

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**GERÇEK ZAMANLI BULANIK  
KONTROLÖRLERİNİN GERÇEKLENMESİ**

**YÜKSEK LİSANS TEZİ  
Müh. Özgür Aydın TEKİN**

**Anabilim Dalı : ELEKTRİK MÜHENDİSLİĞİ**

**Programı : KONTROL VE OTOMASYON  
MÜHENDİSLİĞİ**

**OCAK 2007**

**GERÇEK ZAMANLI BULANIK  
KONTROLÖRLERİNİN GERÇEKLENMESİ**

**YÜKSEK LİSANS TEZİ**  
**Müh. Özgür Aydın TEKİN**  
**(504041116)**

**Tezin Enstitüye Verildiği Tarih : 22 Aralık 2006**  
**Tezin Savunulduğu Tarih : 30 Ocak 2007**

**Tez Danışmanı : Prof. Dr. İbrahim EKSİN**  
**Diğer Jüri Üyeleri : Prof. Dr. Müjde GÜZELKAYA**  
**Yrd. Doç. Dr. Osman Kaan EROL**

## **ÖNSÖZ**

Başta danışmanım Prof. Dr. İbrahim Eksin olmak üzere, Prof Dr. Müjde Güzelkaya, Araş. Gör. Engin Yeşil, Araş. Gör. Kürşat Yalçın ve biricik aileme her türlü destek ve yardımlarından dolayı sonsuz teşekkürlerimi sunarım.

Ayrıca, yüksek lisans öğrenimimde Yurt İçi Yüksek Lisans Burs Programı çerçevesinde bana burs desteği veren TÜBİTAK Bilim İnsanı Destekleme Daire Başkanlığı'na ve İ.T.Ü. Fen Bilimleri Enstitüsü'ne proje kapsamında yüksek lisans tez çalışması desteği için teşekkür ederim.

**Aralık 2006**

**Özgür Aydın Tekin**

## İÇİNDEKİLER

<b>KISALTMALAR</b>	<b>iv</b>
<b>TABLO LİSTESİ</b>	<b>v</b>
<b>ŞEKİL LİSTESİ</b>	<b>vi</b>
<b>SEMBOL LİSTESİ</b>	<b>vii</b>
<b>ÖZET</b>	<b>viii</b>
<b>SUMMARY</b>	<b>x</b>
<b>1. GİRİŞ</b>	<b>1</b>
<b>2. PIC MİKROKONTROLÖR GELİŞTİRME ORTAMI</b>	<b>7</b>
2.1. PIC Geliştirme Kartı	8
2.2. Hata Ayıklayıcı ve Programlayıcı Kartı	10
2.3. Mikrokontrolör Yazılımı Geliştirme Ortamı	11
2.4. PIC18F452 Mikrokontrolörünün Özellikleri	11
2.5. Gerilim Kuvvetlendirici Kartı	14
<b>3. KONTROLÖR YAZILIMININ GERÇEKLENMESİ</b>	<b>16</b>
3.1. Bulanık Çıkarım Yapısının Oluşturulması	18
3.2. Bulanık Çıkarım Mekanizmasının PICC Dilinde Gerçeklenmesi	22
3.3. Ana Kontrol Yazılımı (main( ))	23
<b>4. ISIL SİSTEM ÜZERİNDE DENEYSEL GERÇEKLEMELER</b>	<b>26</b>
4.1. PT326 Modelinin Elde Edilmesi	26
4.2. GATool (MATLAB Genetic Algorithm Toolbox) ile Ölçkleme Katsayılarının Bulunması	29
4.3. Gerçek Zamanlı Çalışma ile Simulink Model Sonuçlarının Karşılaştırılması	30
4.4. MPC555 Geliştirme Kartı ile Yapılan Çalışmalar	31
<b>5. SONUÇ</b>	<b>39</b>
<b>KAYNAKLAR</b>	<b>41</b>
<b>ÖZGEÇMİŞ</b>	<b>43</b>

## KISALTMALAR

<b>ANSI</b>	: American National Standards Institute
<b>IDE</b>	: Integrated Development Environment
<b>ICD</b>	: In-Circuit Debugger
<b>USART</b>	: Universal Synchronous Asynchronous Receiver Transmitter
<b>PC</b>	: Personal Computer
<b>LCD</b>	: Liquid Crystal Display
<b>ADC</b>	: Analog to Digital Converter
<b>PWM</b>	: Pulse Width Modulation
<b>PID</b>	: Proportional Integral Derivative
<b>DSP</b>	: Digital Signal Processor
<b>RAM</b>	: Random Access Memory
<b>EEPROM</b>	: Electrically Erasable Programmable Read Only Memory
<b>ICSP</b>	: In Circuit Serial Programming
<b>IMP</b>	: Implementation
<b>AGG</b>	: Aggregation
<b>GATOOOL</b>	: MATLAB Genetik Algoritma Aracı Çalıştırma Komutu
<b>SPI</b>	: Serial Peripheral Interface
<b>SCI</b>	: Serial Communication Interface
<b>CAN</b>	: Controller Area Network

## TABLO LİSTESİ

	<u>Sayfa No</u>
<b>Tablo 3.1.</b> En Genel Bulanık Kontrolör Kural Tabanı .....	18

## ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1: Isıl Sistem Şeması .....	4
Şekil 1.2: PT326 Proses Kontrol Sistemi Blok Diyagramı.....	4
Şekil 1.3: Termistör Basamak Sıcaklık Yanıtı .....	5
Şekil 1.4: Üç Farklı Pozisyon için Açık Çevrim Birim Basamak Yanıtları .....	5
Şekil 1.5: PT326 Isıl Sistem Basitleştirilmiş Gösterim .....	6
Şekil 2.1: PIC Mikrokontrolör ile Gerçek Zamanlı Bulanık Kontrolör Geliştirme Ortamı .....	7
Şekil 2.2: Gerilim Kuvvetlendirici Kartı Devre Şeması.....	15
Şekil 2.3: Gerilim Kuvvetlendirici Kartı Baskı Devre Çizimi .....	15
Şekil 3.1: PI Tipi Bulanık Kontrolör ile Temel Kontrol Blokları .....	16
Şekil 3.2: $e$ ve $\Delta e$ için Üyelik Fonksiyonları .....	17
Şekil 3.3: $\Delta u$ için Üyelik Fonksiyonları .....	18
Şekil 3.4: Ana Kontrol Yazılımı Akış Diyagramı .....	25
Şekil 4.1: PT326 Isıl Sisteminin Basamak Yanıtı .....	27
Şekil 4.2: Ziegler-Nichols Yöntemi ile PT326 Modelinin Elde Edilmesi.....	28
Şekil 4.3: Basamak Yanıtı Simulink Modeli .....	29
Şekil 4.4: Basamak Yanıtında Gerçek Çıkış ile Model Çıkışının Karşılaştırılması (Referans: 425, 40°C) .....	29
Şekil 4.5: Gatoool ile Minimum IAE Elde Etmek için Kurulan Simulink Modeli .....	30
Şekil 4.6: Gerçek Sistemin ve Simulink Modelin Kapalı Çevrim Basamak Yanıtları (Referans: 425, 40°C) .....	31
Şekil 4.7: phyCORE-MPC555 Rapid Development Kit .....	32
Şekil 4.8: Metrowerks CodeWarrior IDE Ayarları .....	33
Şekil 4.9: MPC555 Hedef Ayarları .....	33
Şekil 4.10: MPC555 Kartı ile Çalışma Ortamı .....	34
Şekil 4.11: Bulanık Kontrol MPC555 Simulink Modeli .....	35
Şekil 4.12: Model Kodunun RAM ya da FLASH Bellek için Üretilmesi Seçimi.....	36
Şekil 4.13: Model Kodunun Karta Yüklenmesi .....	37
Şekil 4.14: MPC555 Kartı Üzerindeki Varsayılan Atlamalar ve Bacak Numaraları .....	38
Şekil 4.15: MPC555 Kartı ile Gerçek Zamanlı Bulanık Kontrol Sistemi için Basamak Yanıtı (Referans: 425, 40°C).....	38

## SEMBOL LİSTESİ

<b>e</b>	: Hata
<b><math>\Delta e</math></b>	: Hatanın deęiřimi
<b><math>\Delta u</math></b>	: Kontrol iřaretinin deęiřimi
<b>ke</b>	: Hata ölçekleme katsayısı
<b>kde</b>	: Hatanın deęiřimi ölçekleme katsayısı
<b>ko</b>	: Kontrol iřaretinin deęiřimi ölçekleme katsayısı
<b>K</b>	: Sistem kazancı
<b>L</b>	: Sistem ölü zamanı
<b>T</b>	: Sistem zaman sabiti



## GERÇEK ZAMANLI BULANIK KONTROLÖRLERİNİN GERÇEKLENMESİ

### ÖZET

Bulanık mantık, özellikle doğrusal olmayan, karmaşık sistemlerin modellenmesi ve kontrolündeki vazgeçilmez önemi nedeni ile endüstriyel alanlarda yaygınlaşarak kullanılmaktadır. Endüstride bu kadar büyük öneme sahip bulanık kontrol yöntemlerinin gerçek zamanlı olarak test edilmesi ve performanslarının gerçek zamanlı olarak gözlenmesi ile elde edilen veriler ışığında kontrolör tasarlanabilmesi kritik öneme sahiptir.

Kontrol dünyasında sıkça kullanılan MATLAB/Simulink yazılımı ile gerçek sistemlerin matematiksel modellerini kontrol etmek üzere yapılan pek çok çalışma vardır. Bu şekilde teorik olarak yüksek başarılı kontrolörler elde etmek mümkündür, fakat bu çalışmaların gerçek sistemler üzerindeki performansları konusunda söz söyleyebilmek ancak tasarlanan kontrolörlerin üretilip gerçek sistem üzerinde denenmesi ile mümkün olabilir ki istenen şekilde çalışmaması halinde baştan tüm tasarım ve üretim süreçlerinin tekrarlanması gerekir, bu da hem zaman hem de para kaybıdır. Bu çalışma ile bu sorun çözülmek istenmektedir. PC üzerinde gerçek dünyadan kopuk bir şekilde çalışan MATLAB/Simulink ile tasarlanan kontrolörlerin dış dünya ile bağlantısını sağlayacak bir mikrokontrolör kartı ile sistemin modeli üzerinde çalışmak zorunda kalınmadan direk olarak bu mikrokontrolör üzerinde kontrol algoritması koşturularak gerçek sistemi kontrol edecek kontrolörler tasarlanacaktır. Daha önceden benzetim ortamında teorik olarak yüksek başarılı çalışacak şekilde tasarlanmış olan kontrolörlerin de gerçek ortamda, gerçek zamanlı çalışma başarımları görülecektir.

Bu çalışma ile teorik olarak elde edilmiş ve iyi benzetim sonuçları veren kontrolörlerin gerçek zamanlı çalışma performanslarının gözlenebileceği ve gerçeğe daha yakın kontrol sistemlerinin tasarlanabileceği bir çalışma platformu oluşturulmaya çalışılmıştır. Bunun için fiyatı uygun, bulunması kolay, endüstride yaygın olarak kullanılan 8-bit PIC mikrokontrolör kullanılmıştır. Kişisel bilgisayar (PC) üzerinde çalışan MATLAB 'Fuzzy Logic Toolbox' arayüzü kullanılarak oluşturulmuş ve '.fis' uzantısı ile diske kaydedilmiş bir bulanık çıkarım yapısı dosyası C dilinde yazılan bir program ile okunarak gerekli parametreler çekilmiş ve bu parametreler ile PIC mikrokontrolörler için kullanılan PICC dilindeki bulanık çıkarım yapısı yeniden oluşturulmuştur. Bu yapıyı kullanarak girişleri bulandırıp, kural tabanını uygulayıp bulanık çıkışı oluşturacak ve bu bulanık çıkışı durulayıp keskin kontrol çıkışını oluşturacak olan bulanık çıkarım fonksiyonları da MATLAB yazılımı ile verilen C kodlarından yararlanılarak PICC dilinde yazılmıştır.

Dış dünyadan gelen işaretleri ADC ile elde eden, bulanık kontrolör için gerekli giriş parametrelerini oluşturan, bulanık kontrol çıkışını hesaplayıp uygun kontrol işaretini PWM ile dış dünyaya uygulayan, bu arada bazı bilgileri LCD ekranda gösteren ve RS-232 ile PC'ye aktaran ana program da PICC dilinde yazıldıktan sonra bütün programlar MPLAB IDE kullanılarak bir proje altında derlenmiştir. Böylece mikrokontrolör için gerekli makine dilindeki '.hex' dosyası elde edilmiş ve bir ICSP PIC programlayıcı üzerinden çeşitli çevre birimleri de üzerinde bulunduran bir mikrokontrolör geliştirme kartına yüklenmiştir. Bu mikrokontrolör kartı ile gerçek bir ölü zamanlı ısıtım sistemi (Process Trainer PT326 deney seti) gerçek zamanlı olarak kontrol edilmiştir. Çalışma esnasında gerekli işaretler gerçek zamanlı olarak PC'ye aktarılmış, kaydedilmiş ve gözlenebilmiştir.

Böylece, matematiksel modeller üzerinden kuramsal olarak elde edilen kontrolörlerin gerçek zamanlı olarak gerçek sistem üzerindeki başarımlarının incelenebileceği ve belki de ileride gerçek sistemlere özel kontrolör tasarlanmasında çok değerli olabileceği düşünülen bir gerçek zamanlı kontrol sistemi düzeneği elde edilmiştir.

Ayrıca, bu tez çalışması için İ.T.Ü. Fen Bilimleri Enstitüsü tarafından desteklenen "Gerçek Zamanlı Bulanık Kontrol Uygulaması" başlıklı proje kapsamında alınan "phyCORE-MPC555 Rapid Development Kit" mikrokontrolör kartı ile de benzer uygulamalar gerçek zamanda gerçekleştirilmiştir.

## **IMPLEMENTATION OF REAL-TIME FUZZY CONTROLLERS**

### **SUMMARY**

Fuzzy Logic is widely used in industrial applications due to its inevitable importance in modeling and control of complicated, nonlinear systems. Design of controllers according to the information obtained by real time testing and observation of real-time performances of these fuzzy control techniques which have great importance in industrial applications is especially important and necessary.

In the world of control, controllers are mostly designed by MATLAB/Simulink software platform using the mathematical models of the real systems to be controlled. Real systems, especially that are nonlinear and complex, are very hard to be modeled perfectly. That's why, although one can design controllers of good performance theoretically, real performances of the controllers can only be seen after the implementation on real world. And, the controllers that are designed by using these approximate models do not always perform qualitative control on real plants as they are expected theoretically. In this case, all the design and implementation process should be done again from the beginning, causing both money and time loss. By this thesis work, this problem is tried to be solved. By the implementation of fuzzy controllers, which are designed by MATLAB/Simulink and simulated on a PC apart from the real world, on a microcontroller board maintaining the connection with the real systems, one will be able to design real-time fuzzy controllers directly for a real plant without having to use the models of them, and the real-time performances of the fuzzy controllers will be analyzed.

In this project, a real time fuzzy control application platform is realized to be able to analyze the real-time performances of various fuzzy control algorithms in the literature which are theoretically of high performance, and to be able to design controllers with better real-time performance. An 8-bit PIC microcontroller which is frequently used in industry and so, easy to find and afford is used. Fuzzy controllers are designed by MATLAB 'Fuzzy Logic Toolbox' on a PC and saved to disk as a '.fis' file. A C program is written to open this '.fis' file, parse the necessary parameters of the fuzzy inference structure stored in this file, and construct the fuzzy inference structure in PICC language for PIC microcontrollers.

Also, the fuzzy inference mechanism functions used for fuzzifying the inputs, applying the rule base to produce the fuzzy output, finally defuzzifying this fuzzy output to produce the crisp control output are written in PICC language by the help of MATLAB 'Stand Alone C Code Fuzzy Inference Engine' tutorial.

Main control program reading the real world signals by ADC unit, producing the inputs for fuzzy controller, getting the fuzzy controllers output and applying the

control signal to the real world by PWM unit, and meanwhile, displaying some data on LCD and also transmitting these data to the PC via RS-232 serial communication port is written in PICC language. Then, all these C codes are compiled and built under MPLAB IDE software environment. The built machine code is embedded into the PIC microcontroller via ICSP PIC programmer. By this PIC microcontroller development board, a real heat process with dead time PT326 is controlled in real-time, and on the run, some measured data is transferred to the workspace of MATLAB running on PC.

Thus, a real-time control system platform where real-time performances of the controllers designed theoretically using the models can be examined is obtained leading valuable future works on design of special real-time controllers for real plants.

Finally, similar real-time applications are achieved using the ‘phyCORE-MPC555 Rapid Development Kit’ microcontroller board supported by the Institute of Science and Technology of Istanbul Technical University.

## 1. GİRİŞ

Bu çalışmanın amacı, genel olarak bir kontrol sisteminin gerçek zamanlı olarak gerçekleştirilebilmesi araştırılarak MATLAB ortamında hazırlanan ve Simulink ile gerçek sistemlerin modeli kullanılarak rahatlıkla kurulup analiz edilebilen bulanık kontrol sistemlerinin gerçek zamanlı olarak gerçek sistem üzerinde çalıştırılması ve analiz edilmesidir. Gerçek sistem olarak ölü zamanlı ısı sistem olan Feedback firmasının Process Trainer PT326 deney seti kullanılmıştır. Şekil 2.1’de 4 numara ile gösterilmiştir. Bulanık kontrolör, MICROCHIP firmasının ürettiği 8-bit mikrokontrolör olan PIC18F452 ile gerçekleştirilmiştir. ‘Visual Studio 6.0’ ortamında yazılan bir C programı ile MATLAB ‘Fuzzy Logic Toolbox’ ile tasarlanıp ‘.fis’ uzantısı ile diske kaydedilen bulanık kontrolör yapısı ‘.fis’ dosyasından okunup ayrıştırılarak, PIC mikrokontrolör yazılımının derlenmesi için kullanılan ANSİ C bir dil olan PICC diline çevrilmiştir. PICC dilinde hazırlanan kontrol yazılımı MPLAB IDE altında derlenerek MPLAB ICD2 hata ayıklayıcı ve programlayıcı işlevi gören Infogate firmasının ürünü icdPIC kullanılarak mikrokontrolör programlanmıştır. Hata ayıklayıcı ve programlayıcı kartı icdPIC Şekil 2.1’de 1 numara ile gösterilmiştir. Böylece MATLAB ortamında tasarlanan ve benzetim ortamında deney ve analizleri yapılan bulanık kontrolör gerçek zamanlı olarak PIC mikrokontrolör üzerinde çalıştırılmıştır. Ayrıca, mikrokontrolörün USART biriminin çıkışına bağlı RS-232 seri port bağlantısı ile sistem çıkışı, kontrol işareti gibi bilgiler PC üzerinde çalışan MATLAB ortamına aktarılabilmiş ve yine mikrokontrolöre çevre birim olarak bağlanan LCD üzerinde de gösterilebilmiştir. Böylece, mikrokontrolör ile elde edilen gerçek ortama ait bilgiler PC ortamında saklanabilmiş ve analiz edilebilmiştir. Dış dünyadan analog verilerin sayısal olarak okunması ve mikrokontrolör ile sayısal olarak üretilen kontrol işaretinin dış dünyaya analog olarak aktarılması PIC mikrokontrolörün 10-bit çözünürlüklü ADC ve PWM birimleri ile sağlanmıştır. Bu birimler 0-5V aralığında çalıştığından 0-10V aralığında çalışan PT326 ısı sistem ile bağlantılar işlemsel kuvvetlendirici ile tasarlanan 0-5V↔0-10V gerilim kuvvetlendirici kartı üzerinden yapılmıştır. Bu kart Şekil 2.1’de 3 numara ile gösterilmiştir.

Giriş bölümünün devamında bulanık kontrolör, gerçek zamanlı uygulama ve PT326 ölü zamanlı ısı sistem hakkında yapılan literatür araştırması özeti verilmiştir. İkinci

bölümde, kullanılan 8-bit PIC mikrokontrolörün üzerinde çalıştığı geliştirme kartı, mikrokontrolörün programlanması için gerekli olan icdPIC hata ayıklayıcı ve programlayıcı kart, mikrokontrolör için yazılım geliştirme ortamı olan MPLAB IDE ortamı ve PIC18F452 mikrokontrolörün özellikleri anlatılmıştır. Üçüncü bölümde, bulanık kontrolör yapısının MATLAB ortamında oluşturulması, PICC diline dönüştürülmesi ve PICC ile yazılan diğer C programlarıyla birlikte ana kontrol yazılımının oluşturulması anlatılmıştır. Dördüncü bölümde ise öncelikle gerçek sistem olarak kullanılan PT326 ısıtıcı sistem deney setinin modelinin PIC mikrokontrolör kullanılarak elde edilmesi ve bu model ile kontrol sisteminin Simulink modelinin oluşturulması anlatılmıştır. Daha sonra düzgün bir sistem yanıtı elde edilmesi için MATLAB genetik algoritma aracı (gatool) ile aranıp bulunan bulanık kontrolör giriş, çıkış ölçekleme katsayılarının mikrokontrolöre girilerek gerçek sistemden elde edilen yanıtlar ile benzetiminden elde edilen sonuçların karşılaştırılması anlatılmıştır. İ.T.Ü. Fen Bilimleri Enstitüsü projesi kapsamında alınan 32-bit MOTOROLA işlemcili Phytex MPC555 kartı ile yapılan benzer çalışmalar dördüncü bölümün sonunda verilmiştir. Sonuç bölümünde de genel bir özet ile birlikte konu ile ilgili gelecekte yapılabilecek çalışmalar ve yorumlara yer verilmiştir.

### **Bulanık Kontrol ve Gerçek Zamanlı Çalışma Üzerine Literatür Özeti:**

Bulanık küme kuramı ilk olarak 1965 yılında Zadeh tarafından bulunmuş ve ilk bulanık mantık kontrol algoritması Mamdani tarafından 1974 yılında buhar makinesi üzerinde uygulanmıştır. Takip eden yıllarda ise artan ivmeyle birçok endüstriyel alanda başarıyla uygulanmıştır. Özellikle doğrusal olmayan, karmaşık yapıları sistemlerin modellenmesi ve kontrolünde vazgeçilmez bir öneme sahiptir [1].

Herhangi bir endüstriyel kontrol tasarım sürecinde, kontrolörlerin gerçek zamanlı uygulanması uzun zaman alan bir işittir. Artan ihtiyaçlara bağlı olarak hızlı bir şekilde kontrol prototiplerinin elde edilmesi ile proje tamamlanma süresinin azaltılmasına olan gereksinimin artmasından dolayı kontrolör tasarımının gerçek zamanlı olarak yapılmasına olan gereksinim artmış ve gerçek zamanlı kontrolörlerin tasarımı daha çok kabul görmeye başlamıştır [1].

Son üretim için kullanılan yazılım ve donanımlardan bağımsız bir şekilde sadece blok diyagramları kullanarak tasarım yapmak tasarım maliyetini düşürmektedir. Ek olarak, bu yöntem ile tasarımcı, sistem modelini ardışıl olarak iyileştirebilir ve kontrolör devredeyken (on the fly) kontrolör parametrelerini ayar edebilir. Bu yöntemin öğretilmesi birçok öncü akademik kurumlarca da kabul görmektedir [1].

Kontrol dünyasında yıllardır kullanılan klasik PID kontrolörlerin bulanık olarak gerçekleştirilmesi üzerine, içlerinde İ.T.Ü. Elektrik-Elektronik fakültesi öğretim üyelerinin yayınları [2] da olmak üzere literatürde birçok yayın vardır ve bu bulanık PID kontrolörlerin gerçek zamanlı olarak uygulanmasına yönelik çalışmalar artmıştır [3].

Bulanık mantık, yalnız PID tipi kontrolörlerde değil kayma kipli kontrolörlerde de kullanılmaktadır. Bulanık kayma kipli kontrolörlerde kayan yüzey eğiminin öz uyarlandığı çalışmalar yapılmıştır.

Yine fakülte öğretim üyelerimizce yapılan bir çalışmada, kural tabanının oluşturulmasında yeni bir yöntem kullanılarak bir bulanık kontrolör tasarlanmıştır. Bu yapıda iki bulanık kural tabanı ve bunlar arasında geçişi sağlayan bir lojik anahtar kullanılmıştır. Geliştirilen bulanık kontrolör, bilinen Mac Vicar-Whelan kontrolör ile kıyaslanmış ve çeşitli benzetim sonuçları verilerek daha iyi performans sağladığı gösterilmiştir. Tasarlanan bu bulanık kontrolörün gerçek zamanlı çalışma performansının da benzetimlerdeki gibi iyi sonuçlar verip vermeyeceğinin, gelecek çalışmalar olarak burada kurulmuş olan gerçek zamanlı çalışma ortamında görülebilecektir.

Doğrusal olmayan sistemlerin modellenmesi ve kontrolü konusu literatürde geniş yer tutmaktadır. Bu tip sistemler için bulanık mantık yöntemi kullanmak etkili sonuçlar vermektedir. Ayrıca, MATLAB Simulink ile gerçek zamanlı akıllı kontrolünün yapılması üzerine literatürde yayın bulmak mümkündür [4]. MATLAB kullanılarak tasarlanan bulanık kontrolörlerin günümüzde yaygın olarak kullanılmaya başlanan sayısal işaret işleyiciler (DSP) ile gerçek zamanlı olarak uygulaması da yapılmaktadır [5, 6].

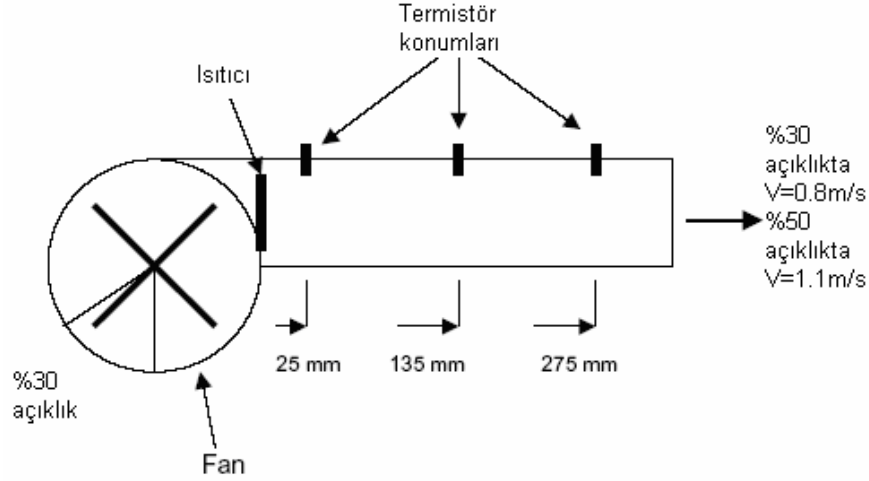
Gerçek zamanlı bulanık kontrol uygulamalarının yaygınlaşmasıyla, MOTOROLA hız gereksinimini karşılamak amacıyla bulanık işlemleri donanımsal olarak yapabilen mikrokontrolörler üretmiştir ve bu mikrokontrolörlerin kullanıldığı kartlar ile gerçek zamanlı bulanık kontrolörlerin tasarlandığı yayınlar literatürde yerlerini almaktadır [7-9].

### **PT326 Ölü Zamanlı Isıl Sistem Üzerine Literatür Araştırması:**

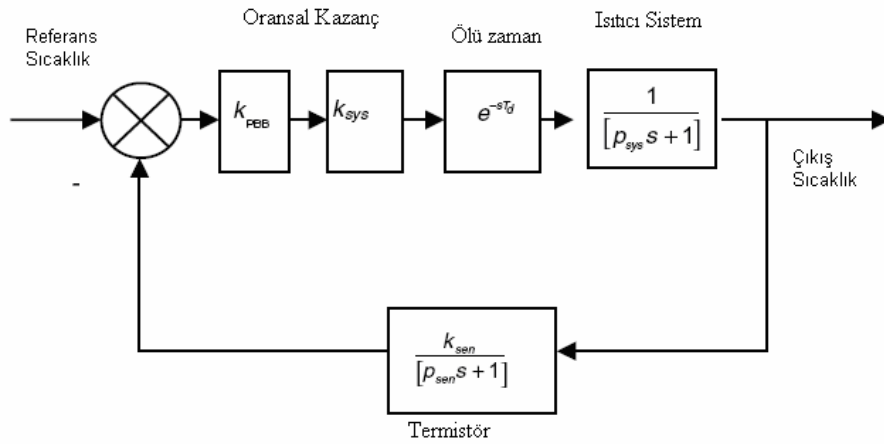
[10]'daki makalede gerçek sistem olarak kullanılan PT326 ısı kontrol sistemi ayrıntılı bir şekilde anlatılmıştır. Sistem, basit bir fan ile ısıtılmış havanın plastik naylon tüpten akmasını sağlar. Hava, fanın hemen çıkışında, tüpün hemen girişindeki ısıtıcı ile ısıtılmaktadır (Şekil 1.1).

Cihaz, oransal kontrol, aç-kapa kontrol veya kontrol işaretinin dışarıdan girilmesine olanak sağlamaktadır. Cihazın temel özelliği, sıcaklık ölçme elemanı termistörün konumunun değiştirilmesi ile ölü zamanın doğal olarak ayarlanabilmesidir. Sistemin blok diyagramı termistörün ve ısıtılmış havanın gecikmeleri tek bir gecikme bloğu olarak Şekil 1.2’de gösterilmiştir [10].

Termistör elemanı lineer değil, zaman sabiti 0.033s olan ölü zamanlı birinci derece sistem gibidir (Şekil 1.3). 30° kapak açıklığı için hava akış hızının 0.8m/s olmasından ısıtıcı sistemin ölü zamanı, yani ısıtılmış havanın termistöre ulaşma zamanı, birinci pozisyon için 0.03s, üçüncü pozisyon için ise 0.34s’dir [10]. Kapak açıklığı 30° iken termistörün üç farklı pozisyonu için sistemin açık çevrim birim basamak yanıtı Şekil 1.4’te verilmiştir.

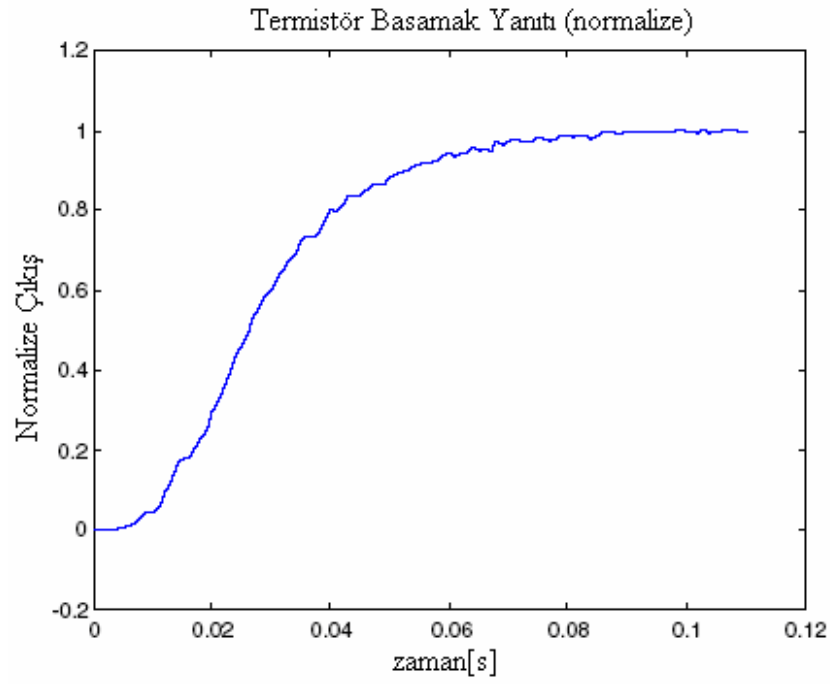


Şekil 1.1: Isıl Sistem Şeması

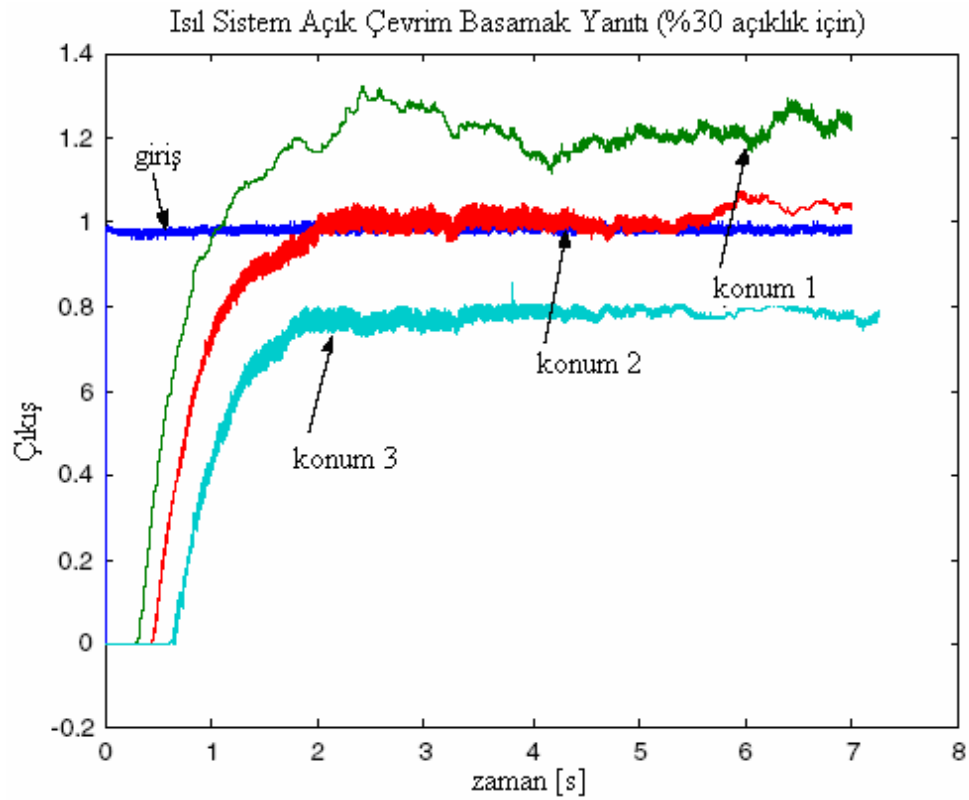


Şekil 1.2: PT326 Proses Kontrol Sistemi Blok Diyagramı





**Şekil 1.3:** Termistör Basamak Sıcaklık Yanıtı

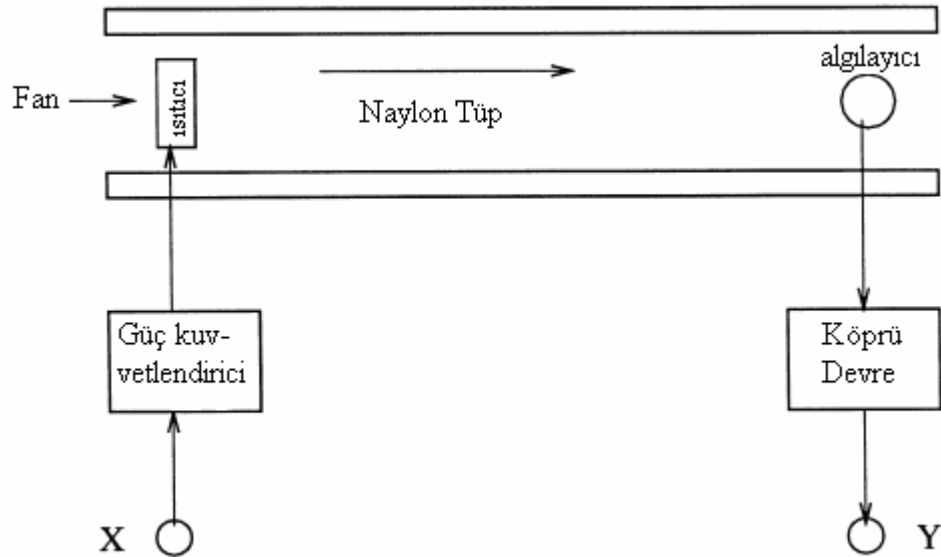


**Şekil 1.4:** Üç Farklı Pozisyon için Açık Çevrim Birim Basamak Yanıtları

[11]'deki makalede üç farklı kontrolörün gerçek zamanlı olarak performanslarının test edilmesinde gerçek sistem olarak kullanılan PT326 ısı kontrol deney setinin

çalışması aşağıda özetlenmeye çalışılmıştır. Isı transfer seti büyük bir kurulu düzeneğin temel özelliklerine sahiptir. Bu set ile hava merkezkaç üfleyici vasıtasıyla atmosferden çekilerek bir ısıtıcı ızgaradan ve devamındaki tüpten geçirilerek tekrar atmosfere verilir. Amaç, tüpten çıkan havanın sıcaklığını kontrol etmektir. Süreç, tüpten akan havayı istenen sıcaklık düzeyine ısıtmaktan oluşur. Kontrol işareti, üfleyicinin hemen bitişiğindeki ısıtıcıya verilen elektriksel gücün miktarını belirler. Şekil 1.5'te sistemin basitleştirilmiş çizimi verilmiştir. Sıcaklık bir probun ucuna tutturulmuş bir termistör ile ölçülmektedir. Termistör, tüp üzerinde uzaklıkları 28mm, 140mm, 280mm olan üç delikten tüpe sokulabilir. Termistör probu doğru akım köprüsünün bir kolunu oluşturur. Köprü çıkış gerilimi bir doğru akım kuvvetlendiricisinden geçirilerek 30-70°C hava sıcaklığı için 0-10V gerilim aralığına düşürülür ve ön paneldeki bir analog göstergede gösterilir [11].

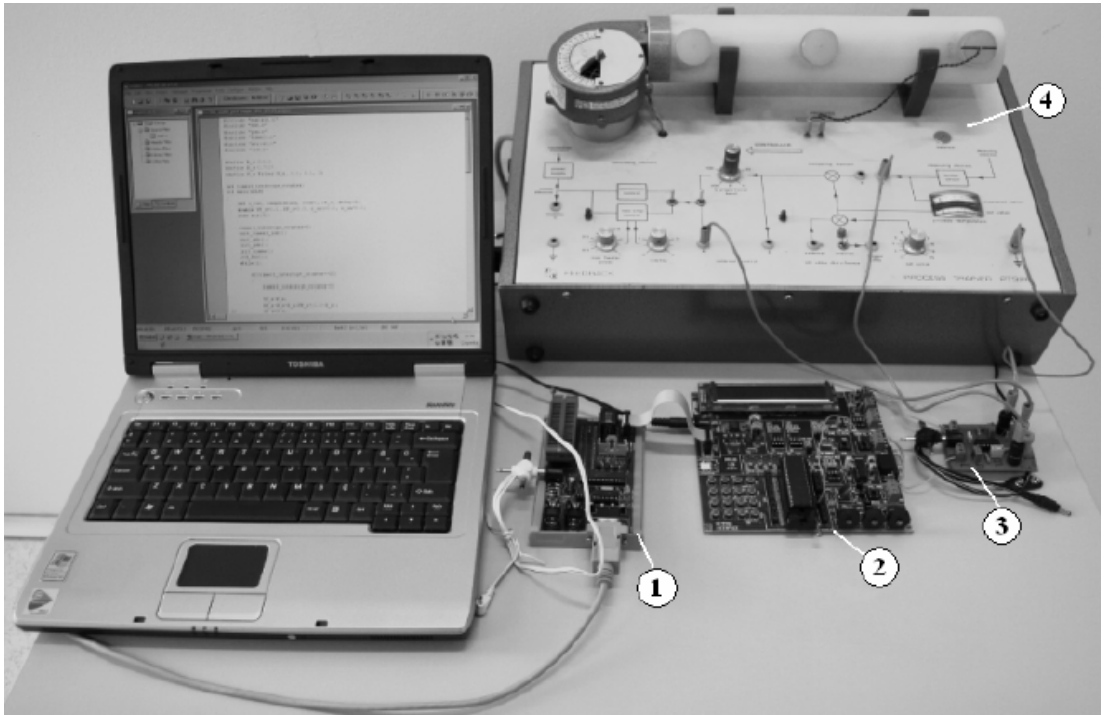
Hava, ızgara şeklindeki ısıtıcıdan geçerken ısı havaya aktarılır. Aktarım oranı, ısıtıcı ızgaranın sıcaklığına ve havanın akış hızına bağlıdır. Havanın, ızgaradan geçip termistör probuna ulaşması için geçen süre ölü zamandır. Yani, ölü zaman, havanın hızıyla ters, ısıtıcı ile termistör arasındaki mesafe ile doğru orantılıdır. Havanın hızı da üfleyicinin üzerindeki kapak açıklığı ile doğru orantılıdır. Bu ısı transfer süreci birinci dereceden sürekli bir transfer fonksiyonu olarak şu şekilde modellenilebilir.  $\frac{f}{(1+\beta s)}e^{-\tau_d s}$ .  $f$  sistem kazancı ve  $\beta$  zaman sabitidir. 20° kapak açıklığı için termistör 280mm uzakta iken ölü zaman 0.5s olarak ölçülmüştür [11].



**Şekil 1.5:** PT326 Isıl Sistem Basitleştirilmiş Gösterim

## 2. PIC MİKROKONTROLÖR GELİŞTİRME ORTAMI

Kullanılan 8-bit PIC mikrokontrolörün üzerinde çalıştığı geliştirme kartı, mikrokontrolörün programlanması için gerekli olan icdPIC hata ayıklayıcı ve programlayıcı kartı, mikrokontrolör için yazılım geliştirme ortamı olan MPLAB IDE ortamı ve PIC18F452 mikrokontrolörün özellikleri bu bölümde anlatılmıştır. PIC mikrokontrolör ile gerçek zamanlı bulanık kontrolör geliştirme ortamı Şekil 2.1'de gösterilmiştir. Şekilde 1 numara ile gösterilen icdPIC hata ayıklayıcı ve programlayıcı kartıdır. 2 numara ile gösterilen PicLAB PIC mikrokontrolör geliştirme kartıdır. 3 numara ile gösterilen 0-5V↔0-10V gerilim kuvvetlendirici kartıdır. PT326 ölü zamanlı ısıl deney seti de 4 numara ile gösterilmiştir.



**Şekil 2.1:** PIC Mikrokontrolör ile Gerçek Zamanlı Bulanık Kontrolör Geliştirme Ortamı

## 2.1 PIC Geliştirme Kartı

PIC geliştirme kartı olarak Era Bilgi Sistemleri firmasının ([www.infogate.org](http://www.infogate.org)) ürünü PicLAB endüstriyel deney seti kullanılmıştır. Ürünün web sayfasından alınan teknik özellikleri aşağıda verilmiştir.

- **Programlayıcı gerektirmez:** PICLab’la birlikte verilen PLab-xx (ya da PICprog) mikrodenetleyicisine - içine özel yazılım yüklenmiş PIC16F87x veya PIC18Fxx2 - nesne koduna çevrilmiş kaynak kodu, PC’nin hyperterminal programı kullanılarak aktarılmaktadır. Bu sayede haricen programlayıcı cihaz kullanmaya gerek kalmadan işlemciye defalarca program yüklenebilmektedir.
- **PIC’i devreden sökmeden programlama:** PLab-xx entegresi yerine başka bir PIC kullanmak isterseniz, PIC’i devreden sökmeden de programlayabileceksiniz. Böylece PIC’in devreye takılması-sökülmesi sırasında bacaklarının kırılması gibi tehlikelerle de karşılaşmayacaksınız. Buna yönelik olarak PICLab kartındaki “devre üzerinde programlama” soketi (ICSP) kullanılmaktadır. Bu özellikten yararlanabilmek için “devre üzerinde PIC programlama” yeteneğine sahip –örn. ProtoPIC Pro gibi– bir programlayıcıya ihtiyacınız vardır.
- **MPLAB ICD (in-circuit debugger) bağlantısı:** Microchip’in PIC16F87x ve PIC18Fxxx ailesi için geliştirdiği ICD ve ICD2 bağlantısına uygun soket devresi ve bağlantı soketi sağlanmaktadır.
- **Güç kaynağı:** PICLab ana kartı üzerinde tam dalga doğrultuculu +5V güç kaynağı vardır.
- **RESET devresi:** PIC’i gerektiğinde elle reset edebilmek için buton kumandalı RESET devresi vardır.
- **PICLab kartında bulunan uygulama devreleri:** PICLab kartındaki tüm uygulama devreleri en uygun PIC portlarına bakır yollar aracılığıyla bağlanmıştır.
  - **Mikrodenetleyici port konfigürasyonu:** PICmicro’nun portlarıyla kart üzerindeki uygulama devre bloklarının bağlantısı atlamalar yardımıyla istenildiği gibi değiştirilebilmekte, böylece kullanıcının her portu dilediği gibi yapılandırarak istediği gibi deney yapmasına olanak sağlanmaktadır.

- **A/D çevrim (yonga üzeri):** Her biri 10 bit çözünürlükte 8 A/D çevirici kanal mevcuttur. Bunlardan üçüne bağlı potansiyometreler aracılığıyla temel A/D çevrim uygulaması yapılmaktadır.
- **LCD gösterge:** En az 2x16 kapasiteli karakter LCD gösterge bağlantısı vardır.
- **Grafik LCD gösterge:** KS0108 uyumlu 128x64 piksel grafik LCD gösterge bağlantısı vardır.
- **LED gösterge:** Çıkış uygulamaları için 8 adet LED vardır.
- **Tuş takımı:** Matris biçiminde bağlanmış 16 adet buton vardır.
- **RS-232 arayüzü:** MAX232 entegresi ile MAX232 uyumlu devre ve entegre bağlantı soketi vardır.
- **Enkoder:** 24 tur iki kanal enkoder vardır.
- **Referans regülatörü:** LM336-5V referans regülatörü ve uyumlu devre vardır.
- **Sıcaklık sensörü (analog):** LM35 sıcaklık sensörü uyumlu devre ve LM35 sensör entegresi veya muadili vardır.
- **PWM çıkış:** 10 bit çözünürlükte PWM çıkış vardır.
- **Hoparlör çıkışı:** Ses uygulamaları için hoparlör sürmeye uygun devre ve bağlantı terminali vardır.
- **Sıcaklık sensörü (dijital):** DALLAS DS18xx sıcaklık sensörü uyumlu devre ve entegre bağlantı soketi vardır.
- **12 bit A/D çevirici (harici):** LTC1298 veya muadili 12-bit Seri A/D uyumlu devre ve entegre bağlantı soketi vardır.
- **12 bit D/A çevirici (harici):** LTC1257 veya muadili 12-bit Seri D/A uyumlu devre ve entegre bağlantı soketi vardır.
- **Seri EEPROM (I2C):** 24Cxx seri EEPROM uyumlu devre ve entegre bağlantı soketi vardır.

- **Seri EEPROM (SPI):** 25Cxx seri EEPROM uyumlu devre ve entegre bağlantı soketi vardır.
- **Seri EEPROM (MicroWire):** 93Cxx seri EEPROM uyumlu devre ve entegre bağlantı soketi vardır.
- **Real Time Clock:** NJU6355 RTC uyumlu devre ve entegre bağlantı soketi vardır.

## 2.2 Hata Ayıklayıcı ve Programlayıcı Kartı

RS-232 seri port üzerinden PC'ye ve devre üzerinde seri programlama ICSP bağlantısı üzerinden de 2.1'de anlatılan geliştirme kartına bağlanan, böylece PC üzerinde geliştirilen mikrokontrolör yazılımının mikrokontrolöre aktarılmasını ve kodun satır satır yürütülerek hata ayıklama (debug) işleminin yapılabilmesini sağlayan, Era Bilgi Sistemleri firmasının ([www.infogate.org](http://www.infogate.org)) ürünü, yüzde yüz MPLAB ICD2 uyumlu icdPIC devre içi hata ayıklayıcı ve programlayıcı kartının web sayfasından alınan özellikleri aşağıda verilmiştir.

icdPIC, PIC16Fxxx ve PIC18Fxxx serisi flaş bellekli mikrokontrolörler ve dsPIC30Fxxx serisi DSP'ler için geliştirilmiş bir hata ayıklayıcı (in-circuit debugger) ve güçlü bir programlayıcıdır.

icdPIC, MPLAB geliştirme ortamında çalışır. PIC'e yüklenen makine kodunun icrasında kısmi çalıştırma, adımlama ve durak noktaları koyma gibi emülasyon kontrollerine olanak tanır.

### Hata ayıklayıcı özellikleri (icdPIC):

- Gerçek zamanda kod icrası
- Makina kodunun adım adım icrası
- Durak noktası koyabilme
- 32kHz-40MHz frekans aralığında çalışabilme
- Kaynak düzeyinde ve sembolik hata ayıklama
- PIC'in uygulama kodunun icrası sırasında tüm dahili yazmaçları görüntülenebilir, içerikleri değiştirilebilir.

### **Programlayıcı özellikleri (icdPIC):**

- PIC16Fxxx, PIC18Fxxx ve dsPIC30Fxxx serisi PICmicro denetleyicileri için programlayıcıdır.
- 40 bacaklı ZIF soket vardır.
- PIC'leri buldukları devrede de programlayabilmek için ICSP bağlantısı vardır.
- Windows 98/NT/2000/XP uyumludur.
- PC bağlantısı RS-232 seri port arabirimi üzerinden yapılır.
- MPLAB v7.00 ve üstü sürümlerle çalışır.
- Microchip MPLAB ICD2 uyumludur.

### **2.3 Mikrokontrolör Yazılımı Geliştirme Ortamı**

MICROCHIP firması, web sitesinde ([www.microchip.com](http://www.microchip.com)) PIC mikrokontrolörler için yazılım geliştirme ortamı olarak MPLAB IDE' yi ücretsiz olarak sunmaktadır. MPLAB IDE 32-bit uygulama olarak MS Windows işletim sisteminde çalışır. Kullanması kolaydır, hızlı uygulama geliştirmek ve hata ayıklamak için gerekli bazı yazılımları ücretsiz olarak sunar. Ayrıca üçüncü parti yazılımlar ve donanımlar için de kullanılabilen tek bir grafik kullanıcı ara yüzü çatısı oluşturur. C kodu kullanarak mikrokontrolör yazılımı geliştirebilmek için PICC derleyicisinin “.exe” uzantılı uygulama dosyasının yerinin gösterilmesi yeterlidir. Hatta derleyici yazılımı yüklenirken MPLAB IDE ile bütünleşme kendiliğinden yapılabilmektedir.

Bu tez çalışmasında MPLAB IDE v7.41 sürümü ve C derleyici olarak da HI-TECH firmasının ([www.htsoft.com](http://www.htsoft.com)) PIC18 ailesi mikrokontrolörler için ürettiği PICC18 ANSI C derleyici yazılımı kullanılmıştır.

### **2.4 PIC18F452 Mikrokontrolörünün Özellikleri**

Kullanılan mikrokontrolörün bazı özellikleri aşağıda verilmiştir. Daha ayrıntılı bilgiye MICROCHIP' in web sayfasından ([www.microchip.com](http://www.microchip.com)) ulaşılabilir.

### **Yüksek Performanslı RISC CPU:**

- C derleyici uyumlu mimari/komut kümesi vardır.
- Lineer adreslenebilen 32K sekizlik (16384 kelime) FLAŞ program belleği, 1536 sekizlik veri belleği (RAM), 256 sekizlik EEPROM veri belleği tümleşik olarak vardır.
- Saniyede 10 milyon komut işleyebilme:
  - DC - 40 MHz osilatör/saat girişi
  - PLL aktif iken 4 MHz - 10 MHz osilatör/saat girişi
- 16-bit komut uzunluğu, 8-bit veri uzunluğu
- Kesmeler için öncelik seviyeleri
- Tek çevrimde 8-bit x 8-bit donanımsal çarpma yapabilir

### **Tümleşik Çevre Birimlerinin Özellikleri:**

- Yüksek akım çıkışlı (batan/kaynak: 25 mA/25 mA)
- Üç adet harici kesme bacağı
- Timer0 birimi: 8-bit programlanabilir ön bölücü ile 8-bit/16-bit zamanlayıcı/sayıcı
- Timer1 birimi: 16-bit zamanlayıcı/sayıcı
- Timer2 birimi: PWM periyodu için de kullanılan 8-bit periyot yazmacı, 8-bit zamanlayıcı/sayıcı
- Timer3 birimi: 16-bit zamanlayıcı/sayıcı
- İkincil osilatör saat seçeneği - Timer1/Timer3
- İki adet Capture/Compare/PWM (CCP) birimi: CCP bacakları şu üç ayrı durumda kullanılmak üzere ayarlanabilir,
  - Yakalama girişi: yakalama 16-bit'tir, maksimum çözünürlük 6.25ns (TCY/16)'dir.



- Karşılaştırma 16-bit'tir, maksimum çözünürlük 100ns (TCY)'dir.
- PWM çıkışı: PWM çözünürlüğü 1 ila 10-bit'tir, maksimum PWM frekansı 8-bit çözünürlük için 156kHz, 10-bit çözünürlük için 39kHz'dir.
- Ana Senkron Seri Port (MSSP) birimi: İki biçimde çalışabilir,
  - 3-hat SPI™ (tüm 4 SPI biçimini destekler)
  - I2C™ Ana ve Köle biçimi
- Adreslenebilir USART birimi vardır: RS-485 ve RS-232'yi destekler
- Paralel Köle Port (PSP) birimi vardır.
- 10-bit çözünürlüklü Analog Dijital Çevirici birimi vardır:
  - Hızlı örnekleme oranı
  - UYKU durumunda iken çevrim aktif
  - Lineerlik  $\leq 1$  bit
- Programlanabilir düşük besleme gerilimi algılama (PLVD): kesme üretebilir.
- Programlanabilir yığın taşması reset (BOR)

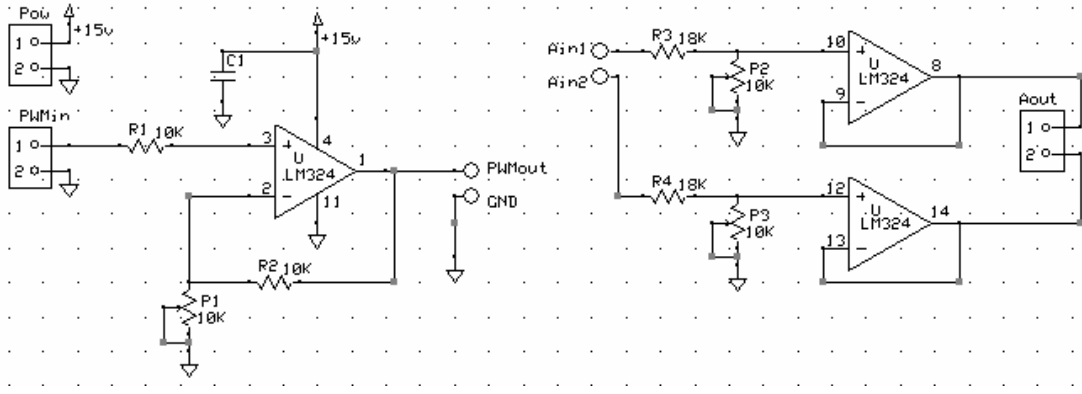
#### **Özel Mikrokontrolör Özellikleri:**

- 100,000 sil/yaz çevrimli geliştirilmiş FLAŞ program belleği
- 1,000,000 sil/yaz çevrimli EEPROM veri belleği
- FLAŞ/EEPROM veri saklama süresi > 40 yıl
- Yazılım kontrolü altında kendi kendini programlayabilir
- Besleme verildiğinde resetlenebilme (POR), Besleme verildi zamanlayıcılı (PWRT), osilatör başlatma zamanlayıcılı (OST), Güvenli çalışma için kendi RC saat üretici ile çalışan bekçi köpeği zamanlayıcılı (WDT)
- Programlanabilir kod korumalı
- Güç tasarrufu için UYKU durumu

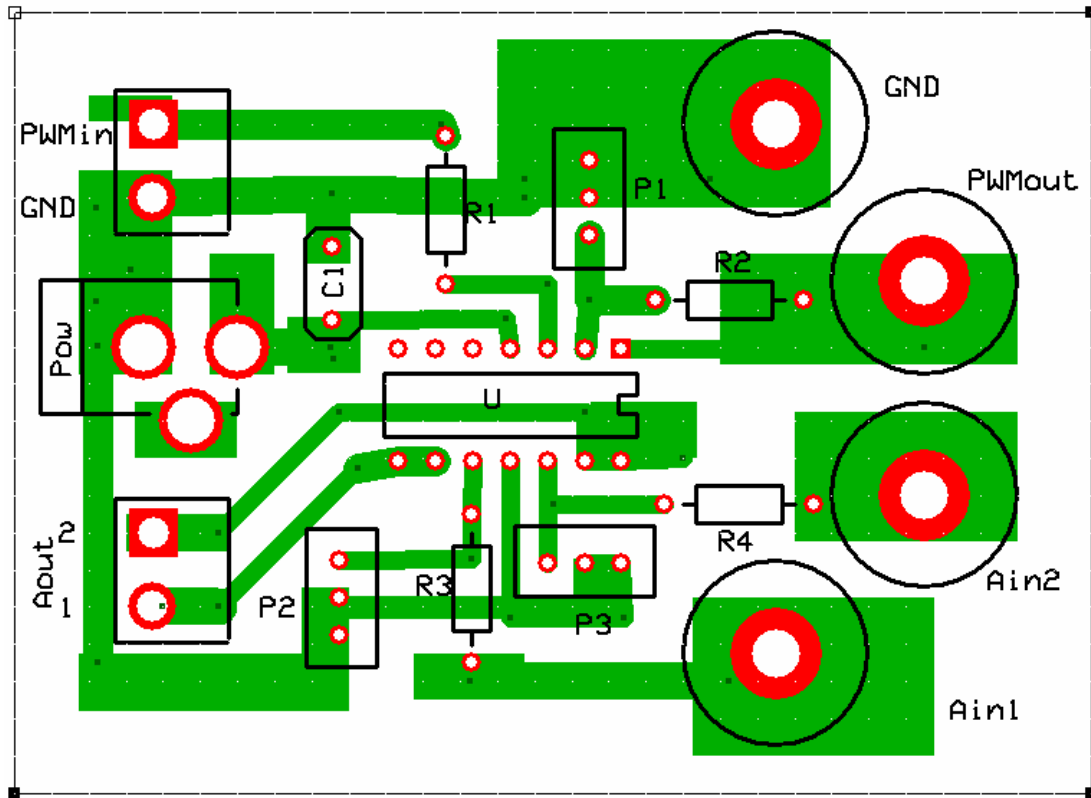
- Seçime bağlı saat işareti:
  - Birincil osilatör kaynağı frekansını PLL iç devresiyle 4 katına çıkarabilir.
  - İkincil osilatör saat girişi (32kHz)
- 5V tek besleme kaynağı ve iki bacak (RB7, RB6) ile devre içi seri programlama (ICSP™)
- İki bacak ile devre içi hata ayıklama (ICD)
- CMOS teknolojisi, düşük güç, yüksek hız FLAŞ/EEPROM teknolojisi
- Geniş besleme gerilimi aralığı (2.0V to 5.5V)
- Endüstriyel ve iyileştirilmiş sıcaklık aralıklarında çalışabilme
- Düşük güç tüketimi:
  - < 1.6 mA @ 5V, 4 MHz
  - 25 µA @ 3V, 32 kHz
  - < 0.2 µA hazırda bekleme akımı

## 2.5 Gerilim Kuvvetlendirici Kartı

PT326 ısı deney seti çıkan havanın sıcaklığına karşılık 0-10V aralığında gerilim çıkışı vermekte, kontrol girişi olarak da 0-10V aralığında gerilim girişi kabul etmektedir. Diğer tarafta, mikrokontrolörün analog giriş okumak üzere kullanılan ADC biriminin ve analog çıkış üretmek üzere kullanılan PWM biriminin çalışma aralığı 0-5V'tur. Bu nedenle, mikrokontrolör kartı ile PT326 seti arasında gerilim uyumunu sağlayacak şekilde Şekil 2.1'de 3 numara ile gösterilen 0-5V↔0-10V gerilim kuvvetlendirici kartı üretilmiştir. Bu kartın devre şeması ve baskı devre çizimi sırasıyla Şekil 2.2 ve Şekil 2.3'te verilmiştir.



Şekil 2.2: Gerilim Kuvvetlendirici Kartı Devre Şeması



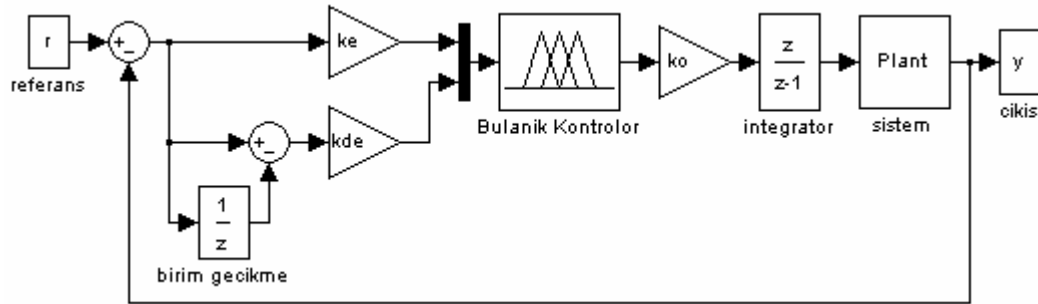
Şekil 2.3: Gerilim Kuvvetlendirici Kartı Baskı Devre Çizimi

### 3. KONTROLÖR YAZILIMININ GERÇEKLENMESİ

Bu tez çalışmasına öncelikle klasik bir bulanık iki girişli PD-I kontrolör yapısının uygulaması olarak başlanmıştır. Daha sonra İ.T.Ü. Fen Bilimleri Enstitüsü tezlerinden Hakkı Murat Genç tarafından 2006'da yapılan tez çalışmasının [13] ve bildirisinin [14] benzeşim çalışmaları bu gerçek zaman çalışmasına aktarılmıştır. Yük. Müh. Hakkı Murat Genç'in tez çalışmasının özeti aşağıda aynen verilmiştir.

Ölü zamanlı sistemlerde, sistemden gözlenen bilgi geçmiş bir ana aittir ve bu gecikmeli işareti kullanmak, kontrol sistemi uygulamalarında başarısız sonuçlara neden olabilir. Bu çalışma, ölü zaman bilgisinin sistem performansını artırmak adına kural tabanının yeniden düzenlenmesinde nasıl kullanılacağı ile ilgilidir. Temel olarak, sistemin ölü zamanından kaynaklanan bilgi gecikmesinin kural tabanının uygun şekilde kaydırılması ile telafisi önerilmektedir. Kural tabanı kaydırma yöntemini etkileyen değiştirgeler (parametreler) detaylı bir biçimde incelenmiş ve kaydırma yöntemi, sistem model değiştirgelerine göre çizelgelenmiştir. Yeni yöntem birçok farklı sistemde denenmiş ve etkisi ortaya konmuştur. Yöntemin, ölü zaman değişimlerine, yapısal ve ayar değiştirgelerinin değişimlerine ve sistem model değiştirgelerindeki belirsizliklere karşı gürbüzlüğü tüm hatlarıyla ortaya konmuştur.

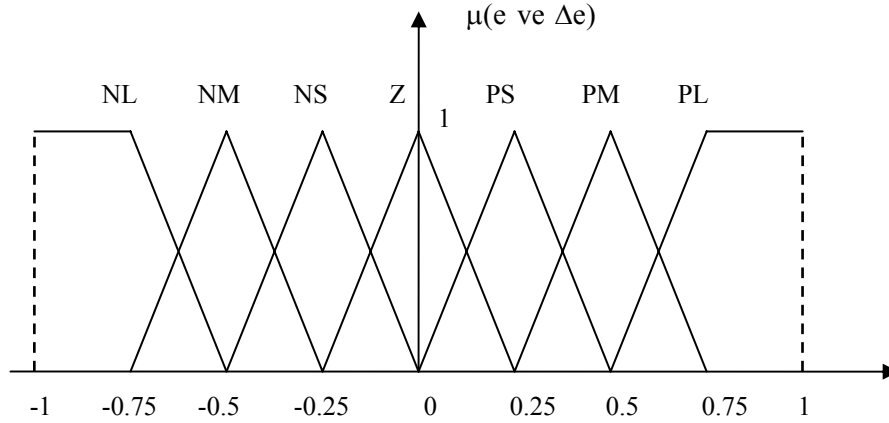
[13]'de ve bu çalışmada, hata ( $e$ ) ve hatanın değişimi ( $\Delta e$ ) giriş, kontrol işaretinin değişimi ( $\Delta u$ ) çıkış olmak üzere çoğu uygulamalar için en iyi çözüm olan en genel bir PI tipi bulanık kontrolör kullanılmıştır. Tüm kontrol çevriminin blok şeması Şekil 3.1'de gösterilmiştir.



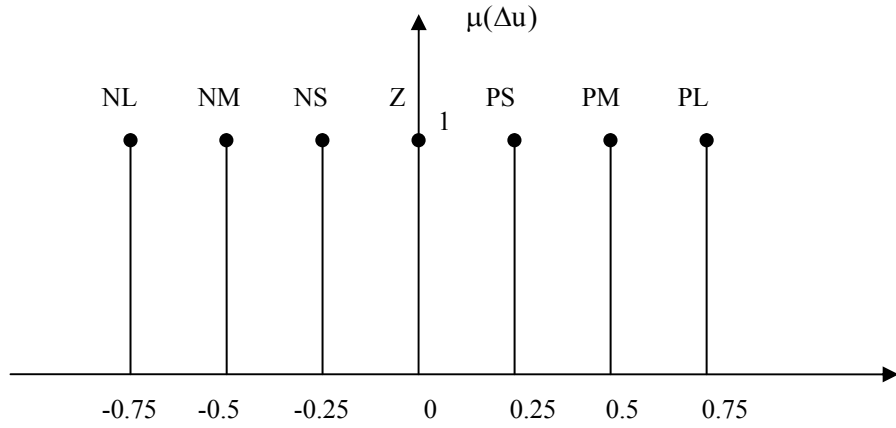
Şekil 3.1: PI Tipi Bulanık Kontrolör ile Temel Kontrol Blokları

Kullanılan PI tipi bulanık kontrolörün yapısı [12]'deki makalede incelenmiştir. Makalede Mizumoto'nun çarpım-toplamı (product-sum) yönteminin kullanıldığı çarpım-toplamı keskin bulanık kontrolörün (product-sum crisp type fuzzy controller) dinamik davranışı incelenmiş ve bu tip bulanık kontrolörün, parametreleri zamanla değişen geleneksel PD tipi kontrolör gibi davrandığı matematiksel olarak gösterilmiştir. Bulanık kontrolörün çıkışına eklenen bir integral alıcı ile de PID tipi bulanık kontrolör elde edilebilmektedir. Böylece makalede, geleneksel PID kontrol teorisine uygun yapıda bir PID tipi bulanık kontrolör elde edilmiş ve benzetim sonuçları verilerek çalışması incelenmiştir.

Girişler ve çıkış için yedişer üyelik fonksiyonu kullanılmıştır. Girişler için kullanılan üyelik fonksiyonları Şekil 3.2'de ve çıkış için kullanılan üyelik fonksiyonları Şekil 3.3'te verilmiştir. Üyelik fonksiyonları NL (negatif büyük), NM (negatif orta), NS (negatif küçük), Z (sıfır), PS (pozitif küçük), PM (pozitif orta), PL (pozitif büyük) olarak isimlendirilmiştir. Tekli (singleton) çıkış ve yedişer üyelik fonksiyonlu girişler için yalnızca 49 kurallı bir bulanık kontrolör yapısı gerçek dünya uygulamaları için gerçekleşmesi kolay ve yeterli çözünürlüklü bir yapıdır. Hata ve hatanın değişimi girişleri için kullanılan en genel haldeki bulanık kontrolör kural tabanı Tablo 3.1'de verilmiştir.



**Şekil 3.2:** e ve  $\Delta e$  için Üyelik Fonksiyonları



**Şekil 3.3:**  $\Delta u$  için Üyelik Fonksiyonları

**Tablo 3.1:** En Genel Bulanık Kontrolör Kural Tabanı

$\Delta e / e$	NL	NM	NS	Z	PS	PM	PL
PL	Z	PS	PM	PL	PL	PL	PL
PM	NS	Z	PS	PM	PL	PL	PL
PS	NM	NS	Z	PS	PM	PL	PL
Z	NL	NM	NS	Z	PS	PM	PL
NS	NL	NL	NM	NS	Z	PS	PM
NM	NL	NL	NL	NM	NS	Z	PS
NL	NL	NL	NL	NL	NM	NS	Z

### 3.1 Bulanık Çıkarım Yapısının Oluşturulması

Kontrol ve Kumanda Sistemleri Ana Bilim Dalımızda kontrol sistemlerinin tasarımı ve incelenmesi için daha çok MATLAB Simulink kullanılmaktadır ve bulanık kontrolör tasarımı için de yine MATLAB 'Fuzzy Logic Toolbox' ara yüzü kullanılmaktadır. Bugüne kadar yapılmış çalışmalarla uyumluluk açısından

mikrokontrolörler için bulanık kontrolör tasarımında kullanılan FUZZYTECH ([www.fuzzytech.com](http://www.fuzzytech.com)) gibi yeni yazılımlar kullanmak tercih edilmemiştir. MATLAB ile uyumlu, bulanık çıkarım için gerekli kayan noktalı işlemleri de donanımsal olarak yapabilen MOTOROLA MPC555 kartı kullanılarak gerçek zamanlı işler yapılmak istenmiştir. İ.T.Ü. Fen Bilimleri Enstitüsü'ne yüksek lisans çalışmasının desteklenmesi için bir proje sunulmuş ve söz konusu kartın alımı için Amerikan PHYTECH firmasının ([www.phytech.com](http://www.phytech.com)) Almanya'daki dağıtıcısı ([www.phytech.de](http://www.phytech.de)) ile Türkiye'deki NETES firması aracılığıyla bağlantıya geçilmiş, sonunda kart elimize ulaşmış ve üzerinde çalışmalar yapmaya başlanmıştır.

Aynı zamanda, MATLAB 'Fuzzy Logic Toolbox' ile oluşturulmuş ve diske '.fis' uzantısı ile kaydedilmiş bulanık çıkarım yapısının bir şekilde açılıp çözülerek gerekli parametrelerin çekilmesi ile bulanık kontrol yapısı C dilinde gerçekleştirilerek öncelikle elimizde hazır bulunan PIC geliştirme kartları ile gerçek zamanlı çalışma işinin yapılması düşünülmüştür.

MATLAB C dili ile yazılmış bir program olduğundan kendi kullandığı bulanık çıkarım mekanizmasının nasıl gerçekleştirildiğini araştırmakla yola çıkılmıştır. MATLAB üreticisi olan MathWorks'ün web sayfasında ([www.mathworks.com](http://www.mathworks.com)) 'support', 'documentation', 'Fuzzy Logic Toolbox' yardım sayfası incelenmiştir. Bu sayfadaki 'Stand-Alone Fuzzy Inference Engine' başlığı altında sunulan yardım bilgileri aşağıda verilmiştir.

### **Bağımsız Uygulama için C Dilinde Bulanık Çıkarım Mekanizması:**

MATLAB Fuzzy Logic Toolbox aracının içerisinde fuzzy klasörünün altında 'fismain.c' ve 'fis.c' isimli iki C dosyası vardır. Bu C dosyaları bulanık çıkarım mekanizmasının MATLAB ortamından bağımsız olarak da çalıştırılabilmesi için kaynak kod olarak sunulmuştur. Bağımsız C dilindeki bulanık çıkarım mekanizması ile bir FIS dosyası ve giriş verisi dosyası okunarak bulanık çıkarım işlemi doğrudan yapılabilir ya da diğer dış uygulama ortamlarını gömülebilir.

Bağımsız bulanık çıkarım mekanizmasını bir UNIX sistemde derlemek için şu komut çalıştırılır.

```
% cc -O -o fismain fismain.c -lm
```

(Not: % UNIX komut satırı simgesidir ve fis.c'nin komut satırında belirtilmesine gerek yoktur, çünkü fis.c zaten fismain.c'de dahil edilmiştir).

Başarılı derleme sonucunda oluşturulan çalıştırılabilir dosya ismi aşağıdaki gibi komut satırında çalıştırılarak mekanizmanın çalışması görülebilir.

```
% fismain
```

Bu komut çalıştırıldığında ekranda şu uyarıyla karşılaşılır.

```
% Usage: fismain data_file fis_file
```

Bu uyarı mesajına göre fismain uygulaması düzgün olarak çalışabilmek için iki dosyaya ihtiyaç duymaktadır: satırlar halinde giriş vektörleri içeren bir veri dosyası ve söz konusu bulanık çıkarım sistemini belirten bir FIS dosyası.

Örneğin, mam21.fis isimli bir FIS yapısı dosyası ele alınsın. Giriş veri dosyası MATLAB kullanılarak şu şekilde hazırlanabilir.

```
[x, y] = meshgrid(-5:5, -5:5);
```

```
input_data = [x(:) y(:)];
```

```
save fis_in input_data -ascii
```

Bu MATLAB komutları ile tüm giriş verisi 121x2'lik bir matris olarak ASCII formatında fis\_in dosyasında saklanır. Bu matrisin her satırı bir giriş vektörüdür.

Artık bağımsız çalışabilen uygulama şu komutla düzgün olarak çalıştırılır:

```
% fismain fis_in mam21.fis
```

Bu komut ekranda 121 adet çıkış üretir. Çıkışlar şu komutla başka bir dosyaya yönlendirilebilir:

```
% fismain fis_in mam21.fis > fis_out
```

Artık fis\_out dosyası 121x1'lik bir çıkış matrisi içermektedir. Genelde çıkış matrisinin her satırı bir çıkış vektörünü temsil eder. Tüm matrislerin dosyalarla yer değiştirmesinin dışında fismain'in kullanımını evalfis.m MATLAB Mex dosyasına benzer.

Mex dosyası ile bağımsız çalışan uygulamanın ürettikleri çıkışları karşılaştırmak için MATLAB'de şu komutlar yürütülebilir:



```
fismat = readfis('mam21');  
  
matlab_out = evalfis(input_data, fismat);  
  
load fis_out  
  
max(max(matlab_out - fis_out))  
  
ans =  
  
4.9583e-13
```

Bu çok küçük fark fis\_out dosyasına sınırlı uzunluklu veri yazılabilmesinden kaynaklanır. Bağımsız çalışabilen uygulama hakkında bilinmesi gereken daha çok şey vardır:

- ANSİ C uyumludur.
- Özel tanımlanmış fonksiyonlar bağımsız çalışan uygulamada kullanılamaz, yani kullanıcı Toolbox ile gelen 11 adet üyelik fonksiyonu ile sınırlıdır. AND, OR, IMP ve AGG fonksiyonları da hazır halleriyle kullanılabilir.
- fismain.c main() ana fonksiyonu içerir ve diğer uygulamalarda kullanımda kolay uyarlanabilmesi için yüklü miktarda açıklama içerir.
- Bağımsız çalışan koda yeni bir üyelik fonksiyonu veya yeni bir çıkarım mekanizması eklemek için bulanık çıkarım süreci için gerekli tüm fonksiyonları içeren fis.c dosyasının değiştirilmesi gerekmektedir.

Bulanık kontrol yapısında girişlerin bulandırılması, ateşleme değerlerinin bulunması, bir araya getirilmesi (aggregation) ve durulanması işleri dinamik olarak yapıldığından, bu yapının mikrokontrolörün veri belleğinde (RAM) yer alması gerekmektedir. Ne yazık ki PIC18F452 mikrokontrolörünün veri belleği 1536 sekizlik ile sınırlı olduğundan MATLAB bulanık çıkarım yapısı aynen gerçekleştirilememiştir. Mamdani tipi bulanık yapı çıkarılmış, yalnızca Sugeno tipine izin verilmiştir. 3.2’de anlatıldığı gibi program belleği sınırlamasından dolayı da girişler için yalnızca üçgen ve trapezoid üyelik fonksiyonları kullanılabilir. Endüstride kullanılan bulanık çıkarım mekanizmaları çoğunlukla bu şekilde olduğundan bu kısıtlamalar üzerinde şimdilik durulmayacaktır.

Bahsedilen kısıtlamalar ile ‘fis.c’ dosyasının içindeki bulanık çıkarım yapısı mikrokontrolör veri belleğine sığacak şekilde düzenlenmiş ve bulanık veri yapısı

(fuzzy data structure) 'fds.h' başlık dosyası içinde oluşturulmuştur. Bu başlık dosyası yine 3.2 kısmında anlatılan PICC dili ile yazılmış olan 'fisP1.c' dosyasının içinde kullanılmıştır.

Bölüm başında üyelik fonksiyonları ve kural tabanı verilmiş olan bulanık çıkarım yapısı MATLAB Fuzzy Logic Toolbox fis editör ile kolayca hazırlanmış ve diske 'xxxx.fis' olarak kaydedilmiştir. Yukarıdaki "Bağımsız Uygulama için C Dilinde Bulanık Çıkarım Mekanizması" yardım bilgilerinden hareketle MATLAB'in sunmuş olduğu 'fis.c' aynen kullanılarak ve 'fismain.c' kodunda değişiklikler yapılarak elde edilen C programı ile 'xxxx.fis' dosyasında kayıtlı bulanık çıkarım yapısındaki parametreler çekilmiş, bir önceki paragrafta anlatılan, mikrokontrolörün veri belleğinde olması gereken bulanık veri yapısındaki yerlerine yazacak olan fis\_init() C fonksiyonunu yeni oluşturulan 'fis\_init.c' dosyasına yazdırılmıştır. Böylece mikrokontrolör ile gerçek zamanlı olarak çalıştırılacak olan bulanık kontrolörün bulanık çıkarım yapısı PIC mikrokontrolör için C derleyicisi olan PICC18'in anlayacağı hale getirilmiştir.

'fismain.c' kodu Visual Studio 6.0 C derleyicisi ile derlenip 'fismain.exe' uygulama programı elde edilmiştir. Daha sonra DOS komut satırından, içinde 'fismain.exe', 'xxxx.fis' dosyalarının bulunduğu klasöre girilmiş ve "fismain.exe xxxx.fis" komutu çalıştırılarak, yine aynı klasörde 'fis\_init.c' dosyası oluşturulmuştur. Bu 'fis\_init.c' dosyasında bulanık çıkarım yapısı vardır ve daha sonra asıl geri beslemeli kontrol işleminde kullanılacak olan diğer PICC kodlarıyla birlikte derlenerek mikrokontrolöre gömülecektir.

### **3.2 Bulanık Çıkarım Mekanizmasının PICC Dilinde Gerçeklenmesi**

MATLAB'in kendi 'fis.c' kodu içerisinde MATLAB bulanık çıkarım yapısı ve bulanık çıkarım mekanizmasında kullanılan üyelik fonksiyonları, 't-norm', 't-conorm' operatörleri, durulama fonksiyonları, girişlerin bulandırılması için gerekli olan fonksiyonlar, ateşleme değerlerinin hesaplanması için gerekli fonksiyon ve durulama işlemi için gerekli diğer fonksiyonlar bulunmaktadır. Fakat bu haliyle doğrudan bir mikrokontrolör üzerinde kullanılması mümkün değildir. En temel sebebi de kodun, mikrokontrolörün 16K kelimelelik program hafızası için çok büyük olmasıdır. Ayrıca ekrana yazdırmak için kullanılan printf() kullanımları vardır fakat iki satırlı ve yirmi dört sütunlu bir LCD'den başka mikrokontrolöre bağlı bir ekran yoktur. Dolayısıyla bu komutlar çıkarılmıştır. 3.1 kısmında anlatıldığı gibi dinamik bulanık çıkarım yapısının, mikrokontrolörün veri belleğine sığmaması nedeniyle

bulanık kontrolörün tipi Mamdani olmadığından buna ilişkin kodlar da 'fis.c' dosyasından çıkarılmıştır. Giriş üyelik fonksiyonu olarak yalnızca üçgen ve trapezoid biçimlerine izin verilmiş, durulama için de yalnızca 'centroid' kullanılmıştır. Bu türler endüstriyel uygulamalarda en yaygın olarak kullanılan türlerdir. Bu nedenle bu kısıtlamalar üzerinde durulmamıştır.

Sonuç olarak giriş üyelik fonksiyonları üçgen veya trapezoid olan Sugeno tipi bulanık kontrolör için çıkarım mekanizması PICC dilinde yazılarak 'fisP1.c' isimli C dosyası oluşturulmuştur. Bu C dosyası program belleğinde büyük yer kapladığından ana fonksiyonun (main()) bulunduğu C dosyası içine dahil edilmeyip, aynı proje çatısı altında fakat ayrı olarak derlenmiştir.

### **3.3 Ana Kontrol Yazılımı (main())**

Kontrol sistemleri geri beslemeli sistemlerdir. Sistem çıkışı bilgisinin analog dünyadan okunup sayısal olarak değerlendirilmesi ve üretilen sayısal kontrol işaretinin analog dünyadaki gerçek sisteme uygulanması gerekmektedir. Bu çalışmada kullanılan, bölüm başında blok diyagramı verilen kontrol çevrimi için bu işlemler sırasıyla PIC mikrokontrolörün içinde tümleşik olarak yer alan sekiz kanallı analog dijital çevirici (ADC) ve çıkışına haricen bir direnç-kondansatör (RC) alçak geçiren filtresi bağlanmış darbe genişlik modülasyonu (PWM) birimleri ile yapılmıştır.

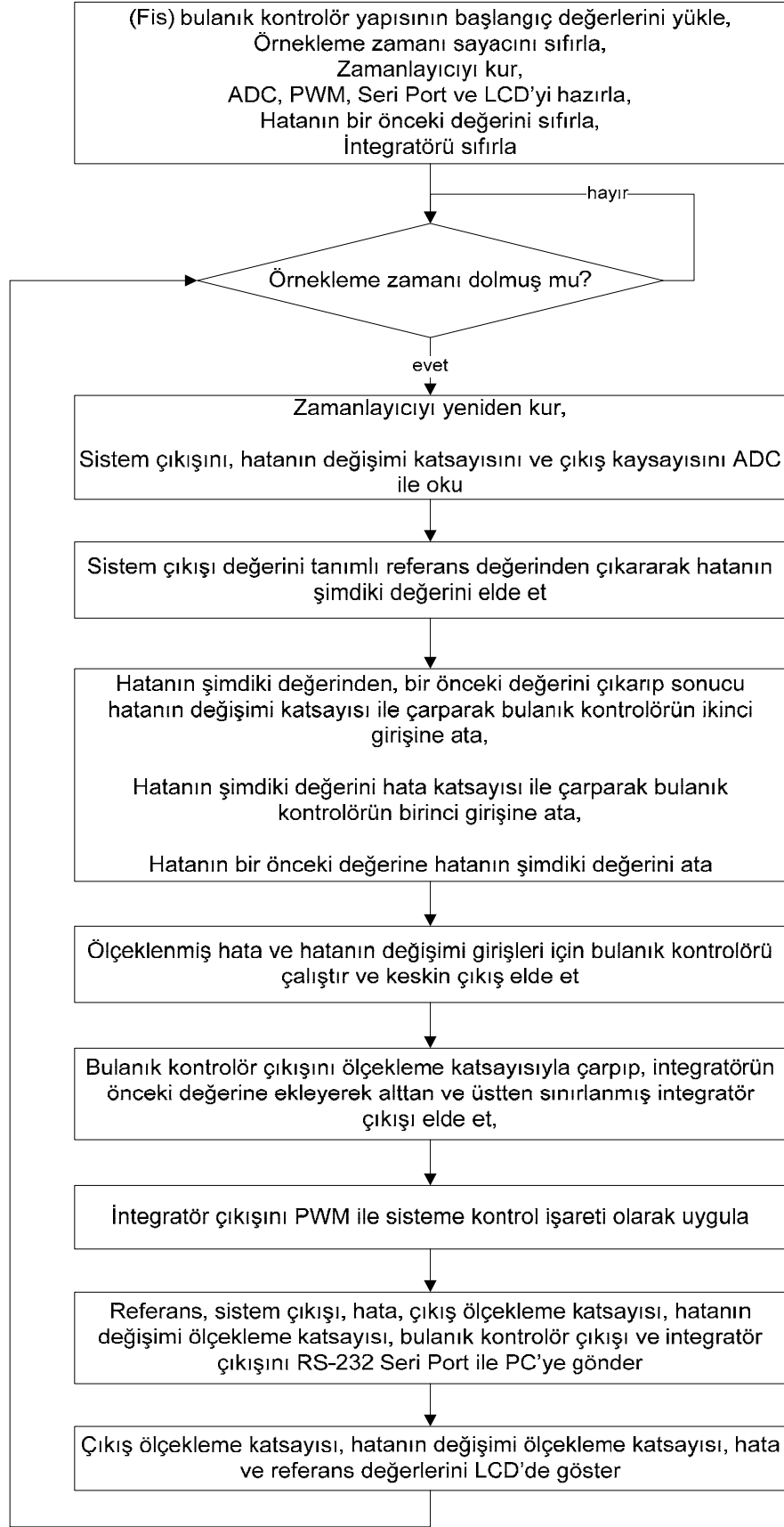
Bahsedilen ADC ve PWM birimlerinin kullanıma hazırlanması ve kullanılması için gerekli olan 'adc.c' ve 'pwm.c' kodları PICC dilinde yazılıp ana programa dahil edilerek kullanılmıştır. Bulanık kontrolörün giriş, çıkış ölçekleme katsayıları da yazılım geliştirme sırasında kolayca değiştirilebilmesi açısından mikrokontrolörün ADC birimi kullanılarak kullanıcıdan alınmıştır ve yine ADC ile okunan hata bilgisi ile birlikte Seiko firmasının ürünü 2x24 karakterlik LCD göstergede gösterilmiştir.

Ayrıca tüm bu hata, ölçekleme katsayıları, üretilen sayısal kontrol işareti bilgisi incelenmek ve kaydedilmek üzere UART RS-232 seri haberleşme ile PC'ye aktarılmıştır. PC üzerinde seri portu dinlemesi ve gelen verileri matris tipi bir değişken içinde saklaması için MATLAB 'seri1.m' dosyası yazılmıştır. Böylece bulanık kontrolör tasarımı için de kullanılan MATLAB yazılımı dışına çıkılmamıştır. LCD ve USART seri haberleşme birimlerinin kullanılabilmesi için gerekli fonksiyonların bulunduğu 'lcd.c' ve 'serial.c' dosyaları oluşturulup ana programa dahil edilmiştir.

Ayrık kontrol için hayati önem taşıyan örnekleme zamanı bilgisi zamanlayıcı kesmesi ile elde edilmiştir. Buna ilişkin 'timer1.c' dosyası yazılıp ana programa dahil edilmiştir.

Bulanık çıkarım yapısını içeren 'fds.h' başlık dosyası da ana programa eklenip, ayrıca derlenen 'fis\_init.c' dosyasının içindeki, bulanık çıkarım yapısının MATLAB '.fis' dosyasından çekilen değerlerini atayacak olan 'fis\_init()' fonksiyonu ve yine ayrı olarak derlenen 'fisP1.c' içindeki, hata ve hatanın değişimi bilgilerini bulanık çıkarım yapısı üzerinde işleyerek çıkarımı yapacak olan 'getFisOutput()' fonksiyonu da harici (extern) olarak ana programa tanıtılmıştır.

Ana programda sonsuz döngü içinde, her 0.1 saniyelik örnekleme zamanında bir, sistem çıkışı, hata değişimi ve çıkış ölçekleme katsayıları değerleri okunur. Yazılımsal olarak girilen referanstan sistem çıkışı çıkarılarak hata bilgisi ve bu hatadan bir önceki çevrimde hesaplanmış olan hata çıkarılarak da hatanın değişimi bilgisi elde edilir. Hata, yazılımsal olarak girilen '0.75/referans' ölçekleme katsayısı ile çarpılarak bulanık kontrolör için önceden belirlenen  $[-0.75, 0.75]$  aralığına ölçeklenir. Hatanın değişimi de PicLAB kartı üzerindeki potansiyometreden okunan ölçekleme katsayısı değeri ile çarpıldıktan sonra hata ve hatanın değişimi değerleri bulanık çıkarım mekanizmasına giriş olarak verilir ve 'getFisOutput()' fonksiyonu ile bulanık kontrolör çıkışı elde edilir. Bulanık kontrolör çıkışı, çıkış ölçekleme faktörü ile çarpılarak, 10-bit çözünürlüklü  $[0, 1023]$  aralığında sınırlı sayısal kontrol işareti PWM birimi ile 0-5V analog doğru gerilim değerine dönüştürülür. Son olarak da gerekli değerler LCD'ye ve UART birimi üzerinden de PC'ye gönderilir. Bu şekilde kontrol çevrimi sürer. Sözle anlatılmış olan bu ana kontrol çevriminin akış diyagramı aşağıda verilmiştir.



Şekil 3.4: Ana Kontrol Yazılımı Akış Diyagramı

#### 4. ISIL SİSTEM ÜZERİNDE DENEYSEL GERÇEKLEMELER

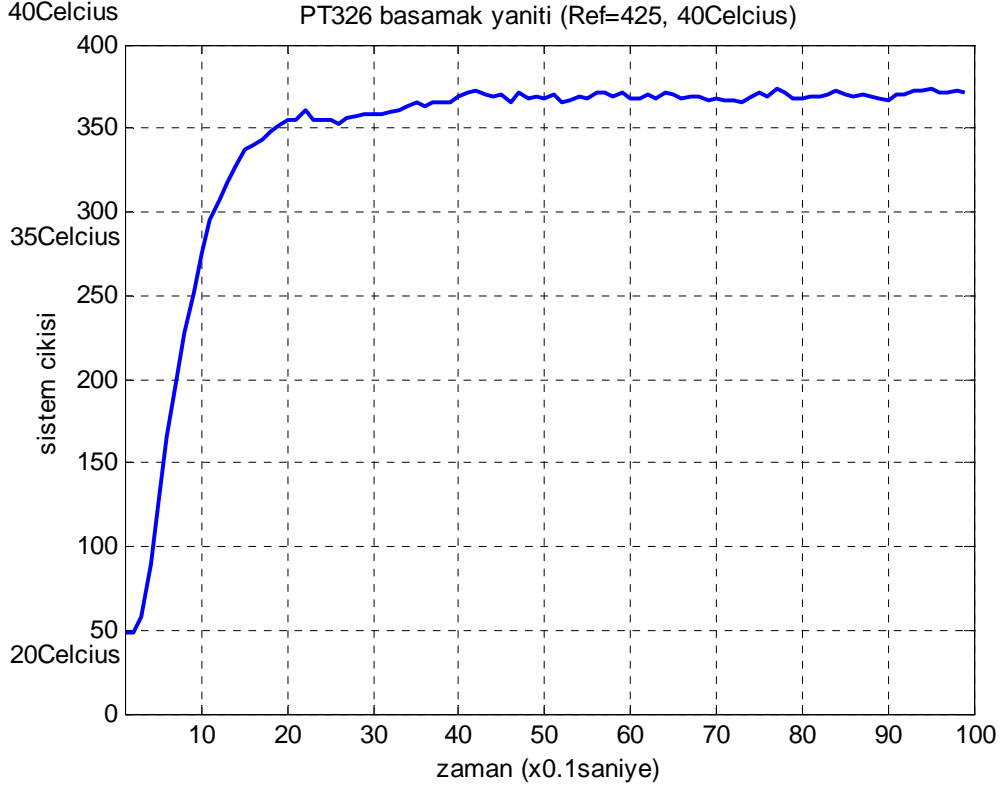
Ölçekleme katsayıları bulanık kontrolün önemli bir parçasıdır. Bulanık çıkarım mekanizmasının iyi çalışabilmesi için kural tabanının iyi tanımlanmış olması gerekir. Kural tabanı ise girişlerin bazı beklenen değerleri için çıkışın olması istenen değerine göre bir uzman tarafından belirlenmiştir. Kural tabanının anlamlı olabilmesi ancak bu girişlerin ve çıkışın baştan kural tabanı düşünülürkenki değer aralıklarına ölçeklenmiş olmasıyla mümkündür. Genelde bulanık kontrolör  $[-1 \ 1]$  aralığında çalışacak şekilde tasarlanır. Bu çalışmada ise Hakkı Murat Genç'in tez çalışmasında kullandığı parametreler kullanılmak istendiğinden girişler ve çıkış  $[-0.75 \ 0.75]$  aralığında olacak şekilde üçgen giriş ve tekli çıkış üyelik fonksiyonları seçilmiştir (Şekil 3.2, Şekil 3.3).

Hata katsayıları için uygun değerler bulunabilmesi için PT326 ısı sistemini model olarak çıkarılarak MATLAB Simulink ile kontrol sisteminin benzetim modeli elde edilmiştir. MATLAB genetik algoritma aracı kullanılarak bu katsayıların değerleri kestirilmiştir. Benzetim ile elde edilen ölçekleme katsayıları PIC mikrokontrolör ile gerçekleştirilen gerçek zamanlı kontrolöre potansiyometreler ile analog olarak girilmiştir. Benzetim sonuçları ile gerçek zamanlı olarak çalıştırılan kontrol sisteminden elde edilen sonuçlar karşılaştırılmıştır. En son olarak da İ.T.Ü. Fen Bilimleri Enstitüsü Projesi kapsamında alınan 32-bit MOTOROLA işlemcili "phyCORE-MPC555 Rapid Development Kit" geliştirme kartı ile MATLAB Simulink ortamında kurulan bulanık kontrolörlü modelin MATLAB Real Time Workshop ile doğrudan bu karta yüklenip kart üzerinde gerçek zamanlı olarak çalışması gerçekleştirilmiştir. Bu kart ile yapılan çalışmalar bölüm sonunda 4.4 kısmında verilmiştir.

##### 4.1 PT326 Modelinin Elde Edilmesi

Kontrol sisteminde gerçek sistem olarak kullanılan PT326 ölü zamanlı ısı sisteminin modelinin elde edilebilmesi için PIC mikrokontrolör kartına, sisteme yaklaşık  $40^{\circ}\text{C}$  karşılığı 425 birimlik basamak giriş uygulayıp sistem çıkışını MATLAB seri port dinleme kodu çalışan PC'ye aktaran program yüklenip, 0.1 saniye örnekleme

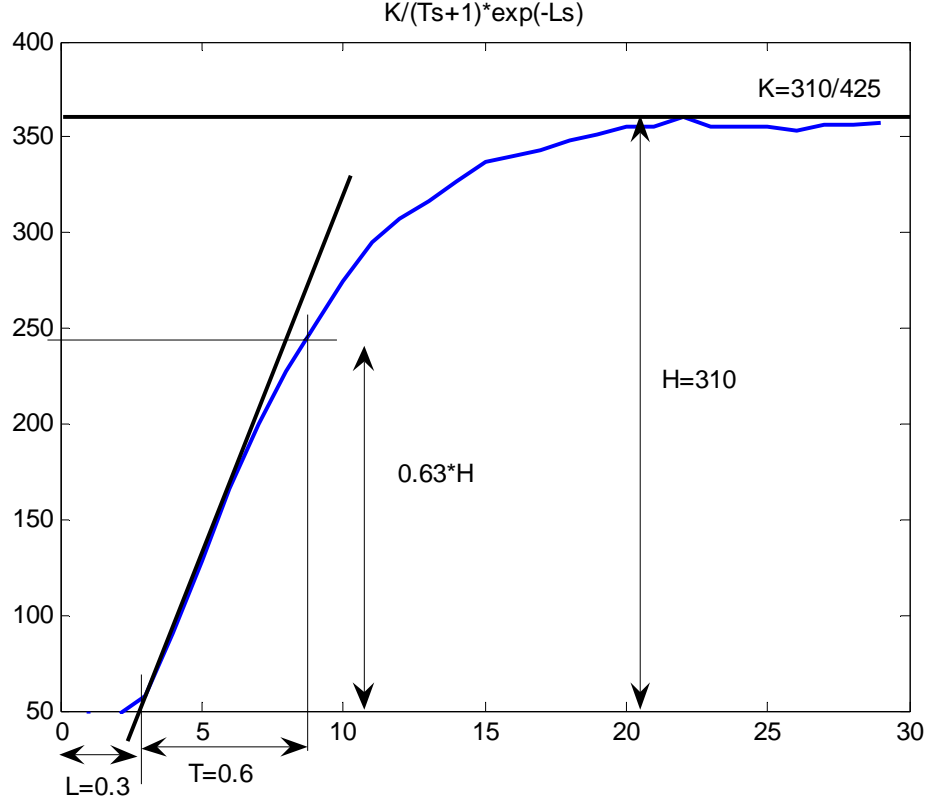
periyodunda gerçek zamanlı olarak çalıştırılmıştır. Sistem yanıtı Şekil 4.1'deki gibi elde edilmiştir. Şekilde yatay eksen 0.1s çarpanla saniye cinsinden zamanı, dikey eksen ise 10-bit çözünürlükle [0, 1023] aralığında sayısal olarak okunan sistem çıkışını göstermektedir. Termistörden okunan °C cinsinden fiziksel hava sıcaklığı değeri de yaklaşık olarak şu şekildedir:  $50 \cong 20^\circ\text{C}$ ,  $300 \cong 35^\circ\text{C}$ ,  $425 \cong 40^\circ\text{C}$ .



**Şekil 4.1:** PT326 Isıl Sisteminin Basamak Yanıtı

Genel olarak basamak yanıtına bakıldığında sistemin birinci dereceden ölü zamanlı bir sistem olduğu söylenebilir ki transfer fonksiyonu,  $K$  kazanç,  $T$  zaman sabiti ve  $L$  ölü zaman olmak üzere  $\frac{K}{Ts+1}e^{-Ls}$  şeklindedir. Görüldüğü gibi sistem çıkışı, ortamın sıcaklığının da zamanla artmasından dolayı pozitif geri besleme altında sürekli olarak artma eğilimindedir. Bu nedenle yaklaşık 3 saniye sonunda sistemin yerleştiğini düşünerek ilk 3 saniyelik kısma Şekil 4.2'de daha yakından bakılıp Ziegler-Nichols yöntemi ile  $K$ ,  $T$  ve  $L$  parametrelerinin kabaca bulunması gösterilmiştir.  $K$  sistem kazancı, sistem çıkışının değişiminin basamak işaretinin değişimine oranıdır. Basamak işareti  $0^\circ$ dan  $425^\circ$ e değiştiğinden ve sistem çıkışı da yaklaşık olarak  $50^\circ$ den  $360^\circ$ a değiştiğinden sistemin kazancı yaklaşık olarak  $K = 310/425 \approx 0.75$  bulunur. Sistem çıkışının en hızlı değiştiği yerdeki teğetin zaman eksenini kestiği nokta sistemin ölü zamanını verir ki yaklaşık olarak  $L \approx 0.3s$  bulunur. Ölü zamanın

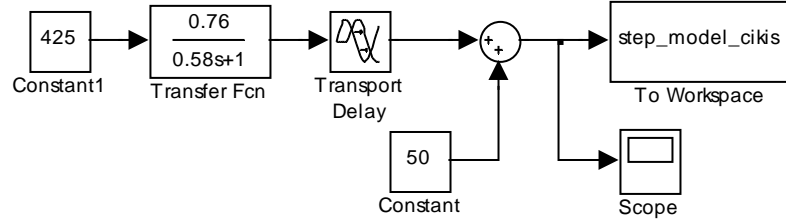
bitiminden, sistem çıkışının %63'lük değişime ulaştığı ana kadar olan zaman dilimi de sistemin zaman sabitini verir ki yaklaşık olarak  $T \approx 0.6s$  bulunur.



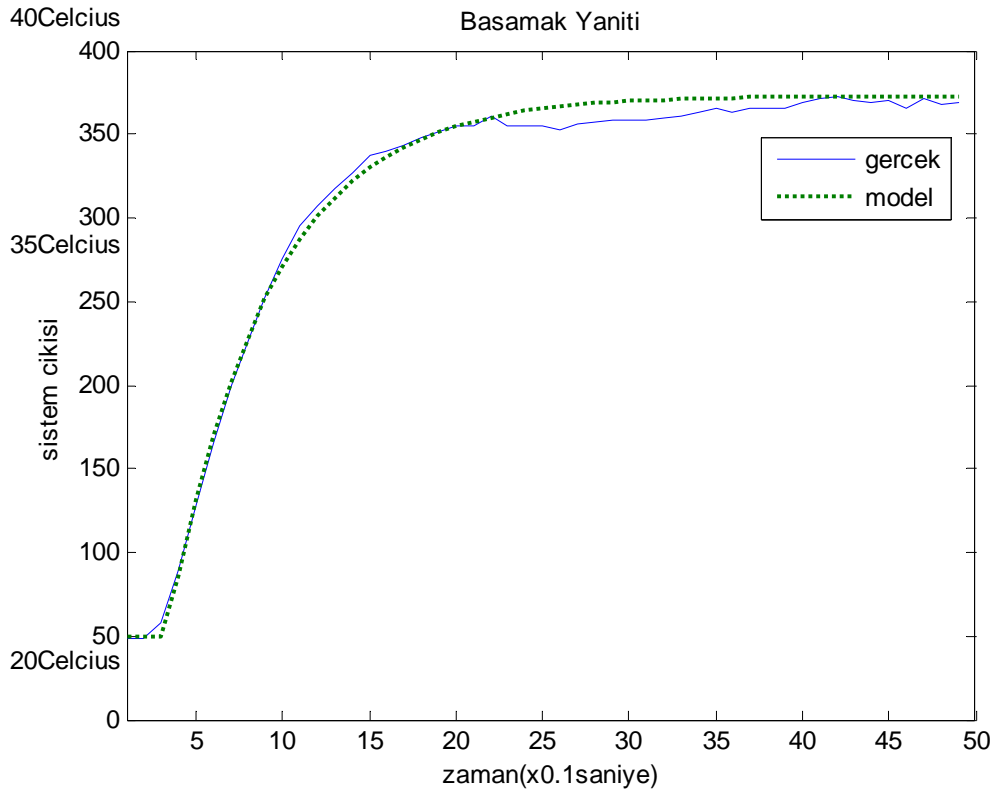
**Şekil 4.2:** Ziegler-Nichols Yöntemi ile PT326 Modelinin Elde Edilmesi

Bu basamak yanıtına bakılarak sıcaklık algılayıcısı en uzak konumda iken PT326 ısı sistemini model, birinci dereceden ölü zamanlı sistemler için Ziegler-Nichols yöntemi kullanılarak  $\frac{0.75}{0.6s+1}e^{-0.3s}$  olarak hesaplanmıştır. Hesabın doğruluğunun denemesi amacıyla da Şekil 4.3'teki MATLAB Simulink modeli oluşturularak model çıkışı ile gerçek sistem çıkışı Şekil 4.4'te karşılaştırılmıştır. Böylece, sistemin birinci dereceden ölü zamanlı yaklaşım modelinin  $\frac{0.76}{0.58s+1}e^{-0.3s}$  olduğuna karar verilmiştir.





**Şekil 4.3:** Basamak Yanıtı Simulink Modeli

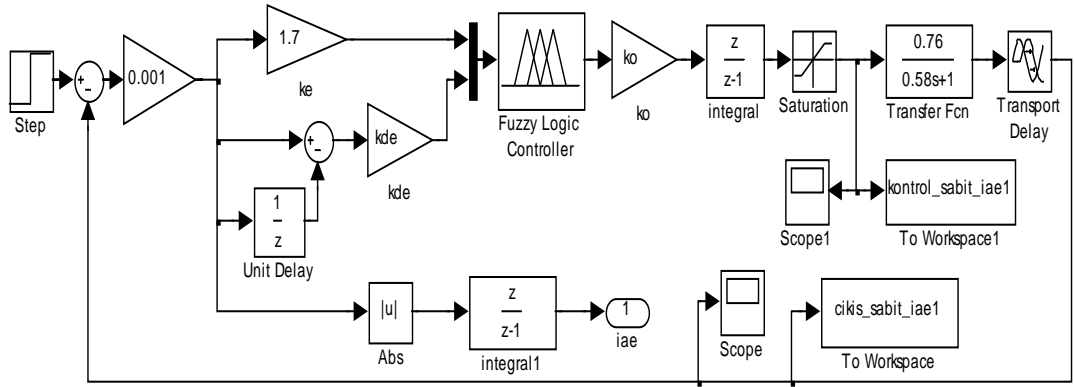


**Şekil 4.4:** Basamak Yanıtında Gerçek Çıkış ile Model Çıkışının Karşılaştırılması  
(Referans: 425, 40°C)

## 4.2 GATOOL (MATLAB Genetic Algorithm Toolbox) ile Ölçekleme Katsayılarının Bulunması

Bulanık kontrol sisteminde girişlerin ve çıkışın bulanık kontrolör tasarıma uygun bir şekilde ölçeklenmesi gerektiğinden bölüm başında bahsedilmişti. Kullanılan bulanık kontrolör yapısı gereği PI tipi bir bulanık kontrolördür. Bu ölçekleme katsayılarının P ve I etkilerinin nasıl olduğu [12]'de incelenmiştir. Dolayısıyla sistemi kararlı kılacak güzel bir kontrolör elde edebilmek için bu ölçekleme katsayılarının iyi bir

şekilde seçilmesi gerekmektedir. Katsayıların seçimi için MATLAB genetik algoritma aracı (gatool) kullanılmıştır. Hatanın değişimi ve çıkış ölçekleme katsayıları genetik algoritma aracı kullanılarak iki boyutlu küresel arama tekniği ile IAE (integral absolute error) küçük olacak şekilde aranmıştır. Hatanın değişimi ve çıkış ölçekleme katsayıları genetik algoritma aracında kullanılan IAE fonksiyonunun girişleri olmak üzere her giriş değeri için Şekil 4.5'teki Simulink modeli koşturulmuş ve yüz koşu sonunda IAE'yi en küçük yapan ölçekleme katsayıları  $kde=7.38$  ve  $ko=96$  olarak bulunmuştur. Daha önce de belirtildiği gibi hata ölçekleme katsayısı, 425/1000 basamak girişi bulanık kontrolörün hata için tanımlanan  $[-0.75 \ 0.75]$  aralığına uyacak şekilde  $ke=1.7$  sabit olarak seçilmiştir.



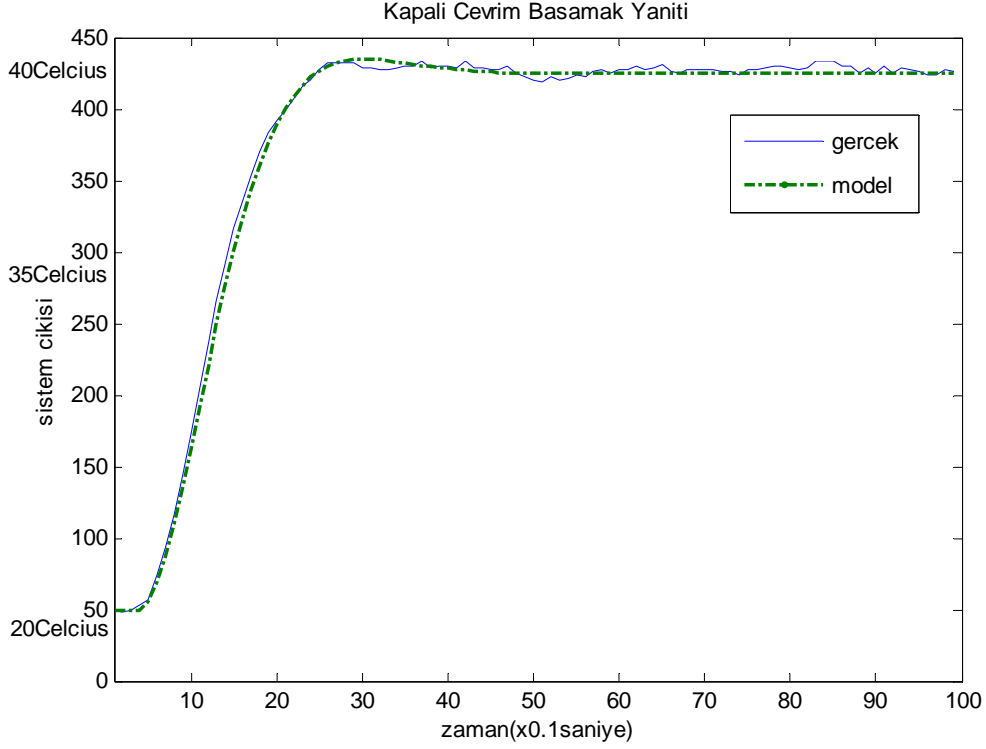
Şekil 4.5: Gatool ile Minimum IAE Elde Etmek için Kurulan Simulink Modeli

Arama sırasında kontrol işareti ve sistem çıkışı gözlenmiş, integratör çıkışındaki kontrol işaretinin doyuma girmemesine dikkat edilmiştir. Bulunan katsayılar kullanıldığında elde edilen sistem yanıtı bir sonraki kısımda Şekil 4.6'da verilmiştir.

### 4.3 Gerçek Zamanlı Çalışma ile Simulink Model Sonuçlarının Karşılaştırılması

PIC mikrokontrolör kartı olarak kullanılan PicLAB kartının üzerindeki, mikrokontrolörün analog giriş bacaklarına bağlanmış iki adet potansiyometre yardımıyla bir önceki kısımda genetik algoritma kullanılarak yapılan global arama sonucu bulunan  $kde$  ve  $ko$  ölçekleme katsayıları LCD'den de gözlenerek kolaylıkla mikrokontrolöre girilmiştir. Bu şekilde kontrol sistemi gerçek zamanlı olarak çalıştırılmış, elde edilen veriler UART ile PC'ye aktarılmıştır. Böylece Simulink ortamında elde edilen sonuçlar ile gerçek ortamda elde edilen sonuçlar karşılaştırılabildiği (Şekil 4.6). Şekilden de rahatlıkla anlaşılabilir gibi Simulink modeli ile gerçek sistemin kapalı çevrim basamak yanıtları çok yakındır. Bu

benzerlikten mikrokontrolör üzerinde gerçekleştirilen gerçek zamanlı bulanık kontrolörün MATLAB Simulink ortamındaki bulanık kontrolör ile aynı şekilde çalıştığı sonucuna varılabilmektedir.



**Şekil 4.6:** Gerçek Sistemin ve Simulink Modelin Kapalı Çevrim Basamak Yanıtları (Referans: 425, 40°C)

#### 4.4 MPC555 Geliştirme Kartı ile Yapılan Çalışmalar

İ.T.Ü. Fen Bilimleri Enstitüsü'ne sunulan “Gerçek Zamanlı Bulanık Kontrol Uygulaması” isimli proje kapsamında alınan “phyCORE-MPC555 Rapid Development Kit” malzemesi Şekil 4.7’de görülmektedir. Bu malzeme tek kart bilgisayar (single board computer) iç birimi ve bu birimin dışarıya bağlantısını sağlayan taşıyıcı karttan oluşur. Bu iç birim; Freescale’in 32-bit PowerPC MPC555 gömülü mikrokontrolörünü, mikrokontrolör içinde dahili 26KB SRAM, 448KB flaş bellek, 64-bit kayan nokta ünitesi, iki adet UART/SPI, iki adet CAN 2.0B, 10-bit çözünürlüklü 10µs çevrim süreli örnekle/tut girişli 16-kanallı çift ADC, 8 kanallı PWM birimi ve harici olarak 256KB RAM, 512KB Flaş, 4KB EEPROM bellek içerir. Adım adım kod icrasını sağlayan BDM debug (hata ayıklama) özelliği vardır.

Taşıyıcı kart ise iki adet RS-232 bağlantısı, iki adet CAN bağlantısı, BDM hata ayıklayıcının 25 pin paralel porta bağlantısı ile iç birimin dış dünyaya bağlanmasını sağlar.

MATLAB “Embedded Target for MOTOROLA-MPC555” aracı ile bu kartın Simulink ile kullanılmasına destek vermektedir. Bu kart için özel ADC, PWM gibi Simulink blokları vardır. Simulink blokları kullanılarak yapılan modelin kartta çalışmasını sağlayacak C kodları MATLAB Real Time Workshop ve MATLAB Real Time Embedded Coder ile otomatik olarak üretilip paralel port ya da seri porttan kartın RAM veya FLAŞ belleğine yüklenerek modelin gerçek zamanda çalışması sağlanmaktadır. Bunun için MATLAB ayrıca ‘Metrowerks CodeWarrior Development Studio for MPC5xx’ derleyicisine ihtiyaç duymaktadır. Alınan kart için PHYTEC 30 günlük deneme lisansı ile birlikte bu derleyici yazılımını sağlamaktadır. Kit ile ilgili detaylı bilgi [www.phytec.com](http://www.phytec.com) adresinden edinilebilir.

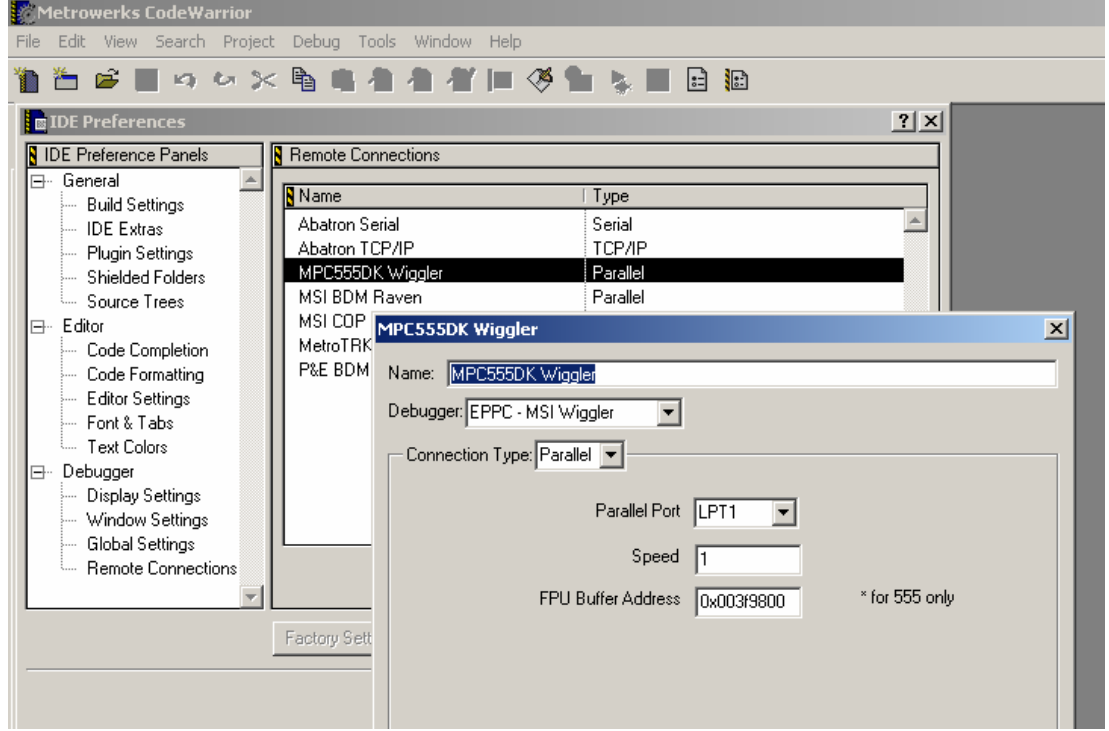


**Şekil 4.7:** phyCORE-MPC555 Rapid Development Kit

MATLAB yardım menüsünden “Embedded Target for MOTOROLA-MPC555” altında “Getting Started” altında “Setting Up and Verifying Your Installation” ve “Setting Target Preferences” alt başlıklarında anlatıldığı şekilde kart üzerindeki atlamalarla (jumper) ilgili donanımsal ve CodeWarrior’la ilgili yazılımsal ayarlar yapıldı. Ayarlarla ilgili izlenmesi gereken her adımı MATLAB yardımında bulmak mümkündür. Burada yalnızca önemli görülen ayarlar verilmiştir.

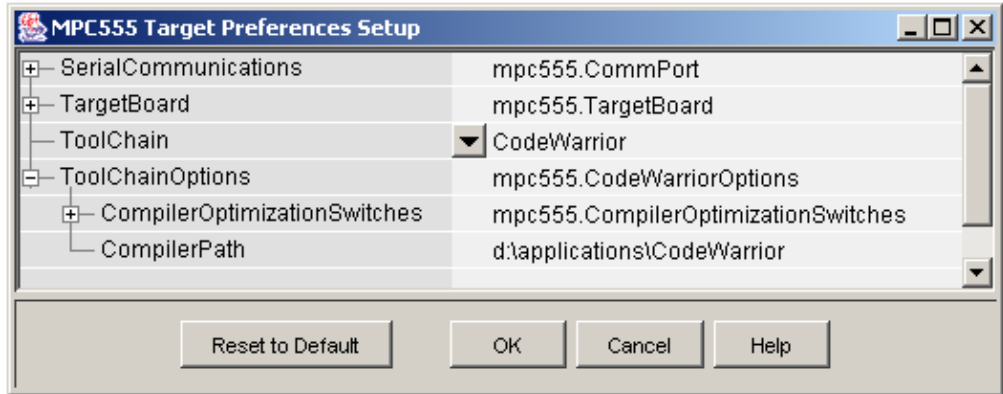
Metrowerks CodeWarrior IDE yazılımı CD’den yüklenip çalıştırıldıktan sonra, açılan IDE ortamında Edit → Preferences → Build Settings → ‘Build Before Running’ için ‘Never’ seçilmesi gerekmektedir.

CodeWarrior IDE ortamında Edit → IDE Preferences seçildiğinde açılan pencerede Debugger → Remote Connections seçildiğinde ‘MPC555DK Wiggler’ isimli bir bağlantı yapılandırılmalıdır, yoksa bu isimle yeni bir bağlantı eklenmelidir. Debugger ‘EPPC MSI Wiggler’ olmalıdır. Bağlantı tipi ‘Parallel’ olmalı, ‘Speed’ 1 olmalıdır, ‘FPU Buffer Address’ için de ‘0x3f9800’ girilmelidir (Şekil 4.8).



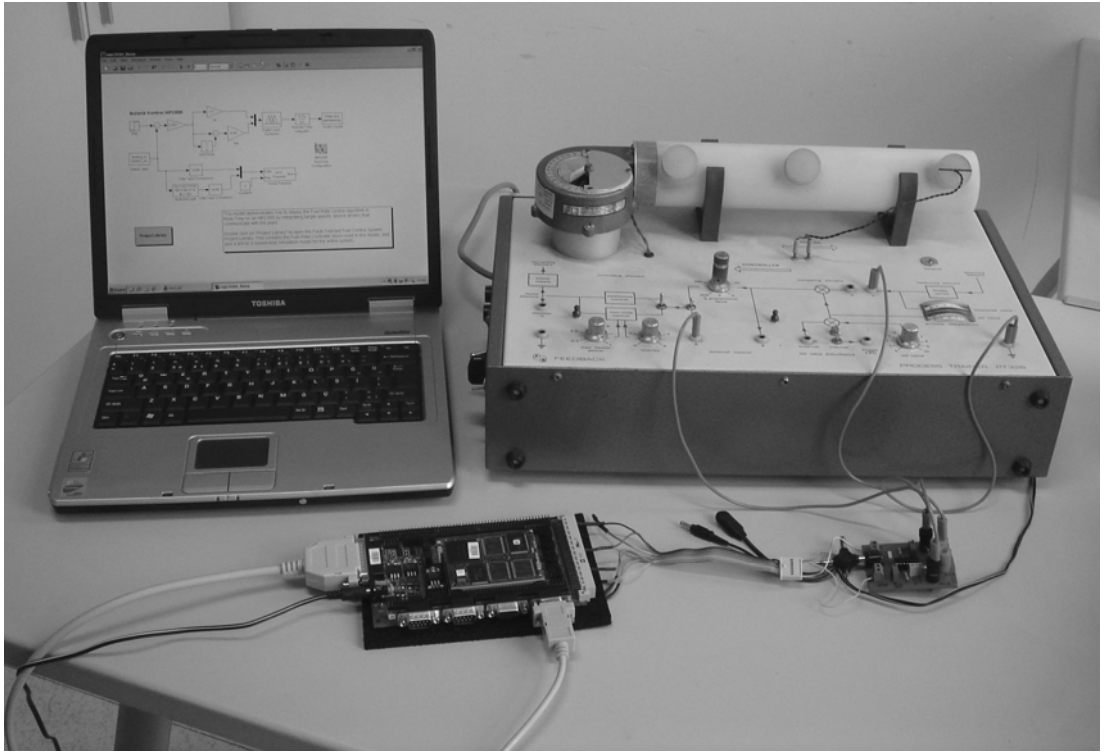
Şekil 4.8: Metrowerks CodeWarrior IDE Ayarları

MATLAB Start → Simulink → Embedded Target for Motorola MPC555 → Target Preferences adımı izlendiğinde ‘Toolchain’ menüsünden ‘CodeWarrior’ seçilmeli ve ‘ToolChainOptions’ altında ‘CompilerPath’ kısmına CodeWarrior derleyicisine ulaşmak için kullanılan yol girilmelidir (Şekil 4.9).



Şekil 4.9: MPC555 Hedef Ayarları

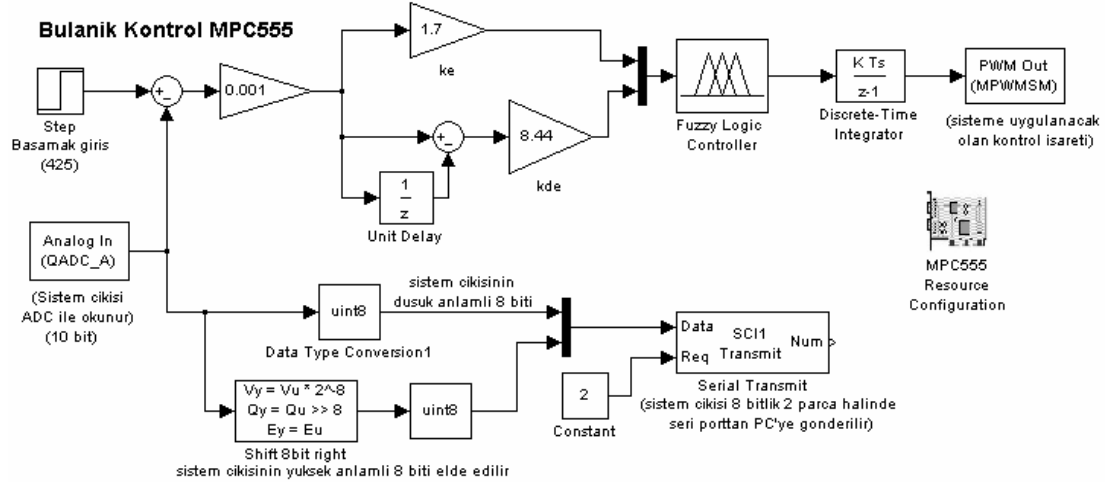
Daha önceki kısımlarda anlatılan PIC mikrokontrolörle gerçekleştirilen bulanık kontrol sistemi MPC555 geliştirme kartı ile gerçekleştirilmiştir. Bellek problemi olmadığından PIC mikrokontrolör için yapılan giriş üyelik fonksiyonlarının yalnızca üçgen ve trapezoid, çıkış üyelik fonksiyonlarının yalnızca Sugeno/tekli, durulama yönteminin yalnızca centroid olması kısıtlamaları yoktur. PT326 deney seti termistör çıkışı (0-10V) ve kontrol işareti girişi (0-10V) yine 0-5V↔0-10V gerilim kuvvetlendirici kartı ile MPC555 kartının A\_AD0 analog giriş ve MPWM0 çıkış bacaklarına bağlanmıştır. Bacak bağlantıları Development Board for phyCORE MPC555 Hardware Manual'den öğrenilebilir. Kartın BDM paralel portu ve SC11 (RS-232 1) seri portu PC'ye bağlanmıştır. Çalışma ortamı Şekil 4.10'da gösterilmiştir.



**Şekil 4.10:** MPC555 Kartı ile Çalışma Ortamı

MATLAB ile Şekil 4.11'deki Bulanık Kontrol MPC555 Simulink Modeli kurulmuştur. Hatanın değişimi ve çıkış katsayıları olarak ITAE kriterine göre Genetik Algoritma Aracı ile bulunan  $k_e=8.44$  ve  $k_o=1.78$  değerleri kullanılmıştır. Çıkış ölçekleme çarpanı ayrık integratör bloğunun kazancı olarak girilmiştir. İntegratörün örnekleme zamanı  $T_s=0.1$  alınmış, ayrıca integratör çıkışı [0 1] aralığında sınırlandırılmıştır çünkü modelde kullanılan PWM bloğunun girişi [0 1] aralığındaki darbe periyot oranıdır. Sistem çıkışı ADC bloğu ile okunup, kontrol işareti PWM bloğu ile gerçek sisteme uygulanmaktadır.

Ayrıca, sistem çıkışı bilgisi, kart üzerindeki birinci seri port üzerinden, PC’de çalışan MATLAB seri port dinleme programı ile MATLAB çalışma ortamına dizi olarak aktarılmaktadır.



**Şekil 4.11:** Bulanık Kontrol MPC555 Simulink Modeli

Şekil 4.11’deki modelin MPC555 kartı üzerinde gerçek zamanlı olarak çalıştırılması için aşağıdaki adımlar izlenmelidir.

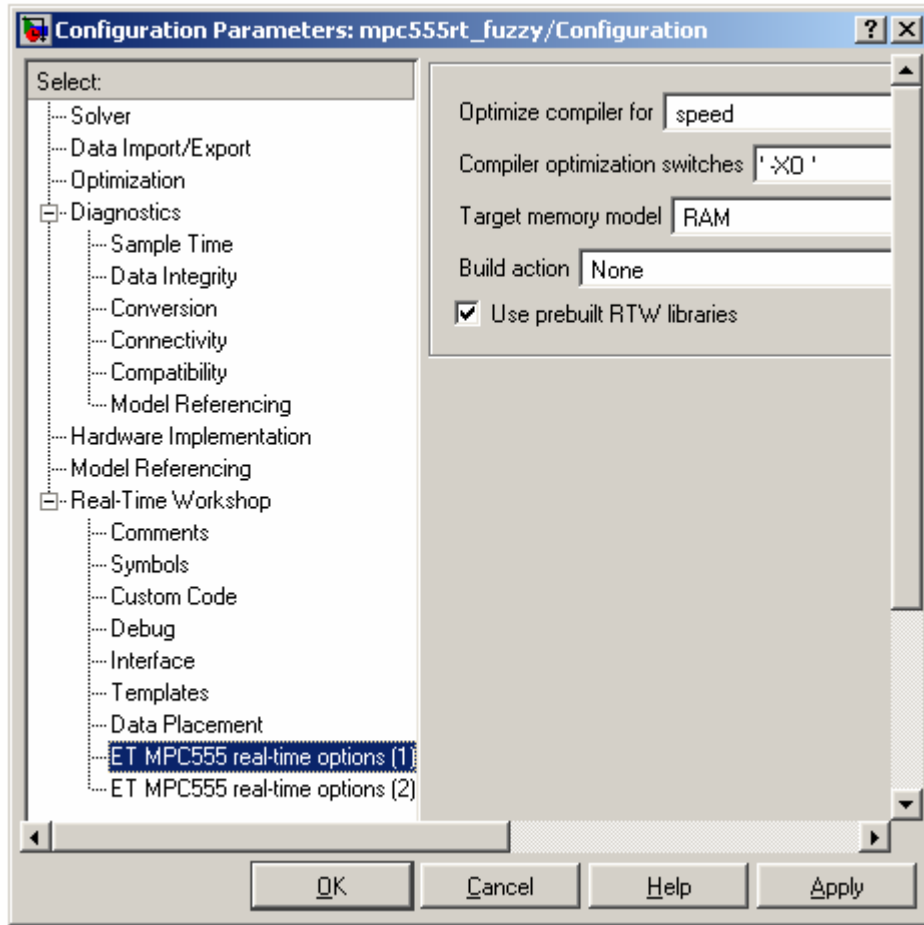
Simulation → Configuration Parameters ayarlarından ‘Solver Type’ ‘Fixed-step’ seçilip ‘Fixed-step size’ 0.1 saniye olarak girilmiştir, ‘Stop time’ ise sonsuz çalışma anlamında ‘inf’ girilmiştir. Bu değerler benzetim için olduğu gibi modelin karta yüklenecek kodu için de geçerlidir.

Simulation → Configuration Parameters ayarlarından ‘Real-Time Workshop’ için ‘RTW target file’ olarak ‘mpe555rt.tlc’ seçilip onaylandıktan sonra sol tarafta ‘Real-Time Workshop’ altında ‘ET MPC555 real-time options(1)’ ayar seçeneği çıkar. Şekil 4.12’de görülen bu pencereden model için oluşturulacak olan MPC555 mikrokontrolör kodunun RAM ya da FLASH belleğe yüklenmek üzere üretilmesi seçilmelidir. Yapılan değişiklikler onaylanarak modele dönülür.

Tools → Real Time Workshop → Build Model adımı izlendiğinde MATLAB çalışma klasörü altında ‘modelin ismi\_mpe555rt’ isiminde bir klasör yaratılarak modelin kartta icrası için gerekli C dosyaları otomatik olarak bu klasör altında oluşturulacaktır. Üretilen makine kodunun RAM ya da FLASH belleğe yüklenecek olmasına göre oluşturulacak olan yükleme dosyası farklı olacaktır.

### Üretilen Model Kodunun RAM Bellekte Çalıştırılması:

Eğer 'Configuration Parameters' altında 'Real-Time Workshop' için 'RTW target file' 'mpc555rt.tlc' seçildiğinde ortaya çıkan 'ET MPC555 real-time options(1)' için 'Target memory model' RAM seçilmişse MATLAB çalışma klasörünün içinde 'modelin ismi.ram.elf' dosyası üretilmiştir. Bu dosya MATLAB Start → Simulink → Embedded Target for Motorola MPC555 → Debug RAM Based Application (via BDM) seçeneği ile kartın RAM belleğine yüklenir (Şekil 4.13). Yükleme işlemi tamamlandığında model kodu kart üzerinde gerçek zamanlı olarak çalışmaya başlar.



Şekil 4.12: Model Kodunun RAM ya da FLASH Bellek için Üretilmesi Seçimi

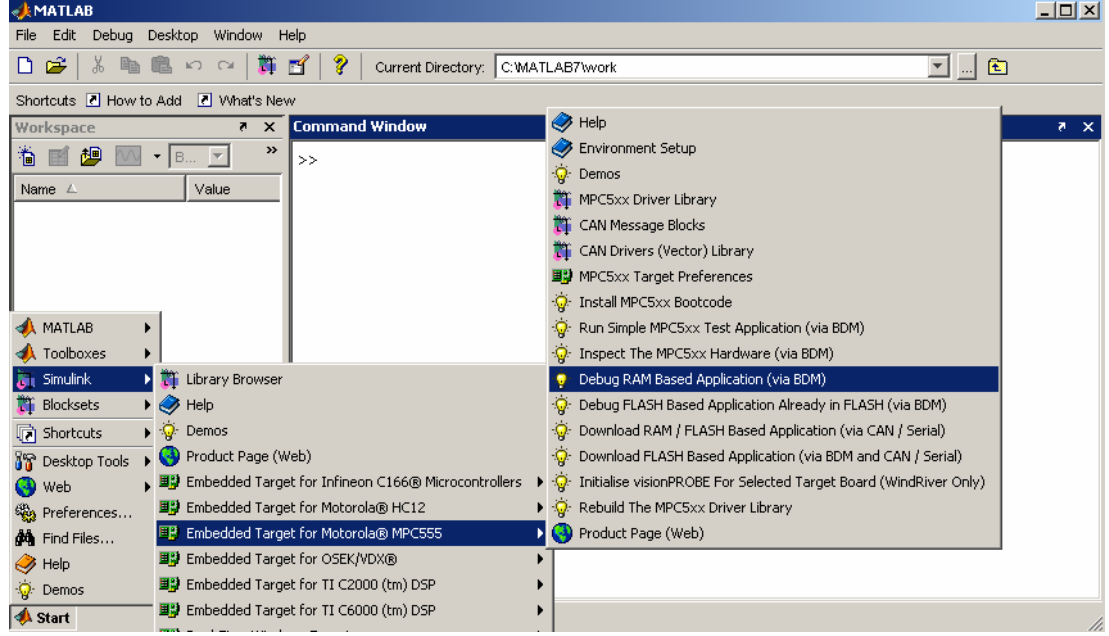
### Üretilen Model Kodunun Flaş Bellekte Çalıştırılması:

FLASH belleğe yükleme yapılabilmesi için (eğer daha önceden yapılmamışsa) öncelikle Start→Simulink→Embedded Target for Motorola MPC555→Install MPC5xx Bootcode adımı ile 'bootcode' yüklemesinin yapılması gerekmektedir.

Eğer 'Configuration Parameters' altında 'Real-Time Workshop' için 'RTW target file' 'mpc555rt.tlc' seçildiğinde ortaya çıkan 'ET MPC555 real-time options(1)' için



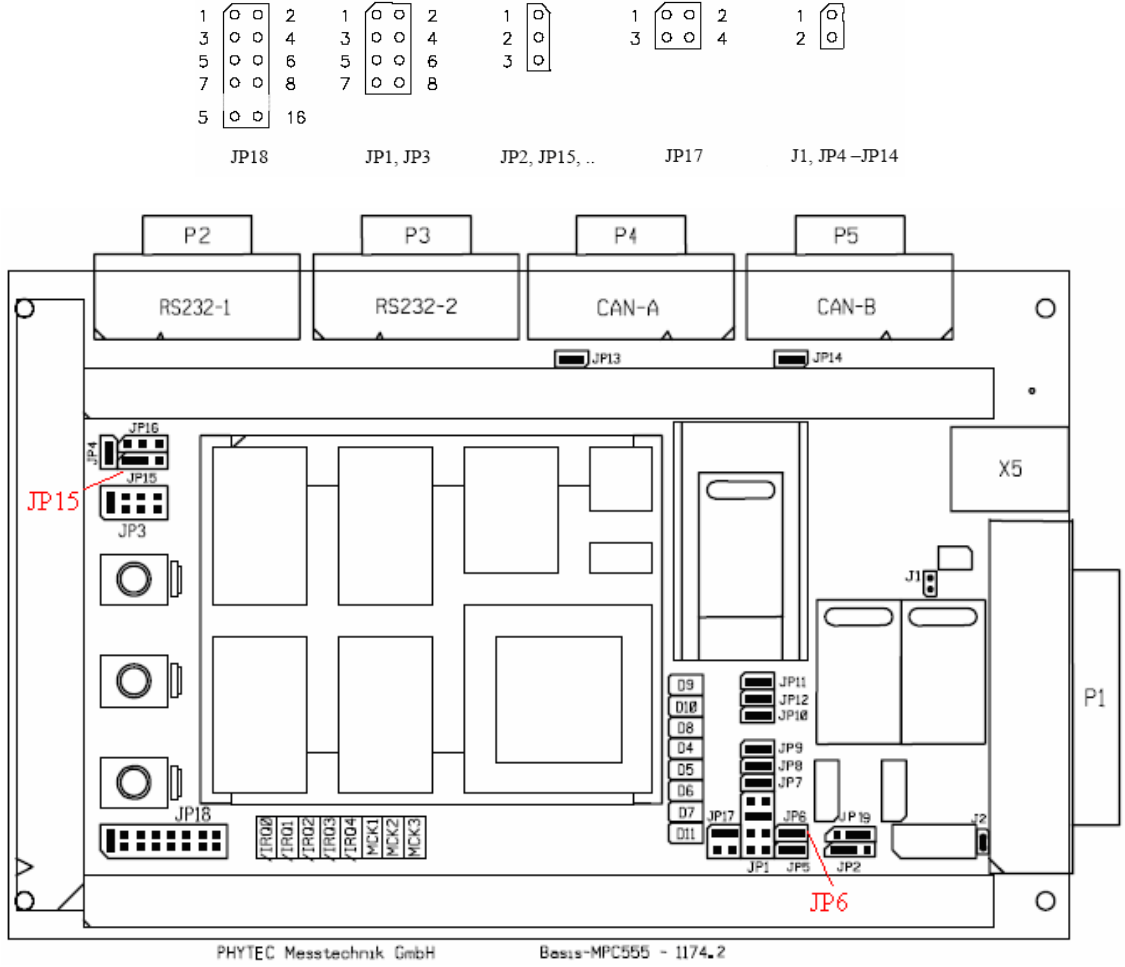
'Target memory model' FLASH seçilmişse MATLAB çalışma klasörünün içinde 'modelin ismi\_flash.s19' dosyası üretilmiştir. Bu dosya Start → Simulink → Embedded Target for Motorola MPC555 → Download FLASH Based Application (via BDM and Serial) seçeneği ile kartın FLASH belleğine yüklenir.



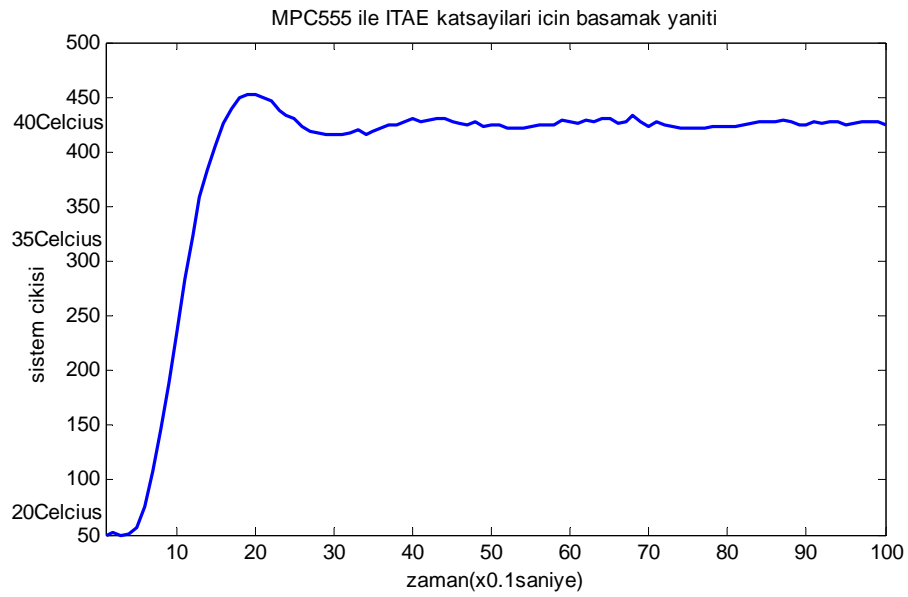
Şekil 4.13: Model Kodunun Karta Yüklenmesi

Kodun RAM ya da FLASH belleğe yüklenmesi sırasında kart üzerindeki atlama (jumper) ayarlarına dikkat etmek gerekir. Kart üzerindeki varsayılan atlamalar Şekil 4.14'te gösterilmiştir. RAM belleğe yükleme sırasında atlamalar varsayıldığı gibidir. Dahili FLASH belleğe yükleme yapmak için J15 atlamasının Şekil 4.14'te görüldüğü gibi 1+2 konumunda olması, harici FLASH belleğe yükleme yapmak için J15 atlamasının bir sağa kaydırılması, yani 2+3 konumunda olması gerekmektedir. FLASH belleğe kod yüklendikten sonra kodun çalışması için güç bağlantısı kesildikten sonra J6 atlaması çıkarılmalı ve paralel port bağlantı kablosu çekilmelidir. Daha sonra güç bağlantısı yeniden yapıldığında FLASH belleğe yüklenen kod koşmaya başlayacaktır.

Kodun koşması ile birlikte, MATLAB ortamında çalışan seri port dinleme programı ile elde edilen sistem yanıtı Şekil 4.15'te verilmiştir. ITAE katsayıları ile elde edilen sistem yanıtı beklendiği gibi IAE katsayılarına göre daha hızlı ancak daha büyük aşımlıdır. Burada asıl başarı kontrolör performansının en iyi olması değil bulanık kontrol sisteminin gerçek zamanlı olarak çalıştırılabilmiş olması ve böylece kontrolörün gerçek sistem üzerindeki başarımının gözlenebilmiş olmasıdır.



Şekil 4.14: MPC555 Kartı Üzerindeki Varsayılan Atlamalar ve Bacak Numaraları



Şekil 4.15: MPC555 Kartı ile Gerçek Zamanlı Bulanık Kontrol Sistemi için Basamak Yanıtı (Referans: 425, 40°C)

## 5. SONUÇ

Bu tez çalışmasında, öncelikle endüstride yaygın olarak kullanılan düşük maliyetli bir 8-bit PIC18F452 mikrokontrolör ile MATLAB Simulink ortamında tasarlanmış olan bir bulanık kontrolör sisteminin gerçek sistem üzerinde gerçek zamanlı olarak çalışması gerçekleştirilmiştir. Bunun için MATLAB ile birlikte verilen bulanık çıkarım mekanizmasını sağlayan 'fis.c' ve 'fismain.c' kaynak kodları düzenlenerek MATLAB 'Fuzzy Logic Toolbox' arayüzü ile kolayca oluşturulan '\*.fis' dosyasındaki parametreleri çekip PICC diline çeviren bir C programı yazılmıştır. Bütün kontrol çevrimini sağlayacak diğer C kodları da PICC dilinde yazılıp PIC mikrokontrolöre yüklenerek bulanık kontrolörün gerçek zamanlı çalışması sağlanmıştır. Çalışma esnasında sistem çıkışı, kontrol işareti gibi veriler seri port üzerinden MATLAB'a aktarılmıştır. Böylece MATLAB ortamında kontrolörün çalışma performansı gözlemlenmiştir.

Bu şekilde PIC mikrokontrolör ile gerçekleştirilen sistemin düşük maliyetli olması ve endüstride yaygın olarak kullanılması avantajları vardır. Ancak, bellek sınırlaması vardır. Bu nedenle bulanık kontrolör giriş üyelik fonksiyonlarının yalnızca üçgen ve trapezoid, çıkış üyelik fonksiyonlarının yalnızca Sugeno/tekli, durulamanın yalnızca centroid olma kısıtlamaları getirilmiştir. Daha da önemlisi MATLAB ortamında tasarlanan kontrol sisteminin mikrokontrolör üzerinde çalıştırılabilmesi için MATLAB yazılımına ek olarak tasarımcının PIC mikrokontrolör ve C ile programlama bilgisinin olması gerekmektedir. Bu durum düşünüldüğünde MATLAB Simulink ortamında bloklarla oluşturulan bir kontrol sisteminin doğrudan bir karta yüklenip, kart üzerinde gerçek zamanlı olarak çalıştırılabilmesi, maliyeti PIC mikrokontrolör sistemine göre daha yüksek olsa da çok daha kolay bir süreç olacaktır.

Bu amaçla, İ.T.Ü. Fen Bilimleri Enstitüsü proje desteği ile 32-bit MOTOROLA işlemcili ve 64-bit kayan nokta birimi olan 'phyCORE MPC555 Rapid Development Kit' geliştirme kartı alınmıştır. MATLAB Simulink ile oluşturulan model yine MATLAB ile direk olarak karta yüklenebilmektedir. Böylece kullanıcının mikrokontrolör mimarisi veya C programlama hakkında derin bilgisinin olması şartı kaldırılmıştır.

Hem PIC mikrokontrolör kartı hem de MOTOROLA MPC555 kartı ile yapılan gerçek zamanlı çalışma sonuçları, Simulink benzetim model sonucu ile karşılaştırılarak verilmiştir. Örnek sistem olarak lineer olmayan ölü zamanlı PT326 ısıtıcı deney seti kullanılmıştır. Sistem, bulanık kontrolör ile gerçek zamanlı olarak kontrol edilmiştir.

Bu haliyle hem PIC mikrokontrolör hem de MPC555 kartı üzerinde çalışan gerçek zamanlı kontrol sistemine MATLAB ortamından müdahale edilememektedir. Veri aktarımı karttan MATLAB ortamına olmak üzere tek yönlüdür. Gelecek çalışma olarak, kontrol algoritması kart üzerinde koşarken MATLAB ortamından komut ve parametrelerin seri port üzerinden karta aktarılması ile kontrol algoritmasındaki bazı parametrelerin hızlı ve kolay bir şekilde MATLAB ortamından değiştirilmesi ile gerçek zamanlı kontrolör geliştirme sürecinin hızlandırılması ve kolaylaştırılmasının sağlanabileceği düşünülmektedir.

## KAYNAKLAR

- [1] **Teng, F.C.**, 2000. Real time control using MATLAB simulink, *Systems, Man, and Cybernetics International Conference Proceedings of IEEE*, **4**, October, 3039-3044.
- [2] **Karasakal, O., Yeşil, E., Güzelkaya, M. and Eksin, İ.**, 2004. A PLC implementation of fuzzy PID-type controller with a function tuner based self-tuning, *Journal of Naval Science and Engineering*, **2(1)**, 41-50.
- [3] **Taneva, A., Muskinja, N., Petrov, M. and Tovornik, B.**, FPID controller: real time application, *Intelligent Systems 2<sup>nd</sup> International Conference Proceedings of IEEE*, **3**, June 2004, 39-42.
- [4] **Yingkai, Z. and Yiling, W.**, 2002. Intelligence control of nonlinear systems based on MATLAB simulation and the real-time control platform, *4<sup>th</sup> World Congress on Intelligent Control and Automation Proceedings of IEEE*, **1**, June, 786-789.
- [5] **Butt, C.B., Hoque, M.A. and Rahman, M.A.**, Simplified fuzzy-logic-based MTPA speed control of IPMSM drive, *IEEE 38<sup>th</sup> IAS Annual Meeting on Industry Applications Conference Record*, **1**, October 2003, 499-506.
- [6] **Sivakumaran, T.S., Natarajan, S.P. and Venkatachalam, P.**, 2005. Simulation and real-time implementation of fuzzy logic controller for parallel loaded resonant converter, *INDICON Annual of IEEE*, December, 127-132.
- [7] **Sibigtroth, J.M.**, 1996. Fuzzy extends a general purpose instruction set, *Northcon96*, Seattle WA USA, November, 216-220.
- [8] **Jackson, A.**, 1997. A new microcontroller with fuzzy inference instructions simplifies controller designs, *IEEE Aerospace Conference Proceeding Volume 3*, Snowmass at Aspen CO USA, February, 491-503.

- [9] **Singh, P., Vinjamuri, R., Wang, X. and Reisner, D.**, 2005. Design and implementation of a fuzzy logic-based state-of-charge meter for Li-ion batteries used in portable defibrillators, *Power Sources Journal of Elsevier*.
- [10] **White, A.S. and Sinclair, R.**, 2004. Quantitative validation techniques a database. (I). Simple examples, *Simulation Modeling Practice and Theory*, **12**, 451-473.
- [11] **Ng, G.W. ve Cook, P.A.**, 1998. Real-time control of systems with unknown and varying time-delays, using neural Networks, *Engineering Applications of Artificial Intelligence*, **11**, 401-409.
- [12] **Qiao, W.Z. and Mizumoto, M.**, 1996. PID type fuzzy controller and parameters adaptive method, *Fuzzy Sets and Systems*, **78**, 23-35.
- [13] **Genç, H.M.**, 2006. A new rule-base modification scheme for the time delay systems, *Ms.C. Thesis*, I.T.U. Institute of Science and Technology, Istanbul.
- [14] **Genç, H.M., Eksin, İ., Güzelkaya, M. and Yeşil E.**, 2006. A rule-base modification for time delay systems, *Artificial Intelligence and Soft Computing Proceeding*, Palma De Mallorca, Spain, August 2006, 41-47.

## ÖZGEÇMİŞ

Özgür Aydın TEKİN 1981 yılında Mersin’de doğdu. 1999 yılında İçel Mersin Anadolu Lisesi’nden mezun oldu. 2004 yılında İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği ve Bilgisayar Mühendisliği diplomalarını alarak İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Kontrol ve Otomasyon Mühendisliği yüksek lisans programına başladı. Yaklaşık bir yıl ARÇELİK A.Ş. Araştırma-Geliştirme bölümünde görev yaptı. Şubat 2005’ten itibaren İstanbul Teknik Üniversitesi Kontrol ve Kumanda Sistemleri Ana Bilim Dalı’nda Araştırma Görevlisi olarak çalışmaktadır.