

PROJE RAPORU

Proje Adı

Küçük Çocukların Televizyonu Yakından İzleme Problemine Bulanık Mantık Yaklaşımı.

İçindekiler

| | |
|--|----|
| 1. Giriş..... | 2 |
| 1.1 Projenin Amacı..... | 2 |
| 2. Yöntem..... | 3 |
| 2.1 Keskin Mantık ve Bulanık Mantık | 3 |
| 2.2 Bulanık Çıkarım İşlemi (FIS-Fuzzy Inference Process)..... | 4 |
| 2.3 Küçük Çocukların Televizyonu Yakından İzlemesini Engelleyen Uygulamanın Geliştirilmesi | 6 |
| 2.3.1 Sistemin Çalışmasına Uygun Sayısal Girişlerin Atanması..... | 8 |
| 2.3.2 Giriş ve Çıkış Üyelik Fonksiyonları..... | 9 |
| 2.3.3 Kural Tabanı | 11 |
| 2.3.4 Sonuçların Birleştirilmesi | 13 |
| 2.3.5 Durulama İşlemi | 14 |
| 3. Sistemin Uygulanması..... | 15 |
| Sonuçlar..... | 18 |
| Öneriler..... | 18 |
| Kaynakça..... | 19 |
| Ekler | 20 |
| EK1: Çalışmada kullanılan ana fonksiyon | 20 |

1. Giriş

Günümüzde televizyonun olmadığı ev neredeyse yok gibidir. Birçok anne baba çocuklarını yetiştirirken onların televizyonu yakından izlemelerine şahit olmuş ve bunu kendilerince yöntemlerle engellemeye çalışmıştır. Sevdiği çizgi filmi veya reklamları televizyona yapışık gibi izleyen bir çocuk görüntüsünden rahatsız olmayacak anne baba herhalde yoktur.

Çocukların neden televizyonu yakından izledikleri konusunda yapılan bazı incelemelerde bunun neden değil sonuç olduğu ve çocuğun göz bozukluğu kusurlarından birine sahip olabileceği söylenmektedir. Bunun yanı sıra çocuğun dikkat çekmek, uyaranların canlılığı vb. gibi sebeplerden dolayı da yakından izlediği düşünülmektedir(“Televizyonu izleme uzaklığı ne olmalı?”).

Televizyonu yakından izlemenin erken dönemde çocuklara ve özellikle gözlere büyük bir zararının olmadığı tespit edilmiştir. Gözlerin yorulması parlak yüzeye bakılması az da olsa elektromanyetik alana maruz kalma vb. zararlarının olabileceği tespit edilmiştir(“Yakından televizyon izlemenin zararları”,2012)(“ Teknolojinin Çocuklara Zararları Nelerdir”)(“ Televizyonu Yakından İzlemenin Zararları”). En önemli etkisi ise televizyon izleme alışkanlığını kazandırmasıdır ve onun gelişim açısından küçük çocuklar üzerindeki zararları ise sayılmayacak kadar çoktur.

Her şeye rağmen hiçbir anne babanın da çocuklarının televizyonu yakından izlemesine razı olabileceği düşünülemez. Hele ki uzmanların 0-4 yaş arasında televizyonun mümkünse hiç izlenmemesini tavsiye ettiği günümüzde.

Fakat anne babalar sosyal hayatın karmaşası içinde bu konuda yalnız kalmıştır. Televizyonun cazibedarlığı ve sosyal hayatın yoğunluğu onları çaresiz bir durumda bırakmıştır. Çocukların televizyonu yakından izlenmesini yaşamayan anne baba neredeyse yok gibidir.

Burada etkili televizyon izleme mesafesi akla gelmektedir. Yapılan çalışmalar televizyonun ekran boyutunun (köşeden köşeye) 2 katı mesafesinin uygun olabileceği sonucunu vermiştir. Etkili ve sağlıklı bir izleme için bu mesafenin dışında olmak gerekir yaklaştıkça yukarıda anlatılan sorunların ortaya çıkma olasılıkları artmaktadır(“Led TV İzleme Mesafeleri”).

Peki, çocuktaki bu tür bir davranışı giderme adına çözüm önerileri nelerdir? Genelde verilen çözümlerin büyük bir çoğunluğu tavsiyelerden oluşmaktadır(“Televizyonu izleme uzaklığı ne olmalı?”). Yaptığımız araştırmalarda, kullanılan bir cihaz, araç veya televizyonlara ait bir özellik bulamadık.

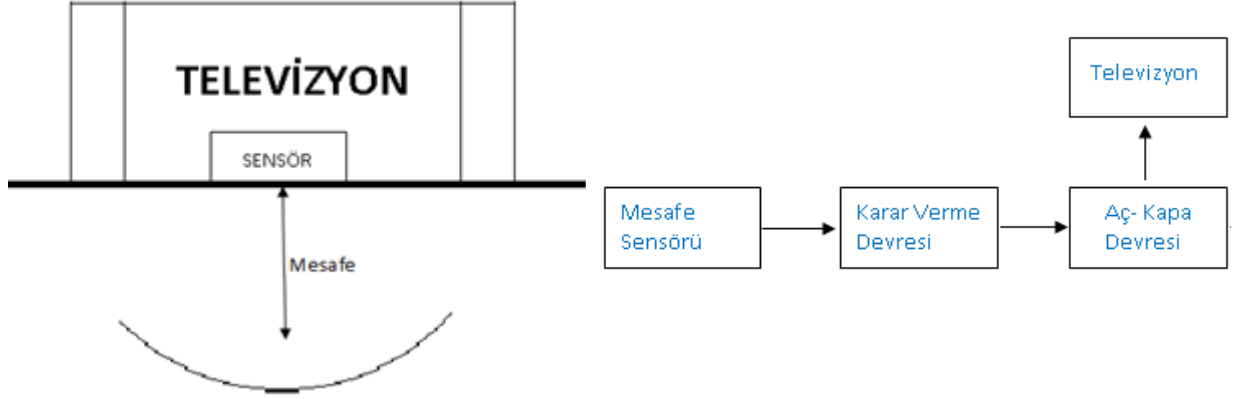
Özellikle küçük kardeşimin de (yaş 9) televizyonu yakından izlemesi ve buna engel olamamızın beni ve arkadaşımı küçük çocukların televizyonu yakından izlemesine engel olacak ve onlarda bu davranışı kazandırabilecek, yakından izleme davranışı varsa düzeltilebilecek bir cihaz geliştirme konusunda proje çalışması yapmaya yönlendirdi.

1.1 Projenin Amacı

Küçük çocukların televizyonu yakından izlemelerini engelleyen ve çocuklarda yakından televizyon izlememe davranışı kazandıran bir sistem geliştirmek.

2. Yöntem

İlk çalışmalarımız da mesafe ölçebilen bir sensör ile mesafe ölçülüp tespit edilen sınırın içine girildiğinde televizyonun kapatılması ve çıkıldığında ise televizyonu açılması şeklinde çalışabilecek bir yapı üzerinde duruldu.



Şekil 1: Sistemin blok şeması.

Fakat bu yaklaşım keskin bir şekilde çalışıyor ve küçük çocuğun televizyon önündeki davranışlarını ya da televizyonun önünde olan herhangi bir kişinin davranışı tam anlamıyla kestiremiyordu. Çalışma şekli kullanım açısından uygunluk göstermedi. Bu denetleme yapısının eksik kaldığını gördük.

Bu noktada televizyonun bulunduğu yerdeki koşullara uyum sağlayabilecek ve televizyon önündeki davranışları analiz ederek yakından izleme olayını tespit edip belirlenen kurallar çerçevesinde çocuklarda davranış geliştirmeye yardımcı olacak bir yöntem üzerinde durulmaya başlandı. Geliştirilecek cihazın çözüm algoritmasının insani düşünceye ve hayata uyumlu olması gerekiyordu. Bulanık mantık yaklaşımlarının buna uygun olduğunu gördük (Elmas, 2007) ve çalışmamızı bulanık mantık ile çözüm geliştirme üzerinde yoğunlaştırdık.

2.1 Keskin Mantık ve Bulanık Mantık

Geleneksel mantık, olayları 0 veya 1 (var veya yok, doğru veya yanlış) şeklinde ele alır. Bir olay için yalnızca iki durum söz konusudur. Bulanık mantık ise olayları 0 ve 1 arasındaki gerçek değerleriyle ifade eder ("Fuzzy Logic").

1965 yılında Lotfi A. Zadeh tarafından ilk defa ortaya konmuştur. Pratiksel olarak matematiksel modeli çıkartılamayan problemlerin çözümünde etkin olarak kullanıla gelmiştir (Greg, 1993). Bir kontrol sisteminin çözümünde genel olarak tablo yöntemi ve matematiksel model çıkarma yöntemi kullanılır. Tablo yöntemi bellekte fazla yer kaplar, matematiksel yöntem ise çok karışıktır ve her problemin çözümünde kullanılamayabilir. Bulanık mantık ile herhangi bir matematiksel modele ihtiyaç duymadan yüksek doğrulukta problemler çözülebilir("Fuzzy Logic").

Klasik mantıkta sınıflandırmalar keskindir. Yani bir eleman bir kümenin ya elemanıdır ya değildir. Kısmi üyelik olamaz. Bulanık mantık insan mantığını taklit ederek belirsiz ve yaklaşık

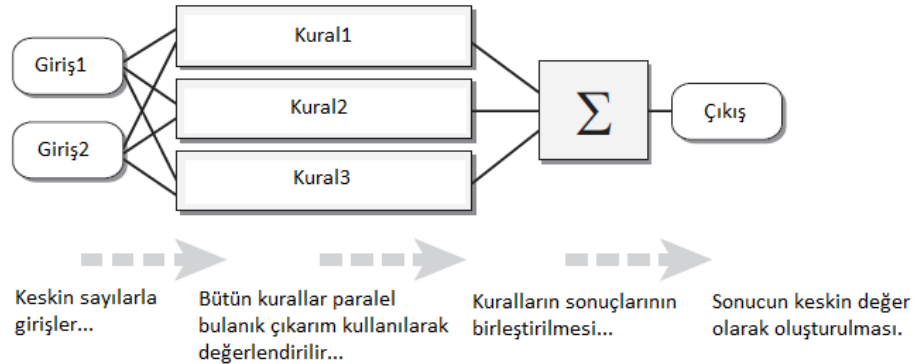
durumlarda işlem yapabilme yeteneğine sahiptir. Bulanık mantıkta bir eleman birden fazla kümenin elemanı olabilir (Aysun ve Kazan).

Bir çocuğun televizyon önündeki hareketlerinin analizi matematiksel olarak ifade edilemez. Dolayısıyla keskin mantık kurallarıyla yapılan çözüm sınırlı kalacaktır. Bu problemde bulanık mantık kullanılması daha iyi sonuçlar verecektir.

2.2 Bulanık Çıkarım İşlemi (FIS-Fuzzy Inference Process)

Bulanık çıkarım işlemi bulanık mantık kullanılarak mevcut keskin girişten sistemin çalışmasına uygun çıkarım yapma işlemidir. En çok kullanılan iki tip çıkarım sistemi Mamdani ve Sugeno ‘dur. Mamdani çıkarım yöntemi esnek, çok geniş kullanım alanı olan ve insani girişlere daha yatkındır. Sugeno ise hesaplama verimliliği yüksek, doğrusal tekniklerle daha iyi çalışabilir ve matematiksel analizler için daha iyidir. Bu iki yöntem arasında Mamdani daha çok kullanılmaktadır. Bunlardan başka yöntemlerde mevcuttur (Elmas, 2007)(“ Fuzzy Logic Tool Box User’s Guide”).

Şekil 2’de iki girişli, tek çıkışlı ve 3 kurala sahip bir sistemin diyagramı vardır. Girişten çıkışa doğru akışa ait bilgiler verilmiştir. Kuralların paralel yapısı bulanık mantık sistemlerin çok önemli bir yönüdür (“ Fuzzy Logic Tool Box User’s Guide”).



Şekil 2: İki giriş, üç kurallı bulanık mantık çıkarım sistemi blok şeması.

Mamdani bulanık çıkarım sistemi 5 bölümden oluşmaktadır.

- Giriş değişkenlerinin bulanıklaştırılması (Fuzzification).
- Öncül önermelerin (kuralların eğer (if) kısımları) bulanık operatörler (AND veya OR) ile bağlanması (Antecedent).
- Gerektirme işlemi (Implication) yeni kuralların eğer (if) ve ise (then) kısımlarının birleştirilmesi.
- Kural sonuçlarının birleştirilmesi (Aggregation).
- Durulama işlemi (Defuzzification).

Bulanık çıkarım işlemine geçmeden önce bulanık kümeler ve kuralların oluşturulması gerekir. Tanımlanan problemle ilgili gerekli analizler yapılarak bulanık kümeler oluşturulmalı ve

sözel ifadelerle bu kümeler belirlenmelidir. Bulanık kümeler üyelik fonksiyonları ile ifade edilirler ve bu fonksiyonlar üçgen, sigmoid, gaussian vb. gibi eğrilerdir. Probleme uygun olan fonksiyon seçilerek belirlenen sözel ifadelere atanırlar (“Fuzzy Logic Tool Box User’s Guide”).

Üyelik fonksiyonları uygun değer aralıkları verilerek oluşturulurlar.

Problemle ilgili kurallar, elde edilen tecrübeler ve bulgulardan faydalanılarak eğer(if)-ise(then) şeklinde sözel ifadeler şeklinde tanımlanırlar. Bu işlemin sonunda bir kural tabanı oluşturulmuş olur. Kurallar oluşturulurken bütün durumların değerlendirilmiş olması gerekir.

Örnek kural tanımlama ifadesi şu şekilde olabilir;

*Eğer (uzaklık YAKIN) **ve** (süre UZUN) ise (televizyonu KAPAT)*

Burada “uzaklık” evrensel kümesinde bulunan “YAKIN” üyelik fonksiyonu ile “süre” evrensel kümesinde bulunan “UZUN” üyelik fonksiyonu kuralın birinci kısmını(eğer kısmı) ifade eder. Kuralın ikinci kısmında (ise kısmı) ise sonuç, “televizyon” evrensel kümesindeki “KAPAT” üyelik fonksiyonu ile ifade edilmiştir. İfadeden de görüleceği gibi kuralın birinci kısmındaki ifadeleri bağlamak için mantıksal operatör “and-ve” kullanılmıştır. Her bir mantıksal operatörün bulanık karşılığı vardır. Örneğin “ve” operatörünün karşılığı “min“, “veya” operatörünün ise “max“ işlemidir. “min” işlemi verilen iki giriş değerinden küçük olanı sonuç olarak verirken, “max” işlemi büyük olanı sonuç olarak verir. Her iki işlemde eğer girişler birbirine eşitse değerin birini sonuç olarak verirler(“Fuzzy Logic Tool Box User’s Guide”).

1. Bulanıklaştırma (Fuzzification)

Sözel ifadelerle tanımlanarak oluşturulan her bir üyelik fonksiyonu yardımıyla sistem girişlerinin üyelik fonksiyonlarına olan üyelik dereceleri hesaplanır. Bu hesaplamaların sonucunda 0-1 arası değerlerden oluşan üyelik fonksiyonları kadar değer kaydedilir (Elmas, 2007)(“Fuzzy Logic Tool Box User’s Guide”). Bu aşamada oluşturulan tüm değerler artık bulanık sayılardır.

2. Bulanık Operatör Uygulama (Antecedent)

Oluşturulan her bir kuralın birinci kısmındaki (if-eğer) ifadelerin bulanıklaştırılmış değerlerine bulanık operatörler (min, max vb.) uygulanır. Bu işlem sonucunda kural sayısı kadar 0-1 arası değer elde edilerek kaydedilir (Elmas, 2007)(“Fuzzy Logic Tool Box User’s Guide”).

3. Gerektirme İşlemi

Kuralların birinci kısmındaki (if-eğer) ifadelerde olduğu gibi ikinci kısmındaki (than-ise) sonuç ifadelerinde bir üyelik fonksiyonu içerirler. Gerektirme işlemine bulanık operatör uygulanarak elde edilmiş bulanık sayı ile sonuç üyelik fonksiyonuna uygulanır. Gerektirme işlemi bulanık operatör uygulanmış kuralın sonucuyla ilgili fonksiyonunu yeniden şekillendirilmesidir. Her bir kural için uygulanmalıdır. İki tane yöntem kullanılır min (minumum) ve prod() olmak üzere (Elmas, 2007)(“Fuzzy Logic Tool Box User’s Guide”).

Gerektirme işlemiyle sonuçların hangi oranda geçerli olduğu tespit edilmiş olur (Yılmaz).

4. Kural Sonuçlarının Birleştirilmesi (Aggregation)

Kuralın geçerlilik dereceleri ve sonuçları hesaba katılarak tüm sisteme ait nihai bir sonuç elde etme süreci kuralların birleştirilmesi olarak adlandırılır. Birleştirme yöntemi olarak maksimum (max) , toplama (sum) veya aritmetik toplama (probor) tercih edilebilir. Amaç tüm sonuçları temsil eden nihai bir sonuç alanı bulmaktır. Bu son şekle sonuç bulanık kümesi adı verilir (Yılmaz).

5. Durulaştırma (Defuzzification)

Bulanık sonucun makinelerin anlayabileceği sayısal bir değere dönüştürme işlemine durulaştırma denir (Yılmaz). En çok tercih edilen durulaştırma yöntemi ağırlık merkezi yöntemidir (Centroid). Aşağıdaki matematiksel formülle ifade edilir.

$$Z = \frac{\sum_{j=1}^q Z_j \cdot \mu_c(Z_j)}{\sum_{j=1}^q \mu_c(Z_j)} \quad (1)$$

Oluşturulan sonuç bulanık kümenin meydana getirdiği sonuç eğrinin orta noktasını bulmayı sağlar ("Fuzzy Logic Tool Box User's Guide"). Bu işlemin sonucu tek bir sayısal değerdir ve bulanık çıkarım işleminin çıkış değeridir.

2.3 Küçük Çocukların Televizyonu Yakından İzlemesini Engelleyen Uygulamanın Geliştirilmesi

Çalışmaya başlamadan önce uzaklık sensörü için hangi sensörleri kullanabiliriz bunun araştırmasını yaptık. Maksimum mesafe olarak 300 cm belirlediğimiz için bu mesafede en az 1-2 cm çözünürlükte ölçüm yapabilecek sensörlerin infrared ya da ultrasonik sensörler olabileceğini tespit ettik. Mesafe açısından sharp GP2Y0A02YK infrared sensörü ile HC-SR04 ultrasonik mesafe sensörlerini inceledik. Buna göre sensörleri incelediğimizde tablo 1'deki gibi değerleri gördük.

| | Sharp GP2Y0A02YK | HC-SR04 |
|----------------------------|-----------------------------|----------------|
| Mesafe(cm) | 20-150 | 2-400 |
| Çıkış | Analog | PWM gibi |
| Çalışma Gerilimi(V) | 5 | 5 |
| Hassasiyet | ? | 3mm |
| Görüş Açısı | Çok Küçük | 15 derece |

Tablo 1: İncelenen sensörlere ait değerler.

HC-RS04 kullanım kolaylığı fiyat ve diğer özellikleriyle dikkatimizi çekti. Görüş açısının 15° olması bir adet sensörün yetmeyeceği ve en az iki tane sensör kullanmamız gerektiğine karar verdik ("Ultrasonic ranging module").

Bulanık çıkarım işlemi için ise en çok kullanılan ve insani karar verme yeteneğine daha çok yakın olan Mamdani çıkarım yöntemini tercih ettik.

Mamdani çıkarım yönteminin algoritması şu şekildedir (Yılmaz);

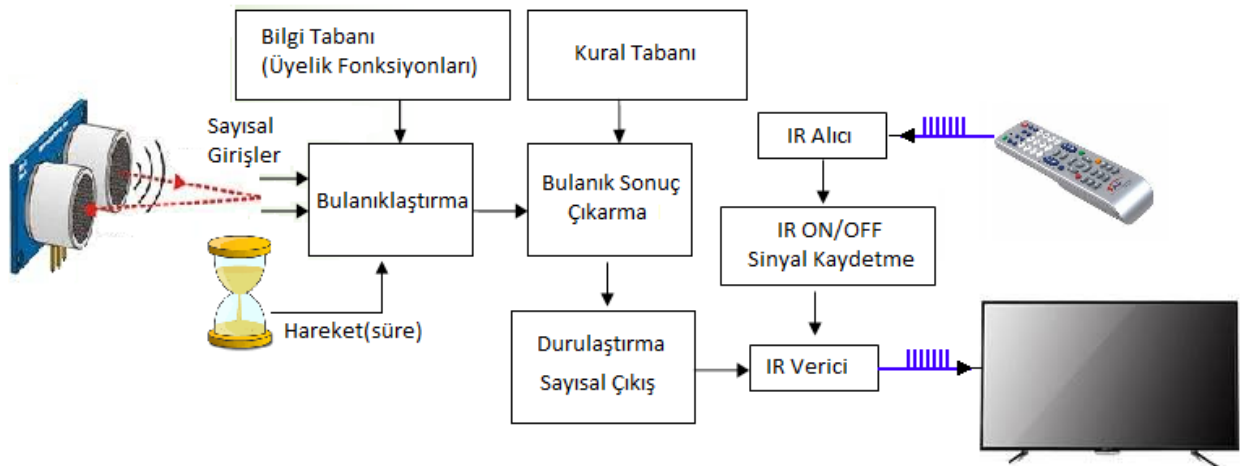
1. Sayısal Girişlerin Atanması, Giriş ve Çıkışlara Ait Sınırların Belirlenmesi.
2. Giriş ve Çıkış Üyelik Fonksiyonları.
 - a. Giriş üyelik fonksiyonlarının tanımlanması ve girişlerin bulanıklaştırılması.
 - b. Çıkış üyelik fonksiyonlarının tanımlanması.
3. Kural Tabanı
 - a. Öncül önermelerin (kuralın eğer kısmı) bulanık operatörler ile bağlanarak kuralların geçerlilik (üyelik) derecelerinin belirlenmesi.
 - b. Gerektirme operatörü ilk kurallara ait sonuç bulanık kümelerin hesaplanması.
4. Sonuçların Birleştirilmesi.
5. Durulama İşlemi.

Televizyon önündeki bir çocuğun hareketlerinin analiz edilebilmesi için mesafeyle beraber o mesafede kalınan sürenin dikkate alınması uygun olacaktır. Yani televizyonun izlendiğini anlamak için hareket-hareketsizlik miktarının incelenmesi gerekecektir. *Bu durumda sistemimiz, aslında televizyon için belirlenen sınır uzaklık miktarında hareket-hareketsizlik analizi yaparak sonuca ulaşacaktır.*

Televizyonun açma kapama yöntemi olarak ta kumandadan açma kapama butonuna basıldığında televizyonun infrared alıcısına giden bilgi önceden yakalanıp kaydedilerek sistem tarafından bulanık çıkarım sonucuna uygun olarak tekrar gönderilmesinin kullanım açısından en uygun seçenek olacağı sonucuna varılmıştır. Buna göre sistemimizin genel blok şeması şekil 3'teki gibidir.

Buna göre donanımda kullanılacak malzeme listesi şu şekildedir;

- Arduino Uno ya da Mega.
- IR alıcı-verici (1'er adet)
- HC-RS04 ultrasonik sensör (2 adet).
- Led (2 adet).
- Direnç, güç kaynağı vb.



Şekil 3: Sistemin genel blok şeması.

Öncelikle Mamdani bulanık çıkarım sistemine uygun olarak bulanık mantıkla ilgili işlemler yapılması gerekmektedir. Yapılan işlemler bundan sonraki bölümde anlatılacaktır.

2.3.1 Sistemin Çalışmasına Uygun Sayısal Girişlerin Atanması

Bu durum için basit bir deney düzeneği hazırlanarak tek sensörle ölçme mesafesi ve televizyon için belirlenen sınır değerine göre ara mesafeler tayin edilmiştir. Bu mesafe değerlerinde çeşitli ve rastgele hareket ile ilgili durumlar incelenmiş ve hareketle ilgili süre değerleri belirlenmiştir. Yapılan çalışma ile elde edilen başlangıç değerleri tablo 2’de gösterilmiştir. Bu değerler sistem tasarımı bittikten sonra ev ortamında sistem denendiğinde elde edilen sonuçlara göre tekrar düzenlenecektir.

| GİRİŞLER | | | |
|-----------|------------|---------------|-------|
| Sensörler | | Zaman(Saniye) | |
| S1-S2 | Mesafe(cm) | Z | Süre |
| ÇokYakın | 0-50 | ÇokHareketli | 0-5 |
| Yakın | 49-100 | Hareketli | 3-8 |
| Sınır | 99-150 | Sınır | 7-12 |
| Uzak | 149-200 | AzHareketli | 11-17 |
| ÇokUzak | 199-300 | Hareketsiz | 15-30 |

Tablo 2: Sayısal giriş değerlerinin belirlenmesi.

Sistemimizde bulanık çıkarım işleminde işlem sonucu tek çıkış iken ayrıca uyarı oluşturmak amacıyla da bir çıkış kullanılmıştır. Uyarı çıkışı bulanık çıkarım işlemine katılmamıştır. Çünkü yaptığımız analizde uyarı işlemi çocuğun sınır uzaklıktan(150 cm) içeri girdiği andan itibaren verilmeye başlayacaktır. Bu durumda keskin mantık kurallarına daha uygundur. Bundan dolayı uyarı çıkışı keskin mantıkla çalışır.

Televizyonun açık veya kapalı olma durumu ise bulanık çıkarım işleminde elde edilen çıkışa göre oluşturulacaktır. Bu durum için belirlenen üyelik fonksiyonlarına ait sayısal değerler tablo 3’de gösterilmiştir.

| ÇIKIŞLAR | | | |
|----------|---|-------|---|
| TV | | Uyarı | |
| açık | 0 | var | 1 |
| kapalı | 1 | yok | 0 |

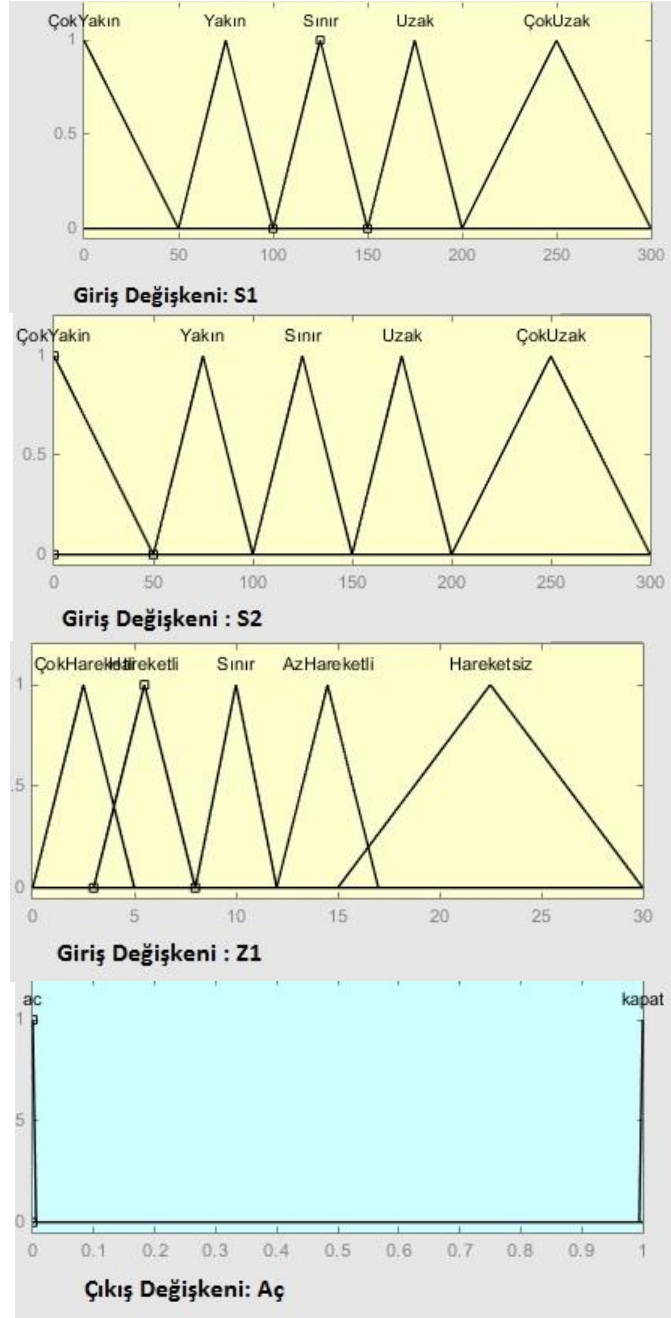
| | |
|-----------|--|
| * açık: | TV kapalı ise aç değilse açık kalsın |
| * kapalı: | TV açık ise kapat değilse kapalı kalsın. |

Tablo 3: Bulanık çıkarım işleminin çıkışına ait sayısal değerler

Tabloda televizyonun açık olması sıfır (0) ile ifade edilmiştir. Kapalı olma durumu ise bir (1) ile ifade edilmiştir. Bulanık çıkarım işleminin sonucu eğer sıfır (açık) ise bu durumda sistemde televizyon kapalı ise televizyonu açar, açık ise hiçbir şey yapmaz. Eğer çıkarım işleminin sonucu bir (kapalı) ise televizyon açıksa sistem televizyonu kapatır, kapalı ise hiçbir şey yapmaz.

2.3.2 Giriş ve Çıkış Üyelik Fonksiyonları

Hesaplama kolaylığı açısından üçgen üyelik fonksiyonları tercih edilmiştir. Belirlenen sayısal değerlere göre girişlere ve çıkışlara ait üyelik fonksiyonları aşağıdaki gibidir.



Şekil 4: Girişlere ve Çıkışa ait üyelik fonksiyon şekilleri.

Çıkış üyelik fonksiyonu da temelde bir üçgendir. Fakat üçgenin tabanına ait bütün değerler ($a=b=c$) birbirine eşittir.

Üçgen üyelik fonksiyonu için gelen herhangi bir girişe ait üyelik değeri hesaplaması için aşağıdaki gibi algoritma takip edilmiştir; (a, b, c üçgeninin tabanına ait değerler, b yükseklik olmak üzere)

1. Üçgenin a, b, c değerleri ile giriş değerini al.
2. $a=b=c$ ise üyelik değerini 1 hesapla.
3. $a=b$ ise ve giriş üçgen içerisinde ise üyelik değerini $(c - \text{Giriş}) / (c - b)$ olarak hesapla.
4. $b=c$ ise ve giriş üçgen içerisinde ise üyelik değerini $(\text{Giriş} - a) / (b - a)$ olarak hesapla.
5. Giriş a, b, c değerlerinin içinde ise üyelik değerini 3 ve 4 adımdan biriyle hesapla.

Yazılan fonksiyon aşağıdaki gibidir;

```
double ucgenUyelikFonk(double x, double a, double b, double c){

    double t1 = (x - a) / (b - a);
    double t2 = (c - x) / (c - b);

    if ((a == b) && (b == c)) return (double)(a==x);
    if (a == b){
        if ((a <= x) && (x <= c)) return (double)(t2);
    }
    if (b == c){
        if ((a <= x) && (x <= b)) return (double)(t1);
    }
    t1 = min(t1, t2);
    return (double)max(t1, 0);
}
```

Bu algoritmaya uygun yazılan “ucgenUyelikFonk” fonksiyonu yardımıyla da giriş değeri bulanıklaştırılmıştır. Bunun işlem için gerçekleştirilen adımlar ise şu şekildedir;

1. Her bir giriş için bulanık değerleri tutacak dizileri tanımla.
2. Her bir girişte tanımlanan üyelik fonksiyonu kadar girişleri bulanıklaştırarak ilgili diziye aktar.

Yazılan fonksiyon aşağıdaki gibidir;

```
void girisleriBulaniklastir(double g1, double g2){
    int x = 0;

    for (x = 0; x < 5; x++){
        bulanikS1[x] = ucgenUyelikFonk(g1, mfi_S1_2_ua[x], mfi_S1_2_ub[x], mfi_S1_2_uc[x]);
    }
    for (x = 0; x < 5; x++){
        bulanikZ1[x] = ucgenUyelikFonk(g2, mfi_Z1_ua[x], mfi_Z1_ub[x], mfi_Z1_uc[x]);
    }
}
```

Bu işlem sonucunda girişlerin üyelik fonksiyonu sayısı kadar elemana sahip giriş sayısı kadar dizi oluşturulmuş oldu. Bulanık değerleri tutan bu diziler sonraki işlemlerde kullanılacaktır.

2.3.3 Kural Tabanı

Sistemimizin kurallarının eğer kısmında, mesafe değerleri “veya” mantıksal bağlacı ile bağlanırken, işlemin sonucu hareket durumunu ifade eden süre değişkeni ile “ve” bağlacı ile bağlanmaktadır. Buna göre çıkışımıza ait mantıksal ifade şu şekildedir;

(S1,2 :Mesafe sensörü, Z1 : Zaman)

Çıkış = ((S1) veya (S2)) ve (Z1)

= (S1+S2).(Z1)

Bu ifadede Z1 paranteze mantıksal olarak dağıtılsa;

Çıkış = (Z1.S1)+(Z1.S2) şeklinde olur.

Dolayısıyla sistemimizde giriş ve her birinde 5 tane üyelik fonksiyonunda olduğu düşünüldüğünde her bir durumda değerlendirileceği için toplamda 125 kural çıkmaktadır. Bu durum ise yazılımda dizi kullanacağımız için RAM kullanım miktarının artması anlamına gelmektedir.

Bu durumu çözmek için ise, yukarıdaki mantıksal çıkış ifadesi önce bulanık operatör karşılıklarıyla tekrar yazılmıştır. Buna göre ifade;

Çıkış = max (min (Z1, S1), min (Z1, S2)) şeklinde olmuştur.

Burada sistemi iki girişli hale getirmek için, S1 ve S2 ‘nin üyelik fonksiyonuna ait belirlenen değerlerde aynı olduğundan dolayı önce min (Z1, S1) sonrasında ise min (Z1, S2) işlemi gerçekleştirilmiş çıkan sonuç ise max yöntemiyle birleştirilerek çıkış elde edilmiştir. Böylelikle kurallarımızın sayısı 25 olmaktadır. (Bknz Ek 1)

Bundan sonra kurallar tek tek incelenmiş ve tablo 4’teki gibi yazılmıştır. Sonrasında ise bu kurallar 5x5 matris oluşturacak şekilde tekrar tablo 5’te de gösterilmiştir.

| Üyelik Fonk. Sözel İfadeleri ile Kurallar | | | | |
|---|--|--------------|-------------|-------|
| Kural No | if (eğer) | | then(ise) | |
| | ((Sensörler-S1_2) ve (Süre-Z1)) | | | |
| | s1 | z1 | TV | uyarı |
| 1 | ÇokYakın | ÇokHareketli | açık | var |
| 2 | ÇokYakın | Hareketli | açık | var |
| 3 | ÇokYakın | Sınır | açık | var |
| 4 | ÇokYakın | AzHareketli | kapalı | var |
| 5 | ÇokYakın | Hareketsiz | kapalı | var |
| 6 | Yakın | ÇokHareketli | açık | var |
| 7 | Yakın | Hareketli | açık | var |
| 8 | Yakın | Sınır | açık | var |

| | | | | |
|----|---------|--------------|--------|-----|
| 9 | Yakın | AzHareketli | kapalı | var |
| 10 | Yakın | Hareketsiz | kapalı | var |
| 11 | Sınır | ÇokHareketli | açık | var |
| 12 | Sınır | Hareketli | açık | var |
| 13 | Sınır | Sınır | açık | var |
| 14 | Sınır | AzHareketli | kapalı | var |
| 15 | Sınır | Hareketsiz | kapalı | var |
| 16 | Uzak | ÇokHareketli | açık | yok |
| 17 | Uzak | Hareketli | açık | yok |
| 18 | Uzak | Sınır | açık | yok |
| 19 | Uzak | AzHareketli | açık | yok |
| 20 | Uzak | Hareketsiz | açık | yok |
| 21 | ÇokUzak | ÇokHareketli | açık | yok |
| 22 | ÇokUzak | Hareketli | açık | yok |
| 23 | ÇokUzak | Sınır | açık | yok |
| 24 | ÇokUzak | AzHareketli | açık | yok |
| 25 | ÇokUzak | Hareketsiz | açık | yok |

Tablo 4: Sistemin çalışması ile ilgili kurallar.

| TV | VE(AND) | Hareket(Z1) | | | | |
|--------------|----------|--------------|-----------|-------|-------------|------------|
| | | ÇokHareketli | Hareketli | Sınır | AzHareketli | Hareketsiz |
| Mesafe(S1_2) | ÇokYakın | açık | açık | açık | kapalı | kapalı |
| | Yakın | açık | açık | açık | kapalı | kapalı |
| | Sınır | açık | açık | açık | kapalı | kapalı |
| | Uzak | açık | açık | açık | açık | açık |
| | ÇokUzak | açık | açık | açık | açık | açık |

Tablo 5:Kuralların matris gösterimi

a. Öncül Önermelere Bulanık Operatör Uygulaması

Bu aşamada ise kuralların eğer kısımlarının bulanık operatör min ile bağlanması gerekmektedir. Kullanılan algoritma şu şekildedir.

1. Kural sayısı kadar elemana sahip bir dizi oluştur.
2. Kuralın eğer kısmındaki girişlerin üyelik fonksiyonlarına ait bulanık değerlere min operatörü uygula.
3. 2. adımı her bir kural için sırayla uygula ve sonucu diziye aktar.

Bu işlem sonucunda dizimizde her bir kuralın geçerlilik derecelerini içeren kural sayısı kadar bulanık değer yer almaktadır.

İşlemle ilgili yazılan fonksiyon aşağıdaki gibidir.

```

void bulanikOperatorUygula(double blg1[], double blg2[]){
    int i = 0;
    for (int x = 0; x < 5; x++){
        for (int y = 0; y < 5; y++){
            dizi_blOperator[i++] = min(blg1[x], blg2[y]);
            //Kuralların if kısımlarına bulanık operator and-->min
            //uygulandı ve diziye alındı...
        }
    }
}

```

b. Gerektirme İşleminin Uygulanması

Bu aşamada ise bulanık operatör uygulanmış dizide yer alan her bir kurala ait bulanık değer ile kuralın sonuç kısmındaki (ise bölümü) üyelik fonksiyonuna “min” bulanık operatörü uygulanır. Kullanılan algoritma şu şekildedir.

1. Kural sayısı kadar elemana sahip bir dizi oluştur.
2. Çıkış üyelik fonksiyonuna ait olan keskin değeri al.
3. Keskin değerle ilgili çıkış üyelik fonksiyonuna göre bulanıklaştır.
4. Hesaplanan bulanık değer ile kuralın ilk kısmındaki değere “min” operatörü uygula ve sonucu dizide sakla.
5. Her çıkış keskin değeri için işlemi bütün kurallara uygula.

Böylelikle keskin giriş için çıkış üyelik fonksiyonları yeniden şekillendirilip birleştirme işlemi için hazırlanmıştır.

Geliştirilen fonksiyon aşağıdaki gibidir;

```

void gerektirmeUygula(double xEkGiris)    //implication işlemi
{
    int indeks = 0;
    double bulanikCikis = 0.0;
    //Butun kuralları her bir çıkış fonk. x ekseni girişi için tara..
    for (int x = 0; x < 25; x++){
        indeks = kurallar[x];
        bulanikCikis = ucgenUyelikFonk(xEkGiris, mfo_Ac_ua[indeks],
                                         mfo_Ac_ub[indeks], mfo_Ac_uc[indeks]);
        dizi_imp[x] = min(bulanikCikis, dizi_blOperator[x]);
    }
}

```

2.3.4 Sonuçların Birleştirilmesi

Herhangi bir çıkış değeri için girişlerin her bir kurala uygun olarak onun üzerindeki etkisi belirlendikten sonra bunların birleştirilmesi gerekmektedir. Bu işlem için kullanılan algoritma aşağıdaki gibidir.

1. Gerektirme işlem sonucunun ilk elemanını al.

2. İkinci elemanı ile “max” operatörü uygula.
3. Sonucu bir sonraki elemanla “max” operatörü uygula.
4. Bu işlemi gerektirme işlemini tutan dizinin elemanları sayısınca yap.
5. Sonucu bul.

İşlem sonucu tek bir bulanık değerdir. Bu değer girişlerin mevcut değerine göre keskin çıkışın ne oranda etkileneceğini gösterir. Durulama işlemi için giriş değeridir.

Geliştirilen fonksiyon aşağıdaki gibidir;

```
double sonuclariBirlestir() //aggregation işlemi
{
    double sonuc;

    double dizi_1[25];

    for (int x = 0; x < 25; x++){
        dizi_1[x] = dizi_imp[x];
    }
    sonuc = dizi_1[0];
    for (int z = 1; z < 25; z++){
        sonuc = max(sonuc, dizi_1[z]);
    }
    return sonuc;
}
```

2.3.5 Durulama İşlemi

Durulama işleminde belli bir çözünürlükte her bir çıkış değeri için gerektirme ve birleştirme işlemleri yapılarak. Elde edilen sonuçlar değerlendirilir. Ağırlık Merkezi (centroid) yöntemi kullanılmıştır. Bu yöntem birleştirme işlemi sonucu oluşan çıkış üyelik fonksiyonlarının yeni değerlerinin orta noktasını bulur. Formülü aşağıdaki gibidir.

$$Z = \frac{\sum_{j=1}^q Z_j \cdot \mu_c(Z_j)}{\sum_{j=1}^q \mu_c(Z_j)}$$

Z_j : Çıkıştaki keskin değerleri ifade eder.

μ_c(Z_j) : Z_j değerine uygulanmış gerektirme ve birleştirme işleminin sonucudur.

Z : Durulama işleminin sonucudur ve keskin bir değerdir.

Formülde pay kısmı çıkış keskin değerlerindeki minimumdan maksimuma toplam değişimi ifade ederken, payda kısmı yeni oluşmuş değerlerin toplam alanını ifade eder. Buna göre kullanılan algoritma şu şekildedir.

1. Çıkışa ait minimum ve maksimum değerlere göre çözünürlüğü 100 kabul ederek adım değerini hesapla.

2. Sıfırdan çözünürlük değerine kadar.

- Adım değerine göre çıkış keskin değerini hesapla.
- Çıkış değerine gerektirme ve birleştirme işlemi uygula.
- Adım değeri ile birleştirme sonucunu çarparak parça değerini hesapla.
- Parça değerlerinin toplayarak alanı hesapla.
- Parça değeri ile çıkış değerinin çarparak değişim değerinin bul ve topla.

3. Durulama sonucu için toplam değişim değerini toplam alana böl.

İşlem sonucu bulanık çıkarımın sonucudur. Sistem bu sonuca göre ne yapması gerektiğine karar verir.

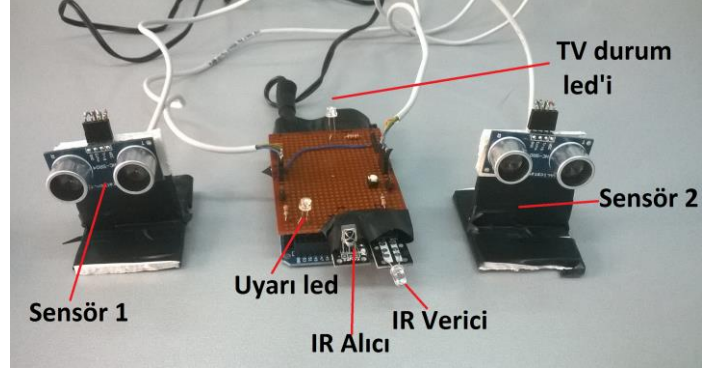
Geliştirilen fonksiyon aşağıdaki gibidir;

```
double durulama() {
    double coz = 101;
    double adim = (1.0 - 0.0) / (coz - 1.0); //
    double x_ekseni = 0.0;
    double degisim = 0.0, alan = 0.0, parca = 0.0;
    double mfCikis = 0.0;

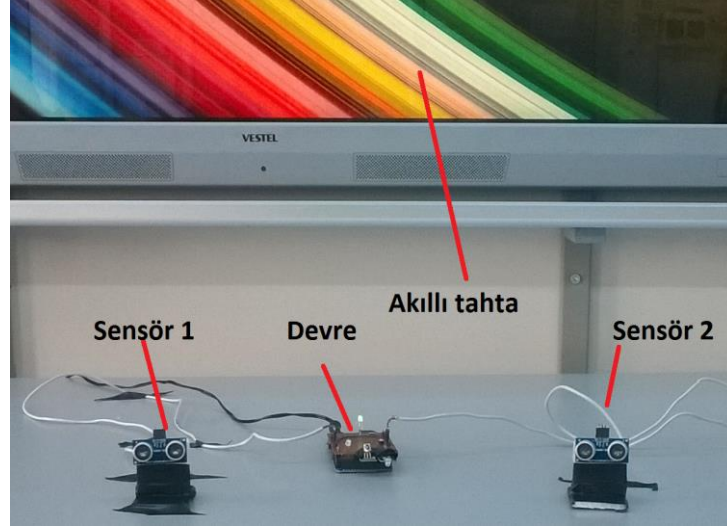
    for (int i = 0; i < coz; i++) {
        x_ekseni = (double) (0 + (adim * i));
        gerektirmeUygula(x_ekseni);
        mfCikis = sonuclariBirlestir();
        parca = adim * mfCikis; //
        degisim += parca * x_ekseni; //pay
        alan += parca; //payda
    }
    return ((alan == 0) ? ((1.0 + 0.0) / 2.0) : (degisim / alan));
}
```

3. Sistemin Uygulaması

Sistemimizin son durumuna ait görüntüsü resim 1'dedir. Sistemi test etmek amacıyla öncelikli olarak atölye sınıfımızda bulunan akıllı tahtaya adapte ettik. Bu durum resim 2'de görülmektedir. Burada rastgele bir biçimde sensörlerin önünde belirsiz hareketler yaparak sistemin çalışmasını denedik. Sistemimizin bulanık mantık çıkarım işleminde kullandığımız kural tabanına uygun olarak çalıştığını gördük. Bu aşamadan sonra ev ortamında da çalışmayı test etmek istedik. Bununla ilgili yapılan ev çalışmaları aşağıdaki gibidir.



Resim 1: Sistemimizin mevcut son durumu.



Resim 2: Akıllı tahta ile yaptığımız çalışma görüntüsü.

1- Kendi evimizde yaptığımız çalışmalar.

Evdeki sistemin konulacağı ortamla ilgili bazı veriler şu şekildedir:

- Evdeki çocuk sayısı: 1
- Çocukların yaşları: 9 (Erkek)
- Sistemin kurulduğu odanın büyüklüğü: 4×6 m
- Televizyonun yeri: televizyon ünitesinde 4 m'lik duvarın ortasında
- TV Ekranı :106 cm
- Test süresi: 1 hafta



Resim 3: Ev ortamında yaptığımız çalışmanın görüntüsü.

Evdeki televizyon kumandasının açma-kapama düğmesi sisteme tanıtıldı. Daha sonrasında sistem 1 hafta süre ile test edildi.

Sistem bağlanmadan önceki durum: Evdeki küçük çocuk ünitenin önüne gelip sevdiği çizgi film ve programları yaklaşık 40 cm’lik mesafeden devamlı izliyordu. Ailesinin uyarılarına kulak asmıyordu. En sonunda baba televizyonu kapatıyor veya mutfaktaki televizyonu izlemesi için gönderiyordu.

Sistem bağlandıktan sonraki durum:

Çocuğun yaşı büyük olduğu için sistem bağlı iken yakından izlemiyordu. Sensörlerin görmediği noktalardan izlemeye çalışıyordu. Sistemin bağlı olmadığı mutfaktaki televizyonu daha çok izlemeye başladığı gözlemlendi. Çocuğun yakından izlemekteki ısrarı bizde bir göz bozukluğu olabileceği düşüncesini uyandırdı. Çocuğun ablasına gerekli uyarıları yaptık.

2- Elektronik öğretmenimizin evinde yaptığımız çalışma.

Bütün öğretmenlerimizden küçük yaşta çocuğu olanlardan rica ettiğimizde elektronik öğretmenimiz kabul etti. Sistemin kurulacağı yerle ilgili bazı veriler şunlardır;

- Evdeki çocuk sayısı: 2
- Çocukların yaşları: 2 (Kız), 11 (Erkek)
- Odanın büyüklüğü: 3×5 m
- Televizyonun yeri: sehpa üzerinde 3 m’lik alanda
- TV Ekranı :80 cm
- Test süresi: 10 gün

Sistem kurulmadan önceki durum: 11 yaşındaki erkek çocuktan herhangi bir yakından izleme davranışı yokken, 2 yaşındaki kız çocuğunda özellikle çizgi film izlerken yakından izleme davranışı vardır. Anne, baba uygun şekilde oyuncakla ya da dikkatini çekecek başka bir şeyle kandırdığında bir süreliğine çocuğun televizyondan uzaklaştığı fakat sonra tekrar televizyonun önüne geldiği baba tarafından söylenmiştir.

Sistem kurulduktan sonraki durum: Zamanla küçük çocuk sensörlerin görmediği noktaları belirlemiş ve televizyonu oradan izlemeye başlamış. Sensörler uygun yerlere konulduğunda ise kör noktalar ortadan kalkmış. Anne, babanın televizyonun kapanacağına dair verdiği uyarıları dinleyerek geri çekilmiştir.

Büyük çocuk kumandadan televizyonu kendisi kapatarak sistemin çalışma döngüsünü bozmuş ve televizyonun devamlı açık kalmasını sağlamıştır. Bu durum sistemin açık noktası olmuştur. Düzeltmek için sistemin kumandadan açma-kapama düğmesine basıldığını algılayıp televizyonun kapalı durumuna göre çalışmasını sürdürmesi gerekmektedir.

Küçük çocuk sistem söküldüğünde sistemin orada olmadığını anlayarak tekrar yakından izlemeye devam etmiştir. Bundan dolayı sistem yerleştirilirken televizyondan ayrı bir cihaz gibi değil de, televizyona ait bir sistem gibi yerleştirilmesi gerekmektedir.

Sonuçlar

Yaptığımız çalışmalar ile aşağıdaki sonuçları elde ettik;

- Büyük çocuklarda sistemimiz alışkanlık kazandırmamış fakat televizyonu yakından izlemelerini engellemiştir.
- Sistem yerleştirilirken 2 sensör arası mesafe 30 cm'i geçmemeli, aksi takdirde kör noktalar oluşabilir. Büyük ekranlı televizyonlarda 3. sensöre ihtiyaç duyulabilir.
- Sistem ayrı bir cihaz gibi değil de televizyonun bir parçasıymış gibi yerleştirilmelidir. Alışkanlık kazandırması için bunun yapılması gerekir.
- Alışkanlık kazandırması için 10 günlük süre yetmemiştir. Daha uzun süre ile sistemin uygulanması gerekir.
- Sistem bağlı olmasına rağmen eğer çocukta yakından izleme davranışı devam ediyorsa bir göz kusuru olabilir. Çocuğun bir göz doktoruna gösterilmesi uygun olur.

Öneriler

- Televizyon üreticileri tarafından üretilen televizyonlarda böyle bir özelliğin olması, sistemin doğal olmasını sağlayacağından yakından izlememe alışkanlığını daha iyi kazandırılacaktır. Televizyon üreticilerinin televizyonlarına böyle bir özellik koymaları uygun olacaktır. Bu durum televizyon maliyetine de fazla etki etmez. Özellikle anne, babalar tarafından böyle bir televizyon tercih sebebi olacaktır.
- Günümüzde özellikle akıllı televizyon (Smart TV) adı verilen televizyonların kullanımı yaygınlaşmaktadır. Bu televizyonların birçoğunda da kamera bulunmaktadır. Kamera ile mesafe ve çocuğun yüzünün televizyona dönük olduğu anlaşılabilir. Dolayısıyla sistemimizde kullanılan algoritma kolaylıkla yazılımsal olarak bu televizyonlara entegre edilebilir. Böylece sensörleri kullanmaya gerek kalmaz ve sistem televizyonun bir özelliği haline gelmiş olur.

Kaynakça

Elmas, Ç. 2007. Yapay Zeka Uygulamaları, Seçkin Yayıncılık, Ankara

E. T. Aysun & R. Kazan. Bulaşık Makinesinin Bulanık Mantık ile Modellenmesi, Mühendis ve Makine (cilt 48, sayı 565)

Fuzzy Logic, <http://www.massey.ac.nz/~nhreyes/MASSEY/159741/Lectures/Lec2012-3-159741-FuzzyLogic-v.2.pdf>, Erişim Tarihi:10.10.2015

Fuzzy Logic Tool Box User's Guide.

http://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf, Erişim Tarihi :1.10.2015

Led TV İzleme Mesafeleri. http://forum.donanimhaber.com/m_67371403/tm.htm, Erişim Tarihi: 3.10.2015

Sefi, Ö. Teknolojinin Çocuklara Zararları Nelerdir, Bilgi Ustam:

<http://www.bilgiustam.com/teknolojinin-cocuklara-zararlari-nelerdir/>, Erişim Tarihi: 1.10.2015

Televizyonu Yakından İzlemenin Zararları,(2013).

<http://www.hastalikedavi.net/televizyonu-yakindan-izlemenin-zararlari.html>, Erişim Tarihi: 1.10.2015

Televizyonu izleme uzaklığı ne olmalı? <http://www.ntv.com.tr/saglik/televizyon-izleme-uzakligi-ne-olmali,KjoU-s-z7kWeNNHrsVoqHA>, Erişim Tarihi :3.10.2015

Ultrasonic ranging module, <http://www.robotshop.com/media/files/PDF/datasheet-sen026.pdf>, Erişim Tarihi: 12.10.2015

V. Greg (1993), Fuzzy Logic in

http://www.chebucto.ns.ca/Science/AIMET/archive/ddj/fuzzy_logic_in_C/, Erişim Tarihi : 10.10.2015

Yakından televizyon izlemenin zararları, (2012). <http://www.izafet.net/threads/yakindan-televizyon-izlemenin-zararlari.671234/>, Erişim Tarihi: 1.10.2015

Yılmaz, S. Bulanık Sistemler ,

http://www.yarbis1.yildiz.edu.tr/web/userCourseMaterials/eakdogan_784d798d417e811a3ba5a79ff8c46a30.pdf, Erişim Tarihi: 2.10.2015

Ekler

EK1: Çalışmada kullanılan ana fonksiyon

```
void loop(){
    olc=millis();
    s1=HC_SR04(trig1,echo1);
    s2=HC_SR04(trig2,echo2);

    girisleriBulaniklastir(s1,z1);
    bulanikOperatorUygula(bulanikS1, bulanikZ1);
    tv1= durulama();

    girisleriBulaniklastir(s2,z1);
    bulanikOperatorUygula(bulanikS2, bulanikZ1);
    tv2= durulama();

    tv=max(tv1,tv2);

    if(tv==kapali){ //1.0 * kapalı:TV açık ise kapat değilse kapalı kalsın.
        digitalWrite(tvLed,HIGH);
        if(tvAcik){
            tvOnOff(1);
            // tvOnOff(2);//esra tv
            tvAcik=0;
        }
    }
    else if(tv==acik){//0.0 * açık: TV kapalı ise aç değilse açık kalsın

        digitalWrite(tvLed,LOW);
        if(!tvAcik){
            tvOnOff(1);
            //tvOnOff(2);//esra tv
            tvAcik=1;
            delay(5000);//tv nin açılmasını BEKLE
        }
    }

    if((s1<UzSinir)||(s2<UzSinir)){ //150

        digitalWrite(uyariLed,HIGH);
        //uyariLedBlink(25);

        olc_ms2=millis();
        if((olc_ms2-olc_ms)>750){
            olc_ms=millis(); //sureyi yeniden başlat
            z1++;
        }

        if((s1<UzYakin)||(s2<UzYakin)){ //100
```

```

        uyariLedBlink(25);
    }
}
else{
    digitalWrite(uyariLed,LOW);
    z1=0.0;
}

Serial.print("S1 : "); Serial.println(s1);
Serial.print("S2 : "); Serial.println(s2);
Serial.print("Z1 : "); Serial.println(z1);

Serial.print("TV : "); Serial.println(tv);
Serial.print("sure : "); Serial.println(millis()-olc);
Serial.println("-----");
}

```