



**SERVO SİSTEMLERİN KONTROLÜ İÇİN ENDÜSTRİYEL
AMAÇLI BİR MİKROKONTROLÖR TASARIMI**

Elektrik-Elektronik Müh. Ahmet TOP

**Yüksek Lisans Tezi
Elektrik-Elektronik Mühendisliği
Teknolojileri Anabilim Dalı
Danışman: Prof. Dr. Muammer GÖKBULUT**

EKİM-2016

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SERVO SİSTEMLERİN KONTROLÜ İÇİN ENDÜSTRİYEL AMAÇLI BİR
MİKROKONTROLÖR TASARIMI

YÜKSEK LİSANS TEZİ

Ahmet TOP

132138105

Tezin Enstitüye Verildiği Tarih : 16 Ağustos 2016
Tezin Savunulduğu Tarih : 4 Ekim 2016

Tez Danışmanı : Prof. Dr. Muammer GÖKBULUT (F.Ü)

Diğer Jüri Üyeleri : Doç. Dr. Melih Cevdet İNCE (F.Ü)

Yrd. Doç. Dr. Mehmet ÜSTÜNDAĞ (Bingöl Ü.)

EKİM-2016

ÖNSÖZ

Çalışmalarında, yardım ve katkılarıyla beni yönlendiren danışman hocam Sayın Prof. Dr. Muammer GÖKBULUT'a, yazılımsal ve donanımsal sorunlarımda yardımlarını esirgemeyen sayın Doç.Dr. Ömür AYDOĞMUŞ'a ve tezimin her aşamasında yanımda olan değerli arkadaşım Ozan ÇAKMAK'a teşekkürlerimi sunarım.

Ahmet TOP
ELAZIĞ-2016



İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER	III
ÖZET	VI
ŞEKİLLER LİSTESİ	VIII
TABLOLAR LİSTESİ	XI
KISALTMALAR LİSTESİ	XII
1. GİRİŞ	1
2. MİKROİŞLEMCİLER	4
2.1 Mikroişlemcileri Ayıran Özellikler	5
2.1.1. Kelime Uzunluğu	5
2.1.2. Komut İşleme Hızı	5
2.1.3. Adresleme Kapasitesi	7
2.1.4. Kaydedici Sayısı.....	7
2.1.5. Farklı Adresleme Modları	7
2.1.6. İlave Edilecek Devrelerle Uyumluluk.....	7
2.2. Merkezi İşlemci Biriminde İletişim Yolları	8
2.2.1. Veri Yolu	8
2.2.2. Adres Yolu	9
2.2.3. Kontrol Yolu.....	10
2.3. Mikroişlemci Mimarileri	10
2.3.1. Hafıza Organizasyonu Açısından Mikroişlemci Mimarileri	10
2.3.1.1. Von Neuman Mimarisi	11
2.3.1.2. Harvard Mimarisi	11
2.3.2. Komut İşleme Tekniği Açısından Mikroişlemci Mimarileri.....	12
2.3.2.1. CISC Mimarisi	13
2.3.2.2. RISC Mimarisi	14
2.3.2.3. CISC ve RISC Mimarilerinin Karşılaştırılması	15
2.4. Hafıza Ve Hafıza Çeşitleri	15
2.4.1. Sadece Okunabilir Bellek (Read Only Memory - ROM).....	16
2.4.2. Programlanabilir ROM Bellek (PROM)	16

2.4.3.	Silinebilir Programlanabilir ROM Bellek (EPROM).....	16
2.4.4.	Elektriksel Yolla Deęiřtirilebilir ROM Bellek (EEPROM).....	16
2.4.5.	Rasgele Eriřilebilir Bellekler (RANDOM ACCESS MEMORY - RAM).....	17
2.4.6.	Flash Bellek.....	17
3.	MİKRODENETLEYİCİLER	18
3.1.	Mikrodenetleyicilerin Saęladığı Üstünlükler	18
3.2.	Mikroiřlemci ve Mikrodenetleyici Arasındaki Farklar	18
3.3.	Mikrodenetleyici Seçiminde Dikkat Edilmesi Gereken Ölçütler.....	19
3.4.	Endüstride Kullanılan Mikrodenetleyiciler	20
3.5.	Uygulama İçin Kullanılan Mikrodenetleyiciler	22
3.5.1.	PIC16F877 Mikrodenetleyicisi	22
3.5.1.1.	PIC16F877 Mikrodenetleyicisinin Temel Özellikleri.....	24
3.5.2.	Arduino Due	25
3.5.3.	DSPIC30F3014 Mikrodenetleyicisi	26
4.	MİKRODENETLEYİCİ SEÇİMİ.....	28
4.1	PIC16F877A ile A/D-D/A devresi	28
4.1.1	A/D Modülü	28
4.1.2	D/A Modülü	32
4.1.3	Devrenin Kurulumu.....	32
4.2	Arduino Due ile A/D-D/A devresi	37
4.3	DSPIC30F3014 ile A/D-D/A devresi.....	39
4.3.1	A/D Modülü	39
4.3.2	DAC Modülü.....	40
4.3.3	Devrenin Kurulumu.....	40
5.	KONTROL SİSTEMİ.....	43
5.1.	Sistem	43
5.2.	Otomatik Kontrol Sistemi	43
5.2.1.	Açık Çevrim Kontrol Sistemi.....	44
5.2.2.	Kapalı Çevrim Kontrol Sistemi	44
5.3.	Temel Kontrol Yapıları ve Kontrol Etkileri	45
5.3.1	Açık-Kapalı kontrol (on-off)	45
5.3.2	Oransal (P) Kontrol	47
5.3.3	Integral (I) Kontrol	48

5.3.4	Türev (D) Kontrol	48
5.4.	Otomatik Kontrol Formları	49
5.4.1	Oransal + İntegral (PI) Kontrol	49
5.4.2	Oransal+Türevsel (PD) Kontrol	50
5.4.3	Oransal+İntegral+Türevsel (PID) Kontrol	51
5.5.	Bulanık Mantık (fuzzy) Kontrol.....	51
5.5.1.	Bulanık Sistemin Çalışması.....	53
5.5.2.	Bulanık Sistemin Birimleri.....	53
5.5.2.1.	Bulanıklaştırma Birimi:	53
5.5.2.2.	Kural Tabanı:.....	53
5.5.2.3.	Çıkarım Birimi:	54
5.5.2.4.	Durulaştırma (Berraklaştırma) Birimi:	54
6.	MİKRODENETLEYİCİ KONTROLLÜ ISIL SİSTEM TASARIMI.....	56
6.1.	Mikrodenetleyici ile Sıcaklık Kontrolü.....	56
6.2.	Sistem Mimarisi	56
6.3.	Mimari Bileşenleri.....	57
6.3.1	Algılayıcı	57
6.3.2.	Isıtıcı.....	58
6.3.3	Isıtılan Ortam.....	58
6.3.4.	Mikrodenetleyici	59
6.3.5.	Dimmer Devresi	59
6.3.6.	Zero Cross Devresi	60
6.3.7.	Set Girişi ve Değer Okuma	62
6.4.	ON/OFF, P,PI, PD, PID Denetleyicilerin Sıcaklık Kontrolündeki Etkileri	62
6.4.1.	Genel Akış Şeması	64
6.4.2.	On/Off Denetleyici	65
6.4.3.	Oransal (P) Denetleyici	66
6.4.4.	Oransal+Integral (PI) Denetleyici	69
6.4.5.	Oransal+Türevsel (PD) Denetleyici	70
6.4.6.	Oransal+Integral+Türevsel (PID) Denetleyici	71
7.	BULANIK MANTIK VE PID İLE SERVO SİSTEM KONTROLÜ.....	73
7.1.	Enkoderli DC Motor Kontrolü	74
7.1.1.	Sistemin Mimari Bileşenleri.....	74

7.1.1.1. Motor ve Geri besleme elemanı	74
7.1.1.2 Motor Sürücü ve Mikrokontrolör	76
7.1.1.3 Güç Kaynağı.....	77
7.1.2. PID Denetleyicinin Enkoderli Motor Kontrolündeki Etkisi.....	77
7.1.2.1. Oransal Kontrol (P)	81
7.1.2.2. Oransal+Türevsel Kontrol (PD)	81
7.1.2.3. Oransal+İntegral +Türevsel Kontrol (PID).....	82
7.1.3. Bulanık Mantık Denetleyicinin Enkoderli Motor Kontrolündeki Etkisi.....	83
7.2. Takogeneratörlü DC Motor Kontrolü.....	88
7.2.1 Sistemin Mimari Bileşenleri.....	88
7.2.1.1 Motor ve Geri besleme elemanı	88
7.2.1.2 Motor Sürücü, Güç Kaynağı ve Mikrodenetleyici	89
7.2.2 PID Denetleyicinin Takogeneratörlü Motor Kontrolündeki Etkisi.....	89
7.2.2.1. Oransal Kontrol (P)	93
7.2.2.2. Oransal+Integral Kontrol (PI)	94
7.2.3. Bulanık Mantık Denetleyicinin Takogeneratörlü Motor Kontrolündeki Etkisi	96
8. SONUÇLAR.....	100
KAYNAKLAR.....	104
ÖZGEÇMİŞ.....	109
EK.....	111

ÖZET

Bu tez çalışmasında, endüstride yaygın olarak kullanılan PIC16F877A, dsPIC ve Arduino Due gibi çeşitli mikrodenetleyicilerin, mimari yapı, bellek, örnekleme frekansı ve işlem hızı gibi özellikleri incelenerek endüstriyel kontrol sistemlerinde kullanılabilirliği araştırılmıştır. Mikrodenetleyicilerin örnekleme frekansı ve işlem hızı açısından endüstriyel kontrol sistemleri sınıflandırıldığında zaman sabitesi yüksek olan süreç kontrol ve zaman sabitesi çok küçük olan elektrikli sürücü sistemlerin kontrolü olarak iki kategoriye ayırmak mümkündür. Diğer taraftan, Oransal, İntegral ve Türev (PID) ya da yapay sinir ağları ve bulanık mantık gibi gerçekleştirilecek akıllı kontrol algoritmalarının işlem yükü de mikrodenetleyicinin ve kontrol sisteminin performansını etkileyen önemli faktörlerdir. Dolayısıyla, mikrodenetleyicili endüstriyel kontrol sistemlerinde akıllı kontrol yöntemlerinin uygulanabilirliğini araştırmak bu çalışmanın önemli bir amacını oluşturmaktadır.

Tez çalışmasının ilk aşamasında, Microchip tarafından geliştirilen PIC16F877A ve dsPIC ile Atmel firmasının ürünü olan AT91SAM3X8E mikroişlemcisi ile tasarlanmış olan Arduino Due geliştirme platformunun aynı şartlar altında performansları karşılaştırılmıştır. Mikrodenetleyicilerin analog girişlerinden analog-dijital çevirici (ADC) vasıtasıyla belirli frekanstaki bir sinüsoidal sinyal mikrodenetleyiciye alınmış, mikrodenetleyicide PID kontrol algoritması ve bazı matris işlemleri yapıldıktan sonra alınan sinüsoidal sinyal dijital-analog çevirici (DAC) üzerinden çıkışa yüklenmiştir. Böylece, sinyaldeki gecikme ölçülerek mikrodenetleyicilerin işlem yapma hızları ve dolayısıyla kontrol sistemleri için seçilebilecek en uygun örnekleme frekansları belirlenmiştir. Mikrodenetleyicilerin performanslarını gösteren sonuçlar hem simülasyon hem de uygulama çalışmaları olarak sunulmuştur. PIC16f877A mikrodenetleyicisinin, Micro c PRO programı kullanılarak yazılımı programlanmış ve Proteus 7 Professional'ın ISIS alt programında simülasyonu yapılmıştır. Arduino Due'nin yazılımı için kendi ara yüz programı kullanılarak uygulama sonuçları elde edilmiştir.

Tez çalışmasının ikinci aşamasında, performansı daha iyi olan Arduino Due geliştirme platformu kullanılarak düşük örnekleme frekansının kullanılabileceği bir sıcaklık kontrol sistemi ile çok daha yüksek örnekleme frekansına ihtiyaç duyan bir doğru akım motorunun kapalı çevrim hız kontrolü gerçekleştirilmiştir. Doğru akım motorunun hız bilgisi hem enkoder hem de tako generatörden alınarak geri besleme yapılmıştır. Kontrol yöntemi olarak PID ve bulanık mantık kontrol yöntemleri kullanılmıştır. Kontrol sistemlerinin performansını gösteren deneysel sonuçlar, Arduino IDE programının seri portundan alınarak matlab programında çizdirilmiştir. Çalışılan kontrol sistemlerinin etkinliğini ve bazı sakıncalı yönlerini de gösteren deneysel sonuçlar sunulmuştur.

Anahtar Kelimeler: Mikrodenetleyiciler, Microchip, Arduino, Sıcaklık Kontrolü, DC Motor Kontrolü, PID ve Bulanık Mantık Kontrol.

ABSTRACT

An Industrial Microcontroller Design for Servo Control Systems

In this thesis, various microcontrollers which are widely used in industry such as PIC16F877A, dsPIC and Arduino Due are analyzed and an industrial microcontroller is designed for some processes and servo systems. Sampling frequency and processing speed of microcontrollers are the most important factors affecting the performance of process control having high time constant and electrical drives having low time constant. On the other hand, computing load of the control algorithms which will be implemented in microcontroller, such as Proportional, Integral and Derivative (PID) or intelligent control algorithms such as artificial neural networks and fuzzy logic control are also important factors affecting performance of microcontroller and control system. Therefore, to investigate the feasibility of intelligent control method in the micro-controller based industrial control system is the main objective of this study.

Firstly, the performances of PIC16F877A and dsPIC developed by Microchip and Arduino Due development platform designed by Atmel microprocessor with AT91SAM3X8E microcontroller are compared under same operating conditions. A sinusoidal signal with constant frequency at the analog input of the microcontroller are taken into microcontroller using analog-to-digital converter (ADC) and, the PID control algorithm and some matrix operations are implemented in microcontroller and then the sinusoidal signal are loaded to digital-to-analog converter (DAC). Thus, the processing speed of the microcontroller and the most suitable sampling frequency for control systems are determined by measuring the delay between the input and output signal. Simulation and experimental results showing the performance of microcontrollers are presented. Application software of the PIC16F877 microcontroller are programmed with mikro C PRO program and the simulation is implemented in ISIS subroutine of Proteus 7 Professional. Application results for Arduino Due's are obtained from the prepared program by using the Arduino Due's interface program. Results shows that the performance of Arduino Due is better than the others.

In the second stage of the thesis, by using the Arduino Due platform, closed loop control of a temperature control system which requires low sampling frequency and the speed control of DC motor which requires higher sampling frequency are implemented. Speed information of the DC motor for feedback are obtained both from encoders and tacho generators. PID and fuzzy logic control methods are applied to temperature and DC motor control systems. Experimental results showing the performance of the control systems are evaluated in MATLAB program by getting from the serial port of the Arduino IDE program. Experimental results showing effectiveness and drawbacks of the control system under tests are evaluated.

Keywords: Microcontrollers, Microchip, Arduino, Temperature Control, DC Motor Control, PID and Fuzzy Logic Control.

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 1.1. Endüstride kullanılan bazı süreç kontrol cihazları	3
Şekil 2.1. Bir mikroişlemci sisteminin temel bileşenleri	4
Şekil 2.2. Çeşitli kelime uzunlukları	5
Şekil 2.3. Komut çevrim süresi.	6
Şekil 2.4. Bellek biriminden işlemciye veri yolu kullanarak veri aktarımı.....	8
Şekil 2.5. İşlemcide işlenen verilerin, veri yolu kullanılarak G/Ç birimine iletilmesi.....	9
Şekil 2.6. Verilerin iletileceği bölgenin adres yolu kullanılarak tespiti	9
Şekil 2.7. Kontrol yolu	10
Şekil 2.8. Von Neuman Mimarisi.....	11
Şekil 2.9. Harvard Mimarisi	12
Şekil 2.10. CISC bir komutun çalışma evreleri.....	13
Şekil 2.11. CISC bir komutun mikrokod çevrimi	14
Şekil 2.12. RISC bir komutun çalışma evreleri.....	14
Şekil 3.1. PIC16F877 Harvard Mimarisi	23
Şekil 3.2. Ardışıl Adreslerden Alınan Komutların İşlenmesi	23
Şekil 3.3. PIC16F87X Dış Mimarisi	23
Şekil 3.4. PIC16F87X Dış Görünüm	24
Şekil 3.5. Arduino Due.....	26
Şekil 4.1. A/D modülü blok diyagramı	29
Şekil 4.2. PIC16F877A ADCON0 Kaydedicisi	29
Şekil 4.3. PIC16F877A ADCON1 Kaydedicisi	30
Şekil 4.4. 4 bitlik 2R-R merdiven tipi D/A dönüştürücü.....	32
Şekil 4.5. Devrenin simülasyonda (a) ve breadboard (b) üzerinde kurulumu.....	33
Şekil 4.6. Devrenin simülasyondaki (a) ve dijital osiloskoptaki (b) (c) giriş-çıkış dalga şekilleri	34
Şekil 4.7. Devrenin simülasyon (a) ve analog osiloskoptaki (b) (c) 2.3kHz giriş sinyali sonucu giriş-çıkış dalga şekilleri	35
Şekil 4.8. Devrenin simülasyonda (a) ve dijital osiloskopta (b) PI uygulanmış giriş-çıkış dalga şekilleri.....	36
Şekil 4.9. ADC-DAC devresi deney düzeneği	36
Şekil 4.10. Arduino Analog-Dijital Dönüşüm Blok Diyagramı.....	37
Şekil 4.11. Arduino Due ADC-DAC devre bağlantısı	37
Şekil 4.12. Arduino Due ADC-DAC giriş çıkış dalga şekilleri	38
Şekil 4.13. 2.30kHz giriş sinyali ile giriş-çıkış dalga şekilleri.....	38
Şekil 4.14. Dspic ADC modülü.....	39
Şekil 4.15. Devrenin breadboard üzerinde kurulumu.....	40
Şekil 4.16. 50 Hz'lik giriş-çıkış sinyali	41
Şekil 4.17. Devrenin analog (a) ve dijital osiloskopta (b) 2,3kHz uygulanmış giriş-çıkış dalga şekilleri.....	41

Şekil 5.1. Açık çevrim kontrol sistemi.....	44
Şekil 5.2. Açık çevrim kontrol sistemi.....	44
Şekil 5.3. Açık-Kapalı Sıcaklık Kontrol (İdeal).....	45
Şekil 5.4. İdeal Açık-Kapalı Kontrol Transfer Eğrisi	46
Şekil 5.5. Histerisizli Açık-Kapalı Kontrol Eğrisi	46
Şekil 5.6. Histerisizli Açık-Kapalı Kontrol Transfer Eğrisi.....	46
Şekil 5.7. Oransal+İntegral Kontrol Eğrisi.....	50
Şekil 5.9. Oransal+İntegral+Türevsel Kontrol Eğrisi	51
Şekil 5.10. Bulanık sistemin yapısı	53
Şekil 6.1. Mikrodenetleyicili sıcaklık kontrol genel diyagramı	56
Şekil 6.2. LM35 Sıcaklık sensörü	57
Şekil 6.3. Yükseltici devresi.....	58
Şekil 6.4. Rezistans	58
Şekil 6.5. Isıtılan ortamın iç kısmı	59
Şekil 6.6. Dimmer devresi.....	60
Şekil 6.7. Sıfır geçiş dedektörü devresi.....	60
Şekil 6.8. Zero-Cross Devresi Sinyalleri.....	61
Şekil 6.9. Sıfır Geçiş Dedektörü ve Triyak Sürme Devresi Sinyalleri.....	61
Şekil 6.10. Sistemin genel görünümü.....	62
Şekil 6.11. Isıtıcıya 220 V giriş gerilimi uygulanmasıyla elde edilen çıkış karakteristiği..	63
Şekil 6.12. Isıtıcıya 110 V giriş gerilimi uygulanmasıyla elde edilen çıkış karakteristiği.	63
Şekil 6.13. Sıcaklık kontrol PID akış diyagramı	64
Şekil 6.14. On/off kontrolör,31-60 °C, histeresis yok.....	65
Şekil 6.15. On/off kontrolör,31-60 °C, histeresis:3.....	65
Şekil 6.16. Oransal kontrolör,30-60 °C, Kp:0.2 sıcaklık-zaman grafiği	66
Şekil 6.18. Oransal kontrolör,31-60 °C, Kp:1 sıcaklık-zaman grafiği.....	67
Şekil 6.19. Oransal kontrolör,31-60 °C, Kp:1 PID çıkışı-zaman grafiği	67
Şekil 6.20. Oransal kontrolör,31-60 °C, Kp:2.25 sıcaklık-zaman grafiği	67
Şekil 6.21. Oransal kontrolör,31-60 °C, Kp:2.25 PID çıkışı-zaman grafiği	68
Şekil 6.22. Oransal kontrolör,31-60 °C, Kp:5 sıcaklık-zaman grafiği	68
Şekil 6.23. Oransal kontrolör,31-60 °C, Kp:5 PID çıkışı-zaman grafiği	69
Şekil 6.24. Oransal+Integral kontrolör,23-60 °C Kp:2 Ki:0.02 sıcaklık-zaman grafiği	69
Şekil 6.25. Oransal+Integral kontrolör,30-60 °C Kp:2.25 Ki:2 sıcaklık-zaman grafiği	70
Şekil 6.26. Oransal+Türev kontrolör,32-60 °C, Kp:3 Kd:0.2 sıcaklık-zaman grafiği	70
Şekil 6.27. Oransal+Türev kontrolör,32-60 °C, Kp:3 Kd:4 sıcaklık-zaman grafiği	71
Şekil 6.28. Oransal+Integral+Türevsel kontrolör,30-60 °C, Kp:1.5 Kd:0.02 Ki:0.02 sıcaklık-zaman grafiği	71
Şekil 6.29. Oransal+Integral+Türevsel kontrolör,29-60 °C, Kp:3 Kd:4 Ki:0.01 sıcaklık-zaman grafiği	72
Şekil 7.1. Enkoderli redüktörlü DC motor	75
Şekil 7.2. Enkoder çıkışı	75
Şekil 7.3. Motor miline bağlı olan enkoder	76
Şekil 7.5. Güç Kaynağı	77

Şekil 7.6. PID kontrol sisteminin blok şeması	78
Şekil 7.7. Motorun kontrolsüz devir-zaman çıkış sinyali.....	78
Şekil 7.8. Motor kontrol PID akış diyagramı	80
Şekil 7.9. $K_p=20, K_d=0, K_i=0$ durumunda devir çıkış sinyali.....	81
Şekil 7.10. $K_p=20, K_d=0, K_i=0$ durumunda gerilim çıkış sinyali	81
Şekil 7.11. $K_p=20, K_d=5, K_i=0$ durumunda devir çıkış sinyali.....	81
Şekil 7.12. $K_p=20, K_d=5, K_i=0$ durumunda gerilim çıkış sinyali	82
Şekil 7.13. $K_p=20, K_d=5, K_i=0.1$ durumunda devir çıkış sinyali.....	82
Şekil 7.14. $K_p=20, K_d=5, K_i=0$ durumunda gerilim çıkış sinyali	82
Şekil 7.15. $K_p=15, K_d=5, K_i=2$ durumunda devir çıkış sinyali.....	83
Şekil 7.16. Bulanık mantık blok diyagramı.....	83
Şekil 7.17. giriş üyelik fonksiyonları	84
Şekil 7.18. çıkış üyelik fonksiyonları.....	84
Şekil 7.19. Motor kontrol bulanık mantık akış diyagramı	86
Şekil 7.20. DC motor Bulanık Mantık kontrollü devir-zaman grafiği	87
Şekil 7.21. DC motor Bulanık Mantık kontrollü değişik devirler için devir-zaman grafiği.....	87
Şekil 7.22 Maxon DC motor	88
Şekil 7.23. Motor ve takogeneratör	89
Şekil 7.24 Takogeneratörlü motor PID blok şeması	90
Şekil 7.26. Takogeneratörlü motor PID kontrol akış diyagramı	92
Şekil 7.27. Oransal kontrol devir-zaman grafiği	93
Şekil 7.29. Oransal kontrol hata-zaman grafiği.....	94
Şekil 7.30. Oransal+integral kontrol devir-zaman grafiği.....	94
Şekil 7.31. Oransal+integral kontrol gerilim-zaman grafiği	95
Şekil 7.32. Oransal+integral kontrol hata-zaman grafiği	95
Şekil 7.33. giriş üyelik fonksiyonları	96
Şekil 7.34. çıkış üyelik fonksiyonları.....	96
Şekil 7.35. Takogeneratörlü motor bulanık mantık kontrol akış diyagramı	98
Şekil 7.36. Bulanık Mantık kontrollü devir-zaman grafiği	99

TABLolar LİSTESİ

	<u>Sayfa No</u>
Tablo 2.2. Karşılaştırmalı hız tablosu	6
Tablo 3.1. Günümüzde yaygın olan mikrodenetleyiciler ve üretici firmaları.....	20
Tablo 3.2. PIC16F877 Mikrodenetleyicisinin Temel Özellikleri [25].	24
Tablo 3.3. Arduino Due'nin özellikleri.....	25
Tablo 3.4. DSPIC30F3014 mikrodenetleyicisinin özellikleri.....	27
Tablo 4.1. 0-5V analog sinyalin 10 bitlik ADC için dijital çıkış değerleri.....	31
Tablo 7.1 Kural Tabanı	85
Tablo7.2. Maxon DC motor verileri	88
Tablo 7.3. Kural Tabanı	97



KISALTMALAR LİSTESİ

RAM	: Random Access Memory (Rastgele Erişilebilir Bellek)
ROM	: Read Only Memory (Sadece Okunabilir Bellek)
CPU	: Central Process Unit (Merkezi İşlem Birimi)
EPROM	: Erasable Programmable Read Only Memory (Silinebilir Programlanabilir ROM bellek)
EEPROM	: Electronically Erasable Programmable Read Only Memory (Elektriksel Yolla Değiştirilebilir ROM Bellek)
CISC	: Complex Instruction Set Computer (Karmaşık Komut Seti Kullanan Bilgisayarlar)
RISC	: Reduced Instruction Set Computer (Azaltılmış Komut Seti Kullanan Bilgisayarlar)
MİB	: Merkezi İşlem Birimi
ALU	: Aritmetik Mantıksal Birim
OTEPROM	: One Time Programmable ROM (Bir kez programlanabilir ROM)
I/O	: Input/Output (Giriş/Çıkış)
ARM	: Acorn RISC Machine
PIC	: Peripheral Interface Controller (Çevresel Arabirim Denetleyicisi)
DAC	: Digital to Analog Converter (Dijital-Analog Dönüştürücü)
ADC	: Analog to Digital Converter (Analog-Dijital Dönüştürücü)
LSB	: Least Significant Bit (en değersiz bit)
MSB	: Most Significant Bit (en ağırlıklı bit)
OP-AMP	: Operational Amplifiers (İşlemsel Kuvvetlendirici)
P	: Oransal (proportional)
PI	: Oransal+Integral (proportional+integral)
PD	: Oransal+Türev (proportional+derivative)
PID	: Oransal+Integral+Türev (proportional+integral+derivative)
PWM	: Darbe Genişlik Modülasyonu (Pulse Width Modulation)

1.GİRİŞ

Günümüz teknolojisinde endüstriyel uygulamalarda otomatik kontrolün hızla yayılması ve bu kontrol mekanizmalarının hızlı ve güvenilir bir şekilde kontrol edilme gereksinimi mikrodenetleyicilerin yaygın bir şekilde kullanılmasını sağlamıştır. Kullanılan yere ve kullanım amacına göre mikroişlemci veya mikrodenetleyiciler kullanılabilir. Bu mikrokontrolörler ile düşük frekanslı sistemlerin (sıcaklık, nem, basınç vb.) kontrolü yaygın biçimde gerçekleştirilmektedir. Ancak, servo sistemler gibi yüksek örnekleme frekansına ihtiyaç duyan sistemlere, ileri kontrol yöntemlerinin (bulanık kontrol, yapay sinir ağları ile kontrol vs.) uygulanabilmesi için gelişmiş mikro işlemcili düzeneklere ihtiyaç duyulmaktadır.

Sıcaklık kontrolü, endüstride ve günlük yaşamda sıkça karşılaşılan bir parametre kontrolüdür. Termostatlarda yer alan, sıcaklığın belli değerinde sistemin ısısının artırılması ya da durdurulması işlemi gibi basit uygulamaların yanında çeşitli kimyasal süreçlerde, ortam sıcaklığının oldukça hassas biçimde kontrol edilmesi gereken endüstri uygulamalarında ve medikal odalarda küvöz vb. ortamların izlenmesinde bu kontrolün verimli biçimde gerçekleştirilebilmesi esastır. Sıcaklık kontrolünde amaçlanan, temelde ortam sıcaklığını belli bir değerde sabit tutmak olabileceği gibi tasarlanan sistemin ya da tesisin konumuna ve kullanım amacına göre sıcaklığı bir değer aralığında sabit tutma hedefi de taşıyabilir. Her durumda istenilen bir değer, ortamdaki elde edilen veri sonucunda hesaplanan bir fark değeri ve bu fark değeri işleyecek verimli bir denetleyici mevcut olacaktır[1,2]. Denetleyicinin karşılaştırma algoritması ve tekniği, elektronik bileşenlerin gelişmesiyle birlikte çeşitli alternatifler değerlendirilebilecek noktaya ulaşmıştır[3]. Bu alandaki ilk çalışmalarda karşılaştırıcı olarak işlemsel yükselteçler kullanılsa da günümüzde mikrodenetleyicilerin gelişmesi, fiyatlarının ucuzlaması ve diğer çevre birimleriyle etkileşimlerinin genişlemesi nedeniyle bu entegre devrelerin dahili yeteneği olan darbe genlikli modülasyon (PWM) tekniğinin, karşılaştırıcı amaçlı kullanılması avantaj olacaktır[4]. Bu yetenek, DC motor hız kontrolü ve güç elektroniklerinde DC-DC dönüştürücü gibi pek çok uygulamada da kullanım alanı bulmaktadır[5].

Endüstriyel kontrol alanındaki teknolojik gelişmeler ve otomatik kontrolün hızla artan konumu, hızı sabit olan bir motordan daha fazlasının gerekli olduğu pozisyon ayarı, yüksek kararlı sistemler, periyodik çalışma ve dinamik yük gibi durumlarda servo sistemleri vazgeçilmez bir tercih haline getirmiştir. Endüstriyel otomasyonda; hız, moment, pozisyonlama, ölçme ve bilgisayar destekli üretim gibi alanlarda sıklıkla servo sistemler kullanılmaktadır [6].

Servo sistemler ile yapılan bir uygulamada, yüksek düzeyde duyarlılık ve kontrol elde edilebilmektedir. Ayrıca motorun hareketlenme ve duruş anında, motorun kontrolünü daha hassas bir şekilde yapmak mümkündür. Bu şekilde motorun kalkış ve duruş sırasında, miline bağlı olan yükün zarar görmemesi sağlanılabileceği gibi aynı zamanda motorun kalkış anında aşırı akım çekmesi de engellenmiş olur [7].

Sürücülerde kullanılan yarı iletken elemanlar motorun konum ve hız bilgisine göre uygun sırada iletme veya kesime geçirilerek motorun istenilen kontrolü yapılır. İletim ve kesime girme sürelerini kontrol edebilmek için mikrodenetleyici kullanılır [8]. Bu kontrolörler, kontrol edilecek sisteme göre seçilebilir. Gelişmişlik düzeylerine göre hassasiyetleri de değişir. Çeşitli firmalar tarafından farklı isimlerde üretilirler. Ayrıca bu mikro kontrol elemanlarından gelişmiş olanlarına P,PI, PID, fuzzy... vb. gibi algoritmalar eklenerek de üretilecek olan PWM (Darbe Genişlik Modülasyonu) ile sürücü içerisindeki yarı iletken elemanların kontrolü yapılabilir [9].

Motorlarla oluşturulan bir sistemde çıkış büyüklüğü, iki şekilde olabilir. Bunlar; motor milinden alınan mekanik güç bilgisi veya generatör çıkış uçlarından alınan elektriksel güç olup bu büyüklükler, istenilen fiziksel büyüklüklere dönüştürülerek istenilen kontrol gerçekleştirilebilir[10]. Motor dışındaki sistemler için de bu yöntem geçerlidir. Bu sistemlerin kontrolünü sağlayabilmek için endüstride kullanılan çeşitli süreç kontrol cihazları bulunmaktadır. Bu cihazlardan bazılarının kontrol opsiyonları hakkında bilgiler aşağıda verilmiştir [11-12].



Şekil 1.1. Endüstride kullanılan bazı süreç kontrol cihazları

Bu kontrol cihazlarının özelliklerinden de anlaşılacağı üzere denetleme, kontrol etme, çalıştırma süreleri yüksek olduğundan dolayı anlık kontrol gerektiren sistemlerin kontrolü için kullanımları uygun değildir. Genellikle sıcaklık seviyesi, nem oranı, basınç düzeyi vb. gibi fazla hassasiyet istemeyen sistem kontrolleri için tercih edilmektedirler. Şekil 1.1 'de görülen ve endüstride kullanılan süreç kontrol cihazlarında hassasiyet istemeyen sistemler için kullanıldığında hız ve işlem kabiliyeti gerekmediği için basit mikrodenetleyicilerle kontrol edilirken, daha hassas sistemler için gelişmiş mikrodenetleyiciler kullanılmaktadır.

Bu tez çalışmasındaki amaç, farklı birkaç mikrodenetleyicinin kapalı çevrim kontrol performansını (PID algoritmalarını gerçekleştirme, işlem yoğunluğu ve bir döngüde yapılması gereken işlemlere bağlı olarak minimum örnekleme periyodu vs.) incelemek. Gelişmiş bir mikrodenetleyici kartı belirlenerek endüstriyel standartlarda (Şekil 1.1 de görüntüleri verildiği gibi 0-10 volt ve 0-20 mA giriş-çıkış, kontrol algoritmalarının ve parametrelerinin seçimi, vs.) bir mikrodenetleyici cihazı tasarlamaktır. Tasarlanan mikrodenetleyici cihazının, düşük örnekleme frekansına ihtiyaç duyan süreçleri ve yüksek örnekleme frekansına ihtiyaç duyan servo sistemleri kontrol performansını değerlendirmektir.

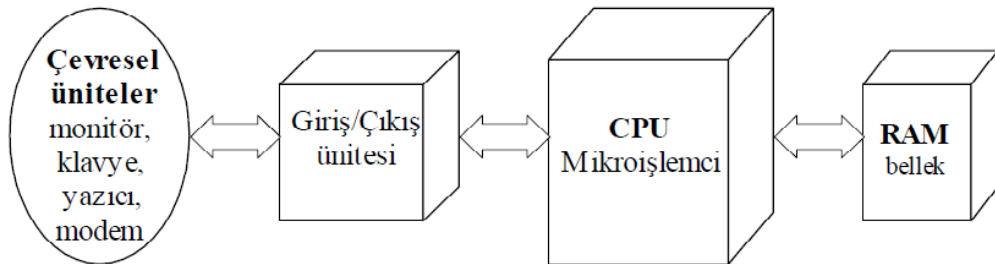
2. MİKROİŞLEMCİLER

Bir sisteminde meydana getirilen programların işletilmesi ve yürütülmesi görevini yürüten ve tüm devreyi merkezi bir şekilde kontrol eden bütünleşmiş devreler mikroişlemci olarak adlandırılır [13]. Tarihteki ilk bütünleşmiş devreler, 1971 yılında hesap makinesi olarak kullanılmak üzere üretilen Intel firmasının 4004 adlı mikroişlemcisidir. Bir seferde işleyebileceği verinin 4 bit olması, bu işlemciye 4 bitlik işlemci denilmesinin ana sebebidir. 1974 ve 1976 yılları arasında bakıldığında işleyebileceği veriler 8 bitlik olan ilk genel amaçlı mikroişlemciler tasarlanmıştır [14]. Mikroişlemcilerin kullanılabilmesi için hafıza, giriş/çıkış üniteleri ve çevresel üniteler gibi bazı temel bileşenlere ihtiyaç duyulur.

Giriş / Çıkış (Input / Output) : Mikroişlemcinin dış dünya ile iletişime geçmesini ve bilgi alıp vermesini sağlayan, analog ve dijital verilerden oluşabilen en temel bileşenlerden biridir.

CPU (Central Processing Unit – Merkezi İşlem Birimi) : 4, 8, 16, 32 ve 64 bitlik sözcük uzunluklarında çalışarak verilerin koordinasyonunu sağlar, komutları idare eder ve istenilen tüm hesapları yapar.

Hafıza: Hafıza olarak RAM, ROM, PROM, EPROM, EEPROM kullanılabileceği gibi bunlardan herhangi birinin birleşimi olabilir. Hafıza birimi, program ve veri depolamak için kullanılır [15].

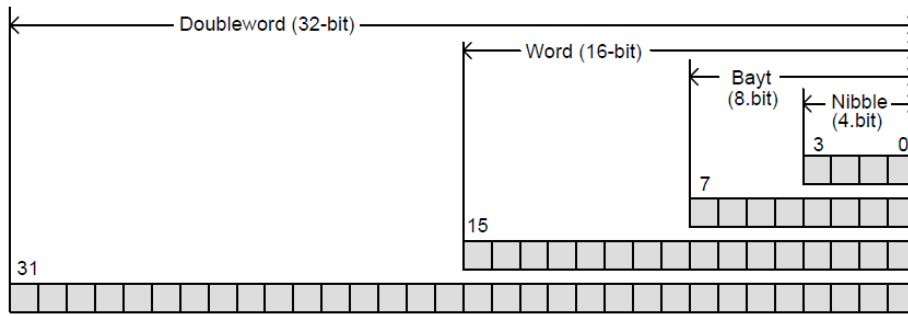


Şekil 2.1. Bir mikroişlemci sisteminin temel bileşenleri

2.1 Mikroişlemcileri Ayıran Özellikler

2.1.1. Kelime Uzunluğu

Osilatörden gelen her bir darbeye göre işlemcinin komutları işleyebileceği bit adetine kelime uzunluğu adı verilir. İşlemciler bu sürede komutları işler veya bellekte bulunan verilerle işlem yapar. İşlenen bu veriler işlemcinin niteliğine göre 4, 8, 16, 32 ve 64-bit olabilir. Kelime uzunluğu ile veri yolu uzunluğu birbirine eşit ifadelerdir. İşlemcilerin sınıflandırılması, dört, sekiz, on altı, otuz iki ve altmış dört bit şeklindedir[16].

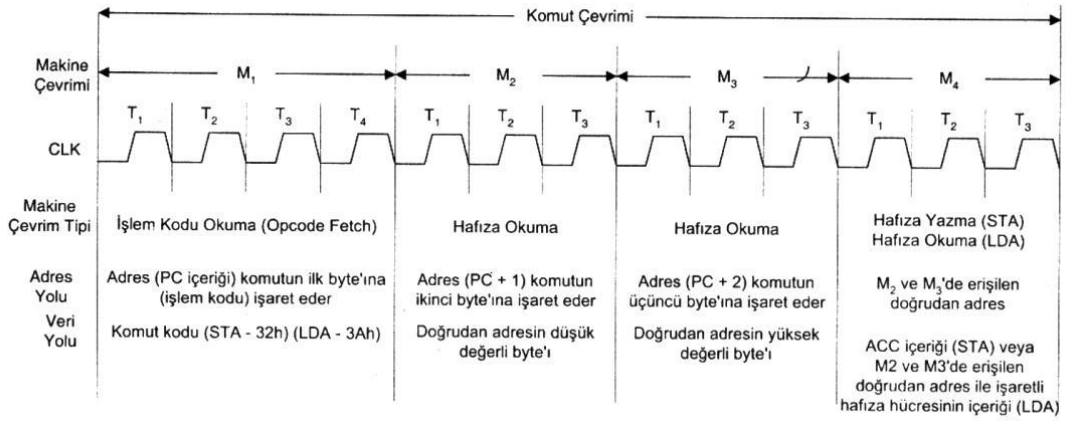


Şekil 2.2. Çeşitli kelime uzunlukları

16-bitlik bir işlemci içerisinde, 16-bit kelime uzunluğuna sahip iki sayının aritmetiksel işlemi, tek adımla yapılırken 8-bitlik işlemcide bu işlem birden fazla adımla yapılmaktadır.

2.1.2. Komut İşleme Hızı

Merkezi İşlemci Birimi (Central Processing Unit-CPU) osilatörden gelen her darbe sinyalinde sonraki işlem adımına geçer. İşlemcinin hızını irdelerken osilatör frekansına ve her komutu kaç çevrimde gerçekleştirdiğine bakmak gerekir. Osilatör frekansı mikroişlemcinin içerisinde bulunduğu gibi dışarıdan harici olarak da verilebilir. Komut çevrim süresi ise işlenecek olan bir komutun işlevini gerçekleştirebilmesi için geçen süredir [16].



Şekil 2.3. Komut çevrim süresi.

Her bir mikroişlemcinin komut çevrim süresi farklı sayıda clock çevrimleri ile ifade edilmektedir. Tablo 2.2'de bir işlemcinin komut işleme süreleri birbirine eşit olan komutlara bakarak karşılaştırılma yapılmıştır [16].

Tablo 2.2. Karşılaştırmalı hız tablosu

Harici Clock Girişi MHz	Mikroişlemci	Komut	Saat Periyodu	Süre ms
2	6502A	ADC	2	1
2	68B00	ADCA	2	1
5	8085A	ACI	7	1,4
4	Z80A	ADC	7	1,75

Yukarıdaki tabloda görüldüğü üzere sadece osilatör frekansına bakarak değerlendirme yapmak yanlış sonuçlar verebilmektedir. Clock frekansı genellikle gerçek işlem yapma frekansını göstermese de bir mikroişlemcinin hızını direkt olarak etkilemektedir. Bir mikroişlemcinin hızını artıran temel unsurlar aşağıdaki gibi sıralanabilir:

- ✓ Merkezi işlem birimi tasarım biçimi
- ✓ Kelime uzunluğu
- ✓ İşlemci komut kümesi çeşidi
- ✓ Bilgisayar belleğine ve giriş/çıkış aygıtlarına erişim hızı [16]

2.1.3. Adresleme Kapasitesi

Bir işlemci için adresleme kapasitesi, adresleme yapabileceği veya direkt olarak erişebileceği belleğin büyüklüğüdür. Bu büyüklük işlemcinin adres hattı adetine bağlıdır. 16-bit adres hattına sahip bir mikroişlemcinin adres büyüklüğü 216 ile 65536 arasında olmaktadır. Bu değer yaklaşık olarak 64 KB ile ifade edilir [16].

2.1.4. Kaydedici Sayısı

Kaydedicileri, genel amaçlı ve özel amaçlı kaydediciler olarak iki başlık altında toplamak mümkündür. Bütün mikroişlemcilerde bu kaydedicilere dâhil edebileceğimiz değişik görevlere sahip, değişik nitelikte ve miktarda kaydediciler vardır. Bu kaydediciler 8, 16, 32 ve 64-bit olabilirler. Kaydedici sayıları programı daha sade ve anlaşılır yapmalarının yanısıra programcılarını işini de kolaylaştırırlar. Herhangi bir mikroişlemciyi programlamaya başlamadan önce kullanılacak olan kaydedicilerin işlevlerinin ve isimlerinin programcı tarafından öğrenilmesi gerekir [16].

2.1.5. Farklı Adresleme Modları

Bir komut işlenirken gerekli veriler bir bellekten alınabilir, bir belleğe aktarılabilir veya bellek ile kaydedici arasında ya da iki kaydedici arasında alışverişi sağlanabilir. Bunun için farklı erişim yöntemleri kullanılır. İşlenecek olan bilgiye mikroişlemcinin farklı şekilde erişmesi, ‘adresleme yöntemleri’ olarak ifade edilir. Farklı adresleme modlarının olması programcıya esneklik avantajı sağlar [16].

2.1.6. İlave Edilecek Devrelerle Uyumluluk

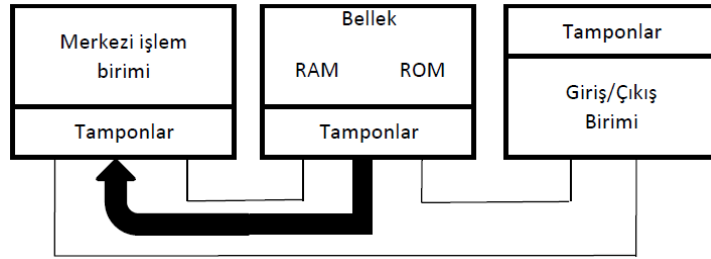
Mikroişlemcili sisteme eklenecek olan devrelerin, bellek entegrasyonlarının ve giriş çıkış birimlerinin performansları ve hızları en azından mikroişlemciyle aynı veya yüksek olması gerekir. Eğer bu birimlerin hızları mikroişlemci hızından düşük ise işlemcinin hızı diğer elemanlardaki yavaşlıktan ötürü düşer [16].

2.2.Merkezi İşlemci Biriminde İletişim Yolları

Mikroişlemcide işlenecek olan komutları taşıyan hatlarla birlikte, işlenecek olan verileri taşıyan hatlar ve kesmeleri kontrol eden sinyalleri taşıyan hatlar vardır. İşlenecek verileri işlemciye almak veya işlenen verileri çevre birimlere yollamak için aynı hatlardan faydalanılır. Tüm bu yollara iletişim yolları adı verilir. Bir mikroişlemciye iletişim yollarını veri, adres ve kontrol olmak üzere üç gruba ayırmak mümkündür [16].

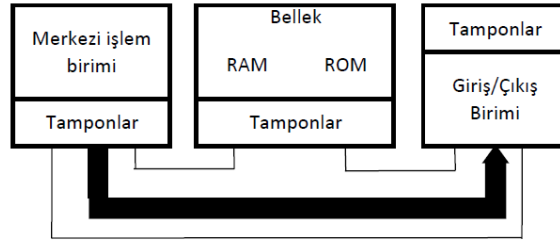
2.2.1.Verii Yolu

Hafıza ve giriş/çıkış birimlerinden CPU'ya veya CPU'dan bu birimlere veri iletmeye kullanılan yollar, veri yolu olarak tanımlanır. Veri yolunun genişliği, mikroişlemci kaydedici genişliği ve kullanılan kelime uzunluğuyla direkt olarak bağlantılıdır. 16-bitlik mikroişlemcilerde veri yolu 16 hattı içerirken 8-bitlik işlemcilerde 8 hattı içerir. Mikroişlemcide işlenen veriler veri yolu üzerinden ilgili birimlere aktarıldığı veya mikroişlemciye işlenmek üzere iletilen veriler veri yolu üzerinden iletildiği için veri yolunda çift yönlü iletişim mümkün olmaktadır [16].



Şekil 2.4. Bellek biriminden işlemciye veri yolu kullanarak veri aktarımı

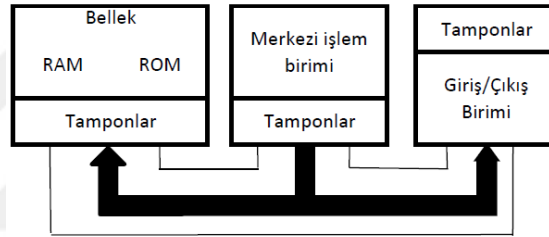
Bellekte bulunan ve CPU tarafından işlenmesi istenilen veriler, veri yolu üzerinden iletilir (Şekil 2.4). Bellekteki çift yönlü aktarımında verileri kısa süre tutmak amacıyla kullanılan kaydediciler yani tamponlardan faydalanılır. Ara birim olarak ise giriş/çıkış birimi kullanılır. CPU ile G/Ç birimi arasında veri iletiminde veri yolundan faydalanılır. [16].



Şekil 2.5. İşlemcide işlenen verilerin, veri yolu kullanılarak G/Ç birimine iletilmesi

2.2.2. Adres Yolu

Verinin gönderileceği veya okunacağı adres bölgesini temsil eden bilgilerin taşınmasında kullanılan hatlar, adres yolu olarak isimlendirilir. Adres yolu, veri yolunun aksine tek yönlüdür ve paralel iletişim sağlayacak şekildedir.



Şekil 2.6. Verilerin iletileceği bölgenin adres yolu kullanılarak tespiti

Merkezi işlem biriminde işlenen dataların bellekte muhafaza edilmesi veya diğer elemanlara yollanabilmesi durumunda, verinin tutulacağı veya yollanacağı yerin adresi, mikroişlemci içerisindeki program sayıcı yardımıyla adres yolu üzerine yerleştirilir. Yerleştirilen bu bilginin işaret ettiği adres dâhili veya harici bellekte olabilir. Adresin bulunabilmesi için bu bilginin kodu deşifre edilir ve bulunan adres bölgesindeki veri, veri yoluna aktarılır. Yapılan bu işlemleri kontrollü bir şekilde gerçekleştirmek için zamanlama ve kontrol birimleri faaliyet gösterir[16].

Mikroişlemcide adres ve veri yolları farklı sayıda hattı içerebilir. Adres yolu 16 bitlik olan bir mikroişlemcide veri yolu 8 bitlik olabilir. Günümüz mikroişlemcilerinde devamlı olarak veri yolları artırılırken, adres yolları hattı büyük oranda korunmaktadır. Adres yollarının artmamasının nedeni şu anda kullanılan adresleme kapasitesinin yüksek ve ilerideki ihtiyaçları karşılayabilecek düzeyde olmasıdır [16].

2.2.3. Kontrol Yolu

Mikroişlemcili ile çevre birimleri arasındaki alış verişi gerçekleştiren sinyallerin iletilmesi amacıyla kullanılan yollar kontrol yolu olarak isimlendirilir. Her mikroişlemcide farklı sayıda hattı içeren kontrol yolu bulunabilir, bunun sebebi her bir mikroişlemciye ait komut kümesi ve belirli amaçlar için kullanılan sinyallerin farklı olmasıdır. Kontrol yolunda bulunan sinyaller üç farklı işlemi gerçekleştirmek amacıyla kullanılır:

Sinyal seçimi: Sistem içerisinde kullanılacak olan sinyallerin ve sinyallerin uygulanacağı yerin belirlenmesi sağlayan sinyaller

Yön belirleme: Sistemdeki verinin ne yöne gideceğini belirleyen sinyaller

Zamanlama: Yapılacak işlemlerin zamanlamasını ve sırasını belirleyen sinyaller [16].



Şekil 2.7. Kontrol yolu

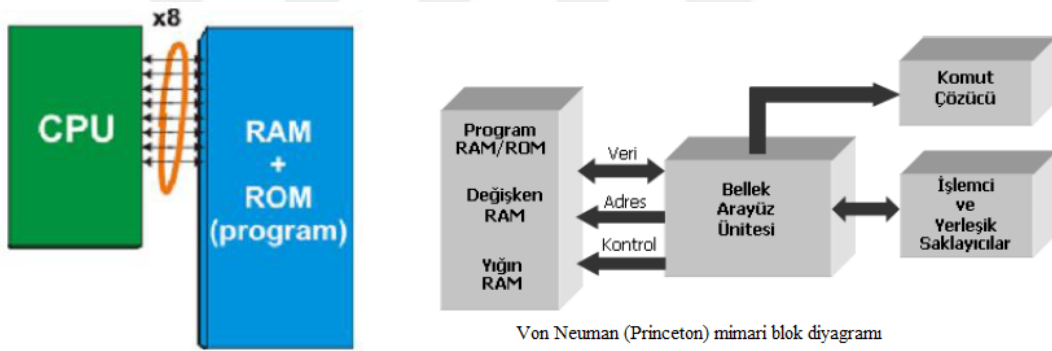
2.3. Mikroişlemci Mimarileri

2.3.1. Hafıza Organizasyonu Açısından Mikroişlemci Mimarileri

Bellek kullanımı bakımından mikroişlemci mimarileri Von Neuman ve Harvard olmak üzere iki gruba ayrılırlar. Önceki yıllarda teknolojiye uygun olarak Von Neuman mimarisi tercih edilirken ilerleyen yıllarda teknolojinin de gelişmesi ile Harvard mimarisi 70'li yıllarda özellikle mikrodenetleyici tasarımında çok kullanılır olmuştur. Her iki mimarinin üstün olan özelliklerini alarak performans artışı elde etmek için iki mimarinin özelliklerini içeren işlemciler de mevcuttur [17].

2.3.1.1. Von Neuman Mimarisi

Bu mimarinin en temel özelliği, bellek birimini işlem biriminden ayırmasıdır. Veri ve komutun aynı bellekte kullanıldığı bu mimaride, hem komut hem de verinin iletilmesinin gerektiği durumlarda veri iletişim sistemlerinin, komut iletişim işlemlerini beklemesi gerekmektedir. Bu mimariyi kullanan mikroişlemcilerde iletimin hızı mikroişlemcinin hızına yetişemediği için darboğaz olarak adlandırılan olay gerçekleşebilir. Bu sakınca ile birlikte meydana gelebilecek diğer bir sakıncalı durum ise; veri ve komutların aynı bellek içerisinde bulunması nedeni ile bu iki bilginin birbirini ile karışma ihtimalinin bulunmasıdır [18].

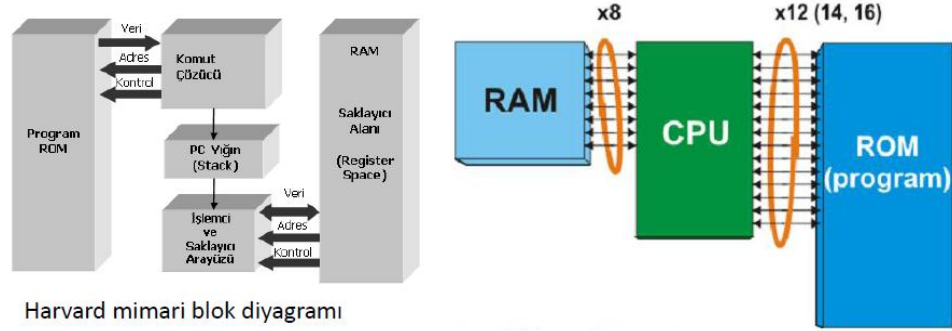


Şekil 2.8. Von Neuman Mimarisi

Belirtilen iki sakıncayı ortadan kaldırmak ve bu mimariyi kullanan sistemlerin performansını arttırmak için önbellek sistemi geliştirilmiştir. Ana bellekten alınan komut ve veriler ayrı önbelleklere yerleştirilerek darboğazın ortadan kaldırılması ile birlikte verilerin ayrıştırılması da sağlanmaktadır [18].

2.3.1.2. Harvard Mimarisi

Von Neuman mimarisindeki dezavantajları ortadan kaldırmak için komutlar ve veri ile ilgili bilgilerin farklı belleklerde saklandığı 'Harvard Mimarisi' tasarlanmıştır. Bu mimari ile işleyen sistemlerde, veri ve komutları ileten yollar birbirinden ayrılmıştır. Yani bir taraftan komut işlenirken aynı zamanda bir veriye ihtiyaç duyulduğunda iki verinin de birbirine karışmadan iletilmesi mümkün olacaktır. [18].



Şekil 2.9. Harvard Mimarisi

2.3.2. Komut İşleme Tekniği Açısından Mikroişlemci Mimarileri

Farklı yapıda komut setlerinin mikroişlemcilerde kullanımı ile iki farklı yapıda mikroişlemci/mikrodenetleyici mimarisi ortaya çıkmıştır:

- Complex Instruction Set Computer –CISC (Karmaşık Komut Seti Kullanan Bilgisayarlar)
- Reduced Instruction Set Computer –RISC (Azaltılmış Komut Seti Kullanan Bilgisayarlar)

CISC işlemciler, 1960'larda geliştirilmiş ve ilk mimari yapı olma özelliğini taşımaktadır. Bu mimarinin, Karmaşık komut seti ve sınırlı bellek kapasitesinin işlemlerin belirli bir hızda yapılması gereksinimini karşılayamaması nedeni ile RISC işlemci mimarisinin oluşmasına neden oldu. İşlemci tarafından kullanılan komutların basitleştirilmiş olması ve komutların azaltılması, komut işleminin hızını artırmıştır [18].

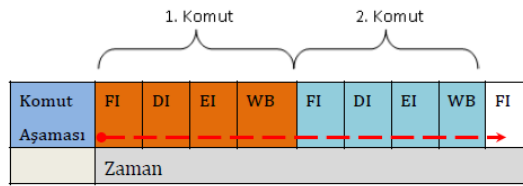
2.3.2.1. CISC Mimarisi

Bu mimari; aktif bir şekilde bellek kullanımı ve kolay programlanabilme sağlamak amacıyla geliştirilmiş olup Tasarımı çok karmaşık yapmasına karşın yazılımı basitleştirmektedir [19]. CISC mimaride donanımın daha hızlı olduğu düşünüldüğü için karmaşık kodları çözmek için donanıma yoğunluk verilmiştir. Bu nedenle maliyeti arttıran ve fazla güç tüketen bir sistemdir [18].

CISC tasarım felsefesinde aynı anda yalnızca bir komut üzerinde işlem yapılmaktadır. Buna 'kademeli komut işleme' tekniği denilmektedir. Kademeli komut işleme tekniğinde oluşacak işlem sırası aşağıdaki gibidir;

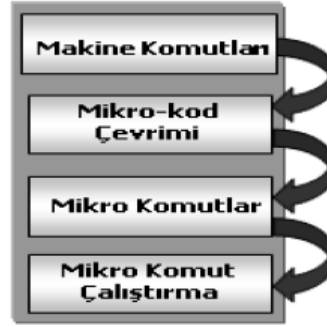
1. Bellekten komutu al getir (FI-Fetch Instruction)
2. Getirilen komutun kodunun çözülmesi (DI-Decode Instruction)
3. Komutun ALU'da çalıştırılması (EI-Execute Instruction)
4. Sonucun ilgili kaydediciye yüklenmesi (WB- Write back Result)

Yukarıdaki adımların bir tanesi bitmeden diğerine geçilemeyeceği için bir komutun işlenmesi bitmeden diğeri işlenemez.



Şekil 2.10. CISC bir komutun çalışma evreleri

CISC mimarisinde fazlaca sayıda bulunan komutları işleyebilmek için mikro-kod kullanılmaktadır. Farklı uzunlukta olan bu komutların çözümünde kod çözücülere ihtiyaç vardır [18].



Şekil 2.11. CISC bir komutun mikrokod çevrimi

2.3.2.2. RISC Mimarisi

RISC mimarisini oluşturan şirketlerden ilki IBM firmasıdır. Bu mimarinin gelmesi ile CISC'in avantajları geçersiz hale geldi. Çünkü bellek hızı artırılmış ve assembly yerine yüksek seviyeli diller kullanılmaya başlanmıştır [15]. RISC mimarisine göre:

- Bütün komutlar tek bir çevrimde çalıştırılmalıdır:
- Belleğe sadece “load” ve “store” komutlarıyla erişilmelidir.
- Bütün icra birimleri mikrokod kullanmadan donanımdan çalıştırılmalıdır. [15].

RISC işlemlerde CISC'in aksine aynı anda birden fazla komut işlenmesini sağlayan teknoloji mevcuttur. Bu işlem Kanal Komut İşleme Tekniği (pipelining) olarak isimlendirilmiştir. Bu teknolojinin kullanımı, daha fazla bellek ve daha gelişmiş derleme tekniği gereksinimlerini ortaya çıkarır. Bu gereksinimleri tam anlamıyla karşılayabilen çok geniş ölçekli entegre (VLSI) teknolojisinin 1990'larda geliştirilmesi ile RISC işlemciler büyük sistemlerin yanı sıra PC'lerde kullanılmaya başlandı.

Komut evreleri							
FI	DI	EI	WB	FI	DI	EI	WB
	FI	DI	EI	WB	FI	DI	EI
		FI	DI	EI	WB	FI	DI
			FI	DI	EI	WB	FI
				FI	DI	EI	WB
Tamamlanan Komut adedi			1	2	3	4	5

Şekil 2.12. RISC bir komutun çalışma evreleri

2.3.2.3. CISC ve RISC Mimarilerinin Karşılaştırılması

RISC ve CISC mimarileri, hızlarına, kullanılan transistör sayılarına, komut işleme tekniklerine vb. bakımından karşılaştırılabilirler. İki mimari arasındaki farklar şöyle sıralanabilir:

1- Hız: İki işlemci mimarisi arasındaki hız farkı, kullanılan komut işleme tekniklerinden kaynaklanmaktadır. CISC işlemciler, RISC işlemcilere göre daha yavaştır [18].

2- Komut işleme biçimi: RISC işlemcilerde ‘kanal komut işleme tekniği’ (pipeline) kullanılırken, CISC işlemcilerde ‘kademeli komut işleme’ tekniği kullanılır. CISC tekniğinde bir komutun işlenmesi bitmeden diğer komut işlenemez yani aynı anda tek bir komut işlenebilir. RISC tekniğinde ise, bunun tam tersidir [18].

3- Transistör sayısı: CISC işlemciler donanıma ağırlık verdiği için kullanılan transistör sayısı daha fazladır. Bundan kaynaklı olarak sistem için daha geniş bir alana gereksinimi vardır ve daha fazla ısı ortaya çıkarır, bunun sonucu olarak soğutma ihtiyacı ortaya çıkar [18].

4- Donanımsal yapı (Tasarım şekli): RISC işlemciler, CISC işlemcilere göre daha basit yapıdadırlar [18].

5- Komut yapısı: RISC mimaride aynı uzunlukta basit komutlar kullanılır [18].

2.4. Hafıza Ve Hafıza Çeşitleri

Hafıza birimleri, programın ve programın üzerinde çalışacağı verilerin saklanması için kullanılır. Önceleri mekanik çarklar ve delikli kartlar sonrasında röleler daha sonraları ise çekirdek hafızalar kullanılmıştır. Yakın zamanda ise kullanılan hafızalar yarı iletken teknolojisine dayanmaktadır [20].

Yarı iletken hafızalar belleğe yazılmış olan verileri hafızasında tutabilmek için enerji gerektirdikleri için enerji kesilmesi durumunda bilgiler yok olur. Bu sorunu bertaraf etmek için zamanla çeşitli yöntemler geliştirilmiştir. Yarı iletken hafızalar bilgileri depolamak

açısından Sadece Okunabilir Bellek ve Okunabilir/Yazılabilir Bellek olmak üzere ikiye ayrılırlar [20].

2.4.1. Sadece Okunabilir Bellek (Read Only Memory - ROM)

“Factory Mask ROM” olarak da adlandırılan bu belleklerin içine bilgi yazılması firma tarafından gerçekleştirilir. ROM’ların sadece üretim sırasında programlanabilmeleri ve içindeki programların değiştirilememesi, önemli kısıtlama olarak görülebilir [20].

2.4.2. Programlanabilir ROM Bellek (PROM)

Üretildikleri an bütün hücreleri lojik 0 veya 1 yüklü hafızalardır. Her hücre içinde bir sigorta bulunmaktadır. Sigortalar özel bir yöntem ve aygıt aracılığı ile arttırılabilir. Bir hücrenin sigortasının atmış olması, o gözün lojik konumunun değişmiş olması demektir. OTPROM (One Time Programmable ROM – Bir Defa Programlanabilen Salt Okunur Hafıza) olarak da adlandırılır [20]. Bu tip hafızada da tekrar programlanabilme özelliği yoktur. Sadece bir defa programlanabilir.

2.4.3. Silinebilir Programlanabilir ROM Bellek (EPROM)

“EPROM”lar bellek hücrelerine elektriksel sinyal uygulanarak programlanır. İçine yazılan programın silinmemesi için cam pencereli kısım ışık geçirmeyen bantla örtülmelidir. Bu bellekteki bilgileri silip yeniden yazma yapabilmek için bandı kaldırıp ultraviyole altında belirli bir süre tutmak gerekir ve daha sonra tekrar programlanabilir [16].

2.4.4. Elektriksel Yolla Değiştirilebilir ROM Bellek (EEPROM)

Elektriksel olarak silinebilir ve programlanabilir belleklerdir. Bu hafıza birimlerinde, hafıza hücrelerine istenen bir değer kaydedilebilir ve kaydedilen veri yeni bir yazma işlemine kadar hafızada kalır. Gerçekte EEPROM silinmesi EPROM’un silinmesi ile aynı anlama gelmektedir [20].

2.4.5. Rasgele Erişilebilir Bellekler (RANDOM ACCESS MEMORY - RAM)

Mikroişlemcinin çalışması geçici işlemlerin yapıldığı birimi RAM belleklerdir. Özel bir sıra takip etmeden herhangi bir adrese erişildiği için bu şekilde isimlendirilir. RAM tipi entegreler okuma ve yazmada kullanıldıklarından merkezi işlem birimi, bu entegreleri kontrol ederken okuma ve yazma sinyalleri göndermesi gerekir. Ayrıca entegrenin istendiği zaman aktif duruma geçmesini sağlayacak entegre seçimi (CS =Chip Select) pini bulunmaktadır [21].

2.4.6. Flash Bellek

EEPROM' da byte byte bloklara yazılırken, FLASH' da blok boyutu değişken ve daha büyük olabilir. EEPROM' da 1 byte programlaması için gerekli süre örneğin 1ms iken FLASH' da 32 Kbyte için gereken süre 10ms' dir [20].

3.MİKRODENETLEYİCİLER

Mikroislemcilere en az giriş-çıkış arabirimleri ile bellek modülü eklenerek tek bir yongada birleştirilmesi ile oluşan devre elemanı mikrodnetleyici olarak adlandırılır. Mikrodnetleyiciler, genellikle I/O arabirimleri ve bellek haricinde analog-sayısal dönüştürücü, seri giriş-çıkış arabirimi, darbe sinyali çıkışı, USB bağlantı kontrolcüsü, kızıl ötesi ve LCD ekran sürücüsü gibi uygulamaların yükünü azaltan, pratik modülleri de barındırabilir [20].

3.1. Mikrodnetleyicilerin Sağladığı Üstünlükler

Mikrodnetleyicilerin mikroislemcilere göre birtakım üstünlükleri bulunmaktadır. Bu üstün yanlar şöyle sıralanabilir;

- ✓ Mikroislemcili sistem mikrodnetleyicili sisteme göre daha karmaşık ve masraflıdır.
- ✓ Mikrodnetleyicili bir sistem için bir osilatör kaynağının olması yeterlidir.
- ✓ Mikrodnetleyicilerin küçük ve ucuz olmaları [16].

3.2. Mikroislemci ve Mikrodnetleyici Arasındaki Farklar

Mikroislemci ikili sayı sistemine göre çalışan, komut dizilerini işleyen, aritmatiksel ve mantıksal işlemleri yapan ve bunları denetleyen bir sistemdir. Mikrodnetleyici ise giriş çıkış birimlerini düzenleyen, programlayan ve içerisinde bu ekstra çevresel birimleri bulunduran mikroislemcili sistemlerdir [22]. Buna göre mikroislemci ve mikrodnetleyici arasındaki farkları belirlemek için her ikisinin de donanımını, kullandıkları alanları ve komut kümelerini incelememiz gerekir [21].

Donanım Olarak Karşılaştırılması: Mikroislemciler üzerlerinde kontrol birimi, aritmetik/mantıksal birim ve hafıza birimlerini bulundururlar. Mikrodnetleyiciler ise mikroislemcilere ek olarak dâhili veri ve kod belleği, zamanlayıcı/sayıcı, kesme

denetleyicisi, seri/paralel arabirimler, adres yolu ve veri yolu gibi çeşitli çevresel birimler içerirler.

Uygulama Alanları: Mikroişlemcili bir sistem, çeşitli kullanıcı programlarını yürütmek, çok sayıda veri işlemek, duyarlı sayısal hesaplamaları gerçekleştirmek için kullanıldığında yüksek verim sağlar. Mikrodenetleyiciye dayalı bir sistem ise, yeniden programlama gerektirmeyen sürekli sabit bir programın yürütüldüğü geniş kapsamlı sistemlerdir [22].

3.3. Mikrodenetleyici Seçiminde Dikkat Edilmesi Gereken Ölçütler

Bir mikrodenetleyici seçerken dikkat edilmesi gereken ölçütleri aşağıdaki gibi sıralayabiliriz.

Maliyet ve Temin Etme: Bir mikrodenetleyicinin maliyetinin düşük olması ve piyasada yaygın olarak bulunması gerekir [22].

Mikrodenetleyicinin Çalışma Hızı: Zamanın etkili olduğu uygulamalarda seçilecek ürünün hızı oldukça önemlidir [22].

Bellek Büyüklüğü ve Türü: Bellek büyüklüğü uygulama için yazılacak programı doğrudan etkiler. Günümüzde EEPROM veya FLASH kullanan kontrolörler, program geliştirmeye olanak sağladığı için tercih edilmektedir. Özellikle belleği daha hızlı silinip, yazılabilen FLASH belleğe sahip mikrodenetleyiciler daha çok tercih edilmektedir [22].

Giriş/Çıkış Port Sayısı: Yapılan uygulamaların ihtiyacına göre belirlenmektedir.

Zamanlayıcı/Sayıcı Adedi: Veri iletim hızının ayarlanması, PWM üretilmesi vb. daha birçok işlemlerde kullanılan bu birim uygulamalar için oldukça önemlidir [22].

Analog/Dijital Dönüştürücü: Günümüzde endüstriyel uygulamaların çoğunda analog/dijital dönüştürücüler gerekmektedir. Böylelikle üretici firmalarda bu dönüştürücüleri mikrodenetleyici içerisinde olmasına dikkat etmektedirler. A/D dönüştürücülerde dikkat edilmesi gereken iki özellik, ölçüm hassasiyetini gösteren bit sayısı ve kaç farklı sayıda noktadan ölçüm yapabileceğini gösteren kanal sayısıdır [22].

Enerji Tasarrufu: Seçilecek kontrolörün düşük güç ve uyku modlarına sahip olması gerekir.

Geliştirme Araçları: Kontrolör ile yapılan uygulamalarda kullanılan assembler ve C derleyicilerin kolay bulunması ve maliyetlerinin düşük olması önemli bir etkidir [22].

Müşteri Hizmetleri: Önemli birçok firma, ürünleri ile ilgili birçok programı kullanıcılara sunarak kullanıcıların işlerini kolaylaştırmaktadır.

3.4. Endüstride Kullanılan Mikrodenetleyiciler

Başlangıçta 'Intel' ve 'Texas Instruments' firmaları tarafından üretilen mikrodenetleyiciler günümüzde, birçok firma { Motorola, Microchip, Hitachi, Siemens, AMD, Intel, Atmel, Dallas Semiconductor, vb.} tarafından üretilmektedir [23-24].

Tablo 3.1. Günümüzde yaygın olan mikrodenetleyiciler ve üretici firmaları

Firma	Mikrodenetleyici serisi
Motorola	68HC
Atmel	AVR
Intel	8051 veya MSC-51
Microchip	PIC
Texas Instruments	MSP430 ve ARM

PIC (Peripheral Interface Controller- Çevresel Arabirim Denetleyicisi) mikrodenetleyicisi, Microchip firması tarafından üretilen, veri yolu genişliğine göre 8,16 ve 32-bit olarak sınıflandırılabilen ve yaygın kullanımı olan bir mikrodenetleyicidir [25].

Bu denetleyicinin; Fiyatının uygun olması, mantıksal işlemlerde performansının yüksek olması, 8-16 ve 32 bitlik çeşitlerinin olması, yüksek frekanslarda çalışabilme özelliği, uyku modunda 1 μ A gibi küçük bir akım çekmesi, farklı bir çok dille (Basic, C, Pascal, Assembly) programlanabilmesi, Harvard ve RISC mimarisine sahip olması, üretici firmanın başta simülatör programı MPLAB olmak üzere birçok kaynağı ücretsiz

yayınlanması, 6 pinden 80 pine kadar bir çok ürün çeşidinin olması tercih edilme sebepleridir [25].

DSPIC, Microchip firması tarafından üretilen 16 bitlik mikrodenetleyicidir. dsPIC mikrodenetleyicisi motor uygulamaları açısından 8 bitlik PIC 'lere göre; daha hızlıdır ve daha geniş program hafızasına sahiptir. DSPIC' in Önemli Özellikleri:

- RISC mimarisi kullanma,
- C programlama dili ile programlanabilmesi,
- 16 bit hafızaya sahip olma,
- 29 farklı kesme kaynağı,
- 6 bağımsız PWM kanalı bulundurma

Çolak İ. ,Şahin M ve Esen Z. yaptıkları çalışmada PID kontrol ile yapay sinir ağları tabanlı bir kontrol sistemi tasarlayarak fırçasız DC motorun pozisyon ve hız kontrolünü gerçekleştirmişlerdir. Bu sistemi Matlab/Simulink kullanarak simülasyonda, dsPIC kullanarak ise deneysel olarak gerçekleştirmişlerdir ve bu sonuçlar tamamen benzer çıkmıştır[26]. Alonso G. A., Ramírez A. ve Muñoz R.'nin yaptıkları çalışmada ise dsPIC30F6010 mikrodenetleyicisi kullanılarak yapay sinir ağları ile ayırık dalgacık dönüşümünün uygulaması yapılmış ve sıvı ortamdaki üç aminoasidin miktarları tespit edilmeye çalışılmıştır. Kullanılan bu mikrodenetleyici ile 39,20 ms gibi kısa bir sürede bu amino asitlerin miktarlarının elde edildiği gözlemlenmiştir[27].

Microchip firması dışında Atmel tarafından geliştirilen AVR serisi ve Texas Instruments'in geliştirdiği MSP430 ve ARM ailesi ürünleri endüstride geniş yer tutmaktadır. Bu denetleyicileri kullanarak arduino, cortex gibi geliştirme, programlama kartları oluşturulmuştur.

Arduino kartlarının donanımında bir adet Atmel AVR mikrodenetleyici (ATmega328, ATmega2560, ATmega32u4 gibi) ile birlikte programlama ve diğer devrelere bağlantı için gerekli yan elemanlar bulunur. Arduino kartlarında programlama için harici bir programlayıcıya ihtiyaç duyulmaz, çünkü karttaki mikrodenetleyiciye önceden bir bootloader programı yazılıdır. Arduino yazılımı bir geliştirme ortamı (IDE) ve kütüphanelerden oluşur. IDE, Java dilinde yazılmıştır ve Processing adlı dilin ortamına dayanmaktadır. Kütüphaneler ise C ve C++ dillerinde yazılmıştır ve AVR-GCC ve AVR

Libc. İle derlenmiştir.14-54 adet I/O pini, 32Kb-16Mb flash hafızası, 16-84 MHz'lik hızı,6-14 adet sağladığı PWM çıkışı, analog giriş/çıkış pinleri, kullanım kolaylığı vs.özelliklerine sahip olması, bu ürünlerin başlıca tercih nedenleridir. Arduino uno, leonardo, yun, due, mega vs. gibi çeşitleri vardır [28-31].

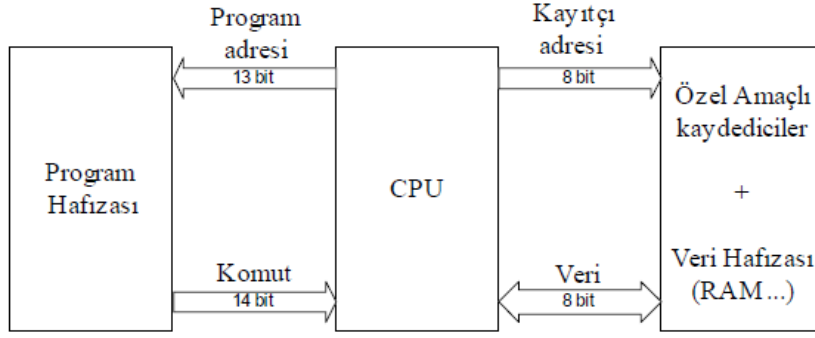
Munadi ve Akbar M.A. yaptıkları çalışmada potansiyometre üzerinden Arduino mega 2560 denetleyicisini kullanarak Matlab/simulink ile robot kollarında kullanılan DC servo motorun PID kontrol ve fuzzy lojik kontrolde vereceği cevaplar incelenmiştir. Fuzzy lojik kontrolün kalıcı durum hatası ve maksimum aşma açısından PID'ye göre daha iyi olduğu ve bu uygulamada fuzzy kullanımının daha doğru olacağı anlaşılmıştır [32].

ARM tabanlı işlemcilerin maliyetinin az olması, tasarruflu güç kullanımları ve performanslarının yüksek olması tercih sebepleridir. Dünyadaki 32 bit işlemcilerin %75'i ARM mimarisini kullanır. 90MHz ve üzeri hız, en az 20 I/O pini, yüksek hafıza vb. bu işlemcileri öne çıkaran özellikleridir. Bu işlemciler de arduino 'larda olduğu gibi geliştirme platformları üzerine kurulu olarak kullanılabilirler. ARM mimarisinde 3 temel grup kullanılır. Klasik ARM işlemciler(ARM7,ARM9,ARM11) ,gömülü sistemlere yönelik ARM işlemciler(Cortex-M0, Cortex-M1, Cortex-M3, Cortex-M4...) ve uygulamaya yönelik ARM işlemciler(Cortex-A5, Cortex-A8...) [29]. Franc H. Ve Şafari R. ARM Cortex M3 mikrodnetleyicisi C dili kaynak kodlarıyla programlayarak otomatik kapı içerisindeki bir DC motorun fuzzy lojik ile pozisyon, PI ile hız kontrolünü gerçekleştirmişlerdir ve değişik kapı kanat ağırlıkları için sistemin davranışlarını analiz etmişlerdir. Sonucunda ise bu kontrol metotlarının yanıt ilerleme şekli ve kontrol sağlamlığı açısından çok kullanışlı olduğunu gözlemlemişlerdir[33].

3.5. Uygulama İçin Kullanılan Mikrodnetleyiciler

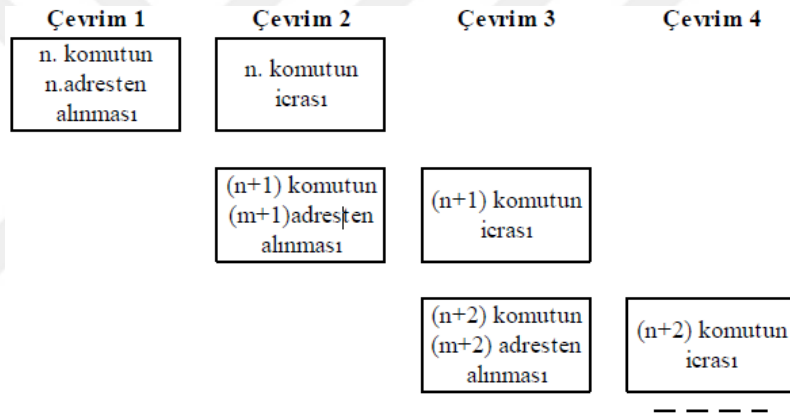
3.5.1. PIC16F877 Mikrodnetleyicisi

PIC 16CXX ailesinin özelliklerini taşıyan PIC16F877, Harvard ve RISC mimarisine, 8 bit veri yolu genişliğinde,14 bit program belleğine sahip Microchip firması tarafından üretilen bir mikrodnetleyicidir.

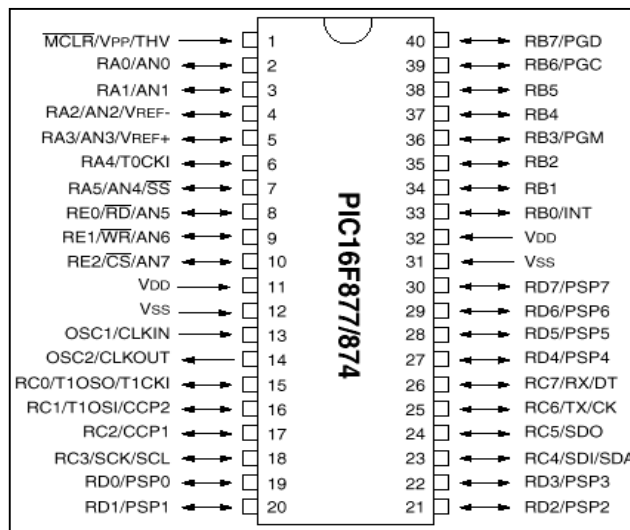


Şekil 3.1. PIC16F877 Harvard Mimarisi

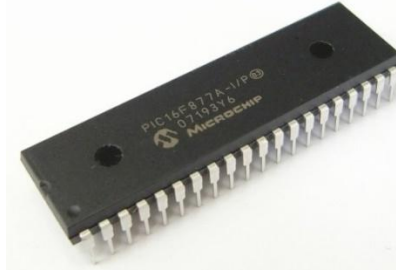
Her komuta bir çevrim süresinde (saykıl, cycle) erişilir ve komut yazmacına yüklenir. Dallanma komutları dışındaki bütün komutlar, aynı çevrim süresinde çalıştırılırlar. Bu sırada program sayacı, PC bir artar.



Şekil 3.2. Ardışıl Adreslerden Alınan Komutların İşlenmesi



Şekil 3.3. PIC16F87X Dış Mimarisi



Şekil 3.4. PIC16F87X Dış Görünüm

Kırk Pinli dış mimariye sahip olan denetleyici 20 MHz clock frekansına kadar çalışabilmektedir. Bu da her bir komut saykılını 200 ns de yaptığı anlamına gelir.

3.5.1.1. PIC16F877 Mikrodenetleyicisinin Temel Özellikleri

Tablo 3.2. PIC16F877 Mikrodenetleyicisinin Temel Özellikleri [25].

Parametre ismi	Değer
Program hafıza tipi	Flash
Program belleği (KB)	14
CPU Hızı (MIPS)	5
RAM Bytes	368
Data EEPROM (bytes)	256
Dijital iletişim çevre birimi	1-UART, 1-A/E/USART, 1-SPI, 1-I2C1-MSSP(SPI/I2C)
Yakalama/karşılaştırma/PWM birimi	2 CCP
Zamanlayıcılar	2 x 8-bit, 1 x 16-bit
ADC	10 bitlik 8 kanal
Karşılaştırıcılar	2
Sıcaklık oranı (C)	-40-125 derece
Çalışma voltajı (V)	2-5V
Pin sayısı	40

3.5.2. Arduino Due

Arduino Due, Atmel firması tarafından geliştirilmiş olan AT91SAM3X8E mikroişlemci tabanlı bir geliştirme platformudur. Arduino kartları hakkında detaylı bilgi Endüstride Kullanılan Mikrodenetleyiciler başlığı altında verilmiştir. Kullanılan mikrodenetleyicinin özellikleri Tablo 3.1 verilmiştir.

Tablo 3.3. Arduino Due'nin özellikleri

Mikrodenetleyici	AT91SAM3X8E
Çalışma gerilimi	3.3V
Önerilen giriş gerilimi	7-12V
Sınır giriş gerilimi	6-16V
Dijital giriş/çıkış pinleri	54 (12 adet PWM çıkış)
Analog giriş pinleri	12
Analog çıkış pinleri	2 (DAC)
Bütün I/O hatlarında DC çıkış akımı	130 mA
3.3V pini için DC akımı	800 mA
5V pini için DC akımı	800 mA
Flash Bellek	512 KB
SRAM	96 KB
Saat hızı	84 MHz
Uzunluk	101.52 mm
Genişlik	53.3 mm
Ağırlık	36 g



Şekil 3.5. Arduino Due

3.5.3 DSPIC30F3014 Mikrodenetleyicisi

DSPIC30F3014, Microchip firması tarafından üretilen 16 bitlik dijital sinyal kontrolörüdür. Harvard mimarisine sahip olan bu denetleyici, 20 MHz lik osilatör ile çalışabilmekte ve yazılımsal olarak bu hız yaklaşık 100 MHz e kadar çıkartılabilmektedir. Giriş/çıkış pinlerinden 25 mA akım çekilebilmektedir. Pic16f877a'dan farklı olarak 12 bitlik ADC'ye sahiptir.



MCLR	1	40	AVDD
AN0/VREF+/CN2/RB0	2	39	AVSS
AN1/VREF-/CN3/RB1	3	38	AN9/RB9
AN2/SS1/LVDIN/CN4/RB2	4	37	AN10/RB10
AN3/CN5/RB3	5	36	AN11/RB11
AN4/CN6/RB4	6	35	AN12/RB12
AN5/CN7/RB5	7	34	EMUC2/OC1/RD0
PGC/EMUC/AN6/OCFA/RB6	8	33	EMUC2/OC2/RD1
PGD/EMUD/AN7/RB7	9	32	VDD
AN8/RB8	10	31	VSS
VDD	11	30	RF0
VSS	12	29	RF1
OSC1/CLKI	13	28	U2RX/CN17/RF4
OSC2/CLKO/RC15	14	27	U2TX/CN18/RF5
EMUC1/SOSCI/T2CK/U1ATX/CN1/RC13	15	26	U1RX/SD1/SDA/RF2
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	16	25	EMUC3/U1TX/SDO1/SCL/RF3
INT0/RA11	17	24	EMUC3/SCK1/RF6
IC2/INT2/RD9	18	23	IC1/INT1/RD8
RD3	19	22	RD2
VSS	20	21	VDD

Şekil 3.6. DSPIC30F3014

Bu mikrodenetleyiciye ait bazı özellikler aşağıdaki tabloda görülmektedir.

Tablo 3.4. DSPIC30F3014 mikrodenetleyicisinin özellikleri

Mikrodenetleyici	dsPIC30F3014
Mimarisi	16-bit
CPU hızı (MIPS)	30
Hafıza tipi	Flash
Program belleği (KB)	24
RAM Bytes	2,048
Çalışma sıcaklığı C	-40 to 125
Çalışma voltajı	2.5 to 5.5
I/O Pinleri	30
Pin sayısı	40
İçsel osilatör	7.37 MHz, 512 kHz
Dijital iletişim birimi	2-UART 1-SPI 1-I2C
Analog birimi	1-A/D 13x12-bit
Zamanlayıcı	3 x 16-bit 1 x 32-bit

4. MİKRODENETLEYİCİ SEÇİMİ

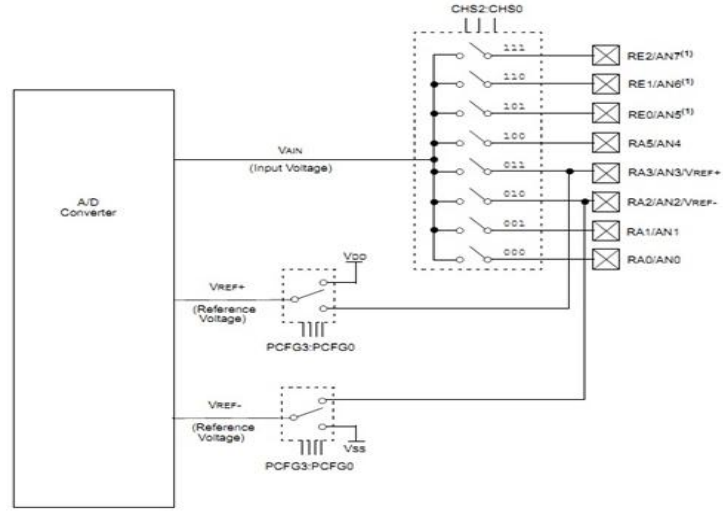
Mikrodenetleyicilerin performans analizini yapmak için iki adet mikrodenetleyici kullanılmıştır. Bunlar; Microchip firması tarafından üretilen PIC ailesi ürünü PIC16F877A ve Atmel firması tarafından üretilen AT91SAM3X8E mikroişlemci tabanlı bir denetleyici olan Arduino Due'dir. Bu uygulamalarda iki entegreye de analog bir sinyal verilmiş ve çıkışında yine analog bir sinyal elde edilmeye çalışılmıştır. Buradaki amaçlar;

- ✓ Girişe uygulanan analog sinyal ile çıkışta elde edilen analog sinyal arasında genlik ve frekans farklılıklarını incelemek.
- ✓ Mikrodenetleyicilerin okuma ve yazma sonucunda oluşan gecikmeyi görebilmek
- ✓ Girişe verilen analog sinyali mikrodenetleyici içerisinde dijitalle dönüştürdüktan sonra birtakım işlemlerden geçirerek çıkışından aldığımız analog sinyaldeki gecikme ve genlik değerlerini gözlemlemek
- ✓ Analog girişe verilen sinyalin mikrodenetleyici içerisinde P,PI vb. gibi kontrol algoritmalarını gerçekleştirilme kabiliyetini incelemek

4.1 PIC16F877A ile A/D-D/A devresi

4.1.1 A/D Modülü

Doğada doğal olarak bulunan tüm sistemler analog düzendedir. Kullandığımız mikrodenetleyiciler ise dijital sistemlerdir. Bu nedenle dış dünya ile iletişim kurmak ve dış dünyadan veri almak için bu iki sistemin bir şekilde birbirini anlaması gerekir. Bu iş için de analog sinyallerin (ısı, ışık, sesi nem, gerilim, akım, vb.) dijital sistemlerin anlayacağı dijital sistemlere çevrilmesi gereklidir. Bu işlem için ADC devreleri ve entegreleri bulunmaktadır. PIC16F877A mikrodenetleyicisinde ADC modülü bulunmaktadır. Böylece harici bir devre veya entegre kullanmadan analog sinyaller dahili ADC modülü sayesinde dijital sinyallere çevrilebilir [34]. Şekil 4.1'de PIC16F877A da bulunan ADC modülünün blok diyagramı görülmektedir.



Şekil 4.1. A/D modülü blok diyagramı

PIC16F877A mikrodenetleyicisinde 8 adet ADC giriş ucu (AN0...AN7) vardır. ADC birimi 4 adet kaydediciye sahiptir. Çevrim sonucunun yazıldığı ADRESH (ADRES High Register) ve ADRESL (ADRES Low Register) ile kontrol kaydedicileri, ADCON0 (A/D Control Register0) ve ADCON1 (A/D Control Register1). ADCON0 ve ADCON1 kaydedicileri ve bitlerinin açıklamaları Şekil 4.2 ve Şekil 4.3’de görülmektedir.

ADCON0: A/D CONTROL REGISTER							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7						bit 0	
bit 7-6	ADCS<1:0> : A/D Conversion Clock Select bits						
	<u>If ADCS2 = 0:</u>						
	00 = FOSC/2						
	01 = FOSC/8						
	10 = FOSC/32						
	11 = FRC (clock derived from the internal A/D module RC oscillator)						
	<u>If ADCS2 = 1:</u>						
	00 = FOSC/4						
	01 = FOSC/16						
	10 = FOSC/64						
	11 = FRC (clock derived from the internal A/D module RC oscillator)						
bit 5-3	CHS<2:0> : Analog Channel Select bits						
	000 = Channel 0 (RA0/AN0)						
	001 = Channel 1 (RA1/AN1)						
	010 = Channel 2 (RA2/AN2)						
	011 = Channel 3 (RA3/AN3)						
	100 = Channel 4 (RA4/AN4)						
	101 = Channel 5 (RB6/AN5)						
	110 = Channel 6 (RB7/AN6)						
bit 2	GO/DONE : A/D Conversion Status bit						
	<u>If ADON = 1:</u>						
	1 = A/D conversion in progress (setting this bit starts the A/D conversion)						
	0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)						
bit 1	Unimplemented : Read as '0'						
bit 0	ADON : A/D On bit						
	1 = A/D converter module is operating						
	0 = A/D converter module is shut off and consumes no operating current						

Şekil 4.2. PIC16F877A ADCON0 Kaydedicisi

ADCON1 REGISTER																																																																																																																																																																																																												
U-0		U-0		R/W-0		U-0		R/W-0		R/W-0		R/W-0																																																																																																																																																																																																
ADFM		—		—		—		PCFG3		PCFG2		PCFG1																																																																																																																																																																																																
											bit 7	bit 0																																																																																																																																																																																																
<p>bit 7 ADFM: A/D Result Format Select bit 1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'. 0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.</p> <p>bit 6-4 Unimplemented: Read as '0'</p> <p>bit 3-0 PCFG3:PCFG0: A/D Port Configuration Control bits:</p> <table border="1"> <thead> <tr> <th>PCFG3: PCFG0</th> <th>AN7⁽¹⁾ RE2</th> <th>AN6⁽¹⁾ RE1</th> <th>AN5⁽¹⁾ RE0</th> <th>AN4 RA5</th> <th>AN3 RA3</th> <th>AN2 RA2</th> <th>AN1 RA1</th> <th>AN0 RA0</th> <th>VREF+</th> <th>VREF-</th> <th>CHAN/ Refs⁽²⁾</th> </tr> </thead> <tbody> <tr><td>0000</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>VDD</td><td>VSS</td><td>8/0</td></tr> <tr><td>0001</td><td>A</td><td>A</td><td>A</td><td>A</td><td>VREF+</td><td>A</td><td>A</td><td>A</td><td>RA3</td><td>VSS</td><td>7/1</td></tr> <tr><td>0010</td><td>D</td><td>D</td><td>D</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>VDD</td><td>VSS</td><td>5/0</td></tr> <tr><td>0011</td><td>D</td><td>D</td><td>D</td><td>A</td><td>VREF+</td><td>A</td><td>A</td><td>A</td><td>RA3</td><td>VSS</td><td>4/1</td></tr> <tr><td>0100</td><td>D</td><td>D</td><td>D</td><td>D</td><td>A</td><td>D</td><td>A</td><td>A</td><td>VDD</td><td>VSS</td><td>3/0</td></tr> <tr><td>0101</td><td>D</td><td>D</td><td>D</td><td>D</td><td>VREF+</td><td>D</td><td>A</td><td>A</td><td>RA3</td><td>VSS</td><td>2/1</td></tr> <tr><td>011x</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>VDD</td><td>VSS</td><td>0/0</td></tr> <tr><td>1000</td><td>A</td><td>A</td><td>A</td><td>A</td><td>VREF+</td><td>VREF-</td><td>A</td><td>A</td><td>RA3</td><td>RA2</td><td>6/2</td></tr> <tr><td>1001</td><td>D</td><td>D</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>A</td><td>VDD</td><td>VSS</td><td>6/0</td></tr> <tr><td>1010</td><td>D</td><td>D</td><td>A</td><td>A</td><td>VREF+</td><td>A</td><td>A</td><td>A</td><td>RA3</td><td>VSS</td><td>5/1</td></tr> <tr><td>1011</td><td>D</td><td>D</td><td>A</td><td>A</td><td>VREF+</td><td>VREF-</td><td>A</td><td>A</td><td>RA3</td><td>RA2</td><td>4/2</td></tr> <tr><td>1100</td><td>D</td><td>D</td><td>D</td><td>A</td><td>VREF+</td><td>VREF-</td><td>A</td><td>A</td><td>RA3</td><td>RA2</td><td>3/2</td></tr> <tr><td>1101</td><td>D</td><td>D</td><td>D</td><td>D</td><td>VREF+</td><td>VREF-</td><td>A</td><td>A</td><td>RA3</td><td>RA2</td><td>2/2</td></tr> <tr><td>1110</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>D</td><td>A</td><td>VDD</td><td>VSS</td><td>1/0</td></tr> <tr><td>1111</td><td>D</td><td>D</td><td>D</td><td>D</td><td>VREF+</td><td>VREF-</td><td>D</td><td>A</td><td>RA3</td><td>RA2</td><td>1/2</td></tr> </tbody> </table> <p>A = Analog input D = Digital I/O</p>													PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs ⁽²⁾	0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0	0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1	0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0	0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1	0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0	0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1	011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0	1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2	1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0	1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1	1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2	1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2	1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2	1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0	1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2
PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs ⁽²⁾																																																																																																																																																																																																	
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0																																																																																																																																																																																																	
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1																																																																																																																																																																																																	
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0																																																																																																																																																																																																	
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1																																																																																																																																																																																																	
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0																																																																																																																																																																																																	
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1																																																																																																																																																																																																	
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0																																																																																																																																																																																																	
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2																																																																																																																																																																																																	
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0																																																																																																																																																																																																	
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1																																																																																																																																																																																																	
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2																																																																																																																																																																																																	
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2																																																																																																																																																																																																	
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2																																																																																																																																																																																																	
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0																																																																																																																																																																																																	
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2																																																																																																																																																																																																	

Şekil 4.3. PIC16F877A ADCON1 Kaydedicisi

m bit değerli bir analog dijital dönüştürücü en çok 2^m adet değer ile bir analog işareti örnekleyebilir. Bu ADC modülünün çözünürlüğü 10 bit olduğundan $2^{10}=1024$ adet değer ile örnekleme yapabilir. Bu bit sayısı da dönüştürücünün çözünürlüğünü ifade eder. Çözünürlüğün yüksek olması demek daha iyi dönüşüm yapılacağı anlamına gelir. ADC'nin dijital bilgiye dönüştürebildiği en küçük gerilim değerine adım büyüklüğü denir. Adım büyüklüğü şu şekilde hesaplanır;

$$\text{Adım Büyüklüğü} = \frac{V_{ref(+)} - V_{ref(-)}}{2^m} \quad 4.1$$

Katalog değerlerine göre pozitif referans gerilimi 5V, negatif referans gerilimi ise 0V olarak alınmıştır.

$$\text{Adım Büyüklüğü} = \frac{V_{ref(+)} - V_{ref(-)}}{2^m} = \frac{5V - 0}{1024} = 0.0048828125 V$$

Maksimum değeri $V_{in}=V_{ref(+)}=5$ V, minimum değeri $V_{in}=V_{ref(-)}=0$ V olan bir sinyalin 10 bitlik ADC entegresi çıkışındaki dijital değerleri Tablo 7.1’de daha iyi görülebilir. 10 bitlik ADC için, dijital değer 0000000000 (desimal 0)’dan başlar 1111111111 (desimal 1023)’e kadar gider. 0’dan başladığı için son değer 1024 değil 1023 olur.

Tablo 4.1. 0-5V analog sinyalin 10 bitlik ADC için dijital çıkış değerleri.

Analog Giriş (V)	Sayısal Bilgi Karşılığı
0V ile 0,0048828125V arasında ise	0000000000
0,0048828125V ile 0,009765625V arasında ise	0000000001
0,009765625V ile 0,0146484375V arasında ise	0000000010
0,0146484375V ile 0,01953125V arasında ise	0000000011
0,01953125V ile 0,0244140625V arasında ise	0000000100
...	...
...	...
...	...
4,990234375V ile 4,9951171875V arasında ise	1111111110
4,9951171875V ile 5V arasında ise	1111111111

ADC birimi girişindeki analog sinyalin dijital bilgiye çevrilmesi işlemi için bir süre geçer. Bir analog sinyalin dijital bilgiye çevrilmesi için gereken süreye çevrim süresi denir. ADC işlemi için kullanılacak clock sinyali frekansı ne kadar yüksek ise çevirme işlemi o kadar hızlı olur. Analog sinyalin dijital bilgiye çevrilmesi işleminde, analog sinyalin hesaplanan dijital karşılığı arasında oluşan farka kuantalama hatası denir.

ADC biriminin uçlarına gelen sinyalin gerilim değerini hesaplamak için ise aşağıdaki formül kullanılır.

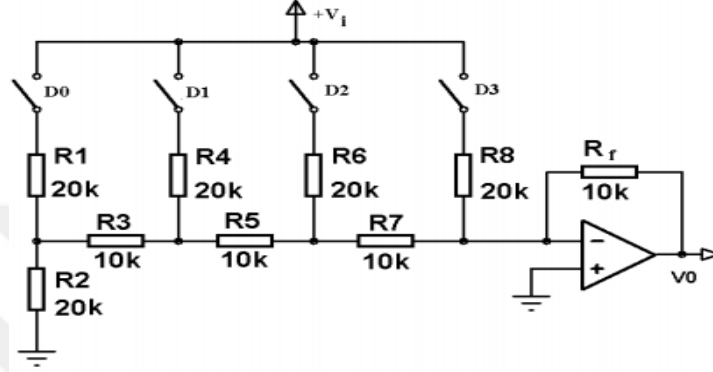
$$ADC \text{ girişine gelen sinyal gerilimi} = \text{Sinyal çıkış değerinin desimal değeri} * \text{Adım büyüklüğü}$$

Örnek olarak 10 bitlik ADC ucuna gelen analog sinyalin dijital karşılığı çevrim sonucunda 1000000000 olarak bulunduysa, bu dijital bilginin gerilim değeri verilen formüle göre aşağıda gösterildiği gibi hesaplanır.

Sinyalin gerilimi=512 (binary olarak 1000000000)x0,0048828125=2,5 V

4.1.2 D/A Modülü

PIC16F877A’da DAC modülü bulunmadığı için bu dönüşüm R-2R merdiven tipi dönüştürme devresi kullanılarak yapılmıştır.



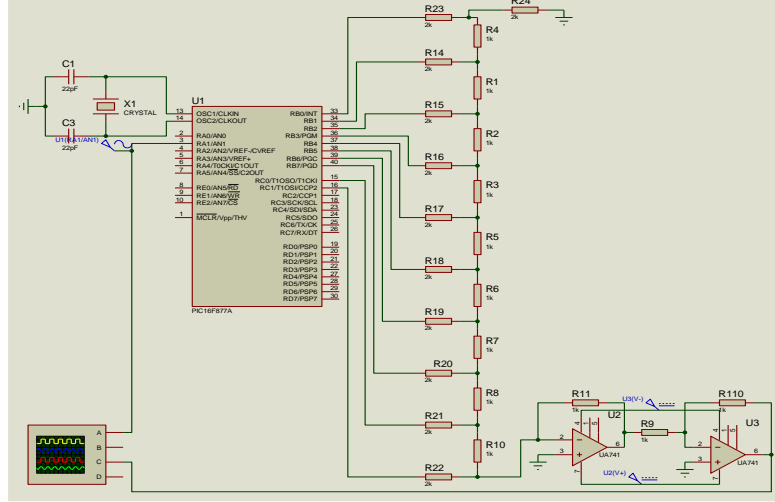
Şekil 4.4. 4 bitlik 2R-R merdiven tipi D/A dönüştürücü

Şekilde görüldüğü gibi devre R ve 2*R değerlerinden oluştuğundan ve çıkıştaki dalga şeklinin merdiven basamağı şeklinde oluşmasından dolayı bu şekilde isimlendirilmiştir. Şekil 4.4’te verilen R-2R merdiven tip DAC devresinde D0, D1, D2, D3 ile gösterilmiş dijital girişlerin op-amp eviren girişine etkileri farklıdır. Önünde büyük direnç değeri olan dijital giriş op-amp girişine daha az akım ulaştıracaktır ve bunun sonucu olarak da çıkıştaki etkisi daha az olacaktır. D0 en değersiz bit (LSB) olup devrenin çözünürlüğünü belirler. D3 ise en ağırlıklı bit (MSB) olup çıkışa tam skala değerinin yarısı olarak etki eder. Devrenin çıkış gerilimini veren formül aşağıda verilmiştir.

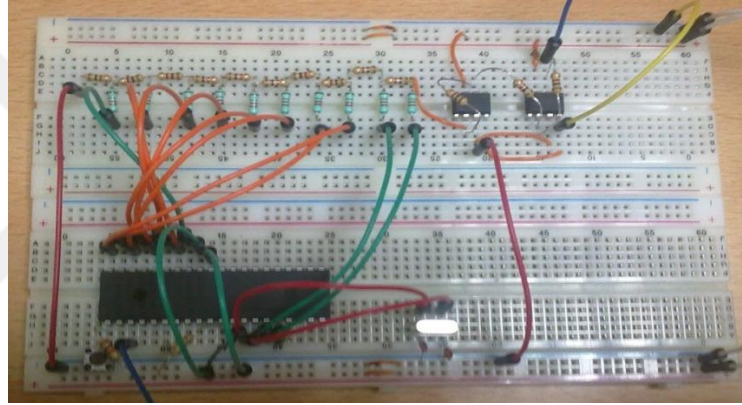
$$\text{DAC çıkış gerilimi } V_o = \frac{-V_{ref}}{16} (D_0 + 2D_1 + 4D_2 + 8D_3)$$

4.1.3 Devrenin Kurulumu

Devre ilk başta elektronik benzetim programı olan Proteus 7 Professional’in ISIS arayüzünde kurularak devrenin çalışması ve sonuçlar gözlemlenmiştir.



(a)

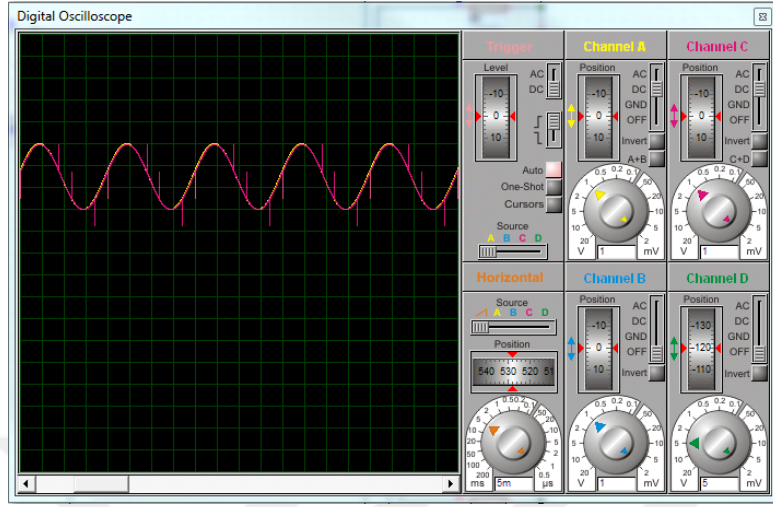


(b)

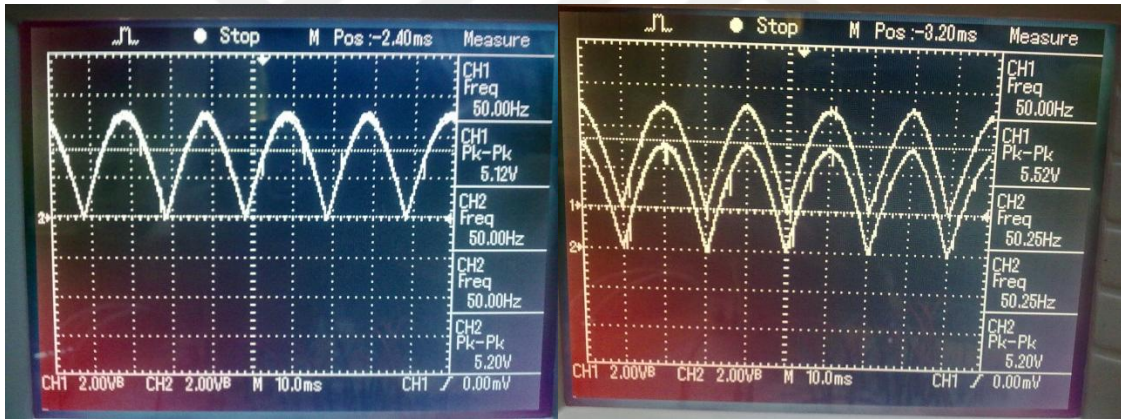
Şekil 4.5. Devrenin simülasyonda (a) ve breadboard (b) üzerinde kurulumu

Devrede mikrodenetleyici olarak bir adet PIC16F877A, osilatör frekansı için iki adet 22pF'lık kondansatör ve bir adet 20MHz'lik kristal osilatör, D/A dönüştürme işlemi için 11 adet 2k ohm direnç, 9 adet 1k ohm direnç ve iki adet UA741 işlemsel kuvvetlendirici, ve sonuçları görebilmek için ise bir adet osiloskop kullanılmıştır. Giriş sinyali olarak devreye 0-5V arasında 50 Hz frekansa sahip sinüs uygulanmıştır. Burada iki adet op-amp kullanılmasının sebebi R-2R'nin çıkışına bağlanan op-amp evirici devresi olduğundan gerilimi tersine çevirmiştir. Onun çıkışına da eviren bir op-amp bağlanarak gerilimin tekrar evirilmesi sağlanmış ve doğru gerilim değerleri alınmıştır. Micro C PRO for PIC programında C dilinde yazılmış olan program makine dili olan hex kodlarına

dönüştürülerek PIC'e yüklenmiştir. Programın hex kodları Picket2 ve program arayüzü kullanılarak PIC16F877A'ya yüklenmiştir.



(a)

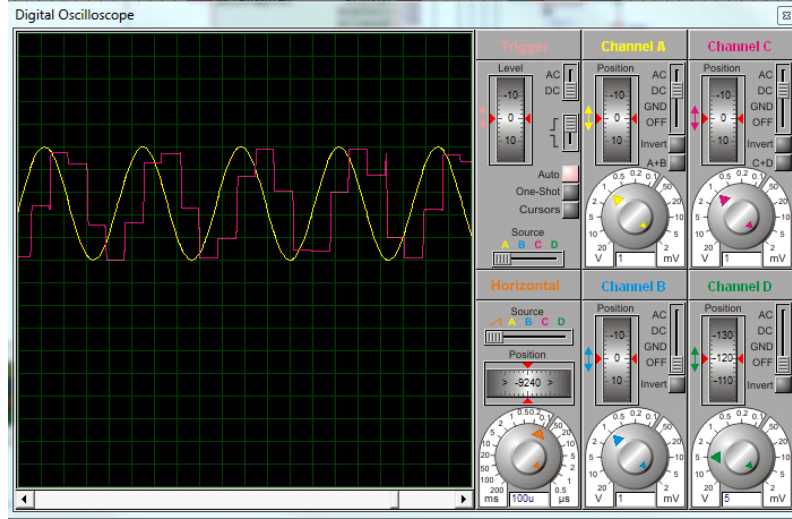


(b)

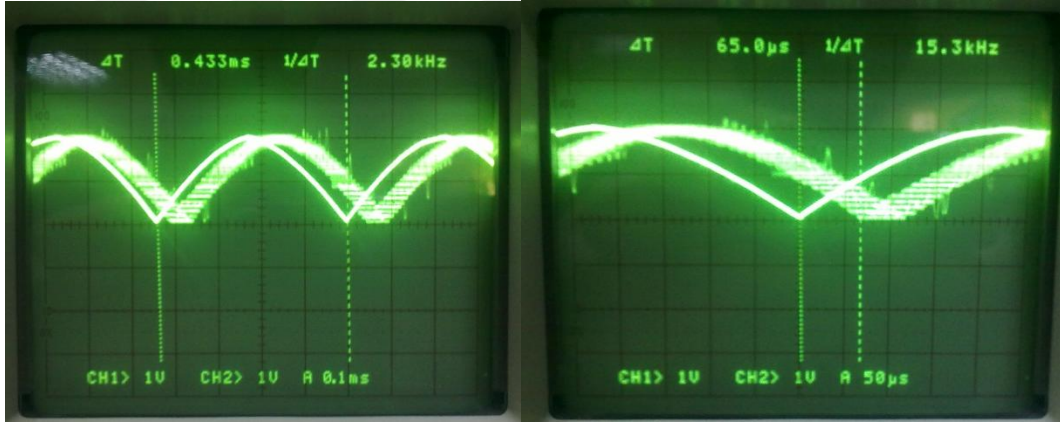
(c)

Şekil 4.6. Devrenin simülasyondaki (a) ve dijital osiloskoptaki (b) (c) giriş-çıkış dalga şekilleri

Şekil 4.6 (c)'de çıkış sinyalinin daha iyi görülebilmesi için osiloskopta bir kare aşağı indirilmiştir. Devrede herhangi bir gecikme olmadığı ve işlem yapılmadığı zaman analog kanaldan okunan sinyal direk olarak portlara verilmektedir. Ayrıca girişe uygulanan sinyal 50Hz'lik bir sinyal olduğundan mikrodenetleyici bu hıza yetişebilmektedir. Girişe 2.3 kHz'lik bir sinyal verildiğinde analog sinyalden ard arda alınan örnekler arasında gecikme yaşanacağı için Şekil 4.7'de de görüldüğü gibi hem bir zaman gecikmesine hem de sinyalin genliğini tam olarak takip edememesine neden olmuştur.



(a)

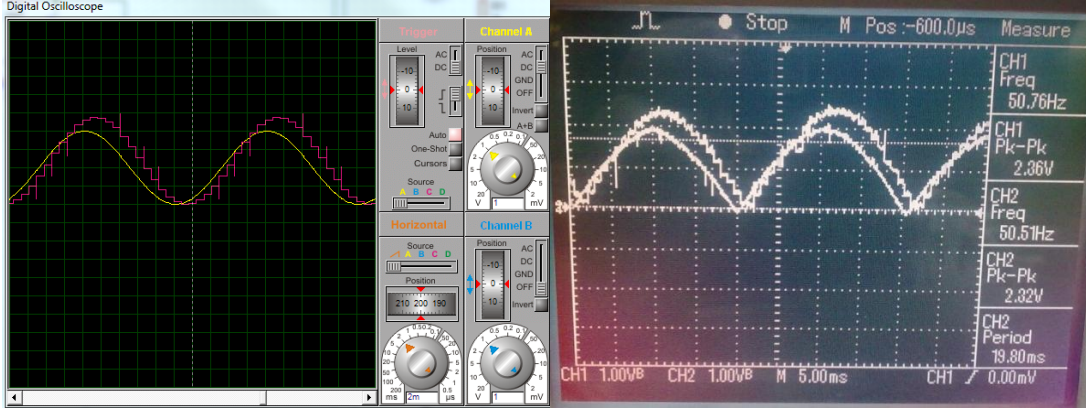


(b)

(c)

Şekil 4.7. Devrenin simülasyon (a) ve analog osilaskoptaki (b) (c) 2.3kHz giriş sinyali sonucu giriş-çıkış dalga şekilleri

Mikrodenetleyicinin programlanabilirliğini incelemek için okunan analog sinyal PI kontrolörden geçirilerek çıkışa verilmiştir. Bu programda okunan analog sinyal, referans değer ile geri beslemeden gelen sinyalin farkı olan hata sinyali olarak kabul edilmiştir. PI kontrol uygulamak için hatanın oransal katsayı (K_p) ile çarpımı ve hatanın integralinin de integral katsayısı (K_i) ile çarpımının toplamı alınmıştır. Ayrıca ileride daha fazla komut ekleneceğini göz önünde bulundurarak 100 komutluk bir döngü oluşturulmuştur.

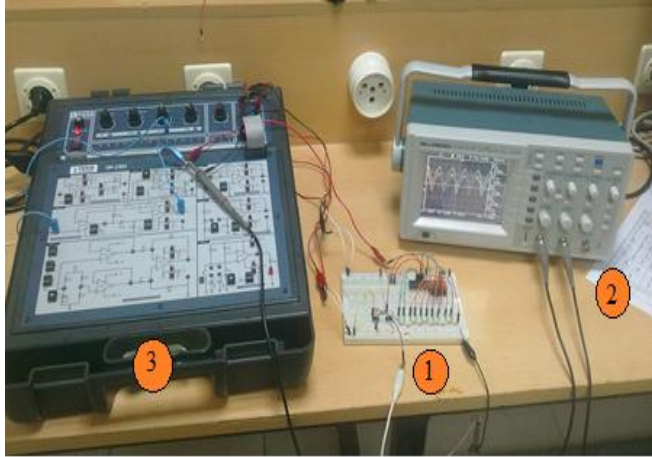


(a)

(b)

Şekil 4.8. Devrenin simülasyonda (a) ve dijital osiloskopta (b) PI uygulanmış giriş-çıkış dalga şekilleri

PIC'e negatif gerilim verilemediği için simülasyonda 0-5V aralığında bir sinyal kullanılmıştır. Breadboardda kurulan devrede ise 0-5V aralığında tam dalga doğrultulmuş sinüs verilmiştir. Bu giriş sinyali için ve devredeki PIC ve op-amp'ların besleme gerilimleri için Şekil 4.9'de görülen OP 2300 OP-AMP eğitim seti kullanılmıştır.



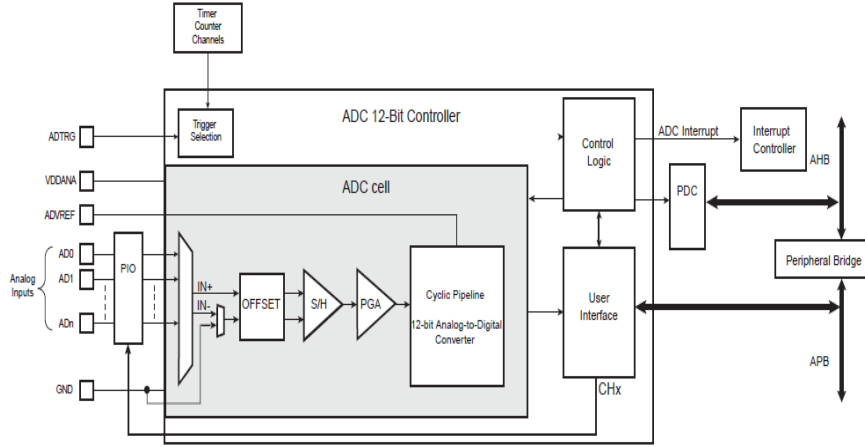
1-PIC16F877A ADC- DAC devresi

2-Osiloskop

3-OP 2300 OP-AMP eğitim seti

Şekil 4.9. ADC-DAC devresi deney düzeneği

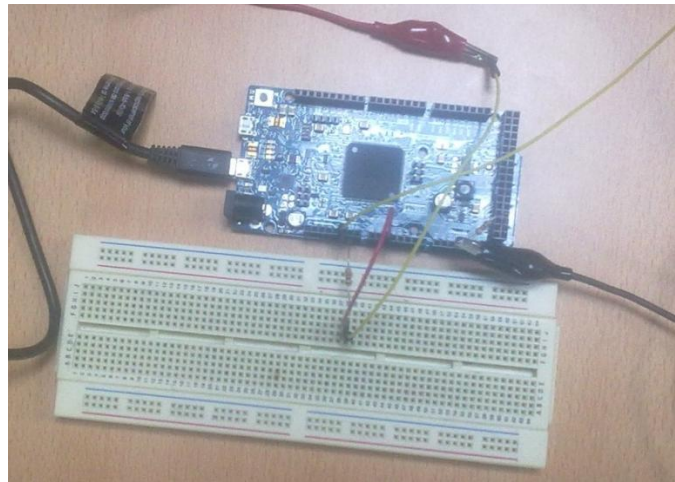
4.2 Arduino Due ile A/D-D/A devresi



Şekil 4.10. Arduino Analog-Dijital Dönüşüm Blok Diyagramı

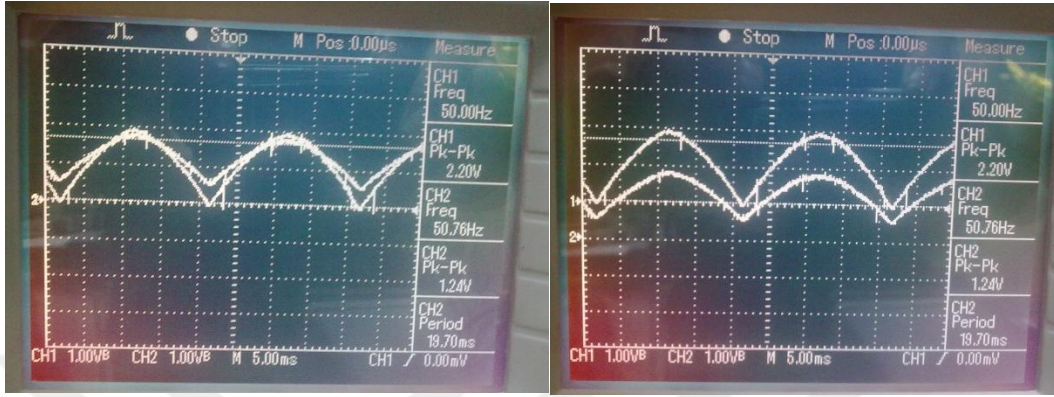
Arduino Due, diğer Arduino'larla karşılaştırıldığında daha üstün olduğu anlaşılmaktadır. Arduino Due'nin kendi devresi içerisinde 12 bitlik DAC ve ADC modülleri bulunmaktadır.

Arduino Due C tabanlı kendi arayüz programından programlanarak USB ile mikroişlemciye yüklenmiştir. Beslemesini USB den aldığından çalışması için ek bir beslemeye ihtiyaç duyulmamıştır. Sadece Şekil 7.12'de görüldüğü gibi direnç üzerinden giriş gerilimi verilmiş ve yine direnç üzerinden doğrudan çıkış gerilimi alınmıştır.



Şekil 4.11. Arduino Due ADC-DAC devre bağlantısı

Analog giriş olarak PIC’de olduğu gibi OP 2300 OP-AMP eğitim setinden faydalanarak 50 Hz frekanslı tam dalga doğrultulmuş sinüs verilmiştir. Programda sinyalde bir değişiklik yapılmadan Arduino’nun kendi DAC’ından direkt çıkışa verilmiştir.

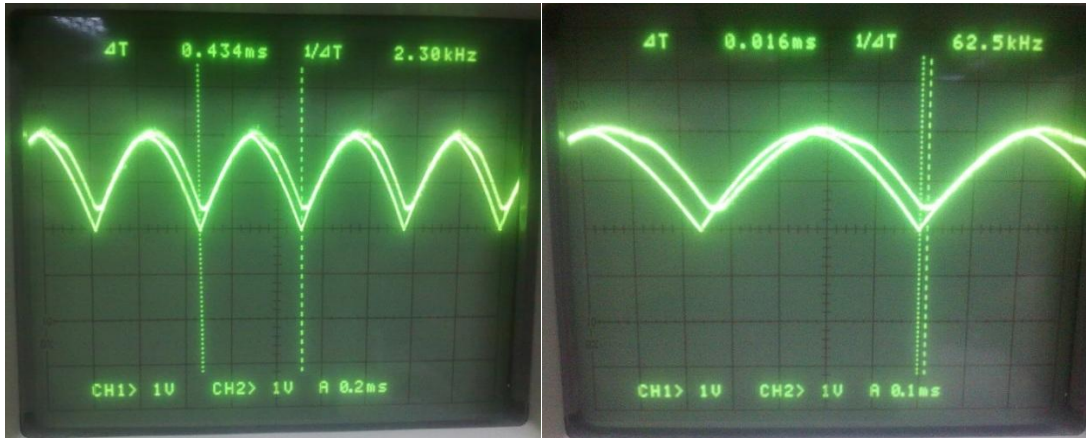


(a)

(b)

Şekil 4.12. Arduino Due ADC-DAC giriş çıkış dalga şekilleri

Şekil 4.12 (b)’de sonucun daha iyi görülebilmesi için sinyal bir kare aşağı indirilmiştir. Arduino’nun hız performansını görebilmek için Şekil 4.13 (a)’da görüldüğü gibi 2.3kHz frekanslı tam dalga doğrultulmuş sinüs verilmiştir.



(a)

(b)

Şekil 4.13. 2.30kHz giriş sinyali ile giriş-çıkış dalga şekilleri

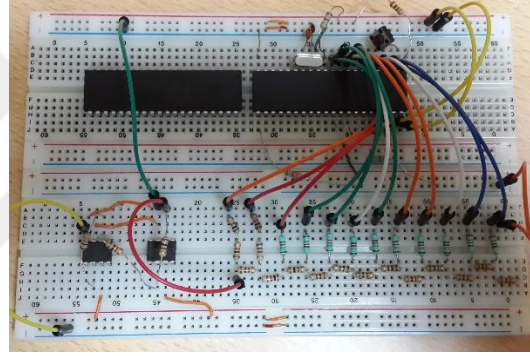
2.30kHz giriş sinyali uygulandığında genlikte bir bozulma olmadığı ancak giriş ile çıkış sinyalleri arasında 16 ms lik bir gecikme olduğu Şekil 4.13 de görülmektedir.

4.3.2 DAC Modülü

Dspic 'in DAC çıkışı olmadığından dolayı Pic16f877a uygulamasında kullanılan 2R merdiven tipi dönüştürücü devresi kullanılmıştır. Bu devreye ek olarak bu denetleyicide 12 bit ADC olduğu için iki çıkış daha eklenmiştir.

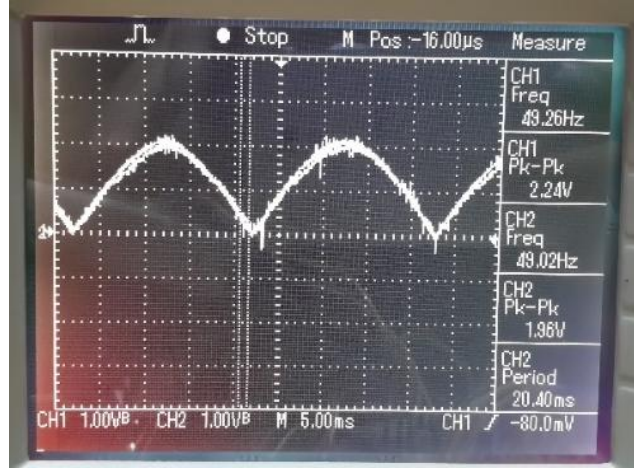
4.3.3 Devrenin Kurulumu

Bu denetleyici, Proteus simülasyon programında yer almadığı için devre direkt olarak board üzerine kurularak çalıştırılmıştır.



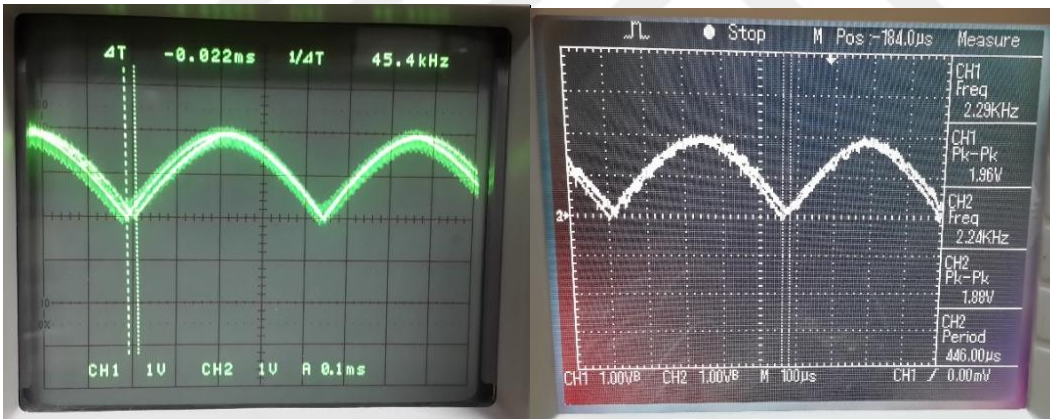
Şekil 4.15. Devrenin breadboard üzerinde kurulumu

Devrede mikrodenetleyici olarak bir adet DSPIC30F3014, osilatör frekansı için iki adet 22pF'lık kondansatör ve bir adet 20MHz'lik kristal osilatör, D/A dönüştürme işlemi için 13 adet 2k ohm direnç, 11 adet 1k ohm direnç ve iki adet UA741 işlemsel kuvvetlendirici ve sonuçları görebilmek için ise bir adet osiloskop kullanılmıştır. Micro C PRO for DSPIC programında C dilinde yazılmış olan program makine dili olan hex kodlarına dönüştürülerek PIC'e yüklenmiştir. Programın hex kodları Pickit2 ve program arayüzü kullanılarak PIC16F877A'ya yüklenmiştir. Arduino ve pic16f877a'da olduğu gibi bu uygulamada da dspic' e 50 ve 2,3 kHz'lik tam dalga doğrultulmuş sinüs sinyalleri verilmiş, matris, döngüsel işlemler gibi kodlardan geçirilmiş ve tekrar çıkıştan analog sinyal alınmıştır.



Şekil 4.16. 50 Hz'lik giriş-çıkış sinyali

Yukarıdaki şekilde de görüldüğü gibi 50 Hz 'lik bir sinüs sinyalinde herhangi bir gecikme ve gerilim bozukluğu olmadan denetleyici sinyali çıkışa aktarmıştır. Performansın daha iyi incelenebilmesi için giriş dalga şeklinin frekansı 2,3 kHz'e çıkarılarak mikrodenetleyiciye uygulanmıştır.



(a)

(b)

Şekil 4.17. Devrenin analog (a) ve dijital osiloskopta (b) 2,3kHz uygulanmış giriş-çıkış dalga şekilleri

2,3 kHz sinüs dalgası uygulandığında çıkış dalga şeklinde gerilim genliğinde bir değişme meydana gelmedi. Ancak şekilden de görüldüğü gibi 22 ms' lik bir gecikme olmuştur.

Deneyel sonuçlar incelendiğinde 50 Hz'lik sinyal uygulandığında 3 denetleyici de sorunsuz olarak giriş sinyalini bozulma ve gecikme olmadan takip etmiştir. Ancak giriş sinyali 2.3 kHz'e çıkarıldığında pic16f877a'da hem genlikte bozulmalar olmuştur hem de giriş ile çıkış sinyali arasında 64 ms lik bir gecikme olmuştur. Dspic'e bakıldığında ise

genlik bozulması olmamış ancak iki sinyal arasında 22 ms lik gecikme yaşanmıştır. Arduino da ise genlik bozulması yaşanmamış ve 16 ms lik gecikme olmuştur. 84 MHz osilatör frekansı ve 12 bit çözünürlüklü Arduino Due'nin, 20 MHz osilatör frekanslı ve 10 bit çözünürlüğe sahip PIC16F877A'ya ve 20 MHz osilatör frekanslı ve 12 bit çözünürlüğe sahip DSPIC30F3014'e göre daha net ve daha hızlı sonuçlar verdiği görülmüştür. PIC deki gecikme 4 kat daha fazla olmuştur ve çözünürlüğü daha düşük olduğundan çıkış dalga şekli basamak şeklinde oluşmuştur. Arduino Due'nin diğerlerine göre üstünlükleri;

- ✓ Çalışma hızının yüksek olması
- ✓ Program belleğinin fazla olması
- ✓ Analog kanal sayısı ve çözünürlüğünün yüksek olması
- ✓ Giriş/çıkış sayısının fazla olması
- ✓ DAC modülünün bulunması
- ✓ Devre kurulum kolaylığı
- ✓ Programlamanın devre çalışırken bile USB den yapılabilmesi

Bu sonuçlar dikkate alındığında sıcaklık ve motor kontrol uygulamalarında Arduino Due'nin kullanımı uygun bulunmuştur.

5.KONTROL SİSTEMİ

Kontrol terimi genel anlamda dengelemek, yönlendirmek ya da arzu edilen bir şekilde yönetmek anlamında kullanılır. Fiziksel sistemler esas alındığında girişi ve çıkışı olan fiziksel bir sistemin kontrolü, sistemin çıkışının kullanıcının arzu ettiği değerden sapması durumunda, çıkışı düzeltmek amacıyla sistemin girişinin uygun şekilde değiştirilmesini ifade eder. Sistem girişinin değiştirilmesi kullanıcı tarafından yapılırsa el ile kontrol, insan unsuru olmaksızın başka bir fiziksel kontrol tarafından gerçekleştirilirse otomatik kontrol olarak söylenir. Böylece, kontrol edilecek sistem ve kontrol eden eleman olmak üzere en az iki temel birimden meydana gelen bir düzenek otomatik kontrol sistemini oluşturur[35].

5.1.Sistem

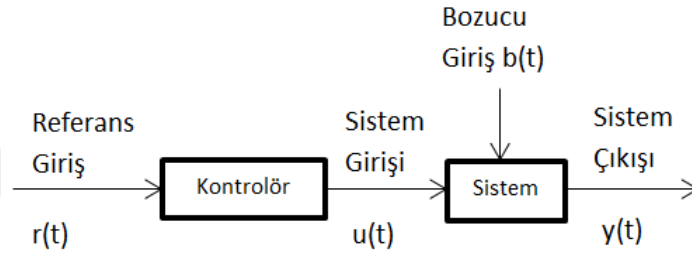
Sistem, belli bir işlevi yerine getirmek üzere birbiri ile ilgili bileşenlerin bir araya getirilmesi ile ortaya çıkan bir bütün ya da düzenektir. Sistem kavramı, genellikle belli bir amacı gerçekleştirmek amacıyla tasarlanan fiziksel bir donanım (örneğin; elektrik motoru, torna tezgâhı, uçaksavar vs.) için kullanılmakla birlikte diğer olgular (örneğin, ekonomik sistem, biyolojik sistem vs.) için de kullanılabilir[35].

5.2.Otomatik Kontrol Sistemi

Kontrol işlemlerinin, kontrol edilmek istenen olay etrafında kurulmuş bir karar mekanizması tarafından, doğrudan insan girişimi olmaksızın gerçekleştirilebilmesidir[36]. Kontrol edilecek sistemden beklenen tepki, genellikle sistemin aç kapa yöntemi ile çalıştırılıp durdurulması, sistemin çıkışının kullanıcının ayarladığı bir değerde sabit tutulması ya da kullanıcı tarafından belirlenen bir değişimi en az hata ile izlemesi olabilir. Kontrol sistemlerinin tasarımında açık çevrim ve kapalı çevrim iki farklı kontrol yöntemi izlenir[35].

5.2.1. Açık Çevrim Kontrol Sistemi

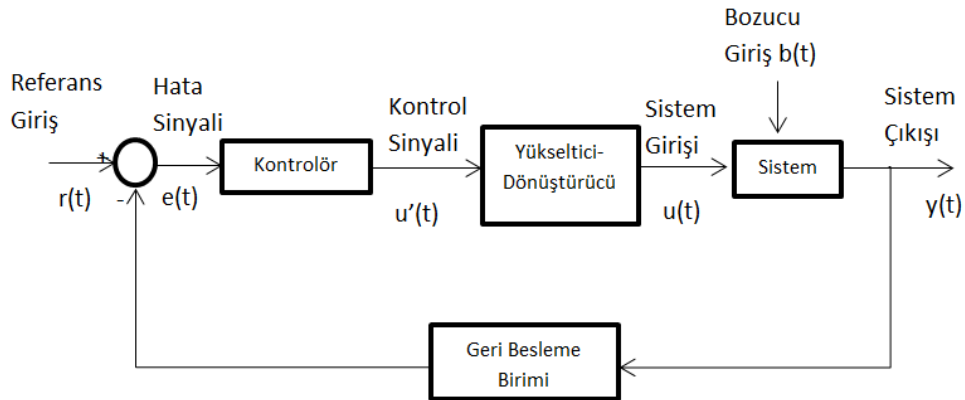
Açık çevrim kontrol, sistemin çıkışından bağımsız olarak kontrolörün kontrol sinyalini belirlediği kontrol yapılarıdır. Şekil 5.1'da blok şeması verilen açık çevrim kontrol sisteminde kontrolör, kontrol edilen sistemin arzu edilen çıkışını temsil eden referans giriş bilgisini kullanarak uygun bir kontrol sinyali üretir ve bu sinyal sistemin girişine uygulanarak sistemden beklenen cevap elde edilmeye çalışılır.



Şekil 5.1. Açık çevrim kontrol sistemi

5.2.2. Kapalı Çevrim Kontrol Sistemi

Kapalı çevrim kontrol, sistemin gerçek çıkışının ölçülerek referans giriş ile karşılaştırıldığı ve aralarındaki hata sinyaline göre kontrol algoritmasının gerçekleştirildiği kontrol sistemidir. Dolayısıyla sistemin gerçek çıkışı, referans giriş değeri ile birlikte sistemin cevabı üzerinde etkilidir[35].



Şekil 5.2. Açık çevrim kontrol sistemi

Kapalı çevrim kontrol sisteminde, kontrol edilen sistemin gerçek çıkış sinyalini ölçen ve bu sinyali, referans giriş sinyali ile karşılaştırmak amacıyla uygun bir seviyeye

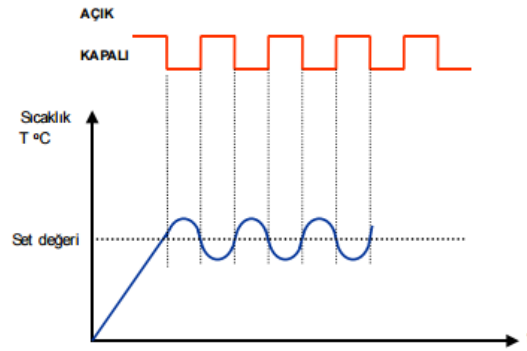
yükselten bir geri besleme birimine ihtiyaç duyulur. Şekil 5.2’de verilen blok şemadan, kapalı çevrim kontrol sisteminin, kontrolör, kontrol edilen sistem, geri besleme birimi yada algılayıcı ve yükseltici/düşürücü birimi olmak üzere 4 temel birimden meydana geldiği görülmektedir[35].

5.3. Temel Kontrol Yapıları ve Kontrol Etkileri

Hata sinyalinin kontrolörde işleyiş şekline göre kontrolörler genel olarak iki konumlu (aç-kapa çıkış üreten) ya da sürekli çıkış üreten kontrolörler şeklinde iki ana gruba ayrılabilir[35].

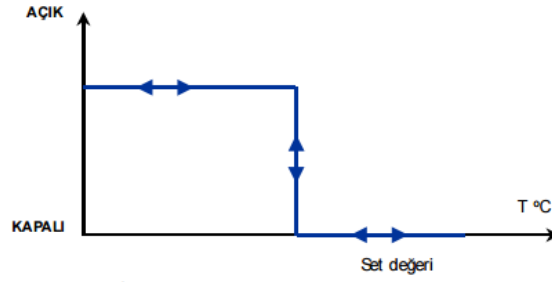
5.3.1 Açık-Kapalı kontrol (on-off)

Bu kontrol yönteminde sistem referans değer etrafında ya tamamen açıktır ya da tamamen kapalıdır. Örneğin bir sıcaklık kontrolünde referans değer 50 derece ise ısıtıcı 50 dereceye gelene kadar ısıtır, sıcaklık 50 olduğunda sistem kapanır sıcaklık bir miktar daha yükselir. Sıcaklık tekrar 50 nin altına indiğinde ısıtıcı tekrar çalışır. Bu şekilde set değeri etrafında bir salınım meydana gelir [37].



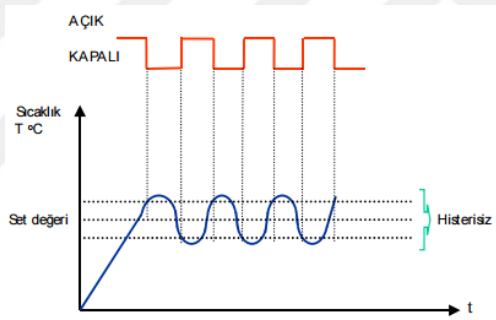
Şekil 5.3. Açık-Kapalı Sıcaklık Kontrol (İdeal)

Bu tip bir kontrolün ideal transfer eğrisi ise Şekil 5.4’ de görülmektedir.



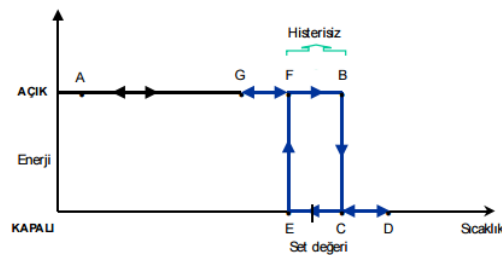
Şekil 5.4. İdeal Açık-Kapalı Kontrol Transfer Eğrisi

Fakat gerçek sistemlere bakıldığında süreçteki bozucu etkenler ve elektriksel gürültüler nedeniyle bu tip ideal aç-kapa sistemi kullanılmaz. Set değer geçişleri idealdeki gibi olursa sistem sürekli set değer etrafında salınıma girer. Bu da ısıtıcının veya kontrol edilen sistemin fazlaca aç kapa yapmasına ve ömrünün kışalmasına neden olur. Pratikte bu durumun önüne geçmek için set değeri geçişlerinde “histerisiz” ya da “sabit band” oluşturulur. Şekil 5.5’te histerisizli açık-kapalı kontrol eğrisi görülmektedir[37].



Şekil 5.5. Histerisizli Açık-Kapalı Kontrol Eğrisi

Bu eğriden de anlaşılacağı üzere sıcaklık artarken referans değeri gördüğünde hemen sistem kapanmaz izin verilen histeresiz değeri kadar daha yükselir ondan sonra kapanır. Aynı şekilde referans değere düştüğü zaman da hemen açma yapmaz bir miktar daha düşme yapar. Bu şekilde sistemin sık aç-kapa yapması engellenmiş olunur. Şekil 5.6’daki histerisizli açık-kapalı kontrol formu transfer eğrisini göstermektedir.



Şekil 5.6. Histerisizli Açık-Kapalı Kontrol Transfer Eğrisi

Yukarıdaki örnekte bu kontrol formu sıcaklık değişkeni için incelenmiş olmasına karşın bu değişken yerine basınç, sıvı seviye, sıvı akış gibi değişkenler de düşünülebilir. Sistemlerde yaygın olarak kullanılmasına rağmen açık-kapalı kontrolün yetersiz kaldığı süreç kontrollerde oransal ya da diğer kontrol formlarına geçilir [37].

Bu sistemin sakıncaları:

- Ölçülen değerler, hiçbir zaman referans değerinde sabit kalmaz, sürekli salınım meydana gelir.
- Sürekli sistem tam açık ve tam kapalı olduğundan enerji tüketimi fazladır.
- Hızlı ve önemli süreç denetimlerinde tamamen yetersizdir [36].

5.3.2 Oransal (P) Kontrol

Oransal kontrolde, sistemden alınan ölçüm bilgisi ile set değerinin farkı alınarak bulunan hata değeri ile orantılı olacak şekilde bir çıkış üretilir ve sürekli bir çalışma durumu söz konusudur. On/off da olduğu gibi sistem ya açık ya da kapalı değildir. O anki ihtiyaca göre sistem değeri sürekli değişim içerisindedir. Hata sinyaline göre oluşturulan çıkış değeri, güç elemanının enerjisini kontrol etmek amacı ile sürücü elemana bilgi gönderir. Geri besleme yani ölçme elemanı sürekli ölçme yaptığı için sistem de sürekli olarak kontrol edilmiş olunur. Sistemin set değerinde bir sapma olduğu anda ölçme elemanı bunun karşılığı olan elektrik sinyalini kontrolöre gönderir. Kontrolör bu bilgiyi referans değer ile karşılaştırarak sürücü elemanı uyarır[38]. Oransal kontrolörlerde kontrol çıkışı $u(t)$ ile kontrolör girişi $e(t)$ arasında sabit bir oransal ilişki vardır.

$$u(t)=K_p.e(t) \quad (5.1)$$

Oransal kontrolörler statik bir işlevi gerçekleştirdiğinden giriş ve çıkış sinyallerinin dalga şekli değişmeyecek ve sadece oransal kazanca bağlı olarak hata sinyali kuvvetlendirilecek ya da zayıflatılarak sisteme uygulanmış olacaktır[35].

Oransal kontrolörler, doğrudan sistemin kazancını artırdığından oransal kazanç artırılarak sistemin cevap hızı artırılacağı gibi kalıcı durum hatası da azaltılabilir. Ancak oransal kontrolör, sistemin tipini değiştirmedığından bir sistemin kalıcı durum hatası hiçbir

zaman sifira getirilemez. Referans deęer ile sistemin yerleřtięi ve sabit kaldıęı sıcaklık arasındaki farka offset denir[37].

5.3.3 Integral (I) Kontrol

İntegral kontrolörler, hata sinyalinin integralini alır ve ayrıca belirli bir kazançla yükselterek/zayıflatarak kontrol girişini belirler.

$$u(t) = K_i \int_0^t e(t) dt \quad (5.2)$$

Burada K_i integral kazancıdır. İntegral etkisi nedeni ile bu tür kontrolörlerde kontrol giriři, hata sinyalinin zamana göre taradıęı alanla iliřkili olacaktır. Hata giriři sıfır olmadığı sürece integral kontrolör, kontrol girişini arttırmaya/azaltmaya devam edeceęinden kalıcı durum hatasını sıfır yapacak şekilde bir kontrol giriři üretir. Bu kontrolörler, temel görev olarak kalıcı durum hatalarını düzeltmekle birlikte bazı sistemlerde geçici rejim kriterleri üzerinde de etkili olabilir[35].

5.3.4 Türev (D) Kontrol

Türev kontrolörler, hata sinyalinin türevine, dięer bir ifade ile deęişim hızına ve yönüne baęlı olarak kontrol girişini üreten kontrolörlerdir.

$$u(t) = K_d \frac{d}{dt} e(t) \quad (5.3)$$

Türev etkisi nedeni ile bu tür kontrolörlerde kontrol giriři, hata sinyalinin zamana göre eęimleri ile iliřkili olacaktır. Hatanın hızlı deęiřtięi anlarda büyük bir kontrol giriři belirleyerek sisteme uyguladıęı için hata oluşmadan düzeltme etkisi yapar ve dolayısıyla genel olarak sistemin geçici rejim kriterlerini düzeltme etkisi vardır. Ancak hatanın sabitleřmesi durumunda türev kontrolün çıkıřı sıfır olacaęından sabit kalıcı durum hatalarını düzeltme etkisi yoktur.

Türev kontrolörler, yalnızca hatanın değişimi sırasında etkili olduğundan kontrol sistemlerinde tek başına kullanılmaz ve genellikle diğer kontrolörlerle, örneğin en azından oransal kontrolörle kullanılabilir[35].

5.4. Otomatik Kontrol Formları

Otomatik kontrol mantığına göre kontrol elemanı olarak kullanılacak olan kontrol sistemi ne olursa olsun girilen set değerine karşı sistemi sürekli ve hassas bir şekilde kontrol etmesi gerekir. Süreç kontrolde gerekli olan bu hassas çalışmayı gerçekleştirebilecek ve hata değerini minimuma indirebilecek farklı kontrol biçimlerivardır [37]. Bunlar:

- 1) Açık-Kapalı kontrol (on-off)
- 2) Oransal kontrol (proportional) P
- 3) Oransal+İntegral kontrol (proportional+integral) P+I
- 4) Oransal+Türevsel kontrol (proportional+derivative) P+D
- 5) Oransal+İntegral+Türevsel kontrol (proportional+integral+derivative) P+I+D

formlarıdır. Bireysel olarak oransal, integral ve türevsel kontrolörlerin etkileri incelenmişti. Çoğu zaman bu kontrolörler tek başlarına kontrol sisteminden beklenen performans kriterlerini karşılayamaz. Bu nedenle temel kontrol etkilerinin birleşiminden meydana gelen PI, PD, PID kontrolörler kullanılır[35].

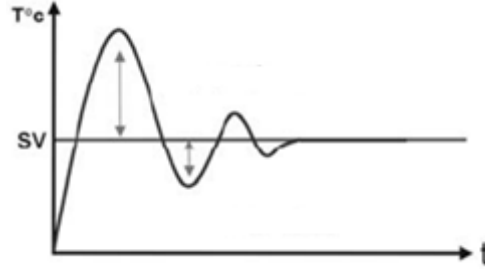
5.4.1 Oransal + İntegral (PI) Kontrol

PI kontrolör, oransal ve integral etkilerinin birleşiminden oluşan kontrolör yapısıdır ve kontrol girişi $u(t)$,

$$u(t)=K_p e(t)+K_i \int_0^t e(t)d(t) \quad (5.4)$$

Oransal ve integral kontrol etkileri dikkate alındığında genel anlamda PI kontrolörün, sistemdeki kalıcı durum hatalarını düzeltereği ancak sistemin bağıl kararlılığını azaltacağı

söylenbilir. Ancak genel olarak sistemin geçici rejim kriterleri üzerinde olumsuz bir etkiye sahiptir[35].



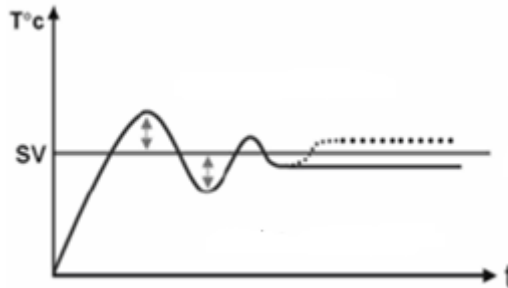
Şekil 5.7. Oransal+İntegral Kontrol Eğrisi

5.4.2 Oransal+Türevsel (PD) Kontrol

PD kontrolörler, oransal ve türev kontrolörlerin birleşiminden meydana gelir ve kontrol sinyali $u(t)$ aşağıdaki gibi hesaplanır.

$$u(t) = K_p e(t) + K_d \frac{-d}{dt} e(t) \quad (5.5)$$

Bu kontrolör kullanılarak, sistemin geçici rejim kontrol taleplerini karşılamak üzere tasarlanabileceği, kalıcı durum hataları üzerinde doğrudan bir olumlu etkisi bulunmamakla birlikte geçici rejimin iyileşmesi sonucunda daha yüksek kontrol kazancı seçilebileceğinden kalıcı durum hatalarını da belirli ölçüde azaltabileceği söylenebilir. Ayrıca PD kontrolörler, türev etkisi nedeni ile gürültüyü de yükselteceğinden özellikle gürültülü sistemlerde pratik olarak kullanılmaları sorunludur[35].



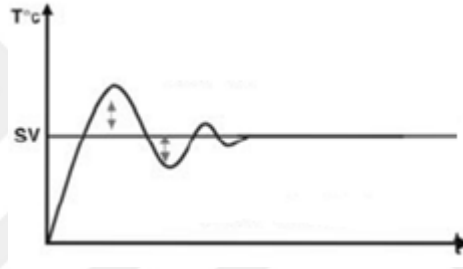
Şekil 5.8. Oransal+Türevsel Kontrol Eğrisi

5.4.3 Oransal+İntegral+Türevsel (PID) Kontrol

PID kontrolörler üç temel kontrolörün birleşiminden meydana gelir ve kontrol sinyali $u(t)$,

$$u(t)=K_p e(t)+K_i \int_0^t e(t)dt+K_d \frac{-d}{dt} e(t) \quad (5.6)$$

Olarak hesaplanır. Bu kontrolör genel anlamda hem geçici hem de kalıcı durum kontrol kriterleri üzerinde etkilidir. Denklem 5.6'ya bakıldığında K_i katsayısı sıfır yapıldığında PD, K_d katsayısı sıfır yapıldığında ise kontrolörün PI kontrolör olabileceği görülmektedir[35].



Şekil 5.9. Oransal+İntegral+Türevsel Kontrol Eğrisi

5.5. Bulanık Mantık (fuzzy) Kontrol

Endüstriyel uygulamalarda daha basit yapılı olması nedeni ile PID kontrol yöntemi daha sık tercih edilen bir yöntemdir [39-41]. Fakat bu yöntemin de bazı olumsuz yönleri vardır. Bunlar sistemin matematiksel modeline gereksinim duyulması ve non-lineer sistemlerde uygun sonuçlar verememesidir. Bu sebepten dolayı özellikle non-lineer sistemlerde başarılı sonuçlar verebilen Bulanık mantık yöntemi kullanılmaya başlanmıştır. Ayrıca bu tip sistemlerde bulanık mantığın yanı sıra durum geri besleme, yapay sinir ağları ve kayma kipli kontrol yöntemleri de başarılı sonuçlar veren kontrol yöntemleridir [42,43].

Bulanık mantık kontrolü ilk olarak 1964 yılında Azeri bilim adamı Prof. Dr. Lotfi A.Zadeh tarafından oluşturulmuş bir kontrol yöntemidir. Genel mantıkta sadece sıfır ve bir seviyeleri yerine bulanık mantıkta giriş ve çıkış değerlerinin birden fazla üyelik fonksiyonlarına üye olabildiği bir kontrol yöntemidir. Bulanık mantık kontrol yapılabilmesi için öncelikle net olan girişlerin bulandırma biriminden geçirilerek girişlerin

bulandırılması gerekmektedir. Bu işlem için giriş üyelik fonksiyonları kullanılır. Bulanırma işleminden sonra uzman kişiler tarafından oluşturulan kural çıkartımı yapılmaktadır. Oluşturulan kurallara göre sistemin çıkışı belirlenir ve kontrolü sağlanır [44].

Klasik mantıkla çalışan sistemler, belirsizlikle ilgilenmezler. Sadece doğruluk değerleri doğru ya da yanlış olan önermelerle ilgilenirler. Bulanık mantık ise belirsizliklerle çalışabilmesi için kurulmuş bir matematik düzendir. Bulanık mantık, günlük hayatta sıkça kullanılan belirsiz sözel ifadeleri sayısal olarak tanımlamaya da elverişlidir.

Bulanık Mantığın sağladığı avantajlar:

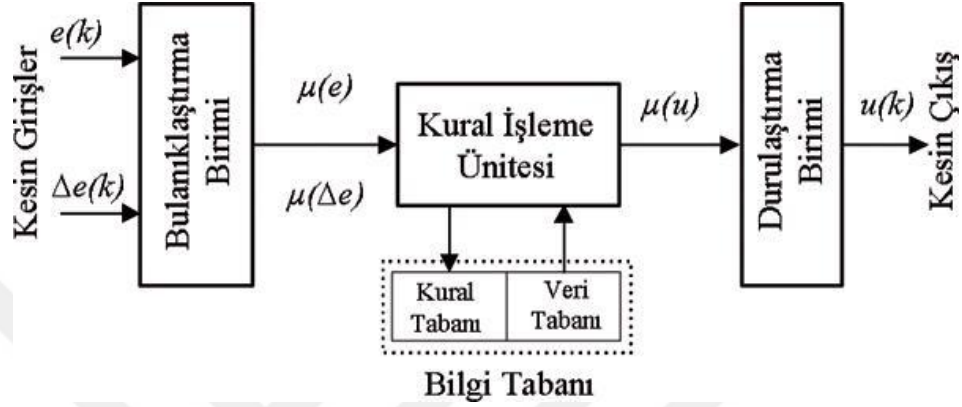
- 1- İnsan mantığına ve düşünce tarzına yakındır.
- 2- PID kontrolde olduğu gibi matematiksel bir modele ihtiyaç duymaz.
- 3- Yazılımın basit olması
- 4- Bulanık Mantık kavramını anlamak kolaydır.
- 5- Üyelik değerlerinin kullanımı sayesinde, diğer kontrol tekniklerine göre daha esnektir.
- 6- Keskin arz etmeyen bilgiler kullanılabilir.
- 7- Doğrusal olmayan fonksiyonların modellenmesine izin verir [45].

Uygulama alanları;

- Otomatik Kontrol Sistemleri
- Bilgi Sistemleri, Uzman Sistemler,
- Görüntü İşleme
- Optimizasyon
- Yönetim ve Karar Destek Sistemleri...[46]

5.5.1. Bulanık Sistemin Çalışması

Bulanık sistem, bulanık olmayan ve gerçek uygulamalarda kullanılan kesin girişleri şekildeki temel birimde işleyerek yine sonuçta bulanık olmayan ve gerçek uygulamalarda kullanılabilir bir çıkış üretir.



Şekil 5.10. Bulanık sistemin yapısı

5.5.2. Bulanık Sistemin Birimleri

5.5.2.1. Bulanıklaştırma Birimi:

Bulanık sistemin girişlerinin kesin değerlerini, üyelik fonksiyonu yardımıyla bulanık değerlere dönüştürür. Dolayısıyla giriş değişkenini alacağı her bir değer için bir üyelik derecesi hesaplar. Bu üyelik fonksiyonları tasarlanacak sisteme göre sigmoidal, pi, üçgensel ya da yamuk üyelik fonksiyonu olarak seçilebilir.

5.5.2.2. Kural Tabanı:

Sistem hakkındaki bilgi sahibi olan uzman kişilerin bilgisi esas alınarak, sisteme gelen girişler karşısında nasıl bir çıkış üretilmesi gerektiğini belirleyen IF (Koşul) THEN (çıkış) kurallarından oluşur. Koşul kısmındaki birden çok giriş AND, OR ya da NOT gibi operatörlerle bağlanabilir.

5.5.2.3. Çıkarım Birimi:

Çıkarım mekanizması, bulanıklaştırma biriminin çıkışlarını yani girişlerin üyelik derecelerini ve kural tabanını kullanarak yeni bir bulanık küme elde eder. Çıkarım biriminde 3 temel işlem gerçekleştirilir.

a-) Her bir kuralın kesinlik derecesini hesaplama: Eğer kural tablosunda satır “i” ile ve sütun “j” ile gösterilirse kural tablosunda kuralların kesinlik dereceleri girişlerin üyelik dereceleri kullanarak aşağıdaki gibi belirlenir.

$$\mu_{ij}^{KD} = \mu_{ij}^{x1}(x_1^*) \cdot \mu_{ij}^{x2}(x_2^*) \quad (5.7)$$

Kuralların kesinlik derecelerinin hesaplanmasında genellikle min ya da cebirsel çarpım kullanılır.

b-) Her bir kuralın kesinlik derecesi ile ilgili kuralın çıkış üyelik kümesi arasındaki İMA işlemi: İma işleminde de genellikle Min ya da cebirsel çarpım kullanılır ve İMA işlemi sonunda elde edilen imalı bulanık çıkış kümesi aşağıdaki gibi ifade edilir.

$$\mu_{ij}(u) = \min(\mu_{ij}^{KD}, \mu_{ij}^U(u)) \quad (5.8)$$

c-) Toparlama işlemi: İMA işlemi sonunda elde edilen bulanık çıkış kümesi, toplama işlemine tabi tutulabilir. Toparlama işleminde ima edilen bulanık çıkış kümelerine max operatörü uygulanarak sonuç ya da toplanmış bulanık çıkış kümesi elde edilir[47].

$$\mu_{out}(u) = \max(\mu_{11}(u), \dots, \mu_{32}(u), \dots, \mu_{43}(u)) \quad (5.9)$$

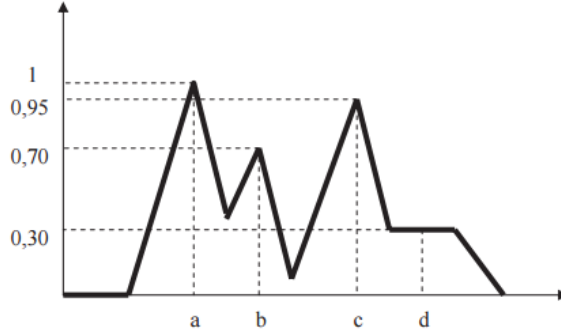
5.5.2.4. Durulaştırma (Berraklaştırma) Birimi:

Çıkarım birimden elde edilen bulanık sonuç çıkış kümesini, bulanık olmayan ve uygulamada kullanılacak gerçek değerlere dönüştürür. 4 adet durulaştırma yöntemi vardır.

- Maksimum üyelik yöntemi
- Ağırlık merkezi yöntemi
- Ağırlık ortalaması yöntemi

- Maksimum üyelikleri ortalama yöntemi

Ağırlık ortalaması yönteminde toplama işlemine gerek yoktur ve doğrudan ima edilmiş bulanık çıkış kümelerine uygulanmaktadır.



Şekil 5.11. Ağırlıklı ortalama yöntemi ile durulaştırma

Yöntemin kullanılabilmesi için ön koşul simetrik üyelik fonksiyonlarının kullanılmış olmasıdır. Çıkarım kümesini oluşturan her bir üyelik fonksiyonunun en büyük üyelik derecesinin bu değere ulaştığı nokta ile çarpılarak bulunur ve A10 eşitliğindeki gibi ifade edilir.

$$du(k) = \frac{\sum_{i=1}^n \mu_{u_i} u_i}{\sum_{i=1}^n \mu_{u_i}} \quad (5.10)$$

Şimdiye kadar anlatılan bulanık denetleyici yapısı standart bulanık denetleyici veya bazen de “Mamdani” tipi bulanık denetleyici olarak isimlendirilir. Buna ek olarak sıklıkla kullanılan diğer denetleyici türü de “Sugeno” tipi denetleyicidir. Bu denetleyiciyi Mamdani tipinden ayıran en büyük fark kurallarla gösterilen çıkışların bulanık kümelerle değil de giriş değişkenlerinin fonksiyonları türünden ifade edilmesidir. Sugeno tipi bir denetleyicide kurallar IF(x1 is A and x2 is B) then y=f(x1,x2) şeklinde ifade edilir. Burada f(x) giriş değişkenlerinin doğrusal veya doğrusal olmayan bir fonksiyonudur. Çıkışlar direk olarak bir fonksiyonla hesaplandığı için bu tip denetleyicide durultucuya ihtiyaç yoktur[65].

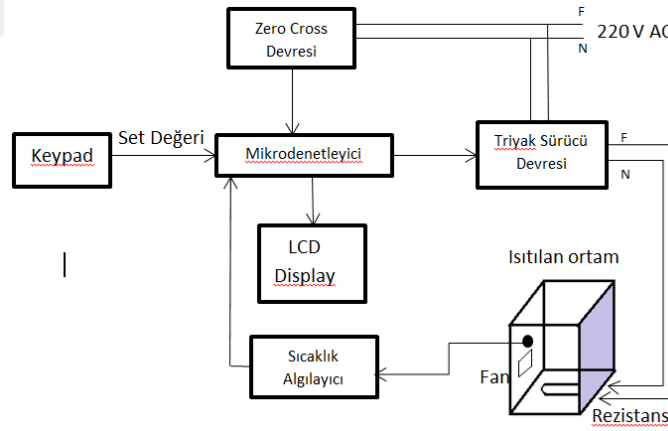
Bu tez de Mamdani tipli bulanık mantık ve durultucu biriminde ise ağırlık ortalaması yöntemi kullanılmıştır.

6. MİKRODENETLEYİCİ KONTROLLÜ ISIL SİSTEM TASARIMI

6.1. Mikrodenetleyici ile Sıcaklık Kontrolü

Sıcaklık, bir cismin üzerine aldığı ısıнын değeri belirten özelliktir. Bir cisim üzerine ısı alırsa sıcaklığı yükselir, ısı verirse sıcaklığı azalır. Birimi santigrad derecedir($^{\circ}\text{C}$). Kontrol açısından düşük örnekleme frekansları ile kontrol edilebilen fiziksel bir büyüklüktür. Bu kontrol esnasında sürece ısı çıkışı sağlayan rezistans ısıtıcı vb. bir aygıt kullanılır. Ortamın sıcaklığını elektriksel büyüklüğe çevirerek mikrodenetleyiciye aktarmak için bir sıcaklık algılayıcısı kullanılır. Bu geri bildirim üzerine, mikrodenetleyici ısıtıcıyı kontrol eder (bazı sistemlerde sistem fanını da)[48].

Bu çalışmada tasarlanan sıcaklık kontrol sisteminin basit bir diyagramı şekil 6.1'de gösterilmiştir.



Şekil 6.1. Mikrodenetleyicili sıcaklık kontrol genel diyagramı

6.2. Sistem Mimarisi

Şekil 6.1'de genel diyagramı verilen mikrodenetleyicili sıcaklık kontrol sisteminin mimarisi, diyagramda gösterilen bileşenlerden meydana gelmektedir. Bu diyagramdaki donanımsal bileşenler mikrodenetleyici yazılımı ile yönetilmektedir.

Sayısal sıcaklık kontrolünün en kolay biçimi on/off tipi bir kontrol kullanmaktır. Bu kontrol formu hemen hemen tüm bina içi termostatlarda kullanılmaktadır[48]. On/off tipi kontrol, basit ve ucuz bir kontrol algoritması olmasına karşın bu çalışmada diğer kontrol

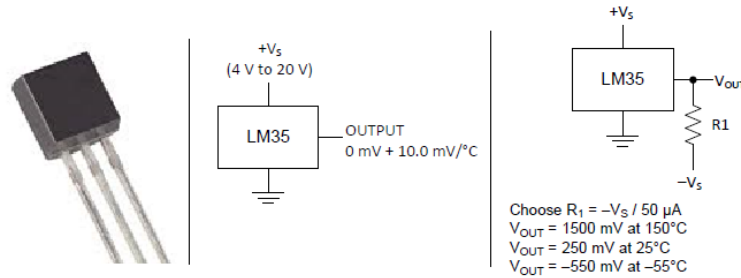
formları da kullanılmıştır. Sürekli değişken kontrol sistemleri set edilen sıcaklık değeri ile ölçülen gerçek sıcaklık değeri arasındaki farkı ölçer ve bu değeri kullanarak ısıtıcıya ne kadar güç aktarılacağını belirler. Bu sayede sistemin hem sürekli aç-kapa yapılması önlenir hemde ısıtıcıya daha az güç aktarımı sağlanarak enerji tasarrufu sağlanır.

6.3. Mimari Bileşenleri

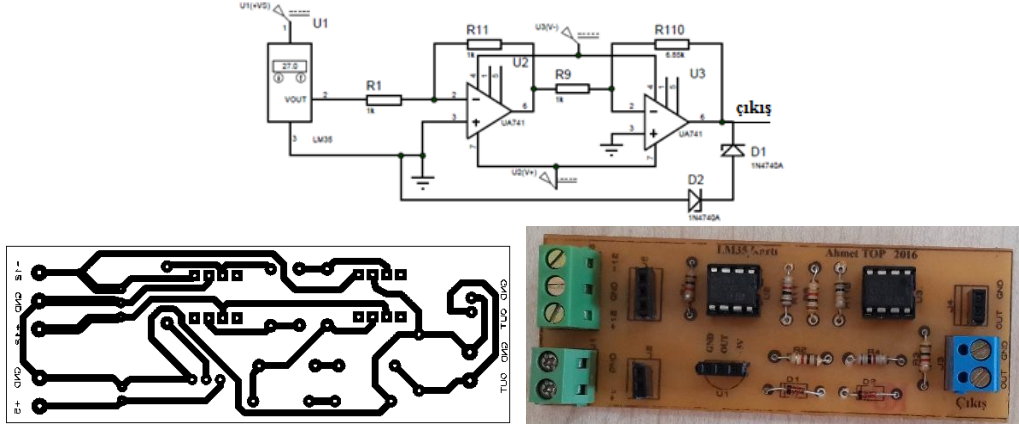
6.3.1 Algılayıcı

Sıcaklık ölçümü için çeşitli tipte algılayıcılar mevcuttur. Isılçift (thermocouple), PT100, termistör ve RTD ler gibi klasik algılayıcılar geniş biçimde kullanım alanı bulmaktadır. Entegre devre biçimindeki algılayıcılar ve uzaktan sıcaklık ölçüm (radiation thermometry) aygıtları gibi yeni nesil algılayıcılar sadece sınırlı uygulamalar için kullanım alanı bulsalar da popüler olmaktadır[48].

Bu çalışmada Texas Instruments firmasının LM35 kodlu analog çıkışlı sıcaklık algılayıcısı kullanılmıştır.



Sıcaklık algılayıcı, 4-20 V DC çalışma aralığına sahip 3 bacaklı bir algılayıcı olup, -55°C ile 150°C sıcaklıkları arasında derece başına 10 mV'luk bir hassasiyetle ölçüm yapmaktadır[2]. Bu algılayıcı bağlantı şeklimize göre 0 ile 10 mV arasında analog çıkış vermektedir. Ancak kontrolü endüstriyel alanlarda kullanılan transmitterlere uygun yapmak istediğimizden dolayı sensör çıkışını 0-10 V endüstriyel standartlara dönüştürmek için sensör çıkışına operational amplifier ile yükseltici bir devre yapılmıştır.



Şekil 6.3. Yükseltici devresi

Kullanılan mikrodenetleyicinin pinlerine 3.3 V'dan fazla gerilim uygulanamadığı için gerilim bölücüler yardımı ile 0-10 V DC 0-3.3 V DC'ye dönüştürülerek mikrodenetleyicinin analog giriş pinine uygulanmıştır.

6.3.2. Isıtıcı

Bu çalışmada kullanılan ısıtıcı, kuru havada ısıtma yapmaktadır. 220 V AC gerilim ile çalışmaktadır ve nominal gerilim altında 1050 W gücünde çalışmaktadır. 46 ohm direnç karakteristiğine sahiptir ve 26x16 cm boyutlarında kıvrımlı tüp rezistanstır.

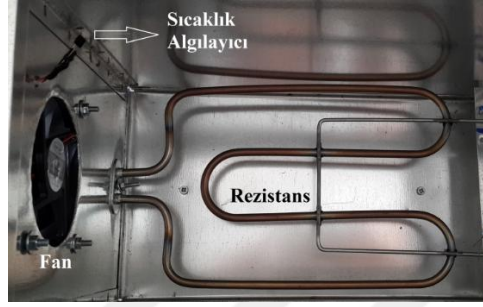


Şekil 6.4. Rezistans

6.3.3 Isıtılan Ortam

Sistem tepkisinin gözlemlenebilmesi için 1 mm kalınlığındaki sac levhalardan 17x30x31 cm boyutlarına sahip dikdörtgen prizması biçiminde bir yapı seçilmiştir. Bozucu etkilerin sistem üzerindeki sonuçlarını da gözlemleyebilmek için ortam yalıtılmamıştır.

Prizmanın tabanından 4 cm yüksekliğe rezistans yerleştirilmiş ve gerilim uygulanabilmesi için uçları dışarı çıkartılmıştır. Sistemin soğutmasında kullanılacak olan fan prizmanın yan yüzeyine 14 cm yüksekliğe monte edilmiştir. Ayrıca kullanılan sensör de fanın yan tarafına monte edilmiştir. Rezistans ve fanın montajı ve herhangi bir arıza durumunda değişilebilmesi için üst kısım kapaklı olarak yapılmıştır.



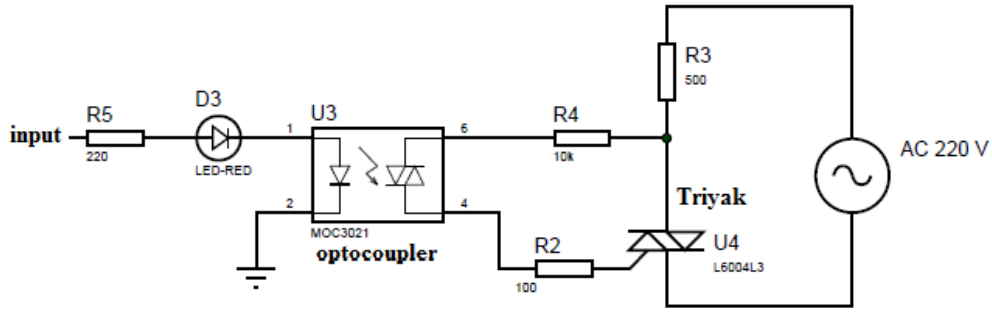
Şekil 6.5. Isıtılan ortamın iç kısmı

6.3.4. Mikrodenetleyici

Mikrodenetleyici olarak bölüm-2 de özellikleri belirtilen Atmel firması tarafından geliştirilmiş olan AT91SAM3X8E mikroişlemci tabanlı bir geliştirme platformu olan Arduino Due kullanılmıştır. Yazılım, Arduino'nun C tabanlı kendi arayüz programından programlanarak USB ile mikroişlemciye yüklenmektedir. Beslemesini USB den aldığından çalışması için ek bir beslemeye ihtiyaç duymamaktadır.

6.3.5. Dimmer Devresi

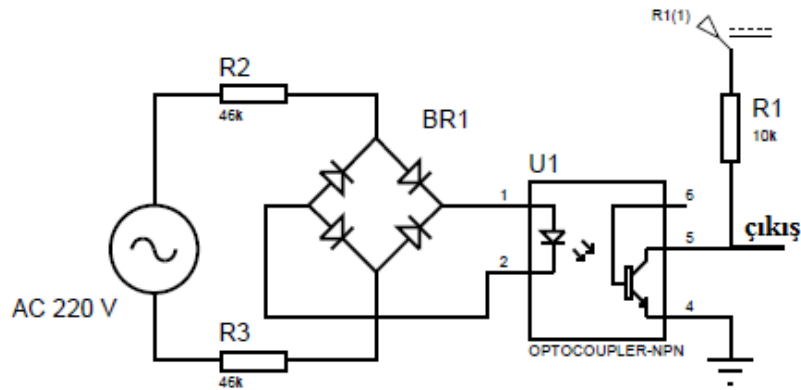
Dimmer devresi, tek fazlı bir tetikleme devresidir. Dimmer devresinde yarım dalga denetimi için bir tane tristör kullanılırken, tam dalga denetiminde iki adet tristör ters-paralel olarak bağlanabilir[49]. Bu çalışmada, rezistansın gerilim ayarı için tam dalga şebeke gerilimi denetlenmiştir ve bunun için NXP Semiconductors firmasının BT138 serisi triyaklarından BT138-600 kullanılmıştır. Bu triyak maksimum 600 V 12 A' e dayanabilmektedir ve motor kontrol, evdüstriyel ve ev aydınlatmalarında, ısıtma kontrolünde vb. alanlarda kullanılabilir[50]. Arduino ile AC gerilimi izole etmek için triyak tetiklemede Texas Instruments firması ürünü olan moc3021 optocoupler entegresi kullanılmıştır. Bu entegre 400 V fototriyak sürme çıkışına sahiptir. Kontrol sürme girişi ise 5 V DC ile yapılabilir[51].



Şekil 6.6. Dimmer devresi

6.3.6. Zero Cross Devresi

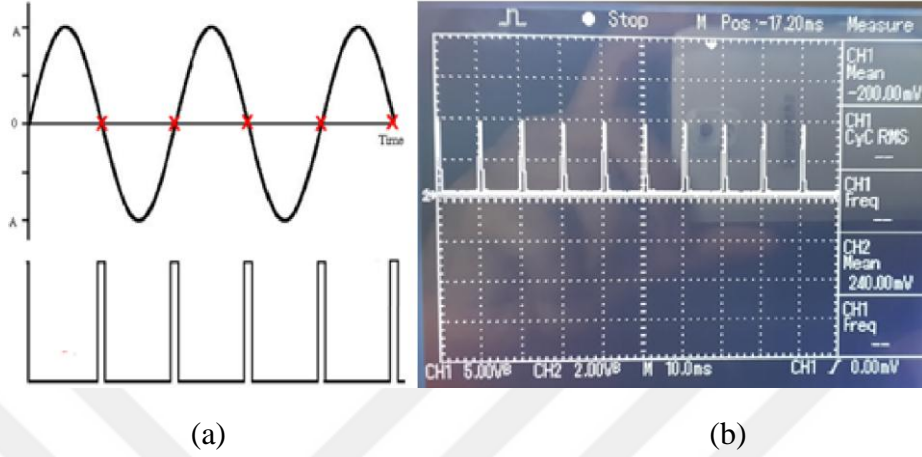
Tristör/triyak kullanılarak yapılan kontrolde, şebeke geriliminin faz açısı kontrol edilerek çıkış geriliminin ortalama ve etkili değerleri değiştirilmektedir. Bunun için şebeke sinyalinin sıfır geçişleri belirlenmiş ve bundan da “sıfır geçiş darbeleri” elde edilmiştir[52]. Irmak E., Calpbınici A., Güler N. yaptıkları çalışmada sinüsoidal akım ve gerilim arasındaki gecikmeyi bulmak için akım ve gerilim sinyallerini sıfır geçiş dedektöründen geçirerek aradaki zaman farkını belirlemişlerdir[53]. Bu çalışma için yapılan sıfır geçiş dedektörü devresi Şekil 6.7’de gösterilmiştir.



Şekil 6.7. Sıfır geçiş dedektörü devresi

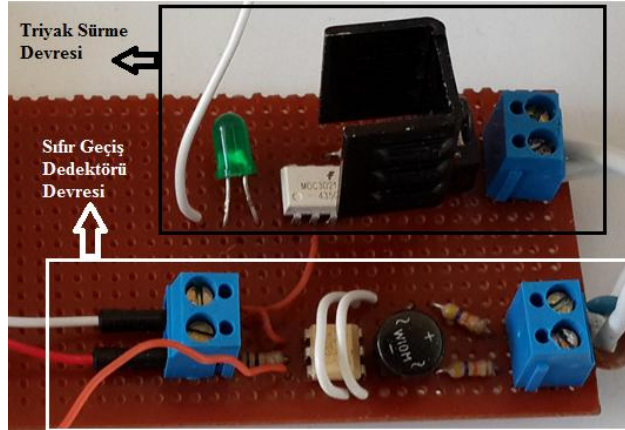
Devrede sinüsoidal gerilimi tam dalga doğrultmak için Micro Commercial Components firmasının ürettiği PB68 köprü diyotu kullanılmıştır. Bu entegre, 560 V etkin voltaja, 800 V peak voltajına sahiptir, ortalama 6 A ‘e kadar çalışabilmektedir ve 8.3 ms 150A peak akımına dayanabilmektedir[54]. AC ve DC sinyalleri izole ederek sıfır geçiş noktalarını almak için ise Texas Instruments firmasının ürettiği 4n35 optocoupleri kullanılmıştır. Bu

entegre, yüksek voltaj elektrik izolasyonuna sahiptir, 8 ms yarı sinüs dalgasında 3.55 kV'luk gerilime dayanabilmektedir, 7 us'lik bir anahtarlama hızına sahiptir[55].



Şekil 6.8. Zero-Cross Devresi Sinyalleri

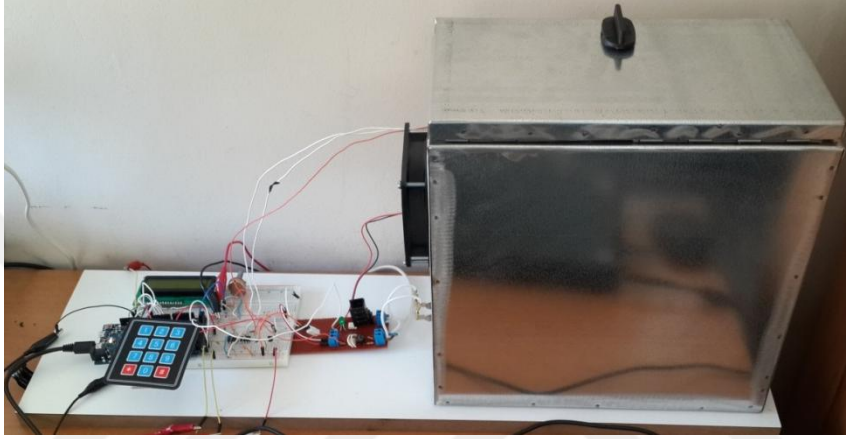
Şekil 6.8.a'da 50 Hz frekansına sahip şehir şebeke sinyalinin dalga şekli ve sıfır geçiş dedektörünün çıkış dalga şekli görülmektedir. Şekil 6.8.b'de ise devrenin osiloskop çıktısı görülmektedir. 50 Hz'lik bir sinyalin periyodu 20 ms'dir ve her sıfır noktası 10 ms'ye denk gelmektedir.



Şekil 6.9. Sıfır Geçiş Dedektörü ve Triyak Sürme Devresi Sinyalleri

6.3.7. Set Giriş ve Değer Okuma

Sisteme istenilen sıcaklık değerinin girilebilmesi için 3x4'lük keypad kullanılmıştır. Yeni set değerinin girilmesi ve onaylanması yine buradan yapılmaktadır. Sistem çalıştığında girilen set değeri ve çalışma durumunda set değeri ve anlık ölçülen değerleri gözlemleyebilmek için yaygın biçimde kullanılan Xiamen Amotec firmasının ürünü olan 2x16 'lık MD216-04 ekranı kullanılmıştır.



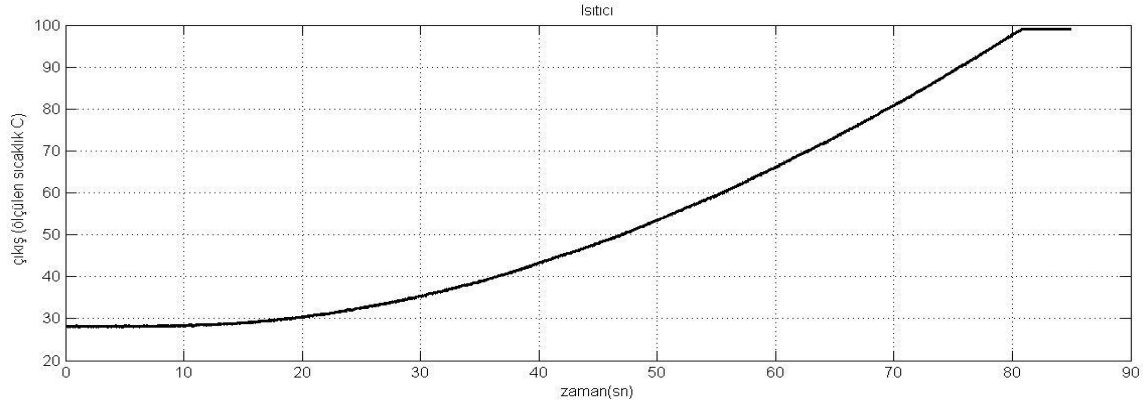
Şekil 6.10. Sistemin genel görünümü

6.4.ON/OFF, P,PI, PD, PID Denetleyicilerin Sıcaklık Kontrolündeki Etkileri

Tasarımı gerçekleştirilen sıcaklık kontrol sistemi, ortam sıcaklığı ile 100 °C sıcaklık aralığını kontrol edebilmektedir. Deneysel çalışmanın sonuçları, kullanılan mikrodenetleyicinin yazılım arayüzünde bulunan seri port ekranından alınmıştır. Bu sonuçlar ise Matlab programına aktarılarak zamana göre çizdirilmiştir.

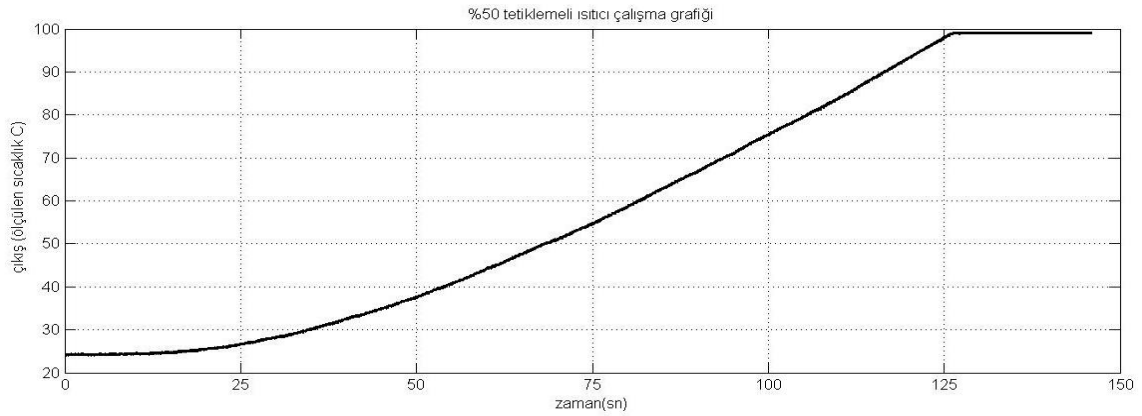
Mikrodenetleyici kullanarak oluşturulan yazılım ile on/off, oransal, oransal+integral, oransal+türev ve oransal+integral+türevsel denetleyici tipleri gerçekleştirilmiştir. Bu denetleyiciler ile çeşitli parametre değerlerinin sistem tepkisi üzerindeki etkileri incelenmiştir. Sistem kontrolünde soğutma bulunmadığından başlangıç sıcaklığı olarak ortam sıcaklığı alınmıştır. Bütün denetleyiciler de bitiş sıcaklığı (set değeri) olarak 60 °C alınmıştır.

Rezistansın tam gerilim (220 V AC) ve yarı gerilim (110 V AC) uygulandığında ısıtılan ortamdaki sıcaklığın değişim grafikleri aşağıda verilmiştir.



Şekil 6.11. Isıtıcıya 220 V giriş gerilimi uygulanmasıyla elde edilen çıkış karakteristiği

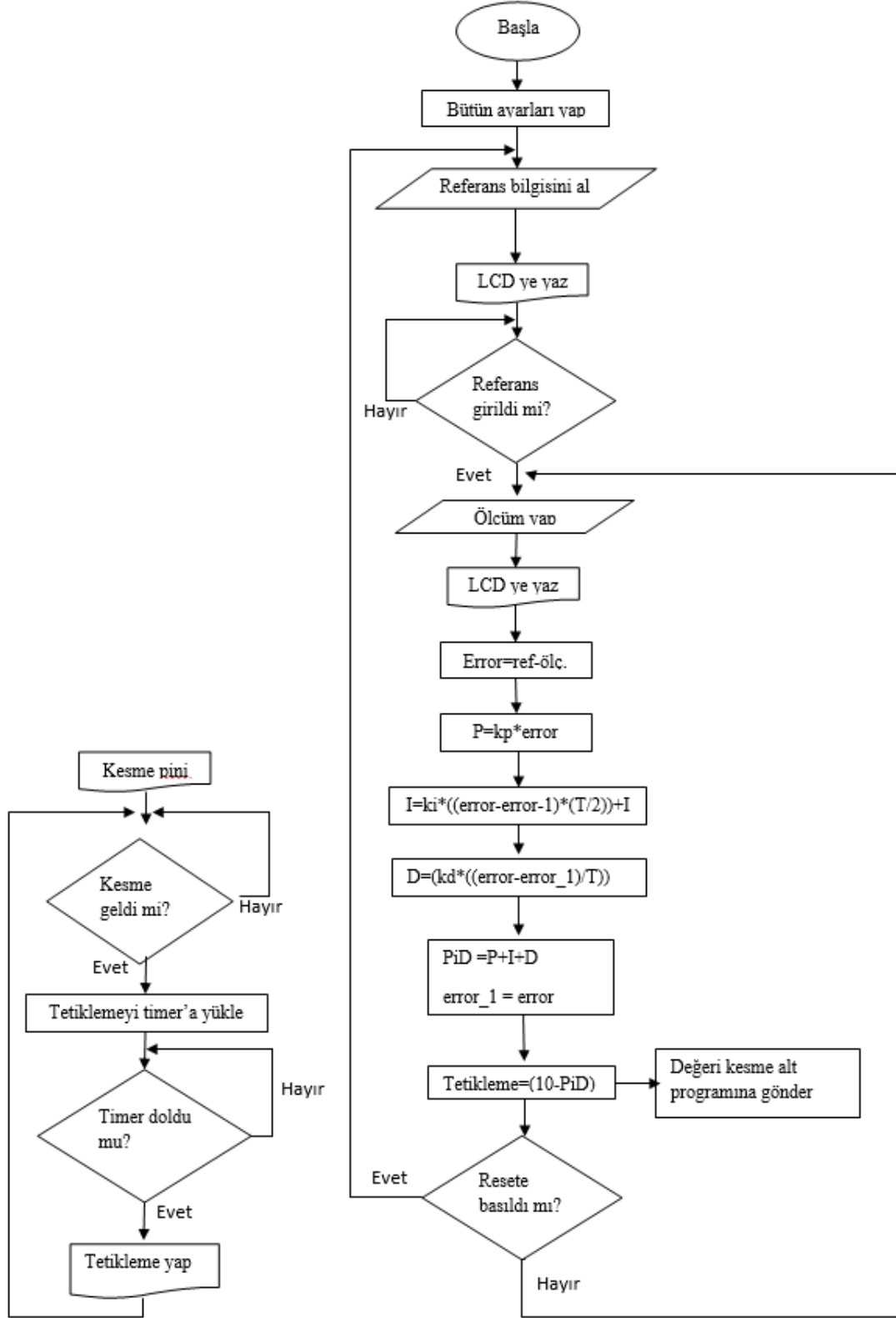
Isıtıcıya 220 V AC gerilim uygulandığında ortam ısısı 29 °C ortam sıcaklığından artmaya başlamış ve 100 °C civarında kararlı duruma ulaştığı gözlemlenmiştir. Geçici durumdan kalıcı duruma geçtiği süre, 81 s 'dir.



Şekil 6.12. Isıtıcıya 110 V giriş gerilimi uygulanmasıyla elde edilen çıkış karakteristiği

Triyak sürücü devresine 90° lik tetikleme verildiğinde ısıtıcıya uygulanan gerilim 110 V olmuştur. Bu durumda ortam sıcaklığından 100 °C'ye yaklaşık 128 s de geldiği gözlemlenmiştir. Bu duruma bakıldığında tetikleme ile gerilim ayarı yapılarak istenilen sıcaklığa daha az güç sarfiyatı yapılarak gelinebileceği gözlemlenmiştir.

6.4.1.Genel Akış Şeması

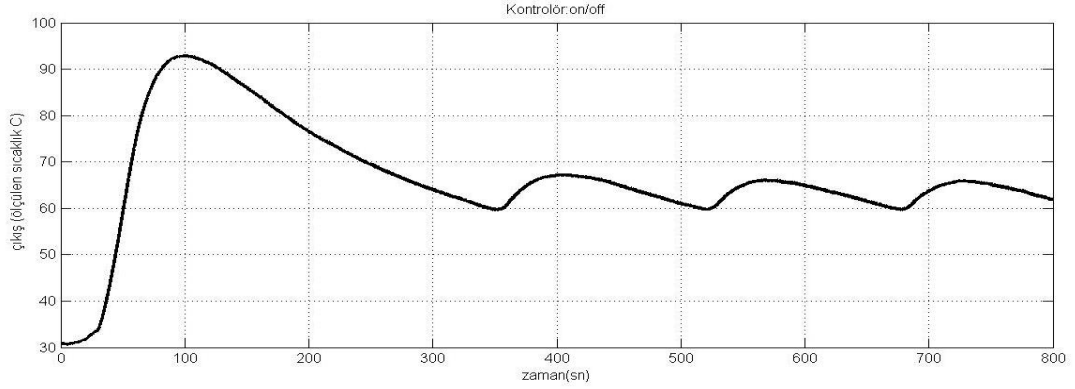


Şekil 6.13. Sıcaklık kontrol PID akış diyagramı

6.4.2. On/Off Denetleyici

On/off kontrolde ısıtıcı ya tam kapalı ya da tam açıktır. Sistemde iki adet on/off kontrol uygulaması denenmiştir.

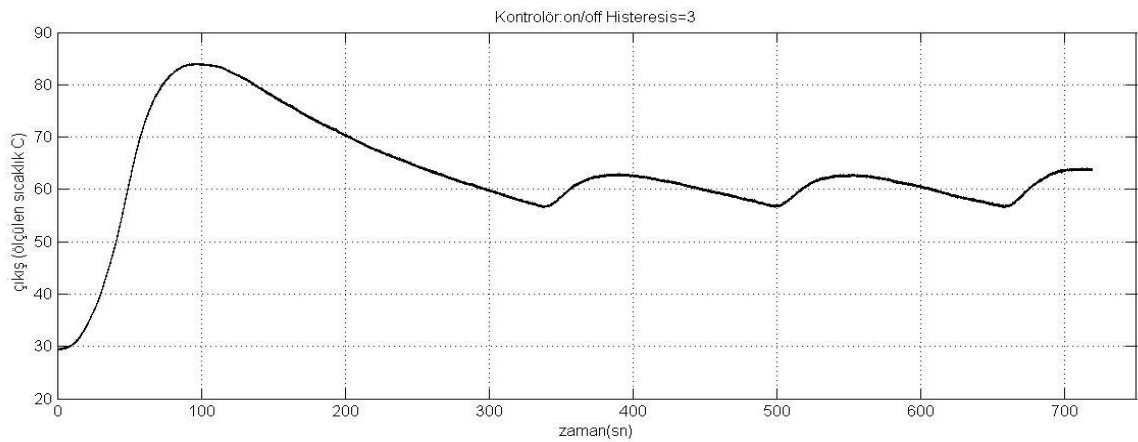
1.ölçüm: Ortam sıcaklığı 31 °C, Set değeri: 60 °C histeresis: yok



Şekil 6.14. On/off kontrolör,31-60 °C, histeresis yok

On/off kontrol daha önce de bahsedildiği gibi referans değere ulaşıldığında sistemi çalıştırır referansın altına düştüğü zaman ise sistemi durdurur. Referansa ulaştıktan sonra sistem kapatılsa bile rezistansın sıcaklığı hemen düşmeyeceğinden dolayı ortam bir miktar daha ısınır yukarıdaki şekilde referans civarında salınım yapmasının sebebi de budur. Yükselme süresi:50.66s, Maks. Aşma: 33.05°C, Tepe süresi:100.4 s, Yerleşme süresi:353 s

2.ölçüm: Ortam sıcaklığı 30 °C, Set değeri: 60 °C histeresisiz: 3

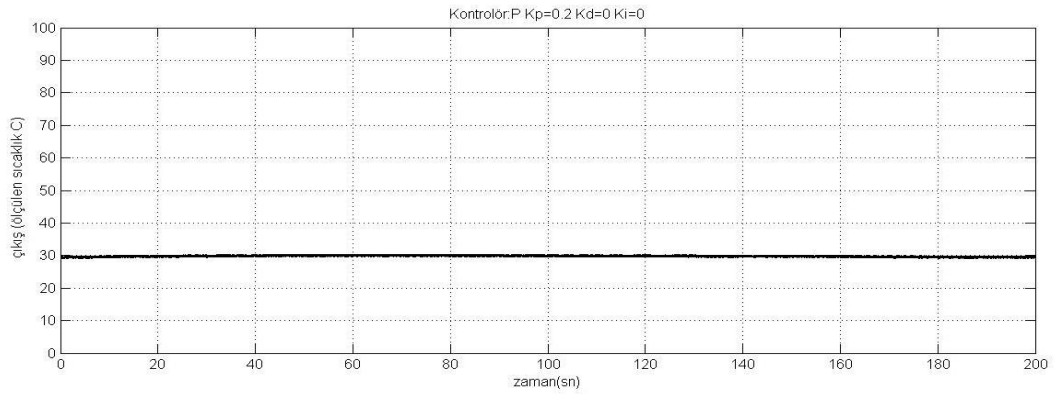


Şekil 6.15. On/off kontrolör,31-60 °C, histeresis:3

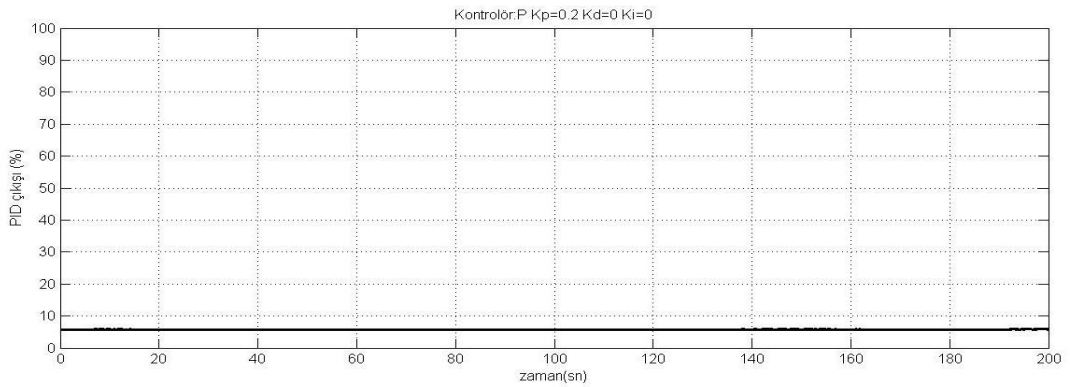
Şekillerde de görüldüğü gibi histeresis olduğunda referans değerinin altında ve üstünde bir bant aralığı oluşturulduğundan hem aşma değeri azalmıştır hem de sıcaklık değeri ortalama 60 °C olmuştur. 1. Ölçümde 60 °C'ye gelir gelmez kontrol sistemi ısıtıcıyı çalıştırdığı için 60 °C'nin üstünde salınım yapmıştır. Yükselme süresi:48.82s, Maks. Aşma: 24.12 °C, Tepe süresi:97.21 s, Yerleşme süresi:300.3 s.

6.4.3.Oransal (P) Denetleyici

1.ölçüm: Ortam sıcaklığı 30 °C, Set değeri: 60 °C $K_p:0.2$



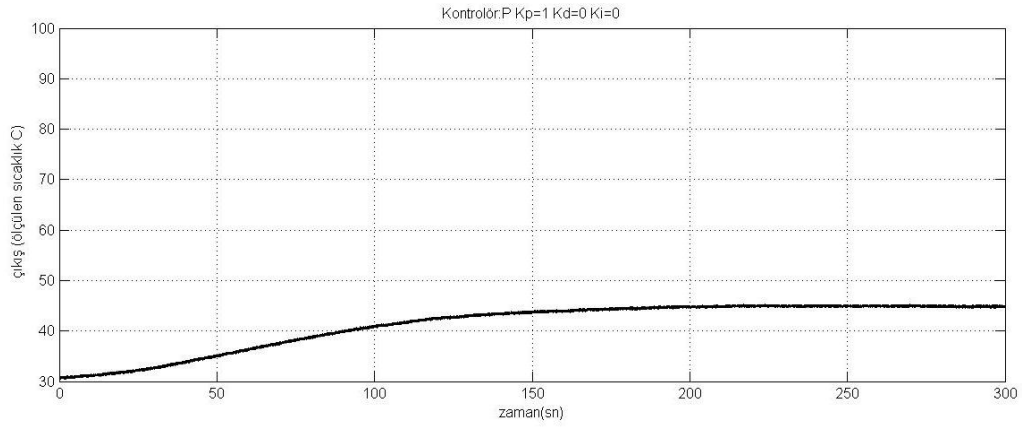
Şekil 6.16. Oransal kontrolör, 30-60 °C, $K_p:0.2$ sıcaklık-zaman grafiği



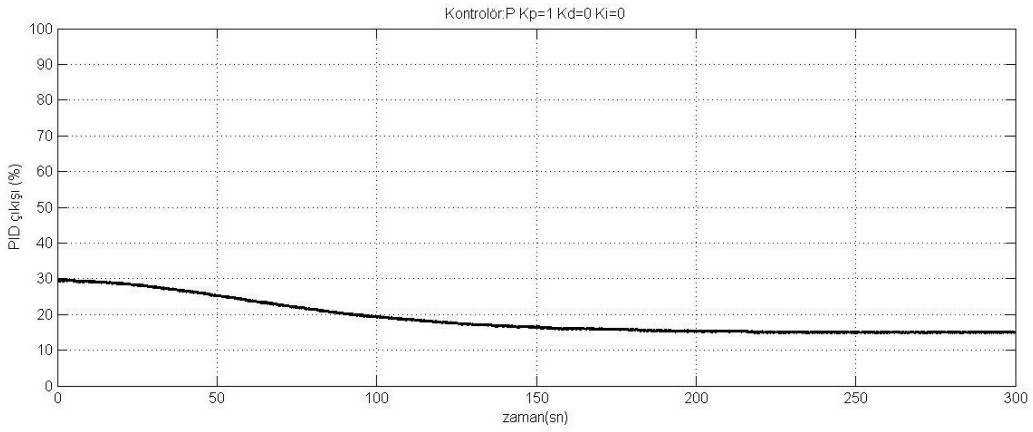
Şekil 6.17. Oransal kontrolör, 30-60 °C, $K_p:0.2$ PID çıkışı-zaman grafiği

K_p değeri 1.ölçümdeki ayarlandığından oransal sonuçta hatanın 0.2 katı alınmaktadır. Bu nedenle de ısıtıcı küçük değerlerde tetiklendiğinden, ortamı ısıtamayarak istenilen değerlere getirememiştir. Referans değeri 60 iken 30 derece de sabitlenmiştir. Yani ortam sıcaklığının üzerine çıkamamıştır.

2.ölçüm: Ortam sıcaklığı 31 °C, Set değeri: 60 °C $K_p:1$



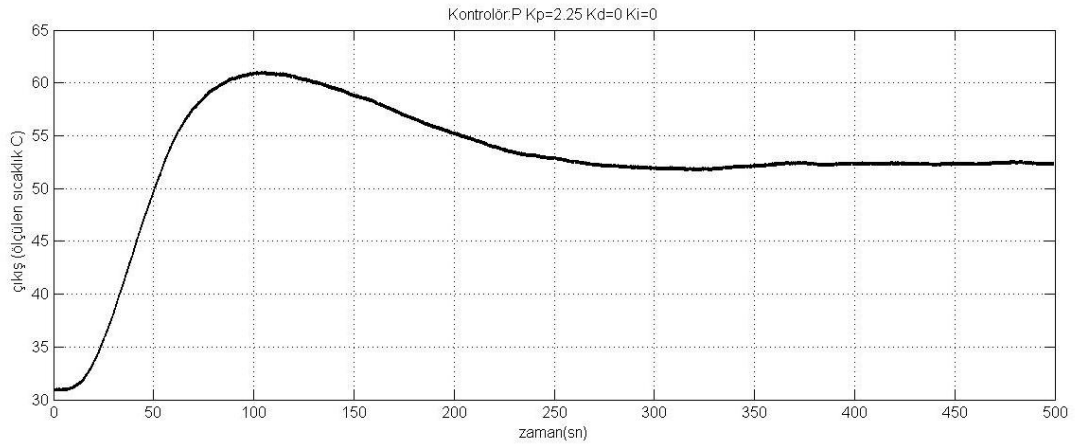
Şekil 6.18. Oransal kontrolör, 31-60 °C, Kp:1 sıcaklık-zaman grafiği



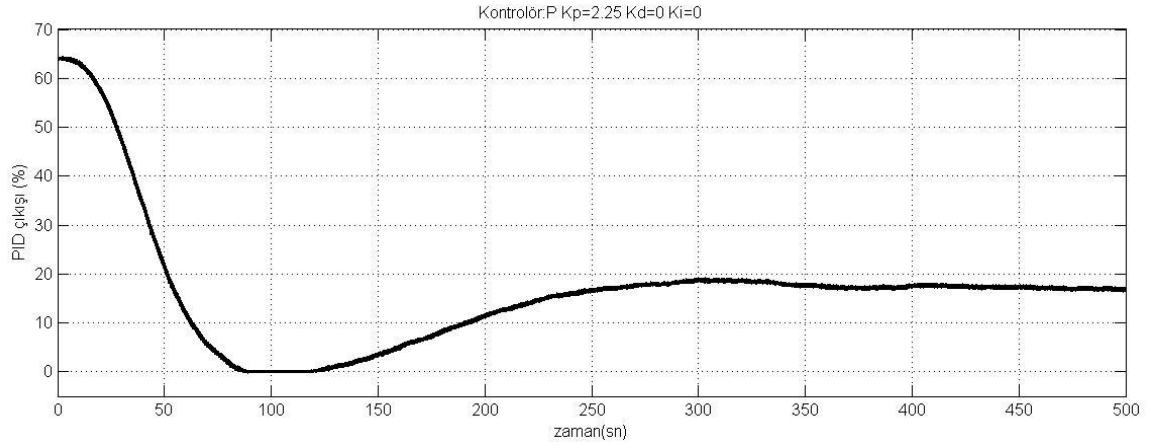
Şekil 6.19. Oransal kontrolör, 31-60 °C, Kp:1 PID çıkışı-zaman grafiği

Oransal katsayı 1 yapıldığında ortam sıcaklığı 200 sn sonra 45 °C'de sabitlenmiştir.

3.ölçüm: Ortam sıcaklığı 31 °C, Set değeri: 60 °C Kp:2.25



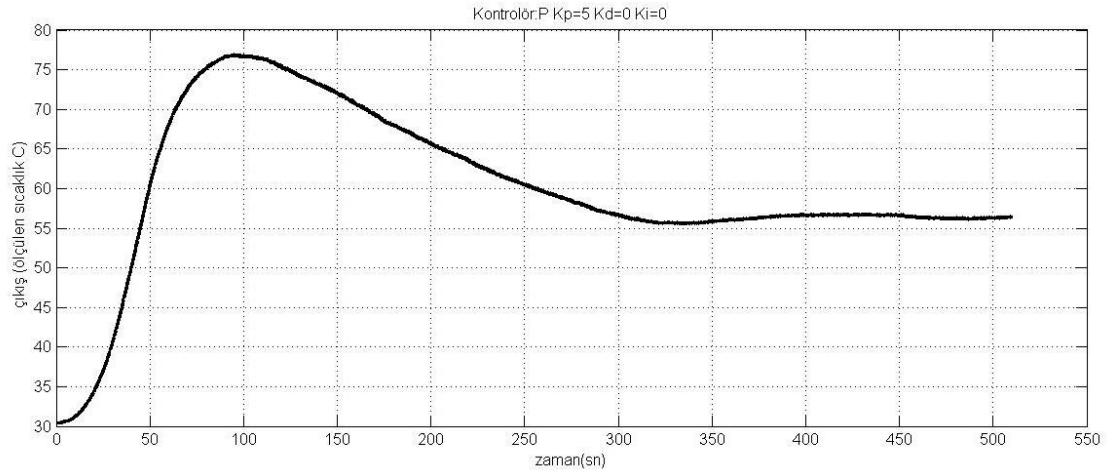
Şekil 6.20. Oransal kontrolör, 31-60 °C, Kp:2.25 sıcaklık-zaman grafiği



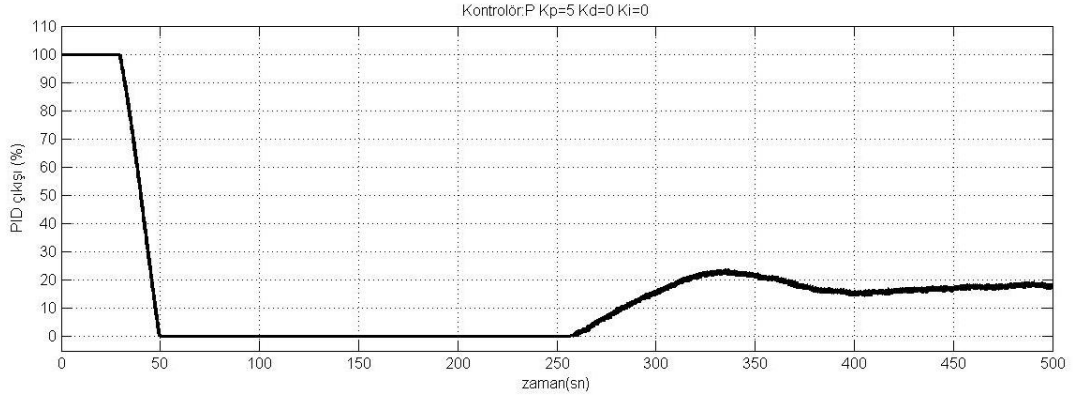
Şekil 6.21. Oransal kontrolör,31-60 °C, Kp:2.25 PID çıkışı-zaman grafiği

Hem yükselme zamanını arttırmak hem de kalıcı durum hatasını azaltmak için Kp katsayısı 2.25 yapıldığında ortam sıcaklığı 100 sn de 61 dereceye gelmiş ancak sürekli durumda 52 °C de sabitlenmiştir.

4.ölçüm: Ortam sıcaklığı 31 °C, Set değeri: 60 °C Kp:5



Şekil 6.22. Oransal kontrolör,31-60 °C, Kp:5 sıcaklık-zaman grafiği

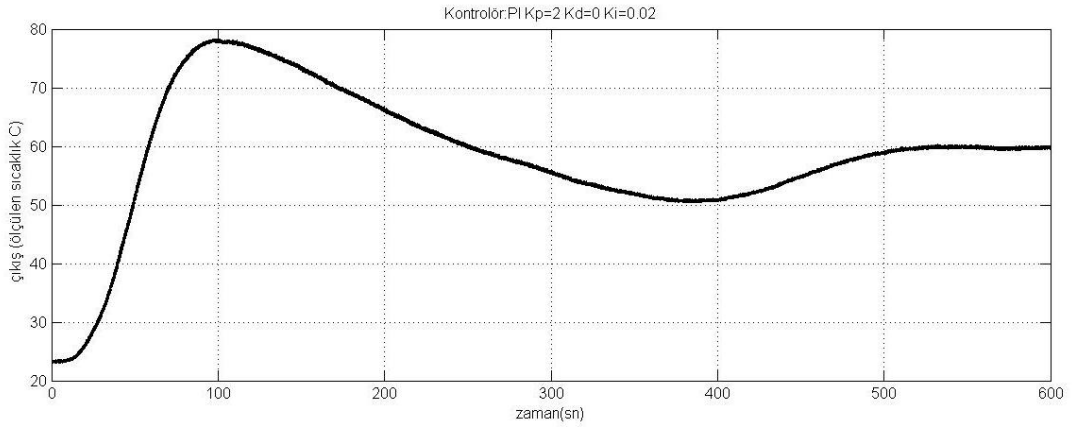


Şekil 6.23. Oransal kontrolör,31-60 °C, Kp:5 PID çıkışı-zaman grafiği

Oransal parametre 5'e ayarlandığında 50 sn de 60 °C'ye ulaştığı ancak 76 °C değerine kadar yükseldiği görülmektedir. Bunun sebebi, ortam sıcaklığı 60 °C'ye geldiğinde hata değeri 0 olduğundan P çıkışı da sıfır olmaktadır ancak rezistansın kendi ısısı hemen düşmediği için ortamın ısısını da arttırmaktadır.

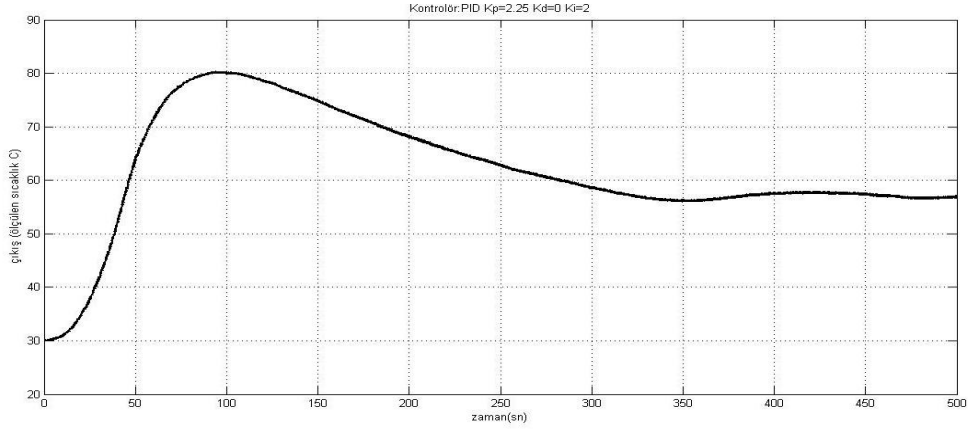
6.4.4.Oransal+Integral (PI) Denetleyici

1.ölçüm: Ortam sıcaklığı 23 °C, Set değeri: 60 °C Kp:2 Ki:0.02



Şekil 6.24. Oransal+Integral kontrolör,23-60 °C Kp:2 Ki:0.02 sıcaklık-zaman grafiği

2.ölçüm: Ortam sıcaklığı 30 °C, Set değeri: 60 °C Kp:2.25 Ki:2

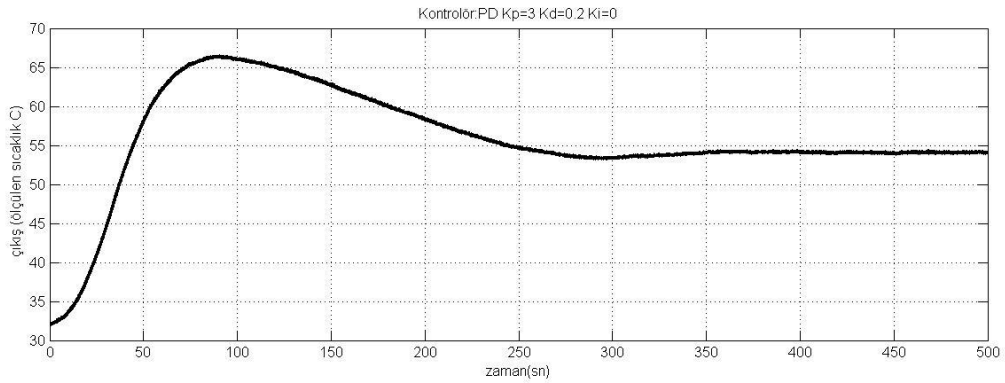


Şekil 6.25. Oransal+Integral kontrolör,30-60 °C Kp:2.25 Ki:2 sıcaklık-zaman grafiği

Sistemin oransal kontroldeki kalıcı durum hatalarını gidermek için PI kontrolör kullanılmıştır. İlk olarak Kp:2 ve Ki:0.02'ye ayarlanmıştır. Bunun sonucunda sistemde hem aşma hem de salınımların olduğu fakat referans değeri de sabitlendiği görülmektedir. Kp:2.25 Ki:2 olarak ayarlandığında ise kalıcı durum hatasının azaldığı fakat sıfırlanmadığı görülmektedir.

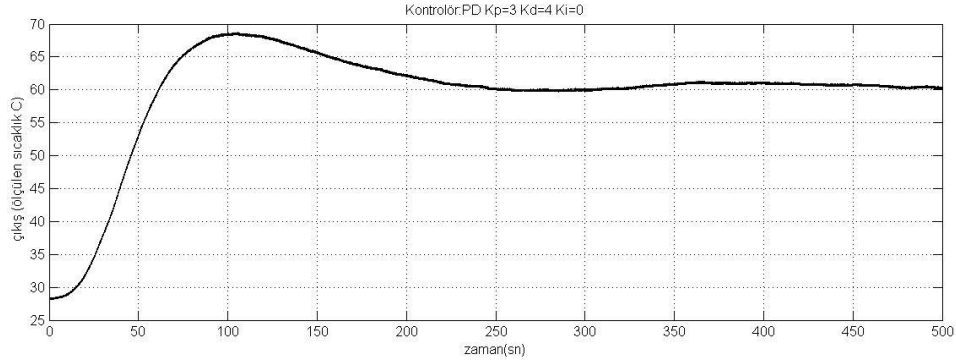
6.4.5.Oransal+Türevsel (PD) Denetleyici

1.ölçüm: Ortam sıcaklığı 32 °C, Set değeri: 60 °C Kp:3 Kd:0.2



Şekil 6.26. Oransal+Türev kontrolör,32-60 °C, Kp:3 Kd:0.2 sıcaklık-zaman grafiği

2.ölçüm: Ortam sıcaklığı 28 °C, Set değeri: 60 °C Kp:3 Kd:4

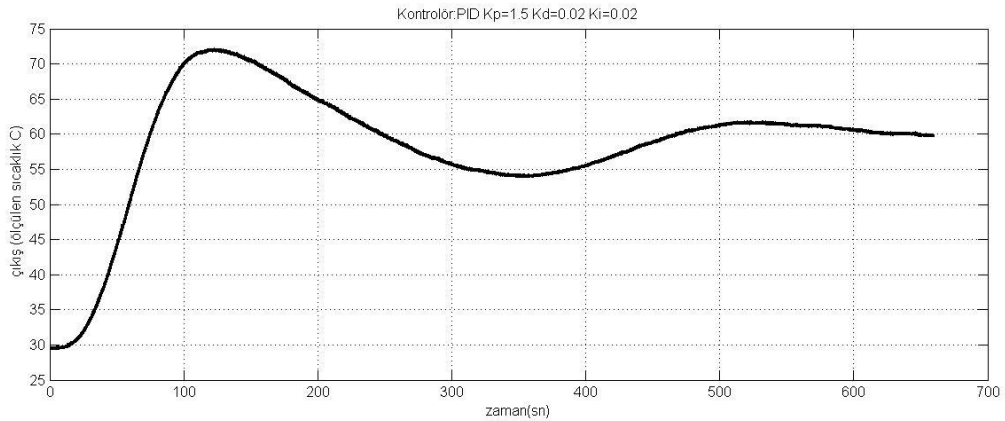


Şekil 6.27. Oransal+Türev kontrolör,32-60 °C, Kp:3 Kd:4 sıcaklık-zaman grafiği

Türevsel etki sistemin geçici rejiminde meydana gelen osilasyonları ve aşmaları düzeltmek için kullanılmıştır. Yukarıdaki şekillerden de görüldüğü gibi kalıcı durum sıfırlanmamıştır ancak oransal ve oransal+integral kontrolde 75-80 °C'leri ulan aşma değeri türevsel etki sayesinde 70 °C'yi aşmamıştır.

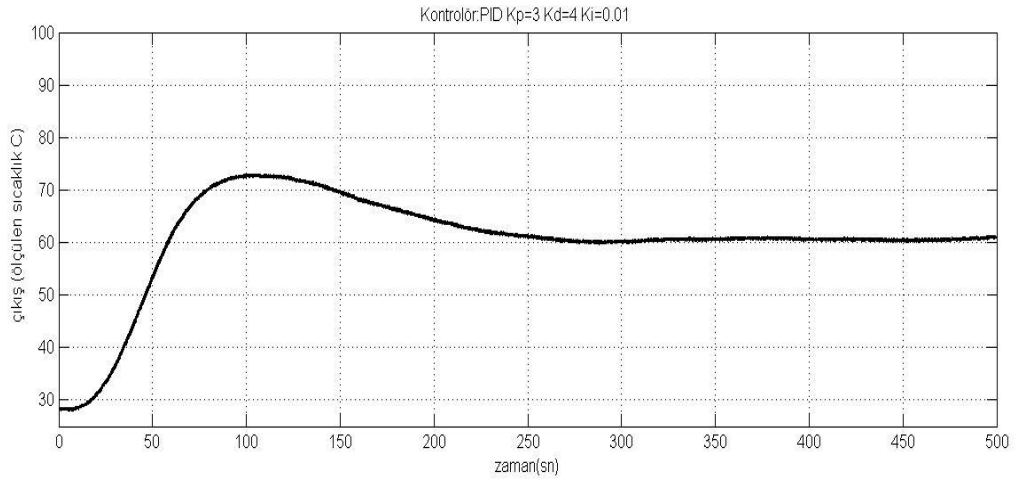
6.4.6.Oransal+Integral+Türevsel (PID) Denetleyici

1.ölçüm: Ortam sıcaklığı 31 °C, Set değeri: 60 °C Kp:1.5 Kd:0.02 Ki:0.02



Şekil 6.28. Oransal+Integral+Türevsel kontrolör,30-60 °C, Kp:1.5 Kd:0.02 Ki:0.02 sıcaklık-zaman grafiği

2.ölçüm: Ortam sıcaklığı 29 °C, Set değeri: 60 °C Kp:3 Kd:4 Ki:0.01



Şekil 6.29. Oransal+Integral+Türevsel kontrolör,29-60 °C, Kp:3 Kd:4 Ki:0.01 sıcaklık-zaman grafiği

Daha önceki ölçümlerde görüldüğü üzere P kullanıldığında fazlaca aşma meydana gelmiş ve kalıcı durum hatası oluşmuştur. PI kontrolde kalıcı durum hatası düzeltilmiş ancak geçici rejimde aşma daha da artmıştır. PD kontrolde geçici zamandaki aşma azaltılmış fakat referans değeri de sabitlenememişti. Hem kalıcı hem de geçici rejimi düzeltmek için PID kontrolör kullanıldığında ve parametreler uygun seçildiğinde sistemin geçici durumda fazla aşma yapmadan kısa sürede sürekli zamanda referans değere oturduğu görülmektedir.

7.BULANIK MANTIK VE PID İLE SERVO SİSTEM KONTROLÜ

Servo, herhangi bir mekanizmanın işleyişini hatayı algılayarak yan bir geri besleme düzeneğinin yardımıyla denetleyen ve hatayı gideren otomatik aygıttır. Servo sistem ise kontrol edilen hassas mekanizma ve ondan geri bildirim alan elemanın tümüne verilen isimdir. Bu çalışmada servo sistem olarak bir DC motor sistemi kullanılmıştır ve motorun hızı PID ve bulanık mantık ile kontrol edilmeye çalışılmıştır. DC motorlar direkt olarak DC güç kaynağından beslenebilen motorlardır ve elektrik enerjisini direkt olarak mekanik enerjiye dönüştürebilmektedir. DC motorlar üretim araçları, endüstriyel robot vb. birçok uygulama ile uyumludurlar. AC motorlar ile kıyaslandığında maliyetleri düşüktür. Ayrıca DC motorlar, çok basit ve kararlı kontrol yöntemleri ile uyum sağlamaktadırlar. Diğer bir avantajı ise yüksek verimlik ve ani oluşan yük artışlarına yüksek başlama torkudur [57]. Fakat periyodik bakım ihtiyaçları, çıkışların mekanik olarak kısa zamanda aşınması, akustik gürültü, parlama, fırçanın verime etkisi gibi bazı alanlarda yetersiz kalan DC motorlara alternatif olarak fırçasız DC motorlar ortaya çıkmıştır [58]. Gürültülü çalışmamaları, bakım gerektirmemeleri, dinamik cevaplarının hızlı olması, tork karakteristiğinin iyi olması verimli çalışmaları gibi avantajları sayesinde fırçasız DC motorlar tercih edilebilmektedir [59] Bu çalışmada kontrol sistemlerine ağırlık verildiğinden motorların verim, maliyet vb. özellikleri göz önünde bulundurulmamıştır.

Yüksüz durumdaki bir motorun sürülmesi oldukça zordur. Değişken yüklerde ise motorun hızını sabitlemek için motorun giriş gerilimini yükte doğru orantılı olarak değiştirmek gerekir. Fakat sisteminin transfer fonksiyonu doğrusal olmayan bir denklemdir. Bozucu etkiler sistemi kararsız kılar. Bu bozucu etkileri gidermek için sistemi dengelemek gerekir[60].

Kocaoğlu S. ve Kuşçu H. Yaptıkları çalışmada PIC kullanarak bir DC motorun hız ve konum kontrolünü yapmak için gerekli PID parametrelerinin belirlenmesini amaçlamışlardır. Öncelikle motor parametrelerini çıkarıp inceleme yapmış ve bir uygulama yaparak sonuçları gözlemlemişlerdir. Sonuç olarak girilen PID parametrelerine karşılık istenilen referans değerlerde motorun kontrolü sağlanmış ve sistem deney seti olarak kullanılabilir hale getirilmiştir[60].

Bulut M., Kurt B. ve Demirtaş M. bir doğru akım motorunun hız kontrolünü gerçekleştirmek için oransal-integral (PI) tipi bulanık kontrolör tasarlamışlardır. Alınan sonuçlara göre genetik algoritmalar vasıtası ile otomatik olarak tasarlanan kontrolörün performansının insan bilgi temelli kontrolöründeki ile karşılaştırıldığında daha iyi olduğu anlaşılmıştır[61].

Akyazı, Ö. , Zenk H. ve Akpınar A.S. ise bulanık mantık kontrolde kullanılan farklı üyelik fonksiyonlarının (üçgen, yamuk, gaussian, cauchy, çan) etkilerini sürekli mıknatıslı bir doğru akım motoru üzerinde yaptıkları çalışmada incelemiş ve alınan sonuçlara göre en fazla hata çan üyelik fonksiyonunda çıktığı en iyi sonuçların üçgen ve yamuk üyelik fonksiyonlarında çıktığı ve diğerlerinin ortalama sonuç verdiği anlaşılmıştır [62].

Köse F., Kaplan K. ve Ertunç M. ise STM32F407 Discovery geliştirme kitini kullanarak sabit mıknatıslı bir DC motorun hız kontrolünü PID ve bulanık mantığı kullanarak gerçekleştirmiş ve farkı incelemişlerdir. Alınan sonuçlara göre iki sistemde de kalıcı durum hatalarının benzer çıktığı fakat bulanık mantık kontrolörde aşırı aşımın fazla olduğu ve yükselme zamanının daha kısa olduğu ve oturma zamanının daha geç olduğu gözlemlenmiştir [63].

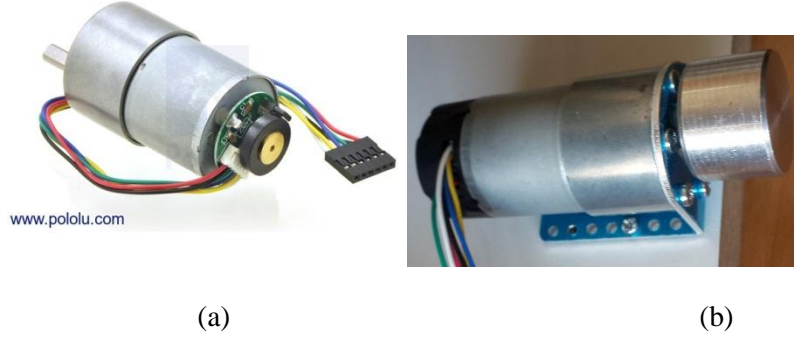
Bu kısımda iki sistem kullanılmıştır. Bunlardan ilki enkoderli DC motor kullanılarak kontrol edilen sistem diğeri ise takogeneratörlü DC motor kullanılarak kontrol edilen sistemdir.

7.1. Enkoderli DC Motor Kontrolü

7.1.1. Sistemin Mimari Bileşenleri

7.1.1.1.Motor ve Geri besleme elemanı

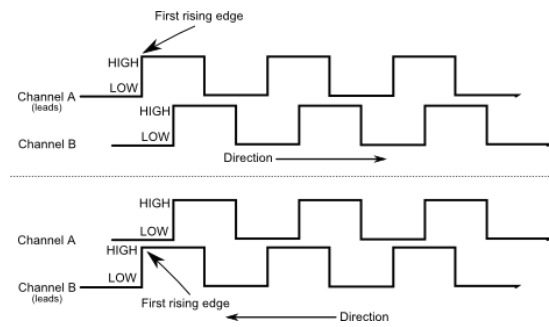
Sistemde kullanılan motor, Pololu marka 12 V ile çalışan, maksimum yükte 5.6 A çekebilen, 10000 rpm dönüş hızına sahip miline bağlı redüktörü ve arka kısmında yine miline bağlı artımsal enkoderi bulunan fırçalı bir DC motordur [64].



Şekil 7.1. Enkoderli redüktörlü DC motor

Redüktöründe bulunan dişliler ile 50:1 oranında bir dönüştürme oranına sahiptir ve kontrol bu redüktör çıkışına göre ayarlanmıştır. Motor 10000 rpm olduğundan redüktör çıkışı maksimum 200 rpm olmaktadır. Şekil 7.1. b’de görüldüğü gibi motor, metal gearmotor bracket ile sabitlenerek çalışılmıştır.

Geri besleme olarak motor miline bağlı quadrature enkoder bulunmaktadır. Motor miline bağlı bir manyetik disk ve onun önünde bulunan bir hall effect sensörden oluşmaktadır. Enkoder 3.5-20 V besleme gerilimi ile çalışmakta ve motor hızına göre değişken frekanslarda kare dalga üretmektedir. Enkoder, A ve B olmak üzere iki kanala sahiptir. İki kanaldan da birbirleri arasında 90 derece faz farkı bulunan kare dalga sinyali alınmaktadır[64].



Şekil 7.2. Enkoder çıkışı

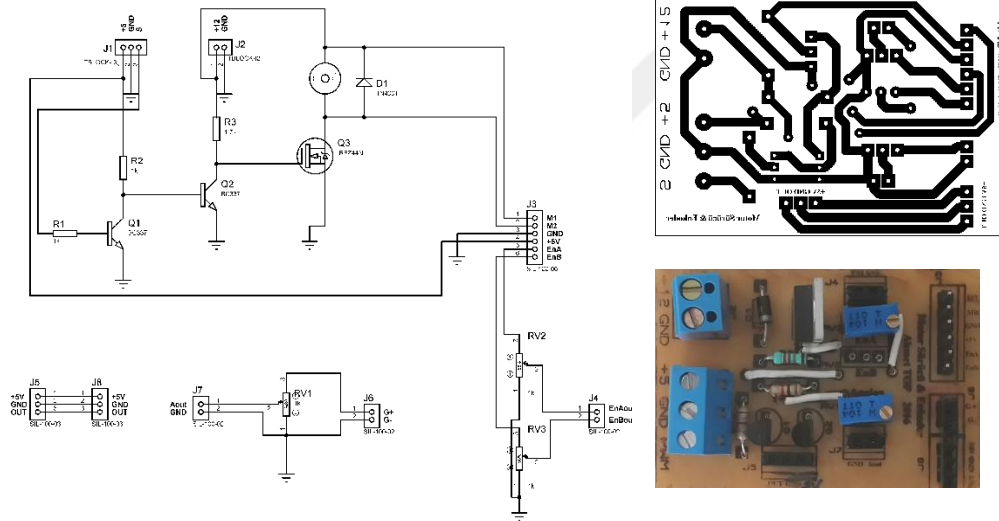
Her bir kanaldan bir turda 16 pals alınmaktadır. Motor dişli oranı 50:1 olduğundan redüktör çıkışı bir tur döndüğünde 800 pals alınmaktadır.



Şekil 7.3. Motor miline bağlı olan enkoder

7.1.1.2 Motor Sürücü ve Mikrokontrolör

Motorun sadece tek yönde hız kontrolü yapıldığı için sürme işlemi tek bir mosfet ile yapılmıştır. Kullanılan mikrodenetleyicinin çıkış voltajı düşük olduğundan mosfetin kontrolü de iki adet transistor ile yapılmıştır.

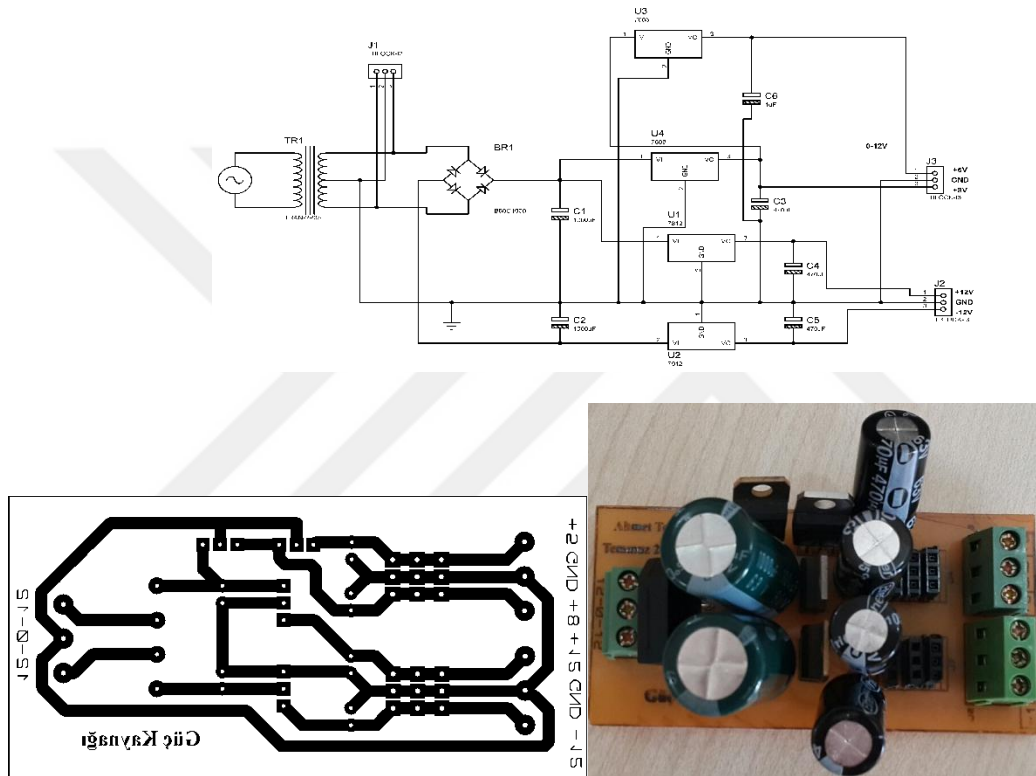


Şekil 7.4. DC motor sürücü

Mikrokontrolör olarak sıcaklık kontrol düzeneğinde kullanılan Arduino Due kullanılmıştır. Enkoderden alınan geri besleme değeri referans değer ile karşılaştırılıp işleminden geçirildikten sonra Arduino'dan alınan PWM sinyalleri ile motor sürücü ve dolayısıyla da motor kontrol edilmiştir. Referans giriş ve değer okuma 3x4 keypad ve lcd ile yapılmaktadır.

7.1.1.3 Güç Kaynağı

Devrede kullanılan mikrodenetleyici için 9V DC, kullanılan DC motorları beslemek için 12 V DC, Enkoderi beslemek için 5 V DC ve sıcaklık kontroldeki lm35 sıcaklık sensörü ve kuvvetlendirme devresinde kullanılan op-amp'lerin beslemelerinde kullanılmak üzere ise -12 V DC ihtiyaçlarını karşılamak üzere bir güç kaynağı tasarlanmıştır.

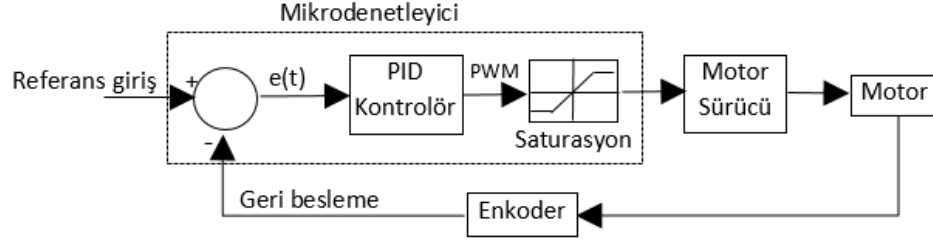


Şekil 7.5. Güç Kaynağı

7.1.2.PID Denetleyicinin Enkoderli Motor Kontrolündeki Etkisi

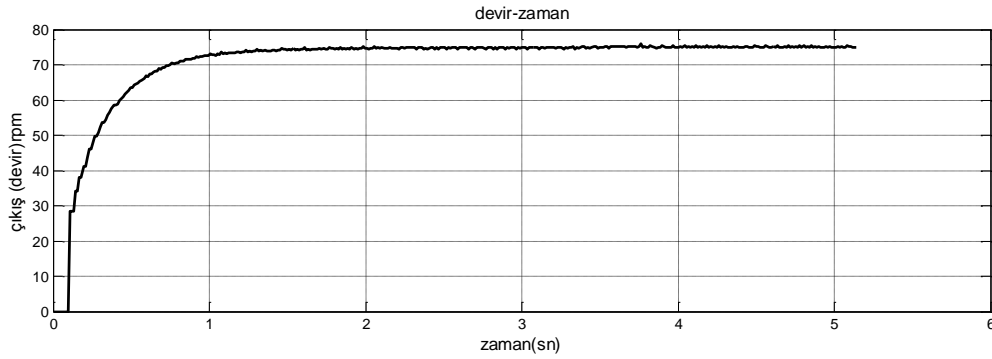
Motor kontrol edilmeden önce güç kaynağından 1 V değerinden başlanarak lineer olarak 12 V 'a kadar arttırılmış ve her bir değer için enkoder çıkışının frekansı osiloskopta ölçülmüştür. Sonuçlar lineer olmadığı için gerilim-frekans grafiği matlab'da çizdirilmiş ve eğri uydurma yöntemi ile 4. Dereceden bir denklem oluşturulmuştur. Mikrodenetleyicide enkoderden harici kesme olarak alınan sinyalin frekansı ölçülmüş ve bu denklemden geçirilerek devir bulunmuştur. Ölçülen devir referans değerden çıkartılarak hata değeri bulunmuştur. Bu işlem periyodu yaklaşık 356 mikro saniyedir. Motor

redüktör çıkışı 0-200 rpm arasında olduğundan kontrol bu aralıkta gerçekleşmektedir. Sistem yazılımı Arduino IDE arayüz programında C++ tabanlı yazılmış olup, yine aynı program üzerinde bulunan seri port'dan alınan veriler Matlab programında çizdirilmiş ve sonuçlar incelenmiştir.



Şekil 7.6. PID kontrol sisteminin blok şeması

Şekil 7.6'de görüldüğü gibi keypad yardımı ile alınan referans bilgisi ile enkoderden alınan geri beslemenin farkı mikrodenetleyici içerisinde alınarak PID algoritmasından ve saturasyondan geçirilerek motor sürücüyeye PWM olarak verilmekte ve motor sürücü de motoru sürmektedir.



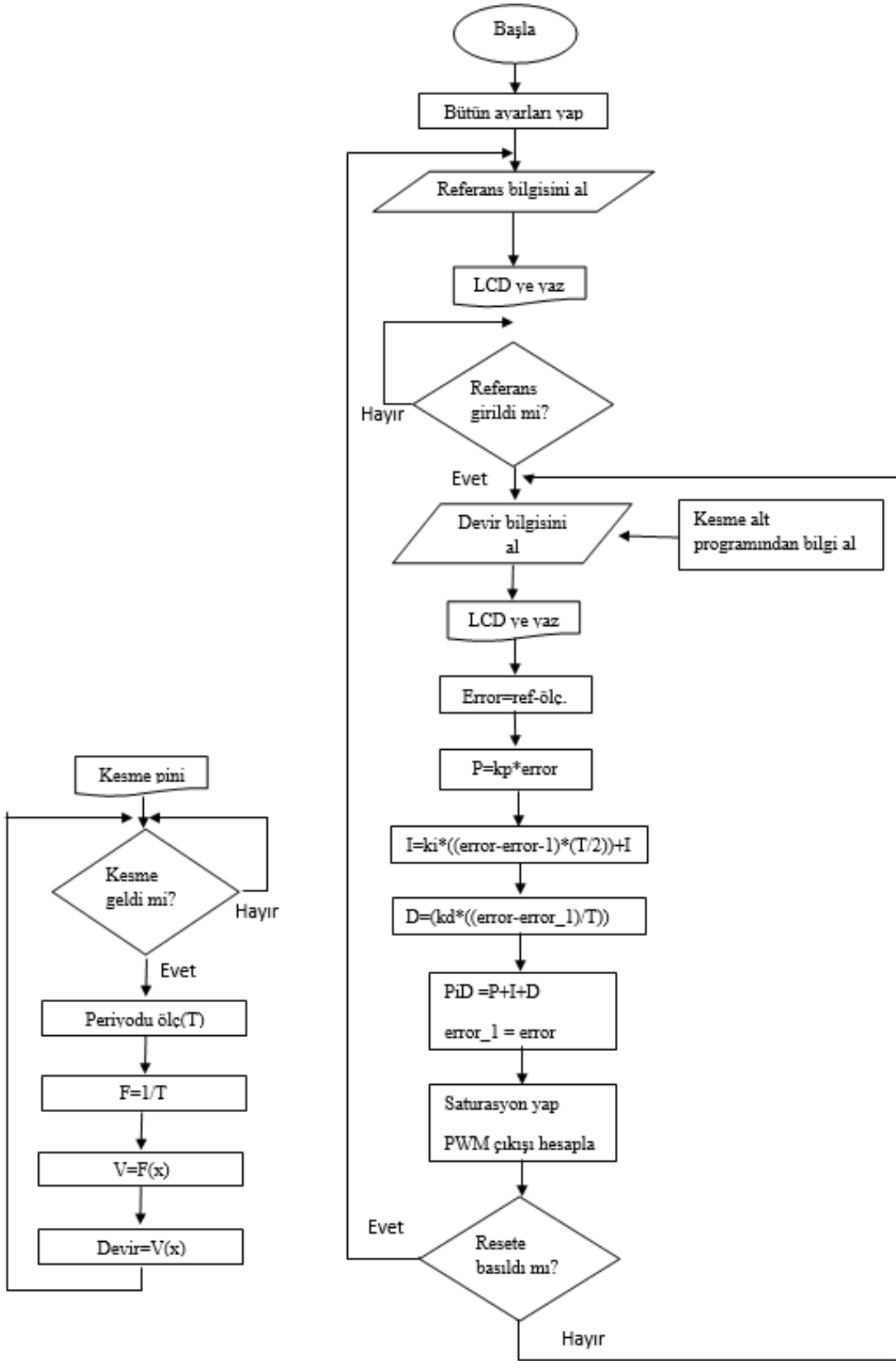
Şekil 7.7. Motorun kontrolsüz devir-zaman çıkış sinyali

Motor, kontrolsüz bir şekilde çalıştırılıp şekil 7.7'de görüldüğü gibi devir-zaman grafiği çıkartıldığında sürekli duruma 2 sn de ulaştığı görülmektedir. Kontrol sistemleri ile sürekli duruma erişme süresi düşürülerek istenilen referans değerlerde motoru sabit tutmak amaçlanmıştır.

Oluşturulan yazılım ile oransal, oransal+integral, oransal+türev ve oransal+integral+türevsel denetleyici tipleri gerçekleştirilmiştir. Bu denetleyiciler ile çeşitli

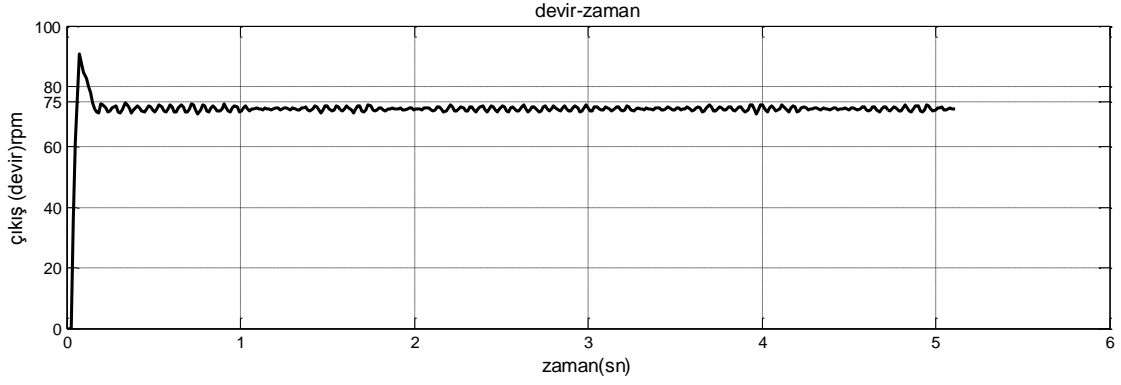
parametre deęerlerinin sistem tepkisi üzerindeki etkileri incelenmiřtir. Deęiřtirilen K_p , K_d ve K_i deęerlerine karřın sistemin verdięi tepkiler ařaęıdaki řekillerde grlmektedir. Btn parametrelerde referans devir deęeri 75 rpm olarak alınmıřtır. PID iin yazılan kodun akıř řeması řekil 7.8.'de grldę gibidir.



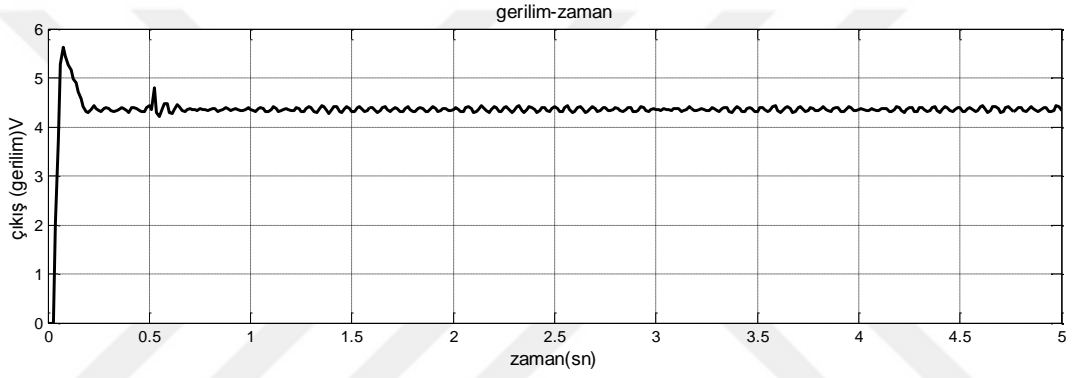


Şekil 7.8. Motor kontrol PID akış diyagramı

7.1.2.1. Oransal Kontrol (P)



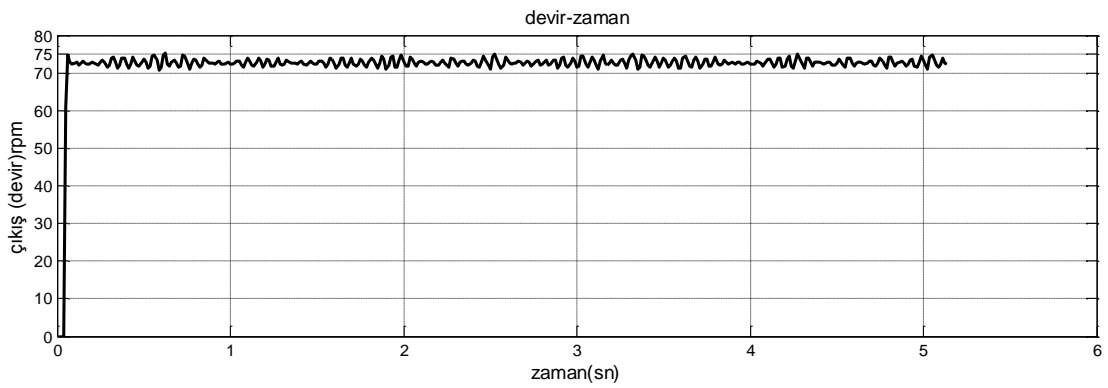
Şekil 7.9. $K_p=20, K_d=0, K_i=0$ durumunda devir çıkış sinyali



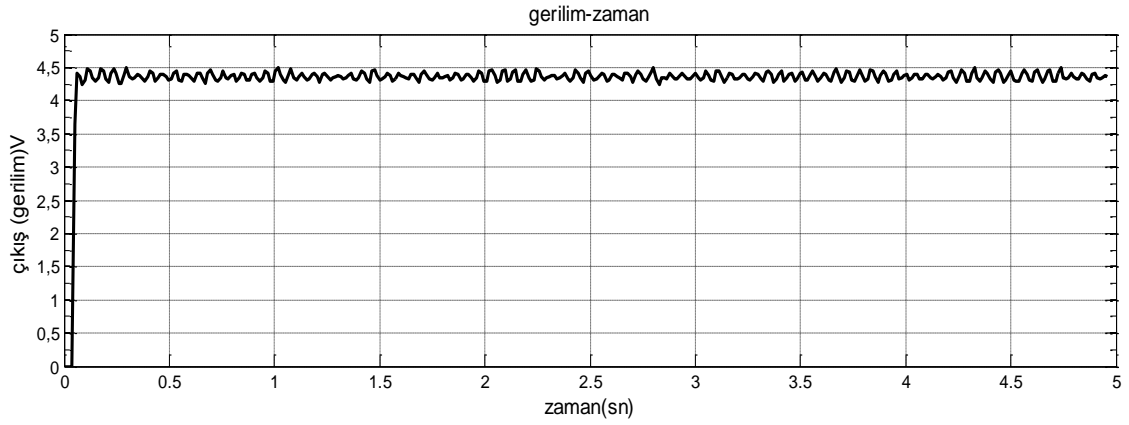
Şekil 7.10. $K_p=20, K_d=0, K_i=0$ durumunda gerilim çıkış sinyali

Motora sadece oransal kontrol uygulandığında 62 ms de referans değerine ulaştığı ve 161 ms de sürekli duruma geçtiği ancak 2.29 rpm lik bir kalıcı durum hatası olduğu görülmektedir. Ayrıca 90 rpm e kadar bir aşma gerçekleşmiştir.

7.1.2.2. Oransal+Türevsel Kontrol (PD)



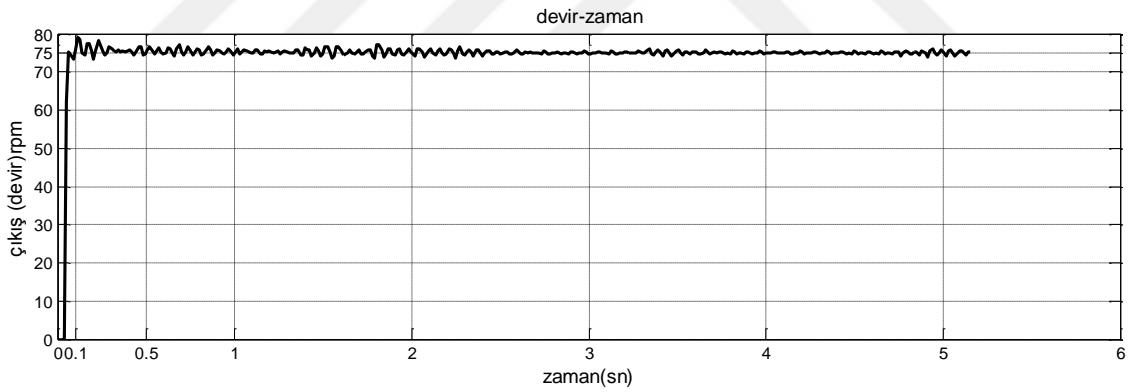
Şekil 7.11. $K_p=20, K_d=5, K_i=0$ durumunda devir çıkış sinyali



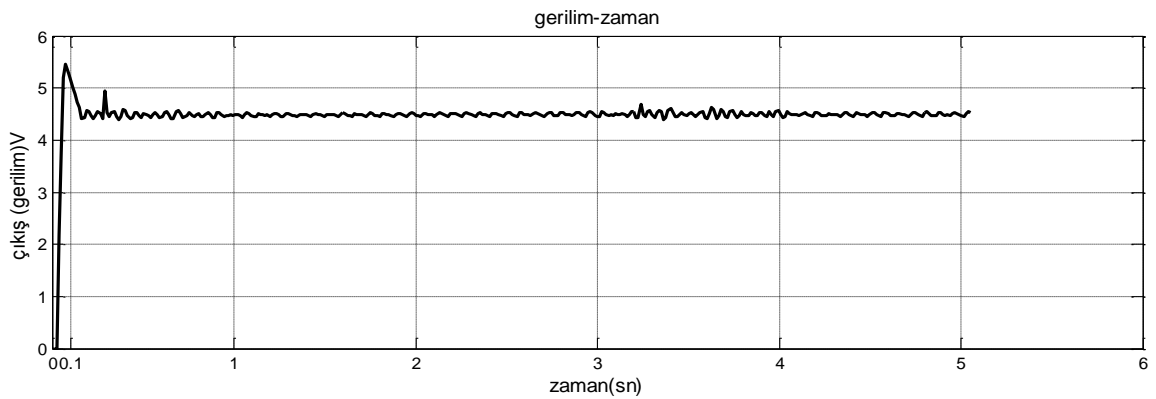
Şekil 7.12. $K_p=20, K_d=5, K_i=0$ durumunda gerilim çıkış sinyali

PD kontrol kullanıldığında şekilde de görüldüğü gibi oransal kontrolde oluşan aşmanın ortadan kalktığı ve sistemin 62.5 ms de kalıcı duruma geçtiği fakat yaklaşık 3 rpm lik kalıcı durum hatası olduğu şekil7.12’de görülmektedir.

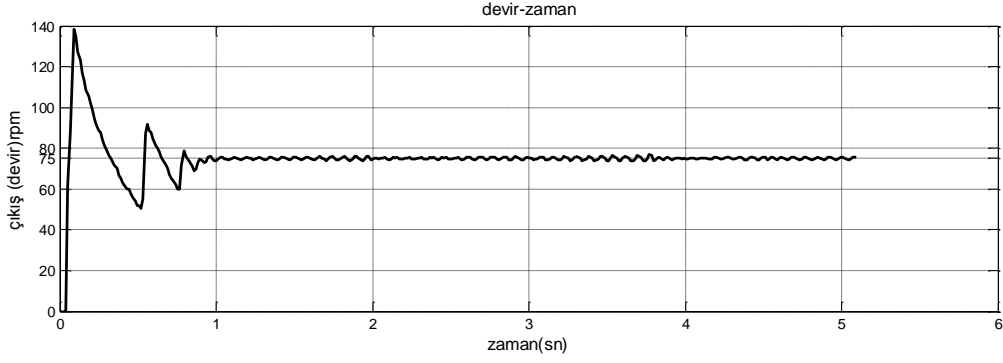
7.1.2.3. Oransal+İntegral +Türevsel Kontrol (PID)



Şekil 7.13. $K_p=20, K_d=5, K_i=0.1$ durumunda devir çıkış sinyali



Şekil 7.14. $K_p=20, K_d=5, K_i=0$ durumunda gerilim çıkış sinyali

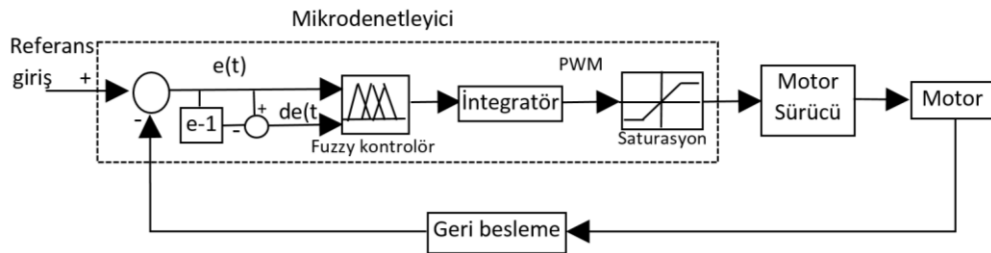


Şekil 7.15. $K_p=15, K_d=5, K_i=2$ durumunda devir çıkış sinyali

Sistemin hem geçici hem de kalıcı durum hatalarını ortadan kaldırmak için PID kontrol uygulandığında, şekil 7.13 de görüldüğü gibi geçici rejimde aşma olmadan ve kalıcı durum hatası meydana gelmeden 101 ms de sürekli duruma geçmiştir. PID parametreleri uygun seçilmediği takdirde şekil 7.15 de olduğu gibi hem gecikmeli hem de osilasyonlu bir sonuç ortaya çıkabilir.

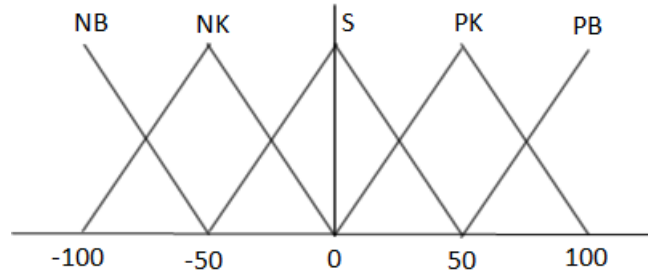
7.1.3. Bulanık Mantık Denetleyicinin Enkoderli Motor Kontrolündeki Etkisi

Bulanık mantık denetleyici ile DC motor kontrolü için iki giriş ve bir çıkışlı üyelik fonksiyonları oluşturulmuştur. Sistemin iki adet girişi vardır, bunlar hata ve hatadaki değişim değeridir. Çıkış olarak ise devir değerini değiştirecek olan PWM değeri alınmıştır.



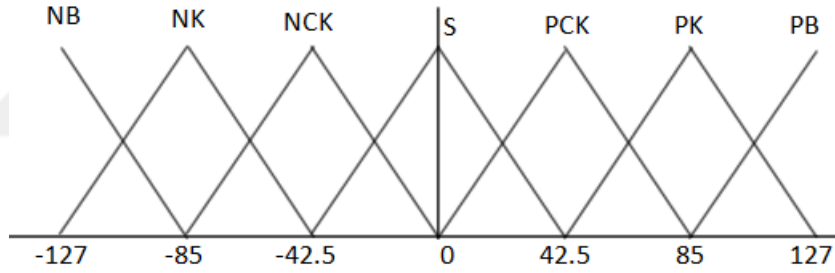
Şekil 7.16. Bulanık mantık blok diyagramı

Şekil 7.16'de görüldüğü gibi hata ve hatanın değişimi mikrodenetleyici tarafından hesaplanarak bulanık mantık hesaplamaları yapılmıştır. Sonuç integratör ve saturasyondan geçirilerek sürücüye PWM sinyali gönderilmiştir.



Şekil 7.17. giriş üyelik fonksiyonları

Şekilde de görüldüğü gibi 5 adet üçgen üyelik fonksiyonu kullanılmıştır. Bunlar NB: negatif büyük NK: negatif küçük S:sıfır PK: pozitif küçük ve PB: pozitif büyük'tür. Burada motor maksimum 100 devir olacak şekilde ayarlanmıştır. Bu kullanıcının isteğine göre 200 devire ya da daha az bir devire ayarlanabilir. Bu hem hata, hem de hatanın değişimi için geçerlidir.



Şekil 7.18. çıkış üyelik fonksiyonları

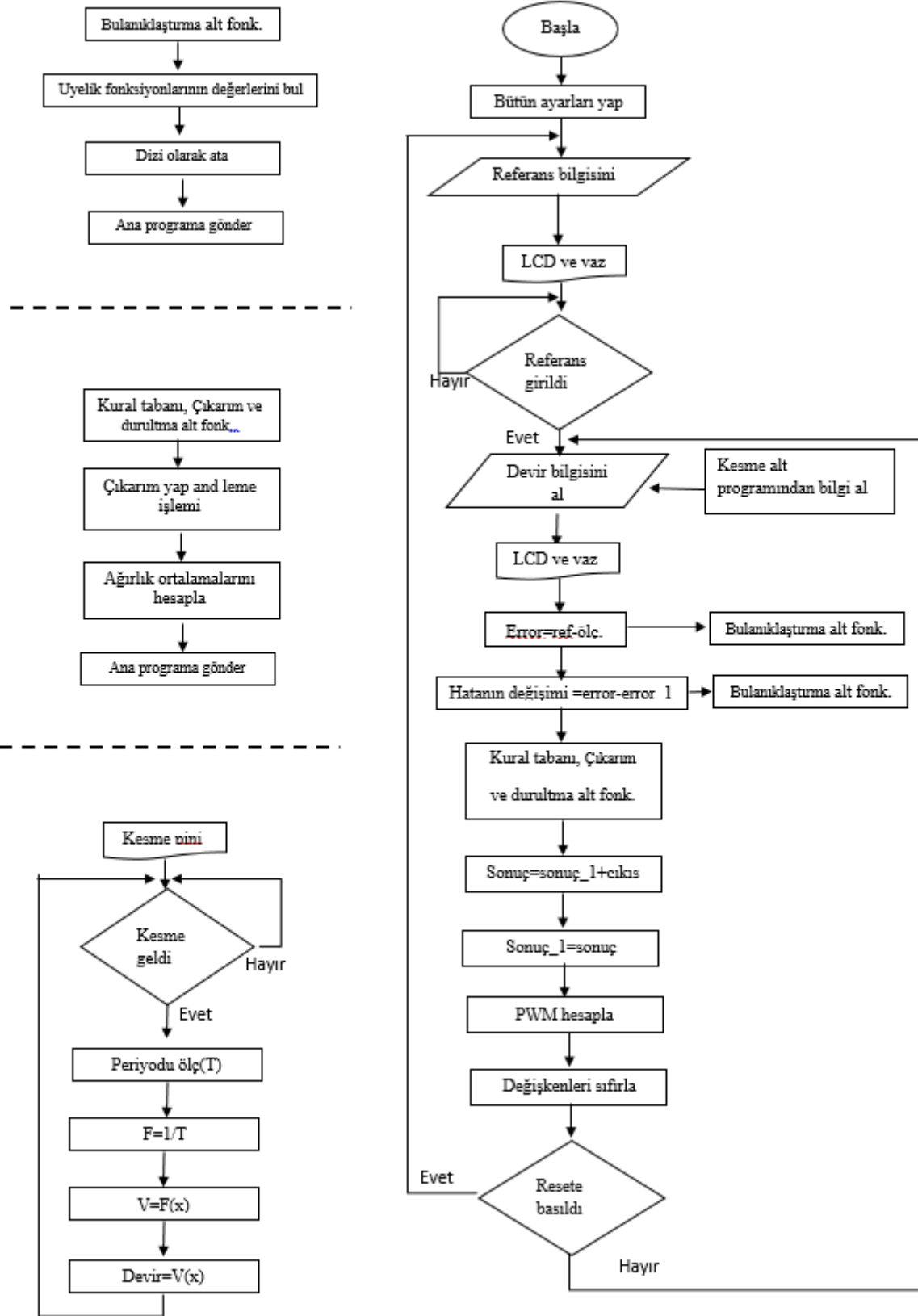
Çıkış üyelik fonksiyonları 7 adettir. Devir sayısı maksimum 100 devire ayarlandığı için PWM değeri de maksimum 127 ye ayarlanmıştır. Çünkü 8 bitlik PWM'in maksimum değeri 255'dir. Buradaki üyelik fonksiyonları NB: negatif büyük NK: negatif küçük, NCK: negatif çok küçük, S:sıfır, PCK: pozitif çok küçük, PK: pozitif küçük ve PB: pozitif büyük' tür. Bulanıklaştırma biriminde hata ve hatadaki değişim, şekil 7.17 'deki üyelik fonksiyonları ile bulandırıldıktan sonra bulunan üyelik dereceleri çıkarım birimine gider. Burada cebirsel çarpım yöntemi kullanılarak bulunan kesinlik dereceleri toplama işlemine girmeden tablo 7.1 'deki kural tabanının dan da geçirilerek durulaştırma birimine gider.

Tablo 7.1 Kural Tabanı

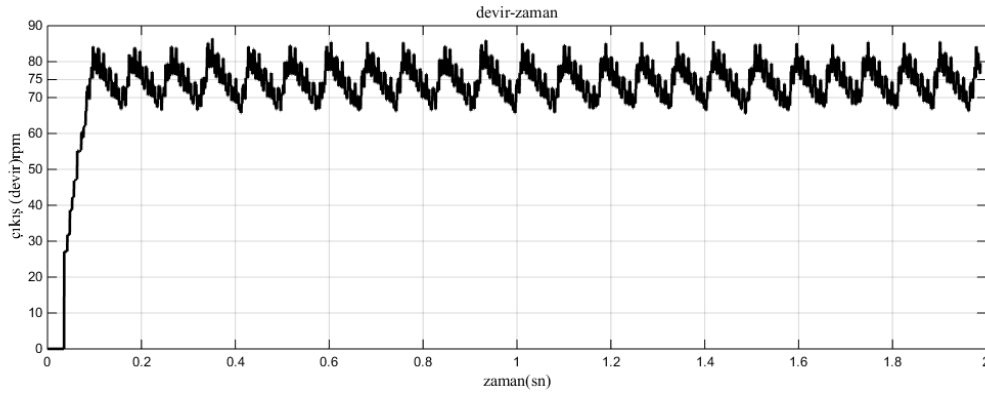
de e	NB	NK	S	PK	PB
NB	NB	NB	NK	NCK	S
NK	NB	NK	NCK	S	PCK
S	NK	NCK	S	PCK	PK
PK	NCK	S	PCK	PK	PB
PB	S	PCK	PK	PB	PB

Durulaştırma biriminde ağırlık ortalaması yöntemi ile bulunan bulanık mantık sonucu PWM değeri olarak motor sürücüye gönderilmiştir.



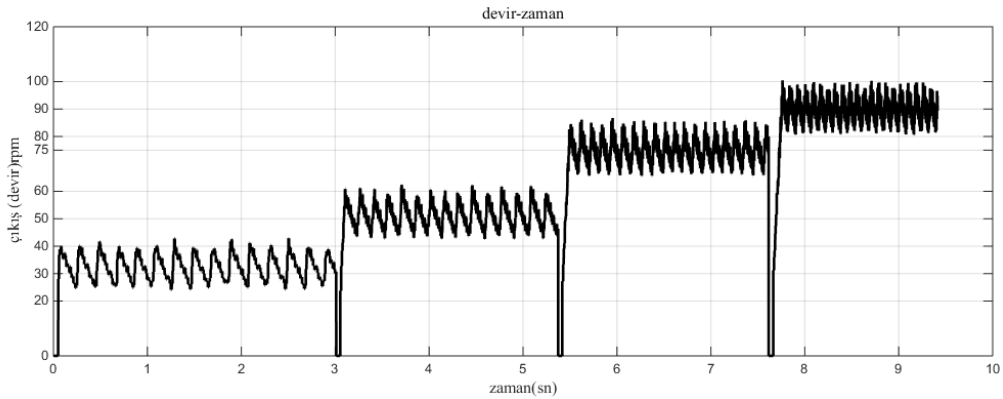


Şekil 7.19. Motor kontrol bulanık mantık akış diyagramı



Şekil 7.20. DC motor Bulanık Mantık kontrollü devir-zaman grafiği

DC motora bulanık mantık kontrol sistemi uygulandığında motor keypad yardımı ile girilen referans değer bilgisine 90.28 ms gibi kısa bir sürede ulaşmış ancak referans değer etrafında salınım yapmıştır. Bu salınımlara neden olan en büyük faktörlerden bir tanesi DC motorun enkoderinin hassas olmaması ve bu nedenle de hassas bir geri besleme bilgisi alınamamasıdır. Salınımlara neden olan ikinci önemli neden ise enkoderden alınan kare dalga sinyalinin salınımlı frekansa sahip olmasıdır. Çünkü programda geri besleme bilgisi kare dalganın periyodu ölçülerek oluşturulmaktadır.



Şekil 7.21. DC motor Bulanık Mantık kontrollü değişik devirler için devir-zaman grafiği

Şekil 7.21 'de aynı grafik üzerinde değişken devirlerde motorun performansı görülmektedir. Girilen devirler sırası ile 30,50, 75 ve 90 rpm'dir. Her referans devirde devirin sıfıra inip tekrar yükselmesinin sebebi yazılan programdan dolayıdır. Keypad de reset tuşuna basılmadan yeni bir referans değer girilememektedir. Bunun yapılmasındaki amaç devir değerini kontrollü bir şekilde değiştirmektir.

7.2. Takogeneratörlü DC Motor Kontrolü

7.2.1 Sistemin Mimari Bileşenleri

7.2.1.1 Motor ve Geri besleme elemanı

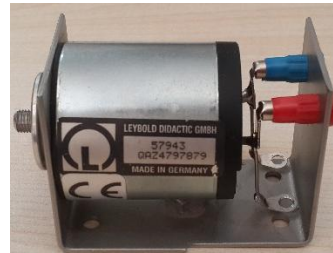
Takogeneratörlü sistemde kullanılan motor, Maxon firması tarafından üretilen 4W gücünde redüktörsüz ve enkodersiz fırçalı DC motordur. Motor ile ilgili bazı veriler aşağıdaki tabloda mevcuttur [67].

Tablo7.2. Maxon DC motor verileri

Parametre	Değeri	Birimi
Nominal voltage	12	V
No load speed	4090	Rpm
No load current	12,3	mA
Nominal speed	2430	Rpm
Nominal torque	13,9	mNm
Starting current	1,23	A
Nominal current	0,508	A
Terminal resistance	9,73	Ohm
Terminal inductance	1,27	mH
Torque constant	27,8	mNm / A
Speed constant	344	rpm / V
Mechanical time constant	31,9	ms
Thermal time constant winding	45,5	S
Thermal time constant motor	988	S
Max. permissible speed	6400	Rpm
Number of pole pairs	1	
Number of commutator segments	7	
Weight of motor	190	g



(a)



(b)

Şekil 7.22 Maxon DC motor

Metal tutucu ile motor sabitlenmiş ve besleme uçları için bağlantı aparatı eklenmiştir. Geri besleme bilgisi için takogeneratör olarak aynı cins motor kullanılmıştır. Aşağıdaki şekilde görüldüğü gibi motorlar birbirine millerinden bağlanmışlardır. Mikrodenetleyici ile kontrol edilen motor döndüğünde miline bağlı olan diğer motoru döndereceği için ikinci motor generatör olarak çalışacak ve milindeki dönme hızı ile orantılı olarak uçlarında gerilim indükleneyecektir. Motorlar maksimum 12 volt ile çalıştırıldığında takogeneratör çıkış uçlarından da yaklaşık 11.6 volt elde edilmektedir. Elektriksel ve mekanik kayıplardan dolayı gerilim düşümü yaşanmaktadır. Bu gerilim direkt olarak analog ölçüm için mikrodenetleyiciye alınmaktadır. Ancak mikrodenetleyiciye maksimum 3.3 V analog giriş yapılabildiğinden takogeneratör uçlarından alınan gerilimin maksimum değeri 3,3 V olacak şekilde potansiyometre ile gerilim bölücü devre yapılarak düşürülmüştür. Mikrodenetleyicide analog olarak okunan bu gerilim, Daha önce takometre ile motordan alınan devir bilgileri yardımı ile oluşturulan denklemden geçirilerek o andaki motorun devri bulunmuş olunuyor.



Şekil 7.23. Motor ve takogeneratör

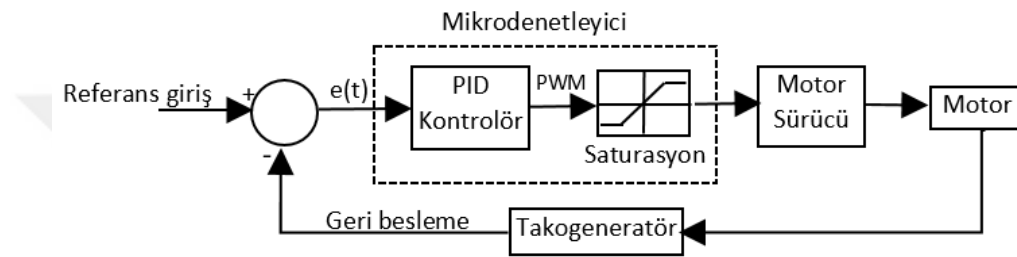
7.2.1.2 Motor Sürücü, Güç Kaynağı ve Mikrodenetleyici

Enkoderli motor için tasarlanan sürücü ve güç kaynağı devreleri bu motor için de kullanılmaktadır. Mikrodenetleyici olarak ise tüm sistemler de kullanılan Arduino Due kullanılmıştır.

7.2.2 PID Denetleyicinin Takogeneratörlü Motor Kontrolündeki Etkisi

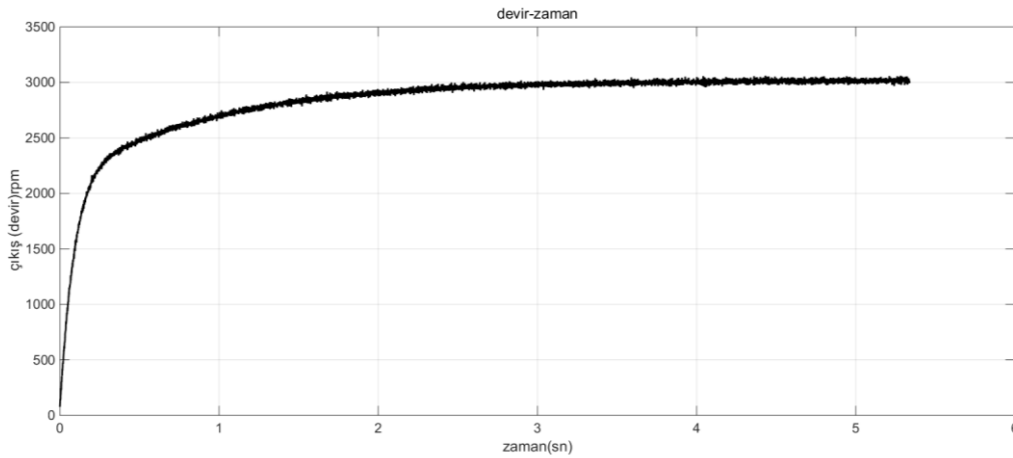
Motor kontrol edilmeye başlanmadan önce takogeneratör bağlı olarak, takometre ile devir ölçümü yapılmış ve 12 V gerilimde devir sayısı 3824 rpm olarak ölçülmüştür. Motor

gerilimi lineer olarak 1 V dan itibaren 1'er V aralıklarla arttırılarak devir sayıları ölçülmüş ve bu bilgiler matlab programına girilerek eğri uydurma yöntemi ile gerilim değerine karşın devir bilgisini verecek 2. Dereceden bir denklem çıkartılmıştır. Takogeneratörden alınan gerilim bilgisi mikrodenetleyici de bu denklemde yerine konularak motorun anlık devir bilgisi hesaplanmıştır. Bu bilgi, referans değerden çıkarılmış ve bulunan hata değeri PID algoritmasından geçirilmiştir. PID sonucu ise PWM değerine dönüştürülerek motor sürücüyeye iletilmiştir. Bu programda kontrol periyodu yaklaşık 950 mikro saniyedir.



Şekil 7.24 Takogeneratörlü motor PID blok şeması

Yukarıdaki Şekilde görüldüğü gibi keypad yardımı ile alınan referans bilgisi ile takogeneratörden alınan geri beslemenin farkı mikrodenetleyici içerisinde alınarak PID algoritmasından ve saturasyondan geçirilerek motor sürücüyeye PWM olarak verilmekte ve motor sürücüy de motoru sürmektedir. Devir bilgisi Arduino'nun seri portundan alınarak matlab'da çizdirilmiştir.

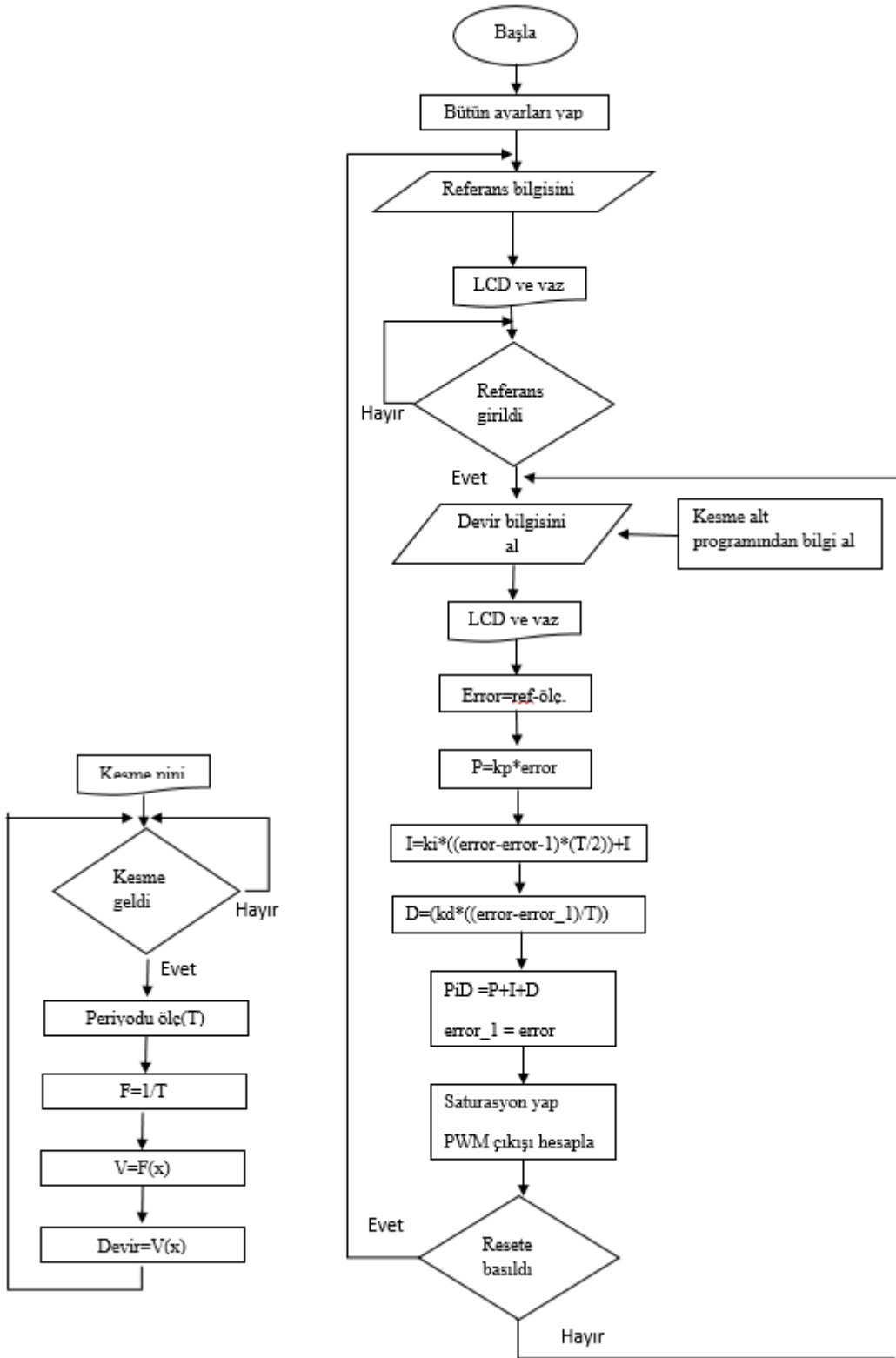


Şekil 7.25 Takogeneratörlü motorun kontrolsüz devir-zaman grafiği

Motor yukarıdaki şekilde görüldüğü gibi kontrolsüz olarak çalıştırıldığında sürekli duruma 4 sn de ulaşmaktadır. PID kontrol yöntemi ile bu süre düşürülerek motorun daha kıs sürede sürekli duruma kalıcı durum hatası olmadan ulaşması sağlanmıştır.

Oluşturulan yazılım ile oransal, oransal+integral, oransal+türev ve oransal+integral+türevsel denetleyici tipleri gerçekleştirilmiştir. Bu denetleyiciler ile çeşitli parametre değerlerinin sistem tepkisi üzerindeki etkileri incelenmiştir. Değiştirilen K_p , K_d ve K_i değerlerine karşın sistemin verdiği tepkiler aşağıdaki şekillerde görülmektedir. Bütün parametrelerde referans devir değeri 3000 rpm olarak alınmıştır. PID için Arduino IDE programında oluşturulan yazılımın akış şeması aşağıdaki şekilde görüldüğü gibidir.

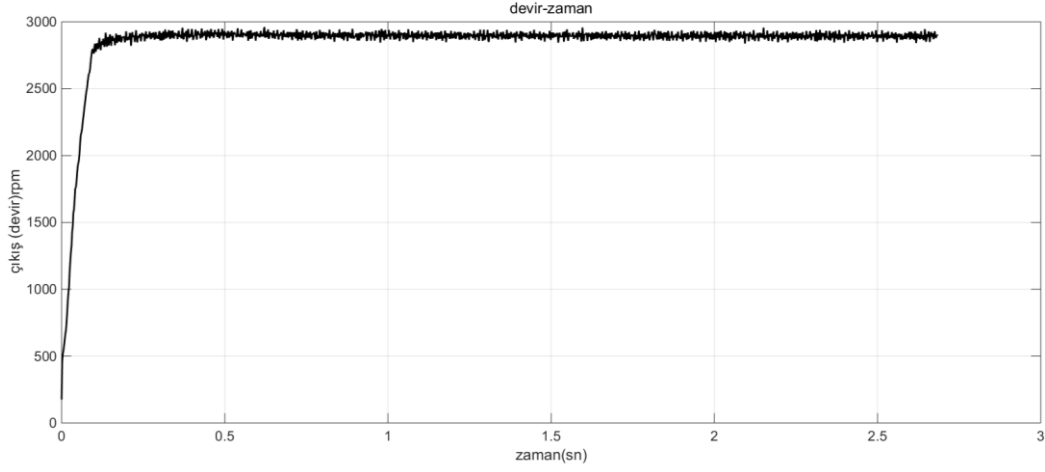




Şekil 7.26. Takogeneratörlü motor PID kontrol akış diyagramı

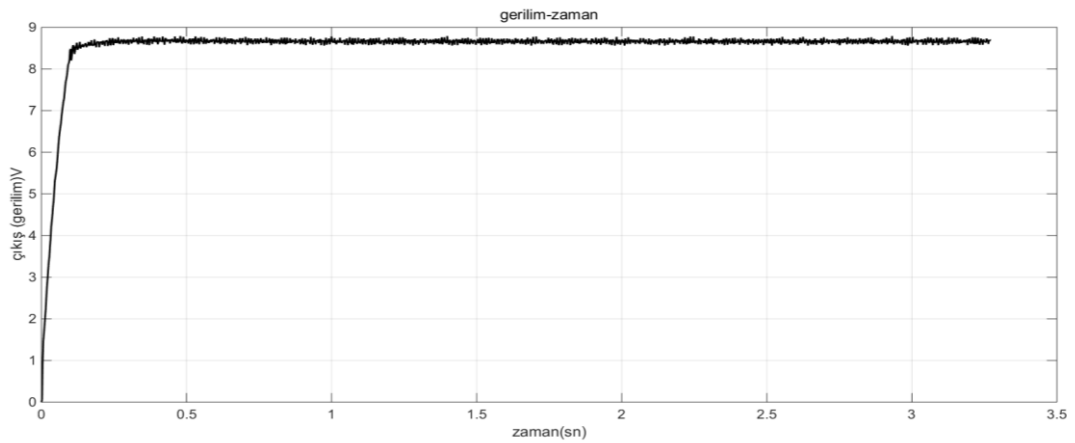
7.2.2.1. Oransal Kontrol (P)

Sistemin oransal kontrolü için yazılımsal olarak farklı birşey yapmadan ki ve kd parametrelerine sıfır değeri verilip kp değeri değiştirilebilir. Oransal katsayı 10 yapıldığında ve motor referans değeri 3000 rpm yapıldığında aşağıdaki grafikler elde edilmektedir.



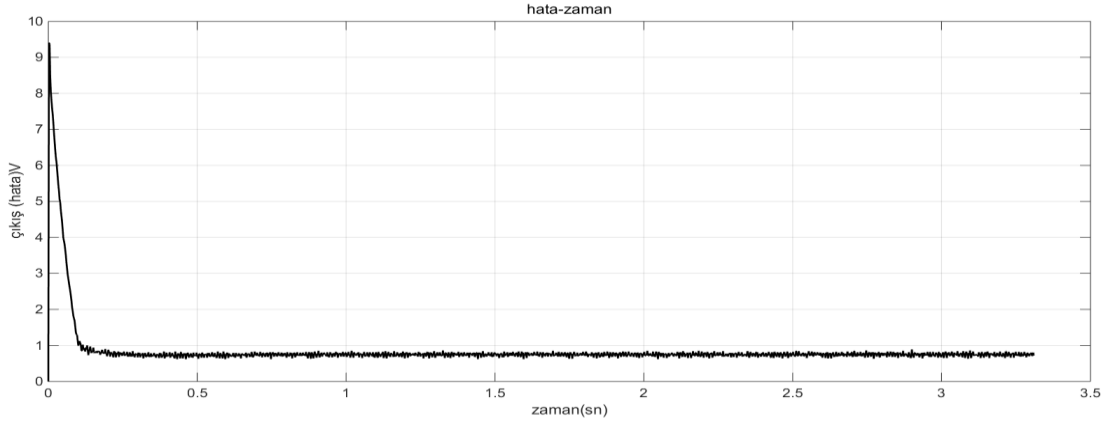
Şekil 7.27. Oransal kontrol devir-zaman grafiği

Şekillerde de görüldüğü gibi sadece oransal kontrol kullanıldığında sistem 194.9 ms gibi bir değerinde hızlıca sürekli duruma geçmiş ancak kalıcı durum hatası meydana gelmiştir istenilen set değerine ulaşamamıştır ve 2909 rpm de sabitlenmiştir.



Şekil 7.28. Oransal kontrol gerilim-zaman grafiği

Gerilim grafiğine bakıldığında 3000 devir için 9.41 V değeri gerekirken motora verilen gerilim 8.6 V’da sabitlenmiştir.

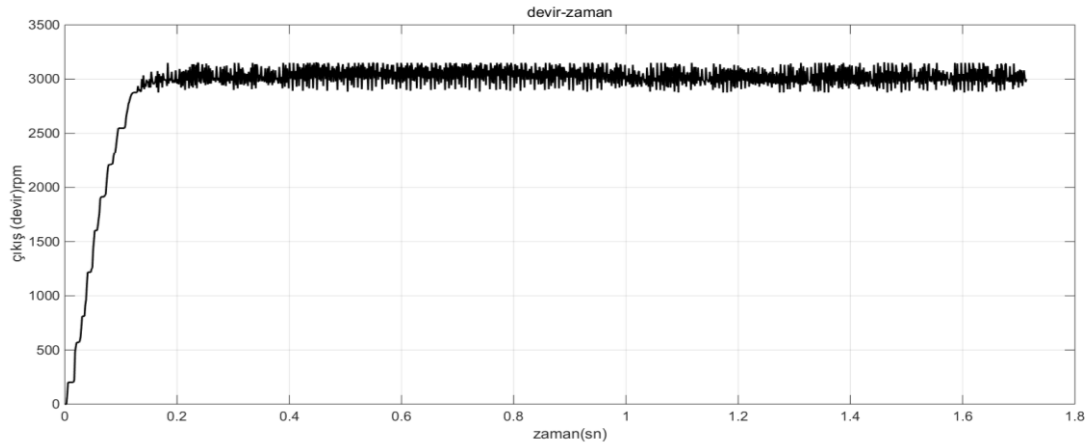


Şekil 7.29. Oransal kontrol hata-zaman grafiği

Hata grafiğine bakıldığında ise error değeri ilk başta sıfır olarak tanımlandığı için sıfırdan başladığı ama ilk döngüden itibaren arttığı ve motor yol aldıkça geçici zamanda gittikçe düştüğü görülmektedir. Sürekli duruma geldikten sonra ise 0.75 de sabitlenmiştir. Yani %3.3’lük bir hata söz konusudur.

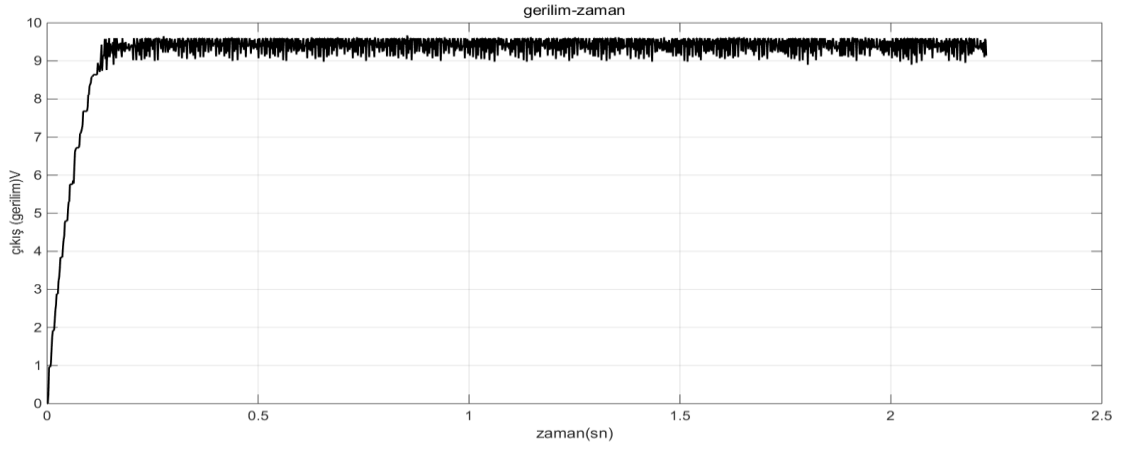
7.2.2.2. Oransal+Integral Kontrol (PI)

Oransal hatada meydana gelen kalıcı durum hatasını yok etmek için integral kontrol de eklenerek PI kontrolör oluşturulmuştur. $K_p:26$, $K_i:5$ ve $K_d:0$ olarak parametre ayarları yapılmış ve referans değer 3000 devir olarak ayarlanmıştır.



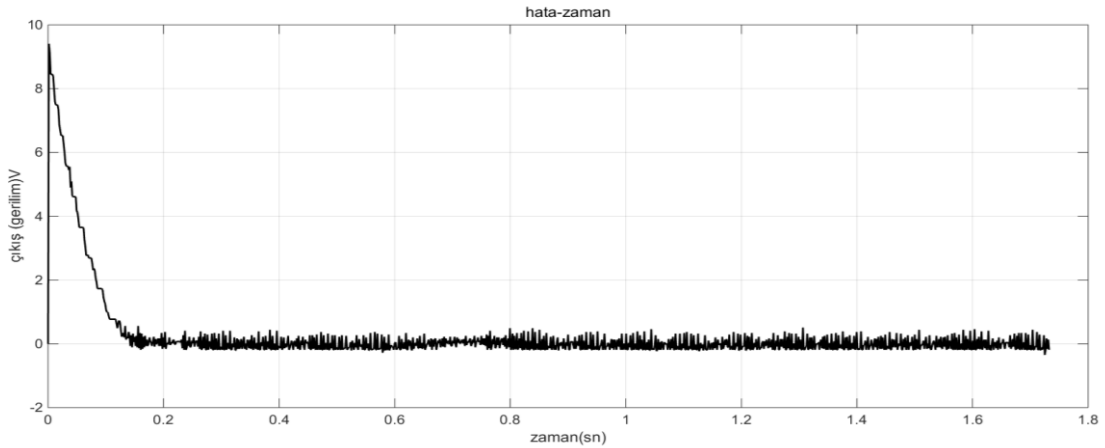
Şekil 7.30. Oransal+integral kontrol devir-zaman grafiği

Yukarıdaki şekilde motorun devir-zaman grafiğine bakıldığında 140 ms de sürekli duruma hatasız bir şekilde ulaştığı görülmektedir. Ancak motor set değeri etrafında salınım yapmaktadır. Bu salınımlar mekaniksel ve elektriksel gürültülerden kaynaklanmaktadır.



Şekil 7.31. Oransal+integral kontrol gerilim-zaman grafiği

Motorun 3000 devirde dönebilmesi için 9.41 V gerilim ile beslenmesi gerekir. Gerilim zaman grafiğine bakıldığında bu değer yakalandığı görülmektedir.

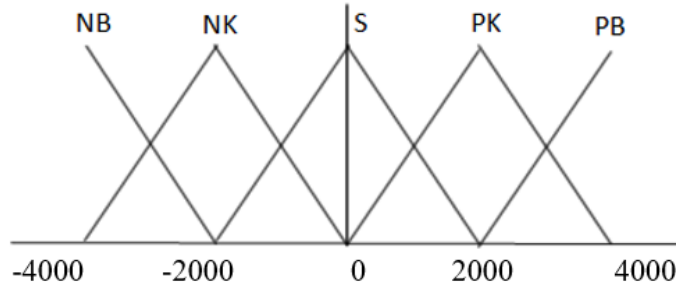


Şekil 7.32. Oransal+integral kontrol hata-zaman grafiği

Hata grafiğine bakıldığında ise PI kontrolle kalıcı durum hatasının ortadan kalktığı ve motorun aşma yapmadan sürekli duruma eriştiği görülmektedir. PI kontrol ile motorun kontrolünde geçici rejimde aşma yaşanmadığı için türevsel kontrol kullanılmamıştır.

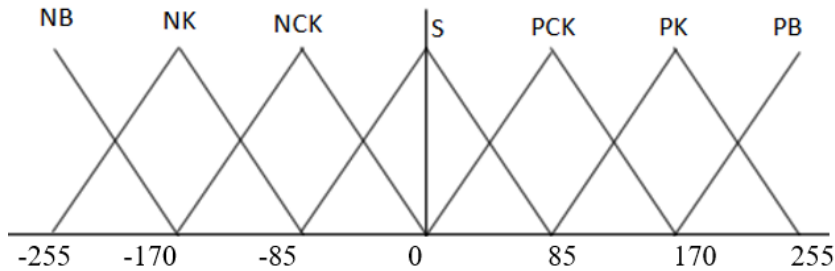
7.2.3. Bulanık Mantık Denetleyicinin Takogeneratörlü Motor Kontrolündeki Etkisi

Bulanık mantık kontrolü için iki giriş ve bir çıkışlı üyelik fonksiyonları oluşturulmuştur. Giriş olarak hata ve hatadaki değişim alınırken çıkış olarak ise PWM sinyali alınmıştır. Bu sistemin blok diyagramı şekil 7.16'da olduğu gibidir. Takogeneratörden alınan devir bilgisi ile keypad den alınan referans bilgisinin farkı mikrodenetleyicide alınarak hata değeri bulunmuştur. Bu hata değeri üyelik fonksiyonlarından geçirilerek üyelik dereceleri bulunup, daha sonra hatanın bir önceki değeri, hata değerinden çıkartılarak hatadaki değişim değeri bulunmuştur. Bu değer fonksiyonlardan geçirilerek hatanın değişimi için üyelik dereceleri hesaplanmıştır. Hem hata hem de hatadaki değişim üyelik dereceleri çıkarım ve durultma için alt fonksiyona gönderilip çıkış değeri hesaplandıktan sonra sonuç integratör ve saturasyondan geçirilerek PWM çıkışı üretilmiştir.



Şekil 7.33. giriş üyelik fonksiyonları

Şekilde de görüldüğü gibi 5 adet üçgen üyelik fonksiyonu kullanılmıştır. Bunlar NB: negatif büyük NK: negatif küçük S:sıfır PK: pozitif küçük ve PB: pozitif büyük 'tür. Burada motor maksimum 4000 devir olacak şekilde ayarlanmıştır. Bu fonksiyonlar hem hata, hem de hatanın değişimi için geçerlidir.



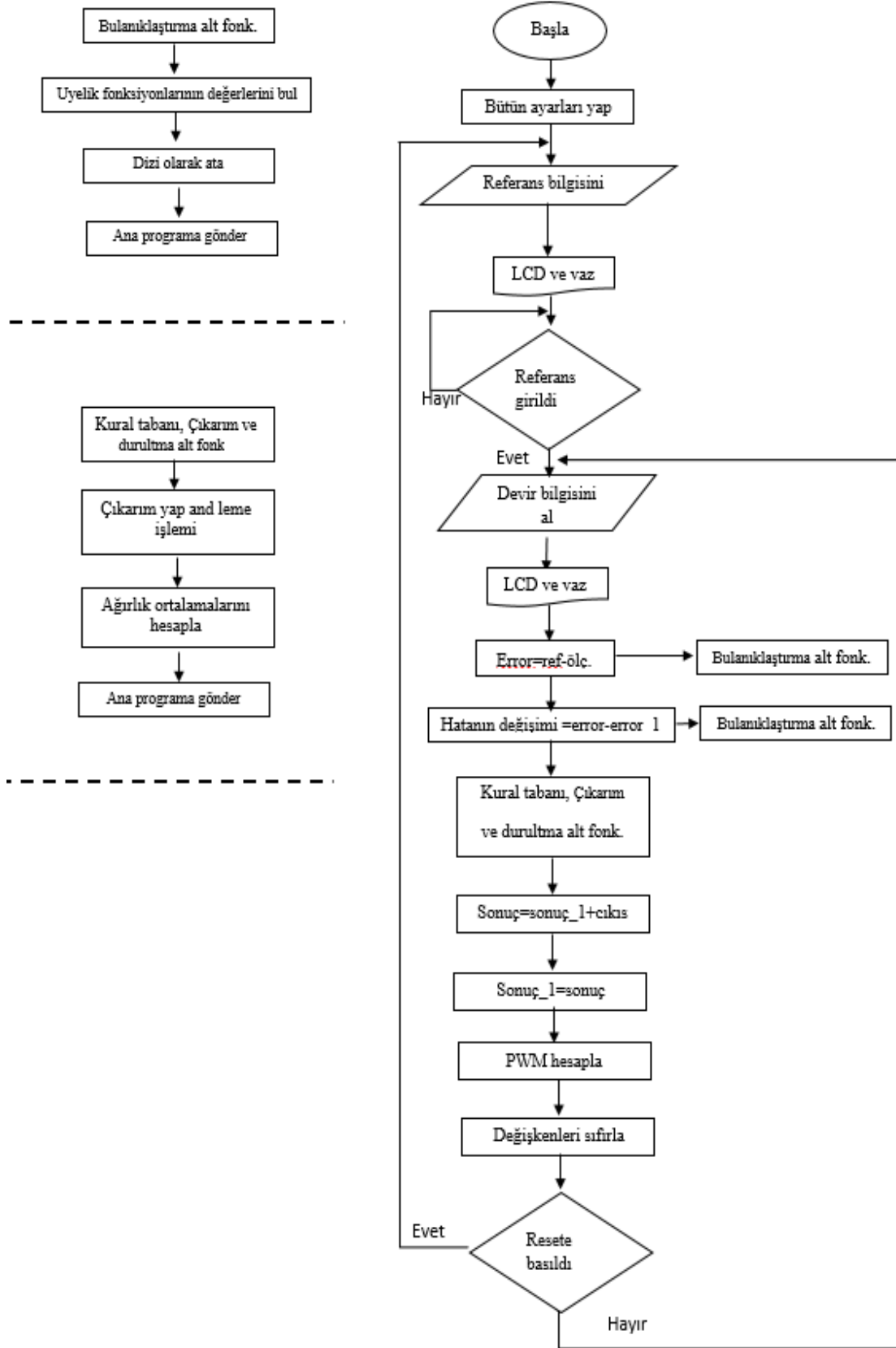
Şekil 7.34. çıkış üyelik fonksiyonları

Çıkış üyelik fonksiyonları 7 adettir. Buradaki üyelik fonksiyonları NB: negatif büyük, NK: negatif küçük, NCK: negatif çok küçük, S:sıfır, PCK: pozitif çok küçük, PK: pozitif küçük ve PB: pozitif büyük' tür.

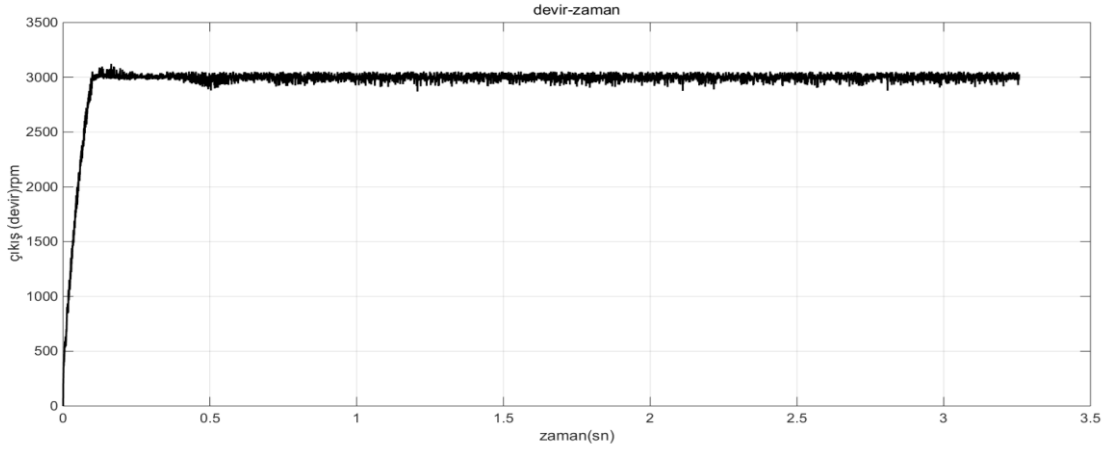
Tablo 7.3. Kural Tabanı

de e	NB	NK	S	PK	PB
NB	NB	NB	NK	NCK	S
NK	NB	NK	NCK	S	PCK
S	NK	NCK	S	PCK	PK
PK	NCK	S	PCK	PK	PB
PB	S	PCK	PK	PB	PB

Durulaştırma biriminde ağırlık ortalaması yöntemi ile bulunan bulanık mantık sonucu PWM değeri olarak motor sürücüye gönderilmiştir.

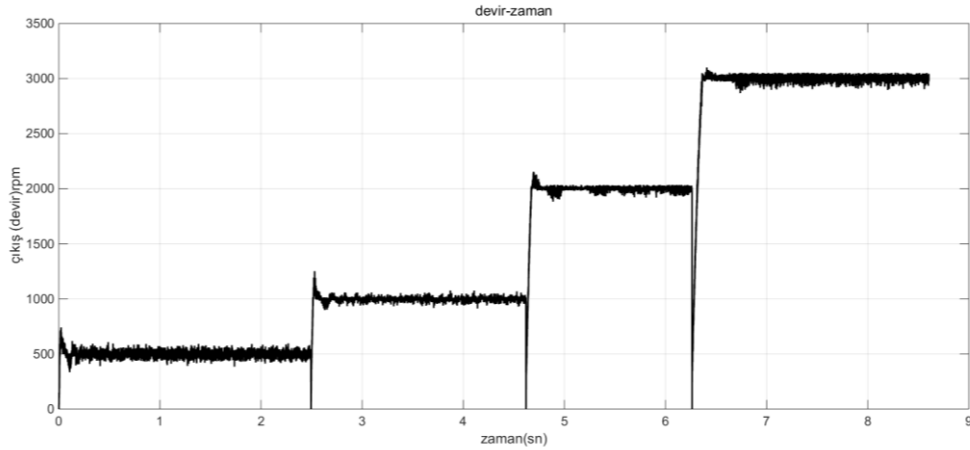


Şekil 7.35. Takogeneratörlü motor bulanık mantık kontrol akış diyagramı



Şekil 7.36. Bulanık Mantık kontrollü devir-zaman grafiği

Motoru bulanık mantık kontrol yöntemi uygulandığında 4 sn de sürekli zamana ulaşan motor, 99.1 ms gibi kısa bir sürede hatasız olarak referans değere ulaşmıştır. Değer etrafında oluşan saçaklar mekanik gürültüden dolayı meydana gelen genlik bozulmalarından kaynaklanmaktadır.



Şekil 7.37. Bulanık Mantık kontrolünde değişik devirler için devir-zaman grafiği

Motor çalışır vaziyette iken motorun referans devir bilgisi keypad ile değiştirildiğinde motorun sorunsuz bir şekilde referansı takip ettiği gözlemlenmiştir. Buradaki referans değerleri sırası ile 500, 1000, 2000 ve 3000'dir. Her referans değişiminde devrin sıfıra inmesinin sebebi yazılımın bu şekilde yapılmasındandır.

8. SONUÇLAR

Bu tezde, mikroişlemciler ve mikrodenetleyiciler'in tanımları yapılarak mimarileri, iletişim yolları, hafıza çeşitleri vb. hakkında bilgileri verilmiştir. Performans analizi için Microchip tarafından geliştirilen PIC16F877A, DSPİC30F3014 ve Atmel firmasının ürünü olan AT91SAM3X8E mikroişlemci tabanlı Arduino Due geliştirme platformu olmak üzere üç adet mikrodenetleyici kullanılmıştır. Sonuçlar hem simülasyonda hem de breadboard üzerinde kurulan devrelerde osiloskop ile gözlemlenmiştir.

Mikrodenetleyicilerin ne kadar hızlı çalıştıklarını gözlemleyebilmek için girişlerine analog sinyal vererek yine çıkışlarından analog sinyal elde etmek için ADC ve DAC yapıları kullanılmıştır. Bunun amacı; mikrodenetleyicilerin girişlerine değişik genlikli sürekli ve hızlı bir şekilde sinyal geldiğinde bunu çıkışa ne oranda doğru ve hızlı bir şekilde iletebileceğini gözlemleyebilmektir. Bunun içinde ilk başta üç mikrodenetleyiciye de 50 Hz frekanslı max 5 V genlikli tam dalga doğrultulmuş sinüs sinyali uygulanmıştır. üç mikrodenetleyicinin arasındaki performans farkını görebilmek için ise uygulanan sinüs sinyali 2.30 kHz'e çıkarılmış ve sonuçlar gözlemlenmiştir. Mikrodenetleyicilerin programlanabilme yeteneğini incelemek için P,PI, PID gibi kontrol algoritmalarının uygulanabilirliğine bakılmıştır ve PI algoritması gerçekleştirilmiştir.

DeneySEL sonuçlar doğrultusunda 84 MHz osilatör frekansı ve 12 bit çözünürlüklü Arduino Due'nin, 20 MHz osilatör frekanslı ve 10 bit çözünürlüğe sahip PIC16F877A'ya ve 20 MHz osilatör frekanslı ve 12 bit çözünürlüğe sahip DSPİC30F3014'e göre daha net ve daha hızlı sonuçlar verdiği görülmüştür. PIC deki gecikme 64 ms, dsPIC deki gecikme 22 ms ve Arduino da ise 16 ms olmuştur. ve Arduinonun ve dsPIC'in çözünürlükleri aynı olduğundan genlik bozulması yaşanmamış ancak 10 bitlik PIC de genlik bozulması oluşmuştur. Arduino Due'nin diğerlerine göre üstünlükleri;

- ✓ Çalışma hızının yüksek olması
- ✓ Program belleğinin fazla olması
- ✓ Analog kanal sayısı ve çözünürlüğünün yüksek olması
- ✓ Giriş/çıkış sayısının fazla olması
- ✓ DAC modülünün bulunması

- ✓ Devre kurulum kolaylığı
- ✓ Programlamanın devre çalışırken bile USB den yapılabilmesi

şeklinde sıralanabilir.

Arduino Due ile servo sistemlerin kontrolüne geçilmeden önce sistem ve mikrodenetleyici düşük frekanslı bir sistem olan sıcaklık kontrolü için programlanarak sonuçlar incelenmiştir. Sıcaklık kontrolü için şekil 6.5 de görüldüğü gibi bir ortam oluşturulmuştur. İstenilen referans sıcaklıkta ortamın ısıtılabilmesi için kullanılan rezistans triyak sürme devresi ile sürülerek PID çıkışlarına göre kontrolü sağlanmıştır. Süreç kontrolünde sistemin tepkisi ve sürekli duruma erişme süresi yavaş olduğundan sistem istenildiği şekilde kontrol edilmiştir. Referans değer bütün uygulamalar için 60°C alınmıştır. On/off kontrolde sistem cevabı beklenildiği gibi referans değer etrafında salınım yapmıştır. Bu salınımı azaltmak için hiteresiz değeri eklendiğinde şekil 6.15 de olduğu gibi salınımların azaldığı görülmüştür. Sadece oransal kontrol kullanıldığında ve Kp katsayısının değeri başlangıçta düşük tutulduğunda referans değere erişmeden yaklaşık 52 derecede sabitlenmiştir. Bu katsayı arttırıldığında ise sıcaklık kısa sürede referans değeri aşarak 76 dereceye kadar çıkmış sonrasında referans değerinin altında 57 derecede sabitlenmiştir. Referans değere ulaşma süresi 50 sn sürmüştür. Kp katsayısı sadece yükselme zamanını etkilediğinden sürekli duruma pek katkısı bulunmamaktadır. Bu durumu düzeltmek için oransal+integral kontrol kullanılmıştır. Bu kontrol ile sistemin kalıcı durum hatası ortadan kaldırılmış ve sıcaklık 60 °C de sabitlenmiştir ancak sistemin geçici durumu bozulmuştur. Oransal+türevsel kontrol kullanıldığında ise sistemin geçici durumunda oluşan salınımların azaldığı, fakat sürekli durumu düzeltmediği ve kalıcı durum hatasının oluştuğu görülmüştür. Sistemin hem kalıcı durum hem de geçici durumunu düzeltmek ve istenilen referansa en kısa zaman ve en iyi doğrulukla ulaşabilmek için oransal+integral+türevsel kontrol kullanılmıştır. Sonuç olarak verilen değişik PID parametreleri ile sistemin geçici rejimde salınımlarının azaldığı ve referans değerinde değerinin sabitlendiği görülmüştür.

Servo sistemlerin kontrolü için iki adet DC motor kullanılmıştır. Bunlardan ilki Enkoderli, reüktörlü DC motor, diğeri ise aynı cins ikinci bir motorun takogeneratör olarak kullanıldığı redüktörsüz DC motordur. Her iki motorda da hem PID hem de ileri kontrol yöntemi olan bulanık mantık denetleyici kontrolü gerçekleştirilmiştir. Geri besleme

elemanı olarak kullanılan enkoderin, birbirine 90 derece faz farkı bulunan ve kare dalga sinyali veren iki adet çıkışı bulunmaktadır. Motorun hızına göre bu kare dalgaların frekansı değişmektedir. Motor hızlandıkça kare dalgaların frekansı artarak sıklaşmaktadır. Arduino ile yazılan programda bu kare dalgaları hız bilgisine çevirebilmek için kesme alt programı kullanılmıştır. Bu alt program ile gelen iki yükselen kenar arasındaki süre ölçülüp frekans bilgisine dönüştürülmektedir. Daha önceden motor değişik gerilimlerde çalıştırılarak bu gerilimlerde enkoder çıkışı osiloskopta incelenmiş ve bir frekans gerilim denklemi oluşturulmuştur. Buradan bulunan gerilim sayesinde devir bilgisine oranlama ile ulaşılmıştır. Buradan elde edilen devir ile referans devir farkı alınarak hata değeri bulunmuştur ve bu hata değeri PID kontrolden geçirilerek motor kontrolü sağlanmıştır. Motor kontrolsüz bir şekilde çalıştırılmış ve sürekli zamana yaklaşık 2 sn süresinde ulaştığı görülmüştür. Oransal, oransal+türevsel ve oransal+integral+türevsel uygulamaların sonuçları incelendiğinde oransal kontrolde 62 ms de referans değerine ulaştığı ve 161 ms de sürekli duruma geçtiği ancak 3 rpm lik bir kalıcı durum hatası olduğu görülmektedir. Ayrıca 90 rpm e kadar bir aşma gerçekleşmiştir. PD kontrol kullanıldığında bu aşmanın ortadan kalktığı ve sistemin 62.5 ms de kalıcı duruma geçtiği fakat yaklaşık 3 rpm lik kalıcı durum hatası olduğu grafiklerden çıkarılmaktadır. PID kontrol uygulandığında ise şekil 7.13 de görüldüğü gibi geçici rejimde aşma olmadan ve kalıcı durum hatası meydana gelmeden 101 ms de sürekli duruma geçmiştir.

Bulanık mantık kontrolde kaynak 62 den yararlanılarak üçgen üyelik fonksiyonları tercih edilmiştir. 5 adet giriş üyelik fonksiyonu, 7 adet de çıkış üyelik fonksiyonu kullanılmıştır. Çıkarım mekanizması için Tablo 7.1' deki kural tabanı oluşturulmuştur. Durultucu biriminde ağırlık ortalaması yöntemi kullanılmıştır. Sonuçlar incelendiğinde motor, istenilen referansa 90.28 ms gibi kısa bir sürede ulaşmıştır ancak referans değer etrafında salınımlar yapmıştır. Aynı sonuç değişik referans değerleri girildiğinde de meydana gelmiştir. Bunun iki sebebi vardır bunlardan ilki motorun bir turda verdiği puls sayısı düşük olduğundan motorun gerilimine göre nonlineer olarak ürettiği kare dalga frekansının çok küçük olmasıdır. Bunun sonucunda örneğin kare dalga periyodu 1.5 ms olduğunda motor devir bilgisi okunana kadar 350 mikro saniye olan program döngüsü 4.2 kez döndüğü için bu bilgi geç gelmektedir. İkincisi ise kare dalganın düşen kenarının salınımlı olmasından dolayı periyot bilgisinin bu salınıma orantılı olarak değişmesidir.

Takogeneratör ile kontrol edilen DC motorun devir bilgisi bir takometre yardımı ile değişik gerilim değerleri için ölçülmüş ve aralarındaki ilişkiye göre matlabda eğri uydurma yapılarak yedinci dereceden bir denklem oluşturulmuştur. Motor gerilim bilgisi takogeneratörden alınarak gerilim bölücüden geçirildikten sonra Arduino nun ADC girişine uygulanmıştır. Buradan alınan gerilim bilgisi ile keypad den girilen set değerinin farkı alınarak hata değeri bulunmuş ve PID algoritmasından geçirilmiştir. Gerilim bilgisi denklemde yerine konularak devir bilgisi bulunmuş ve bu bilgi seri porttan alınarak matlabda çizdirilmiştir. Mekanik ve elektriksel gürültüleri azaltmak ve daha doğru okuma yapabilmek için gerilim 40 defa okunarak ortalaması alınmaktadır. Bu da denetleyici periyodunun 1.4 ms olmasına sebep olmaktadır. Motor kontrolü için oransal kontrol kullanılıp kp değeri 10 yapıldığında motorun 200 ms de sürekli duruma kalıcı durum hatası ile geçtiği görülmektedir. Bu offset değerini ortadan kaldırmak için PI kontrolör kullanılıp katsayılar Kp:26 ve Ki:5 'e ayarlandığında motorun 140 ms de kalıcı durum hatası olmadan sürekli duruma geçtiği görülmektedir. Motorda herhangi bir aşma olmadığından türevsel kontrolöre gerek kalmamıştır. Daha doğru bir ölçüm ve daha yakın parametreler ile sistem kontrolü daha net bir şekilde elde edilebilir.

Bulanık mantık kontrolde ise PID kontrolöre göre daha net ve hızlı sonuçlar elde edilmiştir. Motor 99.1 ms de kalıcı durum hatası olmadan set değerine oturmuştur. Motor kontrolsüz bir şekilde çalıştırıldığında 4 sn de sürekli duruma eriştiği dikkate alınırsa PID ve bulanık mantık ile daha hızlı sonuçlar alındığı kabul edilebilir.

KAYNAKLAR

- [1] Savas, Y., ,1988,Sayısal Göstergeli Sıcaklık Ölçme ve Kontrol Cihazının Tasarım ve Yapımı", Yüksek Lisans Tezi, G. Ü. Fen Bil. Enstitüsü, Ankara, 1-44.
- [2] Kalender, O., 1991,MC 6802 Tabanlı Oransal Sıcaklık Kontrol Devresinin Tasarımı ve Yapımı, Yüksek Lisans Tezi, G. Ü. Fen Bil. Enstitüsü, Ankara, 1-77.
- [3] Basar, E., 2002 PIC16C74B Mikrodenetleyici Tabanlı 5 Bölgeli Isıtma Kontrol Sistemi", Yüksek Lisans Tezi, G. Ü. Fen Bil. Enstitüsü, Ankara, 1-2.
- [4] Duran, F., 2001,PIC Mikrodenetleyicisi ile DC Motor Hız Kontrolü, Yüksek Lisans Tezi, G. Ü. Fen Bil. Enstitüsü, Ankara, 23-47.
- [5] Gök,İ.,2005, Faz Kaymalı Darbe Genislik Modülasyonlu DC-DC Dönüştürücü Kullanan Akü Sarj Cihazı, Yüksek Lisans Tezi, G. Ü. Fen Bil. Enstitüsü, Ankara,1-3.
- [6] Yılmaz, F.H. 2008, "AC ve DC servo sistem eğitim setinin gerçekleştirilmesi", Selçuk... Üniversitesi Fen Bilimleri Enstitüsü Yüksek lisans tezi
- [7] Coskun, İ., Isık M.F., 2004,, "Servo motorun Mikrodenetleyici ile Konum Ve Hız Denetimi", Gazi Üniversitesi Fen Bilimleri Dergisi, Cilt 17, No 3, 115-125
- [8] Microchip, "VF Control of 3-Phase Induction Motors Using PIC16F7X7 Microcontrollers", 2004 Microchip Technology Inc.
- [9] C.Rech, H. Pinheiro, H. A. Grindling, H. L. Hey, J. R. Pinheiro, "Analysis and Design of a Repetitive Predictive-PID Controller for PWM Inverters", Federal University of Santa Maria ZIP CODE: 97 105-900 - Santa Maria, RS – Brazil
- [10] Gulabi, A., 2007 ,"Development of an Embedded SCADA System with PLC and Java .Application for Synchronous Operation of Standard Servo Drives"., M.S. Thesis .Hamburg University, Hamburg,.
- [11] Honeywell DC1000 Series General Purpose universal digital controllers product specification sheet
- [12] Emko Elektronik Sanayi ve Ticaret A.Ş Kullanım Klavuzu. TÜR ESM-4435 02 V05 07/14
- [13] Çobanoğlu, B., Mikroşlemciler Dersi, Pic Mikroşlemciler ve programlama ders notu, Sakarya Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü,
- [14] Soylu E., Öz K., Mikroşlemciler Ders Notu, Karabük Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü, Karabük, 283s.

- [15] Türkoğlu, İ., 2010, Mikro bilgisayarlı sistem tasarımı, Ders Notu, Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü, Elazığ, 157s.
- [16] Milli Eğitim Bakanlığı, 2012, Elektrik Elektronik Teknolojisi, Mikroişlemci ve Mikrodenetleyiciler, 523EO0019, Ankara, 67s.
- [17] Ünal İ., Mikrodenetleyiciler Ders Notu, Akdeniz Üniversitesi Teknik Bilimler Meslek Yüksek Okulu Elektronik ve Otomasyon Bölümü Mekatronik Programı, 73s.
- [18] Çobanoğlu, B., Mikroişlemciler ve Mikrodenetleyiciler II ders notu, Sakarya Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü, 33s.
- [19] Topaloğlu, N., Gürgünoğlu, S., 2003, Mikroişlemciler ve Mikrodenetleyiciler, Seçkin Yayıncılık, Ankara, 304s
- [20] Dilmaç, S., 2004, 8051 ve ADuC8xx Eğitim Notları, Elektronik Sanayi ve Ticaret A.Ş., İstanbul, 385s.
- [21] Milli Eğitim Bakanlığı, 2006, Mikroişlemci ve Mikrodenetleyiciler, Mesleki Eğitim ve Öğretim Sisteminin Geliştirilmesi Projesi, Ankara, 57s.
- [22] Özcerit, A. T., Çakıroğlu, M., Bayılmış, C., 2008, 8051 Mikrodenetleyici Uygulamaları, Papatya Yayıncılık Eğitim, İstanbul, 416s.
- [23] Çobanoğlu B., “PIC mikrodenetleyiciler ve Programlama”, Sakarya Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü.
- [24] T.C Millî Eğitim Bakanlığı Elektrik-Elektronik Teknolojisi,”Mikro İşlemci ve Mikrodenetleyiciler”, 523eo0019 Ankara, 2012
- [25] URL-1, <http://www.microchip.com/pagehandler/en-us/products/picmicrocontrollers>, 11 Eylül 2015
- [26] Çolak İ., Şahin M., Esen Z., 2013, ” Artificial Neural Networks Controller Algorithm Developed for a Brushless DC Motor.”, 12th International Conference on Machine Learning and Applications, IEEE.
- [27] Alonso G. A., Ramírez A., Muñoz R., 2010, ” Hardware Implementation of the Wavelet Transform Coupled with Artificial Neural Network for Quantification Purposes.”, Pan American Health Care Exchanges – Pahce. Conference, Workshops, and Exhibits. Cooperation / Linkages. Inter Cambios De Cuidado Médico Panamericanos. Conferencia, Talleres y Exhibiciones. Cooperación / Enlaces., Lima, Peru, IEEE, March 15 – 19.
- [28] URL-2, http://www.robotiksistem.com/arduino_nedir_arduino_ozellikleri.html ,18 Eylül 2015
- [29] URL-3, <http://www.arm.com/products/processors/index.php> 19 Eylül 2015
- [30] URL-4, http://www.inverter-plc.net/inverter/vektör_kontrol.html, 25 Eylül 2015

- [31] URL-5, <http://arduino.cc/en/Main/Products>, 25 Eylül 2015
- [32] Munadi ve Akbar M.A., 2014,” Simulation of Fuzzy Logic Control for DC Servo Motor using Arduino based on Matlab/Simulink.”, International Conference on Intelligent Autonomous Agents, Networks and Systems Bandung, Indonesia, August 19-21.
- [33] Franc H. Ve Šafariš R.,2010,” ARM-Cortex Microcontroller fuzzy position control on an automatic door test-bed.”, 19th International Workshop on Robotics in Alpe-Adria-Danube Region – RAAD, Budapest, Hungary, June 23–25.
- [34] Çiçek S., 2009, CCS C ile PIC Programlama, Altaş Yayıncılık ve Elektronik, İstanbul, 600s.
- [35] Gökbulut M., Kontrol sistemlerinin analiz ve tasarımı, ders kitabı seçkin yayıncılık, Ankara.
- [36] Kıncay O. , Yapılarda otomasyon ve enerji yönetimi, otomatik kontrol ders notu,Yıldız Teknik Üniversitesi,İstanbul
- [37] Gürsel K., Üiversal gelişmiş kontrol cihazları, PID parametrelerinin ayarlanması, Auto-Tune özellikleri, Elimko Elektronik İmalat ve Kontrol LTD.ŞTİ, Ankara,18s
- [38] MEGEP, 2009, Endüstriyel Otomasyon Teknolojileri, PID kontrolü, Mesleki Eğitim ve öğretim Sisteminin Güçlendirilmesi Projesi, Ankara
- [39] H.P.Huang. , January 2002, On line adaptive tuning for PID controllers , IEEE Control Theory and Applications,. vol. 149, pp. 60-67.
- [40] M.A. Goodrich, Jun 1999, Model predictive satisficing fuzzy logic control ,IEEE Transactions on Fuzzy Systems,. vol. 7, pp. 319- 332.
- [41] C.C. Lee, April 1990 ,Fuzzy Logic in control systems: fuzzy logic controller : 2”, IEEE Transactions on Systems,. vol. 20, pp. 419- 435.
- [42] H.S.Kim, December 2008 ,State-Feedback Control Design for Fuzzy Systems Using Lyapunov Functions With Quadratic Dependence on Fuzzy Weighting Functions”, IEEE Transactions on Fuzzy Systems,. vol. 16, pp. 1655-1663.
- [43] A.Guez, L.E. James, K. Moshe, April 1988,Neural Network Architecture for Control”, IEEE Control Systems Magazine,. vol. 20, pp. 22- 25.
- [44] Taşçı G., Küçükyıldız G., Ertunç H.M., Ocak H., Ekim 2012, PID ve bulanık mantık ile DC motorun gerçek zamanda DSPIC tabanlı konum kontrolü, Otomatik Kontrol Ulusal Toplantısı, Niğde

- [45] Çiftçi, H., 2002, Fuzzy Logic Function Approximation for Some Mathematical Functions, Lisans Tezi, Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği, Eskişehir.
- [46] Özger, H., Bulanık Mantık ve Modelleme, ders notu, İstanbul Teknik Üniversitesi, Hidrolik Anabilim Dalı, İstanbul
- [47] Akpolat, H., Bulanık Mantık Ders Notları, Fırat Üniversitesi, Elazığ.
- [48] Doğan, İ., 2002, "Microcontroller-Based Temperature Monitoring And Control", Newnes, England, 171-219 .
- [49] Ak, M. , Tuna, M. Ve Ergün Amaç A. ,2007, Bir Fazlı Tetikleme devreleri için eğitim amaçlı kullanıcı arayüzü tasarımı, UMES'07-Ulusal Teknik Eğitim, Mühendislik ve Eğitim Bilimleri Genç Araştırmacılar Sempozyumu 309-312
- [50] NXP Semiconductors BT138 series Triacs datasheet.
- [51] Texas Instruments moc3020 thru moc3023 optocouplers/optoisalators datasheet, soes025a – october 1986 – revised april ,1998
- [52] Tıǧlı, İ.C., 2014, Mutfak tipi bir mikrodalga fırının mikroişlemci ile kontrolü, BAÜ Fen Bil. Enst. Dergisi Cilt 16(2) 87-99
- [53] Irmak E., Calpbinici A., Güler N.,2012 Orta Ölçekli Bir İşletmenin Enerji İzleme Sisteminin Tasarlanması, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi Cilt 18, Sayı 2, Sayfa 123-131
- [54] Micro Commercial Components Semiconductors PB605 thru PB610 Single Phase Bridge Rectifier datasheet
- [55] Texas Instruments, 4N35,4N36,4N37 optocouplers datasheet, soes021c – november 1981 – revised april 1998
- [56] Texas Instruments LM35 Precision Centigrade Temperature Sensors datasheet, snis159f –august 1999–revised january 2016
- [57] Siemens Training Education Program, STEP 2000 Series, “Basics of DC drives and related products”.
- [58] Oguntoyinbo, O. 2009, PID control of brushless dc motor and robot trajectory planning and simulation with matlab/simulink, Technology and Communication.
- [59] Yedamale, p., 2003, Microchip Technology Incorporated, Brushless DC motor Fundamentals.
- [60] Kocaoğlu S., Kuşçu H., PIC ile DC motorun hız ve konum kontrolü için gerekli PID parametrelerinin belirlenmesi ve bir uygulama.

- [61] Bulut M., Kurt B. Ve Demirtaş M.,2004, Genetik-Bulanık Kontrolörün DC Motora Uygulanması, Journal of Polytechnic, Vol: 7 No: 4 pp. 277-283.
- [62] Akyazı, Ö. , Zenk H. ,Akpınar A.S., 16-18 May 2011, Farklı Bulanık Üyelik Fonksiyonları Kullanarak Sürekli Mıknatıslı DA Motorunun Hız Denetiminin Gerçeklenmesi, 6 th International Advanced Technologies Symposium (IATS'11), Elazığ, Turkey
- [63] Köse F., Kaplan K. ve Ertunç M., 26-28 Eylül 2013, PID ve Bulanık Mantık ile DC Motorun Gerçek Zamanda STM32F407 Tabanlı Hız Kontrolü, Otomatik Kontrol Ulusal Toplantısı, Malatya.
- [64] URL-6 <https://www.pololu.com/product/1444>, 20 Nisan 2016
- [65] Akolat Z.H. Bulanık Mantık ve Kontrol Uygulamaları Ders notu, Fırat Üniversitesi, Mekatronik Mühendisliği, Kontrol Sistemleri Anabilimdalı
- [66] Microchip, dsPIC30F3014, dsPIC30F4013 datasheet,2004, 220 s.
- [67] Maxon DC motor Teknik datasheet, nisan 2006, 1s

ÖZGEÇMİŞ

Ahmet TOP

Sürsürü Mahallesi Halit Hoca Efendi Bulvarı No:46 Merkez/ELAZIĞ

E-Posta : ahmet_top18@hotmail.com

KİŞİSEL BİLGİLER

Eğitim Durumu : Üniversite (Mezun) – Yük. Lisans (devam ediyor)
Medeni Durumu : Evli
Askerlik Durumu : Yapıldı (2014)
Yuruk : T.C.
Doğum Tarihi : 25.10.1988
Doğum Yeri : Adıyaman

EĞİTİM BİLGİLERİ

Üniversite (Yüksek Lisans) Fırat Üniversitesi
02.2014- devam ediyor Teknoloji Fakültesi, Elektrik/Elektronik Mühendisliği

Üniversite (Lisans) Fırat Üniversitesi - (Örgün Öğretim)
3.39 / 4 Mühendislik Fakültesi, Elektrik/Elektronik Mühendisliği
09.2007- 07.2011

Lise ELAZIĞ ANADOLU LİSESİ 4,4 / 5
07.2006

İŞ TECRÜBESİ

Araştırma Görevlisi
Fırat Üniversitesi Teknoloji Fakültesi
2013- | Elazığ - Türkiye | Tam Zamanlı

Elektrik Mühendisi
Çamlıbel EDAŞ
08.2011 - 06.2012 (11 ay) | Sivas - Türkiye | Tam Zamanlı

Stajyer Mühendis
Zorlu O&M Enerji Tesis. İşletme ve Bakım Hiz. A.Ş.
07.2010 - 09.2010 (2 ay) | Osmaniye | Stajyer

EK

1-) Motor kontrol için PID kontrol programı

```
#include <pwm01.h>
uint32_t pwm_duty = 32767;
uint32_t pwm_freq2 = 10000;
#include <Key.h>
#include <Keypad.h>

#include <LiquidCrystal.h>

const byte Rows = 4; //number of rows on the keypad i.e. 4
const byte Cols = 3; //number of columns on the keypad i.e. 3

char keymap[Rows][Cols] =
{
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte rPins[Rows] = {A1, A2, A3, A4}; //Rows 0 to 3
byte cPins[Cols] = {A5, A6, A7}; //Columns 0 to 2
Keypad kpd = Keypad(makeKeymap(keymap), rPins, cPins, Rows, Cols);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
char deger[3] = {48, 48, 48};

bool izin = 1;
char keypressed;
int referans_deger = 9999;
int darbe = 0, i = 0, k = 0;
float T = 999, F = 0, V = 0;
float ilk = 0;
float son = 0;
float Devir;
float T_son;
float istenen_deger = 0;
float cikis_v = 0;
float olculen_deger = 0;
float kp = 30, kd = 0, ki = 0.1;
float P = 0, I = 0, D = 0;
float PID = 0;
float error_1 = 0;
```

```

float error = 0;
float Tp = pow(12, 10 ^ -3);

int keypad_oku() {
  if (izin) {
    if (keypressed != '#' && keypressed != '*' && i < 3)
      { deger[i] = keypressed;
        i = i + 1;
        k = k + 1; }
    if (keypressed == '#') {
      referans_deger = (((deger[0] - 48) * 100) + ((deger[1] - 48) * 10) + ((deger[2] - 48) *
1));
      int carpan = pow(10, (3 - k));
      referans_deger = referans_deger / carpan;
      lcd.clear();
      deger[0] = 48;
      deger[1] = 48;
      deger[2] = 48;
      i = 0;
      k = 0;
      izin = 0;

    } }
  if (keypressed == '*') {
    deger[0] = 48;
    deger[1] = 48;
    deger[2] = 48;
    referans_deger = 9999;
    i = 0;
    k = 0;
    izin = 1;
    error = 0;

    detachInterrupt(digitalPinToInterrupt(28));
    setup();
  }
  return referans_deger;
}

void setup()
{
  pinMode(29, OUTPUT);
  pinMode(8, OUTPUT);
  pwm_set_resolution(16);
  pwm_setup( 7, pwm_freq2, 2);
  Serial.begin(250000); // haberleşme hızı tanımlanarak başlatıldı.
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.clear();

```

```

lcd.print("Devir Sayisini");
lcd.setCursor(0, 1);
lcd.print("Giriniz(rpm):");
analogWrite(8, 0);
attachInterrupt(digitalPinToInterrupt(28), encoder_kesme, RISING);
while (referans_deger == 9999) {
  keypressed = kpd.getKey();
  if (keypressed) {
    keypad_oku();
  } }

```

```

void loop()
{
  digitalWrite(29, HIGH);
  digitalWrite(29, LOW);
  keypressed = kpd.getKey();
  if (keypressed) {
    keypad_oku(); }

```

```

lcd.setCursor(0, 0);
lcd.print("Ref.Deg:");
lcd.print(referans_deger);
lcd.print("rpm");
lcd.setCursor(0, 1);
lcd.print("Olc.Deg:");
lcd.print(T);
lcd.print("rpm ");

```

```

error = (referans_deger - Devir);
P = (kp * error);
I = (ki * ((error + error_1) * Tp * 0.5)) + I;
D = (kd * ((error - error_1) / Tp));
PID = P + I + D;
error_1 = error;

```

```

cikis_v = round(((255 * PID) / 200));
if (cikis_v >= 255) {
  cikis_v = 255;
}
else if (cikis_v <= 10) {
  cikis_v = 0;
}
else {
  cikis_v = round(((255 * PID) / 200));;
}

```

```

}
cikis_v = map(cikis_v, 0, 255, 0, 65534);
pwm_write_duty( 7, cikis_v );

Serial.println(Devir);

}

void encoder_kesme() // kesme fonksiyonu oluşturulmaktadır
{
  detachInterrupt(digitalPinToInterrupt(28));
  darbe++;
  if (darbe == 1)
  {
    ilk = micros();
  }
  if (darbe == 10 && darbe - 9 == 1) {
    son = micros();
    T = (son - ilk) / 9;
    T = T / 1000;
    darbe = 0;
    ilk = 0;
    son = 0;
  }

  if (T <= 0 && T >= 6) {
    F = 0;
  }
  else {
    F = (1 / T) * pow(10, 3);
  }
  V = (-1.45596683944356e-14 * pow(F, 4) + 1.46425540522710e-10 * pow(F, 3) -
4.28979355504581e-07 * pow(F, 2) + 0.00450312769045400 * F + 0.136337410134447);
  if (V <= 0.36) {
    V = 0;
    Devir = 0;
  }
  else {
    Devir = (V * 200) / 12;
  }

  attachInterrupt(digitalPinToInterrupt(28), encoder_kesme, RISING);
}

```

2-) Motor için Bulanık Mantık Kontrol Programı

```
#include <DueTimer.h>
#include <LiquidCrystal.h>
#include <Key.h>
#include <Keypad.h>
#include <pwm01.h>
uint32_t pwm_duty = 0;
uint32_t pwm_freq2 = 15000;

const byte Rows = 4; //number of rows on the keypad i.e. 4
const byte Cols = 3; //number of columns on the keypad i,e, 3

char keymap[Rows][Cols] =
{
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

byte rPins[Rows] = {A1, A2, A3, A4}; //Rows 0 to 3
byte cPins[Cols] = {A5, A6, A7}; //Columns 0 to 2
Keypad kpd = Keypad(makeKeymap(keymap), rPins, cPins, Rows, Cols);

char deger[3] = {48, 48, 48};

bool izin = 1;
char keypressed;
int referans_deger = 9999, u = 0;
float Toplam1 = 0, Toplam2 = 0, cikis[1] = {0}, sonuc_1 = 0, sonuc = 0;
float NB = 0, NK = 0, S = 0, PK = 0, PB = 0;
float ortalar[5][5], kesinlik_dereceleri[5][5], cikis_v[5][5];
float de[5], d[5], e[5], sayac[30];
float x = 0;
float error = 0, error_1 = 0, hatanin_degisimi = 0;
int darbe = 0, darbe1 = 0, darbe2 = 0, i = 0, j = 0, k = 0;
float T = 999, F = 0, V = 0;
float ilk = 0, son = 0, Devir = 0, T_1 = 0;
float tetikleme = 0, D = 0;

int keypad_oku() {
  if (izin) {
    if (keypressed != '#' && keypressed != '*' && i < 3)
    { deger[i] = keypressed;
      i = i + 1;
      k = k + 1;
    }
  }
}
```

```

    }
    if (keypressed == '#') {
        referans_deger = (((deger[0] - 48) * 100) + ((deger[1] - 48) * 10) + ((deger[2] - 48) *
1));
        int carpan = pow(10, (3 - k));
        referans_deger = referans_deger / carpan;

        deger[0] = 48;
        deger[1] = 48;
        deger[2] = 48;
        i = 0;
        k = 0;
        izin = 0;
    }
}
if (keypressed == '*') {
    deger[0] = 48;
    deger[1] = 48;
    deger[2] = 48;
    referans_deger = 9999;
    i = 0;
    k = 0;
    izin = 1;
    error = 0;
    pwm_stop( 6);

    detachInterrupt(digitalPinToInterrupt(28));
    setup();
}
return referans_deger;
}

```

```

float *uyelik_hesaplama(float &x) {
    if (x >= -100 && x <= -50) {
        NB = -0.02 * x - 1;
        NK = 0.02 * x + 2;
        S = 0;
        PK = 0;
        PB = 0; }

    else if (x > -50 && x <= 0) {
        NB = 0;
        NK = -0.02 * x;
        S = 0.02 * x + 1;
        PK = 0;
        PB = 0; }
}

```

```

else if (x > 0 && x <= 50) {
    NB = 0;
    NK = 0;
    S = -0.02 * x + 1;
    PK = 0.02 * x;
    PB = 0; }

else if (x > 50 && x <= 100) {
    NB = 0;
    NK = 0;
    S = 0;
    PK = -0.02 * x + 2;
    PB = 0.02 * x - 1; }

else {
    NB = 0;
    NK = 0;
    S = 0;
    PK = 0;
    PB = 0; }

d[0] = NB;
d[1] = NK;
d[2] = S;
d[3] = PK;
d[4] = PB;

return d;
}

float *agirlik_merkezleri_hesaplama(float c[5], float f[5]) {
    Toplam1 = 0;
    Toplam2 = 0;
    for (i = 0; i < 5; i++) {
        for (j = 0; j < 5; j++) {
            cikis_v[i][j] = c[i] * f[j];
            if (cikis_v[i][j] == 0) {
                ortalar[i][j] = 0;
            }
            else {
                kesinlik_dereceleri[i][j] = cikis_v[i][j];
                if (i == 0 && j == 0 || i == 1 && j == 0 || i == 0 && j == 1) {
                    ortalar[i][j] = cikis_v[i][j] * (-127.5);
                }

                else if (i == 0 && j == 2 || i == 1 && j == 1 || i == 2 && j == 0) {
                    ortalar[i][j] = cikis_v[i][j] * (-85);
                }

                else if (i == 0 && j == 3 || i == 1 && j == 2 || i == 2 && j == 1 || i == 3 && j == 0) {
                    ortalar[i][j] = cikis_v[i][j] * (-42.5);
                }
            }
        }
    }
}

```

```

    }

    else if (i == 0 && j == 4 || i == 1 && j == 3 || i == 2 && j == 2 || i == 3 && j == 1 || i
== 4 && j == 0) {
        ortalar[i][j] = cikis_v[i][j] * 0;
    }

    else if (i == 1 && j == 4 || i == 2 && j == 3 || i == 3 && j == 2 || i == 4 && j == 1) {
        ortalar[i][j] = cikis_v[i][j] * 42.5;
    }
    else if (i == 2 && j == 4 || i == 3 && j == 3 || i == 4 && j == 2) {
        ortalar[i][j] = cikis_v[i][j] * 85;
    }
    else {
        ortalar[i][j] = cikis_v[i][j] * 127.5;
    }
}
}
}

for (i = 0; i < 5; i++) {
    for (j = 0; j < 5; j++) {
        Toplam1 = Toplam1 + ortalar[i][j];
        Toplam2 = Toplam2 + kesinlik_dereceleri[i][j];
    }
}
cikis[0] = round(Toplam1 / Toplam2);

return cikis;
}

```

```

void setup() {
    pinMode(12, OUTPUT);
    Serial.begin(250000);
    pwm_set_resolution(16);
    pwm_setup( 6, pwm_freq2, 1);
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.clear();
    lcd.print("Devir Sayisini");
    lcd.setCursor(0, 1);
    lcd.print("Giriniz(rpm):");

    attachInterrupt(digitalPinToInterrupt(28), encoder_kesme, CHANGE);

    while (referans_deger == 9999) {

```

```

    keypressed = kpd.getKey();
    if (keypressed) {
        keypad_oku();
    }
}
}
}

```

```

void loop() {

```

```

    keypressed = kpd.getKey();
    if (keypressed) {
        keypad_oku();
    }
    digitalWrite(12, HIGH);
    digitalWrite(12, LOW);

```

```

    error = (referans_deger - (Devir));
    if (Devir > 100) {
        Devir = referans_deger;
    }

```

```

    uyelik_hesaplama(error);

```

```

    for (i = 0; i < 5; i++) {
        e[i] = d[i];
    }

```

```

    hatanin_degisimi = (error - error_1);

```

```

    uyelik_hesaplama(hatanin_degisimi);

```

```

    for (i = 0; i < 5; i++) {
        de[i] = d[i];
    }
    lcd.setCursor(0, 0);
    lcd.print("Ref.Deg:");
    lcd.print(referans_deger);
    lcd.print("rpm");
    lcd.setCursor(0, 1);
    lcd.print("Olc.Deg:");
    lcd.print(T);
    lcd.print("rpm ");

```

```

    agirlik_merkezleri_hesaplama(e, de);

```

```

sonuc = (cikis[0] + sonuc_1);
if (sonuc >= 127.5) {
    sonuc_1 = 127.5;
    sonuc = 127.5;
}
else if (sonuc <= -127.5) {
    sonuc_1 = -127.5;
}
else {
    sonuc_1 = sonuc;
}

```

```

if (sonuc >= 127) {
    u = 127;
}
else if (sonuc <= 0) {
    u = 0;
}
else {
    u = sonuc;
}
u = map(u, 0, 127.5, 0, 32767);
pwm_write_duty( 6, u);
error_1 = error;

```

```

for (i = 0; i < 5; i++) {
    e[i] = 0;
    de[i] = 0;
    for (j = 0; j < 5; j++) {
        cikis_v[i][j] = 0;
        kesinlik_dereceleri[i][j] = 0;
        ortalar[i][j] = 0;
    }
}

```

```

}

```

```

void encoder_kesme() // kesme fonksiyonu oluşturulmaktadır
{
    detachInterrupt(digitalPinToInterrupt(28));
    darbe++;

    if (darbe == 1)
    {
        ilk = micros();
    }
    if (darbe == 2 && darbe - 1 == 1) {
        son = micros();
    }
}

```

```

T = ((son - ilk) * 2);

T = T / 1000;
darbe = 0;
ilk = 0;
son = 0;
}

if (T <= 0 || T >= 6) {
    F = 0;
}
else {
    F = (1 / T) * pow(10, 3);
}
V = (-1.45596683944356e-14 * pow(F, 4) + 1.46425540522710e-10 * pow(F, 3) -
4.28979355504581e-07 * pow(F, 2) + 0.00450312769045400 * F + 0.136337410134447);
if (V <= 0.36) {
    V = 0;
    Devir = 0;
}
else {
    Devir = ((V * 200) / 12) * 1.6;
}
attachInterrupt(digitalPinToInterrupt(28), encoder_kesme, CHANGE);
}

```