

**T.C.  
SÜLEYMAN DEMİREL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DONANIMSAL BULANIK MANTIK MODELLEME SİMÜLATÖRÜ**

**Hüseyin ŞENGÜN**

**Danışman  
Prof. Dr. Tuncay AYDOĞAN**

**YÜKSEK LİSANS TEZİ  
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI  
ISPARTA - 2018**



©2018 [Hüseyin ŞENGÜN]

## TEZ ONAYI

Hüseyin ŞENGÜN tarafından hazırlanan "**Donanımsal Bulanık Mantık Modelleme Simülatörü**" adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü **Elektronik-Bilgisayar Eğitimi Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak başarı ile savunulmuştur.

Danışman **Prof. Dr. Tuncay AYDOĞAN**  
Süleyman Demirel Üniversitesi



Jüri Üyesi **Doç. Dr. Okan BİNGÖL**  
Süleyman Demirel Üniversitesi



Jüri Üyesi **Dr. Öğr. Üyesi Ali KAVURUR**  
Mehmet Akif Ersoy Üniversitesi



Enstitü Müdürü **Prof. Dr. Yasin TUNCER**

## **TAAHHÜTNAME**

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

**Hüseyin ŞENGÜN**



## İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	i
ÖZET .....	ii
ABSTRACT .....	iii
TEŞEKKÜR.....	iv
ŞEKİLLER DİZİNİ .....	v
ÇİZELGELER DİZİNİ.....	vi
KISALTMALAR.....	vii
1. GİRİŞ.....	1
2. KAYNAK ÖZETLERİ.....	4
3. MATERYAL VE YÖNTEM.....	7
3.1. Yapay Zeka .....	7
3.1.1. Yapay sinir ağları.....	7
3.1.2. Uzman sistemler .....	8
3.1.3. Bulanık mantık.....	8
3.1.3.1. Belirsizlik kavramı.....	9
3.1.3.2. Bulanık mantık ve küme teorisi .....	10
3.1.3.3. Bulanık kontrol sistemleri.....	12
3.1.3.4. Bulanık mantık uygulama alanları .....	13
3.1.3.5. Bulanık mantık modelleme araçları .....	14
3.1.3.5.1. Matlab bulanık mantık model geliştirme ortamı.....	14
3.1.3.5.2. FuzzyTech programı.....	16
3.1.3.5.3. Labview programı .....	18
3.2. Tasarlanan Simülatörün Donanım Bileşenleri.....	19
3.2.1. Arduino gömülü sistem kartı .....	19
3.2.2. Potansiyometre.....	20
3.2.3. Nextion HMI TFT LCD ekran.....	21
3.2.4. Toggle switch .....	25
3.3. Simülatör .....	26
4. ARAŞTIRMA BULGULARI .....	28
4.1. Geliştirilen Donanımsal Bulanık Mantık Modelleme Simülatör Tasarımı .....	28
4.1.1. Arduino UNO'ya bulanık mantık kontrol modelini yükleme .....	30
4.1.2. Örnek uygulama ile sistemin çalışması.....	34
5. SONUÇ .....	45
6. KAYNAKLAR .....	47
EKLER.....	50
ÖZGEÇMİŞ.....	56

## ÖZET

Yüksek Lisans Tezi

### DONANIMSAL BULANIK MANTIK MODELLEME SİMÜLATÖRÜ

Hüseyin ŞENGÜN

Süleyman Demirel Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektronik Bilgisayar Eğitimi Anabilim Dalı

Danışman: Prof. Dr. Tuncay AYDOĞAN

Bulanık mantık, günümüzde hemen hemen her alandaki kontrol sistemlerini modellemek ve tasarlamak için yaygın olarak kullanılan bir yapay zeka algoritmasıdır. Bulanık mantık algoritması özetle; bulanık (belirsizlik) özellikteki durum ve kontrol değişkenlerinin önce bulanık üyelik fonksiyonları oluşturulur, daha sonra bu parametreler arasındaki bulanık ilişki kuralları tanımlanır. Bulanık mantık çıkarım algoritması da bulanık kuralları dikkate alarak, bulanık parametrelerin ağırlıklarını hesaplayarak sonuçlar üretir. Problemleri bulanık mantık ile modellemek için herhangi bir programlama dili ile kodlama yapılabileceği gibi günümüzde genellikle geliştirilmiş olan birçok bulanık mantık modelleme paket yazılımları kullanılmaktadır. Ancak bulanık mantık kontrol sistemleri modellerinin donanımsal tasarımlarını hazırlamadan önce doğrulanacağı ve test edileceği genel amaçlı, esnek yapıda donanım simülatörleri bulunmamaktadır.

Bu tez çalışmasında, herhangi bir yazılım modelleme aracında modellenen bir bulanık kontrol sistemi modelinin donanım üzerindeki çalışma başarımının izlenebileceği, doğrulama ve testinin yapılabileceği en çok 5 giriş ve 5 çıkışa sahip donanımsal bir simülatör tasarımı ve uygulaması yapılması amaçlanmıştır. Bu simülatör sayesinde giriş değerlerine göre sistemin çıkış değerleri görüntülenebilecek, uygulamaya geçmeden önce donanımsal simülatör ile bulanık kontrol sistemi modelinin tutarlılığı test edilmiş olacaktır. Tasarlanan Donanımsal Bulanık Mantık Modelleme Simülatörü, yazılımın yükleneceği Arduino UNO, giriş değerleri için kullanılacak Potansiyometreler ve giriş/çıkış değerlerini gösterecek Nextion 3.5" HMI TFT LCD akıllı dokunmatik ekran kullanılmıştır.

**Anahtar Kelimeler:** Bulanık Mantık, Arduino, Nextion HMI TFT LCD, Simülatör.

**2018, 56 Sayfa**

## **ABSTRACT**

**M. Sc. Thesis**

### **FUZZY LOGIC MODELING SIMULATOR ON HARDWARE**

**Hüseyin ŞENGÜN**

**Süleyman Demirel University  
Graduate School of Natural and Applied Sciences  
Department of Electronic and Computer Education**

**Supervisor: Prof. Dr. Tuncay AYDOĞAN**

Fuzzy logic is a widely used artificial intelligence algorithm for modeling and designing control systems in almost all areas at the present time. To summarize, the fuzzy logic algorithm is formed by firstly creating the fuzzy associative functions of fuzzy status and control variables and then the fuzzy relation rules are defined between these parameters. Fuzzy Logic Extraction Algorithm measures the weights of the parameters and finds results in accordance with fuzzy rules. To model problems with fuzzy logic, it is possible to code with any programming language, or many fuzzy logic modeling software packages which are developed recent are used. However, there is no general-purpose, flexible hardware simulator that can be verified and tested before preparing the hardware designs of the fuzzy logic control systems models.

In this thesis study, it is aimed to design and implement a hardware simulator with maximum 5 inputs and 5 outputs that can be used to monitor the performance of the fuzzy control system modeled by any software modeling device on the hardware. Thanks to this simulator, according to input values, output values of the system can be displayed and the coherence of the fuzzy control system model with the hardware simulator will be tested before implementation. The designed hardware fuzzy logic modelling simulator, the software which loaded Arduino UNO, the potentiometers to be used for input values and Nextion 3.5 inch HMI TFT LCD smart touch screen that display input/output values were used.

**Key Words:** Fuzzy Logic, Arduino, Nextion HMI TFT LCD, Simulator.

**2018, 56 Pages**

## **TEŐEKKÜR**

Bu alıőmamda her tűrlű desteęi ve tecrűbesiyle yanımda olan kıymetli danıőman hocam Prof. Dr. Tuncay AYDOęAN'a teőekkűrlerimi sunarım.

Akademik hayatta baőarılı olabilmemi ve bilgileriyle bugűnlere gelmemi saęlayan tűm deęerli bűlűm hocalarıma teőekkűr ederim.

Hayatımın her aőamasında yanımda olan anneme, babama ve alıőmamda desteęini esirgemeyen eőime sevgi ve saygılarımı sunarım.

Hűseyin ŐENGŪN  
ISPARTA, 2018



## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 3.1. Klasik küme ve bulanık küme üyelik derecesi işlevleri .....	10
Şekil 3.2. Üyelik fonksiyonu .....	11
Şekil 3.3. Bulanık mantık karar verme sistemi.....	13
Şekil 3.4. Matlab bulanık mantık blok yapısı.....	15
Şekil 3.5. FIS Editör gösterimi .....	16
Şekil 3.6. FuzzyTech Gösterimi .....	17
Şekil 3.7. LabVIEW bulanık mantık araçları .....	18
Şekil 3.8. Arduino UNO R3 ve Arduino MEGA 2560 genel görünümü .....	20
Şekil 3.9. Potansiyometre Genel Görünümü.....	20
Şekil 3.10. 3.5 Inch Nextion HMI TFT LCD ekran (NX4832T035) .....	22
Şekil 3.11. Nextion HMI TFT LCD ekran seri port bağlantı noktaları .....	22
Şekil 3.12. Nextion editör ile ara yüz tasarım ekranı.....	23
Şekil 3.13. Nextion editor ile ara yüz tasarımının ana ekranı .....	24
Şekil 3.14. USB-TTL Converter.....	25
Şekil 3.15. 6'lı Toggle switch.....	26
Şekil 4.1. Simülatör prototipinin genel görünümü .....	28
Şekil 4.2. Donanımsal bulanık mantık modelleme simülatörü devre şeması.....	29
Şekil 4.3. Matlab fuzzy inference system to arduino c converter ekranı.....	30
Şekil 4.4. 6'lı toggle switch 1. ve 2. konum bağlantı gösterimi .....	34
Şekil 4.5. Sistemin giriş ve çıkış parametreleri .....	35
Şekil 4.6. Tasarımın giriş parametre değerleri.....	35
Şekil 4.7. Tasarımın çıkış parametre değerleri .....	36
Şekil 4.8. Sistemin matlab üyelik fonksiyon editörü görünümü .....	38
Şekil 4.9. Birinci örneğin giriş parametrelerine göre çıkış parametrelerinin kural editöründe gösterimi .....	38
Şekil 4.10. Nextion ekranda birinci örneğin gösterimi .....	39
Şekil 4.11. İkinci örneğin giriş parametrelerine göre çıkış parametrelerinin kural editöründe gösterimi.....	39
Şekil 4.12. Nextion ekranda ikinci örneğin gösterimi .....	40

## ÇİZELGELER DİZİNİ

	<b>Sayfa</b>
Çizelge 3.1. Arduino platformlarının teknik özelliklerinin gösterimi.....	20
Çizelge 4.1. Bulanık çıkarım sisteminin kural tablosu .....	37



## **KISALTMALAR**

FIS	Fuzzy Inference System
FLC	Fuzzy Logic Controller
FTL	Fuzzy Technology Language
GUI	Graphical User Interface
HMI	Human-Machine Interface
IDE	Integrated Development Environment
LCD	Liquid Crystal Display
PIC	Peripheral Interface Controller
PID	Proportional Integral Derivative
TFT	Thin Film Transistor
TTL	Transistor to Transistor Logic
USB	Universal Serial Bus



## 1. GİRİŞ

Bulanık mantık (Fuzzy Logic) kavramı ilk kez Prof. Lutfi A. Zadeh tarafından 1965 yılında California Berkeley Üniversitesinde yapmış olduğu çalışmalarla duyuldu.

Bulanık Mantık insanların her gün kullandığı ve davranışlarının yorumlandığı yapıya ulaşılmasını sağlayan matematiksel bir disiplindir. İnsanlar günlük hayatta; tam olarak tanımlanmamış ve nümerik olmayan dilsel niteleyiciler (soğuk, biraz soğuk, ılık, sıcak, çok sıcak vb. gibi) kullanarak kararlar verir ve problemlerini çözerler (Dereli, 2000).

Bir bulanık küme, çalışma yapılan alana ait her bir elemana matematiksel olarak kümedeki üyelik derecesini temsil eden bir değer atayarak tanımlanır. Bu değer, elemanın bulanık küme tarafından ifade edilen kavrama üyelik derecesini ifade eder. Bundan dolayı bireylerin kümeye ait olması farklıdır. Üyelik dereceleri 0 ile 1 arasındaki gerçel sayılarla temsil edilirler. Tam üye olma ve üye olmama durumu, bulanık kümede sırasıyla 1 ve 0 değerleriyle karşılanır. Bundan dolayı da, klasik küme kavramı bulanık küme kavramının bu iki değere kısıtlanmış özel bir şekli olarak görülebilir (Sarı vd., 2005).

Kontrol sistemlerinde karşılaşılan en büyük problemlerden biri denetim mekanizmasına ait matematiksel modelin oluşturulmasının zor olmasıdır. Bu model oluşturulsa bile geleneksel mantık ile uygulamayı gerçekleştirmek çok karmaşık problemlere yol açacaktır. Bulanık Mantık ise kontrol sistemine ait matematiksel modele ihtiyaç duymadan sadece dilsel değişkenler yardımıyla denetim mekanizmasını oluşturabilmektedir (Özek ve Sincen, 2004).

Bir model kurarken, her zaman beklenen faydanın en yüksek olması amaçlanır. Bu amaç, tüm sistem modellerinde üç temel nitelikte yakından ilişkilidir; karmaşıklık, güvenilirlik ve belirsizlik. Bu nitelikler ve istenilen amaç arasındaki ilişki henüz tam anlamıyla anlaşılabilmiş değildir. Biz sadece faydanın en yüksek olabilmesi için belirsizliğin bu nitelikler arasında en önemli yeri tuttuğunu bilmekteyiz. Fakat yalnız başına değerlendirildiğinde genellikle istenilmeyen bir durum olan belirsizlik, diğer niteliklerle ilişkisinin getireceği sonuçlar itibarıyla önemli bir değer taşır. Belirsizliğin fazlaşmasına izin vermek, karmaşıklığın azalmasına sebep olurken güvenilirliğin azalmasına yol açar. Sistem modellenmesinde, her bir modelleme problemi için optimum seviyede belirsizliğe izin veren metotlar geliştirmek izlenilecek en iyi yoldur (Elmas, 2003).

Bulanık mantığın uygulama alanları her geçen gün artmaktadır. Çünkü, insanların tecrübesi ile öğrenme olayını en yakın şekilde modelleyebilmesine olanak tanımaktadır. Bulanık mantık konusunda Japonya'da oldukça araştırmalar yapılmaktadır.

Günümüzde bulanık mantık ile kontrol edilebilen birçok uygulama görmek mümkündür. Elektronik cihazlar, klima sistemleri, baraj kapaklarının açılıp/kapanması, otomotiv sektörü, doğal afetlerin önceden tahmin edilmesi, kanser teşhisi konulması, trafik lambaları, havacılık ve uzay sanayinde ve bunun gibi birçok alanda bulanık mantık kullanılmaktadır.

Gelecekte kullandığımız bütün elektrik/elektronik cihazlara uzaktan erişim sağlayabileceğiz, bunları kontrol edebileceğiz. Ancak daha ileride ise bu cihazları kontrol etmek için bulanık mantık ile insan gibi kararlar verebilen sistemler oluşturacağız. Birbiriyle haberleşebilen, kendi topladığı verileri yorumlayıp diğer cihaza gönderebilen, onunda verileri yorumlaması ve insan gibi kararlar vererek kontrol edebileceği sistemler çok da uzak görünmemektedir.

Bulanık mantık, günümüzde hemen hemen her alandaki kontrol sistemlerini modellemek ve tasarlamak için yaygın olarak kullanılan bir yapay zeka algoritmasıdır. Bulanık mantık algoritması özetle; bulanık (belirsizlik) özellikteki durum ve kontrol değişkenlerinin önce bulanık üyelik fonksiyonları oluşturulur, daha sonra bu parametreler arasındaki bulanık ilişki kuralları tanımlanır. Bulanık mantık çıkarım algoritması da bulanık kuralları dikkate alarak, bulanık parametrelerin ağırlıklarını hesaplayarak sonuçlar üretir. Problemleri bulanık mantık ile modellemek için herhangi bir programlama dili ile kodlama yapılabileceği gibi günümüzde genellikle geliştirilmiş olan birçok bulanık mantık modelleme paket yazılımları kullanılmaktadır. Ancak bulanık mantık kontrol sistemleri modellerinin donanımsal tasarımlarını hazırlamadan önce doğrulanacağı ve test edileceği genel amaçlı, esnek yapıda donanım simülatörleri bulunmamaktadır.

Bu çalışma ile herhangi bir yazılım modelleme aracında modellenen bir bulanık kontrol sistemi modelinin donanım üzerindeki çalışma başarımının izlenebileceği, doğrulama ve testinin yapılabileceği donanımsal bir simülatör tasarımı ve uygulaması yapılması amaçlanmıştır.

Bu amaç doğrultusunda en çok beş giriş, beş çıkışa sahip sistemler simüle edilmiştir. Giriş parametre [0 1] değerlerini simüle edebilecek analog potansiyometreler ve onların değerlerini gösteren Nextion HMI TFT LCD dokunmatik ekran yer almıştır. Çıkış parametre [0 1] değerlerini simüle edebilmek için Nextion HMI TFT LCD dokunmatik ekran kullanılmıştır. Giriş ve çıkışlar arasındaki bulanık ilişki kuralları ve çıkarım mekanizması ise bir Arduino UNO kartı kullanılmıştır.

Tezin ilerleyen bölümlerinde kaynak taraması ve değerlendirmeleri, yapay zeka ve bulanık mantık, amaçlanan çalışmanın tasarımı ve uygulaması, ayrıca araştırma bulgularına yer verilmektedir.



## 2. KAYNAK ÖZETLERİ

Bu bölümde araştırma konusuna temel oluşturan yapay zeka, bulanık mantık, simulink, nextion hmi tft lcd ekran, arduino ile ilgili çalışmaya katkı sağlayan kaynak özetleri verilerek değerlendirilmiştir.

Sinecen (2002), “Klima Sistem Kontrolünün Bulanık Mantık ile Modellemesi” isimli yüksek lisans tezinde, dilsel değişkenler yardımıyla klasik mantığa göre karmaşık ve güç olan klima sisteminin kontrolünü bulanık mantıkla incelemiştir. Sistemin tasarımı için Matlab Fuzzy Logic Toolbox simülasyon programını kullanmıştır. Sonuç olarak, klasik mantık ve bulanık mantık ile oluşturulan kontrol sistemlerini karşılaştırarak bulanık mantık kontrol sistemlerinin daha ekonomik bir yöntem olduğunu göstermiştir.

Bolat (2006), “Matlab-Simulink + Pic Tabanlı Bulanık Mantık Denetleyici Tasarımı ve Gerçek Zamanlı Sıcaklık Kontrolü Uygulaması” isimli yüksek lisans tezinde, FuzzyTECH programı ile bulanık mantık denetleyici tasarımı yapılarak, C kodları üzerinde gerekli düzenlemelerden sonra kodların mikrodenetleyiciye yüklenmesini sağlamıştır. Oluşturulan sistem ile fırının hem matlab-simulink kullanılarak simülasyonu hem de pic tabanlı kart ile elde edilen sonuçlar karşılaştırılmış, iki durum içinde çıkan sonuçların bir biriyle aynı olduğunu göstermiştir.

Karadeniz (2006), “Çoklu Kompresör Sisteminin Bulanık Mantık ile Kontrolü” isimli yüksek lisans tezinde, bulanık mantık kontrolü kavramı üzerinde durarak, çoklu kompresör sistemlerindeki klasik kontrol algoritmalarından farklı olarak bulanık mantık kontrol tasarımı gerçekleştirmiştir. Bulanık mantık modelleme yazılımı olarak FuzzyTECH programını kullanmıştır. Özellikle belirsizliklerin ve giriş değişkenlerinin çok olduğu kontrol uygulamalarında klasik kontrol algoritmalarının yerine tercih edilmesi gerektiğini bir simülatör sistemi üzerinde göstermektedir.

Nakornnayok vd. (2006), “DC Motor Speed Control Using Fuzzy Logic Based On LabVIEW” isimli makalede, LabVIEW programı ile bulanık mantık denetimi kullanarak bir DC motorun hız denetiminde uygulanması açıklamaktadır. Bulanık mantık denetleyicisi, motor hızı hatasının bulanık kural tabanına ve hız hatasının değişimine bağlı olarak DC motora gönderilen gerekli kontrol gerilimini uygulamak üzere tasarlanmıştır.

Soy (2006), "Gömülü Sistem Üzerinden Bulanık Kontrol ile Ev Otomasyonunu Gerçekleştirilmesi" isimli yüksek lisans tezinde, ev otomasyon sistemini bulanık kontrolör kullanılarak denetlenmeye çalışılmıştır. Bu doğrultuda bulanık kurallar, üyelik fonksiyonları, bulanıklaştırma ve durulaştırma işlemlerini yapan birimleri belirleyerek, mikrokontrolör üzerindeki yazılım sayesinde gerçekleştirilmiştir. İnsan tecrübelerine yakın davranan bulanık mantık kontrol sistemi için konforlu bir iklimlendirme amacıyla giriş ve çıkışları kontrol edebilen sistem tasarlanmıştır. Mikrokontrolör için kodlar C programlama dilinde yazılmıştır.

Tiryaki ve Kazan (2007), "Bulaşık Makinesinin Bulanık Mantık ile Modellenmesi" isimli makalelerinde, hemen hemen her evde kullanılan bulaşık makinesinin bulanık mantık ile modellenmesini ortaya koymuştur. Giriş parametrelerine göre üyelik fonksiyonları belirlenerek, çıkış parametreleri üyelik fonksiyonları ile bulaşık makinesinin daha verimli kullanılmasını sağlamayı amaçlamışlardır.

Taşkın (2009), "Klasik Mantık ve Bulanık Mantık Tabanlı Sıcaklık Dengeleyici Sistemlerin Geliştirilen Bir Prototip Üzerinden Karşılaştırılması" isimli çalışmasında, klasik mantık ile bulanık mantık kontrol sistemlerini karşılaştırarak, insan gibi düşünebilen, öğrenebilen ve kararlar verebilen sistemler için bulanık mantık kontrol sistemlerinin uygun olduğunu belirtmiştir. Klasik mantık ile kontrole kıyasla bulanık mantık ile kontrolün daha basit, ucuz ve hız açısından elverişli olduğunu savunmaktadır.

Erkaymaz ve Çayıroğlu (2010), "Bulanık Mantık ve PIC Kullanılarak Bir Klima Sisteminin Kontrolü" başlıklı makale bulunmaktadır. Bu çalışmalarında, klima sisteminin kontrolü için bulanık mantık sistemi ile mikrodenetleyiciye program yükleyerek gerçekleştirilmiştir. Giriş için sıcaklık ve nem değerleri sensör yardımıyla okunarak mikrodenetleyiciye gönderilmektedir. Çıkış olarak ısıtıcı/soğutucu fanlar ile iklimlendirme sağlanmaktadır. Yaptıkları çalışmada matlab programını kullanmışlardır.

Mahmood (2010), "Bulanık Mantık Kullanılarak Trafik Kontrolünün Tasarımı ve Uygulaması" isimli yüksek lisans tezinde, nüfusun ve trafikteki araç sayısının her geçen gün arttığı, ulaşımda zamandan tasarruf sağlamak için trafik ışıklarının kontrolünü bulanık mantık kontrolü ile tasarımını açıklamıştır. Mevcut trafik kontrol sistemlerinin bu durum karşısında çok yetersiz kaldığını ve trafiğin anlık durumuna göre kararlar verebilen sistemlerin gerekli olduğunu düşünmektedir.

Tayıldız (2013), “Bulanık Mantık Model Geliřtirme Ortamlarının Karřılařtırılması ve rnek Bir Uygulama” isimli tezinde, rnek bir sunucu odası ortamı iin istenen en iyi şartların; FuzzyTECH, MATLAB Fuzzy Logic Toolbox ve LabVIEW PID and Fuzzy Logic Toolkit programları ile tasarımını gerekleřtirmiřtir. Tasarlanan bulanık kontrol sistemi sayesinde, sunucu odasındaki klima sisteminin ıkıř deęerlerini belirlemek iin bu  program eřitli ynleriyle incelenerek karřılařtırılmıřtır.

Munadi ve Amirullah (2014), “Simulation Of Fuzzy Logic Control For Dc Servo Motor Using Arduino Based On Matlab/Simulink” isimli makalelerinde, DC servo motoru kullanarak robot kolunu kontrol etmektedirler. DC servo motorun PID denetleyicisinin tasarımını MATLAB / Simulink'te geliřtirmekte ve Arduino mikrodenetleyicisini kullanarak potansiyometre ile Arduino Mega 2560'a yklenen yazılım sayesinde kontrol edebilmektedir.

Caldo vd. (2015), “Design And Development Of Fuzzy Logic Controlled Dimming Lighting System Using Arduino Microcontroller” isimli makalelerinde, Arduino ile bulanık mantık tabanlı ortam karartma sistemi tasarlamıřlardır. Giriř olarak ıřık kaynakları iin iki yelik fonksiyonu ve ıktı iin bir yelik fonksiyonu ile, sensrlerden gelen veriler doęrultusunda blgede bir hareket varsa ıřıkları otomatik olarak amak ve kapatmak iin sistemi oluřturmuřlardır. Kontrol sisteminin ıkıřı ise led lambalarının karartma yzdesini belirlemek iin kullanılmıřtır. Bu alıřma sayesinde elektrik faturasında tasarrufu saęlamayı hedeflemektedirler.

Nayanmani ve Pranjal Jyoti (2017), “Liquid Level Control Using Fuzzy Logic in LabVIEW with ARDUINO” isimli makalede, LabVIEW programı ile bulanık mantık denetimi kullanılarak potansiyometre ile ayarlanabilen su tankının seviyesine gre led'in ve buzzer'ın deęiřken ıktılara gre farklı deęerler vermesini ele almıřlardır.

Bu zetlenen bulanık mantık kontrol sistemlerinin tamamı iin temelde aynı fakat zelde ayrı ayrı prototipler geliřtirilmiřtir. Bu tez alıřmasında literatrden farklı olarak en ok 5 giriř ve 5 ıkıřa sahip tm bulanık mantık ile modellenmiř kontrol sistemleri donanımsal olarak denenebilecektir.

### **3. MATERYAL VE YÖNTEM**

Bu bölümde yapay zeka, bulanık mantık ve simülatör kavramlarına ve simülatör tasarımıında kullanılan arduino mikrodenetleyicisi, nextion HMI TFT LCD ekran, toggle switch ve potansiyometre'den oluşan donanım elemanlarına değinilmiştir.

#### **3.1. Yapay Zeka**

YZ kavramı ilk defa 1956 yılında Amerika'nın Dartmouth kolejinde 10 bilim adamının katıldığı konferansta doğmuştur. Bu konferansta Mc. Carthy, C. Shannon, H. Simon ve A. Newel, zeka ile donatılmış bilgisayar programları geliştirmeyi önermişlerdir. Newel ve Simon'ın YZ ismine karşı gelmelerine rağmen bu isim günümüze kadar süregelmiştir (Hant, 1986).

1965'te Zadeh tarafından önerilen bulanık mantık teorisi ise yapay zekaya yeni bir bakış açısı kazandırmış, önemini arttırmıştır.

Yapay zeka; zeka ve düşünme gerektiren işlemlerin bilgisayarlar tarafından yapılmasını sağlayacak araştırmaların yapılması ve yeni yöntemlerin geliştirilmesi hususunda çalışılan bilim dalıdır (Sağiroğlu vd., 2003).

Yapay zekaya sahip yapılar, insan gibi düşünmesi istenen sistemlerdir. İnsan gibi karar verip, insanın olmadığı yerlerde karar verme yetisine sahip sistemlerdir (Karaosmanoğlu, 2007).

Klasik sistemlerden farklı olarak kendi başlarına sorgulama, karar verme yeteneğine sahip bu yöntemleri günümüzde üç ana başlık altında değerlendirebilir. Bunlar yapay sinir ağları, uzman sistemler ve bulanık mantıktır. Yapılan uygulama açısından en çok bulanık mantığa değinilecektir.

##### **3.1.1. Yapay sinir ağları**

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacıyla geliştirilen bilgisayar sistemleridir. Yapay sinir ağlarının programlanması çok zor veya mümkün değildir (Öztemel, 2006).

### 3.1.2. Uzman sistemler

Uzman sistemler uzman kişilerin yaptığı tasarım, kompozisyon, planlama, teşhis etme, yorumlama, özetleme, kontrol etme, tavsiyede bulunma gibi fonksiyonları yapmak üzere geliştirilen bilgisayar programlarıdır (Annebel, 1990).

Sadece uzman kişinin çözebileceği problemler için bir uzman sistem geliştirilmektedir. Uzman kişiye her zaman ve her yerde ulaşılamadığı için, uzman kişiden alınan bilgiler ve tecrübeden faydalanılarak elde edilen bilgiler doğrultusunda oluşturulan sistemlerdir.

### 3.1.3. Bulanık mantık

Bulanık mantığın temeli bulanık küme ve alt kümelere dayanır. Klasik yaklaşımda bir varlık ya kümenin elemanıdır ya da değildir. Matematiksel olarak ifade edildiğinde varlık küme ile olan üyelik ilişkisi bakımından kümenin elemanı olduğunda "1", kümenin elemanı olmadığı zaman "0" değerini alır. Bulanık mantık klasik küme gösteriminin genişletilmesidir. Bulanık varlık kümesinde her bir varlığın üyelik derecesi vardır. Varlıkların üyelik dereceleri (0,1) aralığında herhangi bir değer olabilir ve üyelik fonksiyonu  $\mu(x)$  ile gösterilir. Bir bulanık kümenin üyelik fonksiyonu tanım kümesindeki bütün elemanları, çoğunlukla birim aralık olarak kabul edilen bir aralığa atanır (Altaş, 1999).

Bulanık mantık kuramı, Azeri asıllı Amerikalı sibernetikçi Lutfi A. Zadeh tarafından 1965 yılında ortaya atılmıştır. Bulanık mantık iki değerli mantık kuramlarına bir alternatif oluşturmuş, sibernetik ve onunla doğrudan ilgili bilimlerde akıl almaz derecede hızlı gelişmeleri tetiklemiş, sibernetik ve yapay zekâ çalışmalarını hızlandırmıştır. Lutfi A. Zadeh mantık, sibernetik, bilgisayar, yapay zekâ ve otomatik makinelerle ilgili kuramları ve keşifleri ile teknolojik alanda önemli pay sahibidir. İnsana özgü eylemlerde bulunabilen makinelere olanak tanıyan sibernetiğin önemi her geçen gün artmaktadır.

Bulanık mantık ortaya atıldığı ülke olan Amerika'da çok da yaygınlaşma fırsatı bulamamıştır. Bunun nedenleri olarak klasik mantık kavramının ve bilimin kesin doğruluğunun şüphe götürmez gerçek olduğu düşüncesidir. Kullanılan bulanık (fuzzy) kelimesi de çokları tarafından anlaşılmasız ve gereksiz olarak bulunmuştur. Ancak sistemlerin daha esnek kontrol ve dizaynına olanak veren bulanık mantık kavramı doğduğu ülkeden çok uzaklarda Japonya'da sahiplenilmiş ve üzerine çok sayıda araştırma yapılmıştır. Günümüzde bulanık mantığın en yaygın bir kontrol yapısı olarak

kullanıldığı ülke Japonya'dır. Japonya'da deprem çalışmalarında, hızlı tren çalışmalarında ve benzeri çok sayıda çalışmada bulanık mantık kullanılmaktadır. Bulanık mantık günümüzde yaygın olarak kullanılmaktadır. Kullanıldığı alanlar daha çok kontrol sistemleri dizaynının da, bilgi işlem süreçlerinde, mühendislik alanlarında, yüz tanıma gibi üst düzey kontrol isteyen çalışmalarda kullanılmaktadır (Taçyıldız, 2013).

Lutfi A. Zadeh: "Bulanık mantık her şeyin, doğrunun da, bir derece meselesi olduğu insani akıl yürütme için bir modeldir. Temelde, sözcükle hesaplama anlamı sunmaktadır" demiştir (Zadeh, 1998).

Bulanık küme teorisi, belirsiz durumları tanımlamak için üyelik durumlarına göre kademeli geçiş özelliği sayesinde önemli yere sahiptir.

Bulanık küme teorisi; belirsizliğin ölçülmesinde güçlü ve anlamlı araçlar sunmasına ek olarak, dilimizde ifade edilen belirsiz kavramların anlamlı bir biçimde temsilini sağlar. Ancak Aristo mantığı üzerinde temellenen klasik küme teorisi, verilen bir alana ait bütün bireyleri incelenen özelliğe göre, kümeye ait olan elemanlar ve ait olmayanlar olmak üzere ele almaktadır. Kümeye üye ve üye olmayan elemanlar arasında kesin ve belirsiz olmayan bir ayırım vardır. Dilimizde ifade edilen ve üzerinde çalışılan çoğu sınıflandırma kavramı, bu türde bir karakter de değildir. Örneğin; uzun boylu insanlar kümesi, pahalı otomobiller kümesi, yakın sürüş mesafesi, güvenilir kar araçları, birden çok büyük sayıların oluşturduğu küme gibi kavramlar klasik küme teorisinin öngördüğü şekilde incelenemezler (Sarı vd., 2005).

### **3.1.3.1. Belirsizlik kavramı**

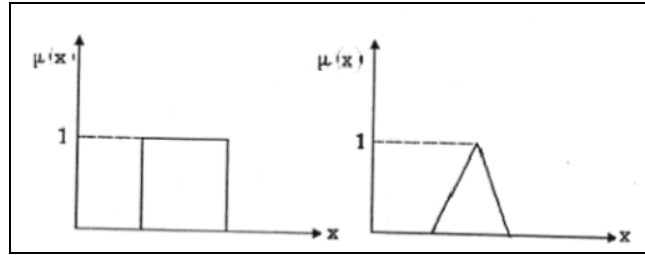
Gerçek dünyanın karmaşıklığı, belirsizliklerin ve karar verememe durumlarından kaynaklıdır. İnsanların bilgi ve tecrübe eksikliğinden dolayı kararlar alamaması veya yanlış kararlar alması belirsizliklerin doğmasına sebep olmaktadır. Bilgisayarların bu tür belirsizlikleri ortadan kaldırabilmesi için, belirsiz kavramların sayısal değerlere dönüştürülmesi, bilgisayar tarafından anlaşılması gerekmektedir. İnsan gerçek yaşamdan bir olayı, bilgi ve tecrübesine dayanarak olumlu veya olumsuz karara bağlayabilirken, gelişen olayları yorumlayarak davranış sergilemektedir. Genel olarak kişiden kişiye değişen, tecrübe ve bilgiye bağlı ortaya çıkan belirsizliklere ve kesin olmayan bilgi kaynağına bulanık kaynak denir.

### 3.1.3.2. Bulanık mantık ve küme teorisi

Aristo mantığına göre çalışan ve şimdiye kadar alışlagelen klasik küme kavramında, bir kümeye giren elemanların oraya ait olmaları durumunda üyelik dereceleri 1'e ait olmamaları durumunda ise 0'a eşit varsayılmıştır. Bu iki değer arasında hiçbir üyelik derecesi düşünülemez. Oysaki bulanık kümeler kavramında 0 ile 1 arasında değişen, değişik üyelik derecelerinden söz etmek mümkündür. Aslında ZADEH küme elemanlarının üyelik derecelerinin 0 ile 1 arasında değişebileceğini ileri sürerek, kümeler teoreminde geniş uygulamaya sahip ve gerçek hayatla uyumlu olan bulanık küme teoremini geliştirmiştir (Baykal ve Beyan, 2004).

Klasik kümelere bakıldığında, bir elemanın hangi kümeye ait olduğu kesindir ve ani değişen üyelik dereceleri ile gösterilmektedir. Bulanık kümelerde ise üyelik derecesi sürekli ve yumuşak bir geçişle gösterilmektedir. Her iki durumda Şekil 3.1.'de gösterilmektedir.

Bu geçişte belirsizlik, hayal gücü, sezgi gibi görüşler rol oynar. Üyelik işlevi, bu tür görüşlerin karışık bir şekilde elemanlara yayılmasını temsil eder. Buradan, bulanık kümenin değişik üyelik derecesinde elemanları olan bir topluluk olduğu sonucu çıkarılabilir. Klasik kümelerde bir elemanın kümeye ait olabilmesi için üyelik derecesinin mutlaka 1'e eşit olması gerekirken, bulanık kümede nerede ise bütün elemanların değişik derecelere kümeye ait olmaları mümkündür.



Şekil 3.1. Klasik küme ve bulanık küme üyelik derecesi işlevleri

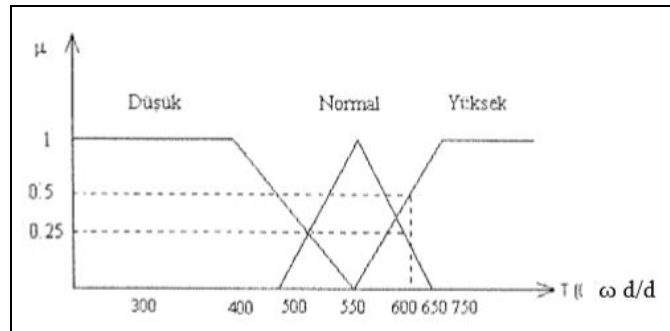
Burada notasyon olarak bulanık kümeler büyük harflerin altına bir çizgi işaretinin konulması ile gösterilecektir. Genelde, bir klasik X kümesinin elemanları;  
 $X = \{x_1, x_2, x_3, \dots\}$  şeklinde gösterilirken, bunun bulanık gösterimi;  
 $\underline{X} = \{\mu(x_1)/x_1 + \mu(x_2)/x_2 + \dots\} = \{\sum \mu(x_i)/x_i\}$  şeklindedir. Bulanık kümenin sürekli olması durumunda ise;  $\underline{X} = \int \mu(x)/x dx$  olur.

Her iki notasyonda da bölüm işareti bölmeyi göstermez. Sadece alttaki gerçek sayıya yani küme elemanlarına üstteki üyelik derecesinin karşı geldiğini belirtir. Yukarıdaki

eşitlilerin ilkinde toplam işareti de bildiğimiz toplamı değil, küme elemanlarının topluluğunu ifade etmek içindir. İkinci notasyonda integral işareti de asla bilinen integral anlamına gelmez yine topluluğu gösteren bir işaret olarak algılanmalıdır. Örneğin sıcaklık kelimesinin İstanbul için bulanık küme olarak gösterilirse; 'sıcaklık' = { 0.1/18 + 0.3 /20 + 0.5/22 + 0.7/24 + 0.8/26 + 0.9/28 + 1.0/30} şeklinde olabilir. 18°C'lik sıcaklığın bu kümeye üyelik derecesi 0.1 iken, 28°C'lik sıcaklığın bu kümeye üyelik derecesi 0.9'dur (Karabağ, 2004).

Geleneksel mantıkta bir değişkenin gerçeklik değeri ya 1 ' ya da '0' olabilir. Ancak bulanık mantıkta bu değer 0 ile 1 aralığı içindeki tüm değerleri alabilir. Bu değere üyelik derecesi adı verilir. Devamlı bir değişken için üyelik derecesi, ya da bir girdinin hangi üyelik değerini alacağını belirleyen eğriler üyelik fonksiyonlar şeklin her bir değerini şekillendirir. Kullanılan matematiğin basit olması, doğrusal olmayanı olarak adlandırılır. Üyelik fonksiyonları dilsel terimlerin üyelik dereceleri ile teknik sistemleri modelleyebilme yeteneği, günlük dile dayalı olması ve esnek olması bulanık mantığa dayalı sistemlerin kullanımının getirdiği en önemli avantajlar arasındadır (Arınc, 2003).

Üyelik fonksiyonunun  $\mu(x) = 1$  değeri  $x$ 'in A bulanık kümesinin kesin bir elemanı olduğunu tanımlar. Aynı şekilde  $\mu(x) = 0$  değeri  $x$ 'in A bulanık kümesinin dışında olduğunu gösterir.  $\mu_A(x)$  üyelik fonksiyonunun [0-1] aralığında aldığı her değer  $x$ 'in A bulanık kümesindeki üyeliğinin belirsiz değeridir. Bu yüzden kesin olmayan büyüklükler üyelik fonksiyonları tarafından belirtilmiş bulanık kümeler ile temsil edilir (Oğuz, 2000).



Şekil 3.2. Üyelik fonksiyonu

Bu durum Şekil 3.2.'deki gibi bir örnek üzerinde sözel olarak ifade edilebilir. Bir motor hızının şekilde görülen üyelik fonksiyonuyla tanımlanabilir. Motorun hızı "Düşük", "Normal" ve "Yüksek" olarak sözel bulanık terimler ile tanımlanabilir. Üyelik derecesi

[0-1] arasında değerlerdir. Motor hızı 300 d/d ise  $\mu(x)$ 'e göre hız "Düşük" grubuna girecektir. Eğer hız 600 devre/dakika ise hız "Yüksek" bulanık kümesinin 0.5 oranında ve aynı zamanda normal bulanık kümesinin 0.25 oranında üyesi olacaktır (Arıncı, 2003).

### 3.1.3.3. Bulanık kontrol sistemleri

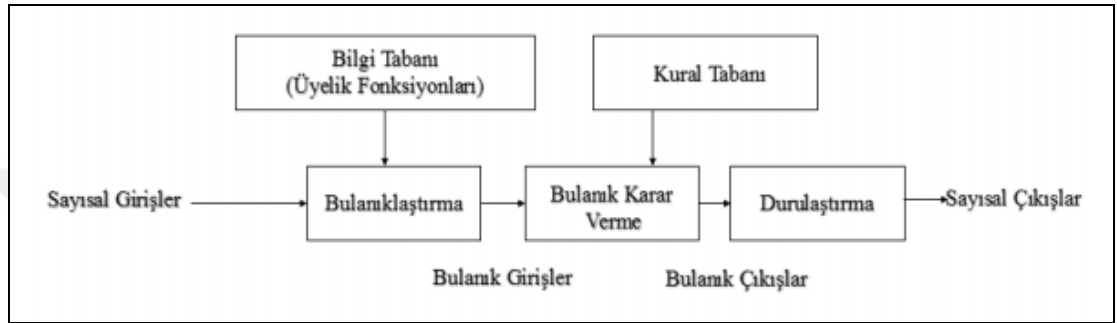
Klasik kontrolörler, matematiksel modeli mevcut olan durumlara uygulanır ve kontrolör bu modele göre tasarlanır. Bulanık kontrolörler ise matematiksel olarak ifade edilemeyen, çok değişkenli süreçlere uygulanmaktadır. Bulanık kontrolör alanında uzman kişinin bilgisine başvurulmuş olarak oluşturulan uzman bir sistemdir. Uzmanın elde edilen bilgiler dilsel cümlelerle ifade edilerek if-else kuralları haline dönüştürülerek bulanık kontrol algoritmasının kurallarını oluşturur.

Bulanık kontrolde ise, kontrolör sabitlerinin belirlenmesinde etkin olan, sistemin davranışdır. Yani, değişik şartlar altında değişik referans değerlerine göre, sistemin vermiş olduğu cevaplar, bir uzman sistem tarafından değerlendirilerek, kontrolör sabitleri belirlenir. Bu uzman sistemler, önceleri insan olmuş, daha sonra yerini yapay sinir ağları ve genetik algoritmalar bu alanda kullanılmaya başlamıştır. Bir bulanık kontrolör tasarımında, kontrolörün sağlıklı çalışabilmesi için, kontrol edilecek sistemin iyi incelenmesi ve tanınması gerekir. Bulanık kontrolörde kullanılan sabitler, bulanık küme sınırları ve etki değerleridir (Korkmaz, 1999).

Bulanık kontrolörün temel dayanağı ise; uzman kişinin bilgi, tecrübe ve kabiliyetini oluşturulan tasarımda bilgi tabanı olarak kullanmaktır. Kontrol işlemleri, tecrübe ve bilgiye dayanan dilsel kurallarla gerçekleştirilir.

Bulanık sistemler; girdilerin tümünü çıktılarının tümüyle ilişkilendirerek küme ve kuralların tanımlamasını yapmaktadır. Bu nedenle de bulanık sistemlerin çalışma yapısı matematiksel bir neden-sonuç fonksiyonunun çalışma şekline benzemektedir. Bulanık sistemlerdeki en önemli kavramlardan bir tanesi bulanık kurallardır. Bulanık kurallar, kontrol amacını gerçekleştirmek üzere tasarlanmış bulanık kontrol kurallarını içermektedir. Bu aşamanın temel amacı ise uzman bilgilerin neden-sonuç ilişkisi içerisinde ifade etmektir. Şekil 3.3.'de (Serhat, 2006) görüldüğü gibi dış dünyadan bilgisayara ölçüm yoluyla alınan ve kesin bir nümerik (sayısal) değere sahip olan giriş verisi, bilgi tabanındaki üyelik fonksiyonları tarafından sözel ifadelere ve giriş verisinin bu ifadeyi ne oranda desteklediğini gösteren üyelik derecelerine dönüştürülür. Bu aşamaya bulanıklaştırma adı verilir. Bulanıklaştırma sonunda elde edilen sözel ifadeler,

insanların karar verme sürecinde olduğu gibi, kural tabanındaki önermelerle karşılaştırılır ve yine sözel yargı sonuçlarına varılır, bu sonuçların hangi oranda geçerli olduğunu yine girişteki üyelik dereceleri belirler. Bu kısma bulanık karar verme süreci adı verilir. Bulanık karar verme sürecinin çıkışında yargı sonuçlarını ifade eden sözel ifadeler ve bunların destek dereceleri bulanık çıkışlar olarak adlandırılır. Eğer bilgisayar çıkışta bir makineye bilgi yolluyorsa, bulanık çıkışlar yine makinelerin anlayacağı dil olan sayısal çıkış değerlerine dönüştürülmelidir. Bu dönüştürme işlemi durulaştırma katında yapılır. (Serhat, 2006).



Şekil 3.3. Bulanık mantık karar verme sistemi

#### 3.1.3.4. Bulanık mantık uygulama alanları

Bulanık mantığın uygulama alanı her geçen gün artmaktadır. Özellikle sanayileşen toplumlarda insan gücünden, hız ve maliyetten kurtulmak isteyen sanayi şirketlerinde yaygın kullanım alanına sahiptir.

Bulanık mantık ilk kez 1973 yılında, Londra'ki Queen Mary College'da profesör olan Ebrahim H. Mamdani tarafından bir buhar makinasında uygulandı. Ticari olarak ise ilk defa, 1980 yılında, Danimarka'daki bir çimento fabrikasının fırınına kontrol etmede kullanılmıştır (Işıklı, 2008).

Bazı bulanık mantık kullanım alanları; insansız hava araçları, otomatik fren sistemi, otomatik vites kontrolü, robot teknolojisi, görüntü işleme, ev elektronik aletleri, trafik lambaları, yapay zeka, karar analizi, veri tabanı sorgulama, sistem analizi, veri işleme, mühendislik, genetik algoritmalarıdır.

Bulanık mantığın üstünlükleri;

- Bulanık sistemlerin tasarımında insan düşünce tarzına ait dilsel terimler kullanılır.
- Matematiksel olarak modellenmesi zor veya imkânsız sistemlerde, zamanla değişen ve doğrusal olmayan sistemlerde başarılı sonuçlar göstermektedirler.

- Bazı sistemlerdeki uygulamalarda sonuç çıkarım sırasında karmaşık matematik işlemlerden kaynaklanan işlem yükünü azaltarak daha hızlı sonuç üretir.

Bulanık mantığın sakıncaları;

- Uzman bilgi ve deneyimlerine ihtiyaç bulunmaktadır.
- Karalılık, gözlemlenebilirlik ve kontrol edilebilirlik analizi yapılmasında ispat edilmiş bir yöntem yoktur.
- Bulanık mantık üyelik işlevleri sisteme özgüdür. Başka bir sisteme uyarlanması zordur.
- Deneme, yanılma yapmadan ne kadar üyelik işlevlerinin kullanılması gerektiğini kestirmek güçtür (Vardar, 2004).

### **3.1.3.5. Bulanık mantık modelleme araçları**

#### **3.1.3.5.1. Matlab bulanık mantık model geliştirme ortamı**

Matlab Mathworks firması tarafından geliştirilmiş olan dördüncü nesil bir programlama dilidir. Matlab kelimesi Matrix Laboratory kelimelerinin kısaltmasıdır ve matris laboratuvarı anlamına gelir. Matlab'ın simülasyon aracı Simulink'tir. Matlab ile bulanık mantık işlemlerini yapabilmek için Matlab içerisine Fuzzy Logic Toolbox'ın yüklenmesi gereklidir ([www.mathworks.com](http://www.mathworks.com)).

Matlab programında yer alan Fuzzy Logic Toolbox, bulanık mantık sistem tasarımı için geliştirilmiştir. Programlama dilleriyle birçok kod satırının yapacağı işlemleri, içerdiği fonksiyonlarla tasarımcıların kısa sürede bulanık sistemi oluşturmasına olanak sağlamaktadır.

Kullanıcılar Fuzzy Logic Toolbox'ta tasarımlar yapabilmek için bulanık mantık sistem tasarımı hakkında bilgi sahibi olmaları gerekir. Programın menüleri ve yardım dosyaları ile kolaylıkla bulanık sistem tasarımları yapabilirler.

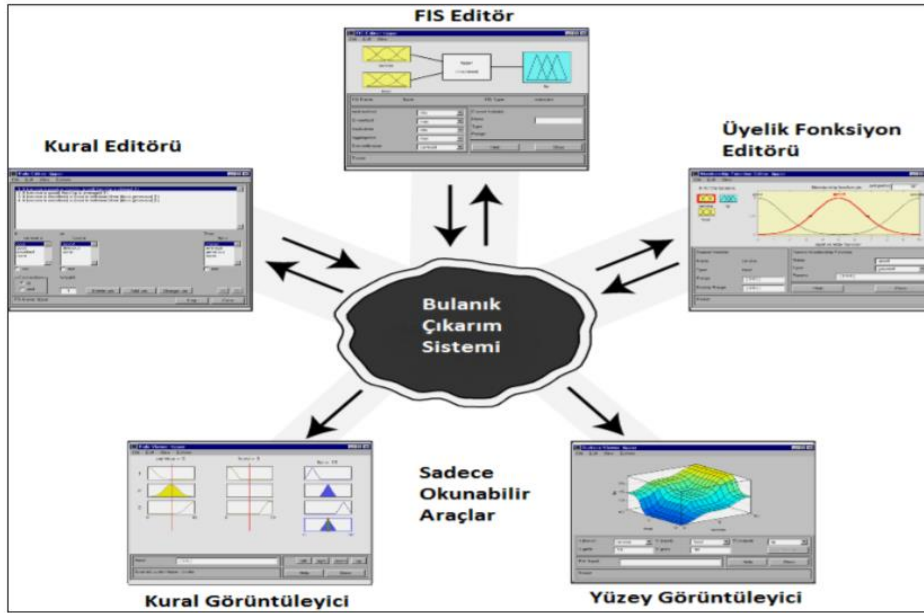
Fuzzy Logic Toolbox ile probleme en yakın bulanık çıkarım metodu seçilmelidir. Sık kullanılan mamdani ve sugeno metotları sistemi kolaylaştırmaktadır. Mamdani metodunda, sistemdeki giriş ve çıkışlara ait üyelik fonksiyonları kesin olmayan, uzman kişinin tecrübesi ile belirleniyorsa kullanılır. Sugeno ise daha çok istatistiki analizlerde tercih edilmektedir.

Matlab programında bulanık mantık tasarımında GUI 'ler kullanılır. Bu GUI 'ler aracılığı ile klasik bulanık sistem geliştirme ve model tanımlamak mümkündür. Bulanık çıkarım

sistemleri analiz ve geliřtirmek, sinirsel bulanık ıkarım sistemleri oluřturmak ve bulanık kmeleme yapmak iin Fuzzy Logic Toolbox kullanılır. Bu Toolbox 'ta bulunan bileřenler řunlardır:

- 1- FIS Editr (Fuzzy Inference System – Bulanık ıkarım Sistemi)
- 2- yelik Fonksiyonu Editr (Membership Function Editor)
- 3- Kural Editr (Rule Editor)
- 4- Kural Grntleyici (Rule Viewer)
- 5- Yzey Grntleyici (Surface Viewer)

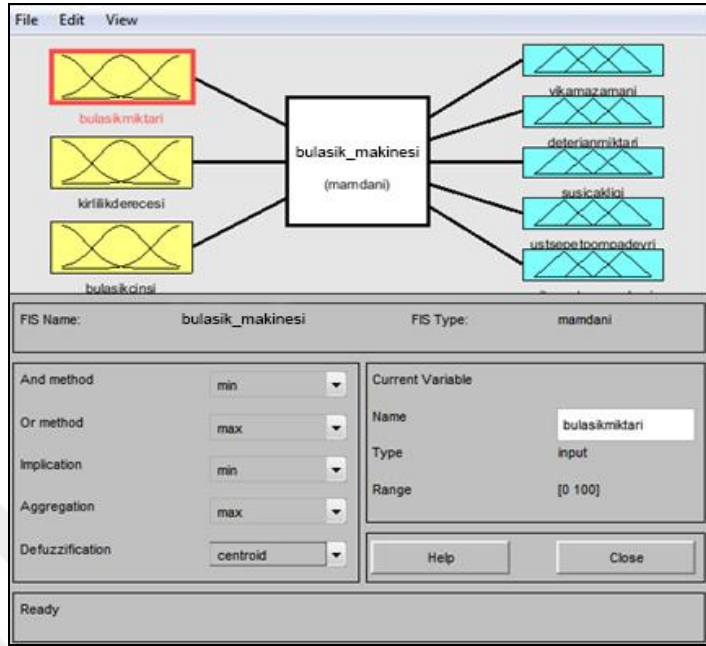
Tm bu aıklanan GUI' ler birbirine dinamik olarak baėlıdır. FIS zerinde yapılan herhangi bir deėiřiklik aynı anda tm GUI'leri deėiřtirir ve bu deėiřim aynı anda grlebilir. rneėin eėer yelik Fonksiyon Editr'nde yelik fonksiyonlarının isimleri deėiřtirilirse bu deėiřiklik aynı anda Kural Editr penceresinde de grlr. İstenen sayıda FIS sistemi iin aynı anda ok sayıda editr yada tm editrler aık olabilir. Matlab programında bulanık mantık sisteminin blok yapısı Őekil 3.4.'de gsterilmiřtir. (Tayıldız, 2013).



Őekil 3.4. Matlab bulanık mantık blok yapısı

FIS editr bulanık sistemin ka giriř ka ıkıř olacaėını, bunların isimleri, aralık deėerleri gibi nemli bilgilerin yer aldıėı editrdr. yelik fonksiyon editr, her giriř ve her ıkıř deėiřkenleri iin yelik fonksiyonlarının řekillerini ve alacaėı deėerleri tanımlar. Kural editr, sistemin davranıřını tanımlayan kuralların oluřturulması ve dzenlenmesi iin kullanılır. Kural grntleyici, giriř deėiřkenlerinin deėerleri

değiştirilerek, çıkış değişkeninin değerini görüntüler. Yüzey görüntüleyici, sistemin 3 boyutlu yüzey haritasını görüntüler.



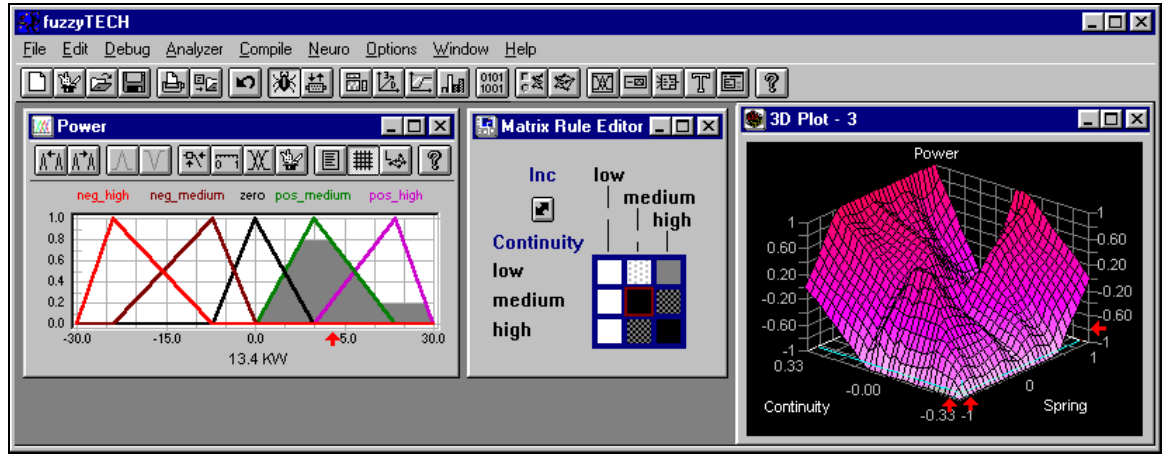
Şekil 3.5. FIS Editör gösterimi

Şekil 3.5.'de FIS Editör'ünde 3 giriş ve 5 çıkışlı bulanık sistemin gösterimi verilmiştir. FIS Editör penceresi 3 menüden oluşmaktadır. File menüsü altında New FIS seçildiğinde Sugeno ve Mamdani olmak üzere iki metod seçeneği gelir. Sugeno metodu giriş değişkenlerine göre sabit ya da lineer bir çıkış üretirken, Mamdani metodunda çıkışlar nonlineer fonksiyonlar olabilir.

### 3.1.3.5.2. FuzzyTech programı

Fuzzytech, bulanık mantık ile modelleme yapmak ve test edebilmek için geliştirilmiş dünyanın sayılı paket programlarından. Sistem bir kere tasarlandığında Fuzzytech oluşturulan tasarımı FTL dosya yapısı olarak kaydeder. FTL "Fuzzy Technology Language" ifadesinin kısaltmasıdır ve bulanık mantık yapısının programlama dili olarak düşünülebilir. FuzzyTech her türlü grafik arayüz desteğini verdiği için FTL'de kod yazılmasına gerek yoktur. Bunun yanısıra Fuzzytech FTL'de hazırlanan yapıyı, bulanık mantık sisteminin çalıştırılacağı donanım için gerekli gömülü kod haline çevirebilir. Bu sebeple mikrokontroller ve PC tabanlı uygulamalar için vazgeçilmez bir araçtır. Birçok modeli(edition) vardır. Fuzzytech MCU-MP modeli kullanılarak oluşturulan bulanık mantık modellemesi sonucunda ASCII, C, ASM, Matlab dosya tabanlı kodlar üretilebilir. Bunun dışında en gelişmiş Fuzzytech modeli olan Fuzzytech Professional modeli

kullanılarak Visual C++/MFC, Visual Basic, Matlab, ANSI C, Java tabanlı kodlar elde edilebilir. Şekil 3.6.'da FuzzyTech programının gösterimi verilmiştir (Karadeniz, 2006).



Şekil 3.6. FuzzyTech Gösterimi

FuzzyTECH programı 1990 yılından itibaren çok sayıda uygulamada diğer geliştirme ortamlarından daha başarılı sonuçlar elde ederek, dünyanın önde gelen çok sayıda büyük şirketiyle bulanık mantık çalışmaları yapmıştır. Audi, Intel Corporation, AEG, Siemens, Toshiba, Volkswagen bu şirketlerden birkaçıdır.

FuzzyTech programının özellikleri;

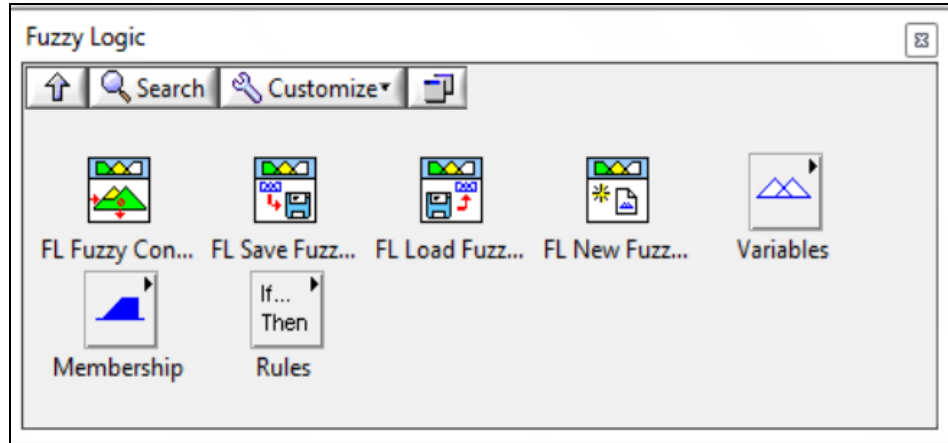
- FuzzyTECH programı teknik uygulamalar için özelleştirilmiş bir programdır.
- FuzzyTECH tasarımlara bulanık mantığın avantajlarını yerleştirmeyi sağlar.
- FuzzyTECH diğer geliştirme ortamlarından daha başarılı bir şekilde bulanık mantık uygulamaları yapmayı sağlar.
- FuzzyTECH kontrol sistemi üreticileri tarafından diğer bulanık mantık araçlarından daha fazla lisans almıştır.
- FuzzyTECH tasarımın gerçekleştirileceği hedef sistemi piyasada bulunan diğer bulanık mantık araçlarından daha fazla destekler.
- FuzzyTECH karmaşık ve hızlı sistemleri diğer bulanık mantık araçlarından daha fazla destekler.
- FuzzyTECH programının bulanık tasarım sihirbazı çözümümüze daha kolay ulaşmamızı sağlar.
- FuzzyTECH dosya indirme sayfasından animasyonlu simülasyon örnekleri indirip test etmeye imkan vermektedir.

FuzzyTECH programı son nesil Windows ve MS Office programlarının ara yüzlerine benzer bir ara yüz sunmaktadır. Programın üst kısmında başlık çubuğu, menü çubuğu ve araç çubukları bulunmaktadır. Windows gezgini Windows üzerinde nasıl kullanıcılara bilgisayarlarında olan dosyaları gösteriyorsa, FuzzyTECH programında bulunan dallanma yapısı da aynı şekilde kullanıcılara çalıştıkları dosyaların tamamına erişmelerini sağlar. Editör ve Analiz pencereleri grafiksel bir bulanık mantık sisteminin her bir bileşeninin tasarımına olanak sağlar. Analiz aşamasında ekran görüntüsü değişmektedir. Aynı anda hem analiz sonuçları hem de tasarlanan sistem görülebilmektedir (Taçyıldız, 2013).

### 3.1.3.5.3. Labview programı

Labview grafik tabanlı bir program geliştirme yazılımıdır. Metin tabanlı yazılım geliştirme ortamlarına kıyasla Labview programları blok diyagram halinde oluşturur.

Labview PID ve Fuzzy Logic Toolkit aracı bulanık mantık uygulamaları geliştirmede iki şekilde kullanılabilir. Birincisi Fuzzy System Designer kullanarak ya da Fuzzy Logic VI'lar yardımıyla. Labview programları veya VI (sanal cihazlar)'lar gerçek olmayan belgeler olarak adlandırılır. Çünkü görünüşleri ve çalışmaları osiloskop ve milimetreler gibi gerçek fiziki belgeleri taklit ederler. Şekil 3.7.'de LabVIEW programındaki bulanık mantık araçları gösterilmektedir (Taçyıldız, 2013).



Şekil 3.7. LabVIEW bulanık mantık araçları

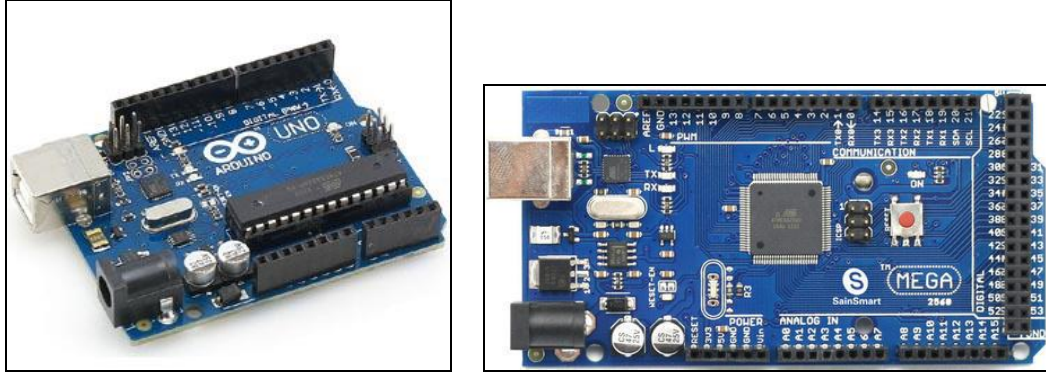
## 3.2. Tasarlanan Simülatörün Donanım Bileşenleri

### 3.2.1. Arduino gömülü sistem kartı

Arduino bir giriş/çıkış kartıdır. Processing/Wiring diliyle program geliştirme ortamı sunan, üzerine eklenebilen shield(kalkan)ler sayesinde fonksiyonu arttırılabilen mikrodenetleyicili programlama platformudur. Arduino'nun diğer adı Genuino'dur.

Hiç programlama deneyimi olmayan kullanıcılara yönelik programlama öğretmek amacıyla geliştirilmiş bir programlama yapısına sahip ve aynı zamanda proje geliştirme ortamıdır. Genellikle çevresiyle etkileşimde bulunabilen nesnelere oluşturmak için kullanılır. Arduino'nun dünya genelinde bu kadar popüler olmasını sağlayan birçok özelliği ve kullanım kolaylığı bulunmaktadır. Kütüphane desteği ile mikrodenetleyiciler hakkında çok fazla bilgisi olmayan kullanıcıların, internetten kolaylıkla yardım alabilmesini sağlar. Bu kütüphaneler "libraries" klasörü altında bulunmaktadır. İhtiyaç duyulan özelliklere göre internetten kütüphaneler indirilebilir. Açık kaynak kodlu olduğu için herkes istediği amaç doğrultusunda kullanma özgürlüğüne sahiptir. Bütün işletim sistemlerinde geliştirme ortamının kurulumu kolay ve çalıştırılabilir olması büyük avantaj sağlamaktadır. Birlikte çalışabilecek, ek özellikler ve fonksiyon katacak shield'lerin bulunması projelerde kullanımını yaygınlaştırıyor. Bu platformun arkasında her geçen gün büyüyen bir topluluk bulunmaktadır. Her gün yeni kullanıcılar ve yeni projelerle topluluk katlanarak büyümektedir. Arduino kartlarında farklı özelliklerde Atmel mikrodenetleyicisi bulunmaktadır. Giriş/çıkış işlemleri için arduino modeline göre farklı sayılarda pinlere sahiptir. Arduino kartları programlamak için harici programcıya ihtiyaç yoktur, bootloader programı üzerinde gelmektedir.

Arduino, analog ve dijital sinyalleri işleyebilir. Sensör shieldleri ile giriş sinyallerini işleyerek, çevreyle etkileşim içerisinde olan sistemler tasarlanabilir. Tasarlanan sisteme özgü tepkiler verebilir. Arduino her ihtiyaca ve çözüme göre bir çok kart ve modül üreterek, kullanıcıların sistemlerine en uygun kartı seçmelerini sağlamaktadır. Şekil 3.8.'de Arduino UNO R3 ve Arduino Mega 2560 geliştirme kartlarının genel görünümü verilmiştir.



Şekil 3.8. Arduino UNO R3 ve Arduino MEGA 2560 genel görünümü

Çizelge 3.1.'de Arduino platformunda en sık kullanılan kartlardan arduino UNO ve arduino mega 2560 kartlarının teknik özellikleri verilmiştir (www.arduino.cc).

Çizelge 3.1. Arduino platformlarının teknik özelliklerinin gösterimi

	Arduino Uno	Arduino Mega 2560
Mikrodenetleyici	Atmega328	Atmega 2560
Çalışma Gerilimi	+5V DC	+5V DC
Tavsiye Edilen Besleme Gerilimi	7-12 V DC	7-12 V DC
Besleme Gerilimi Limitleri	6-20 V	6-20 V
Analog Giriş Pinleri	6 tane	16 tane
I/O Birimleri İçin Maksimum DC Çıkış Akımı	40 mA	40 mA
Saat Frekansı	16 MHZ	16 MHZ
EEPROM	1 KB	4 KB
I/O Sayısı	14 tane (6 PWM)	54 tane (15 PWM)

### 3.2.2. Potansiyometre

Direnç değerini ayarlayabilen, üç bacaklı elektriksel rezistansa potansiyometre denir. Gerilimi bölme işlevi görürler. Potansiyometre ile ayarlı orta uç direnç üzerinde gezinerek direncin değeri değiştirilebilir. Direnç değerinin değiştirilmesiyle gerilim bölme, yani çıkış gerilimini istenilen bir değere ayarlayabilme imkanı sunar. Şekil 3.9.'da potansiyometrenin genel görünümü verilmiştir.



Şekil 3.9. Potansiyometre Genel Görünümü

### 3.2.3. Nextion HMI TFT LCD ekran

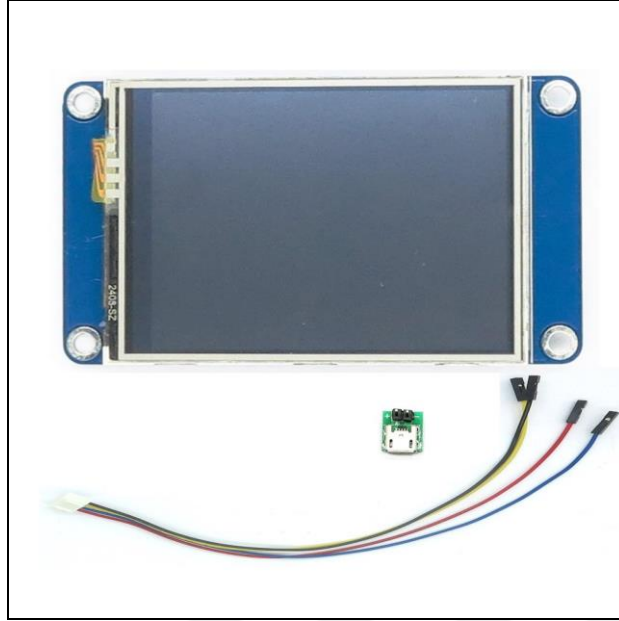
Nextion, bir insan ile bir süreç, makine, uygulama veya cihaz arasında kontrol ve görselleştirme arabirimi sağlayan HMI (Human Machine Interface / İnsan Makine Arayüzü) çözümdür.

Sadece bir ekran olması yanı sıra dokunmatik özelliğe sahiptir. Bu özellik sayesinde veri girişi ve girilen verileri işleyip sonuçlar üretmesi, süreçleri başlatması gibi nedenlerden dolayı akıllı ekran olarak tanımlamak mümkündür. Nextion ekranlara Arduino veya diğer mikro işlemciler ile seri port üzerinden haberleştilirerek birçok iş yaptırmak ve mikro işlemci üzerinden daha az pin kullanarak çözümler üretmek mümkündür.

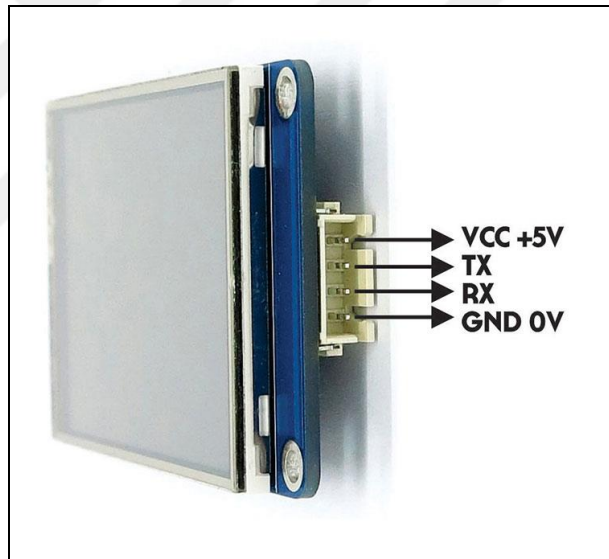
Ekran donanım kısmının yanında yazılım kısmından da oluşmaktadır. Nextion Editor programıyla ekran tasarımı yapılabilmektedir. Nextion Editor ile butonlar, metinler, ilerleme çubuğu, kaydırma çubuğu gibi bir çok nesne ekleyerek kapsamlı ve etkileşimli ara yüzler yapmak mümkündür.

Bu çalışmada 3.5 Inch Nextion HMI TFT LCD ekran (NX4832T035) kullanılmıştır. Yapılan tez çalışmasında bu ekranın özellikleri yeterli olacağı için seçilmiştir.

Şekil 3.10.'da 3.5 Inch Nextion HMI TFT LCD ekranın genel görünümü ve bağlantı kablosu gösterilmiştir.



Şekil 3.10. 3.5 Inch Nextion HMI TFT LCD ekran (NX4832T035)



Şekil 3.11. Nextion HMI TFT LCD ekran seri port bağlantı noktaları

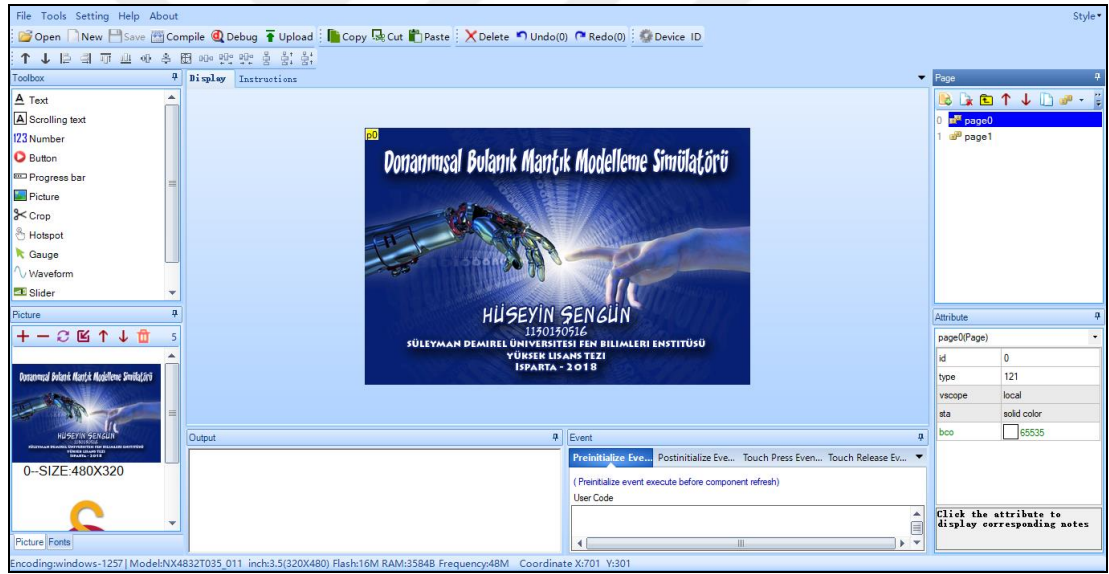
Şekil 3.11.'de Nextion HMI TFT LCD ekranın seri port bağlantı noktaları ve pinleri gösterilmiştir.

3.5 inç nextion nx4832t035 hmi tft lcd akıllı ekran özellikleri;

- 480 x 320 Çözünürlüğü
- RGB 65K canlı renkler için geçerlidir
- Entegre 4 telli Dirençli Dokunmatik Panelli TFT Ekran
- Herhangi bir TTL Seri Ana Makineye kolay 4 pinli arabirim

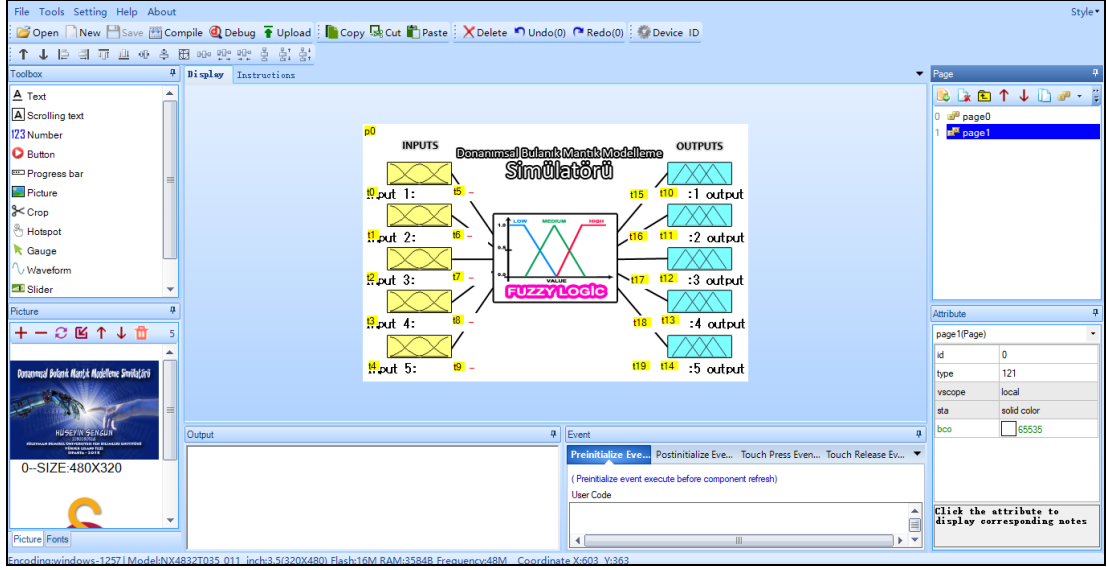
- Kullanıcı Uygulama Kodu ve Verileri için 16M Flash Bellek
- 3.5KByte RAM
- Yazılım yükseltmesi için dahili mikro SD kartı
- Görme Alanı: 73.44mm (L) × 48.96mm (W)
- Ayarlanabilir Parlaklık: 0 ~ 180 nit, ayar aralığı % 1
- 5V 500mA DC önerilen güç kaynağı
- 5V 145mA güç

Nextion HMI TFT LCD ekrana ara yüz yapmak için kullanılan yazılımın adı Nextion Editor programıdır. Bu yazılım sayesinde ekrana butonlar, metinler vb. nesnelere eklemek, eklenen nesnelere görevler vermek için geliştirilmiş kullanışlı bir programdır. Arduino'dan gelen verileri istenilen nesnelere göstermek, ya da kullanıcının dokunduğu nesnelere görevi yerine getirerek arduino'ya işlemler yaptırması mümkündür. Şekil 3.12.'de geliştirilen simülasyonun karşılama ekran tasarımı görülmektedir.



Şekil 3.12. Nextion editör ile ara yüz tasarım ekranı

İlk ara yüz tasarımı simülatöre enerji verildiğinde ekranda görünmekte, ekrana dokunarak ana ekran tasarımı ekranda gösterilmektedir. Sistemin giriş ve çıkışları ana ekran tasarımındaki gerekli yerlerde, parametreleri ve etiketleri ile birlikte görüntülenmektedir.



Şekil 3.13. Nextion editor ile ara yüz tasarımının ana ekranı

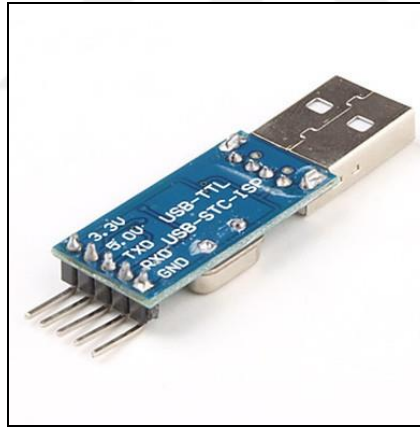
Şekil 3.13.'de simülâtörün nextion editördeki ara yüz tasarımının ana ekranı görülmektedir. Ana ekran tasarımının sol tarafında analog girişlerin etiketleri ve giriş parametre değerleri, sağ tarafta ise çıkış etiketleri ve parametrelerinin gösterileceği text nesnelere yerleştirilmiştir.

Nextion Editor programında yeni bir tasarım yapmadan önce kullanılacak ekran boyutunu seçmek gerekir. Kullanılacak olan Nextion 3.5" HMI TFT LCD ekran 480x320 piksel çözünürlüğe sahiptir. Bu nedenle tasarım aşamasında sayfalarda kullanılacak resimler de ekranda tam olarak gözükebilmesi için bu çözünürlükte seçilmelidir.

Tasarlanan giriş ara yüzü simülâtöre enerji verildiğinde ekranda görünmektedir. Nextion HMI TFT LCD ekranlar dokunmatik özelliğe sahip olduğundan giriş ekran tasarımına dokunulduğunda ana ara yüz tasarımının nextion ekranda görüntülenmesi sağlanmıştır. Ekran ya da tasarımdaki her hangi bir nesneye dokunulduğunda görev vermek için Touch Release Event bölümüne "page sayfa\_numarası" komutu yazılarak gerçekleştirilir. Simülâtör'ün giriş/çıkış değerlerinin gösterildiği ara yüz ekranı ana ekrandır. Burada analog giriş değerlerinin, bu girişlerin parametrelerinin, çıkış değerleri ve bu çıkış değerlerinin parametrelerini göstermek için nextion editör programındaki toolbox'tan text nesnelere eklenmiştir. Her bir text nesnesinin name'i farklıdır ve arduino'dan hangi değer hangi text nesnesinde gösterileceği kodlarla belirlenebilmektedir. Hem giriş ara yüzü hem de ana ara yüzün arka plan resmi photoshop programıyla oluşturulup, picture nesnesi ile ekran tasarımına eklenmiştir.

Oluşturulan ara yüz nextion editor yazılımındaki simülatör sayesinde kontrol edilebilir. Nextion HMI TFT LCD ekrana yüklenmeden önce son değişiklikler yapılarak, ekranda görünecek hali ve nesnelere verilen görevlerin çalışıp çalışmadığı editörün debug özelliği sayesinde test edilmiş olur.

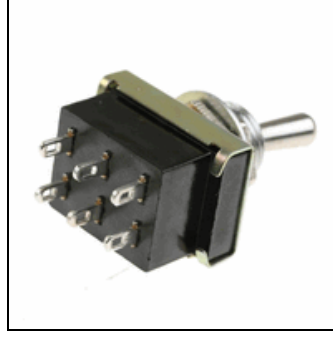
Oluşturulan ara yüz tasarımı kaydedildiğinde HMI uzantısı ile kaydedilir. Bu dosya daha sonra editör ile açılarak düzenlemeler yapılabilir. Fakat Nextion HMI TFT LCD ekrana yükleyeceğimiz tasarım HMI uzantılı dosya değildir, ara yüz tasarımı editörde Debug edildiğinde editör aynı isimde TFT uzantılı bir dosya oluşturur. Bu dosya C:\Kullanıcılar\Kullanıcı\_adi\AppData\Roaming\Nextion Editor\bianyi klasörünün içindedir. Oluşturulan TFT uzantılı ara yüz tasarımı iki yöntemle Nextion HMI TFT LCD ekrana yüklenebilir. Birinci yöntem, USB-TTL Converter ile ara yüz tasarımı Nextion Editor ile upload edilerek yüklenebilir. İkinci yöntem ise ara yüz tasarımı MicroSD hafıza kartına yüklenerek Nextion HMI TFT LCD ekrana takılıp enerji verilerek yükleme işlemi gerçekleştirilir. Yükleme işlemi gerçekleştirildikten sonra MicroSD kart çıkartılır ve enerji kesilip tekrar verilir. Şekil 3.14.'de USB-TTL Converter donanım elemanının pinleri ve bu pinlerin isimleri gösterilmiştir.



Şekil 3.14. USB-TTL Converter

#### 3.2.4. Toggle switch

6 bacaklı ve 2 konumlu bir anahtardır. Kablo değiştirmek veya motor dönüş yönünü değiştirmek için kullanılır. 6'lı Toggle switch hem iki yönlü hem de iki bağımsız anahtar görevi görmektedir. Anahtarın konumu değiştirildiğinde orta bacağı ile diğer bacak kısa devre olur, diğer konuma getirildiğinde diğer bacaklar ile orta bacak kısa devre olur. Şekil 3.15.'de 6'lı Toggle switch anahtarının genel görüntüsü verilmiştir. Çalışma prensibi ilerleyen bölümlerde anlatılmaktadır.



Şekil 3.15. 6'lı Toggle switch

### 3.3. Simülâtör

Simülasyon kavramı genel olarak, bir sistem, süreç ya da durumun taklit edilmesi olarak tanımlanabilir. Dolayısıyla simülasyon, söz konusu sistem, süreç ya da durumu temsil edecek bir modeli içerir. Bu model, temsil edilen sistemle ilgili, gerçek hayatta gerçekleştirilmesi riskli, pahalı ya da zaman gerektiren deneme, inceleme ve çalışmaların gerçekleştirilmesine olanak sağlar. Başka bir deyişle simülasyon, risk, maliyet ve zaman etkenleri açısından tasarruf edici bir kuvvet çarpanı olarak öne çıkmaktadır (Mevlütöğlü, 2013).

Simülasyonlar, gerçek hayattaki durumların benzerlerini bilgisayar ortamında oluşturmasıyla, sistemin çalışması hakkında bilgi veren gerçeğin taklit edilmiş benzetimleridir. Düşük maliyet, güvenlik, hız gibi nedenlerden dolayı simülasyon sistemlerinin kullanımı oldukça popülerdir. Örneğin uzay boşluğunu anlatmak isterken öğrencileri uzaya götürmek yerine uzay boşluğu oluşturabilen yer çekimsiz simülâtörler aracılığıyla gerçekleştirmek, eğitimi fiziki olarak gerçekleştirmekten daha ucuz, daha kısa sürede ve kolaydır.

Simülasyon (benzetim) gerçek hayattaki sistemlerin bir benzerinin farklı bir ortamda kurularak düşük maliyet ile sistemin nasıl çalıştığının anlaşılması yolunda yapılan faaliyetlerdir. Günümüzde simülasyon çalışmalarının büyük çoğunluğu bilgisayarlar üzerinde yapılmaktadır (<http://web.itu.edu.tr/soylemezm/simux/simux.pdf>).

İyi bir simülasyonun gerçek sisteme veya duruma yakın sonuçları hızlı bir şekilde üretmesi ayrıca simülasyon giriş/çıkış parametrelerinin değiştirilebilir olması, kullanıcının kolay bir şekilde simülâtörü kullanabilmesi ve gerçek standartlarla uyumlu olması gerekmektedir.

Simülâtörler, benzetimi yapılan sistemlerin ya sanallaştırılması ya da donanımsal modüllerinin oluşturulması ile gerçekleştirilir. Güvenlik, maliyet, zaman, mekan gibi

nedenlerden dolayı gerçek hayatta uygulanamayacak durumlarda simülasyon ile tasarımı sanallaştırmak kullanılan uygun bir yöntemdir. Özellikle eğitimde kullanılan bu tür simülasyonlarla, öğrenmek için gerçek ortama geçiş yapmadan önce bire bir gerçeğiyle denemeler yapması, uygulaması hem eğitmen, hem de eğitilen açısından daha güvenli olmaktadır. Simülasyon, sanal ortamdaki sistemin donanımsal ortamlarda görselleştirilmesi, elle tutulur hale getirilerek elde edilen sonuçların daha anlaşılır daha somut bir şekilde ortaya konulması içinde donanımsal simülatörler olarak da kullanılmaktadır.

Simülasyon ihtiyacı olan durumlar;

- Deney yapmaya müsait olmuyorsa,
- Henüz uygulamaya geçilmediyse,
- Tasarım çok karmaşıkça,
- Tasarımın uygulamadan önce kontrol edilecekçe,
- Eğitim amacıyla kullanılır.

Simülasyonun özellikleri;

- Kullanıcının kolay anlaması gerekir.
- Amaca ve hedefe uygun olmalıdır.
- Kullanıcının denetiminde olmalıdır.
- Değişiklik ve güncelleme kolaylıkla yapılabilmelidir.
- Basitten zora doğru olmalıdır.

Modelleme ve simülasyon teknolojilerinin günümüzde pek çok kullanım alanı bulunmaktadır. Bunlardan birkaçı aşağıdaki gibidir.

- Eğitim: Hava, kara, deniz gibi farklı platformalara ait sistemlerin (uçak, otomobil, gemi vb) kullanımı amaçlı eğitim.
- Planlama
- Eğlence: etkileşimli eğlence araç ve ortamları hazırlanması.
- Tasarım: konsept tasarım.
- Ar-ge: yeni tasarımların sanal ortamda diğer sistemlerle etkileşimlerinin incelenmesi (Mevlütöğlü, 2013).

#### 4. ARAŞTIRMA BULGULARI

Bulanık mantık her geçen gün önemini giderek arttırmaktadır. Cihazların akıllı olması için kullanılan sistemlerin önümüzdeki yıllarda daha da yaygınlaşacağı düşünüldüğünde bulanık mantık ile yönetilen, idare edilen cihazlar her alandaki ihtiyaçlara cevap verebilecek hale geleceklerdir.

Bulanık mantık ile yapılan sistem tasarımları ya bilgisayar ortamında simülasyon halinde kalmakta ya da donanımsal sistemlere uygulanamamaktadır. Yapılan çalışmaların testi sadece bilgisayar ortamında yapılabilmektedir.

Bu çalışmada herhangi bir bulanık mantık tasarım editöründe yapılan kontrol sistemi modelinin arduino gömülü sistem kartına yüklenebilecek şekilde dönüşümü ve gerekli ayarlamaları yapılarak tasarlanan simülatöre aktarılması sağlanacaktır. Böylelikle bilgisayar yazılımı editörü üzerinde simüle edilmiş olan sistem tasarımı daha somut bir ortamda çalışabilecek donanımsal simülatör ortaya çıkmış olacaktır.

Bu bölümde Donanımsal Bulanık Mantık Modelleme Simülatör Tasarımı ve örnek uygulamalarla sistemin çalışması ele alınacaktır.

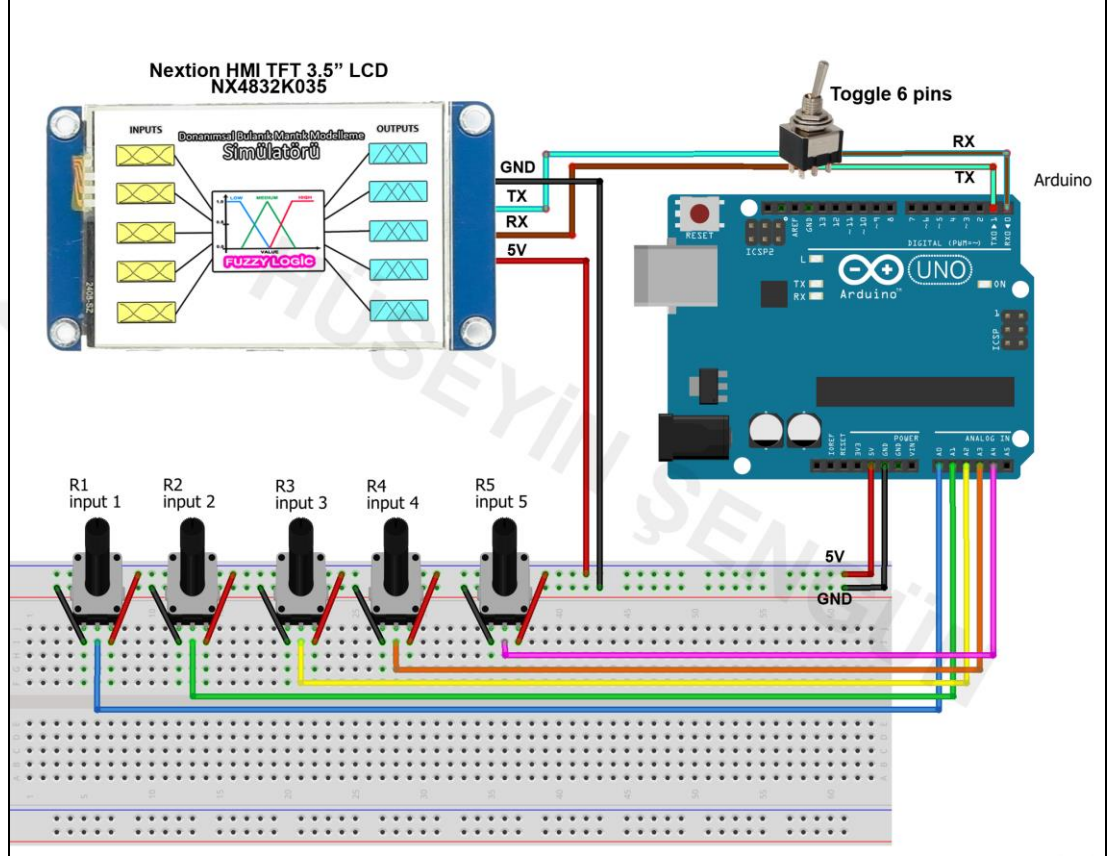
##### 4.1. Geliştirilen Donanımsal Bulanık Mantık Modelleme Simülatör Tasarımı

Çalışmanın donanımsal bölümünde Şekil 4.1.'de görülen 21 cm x 13 cm x 12 cm ölçülerinde bir simülatör prototipi geliştirilmiştir. Prototip plexiglass'tan sisteme uygun olarak özel tasarlanmış ve birleştirilerek hem kullanım hem de koruma amacı taşımaktadır.



Şekil 4.1. Simülatör prototipinin genel görünümü

Bu çalışma ile herhangi bir yazılım modelleme aracında modellenen bir bulanık kontrol sistemi modelinin donanım üzerindeki çalışma başarımının izlenebileceği, doğrulama ve testinin yapılabileceği donanımsal bir simülâtör tasarımı ve uygulaması yapılmıştır.



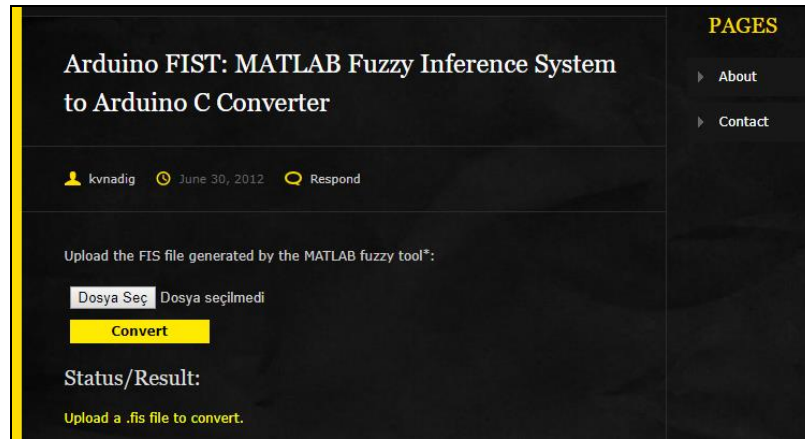
Şekil 4.2. Donanımsal bulanık mantık modelleme simülâtörü devre şeması

Bu amaç doğrultusunda simülâtörün devre şeması Şekil 4.2.'de gösterilmektedir. Donanımsal bulanık mantık modelleme simülâtörünün devre şemasına bakıldığında, bulanık girişler için beş adet 10K potansiyometre kullanılmıştır. Potansiyometrelerden ayarlanan giriş değişkenleri arduino UNO'nun analog girişlerine gönderilmektedir. Arduino UNO'ya düzenlenip yüklenen kod sayesinde giriş değerlerine göre bulanık çıkarım elde edilmekte ve arduino'dan nextion HMI TFT LCD ekrana gönderilmektedir. Bu ekranın sol tarafında giriş parametreleri ve giriş değerleri anlık görüntülenmektedir. Ekranın sağ tarafında da çıkış değerleri ve parametreleri gösterilmektedir.

#### 4.1.1. Arduino UNO'ya bulanık mantık kontrol modelini yükleme

Matlab Fuzzy Logic Toolbox ile oluşturulan herhangi bir bulanık kontrol sistemi, Matlab Fuzzy Logic Toolbox'taki kural görüntüleyicisi ile giriş değişkenleri değiştirilerek, çıkış değerleri görüntülenebilmektedir. Bu durum sadece bilgisayar ortamında simüle edilmiştir. Bulanık çıkarım sistemi (FIS) editörü ile bulanık kontrol sisteminin giriş ve çıkış değişkenleri belirlenmektedir. Oluşturulan bulanık kontrol sistemi kaydedildiğinde matlab programı bize fis uzantılı bir dosya vermektedir. Matlab Fuzzy Logic Toolbox direkt olarak C/C++ programlama dilinde dosya vermemektedir. Bu uzantıya sahip dosya Arduino mikrodenetleyicisine yüklenememektedir. Bunun için mikrodenetleyiciye uygun kod satırları haline getirmek için dönüştürme işlemi yapılması gerekmektedir. Bulanık kontrol sistemindeki bütün kurallar mikrodenetleyiciye uygun if-else kod satırları haline getirilir. Arduino UNO C/C++ kodlarına uyumlu olduğu için, tasarlanan bulanık sistemin C/C++ programlama dilindeki kodlara dönüştürülmesi gerekmektedir.

Dönüşüm için [www.makeproto.com](http://www.makeproto.com) sitesindeki "MATLAB Fuzzy Inference System to Arduino C Converter" özelliği kullanılmıştır. Siteye yüklenen fis uzantılı dosyayı Arduino mikrodenetleyicisine uygun olan C programlama dili haline dönüştürme işlemini yapmaktadır. Şekil 4.3.'de matlab bulanık mantık sistemini arduino C koduna dönüştürücü ekranı görülmektedir. Dosya Seç butonuna tıklayarak fis uzantılı dosyayı siteye yükleyip, convert butonu ile dönüştürme işlemi gerçekleştirilmiştir.



Şekil 4.3. Matlab fuzzy inference system to arduino c converter ekranı

Bu dönüştürme işleminden sonra site 2 dosya üretmektedir. Birinci dosya header dosyasıdır. h uzantılı olan bu dosya, bir C, C++ veya Objective-C kaynak kodu belgesi tarafından başvuru başlık dosyasıdır. Değişkenler, sabitler ve bir programlama projesi içerisinde diğer dosyalar tarafından kullanılan fonksiyonları içerir. Yaygın

olarak kullanılan fonksiyonların yalnızca bir kez yazılmasına ve ihtiyaç duyulduğunda diğer kaynak dosyaları tarafından bunlara başvurulmasına olanak sağlar. İkinci dosya ise ino uzantılı dosyadır ve bulanık kontrol sisteminin kurallarının C kodlarına dönüştürüldüğü Arduino dosyadır. Dönüştürme işleminden sonra arduino IDE programıyla açılan ino uzantılı arduino dosyasında bulanık çıkarım sisteminin giriş/çıkışları, kural tablosundaki kuralları, değişken parametreleri ve üyelik fonksiyonlarının yer aldığı kodları içermektedir. Bu dosya Arduino'da program yazmak için kullanılan Arduino IDE programıyla açılmaktadır.

---

```
#include "fis_header.h"
// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 25;
float olculendeger;
String input1="Ortam sic: ";
String input2="Hedef sic: ";
String output1="Fan hizi: ";
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
// Setup routine runs once when you press reset:
void setup()
{
  pinMode(0 , INPUT);
  pinMode(1 , INPUT);
  Serial.begin(9600);
  pinMode(2 , OUTPUT);
}
```

---

```
const int fis_gcI = 2; // Bulanık çıkarım sistemindeki giriş sayısı
const int fis_gcO = 1; // Bulanık çıkarım sistemindeki çıkış sayısı
const int fis_gcR = 25; // Bulanık çıkarım sistemindeki kural sayısını gösterir.
```

PinMode ile analog giriş ve çıkış değerleri için kullanılacak pin numaraları belirtilmektedir.

---

```

void loop()
{
  g_fisInput[0] = analogRead(A0);
  g_fisInput[0]=map(g_fisInput[0],0,1023,0,100);
  Serial.print("t0.txt=");
  Serial.write(0x22);
  Serial.print(input1);
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.print("t5.txt=");
  Serial.write(0x22);
  Serial.print(g_fisInput[0]);
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
  // Read Input: Hedef_Sıcaklığı
  g_fisInput[1] = analogRead(A1);
  g_fisInput[1]=map(g_fisInput[1],0,1023,0,100);
  Serial.print("t6.txt=");
  Serial.write(0x22);
  Serial.print(g_fisInput[1]);
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.print("t1.txt=");
  Serial.write(0x22);
  Serial.print(input2);
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
}

```

---

Döngüde giriş pinlerinden sürekli okunan değerler nextion ekrandaki ilgili textlere anlık olarak aktarılmaktadır. Milisaniye cinsinden sistemin çalışması nextion HMI TFT LCD ekranda görüntülenmektedir. Böylelikle giriş değişkenleri ve etiketlerinin ekrana yerleştirilen konumlarında gösterimi yapılmıştır. Potansiyometrelerden okunan analog giriş değerleri 0-1023 aralığın da olduğu için, map fonksiyonu ile uygulamaya göre 0-100 arasındaki değerler alması sağlanmıştır. Sistem tasarımcısı bulanık mantık sistemine göre giriş değerleri için belirttiği al/üst limitlerini map fonksiyonu ile değiştirebilmektedir. Döngünün devamında çıkış değerleri için kodlar aşağıda yer almaktadır.

---

```
g_fisOutput[0] = 0;
fis_evaluate();
// Set output vlaue: Çıkış
analogWrite(2 , g_fisOutput[0]);
Serial.print("t10.txt=");
Serial.write(0x22);
Serial.print(output1);
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("t15.txt=");
Serial.write(0x22);
Serial.print(g_fisOutput[0]);
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("ortam sicakligi: ");
Serial.println(g_fisInput[0]);
Serial.print("hedef sicakligi: ");
Serial.println(g_fisInput[1]);
Serial.print("klima cikisi PWM:");
Serial.println(g_fisOutput[0]);
Serial.println("");
delay(100);
}
```

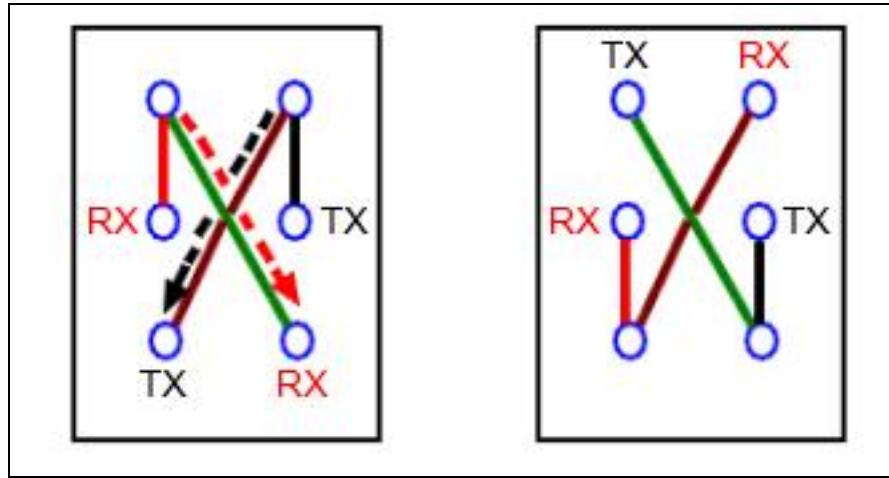
---

Döngüde çıkış değerlerinin hangi etiketle hangi textte gösterileceği belirtilmiştir. Kodlarda gerekli düzenlemelerin yapıldığı bölüm parametrelerin nextion ekranda hangi nesnelere gösterileceğidir. Uygulama olarak seçeceğimiz örneklerin düzenlenmiş kodları tezin ilerleyen kısımlarında gösterilecektir.

Arduino üzerinden Nextion ekrana tasarımın atılması veya arduino IDE programıyla düzenlenen kodların arduino'ya yüklenmesi sırasında Nextion kablolarından siyah kablonun(GND) Ardunio üzerindeki ground pinine, kırmızı kablonun(+5V) Arduino üzerindeki +5V pinine, sarı kablonun(RX) Arduino üzerindeki RX pinine ve son olarak mavi kablonun(TX) Arduino üzerinde TX pinine bağlantısına dikkat edilmesi gerekir.

Arduino IDE'de düzenlenen kodların arduino'ya yükledikten sonra, arduino ve nextion HMI TFT LCD ekranın haberleşebilmesi, gönderilen/alınan komutların iletilebilmesi için RX ve TX kablolarının çapraz bağlanması gerekmektedir. Arduino'ya kod yükleme ve bu kodların sonucunda değerlerin nextion ekranda görüntülenmesi için sürekli kablo değiştirme işlemi yapmak yerine 6 bacaklı toggle switch anahtarı kullanılmıştır.

Bu anahtar ile kablo deęişimi için önce köşelerdeki bacaklar çapraz bağlanır. Şekil 4.4.'de 6 bacaklı toogle switch ile RX/TX kablolarının nasıl bağlandığı gösterilmektedir.



Şekil 4.4. 6'lı toogle switch 1. ve 2. konum bağlantı gösterimi

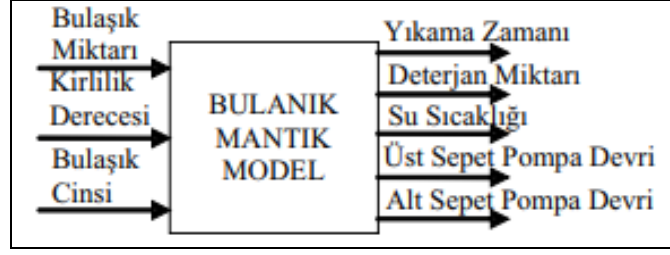
6 bacaklı toogle switch birinci konumuna getirildiğinde arduino IDE programında oluşturulan sistemin kodlarını arduino gömülü sistem kartına yükleme işlemi gerçekleştirilmektedir. 6 bacaklı toogle switch ikinci konumuna getirildiğinde ise arduino gömülü sistem kartındaki kodların sonucu nextion HMI TFT LCD ekranda anlık olarak gösterimi sağlanmıştır.

#### 4.1.2. Örnek uygulama ile sistemin çalışması

Donanımsal bulanık mantık modelleme simülatörünün daha iyi anlaşılması için örnek bir uygulama üzerinden sistemin çalışmasını anlatmak daha uygun gelecektir.

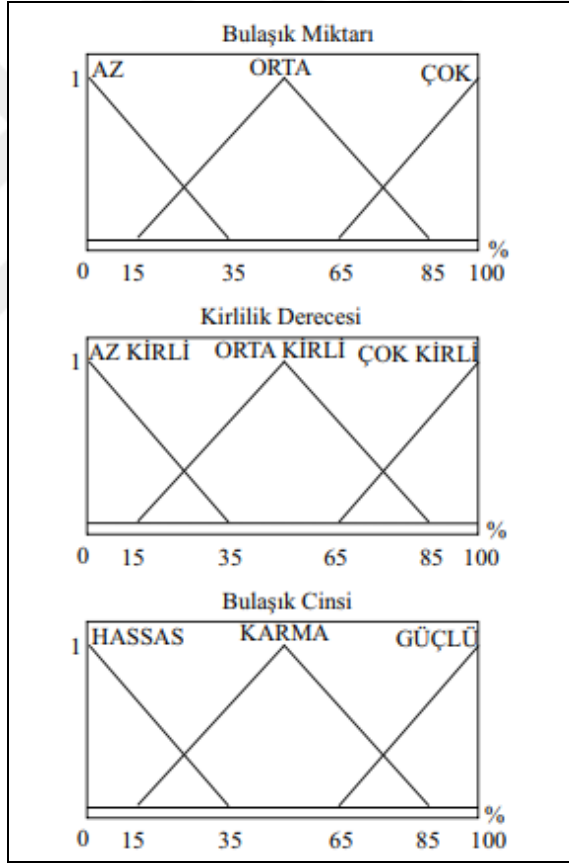
Örnek uygulama olarak Tiryaki ve Kazan'ın Bulaşık Makinesinin Bulanık Mantık ile Modellenmesi adlı makalesindeki bulanık sistem tasarımı kullanılmıştır.

Bu çalışmada bulanık mantık ile bulaşık makinesinin modellenmesi anlatılmaktadır. Bulanık mantık sistem tasarımında ilk olarak tasarımın giriş ve çıkışları belirlenir. Bulaşık makinesinin çalışmasına bakıldığında kirlerin temizlenmesi ve bulaşıkların çizilmemesi gerekmektedir. Bu amaç doğrultusunda bulanık mantık modelinin çıkışlarını oluşturan parametreleri belirlemek çok önemlidir. Bulanık mantık modelinin girişleri ve çıkışları Şekil 4.5.'de gösterilmiştir.

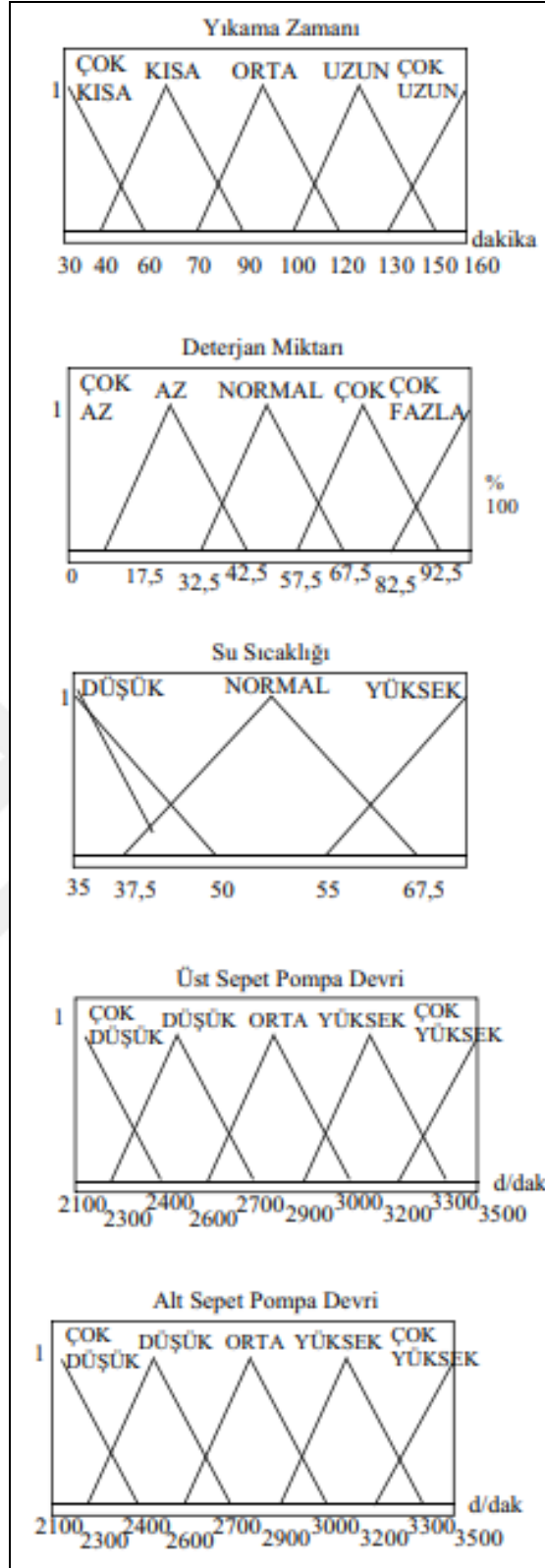


Şekil 4.5. Sistemin giriş ve çıkış parametreleri

Bu sistem tasarımıyla bulaşık miktarı, bulaşığın kirlilik derecesi ve bulaşık cinsine göre en kısa, en az enerji ve deterjan miktarı ile sistemin çalışması amaçlanmıştır. Giriş ve çıkış parametreleri sistemin daha kararlı ve amaca uygun çalışması için titizlikle belirlenmelidir. Şekil 4.6.'da sistemin giriş parametreleri gösterilmiştir. Şekil 4.7. ise çıkış parametreleri ve değerleri göstermektedir.



Şekil 4.6. Tasarımın giriş parametre değerleri



Şekil 4.7. Tasarımın çıkış parametre değerleri

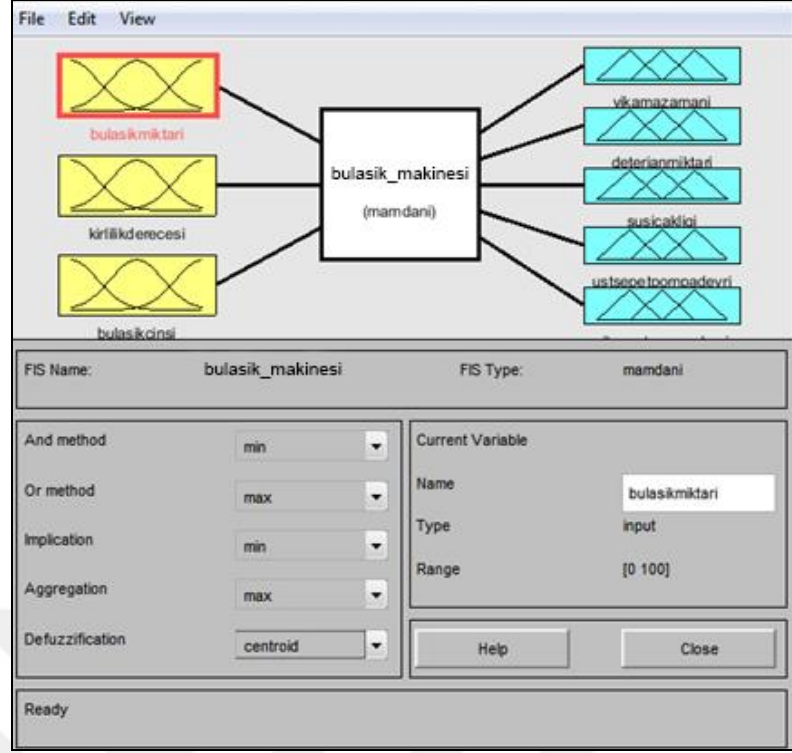
Modeli kurmak için gerekli parametrelerin üyelik fonksiyonları, alt ve üst limit değerleri belirlendikten sonra, sisteme etki eden parametreler arasında gerekli ilişkileri kurmak için 27 adet kural oluşturulmuştur (Tiryaki ve Kazan, 2007).

Çizelge 4.1. Bulanık çıkarım sisteminin kural tablosu

Girişler			Çıktılar				
Bulaşık Miktarı	Kirlilik	Bulaşık Cinsi	Yıkama Zamanı	Deterjan Miktarı	Su Sıcaklığı	Üst Sepet Pompa Devri	Alt Sepet Pompa Devri
az	az	hassas	çok kısa	çok az	düşük	çok düşük	çok düşük
az	az	karma	çok kısa	çok az	normal	düşük	düşük
az	az	güçlü	çok kısa	çok az	normal	orta	orta
az	orta	hassas	kısa	az	düşük	düşük	düşük
az	orta	karma	kısa	az	normal	düşük	düşük
az	orta	güçlü	kısa	az	yüksek	orta	orta
az	çok	hassas	kısa	az	yüksek	düşük	düşük
az	çok	karma	orta	normal	yüksek	düşük	çok yüksek
az	çok	güçlü	orta	normal	yüksek	orta	yüksek
orta	az	hassas	orta	az	düşük	düşük	düşük
orta	az	karma	orta	az	normal	orta	orta
orta	az	güçlü	orta	normal	yüksek	orta	orta
orta	orta	hassas	orta	normal	normal	düşük	orta
orta	orta	karma	orta	normal	normal	orta	yüksek
orta	orta	güçlü	orta	normal	normal	yüksek	yüksek
orta	çok	hassas	orta	normal	normal	düşük	orta
orta	çok	karma	orta	normal	yüksek	orta	yüksek
orta	çok	güçlü	orta	normal	yüksek	yüksek	çok yüksek
çok	az	hassas	kısa	az	normal	orta	yüksek
çok	az	karma	kısa	normal	normal	yüksek	yüksek
çok	az	güçlü	orta	normal	yüksek	yüksek	yüksek
çok	orta	hassas	uzun	çok	normal	orta	yüksek
çok	orta	karma	uzun	normal	yüksek	orta	yüksek
çok	orta	güçlü	uzun	çok fazla	yüksek	orta	çok yüksek
çok	çok	hassas	uzun	çok fazla	yüksek	düşük	orta
çok	çok	karma	çok uzun	çok fazla	yüksek	düşük	çok yüksek
çok	çok	güçlü	çok uzun	çok fazla	yüksek	yüksek	çok yüksek

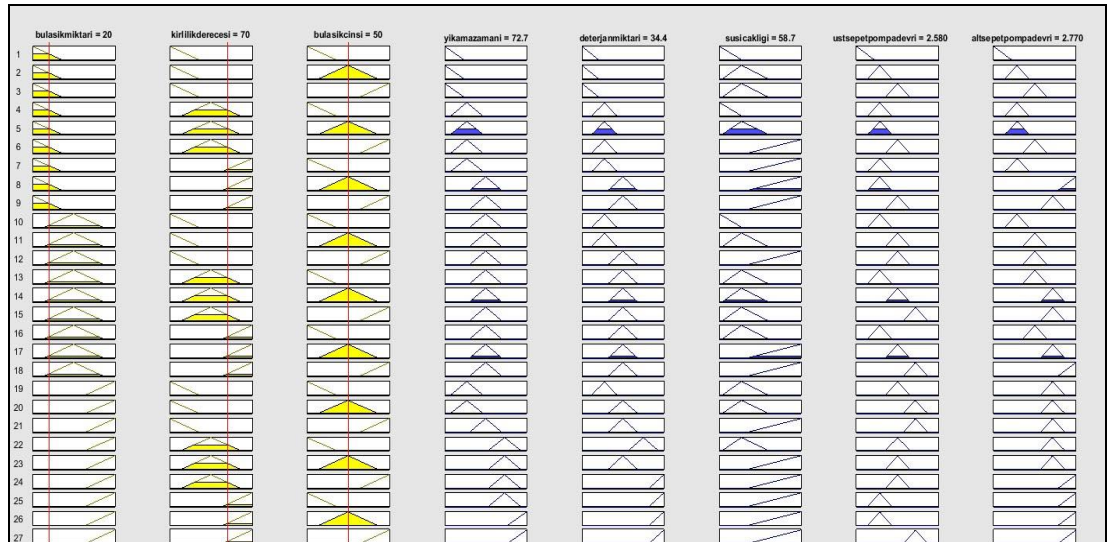
Çizelge 4.1.'deki kural tablosuna göre herhangi bir bulanık mantık sistem tasarımı yapılabilen programda sistem tasarımı yapılmıştır. Bu tez çalışmasında Matlab Fuzzy Logic Toolbox ile sistem tasarımı gerçekleştirilmiştir.

Şekil 4.8.'de Matlab Fuzzy Logic Toolbox ile bulanık çıkarım sisteminin üyelik fonksiyon editöründeki görünümü verilmiştir.



Şekil 4.8. Sistemin matlab üyelik fonksiyon editörü görünümü

Şekil 4.9.'da matlab kural görüntüleyicide giriş parametreleri ve bunlara karşılık modelin çıkış parametre değerleri gösterilmektedir.

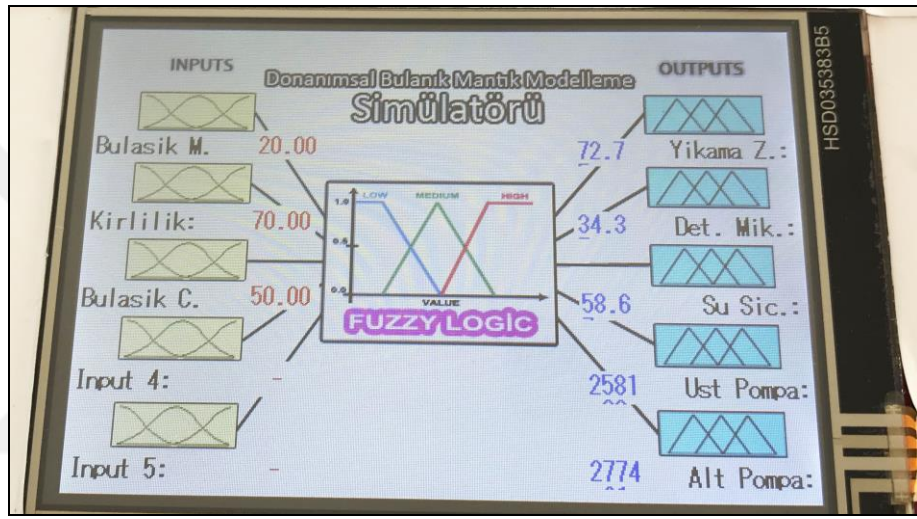


Şekil 4.9. Birinci örneğin giriş parametrelerine göre çıkış parametrelerinin kural editöründe gösterimi

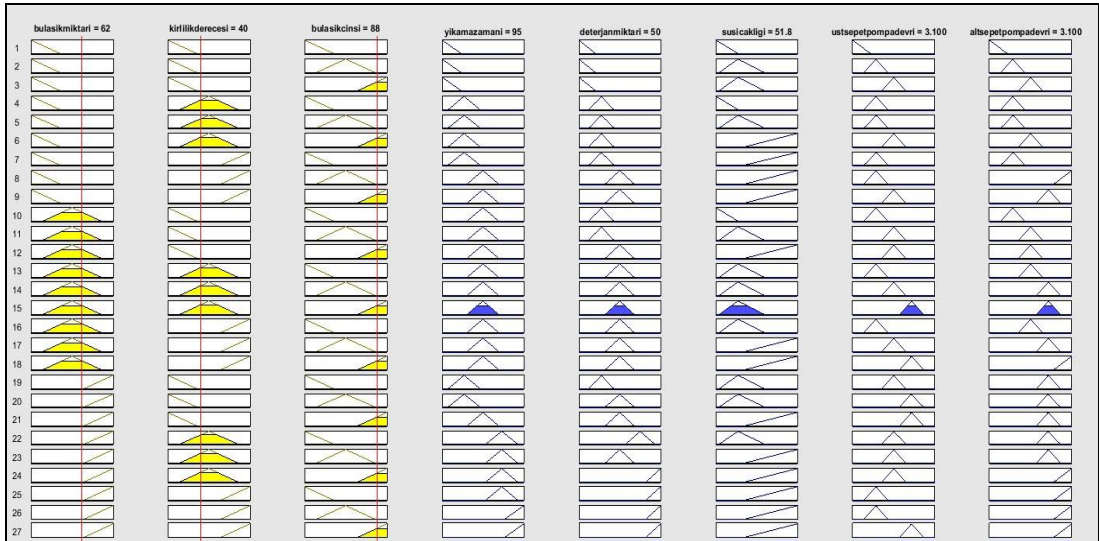
Şekil 4.9.'da giriş parametreleri bulaşık miktarı %20, kirlilik derecesi %70 ve bulaşık cinsi %50 olarak girilmiştir. Giriş parametrelerine göre çıkış parametreleri ise, yıkama

zamanı 72,7 dk, deterjan miktarı %34,4, su sıcaklığı 58,7 C<sup>0</sup>, üst sepet pompa devri 2580 dev/dk, alt sepet pompa devri 2770 dev/dk olarak gösterilmiştir.

Oluşturulan bulanık mantık tasarımının gerekli dönüşüm işlemleri yapılarak elde edilen kodun arduino UNO'ya yüklenmesi ve gerekli düzenlemelerin yapılmasıyla şekil 4.9.'daki giriş parametreleri tasarladığımız donanımsal bulanık mantık modelleme simülatorü'ne potansiyometreler vasıtasıyla girilerek, çıkışların matlab'ta oluşturduğumuz simülasyonla karşılaştırılması ve doğruluğu test edilmiştir. Şekil 4.10.'da doğruluğu test etmek için giriş parametreleri ve çıkış parametreleri gösterilmiştir.

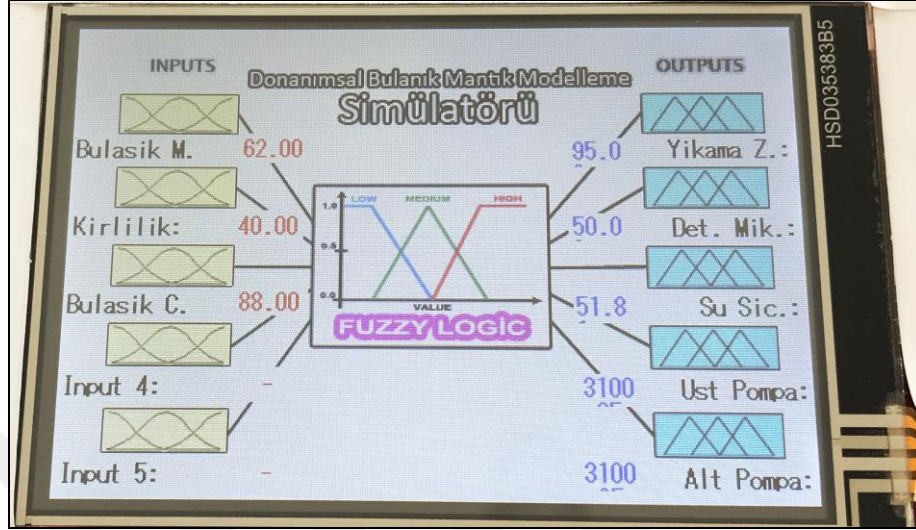


Şekil 4.10. Nextion ekranda birinci örneğin gösterimi



Şekil 4.11. İkinci örneğin giriş parametrelerine göre çıkış parametrelerinin kural editöründe gösterimi

Şekil 4.11.'de giriş parametreleri bulaşık miktarı %62, kirlilik derecesi %40 ve bulaşık cinsi %88 olarak girilmiştir. Giriş parametrelerine göre çıkış parametreleri ise, yıkama zamanı 95 dk, deterjan miktarı %50, su sıcaklığı 51,8 C<sup>0</sup>, üst sepet pompa devri 3100 dev/dk, alt sepet pompa devri 3100 dev/dk olarak gösterilmiştir.



Şekil 4.12. Nextion ekranda ikinci örneğin gösterimi

Şekil 4.12.'de örnek uygulamanın Matlab Fuzzy Logic Toolbox tasarımıdaki giriş/çıkış parametreleri ile donanımsal bulanık mantık modelleme simülatorü'ndeki giriş/çıkış parametreleri karşılaştırılmış ve sonuçların birebir aynı olduğu doğrulanmıştır.

Aşağıda arduino IDE programında giriş parametreleri için okunan değerlerin nextion HMI TFT LCD ekranda gösterimini sağlayan kod blokları gösterilmiştir. Bulaşık makinesi örneğinde 3 giriş parametresi olduğundan, potansiyometrelerden okunacak giriş parametrelerinin ekrandaki hangi text nesnesinde gösterileceği belirtilmiştir. t0.txt birinci giriş parametresinin açıklama yazısı olup "Bulaşık Miktarı" texti gönderilmiştir. analogRead(0) pininden okunan bulaşık miktarı parametresi ise t5.txt nesnesine yazdırılmaktadır. Ayrıca potansiyometrelerden ayarlanan giriş parametrelerinin map fonksiyonu ile 0-100 arasında değerler alabilmesi sağlanmıştır.

---

```
#include "fis_header.h"
// Number of inputs to the fuzzy inference system
const int fis_gcI = 3;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 5;
// Number of rules to the fuzzy inference system
const int fis_gcR = 27;
FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
// Setup routine runs once when you press reset:
void setup()
{
  // initialize the Analog pins for input.
  // Pin mode for Input: bulasikmiktari
  pinMode(0 , INPUT);
  // Pin mode for Input: kirlilikderecesi
  pinMode(1 , INPUT);
  // Pin mode for Input: bulasikcinsi
  pinMode(2 , INPUT);
  Serial.begin(9600);
  // initialize the Analog pins for output.
  // Pin mode for Output: yikamazamani
  pinMode(3 , OUTPUT);
  // Pin mode for Output: deterjanmiktari
  pinMode(4 , OUTPUT);
  // Pin mode for Output: susicakligi
  pinMode(5 , OUTPUT);
  // Pin mode for Output: ustsepetpompadevri
  pinMode(6 , OUTPUT);
  // Pin mode for Output: altsepetpompadevri
  pinMode(7 , OUTPUT);
}
```

```

void loop()
{
  // Read Input: bulasikmiktari
  g_fisInput[0] = analogRead(0);
  g_fisInput[0] = map(g_fisInput[0], 0, 1023, 0, 100);
  Serial.print("t0.txt=");
  Serial.write(0x22);
  Serial.print("Bulasik M.:");
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.print("t5.txt=");
  Serial.write(0x22);
  Serial.print(g_fisInput[0]);
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
  // Read Input: kirlilikderecesi
  g_fisInput[1] = analogRead(1);
  g_fisInput[1] = map(g_fisInput[1], 0, 1023, 0, 100);
  Serial.print("t1.txt=");
  Serial.write(0x22);
  Serial.print("Kirlilik:");
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.print("t6.txt=");
  Serial.write(0x22);
  Serial.print(g_fisInput[1]);
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
  // Read Input: bulasikcinsi
  g_fisInput[2] = analogRead(2);
  g_fisInput[2] = map(g_fisInput[2], 0, 1023, 0, 100);
  Serial.print("t2.txt=");
  Serial.write(0x22);
  Serial.print("Bulasik C.:");
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.print("t7.txt=");
  Serial.write(0x22);
  Serial.print(g_fisInput[2]);
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
}

```

```

g_fisOutput[0] = 0;
g_fisOutput[1] = 0;
g_fisOutput[2] = 0;
g_fisOutput[3] = 0;
g_fisOutput[4] = 0;

fis_evaluate();

// Set output vlaue: yikamazamani
analogWrite(3 , g_fisOutput[0]);
Serial.print("t10.txt=");
Serial.write(0x22);
Serial.print("Yikama Z.");
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("t15.txt=");
Serial.write(0x22);
Serial.print(g_fisOutput[0]);
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
// Set output vlaue: deterjanmiktari
analogWrite(4 , g_fisOutput[1]);
Serial.print("t11.txt=");
Serial.write(0x22);
Serial.print("Det. Mik.");
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("t16.txt=");
Serial.write(0x22);
Serial.print(g_fisOutput[1]);
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
// Set output vlaue: susicakligi
analogWrite(5 , g_fisOutput[2]);
Serial.print("t12.txt=");
Serial.write(0x22);
Serial.print("Su Sic.");
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("t17.txt=");
Serial.write(0x22);
Serial.print(g_fisOutput[2]);
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);

```

```

analogWrite(6 , g_fisOutput[3]);
Serial.print("t13.txt=");
Serial.write(0x22);
Serial.print("Ust Pompa:");
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("t18.txt=");
Serial.write(0x22);
Serial.print(g_fisOutput[3]);
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
// Set output vlaue: altsepetpompadevri
analogWrite(7 , g_fisOutput[4]);
Serial.print("t14.txt=");
Serial.write(0x22);
Serial.print("Alt Pompa:");
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("t19.txt=");
Serial.write(0x22);
Serial.print(g_fisOutput[4]);
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
delay(100);
}

```

Yukarıdaki kodlarda çıkış parametrelerinin nextion HMI TFT LCD ekrandaki hangi text nesnesinde gösterilmesini sağlayan kod blokları gösterilmiştir. Bulaşık makinesi örneğinde 5 çıkış parametresi vardır. Giriş parametrelerine göre bulanık çıkarım sisteminde hesaplanan duruma göre çıkış parametreleri değişmektedir. T10.txt birinci çıkış parametresinin açıklama yazısı olup “Yıkama Zamanı” texti gönderilmiştir. (analogWrite(3 , g\_fisOutput[0]) pininden okunan yıkama zamanı parametresi ise t15.txt nesnesine yazdırılmaktadır.

Giriş ve çıkış parametreleri tek tek hangi nesnede gösterileceği döngü içerisinde belirtildikten sonra değer okuma/yazma işleminin hangi sürede tekrar edeceği delay(100) komutuyla yapılır.

Bu düzenlemeler yapıldıktan sonra bulanık kontrol sistemindeki bütün kuralların yer aldığı kodlar EK A.’da gösterilmektedir. Bu kodlarda herhangi bir değişiklik yapmaya gerek yoktur.

## 5. SONUÇ

Bulanık mantık her şeyin, doğrunun da, bir derece meselesi olduğu insani akıl yürütme için bir modeldir. Temelde, sözcükle hesaplama anlamı sunmaktadır (Zadeh, 1998).

Simülatörler, benzetimi yapılan sistemlerin ya sanallaştırılması ya da donanımsal modüllerinin oluşturulması ile gerçekleştirilir. Güvenlik, maliyet, zaman, mekan gibi nedenlerden dolayı gerçek hayatta uygulanamayacak durumlarda simülasyon ile tasarımı sanallaştırmak kullanılan uygun bir yöntemdir. Özellikle eğitimde kullanılan bu tür simülasyonlarla, öğrenmek için gerçek ortama geçiş yapmadan önce bire bir gerçeğiyle denemeler yapması, uygulaması hem eğitmen, hem de eğitilen açısından daha güvenli olmaktadır. Simülasyon, sanal ortamdaki sistemin donanımsal ortamlarda görselleştirilmesi, elle tutulur hale getirilerek elde edilen sonuçların daha anlaşılır daha somut bir şekilde ortaya konulması açısından donanımsal simülatörler de kullanılmaktadır.

Bu yüksek lisans tezinde, bilgisayar ortamındaki yazılım araçlarında tasarlanmış herhangi bir bulanık kontrol modelinin, tasarlanan donanımsal bulanık mantık modelleme simülatörünün elektriksel giriş/çıkışlarında donanımsal olarak gösterimi sağlanmış ve donanım üzerinde doğrulanmıştır. Bu sayede bilgisayar ortamında sanallaştırılmış simülasyonlardan ziyade donanımsal simülatörler ile tasarlanan sistemleri gerçek hayata uygulamak daha kolay hale gelmiştir.

Çalışma kapsamında geliştirilen simülatör donanımı arduino UNO, nexiton HMI TFT LCD, giriş parametreleri için potansiyometreler, toggle switch'ten oluşmaktadır. Yazılım olarak ise, arduino IDE programıyla bulanık çıkarım sisteminin üyelik fonksiyonları ve kurallarını barındıran kodların oluşturulması ve arduino mikrodenetleyici kartına yüklenmesi sağlanmıştır. Diğer önemli yazılım ise nextion HMI TFT LCD ekranın tasarımı için kullanılan nextion editör programıdır.

Araştırma sonunda bilgisayar ortamında tasarlanan bulanık kontrol modelleri için geliştirilen donanımsal bulanık mantık modelleme simülatörü sayesinde, 5 giriş ve 5 çıkışa kadar her türlü tasarım gerekli düzenlemeler yapıldıktan sonra kolaylıkla yüklenebilmektedir.

Daha sonraki çalışmalarda hafıza sorunu yaşamamak için mikrodenetleyici kart değiştirilebilir. Nextion HMI TFT LCD ekranın boyutu büyütülebilir. Bu sayede giriş

parametreleri potansiyometre kullanmadan ekran üzerine eklenecek butonlar sayesinde girilebilecektir.



## 6. KAYNAKLAR

- V. Daniel Hant, 1986. Artificial Intelligence and Expert System Source Book, Chapman and Hall Company, s.2, New York.
- Annebel C. Beerel, 1990. Expert Systems and Artificial Intelligence An Information Manager's Guide, Library Association Publishing, s.8, London.
- Zadeh, L. A., 1998. Commercialism and Human Values, Azerbaijan International, Spring (6.1), Azerbaijan.
- Korkmaz, K., 1999. Fırçasız DC Motorların Bulanık Mantık Yöntemi Kullanarak ADSP-2101 DSP İle Kontrolü. Marmara Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi. 22-24s, İstanbul.
- Altaş İsmail H., 1999. Bulanık Mantık: Bulanıklılık Kavramı. Enerji, Elektrik, Elektromekanik-3e, Bileşim yayıncılık A.Ş., İstanbul, Temmuz, Sayı 62, ss. 80-85.
- Dereli T., 2000. Toplam Kalite Yönteminin Işığı Altında Yapay Zekanın Endüstriyel Problemlerin Çözümünde Kullanımı Semineri. Gaziantep Üniversitesi Mühendislik Fakültesi, Gaziantep.
- Oğuz, Y., 2000. Senkron Generatorlerin Uyarım Devresinin Bulanık Mantıkla Kontrolü. Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 12-20s, İstanbul.
- Şen, Z., 2001. Bulanık Mantık ve Modelleme İlkeleri. Birsen Yayınevi, 29-69, 115-129s, İstanbul.
- Sinecen M., 2002. Klima Sistem Kontrolünün Bulanık Mantık İle Modellenmesi. Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 88s, Denizli.
- Arınç, R., 2003. Bulanık Mantık Yöntemiyle Asenkron Motor Hız Kontrolü ve Simülasyonu. Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 5-21s. İstanbul.
- Sağiroğlu, Ş., Beşdok, E., Erler, M., 2003. Mühendislikte Yapay Zekâ Uygulamaları-I. Papatya Yayıncılık, Kayseri, 40-42.
- Elmas Ç., 2003. Bulanık Mantık Denetleyiciler. Seçkin Yayıncılık, 75s, Ankara.
- Baykal N., Beyan, T., 2004. Bulanık Mantık Uzman Sistemler ve Denetleyiciler. Bıçaklar Kitabevi, 103-107s, Ankara.
- Karabağ, Y., 2004. Bir Fazlı Eviricilerin Bulanık Mantık İle Denetimi. Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 61-77s, Kocaeli.
- Özek A., Sinecen M. 2004. Klima Sistem Kontrolünün Bulanık Mantık ile Modellemesi. Pamukkale Üniversitesi Mühendislik Fakültesi Mühendislik Bilimleri Dergisi Cilt 10, Sayı 3, Sayfa 353-358.
- Vardar, K., 2004. Doğru Akım Motor Hızının Bulanık Mantık İle Kontrolü. Dumlupınar Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 1-5s, 15-23s, Kütahya.

- Sarı M., Murat Y. Ş., Kırabalı M., 2005. Bulanık Modelleme Yaklaşımı ve Uygulamaları. Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü Dergisi, Sayı 9, Sayfa 2.
- Serhat Y., 2006. Bulanık Mantık ve Mühendislik Uygulamaları Ders Notları. KOÜ Yayınları Yayın No: 289, Kocaeli.
- Karadeniz S., 2006. Çoklu Kompresör Sisteminin Bulanık Mantık İle Kontrolü. Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 71s, Denizli.
- Bolat Y., 2006. Matlab-SIMULINK + PIC tabanlı bulanık mantık denetleyici tasarımı ve gerçek zamanlı sıcaklık kontrolü uygulaması. Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 101s, İstanbul.
- Soy H., 2006. Gömülü Sistem Üzerinden Bulanık Kontrol ile Ev Otomasyonunu Gerçekleştirilmesi. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 96s, Konya.
- Nakornnayok A., Tipsuwanporn V., Teanthong T., 2006. DC Motor Speed Control using Fuzzy Logic based on LabVIEW, Department of Electronics Engineering, Srinakharinwirot University, Busan, South Korea.
- Öztemel, E., 2006. Yapay Sinir Ağları, Papatya Yayıncılık, s.29, İstanbul.
- Karaosmanoğlu, G., 2007. Visual Prolog Programı ve Zeki Öğretim Sistemleri. Haliç Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 23-45s, İstanbul.
- Eğrisöğüt Tiryaki A., Kazan R., 2007. Bulaşık Makinesinin Bulanık Mantık ile Modellenmesi. Mühendis ve Makine Dergisi, Cilt : 48 Sayı: 565. Ankara.
- Işıklı Ş., 2008. Bulanık Mantık ve Bulanık Teknolojiler. Ankara Üniversitesi, Dil ve Tarih-Coğrafya Fakültesi, Felsefe Bölümü Dergisi, 19: 105-126s, Ankara.
- Taşkın M., 2009. Klasik mantık ve bulanık mantık tabanlı sıcaklık dengeleyici sistemlerin geliştirilen bir prototip üzerinden karşılaştırılması. Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi. 76s, Edirne.
- Erkaymaz H. ve Çayıroğlu İ., 2010. Bulanık mantık ve PIC kullanılarak bir klima sisteminin kontrolü. Mehmet Akif Ersoy Üniversitesi, Fen Bilimleri Enstitüsü Dergisi S:2:167-180,
- Mahmood M. S., 2010. Bulanık mantık kullanılarak trafik kontrolünün tasarımı ve uygulaması. Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 78s, Ankara.
- Taçyıldız B., 2013. Bulanık Mantık Model Geliştirme Ortamlarının Karşılaştırılması ve Örnek Bir Uygulama. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 92s, Sakarya.
- Mevlütöğlü, A., 2013. Uçuş Eğitiminde Simülatörler. Mühendis ve Makine Dergisi, Sayı:636, 17-21s. Ankara.

Munadi, M., Amirullah A., 2014. Simulation of fuzzy logic control for DC servo motor using Arduino based on MATLAB/Simulink. Intelligent Autonomous Agents, Networks and Systems (INAGENTSYS), 2014 IEEE International Conference.

RB Caldo, JT Seranilla, DJ Castillo., 2015. Design and development of fuzzy logic controlled dimming lighting system using Arduino microcontroller. Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2015 International Conference.

Nayanmani D., Pranjal Jyoti H., 2017. Liquid Level Control Using Fuzzy Logic in LabVIEW with ARDUINO. International Journal of Advance Research, Ideas and Innovations in Technology.

[www.mathworks.com/products/matlab/](http://www.mathworks.com/products/matlab/): Erişim Tarihi: 01.04.2018.

[www.arduino.cc/](http://www.arduino.cc/): Erişim Tarihi: 01.05.2018.

[http://www.makeproto.com/projects/fuzzy/matlab\\_arduino\\_FIST/index.php](http://www.makeproto.com/projects/fuzzy/matlab_arduino_FIST/index.php): Erişim Tarihi: 02.06.2018.

<http://web.itu.edu.tr/soylemezm/simux/simux.pdf>: Erişim Tarihi: 01.06.2018.

## EKLER

### EK A. Örnek Uygulama Kodlarının Devamı

---

```
//*****  
// Support functions for Fuzzy Inference System  
//*****  
// Triangular Member Function  
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)  
{  
    FIS_TYPE a = p[0], b = p[1], c = p[2];  
    FIS_TYPE t1 = (x - a) / (b - a);  
    FIS_TYPE t2 = (c - x) / (c - b);  
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);  
    if (a == b) return (FIS_TYPE) (t2 * (b <= x) * (x <= c));  
    if (b == c) return (FIS_TYPE) (t1 * (a <= x) * (x <= b));  
    t1 = min(t1, t2);  
    return (FIS_TYPE) max(t1, 0);  
}  
  
FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)  
{  
    return min(a, b);  
}  
  
FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)  
{  
    return max(a, b);  
}  
  
FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP pfnOp)  
{  
    int i;  
    FIS_TYPE ret = 0;  
  
    if (size == 0) return ret;  
    if (size == 1) return array[0];  
  
    ret = array[0];  
    for (i = 1; i < size; i++)  
    {  
        ret = (*pfnOp)(ret, array[i]);  
    }  
  
    return ret;  
}
```

```

//*****
// Data for Fuzzy Inference System
//*****
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_trimf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 3, 3, 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 5, 5, 3, 5, 5 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { 0, 0, 35 };
FIS_TYPE fis_gMFI0Coeff2[] = { 15, 50, 85 };
FIS_TYPE fis_gMFI0Coeff3[] = { 65, 100, 100 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2, fis_gMFI0Coeff3 };
FIS_TYPE fis_gMFI1Coeff1[] = { 0, 0, 35 };
FIS_TYPE fis_gMFI1Coeff2[] = { 15, 50, 85 };
FIS_TYPE fis_gMFI1Coeff3[] = { 65, 100, 100 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2, fis_gMFI1Coeff3 };
FIS_TYPE fis_gMFI2Coeff1[] = { 0, 0, 35 };
FIS_TYPE fis_gMFI2Coeff2[] = { 15, 50, 85 };
FIS_TYPE fis_gMFI2Coeff3[] = { 65, 100, 100 };
FIS_TYPE* fis_gMFI2Coeff[] = { fis_gMFI2Coeff1, fis_gMFI2Coeff2, fis_gMFI2Coeff3 };
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff, fis_gMFI2Coeff };

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { 0, 30, 60 };
FIS_TYPE fis_gMFO0Coeff2[] = { 40, 65, 90 };
FIS_TYPE fis_gMFO0Coeff3[] = { 70, 95, 120 };
FIS_TYPE fis_gMFO0Coeff4[] = { 100, 125, 150 };
FIS_TYPE fis_gMFO0Coeff5[] = { 130, 160, 160 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2, fis_gMFO0Coeff3,
fis_gMFO0Coeff4, fis_gMFO0Coeff5 };
FIS_TYPE fis_gMFO1Coeff1[] = { 0, 0, 20 };
FIS_TYPE fis_gMFO1Coeff2[] = { 12.5, 27.5, 42.5 };
FIS_TYPE fis_gMFO1Coeff3[] = { 32.5, 50, 67.5 };
FIS_TYPE fis_gMFO1Coeff4[] = { 57.5, 75, 92.5 };
FIS_TYPE fis_gMFO1Coeff5[] = { 82.5, 100, 100 };
FIS_TYPE* fis_gMFO1Coeff[] = { fis_gMFO1Coeff1, fis_gMFO1Coeff2, fis_gMFO1Coeff3,
fis_gMFO1Coeff4, fis_gMFO1Coeff5 };
FIS_TYPE fis_gMFO2Coeff1[] = { 35, 35, 50 };
FIS_TYPE fis_gMFO2Coeff2[] = { 37.5, 50, 67.5 };
FIS_TYPE fis_gMFO2Coeff3[] = { 55, 90, 90 };
FIS_TYPE* fis_gMFO2Coeff[] = { fis_gMFO2Coeff1, fis_gMFO2Coeff2, fis_gMFO2Coeff3 };
FIS_TYPE fis_gMFO3Coeff1[] = { 2100, 2100, 2400 };
FIS_TYPE fis_gMFO3Coeff2[] = { 2300, 2500, 2700 };
FIS_TYPE fis_gMFO3Coeff3[] = { 2600, 2800, 3000 };
FIS_TYPE fis_gMFO3Coeff4[] = { 2900, 3100, 3300 };
FIS_TYPE fis_gMFO3Coeff5[] = { 3200, 3500, 3500 };
FIS_TYPE* fis_gMFO3Coeff[] = { fis_gMFO3Coeff1, fis_gMFO3Coeff2, fis_gMFO3Coeff3,
fis_gMFO3Coeff4, fis_gMFO3Coeff5 };
FIS_TYPE fis_gMFO4Coeff1[] = { 2100, 2100, 2400 };
FIS_TYPE fis_gMFO4Coeff2[] = { 2300, 2500, 2700 };
FIS_TYPE fis_gMFO4Coeff3[] = { 2600, 2800, 3000 };
FIS_TYPE fis_gMFO4Coeff4[] = { 2900, 3100, 3300 };
FIS_TYPE fis_gMFO4Coeff5[] = { 3200, 3500, 3500 };
FIS_TYPE* fis_gMFO4Coeff[] = { fis_gMFO4Coeff1, fis_gMFO4Coeff2, fis_gMFO4Coeff3,
fis_gMFO4Coeff4, fis_gMFO4Coeff5 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff, fis_gMFO1Coeff, fis_gMFO2Coeff,
fis_gMFO3Coeff, fis_gMFO4Coeff };

```



```

int fis_gRO13[] = { 3, 3, 2, 3, 4 };
int fis_gRO14[] = { 3, 3, 2, 4, 4 };
int fis_gRO15[] = { 3, 3, 2, 2, 3 };
int fis_gRO16[] = { 3, 3, 3, 3, 4 };
int fis_gRO17[] = { 3, 3, 3, 4, 5 };
int fis_gRO18[] = { 2, 2, 2, 3, 4 };
int fis_gRO19[] = { 2, 3, 2, 4, 4 };
int fis_gRO20[] = { 3, 3, 3, 4, 4 };
int fis_gRO21[] = { 4, 4, 2, 3, 4 };
int fis_gRO22[] = { 4, 3, 3, 3, 4 };
int fis_gRO23[] = { 4, 5, 3, 3, 5 };
int fis_gRO24[] = { 4, 5, 3, 2, 5 };
int fis_gRO25[] = { 5, 5, 3, 2, 5 };
int fis_gRO26[] = { 5, 5, 3, 4, 5 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5,
fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9, fis_gRO10, fis_gRO11, fis_gRO12, fis_gRO13,
fis_gRO14, fis_gRO15, fis_gRO16, fis_gRO17, fis_gRO18, fis_gRO19, fis_gRO20,
fis_gRO21, fis_gRO22, fis_gRO23, fis_gRO24, fis_gRO25, fis_gRO26 };
// Input range Min
FIS_TYPE fis_gIMin[] = { 0, 0, 0 };
// Input range Max
FIS_TYPE fis_gIMax[] = { 100, 100, 100 };
// Output range Min
FIS_TYPE fis_gOMin[] = { 30, 0, 35, 2100, 2100 };
// Output range Max
FIS_TYPE fis_gOMax[] = { 160, 100, 90, 3500, 3500 };
//*****
// Data dependent support functions for Fuzzy Inference System
//*****
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;
    for (r = 0; r < fis_gcR; ++r)
    {
        int index = fis_gRO[r][o];
        if (index > 0)
        {
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else if (index < 0)
        {
            index = -index - 1;
            mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else
        {
            mfOut = 0;
        }
        fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
    }
    return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}
FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) / (FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;
    // calculate the area under the curve formed by the MF outputs
    for (i = 0; i < FIS_RESOLUTION; ++i) {
        dist = fis_gOMin[o] + (step * i);
        slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
        area += slice;
    }
}

```

```

    momentum += slice * dist;
}
return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) : (momentum / area));
}
//*****
// Fuzzy Inference System
//*****
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
    FIS_TYPE fuzzyInput2[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, fuzzyInput2, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyOutput1[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyOutput2[] = { 0, 0, 0 };
    FIS_TYPE fuzzyOutput3[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE fuzzyOutput4[] = { 0, 0, 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, fuzzyOutput1, fuzzyOutput2,
    fuzzyOutput3, fuzzyOutput4, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;
    // Transforming input to fuzzy Input
    int i, j, r, o;
    for (i = 0; i < fis_gcI; ++i)
    {
        for (j = 0; j < fis_gIMFCount[i]; ++j)
        {
            fuzzyInput[i][j] =
                (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i], fis_gMFCoeff[i][j]);
        }
    }
    int index = 0;
    for (r = 0; r < fis_gcR; ++r)
    {
        if (fis_gRType[r] == 1)
        {
            fuzzyFires[r] = FIS_MAX;
            for (i = 0; i < fis_gcI; ++i)
            {
                index = fis_gRI[r][i];
                if (index > 0)
                    fuzzyFires[r] = fis_min(fuzzyFires[r], fuzzyInput[i][index - 1]);
                else if (index < 0)
                    fuzzyFires[r] = fis_min(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
                else
                    fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
            }
        }
        else
        {
            fuzzyFires[r] = FIS_MIN;
            for (i = 0; i < fis_gcI; ++i)
            {
                index = fis_gRI[r][i];
                if (index > 0)
                    fuzzyFires[r] = fis_max(fuzzyFires[r], fuzzyInput[i][index - 1]);
                else if (index < 0)
                    fuzzyFires[r] = fis_max(fuzzyFires[r], 1 - fuzzyInput[i][-index - 1]);
                else
                    fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
            }
        }
    }
}

```

```
    }  
  }  
  fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];  
  sW += fuzzyFires[r];  
}  
if (sW == 0)  
{  
  for (o = 0; o < fis_gc0; ++o)  
  {  
    g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);  
  }  
}  
else  
{  
  for (o = 0; o < fis_gc0; ++o)  
  {  
    g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);  
  }  
}  
}
```

## ÖZGEÇMİŞ

Adı Soyadı : Hüseyin ŞENGÜN  
Doğum Yeri ve Yılı : Burdur, 1982  
Medeni Hali : Evli  
Yabancı Dili : İngilizce  
E-Posta : hsengun@mehmetakif.edu.tr



## Eğitim Durumu

Lise : Burdur Anadolu Ticaret Meslek Lisesi  
Lisans : SDÜ, Teknik Eğitim Fakültesi, Elektronik - Bilgisayar Eğitimi,  
Bilgisayar Sistemleri Öğretmenliği

## Çalıştığı Kurumlar ve Yıl

Burdur Kız Teknik ve Anadolu Teknik Meslek Lisesi	2007-2008
Burdur USO Anadolu Lisesi	2008-2009
Mehmet Akif Ersoy Üniversitesi Tefenni Meslek Yüksekokulu	2009- .....