

T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

77724

BULANIK TABANLI VE PROGRAMLANABİLİR
MANTIK KONTROLÖRLERİ İLE TRAFİK
SİNYALİZASYON SİSTEMİ

77724

Murat ŞEKER

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

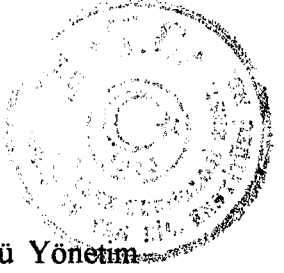
77724

TEZ DANIŞMANI

Yrd. Doç. Dr. M. Ragıp BAŞBUĞ

GEBZE

1998



Bu tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 3.7.98. tarih ve 98/174 sayılı kararıyla oluşturulan jüri tarafından Elektronik Mühendislik Anabilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

JÜRI

ÜYE
(Tez Danışmanı)

: Y. Doç. Dr. Ragıp M. Başbuğ

ÜYE

: Y. Doç. Dr. E. T. Karacuta

ÜYE

: Y. Doç. Dr. M. Halis SAKA

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 23/07/1998. tarih ve 98/20. sayılı kararı.

İmza / Mühür

ÖZET

Mikrobilgisayarların denetim sistemlerinde yaygın olarak kullanılmaya başlanması ile birlikte, kesin matematiksel tanımlamalar yerine bilgiye dayalı denetim algoritmaları ortaya çıkmıştır. Bunların belli başlıları bulanık mantık, yapay sinir ağları ve genetik algoritmalar olup son yıllarda teorik ve pratik pek çok çalışmaya konu oluşturmaktadırlar. Bu algoritmalar özellikle matematiksel modellenmesi çok güç olan doğrusal olmayan sistemlere uygulanmaktadır

Bulanık mantık işlemleri, bir problemin analizi ve tanımlanması, değişken kümelerin ve mantık ilişkilerinin geliştirilmeden bulunan bilgilerin bulanık kümelere dönüştürülmesi ve modelin yorumlanması işlemlerin oluşmaktadır. Bulanık mantık algoritması her türlü problem için uygun olmayabilir. Başka bir modelin uygun olduğu durumda bulanık mantık kullanmak istenen sonucu vermeyebilir. Bir veya birden fazla denetim değişkeninin sürekli olan, işlemin matematiksel bir modeli bulunmadığı veya bulunsa da bunu şifrelemenin zor olduğu ya da gerçek zaman işlemleri için yeterince hesaplamanın çok karmaşık olduğu durumlarda bulanık mantık uygulanabilir.

Kavşaklarda trafik denetimi yaygın olarak girişlerdeki yoğunluğa göre belirlenmiş periyotlarla ışıkların kontrolünü sağlayan mantık devresiyle yapılır. Ancak günün farklı saatlerinde girişlerdeki yoğunluk değiştiğinden, kavşağa yaklaşan araçlar sensörler yardımıyla sayılarak uygun geçiş sürelerinin sağlanması yoluna gidilmiştir.

Bu çalışmada, kavşak kontrolünde bulanık mantık yöntemlerinin kullanılabilirliği ve getirdiği avantajlar gösterilmiştir. Bu amaçla, kırmızı ve yeşil ışıklardaki trafik yoğunluğunu belirten üyelik fonksiyonları tanımlanmıştır. Kırmızı ve yeşil ışık ile çıkış üyelik fonksiyonlarının aynı olması teoride ve kural tablosu oluşturmada kolaylık sağlaması bakımından uygun görülmüştür. Her giriş için üyelik fonksiyonlarına eşittir, ek olarak uygulama esnasında gerekli görülen kurallar seçilmiş ve uygulanmıştır.

Trafik denetim uygulamasında klasik, matematiksel algoritmalar yerine Bulanık Mantık kullanılmasında ki amaç; bulanık mantığın trafik ortamının denetimi için alternatif bir metot olarak kullanılabileceğinin kanıtıdır. Bulanık denetim ile kavşağa giren araçların algılanıp buna göre sinyal değiştirmek suretiyle değişken trafik yoğunluklarının olduğu kavşaklarda daha iyi sonuçlar alınmasını sağlamaktadır.

Dört kavşak girişinde yapılan ölçümler sonucunda, trafik akışını zedelemeyen ışık yanma sürelerinin azaltılabileceği görülmüştür. çıkış üyelik fonksiyonlarının sınırlarının çok esnek alınmasına rağmen %30-35 arasında kırmızı ve yeşil yanma süresinde tasarruf sağlanmıştır. Bu sonuçlar esnek olarak kullanılsa da, bulanık mantık kurallarını, klasik metotlara göre daha avantajlı olduğu kanıtlanmıştır. Bulanık mantık uygulanarak bekleme sürelerinin daha azaltılacağı ve bundan kaynaklanan yakıt giderlerinin düşürüleceği, ülke ekonomisine katkıda bulunacağı kanıtlanmıştır.

SUMMARY

When the microcomputers started to be used in control systems, in place of using exact mathematical definitions, control algorithms which depends on information, are created. Most important ones of these algorithms are fuzzy logic, artificial neural networks, and genetic algorithms, and in recent years, they are the subjects of a lot of theoretical and practical studies. These algorithms are specially applied to the nonlinear systems whose mathematical modelling are quite difficult.

Fuzzy logic operations include the definition and analysis of a problem, the transformation of knowledge which comes from the development of variable sets and logic relations into fuzzy sets and comments about the model. Fuzzy logic algorithms may not be suitable for every type problems. When a different model is useful for the problem, the use of fuzzy logic may not give the wanted result. If one or more control parameters are variable or if there is not a mathematical model for the operation or if it is found, if the ciphering of the real time model is difficult or in the case of required calculation is too complicated, fuzzy logic may be applied to the problem.

In this study, three membership functions are defined for the density of traffic at red and green lights. These functions are low, medium and high. Because it gives simplicity, in the creation of rule table and for the theory, red light, green light and output membership functions are chosen as equal. Number of rules is equal to the subtraction of input combinations number from the number of membership functions. However, in the case of application, required rules were chosen and applied.

The reason for the use of fuzzy logic in place of classical mathematical algorithms for traffic control is to prove that the fuzzy logic can be an alternative method for the control of traffic medium. The use of fuzzy control enables the sense of the number of vehicles coming to the cross and using this information, signals are changed; so, for crosses in which variable traffic density occurs, the use of this technique gives better results. In cross traffic control, it was thought that predefined variation time and sensors which can change the on time of the lights can be used, but because the density of traffic on the cross is changing for various hours of the day, it was seen that, this not practical. For this cross, the use of fuzzy logic when the traffic density is high and the use of constant period light times for the other times was seen quite reasonable. Also, it was shown that fuzzy logic rules can be generated in a road which connects two crosses.

Taking measurements at the inputs of for different crosses, it is seen that lights on periods can be reduced without affecting the flow of traffic. Although the boundaries of the output membership functions are chosen quite flexible, on periods of red and green lights were reduced by 30-35%. Despite these results are used quite flexible, it is proved that fuzzy logic rules are more advantageous than the classical methods. With the application of the fuzzy logic, it is proved that, waiting states and energy consumptions can be reduced and, so, this results in an addition for the economy of the country.



TEŞEKKÜR

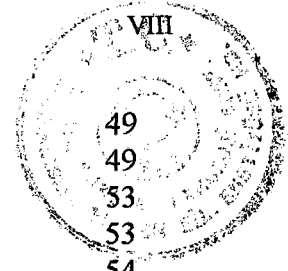
Tez konusu öyle seçilmelidir ki yanıtı bilim ve insanlık için önemli olsun. Neyin önemli neyin önemsiz olduğu konusunda bilim dünyası hem fikirdir. Yapılan tüm bilimsel çalışmaların biraz daha fazla günlük hayatta insanlara fayda sağlaması dileklerimle...

Tezin çalışmaları sırasında bana yardımcı olan, yol gösteren tez danışman Hocam Bilgisayar Bölümü Öğretim Üyesi Yrd. Doç. Dr. M. Ragıp BAŞBUĞ' a, Bölüm Başkanımız Sayın Yrd. Doç. Dr. Ertuğrul KARAÇUHA' ya ve Enstitümüze geldiği günden beri bana yol gösteren her türlü problemime yardımcı olan Sayın Hocam Bilgisayar Bölüm Başkanı Yrd. Doç. Dr. E. Murat ESİN' e, gözlemlerin yapılmasında yardımcı olan arkadaşlarım Arş. Gör. E. Selim YILDIRIM, Arş. Gör. Serkan AKSOY, katıldığım seminerlere başkanlık eden Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik Bölüm Başkanı Sayın Hocam Prof. Dr. Burhanettin CAN' a ve seminer grubu arkadaşlarım Y. Müh. Ahmet AKBAŞ ve Mustafa ONAT' a bulundukları katkılardan dolayı, Ayrıca hayat arkadaşım, nişanlım Biyoloji Bölümü Arş. Gör. Mine Gül' e bu çalışmanın yapıldığı zamanda bana göstermiş olduğu hoşgörü, anlayış ve desteğe teşekkür ederim.

Dar gelirlerine rağmen eğitimim için fedakarlık eden, bugüne kadar tüm kararlarımda beni destekleyen, manevi desteklerini her zaman yanımda hissettiğim değerli Aileme bu çalışmayı armağan ederim.

İÇİNDEKİLER

	SAYFA
ÖZET	IV
SUMMARY	V
TEŞEKKÜR	VI
İÇİNDEKİLER DİZİNİ	VII
SİMGELER VE KISALTMALAR	IX
ŞEKİLLER DİZİNİ	X
ÇİZELGELER DİZİNİ	XII
1. GİRİŞ	1
2. PROGRAMLANABİLİR MANTIK KONTROLÖRLERİNE GENEL BAKIŞ	5
2.1. Tarihçe	5
2.2. PLC' lerin Yapısı	7
2.3. Merkezi İşlem Birimi	10
2.4. Bellek	13
2.4.1. Bellek Haritası	14
2.5. Giriş/Çıkış Birimleri	16
2.5.1. Ayrık Giriş/Çıkış	17
2.5.2. Sözcük Temelli Veri Giriş/Çıkış	20
2.5.3. Analog Giriş/Çıkış	21
2.5.4. Özel Giriş/Çıkış Modülleri	21
2.6. Programlama	22
2.6.1. PLC Komutlarının Sınıflandırılması	22
2.6.2. Merdiven Dili	23
2.6.2.1. Röle Türü Komutlar	23
2.6.2.2. Veri Transfer ve Aritmetik İşlem Komutları	24
2.6.2.3. Sayma ve Zamanlama Komutları	25
2.6.2.4. Program Denetim İşlemleri	26
2.6.2.5. Giriş/Çıkış Komutları	27
2.6.3. Boolean Cebri	27
2.6.4. İşlevsel Bloklar Kullanan Diller	27
2.6.5. İngilizce Bildirim Kullanan Diller	28
2.7. PLC ile Röle Sistemlerinin Karşılaştırılması	28
2.8. PLC ile Bilgisayarlı Denetim Karşılaştırılması	30
3. BULANIK KÜME TEORİSİ	31
3.1. Üyelik Derecesi Kavramı	31
3.2. Bulanık Mantık İşlemleri	32
3.3. Bulanık Denetim	34
3.3.1. Bulanık Mantık Denetleyicilerinin Yapısı	39
3.3.2. Kuralların Oluşturulması	40
3.3.3. Bulanık Servo Denetleyici	41
3.3.4. Bulanık İşlevlerinin Bir Tablo Şeklinde Gösterimi	43
3.3.5. Bulanık Denetim Uygulama Alanları	44
3.3.6. Bulanık Denetimin Avantaj-Dezavantajları	45



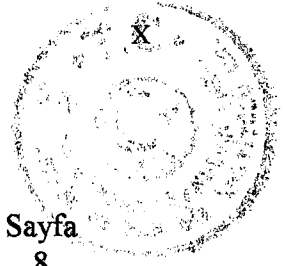
4. KAVŞAK TRAFİK DENETİM UYGULAMASI	
4.1. Trafik Kontrol Sistemleri	49
4.2. Kavşak Kontrolünün Formüle Edilmesi	49
4.2.1. Periyot Süresi Hesabı	53
4.2.2. Kayıp Zaman Hesabı	53
4.3. Kavşak Trafik Işıkları Sürelerinin Bulunması	54
4.3.1. Yeşilde Geçen ve Kırmızıda Bekleyen Araçların Ortalaması	57
4.3.2. Araçların Frekanslarının Belirlenmesi	68
4.3.3. Bulanık Alt Kümelerin Belirlenmesi	73
4.3.4. Üyelik Fonksiyonlarının Oluşturulması	78
5. SONUÇLAR	87
6. TARTIŞMA VE ÖNERİLER	94
KAYNAKLAR	96
ÖZGEÇMİŞ	97
EK-1 LSS Kavşak Programı	99
	100



SİMGELER VE KISALTMALAR

$\mu(x)$	Üyelik Derecesi
$y(t)$	Çıktı
$y(t)^*$	Yaklaşık Çıktı
K_p	Oransal Kazanç
K_i	Integral Kazancı
K_d	Türevsel Kazanç
u	Denetim Sinyali
e	Hata Sinyali
ce	Hata Değişimi
KE, KCE	Kazanç Katsayısı
g_1, g_2	Minumum Üyelik Ağırlıkları
PLC	Programlanabilir Mantık Kontrolörü
G/Ç	Giriş/Çıkış
RAM	Rasgele Erişimli Bellek
CPU	Merkezi İşlem Birimi
A.A.	Alternatif Akım
D.A.	Doğru Akım
A/D	Analog /Dijital
D/A	Dijital/Analog
TTL	Transistör-Transistör-Logic
PID	Oransal-Integral-Türevsel Kazanç
Ç.M.	Çıkarım Motoru
G	Yeşil Işık Süresi
t_s	Emniyetli Geçiş Süresi
t_R	Kavşağı Boşaltma Zamanı
t_A	Çarpışma Noktasına Gelen Araç
Δt	Kayıp Zaman
C	Periyot
m	Metre
L_A	Kavşak Giriş Mesafesi
L_R	Kavşak Boşaltma Mesafesi
X_i	Araç Değeri
N_i	Araç Frekansı
σ	Araç Sapması
$\bar{\sigma}$	Sapmaların Ortalaması
$\bar{X}$	Araç Ortalaması
$\bar{\bar{X}}$	Araç Ortalama Ortalaması
D	Düşük
O	Orta
Y	Yüksek
$K(t)$	Çıkış Fonksiyonu
$K(s)$	Kırmızıda Bekleyen Araç Sayısı
$Y(s)$	Yeşilde Geçen Araç Sayısı
K_y	Yeşil Çıkış Süresi

ŞEKİLLER DİZİNİ



Şekil	Sayfa
2.1. Programlanabilir mantık kontrolörlerin yapısı	8
2.2. İşletim sisteminin işlevleri	10
2.3. Çift mikro işlemcili PLC	11
2.4. PLC' lerde tarama	12
a) Normal tarama	
b) Anında I/O yenilemeli tarama	
2.5. Bellek türleri	14
2.6. Temel sınırları gösteren bellek haritası	15
2.7. Ayrıntılı bellek haritası	15
2.8. Çıkış elemanları için bağlantı şekli	17
a) A.A. veya d.a. besleme gerektiren elemanlar	
b) Çıkışları TTL düzeeyde olan elemanlar	
2.9. Çıkış elemanları için bağlantı şekli	18
a) A.A. veya D.A. besleme gerektiren elemanlar	
b) TTL düzeyde giriş gerektiren elemanlar	
2.10. Giriş devresinin blok şeması	18
2.11. Çıkış devresinin blok şeması	19
2.12. Tipik çıkış devreleri	19
2.13. Tipik Röle kontak çıkışı	20
2.14. BASIC türü dilde örnek merdiven diyagramı	28
2.15. Röle, hard-wired, PLC sistemlerinin göreceli maliyetleri	29
3.1. Bulanık küme ile klasik kümenin karşılaştırılması	32
3.2. Sıcaklık Giriş parametresinin üyelik derecesi	33
3.3. Bulanık denetim sistemi tanımı	35
3.4. Tipik bir bulanık kontrol sistemi için blok diyagramı	36
3.5. Uyarlamalı bulanık kontrol sistemi	37
3.6. İki araba üyelik işlevleri	38
3.7. Fren şiddetinin bulunması	39
3.8. Bir bulanık denetleyicinin temel yapısı	39
3.9. Hata (e) ve hata değişiminin dilsel değerlerinin üyelik işlevleri	41
3.10. Denetim girdisinin dilsel değerlerinin üyelik işlevleri	42
4.1. Trafik denetim sisteminin genel yapısı	50
4.2. Kavşak bilgisayara bağlantı şeması	52
4.3. Kayıp zamanlar	54
4.4. Çarpışma noktası	56
4.5. Birinci faza göre hareket eden araçların hareket yönleri	58
4.6. İkinci faza göre hareket eden araçların hareket yönleri	59
4.7. A1 ışığı için Yeşil ışıktaki geçen araçların frekansları	73
4.8. A1 ışığı için kırmızıda bekleyen araçların frekansı	74
4.9. B1 ışığı için Yeşil ışıktaki geçen araçların frekansları	74
4.10. B1 ışığı için kırmızıda bekleyen araçların frekansı	75
4.11. A2 ışığı için Yeşil ışıktaki geçen araçların frekansları	75
4.12. A2 ışığı için kırmızıda bekleyen araçların frekansı	76
4.13. B2 ışığı için Yeşil ışıktaki geçen araçların frekansları	76
4.14. B2 ışığı için kırmızıda bekleyen araçların frekansı	77
4.15. Üyelik fonksiyonu tanımı	78
4.16. K(t) çıkış üyelik fonksiyonları	78

4.17. Düşük, orta, yüksek için üyelik derecelerinin belirlenmesi	79
4.18. Düşük için üyelik derecelerinin belirlenmesi	79
4.19. Orta için üyelik derecelerinin belirlenmesi	80
4.20. Yüksek için üyelik derecelerinin belirlenmesi	80
4.21. A1 girişi için yeşil ışıktaki geçen araçların üyelik fonksiyonları	81
4.22. A1 girişi için kırmızı bekleyen araçların üyelik fonksiyonları	81
4.23. B1 girişi için yeşil ışıktaki geçen araçların üyelik fonksiyonları	81
4.24. B1 girişi için kırmızı bekleyen araçların üyelik fonksiyonları	82
4.25. A2 girişi için yeşil ışıktaki geçen araçların üyelik fonksiyonları	82
4.26. A2 girişi için kırmızı bekleyen araçların üyelik fonksiyonları	82
4.27. B2 girişi için yeşil ışıktaki geçen araçların üyelik fonksiyonları	83
4.28. B2 girişi için kırmızı bekleyen araçların üyelik fonksiyonları	83
4.29. Yeşil ışıktaki geçen araçların üyelik fonksiyonları	83
4.30. Kırmızı ışıktaki bekleyen araçların üyelik fonksiyonları	83
4.31. Kırmızı ışıktaki bekleyenler için kurallara dayanarak oluşturulmuş üyelik fonksiyonları	88
4.32. Kırmızı ışıktaki bekleyen araçlar için ağırlık merkezi	89
4.33. Yeşil ışık için kurallara dayanarak oluşturulmuş üyelik fonksiyonları	90
4.34. Yeşil ışık için üyelik fonksiyonlarının ağırlık merkezi	91
4.35. Araç sayısı ve geçme süreleri	92

ÇİZELGELER DİZİNİ

Çizelge	Sayfa
3.1. Bulandırıcı denetleyici için kural tabanı	42
3.2. Üyelik işlemlerinin bir matris şeklinde ifadesi	44
4.1. A1 girişi için araç yoğunlukları	60
4.2. B1 girişi için araç yoğunlukları	62
4.3. A2 girişi için araç yoğunlukları	64
4.4. B2 girişi için araç yoğunlukları	66
4.5. A1 girişi için yeşilde geçen araç sayısı	68
4.6. A1 girişi için kırmızı ışıktaki bekleyen araç sayısı	69
4.7. B1 girişi için yeşilde geçen araç sayısı	69
4.8. B1 girişi için kırmızı ışıktaki bekleyen araç sayısı	70
4.9. A2 girişi için yeşilde geçen araç sayısı	70
4.10. A2 girişi için kırmızı ışıktaki bekleyen araç sayısı	71
4.11. B2 girişi için yeşilde geçen araç sayısı	71
4.12. B2 girişi için kırmızı ışıktaki bekleyen araç sayısı	72
4.13. Yeşilde geçen araçların ortalamalarının ortalaması	77
4.14. Kırmızıda bekleyen araçların ortalamalarının ortalaması	77
4.15. Bulanık alt küme teorisine uygun bulanık etiketler	78
4.16. Bulanık alt küme terimleri için üyelik fonksiyonları (kırmızı için)	84
4.17. Bulanık alt küme terimleri için üyelik fonksiyonları (yeşil için)	84
4.18. Kırmızı ışıktaki bekleyen araç sayısı ile ilgili kural tablosu	84
4.19. Araç sayıları ve geçme süreleri	92

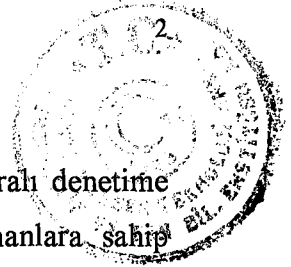
1. GİRİŞ

Mikrobilgisayarların denetim sistemlerinde yaygın olarak kullanılmaya başlanması ile birlikte, kesin matematiksel tanımlamalar yerine bilgiye dayalı denetim algoritmaları ortaya çıkmıştır. Bunların belli başlıları bulanık mantık, yapay sinir ağları ve genetik algoritmalar olup son yıllarda teorik ve pratik pek çok çalışmaya konu çalışmaya konu oluşturmaktadırlar. Bu algoritmalar özellikle matematiksel modellemesi çok güç olan doğrusal olmayan sistemlere uygulanmaktadır. Algoritmalar sistem için uygun denetim algoritmaları üretmek kadar sistemin tanımlanması amaçları ile de kullanılmaktadır. Çoğunlukla bu algoritmaların bir kaç bir arada kullanılır[1].

Klasik mantık, “Her önerme ya doğrudur ya da yanlıştır” varsayımına dayanmaktadır. Oysaki bazı önermelerin doğruluk değerleri, ölçümlerin temel sınırlamalarından dolayı belirsiz olabilmektedir. Bulanık mantık, klasik mantıkta iki önerme arasında belirsiz adı verilen üçüncü bir önermeyi ortaya koymuştur. Böylece klasik iki değerli mantığın doğru ve yanlış olan doğruluk değerleri daha esnek hale getirilmiştir.

Bulanık mantık işlemleri, bir problemin analizi ve tanımlanması, değişken kümelerin ve mantık ilişkilerinin geliştirilmeden bulunan bilgilerin bulanık kümelere dönüştürülmesi ve modelin yorumlanması işlemlerin oluşmaktadır. Bulanık mantık algoritması her türlü problem için uygun olmayabilir. Başka bir modelin uygun olduğu durumda bulanık mantık kullanmak istenen sonucu vermeyebilir. Bir veya birden fazla denetim değişkeninin sürekli olan, işlemin matematiksel bir modeli bulunmadığı veya bulunsu da bunu şifrelemenin zor olduğu ya da gerçek zaman işlemleri için yeterince hesaplamanın çok karmaşık olduğu durumlarda bulanık mantık uygulanabilir[2].

Bulanık mantıkta, değişkenlere kesin değerler atamak yerine linguistik tanımları verilir. Bulanık küme olarak ifade edilen bu tanımlardan sonuç çıkarma yapılır. Bulanık mantıkta bir şey ne tam doğru ne de tam yanlış olmaktadır. Diğer bir deyişle doğruluk derecelerine göre ayrılabilir. Bulanık mantıkta klasik mantığın aksine kısmi üyeliğe geçiş bölgesi oluşturulur[2].



Trafik ışıkları yolu ile trafik denetim işlemi zaman- tabanlı sıralı denetime örnek teşkil eder. Zaman tabanlı sıralı denetim geri besleme elemanlara sahip değildir ve çıkışın bir karşılaştırmasını yapmadığında açık döngü denetim sistemidir. Trafik denetiminde zaman röleleri ve günümüzde ise mikroişlemci tabanlı PLC (Programlanabilen Mantık Kontrolör) ler kullanılmaktadır[1].

PLC' leri sayısal ve analog giriş/çıkış modülleri yolu ile çeşitli türden makine, tesis ve süreçlerin mantık, sıralama, sayma ve denetim için gerekli aritmatiksel işlem gibi belli işlevleri yerine getirmek üzere komutların saklanması amacı ile programlanabilen, bellek kullanabilen, sayısal çalışan bir elektronik aygıttır[3].

Trafik genel olarak, taşıtlar ve yayalar olmak üzere iki temel akıştan oluşur. Bu akışlar, çeşitli nedenlerle birbirini keser ve denetlenmediğinde zaman kaybına neden olan bir karmaşa doğururlar. Bu karmaşıklığı çözmenin en etkin yöntemlerinden birisi trafik ışıkları kullanmaktır. Trafik ışıkları karmaşayı önleseler de iyi düzenlenmedikleri sürece birikmelere yol açan gereksiz duraklamalar yaratır. Bekleme kayıplarını en aza indirmek için bir çizelge yapması, en iyi kırmızı ve yeşil yanma zamanlarının optimize edilmesi gerekir[2].

Işıklar ile düzenlenen kavşaklardaki trafik akışının matematiksel tanımı uzun süreden beri araştırmacıların ilgisini çeken bir konudur. Bu amaçla sayısız model geliştirilmiş ve bunları çözmek için daha çok optimize olan, bilgisayarlı yöntemler önerilmiştir. Bunlar içinde birleşik optimize tekniklerini kullananlar, karma tamsayılı programlama kullananlar ve sistem dinamiği kullananlar sayılabilir[2].

İki trafik denetim metodu vardır. Birincisi ışıkları değiştirmek için periyodu (yeşilden yeşile) ön düzenlemede kullanılır. Diğeri, devir zamanında veya ışıklardaki değişkenliği harekete geçiren yakın duyarlı algılayıcılarla devir zamanının ön ayarlamayı birleştirir. Yeşil ışıkların düzenli devrine gereksinim duyulmayan nisbeten az gidilen cadde olayında arabaların mevcut olduğu ışıktaki değişimi harekete geçirecektir. Bu tip denetim kavşakta trafik akış modeline ait bazı ön bilgiye bağlı olup öyle ki işaret devir zamanları, yakınlık sensörlerin yerleri kavşak için

belirlenebilir. Bir kavşaktaki trafik akış biçiminin daha geniş dizisine, denetlenebilen bulanık mantık denetimi klasik denetime alternatiftir. Bulanık kontrol edilmiş trafik ışık lambası yalnızca arabaların varlığına işaret eden yakınlık sensörler yerine arabaları sayan sensörler kullanılır. Bu kavşaktaki yolların trafik yoğunluğuyla denetim sağlar ve trafik akış biçiminin değişimine ait daha iyi bir değerlendirmeye izin verir. Trafik dalgalanma dağılımları olarak bulanık denetleyici trafik ışıklarını değiştirebilir[4].

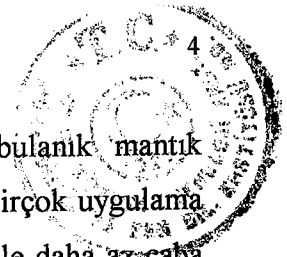
Tüm bu modeller kullanılarak hem kuramsal, hem de uygulanabilir sonuçlar elde edilmiştir. Ne var ki problemin karmaşık yapısı, bu sonuçların ancak basit kavşaklar için elde edilmesine imkan tanımıştır. Özellikle değişik yönlerden gelen birçok trafik akışının kestiği yerlerde bu modellerden birisini doğrudan kullanmak olası olmamaktadır[4].

Bu çalışma bulvardaki ışıklar arkasında oluşan taşıt birikimini en küçükleyecek şekilde bulmayı amaçlar ve bu çalışmada bulanık mantık kullanılmasıdaki amaç, trafik denetiminde bulanık mantığın kullanılmasının diğer klasik ve matematiksel algoritmalarından oluşan denetim sistemlerine kıyasla belli bir zaman aralığında trafik akış hızının daha iyi performans göstermesidir.

Klasik metotların çoğunda, kavşaklardaki trafik denetimi istatistiki verilere dayanarak ışık sürelerini belirleyip değiştirmeye giderken, bulanık denetim ise kavşağa giren araçların sayısını belirleyip buna göre sinyali değiştirmektedir. Bu ise ani değişken trafik yoğunlukları durumunda daha iyi sonuçlar alınmasını sağlamaktadır.

Bu çalışmada ayrıca kavşaklarda denetim görevini üstlenecek kontrolör elemanı olarak PLC seçilmiştir. Bulanık mantıkta elde edilen kırmızı bekleme, yeşilde geçme süresi LSS (Ladder System Support) programına aktarılmıştır. Burada ki sensörler kullanılıp araçlar yoğunlukları sürekli algılanıp optimizasyon yapılmadığı durumlarda bu sistemin kullanılması önerilmiştir. Bu şekilde dahi olsa sürelerin bulanık mantık metodu kullanılarak bulunması klasik metodlara göre %30-35 arasında fayda sağlamaktadır.

Ayrıca ülkemiz için problem haline trafik denetiminde bulanık mantık kullanımı ile denetim sistemlerine basit çözümler üretmiştir. Zaten birçok uygulama kalıcı çözüm üretmeyi beklememektedir. Bulanık mantık kullanımı ile daha az çaba ile daha iyi iş yapılabilceğı bu çalışma ile kanıtlanmıştır. Bulanık mantık mühendislik bilgisi ve deneyimini etkin bir çözüme aktarmanın tek yolu olduğı görülmüştür.



2. PROGRAMLANABİLİR MANTIK KONTROLÖRLERİNE GENEL BİR BAKIŞ



2.1. Tarihçe

Endüstriyel tesislerin denetim sistemlerinin ilk kuruluşu 19. yüzyılın sonlarında olmuştur. Bu yıllarda montaj hatlarının çeşitli kısımlarının otomasyonu için, özel olarak tasarlanmış, mekanik düzenler kullanılmıştır. 1920' li yıllardan itibaren ise röleler, kontaktörler kullanılmaya başlanmış ve böylece denetim sistemlerinde daha ileri bir düzeye ulaşmıştır[3].

Rölelerin ömürlerinin ve çalışma frekanslarının sınırlı olması gibi nedenler, anahtarlama amaçları ile kullanılabilecek yeni elemanlar aranmasına yol açmış ve 1950'li yıllarda germanyum transistörler uygulamaya girmiştir. Daha sonraki aşama, 1970'li yıllarda, ayırık elemanların yerine tümleşik elemanların alması ile gerçekleşmiş ve böylece çok karmaşık denetim sistemlerinin inşa edilebilmesi mümkün olmuştur[3].

PLC' lerin tarihçesi ise 1968 yılına, General Motors Firmasının Hydramatik Bölümünün esnek olmayan ve maliyeti yüksek röleli denetim sistemleri yerine kullanılabilecek bilgisayar temelli, esnek ve endüstrinin mühendisleri tarafından kolayca programlanabilecek ve bakımı yapılabilecek bir denetim sisteminin tasarım kriterlerini sıralamasına kadar geri gider. Sonuç olarak ortaya çıkan cihazlar, sadece rölelerin yerini almaktan çok öteye gidebilmelerine rağmen uygulama alanları tekrarlı işlemler yapan makine ve süreçlerle sınırlı kaldı. Çünkü bunlar sadece açık/kapalı denetim yapabilecek yetenekte idiler[3].

1970-1974 arasında mikroişlemci teknolojisindeki ilk gelişmelerle birlikte PLC' lerin esnekliği ve akıllığı daha da arttı. Operatörlerle etkileşim, aritmetik işlem, veri üzerinde işlem ve bilgisayarlarla iletişim gibi yetenekler PLC' lerin uygulama alanlarında yeni ufuklar açtı. Cihazların bir ekran aracılığı ile, alışagelmış röle sembolleri kullanılarak kolaylıkla programlanabilmesi mümkün oldu. Aritmetik işlem yeteneği ve daha gelişmiş komut setleri PLC' lerin nümerik data veren

sensörlerle doğrudan kullanılabilmesi ve bundan gelen verilere dayalı mantık sıralama işlemlerinin yapılabilmesini sağladı[3].

1975-1979 yılları arasında ise gerek donanım gerekse yazılım açısından yeni ilerlemeler kaydedildi. Bunlar arasında daha yüksek bellek kapasitesi, analog denetim, konumlandırma denetimi gibi yetenekler sayılabilir. Daha yüksek bellek kapasitesi daha büyük uygulama alanlarının, değişen koşullar altında değişik reçetelerin uygulanmasına olanak sağladı. Analog denetim kolaylığı ise açık/kapalı denetimin büyük bir eksikliğini ortadan kaldırdı ve analog verilerin doğrudan PLC'ye verilebilmesi ile analog denetimin PLC içerisinde açık/kapalı denetimle etkileşimli bir şekilde yapılabilmesini sağladı[3].

Konumlandırma denetimi yeteneği adımlayıcı motor çıkışı ve kodlayıcı geri beslemesi ile gerçekleştirildi. Böylece süreç denetiminde gerekebilecek bir başka yetenekte sağlanmış oldu[3].

Donanımda kaydedilen bu ilerlemeler yazılımda da paralel ilerlemeleri gerektirdi ve analog denetim konumlandırma denetimi gibi yeteneklerin kolaylıkla kullanılabilmesini sağlayacak, bilgisayar dillerine benzer diller geliştirildi[3].

1980'li yıllarla birlikte PLC endüstrisinde yeni teknolojik ilerlemeler kaydedildi. “*Bit - Slice*” teknolojisinin kullanımı ile daha hızlı bir tarama (scan) yapılabilmesini çok az sayıda röle kullanan sistemler yerine kullanılabilecek PLC'lerin piyasaya sürülmesi, mikro işlemci temelli akıllı giriş/çıkışla dağıtılmış işleme, ısı ve basınç ölçen sezicilerin doğrudan PLC'ye bağlanabilmesine olanak veren arayüzler donanım açısından kaydedilen ilerlemelere birer örnek olarak sayılabilir. Diğer önemli bir ilerlemede çeşitli denetleyicilerin ortak bir yol üzerinden bağlanabilmelerine olanak sağlayan aile yaklaşımı oldu; böylece ufak sistemlerden başlayarak, zamanla, denetleyici sistemi büyütme olanağı oldu. Günümüzde ileri düzeyde ki PLC sistemlerinin 8000 G/Ç (Giriş/Çıkış) noktası, 128 k'ya varan belleği vardır[3].

Yazılım açısından kaydedilen gelişmeler arasında ise BASIC gibi yüksek düzeydeki dillerin bazı denetleyicilerde kullanılabilmesi, diagnostik olanaklarının

sistem düzeyinden süreç düzeyine çıkarılması (sadece sistem çalışmasında değil süreçteki çalışma hatalarının da bulunabilmesi), kayan nokta aritmetiğinin kullanılabilmesi sayılabilir[3].

Geçmişte kaydedilen bu hızlı gelişmelerin gelecekte de sürmesi doğaldır. Donanım açısından beklenebilecek gelişmeler arasında kalıcı (nonvolatile) **RAM** (random access memory, rasgele erişimli bellek), çeşitli PLC' lerde kullanılabilecek programlama birimleri, fiber optik iletişim arayüzü ve daha hızlı tarama sayılabilir.

Gelecekte denetim filozofisi açısından da bazı değişikliklerin olması kaçınılmazdır. Denetim stratejisinin merkezi olmak yerine dağıtılmış bir biçimde olması , PLC' lerin robotlar, nümerik denetim ve CAD/ CAM sistemleri ile birlikte tümleşik bir biçimde kullanılması , birden fazla PLC' nin hiyerarşik biçimde çalışması beklenebilecek gelişmeler arasında sayılabilir.

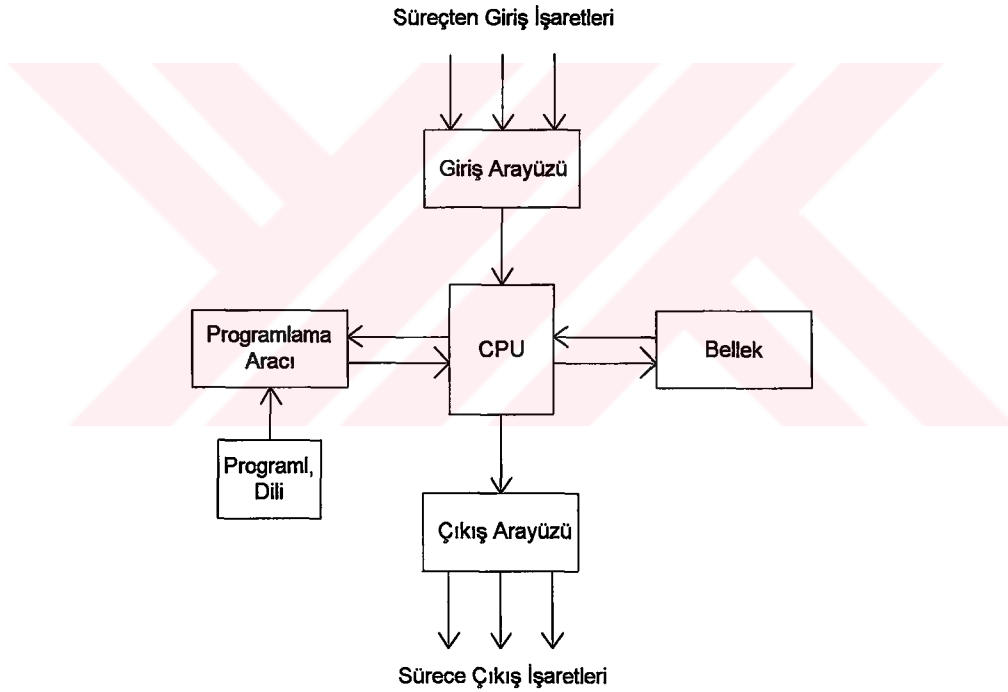
2.2. PLC' lerin Yapısı

Genel olarak bir PLC' de, belleğindeki programın akışı içinde, girişleri (ki bunlar denetlenecek sistemin denetim veya geri besleme işaretleridir) okuyup programda istenen denetim işlemleri üreten ve çıkışlara yazan özel amaçlı bir mikrobilgisayardır. Amerika' nın *National Electrical Manufacturers Association* (NEMA) programlanabilir mantık kontrolörleri şu şekilde tanımlar : Bir PLC, makina ve süreçleri denetlemek için mantık, sıralama, zamanlama, sayma ve aritmetik gibi bazı belirli işlevleri gerçekleştirmeyi sağlayacak komutların depolandığı programlanabilir bir belleği olan bir sayısal araçtır[3].

Bütün PLC' ler, büyüklük, karmaşıklık ve fiyat gibi faktörlerden bağımsız Şekil 2.1.' de gösterilen bazı temel birimlerden oluşurlar. Bunların bazıları donanım birimleridir, bazıları ise PLC yazılım ve programlarının işlevsel özelliklerini yansıtır. Bütün PLC' lerde güç besleme sistemine endüstriyel tesisin fiziksel ve elektriksel ortamına uygun muhafazaya ek olarak şunlar bulunur :

- * Giriş arayüzü
- * Merkezi işlem birimi - CPU
- * Çıkış arayüzü
- * Programlama dili
- * Programlama aracı

Giriş arayüzü, denetlenen makine veya sürece bir bağlantı sağlar. Ana işlevi, dış dünyadan gelen, kontak kapanması , analog süreç gerilim ve akım değerleri, v.b. işaretleri almak ve CPU' nun kullanacağı bir formata sokmaktır. Modüler bir yapıda olup daha fazla giriş yapılması gereken durumlarda modüller eklenerek genişletme yapılabilir. Eklenebilecek modül sayısının sınırı CPU' ya ve belleğe bağlıdır.



Şekil 2.1. Programlanabilir denetleyicilerin yapısı

CPU ve bellek PC' ye zeka veren birimlerdir. Çalışma şekilleri, en basit anlamda aşağıda belirtilenlerin, sıralı olarak, tekrar tekrar yapılmasından ibarettir.

1. Giriş arayüzündeki bilgileri inceleyerek denetlenen sürecin durumunu gözle.
2. Bu bilgileri kullanarak bellekte saklı uygulama programını yürüt.



3. Bir denetim işlevi gerekiyor mu kara ver.
4. Gerekiyorsa, çıkış arayüzüne işaret göndererek denetim işlevini gerçekleştir.
5. Birinci adıma git ve devam et.

Çıkış arayüzünün işlevi ise giriş arayüzünün tam tersidir. Yapısı onunki gibi modüler olup CPU' dan gelen işaretleri alır ve onları valf, ısıtıcı , pompa gibi dış birimler tarafından gerçekleştirilecek denetim işlevlerine uygun bir formata sokar.

- * Giriş arayüzünün terminallerine bağlı bir devrede A.A. (Alternatif Akım) güç olup olmadığını sezmek.
- * Çıkış arayüzünün terminallerine bağlı devrelere A.A. güç uygulamak veya kesmek.
- * D.A. (Doğru Akım) devreler için benzer sezme ve anahtarlama işlemleri yapmak.
- * Yüksek hızlı giriş darbelerini kullanarak önceden ayarlı düzeylerden aşağı veya bu yüzeylere kadar saymak.
- * Sürecin çeşitli noktalarından denetim bilgilerini alıp bunları CPU' ya uygun bir formata sokmak.
- * Termokulp , basınç sezicisi gibi çeşitli birimlerden işaret alıp bunları CPU' ya uygun bir formata sokmak.
- * Belirli bazı süreç düzeylerinde dış birimleri denetleyecek işaretleri üretmek.
- * ASCII kodunda iletişim sinyallerini almak ve üretmek.
- *Mil kodlayıcısı , “thumbwhell” anahtarlar tarafından üretilen işaretlerdir veya bilgiler gibi birden fazla bitten oluşan işaretleri almak.

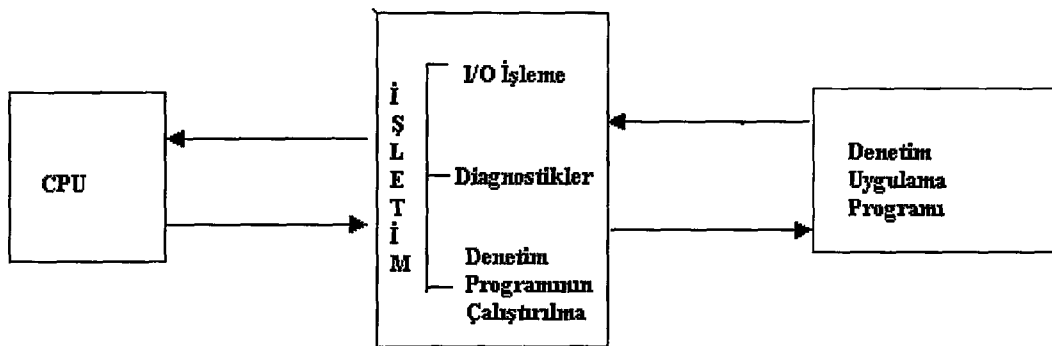
Kullanıcı tarafından yazılan ve bellekte saklanan program, verilen bir süreç durumu için doğru denetim çıkış işaretlerinin üretilmesini sağlar. Bu programın yazılış biçimi çeşitli türlerden biri olabilir. İlk olarak ortaya atılan dil, röleli denetim sistemlerinde kullanılan merdiven diyagramlarına benzer bir biçimde idi. Bunlar hala, zamanlama işlevlerinde kullanılan bloklar gibi bazı ek kolaylıklarla birlikte popüler bir şekilde kullanılmaktadır[3].

Merdiven diyagramı türü dile benzer diğer bir dilde denetim yönetiminin boolean dilinde gösterilmesidir. Daha sonra gelişmeler BASIC' e benzer üst düzey dilleri yönünde olmuştur.

Son olarak programlama aracı, programlayıcı ile PLC arasındaki bağlantıyı sağlar. En basit şekli, PLC' nin bir yanındaki bir tuş takımı veya PLC' ye bir kablo ile bağlanan, elde tutulan, yine bir dizi tuşlardan oluşturulmuş bir ünedir. Daha gelişmiş şekilleri ekran ve klavye kullanılır ve böylece programın daha iyi bir şekilde görülmesini ve otomatik program dökümantasyonunu sağlarlar. Bazı PLC' ler ise bir üst düzey denetleyici bilgisayar tarafından programlanacak şekilde yapılandırıldı[3].

2.3. Merkezi İşlem Birimi

Merkezi İşlem Birimi , bellekle birlikte bir PLC' ye zeka veren kısmı olup bellekte saklı programın gerektirdiği bütün aritmetik, mantık ve veri işleme gibi operasyonlar bu birimde gerçekleştirilir. Kalbinde mikroişlemci denilen, çok yüksek oranda tümleşik devre (VLSI- Very Large Scale Intergation) teknolojisinin bir ürünü yer alır. Burada bir PLC' deki mikroişlemcinin gücünün iletişim sistemi denilen bir programla belirlendiğini vurgulamalıdır. İki ayrı yapımcı aynı işlemciyi kullanabilir, ama değişik işletim sistemleri nedeniyle sonuç PLC' lerin yetenekleri farklı olabilir. İşletim programı yapımcı tarafından hazırlanmış olup genelde PLC kullanıcısına açık değildir. Şekil 2.2' de bu programın genel işlevleri ve denetim uygulama programı ile ilişkisi gösterilmiştir[3].



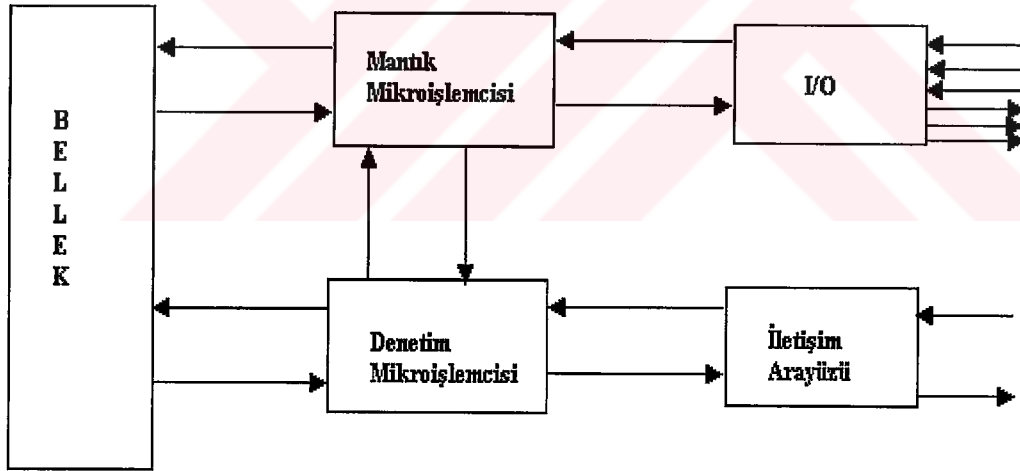
Şekil 2.2. İşletim sisteminin işlevleri.

İşletim sistemi diyagnostik prosedürleride içerir. Bu prosedürler genelde ilk başlangıçta kendinden çalışan veya kullanıcı tarafından çalıştırılan olmak üzere iki temel türe ayrılır. Birinci tür diyagnostikler genelde, sadece işletim sisteminin

kullandığı belleği test ederler. Kullanıcı tarafından çalıştırılan diyagnostikler ise bütün belleği ve iletişim kapıları gibi diğer unsurları teste tabi tutarlar.

G/Ç alt sistemler ile iletişimin doğru yapıp yapılmadığını test eden sistem prosedürleride vardır.

Yukarıda belirtilen bütün operasyonlar bazı PLC' lerde bir tek mikro işlemci ile gerçekleştirilir. Günümüzde daha yaygın olan yaklaşım sistemin görevlerini birkaç mikro işlemci arasında bölmektir. Bu yaklaşımda (Multi- Processing) her bir mikro işlemcinin belirli görevleri vardır. Şekil 2.3' de yaygın olarak kullanılan çift mikro işlemcili bir yaklaşım gösterilmiştir. Burada mikro işlemcilerden biri denetim diğeri ise mantık işlemini üstlenir. Denetim mikro işlemcisi denetim çevrimindeki denklemler operatörler ile etkileşim gibi daha karmaşık hesap ve veri işlemesi ile ilgili işlemleri yapar, diğeri ise zamanlama, mantık ve sayma gibi işlemlerle birlikte uygulama programının taramasını gerçekleştirir. Programın bir yerinde kendi yeteneğinin üstüne çıkan bir işlemle karşılaşırse bunu denetim mikro işlemcisine aktarır ve taramaya devam eder.

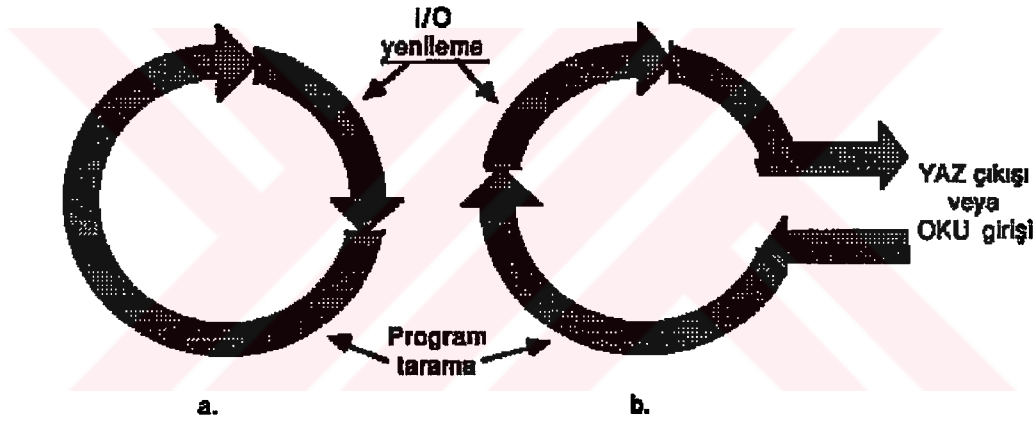


Şekil 2.3. Çift mikro işlemcili programlanabilir denetleyici.

Son yıllarda ortaya çıkan yeni bir gelişmede akıllı G/Ç arayüzleridir. Bu G/Ç' ların bir mikro işlemcisi ve belleği ve de uygulama programına bağlı olmayan bir mini işletim programları vardır. Böyle akıllı bir modüle örnek olarak PID (Oransal-Integral-Türevsel) denetim modülü gösterilebilir.

Uygulama programının yürütülmesi sırasında CPU tüm girişleri okur, denetim mantığının gerektirdiği işlemleri yaparak çıkış terminallerine güç verir veya keser. Bu okuma, programı gerçekleştirme ve çıkışların durumlarını buna göre yineleme işlemine “ TARAMA ” adı verilir. Şekil 2.4.a’da bir PLC taraması sembolik olarak gösterilmiştir. Bir tarama için gerekli süre 1-100 μ s arasında değişebilir. Yapımcılar bu süreyi genellikle uygulama programının kullandığı bellek miktarına bağlı olarak verirler (örneğin 5 ms/1 K program)[3].

Normal olarak tarama, girişlerin durumunu okuma, denetim denetim mantığını gerçekleştirme ve çıkışları yenilemeden oluşan sürekli ve sıralı bir süreçtir. Ama bazı kritik durumlarda belirli bir G/Ç yenilemesini, taramanın sonunu beklemeden, anında yapılması gerekebilir. Bazı PLC’ lerde yazılımdaki özel komutlarla bu işlem, şekil 2.4.b’ de gösterilen biçimde, gerçekleştirilebilir.



Şekil 2.4. PLC’ lerde tarama.

a. Normal tarama.

b. Anında I/ O yenilemeli tarama.

Tarama hızı PLC seçiminde önemli bir parametredir. Örneğin sürecimizde 5 ms içerisinde iki defa durum değiştiren işaretler var ise ve programda bunların izlenmesi gerekiyorsa 10 ms’ lik tarama hızı olan bir PLC kullanılamaz, çünkü tarama sırasında bu işaret durumu değiştirip, yine eski durumuna örneğin 1 iken 0 ve tekrar 1’ e dönmüş olabilir. Böyle bir durum tarama hızı daha süratli olan bir PLC kullanımını veya özel bir G/Ç modülünü gerektirir[3].

2.4. Bellek

Bellekte aşağıda belirtilen dört ayrı tür bilgi yer alır.

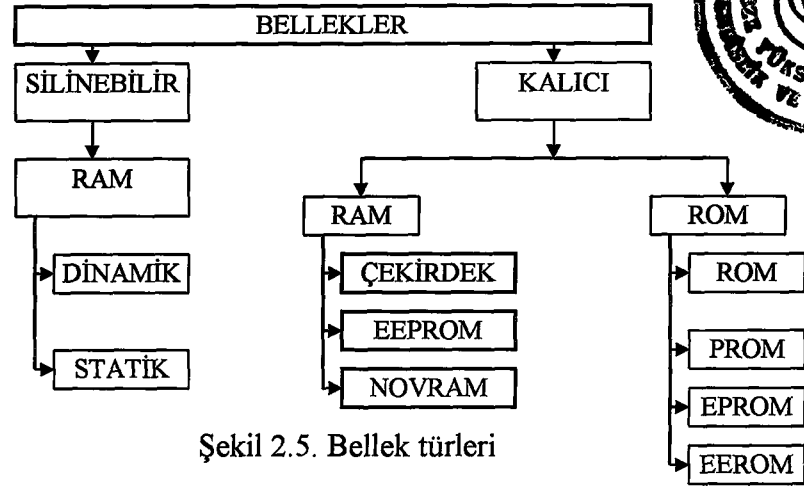
1. Yönetim programı veya işletim sistemi: İşletim sistemi genel amaçlı bir mikrobilgisayarı, bir PLC haline getiren programdır. Çevre birimleri ile iletişimi sağlar, denetim uygulama programını yürütür, genel ev bakımı (housekeeping) işlemlerini gerçekleştirir. CPU ve G/ Ç gibi PLC' nin asli bir kısmıdır.

2. Denetim uygulama programı: Elimizdeki belirli denetim işlemi için yapılmış programdır. Merdiven diyagramı şeklinde veya daha üst düzey bir dilde yazılmış olabilir.

3. Veri tabloları: Bu alanda denetim işleminin gerçekleştirilmesinde kullanılacak denklem sabitleri, zaman ve sayma işlemleri için ön değerler uygulama programının kullanacağı diğer sabitler gibi veriler depo edilir. Burada ayrıca son olarak okunan sistem girişleri ve son olarak dışarı verilen sistem çıkışları da bulunur.

4. Yaz boz alanı: Burada program tarafından kullanılacak geçici değerler bulunur. Ana bellek (veri tablolarının bulunduğu yer) yerine buranın kullanımı bu verilere çok daha kısa bir sürede ulaşılmasını sağlar.

Yukarıda açıklanan çeşitli bilgi türleri, özellikleri dolayısıyla daha değişik bellek türleri gerektirirler. Örneğin işletim sistemi devamlı kalmalı, güç kesilmelerinden etkilenmemelidir. Fakat böyle bir bellek uygulama programı için uygun olmayabilir, çünkü kullanıcı bu programı kolaylıkla değiştirebilme olanağına sahip olmak ister[3].



Şekil 2.5. Bellek türleri

Bellek türleri, şekil 2.5’de gösterildiği üzere uçucu ve kalıcı olmak üzere iki genel sınıfa ayrılırlar. İsimlerden de anlaşılabilceği üzere uçucu bellekteki bilgiler, bellek beslemeden yoksun kalırsa kaybolurlar. Eğer bu bilgilerin kaybolması istenmiyorsa bir yedek (Backup) pil kullanılmalıdır. Kalıcı belleklerde ise bilgi kalıcıdır, besleme gerilim kesildiğinde kaybolmaz[3].

2.4.1. Bellek Haritası

Bellek haritaları belleğin, çeşitli işlevler için kullanılan kısımlarını belirterek coğrafyasını gösterir. En basit olarak bellek, şekil 2.6’da gösterildiği gibi, iki kısma ayrılır.

- 1.) İşletim sistemi tarafından kullanılan alan
- 2.) Denetim uygulaması için kullanılabilecek alan veya bellek alanı

Birinci kısımda işletim sistemi programına ek olarak, bu program tarafından ara sonuçları ve çeşitli geçici bilgileri saklamak için kullanılacak bir yaz -boz (scratch) alanı bulunur. İkinci kısım ise uygulama programına ayrılmış olup burada denetim komutları veri tabloları saklanır[3].

İŞLETİM SİSTEMİ
YAZ-BOZ ALANI
UYGULAMA PROGRAMLARI

Şekil 2.6. Temel sınırları gösteren bellek haritası

Şekil 2.7’ da daha ayrıntılı bir bellek haritası gösterilmiştir. Görülebileceği üzere yukarıda açıklanan işletim sistemi ve yaz - boz alanlarının kapsadığı bölümlere ek olarak iki temel bölge vardır, bir bölge uygulama programının kendisine diğer bölge ise veri depolanmasına atanmıştır. Veri depolanan bölge kendi içerisinde işlevsel bazı ek bölgelere ayrılmıştır[3].

İşletim Sistemi
Yaz – Boz Alanı
Bir Temelli Veri Tablosu
Sözcük Temelli Veri Tablosu
İç Bit Temelli Veri Tablosu
İç Sözcük Temelli Veri Tablosu
Sabitler
Değişkenler
Uygulama Programları

Şekil 2.7. Ayrıntılı bellek haritası.

Bit Temelli Veri Tablosu: Bu tabloda her bir sayısal giriş (örneğin limit anahtarları, alçak/yüksek, düzey/basınç sensörleri v.b.) ve çıkış (örneğin lamba, kontaktör v.b.) için bir bit ayrılmıştır. Örneğin 64 tane bit temelli giriş ve çıkışımız varsa bu tabloda o kadar bitliktir. Taramanın G/Ç yenileme kısmından sonra bu tablodaki bütün bit giriş ve çıkışların durumuna göre 1 veya 0 yapılırlar[3].

Sözcük Temelli Veri Tablosu: Bu tabloda birden fazla bitlik giriş ve çıkışlar bulunur. Bunlar çeşitli formatlarda kodlanmış olurlar. Örneğin numarator

(Thumbwheel) anahtarlardan gelen işaretler BCD kodludur buna karşın A/D (Analog/Dijital) çeviricilerden gelenler ikili veya işaretli ikili olabilir. Çıkışlara örnek olarak da hız ayarlayıcılarına ve çok parçalı (Multiple Segment) göstergelere giden işaretleri verebiliriz. Bunlardan ilki genellikle olağan ikili bir işaret olup önce bir D/A (Dijital/Analog) dan geçer sonra da hız ayarlayıcısına gider. İkinci tür çıkış ise BCD formatta göstergeleri doğrudan sürmekte kullanılır[3].

İç Bit Temelli Veri Tablosu: Burada programının başka kısımları için giriş olarak kullanılan iç çıkışların durumu saklanır. Bu bitler iç çıkışlar veya iç röleler olarak bilinir, ve elektromekanik denetleyicilerdeki denetim rölelerinin görevini görürler ve hiçbir dış elemanı kumanda etmezler[3].

İç Sözcük temelli Veri Tablosu: İç temelli veriler gibi bunlarda tamamı ile iç amaçlar için kullanılacak çeşitli sözcük temelli çıkışlardır[3].

Sabitler Ve Değişkenler: Bunlar zamanlayıcı ve sayıcı için öndeğerler karşılaştırma işlemleri için referans değerler gibi sabitlerde, zamanlayıcı ve sayıcıların o andaki değerleri, matematik işlemlerin sonuçları gibi değişkenlerdir[3].

Belleğin geri kalan kısımları uygulama programına ayrılmıştır. Süreci veya makineyi denetlemek için PLC' ye girilen bütün komutlar burada saklanır. İç veya dış giriş veya çıkışları tanımlayacak bütün adresler burada yazılıdır. Program yürütüldüğü zaman CPU işletim programının denetimi altında bu alanın içeriğini çözümler ve veri alanındaki bit sözcükleri kullanacak gerekli denetim kararlarını verir. Bu kararlar bit ve sözcükleri kullanacak gerekli denetim kararlarını verir. Bu kararlar bit ve sözcük temelli veri alanlarında ki çıkış tablolarına aksettirilir, ve çıkış arayüzü bu yenilenen bellek içeriklerine göre dış dünya için denetim işaretlerini üretir[5].

2.5. Giriş/Çıkış Birimleri

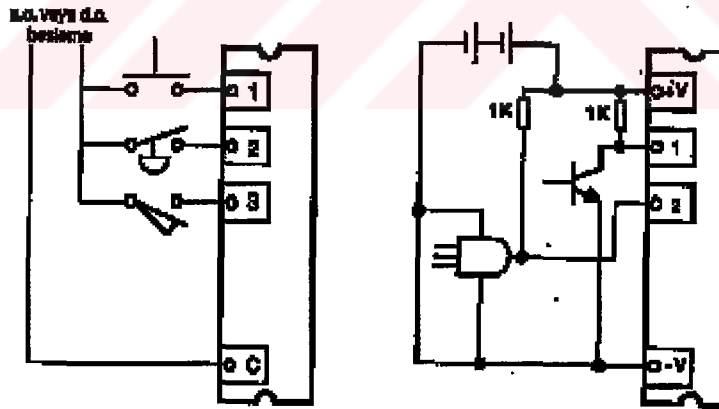
Giriş/çıkış birimleri CPU' nun dış dünya ile bağlantısını sağlar, önceleri yalnız ayrıık açık kapalı elemanlar için arayüzler vardı ve PLC' lerin denetim işlevleri bu nedenle oldukça sınırlı kalmıştı. Günümüzde ise düşünülebilecek hemen hemen her tür G/Ç elemanının CPU ile bağlantısını sağlayabilecek bir arayüz vardır, bu

arayüzlerden bazılarına, örneğin PID bağlarına, zekada eklenmiştir. Böylece CPU' nun görevleri hafifletilmiş, dolayısıyla da tarama hızı artırılmıştır.

2.5.1. Ayrık Giriş/Çıkış

En yaygın olarak kullanılan arayüz türü ayrık G/Ç arayüz olup CPU' nun "1" veya "0" türü çıkışı olan giriş elemanları veya "1" veya "0" türü bir çıkış ile denetlenecek çıkış elemanları ile bağlantısını sağlar. Giriş elemanlarına örnek olarak puşbuton anahtarlar, limit anahtarları, düzey anahtarları, motor kontaktör veya röle kontakları, seçici anahtarlar, fotoelektrik gözler sayılabilir. Ayrık G/Ç arayüzü ile denetlenebilecek çıkış elemanları için ise alarmlar, denetim röleleri, selonoidler, motor starterleri vb. birer örnek olarak gösterilebilir[3].

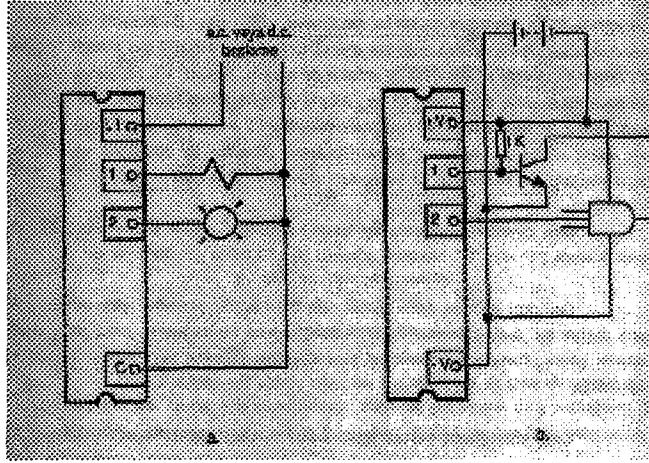
G/Ç elemanlarının çeşitli düzeyde gerilimlerle beslenmesi gerekebileceğinden G/Ç arabağlarının nominal gerilim (A.A. veya D.A.) değerleride çeşitlilik gösterir. Şekil 2.8.'de A.A. veya D.A. besleme gerektiren giriş elemanları ile TTL düzeyde çıkış veren elemanların arabağa bağlantı şekli gösterilmiştir. Şekil 2.9' da ise çıkış elemanlarının arabağ bağlantılarının tipik olarak nasıl yapılacağını göstermektedir[3].



Şekil 2.8. Çıkış elemanları için bağlantı şekli.

- a. A.A. veya d.a. besleme gerektiren elemanlar
- b. Çıkışları TTL düzeyde olan elemanlar

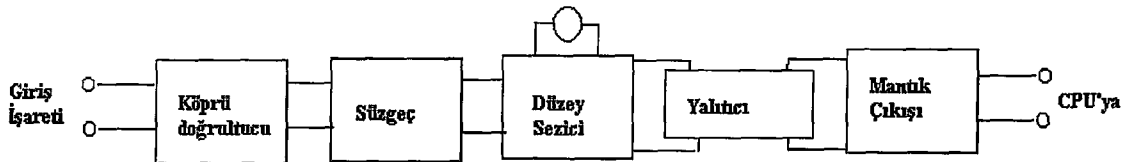
Gerek girişlerde, gerekse çıkışlarda güç ile elektronik kısmın elektriksel olarak izolasyonu, CPU' nun güç kısmında oluşabilecek çeşitli geçici olaylardan etkilenmemesi ve zarar görmemesi için gereklidir[3].



Şekil 2.9. Çıkış elemanları için bağlantı şekli.

- a. A.A. veya D.A. besleme gerektiren elemanlar
- b. TTL düzeyde giriş gerektiren elemanlar

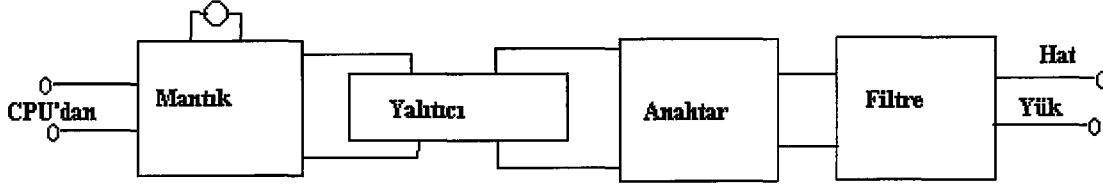
Şekil 2.10.'de giriş devresinin blok şeması gösterilmektedir. Giriş işareti bir köprü doğrultucudan geçirildikten sonra güç tarafındaki elektriksel gürültülere ve işaret titremesine karşı koruma sağlayan bir süzgeçten geçer. Bu süzgeç işarete tipik olarak 10 -25 milisaniye kadar bir gecikmeye neden olur. Gecikmenin tam değeri yapımcıdan yapımcıya , arayüz türünden arayüz türüne göre değiştiği gibi işaretin "0" dan "1" e yoksa "1" den "0" düzeyine indiğine de bağlıdır. "0" dan "1" e geçişlerde gecikme genellikle daha azdır. Eğer işaret en az süzgeç gecikmesi süresi kadar bir süre belirli bir düzeyi geçer ve bu durumda kalırsa düzey sensör bu işareti geçerli bir "1" giriş olarak algılar ve çıkışını bir yapar. Bu işaret genellikle bir optoküplör veya darbe transformatöründen oluşturulan bir yalıtıcıdan geçerek CPU' ya girer[3].



Şekil 2.10. Giriş devresinin (A.A. veya D.A.) blok şeması

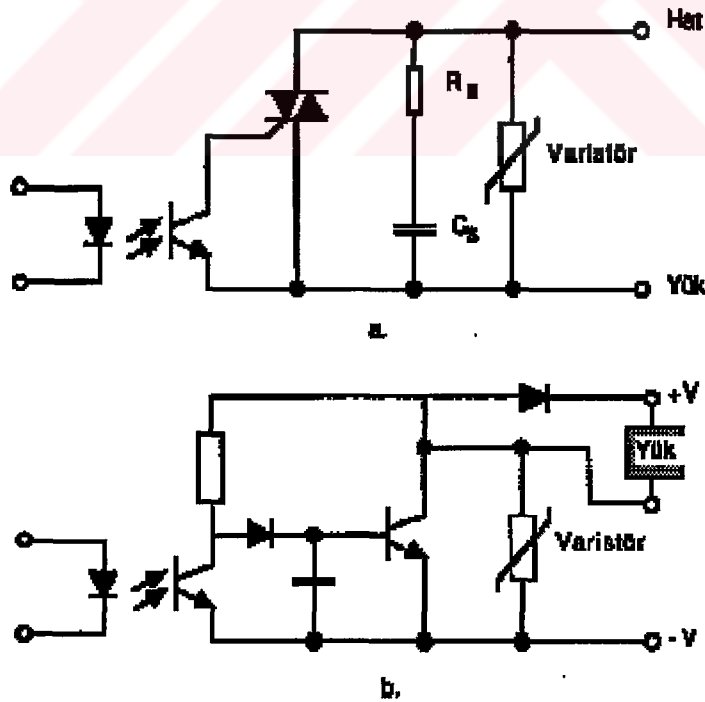
Giriş devresinde yukarıda açıklanan kısımlara ek olarak birde giriş işaretinin hangi düzeyde ("1" veya "0") olduğunu gösteren bir LED lamba yer alır. Çok hızlı

değişen darbe girişler için Şekil 2.10.'de gösterilen giriş devresinden daha değişik bir devre kullanılır. Bu tür arayüz tarama hızına uygun, daha uzun süreli darbeler çıkarır. Darbelerin tetiklenmesi giriş sinyalinin yükselen veya alçalan kenarlarından yapılabilir[3].



Şekil 2.11. Çıkış devresinin (A.A. veya D.A.) blok şeması

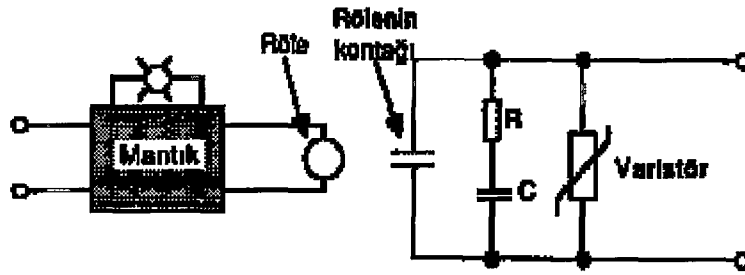
Çıkış arayüzlerine gelince, Şekil 2.11'de bir çıkış devresinin blok şeması, Şekil 2.12'de ise A.A veya D.A. çıkışlar için tipik birer gerçekleştiriliş şekli gösterilmiştir. A.A. çıkışlarda anahtar eleman olarak bir TRIAC' dan yararlanılır. Bu elemanın ani gerilim değişmelerinden dolayı istenmeyen bir şekilde tetiklenmesini önlemek için R_s ve C_s den oluşan bir "snubber" devresi, elemanı gerilim darbelerinden korumak için ise bir varistor kullanılır.



Şekil 2.12. Tipik çıkış devreleri

Giriş arayüzlerinde olduğu gibi çıkış arayüzlerinde de çıkış durumunu gösteren bir LED yer alır. Ayrıca giriş arayüzlerinde değinilen çıkış arayüzleri içinde geçerli olduğuna burada işaret etmeliyiz. Bu gecikme D.A. çıkışlar için biraz daha azdır[3].

D.A. çıkış devresi A.A. için olana benzer. Fakat burada yük bir güç transtörü ile anahtarlanır. Anahtarı korumak için bir varistöre ek olarak Şekil 2.12.b.'de gösterildiği gibi, bir başa döndürme diyodu kullanılır. Bu diyot transistörü endüktif yük durumunda oluşabilecek gerilim darbelerinden korur[3].



Şekil 2.13. Tipik röle kontak çıkışı.

PLC' lerde yukarıda değinilen A.A. ve D.A. çıkışlara ek olarak TTL (Transistör- Transistör- Lojik) düzey ve röle kontağı çıkışları da bulunur. Şekil 2.13.'da röle çıkış gösterilmiştir. Böyle bir çıkışta elektriksel izolasyon doğal olarak bulunur. Gösterildiği gibi bir RC devre ile süzgeçleme bir varistör ile de gerilim darbelerine karşı koruma yapılır. Kontak uçları A.A. veya D.A. yükleri anahtarlamak veya analog işaretleri çoklamak için kullanılır. Tipik olarak bu kontakta 2 A kadar bir akım geçirilebilir[3].

2.5.2. Sözcük Temelli Veri Giriş/Çıkış

PLC' lerin yetenekleri gelişip, salt mantık işlemlerinden daha üst düzeyde aritmetik, sayma, zamanlama gibi işlemler yapabilme olanağına kavuşmaları ile birlikte birden fazla bittin oluşan veri giriş çıkışlarının yapılabilmesi zorunluluğu doğdu; böylece bit temelli arayüzlere ek olarak sözcük temelli arayüzlerde doğdu .Bu arayüzler çok bitli çıkış veren veya elemanlarla kullanılanlara analog çıkış veren veya giriş gerektiren elemanlarla kullanılanlar olmak üzere iki temel gruba ayrılabilir. Çok bitli elemanlara örnek olarak *thumbwhell* anahtarlar , kodlayıcılar yedi parçalı göstergeler, analog elemanlara örnek ise basınç sensörleri, termokulplar,

potansiyometreler, akış sezicileri, analog valfler, motor tahrik sistemleri sayılabilir. Arayüzler veriyi genellikle paralel olarak alırlar. Ama seri kabul eden türleride vardır[5].

2.5.3. Analog Giriş/Çıkış

Analog G/Ç modülleri genellikle hem tek kutuplu hem de çift kutuplu gerilimleri veya akımları kabul edebilecek veya üretebilecek şekilde imal edilirler. A/D çeviriciler kullanımı ile gerilim ve akımlar sezilip denetim programı tarafından sezilip denetim programı tarafından incelenmek üzere eşdeğer sayısal sözcüklere dönüştürülebilir. Program tarafından üretilen sayısal değerler ise sürecin çeşitli elemanlarını denetlemek veya sürmek üzere analog gerilim veya akımlara çevrilebilir. Bu dönüştürmeler sırasında önemli olan bir nokta ayrıştırma değildir. 8 bitlik bir sözcük uzunluğu kullanan bir çevirici tam skalada 256'da 1'lik bir yeteneğine sahip olacak yani 10 voltluk bir işaret ancak 0.04 volt kadarlık bir hassasiyetle gözlenebilecek veya yeniden üretilebilecektir. Buna karşın 10 bitlik bir A/D de ayrıştırma 1024'de 1, 12 bitlikte 4096 olacaktır. Seçim uygulamaya bağlı olup genellikle geniş bir alanda değişen işaretlerin küçük değerlerinde yeterli bir ayrıştırma sağlayacak şekilde yapılır[3].

2.5.4. Özel Giriş/Çıkış Modülleri

Şimdiye kadar salt standart G/Ç modülleri açıklanmış olup bunlar bir kullanıcının gereksinim duyacağı G/Ç işlemlerinin %90'ından fazlasını kapsar. Bununla birlikte bazı uygulamalarda PLC ve G/Ç modülleri yetersiz kalır. Örneğin :

- * Sadece birkaç milisaniyelik gecikmelerin kabul edilebileceği , çok hızlı değişmelerin yer aldığı uygulamalarda ,
- * Yoğun ASCII veri transferi gerektiren uygulamalarda ,
- * Yoğun matematik işlemleri gerektiren uygulamalarda ,
- * PID denetim veya ileri düzeyde hassas konum denetimi gibi karmaşık süreç denetimi uygulamalarında,

Özel olarak tasarlanmış G/Ç modülleri kullanmak gerekir. Bu modüller genellikle akıllı olup böylece ana CPU ile sadece parametre ve veri yenileme amaçları için

ilişki kurarlar, zamanlama ve denetimi kendi içlerinde gerçekleştirirler. Bütün bunlar CPU' nun tarama hızından ayrı yapılır. Çeşitli denetim görevlerinin, bu şekilde, akıllı G/Ç modüllerine verilmesi dağıtılmış denetim olarak bilinir[3].

2.6. Programlama

PLC' lerin programlama yöntemleri temelde dört gruba ayrılır.

1. Merdiven Dili
2. Boolean Dili
3. İşlevsel Bloklar
4. İngilizce Bildirimler

Bunların ilk ikisi temel PLC dilleri olup diğer ikisi üst düzey dil sınıfına girerler. Temel PLC dilleri ile sadece röle kontağı kumandası, zamanlama, sayma sıralama ve mantık işlemleri yapılabilir. Daha ileri düzeyde komutlar gerektiren denetim uygulamaları, belirtilen üst düzey dillerin geliştirilmesine neden olmuştur. Bir PLC' nin büyüklüğüne ve yeteneklerine yukarıdaki yöntemlerin bir veya birkaçı beraberce kullanılabilir[3].

2.6.1. PLC Komutlarının Sınıflandırılması

PLC' lerin programlanmasında kullanılan komutlar genel olarak aşağıda belirtilen altı komut grubundan birine girerler.

1. Röle Mantığı
2. Zamanlama ve Sayma
3. Aritmetik İşlemler
4. Veri Transferi
5. Veri Üzerinde İşlemler
6. Program Akış Diyagramları

Merdiven Diyagramları ve Boolean Nümerikleri kullanılarak programlama yöntemlerinde en basit olarak sadece röle mantığı ve zamanlama ve sayma olanakları bulunur. Bu temel komut kümesi, diğer komut gruplarından bazı komut olanakları ile, PLC' nin türüne ve büyüklüğüne bağlı olarak, az veya çok zenginleştirilmiş

olabilir. Üst düzey diller yukarıda sayılan tüm gruplar için zengin komut kümeleri içerir[3].

2.6.2. Merdiven Dili

Merdiven Dili, elektromekanik, röleli denetleyicilerin alışlagelmiş merdiven diyagramlarına benzer bir şekilde yazılır. Merdivenin her basamağı bir denetim işlevini yerine getirir. Solda giriş işaretleri, sağda ise bir sargı işareti ile temsil edilen çıkış bulunur. Burada şunu belirtmek gerekir ki bir merdiven basamağının çıkışı, mutlaka, bir dış elemana güç bağlanmasını sağlayan gerçek bir çıkış olmayıp röleli denetim devrelerindeki denetim rölelerine benzer bir şekilde bir iç çıkış olabilir. Aynı referanslı kontaklar merdiven diyagramının diğer kısımlarında denetim amaçları ile kullanılırlar. Denetim rölelerinden farkları, kullanılacakları kontak sayısında hiçbir kısıtlama olmamasıdır. Halbuki elektromekanik rölelerin kontak sayısı sınırlıdır[3][7].

2.6.2.1. Röle Türü Komutlar

Normally Open Contact (Normal Açık Kontak) : Eğer bir işaretin varlığı bir çıkışı ON yapacaksa kullanılır. Tarama sırasında belirtilen adresin ki bu adres bir giriş, bir çıkış veya bir iç çıkış olabilir, ON durumunda olup olmadığına bakılır. Eğer ON ise normal olarak açık kontak kapanır ve mantık akışına izin verir, OFF ise normal olarak açık kontak durumunu alır ve mantık akışını keser.

Normally Closed Contact (Normal Olarak Kapalı Kontak): Yukarıda açıklanan tam tersi yönünde işlem görür. Eğer bir işaretin olmaması bir çıkışı ON yapacaksa kullanılır. Tarama sırasında belirtilen adresin OFF durumunda olup olmadığına bakılır. Eğer OFF sa normal kapalı durumunda kalır ve mantık akışın izin verir, ON sa açılır ve mantık akışına izin vermez.

Energize Coil (Sargıyı Uyar): PLC' ye bağlı bir dış elemanı veya bir iç çıkışı denetlemekte kullanılır. Eğer merdiven basamağında mantık akışı sürekli ise belirtilen çıkış ON yapılır. Aynı zamanda belirtilen adresi taşıyan bütün normal açık kontaklar kapanır normal olarak kapalı olanlar açıktır. Eğer merdiven basamağındaki

mantık akışı kesilirse çıkış OFF durumuna geçer ve kontaklarda normal durumlarını alırlar.

De - Energize Coil (Sargının Uyarısını Kes): Yukarıda açıklananın tam tersi bir şekilde işlev görür. Eğer merdiven basamağında mantık akışı varsa belirtilen çıkış OFF yapılır.

Latch Coil (Sargıyı Kilitler): Eğer bir çıkışın, o çıkışın ON olmasını sağlayan kontakların durumu değişse bile ON olarak kalması isteniyorsa kullanılır. Mantık akışı veya PLC enerjisinin kesilmesi çıkışın ON durumunu etkilemez.

Unlatch Coil (Sargı Kilidini Aç): ON durumuna girmiş kilitli bir sargıyı OFF yapmak için kullanılır. Eğer merdiven basamağında mantık akışı sürekli ise belirtilen adres OFF durumuna sokulur.

2.6.2.2. Veri Transfer ve Aritmetik İşlem Komutları

Get Word (Sözcüğü Al): Belirtilen adresin içeriğini alır ve başka işlemler için hazır tutar. Bazı PLC' ler de aritmetik veya karşılaştırma işlemlerinden önce istenilen bellek gözünden veri bu komutla alınır.

Put Word (Sözcüğü Yerleştir): Bir basamakta yapılmış olan işlemlerin sonucunu belirtilen adrese koyar.

Add (Topla): Belirtilen bellek gözlerindeki iki veriyi toplar. İçerikleri toplanacak bellek gözlerini belirtmek için bazı PLC' lerde, yukarıda açıklandığı üzere GET komutu kullanılır, diğer bazılarında ise bellek adresleri kontak sembolleri üzerinde belirtilir. Sonuç ADD sembolü üzerindeki adrese yerleştirilir.

Sub (Çıkart): Yukarıda açıklanan ADD komutuna benzer bir şekilde çıkarma işlemi yapar.

Mul (Çarp): GET komutu veya başka bir şekilde almış iki veriyi alır, çarpar ve sonucu iki bellek gözüne yerleştirir.

Div (Böl): MUL komutuna benzer bir şekilde bölme işlemi yapar. Sonucu tam sayı kısmı belirtilen iki adrese, kesir kısmı ise ikinci adrese yerleştirilir.

2.6.2.3. Sayma ve Zamanlama Komutları

Bu komutlar belirli bir süre veya olay sayısından sonra bir çıkışı ON veya OFF yapmakta kullanılır. Temelde iki işlemde aynıdır, çünkü her ikisinde de bir sayma işlemi gerçekleştirilir. İlkinde sayılan saat darbeleri, ikincisinde ise bir olayın kaç defa olduğu sayılır[3].

Time Delay Energize (Gecikme İle Sargıyı Uyar) (TON): Eğer bir basamakta mantık sürekliliği varsa zamanlayıcı belirli bir saatle saymaya başlar ve sayı belirtilen değere ulaşınca çıkış uyarılır. Bu çıkış programın diğer kısımlarında normal olarak açık veya normal olarak kapalı kontaklar şeklinde kullanılabilir. Eğer zamanlama bitmeden mantık sürekliliği kaybolursa zamanlayıcı sıfırlanır.

Time Delay De - Energize (Gecikme İle Sargıdan Uyarı Kes) (TOF): Bu komut ile bir çıkışın OFF yapılması belirli bir zaman gecikmesi ile olur. Basamakta mantık sürekliliği varsa çıkış hemen ON olur, ama mantık sürekliliği kaybedildiyse çıkışın hemen OFF olması için belirtilen sürenin gecikmesi gerekir. Eğer zaman bitmeden mantık sürekliliği kaybedildiyse çıkışın OFF olması için belirtilen sürenin geçmesi gerekir. Eğer zamanlama bitmeden mantık sürekliliği oluşursa çıkış ON olarak kalır ve zamanlayıcı sıfırlanır.

Retentive ON - Delay Timer (Saklayan ON - Gecikmeli Saklayıcı) (RTO): Eğer mantık sürekliliği veya besleme kaybolduğu zaman bile o ana kadar geçen zamanın saklanması gerekiyorsa bu komut kullanılır. Zamanlayıcının sıfırlanması ancak özel bir komutla gerçekleştirilir. Kullanılış şekli aynı TON komutu gibidir.

Retentive Timer Reset (Saklayan Zamanlayıcı Sıfırla) (RTR): Yukarıda değinilen özel komut olup kalıcı zamanlayıcı sıfırlamakta kullanılır. Basamakta mantık sürekliliği varsa belirtilen saklayan zamanlayıcının sayacı sıfırlanır.

Up Counter (Yukarı Sayıcı) (CTU): Bu komutla basamaktaki mantık sürekliliğinin OFF dan ON a her geçişinde sayıcının içeriği bir artar. Öngörülen sayıya ulaşınca da çıkış ON yapılır. Bu andan itibaren sayıcının türüne göre ya sayaç otomatik olarak sıfırlanır yada bir sıfırlama komutu kullanılır.

Down Counter (Aşağı Sayıcı) (CTD): Yukarı sayıcıya benzer ama sayma işlemi öngörülen sayıdan aşağı doğru yapılır. Sıfıra ulaşınca çıkış ON olur. Bundan sonraki olaylarda sayıcı değeri ya sıfırda kalır yada maksimum negatif sayıya ulaşınca kadar devam eder.

Counter Reset (Sayıcıyı Sıfırla) (CTR): Bu komut CTU ve CTD sayıcılarında sayaçları sıfırlamakta kullanılır. Sembolün üzerindeki numara sıfırlanmak istenen CTU veya CTD nin adresiyle aynıdır. Merdiven basamağında mantık sürekliliği varsa belirtilen sayıcı sıfırlanır.

2.6.2.4. Program Denetim İşlemleri

Master Control Relay (Ana Denetim Rölesi) (MCR): Bu komut merdiven basamaklarından bir grubun etkinleştirilmesini veya etkisizleştirilmesini sağlar. MCR komutunun bulunduğu basamakta mantık sürekliliği varsa (END MCR) komutunu taşıyan basamağa kadar bütün basamaklar dikkate alınır. Aksi durumda bu basamaklar işlenmez ve doğrudan (END MCR) komutundan sonraki basamağa geçilir.

Label (Etiketle) (LBL): Aşağıda açıklanacak olan JMP ve JSB komutları bazı basamakların bir adres taşımasını gerektirir. Bu komutla basamaklara birer etiket verilebilir. Bulunduğu basamağın mantık sürekliliğine bir etkisi yoktur basamakta daima ilk eleman olarak kullanılır. Bir programda bir etiket, doğal olarak, sadece bir defa kullanılabilir.

Jump To Label (Etikete Atla) (JMP): Bu komut, eğer bazı koşullar gerçekleşmişse program akışında bir değişiklik yaratır. Komutun bulunduğu basamakta mantık sürekliliği JMP komutunun belirlediği etiketli basamağa atlanır.

Jump To Subroutine (Alt Yordama Atla) (JSB): Komutun bulunduğu basamakta mantık sürekliliği varsa JSB komutunun belirlediği etiketli basamağa atlanır ve bir RET komutu ile karşılaşıncaya kadar program akışı oradan devam eder.



Return Coil (Geriye Dön) (RET): Bir alt yordamdan çıkılmasını sağlayan komuttur. Karşılaşıldığı zaman alt yordama girilmesine neden olan JSB komutunun bulunduğu basamağı takip eden basamağa dönlür.

2.6.2.5. Giriş/Çıkış Komutları

Bazı G/Ç' ların, normal G/Ç yenileme sırasında yapılana ek olarak programın yürütülmesi sırasında da yenilenmesi gerekebilir. Bu durumlarda aşağıdaki komutlar kullanılır[3].

Immediate Input (Hemen Giriş Yenile) (II): Programın taraması sırasında bu komutla karşılaşıldığı zaman programın akışı kesilir ve belirtilen giriş okunarak bellek haritasındaki yerine yerleştirir.

Immediate Output (Hemen Çıkış Yenile) (IO): Yukarıda açıklanan komuta benzer bir şekilde çıkışı hemen yeniler.

2.6.3. Boolean Cebri

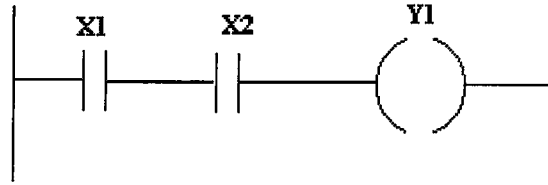
Boolean dili temel bir PLC programlama dili olup yaygın olarak kullanılmasının iki temel nedeni vardır. Bunlardan birincisi birçok denetim uygulama programının bir ara röle mantık diyagramı kullanılmadan geliştirilmesidir. Böyle bir durumda Boolean dili, denetim yönetiminin PLC' ye rahatlıkla uygulanabilmesini sağlar. Daha önemli diğer neden programa aracı olarak kullanılan taşınabilir küçük el cihazlarının tasarım şeklinden kaynaklanır. Bunların tuş sayısı ve gösterge yetenekleri kısıtlıdır, bir CRT ekranları genellikle yoktur. Boolean dilinin formatı, sınırlı özellikler altında bile kolaylıkla kullanılmasını sağlar[3][8].

2.6.4. İşlevsel Bloklar Kullanan Dil

Bu tür bir programlama dilinde kullanıcı merdiven diyagramı formatı kullanarak daha karmaşık işlevler gerçekleştirilebilir. Dil zamanlama ve sayma, aritmetik işlemler, veri işleme gibi belirli özel görevleri yerine getiren bloklar üzerine kurulmuştur. Bloklar normal olarak açık veya normal olarak kapalı türü kontaklarla denetlenir. Bloğun görevini yerine getirirken gereksinim duyacağı parametreler blok için belirtilen yerlere yerleştirilir[3].

2.6.5. İngilizce Bildirim Kullanan Diller

Bu diller, bilgisayarlarda kullanılan programlama dillerinin PLC' lere uyarlanmış birer türevidir. Ayrılıkları daha kolay ve kullanıcıya yönelik olmalarıdır. En yaygın olarak kullanılanları BASIC diline benzer. Örneğin şekil 2.13'de gösterilen merdiven basamağının programlanması yine şekil üzerinde gösterildiği biçimde yapılır[3].



Şekil 2.14.BASIC türü dilde örnek merdiven diyagramı.

```

100  REM Begin AND operation
150  Y1=0
200  IF X1=1 THEN 300
250  GO TO 100
300  IF X2=1 THEN 500
400  GO TO 100
500  Y1=1
600  GO TO 200
700  END
  
```

2.7. PLC ile Röle Sistemlerinin Karşılaştırılması

Belirli bir işlevi yerine getirmek için tasarlanmış geleneksel bir kontaklı kumanda devresi, aynı yapı kullanılarak bir PLC' li kumanda devresi, genellikle, kontaklı kumanda devresi ile işlemi sağlar. Bununla birlikte, her tür devre arasında, aşağıda açıklanan nedenlerden dolayı farklı çalışma biçimleri ortaya çıkabilir.

1. Kontaklı kumanda devrelerinin işleyişi ile PLC' li bir kumanda devresinin işleyişi farklıdır. PLC' li kumanda devrelerinde, çıkışların hesaplanması, programda yazılan sıraya göre olur. Kontaklı kumanda devrelerinde ise, böyle bir çalışma biçimi söz konusu değildir, devre elemanları paralel çalışır.

2. Kontaklı kumanda devrelerinde, kontakların fiziksel yapısından

kaynaklanan kısıtlamalar vardır. Herhangi bir kontaktör yada röle bobinine ilişkin normalde açık (NA) ve normalde kapalı (NK) kontaklar içeren geri beslemeli bir kontaklı kumanda devresi, öngörüldüğü gibi çalışmayabilir.

3. Az sayıda denetim işlevinin yapıldığı durumlarda başlangıçtaki yatırım PLC için daha fazladır.

4. PLC' li sistem, elektromekanik denetime (EMD) nazaran daha uzun süre çalışabilir durumda kalır. PLC' de ki diagnostik programlar, tamir için ortalama zamanı düşürür.

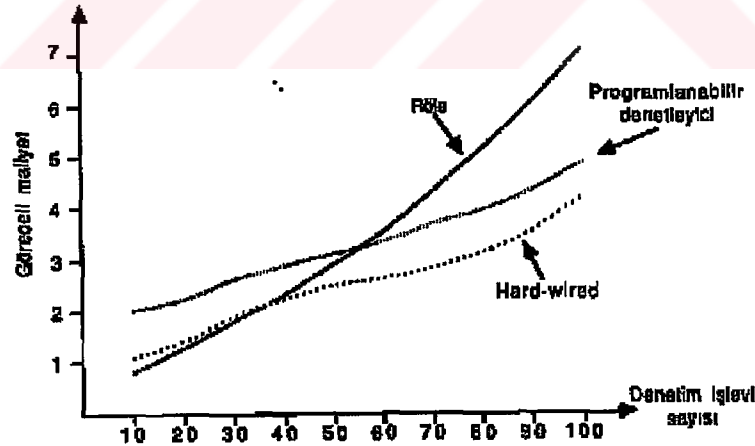
5. Arızalar arası ortalama zaman bir PLC için 8000 saatin üzerindedir.

6. PLC' lerin çalışma hızı çok daha yüksektir.

7. Zamanla gereksinimler veya denetlenecek sistemin yapısı değişecek olursa PLC temelli bir sistem çok az bir ek donanım ile program değiştirilerek çok daha kısa zamanda yeni sisteme uyarlanabilir. Oysa EMD denetimi değiştirmek hiç kolay değildir.

8. PLC' ler çok daha az yer kaplar ve daha az enerji tüketirler.

PLC' ler, EMD' ler ve eldeki amaca uygun bir şekilde bağlanmış (Hard-Wired) katı hal sistemlerin gerçekleştirilecek denetim işlevi sayısına bağlı olarak göreceli fiyatlarının genel bir karşılaştırması aşağıda şekil 2.15.'de verilmiştir[3].



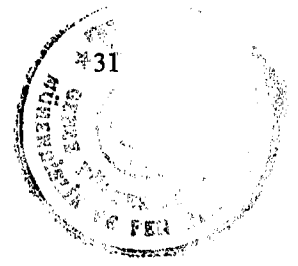
Şekil 2.15. Röle, hard-wired, plc sistemlerinin göreceli maliyetleri

2.8. PLC ile Bilgisayarlı Denetim Karşılaştırılması

Yukarıda açıklanan ve ileri ki bölümlerde açıklanacak olanlardan bir PLC'nin yapısının ve çalışma şeklinin bir bilgisayarın yapısı ve çalışma şekline çok benzediği görülebilir. Bu da , doğal olarak PLC' lerle özellikle mikro işlemci temelli denetleme sistemleri arasında ne gibi bir fark olduğu sorusunu akla getirebilir. İlk ayrılık PLC' lerin, endüstriyel ortamlarda görülen yüksek düzeyde elektriksel gürültü, elektromanyetik parazitler, mekanik titreşimler, yüksek sıcaklıklar gibi ters koşullar altında çalıştırabilmek için özellikle tasarımılanmış olmasından kaynaklanır. Buna karşın bir mikro işlemciyi benzer koşullar altında çalıştırabilmek için bazı özel önlemler almak gerekir[5].

İkinci ayrılık ise PLC' lerin donanım ve yazılımlarının o tesisin mühendis ve teknisyenlerince kolaylıkla kullanılabilecek bir düzeyde tasarımılanmış olmasıdır. Tesisdeki elemanlarla PLC arasındaki bağlantı modüler G/Ç üniteleri ile kolaylıkla gerçekleştirilebilir. Diyagnostik yazılımlar aracılığı ile yanlış çalışan birimler kısa zaman bulunabilir ve değiştirilebilir. Yazılım alışıla gelmiş röle sembollerini veya kolayca öğrenilebilen basit bir dille gerçekleştirilmiştir. Bütün bu özellikler PLC' lerin tesise entegrasyonu çıkabilecek sorunları en az bir düzeye indirir[5].

Son bir ayrılıkta genelde, tek bir programı sıralı bir şekilde baştan sona kadar gerçekleştirilmesidir. Hal bu ki bilgisayarlar birden fazla programı , değişik sıralarla , esnek bir şekilde gerçekleştirme yeteneğine sahiptir. Bununla birlikte bu ayrılık PLC' lerin son yıllarda daha da akıllı bir düzeye gelmesiyle yavaş yavaş ortadan kalkmaktadır[5].



3. BULANIK KÜME TEORİSİ

Bulanık mantık, bulanık küme teorisine dayanmaktadır. Bulanık küme teorisi genel bir matematiksel yaklaşımdır. Bu yaklaşım ile çözülmesi güç olan problemler genel bir yapıya kavuşturularak daha kolay bir sonuca gidilir. Bulanık küme teorisi, bilinen klasik küme teorisinin genel bir halidir. Örnek olarak, bir koyun memeliler kümesinin bir elemanıdır; fakat bir balık bu kümenin bir elemanı değildir. Bu tür küme, klasik bir küme anlayışıdır. Bulanık küme, gerçekte birden fazla klasik kümenin varlığını kabul eder. Klasik küme, platyus denen hayvanla ilgili büyük bir soruna sahiptir. Bu hayvan, tipik memeli özellikleri gereği memeliler kümesinde yer alırken; bir ördek gibi bataklıklarda yaşaması sebebiyle kuşlar kümesinde de yer alabilmektedir. Fakat bulanık küme teorisi böyle bir istisna ile uğraşmak zorunda değildir.

Bulanık küme teorisi, kısmi üyeliğe izin veren bir mantık sistemidir. Yani bir kümenin tam üyeliği ile o kümenin üyesi olmama durumları arasında derece derece geçişe izin verir. Verilen bir elemanın bir kümede kısmi üyeliğinin bulunması demek, aynı zamanda bu elemanın bu kümenin üyesi olmama durumunun da kısmen başlaması demektir. Çünkü, bulanık küme teorisi hem tam üyeliğe hem de hiç üye olmamaya izin verir. İşte bundan dolayı bulanık küme teorisi, klasik küme teorisinin genelleştirilmiş bir hali denebilir[2].

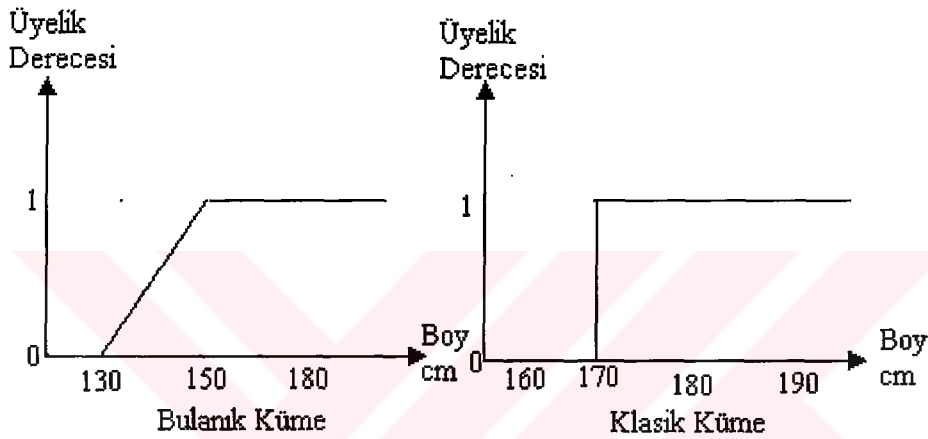
3.1. Üyelik Derecesi Kavramı

Bulanık küme mantığı, klasik mantıktan gelmektedir. Kesin kümeler, ihtiyaçları karşılasa da karşılamasa da kesin üyeliklere sahiptir. Sadece doğru ve yanlış bulunduğu kesin bir kümede sadece 0 veya 1 bulunabilir. Oysa bulanık mantıkta 0 ile 1 arasında değişen birçok değer alınabilir. Yani bulanık mantıkta birşey ne tanı doğru ve ne de tam yanlış olmaktadır. Doğruluk derecelerine göre ayrılabilir. Bulanık sistemler ile gerçekleştirilen çözümlerde kullanılan kümelerde kısmi üyeliğe geçiş bölgesi oluşturulmaktadır.[2]

Uzun boylu insan kime denir sorusunun cevabını her iki mantığa göre değerlendirelim. Şekil 3.1' de görüldüğü gibi klasik küme mantığına göre 160 cm. boyundaki bir kişi uzun boylu insanlar kümesi içinde değildir. Hatta 169 cm. bir



insan bile bu kümede yer almamaktadır. Oysa bulanık mantığa göre 160 cm. boyundaki kişiye kısa denilmez. Çünkü kısmen de olsa uzun boylu insanlar kümesi içindedir. Hatta bulanık küme içinde tam üyelikten hiç üye olmamaya kadar uzanan bölgede kısmi üyeliğe izin verdiği için 150 cm. boyunda olan bir insana uzun boylu insanlar kümesinde değil denilemez. Bunun gibi bir insanın uzun boylu olması, bulanık küme mantığında derecelerine ayrılabilir. Uzundan kısaya giden yol kısmen yanlış ve kısmen doğru olacağı için bu geçişi sağlayacak bir istek kümesine ihtiyaç olacaktır[2].



Şekil 3.1. Bulanık küme ile klasik kümenin karşılaştırılması

3.2. Bulanık Mantık İşlemleri

Bulanık küme teorisi kullanılarak herhangi bir elemanın belirlenen bir bulanık kümesindeki üyelik derecesi bulunabilmektedir. Böylece $\mu_A(x)$ $A:[0,1]$ ifadesi, 0-1 aralığındaki değerleri kapsayan bir A kümesindeki x elemanın üyelik derecesini gösterir. Bu üyelik dereceleri dilsel bulanık kümeleri üzerinde yüzdelerle ifade edilir. Dilsel değişkenler soğuk, ılık, sıcak ve çok sıcak gibi kümeler olabilir. Sonuç olarak böyle bir durumda %30 soğuk, %10 sıcak, %15 ılık gibi üyelik dereceleri elde edilir[2].

Bulanık mantıkta kullanılan en yaygın tanımlar, bulanık mantık teorisini bulan Lotfi A. Zadeh tarafından verilen terimlerden oluşmaktadır. Bu tanımlara göre;
A ve B : Bulanık değişkenler, yani tanımlanan ve dilsel olabilen kümeler,
AND : Minimum üyelik derecesini bulmak için kullanılan mantıksal işlem,

OR : Maksimum üyelik derecesini bulmak için kullanılan mantıksal işlem

NOT : $1-\mu_A(x)$,

$$\mu(A \text{ AND } B) = \min(\mu_A, \mu_B), \quad (3.2)$$

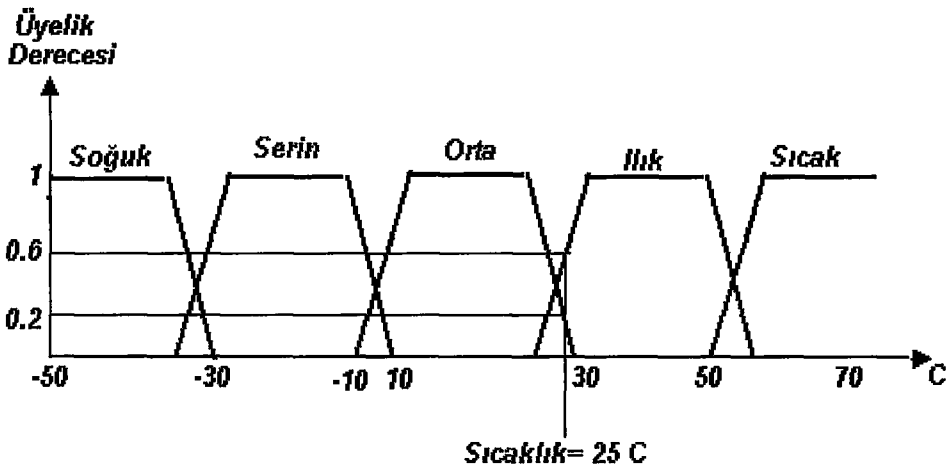
$$\mu(A \text{ OR } B) = \max(\mu_A, \mu_B) \quad (3.3)$$

$$\mu(\text{NOT } A) = 1 - \mu_A \quad (3.4)$$

olarak verilmektedir.

Bulanık mantık ile yapılan bir çözümün ilk aşamasında, hedef sistemin giriş ve çıkışlarına üyelik fonksiyonları tayin edilir. Üyelik fonksiyonları yüksek, orta ve düşük gibi nicel tanımlamalardır. Bunlar giriş ve çıkış değişkenlerinin değer aralıkları olarak kabul edilir. Tipik olarak bir üyelik fonksiyonu, x-y düzleminde bir eğri olarak gösterilir. Bu düzlemde x-ekseni, giriş ve çıkış değerlerinin değer aralığını gösterirken. y-ekseni ise 0-1 arasında olmak üzere değişkenin üyelik derecesini verir[2].

Şekil 3.2' de giriş sıcaklığı, atanan beş üyelik fonksiyonu ile verilmiştir. Her giriş değerine karşılık gelen bütün üyelik fonksiyonlarındaki üyelik derecesi, y-ekseni ile gösterilmiştir. Mesela giriş sıcaklığı 25 °C ise, y-eksenine bakıldığında bu giriş değeri için, ılık ve orta fonksiyonlarındaki üyelik derecesi sırasıyla 0.6 ve 0.2 olur. Bu giriş değeri için diğer bulanık kümelerindeki üyelik derecesi sıfırdır. Bu örnekte üyelik fonksiyonunun şekli yamuk olarak seçilmiştir. Bulanık mantıkta üyelik fonksiyonu için üçgen veya çan eğrisi kullanılmaktadır.



Şekil 3.2. Sıcaklık giriş parametresinin üyelik derecesi



Giriş ve çıkış üyelik fonksiyonları atandıktan sonra sistemin davranışını tanımlamak için bulanık kurallar (rules) üretilmektedir. bulanık mantığın kuralları tipik olarak **IF-THEN** şeklinde olmaktadır. Mesela tipik bir bulanık kuralı,

EĞER sıcaklık Düşük **VE** basınç Orta ise **O HALDE** güç Hızlı' dır.

Bu kural, iki varsayım ile bir karar içermektedir. Bulanık mantıkta varsayımlar, **AND** (**VE**) işlemi ile kurallar da **OR** (**VEYA**) işlemi ile ayrılır. Bulanık varsayım ve karar içeren bir örnek aşamada verilmiştir:

EĞER X küçük ise **O HALDE** Y büyük' tür.

EĞER X orta ise **O HALDE** Y orta' dır.

EĞER X büyük ise **O HALDE** Y küçük' tür.

Bu kurallar bir fonksiyon ile gösterilirse:

$F = \text{küçük} * \text{büyük} + \text{orta} * \text{orta} + \dots + \text{büyük} * \text{küçük}$ olur.

Bu örnek bu şekilde ifade edilebilir,

EĞER X A_1 ise **O HALDE** Y B_1 ' dir.

EĞER X A_2 ise **O HALDE** Y B_2 dir.

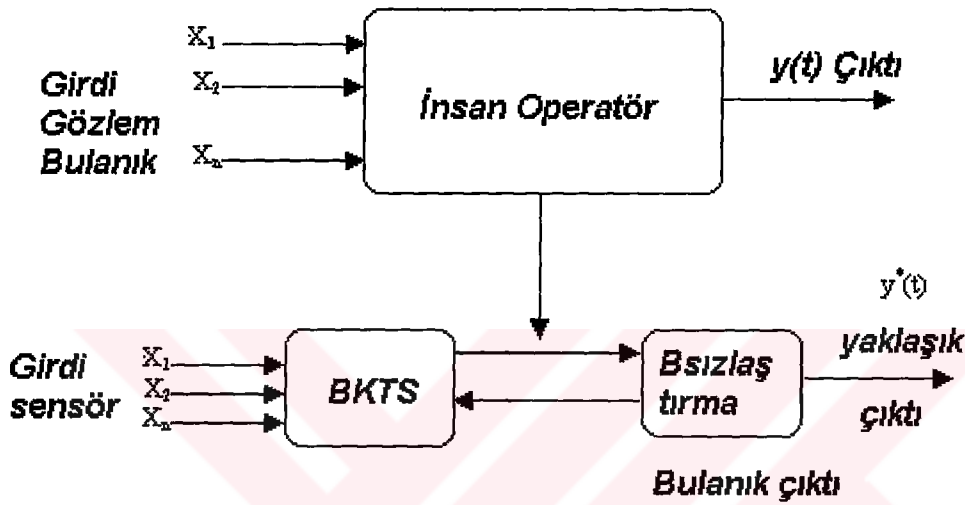
.....

EĞER X A_N ise **O HALDE** Y B_Y ' dir.

3.3. Bulanık Denetim

Bulanık denetim sistemleri, sistemden gelen kesin harekete sevkedici unsurların etkileri bir denetim karar mekanizmasıyla temsil edilen bulanık kurallar kümesi diye anılan kural-tabanlı sistemlerdir. Bulanık denetim sistemlerinin amacı normal olarak bir bulanık kural-tabanı sistemi vasıflı bir operatörün yerini almak veya yerine koymaktır. Bu tanım için bir blok diyagramı şekil 3.3'de gösterilir. Gösterildiği gibi, operatörün nicelikleri bir gözetme kartı veya bir ölçü aletiyle okunarak gözlenir ve sonra bir aksiyon (iş) tanımı yapılır. Benzer bir tarzda bulanık denetleyici, şekil 3.4'de gösterilen, sensörlerin sayısından direkt olarak kesin-tam veri kullanır;

bulanıklığın prosesi esnasında bunlar, dilbilimine veya bulanık üyelik fonksiyonlarına dönüştürülür. Daha sonra bunlar bulanık çıktı sonuçları ve bir uzman sistem gibi bir çıkarım mekanizmasındaki "IF - THEN" bulanık kurallar setine gider. Bulanık çıktı veya çıktılar "centroidal" metot gibi bazı ağırlıklı ortalamalar tarafından "defizzification" olarak adlandırılan bir prosesten geçerek kesin-tam değerlere, daha sonra geri dönüştürülebilecek. Sonuç, bir y^* çıktı değeridir. Bu yolla, tasarımcı gerçek y çıktısı için yaklaşık bir y^* çıktısı elde eder[9].



Şekil 3.3. Bulanık Denetim Sistemi Tanımı

Daha önce belirtildiği gibi, bir bulanık denetleyici tipik olarak öncekiler (IF kısmı) ve izleyenler neticeler THEN kısmı ile bir IF - THEN kurallar setinden kendi üyelik fonksiyonlarını alırlar. Farklı kurallardan çıkan sonuçlar sayısal olarak birleştirilir (tipik olarak MAX yoluyla birleşim) ve tek bir gerçek sayı (ikili) çıktı sağlamak için sonuçlandırılır (tipik olarak birleştirilmiş dağılımın centroid'i alınarak). Bir uzman sistemin yapısında, düzenli uzman sistemlere benzer tipik kurallar (rules) bir operatörün bilgisi sonuç olabilir. Şöyleki;

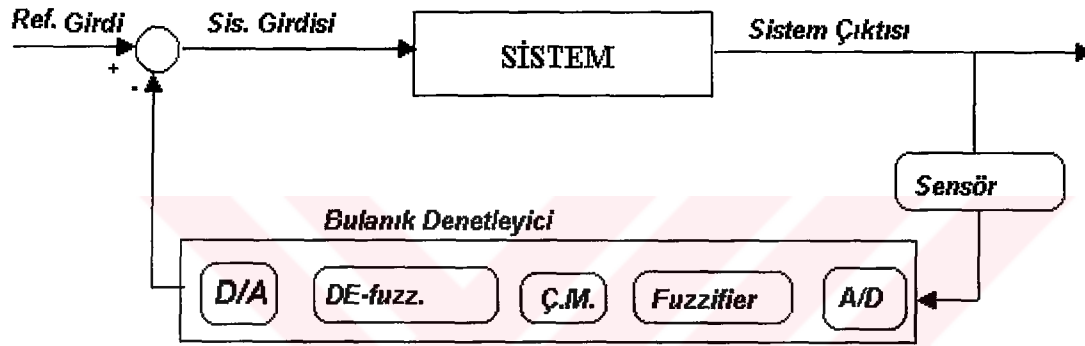
EĞER ısı sıcak ise **O HALDE** şu andaki seviyeyi orta seviyeye düşür

Bu kuralda, sıcak ve orta, bulanık değişkenlerdir. Bu gibi doğal dil kuralları bilgisayar diline daha sonra dönüştürülebilir, şöyleki;

EĞER (A, A1 ise **VE** B, B1 ise **VE** C, C1 ise **VE** D, D1 ise)

O HALDE (E, E1 **VE** F, F1) dir.

Bu gibi kurallar setini kullanarak, sınırlı sayıda ölçülebilir kurallar girilerek bir operatörün denetim işini yaptığı gibi doğal dil açıklamalarından türetilir. Bazı pratik sistemlerde, örneğin havalandırma sisteminde, kullanıcı veya operatör, istenen hızla istenen serinliğe (veya sıcaklığa) kadar sıklıkla düğmeleri ayarlar ve ince ayar yapar. Bu gibi operatör bilgisi bir havalandırma ünitesi sistemi için bir bulanık denetleyicinin dizaynında kullanılabilir. Birçok yaygın bulanık denetleyici dizaynı tipik bulanık denetim yapısını gösterir. Denetleyici bulanıklık (ikili bulanık dönüştürücüye) ve bulanıklıktan arındırmayı (bulanık ikiliye dönüştürücü) gösterir[9].

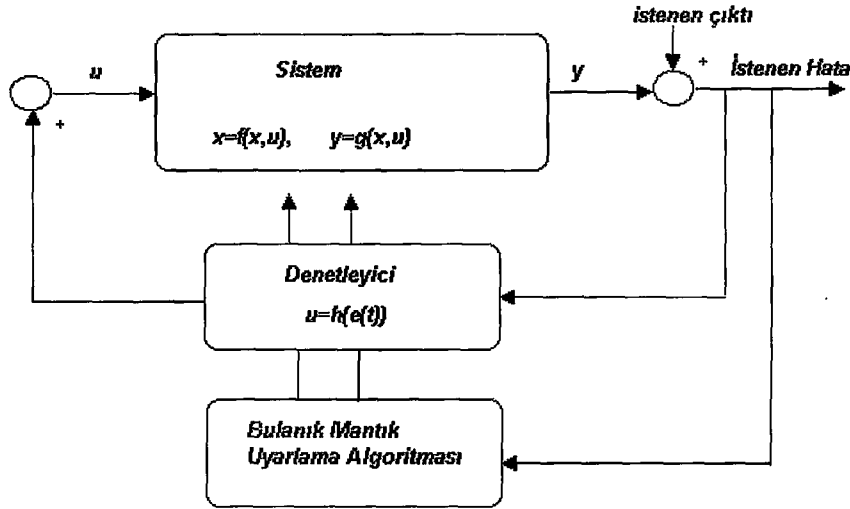


Şekil 3.4. Bulandırıcı (Fuzzifier), Bulanıklıktan Arındırma (DE-Fuzzifier) ve Çıkarım Mekanizmasını Gösteren Tipik Bir Bulanık Kontrol Sistemi için Blok Diyagramı

Sıradan uyarlamalı denetim kanununa benzer bir bulanık denetim sistemini yerine getirmenin alternatif bir yolu PID

$$u = K_p \cdot e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (3.5)$$

gibi standard bir tam-kesin mantık denetleyici kullanmak ve sonra geleneksel denetleyicilerin kazançlarını, yani, K_p , K_i ve K_d ayarlamak için bulanık **IF - THEN** kurallarını kullanmaktır. Burada $e(t)$ değişkeni, $y(t)$ sistem çıktısı için arzu edilen $y_d(t)$ çıktısına ulaşmada izlenen hatadır. Şekil 3.5. böyle bir adaptif (uyarlamalı) bulanık denetim yapısını gösterir.



Şekil 3.7.Uyarlamalı (Kendi Kendini Uyarlayan) Bulanık Kontrol Sistemi

Bulanık denetim problemi, herhangi bir denetim problemi gibi, aşağıda gösterildiği şekilde bir $h(\cdot)$ haritalama değerlendirmesidir.

$$h : e \rightarrow u \quad (3.6)$$

burada u ve e sırasıyla denetim ve hata sinyalleridir. $h(\cdot)$ 'nin seçimi çok önemli denetim problemidir. İki-sensör durumu için, e ve Δe ye karşı u 'nun diyagramı. İki hata değişkeni (e ve Δe) ve bir denetim sinyali denetim yüzeyi olarak adlandırılan bir yüzeyle sağlanır[9][10].

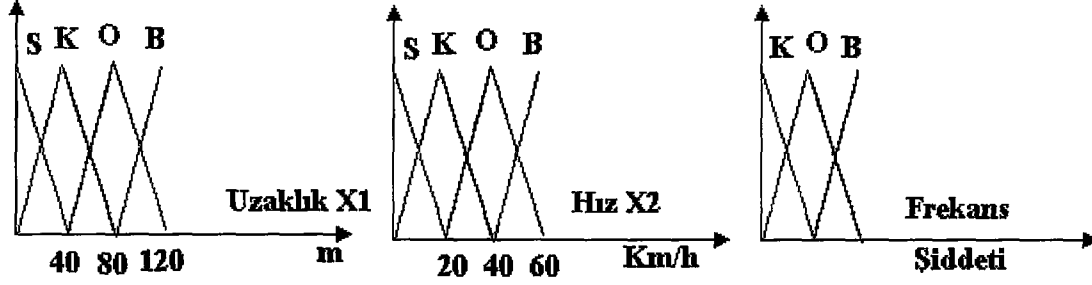
Aslında günlük hayatımızın hemen her anında farkında olmadan bulanık denetim uyguluyoruz. Örneğin arabayı parkederken, gidilecek yere en çabuk varacak yolu seçerken, yol kavşağından geçerken. Bu örneklere benzer bir başka örnek de yolda arabayla, bir başka arabayı izlemek olabilir. Öndeki araba ile sizin aranızdaki uzaklık X_1 metre, sizin hızınız ise X_2 km/saat olsun. İlk olarak yıllardır araba kullanıyor olmanız dolayısıyla elde ettiğimiz deneyimlerimizi ve ustalığınızı birer dilsel kurallar (linguistic rules) şeklinde açıklayalım[11].

Kural 1: Eğer iki araba arasındaki uzaklık az (küçük-K) ve hızınız orta (orta-O) ise sert (büyük-B) fren yap.

Kural 2: Eğer iki araba arasındaki uzaklık orta (O) ve hızınız yüksek (B) ise orta sertlikte (O) fren yap.



Burada iki araba arasındaki uzaklık ve arkadaki arabanın hızı (X_1 ve X_2) öncel kısımlar (antecedent parts), frenleme oranı (Y) ise sonuçsal kısım (consequent parts) olarak anılır. İkinci olarak yapılacak şey, öncel ve sonuçsal kısımların üyelik işlevlerinin belirlenmesidir. Bunlar için şekil 3.6'te gösterilen biçimde 3 etiketli üçgen üyelik işlevleri kullanalım.



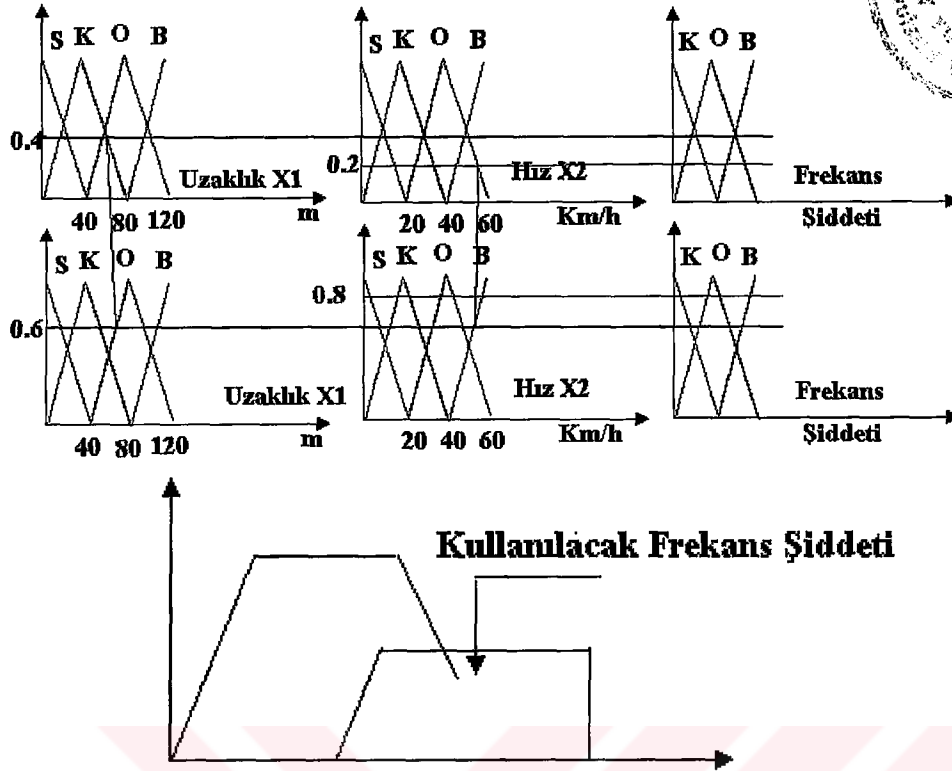
Şekil 3.6. İki araba üyelik işlevleri

Şimdi de kurallarımızı kullandığımız sembol ve etiketlerle kodlayalım:

Kural 1: **EĞER** $X_1=K$ ve $X_2=0$ ise **O HALDE** $Y=B$

Kural 2: **EĞER** $X_1=O$ ve $X_2=B$ ise **O HALDE** $y=O$

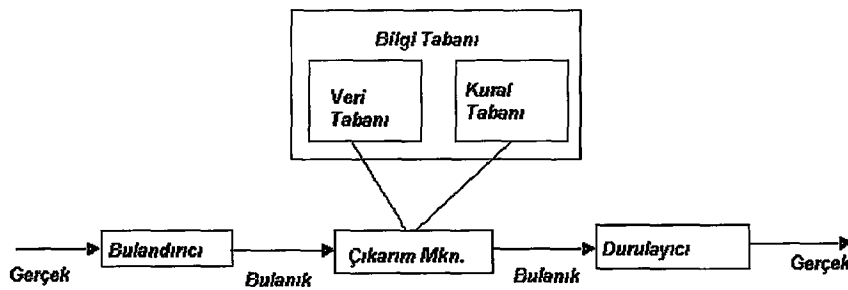
Artık çıkarım mekanizmamızı çalıştırıp belirli bir hız ve uzaklık durumu için en şiddeti değerini bulabiliriz. Bir örnek olarak $X_1=64m$ ve $X_2=56 km/saat$ olsun. Yapacağımız ilk iş değerlerine karşılık gelen üyelik ağırlıklarını bulmaktır. Şekil 3.7. a ve b'den bu değerlerin Kural 1 için $\mu_{11}=4$, $\mu_{12}=0.2$ olduğu, Kural 2 için $\mu_{21}=0.6$ ve $\mu_{22}= 0.8$ için olduğu görülmektedir. Şimdi de kuralların ağırlıklarını bulalım. Bu işlem öncel kısımlarının ağırlıklarının en küçüğünün (minimum) olması ile yapılır. (Bulanık mantıkta "ve" leme en küçük ağırlığın, "veya" lama ise en büyük ağırlığın alınması ile gerçekleştirildiği daha önce açıklanmıştı.). Böylece $\mu_1=0.2$ ve $\mu_2=0.6$ bulunur. Bu ağırlıklardan, uygulanması gereken fren şiddetini bulabilmek için izlenecek yöntemlerden biri (bulanık mantıklı düşünme- fuzzy reasoning) şekil 3.7 a, b, c'den görülebilmektedir. Sonuçsal kısım üyelik işlevleri kuralların ağırlıklarına ayarlanır. Böylece şekil .6 a ve b de gösterilen iki taralı alanın ortaya çıkar. Bu iki taralı alanın toplanıp sonucun ikiye bölünmesi (yani ağırlık merkezlerinin bulunma ile) fren şiddeti bulunur. Bu işlem, daha sonra da görüleceği üzere. bir durulandırıcı (Bulanıklıktan arındırma) ile gerçekleştirilir. Şekil 3.7. c' den bu şiddetin ortanın biraz üstü olması gerektiği görülebilmektedir.



Şekil 3.5.a.b.c. Fren şiddetinin bulunması

3.3.1. Bulanık Mantık Denetleyicinin Yapısı

Bu kısımda bir bulanık denetleyici, daha önceki bölümlerde verilen ön bilgilerin ışığı altında, daha ayrıntılı bir biçim de incelenecektir. Bir bulanık denetleyici, şekil 3.8' de gösterildiği üzere, 4 ana bölümden oluşur:



Şekil 3.8. Bir bulanık denetleyicinin temel yapısı

a. Bulandırıcı (fuzzifier) : Bu bölüm giriş değişkenlerini (gerçek Sayıları) ölçer, onlar üzerinde bir ölçek değişikliği yapar ve bulanık kümelerle dönüştürür, yani onlara birer etiket vererek dilsel nicelik kazandırır.

Bu kısımda bir bulanık denetleyici, daha önceki bölümlerde verilen ön bilgilerin ışığı altında, daha ayrıntılı bir biçim de incelenecektir.

b. Çıkarım mekanizması (inference engine) : Burada kurallar üzerinde bulanık mantık yürütülür (fuzzy reasonin), Yani insan beyninin düşünüş şeklinin bir benzetimi yapılmaya çalışır.

c. Veri tabanı (data base) : Çıkarım mekanizması, kural tabanında kullanılan bulanık kümelerin üyelik işlevlerini bu bölümde alır.

d. Kural tabanı (rule base) : Denetim amaçlarına uygun dilsel denetim kuralları burada bulunur ve çıkarım mekanizması buradan verilir.

e. Durulandırıcı (defuzzifier) : Çıkarım mekanizmasının bulanık küme çıkışları(birden fazla çıkış olabilir) üzerinde gerekli ölçek değişikliklerini yapar ve bunları gerçek sayılara dönüştürür.

3.3.2. Kuralların Oluşturulması

Bir bulanık denetleyicinin gerçekleştirilişinde, denetlenecek sistemin bir matematiksel modelinden daha çok o sistemi çalıştıran operatörün sistem davranışı konusunda sahip olduğu bilgiler daha önemlidir. Tasarım sırasında genellikle bu tür bilgilerden yararlanılır. Böyle bir yaklaşım, uzun yıllar boyunca kazanılan deneyimler denetleyici içerisine yorumlanmış halde kolaylıkla yerleştirilmesine olanak sağlar. Bu yararın yanında getirdiği sakınca, denetleyici tasarımında kullanılacak belirli kriterlerin olmaması, tasarımın belirli bir otomasyona bağlanamamasıdır. Bununla birlikte bulanık denetleyicinin en önemli kısmını oluşturan kural tabanının oluşturulması için kullanılabilecek çeşitli yaklaşımlar literatürde açıklanmıştır. Bunlardan bazıları:

* bir uzmanın bilgi ve/veya deneyimlerine dayanır.

* sürecin bir bulanık modelinin kullanılmasına dayanır.

* operatörün süreç üzerinde yaptığı işlemlere dayanır.

* öğrenen algoritmalar kullanır.

Bunlardan birinci ve sonuncu yaklaşım, özellikle küçük çaplı denetim işlemlerinde en yaygın olarak görülen yaklaşımlardır. Araba frenlemesi örneğinde birinci yaklaşım kullanıldığı gibi aşağıda açıklanacak olan bulanık servo denetleyicide de bu yaklaşım izlenmiştir[11].

3.3.3. Bulanık Servo Denetleyici

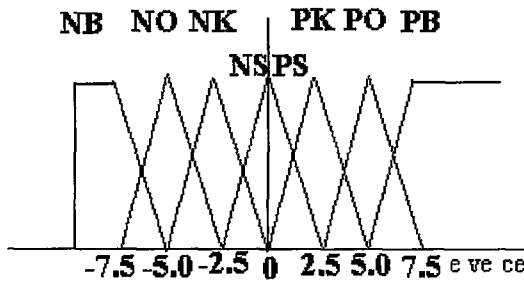
Servo denetleyicilerde denetim amacı izleme hatasını en aza indirmektir. Hata ne kadar büyük ise denetim girişinin belirlenmesinde hatanın değişim oranı da oldukça önemli rol oynar. Dolayısıyla da denetim kurallarından oluşacak savlarımızda dilsel değişkenler olarak hata ve hata değişim oranlarını kullanmalıyız. Bunları

$$e = \text{hata} = (\text{istenen konum} - \text{o andaki konum}) * KE \quad (3.7)$$

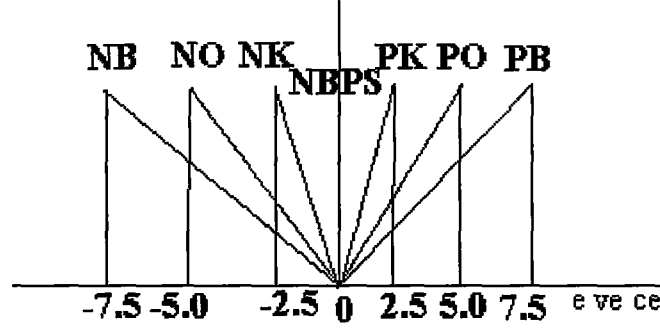
$$ce = \text{hata değişimi} = (\text{şimdiki hata} - \text{bir önceki hata}) \times KCE \quad (3.8)$$

olarak tanımlayalım. Burada KE ve KCE birer kazanç katsayılarıdır.

Denetim kurallarımızın denetim çıktımız u olup bu değer e ve ce' nin sonuçsal $u=f(e,ce)$ şeklinde uygun bir işlevi olacaktır. O halde bulanık denetleyicimizi Oransal-Türevsel (PD) denetleyici olarak düşünebiliriz. Şimdi de e ve ce için PB, PO, PK, PS, NS, NK ve NO olmak üzere 8 bulanık küme tanımlayalım. Burada P=pozitif, T=Negatif, B=Büyük S=Sıfır O=Orta, K=Küçük olarak kullanılmış olup, örneğin PO pozitif orta anlamına gelir. Üyelik işlevleri olarak şekil 3.9.' de gösterilen monotonik üyelik işlevlerini kullanalım[11].



Şekil 3.9. Hata (e) ve hata değişiminin dilsel değerlerinin üyelik işlevleri



Şekil 3.10. Denetim girdisinin dilsel değerlerinin üyelik işlevleri

Şimdi de denetim kurallarımızı belirleyelim. Bulanık denetleyicilerde kurallar genellikle Çizelge 3.1.'de gösterildiği üzere bir tablo şeklinde yazılır. Bu tablonun elemanları, servo sistemleri hakkındaki deneyimlerimizin kullanılması ile, gösterilen biçimde doldurulabilir. Burada 14 tane kural bulunmaktadır. Bunlardan iki tanesi birer örnek olarak aşağıda verilmiştir.

EĞER $e=NB$ ve $ce=PB$ ise **O HALDE** $u=NB$.

EĞER $e=NS$ ve $ce=PK$ ise **O HALDE** $u=PS$.

Çizelge 3.1. Bulandırıcı denetleyici için kural tabanı

		<u>E</u>							
	NB	NO	NK	NS	PS	PK	PO	PB	
NB		PO							
PO		*							
PK		*		PS	PK				
PS									
NS	NB		NK	NS	PS	PK		PB	
NK				NK	NS	*	*		
NO									
NB			NB						

Tabloda bazı elemanların olmayışı, (örneğin $e=NK$ ve $ce=PK$ durumu) böyle bir durumda denetim çıktısı olmayacağı anlamına gelmez. Üyelik işlevlerinin örtüşümünden dolayı çevredeki değerlerden denetim çıktısı bulunabilir. Bir örnek olarak $e = -3$ ve $ce = 2$ durumunu ele alalım. Bu değerler için şekil 3.9'den $e = NO$, $e = NK$, $ce = PS$ ve PK etiketlerini okuruz ve bunların oluşturduğu dört



kombinasyondan ikisi için tabloda kural olmadığını diğer ikisi için ise aşağıdaki kuralların uygulanması gerektiğini görebiliriz.

EĞER $e=NO$ ve $ce=PS$ ise **O HALDE** $u=NB$.

EĞER $e=NK$ ve $ce=PS$ ise **O HALDE** $u=PK$.

Bundan sonra yapmamız gereken şey bu iki kural için minimum üyelik ağırlıklarını (g_1 ve g_2) bulmak ve daha sonrada durulaştırma işlemini yapmaktır. Şekil 3.10' da gösterilen üyelik işlevlerinin monotonik biçimde olması dolayısıyla burada daha önce açıklanandan daha farklı bir yöntem uygulanacaktır. Şöyleki g_1 ve g_2 değerlerine karşı gelen iki u değeri (u_1 ve u_2) bulunur ve,

$$u = \frac{g_1 u_1 + g_2 u_2}{g_1 + g_2} \quad (3.9)$$

formülünden denetim girdisi hesaplanır. Bu durulaştırma yöntemi ağırlıklı ortalama alma yöntemi olarak bilinir.

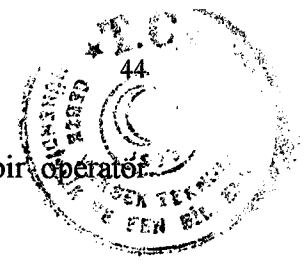
3.3.4. Üyelik İşlevlerinin Bir Tablo Şeklinde Gösterimi

Yukarıda açıklanan örnekte oldukça fazla bir zaman ve uğraş, belirli girişlere karşı gelen üyelik ağırlıklarını bulmak için harcanır. Eğer biz hem girişleri hem de üyelik ağırlıklarını sürekli değerler yerine kuantalanmış değerler olarak ele alırsak matris şeklinde bir "Gözetme" tablosu yapmamız mümkündür.

Benzer bir kuantalama işlemi çıkış değeri u için de yapılabilir ve Çizelge 3.2 'da gösterilen üyelik işlevleri benzer bir tabloya dönüştürülebilir. Hatta, bu şekilde oluşturulan iki tablo ile Çizelge 3.1.' de verilen kural tablosunun birleştirilmesi sonucu tek bir tablo ortaya çıkartılabilir. Bu tabloda x ekseni hata e 'nin kuantalanmış değerleri y ekseni hatanın değişme oranı ce 'nin kuantalanmış değerleri olup bu iki değere karşı gelen tablodaki değer doğrudan denetleyici çıkışıdır. Böyle bir yaklaşım (look-up tablosu) bir bilgisayar içerisinde çok hızlı bir şekilde gerçekleştirilebilir, böylece çıkarım mekanizmasının hızı artırılmış olur.

Yukarıda açıklanan benzer bir tablonun elemanları deneme yanılma yolu ile bulunur. Bu konudaki çalışmalar hala sürmekte olup amaç, özellikle değişen koşullar

altında sürecin, alınması gereken denetim önlemlerini kendi kendine, bir operatör olmadan da öğrenmesidir[11].



Çizelge 3.2. Üyelik işlemlerinin bir matris şeklinde ifadesi.

E ve ce																	
Etk	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
PD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0.8	1
PO	0	0	0	0	0	0	0	0	0	0	0	0.2	0.6	1	0.6	0.2	0
PK	0	0	0	0	0	0	0	0	0	0.4	0.8	0.8	0.4	0	0	0	0
PS	0	0	0	0	0	0	0	0	1	0.8	0.4	0	0	0	0	0	0
NS	0	0	0	0	0	0	0.4	0.8	1	0	0	0	0	0	0	0	0
NK	0	0	0	0	0.4	0.8	0.8	0.4	0	0	0	0	0	0	0	0	0
NO	0	0.2	0.6	-1	0.6	0.4	0	0	0	0	0	0	0	0	0	0	0
NB	-1	0.8	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3.3.5. Bulanık Denetimin Uygulama Alanları

Bulanık denetimin ilk uygulamaları genellikle endüstriyel alanlarda çimento sanayinde ve su arıtma sistemlerinde olmuştur. Daha sonraları literatürde nükleer reaktör, asansör ve vinç denetimi gibi daha değişik uygulamalar görülmüştür. Bu gelişim içerisinde en önemli olay bulanık denetimin, Kuzey Japonya'nın Sendai'de kentindeki metro sisteminde çok başarılı bir biçimde kullanılması olmuştur. Bu olay bulanık denetim uygulamalarına büyük bir ivme kazandırmış, Japonya'da adeta bir patlama yaratmıştır. 1987 yılında başlayan bu patlama 1990'da bir zirveye ulaşarak bulanık denetimin ev aletlerinden borsa portföyü denetimine, fotoğraf makinalarından hasta izleme uzman sistemlerine kadar uzanan çok geniş bir yelpaze içerisinde kullanılması ile sonuçlanmıştır. Günümüzde artık bulanık denetim uygulamalarına yönelik özel yazılım ve donanımlar piyasadan hazır bir şekilde temin edilebilmektedir. Örneğin Omron firması bulanık denetim benzetim paketleri ve bulanık mikro işlemci olarak adlandırdıkları özel tümleşik devreler pazarlamaktadır. Benzer paketler Togai firması pazarlamaktadır.

Bulanık denetimin Japon toplumu içerisinde gördüğü yüksek oranda kabule bir örnek bir ,japon ev kadınının, çocuk yetiştirmede kazandığı deneyimlerden yararlanarak geliştirdiği bebek bakım uzman sistemidir. Bu bulanık uzman sistem, çocuğun karakterini, fiziksel durumunu ve çevre koşullarını da içeren bir bilgi tabanına dayandırılmıştır ve anneye bebeğine ne kadar süt vermesi gerektiğini söyler.

Sistem anneler arasında oldukça popüler olmuş ve mucit anneyi zengin etmiştir. Diğer benzer bir uzman sistem de Maruman firması tarafından geliştirilmiş olup golf oyuncusuna golf sopasının seçiminde yardımcı olur[11].



Yukarıda da değinildiği gibi bulanık denetim günümüzde, özellikle Japonya'da Çok geniş uygulama alanları bulmuştur. Bunlardan birinci düzeyde amaç, uzman Operatör yerine bulanık kural tabanlı bir sistem kullanılmasıdır. Sendai metrosu, Çimento değirmeni, asansör denetimi, vs bu yüzeyde yer alır. Daha üst ikinci bir düzey ise hedef, bir uzman yerine bulanık mantık temelli bir karar verici sistem kullanımıdır. Örnek olarak Yamaichi Securities firmasının geliştirdiği uzman sistem verilebilir. Bu sistem, borsada kara Pazar olarak bilinen 1988 yılının Ekim ayındaki büyük çöküşü 18 gün önceden haber verebilmiştir[11].

3.3.6. Bulanık Denetimin Avantajları- Dezavantajları

Günlük hayatta karşımıza çıkan denetim problemlerinin büyük bir kısmı en iyi olarak dilsel niteleyicilerle açıklanabilir, ayrıca denetimin de çok hassas olması gerekmez. İşte bu tür matematiksel birçok problemde, bulanık denetim insan düşünüş tarzına uygun düştüğü için belirgin bir avantaja sahiptir.

Bulanık denetimin en iyi uygulama alanları doğrusal olmayan, iyi tanımlanmamış (ill- defined), zamanla değişen sistemlerdir. Bu tür sistemlerin denetimine kolay anlaşılır bir çözüm getirir. Eğer sistem basit bir matematiksel ilişki ile tanımlanabiliyorsa o zaman belki de geleneksel (klasik), Pm (Oransal, İntegral ve Türevsel) türü basit bir denetim yaklaşımı yeterli olacaktır. Fakat sistem karmaşılaştıkça geleneksel veya modern denetim kuramlarının uygulanması o derece güçlü bir ve bulanık denetime yönelmek ekonomik bir alternatif olarak ortaya çıkar[11].

Bulanık denetim genellikle daha küçük bir yazılımla, daha hızlı bir şekilde sonuca ulaşır. Bunun nedeni bulanık mantıkta işaretlerin bir ön işlemeye tabi tutulmaları ve geniş bir alana yayılmış değerlerin az sayıda üyelik işlevlerine indirgenmeleridir. Bu az sayıda değerler üzerinde uygulanacak kural sayısının da az olması dolayısıyla sonuca ulaşılması daha az zamanda gerçekleşir. Geleneksel bilgisayar ortamında durum böyle iken özel geliştirilmiş bir donanımla hızın daha da arttırılabileceği açıktır. Sanyo Fisher (ABD) firmasının mühendislerini video kayıt

cihazında bulanık denetim kullanmaya iten asıl nedenlerden biri kullanmayı düşündükleri mikrobilgisayarın belleğinin gerçekleştirmeyi düşündükleri tüm işlevler için yetersiz kalması olmuştur. Bulanık denetim kullanmak gerekmemiştir. Artık video kayıt cihazları ve SLR fotoğraf makinalarında bulanık mantık denetleyici çok yaygın bir duruma gelmiştir. Böylece kayıt cihazlarının elle tutulmasından dolayı resimde oluşan sallantılar ortadan kaldırılmış, fotoğraf makinalarında odaklama daha iyi bir şekilde (%76.3 başarı oranına karşı %96.5 başarı oranı) ve merceğin ileri geri gidip gelmesi durumu ortaya çıkmadan hızla yapılabilir olmuştur[11].

Bulanık denetimin diğer bir avantajı, doğrudan kullanıcı girişlerine, kullanıcının deneyimlerinden yararlanılmasına imkan sağlamasıdır. Buna en güzel örnek elektronik olarak sürekli değiştirilebilen aktarımdır. Bilindiği üzere geleneksel otomotik vites değiştirmede vites değişimi motorun belirli hızlara ulaşması ile otomatik olarak gerçekleşir. Manuel vitesli bir arabada ise sürücü daha fazla bir hürriyete sahiptir ve yol, yük ve kendi kullanım stiline göre belirli durumlarda vites değiştirir. Subaru'nun justy tipi otomobilinde kullanılan ECTV-II değiştirilmesi, bir kaybın konumunun, bulanık mantık kullanılarak değiştirilmesi ile sağlanır. Böylece arabanın ivmesi ve performansı sürekli olarak ayarlanabilir. Subani test şöförlerine bu arabayı kullandırmış, ivme ve performans açısından onlara en iyi gelen aktarım oranını öğrenerek bulanık mantığın üyelik işlevlerini bu bilgiye göre ayarlamıştır.

Otomotik vites değişiminde bulanık mantık yaklaşımı konusunda Honda ve Nissan gibi bazı diğer araba üreticileri de çalışmalar yapmaktadırlar. Hedefler daha rahat bir yolculuk, daha az benzin tüketimi, mekanik aksamda daha az yıpranma ve vites değişiminde bir insanın yaptıklarına benzerlik olarak sıralanabilir. Otomotiv sanayinde üzerinde çalışılan diğer uygulama alanları, otomatik benzin enjeksiyonu ve kaymasız frenleme (ABS) alanlarıdır[11].

Uygulamalara örnekler verilirken, geleneksel denetleyicilerle kıyaslamalar yapılmış ve bulanık mantık temelli denetleyicilerin geleneksel denetleyicilere kıyasla daha iyi bir başarımla sonuçlanabileceği açıklanmıştır. Buna diğer bir örnek Mitsubishi'nin bulanık denetimli klima cihazlarıdır. Bu cihazlarda tek bir algılayıcı (sıcaklık algılayıcısı) kullanılmakta ve bu algılayıcıdan elde edilen sıcaklık hatası ve sıcaklık değişimi bilgilerin kullanımı ile ısıtma/soğutma denetlenmektedir. Kullanılan kural sayısı 50 tanedir. Sistemin ilk tasarımı 1988'de, benzetim

çalışmaları 1988 yazında yapılmış, üretime ise Ekim 1989'da geçilmiştir. Geleneksel denetleyici ile karşılaştırıldığında ısıtma ve soğutma sürelerinin 5 katı azaldığı, sıcaklık kararlılığının 2 misli arttığı, kullanılan güçte ise %20 oranında bir azalma olduğu görülmüştür. Bu örnekte başarımdaki iyileşme kadar tasarımdan üretime geçme süresindeki kısalığına da dikkat etmek gerekir[11].

Bulanık mantık denetleyicilere yöneltilen bir eleştiri bu tür denetleyicilerin süreç hakkında daha fazla bilgi, dolayısıyla da daha pahalı ve güvenilirliği daha az bir çözüm oldukları yönündedir. Bu her zaman doğru değildir. Örneğin yukarıda açıklanan Mitsubishi klima cihazında geleneksel denetleyicili cihazda kullanılan daha az (sadece bir tane) algılayıcı kullanılmıştır. Bir başka örnekte, National Panasonic firmasının piyasaya sürdüğü bulanık mantık denetleyicili çamaşır makinesinde, iki tane algılayıcı kullanılmaktadır fakat bunlar ucuz, hassas bilgi vermeyen algılayıcılardır. Bunlardan biri basit bir optik algılayıcı olup kazandaki suyun türbülansı, kirlilik derecesi ve kirlilik türü hakkında kaba bilgiler verir. Diğeri ise ağırlık sezicisidir ve kazandaki çamaşır hacmini ölçtüğü gibi uygun su akışı ve hacmini ayarlar. Bu iki algılayıcıdan gelen bilgiler ışığı altında çamaşır yıkama süresi seçilir.

Yapılan ikinci bir eleştiri bulanık mantık denetleyicilerin doğrusal denetleyicilere kıyasla sağladığı daha iyi performansın bir doğrusal olmayan denetleyici aracılığıyla da sağlanabileceği yönündedir. Bu gerçekten de doğru olabilir, fakat büyük bir olasılıkla doğrusal olmayan denetleyici bir 4 bitlik mikroişlemci ile gerçekleştirilemeyecektir.

Yukarıda açıklanan yararlar, bulanık denetimi endüstriyel uygulamalar için uygun bir aday konumuna sokmuşlardır. Bununla birlikte uygulamada karşılaşılan bazı güçlükler de yok değildir ki bunlar şöyle sıralanabilir:

1. Bulanık denetimde kullanılan kurallar deneyime çok bağlıdır.
2. Üyelik işlevlerinin seçiminde belirli bir yöntem yoktur. En uygun işlev deneme ile bulunur.
3. Denetlenen sistemin bir kararlılık analizi yapılamaz, sistemin nasıl cevap vereceği önceden kestirilemez. Yapılabilecek tek şey benzetim çalışmasıdır.



Yukarıda değinilen güçlüklerden ikincisi konusunda son zamanlarda bazı aşamalar kaydedilmiştir. Yapay sinir ağları ile bulanık mantık kuramının entegrasyonu önerilmiş (Neuro-fuzzy Systems) ve üyelik işlevlerinin bu tür bir yaklaşımla otomatik olarak öğrenilmesi önerilmiştir. Hatta Sanyo firması piyasaya "Neuro-fuzzy Fan" adı altında bir fan sürmüştür.



4. KAVŞAK TRAFİK DENETİM UYGULAMASI



4.1. Trafik Kontrol Sistemleri

Trafik kontrol sistemleri konusunda, ileri düzeyde tesis edilmiş bir şehir trafik kontrol sisteminde şu Yöntemlerin kullanıldığını görmekteyiz.

- 1.) Akıllı olmayan bireysel trafik ışıkları.
- 2.) Yeşil çizgi uygulamalı ardışık kavşaklar.
- 3.) Akıllı yerel trafik ışıkları.
- 4.) Merkezi bilgisayar kontrollü trafik ışıkları

Kullanılan bu dört yöntemi kısaca açıklayalım:

Akıllıca olmayan bireysel trafik ışıkları kontrol sistemi periyodik olarak çalışır. Bu periyodun görev çemberi günün değişik saatlerindeki trafik akışına göre sabit yada ayarlanabilir olabilmektedir. Bu kontrol sisteminde sürücüler ve yayalar geçişlerini engelleyen bir akış olmadığında bile, ışıktaki beklemek zorundadırlar. Bu bekleme ise zaman ve enerji kaybına neden olmaktadır.

Aynı başlangıç zamanına göre ayarlanmış ve bağımsız olduğu halde taşıtın ulaşabileceği her kavşakta yeşil ışıkla karşılaşması garanti edilmiş zaman ayarlamalı sistemlerdir. Bu sistem dikkatlice yerleştirildiği zaman sürücüler yasal hızla seyrettikleri ve anormal bir durum olmadığı sürece hiçbir ışıktaki beklemek zorunda kalmazlar. Ancak senkronizasyon geri beslemesi olmadığından zamanla uyumunu kaybedebilir.

Diğer bir yöntem olan yerel ve akıllı trafik ışıklarında bir kavşağın trafiği, her kavşağın trafik yüküne göre kontrol edilir. Sistem içinde geri besleme olmasına rağmen kavşaklar arasında hiçbir haberleşme yoktur. Bu nedenle ışıklar bağımsız olarak çalışırlar.

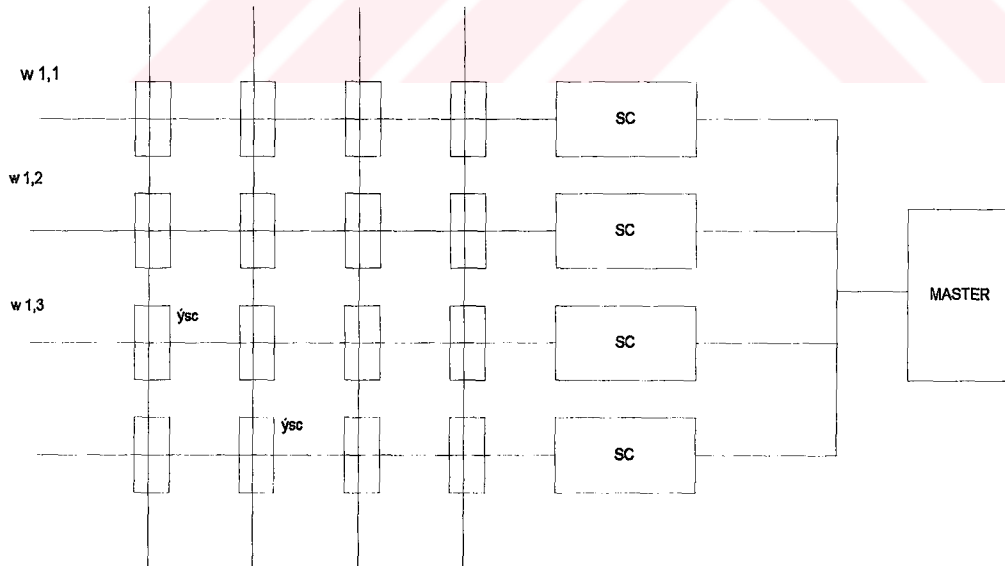
Bugünün en gelişmiş trafik kontrol sistemi, merkezi bilgisayarlarla kontrol edilmiş olanlardır. Bu metot ilk olarak Kanada’da kullanılmaktadır. Bu sistemde, merkezdeki bilgisayar, alıcılar yoluyla kavşağa giren yolların trafik yükü hakkında

bilgi toplar, değerlendirir ve trafik ışıklarını kontrol eder. Merkezdeki bilgisayar ve alıcılar arasındaki haberleşme network ile gerçekleştirilebilmektedir.

Akıllı bir trafik kontrol sistemini tesis etmek istediğimizde sistemde bir geri besleme bir de tekrar birimi bulundurulmalıdır. Başka bir deyişle kapalı çevrim otomatik kontrol sistemine gerek duymaktayız. Geri besleme yolu üzerinde, yolların trafik yoğunluğunu algılayan alıcılar bulunmalıdır. Hatta mimari daha da ileri götürülerek dağıtılmış bilgisayar kontrol sistemi kurulabilir.

Sistemde her kavşağa bir kavşak bilgisayarı yerleştirilir. Aynı yol üzerindeki kavşak bilgisayarları aynı haberleşme hattı üzerinden yol bilgisayarına bağlanır ve tüm yol bilgisayarları da başka bir haberleşme hattı ile merkezdeki asıl bilgisayara bağlanırlar. Bu iki haberleşme çizgisini daha iyi ayırabilmek için birincisini veri anayolu I, ikincisini veri anayolu II olarak tanımlarız. Böylelikle seçilen dağıtılmış bilgisayarlar kontrol mimarisi vasıtasıyla karar birimi birçok bilgisayarın içinde bulunacaktır.

Yukarıdaki açıklamaların ışığında dağıtılmış bilgisayar kontrollü trafik sisteminin genel yapısını (Şekil 4.1) deki gibi çizebiliriz:



Şekil 4.1. Trafik kontrol sisteminin genel yapısı

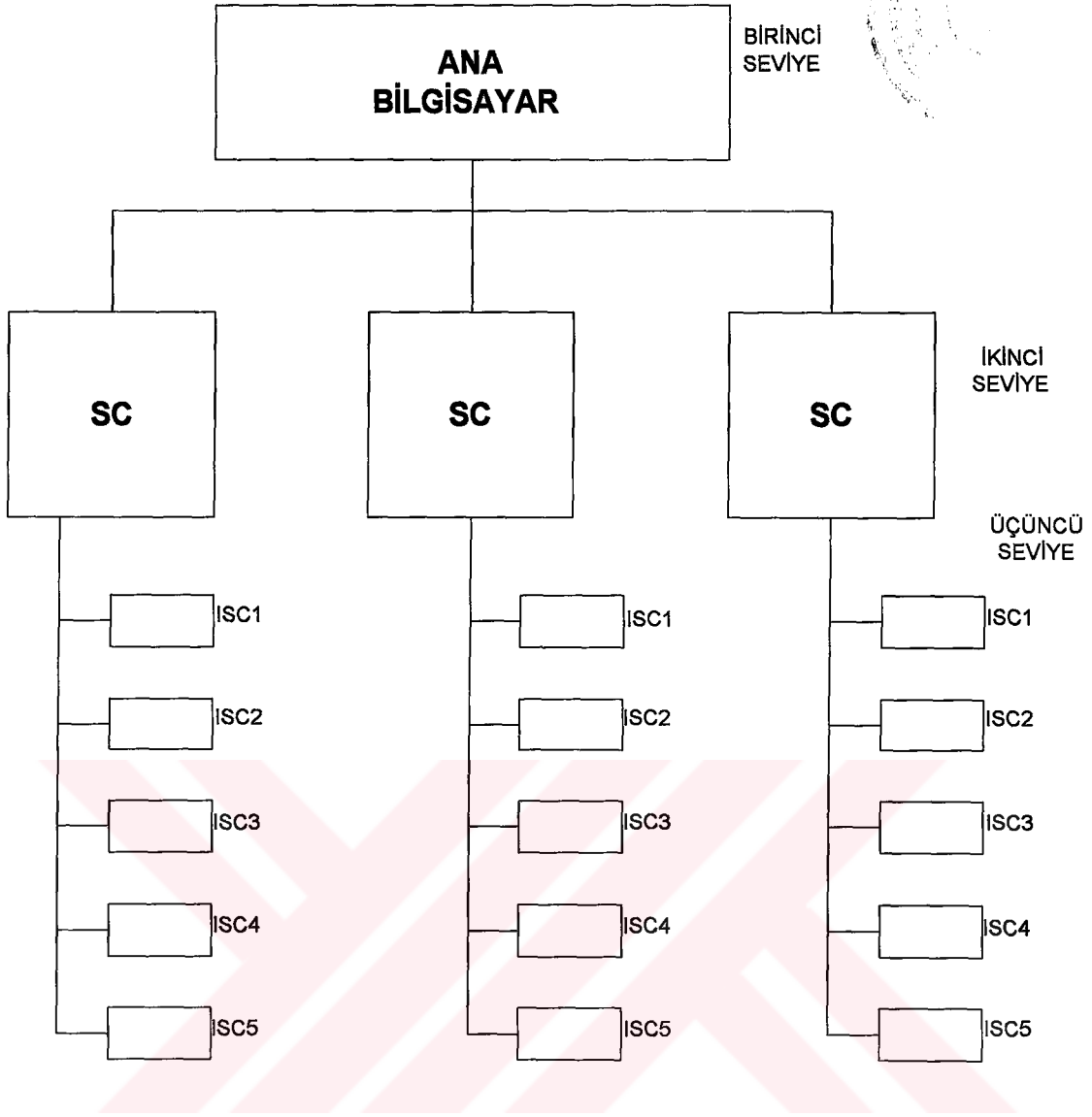
Dağıtık bilgisayar kontrollü trafik kontrol sisteminin bazı üstünlükleri vardır. Bunlar aşağıda belirtilmiştir:

1.) Düşük Maliyet: Dağıtılmış bilgisayar kontrollü trafik kontrol sistemi birçok sayıda mikro bilgisayara gerek duymaktadır. Bu sistemin maliyeti merkezi bilgisayar kontrollü trafik sistemi uygulamasından daha fazla imiş gibi görünüyorsa da aynı veri anayoluna bağlanmış birçok mikrobilgisayarın maliyeti aslında büyük bir merkezi bilgisayar sisteminden daha düşüktür. Yani veri anayolunun kullanımı montaj maliyetini azaltmada bize yardımcı olur.

2.) İlerde Düzeyde Kontrol Kolaylığı: Çok seviyeli kontrol mimarisi vasıtası ile dağıtılmış bilgisayar kontrollü trafik sistemi, ileri düzeyde kontrol kolaylığı sağlamıştır. Bu sistem Şekil 4.2' de belirtilmiştir.

3.) Esnek Tasarım Kolaylığı: Veri anayolu tekniği nedeni ile daha esnek ve daha kolay tasarım özelliği kazanmıştır. Böylelikle mevcut olan sisteme yeni bir kavşak birimi eklemek kolaylaşmıştır.

4.) Kendi Kendini Kontrol Özelliği: Dağıtılmış bilgisayar kontrollü trafik sistemi mimarisi, kendi kendini kontrol imkanını sağlamıştır. Kavşak birimi, sensörlerin ve ışıkların durumunu kontrol etmekle, yol bilgisayarı ise kendine bağlı olan kavşak birimlerinin gözleminden sorumludur (Şekil 4.2).



Şekil 4.2. Kavşak Bilgisayarlarının ana bilgisayara bağlantı şeması

4.2. Kavşak Kontrolünün Formüle Edilmesi

Trafik akımlarının kontrol altına alınması, sürücü ve yayalara gerekli bilgilerin verilmesi ikazların yapılması için diğer trafik ışıkları ile kullanılan ve genel olarak elektrikli renkli ışıklarla çalışan trafik işaretlerine SİNYAL denir. Bunların oluşturduğu sisteme de SİNYALİZASYON SİSTEMİ denir.

Bir sinyal sistemi içinde sistemin 24 saatlik veya bir haftalık programı boyunca aralarında zaman ve yanış süresi bakımından hiçbir farklılık göstermeyen tüm oto, yaya ve flaş vericilerin tümüne birden SİNYAL GRUBU denir.

Bir kavşaktaki sinyal gruplarının hangi saniyede hangi sinyali gösterdiklerini belirten diyagramlara SİNYAL DİYAGRAMI denir.

Bir kavşaktaki sinyal gruplarının tümü birden dikkate alındığında bunların en az birinde bir değişiklik meydana geldiğinde an, bir STEP olarak adlandırılır.

Bir kavşağın sinyal diyagramının uygulanması esnasında bir stepten hareketle, bütün sinyal diyagramını işleyip tekrar aynı stepe gelmesi için geçen süreye PERİYOD denir.

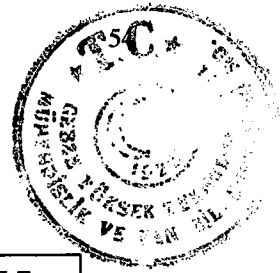
Bir kavşağın sinyalle edilmesi esnasında bu kavşakta uygulanabilecek birbirinden farklı trafik akış kombinasyonlarından her birine FAZ denir.

Bir kavşakta uygulanan farklı sinyal diyagramlarından herbirine PROGRAM denir. Bir kavşakta uygulanan iki muhtelif programın arasında şekil bakımından bir fark varsa bu iki değişik programa STRÜKTÜR PROGRAMI denir. Bir kavşakta uygulanan iki ayrı programın arasında şekil bakımından bir fark yoksa bir veya birkaç stepin süreleri arasında fark varsa bu iki programa ZAMAN PROGRAMI denir.

4.2.1. Periyod Süresi Hesabı

Sinyalize sistemlerinde periyod kesişen veya ayrı ayrı fazlarda hareket eden muhtelif kolların yeşil zamanlarının veya ve kayıp zamanlarının toplamına eşittir.

$$C = G' + G'' + \dots + G^n + \Delta t \quad (4.1)$$



Bu formülde;

G: Her bir bağımsız akımın sahip olduğu yeşil zamandır.

Δt : Kayıp zamanların toplamıdır.

SARI+KIRMIZI (t_1)	YEŞİL ZAMAN(G_1)	SARI ZAMAN	KIRMIZI Z.
KIRMIZI ZAMAN	KIRMIZI+SARI ZAMAN (t_2)	YEŞİL ZAMAN (G_2)	SARI Z.

PERİYOD (C) →

Şekil 4.3. Kayıp zamanlar

Şekil 1 ve 2 de görülen 2 fazlı sistemin için periyot süresi;

$$C = G' + G'' + \Delta t \quad (4.2)$$

formülü ile belirlenir.

4.2.2. Kayıp Zaman Hesabı

$$\Delta t = n \cdot 2 + R + t_{s1} + \dots + t_{sn} \quad (4.3)$$

periyot süresinin bir kısmı da kayıp zamandır. Bu araçların hareket etmediği zaman olarak adlandırılır.

1. R bütün araçların durduğu sadece yayaların geçtiği zaman, her zaman kullanılmaz.
2. Sarı + Kırmızı zamanlar (3 saniye)
3. Bir taşıt dizisinde sonuncu taşıtın kavşağı emniyetle boşaltma zamanı veya koruma zamanı t_s ; bu süre kavşağın geometrisine bağlı olarak değişir.

Bir yönün sonuncu taşıtı kavşağı boşaltma zamanı (t_R) ile kesişen yönün kavşakta çarpışma noktasına gelene kadar geçen zaman (t_A) farkları emniyetli koruma zamanı olarak düşünülür.

Kavşağı boşaltma zamanının 1. Saniyesi ile 4. Saniyesi arasında arabaların hızının 7 m/s olarak gidildiği kabul edilirse, 5. Saniyeden sonra 8 m/s olarak kabul edilir.



L_R : Kavşağı boşaltma mesafesi (m)

L_A : Kavşağa giriş mesafesi (m)

t_A : Kavşağa giriş zamanı (s)

t_R : Kavşağı boşaltma zamanı (s)

olmak üzere;

$L_R < 28$ metre için

$$t_R = \frac{L_R + 5,00}{7,00} \quad (4.4)$$

$L_R > 28$ metre için

$$t_R = \frac{L_R + 5,00 - 28,00}{8} + 4,00 = \frac{L_R + 9}{8} \quad (4.5)$$

t_A zamanının hesabı için şekil 4.4' de görüldüğü gibi kavşağa giren bir vasıtanın kavşak dur çizgisini yeşil ışığın başında ve durmadan geçip, çarpışma noktasına gelirse, bu tehlikenin önlenmesi için A vasıtasına verilecek yeşil ışık başlangıcı geciktirilmelidir. Kavşağa giriş hızı, $X = 50$ Km/saat olarak düşünülürse:

$$t_A = \frac{L_A}{13,86} \quad (4.6)$$

Sonuç olarak emniyetli geçiş zamanı t_s ;

$$t_s = t_R - t_A \quad (4.7)$$

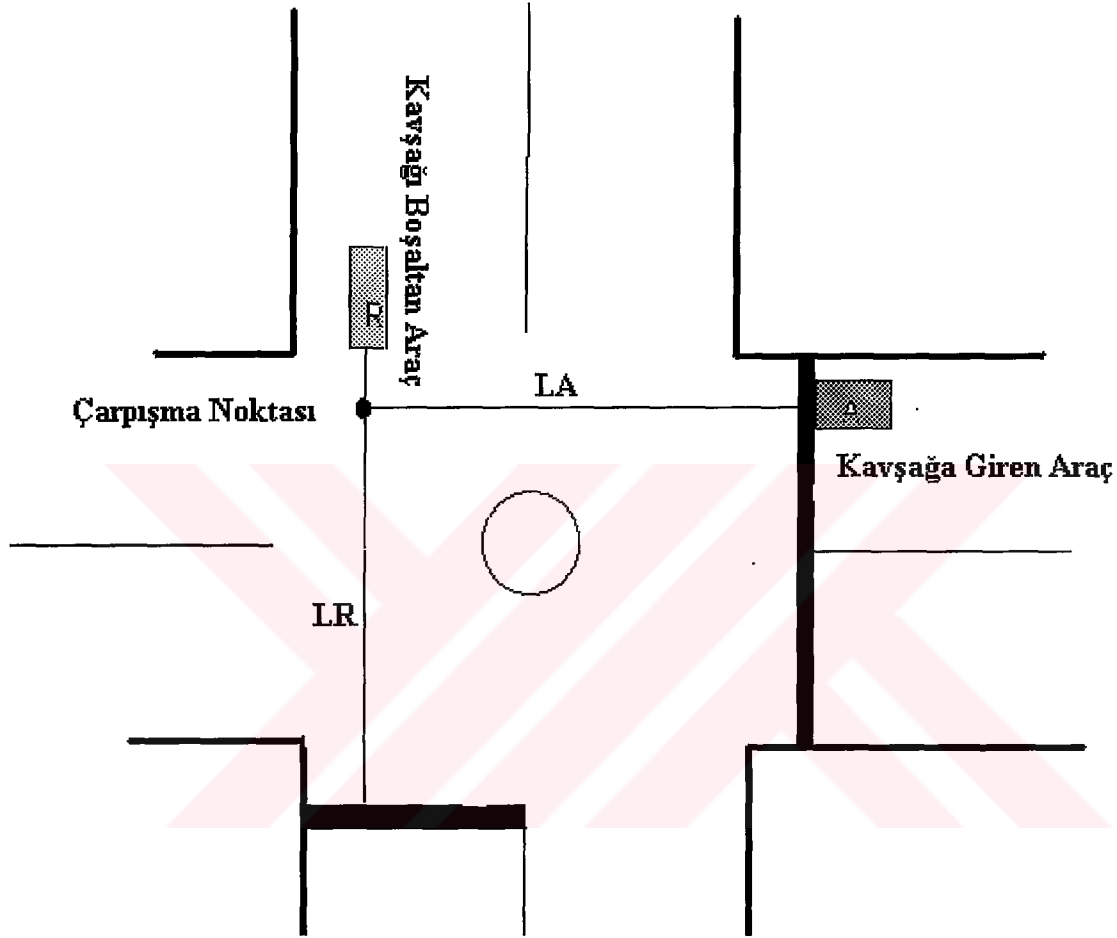
$L_R < 28$ metre için

$$t_s = \frac{L_R + 5,00}{7,00} - \frac{L_A}{13,86} \quad (4.8)$$

$L_R > 28$ metre için

$$t_s = \frac{L_R + 9,00}{8,00} - \frac{L_A}{13,86} \quad (4.9)$$

Burada t_s yeşil ışığın sonunda 3 saniye olarak sarı zamanın sonuncu birinci saniyesinden itibaren sayılır.



Şekil 4.4. Çarpışma noktası



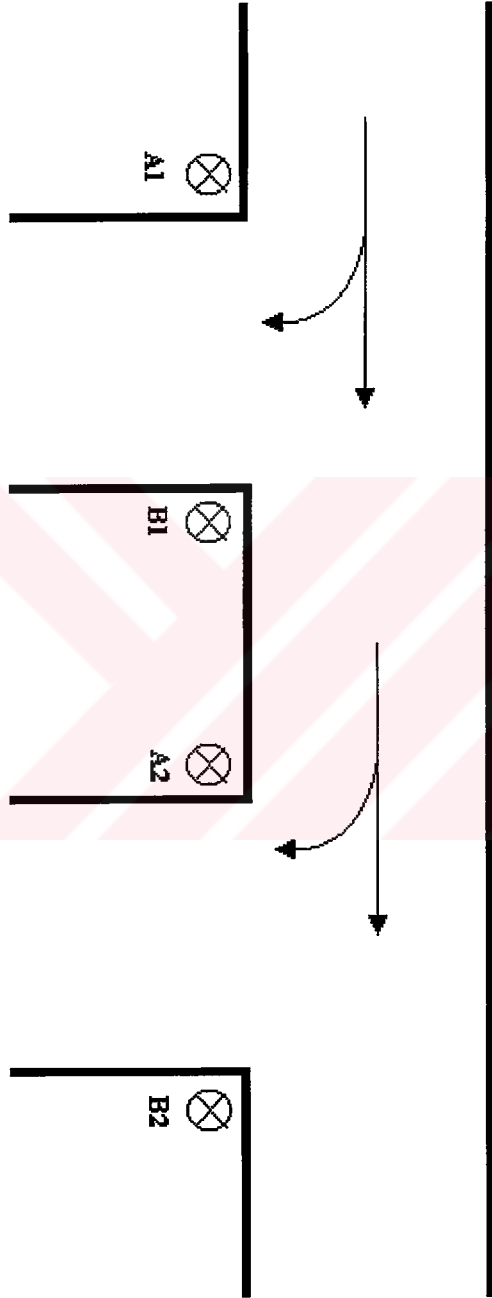
4.3. Kavşak Trafik Işıkları Sürelerinin Bulunması

Kavşaklardaki trafik denetimi için önerilen yöntemler iki ana küme halinde toplanabilir. Bunlardan ilkinin oluşturanlar sinyal döngü sürelerini sabit tutup yeşil ışık sürelerini en iyiye amaçlayan, normal ve doymuş akışların bilindiğini varsayan modellerdir. İkinci kümeyi oluşturan modellerde de, yollardaki normal ve doymuş akışların bilindiği varsayılır ve en iyi yeşil yanma süreleri hesaplanır. Ancak bu süreler bulunurken farklı olarak kavşağı oluşturan akışlardaki kesişmeler ve uyumlarda gözönüne alınır.

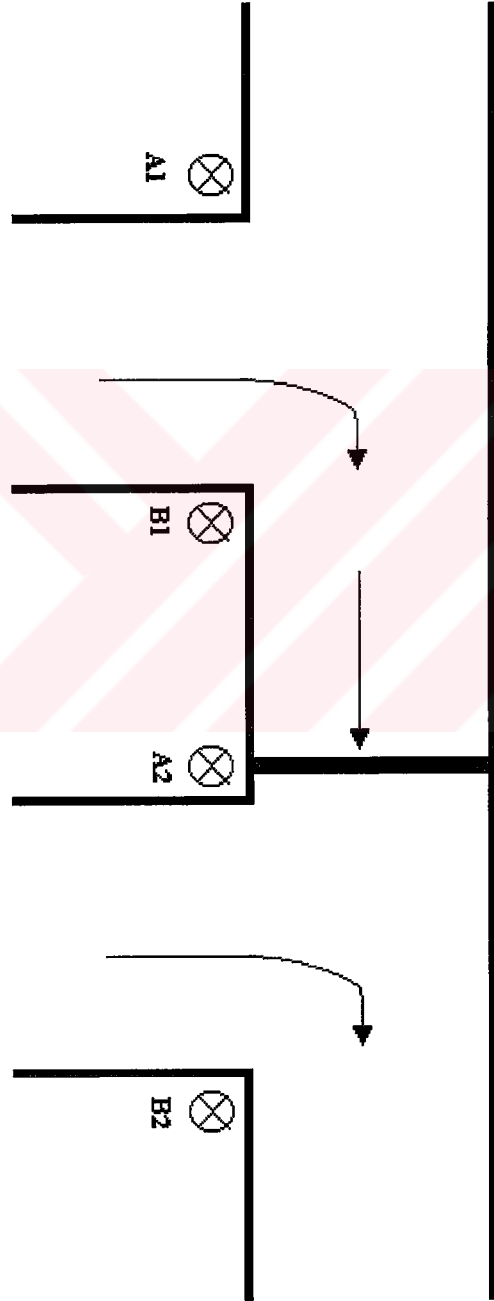
Tüm modeller kullanılarak hem kuramsal, hem de uygulanabilir sonuçlar elde edilmiştir. Ne var ki problemin karmaşık yapısı bu sonuçların ancak basit kavşaklar için elde edilmesine imkan sağlamıştır.

Klasik metotların çoğunda, kavşaklardaki trafik denetimi istatistiki verilere dayandırılarak ışık sürelerini belirleyip değiştirmeye giderken, bulanık denetim ise kavşağa giren araçların sayısını belirleyip buna göre sinyali değiştirmektedir. Bu ise değişken trafik yoğunlukları durumunda daha iyi sonuçlar elde edilmesini sağlamaktadır.

Bu çalışmada sabah 08.00-09.30 saatleri arasında elde edilen gözlem verileri Çizelge 4.1' den Çizelge 4.4' e kadar yeşil ışıktaki geçen, kırmızı ışıktaki bekleyen araçlar gösterilmiştir. Kavşak yapısı ise Şekil 4.5 ve Şekil 4.6' da trafik akışlarıyla beraber gösterilmiştir.



Şekil 4.5. Birinci faza göre hareket eden araçların hareket yönleri

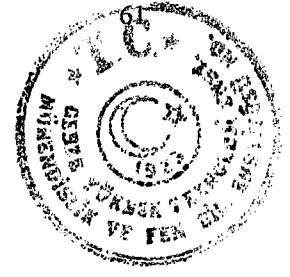


Şekil 4.6. İkinci faza göre hareket eden araçları hareket yönleri

Çizelge 4.1. A1 Girişi Araç Yoğunlukları

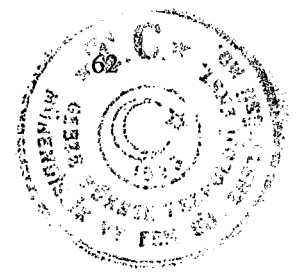
Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
29	11	27	12	32	13
30	13	28	14	37	14
29	16	30	13	36	17
29	13	28	15	41	17
35	14	31	14	37	18
34	15	33	15	36	19
35	16	34	17	38	20
37	16	36	15	38	17
37	15	35	18	39	19
37	18	38	20	39	16
39	19	36	21	35	17
40	17	37	16	37	16
39	17	33	18	36	16
37	21	38	19	33	20
36	20	36	18	37	21
35	19	36	17	35	18
34	20	35	21	36	15
37	18	40	15	35	15
36	19	41	18	35	17

Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
30	17	32	12
27	19	35	16
28	18	37	13
27	20	38	17
27	14	34	15
32	16	36	18
37	16	38	17
38	18	38	19
38	19	39	16
39	15	41	21
40	18	36	15
36	22	35	17
38	24	36	18
39	23	33	18
41	20	36	21
49	19	37	20
39	17	37	17
37	19	40	18
37	21	38	19



Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
32	13	29	10	31	9
30	14	28	12	28	13
35	16	30	14	27	16
36	16	36	15	34	14
36	15	35	15	36	15
34	17	37	16	41	17
37	19	37	19	38	21
38	15	39	22	36	22
39	17	41	24	37	20
38	17	35	23	39	17
37	20	37	20	40	16
41	21	36	18	42	15
41	22	33	15	41	17
40	17	34	19	51	16
40	18	36	16	44	16
38	19	37	18	37	14
37	18	39	19	36	13
37	16	40	18	37	18
36	14	42	14	35	16

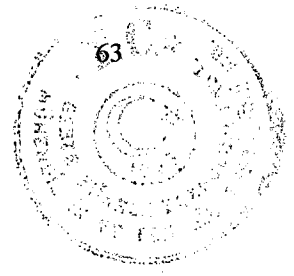
Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
25	11	31	14
32	14	27	17
30	13	29	18
34	16	30	16
40	17	35	13
41	22	42	17
42	20	38	18
38	23	36	19
39	24	38	20
36	19	37	17
40	18	37	19
36	16	40	20
35	21	41	18
33	19	36	14
38	19	37	17
36	20	38	16
34	21	39	17
37	15	34	15
38	17	36	13



Çizelge 4.2. B1 Girişi Araç Yoğunlukları

Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
29	13	30	14	27	13
26	12	32	17	28	13
28	15	31	17	29	14
30	15	35	15	29	16
35	17	33	16	30	17
37	18	36	17	35	15
37	19	38	19	36	15
36	21	37	19	33	14
38	16	37	17	32	17
39	18	39	21	35	18
40	18	40	22	37	21
38	17	41	20	36	20
37	19	37	17	37	19
38	18	33	19	37	18
37	16	36	18	35	17
36	19	37	17	39	16
34	10	35	16	36	17
33	19	38	19	33	18
36	19	37	17	32	15

Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
28	9	27	10
31	11	27	12
32	14	28	11
29	13	29	9
30	15	35	16
34	16	39	16
37	16	38	16
38	18	38	14
38	19	37	18
39	19	36	17
36	18	38	18
36	16	39	18
35	17	40	16
38	17	38	17
39	18	41	17
37	18	42	15
35	17	39	16
34	15	36	19
35	13	36	17



Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
27	11	30	12	31	9
28	12	32	14	29	11
33	15	29	15	27	10
32	16	31	13	30	12
35	17	34	12	33	13
40	18	37	15	29	11
39	19	36	14	34	12
38	18	35	15	36	16
36	18	35	16	38	15
39	17	38	16	40	17
38	16	36	17	39	17
39	15	35	17	37	18
38	16	34	16	38	20
38	17	37	19	33	15
35	16	38	19	37	15
34	17	33	16	35	16
36	18	35	18	35	18
37	19	32	17	36	17
35	18	37	17	37	15

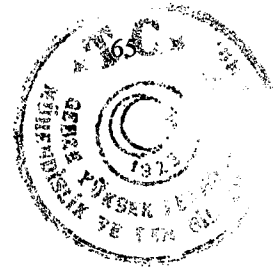
Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
28	9	31	11
29	12	32	13
27	13	34	9
33	14	31	10
34	16	36	16
39	14	38	15
36	15	37	16
37	16	35	16
37	17	35	17
38	20	36	19
35	19	37	17
36	18	39	15
37	17	37	16
39	16	37	18
35	15	37	17
37	16	33	17
35	18	35	16
36	15	31	16
33	17	33	15

Çizelge 4.3. A2 Girişi Araç Yoğunlukları



Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
35	20	33	19	40	17
38	22	35	22	41	16
37	23	34	23	38	18
37	24	35	21	38	21
40	23	35	20	39	22
39	26	36	20	40	23
40	22	39	23	46	21
45	21	39	24	48	24
44	20	39	27	49	27
46	22	36	28	50	26
48	26	37	28	51	27
34	29	39	26	48	29
36	21	45	25	49	29
46	22	36	26	51	28
44	23	46	24	43	25
45	25	45	26	49	22
43	26	48	26	47	24
39	27	47	27	45	25
41	24	45	28	36	26

Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
37	19	33	23
38	17	44	22
36	21	45	22
33	22	45	24
38	26	43	24
41	27	42	28
44	25	46	25
46	24	49	26
49	27	43	25
50	26	43	26
49	23	42	24
48	25	43	25
46	26	43	28
45	21	44	26
34	22	51	28
45	22	48	28
44	25	47	25
47	24	49	26
46	24	48	27



Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
38	19	35	21	38	17
37	17	35	26	45	16
40	16	38	27	45	18
39	20	36	28	41	17
41	18	39	27	40	25
44	21	40	27	46	26
49	24	37	30	47	26
42	21	33	27	48	28
37	22	38	26	49	29
49	19	41	27	46	27
49	24	44	28	47	26
34	23	42	29	45	26
48	27	47	25	44	29
47	26	48	23	46	26
46	25	48	24	47	25
42	27	49	26	47	22
45	27	44	25	48	25
44	28	43	26	46	23
43	25	44	24	45	22

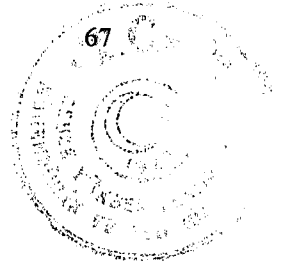
Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
41	21	35	22
42	25	36	23
33	23	41	24
38	23	42	25
40	26	44	24
43	28	48	25
47	29	45	27
46	26	46	28
49	25	49	29
43	20	42	25
46	24	46	27
48	23	48	29
47	26	47	26
47	27	46	28
48	24	47	28
45	25	48	27
45	28	50	26
47	26	51	27
42	27	44	25



Çizelge 4.4. B2 Girişi Araç Yoğunlukları

Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
34	16	36	19	33	18
35	17	35	21	36	21
34	18	34	22	35	24
37	19	37	26	38	24
35	17	37	24	37	26
37	19	40	23	39	25
36	21	41	26	36	27
38	23	37	24	40	26
39	24	38	28	44	25
40	25	36	25	41	28
41	26	38	25	44	29
41	25	41	23	39	27
44	28	42	26	35	27
42	29	43	27	45	25
39	27	39	24	46	29
39	28	37	27	42	27
41	25	41	28	42	30
40	26	38	29	43	26
41	27	35	26	41	27

Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
36	18	34	16
37	16	35	18
34	20	38	20
33	21	35	23
36	22	37	26
36	23	38	27
41	27	36	27
42	28	40	27
45	27	39	25
39	27	44	29
41	29	45	30
44	30	42	32
43	29	40	29
40	29	43	28
41	31	39	29
44	28	38	30
43	29	37	27
39	27	40	29
39	28	41	25



Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
35	16	36	17	37	19
38	17	37	19	34	21
36	19	34	15	39	22
37	20	35	19	41	23
35	19	37	17	38	20
41	19	39	18	39	25
44	21	36	21	39	26
39	23	42	24	40	27
38	26	43	27	44	29
41	28	40	22	45	26
42	27	44	23	42	27
45	27	45	25	43	25
43	25	39	20	41	25
44	28	40	21	43	25
42	29	42	25	39	26
39	27	41	26	38	20
38	27	41	29	40	26
40	28	40	24	38	26
40	28	38	25	38	22

Yeşil Geçen	Kırmızı Bekleyen	Yeşil Geçen	Kırmızı Bekleyen
36	18	33	19
38	21	35	20
39	22	36	21
35	19	38	22
36	27	37	23
38	22	39	21
39	25	40	27
40	26	38	23
41	27	44	24
44	29	42	25
41	30	43	24
39	24	41	24
38	27	42	26
40	26	53	25
43	25	52	24
42	26	41	26
38	25	44	26
40	24	43	27
41	23	40	29

4.3.1. Yeşilde Geçen ve Kırmızıda Bekleyen Araçların Ortalamaları

Çizelge 4.5. A1 Işığı için Yeşilde Geçen Araç Sayısı

<u>Xi</u>	<u>Ni</u>	<u>Xi*Ni</u>
25	1	25
26	0	0
27	6	162
28	5	140
29	5	145
30	7	210
31	3	93
32	5	160
33	6	198
34	11	374
35	17	595
36	30	1080
37	31	1147
38	21	798
39	14	546
40	11	440
41	11	451
42	4	168
43	0	0
44	0	0
45	0	0
46	0	0
47	0	0
48	0	0
49	1	49
50	0	0
51	1	51
TOPLAM	190	6832

$$\bar{X} = \frac{\sum Xi * Ni}{\sum Ni} \quad (4.10)$$

$$\bar{X} = \frac{6832}{190} \cong 36$$



Çizelge 4.6. A1 Işığı İçin Kırmızı Işıқта Bekleyen Araç Sayısı

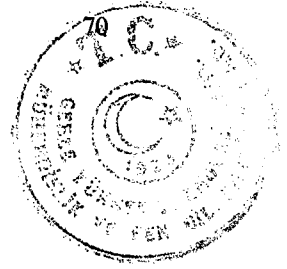
Xi	Ni	Xi*Ni
9	1	9
10	1	10
11	2	22
12	3	36
13	11	143
14	15	210
15	20	300
16	25	400
17	29	493
18	25	450
19	22	418
20	15	300
21	11	231
22	4	88
23	3	69
24	3	72
TOPLAM	190	3251

$$\bar{X} = \frac{3251}{190} \cong 18$$

Çizelge 4.7. B1 Işığı İçin Yeşilde Geçen Araç Sayısı

Xi	Ni	Xi*Ni
26	1	26
27	6	162
28	6	168
29	9	261
30	6	180
31	7	217
32	8	256
33	13	429
34	9	306
35	25	875
36	24	864
37	31	1147
38	22	836
39	15	585
40	5	200
41	2	82
42	1	42
TOPLAM	190	6610

$$\bar{X} = \frac{6610}{190} \cong 35$$



Çizelge 4.8. B1 Işığı İçin Kırmızıda Bekleyen Araç Sayısı

<u>Xi</u>	<u>Ni</u>	<u>Xi*Ni</u>
9	5	45
10	4	40
11	6	66
12	8	96
13	9	117
14	9	126
15	24	360
16	34	544
17	38	646
18	26	468
19	19	361
20	4	80
21	3	63
22	1	22
TOPLAM	190	3034

$$\bar{X} = \frac{3034}{190} \cong 16$$

Çizelge 4.9. A2 Nolu Işıқта Yeşilde Geçen Araç Sayısı

<u>Xi</u>	<u>Ni</u>	<u>Xi*Ni</u>
33	5	165
34	4	136
35	7	245
36	8	288
37	7	259
38	10	380
39	9	351
40	8	320
41	8	328
42	9	378
43	11	473
44	14	616
45	18	810
46	18	828
47	16	752
48	17	816
49	14	686
50	3	150
51	4	204
TOPLAM	190	8058

$$\bar{X} = \frac{8058}{190} \cong 42$$



Çizelge 4.10. A2 Nolu Işık İçin Kırmızıda Bekleyen Araç Sayısı

<u>Xi</u>	<u>Ni</u>	<u>Xi*Ni</u>
16	3	48
17	5	85
18	3	54
19	4	76
20	6	120
21	11	231
22	16	352
23	16	368
24	20	480
25	25	625
26	31	806
27	23	621
28	17	476
29	9	261
30	1	30
TOPLAM	190	4633

$$\bar{X} = \frac{4663}{190} \cong 25$$

Çizelge 4.11. B2 Işığı için Yeşilde Geçen Araç Sayısı

<u>Xi</u>	<u>Ni</u>	<u>Xi*Ni</u>
33	3	99
34	7	238
35	13	455
36	15	540
37	15	555
38	22	836
39	23	897
40	20	800
41	24	984
42	14	588
43	12	516
44	13	572
45	6	270
46	1	46
47	0	0
48	0	0
49	0	0
50	0	0
51	0	0
52	1	52
53	1	53
TOPLAM	190	7501

$$\bar{X} = \frac{7501}{190} \cong 40$$



Çizelge 4.12. B2 Işığı İçin Kırmızıda Bekleyen Araç Sayısı

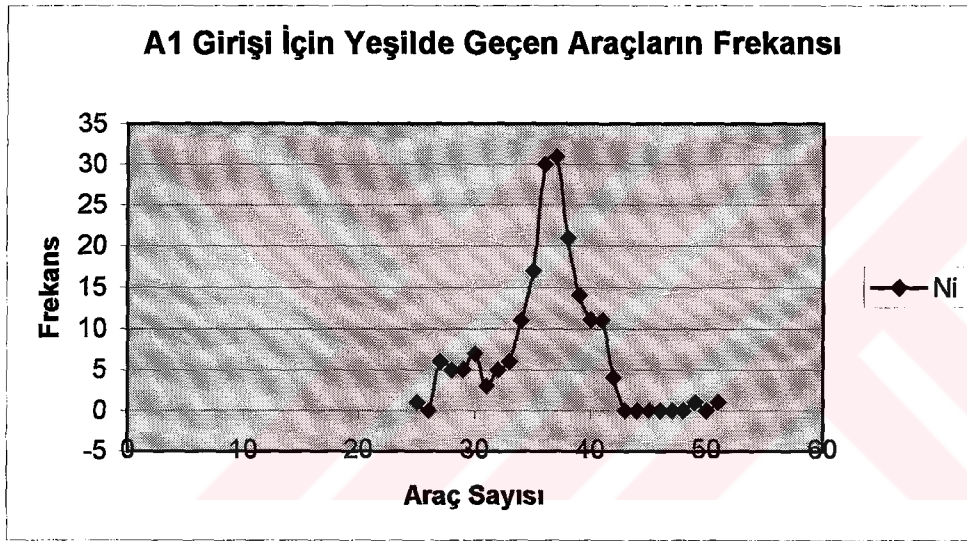
Xi	Ni	Xi*Ni
15	1	15
16	4	64
17	5	70
18	6	108
19	11	209
20	7	140
21	11	231
22	8	176
23	11	253
24	14	336
25	23	575
26	23	598
27	29	783
28	13	364
29	17	493
30	5	150
31	1	31
32	1	32
TOPLAM	190	4628

$$\bar{X} = \frac{4628}{190} \cong 25$$

4.3.2. Girişlerin Yeşil Işıktaki Geçen ve Kırmızıda Bekleyen Araçların Frekanslarının Belirlenmesi

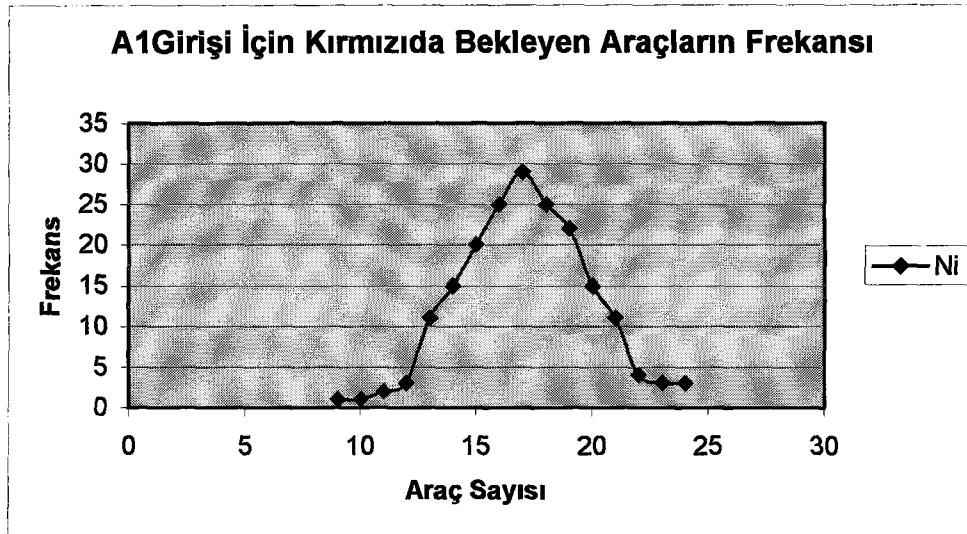
Yeşilde geçen ve kırmızıda bekleyen araçlar için standart sapma hesabı aşağıdaki formüle göre yapılmaktadır;

$$\sigma = \sqrt{\frac{\sum (X_i - \bar{X})^2 * N_i}{\sum N_i}} \quad (4.11)$$



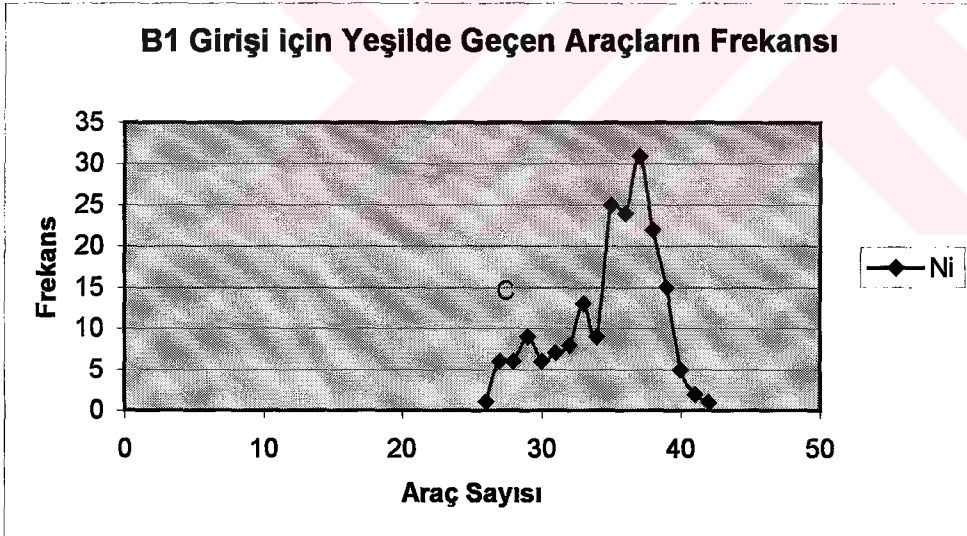
Şekil 4.7. A1 Işığı için Yeşil Işıktaki Geçen Araçların Frekansı

$$\sigma = \sqrt{\frac{2944}{190}} = 3,93 \cong 4$$



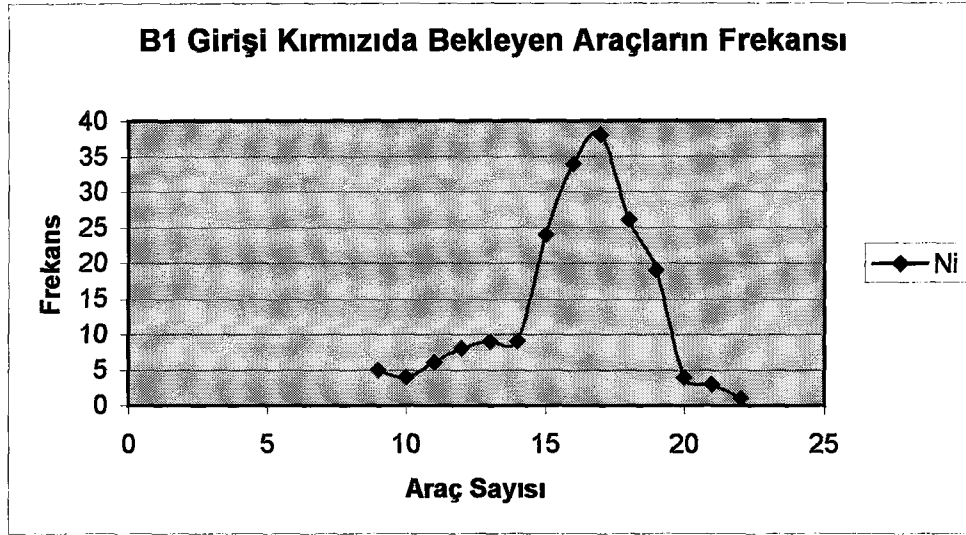
Şekil 4.8. A1 Işığı için Kırmızı Işıktaki Bekleyen Araçların Frekansı

$$\sigma = \sqrt{\frac{1603}{190}} = 2,90 \approx 3$$



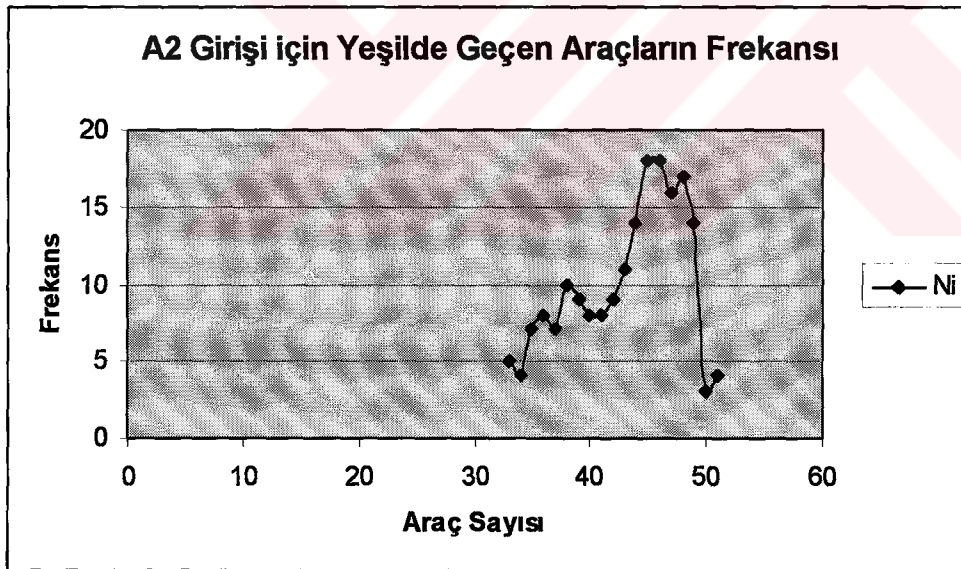
Şekil 4.9. B1 Işığı için Yeşil Işıktaki Geçen Araçların Frekansı

$$\sigma = \sqrt{\frac{2310}{190}} = 3,48 \approx 4$$



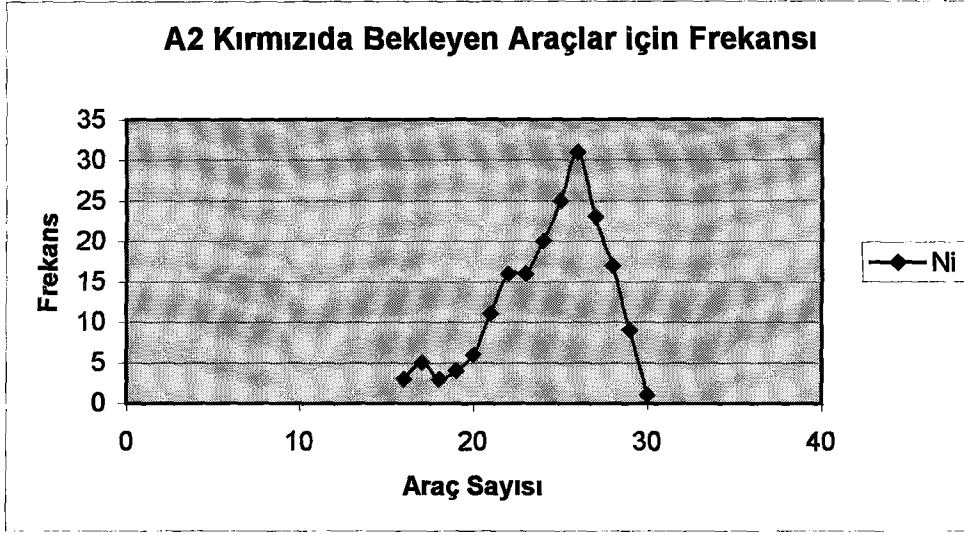
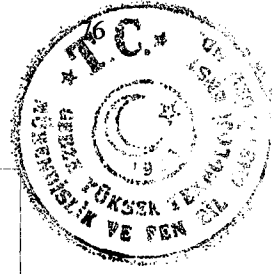
Şekil 4.10. B1 Işığı için Kırmızıda Bekleyen Araçların Frekansı

$$\sigma = \sqrt{\frac{1464}{190}} = 2,77 \cong 3$$



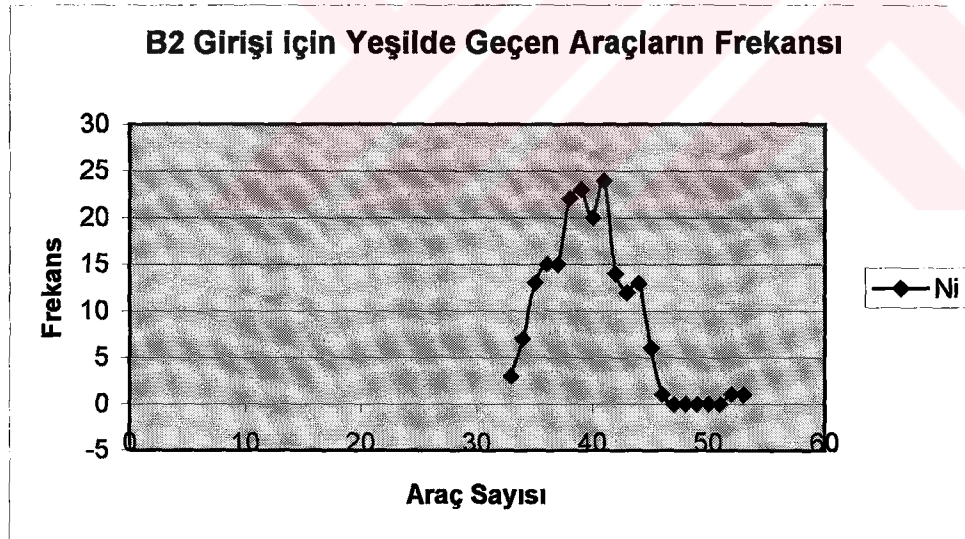
Şekil 4.11. A2 Işığı için Yeşil Işıktaki Geçen Araçların Frekansı

$$\sigma = \sqrt{\frac{4626}{190}} = 4,91 \cong 5$$



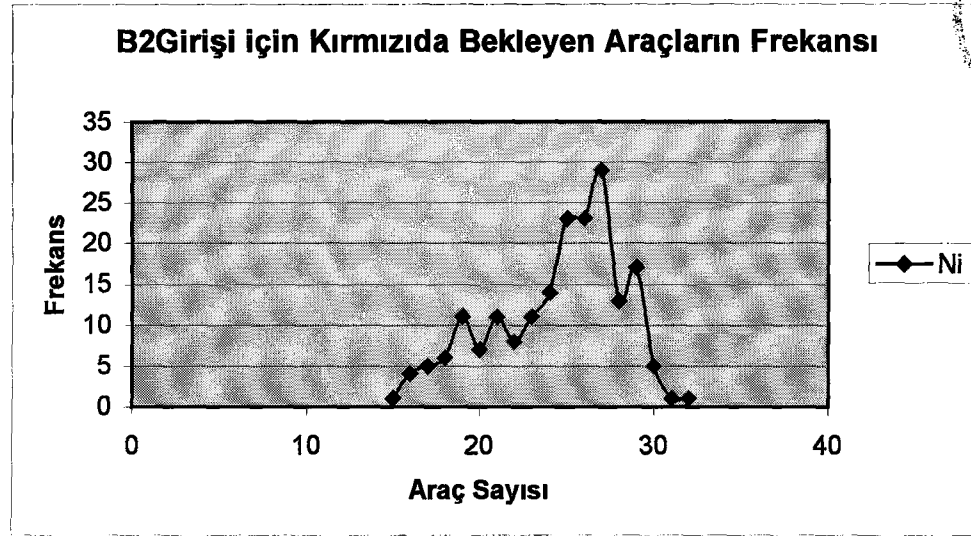
Şekil 4.12. A2 Işığ1 için Kırmızı Işıktaki Bekleyen Araçların Frekansı

$$\sigma = \sqrt{\frac{1853}{190}} = 3.12 \cong 4$$



Şekil 4.13. B2 Işığ1 için Yeşil Işıktaki Geçen Araçların Frekansı

$$\sigma = \sqrt{\frac{2105}{190}} = 3.32 \cong 4$$



Şekil 4.14. B2 Işığ1 için Kırmızıda Bekleyen Araçların Frekansı

$$\sigma = \sqrt{\frac{2653}{190}} = 3,73 \cong 4$$

Çizelge 4.13. Yeşilde Geçen Araçların Ortalamalarının Ortalaması

$\bar{X}_i$	N_i	$\bar{X}_i * N_i$	$\bar{X} = \frac{153}{4} \cong 38$ $\sigma = \sqrt{\frac{33}{4}} \cong 3$
42	1	42	
40	1	40	
36	1	36	
35	1	35	
Toplam	4	153	

Çizelge 4.14. Kırmızıda Bekleyen Araçların Ortalamalarının Ortalaması

$\bar{X}_i$	N_i	$\bar{X}_i * N_i$	$\bar{X} = \frac{84}{4} \cong 21$ $\sigma = \sqrt{\frac{66}{4}} \cong 4$
25	2	50	
18	1	18	
16	1	16	
Toplam	4	84	

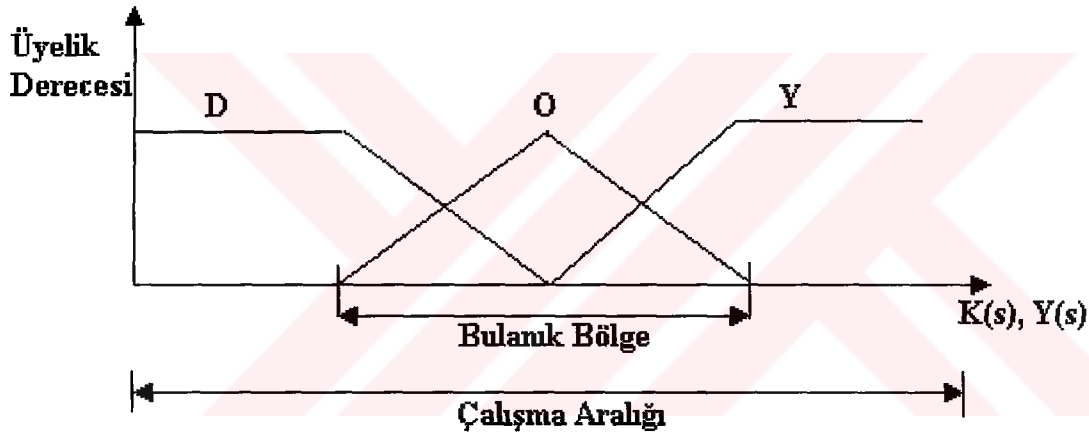
4.3.3. Bulanık Alt Kümelerin Belirlenmesi

Bulanık kümelerin kullanılması son yıllara rastlamaktadır. Bunun sebebi bulanık kuralların mikroişlemcisiz yapılmasının pratik olmamasıdır. Bu çalışmada trafik sistemi Çizelge 4.15’ de görüldüğü gibi bulanık alt küme teorisine uygun olarak 3 adet üyeye ait semboller belirtilmiştir.

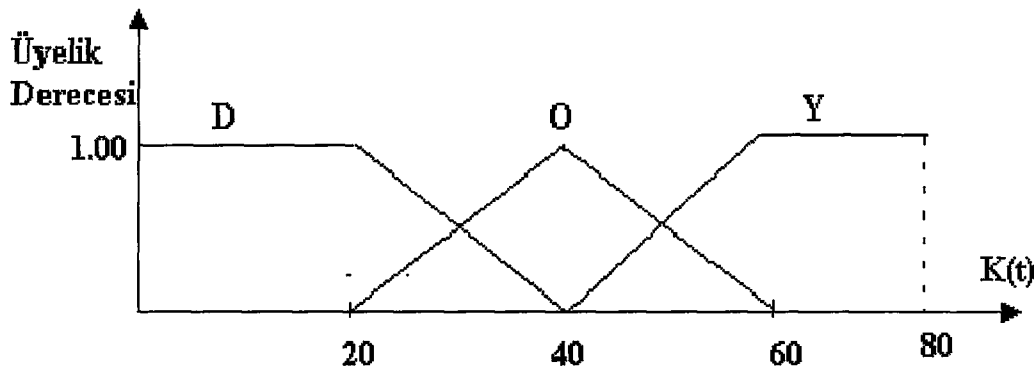
Çizelge 4.15. Bulanık Alt Küme Teorisine Uygun Sembolik Etiketler

Sembol	Bulanık Alt Küme Terimi
D	Düşük
O	Orta
Y	Yüksek

Kırmızı ışıktaki bekleyen, yeşil ışıktaki geçen araç sayısı için üyelik fonksiyonu aşağıda şekil 4.15’ de, çıkış üyelik fonksiyonu şekil 4.16’ da verilmiştir.

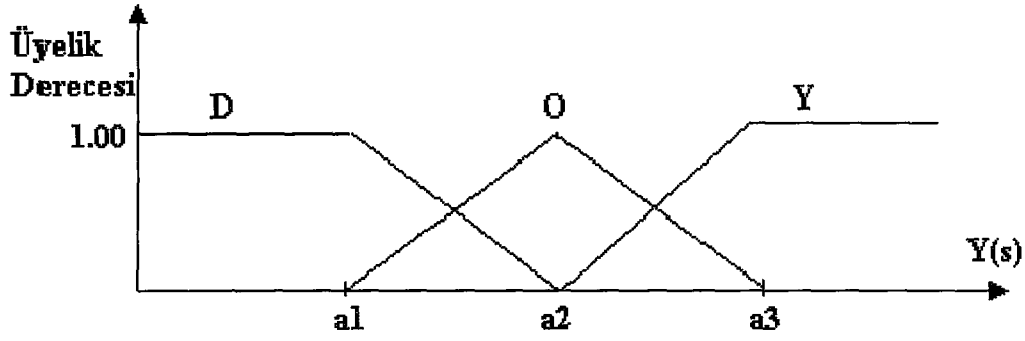


Şekil 4.15. Kırmızı ışıktaki bekleyen yeşil ışıktaki geçen araçlar için üyelik fonksiyonu



Şekil 4.16. K(t) Çıkış Üyelik Fonksiyonları

Bulanık alt küme terimleri için üyelik derecelerinin belirlenmesi Şekil 4.17'ye göre yapılmıştır.



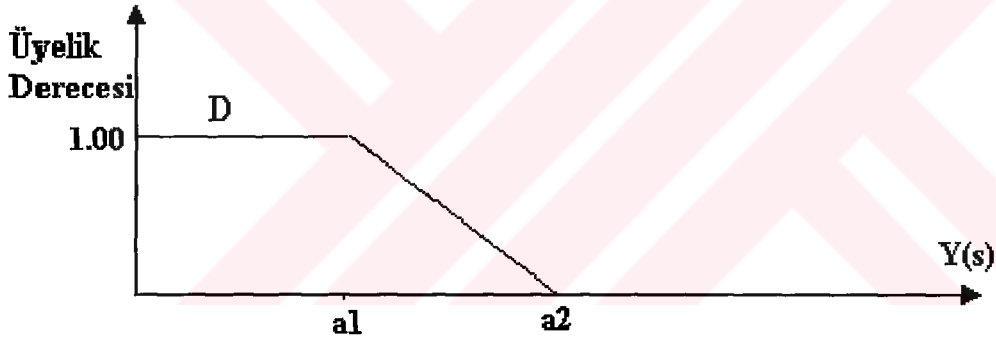
Şekil 4.17. Düşük (D), Orta (O), Yüksek (Y) için Üyelik Derecelerinin Belirlenmesi

$$A = (a_1, a_2, a_3)$$

A: Bulanık küme

a_1, a_2, a_3 : Bulanık alt kümeler

1. Düşük Bulanık Alt Kümesi için Üyelik Dereceleri:

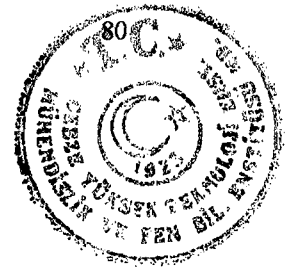


Şekil 4.18. Düşük (D) için Üyelik Derecelerinin Belirlenmesi

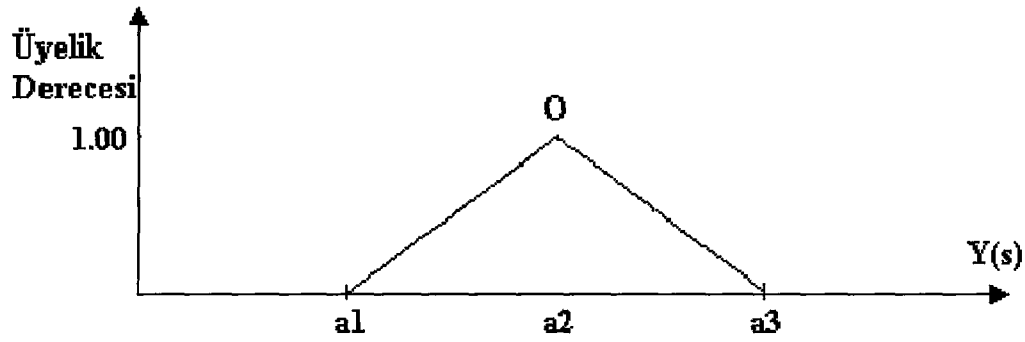
$$\mu_D(x) = 1 \quad 0 \leq x \leq a_1 \quad (4.12.)$$

$$\mu_D(x) = \frac{(a_2 - x)}{(a_2 - a_1)} \quad a_1 \leq x \leq a_2 \quad (4.13.)$$

$$\mu_D(x) = 0 \quad x > a_2 \quad (4.14.)$$



2. Orta Bulanık Alt Kümesi için Üyelik Dereceleri:



Şekil 4.19. Orta (O) için Üyelik Derecelerinin Belirlenmesi

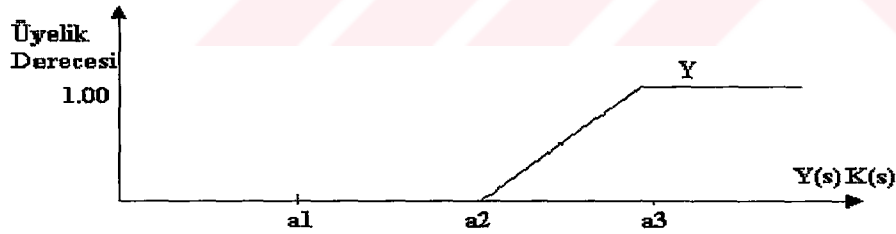
$$\mu_O(x) = 0 \quad x < a1 \quad (4.15.)$$

$$\mu_O(x) = \frac{(x - a1)}{(a2 - a1)} \quad a1 \leq x \leq a2 \quad (4.16.)$$

$$\mu_O(x) = \frac{(a3 - x)}{(a3 - a2)} \quad a2 \leq x \leq a3 \quad (4.16.)$$

$$\mu_O(x) = 0 \quad x > a3 \quad (4.17.)$$

3. Yüksek Bulanık Alt Kümesi için Üyelik Dereceleri:

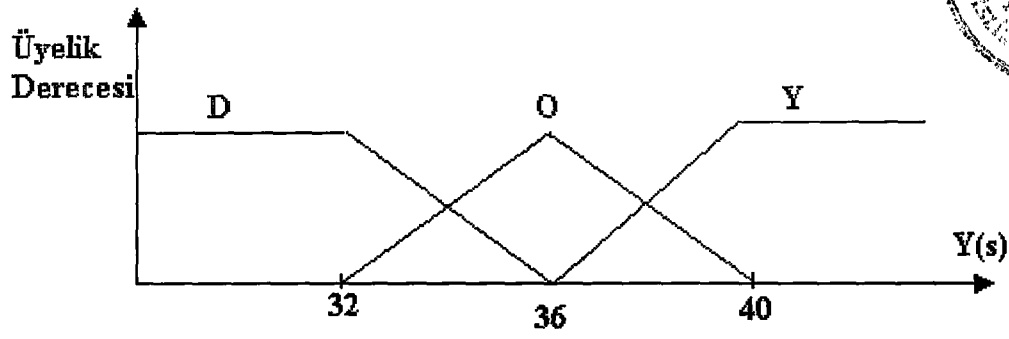
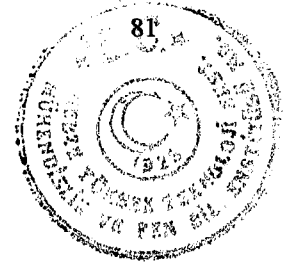


Şekil 4.20. Yüksek (Y) için Üyelik Derecelerinin Belirlenmesi

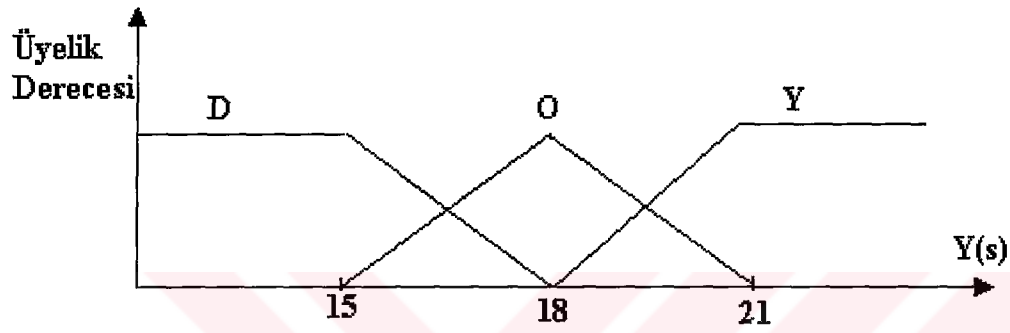
$$\mu_Y(x) = 0 \quad x < a2 \quad (4.18.)$$

$$\mu_Y(x) = \frac{(x - a2)}{(a3 - a2)} \quad a2 \leq x \leq a3 \quad (4.19.)$$

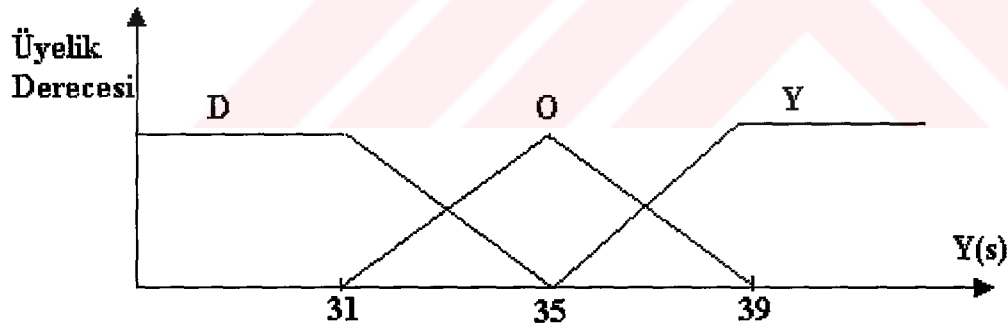
$$\mu_Y(x) = 1 \quad x > a3 \quad (4.20.)$$



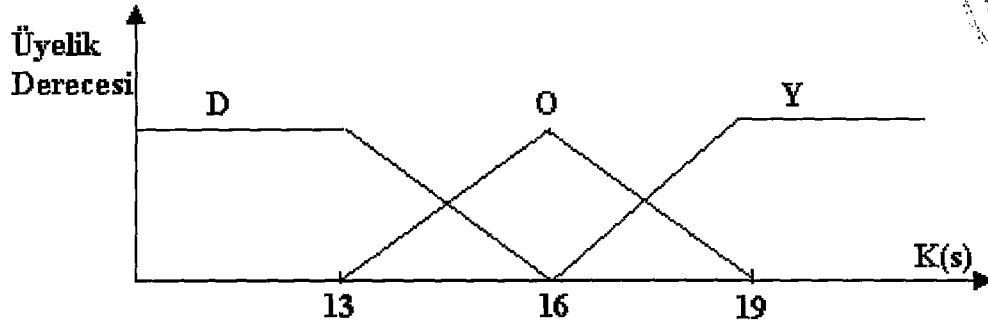
Şekil 4.21. A1 Girişi için Yeşil Işıktaki Geçen Araçların Üyelik Fonksiyonu



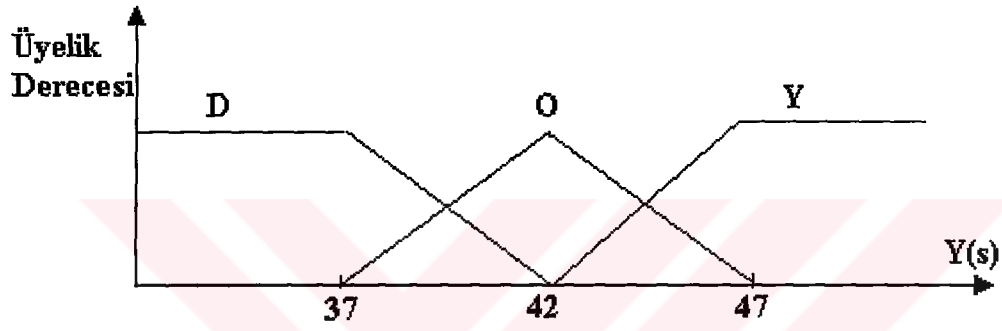
Şekil 4.22. A1 Girişi için Kırmızı Işıktaki Bekleyen Araçların Üyelik Fonksiyon



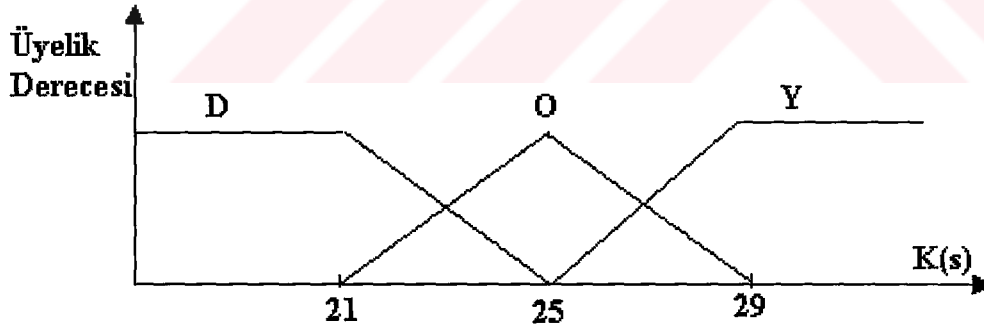
Şekil 4.23. B1 Girişi için Yeşil Işıktaki Geçen Araçların Üyelik Fonksiyon



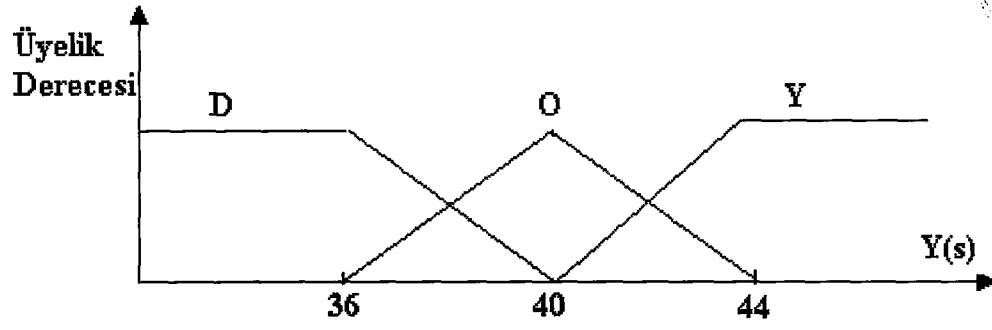
Şekil 4.24. B1 Girişi için Kırmızı Işıktaki Bekleyen Araçların Üyelik Fonksiyon



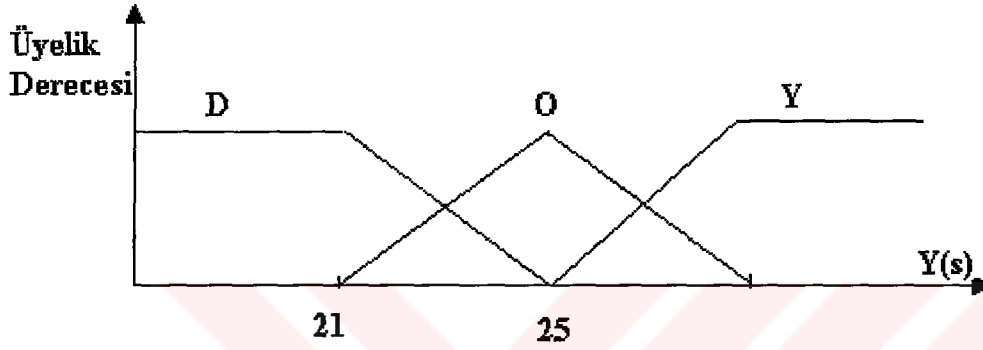
Şekil 4.25. A2 Girişi için Yeşil Işıktaki Geçen Araçların Üyelik Fonksiyon



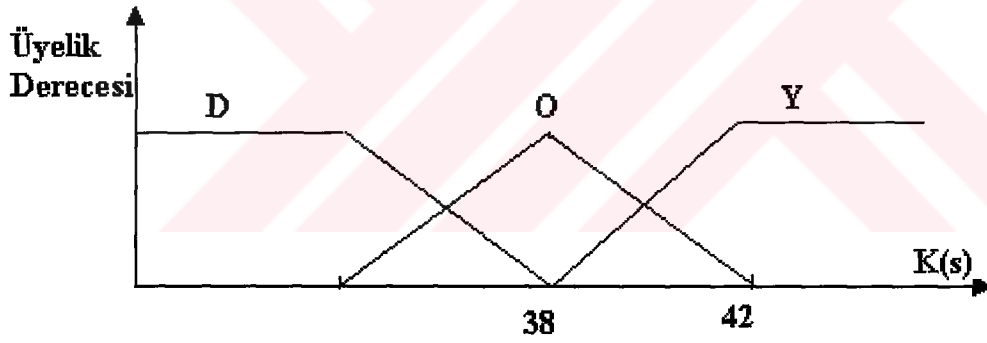
Şekil 4.26. A2 Girişi için Kırmızı Işıktaki Bekleyen Araçların Üyelik Fonksiyon



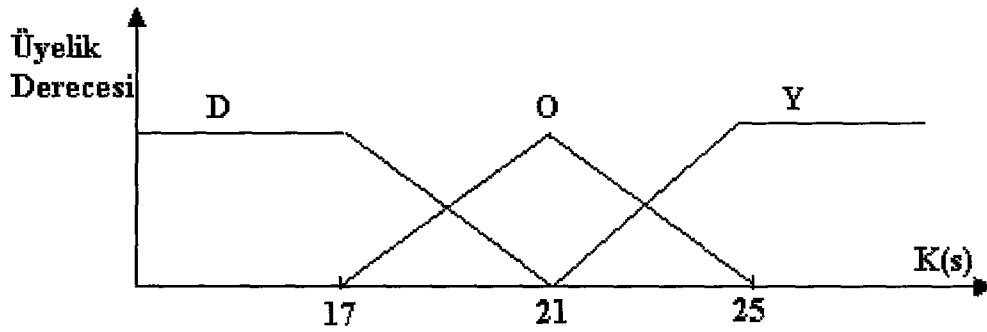
Şekil 4.27. B2 Girişi için Yeşil Işıktaki Geçen Araçların Üyelik Fonksiyonu



Şekil 4.28. B2 Girişi için Kırmızı Işıktaki Bekleyen Araçların Üyelik Fonksiyonu



Şekil 4.29. Yeşil Işıktaki Geçen Araçların Üyelik Fonksiyonu



Şekil 4.30. Kırmızı Işıktaki Bekleyen Araçların Üyelik Fonksiyon

Çizelge 4.16. Bulanık alt küme terimleri için üyelik fonksiyonları (kırmızı için)

	17	18	19	20	21	22	23	24	25
D	1	0.75	0.5	0.25	0	0	0	0	0
O	0	0.25	0.5	0.75	1	0.75	0.5	0.25	0
Y	0	0	0	0	0	0.25	0.5	0.75	1

Çizelge 4.17. Bulanık alt küme terimleri için üyelik fonksiyonları (Yeşil için)

	34	35	36	37	38	39	40	41	42
D	1	0.75	0.5	0.25	0	0	0	0	0
O	0	0.25	0.5	0.75	1	0.75	0.5	0.25	0
Y	0	0	0	0	0	0.25	0.5	0.75	1

Çıkış üyelik fonksiyonu Şekil 4.17’ da ki gibi tanımlı ise bulanık Mantıkta eşdeğer önermeleri şu şekilde olabilir.

“EĞER $K(s) = O$ VE $Y(s) = D$ ise O HALDE $K(t) = Y$ ”

Diğer önermelerde sıralanarak aşağıdaki Çizelge 4.18 oluşturulur.

Çizelge 4.18. Kırmızı Işıқта Bekleyen Araç Sayısı ile ilgili kural Çizelgesu

$K(s)/Y(s)$	Y	O	D
Y	D	D	O
O	D	O	Y
D	O	Y	Y

Yukarıdaki önerme Çizelgesundan aşağıdaki önermeleri üretebiliriz. Bu önermelerin tanımlanması ile kırmızı ve yeşil sürelerin belirlenmesi sağlanır. Kırmızı ışık yanma süresini bulmak için;

Kural 1.

“EĞER Kırmızıda bekleyen araç sayısı Yüksek VE Yeşilde geçen araç sayısı Yüksek İSE Kırmızı yanma süresini Düşür.”



Kural 2.

“EĞER Kırmızıda bekleyen araç sayısı Yüksek VE Yeşilde geçen araç sayısı Orta İSE Kırmızı yanma süresini Düşür.”

Kural 3.

“EĞER Kırmızıda bekleyen araç sayısı Yüksek VE Yeşilde geçen araç sayısı Düşük İSE Kırmızı yanma süresini Orta yap.”

Kural 4.

“EĞER Kırmızıda bekleyen araç sayısı Orta VE Yeşilde geçen araç sayısı Yüksek İSE Kırmızı yanma süresini Düşür.”

Kural 5.

“EĞER Kırmızıda bekleyen araç sayısı Orta VE Yeşilde geçen araç sayısı Orta İSE Kırmızı yanma süresini Orta yap.”

Kural 6.

“EĞER Kırmızıda bekleyen araç sayısı Orta VE Yeşilde geçen araç sayısı Düşük İSE Kırmızı yanma süresini Yükselt.”

Kural 7.

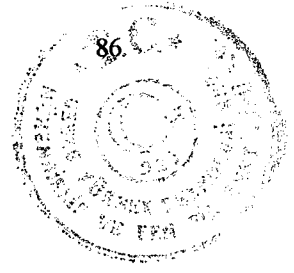
“EĞER Kırmızıda bekleyen araç sayısı Düşük VE Yeşilde geçen araç sayısı Yüksek İSE Kırmızı yanma süresini Orta yap.”

Kural 8.

“EĞER Kırmızıda bekleyen araç sayısı Düşük VE Yeşilde geçen araç sayısı Orta İSE Kırmızı yanma süresini Yükselt.”

Kural 9.

“EĞER Kırmızıda bekleyen araç sayısı Düşük VE Yeşilde geçen araç sayısı Düşük İSE Kırmızı yanma süresini Yükselt.”



Aynı şekilde Yeşil ışık yanma süresini bulmak için kurallar üretilirse;

Kural 1.

“EĞER Yeşilde geçen araç sayısı Yüksek VE Kırmızıda bekleyen araç sayısı Yüksek İSE Yeşil ışık yanma süresini Yükselt.”

Kural 2.

“EĞER Yeşilde geçen araç sayısı Yüksek VE Kırmızıda bekleyen araç sayısı Orta İSE Yeşil ışık yanma süresini Yükselt.”

Kural 3.

“EĞER Yeşilde geçen araç sayısı Yüksek VE Kırmızıda bekleyen araç sayısı Düşük İSE Yeşil ışık yanma süresini Yükselt.”

Kural 4.

“EĞER Yeşilde geçen araç sayısı Orta VE Kırmızıda bekleyen araç sayısı Yüksek İSE Yeşil ışık yanma süresini Orta yap.”

Kural 5.

“EĞER Yeşilde geçen araç sayısı Orta VE Kırmızıda bekleyen araç sayısı Orta İSE Yeşil ışık yanma süresini Orta yap.”

Kural 6.

“EĞER Yeşilde geçen araç sayısı Orta VE Kırmızıda bekleyen araç sayısı Düşük İSE Yeşil ışık yanma süresini Orta yap.”

Kural 7.

“EĞER Yeşilde geçen araç sayısı Düşük VE Kırmızıda bekleyen araç sayısı Yüksek İSE Yeşil ışık yanma süresini Düşür.”

Kural 8.

“EĞER Yeşilde geçen araç sayısı Düşük VE Kırmızıda bekleyen araç sayısı Orta İSE Yeşil ışık yanma süresini Düşür.”

Kural 9.

“EĞER Yeşilde geçen araç sayısı Düşük VE Kırmızıda bekleyen araç sayısı Düşük İSE Yeşil ışık yanma süresini Düşür.”

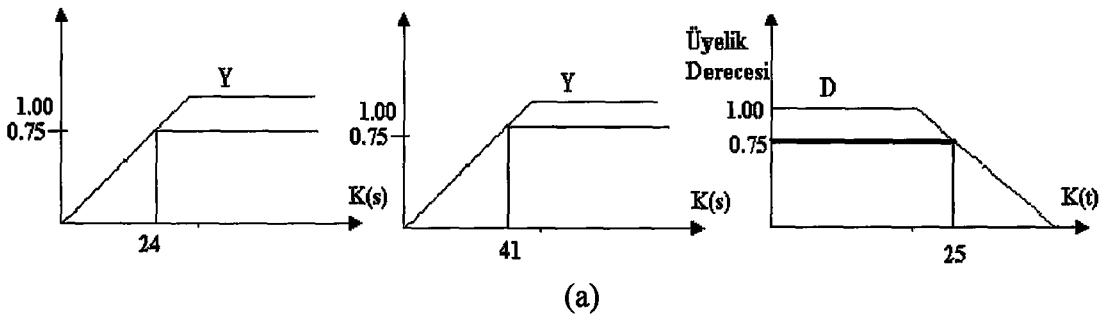
4.3.4. Kırmızı ve Yeşil Işık Süreleri için Üyelik Fonksiyonlarının Oluşturulması

Çıkış üyelik fonksiyonlarının taralı kısımlarının üst üste yerleştirilmiş durumlarının ağırlık merkezinin $K(t)$ çıkış eksenindeki izdüşümü, kırmızı ve yeşil ışık denetim sürelerini verecektir. Ancak;

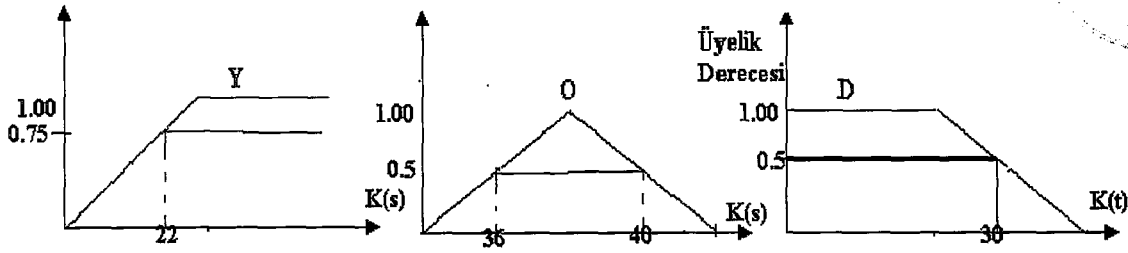
- Küme sayısı ve üyelik fonksiyonları sistemin performansına etki eden en önemli iki kuraldır.
- Üyelik fonksiyonları birbirine benzer veya simetrik olmak zorunda değildir. Ancak üçgen yada yamuk şeklindeki fonksiyonlar teknik olarak daha kolay gerçekleştirilebilirler.
- Önermede “VE” operatörü kullanılırsa en küçük doğruluğa sahip durum, “VEYA” operatörü kullanılırsa en büyük doğruluğa sahip durum çıkış oluşturmak için kullanılır.

1. Kırmızı Yanma Süresi için Üyelik Fonksiyonlarının Oluşturulması

EĞER $K(s) = Y$ VE $Y(s) = Y$ O HALDE $K(t) = D$

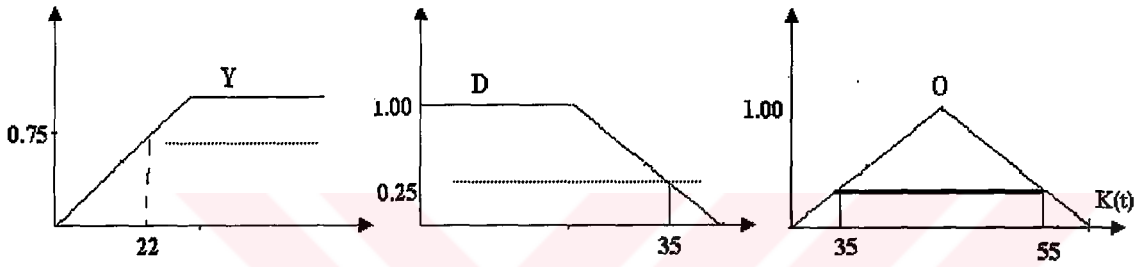


EĞER $K(s) = Y$ VE $Y(s) = O$ O HALDE $K(t) = D$



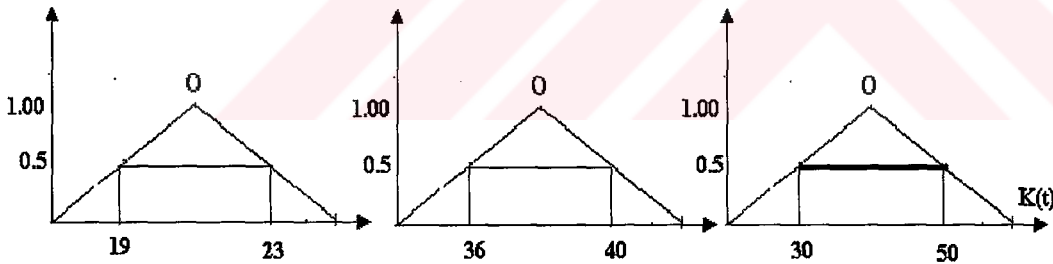
(b)

EĞER $K(s) = Y$ VE $Y(s) = D$ O HALDE $K(t) = O$



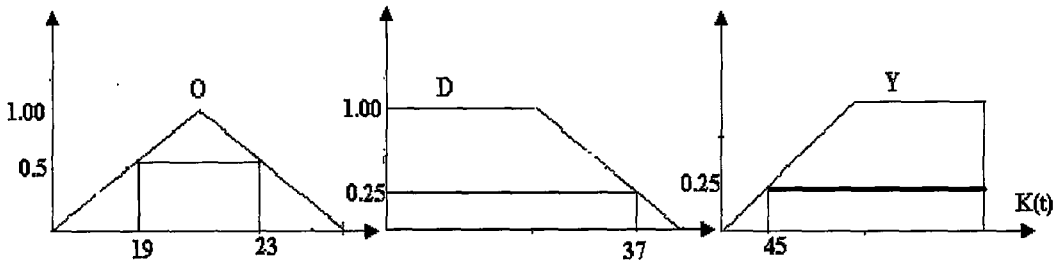
(c)

EĞER $K(s) = O$ VE $Y(s) = O$ O HALDE $K(t) = D$



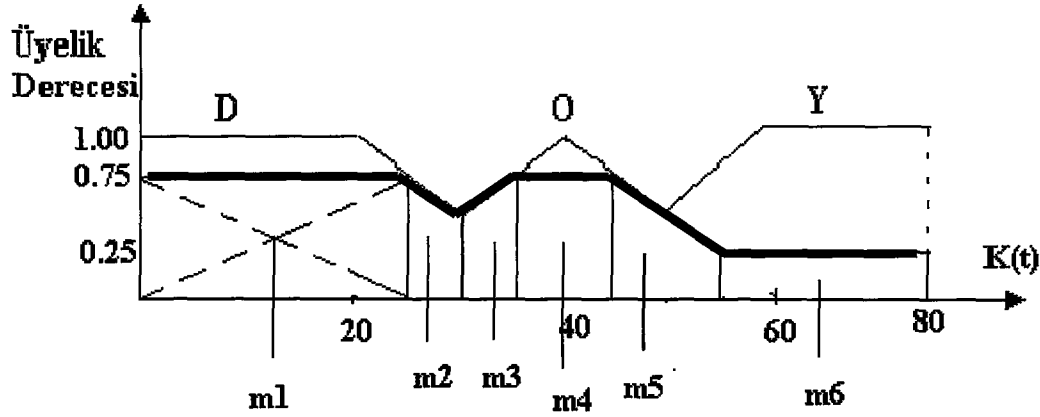
(d)

EĞER $K(s) = O$ VE $Y(s) = D$ O HALDE $K(t) = D$



(e)

Şekil 4.31. Kırmızı ıřıkta bekleyenler iin kurallara dayanarak oluřturulmuř yelik fonksiyonları



Şekil 4.32. Kırımızı ışıktaki bekleyen araçlar için üyelik fonksiyonlarının ağırlık merkezi

Şekil 4.32' de gösterilen alanın ağırlık merkezi gerçek zamanda kırmızıda bekleyen araçların, bekleme zamanını verecektir.

Alan gerekli parçalara ayrılarak, ağırlık hesaplanır

$$m1 = 0.75 * 25 = 18.75$$

$$m2 = (0.5 + 0.75) * 5 / 2 = 3.125$$

$$m3 = (0.5 + 0.75) * 5 / 2 = 3.125$$

$$m4 = 0.75 * 10 = 7.5$$

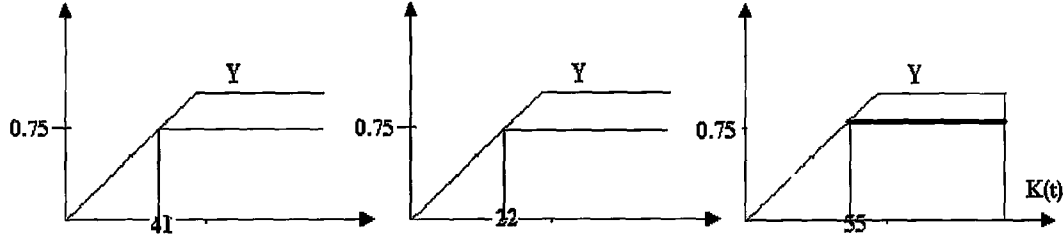
$$m5 = (0.25 + 0.75) * 10 / 2 = 5$$

$$m6 = 0.25 * 25 = 6.25$$

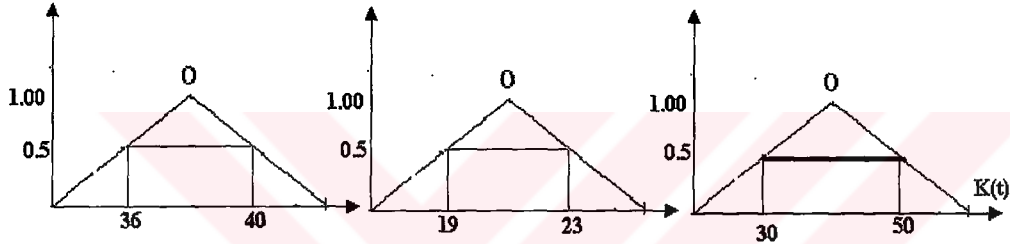
$$X_K = \frac{m1 * x1 + m2 * x2 + \dots + m6 * x6}{m1 + m2 + \dots + m6} = \frac{1388}{44} \approx 32 \text{ saniye olarak bulunur.}$$

2. Yeşil Yanma Süresi için Üyelik Fonksiyonlarının Oluşturulması

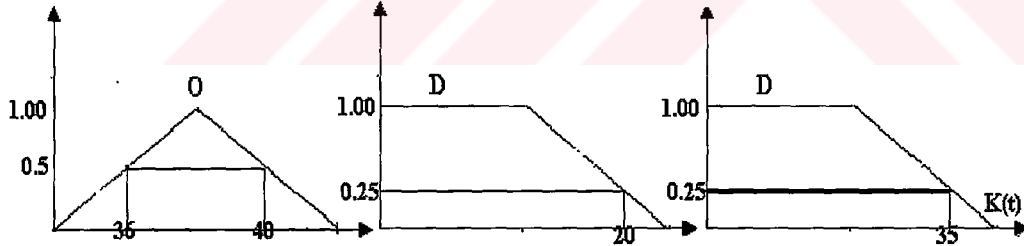
a) EĞER $Y(s)=Y$ VE $K(s)=Y$ O_HALDE $Y(t)=Y$



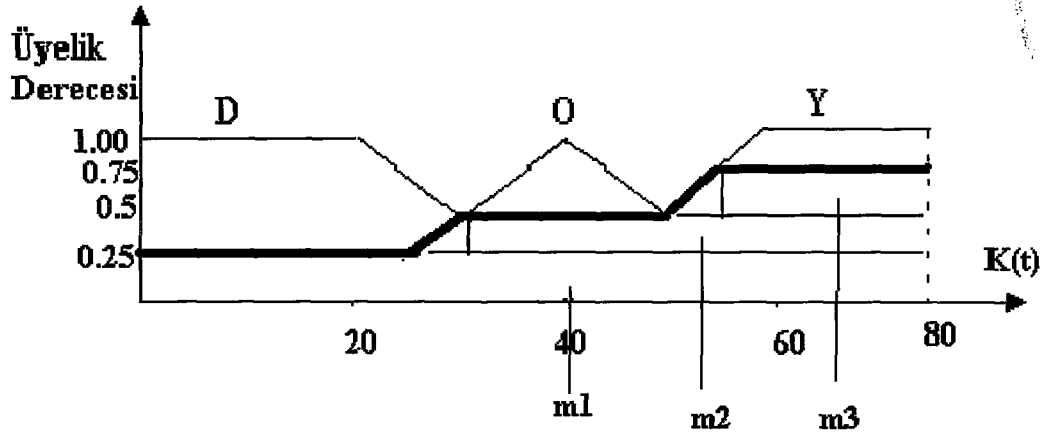
b) EĞER $Y(s)=O$ VE $K(s)=O$ O_HALDE $Y(t)=O$



c) EĞER $Y(s)=O$ VE $K(s)=D$ O_HALDE $Y(t)=D$



Şekil 4.33. a. b. c. Yeşil ışık için kurallara dayanarak oluşturulmuş üyelik fonksiyonları



Şekil 4.34. Yeşil ışık için üyelik fonksiyonlarının ağırlık merkezi

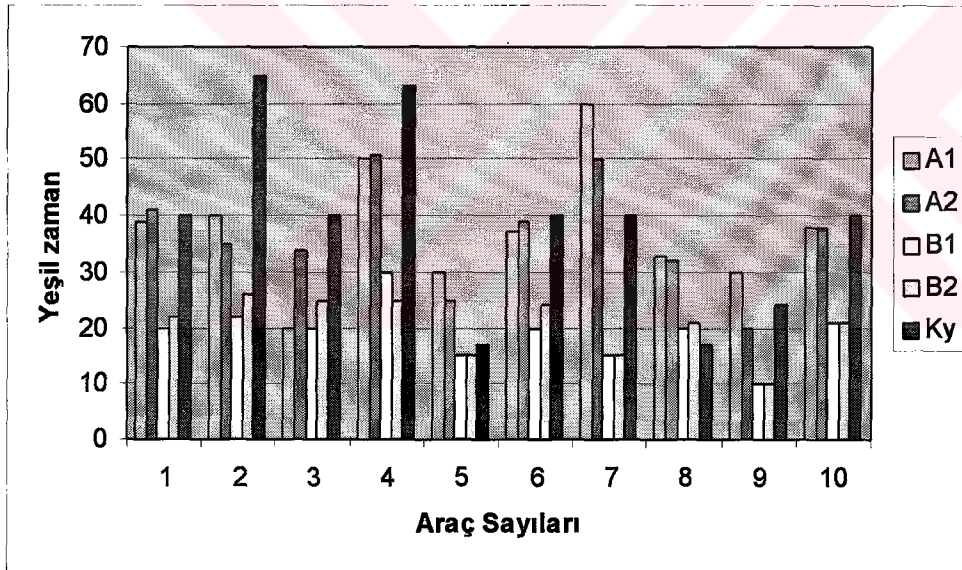
$$\begin{aligned} m1 &= 0.25 \cdot 80 = 20 \\ m2 &= 0.25 \cdot 50 = 12.5 \\ m3 &= 0.25 \cdot 25 = 6.25 \end{aligned}$$

$$X_Y = \frac{m1 \cdot x1 + m2 \cdot x2 + m3 \cdot x3}{m1 + m2 + m3} = \frac{1880}{38.75} \cong 48 \text{ saniye olarak yeşil ışık süresi bulunur.}$$

Bulanık mantık kuralları uygulanarak denetimi yapılan iki fazlı kavşak için A1-A2 ışıkları için yeşilde geçme zamanı çeşitli araç değerlerinde yeşilde geçme süreleri tespit edilmiştir. Sonuçlar Çizelge 4.19 ve Şekil 4.35' de gösterilmiştir.

Çizelge 4.19. Araç sayıları ve geçme süreleri

A1	A2	B1	B2	Ky
39	41	20	22	40
40	35	22	26	65
20	34	20	25	40
50	51	30	25	63
30	25	15	15	17
37	39	20	24	40
60	50	15	15	40
33	32	20	21	17
30	20	10	10	24
38	38	21	21	40



Şekil 4.35. Araç sayısı ve geçme süreleri

Kavşaklardaki trafik denetimi için önerilen yöntem iki ana küme halinde toplanabilir. Bunlardan ilkinin oluşturanlar sinyal döngü sürelerini sabit tutup yeşil ışık sürelerini optimize etmeyi amaçlayan, normal ve doymuş akışların bulunduğu varsayan modeldir. İkinci kümeyi oluşturan modellerde yollardaki normal ve doymuş akışların bulunduğu varsayılır ve en iyi yeşil süreler hesaplanır. Bu süreler hesaplanırken tüm fazlar dikkate alınır. Bulanık mantık algoritmaları kullanılarak hesaplanan yeşil yanma süresi ve kırmızıda bekleme süreleri kullanılarak sabah 08:00 ile 09:00 arasında çalıştırılacak periyodu sabit bir sistem için PLC kullanılabilir. Bu kavşağın LSS programı ile yapılan kavşak PLC kavşak denetim programı EK- A' da gösterilmiştir. Kavşak periyodu :

Periyot Hesabı (C)

LA= 5 m.
LR= 12 m.

Bu verilerle emniyetli geçiş süresini hesaplırsak;

$L_R < 28$ metre için

$$t_s = \frac{L_R + 5,00}{7,00} - \frac{L_A}{13,86} = 3 \text{ s.}$$

Kavşaktaki tüm fazların bir periyodu;

$$C = G_1 + G_2 + \Delta t$$

$$C = 48 \text{ s} + 32 \text{ s} + (2 \cdot 2 + 3 + 3) \text{ s} = 90 \text{ saniye olarak bulunur.}$$

5. SONUÇLAR

Bu çalışmayla şekil 4.5-6' da gösterilen kavşakta trafik ışıklarının süresi günün her saatinde sabit olup iki fazlı trafik akışında çalışmaktadır. Işık süreleri fazlar için 63 s. Yeşil yanma süresi ve 43 s. Kırmızı yanma süresidir. Kavşakta bulanık mantık algoritmaları kullanılarak aşağıdaki faydalar elde edilmiştir.

1. Bu çalışmada bulanık mantığın gerektirdiği metotlardan biri olan ağırlık ortalama yönteminin kullanılması ile elde edilen sonuçlar; kırmızı ışığın yanma süresi 32 s. yeşil ışığın yanma süresi 48 s. olarak bulunmasıyla önceki çalışma sürelerine göre daha iyi sonuçlar elde edilmiştir.

2. Trafik akışında kırmızıda bekleme süresinin daha az olması yakıt ve Zaman bakımından tasarruf sağlamıştır. Bir aracın saatte boşta çalışırken harcadığı süper benzin bedeli 180.000 TL dir. 1 saniyede kayıp enerji maliyeti 50 TL dir. Bu kavşakta yapılan optimizasyon sonucu bir periyot sonunda bir araçtan kazanılan miktar 2200 TL. dir. Bu kavşağın bir günlük kırmızıda bekleyen araç ortalaması (24:00 – 06:00 saatleri dahil edilmemiştir) 13 araçtır. Bir gün boyunca elde edilecek kar ise; **20.592.000 TL.** dir. Bir yılda elde edilecek kazanç ise **7.516.080.000 TL.** dir. Bu aynı zamanda; **27941 USD** eder.

3. Bu çalışmada yayaların ihmal edilmesi, özellikle yayaların takibinin zor olmasından kaynaklanmaktadır.

4. Zaman üyelik fonksiyonu daha dar zaman aralığına yerleştirilerek periyot süresini kısaltıp kırmızı ve yeşil zamanı daha kısa tutulabilirdi, ancak bu çalışmada periyot süresi genel normda kalınarak kırmızı ve yeşil sürelerin en esnek durumları gözönünde tutulmuştur. Periyodu azaltıp, periyot sayısını daha sık yaparak yayaları daha fazla düşünen bir sistem oluşturulabilir. Daha hızlı trafik akışı sağlanabilir. Araç beklemeleri daha azaltılabilir.

5. Trafik ışıklarının parlakları zaman içerisinde etkisini yitirmektedir ve aynı zamanda daha fazla enerji tüketmektedir. Bu lambaların yerine LED' lerden oluşan lambaların kullanılmasıyla daha az enerji tüketimi sağlanacak ayrıca lambaların ömürleri uzatılacaktır. Mevcut sistemde kullanılan lambalar 150 Watt' lık

halojen lambalardır. Kazaların önlenmesi için kırmızı ve yeşil ışık lambaları senede iki defa daha az yanan sarı lamba ise bir defa değiştirilmesi gerekmektedir. Bu lambalar büyük şehirlerde çok büyük miktarda elektrik enerjisi tüketimine sebep olmaktadır. Bu lambalar LED tabanlı trafik ışıkları ile değiştirilirse; bir 1 tane trafik ışığının değiştirilmesi sonucunda yılda **1.4 KW** enerji tasarrufu sağlanacaktır.

6. Sürekli aynı periyot çalışan kavşak lambalarının kontrolü için PLC önerilmiştir. Böylece daha kolay bir şekilde periyot süresi değiştirilebilir. Ayrıca; bütün kumanda fonksiyonları yazılımla gerçekleştirildiğinden, farklı bir uygulama için adaptasyonu kolaydır. Bilgisayar ve diğer kontrolörlerle haberleşme imkanının olması avantajdır.



6. TARTIŞMA VE ÖNERİLER

Bu çalışma; büyük şehirlerimizde araç kullananların ruh sağlığını önemli bir şekilde etkileyen trafik ışık optimizasyonu üzerinde durulmuştur. Gebze’ de trafik yoğunluğu fazla olmayan bu kavşakta yapılan çalışma gelecekte bu konuyla ilgilenenlere bir yöntem göstermektedir. Sistemin simülasyonu yapılmamıştır, iyi bir simülasyon programının pahalı,her zaman kolaylıkla bulunamayacak düzeyde yaratıcılık gerektirmesi ve zaman alıcı bir işlem olması, Simülasyon sonuçları genellikle sayısaldir bu ise trafik denetiminde gerçek hayatta fazla önemli değildir, Trafik denetimine kalıcı bir çözümün bulunması açısından;

1. Yayaların dikkate alınmaması, trafik denetiminin onlarıda dikkate alacak şekilde düzenlenmesi,
2. Sistemin önce bölgelere daha sonrada tüm şehri dikkate alacak şekilde ele alınması,
3. Kavşakları birbirine bağlayan yolların kuyruklanması önleyecek kuralların oluşturulması,
4. Şehirlerde trafikten sorumlu, gerekli tedbirleri alan kişi ve kurumların yetki ve sorumluluklarını belirten, düzenleyen bir çalışmanın yapılmasında gereksinim vardır.
5. Ayrıca Üniversitelerde ve Yüksek Teknoloji Enstitülerinde trafik kursülerinin kurulması gereklilik haline gelmiştir.



KAYNAKLAR

1. YÜKSEL İ., "Otomotik Kontrol Sistem Dinamiği ve Denetim Sistemleri", Uludağ Üniversitesi, 1997.
2. GÜMÜŞTAŞ R., "Trafik Akış Sistemlerinin Bulanık Mantıkla Denetimi", Sakarya Üniversitesi, Yüksek Lisans Tezi, 1995.
3. KAYNAK, O., "Programlanabilir Denetleyiciler", Boğaziçi Üniversitesi, Mühendislik Fakültesi, Aralık 1988.
4. ROBERT, L. K. and Keith R. B., "Simulation of Traffic Flow and Control Using Fuzzy and Conventional Methods", Fuzzy Logic and Control, ed.: JAMSHIDI, M., VADIEE, N. and J. ROSS, T., University of New Mexico, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1993.
5. KURTULAN S., "Programlanabilir Lojik Kontrolörler", İ.T.Ü. Kontrol ve Bilgisayar Mühendisliği Bölümü, Bileşim Yayıncılık, 1996.
6. ADALI, E., "Mikroişlemciler Mikrobilgisayarlar", İ.T.Ü. Kontrol ve Bilgisayar Mühendisliği Bölümü, Sistem Yayıncılık, 1993.
7. OTTER, J.D., "Programlanabilir Mantık Denetleyiciler", Milli Eğitim Bakanlığı, 1994.
8. OMRON, "SYSMAC Programmable Controllers C200HS Operation Manual", 1995.
9. JAMSHIDI, M., "Fuzzy Logic", Fuzzy Logic and Control, JAMSHIDI, M., VADIEE, and J. ROSS, T., University of New Mexico, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1993.
10. ÖZDAŞ N., DİNİBÜTÜN T. ve KUZUCU A., "Otomotik Kontrol Temelleri" İ.T.Ü. Makine Fakültesi, Birsen Yayıncılık, 1995.
11. KAYNAK, O., "Bulanık Denetim ve Endüstriyel Uygulamaları", TÜBİTAK, Marmara Araştırma Merkezi, CAD/CAM ve Robotik Bölümü, 28 Ocak 1993.
12. SHOURESHİ, R. "Intelligent Control Systems: Are They for Real", Transactions of the ASME, Vol.115, pp.392-401, June-1993.
13. SONG DA-HE "A Mathematical Model and Algorithm of Fuzzy Controller", Applications of Fuzzy Set Methodologies in Industrial Engineering. Department of Computer Science and Technology, Northern Jiaotong University, China.

14. VADIEE, N., "Fuzzy Rule-Based Expert Systems I", Fuzzy Logic and Control, ed.: JAMSHIDI, M., VADIEE, N. and J. ROSS, T., University of New Mexico, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1993.
15. RALSTON, P.A.S and WARD, T.L., " Fuzzy Control of Industrial Processes", Applications of Fuzzy Set Methodologies in Industrial Engineering, ed.: EVANS, G.W., KARWOWSKI, G. and WILHELM, M., ELSEVIER Amsterdam Oxford-Newyork-Tokyo, 1989.
16. GERİŞKOVAN, I. " PID, Fuzzy ve PID-Fuzzy Kontrol Algoritmalarına Genel Bir Bakış", Otomasyon, Haziran-1995.
17. JAMSHIDI M " Fuzzy Logic", Fuzzy Logic and Control, JAMSHIDI, M., VADIEE, N. and J. ROSS, T., University of New Mexico, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1993.
18. CHUEN CHIEN, L., " Fuzzy Logic in Control Systems: Fuzzy Logic Controller Part I" IEEE Trans. on Systems, Man and Cybernetics, V. 20, pp. 404-418 1990
19. DOĞAN İ., "İstanbul Büyükşehir Belediyesi Trafik Sinyalizasyon Projesi için Öneriler", Londra, 1997.



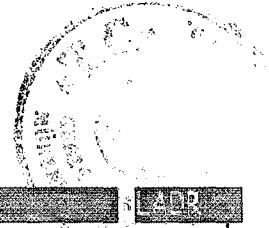
ÖZGEÇMİŞ

1971’ de Sivas’ da doğdu. İlk, orta, lise eğitimini aynı şehir de tamamladı. 1989’ da Gazi üniversitesine girdi bir sene okuduktan sonra 1990 yılında Yıldız Teknik Üniversitesi Elektrik mühendisliği bölümüne girdi. “İstanbul 2000’ daki Elektrik Enerjisi Planlaması” adlı tez çalışmasını yaparak 1994-Haziran’ da mezun oldu. 1994-Ekim’ de Gebze Yüksek Teknoloji Enstitüsü Elektronik Mühendisliği Bölümünde yüksek lisansa başladı. Emlak Bankası Mutlukent Projesinde Borova Yapı Endüstrisinde kontrol mühendisi olarak çalışırken 1995-Kasım ayında Gebze Yüksek Teknoloji Enstitüsü Elektronik Mühendisliği Bölümünde Araştırma Görevlisi olarak atandı. Halen bu görevi sürdürmektedir.



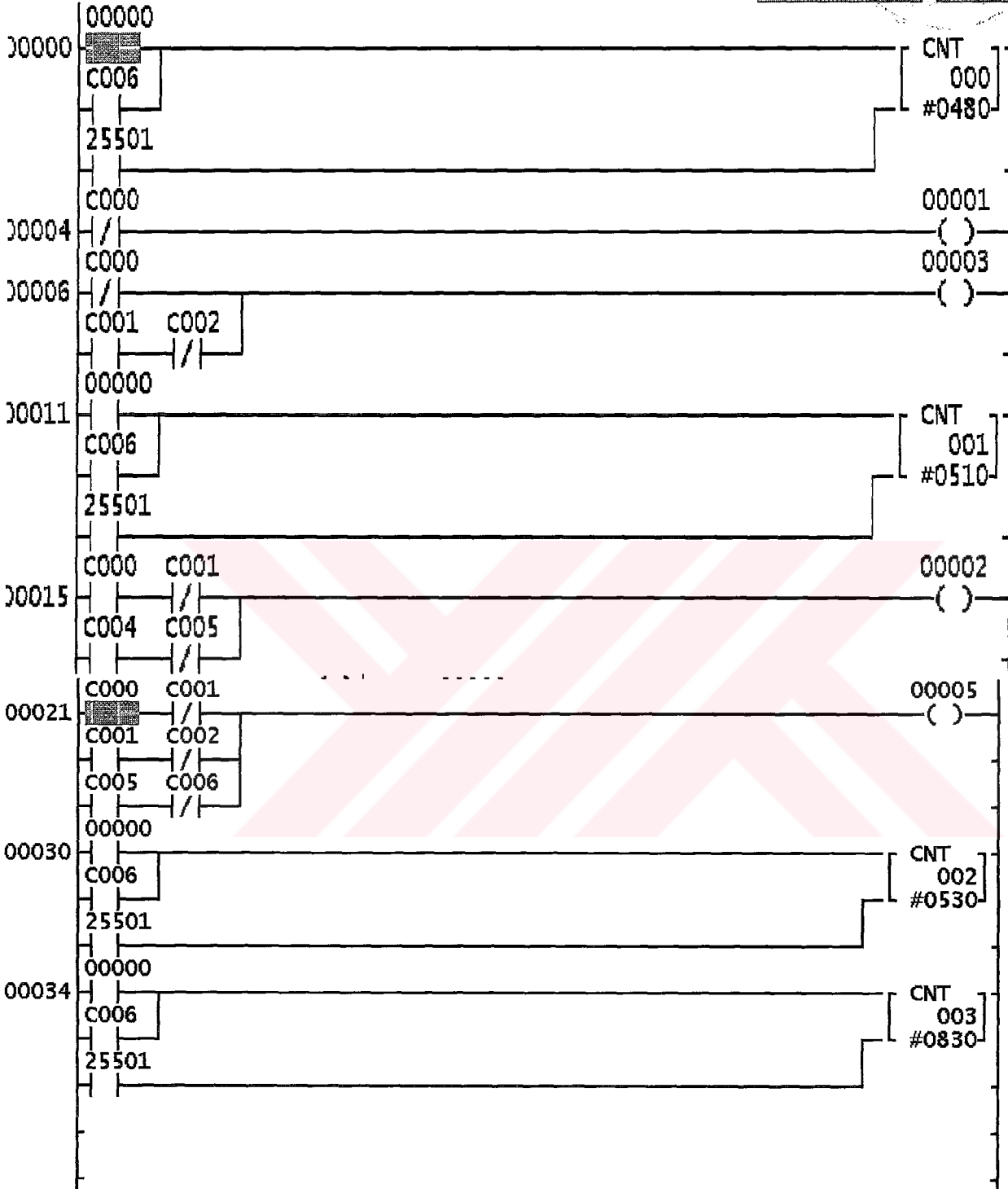
EK-1

LSS KAVŞAK PROGRAMI

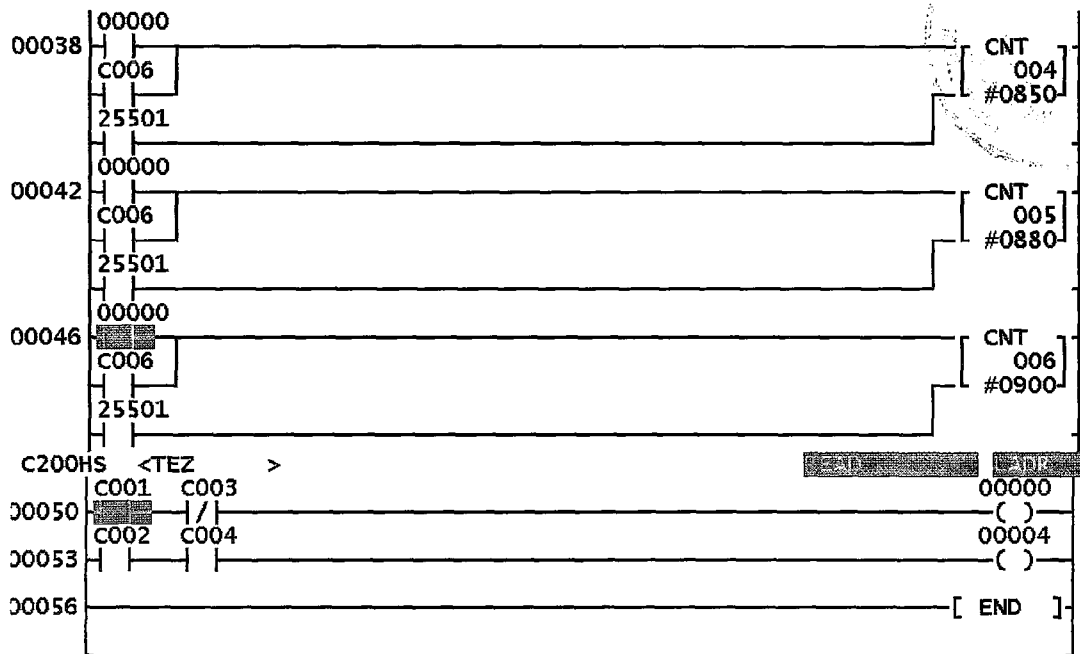


C200HS <TEZ >

READ READ



1 2 3 4 5 6 7 8 9 10



< < <

ADDRESS	MNEMONIC	OPERAND	COMMENT
00000	LD	00000	
00001	OR	CNT 006	
00002	LD	25501	
00003	CNT	000	
		# 0480	
00004	LD NOT	CNT 000	
00005	OUT	00001	
00006	LD NOT	CNT 000	
00007	LD	CNT 001	
00008	AND NOT	CNT 002	
00009	OR LD		
00010	OUT	00003	
00011	LD	00000	
00012	OR	CNT 006	
00013	LD	25501	
00014	CNT	001	
		# 0510	
00015	LD	CNT 000	
00016	AND NOT	CNT 001	
00017	LD	CNT 004	
00018	AND NOT	CNT 005	
00019	OR LD		
00020	OUT	00002	
00021	LD	CNT 000	
00022	AND NOT	CNT 001	
00023	LD	CNT 001	
00024	AND NOT	CNT 002	
00025	OR LD		
00026	LD	CNT 005	
00027	AND NOT	CNT 006	
00028	OR LD		
00029	OUT	00005	
00030	LD	00000	
00031	OR	CNT 006	
00032	LD	25501	
00033	CNT	002	
		# 0530	
00034	LD	00000	
00035	OR	CNT 006	
00036	LD	25501	
00037	CNT	003	
		# 0830	
00038	LD	00000	
00039	OR	CNT 006	
00040	LD	25501	
00041	CNT	004	
		# 0850	
00042	LD	00000	
00043	OR	CNT 006	
00044	LD	25501	

< < <

ADDRESS	MNEMONIC	OPERAND	COMMENT
00045	CNT	005	
		# 0880	
00046	LD	00000	
00047	OR	CNT 006	
00048	LD	25501	
00049	CNT	006	
		# 0900	
00050	LD	CNT 001	
00051	AND NOT	CNT 003	
00052	OUT	00000	
00053	LD	CNT 002	
00054	AND	CNT 004	
00055	OUT	00004	
00056	END		

