

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**EOG'NİN KODLANMASINA DAYANAN BİLGİSAYAR
TABANLI GÖZLE YAZI YAZMA SİSTEMİ GELİŞTİRİLMESİ**

HESNA ÖZBEK ÜLKÜTAŞ

YÜKSEK LİSANS TEZİ

2015

EOG'NİN KODLANMASINA DAYANAN BİLGİSAYAR TABANLI GÖZLE YAZI YAZMA SİSTEMİ GELİŞTİRİLMESİ

HESNA ÖZBEK ÜLKÜTAŞ

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
BİYOMEDİKAL Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2015

“EOG’NİN KODLANMASINA DAYANAN BİLGİSAYAR TABANLI GÖZLE YAZI YAZMA SİSTEMİ GELİŞTİRİLMESİ” başlıklı bu çalışma, jürimiz tarafından, 29/04/2015 tarihinde, BİYOMEDİKAL MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Başkan

Yrd. Doç. Dr. Derya YILMAZ

Üye (Danışman)

Yrd. Doç. Dr. Metin YILDIZ

Üye

Yrd. Doç. Dr. Aykut ERDAMAR

ONAY

..../04/2015

Prof. Dr. Emin AKATA
Fen Bilimleri Enstitüsü Müdürü

TEŐEKKÖR

Tezim süresince hem teorik ve teknik bilgisi ile hem de manevi desteęi ile her daim yanımda olan tez danışmanım Yrd. Doę. Dr. Metin YILDIZ'a, deęerli görüşlerini benimle paylaşan hocalarım Yrd. Doę. Dr. Aykut ERDAMAR'a ve Yrd. Doę. Dr. Derya YILMAZ'a, bilgi ve tecrübesini benden esirgemeyen sevgili Yeşim GÜNHAN'a ve hayatımın her aşamasında desteęini esirgemeyip yanımda olan aileme, sonsuz teşekkürlerimi sunarım.

ÖZ

EOG'NİN KODLANMASINA DAYANAN BİLGİSAYAR TABANLI GÖZLE YAZI YAZMA SİSTEMİ GELİŞTİRİLMESİ

Hesna Özbek ÜLKÜTAŞ

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Biyomedikal Mühendisliği Anabilim Dalı

İşaret diliyle ya da konuşarak iletişimini sağlayamayan (Amyotrofik Lateral Skleroz (ALS), Locked-in Sendromu, felç gibi hastalıkları olan) kişilerin iletişimi için çok çeşitli sistemler önerilmiştir. Bunlar içinde, bu hastaların gözlerinin kontrolünü nadiren kaybettikleri bilgisine dayalı olarak, gözle yazı yazma sistemleri öne çıkmaktadır. Göz hareketlerinin kamera veya elektrookülogram (EOG) ile takip edilip yazıya çevrildiği şimdiye kadarki sistemlerin yazma hızları, genellikle 10 kelime/dakika civarındadır. Bu çalışmada ilk önce, daha önce gözle yazı yazmak için geliştirilmiş bir sistem üzerinde yazma hızı ve yazılmak istenen karakterin doğru tespit edilebilme oranını etkileyen faktörleri tespit etmeye dönük çalışmalar gerçekleştirilmiştir. Bu çalışmanın sonuçları dikkate alınarak; gözlerin 4 farklı yön, 2 farklı bakma açısı ve bakılan yönde farklı bekleme süreleriyle durulması ile karakterlerin kodlandığı yeni bir gözle yazma yöntemi önerilmiştir. Önerilen yöntemin uygulanması için bilgisayar tabanlı bir sistem geliştirilmiştir. Geliştirilen sistem donanım ve yazılım olmak üzere iki kısımdan oluşmaktadır. Donanım kısmı yatay ve düşey göz hareketleri sırasında ortaya çıkan EOG sinyallerini alıp işleyerek bilgisayara atarken, yazılım kısmı bu sinyalleri anlamlandırıp seslendiren ve yazıya çeviren programları içermektedir. Önerilen yeni yöntemin etkinliği yazma hızı ve karakterlerin doğru kodlanma oranı parametrelerine göre analiz edilmiştir. Önerilen yeni yöntem ve geliştirilen sistemle ulaşılan azami yazma hızı 17,5 kelime/dakikadır. 126 harften oluşan bir metnin yazılmasının ilk denenmesinde %92,5'lik doğru kodlama oranı sağlanmışken dördüncü denemeden itibaren %100'lük doğru kodlama seviyesine ulaşılmıştır. Ayrıca tekrarlamalar sonucu yazma hızı ve doğruluğunun birlikte arttığı görülmüştür. Buna göre geliştirilen sistemin, etkin kullanımı için harflerin nasıl kodlanacağını öğrenildiği bir alışma dönemi gerekmesine rağmen, bu dönem atlatıldığında, daha önce önerilen benzer

sistemlere göre en az %50 daha hızlı şekilde gözle yazı yazılmasına olanak vereceđi sonucuna varılmıřtır.

ANAHTAR SÖZCÜKLER: EOG, Beyin Bilgisayar Arayüzü, EOG tabanlı gözle yazı yazma sistemleri, kameralı göz takip sistemleri

Danışman: Yrd. Doç. Dr. Metin YILDIZ, Başkent Üniversitesi, Biyomedikal Mühendisliđi Bölümü.

ABSTRACT

DEVELOPMENT OF COMPUTER BASED EYE-WRITING SYSTEM BY USING EOG

Hesna Özbek ÜLKÜTAŞ

Başkent University Institute of Science and Engineering

Department of Biomedical Engineering

There are various recommended methods for patients with Amyotrophic Lateral Sclerosis (ALS) and Locked-in Syndrome (LIS) as well as for paralyzed patients who are not able to speak and/or not able use sign language for communication. In very few cases, the eye muscles become paralyzed for these patients. Thus, based on this evidence, the methods that code eye movements into characters used for writing surpass the others. The speed of existing methods that code eye movements into characters by using camera or the electrooculogram (EOG) is about 10 words per minute in general. In this study, firstly, the methods developed for coding eye movements into characters were analyzed in order to determine coding speed and the accuracy rate of the character that is aimed by an eye movement in these techniques. Based on the results of this study, a new eye movements coding technique, which depends on the eye movements in 4 different directions and 2 different view angles and different waiting time in each of these directions with characters coded based on in each specific eye movement direction, is suggested. Computer based system is developed for implementation of suggested technique. This developed technique consists of two parts; hardware and software. While hardware part records EOG signals that occur during horizontal and vertical eye movements and puts this information into a computer, software part infers and vocalizes these signals by coding these into characters consisting letters, numbers and punctuation marks in order to have proper words and sentences with the help of programs. The efficiency of this newly suggested technique is analyzed through assessing parameters that indicate the speed of coding eye movements into characters and the accuracy rate of coding. The speed of coding eye movements into characters with suggested new technique developed in this study is 17.5 words per minute. While the accuracy rate of coding into right characters in the first trial of a text comprised of

126 letters is 92.5%, this accuracy rate reaches to 100% starting from the fourth trial of the technique. Moreover, this repetition of using this new technique improves both the speed and the accuracy rate of coding into right characters. All these findings imply that even though this newly developed methodology requires a settling-in period needed for grasping how to code efficiently eye movements into characters, after passing this period, this new eye movement coding method developed in this study would be able to improve the speed of coding at least 50% faster than the previously suggested similar ones.

KEY WORDS: EOG, Brain Computer Interface, EOG-based eye typing systems, Eye tracking systems with camera

Advisor: Assistant Professor Metin YILDIZ, Başkent University, Biomedical Engineering Department.

İÇİNDEKİLER LİSTESİ

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	iii
İÇİNDEKİLER LİSTESİ	v
ŞEKİLLER LİSTESİ.....	vi
ÇİZELGELER LİSTESİ.....	vii
ŞİMGELER VE KISALTMALAR LİSTESİ	viii
1. GİRİŞ.....	1
2. GENEL BİLGİLER ve LİTERATÜR ÖZETİ	4
2.1 Kas Sisteminin Kontrol Edilemediği Hastalıklar	4
2.1.1 Amyotrofik Lateral Skleroz (ALS)	4
2.1.2 Locked-in Sendromu	4
2.1.3 Serebral Palsi (SP).....	4
2.1.4 Multiple Skleroz (MS)	5
2.1.5 Hemiparezi (Hemipleji)	5
2.2 Gözle Yazı Yazmaya Duyulan İhtiyaç	5
2.3 Sinir-Kas Hastalarının İletişiminde Kullanılan Sistemler	6
2.3.1 Beyin bilgisayar arayüzü (BBA).....	6
2.3.2 Kameralı gözle yazı yazma sistemleri	7
2.3.3 EOG tabanlı gözle yazı yazma sistemleri.....	10
2.3.3.1 <u>EOG</u>	10
2.4 Gözle Yazı Yazma Sistemlerinin Karşılaştırılması	14
3. MATERYAL ve METOD	15
3.1 Daha Önce Geliştirilmiş EOG Tabanlı Gözle Yazı Yazma Sisteminin İncelenmesi	15
3.2 Önerilen İlk Gözle Yazma Prosedürü	18
3.3 Önerilen Yeni Gözle Yazma Prosedürü.....	18
3.4 Gerçek Zamanlı Gözle Yazı Yazma Sistemi Geliştirilmesi	21
3.4.1 Donanım kısmı	24
3.4.1.1 <u>Toprak sürücü devresi</u>	24
3.4.1.2 <u>Enstrümantasyon yükselteci</u>	27
3.4.1.3 <u>İkinci derece butterworth yüksek ve alçak geçiren filtre devresi</u>	28
3.4.1.4 <u>Kazanç yükselteci devresi</u>	29
3.4.1.5 <u>Optik aktarıcı devresi</u>	30
3.4.1.6 <u>Verilerin bilgisayara aktarılması</u>	30
3.5 Gerçek Zamanlı Gözle Yazı Yazma Sistemi Yazılımı.....	31
4. SONUÇLAR	39
5. YORUM VE ÖNERİLER	53
6. KAYNAKÇA	55
EKLER LİSTESİ	58

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1	BBA tabanlı bir sistem için genel blok şema [14]	7
Şekil 2.2	Darius ve ark.'nın 2003'deki çalışmalarında karakterlerin örnek yazılışını gösteren arayüz [33]	8
Şekil 2.3	Majaranta ve ark.'nın 2004'deki çalışmalarında karakterlerin örnek yazılışını gösteren arayüz [32]	8
Şekil 2.4	MacKenzie ve ark.'nın 2008'deki çalışmalarındaki kullanıcı arayüzü [35]	9
Şekil 2.5	Kristensson ve ark.'nın 2012 çalışmalarındaki kullanıcı arayüzü [16]	10
Şekil 2.6	Tsai ve ark.'nın 2007 'deki çalışmalarında karakter yazımı için kullanılması gereken hareket sayısı [5]	11
Şekil 2.7	Wobbrock ve ark.'nın 2007'deki çalışmalarından alınmıştır [6]	11
Şekil 2.8	Akan ve ark.'nın 2007'deki çalışmalarından alınmıştır [31]	12
Şekil 2.9	Porta ve Turina'nın 2008'deki çalışmalarından alınmıştır [7]	13
Şekil 2.10	Uşaklı ve ark.'nın 2009'daki çalışmalarından alınmıştır [4]	13
Şekil 3.1	Altı farklı göz hareketi sırasında kaydedilen EOG sinyalleri	16
Şekil 3.2	Yatay ve düşey göz hareketlerinin algılanması için yapılan elektrot bağlantı şekli [36]	16
Şekil 3.3	Bir yönde farklı bakma açıları ve bekleme sürelerinde oluşacak EOG sinyalleri	19
Şekil 3.4	Donanım blok diyagramı	25
Şekil 3.5	Devre şeması	26
Şekil 3.6	Toprak sürücü devre kısmı	27
Şekil 3.7	İkinci derece butterworth yüksek ve alçak geçiren filtre	29
Şekil 3.8	Kazanç yükseltici devresi	29
Şekil 3.9	Optik aktarıcı devre şeması	30
Şekil 3.10	National Instruments marka USB-NI6009 model DAQ kartı	31
Şekil 3.11	Yazılım akış diyagramı	34
Şekil 3.12	Tasarlanan arayüzde dil seçme ekranı	36
Şekil 3.13	Tasarlanan İngilizce arayüz ekranı	37
Şekil 3.14	Tasarlanan Türkçe arayüz ekranı	38
Şekil 4.1	Dört farklı açı ile bakılması sırasında ortaya çıkan EOG sinyalleri ve türevleri	42
Şekil 4.2	Yukarı yönde bakışta kodlanan i harfinin art arda 10 kez yazılması	44
Şekil 4.3	Sol yönde bakışta kodlanan r harfinin art arda 10 kez yazılması	45
Şekil 4.4	Yukarı yöndeki farklı açı ve bekleme süreleri ile harflerin kodlanması ..	46
Şekil 4.5	Türk alfabesindeki tüm harflerin yazılması	47
Şekil 4.6	"i love you" cümlesinin yazılması	48
Şekil 4.7	Nerede kelimesinin yazımını gösteren arayüz ekranı	50

ÇİZELGELER LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1 Literatürdeki çalışmaların hız ve doğruluk kıyaslaması	14
Çizelge 3.1 Türkçede en çok kullanılan harflere göre oluşturulan alfabe	20
Çizelge 3.2 Önerilen yeni gözle kodlama yöntemi.....	22
Çizelge 3.3 Sistemden beklenen maksimum hız tespiti.....	23
Çizelge 3.4 Harflerin yerlerinin gösterildiği tablo.....	33
Çizelge 4.1 Yıldız M. çalışmasının performans testi sonuçları	39
Çizelge 4.2 Günlük hayatta kullanılabilecek kelimelerin ortalama, en hızlı, en yavaş yazma süreleri.....	51

SİMGELER VE KISALTMALAR LİSTESİ

EOG	Elektrookülografi
EEG	Elektroensefalogram
ALS	Amyotrofik Lateral Skleroz
BBA	Beyin Bilgisayar Arayüzü
MS	Multiple Skleroz
SP	Serebral Palsi
CMRR	Ortak Mod Sinyalini Bastırma Oranı
DAQ	Data Acquisition
MATLAB	Matrix Laboratory
GUI	Graphical User Interface
GTP	Görsel Tetiklenen Potansiyel

1. GİRİŞ

Çevre ile iletişimini konuşarak ya da işaret dili ile sağlayamayan, Amyotrofik Lateral Skleroz (ALS) gibi sinir kas hastalıkları bulunan veya felçli kişilerin istek ve düşüncelerini çevreye aktarması için pek çok sistem önerilmiştir.

Bu yöntemlerden bazıları beyin bilgisayar arayüzü (BBA) genel ismi ile anılmaktadır. Bu sistemler genel olarak; kişinin isteklerini yazıya dökebilmek için beyninden kaydedilen sinyallerle yazmak istediği kelimeye ait harfleri bir bilgisayar ekranından seçmesi mantığına dayanmaktadır. Bu tür sistemlerin dezavantajı yazı yazma hızının dakikada 5-10 harf gibi çok yavaş bir seviyede olmasıdır.

Bu tür hastaların gözlerini kullanma kabiliyetlerini kaybetmediği bilgisinden yola çıkan bazı çalışmalarda ise; kişinin yazmak istediği kelimeye ait karakterleri göz hareketleri ile seçmesine dayanan benzer bir yöntem kullanılmaktadır. Bu sistemler genellikle bir bilgisayar ekranında görülen klavye üzerinde yazılmak istenen harfin üzerine odaklanılmasına ve bu odaklanmanın ya bir kamera ya da gözlerin ürettiği EOG sinyalleri vasıtasıyla belirlenip, yazılması ilkesine dayanmaktadır [1,2,3].

Çeşitli yönlerdeki bakışların kombinasyonları ile karakterlerin kodlandığı gözle yazı yazma sistemleri, bilgisayar ekranında takip edilecek bir sanal klavyeye ihtiyaç duymamaları bakımından öncekilere göre bir üstünlüğe sahiptir. Bu yöntem ile dakikada yazılabilen karakter sayısında önceki sistemlere göre artışlar sağlanmıştır. Buna karşın, karakter kodlaması ile ilgili göz hareketlerinin öğrenilmesi zorluğu, bir karakterin çok sayıda göz hareketi ile kodlanması gibi dezavantajları mevcuttur [19].

Diğer bir yöntemde ise klavye kullanmak yerine karakterlerin gözle çizilmesi gündeme gelmiş fakat bu da istenen artışı sağlamamıştır [5]. Bir karakteri yazmak için çok sayıda göz hareketi yapılması gerektiğinden sistemin çalışma hızı düşüktür.

Bu çalışmada göz hareketlerinin yazıya çevrilmesi için; göz hareketleri sırasında, göz dinlenim potansiyellerinde oluşan sapma sonucu ortaya çıkan EOG sinyalleri kullanılmıştır. Göz hareketleri sırasında oluşan EOG sinyallerinin yazıya çevrilmesi

iin bir kodlama teknięiyle daha nceki sistemlere gre daha az sayıda gz hareketi ile karakterlerin kodlanmasına dayanan, bir yntem nerilmiřtir.

Bu alıřmanın ana amacı daha nce geliřtirilmiř EOG tabanlı gzle yazı yazma tetkiklerine gre daha hızlı ve daha az hata ile yazı yazılmasını saęlayabilecek bir yntem nermek ve bunu uygulamaktır.

Mmkn olduęu kadar az hata ile ve hızlı řekilde gzle yazı yazmaya yarayacak bir yntem nerebilmek iin, daha nce Yıldız M. 2011 tarafından geliřtirilen gzle yazı yazma sistemi eřitli durumlar iin test edilmiřtir. Bu alıřmadan elde edilen sonular ışığında yeni bir gzle yazma yntemi nerilmiřtir.

nerilen yeni gzle yazı yazma ynteminin uygulanması iin bilgisayar tabanlı bir sistem geliřtirilmiřtir. Geliřtirilen sistem donanım ve yazılım olmak zere iki kısımdan oluřmaktadır. Donanım kısmı yatay ve dřey gz hareketleri sırasında ortaya ıkan EOG sinyallerini alıp iřleyip bilgisayara atarken, yazılım kısmı bu sinyalleri anlamlandırıp seslendiren ve yazıya eviren program paralarını iermektedir.

Belirlenen yntemde kullanıcı saę, sol, yukarı ve ařaęı olmak zere 4 yne, 2 farklı aı ile (yakın/uzak bakıř) ve 4 bekleme sreli baktığında tek ynde 8, 4 ynde ise 32 karakter kodlanabilmektedir. Trkede en ok kullanılan karakterlerin en kısa srede kodlanan bakıř ve bekleme sresi atanması ile yazma hızının artması hedeflenmiřtir.

Geliřtirilen sistem yazma hızı ve karakterlerin doęru kodlanma oranı parametrelerine gre analiz edilmiřtir. Hastalar zerinde denenmesi etik kurul iznine baęlı olduęundan kendi zerimizde yapılan denemeler sonucu ulařılan maksimum yazma hızı; dakikada 17,5 kelime, 126 harften oluřan bir metnin yazılmasının ilk denenmesinde % 92,5 lik doęru kodlama oranı saęlanmışken drdnc denemeden itibaren %100 lk doęru kodlama seviyesine ulařılmıştır. Ayrıca tekrarlamalar sonucu yazma hızı ve doęruluęunun birlikte arttığı grlmřtr. Literatrde ilk defa olarak, kullanılan karakterleri seslendirerek, kodlanacak karakterin seimi kolaylařtırılmış bu da yanlış kodlanan karakter sayısını azaltmıştır.

Geliştirilen yöntem ile farklı yönlerdeki bakışlara farklı dillerin alfabelerindeki harfler atanarak, birçok farklı dil için kullanılabilir. Bu sistem dünyadaki bu tür hastalıklardan muzdarip kişilerin çevreleri ile daha hızlı iletişim kurmalarını sağlayabilecektir. Ayrıca geliştirilen sistemin gömülü sistem olarak tasarlanması mümkün olup, daha ucuz bir şekilde imal edilebilir.

Bu çalışma dört bölümden oluşmaktadır. İlk bölümde genel bilgiler başlığı altında bu sistemin kullanımını gerektirecek hastalıklar ve görülme sıklığı hakkında bilgi verilmiştir. İkinci bölümde literatürde daha önce bu konuda yapılan çalışmalar üç ana başlık altında incelenmiştir. Bunlar; BBA, kameralı gözle yazı yazma sistemleri ve EOG tabanlı gözle yazı yazma sistemleridir. Bu sistemler kullanılarak yapılan çalışmaların avantaj ve dezavantajları anlatılmıştır. Üçüncü bölümde sistemi tasarlama aşamasında kullanılan yöntemler anlatılmaktadır. Dördüncü bölümde ise gerçekleştirilen çalışmaların sonuçları verilip bu sonuçlar yorumlanmıştır.

2. GENEL BİLGİLER ve LİTERATÜR ÖZETİ

Aşağıda çevresi ile iletişimini konuşarak veya işaret diliyle sağlayamayan hastalık tipleri ve bu hastalıklara sahip kişilerin iletişimi için geliştirilen sistemlerle ilgili genel bilgiler verilmiştir.

2.1 Kas Sisteminin Kontrol Edilemediği Hastalıklar

2.1.1 Amyotrofik Lateral Skleroz (ALS)

ALS, merkezi sinir sisteminde omurilik ve beyin sapı adı verilen bölgede motor sinir hücrelerinin kaybindan kaynaklanan bir hastalıktır. Bu hücrelerin kaybı kaslarda güçsüzlük ve kas erimesine yol açar. Zihinsel fonksiyonlar ve bellek ise bozulmaz. Kaslardaki zayıflık ellerde ya da bacaklarda, ağız-yutak bölgesinde ya da dilde başlayabilir ve sürekli ilerleyerek yayılır. Bu yayılma "bulber" alandaki kasları da tutabileceği için konuşma ve yutma güçlüğüne neden olabilir. İleri evrelerinde solunum yetersizliğine de yol açabilir. Genellikle erişkin yaşlarda (40-50) ve erkeklerde, kadınlara göre biraz daha sık görülür. Görülme sıklığı 100.000 de 1-1,5 civarındadır [26].

2.1.2 Locked-in Sendromu

Locked-in Sendromu ilk defa 1966 yılında Posner ve Plum adlı araştırmacılar tarafından "beyin sapındaki kortikospinal ve kortikobulbar yolların hasarlanması sonucu sırasıyla kuadripleji (ön ve arka bacakların birlikte felci) ve anartri (dil tutukluğu)'nin bir arada gözlenmesi" olarak tanımlanmıştır. Locked-in Sendromu, bir insanın şuuru yerinde olmasına rağmen bedenen neredeyse tamamen felç halinde olması ve bundan dolayı kendini dil veya hareketlerle ifade edememesi durumuna denir. Bu durumda olan hastaların dışarıyla iletişimi için kullanabilecekleri yalnızca göz hareketi kalmıştır. Bu imkânın da kalmamış olması durumunda son çare, bir BBA'dır [24,25].

2.1.3 Serebral Palsi (SP)

Serebral Palsi, gelişimini sürdüren beyinde kalıcı bir hasar sonucu kol, bacak ve gövdede gelişen, ilerleyici özellik taşımayan motor bozukluğu ifade eder. Bunun sonucunda hareket ve duruş bozuklukları oluşur; ayrıca görme, konuşma ve yutma işlevlerinde de sorun yaşanır. Beyindeki hasar kalıcı olmasına ve ilerlememesine karşın, çocuk büyüdükçe ve yeni motor beceriler edindikçe hareket bozukluğunun

biçimi değişmeye eğilim gösterir. Yaklaşık 400 canlı doğumdan 1'inde Serebral Palsi tanısı konmaktadır. Ülkemizde sağlıklı veriler olmamasına karşın bu oranın 150 canlı doğumda 1 olduğu tahmin edilmektedir. Bu oran değişik ülkelerde binde 1 ila 5 arasında değişmektedir [23,28].

2.1.4 Multiple Skleroz (MS)

Multiple Skleroz beyin ve omurilikte iltihabın izlendiği bir nörolojik hastalıktır. MS' te hastalığa neden olan yanlış çalışan bağışıklık sistemi hücreleri, beyin ve omurilikteki hücrelerin (nöronların) sinir kılıflarının hasar görmesine neden olmaktadır. Bunun sonucunda nöronlar işlevlerini doğru şekilde yerine getirememektedir. Nöronlar; görme, konuşma, yürüme gibi fonksiyonlar üzerinde kişinin kontrol kabiliyetini bozar. Koordinasyon bozulur, kuvvet azalır. Aşırı halsizlik yapar. Uyuşma, hissetmeme görülebilir, hasta felç olduğunu zannedebilir. MS'in görülme sıklığı dünya üzerinde değişik coğrafya ve iklimlerde farklılıklar gösterir. MS'in Türkiye'deki görülme sıklığı yüz binde 40-60 civarındadır. Hastalık çoğunlukla 20-30 yaş civarında görülür. Daha küçük yaşlar ve 60 yaş sonrası nadiren ortaya çıkar. Hastalık kadınlarda erkeklere oranla 2-3 kat daha fazla görülmektedir [22,29].

2.1.5 Hemiparezi (Hemipleji)

Vücudun bir yarısında istemli hareket etkinliğinin tümüyle kaybolmasıyla beliren patolojik durumdur. Yarım felç olarakta tanımlanmaktadır. Hemiparezi vücudun bir yarısındaki kaslarda kuvvetsizlik anlamına gelen patolojik durumdur. İstemli hareketler tümüyle kaybolmuştur. Tüm dünyada 7.000 kişide bulunduğu tahmin edilmektedir [21].

2.2 Gözle Yazı Yazmaya Duyulan İhtiyaç

Çevresi ile etkileşimini konuşarak veya işaret dile ile sağlayamayan azımsanmayacak sayıda sinir kas hastalığı bulunan hasta vardır. Bu kas hastalıklardan sadece ALS hastalığının görülme sıklığı 100000 da 1-1,5 seviyesindedir. Türkiye'de ki felçli sayısı ise 140 bin ile 210 bin arasında olduğu sanılmaktadır. Yani 10000 de 2-3 civarında insan felçlidir. Bunların büyük bir çoğunluğu bu sisteme ihtiyaç duymayacak ölçüde çeşitli uzuvlarını hareket ettirebilmektedir. Felçlilerin %10 kadarının gözlerini kullanarak yazı yazmaya

ihtiyaç duyacağı düşünülecek olursa; bu 100000 de 2-3 ilave kişinin sistemden faydalanma potansiyeli olduğu anlamına gelir. Gözleri vasıtasıyla bilgisayar ekranı üzerinde kelime yazabilme, böylece ihtiyaçlarını ifade edebilme ileri yaştaki yaşlılar için de önemli kolaylık sağlayabilir. 2030'da 60 ve üzeri yaştaki nüfusun genç nüfusa oranı bire üç olacağı tahmin edilmektedir. Ortalama ömrün giderek arttığı ve toplumdaki felçli ve diğer kas hastalıkları bulunan kişilerin fazlalığı düşünüldüğünde; gözle yazı yazma sistemine duyulan ihtiyacın ileriki yıllarda giderek artacağı düşünülmektedir [17].

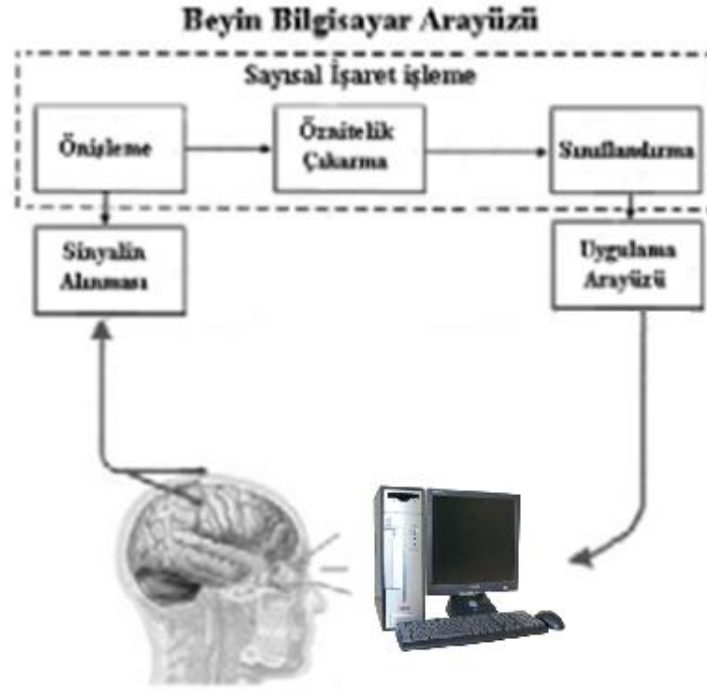
2.3 Sinir-Kas Hastalarının İletişiminde Kullanılan Sistemler

Bu bölümde, yukarıda sayılan sinir ve kas hastalıklarına bağlı olarak çevresi ile konuşarak veya işaret diliyle iletişim kuramayan kişiler için geliştirilen sistemlerden bazıları incelenmiştir.

2.3.1 Beyin bilgisayar arayüzü (BBA)

Jacques J. Vidal 1973 yılındaki çalışmasında EEG sinyallerini algılayıp, çözen, belli örüntüleri bulup bunları önceden tanımlayıp hareket komutları olarak yorumlayan karmaşık bir bilgisayar sistemi geliştirmiştir [31]. Bağımlı BBA olarak tanımlanan bu sistem kullanıcının göz hareketlerini kontrol etme kabiliyetine dayanmaktaydı. BBA'yı kullanabilme yeteneği, beynin ürettiği sinyalleri kaydedip bunları örüntü çözümü ve sınıflandırılmasına uygulayabilmesidir. Beyinde iki tür iletişim gerçekleşir. Bunlar elektriksel ve kimyasaldır. Her iki iletişimde görülebilir etkileri vardır ve bunları bazı cihazlarla tespit etmek mümkündür. BBA açısından önemli olan beyindeki elektriksel iletişimdir. Beyindeki eylem potansiyellerinin tetiklenmesi ve bunların aksonlar boyunca iletilmesi, kafatası yüzeyinde tespit edilebilir fiziksel aktiviteye yol açar. Bağımlı bir BBA sistemi beynin normal çıktı kanallarını kullanır. Bu nedenle bu sistemler işlevsel bir sinir sistemi gerektirir. Popüler bir BBA örneğinde ise, bedeninin büyük kısmı felçli olan bir hastaya ekranda tek tek harfler gösterilir. Hasta seçmek istediği harfi ekranda görünce konsantre olup o harfe bakar. Bu durum, görsel olarak tetiklenen bir potansiyele yol açar Görsel Tetiklenen Potansiyel (GTP) ve bu sinyal kişilerin hangi harfe baktığının tespitinde kullanılmaktadır. Bu hastanın kodlamak istediği

harfe diğerlerine göre daha fazla GTP uygulamasına dayanarak gerçekleşir [17,18]. Şekil 2.1’de BBA tabanlı sistem blok şeması görülmektedir.



Şekil 2.1 BBA tabanlı bir sistem için genel blok şema [14]

BBA sistemleri her ne kadar kullanım kolaylığı ve düzgün öğrenme eğrisinden dolayı faydalı olsa da bunları kullanan hastaların sinir sistemlerinin hasar görmemiş olması gerekir. Ayrıca bu tür sistemlerin dezavantajı yazı yazma hızının dakikada 5-10 harf gibi çok yavaş bir seviyede olmasıdır [19]. Sözü edilen hastaların gözlerini hareket ettirebilme yetisinin bulunması, çalışmaları kameralı ve EOG tabanlı sistemlere yönlendirmiştir.

2.3.2 Kameralı gözle yazı yazma sistemleri

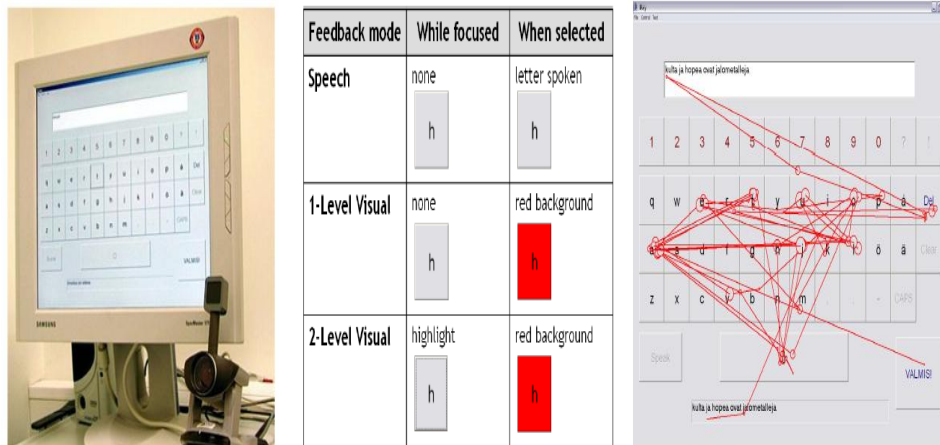
Darius ve ark.'nın 2003'deki çalışmaları sanal klavye kullanılarak ekrandan karakteri seçme prensibine dayanmaktadır [33]. Cep telefonlarının mesaj menüsünde olduğu gibi bir kaç harf aynı butonda olacak şekilde karakterler gruplandırılmıştır. Kullanıcı kodlamak istediği harfi, ekrandaki harf gruplarından bulup onun üzerinde belirlenen zaman süresince (çalışmada 400 ms belirlenmiş) durduğunda gruptaki ilk harf kodlanacaktır. Harf kodlandığında ekranda genişleyerek görülmektedir. Eğer kodlamak istediği harf, gruptaki ikinci harf ise bekleme süresi iki katına çıkacaktır. Şekil 2.2'de sistemin çalışmasının örnek gösterimi bulunmaktadır. Harflerin tek tek

yazılmaması avantajken, ilk kullanımda kullanıcının harflerin yerini öğrenme süresi ve harf seçildiğinde üzerinde belli bir süre beklenmesi yazma hızını düşürmüştür [33]. İlgili çalışmada yazı yazma hızı hakkında bilgi verilmemiştir.



Şekil 2.2 Darius ve ark.'nın 2003'deki çalışmalarında karakterlerin örnek yazılışını gösteren arayüz [33]

Majaranta ve ark.'nın 2004'deki çalışmalarında kameralı göz takip sistemi kullanmışlardır. Kullanıcı bilgisayar ekranında gördüğü sanal klavye üzerinde gözlerini hareket ettirip, yazmak istediği harfin üzerinde odaklanarak istediği harfi ekrana yazmıştır. Şekil 2.3'te sistemin arayüzü görülmektedir. Bir harfin ekrana yazılmasından sonra bekleme süresinin 900 ms olması sistemin yazma hızını yavaşlatmış ve gözün tüm hareketlerinin ekranda kırmızı çizgiler şeklinde görülmesi yeni kodlanacak harf için karışıklığa neden olmuştur [32].



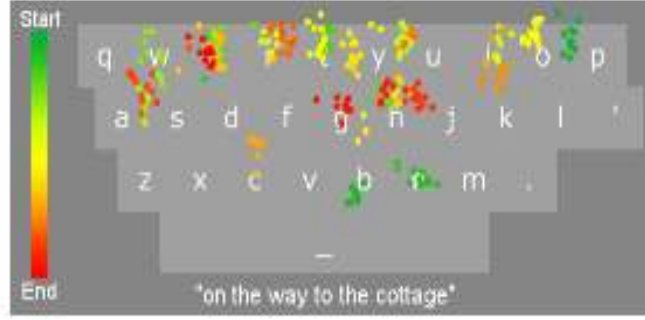
Şekil 2.3 Majaranta ve ark.'nın 2004'deki çalışmalarında karakterlerin örnek yazılışını gösteren arayüz [32]

MacKenzie ve ark.'nın 2008'deki çalışmalarında ise kullanıcı bilgisayar ekranında gördüğü sanal klavye üzerinde gözlerini hareket ettirek yazmak istediği kelimenin harflerini ekrana tek tek kodlamak yerine ilk iki harfini kodlamakta ve ardından o iki harfle başlayan kelimeler ekranda listelenmektedir [35]. Tüm harfleri tek tek kodlanmaması avantajken, tüm kelimelerin sistemde bulunmaması, bulunanların ise ekrana sığmaması ayrıca yalnızca İngilizce kelimeler için çalışması dezavantajdır. Şekil 2.4'de sistemin arayüzü ve kullanımına örnek görülmektedir. Sistemin yazma hızı hakkında bilgi verilmemiştir.



Şekil 2.4 MacKenzie ve ark.'nın 2008'deki çalışmalarındaki kullanıcı arayüzü [35]

2012'de Kristensson ve ark.'nın, Tobii adlı göz takip sistemi yardımıyla çalışan sistemlerinin çalışma prensibi, T9 klavye çalışma prensibine benzemektedir [16]. Akıllı telefonların mesaj menüsünde bulunan klavyeye benzer bir klavyeyi bilgisayar ekranında gören kullanıcı yazmak istediği kelimenin ilk bir kaç harfinin üzerinde gözlerini gezdirdiğinde sistem otomatik olarak yazılmak istenen kelimeye benzer tüm alternatifleri ekrana yazmakta ve kullanıcı böylelikle tüm harfleri yazmadan alternatifler içinden yazmak istediği kelimeyi ekrana yazabilmektedir. Bu sistemin önceki sistemlerden çok daha hızlı çalışması avantajken, sistemde maliyeti oldukça yüksek olan Tobii adlı göz takip sistemi kullanılması dezavantajdır [16]. Şekil 2.5'de sistemin çalışmasının örneği görülmektedir.



Şekil 2.5 Kristensson ve ark'nın 2012 çalışmalarındaki kullanıcı arayüzü [16]

2.3.3 EOG tabanlı gözle yazı yazma sistemleri

2.3.3.1 EOG

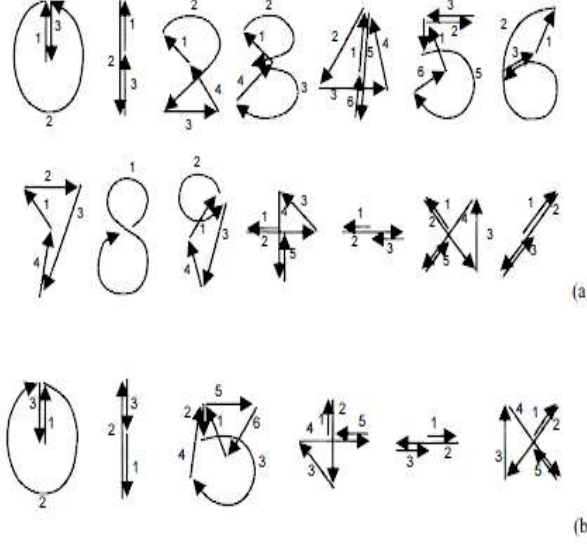
EOG ilk kez 1936 yılında Marg tarafından kullanılan bir terim olup, kornea ile gözün arka segmenti arasında mevcut olan yaklaşık 6 milivoltluk istirahat potansiyelini ifade etmektedir. EOG klinikte; pigment epitelinin primer distrofileri, toksik retinopatiler, fotoreseptör distrofileri veya disfonksiyonları, koroid distrofileri, retina dekolmanı, vasküler hastalıklarda kullanılır [18,32].

1951'de Mowrer ilk kez göz hareketlerini otomatik kaydeden bir sistem geliştirerek, gözün nereye baktığı ve baktığı yöndeki nesneleri algılamak (görmek) için nasıl hareketler yaptığını inceleme imkanı sağlamıştır [30]. Göz takibi veya bakış takibi olarak adlandırılan bu teknik aslında kişinin nereye baktığının bulunmasını sağlamaktaydı. Günümüzde teknolojik gelişmelere paralel olarak, yöntemin kullanımı gelişmiş ve artmıştır. EOG bir çok çalışmada kişilerin gözleri ile iletişimi için bir gereç olarak kullanılmıştır.

Tsai ve ark. 2007, çalışmalarında, sayılar ve bazı matematiksel operatörlerin göz ile yazılıyormuş gibi kodlanması sırasında kaydettikleri EOG sinyallerini çeşitli örüntü tanıma (pattern recognition) yöntemleri ile yazıya çevirmişlerdir [5]. Şekil 2.6'da matematiksel operatörlerin ve rakamların yazım şekli görülmektedir. Bu sistemde bir matematiksel operatörün yazımı için en az 3 göz hareketi gerekmektedir. Bu sayı diğer operatörlerde 6 harekete kadar çıkmıştır. İlgili makalede sistemin yazma hızı hakkında bilgi verilmemiştir.

Wobbrock ve ark. 2007, tüm alfabetik karakterlerin, Tsai ve ark. 2007'nin yöntemine benzer şekilde gözle çizildiği bir yöntem geliştirmişlerdir. Şekil 2.7'de

karakterlerin yazımı görülmektedir [6]. Bir karakterin yazımı için gözün en az iki büyük sıçrama yapması gerekmektedir. Bu sistemin de yazma hızı hakkında bilgi verilmemiştir



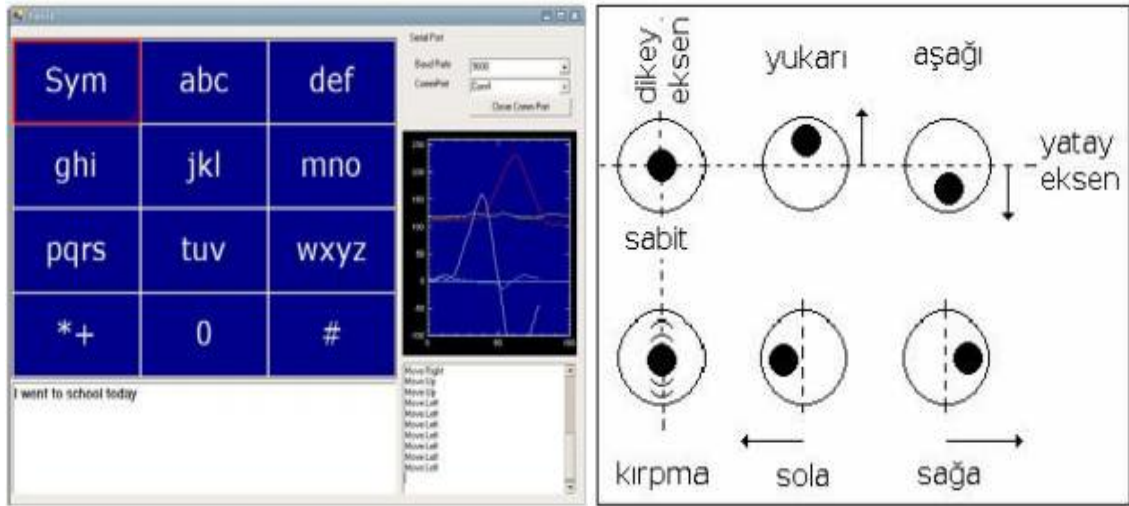
Şekil 2.6 Tsai ve ark.'nın 2007 'deki çalışmalarında karakter yazımı için kullanılması gereken hareket sayısı [5]



Şekil 2.7 Wobbrock ve ark.'nın 2007'deki çalışmalarından alınmıştır [6]

Akan ve ark. 2007, geliştirdikleri uygulamada cep telefonlarında kısa mesaj yazmak için kullanılan ve akıllı klavye olarak bilinen T9 klavye sistemi kullanılmıştır [31]. Yazılım ile bilgisayar ekranı üzerinde cep telefonu tuş takımı şeklinde bir şablon hazırlanmıştır. Bu şablon üzerinde seçilmesi istenilen üçlü harf grubuna EOG sinyali vasıtasıyla erişilir. Seçilmesine karar verilen grup gözün hızlı bir

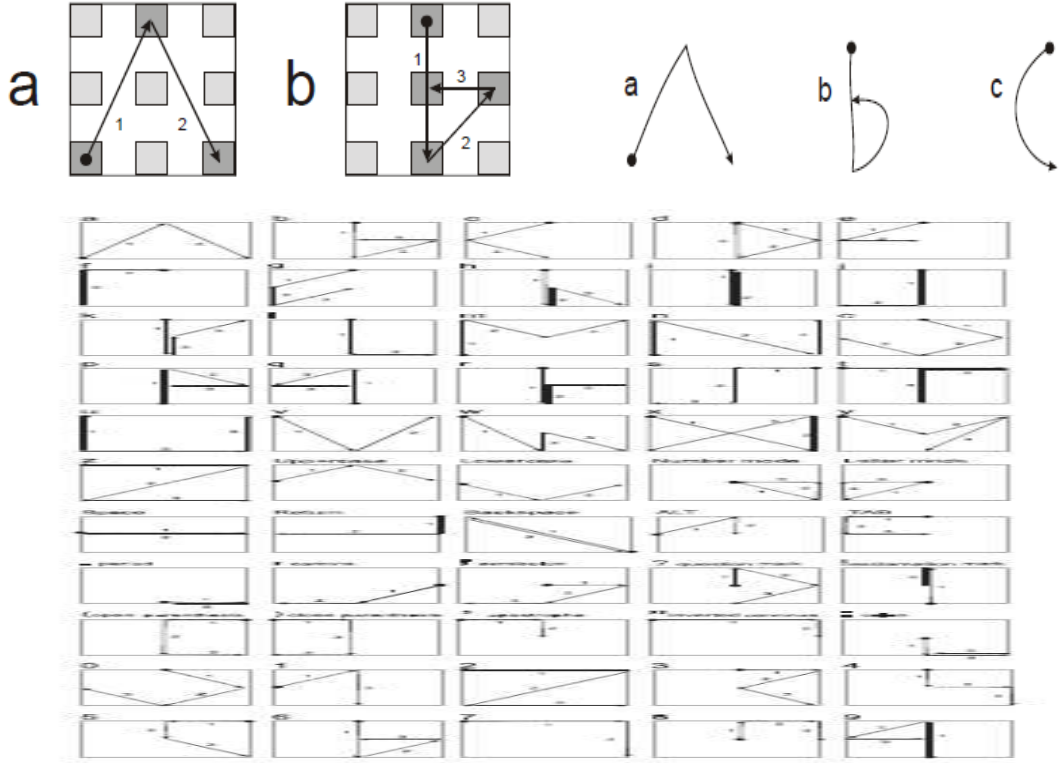
şekilde iki kere kırılması ile belirlenir. Şekil 2.8’de sistemin arayüzü gösterilmiştir. Seçilen üçlü gruplar ekranın başka bir köşesinde grup içerisindeki harflerin oluşturabileceği olası kelimeler olarak listelenir. Bu listeleme işleminde kullanılan kelime dizisinin İngilizce olmasından dolayı sistem yalnızca İngilizce sözcüklerin yazımı için kullanılabilir. Sistem sağlıklı iki kişi üzerinde yapılan denemelerde %85 doğruluk oranı ile çalışmaktadır. 5 harfli bir kelimenin yaklaşık olarak yazdırılabilme hızı 15 saniyedir [31].



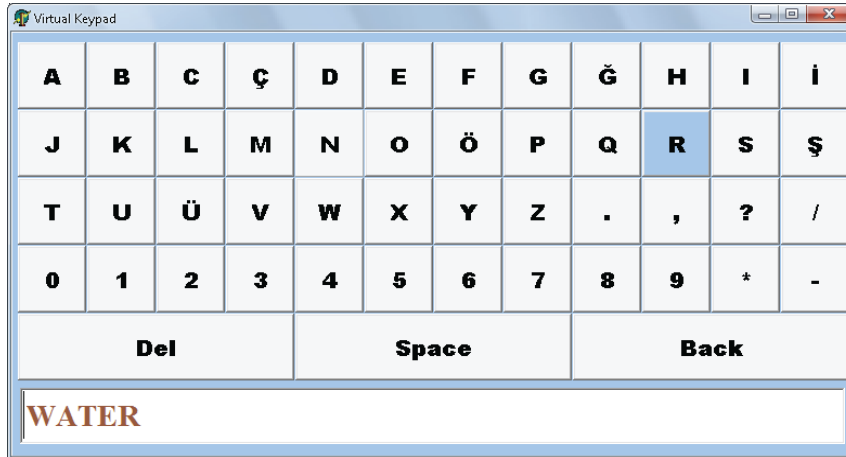
Şekil 2.8 Akan ve ark.'nın 2007'deki çalışmalarından alınmıştır [31]

Porta ve Turina 2008, harf ve rakamların tanınmasını kolaylaştıracak şekilde düzenledikleri, bir dizi göz hareketi ile karakter yazılmasını sağlayan bir yöntem önermişlerdir [7]. Harf ve rakamları ekran üzerindeki bir karenin köşeleri, kenarlarının ortaları ve merkezine yerleştirilmiş 9 nokta arasında çizerek yazıya çevirmişlerdir. Şekil 2.9'da noktalar üzerinde yazılan karakterler görülmektedir. Bu yöntemde bir karakteri kodlamak için 2 ila 4 arası göz hareketi gerekmektedir.

Uşaklı ve ark. 2009 bir bilgisayar ara yüzü ve EOG kullanılarak hastaların bilgisayarla etkileşimini sağlayan bir sistem geliştirmişlerdir [4]. Gerçekleştirilen sistemde gözün yatay ve düşey hareketlerinden elde edilen iki kanallı EOG sinyalleri kullanılarak sanal bir klavye geliştirilmiştir. Sistemin yazma hızı 2,5 kelime/dk'dır. Şekil 2.10'da sistem görülmektedir.



Şekil 2.9 Porta ve Turina'nın 2008'deki çalışmalarından alınmıştır [7]



Şekil 2.10 Uşaklı ve ark.'nın 2009'daki çalışmalarından alınmıştır [4]

Yıldız M. 2011 çalışmasında, gözle yazı yazma sistemleri için karakterlerin daha az sayıda göz hareketi ile oluşturulduğu yeni bir metot önermiştir [19]. Harf, rakam ve noktalama işaretlerinden oluşan 42 farklı karakterin kodlanması için düşey ve yatay düzlemdeki EOG sinyalleri kullanılmıştır. Yöntemin etkinliği öncelikle bilgisayar benzetimleri ile gösterilmiştir. Daha sonra, 3 denekten kaydedilen gerçek EOG sinyalleri ile çevrimdışı olarak test edilmiştir. Bu yöntemle kişiye özgü

yapılacak ayarlamalarla, kodlanan karakterlerin %100'e varan doğrulukla tespit edilebileceği öngörülmüştür. İlgili çalışmada sistemin yazma hızı hakkında bilgi verilmemiştir.

2.4 Gözle Yazı Yazma Sistemlerinin Karşılaştırılması

Çizelge 2.1'de ulaşılabilen literatür çalışmalarında hangi yöntemin uygulandığı, dakikada kaç karakter yazabildiği ve karakter tanıma yüzdeleri verilmiştir. Bu çizelgeye göre çalışmalar kameralı, EOG tabanlı sanal klavyeli ve EOG tabanlı kodlama ile karakter yazılan olmak üzere üç gruba ayrılmıştır.

EOG'nin kodlanmasına dayanan sistemler bilgisayar ekranında takip edilecek bir sanal klavyeye ihtiyaç duymamaları bakımından öncekilere göre bir üstünlüğe sahiptir. Karakterlerin gözle çizildiği yöntem ile dakikada yazılabilen karakter sayısında önceki sistemlere göre artışlar da sağlamıştır. Buna karşın, bu sistemlerin bir karakterin çok sayıda göz hareketi ile kodlanması, karakter kodlaması ile ilgili göz hareketlerinin öğrenilmesinin zorluğu gibi dezavantajları mevcuttur.

Çizelge 2.1 Literatürdeki çalışmaların hız ve doğruluk kıyaslaması

ÇALIŞMA	YÖNTEM	HIZ	DOĞRULUK
Päivi Majaranta ve ark.,2002 [3]	Eye typing - kamera,ekran	9.89 kelime/dk	
Jacob O. Wobbrock, ve ark. 2007 [6]	EyeWrite - kamera,ekran	7.99 kelime/dk	%1.25 hata
Darius Miniotas ve ark. 2000 [13]	Symbol creator - kamera,ekran	8.58 kelime/dk	%3.4 hata
A. B. Usakli, ve ark. 2009 [12]	EOG tabanlı sistem EOG,ekran	2.5 kelime /dk	%5 hata
Yıldız M. 2011 [19]	EOG,ekran	-	%20 hata

3. MATERYAL ve METOD

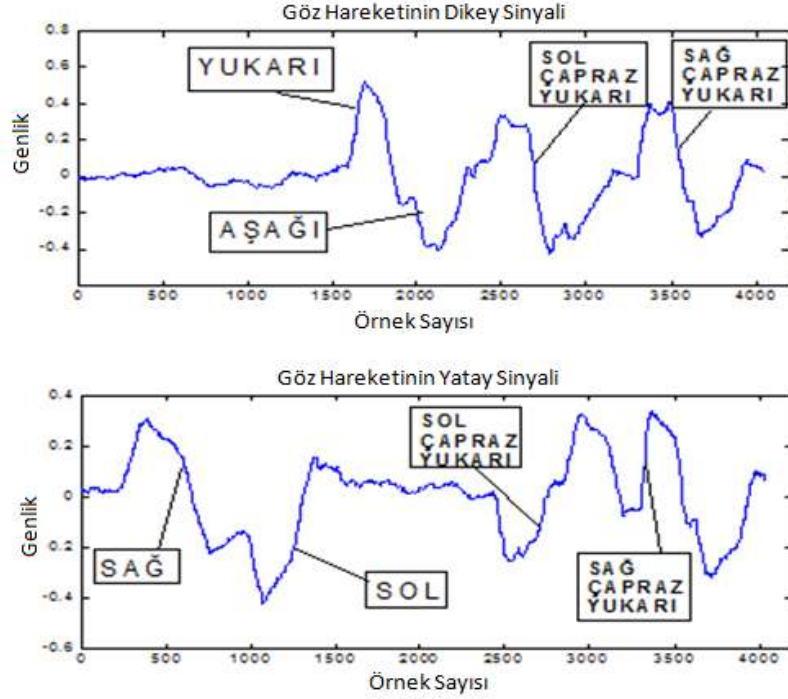
Bu çalışmanın ana amacı daha önce geliştirilmiş EOG tabanlı gözle yazı yazma tetkiklerine göre daha hızlı ve daha az hata ile yazı yazılmasını sağlayabilecek bir yöntem önermek ve bunu uygulamaktır.

Bu amaçla ilk başta daha önce Yıldız M, 2011 [19] tarafından önerilmiş olan EOG tabanlı gözle yazı yazma yönteminin performans testleri yapılarak, bu yöntemin yazma hızını ve doğruluğunu etkileyen faktörler tespit edilmiştir. Buradan elde edilen bilgiler ışığında bu çalışmaya özel yeni bir yöntem geliştirilmiştir.

3.1 Daha Önce Geliştirilmiş EOG Tabanlı Gözle Yazı Yazma Sisteminin İncelenmesi

Yıldız M.'nin çalışmasında önerilen yöntemde, harf, rakam ve bazı noktalama işaretlerinin gözler ile kodlaması sırasında yatay ve düşey düzlemdeki EOG sinyallerinde oluşan değişimlerden yararlanılmaktadır [19].

Yöntem düşey, yatay ve çapraz yöndeki göz hareketleri sırasında düşey ve yatay düzlemde kaydedilen EOG sinyallerinde birbirinden kolayca ayırt edilebilecek 6 farklı EOG örüntüsünün, yazı karakterlerinin kodlanmasında kullanılmasına dayanmaktadır. Gözlerin sağa, sola, yukarı, aşağı, yukarı sağ çapraz ve yukarı sol çapraz olmak üzere altı farklı göz hareketi sırasında kaydedilen EOG sinyalleri ve bunların çok kolay bir şekilde birbirinden ayrılabilceği Şekil 3.1'de görülmektedir. Şekil 3.1 incelendiğinde, karşıya bakılırken yani göz dinlenme durumundayken yatay ve düşey düzlemdeki EOG kayıtlarının 0 mV civarında dar bir aralıkta dalgalandığı görülmektedir. Sağa, sola bakışlar yatay düzlemde, yukarı aşağı bakışlar düşey düzlemde, sağ ve sol çapraz bakışlar ise yatay ve düşey düzlemdeki EOG sinyallerinin her ikisinde birden birbirinden ayrılabilcek pozitif ve negatif genlikli değişimler göstermektedir.



Şekil 3.1 Altı farklı göz hareketi sırasında kaydedilen EOG sinyalleri

Göz hareketleri sırasında ortaya çıkan EOG işaretlerini algılamak için küçük çaplı yüzey elektrotları kullanılmıştır. Düşey göz hareketleri sırasında ortaya çıkan sinyallerin algılanması için aktif elektrotlar sağ gözün altı ve üstüne, referans elektrotu sol gözün yanına, yatay göz hareketleri sırasında ortaya çıkan sinyallerin algılanması için aktif elektrotlar sağ ve sol gözün yanlarına, referans elektrotu alın bölgesine bağlanmıştır. Şekil 3.2’de elektrotların bağlanma şekli görülmektedir.



Şekil 3.2 Yatay ve düşey göz hareketlerinin algılanması için yapılan elektrot bağlantı şekli [36]

Türkçede en çok kullanılan 5 harf tek yöne bakışla kodlanmıştır. Kelimeler arasındaki boşlukların, en çok kullanılan karakterlerden biri olacağı düşünülerek, yukarı sağ çapraz yönündeki tek bakış, kelimeler arası boşluk ya da “-” olarak kodlanmıştır. Alfabadeki geri kalan harfler, rakamlar ve bazı noktalama

işaretlerinin kodlanabilmesi için 8 farklı yöndeki göz hareketinin ikili kombinasyonları (iki farklı yöne bakışın birbiri peşi sıra gerçekleştirilmesi ile) kullanılarak bir alfabe geliştirilmiştir. Ardı ardına gerçekleştirilen iki bakışla ise (2^8) 256 tane yazı karakteri, rakam, noktalama işareti vb. kodlanabilir. Böylece bir karakteri kodlamak için en az 1 en fazla 2 göz hareketi yapılarak bir metin gözle yazılırken öncekilere göre daha az sayıda göz hareketinin yapılması sağlanmıştır.

İki harfin birbirinden ayrılması bir yöndeki bakıştan sonra tekrar karşıya bakılması sonucu oluşan sinyaller ile gerçekleştirilmektedir. İki hareketten oluşan bir karakter kodlanırken; ilk bakıştan sonra, karakterler arasındaki ayrımı sağlamak için belirlenen karşıya bakma süresinden daha kısa bir süre karşıya bakılıp ikinci yöndeki bakışın yapılması gerekmektedir.

Göz hareketlerinin tespiti için gerekli EOG kayıtları; BIOPAC MP 36 fizyolojik veri toplama sistemi vasıtası ile alınmıştır. Örneklem frekansı 200 Hz seçilerek, 12 bit çözünürlükle sayısala çevrilen veriler, MATLAB' ta okunmak üzere "txt" formatında kaydedilmiştir. Göz hareketlerinin tanınip karaktere çevrilmesi ile ilgili çalışmalar çevrimdışı olarak gerçekleştirilmiştir. Bunun için Başkent Üniversitesi Klinik Araştırmalar Etik Kurulundan (Karar Sayısı:12/113) izin alınmıştır. Denekler deney hakkında bilgilendirilip, gönüllü olanlar çalışmaya dahil edilmiştir. Deneye yaşları 19-35 arasında değişen 5 adet bayan, 5 adet erkek katılmıştır.

Yöntemin performansının belirlenmesinde gözle kodlanan belli sayıdaki karakterin bilgisayar yazılımı vasıtasıyla tanınma yüzdesi ve dakikada kaç karakter veya kelime yazılabildiği dikkate alınmıştır

Gerçekleştirilen ilk denemede 10 denekten, sağ, yukarı, sağ çapraz aşağı, sağ-sağ, sol-sağ, sol aşağı çapraz-yukarı olmak üzere 6 farklı karakteri onar tekrarlı olarak kodlaması istenmiştir. Yapılan deneyin sonuçları sonuçlar kısmında detaylı şekilde açıklanmıştır. Bu sonuçlara göre yeni bir gözle yazma prosedürü önerilmiş ve yapılan deney sonuçlarına göre en hızlı olan yöne ve en kısa bekleme süresine, en çok kullanılan karakter gelecek şekilde bir alfabe hazırlanmıştır. Çizelge 3.1'de görülen bu alfabedeki parantez içindeki değerler o karakterlerin kullanım yüzdelerini vermektedir. Bu yüzdeler Türkçede harflerin kullanım yüzdelerine ait çalışmadan yararlanıp, literatürde ortalama bir kelimenin beş karakter olduğu bilgisinden yola çıkarak hesaplanmıştır [8]. Her kelimedenden yani

ortalama her beş karakterden sonra bir boşluk koyulması gerekeceğinden, ortalama her 6 karakterde bir boşluk koyulacaktır. Bu hesaba göre %16,66 oranında boşluk karakteri kullanılacaktır. Diğer karakterlerin kullanım yüzdeleri bu durum dikkate alınarak yeniden hesaplanmış çizelgeye bu yüzdeler yazılmıştır.

3.2 Önerilen İlk Gözle Yazma Prosedürü

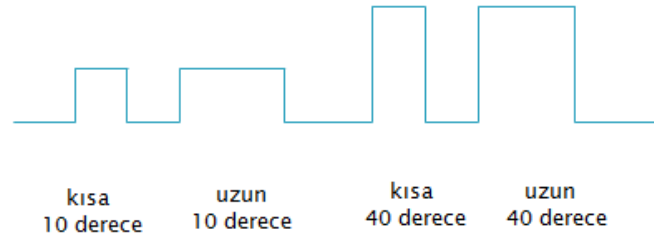
Denemelerden elde edilen sonuçlara göre özetle; tekrar sayısı arttığında beklendiği şekilde hızın da arttığı ancak hızlı kodlama yapılırken doğru tanınma yüzdesinin azaldığı görülmüştür. Doğru tanınma yüzdesindeki azalmanın özellikle ikili karakterlerin doğru kodlanamamasından kaynaklandığı, ayrıca çapraz göz hareketlerinin tanınma yüzdesinin düşüklüğü, yeni gözle yazma prosedürü önerilirken dikkate alınan en önemli unsurlar olmuştur.

Alternatif yöntem önerirken ilk önce; R.Barea ve ark. çalışmalarından esinlenerek bir yönde 4 farklı açıyla bakmanın kullanılabilme olanağı araştırılmıştır [15]. Bu amaçla yapılan deneyde, 10 denek 8 farklı yönde fakat 4 farklı açı ile bakmıştır. Bu deneyin sonuçları da ayrıntılı olarak sonuçlar bölümünde verilmiştir. Bu deney sonucundaki en önemli gözlem, bir yönde 4 farklı açı ile bakmanın hem uygulanmasının, hem de tanınmasının neredeyse imkansız olduğudur. Aynı deney sonucuna göre; bir yöndeki 2 farklı açıyla bakışın ise kolayca ayrılabilceği görülmüş ve nihai öneride bu bilgi kullanılmıştır.

3.3 Önerilen Yeni Gözle Yazma Prosedürü

Gerçekleştirilen deneyler sonucunda çapraz yönlerdeki bakışların tanınma yüzdesini çok düşürdüğü dikkate alınarak sadece 4 yönde bakışın kullanılmasına karar verilmiştir. Yine yapılan deneylerle 4 farklı açıyla bakışın tanınmasının zor olacağı ve hata yüzdesini arttıracacağı değerlendirilerek, yeni kodlama sisteminde sadece 2 açıyla bakışın kullanılmasına karar verilmiştir.

Bu durumda 8 karakterin sadece tek yöndeki bir bakışla kodlanması mümkün olmuştur. Bu yöntem sonucu tek bir yöndeki bakışlarla oluşması beklenen EOG sinyalleri Şekil 3.3 'te gösterilmektedir.



Şekil 3.3 Bir yönde farklı bakma açıları ve bekleme sürelerinde oluşacak EOG sinyalleri

Yeni yöntemde birbiri peşi sıra bakışlarla harf kodlamanın, hem hızı düşürmesi hem de kullanıcıların 2 yöndeki bakış arasındaki kısa bekleme gerçekleştiremeyerek doğru kodlama yapamadığı değerlendirilerek, diğer karakterlerin bir yöndeki bakıştan sonraki bekleme süresi ile kodlanmasına karar verilmiştir. Bekleme süresine göre harf kodlanmasında bir yöndeki bakıştan sonraki her 400 ms'lik bekleme başka bir karaktere karşılık gelmektedir.

Böylece 4 yön, 2 farklı bakma açısı ve 4 bekleme süresi ile 29 karakterin, kelimeler arası boşluğun, son yazılan karakteri silme komutunun ifade edilebileceği Çizelge 3.2'de gösterilen kodlama tekniği oluşturulmuştur.

Önerilen yöntem farklı bekleme sürelerine farklı karakter atanması şeklinde çalıştığından kişilerin bu bekleme sürelerine uymalarını sağlayacak bir yöntem önerilmesine ihtiyaç duyulmuştur. Bunun için daha önceki çalışmalarla insanın duyduğu sese tepki verme süresinin 200 ms civarında olduğu tespit edilmiş olup, bu zamana göre belirlenen sesli uyarı desteği kullanılmıştır. Geliştirilen yöntemle belirlenen 4 yönden birine belirli bir açıyla bakılır bakılmaz o yöndeki ilk harf hoparlörden duyulmakta, buna göre 400 ms içerisinde kişi tekrar karşıya bakarsa ilk sıradaki harfin yazılmak istendiğine karar verilmektedir.

Çizelge 3.1 Türkçede en çok kullanılan harflere göre oluşturulan alfabe

HARFLER	SEMBOL	HARFLER	SEMBOL	HARFLER	SEMBOL
BOŞLUK(16,66)	SAĞ →	B(2,45)	SAĞ - AŞAĞI → ↓	F(0,36)	SAĞ - SAĞ YUKARI → ↗
A(9,73)	SOL ←	S(2,45)	SOL - SAĞ ← →	J(0,008)	SAĞ - SOL AŞAĞI → ↘
E(7,5)	YUKARI ↑	O(2,04)	SOL - YUKARI ← ↑	0	SAĞ - SOL YUKARI → ↖
I(6,89)	AŞAĞI ↓	Ü(1,65)	SOL - AŞAĞI ← ↓	1	SAĞ - SOL AŞAĞI → ↘
N(6,02)	SAĞ YUKARI ÇAPRAZ ↗	Ş(1,62)	YUKARI - SAĞ ↑ →	2	SOL - SAĞ YUKARI ← ↗
R(5,81)	SOL AŞAĞI ÇAPRAZ ↘	Z(1,25)	YUKARI - SOL ↑ ←	3	SOL - SOL AŞAĞI ← ↘
L(4,79)	SOL YUKARI ÇAPRAZ ↖	G(1,12)	YUKARI - AŞAĞI ↑ ↓	4	SOL - SOL YUKARI ← ↖
İ(4,33)	SAĞ AŞAĞI ÇAPRAZ ↘	Ç(1,05)	AŞAĞI - SAĞ ↓ →	5	SOL - SOL AŞAĞI ← ↘
D(4,05)	SAĞ – SAĞ → →	H(0,95)	AŞAĞI – SOL ↓ ←	6	YUKARI - SAĞ YUKARI ↑ ↗
K(3,92)	SOL – SOL ← ←	Ğ(0,94)	AŞAĞI - YUKARI ↓ ↑	7	YUKARI - SOL AŞAĞI ↑ ↘
M(3,11)	YUKARI – YUKARI ↑ ↑	V(0,81)	SAĞ YUKARI- SAĞ YUKARI ↗ ↗	8	YUKARI - SOL YUKARI ↑ ↖
U(2,85)	AŞAĞI – AŞAĞI ↓ ↓	C(0,8)	SOL AŞAĞI - SOL AŞAĞI ↘ ↘	9	YUKARI - SOL AŞAĞI ↑ ↘
Y(2,8)	SAĞ – SOL → ←	Ö(0,72)	SOL YUKARI - SOL YUKARI ↖ ↖	.	AŞAĞI - SAĞ ↓ →
T(2,57)	SAĞ – YUKARI → ↑	P(0,65)	SOL AŞAĞI - SOL AŞAĞI ↘ ↘	?	AŞAĞI -SOL ↓ ←

Eğer bakılan yönde 400 ms'den fazla beklenirse o yön ve açıdaki 2. harf seslendirilmekte, onaylamak ve ortaya dönüş için kullanıcıya 400 ms'lik bir süre verilmektedir. Bu o yöndeki 3. ve 4. harf içinde geçerlidir. Yöntemle bu şekilde 32 karakter sadece tek bir yöndeki bakış ve karşıya bakışla kodlanmaktadır.

Yanlış yazılan karakterin silinmesi gerekebileceği ve bu işlevin çok kullanılmayacağı düşünülerek sol yöndeki uzak bakışın 3. bekleme süresine silme komutu atanmıştır. Boşluk karakteri ise ortalama her 6 karakterde bir kullanılacağından sağ yöndeki uzak bakışın 1. bekleme süresi olan kısmına atanmıştır.

Çizelge 3.3'te Türkçede en çok kullanılan harflerin en kısa sürede kodlanacak yerlere yerleştirilmesi ile elde edilen alfabe ve bu alfabe ile yazı yazılmasından elde edilebilecek en yüksek hızın kestirimi için hesaplamalar gösterilmiştir. Hesaplamalar yapılırken en kolay kodlanan karakterlerin 400 ms'de kodlanabileceği, daha sonraki karakterleri kodlamak içinde 400 ms'lik artışlar gerektiği kabul edilmiştir.

Bu sonuçlara göre bir karakterin ortalama 517,6 ms'de kodlanmasının mümkün olduğu, ve güzün bir sıçramasının 150 ms aldığı dikkate alınarak; yöntemle dakikada en fazla 75 karakterin, yani 15 kelimenin kodlanmasının mümkün olabileceği hesaplanmıştır. Bu da literatürdeki en hızlı yazı yazmaya olanak veren yöntemden %50 daha hızlı bir gözle yazma performansı anlamına gelir. Bu sonuçların motivasyonunda, önerilen yeni yöntemi gerçekleyecek aşağıdaki gerçek zamanlı gözle yazı yöntemi geliştirilmiştir

3.4 Gerçek Zamanlı Gözle Yazı Yazma Sistemi Geliştirilmesi

Önerilen yeni gözle yazı yazma yönteminin uygulanması için bilgisayar tabanlı bir sistem geliştirilmiştir. Geliştirilen sistem donanım ve yazılım olmak üzere iki kısımdan oluşmaktadır. Donanım kısmı yatay ve düşey göz hareketleri sırasında ortaya çıkan EOG sinyallerini alıp işleyerek bilgisayara atarken, yazılım kısmı bu sinyalleri anlamlandırıp seslendiren ve yazıya çeviren programları içermektedir.

Çizelge 3.2 Önerilen yeni gözle kodlama yöntemi

Harfler/Süreleri	Sembol	Harfler/Süreleri	Sembol
Boşluk(16,66) 400 ms	Sağ Uzak Bakış →	S (2,45) 800 ms	Sol Uzak Bakış
A (9,73) 400 ms	Sağ Yakın Bakış →	O (2,04) 1200 ms	Sağ Yakın Bakış ↔
E (7,5) 400 ms	Sol Yakın Bakış ←	Ü (1,65) 1200 ms	Yukarı Uzak Bakış ↑
I (6,89) 400 ms	Yukarı Yakın Bakış ↑	Ş (1,62) 1200 ms	Yukarı Yakın Bakış ↑
N (6,02) 400 ms	Aşağı Yakın Bakış ↓	Z (1,25) 1200 ms	Aşağı Yakın Bakış ↓
R (5,81) 400 ms	Sol Uzak Bakış ←	G (1,12) 1200 ms	Sağ Uzak Bakış →
L (4,79) 400 ms	Yukarı Uzak Bakış ↑	Ç (1,05) 1200 ms	Yukarı Uzak Bakış ↑
İ (4,33) 400 ms	Aşağı Uzak Bakış ↓	H (0,95) 1200 ms	Aşağı Uzak Bakış ↓
D (4,05) 800 ms	Sol Yakın Bakış ←	Ğ (0,94) 1600 ms	Sol Yakın Bakış ←
K (3,92) 800 ms	Sağ Yakın Bakış →	V (0,81) 1600 ms	Sağ Yakın Bakış →
M (3,11) 800 ms	Yukarı Yakın Bakış ↑	C (0,8) 1600 ms	Aşağı Yakın Bakış ↓
U (2,85) 800 ms	Sol Yakın Bakış ←	Ö (0,72) 1600 ms	Sağ Uzak Bakış →
Y (2,8) 800 ms	Aşağı Yakın Bakış ↓	P (0,65) 1600 ms	Yukarı Yakın Bakış ↑
T (2,57) 800 ms	Sağ Uzak Bakış →	F (0,36) 1600 ms	Yukarı Uzak Bakış ↑
B (2,45) 800 ms	Aşağı Uzak Bakış ↓	J (0,008) 1600 ms	Aşağı Uzak Bakış ↓

Çizelge 3.3 Sistemden beklenen maksimum hız tespiti

Harf	Kullanım Yüzdesi	Kodlama süresi (ms)	Sonuç
A	0,118	400	94,4
E	0,091	400	36,4
İ	0,085	400	34
N	0,073	400	29,2
R	0,071	400	28,4
L	0,064	400	25,6
I	0,050	400	20
K	0,048	800	38,4
D	0,043	800	34,4
M	0,037	800	29,6
Y	0,035	800	28
T	0,034	800	27,2
U	0,031	800	24,8
S	0,031	800	24,8
B	0,026	800	20,8
O	0,025	1200	30
Ü	0,020	1200	24
Ş	0,017	1200	20,4
Z	0,015	1200	18
G	0,013	1200	15,6
Ç	0,011	1200	13,2
H	0,011	1200	13,2
Ğ	0,010	1600	16
V	0,010	1600	16
C	0,009	1600	14,4
P	0,009	1600	14,4
Ö	0,008	1600	12,8
F	0,004	1600	6,4
J	0,001	1600	1,6
		Ortalama tek karakter kodlama süresi (ms)	517,6

3.4.1 Donanım kısmı

Bu bölümde, yeni gözle kodlama prosedürü için geliştirilen çift kanallı EOG ölçüm sisteminin tasarlanma aşaması ve bu aşamada yapılan çalışmalar anlatılmaktadır. Gerçekleştirilen iki kanallı EOG algılama ve bilgisayara aktarma donanımının blok diyagramı Şekil 3.4' te gösterilmiştir.

EOG sinyallerinin ölçülmesine yönelik bir sistemin tasarımında sinyallerin doğru bir şekilde kaydedilmesi için yükseltecin yeterince yüksek ortak mod sinyalini bastırma oranına (CMRR) sahip olması, elektronik gürültünün mümkün olduğunca azaltılması, şebeke gürültüsünden ve besleme gerilimi dalgalanmalarından etkilenmemesi ve en önemlisi hastanın elektriksel güvenliğinin sağlanması gerekmektedir. Şekil 3.5'de tüm devrenin elektronik şeması görülmektedir. Devreyi oluşturan katlar aşağıda ayrı ayrı açıklanmıştır.

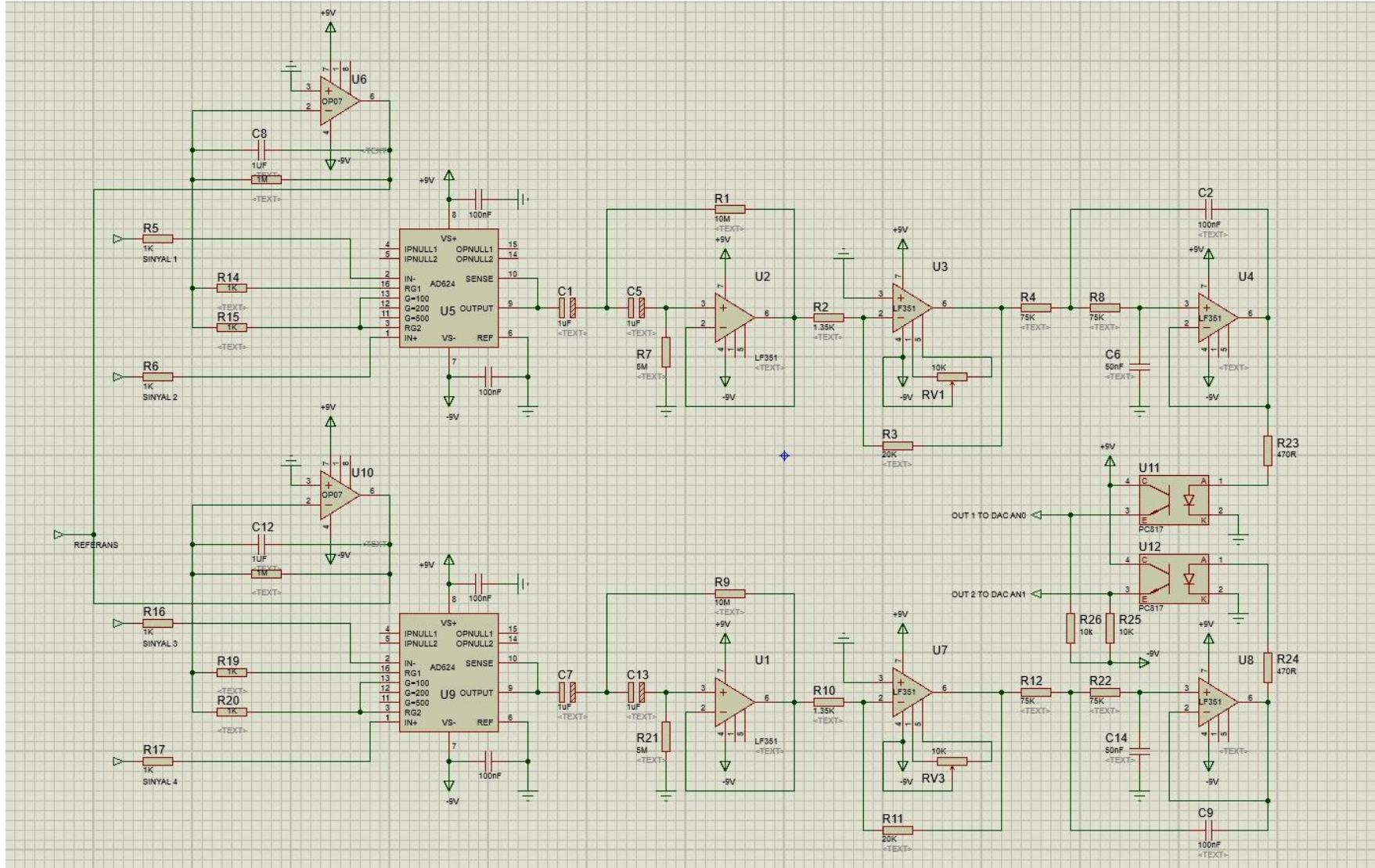
Tasarlanan sistem, yatay ve düşey göz hareketlerini algılayacak şekilde iki kanallı olarak tasarlanmıştır. İki adet 9V'luk pille beslenen devrede, EOG sinyallerinin alınması için küçük çaplı tek kullanımlık gümüş/gümüş klorür (Ag/AgCl) elektrotlar kullanılmıştır. Gerçekleştirilen sistemde yatay (ChH+, ChH-) ve düşey (ChV+, ChV-) göz hareketlerinin algılanmasını sağlayan elektrotlar ve iki adet toprak (GND) elektrodu olmak üzere toplam altı adet elektrot kullanılmıştır.

3.4.1.1 Toprak sürücü devresi

Temelde şebeke kaynaklı işaretlerin insan vücudu aracılığı ile toprağa akması sonucu meydana gelen şebeke gürültülerini engellemek maksadıyla kullanılan devreye "toprak sürücüsü" denilmektedir. Toprak sürücü devresinde vücut üzerindeki ortak mod gerilimi kuvvetlendirildikten sonra hastaya geri beslenir. Bu işlem CMRR'nin de yüksek bir değere sürülmesini sağlar [11].

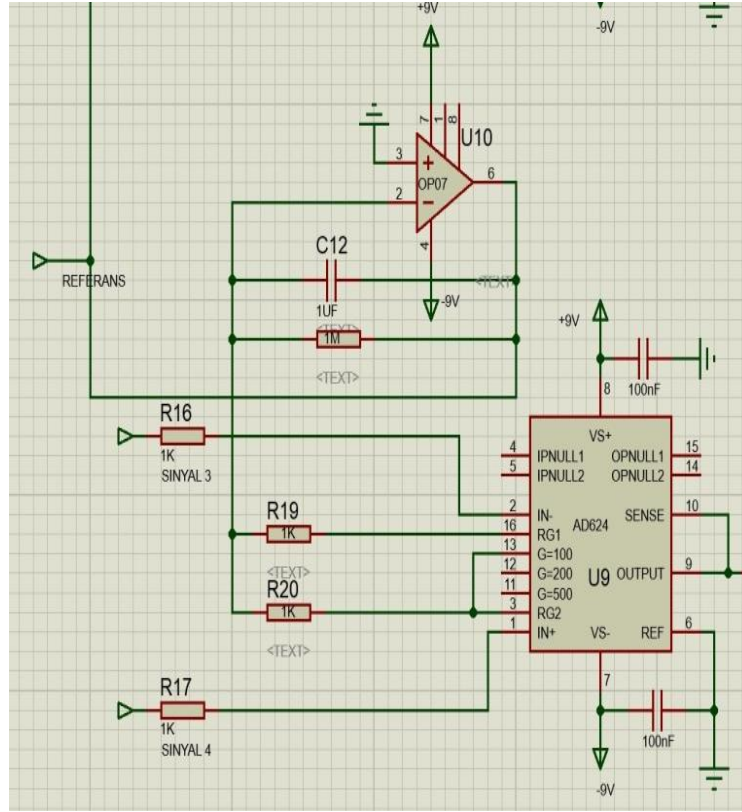


Şekil 3.4 Donanım blok diyagramı



Şekil 3.5 Devre şeması

Şekil 3.6'da toprak sürücüsü devresi görülmektedir. OP07 entegresinin altında bir adet 1 μ F kapasitör ve 1 adet 1 M Ω değerinde direnç vardır. Hastaya uygulanması olası voltajlar çok yüksek direnç üzerinden geldiğinden, çok küçük bir akımın hasta üzerinden geçmesi çarpılmalara karşı hasta güvenliğinin de sağlanmasına yardımcı olmaktadır.



Şekil 3.6 Toprak sürücü devre kısmı

3.4.1.2 Enstrümantasyon yükselteci

Elektrotların vücutla teması sonucu ortaya çıkan yarı hücre potansiyelini minimize etmek için çok yüksek CMRR oranına sahip AD624 enstrümantasyon yükselteci kullanılmıştır. AD624'ün tercih edilme sebebi CMRR'sinin 130 dB olması ve kazanç ayarlarının kendi içindeki hassas kalibre edilmiş direnç tarafından sağlanmasıdır. Enstrümantasyon yükselteci için 100 kazanç değeri, 3 ile 13'üncü uçlar arasında kısa devre yapılarak sağlanmıştır.

3.4.1.3 İkinci derece butterworth yüksek ve alçak geçiren filtre devresi

EOG sinyalleri 0,05 Hz ile 30 Hz frekans aralığındadır [37]. Bu sebepten oluşturulacak devrenin bu frekans aralıklarına yakın değerleri geçirmesi gerekmektedir. Enstrümantasyon yükseltecinin arkasına ikinci derece butterworth yüksek geçiren ve ikinci derece butterworth alçak geçiren filtreler eklenmiştir. 2. derece filtre kullanılması ile istenmeyen frekanslar için 40 dB'lik bir zayıflama sağlanmıştır. Direnç ve kondansatör değerleri 0,02 Hz'in üzerini ve 30 Hz'in aşağısını geçirecek şekilde seçilmiştir. Yüksek geçiren filtrenin kesim frekansının 0,02 Hz seçilmesinin nedeni, elektrotlar yüze takıldıktan sonra kararlılığa ulaşana kadar kısa süre beklenmesini sağlamaktır. Şekil 3.7'de görüldüğü üzere C1,C5 kapasitörleri ve R1,R7 dirençleri kullanılarak oluşturulan ikinci derece yüksek geçiren filtre kullanılmaktadır.

Yüksek geçiren filtrenin kesim frekansı;

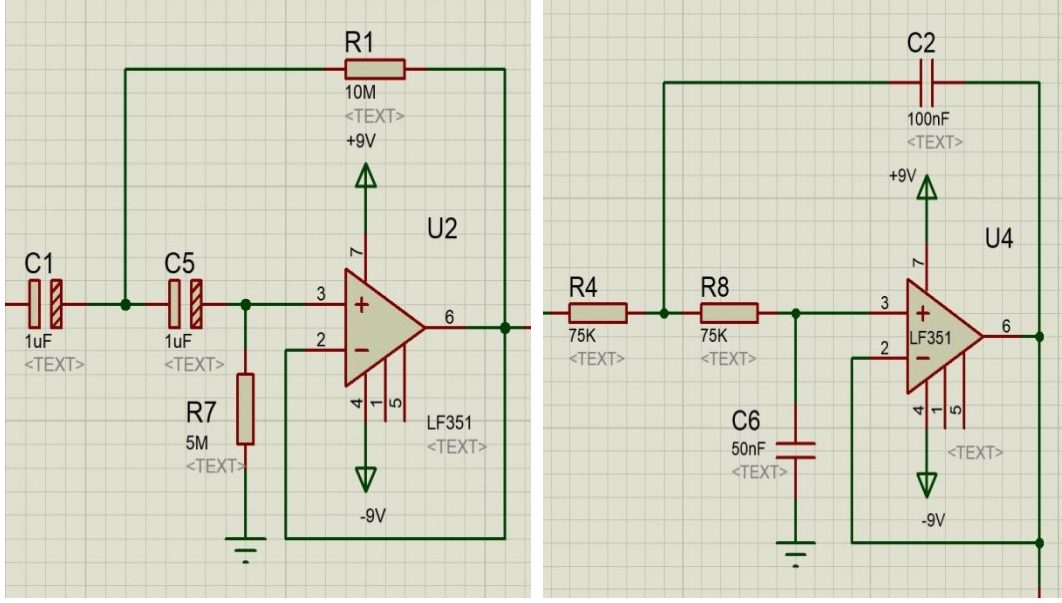
$$f = \frac{1}{2\pi\sqrt{R1 R7 C1 C5}} = \frac{1}{2\pi\sqrt{10 \times 10^6 \cdot 5 \times 10^6 \cdot 1 \times 10^{-6} \cdot 1 \times 10^{-6}}} \cong 0,02Hz \quad (3.1)$$

Şekil 3.7'de yer alan R4,R8 dirençleri ve C2,C6 kapasitörleri ikinci dereceden alçak geçiren filtre görevi görmektedir.

Alçak geçiren filtrenin kesim frekansı;

$$f = \frac{1}{2\pi\sqrt{R4 R8 C2 C6}} = \frac{1}{2\pi\sqrt{75 \times 10^3 \cdot 75 \times 10^3 \cdot 50 \times 10^{-9} \cdot 100 \times 10^{-9}}} \cong 30Hz \quad (3.2)$$

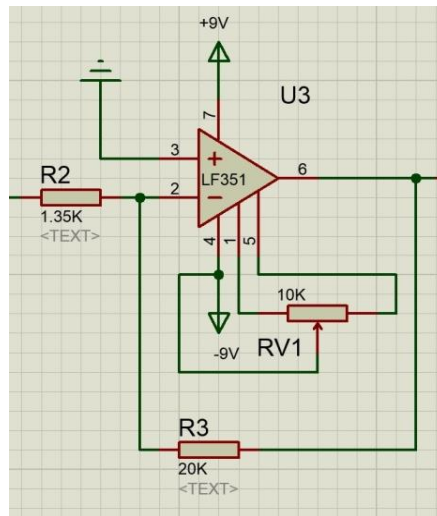
formülleriyle tespit edilmektedir.



Şekil 3.7 İkinci derece butterworth yüksek ve alçak geçiren filtre

3.4.1.4 Kazanç yükseltici devresi

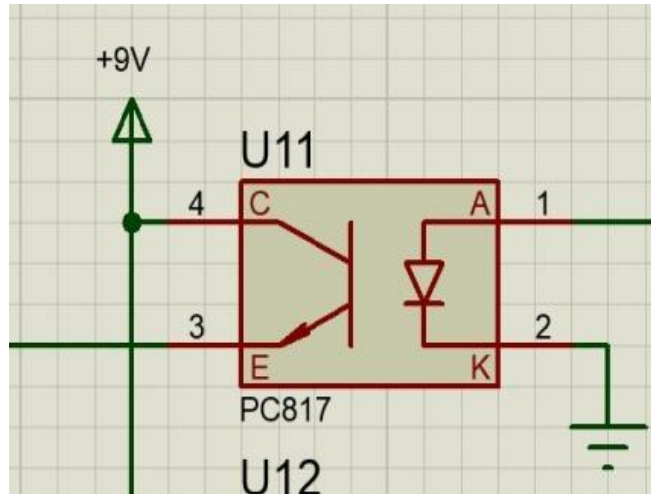
Enstrümantasyon yükseltici çıkışında 100mV'lar mertebesinde olan EOG sinyallerinin voltlar mertebesine çıkarılması için x15'lik kazanç, bu kazanç yükseltici devre kısmıyla sağlanmıştır. Bu durum mV'lar mertebesinde düşürülmüş olan DC bileşenlerinde yükseltilmesine yol açmıştır. Sinyal tabanındaki kaymayı önlemek dinlenme durumundaki EOG seviyesini 0 V seviyesine taşımak için devreye 10 k'lık ofset ayar potansiyometresi eklenmiştir. Şekil 3.8'de kazanç yükseltici devresi görülmektedir.



Şekil 3.8 Kazanç yükseltici devresi

3.4.1.5 Optik aktarıcı devresi

Bilgisayarda oluşabilecek herhangi bir arıza durumunda hasta üzerinde oluşabilecek kaçak akımları engellemek ve kullanıcıyı korumak için optik aktarıcılar kullanılmıştır. Tipik bir verici-alıcı ikilisi gibi art arda bir bağlantı olmaksızın bir tarafta ışık yayan diyot, diğer tarafta bu ışığa duyarlı bir anahtarlama elemanı ile sinyal aktarımı ışık ile gerçekleştirilir. Şekil 3.9'da kullanılan optik aktarıcı devre şeması görülmektedir.



Şekil 3.9 Optik aktarıcı devre şeması

3.4.1.6 Verilerin bilgisayara aktarılması

Donanım kısmından alınan sinyaller bilgisayara aktarılırken National Instruments firmasının üretmiş olduğu USB-NI6009 model veri toplama kartı kullanılmıştır. Bilgisayara USB girişinden bağlanan NI USB6009 Şekil 3.10'da görülmektedir. NI6009'un 8 adet analog girişi ve 2 adet analog çıkışı bulunmaktadır. NI6009 analog sayısal dönüştürücüsünün çözünürlüğü 14 bittir. Ayrıca örnekleme hızı 48kS/s'dir [20]. Yatay ve düşey EOG sinyallerinin bilgisayara aktarılması için USB-NI6009 modelinin iki kanalının kullanımı yeterli olmuştur.



Şekil 3.10 National Instruments marka USB-NI6009 model DAQ kartı

3.5 Gerçek Zamanlı Gözle Yazı Yazma Sistemi Yazılımı

Yazılım kısmı 3 ana bölümden oluşmaktadır. Bunlar veri toplama (DAQ) kartının programlanması, grafik kullanıcı arayüzü ve alınan EOG verilerinin anlamlandırılıp karakterlere çevrilmesi ve seslendirilip ekrana yazılmasını sağlayan bölümlerdir.

Gerçek zamanlı çalışma için programlanan veri toplama kartından gelen bilgilerin alınıp ekrana yazdırılması gerekmektedir. Bu nedenle grafiksel kullanıcı arayüzünün bazı kısımları ve veri toplama kartının programlanması birlikte anlatılmıştır.

Tasarlanan sistemin kullanıcı arayüzünde bulunan grafik pencereleri bir anda 10 sn'lik sinyalleri göstermek üzere ayarlanmış, sinyaller saniyede 10 kez okunarak ekrana aktarılmıştır. Sinyal 2 saniyelik aralıklarla kaydırılıp ekranda yenilenmektedir. DAQ kartı Data Acquisition Toolbox'ında yer alan daqreset, analoginput, addchannel, daqread, daq.createSession vb. komutlar kullanılarak programlanmış ve veri toplanmıştır. Analog input objesi ile sağ-sol ve yukarı-aşağı göz sinyali için 2 adet giriş kanalı eklenmiştir. 0,02 ile 30 Hz frekans bileşenlerine sahip EOG'leri incelemek için örnekleme hızı 100 bit/saniye olarak ayarlanmıştır.

Programda, kartın veri okuma sıklığını belirlemek için analoginput objesinin sample direct count değişkenine 10 örneklik veri koplandığında yani 0,1 saniyede bir karttan veri okunmuştur.

Oluşturulan devre ile alınan yatay ve düşey düzlemdeki göz hareketleri sırasında oluşan EOG sinyallerinden yazılan karakterin tanınması için geliştirilen algoritmanın akış diyagramı Şekil 3.11’de verilmiştir. İlgili algoritma ve sinyal işleme işlemleri MATLAB ortamında uygulanmıştır.

Sistemin algoritması şu şekilde çalışmaktadır; sesle kullanıcıyı destekleme kısmı için alfabedeki tüm harflerin “.wav” uzantısı olacak şekilde değişkenlerinin ataması yapılmıştır. Eğer arayüzdeki başlama butonuna basıldıysa veri toplama kartının ayarlamaları yapılmaktadır. Veri toplama kartının iki adet kanalı giriş olarak saniyede 100 örnek olacak şekilde ayarlanır. Yüze takılan elektrotlar vasıtasıyla gözden EOG sinyalleri alınmaya başlanır. CollectData isimli fonksiyon 20 veride bir kez çağırılır. Ekranın dolması için bu fonksiyonun 50 kez çağırılması gerekmektedir. Ekran dolduktan sonra 10 saniye gelen sinyaller ekranda görülecek şekilde eski veriler hafızadan silinir. Her 10 veride bir sağ-sol veya yukarı-aşağı karakter bulma fonksiyonu çalıştırılır. Göz hareketleriyle oluşan sinyalin yükselen kenar mı alçalan kenar mı olduğuna yani sinyalin pozitif yönde mi yoksa negatif yönde mi genlik ürettiğine bakılır. Darbenin yükselen kenar mı alçalan kenar mı olduğu belirlendikten sonra hareket sinyalinin genliği ve bekleme süresi parametreleri bulunur.

Belirlenen ilk eşik seviyesini geçen sinyal yakın bakmayı ifade etmektedir. Yakına bakan kullanıcı belirlediği yöne bakarken o yön için atanan harfler sırasıyla seslendirilmektedir. Kullanıcının yazmak istediği harf ilk harfse, kullanıcı o harfi duyduğunda ortaya dönmesi beklenmektedir. Eğer kullanıcı ortaya dönmez ise o yönde belirlenen ikinci harfe geçilecektir. Kullanıcının ortaya dönmek isteyipte geri dönemediği durumlar için ise sola bakışın uzak ve uzun süre bakışı sil tuşu olarak tasarlanmıştır. Kullanıcı bu buton sayesinde harfi yanlış yazsa dahi, yazdığı son harfi silip yeni harfi tekrar kodlayabilecektir. Çizelge 3.4’de 4 yön için yakın ve uzak bakış ile 4 farklı bekleme süresine göre hazırlanan harflerin sıralamasına ait tablo görülmektedir.

Çizelge 3.4 Harflerin yerlerinin gösterildiği tablo

Mesafe/Yön	Yakın Bakış	Uzak Bakış
Sağ	A(11,8),K(4,8),O(2,5),V(1,0)	'BOŞLUK',T(3,4),G(1,3),Ö(0,8)
Sol	E(9,1),D(4,3),Ü(2,0),Ğ(1,0)	R(7,1),S(3,1),'SİL',
Yukarı	İ(8,5), M(3,7),Ş(1,7),P(0,9)	L(6,4),U(3,1),Ç(1,1),F(0,4)
Aşağı	N(7,3),Y(3,5),Z(1,5), C(0,9)	I(5,0),B(2,6),H(1,1),J(0,1)

Grafiksel kullanıcı arayüzü, bilgisayarlarda işletilen komutlar ve bunların çıktıları yerine; simgeler, pencereler, düğmeler ve panellerin tümünü ifade etmek için kullanılan genel addır. Grafiksel kullanıcı arayüzü, kullanıcılarının komut satırı kodlarını ezberlemeden fare, klavye gibi araçlar sayesinde bilgisayarları kontrol edebilmesini sağlar. Oluşturulan grafik kullanıcı arayüzü ekranına yerleştirilen her bir buton ve pencere ile ilgili program parçaları kendi kod bölümüne yazılarak sistemin idaresi arayüz üzerinden yapılmıştır.

Geliştirilen grafiksel kullanıcı arayüzü hem Türkçe hem de İngilizce dilleri için hazırlanmıştır. MATLAB kodunu çalıştırdığımızda Şekil 3.12'de görülen 'Dil Seçeneği' sayesinde kullanıcıya yardımcı olacak kişi çalışacağı dili seçecektir. Şekil 3.13'de İngilizce, Şekil 3.14'de ise Türkçe arayüz ekranı görülmektedir.

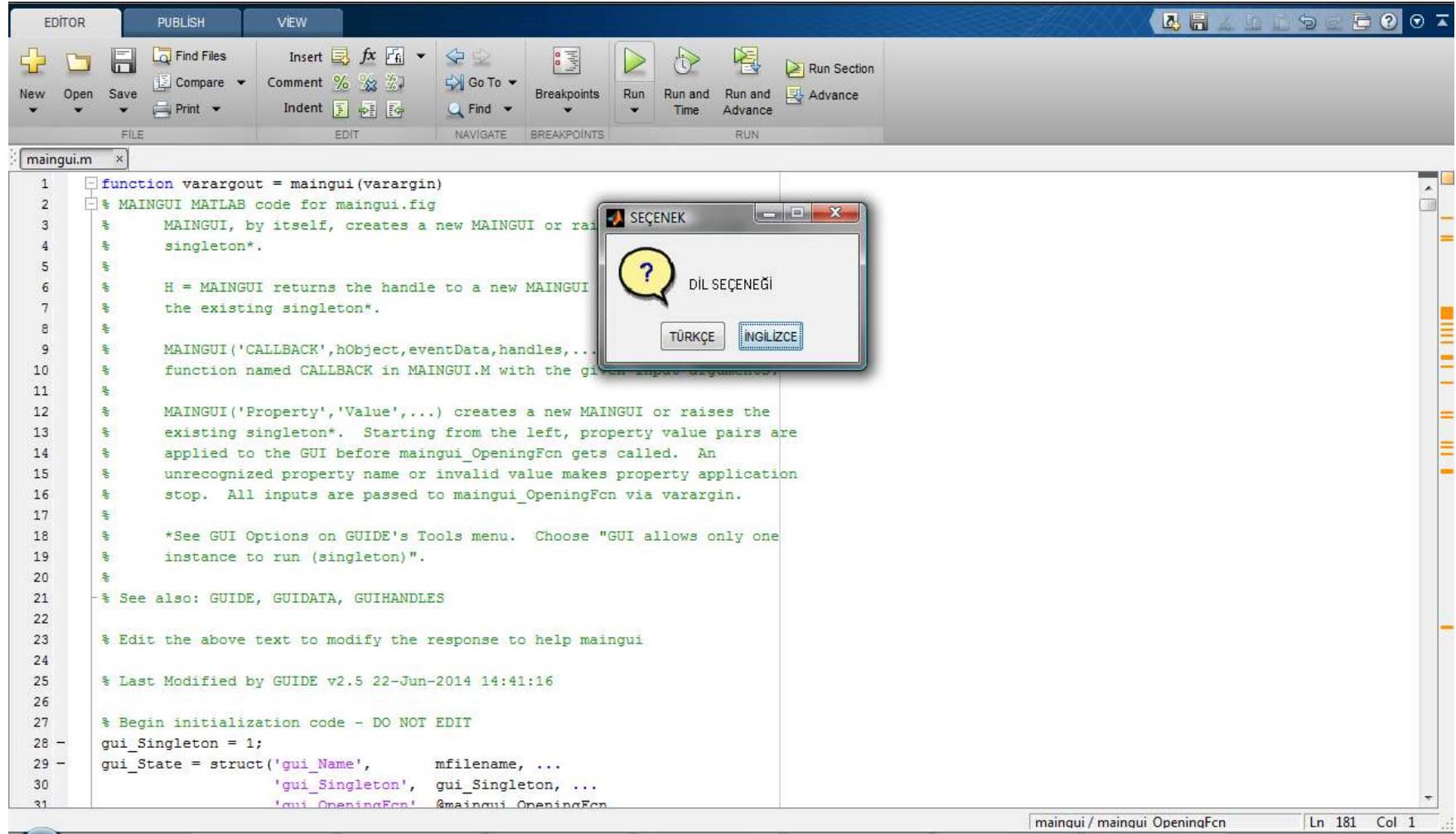
Kontrol panelinde yer alan kısımda bulunan tuşlardan başlatma butonu basıldığında yüze takılan elektrotlar vasıtasıyla gözden sinyaller alınmaya başlanacaktır. Durdurma butonuna basıldığında ise sinyal alma işlemi sonlandırılmaktadır. Hepsini sil butonu, metin yazılması için ayrılan pencereye yazılan harflerin tümünü silme işlemi yapmaktadır. Bu paneldeki son tuş olan GUI kapatma tuşu ise GUI arayüz ekranını kapatmaktadır. Bu tuşlara basma işlemlerini kas hastalıkları bulunan kişiler yapamayacağından, bu aşamalarda ailesinden yardım alması gerekecektir.

Yukarı-Aşağı Hareket ve Sağ-Sol Hareket kutucukları içindeki Genliğin Eşik Değerleri panelleri, yatay ve düşey göz hareketleri anlamlandırılırken kısa mesafeli veya uzun mesafeli bakışların ayırt edilmesi sırasında kullanılacak eşik değerlerinin kullanıcı tarafından girilmesi için sisteme eklenmiştir. Böylelikle farklı miktarda EOG genliği üreten hastalar için sınır değerler seçimlik hale getirilmiştir.

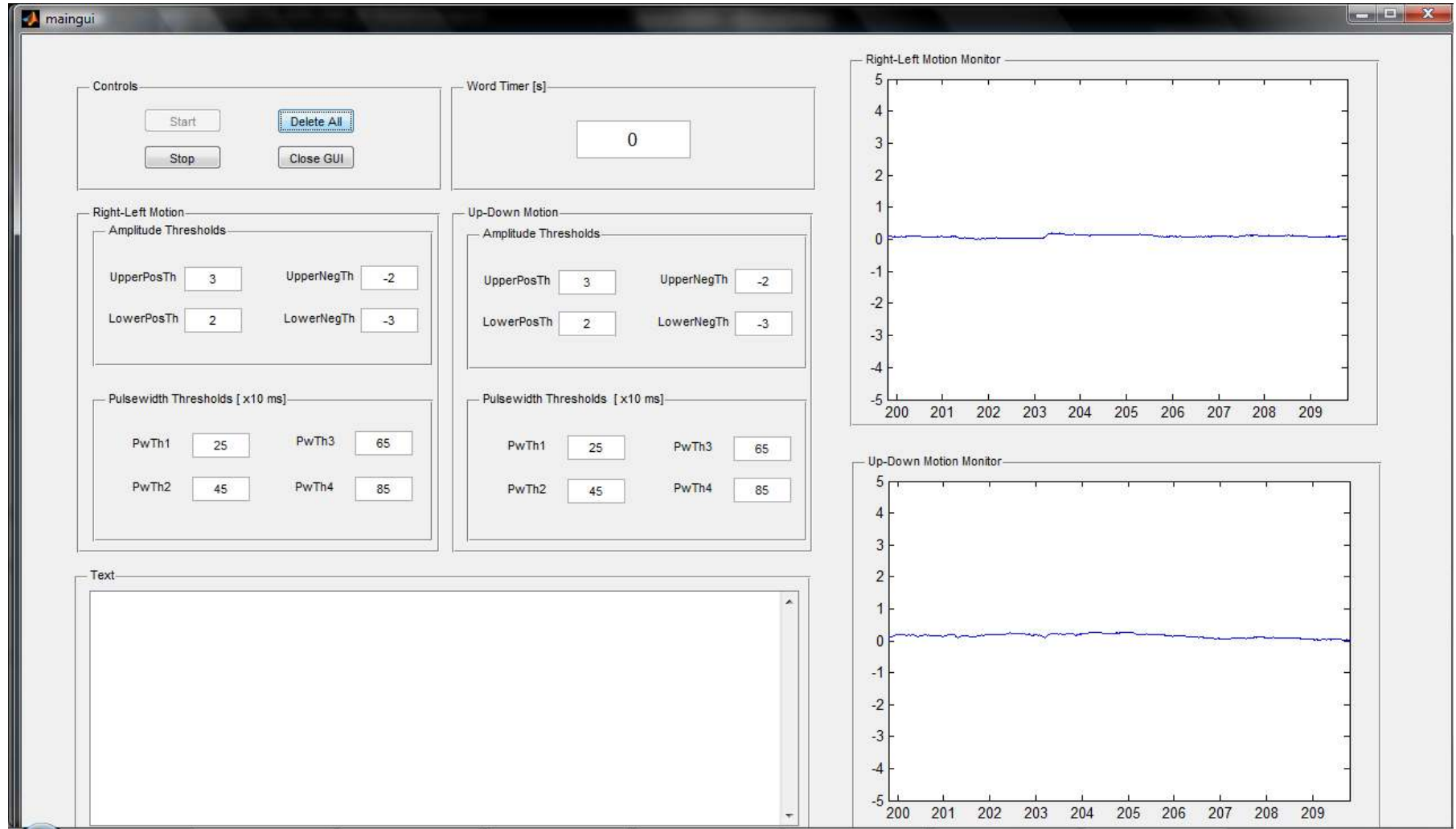
Darbe genişliği eşik değerleri panelinde yer alan DarbeGen 1, DarbeGen 2, DarbeGen 3 ve DarbeGen 4 ise bir yöndeki bakıştan sonra beklemlerle kodlanacak karakterlerin ayrılmasının sağlandığı kısımdır. Kullanıcının başlangıçta harflerin yerini tam öğrenememesinden yazma işlemi yavaş yapacağı, fakat zaman geçtikçe kullanıcının göz hareketleri hızlanacağından sisteme bu darbe genişliğini değiştirebilme opsiyonu eklenmiştir. Bu opsiyon darbe genişliklerinin kullanıcıdan kullanıcıya değişmesi durumunda da işe yarayabilecektir.

Panel üzerinde Sağ-Sol Hareket Monitörü ve Yukarı-Aşağı Hareket Monitörü pencerelerinde alınan yatay ve düşey EOG sinyalleri gösterilerek, yukarıda sayılan ayarların uygun şekilde yapılmasına yardımcı olunması amaçlanmıştır. Kullanıcılara sunulacak son kullanıcı ekranında bu pencereler gösterilmeyebilir.

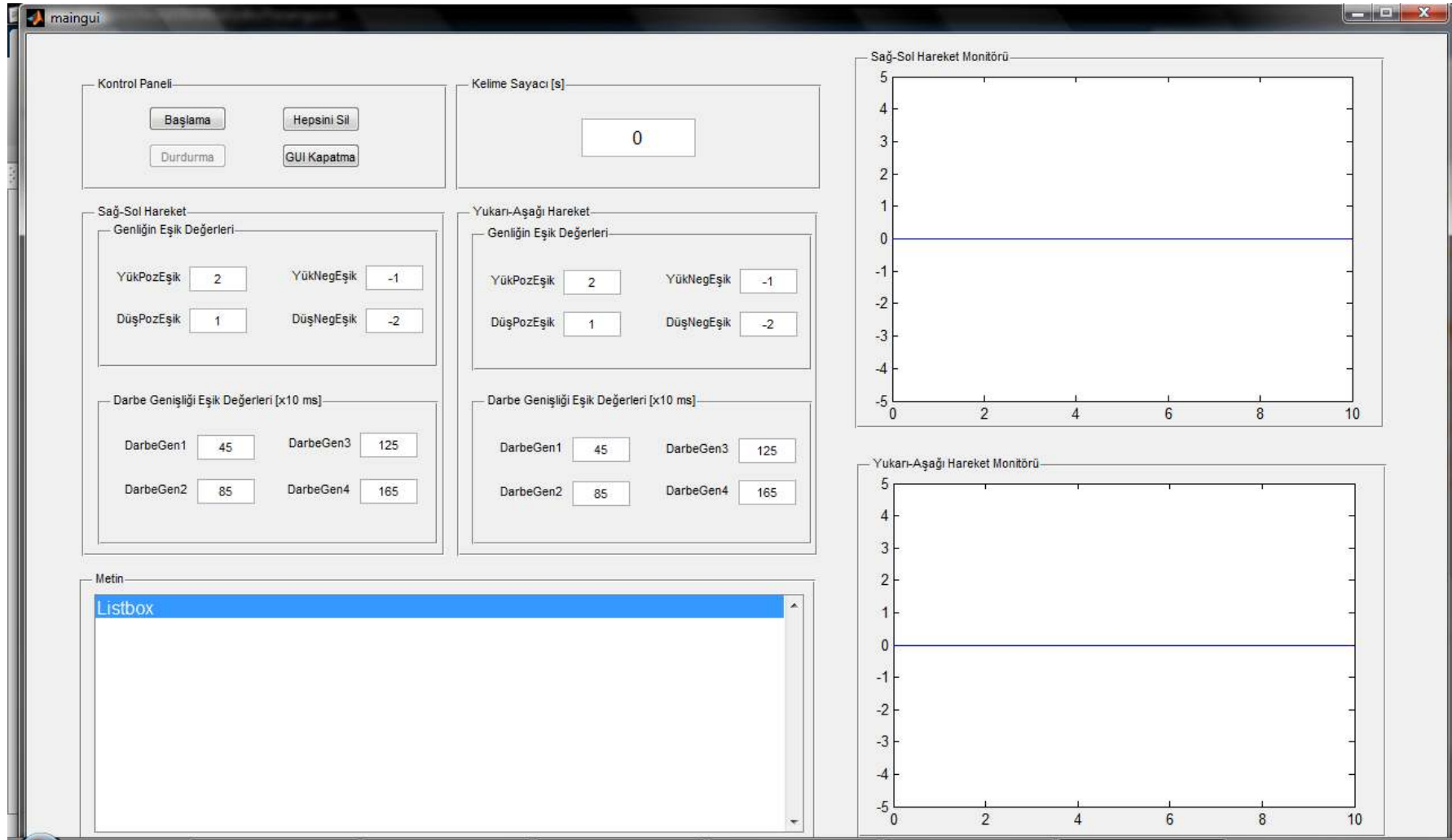
Son panel olan metin penceresinde ise göz hareketlerine göre tespit edilen karakterler yazdırılmaktadır.



Şekil 3.12 Tasarlanan arayüzde dil seçme ekranı



Şekil 3.13 Tasarlanan İngilizce arayüz ekranı



Şekil 3.14 Tasarlanan Türkçe arayüz ekranı

4. SONUÇLAR

Hastaların mümkün olduğu kadar az hata ile ve hızlı şekilde gözle yazı yazmasına yarayacak bir yöntem önerebilmek için, daha önce Yıldız M. 2011 tarafından geliştirilen gözle yazı yazma yöntemi çeşitli durumlar için test edilmiştir.

İlk teste 10 deneğin farklı yöndeki bakışlarla kodlanan 6 karakteri ilk ve onuncu denemede doğru kodlama oranları ve hızlarının değişimi incelenmiştir. Çizelge 4.1’de ilk ve onuncu denemede, ortalama, en az ve en çok tanınma yüzdeleri verilmiştir.

Çizelge 4.1 Yıldız M. çalışmasının performans testi sonuçları

N=10	İlk Deneme		Onuncu Deneme	
Yönler	Hız(sn)	Doğruluk(%)	Hız(sn)	Doğruluk(%)
Sağ →	1,06(0,8-1,2)	92(80-96)	0,69(0,42-0,73)	84(72-88)
Yukarı ↑	1,26(0,88-1,42)	88(76-92)	0,71(0,44-0,84)	80(68-88)
Sağ Çapraz Aşağı ↘	1,29(0,82-1,54)	82(72-88)	0,87(0,58-0,93)	76(68-84)
Sağ-Sağ → →	2,63(2-2,89)	84(72-88)	2,14(1,8-2,28)	72(64-76)
Sol- Sağ ← →	2,83(2,04-2,96)	80(68-84)	2,23(1,82-2,39)	68(64-72)
Sol Aşağı Çapraz- Yukarı ↙ ↑	3,09(2,06-3,32)	74(62-84)	2,20(1,9-2,64)	64(60-72)

Çizelge 4.1 yazma hızı açısından değerlendirildiğinde;

1. Tek yöne bakışla kodlanan karakterlerin çok daha kısa sürede yazılabildiği,
2. Sağ yöne bakışın diğer yönlerle göre en az %8 daha çabuk yapılabildiği,
3. Çapraz yönlerdeki bakışın en yavaş yapılabildiği,
4. Karakterlerin onuncu denemede ilk denemeye göre %30-%45 daha hızlı yazılabildiği, en hızlı yazan kişi açısından değerlendirildiğinde bu değer tek yöndeki bakışla kodlanan karakterler için %50’ye varabildiği,
5. 2 yönde bakışla kodlanan karakterlerin, tek yönde bakışla kodlanarlardan yaklaşık 2 kat daha uzun sürede yapılabildiği görülmüştür.

Çizelge 4.1 karakterlerin doğru kodlanıp, yazılım tarafından tanınabilme doğruluğu yüzdesi açısından değerlendirildiğinde;

İlk denemede;

1. Karakterleri doğru kodlama oranının %68-%96 arasında değiştiği,
2. Sağ ve yukarı bakışların doğru tanınma oranının %90 civarında olmasına karşın, çapraz yönlerdeki ve 2 bakışla kodlanan karakterlerin tanınmasının %80'lerde kaldığı,
3. 2 yöndeki bakış ve çapraz yöndeki bakış içeren karakterlerin doğru tanınanlardan en az olduğu,

Onuncu denemede;

1. Harflerin doğru tanınma yüzdelerinin %60-%88 arasında olduğu,
2. Yine en çok çapraz yönde bakış içeren kodlamaların hatalı yapıldığı görülmüştür.

Bu çalışmada karakterlerin doğru tanınması, iki karakter arasında karşıya bakarak bekleme süresinin, iki yöne bakışla oluşturulan harflerin bekleme süresinden daha büyük olmasına bağlıdır. Deneyler sırasında iki yönde bakışla kodlanan karakterlerin yanlış tanınma oranının daha yüksek olması deneklerin bu ayrımı kolayca uygulayamamasından kaynaklanmıştır.

Tekrar sayısı arttıkça kullanıcıların sisteme alışmasından dolayı tanınma yüzdelerinin ve hızlarının artacağı düşünülse de, hızın ancak %40'larda arttığı, en iyi tanınmanın iki yöne bakışta %76'larda kaldığı görülmüştür. Bu sonuçlar, deneme sayısı arttıkça kullanıcının hızının artmasına rağmen, karakterlerin doğru kodlanma oranının azaldığını göstermektedir.

Geliştirilen yeni sisteme ışık tutması için, ilk denemelerin yapıldığı EOG tabanlı gözle kodlama sisteminin %10-%20 arasında değişen hatalı kodlamaların sebepleri ayrıntılı olarak incelenmiştir.

100 kaydın 34 tanesinde art arda 2 yöndeki bakışla kodlanan karakterlerin doğru tanınmadığı görülmüştür. Bunun, kullanıcıların arka arkaya 2 yöndeki bakış arasındaki kısa süreli olması gereken bekleme süresini doğru ayarlayamamasından kaynaklandığı değerlendirilmiştir. Yeni geliştirilen yöntemde bu kodlama şeklinden vazgeçilmiştir.

100 kaydın 26 tanesindeki hatanın karşıya bakış sırasında 0 civarında olması gereken EOG sinyalinin, taban hattının kaymasından dolayı bu değerden daha negatif veya pozitif olmasından kaynaklandığı görülmüştür. Yeni sistem geliştirilirken yapılan denemeler sonucu taban kaymasını gidermek için yüksek geçiren butterworth filtrenin kesim frekansı 0,05 Hz'in biraz daha altında bir değer olarak 0,02 Hz olarak seçilmiştir.

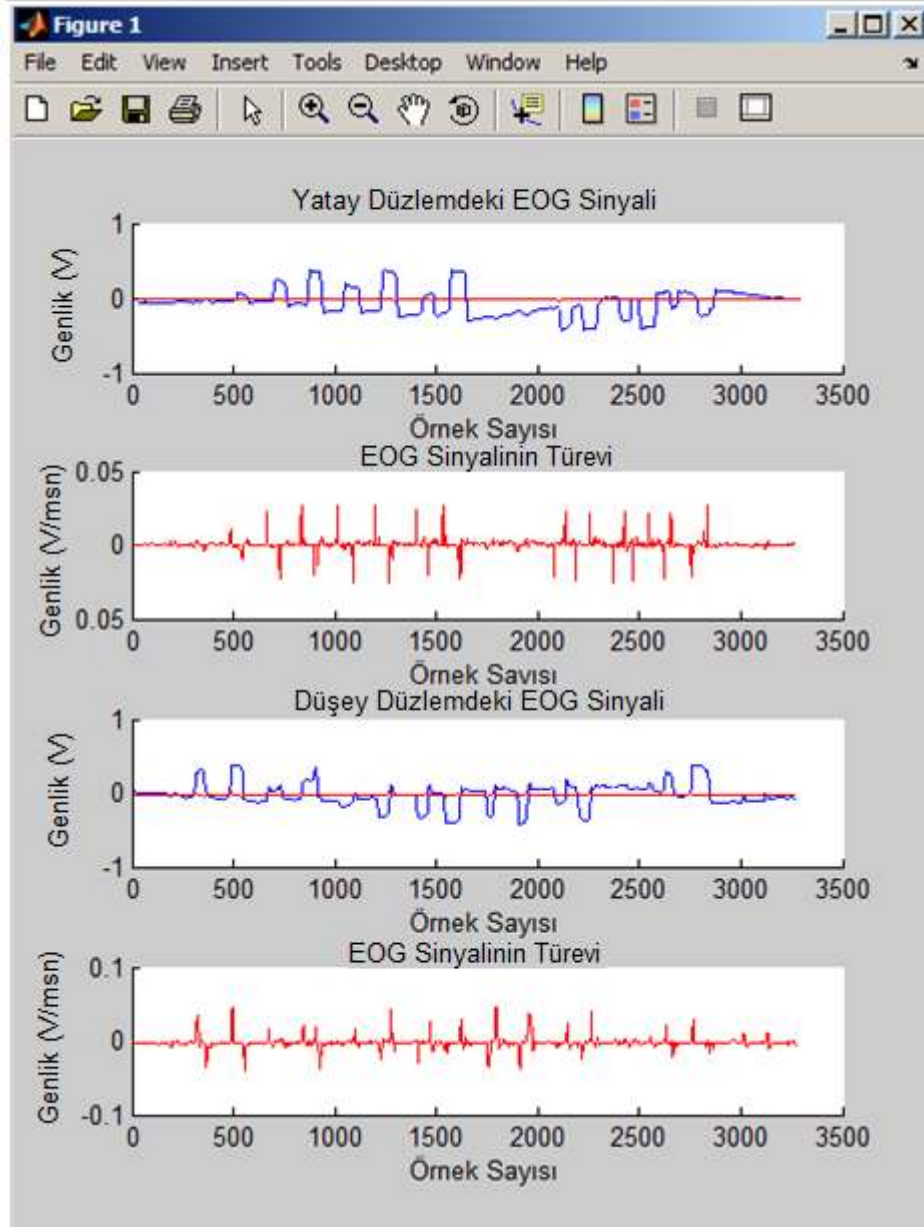
100 kayıttan 21'inde aslında deneğin karakteri kodladığı fakat belirlenen sınır değerini geçmeyen bir EOG üretildiği için karakterin tanınmadığı görülmüştür. Bu durum geliştirilen yeni yöntemde sınır değerlerin yazılımla seçilebilmesine imkan verilmesi ve kodlanan harfin sistem tarafından seslendirilmesi ile önlenilmeye çalışılmıştır.

100 kaydın 6 tanesinde EOG sinyalinin tabanının kaydığı ve hatalı tanımlara bunların yol açtığı görülmüştür. Bu durumun elektrotların kullanıcıya bağlanmasının ardından hemen kayıt alınmasından, kararlı duruma gelmesinin beklenmemesinden kaynaklandığı değerlendirilmiştir. Geliştirilen yeni yöntem uygulanırken bu durum dikkate alınmış ve deneylere elektrot potansiyelleri dengeye geldikten sonra başlanmıştır.

Belirlenen yeni prosedür geliştirilirken bu sonuçlar ile birlikte R.Barea ve ark. 2012 çalışmalarından esinlenilmiştir. Bu çalışmada farklı bakma açılarıyla oluşacak farklı genliklerdeki EOG sinyallerini tespit eden bir yöntem önermişlerdir. Sistem 10°, 20°, 30°, 40° ile farklı yönler bakılması prensibiyle çalışmaktadır. R.Barea ve ark. 2012 yılındaki çalışmalarında, farklı açılarla bakıldığında oluşacak sinyalin genliğinin farklı olması ve ardından alınan türev değerlerinin farklı olmasının bu bakışlarının birbirinden ayrılmasında kullanılabileceğini öngörmüşlerdir.

Bu çalışmanın öngörüsü ele alınarak, kullanıcının dört yöne dört farklı açıyla bakması sonucu oluşacak sinyallerin ayrıştırılmasının olanağı araştırılmıştır. Şekil 4.1' de bu deneye ait bir örnek gösterilmiştir. Şekil 4.1'de ilk ve üçüncü grafik yatay ve düşey göz hareketleri sonucu oluşan EOG sinyallerini göstermektedir. İkinci ve dördüncü grafik ise bu sinyallerin türevlerini göstermektedir. İncelenen kayıtlarda 4 farklı bakma açısıyla bakış sırasında oluşan EOG genliklerinin birbirine çok yakın olması ve sinyalin taban hattının kayması dolayısıyla bunlar arasındaki ayrımın

çokta kolay yapılamayacağı görülmüştür. Bu olumsuzluklardan dolayı geliştirilen yeni yöntemde, 4 açı ile bakma yerine 2 açı (10° ve 40°) ile bakmanın kullanılmasına karar verilmiştir.



Şekil 4.1 Dört farklı açı ile bakılması sırasında ortaya çıkan EOG sinyalleri ve türevleri

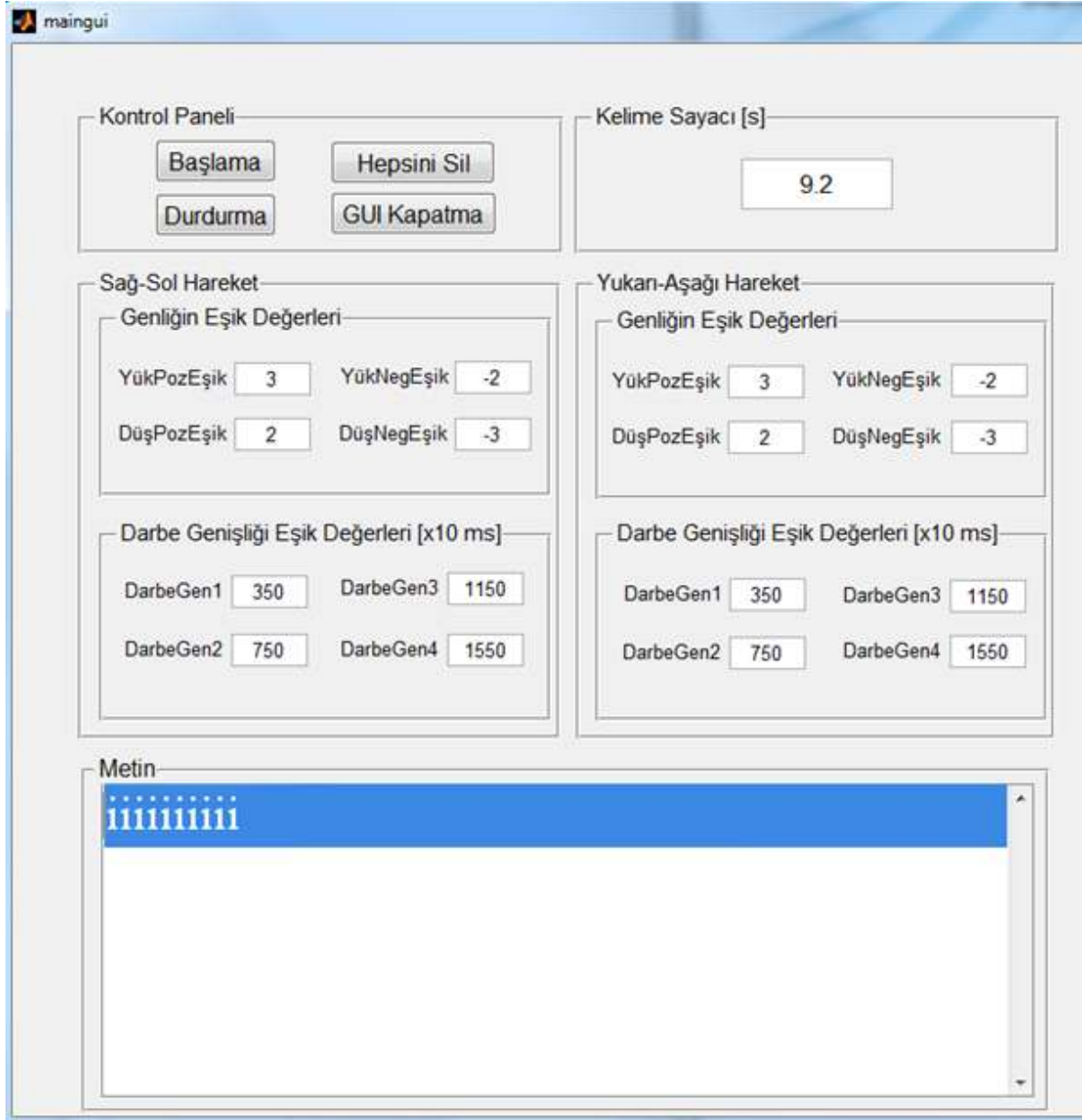
Önerilen yeni yöntemde 4 yöndeki bakışın 2 farklı açı ile yapılması 8 en çok kullanılan harfe atanırken, daha az kullanılan diğer harflerin bir yöne iki açıdan biri ile baktıktan sonra belli süreler beklenerek kodlanması sağlanmıştır. Kullanıcı sağ, sol, yukarı, aşağı olmak üzere 4 yöne, 2 farklı açı ile ve 4 farklı bekleme süresi ile baktığında toplam 32 karakter kodlanmıştır. Kullanıcıların harfleri doğru

kodlayabilmesi, bekleme sürelerini doğru ayarlamasına bağlıdır. Bunu sağlamak için; ayrılmış her bekleme süresinin başında kişi ilgili harf seslendirilerek uyarılmakta, ayrılan sürede bakışını karşıya yönlendirmesi ile karakter tanınıp ekrana yazdırılmaktadır.

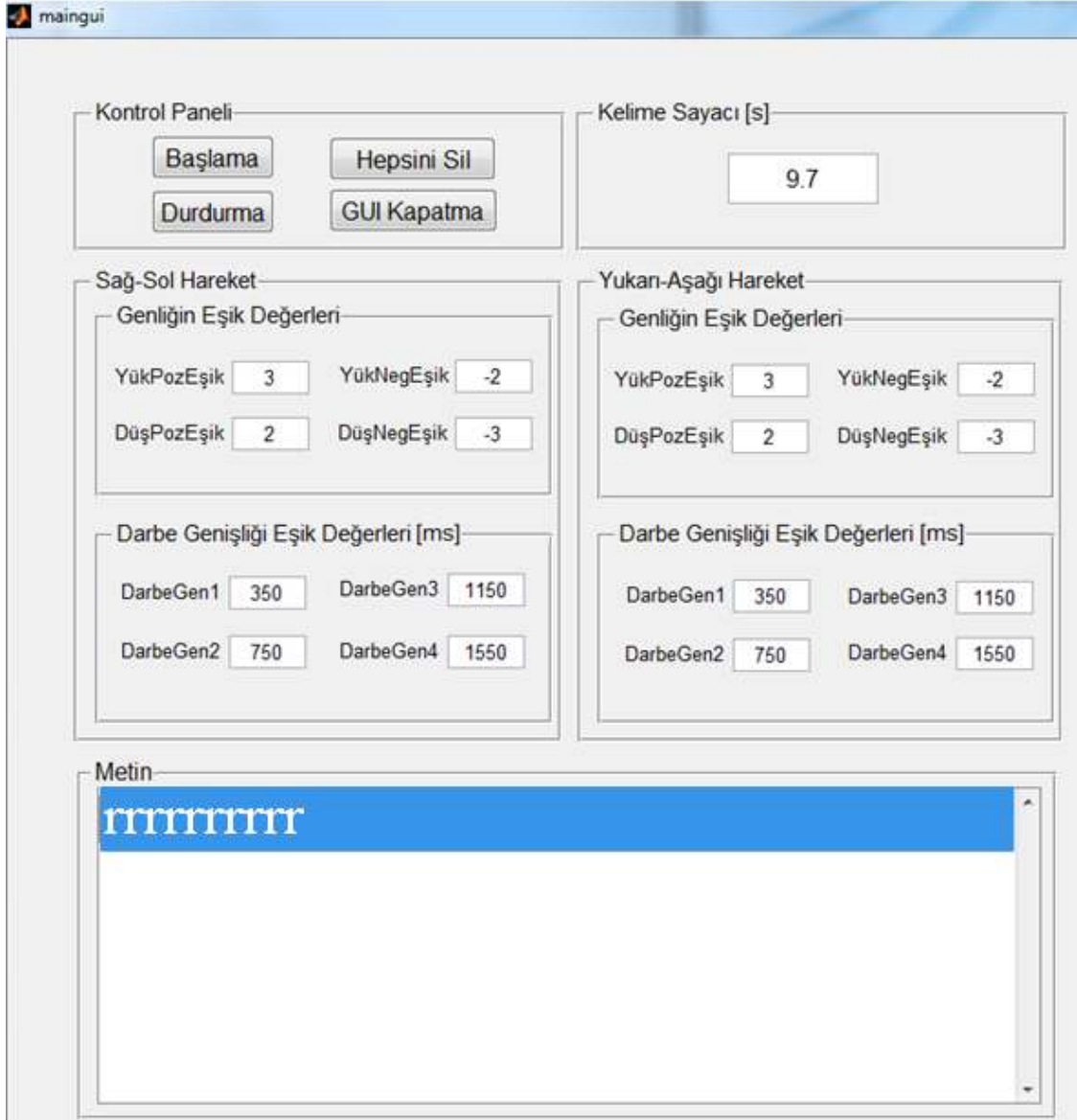
Bu noktadan sonra geliştirilen yeni kodlama sistemi referans alınarak geliştirilen gerçek zamanlı gözle yazı yazma sisteminin performansının testine yönelik çalışmalara yer verilmiştir. Çalışma, kendi tasarladığımız sistem üzerinde denendiğinden ve bunun için gerekli etik kurul izni alınmadığı için sistemin performans testleri sadece kendi üzerimizdeki denemelerle yapılmıştır.

Çeşitli yönlerdeki farklı açılarla yapılan kodlamaların hangi hızla yapılabildiğini tespit için grafiksel kullanıcı arayüzüne kodlamanın başlama ve bitiş sürelerinin farkını gösteren kelime sayıcı eklenmiştir. İlk denemede yukarı, 10 derecelik açı ile en kısa süre bakışla kodlanan i karakteri 10 kez 10'ar tekrarlı şekilde yazılmıştır. Bir deneme sonucu Şekil 4.2'de gösterilmiştir. Altıncı denemeden sonra eşik değerleri ve bekleme süreleri bu yöndeki bakış için karakterin en rahat kodlanabileceği değerlere ayarlanmıştır. Aynı karakterin tekrar kodlanmasıyla, göz bu harekete alışacak ve kodlama daha kolay olacaktır. Bu nedenden dolayı eşik değeri sıralaması 350, 750, 1150, 1550 ms olacak şekilde düşürülmüştür. Şekil 4.2'de i harfinin 10 kez tekrarlanması 9,2 sn sürmektedir. Bir karakterin 0,92 sn de yazılabildiği, bu da dakikada yaklaşık 13 kelimenin yazılabileceği anlamına gelir. Toplam yazılan 100 karakterin 97 tanesi ilk denemede doğru yazılmıştır yani bu deneye göre sistemin doğruluğu %97'dir.

Şekil 4.3'te sola uzun süreli bakış sonucunda oluşan art arda 10 kez kodlanan karakter için kodlamanın yapıldığı kullanıcı arayüzü görülmektedir. DüşPozEşik değeri aşan sağ yön için (350 ms) sinyal text bölümünde "r" harfi olarak anlamlandırılmaktadır. Bu deneyde de kullanıcı art arda aynı karakteri yazdığından yazma hızı artmış eşik değeri sıralaması bu nedenle 350, 750, 1150, 1550 ms olacak şekilde düşürülmüştür. Kullanıcı art arda 10 tekrarlı onuncu denemede aynı 10 karakteri 9,7 sn'de kodlamıştır. Bir karakterin 0,97 sn de yazılabildiği görülmektedir. Bu da dakikada yaklaşık 13 kelimenin yazılabileceği anlamına gelir.

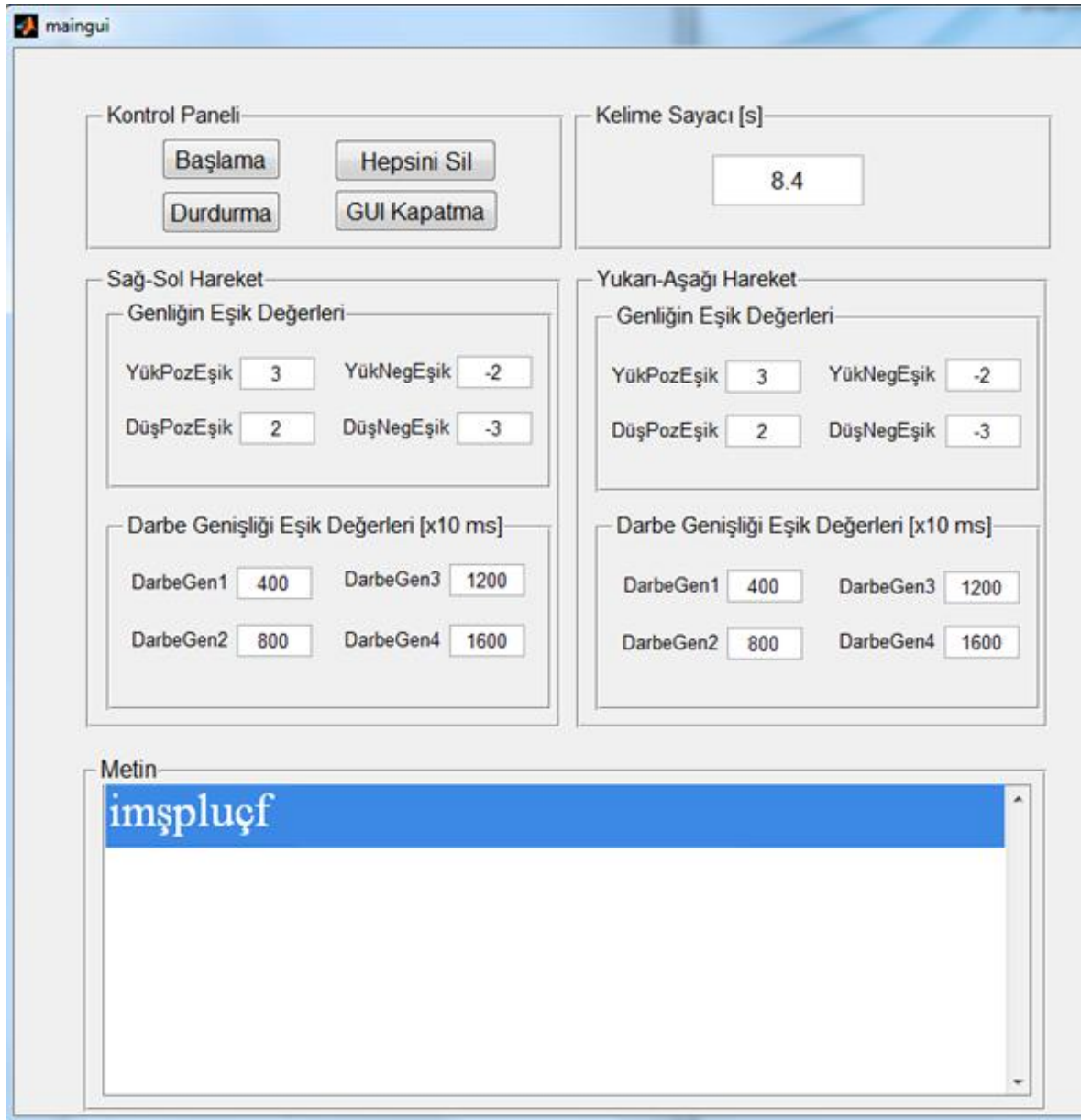


Şekil 4.2 Yukarı yönde bakışta kodlanan i harfinin art arda 10 kez yazılması
Deney sırasında yazılan toplam 100 karakterin 96 tanesi ilk deneme de doğru yazılmıştır. Şekil 4.3'de bu deneyin arayüz ekranı görülmektedir.



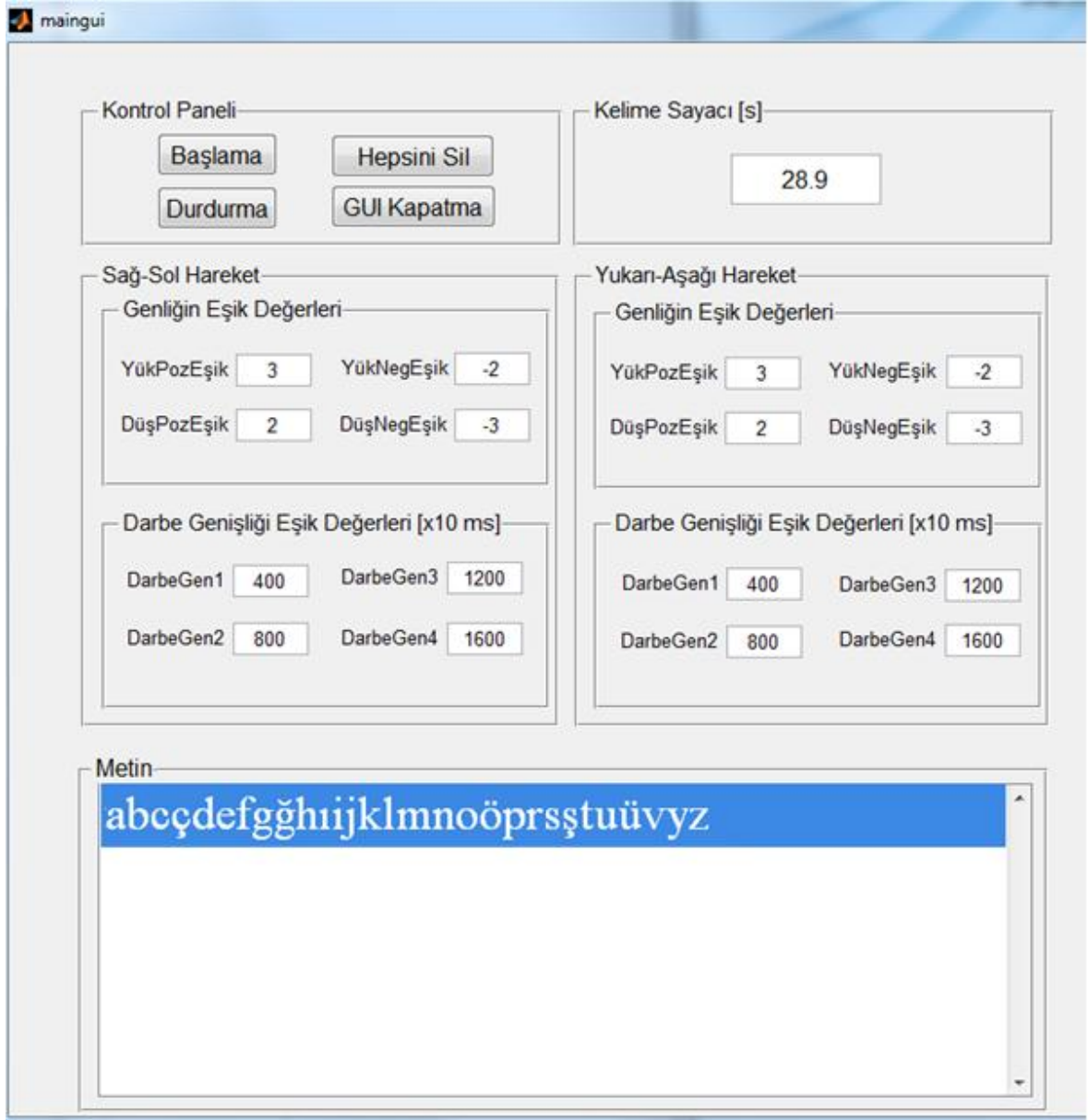
Şekil 4.3 Sol yönde bakışta kodlanan r harfinin art arda 10 kez yazılması

Şekilde görüleceği gibi kullanıcı tek yönde yazılabilecek karakterlerin tümünü birbiri peşi sıra yazmayı denemiştir. Yukarı yön için belirlenen tüm harfleri (sırasıyla i, m, ş, p, l, u, ç, p harflerini) 8,4 saniyede kodlamıştır. Bu deneyde kullanıcı 10 tekrarlı 8 karakteri en hızlı dokuzuncu denemesinde yapmıştır. Bir karakterin 1,05 sn de yazılabildiği görülmüştür. Bu da dakikada 11,5 kelimenin yazılabileceği anlamına gelir. Bu deneyde kodlanan 80 karakterin 75 tanesi doğru kodlanmıştır. Bu durumda deneğin bu deneydeki doğru yazma oranı %94'tür. Şekil 4.4'de bu denemenin arayüz ekranı görülmektedir.



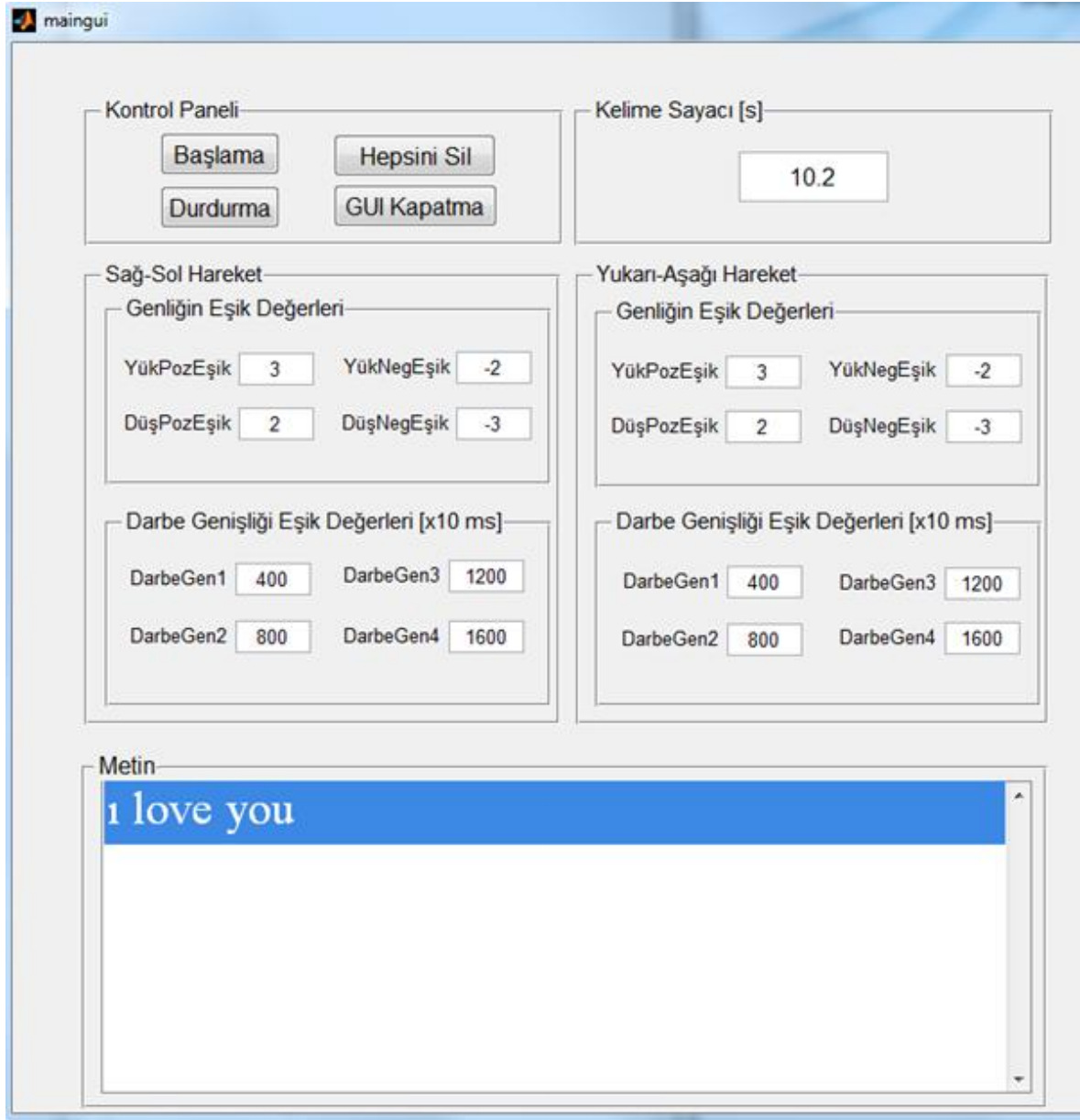
Şekil 4.4 Yukarı yöndeki farklı aç ve bekleme süreleri ile harflerin kodlanması

Bir sonraki denemede olabilecek tüm yönler ve tüm bekleme süreleri bir arada denenmiş yani alfabedeki bütün karakterler yazılmıştır. Kullanıcı sağ, sol, yukarı ve aşağı için atanan tüm harflerin yerini öğrendikten sonra beşinci denemesinde tüm harfleri 28,9 sn'de kodlamıştır. Şekil 4.5'de bir karakterin 0,99 sn de yazılabildiği görülmektedir. Bu da dakikada yaklaşık 12 kelimenin yazılabileceği anlamına gelir. Bu deneyde ilk denemede doğru tanınma oranı %95'dir.



Şekil 4.5 Türk alfabesindeki tüm harflerin yazılması

Kullanıcı alfabenin yerlerine alıştıktan sonra istediği tüm cümleleri ekrana yazabilmektedir. Şekil 4.6'da kullanıcının "i love you" yazma denemesi görülmektedir. Bu yazımı ilk kez yazan kullanıcı, ilk denemede kelimeleri doğru yazamadığından on karakterden beşini tekrar kodlaması gerekmiştir. Beşinci denemede ise kullanıcı harflerin sıralarını ve yerlerini daha iyi öğrendiğinden yanlış kodlanan karakter sayısı üçe inmiştir. Son deneme olan onuncu denemede ise kullanıcı yalnızca bir karakteri yanlış kodlamıştır. Bu deneyde ilk denemede doğru kodlama oranı %90'dır. En hızlı olan son deneme baz alındığında kullanıcı 10 karakteri 10,2 sn'de kodlamıştır



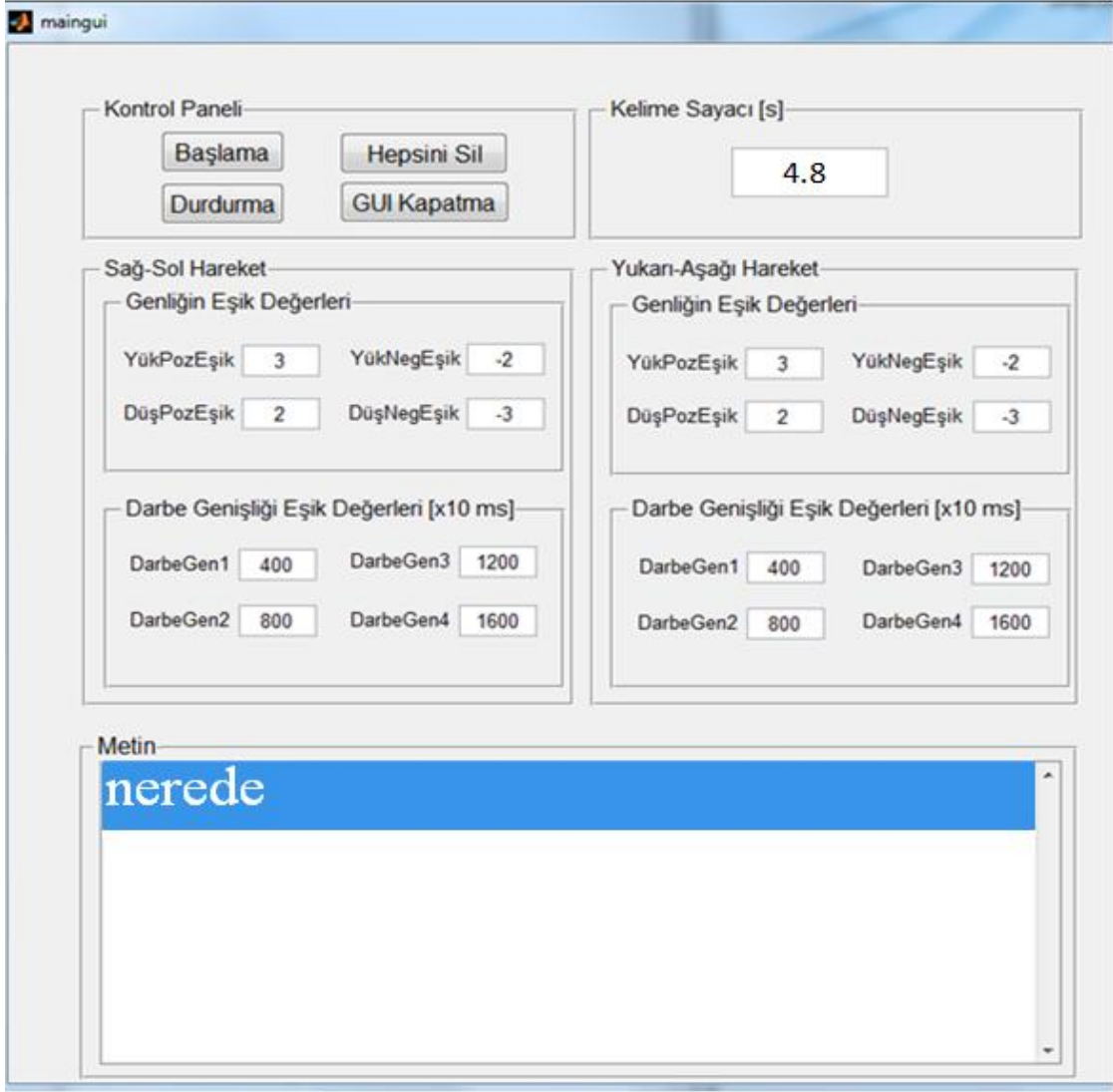
Şekil 4.6 “i love you” cümlesinin yazılması

Daha sonra benzer çalışmalarda kullanılan standart kelimeler olup olmadığı araştırılmış ancak böyle bir kelimeye rastlanmamıştır. Bu sebeple bu tür kas hastalıkları bulunan kişilerin en temel ihtiyaçlarını karşılanacağını düşünüldüğü kelimeler seçilerek denemeler yapılmıştır. Bu kelimeler; acıktım, susadım, tuvaletim var, merhaba, günaydın, iyi geceler, nasılsın, görüşürüz, ışığı aç, ışığı kapat, nerede, üşüyorum, çok sıcak ve ilaçlarımı ver'dir. Bu kelimeleri kullanıcı beşer tekrar yaparak denemiştir. Bu denemelerin hızları Çizelge 4.2'de verilmiştir.. Bu çizelgeye göre; tüm kelimeler için deneme sayısı artıkça yazma hızının arttığı görülmektedir.

Çizelge 4.3’de ise bu denemeler sonucu elde edilen doğru tanınma yüzdeleri görülmektedir. Önerilen yeni yöntem ve geliştirilen sistemle ulaşılan azami yazma hızı 17,5 kelime/dakikadır. 126 harften oluşan bir metnin yazılmasının ilk denemesinde %92,5’lik doğru kodlama oranı sağlanmışken dördüncü denemeden itibaren %100’lük doğru kodlama seviyesine ulaşılmıştır.

Çizelge 4.2 ve Çizelge 4.3’ten görüldüğü gibi, kullanıcının deneme sayısı arttıkça yazma hızı ve karakterleri doğru kodlama yüzdelerinin her ikisi de artmıştır.

Tek kişi üzerinde yapılan tüm deney sonuçlarına göre; bir karakterleri en hızlı yazılması “nerede” kelimesinin beşinci denemesinde 4,8 sn olarak gerçekleşmiştir. Bu denemeye ait arayüz şekil 4.7 de gösterilmiştir. Nerede kelimesi sonuna yazma süresinin belirlenmesi için konulan boşluk karakteriyle birlikte 7 karakterden oluşmaktadır. Bu sonuçlara göre saniyede; $4.8/7 = 0,68$ sn/karakter yazılabildiği, dakikada yazabildiği karakter sayısının $60/0,68=88$ karakter/dakika olduğu ve sonuç olarak yazma hızının, $88/5 =17,6$ kelime/dakika olduğu görülmektedir.



Şekil 4.7 Nerede kelimesinin yazımını gösteren arayüz ekranı

Çizelge 4.2 Günlük hayatta kullanılabilir kelimelerin ortalama, en hızlı, en yavaş yazma süreleri

Kelime	Birinci Deneme Süresi (sn)	İkinci Deneme Süresi (sn)	Üçüncü Deneme Süresi (sn)	Dördüncü Deneme Süresi (sn)	Beşinci Deneme Süresi (sn)	Ortalama (sn)	En Hızlı (sn)	En Yavaş (sn)
susadım	6,8	8,7	6,8	6,6	6,5	7,0	6,5	8,7
acıktım	9,1	7,1	6,9	6,2	6,7	7,2	6,2	9,1
tuvaletim var	14,1	13,7	13,4	13,2	13,2	13,5	13,2	14,1
merhaba	8,3	7,8	7,6	7,2	7,3	7,6	7,2	8,3
günaydın	9,0	9,1	9,0	8,5	8,9	8,9	8,5	9,1
iyi geceler	11,7	11,8	11,2	11,3	11,0	11,4	11,0	11,8
nasılsın	8,1	8,4	7,6	7,7	7,7	7,9	7,6	8,4
görüşürüz	11,1	10,9	11,1	10,9	10,2	10,9	10,2	11,1
ışığı aç	9,4	9,6	8,3	8,5	7,9	8,8	7,9	9,6
ışığı kapat	12,4	11,9	11,4	11,2	11,3	11,6	11,2	12,4
nerede	6,7	5,6	5,2	5,2	4,8	5,5	4,8	6,7
üşüyorum	9,8	9,8	9,6	9,7	9,2	9,6	9,2	9,8
çok sıcak	10,4	10,6	9,8	10,3	9,9	10,2	9,8	10,6
ilaçlarımı ver	16,5	15,9	15,2	15,0	15,4	15,6	15,0	16,5

Çizelge 4.3 Günlük hayatta kullanılabilecek kelimelerin doğru tanınma yüzdeleri

Kelime	Doğru Kodlanan Harf Sayısı	İlk Denemede Doğru Tanınma Yüzdesi	Beşinci Denemede Doğru Tanınma Yüzdesi
susadım	6D,6D,7D,7D,7D	%86	%100
acıktım	6D,7D,7D,7D,7D	%86	%100
tuvaletim var	11D,12D,13D,13D,13D	%85	%100
merhaba	6D,7D,7D,7D,7D	%86	%100
günaydın	7D,7D,8D,8D,8D	%88	%100
iyi geceler	9D,10D,11D,11D,11D	%82	%100
nasılsın	7D,8D,7D,8D,8D	%88	%100
görüşürüz	8D,9D,9D,8D,9D	%89	%100
ışığı aç	8D,7D,7D,8D,8D	%88	%100
ışığı kapat	9D,11D,10D,10D,11D	%82	%100
nerede	5D,6D,6D,6D,6D	%84	%100
üşüyorum	7D,8D,8D,7D,8D	%88	%100
çok sıcak	9D,8D,8D,9D,9D	%89	%100
ilaçlarımı ver	12D,14D,13D,14D,13D	%86	%100
İlk Deneme Ortalama Tanınma Yüzdeleri	%92.5	Onuncu Deneme Ortalama Tanınma Yüzdeleri	%100

5. YORUM VE ÖNERİLER

Geliştirilen yeni EOG tabanlı gözle yazı yazma sisteminden alınan ilk sonuçlar oldukça umut vaat edicidir. Tek kişi üzerinde yapılan denemelerde ulaşılan en hızlı yazma hızı dakikada 17,5 kelime/dk olarak tespit edilmiştir. Yazma hızının 5 tekrar sonunda %18 arttığı dikkate alınırsa, tekrar sayısı arttıp kodlama yöntemi daha iyi öğrenildiğinde bu hızın daha da artacağı öngörülebilir. Daha önce geliştirilmiş EOG tabanlı sistemlerin en hızlısı olan Majaranta ve ark. 2002 çalışmalarındaki [3] yazma hızı dakikada 10 kelime/dk'dır. Geliştirilen sistemle gözle yazma hızı yalnızca 2012'de Per Ola ve ark.'larının [16] dakikada 46 kelime olan gözle yazma hızının gerisinde kalmıştır. Bu sistem Tobii isimli profesyonel göz takip sistemi kullanmakta ve yazmaya başlandığında yazılmak istenebilecek kelimeleri kişinin önüne getirip seçmesini sağlayarak daha hızlı yazmaya imkan vermektedir [16]. Ancak Per Ola ve arkadaşlarının çalışmasında kullanılan göz takip sistemi, geliştirilen EOG tabanlı yöntemle göre çok daha yüksek maliyete sahiptir. Geliştirilen sisteme Per Ola ve ark. 2012, benzer bir kelime tamamlama yazılımı eklenebilirse, yazma hızı daha da artabilir.

Geliştirilen yöntem literatüre göre önemli yenilikler içermektedir. Tüm karakterlerin sadece 4 yöndeki bakışla kodlanıyor olması, karakterlerin tanınmasını; EOG'nin basit bir eşik değeri geçmesinin kontrolü ile tanınabilir hale getirmiştir. Önerilen yöntemle, Türkçedeki kullanım yüzdeleri en yüksek 8 karakter (toplamları %50) sadece 0,4-0,5 saniyede kodlanabilir duruma gelmiştir.

Literatürde ilk defa olarak kullanılan, karakterlerin kodlanmasını seslendirerek doğru kodlamayı destekleme olanağı sayesinde, daha uzun süreli beklemelemlerle kodlanan karakterlerin de kolayca kodlanabilir hale getirilmesi sağlanmıştır. Bu durum hatalı yazma riskini azalttığı gibi hatalı yazımın hemen farkına varılıp, sisteme eklenen "son yazılanı sil" bakışı ile kolayca silinebilmesini sağlanmıştır.

Daha önceki EOG tabanlı sistemlerin çoğu karakterlerin gözle çizilmesi veya sanal klavye üzerindeki yerlerine gözün sıçramaları ile gidilmesi ve harfin seçilmesi için bir süre beklenmesi prensibiyle çalışmaktadır. Bu yöntemler hem kullanıcının gözlerini yormakta hem de sistemin yazma hızını düşürmektedir. Önerilen sistemin kullanımı için önce harflerin hangi yön ve bekleme süresi ile seçileceğinin yani kodlanmasının ezberlenmesi gerekmektedir. Bu zaman alacak ve sistemin

kullanımını ilk başta zorlayacak bir durumdur. Fakat yapılan denemeler harflerin yeri ezberlendikçe bu dezavantajın ortadan kalkabileceğini, yöntemin yazma hızının önceki sistemlere göre %50 daha fazla olması sebebi ile getirdiği yüke katlanılabileceğini göstermiştir.

Sistemi kullanmanın bir diğer zorluğu ise yüze takılan elektrotların kullanıcıyı rahatsız etmesidir. Geliştirilen yöntem kameralı göz takip sistemleri ile uygulamaya da müsait olduğundan bundan sonraki çalışmalarda, yöntemin kameralı bir göz takip sistemi ile uygulanması ile bu rahatsızlıktan kurulunabilir.

Geliştirilen yöntem farklı yönlerdeki bakışlara farklı dillerin alfabelerindeki harfler atanarak, birçok farklı dile kolayca adapte edilebilir. Tasarlanan sistemde bakış yönlerine harf ataması yapıldığından 29 harfe kadar alfabesi bulunan tüm dillerde, harflerin kullanım yüzdelerine göre yeni bir tablo oluşturularak belirlenen hızlarda çalışabilecektir. Alfabesinde daha çok harf olan dillerde ise sistemdeki mevcut 4 bekleme süresine yeni bekleme süreleri eklenerek, biraz daha yavaş da olsa sistemin kullanımı mümkün olabilecektir.

Geliştirilen sistemin gömülü sistem olarak tasarlanması mümkün olup, bilgisayarla birlikte kullanılması gereken sistemlere göre daha ucuz bir şekilde imal edilebilir.

Geliştirilen sistemin hedef kitlesi sinir kas hastalığı olup çevresiyle iletişimini gözleriyle yazarak sağlamak zorunda olanlar olsada, geliştirilen yöntemin sağlıklı veya hasta kişilerin farklı uzuvlarını kullanarak bilgisayarlı sistemlere daha hızlı yazı yazması konusunda geliştirilecek bir sisteme öncülük etmesi mümkündür.

Tez kapsamında geliştirilen gözle yazı yazma sistemi; çevresi ile iletişim kurma olanağı olmayan kişilerin yaşam kalitesinin artırılmasına, sevdiklerine söylemek istediklerini aktarabilmesine, birikimlerini gelecek kuşaklara aktarmalarını sağlamaya ve aileleri ile kurabilecekleri iletişim ile bir nebze olsun mutlu olmalarına yardımcı olabilecektir.

6. KAYNAKÇA

- [1] Wolpach J.R., Birbaumer N., Heetderks W. J. ve ark. "Brain–Computer Interface Technology: A Review of the First International Meeting", IEEE Tran. Rehabilitation Eng., Vol. 8, No. 2, 2000, p164-173.
- [2] Zhang Jiaan , Eye-writing Keyboard and Eye-writing Mouse, National Central University 1995.
- [3] Majaranta, P. ve RAIHA, K. J., "Twenty years of eye typing: systems and design issues" Proc. of Eye Tracking Research and Applications, 2002.
- [4] Uşaklı A.B., Gürkan S., Aloise F., Vecchiato G., Babailoni F., "On the Use of Electrooculogram for Efficient Human Computer Interfaces," Hindawi Publishing Corporation Computational Intelligence and Neuroscience, 2009.
- [5] Tsai J. Z., Lee C.K., Wu C.M., Wu J.J. Kao K.P., A feasibility of an eye writing system based on electro-oculography, Journal Of A Medical And Biologicak Engineering , vol. 28(1), 2007, p 39-46.
- [6] Wobbrock J.O., Rubinstein J., Sawyer M., Duchowski A.T., "Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures", The 3rd Conference on Communication by Gaze Interaction, Leicester, UK, 2007.
- [7] Porta M., Ttrina M. "Eye-S: a Full-Screen Input Modality for Pure Eye-based Communication", ETRA 2008, Savannah, Georgia, March 2008.
- [8] Diri B., Karslıgil M. Y. , "Türk Dilinin Biçimbilim Yapısından Yararlanarak Türkçe Metinlerin Farklı İmgelere Ayrılarak Kodlanması ve Sıkıştırılması", Yıldız Teknik Üniversitesi Elektrik Elektronik Fakültesi - Bilgisayar Mühendisliği.
- [9] Kristensson P. O., Vartanen K., "The Potential of Dwell-Free Eye-Typing for Fast Assistive Gaze Communication" ETRA 2012, Santa Barbara, CA, 28-30 March , 2012.
- [10] Majaranta P., Text Entry by Eye Gaze, Dissertations in Interactive Technology, Faculty of Information Sciences of the University of Tampere, 2009.
- [11] Farksal Yalıtılmış EKG Tasarımı ve Uygulaması Gül Fatma TÜRKER, İnan GÜLER, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü Dergisi, 16-3 2012, 264-268
- [12] Uşaklı A.B., Gürkan S., Aloise F., Vecchiato G., Babailoni F., "On the Use of Electrooculogram for Efficient Human Computer Interfaces," Hindawi Publishing Corporation Computational Intelligence and Neuroscience, 2009.
- [13] Darius Miniotas, Oleg Spakov & Grigori Evreinov Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space, 2000

- [14] Wolpaw J. R., Birbaumer N., Brain–computer interfaces for communication and control.
- [15] Barea R., Boquete L., Ortega S., López E., Rodríguez-Ascariz J.M, “EOG-based eye movements codification for human computer interaction”Electronics Department, University of Alcalá, 28871 Alcalá de Henares, Madrid, Spain.
- [16] <http://www.computerhistory.org/revolution/input-output/14/352/1843>.
- [17] Elektrookülogram Tabanlı Yeni Bir İnsan Bilgisayar Arabirimi ve Sanal Klavye Olarak Uygulaması Ali Bülent Uşaklı¹, Serkan Gürkan.
- [18] http://www.erciyesmedj.com/sayilar/211/buyuk/pdf_EMJ_93.pdf.
- [19] Yıldız M., EOG Tabanlı Yazı Yazma Sistemleri İçin Yeni Bir Kodlama Tekniği, SIU, 2011.
- [20] <http://www.ni.com/pdf/manuals/371303m.pdf>
- [21] <http://tr.wikipedia.org/wiki/Hemiparezi>
- [22] http://tr.wikipedia.org/wiki/Multipl_skleroz
- [23] http://tr.wikipedia.org/wiki/Serebral_palsi
- [24] http://tr.wikipedia.org/wiki/Locked-in_sendromu
- [25] Plum F., Posner J.B.,The diagnosis of stupor and coma. F.A. Davis Co. Philadelphia, Pennsylvania, USA., 1966, pp197.
- [26] http://tr.wikipedia.org/wiki/Amyotrofik_lateral_skleroz
- [27] Fishman GA. The Electro-Oculogram. In: Fishman GA, Birch DG, Holder GA, Brigell MG (eds). Electrophysiologic Testing in Disorders of the Retina, Optic Nerve and Visual Pathway. 2nd ed. Singapore: American Academy of Ophthalmology 2001: 157-170.
- [28] <http://tuncaycentel.com/cp1.htm>
- [29] <http://www.burhanettinuludag.com.tr/yazilar/yazilar/yazilar/MS.html>
- [30] <http://ttefdergi.gazi.edu.tr/makaleler/2008/Sayi2/134-147.pdf>
- [31] Brain Products Press Release, Brain Computer Interfaces, Thorsten Zander, Volume37, 2010
- [32] Majaranta P., Aula A., Räihä K. J., Effects of Feedback on Eye Typing with a Short Dwell Time, Human-Computer Interaction Unit (TAUCHI)Department of Computer Sciences FIN-33014 University of Tampere, Finland, 2004.

- [33] MacKenzie S., Zhang X., Eye Typing Using Word and Letter Prediction and a Fixation Algorithm, Dept. of Computer Science and Engineering, York University, Toronto, Canada, 2008.
- [34] <https://www.youtube.com/watch?v=-QXGiZBDkUw>
- [35] Deng,L. Y., Hsu C. L., Lin T. C., Tuan J. S., Chang S.M., EOG-based Human–Computer Interface system development.

EKLER LİSTESİ

Sayfa

EK 1 EOG Tabanlı Gözle Yazı Yazma Sistemi İçin Oluşturulan Program	59
--	----

EK 1 EOG Tabanlı Gözle Yazı Yazma Sistemi İçin Oluşturulan Program

```
function varargout = maingui(varargin)
% MAINGUI MATLAB code for maingui.fig
%     MAINGUI, by itself, creates a new MAINGUI or raises the existing
%     singleton*.
%
%     H = MAINGUI returns the handle to a new MAINGUI or the handle to
%     the existing singleton*.
%
%     MAINGUI('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in MAINGUI.M with the given input
arguments.
%
%     MAINGUI('Property','Value',...) creates a new MAINGUI or raises
the
%     existing singleton*. Starting from the left, property value pairs
are
%     applied to the GUI before maingui_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to maingui_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help maingui

% Last Modified by GUIDE v2.5 22-Jun-2014 14:41:16

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @maingui_OpeningFcn, ...
                  'gui_OutputFcn',  @maingui_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

end

% --- Executes just before maingui is made visible.
function maingui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```

```

% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to maingui (see VARARGIN)

% Choose default command line output for maingui
handles.output = hObject;

% UIWAIT makes maingui wait for user response (see UIRESUME)
% uiwait(handles.figure1);

global plotBuffer;
global timeBuffer;
global plotCounter;
global dataBufferRL;
global dataHoldRL;
global dataBufferUD;
global dataHoldUD;

% plotBuffer degiskenini sifirla. plotBuffer 2x1000'lik ADC verilerinden
% grafik cizdirmek icin kullanılan buffer
% dataBufferUD/dataBufferRL degiskenlerini sifirla. Bu buffer darbe
% tasnifi icin
% kullanılan buffer
% timeBuffer degiskenini sifirla. ADC'nin basladigi andan itibaren zaman
% degerini tutan buffer.
plotBuffer = zeros(2,1000);
dataBufferRL = [];
dataHoldRL = 0;
dataBufferUD = [];
dataHoldUD = 0;
timeBuffer = [0:0.01:(10-0.01)];
plotCounter = 0;

set(handles.pushbutton1,'enable','on');
set(handles.pushbutton2,'enable','off');

question_string = ['DİL SEÇENEĞİ' ]

choice = questdlg( question_string ,
'SEÇENEK','TÜRKÇE','İNGİLİZCE','İNGİLİZCE');

switch choice

    case 'TÜRKÇE'

        set(handles.uipanel1,'title','Kontrol Paneli');
        set(handles.pushbutton1,'string','Başlama');
        set(handles.pushbutton2,'string','Durdurma');
        set(handles.pushbutton3,'string','Hepsini Sil');

```

```

set(handles.pushbutton4,'string','GUI Kapatma');
set(handles.uipanel12,'title','Kelime Sayacı [s]');
set(handles.uipanel9,'title','Sağ-Sol Hareket');
set(handles.uipanel4,'title','Genliğin Eşik Değerleri');
set(handles.uipanel5,'title','Darbe Genişliği Eşik Değerleri [x10 ms]');
set(handles.text1,'string','YükPozEşik');
set(handles.text2,'string','DüşPozEşik');
set(handles.text3,'string','YükNegEşik');
set(handles.text4,'string','DüşNegEşik');
set(handles.text5,'string','DarbeGen1');
set(handles.text6,'string','DarbeGen2');
set(handles.text7,'string','DarbeGen3');
set(handles.text15,'string','DarbeGen4');

```

```

set(handles.uipanel10,'title','Yukarı-Aşağı Hareket');
set(handles.uipanel17,'title','Genliğin Eşik Değerleri');
set(handles.uipanel8,'title','Darbe Genişliği Eşik Değerleri [x10 ms]');
set(handles.text8,'string','YükPozEşik');
set(handles.text9,'string','DüşPozEşik');
set(handles.text10,'string','YükNegEşik');
set(handles.text11,'string','DüşNegEşik');
set(handles.text12,'string','DarbeGen1');
set(handles.text13,'string','DarbeGen2');
set(handles.text14,'string','DarbeGen3');
set(handles.text16,'string','DarbeGen4');

```

```

set(handles.uipanel6,'title','Sağ-Sol Hareket Monitörü');
set(handles.uipanel11,'title','Yukarı-Aşağı Hareket Monitörü');
set(handles.uipanel3,'title','Metin');

```

```

    case 'İNGİLİZCE'

```

```

end

```

```

% Grafik limitlerini ayarla
handles.myplot1 = plot(handles.axes1,timeBuffer, plotBuffer(1,:));
set(handles.axes1, 'YLim', [-5,5]);
set(handles.axes1, 'XLim', [0,10]);

```

```

handles.myplot2 = plot(handles.axes2,timeBuffer, plotBuffer(2,:));
set(handles.axes2, 'YLim', [-5,5]);
set(handles.axes2, 'XLim', [0,10]);

```

```

% Alfabe karakterlerini GUI açıldığında anda degiskenlere kaydet

```

```

global ch_a ch_b ch_c ch_d ch_e ch_f ch_g ch_h ch_i ch_j ch_k ch_l ch_m
...
ch_n ch_o ch_p ch_r ch_s ch_t ch_u ch_v ch_y ch_z Fs;

```

```

global flag_a flag_a1 flag_b flag_c flag_d flag_d1 flag_e flag_f flag_g1
...
    flag_g flag_h flag_i flag_j1 flag_j flag_k flag_l flag_m1 flag_m ...
    flag_n flag_o flag_p1 flag_p flag_r flag_s flag_t1 flag_t flag_u ...

```

```

    flag_v flag_y1 flag_y flag_z;

flag_a1 = 0;
flag_a = 0;
flag_b = 0;
flag_c = 0;
flag_d1 = 0;
flag_d = 0;
flag_e = 0;
flag_f = 0;
flag_g1 = 0;
flag_g = 0;
flag_h = 0;
flag_i = 0;
flag_j1 = 0;
flag_j = 0;
flag_k = 0;
flag_l = 0;
flag_m1 = 0;
flag_m = 0;
flag_n = 0;
flag_o = 0;
flag_p1 = 0;
flag_p = 0;
flag_r = 0;
flag_s = 0;
flag_t1 = 0;
flag_t = 0;
flag_u = 0;
flag_v = 0;
flag_y1 = 0;
flag_y = 0;
flag_z = 0;

[ch_a,Fs] = audioread('alfabe\a.wav');
[ch_b,Fs] = audioread('alfabe\b.wav');
[ch_c,Fs] = audioread('alfabe\c.wav');
[ch_d,Fs] = audioread('alfabe\d.wav');
[ch_e,Fs] = audioread('alfabe\e.wav');
[ch_f,Fs] = audioread('alfabe\f.wav');
[ch_g,Fs] = audioread('alfabe\g.wav');
[ch_h,Fs] = audioread('alfabe\h.wav');
[ch_i,Fs] = audioread('alfabe\i.wav');
[ch_j,Fs] = audioread('alfabe\j.wav');
[ch_k,Fs] = audioread('alfabe\k.wav');
[ch_l,Fs] = audioread('alfabe\l.wav');
[ch_m,Fs] = audioread('alfabe\m.wav');
[ch_n,Fs] = audioread('alfabe\n.wav');
[ch_o,Fs] = audioread('alfabe\o.wav');
[ch_p,Fs] = audioread('alfabe\p.wav');
[ch_r,Fs] = audioread('alfabe\r.wav');
[ch_s,Fs] = audioread('alfabe\s.wav');
[ch_t,Fs] = audioread('alfabe\t.wav');
[ch_u,Fs] = audioread('alfabe\u.wav');
[ch_v,Fs] = audioread('alfabe\v.wav');
[ch_y,Fs] = audioread('alfabe\y.wav');
[ch_z,Fs] = audioread('alfabe\z.wav');

```

```

% Update handles structure
guidata(hObject, handles);

end

% --- Outputs from this function are returned to the command line.
function varargout = maingui_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
end

function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a
double
end

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
end

% --- Executes on button press in pushbutton1.
% Start butonuna basildiginda calisir.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

set(handles.pushbutton1,'enable','off');
set(handles.pushbutton2,'enable','on');

global stop;
global s;

```

```

global char_array_disp;
global samplingDuration;

% Burada daq cihazini 100 symbol(bit)/s hizinda "analog" veri toplamaya
% ayarliyoruz.
samplingDuration = 0.2; % 200 ms'lik veri paketi
topla
daqreset; % Guncel "device list" icin
daq'i resetle
device = daq.getDevices; % Taninan "daq device"lari
al
s = daq.createSession (device.Vendor.ID); % yeni bir "session" aç
s.Rate = 100; % Ornekleme 100 sym/sec
s.DurationInSeconds = samplingDuration;

s.NotifyWhenDataAvailableExceeds = 20; % 20 data'da bir "listener"
fonksiyonuna git = 200 ms

s.IsContinuous = true; % surekli ADC

% AI1 ve AI3 kanallarını Analog Input kanalı olarak ayarla
[chan1, Idx1] = s.addAnalogInputChannel(device.ID, 1, 'Voltage');
[chan, Idx] = s.addAnalogInputChannel(device.ID, 7, 'Voltage');

% 20 data'da bir "listener" fonksiyonuna git
lh = s.addlistener('DataAvailable',
@ (src,events) collectData (src,events,handles));
s.startBackground(); % "daq device"ı arkaplanda
baslat
stop = false;
tic; % kelime sayacını baslat

while (~stop)

    pause(0.001); % Stop butonuna
basilmadigi sürece
    set(handles.listbox1,'string',char_array_disp); % karakterleri göster

end

end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% Stop butonuna basildiginda calisir
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Stop butonuna basildiginda
set(handles.pushbutton1,'enable','on');
set(handles.pushbutton2,'enable','off');

global stop;
global s;

```

```

stop = true;      % Stop degiskenini "dogru" yap
s.stop();         % "daq device" durdur.

```

```

global plotCounter;
global plotBuffer;
global timeBuffer;
global dataBufferRL;
global dataHoldRL;
global dataBufferUD;
global dataHoldUD;

```

```

% Bufferlari, sayaclari ve bayraklari sifirla fakat ekranda yazılan
kelimeleri silme
plotCounter = 0;
plotBuffer = zeros(2,1000);
timeBuffer = [0:0.01:(10-0.01)];
dataBufferRL = [];
dataHoldRL = 0;
dataBufferUD = [];
dataHoldUD = 0;
global flag_a flag_a1 flag_b flag_c flag_d flag_d1 flag_e flag_f flag_g1
...
    flag_g flag_h flag_i flag_j1 flag_j flag_k flag_l flag_m1 flag_m ...
    flag_n flag_o flag_p1 flag_p flag_r flag_s flag_t1 flag_t flag_u ...
    flag_v flag_y1 flag_y flag_z;

```

```

flag_a1 = 0;
flag_a = 0;
flag_b = 0;
flag_c = 0;
flag_d1 = 0;
flag_d = 0;
flag_e = 0;
flag_f = 0;
flag_g1 = 0;
flag_g = 0;
flag_h = 0;
flag_i = 0;
flag_j1 = 0;
flag_j = 0;
flag_k = 0;
flag_l = 0;
flag_m1 = 0;
flag_m = 0;
flag_n = 0;
flag_o = 0;
flag_p1 = 0;
flag_p = 0;
flag_r = 0;
flag_s = 0;
flag_t1 = 0;
flag_t = 0;

```

```

flag_u = 0;
flag_v = 0;
flag_y1 = 0;
flag_y = 0;
flag_z = 0;

% Grafik limitlerini belirle
handles.myplot1 = plot(handles.axes1,timeBuffer, plotBuffer(1,:));
set(handles.axes1, 'YLim', [-5,5]);
set(handles.axes1, 'XLim', [0,10]);

handles.myplot2 = plot(handles.axes2,timeBuffer, plotBuffer(2,:));
set(handles.axes2, 'YLim', [-5,5]);
set(handles.axes2, 'XLim', [0,10]);

% Update handles structure
guidata(hObject, handles);

end

% --- Executes on button press in pushbutton2.
% Close GUI butonuna basildiginda
function pushbutton4_Callback(hObject, eventdata, handles)

% Butun global ve global olmayan degiskenleri sil

close(handles.figure1);
clearvars;

end

% "listener" fonksiyonu uygulaması
function collectData(src,event,handles)

% ayarlanan "data adeti" dolunca bu fonksiyona girilir.

global plotBuffer;
global timeBuffer;
global plotCounter;
global samplingDuration;

time = event.TimeStamps.';
data = event.Data.'; % Veri vektörünü al

% Listener fonksiyonu ilk 50 kez cagirildiginda (10 sn)
if (plotCounter < 50)

    % plotBuffer degikenini son 20 elemandan itibaren doldur

```

```

plotBuffer = [plotBuffer(:,21:end),data];
set(handles.myplot1, 'XData', timeBuffer, 'YData', plotBuffer(1,:));
set(handles.myplot2, 'XData', timeBuffer, 'YData', plotBuffer(2,:));

findCharacterRL(handles, data(1,:));
findCharacterUD (handles, data(2,:));

plotCounter = plotCounter +1;

else
    plotBuffer = [plotBuffer(:,21:end),data];
    timeBuffer = [timeBuffer(21:end),time];
    set(handles.axes1,
'XLim',get(handles.axes1,'XLim')+samplingDuration);
    set(handles.myplot1, 'XData', timeBuffer, 'YData', plotBuffer(1,:));

    set(handles.axes2,
'XLim',get(handles.axes2,'XLim')+samplingDuration);
    set(handles.myplot2, 'XData', timeBuffer, 'YData', plotBuffer(2,:));

    findCharacterRL(handles, data(1,:));
    findCharacterUD (handles, data(2,:));

end

end

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox1
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
listbox1

end

% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```

% --- Executes on button press in pushbutton3.
% Delete All butonuna basildiginda calisir
function pushbutton3_Callback(hObject, eventdata, handles)
%
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global char_array;
global char_array_disp;

% "Clear all" butonu basildiginda butun "data" matrisleri temizle ve
% yeniden "karakter okuma" konumuna gec
char_array = [];
char_array_disp = [];

% "listbox" 'u temizle
set(handles.listbox1,'string',char_array_disp);

%sayaci yeniden baslat
tic;

end

function findCharacterRL (handles, data)
% Sag-sol göz hareketlerini algılayıp harflere donusturen fonksiyon

global char_data;
global char_array;
global char_array_disp;
global dataBufferRL;
global dataHoldRL;

global ch_a ch_b ch_c ch_d ch_e ch_f ch_g ch_h ch_i ch_j ch_k ch_l ch_m
...
ch_n ch_o ch_p ch_r ch_s ch_t ch_u ch_v ch_y ch_z Fs;

global flag_a flag_a1 flag_b flag_c flag_d flag_d1 flag_e flag_f flag_g1
...
    flag_g flag_h flag_i flag_j1 flag_j flag_k flag_l flag_m1 flag_m ...
    flag_n flag_o flag_p1 flag_p flag_r flag_s flag_t1 flag_t flag_u ...
    flag_v flag_y1 flag_y flag_z;

UpperPosTh = str2double(get(handles.UpperPosThRL,'String'));
LowerPosTh = str2double(get(handles.LowerPosThRL,'String'));
UpperNegTh = str2double(get(handles.UpperNegThRL,'String'));
LowerNegTh = str2double(get(handles.LowerNegThRL,'String'));

PwTh1 = str2double(get(handles.PwTh1RL,'String'));
PwTh2 = str2double(get(handles.PwTh2RL,'String'));
PwTh3 = str2double(get(handles.PwTh3RL,'String'));
PwTh4 = str2double(get(handles.PwTh4RL,'String'));

% UpperPosTh = 3.0;

```

```

% LowerPosTh = 2.0;
% UpperNegTh = -2.0;
% LowerNegTh = -3.0;

% PwTh1 = 25; % 250 msec
% PwTh2 = 45; % 450 msec
% PwTh3 = 65; % 650 msec
% PwTh4 = 85; % 850 msec

% datanın ilk ve son elemanları esik seviyesini altında ve üstünde ise
% darbe oluşmaya başlamıştır. Bu durumda bu darbe dataBufferRL
% değişkenine kopyalanır. Tam bir darbe verisini içeren bu değişken
esikleme
% yapılarak harf üretmek için kullanılır.
if (data(1) < LowerPosTh && data(end) > LowerPosTh)...
    || (data(1) > UpperNegTh && data(end)<UpperNegTh )
    dataHoldRL = 1;

elseif (data(1) > LowerPosTh && data(end) < LowerPosTh)...
    || (data(1) < UpperNegTh && data(end)>UpperNegTh )
    dataHoldRL = 2;
    dataBufferRL = [dataBufferRL, data];

else

end

if (dataHoldRL == 1)
    dataBufferRL = [dataBufferRL, data];
else

end

if (dataHoldRL == 1 || dataHoldRL == 2)

    max_data = max(dataBufferRL);
    min_data = min(dataBufferRL);
    Th_max = max_data/2;
    Th_min = min_data/2;

    if ( max_data>=UpperPosTh)

        pw_data = sum(dataBufferRL>Th_max);

        if (pw_data<PwTh1-5)      % Darbe oluştuğu anda harfi seslendir.
            if (flag_a1==0)
                sound(ch_a,Fs);
                flag_a1 = 1;
            end
            if (dataHoldRL == 2)
                flag_a1 = 0;
            end
        end
    end
end

```

```

end

elseif ( pw_data>=PwTh1-5 && pw_data<PwTh1+5 )
    if (flag_a==0)
        %sound(ch_a,Fs);
        flag_a = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'a';
        char_array = [char_array, char_data];
        flag_a1 = 0;
        flag_a = 0;
    end

elseif ( pw_data>=PwTh1+5 && pw_data<PwTh2+5 )
    if (flag_b==0)
        sound(ch_b,Fs);
        flag_b = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'b';
        char_array = [char_array, char_data];
        flag_a1 = 0;
        flag_a = 0;
        flag_b = 0;
    end

elseif ( pw_data>=PwTh2+5 && pw_data<PwTh3+5 )
    if (flag_c==0)
        sound(ch_c,Fs);
        flag_c = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'c';
        char_array = [char_array, char_data];
        flag_a1 = 0;
        flag_a = 0;
        flag_b = 0;
        flag_c = 0;
    end

elseif ( pw_data>=PwTh3+5 && pw_data<PwTh4+5 )
    if (dataHoldRL == 2)
        char_data = ' ';
        char_array = [char_array, char_data];
        elapsedTime = toc; % bosluk karakteri gelirse sureyi yaz
        set(handles.wordTimer, 'String', num2str(elapsedTime));
        tic; % sayaci yeniden baslat.
    end

end

elseif ( max_data<UpperPosTh && max_data>=LowerPosTh)

    pw_data = sum(dataBufferRL>Th_max);

    if (pw_data<PwTh1-5) % Darbe oluřtuđu anda harfi seslendir.
        if (flag_d1==0)
            sound(ch_d,Fs);

```

```

        flag_d1 = 1;
    end
    if (dataHoldRL == 2)
        flag_d1 = 0;
    end

elseif ( pw_data >=PwTh1-5 && pw_data<PwTh1+5 )
    if (flag_d==0)
        %sound(ch_d,Fs);
        flag_d = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'd';
        char_array = [char_array, char_data];
        flag_d1 = 0;
        flag_d = 0;
    end

elseif ( pw_data >=PwTh1+5 && pw_data<PwTh2+5 )
    if (flag_e==0)
        sound(ch_e,Fs);
        flag_e = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'e';
        char_array = [char_array, char_data];
        flag_d1 = 0;
        flag_d = 0;
        flag_e = 0;
    end

elseif ( pw_data >=PwTh2+5 && pw_data<PwTh3+5 )
    if (flag_f==0)
        sound(ch_f,Fs);
        flag_f = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'f';
        char_array = [char_array, char_data];
        flag_d1 = 0;
        flag_d = 0;
        flag_e = 0;
        flag_f = 0;
    end

elseif ( pw_data>=PwTh3+5 && pw_data<PwTh4+5 )
    % Yeni karakterler icin ayrılmıştır.

end

elseif ( min_data<=LowerNegTh)

    pw_data = sum(dataBufferRL<Th_min);

    if (pw_data<PwTh1-5)      % Darbe oluştuğu anda harfi seslendir.
        if (flag_g1==0)
            sound(ch_g,Fs);
            flag_g1 = 1;

```

```

        end
        if (dataHoldRL == 2)
            flag_g1 = 0;
        end

elseif ( pw_data >=PwTh1-5 && pw_data<PwTh1+5 )
    if (flag_g==0)
        %sound(ch_g,Fs);
        flag_g = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'g';
        char_array = [char_array, char_data];
        flag_g1 = 0;
        flag_g = 0;
    end

elseif ( pw_data >=PwTh1+5 && pw_data<PwTh2+5 )
    if (flag_h==0)
        sound(ch_h,Fs);
        flag_h = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'h';
        char_array = [char_array, char_data];
        flag_g1 = 0;
        flag_g = 0;
        flag_h = 0;
    end

elseif ( pw_data >=PwTh2+5 && pw_data<PwTh3+5 )
    if (flag_i==0)
        sound(ch_i,Fs);
        flag_i = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'i';
        flag_g1 = 0;
        flag_g = 0;
        flag_h = 0;
        flag_i = 0;
    end

elseif ( pw_data>=PwTh3+5 && pw_data<PwTh4+5 )%karakter silme
    if (dataHoldRL == 2)
        char_array = char_array(1:end-1);
    end

end

elseif ( min_data>LowerNegTh && min_data<=UpperNegTh)

    pw_data = sum(dataBufferRL<Th_min);

    if (pw_data<PwTh1-5)        % Darbe oluřtuđu anda harfi seslendir.
        if (flag_j1==0)
            sound(ch_j,Fs);
            flag_j1 = 1;
        end

```

```

        if (dataHoldRL == 2)
            flag_j1 = 0;
        end

elseif ( pw_data >=PwTh1-5 && pw_data<PwTh1+5 )
    if (flag_j==0)
        %sound(ch_j,Fs);
        flag_j = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'j';
        char_array = [char_array, char_data];
        flag_j1 = 0;
        flag_j = 0;
    end
elseif ( pw_data >=PwTh1+5 && pw_data<PwTh2+5 )
    if (flag_k==0)
        sound(ch_k,Fs);
        flag_k = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'k';
        char_array = [char_array, char_data];
        flag_j1 = 0;
        flag_j = 0;
        flag_k = 0;
    end
elseif ( pw_data >=PwTh2+5 && pw_data<PwTh3+5 )
    if (flag_l==0)
        sound(ch_l,Fs);
        flag_l = 1;
    end
    if (dataHoldRL == 2)
        char_data = 'l';
        char_array = [char_array, char_data];
        flag_j1 = 0;
        flag_j = 0;
        flag_k = 0;
        flag_l = 0;
    end
elseif ( pw_data>=PwTh3+5 && pw_data<PwTh4+5 )
    % Yeni karakterler icin ayrilmistir.

end

end

if (dataHoldRL == 2)
    char_array_disp = char_array ;
    dataBufferRL = [];
    dataHoldRL = 0;
end

end

function findCharacterUD (handles, data)

```

```

% Asagi-yukari göz hareketlerini algılayıp harflere donusturen fonksiyon

global char_data;
global char_array;
global char_array_disp;
global dataBufferUD;
global dataHoldUD;

global ch_a ch_b ch_c ch_d ch_e ch_f ch_g ch_h ch_i ch_j ch_k ch_l ch_m
...
ch_n ch_o ch_p ch_r ch_s ch_t ch_u ch_v ch_y ch_z Fs;

global flag_a flag_al flag_b flag_c flag_d flag_dl flag_e flag_f flag_g1
...
    flag_g flag_h flag_i flag_j1 flag_j flag_k flag_l flag_m1 flag_m ...
    flag_n flag_o flag_p1 flag_p flag_r flag_s flag_t1 flag_t flag_u ...
    flag_v flag_y1 flag_y flag_z;

UpperPosTh = str2double(get(handles.UpperPosThUD, 'String'));
LowerPosTh = str2double(get(handles.LowerPosThUD, 'String'));
UpperNegTh = str2double(get(handles.UpperNegThUD, 'String'));
LowerNegTh = str2double(get(handles.LowerNegThUD, 'String'));

PwTh1 = str2double(get(handles.PwTh1UD, 'String'));
PwTh2 = str2double(get(handles.PwTh2UD, 'String'));
PwTh3 = str2double(get(handles.PwTh3UD, 'String'));
PwTh4 = str2double(get(handles.PwTh4UD, 'String'));

% UpperPosTh = 3.0;
% LowerPosTh = 2.0;
% UpperNegTh = -2.0;
% LowerNegTh = -3.0;

% PwTh1 = 25; % 250 msec
% PwTh2 = 45; % 450 msec
% PwTh3 = 65; % 650 msec
% PwTh4 = 65; % 850 msec

if (data(1) < LowerPosTh && data(end) > LowerPosTh)...
    || (data(1) > UpperNegTh && data(end)<UpperNegTh )
    dataHoldUD = 1;

elseif (data(1) > LowerPosTh && data(end) < LowerPosTh)...
    || (data(1) < UpperNegTh && data(end)>UpperNegTh )
    dataHoldUD = 2;
    dataBufferUD = [dataBufferUD, data];

else

end

if (dataHoldUD == 1)
    dataBufferUD = [dataBufferUD, data];
else

end

```

```

if (dataHoldUD == 1 || dataHoldUD == 2)

    max_data = max(dataBufferUD);
    min_data = min(dataBufferUD);
    Th_max = max_data/2;
    Th_min = min_data/2;

    if ( max_data>=UpperPosTh)

        pw_data = sum(dataBufferUD>Th_max);

        if (pw_data<PwTh1-5)      % Darbe oluřtuđu anda harfi seslendir.
            if (flag_m1==0)
                sound(ch_m,Fs);
                flag_m1 = 1;
            end
            if (dataHoldUD == 2)
                flag_m1 = 0;
            end

        elseif ( pw_data>=PwTh1-5 && pw_data<PwTh1+5 )
            if (flag_m==0)
                %sound(ch_m,Fs);
                flag_m = 1;
            end
            if (dataHoldUD == 2)
                char_data = 'm';
                char_array = [char_array, char_data];
                flag_m1 = 0;
                flag_m = 0;
            end

        elseif ( pw_data>=PwTh1+5 && pw_data<PwTh2+5 )
            if (flag_n==0)
                sound(ch_n,Fs);
                flag_n = 1;
            end
            if (dataHoldUD == 2)
                char_data = 'n';
                char_array = [char_array, char_data];
                flag_m1 = 0;
                flag_m = 0;
                flag_n = 0;
            end

        elseif ( pw_data>=PwTh2+5 && pw_data<PwTh3+5 )
            if (flag_o==0)
                sound(ch_o,Fs);
                flag_o = 1;
            end
            if (dataHoldUD == 2)
                char_data = 'o';
                char_array = [char_array, char_data];
                flag_m1 = 0;
                flag_m = 0;
                flag_n = 0;
                flag_o = 0;
            end

        elseif ( pw_data>=PwTh3+5 && pw_data<PwTh4+5 )

```

```

        % Yeni karakterler icin ayrılmıştır.

    end

elseif ( max_data<UpperPosTh && max_data>=LowerPosTh)

    pw_data = sum(dataBufferUD>Th_max);

    if (pw_data<PwTh1-5)      % Darbe oluştuğu anda harfi seslendir.
        if (flag_p1==0)
            sound(ch_p,Fs);
            flag_p1 = 1;
        end
        if (dataHoldUD == 2)
            flag_p1 = 0;
        end

    elseif ( pw_data >=PwTh1-5 && pw_data<PwTh1+5 )
        if (flag_p==0)
            %sound(ch_p,Fs);
            flag_p = 1;
        end
        if (dataHoldUD == 2)
            char_data = 'p';
            char_array = [char_array, char_data];
            flag_p1 = 0;
            flag_p = 0;
        end

    elseif ( pw_data >=PwTh1+5 && pw_data<PwTh2+5 )
        if (flag_r==0)
            sound(ch_r,Fs);
            flag_r = 1;
        end
        if (dataHoldUD == 2)
            char_data = 'r';
            char_array = [char_array, char_data];
            flag_p1 = 0;
            flag_p = 0;
            flag_r = 0;
        end

    elseif ( pw_data >=PwTh2+5 && pw_data<PwTh3+5 )
        if (flag_s==0)
            sound(ch_s,Fs);
            flag_s = 1;
        end
        if (dataHoldUD == 2)
            char_data = 's';
            char_array = [char_array, char_data];
            flag_p1 = 0;
            flag_p = 0;
            flag_r = 0;
            flag_s = 0;
        end

    elseif ( pw_data>PwTh3+5 && pw_data<PwTh4+5 )
        % Yeni karakterler icin ayrılmıştır.

    end
end

```

```

elseif ( min_data<=LowerNegTh)

    pw_data = sum(dataBufferUD<Th_min);

    if (pw_data<PwTh1-5)      % Darbe oluřtuđu anda harfi seslendir.
        if (flag_t1==0)
            sound(ch_t,Fs);
            flag_t1 = 1;
        end
        if (dataHoldUD == 2)
            flag_t1 = 0;
        end

    elseif ( pw_data >=PwTh1-5 && pw_data<PwTh1+5 )
        if (flag_t==0)
            %sound(ch_t,Fs);
            flag_t = 1;
        end
        if (dataHoldUD == 2)
            char_data = 't';
            char_array = [char_array, char_data];
            flag_t1 = 0;
            flag_t = 0;
        end
    elseif ( pw_data >=PwTh1+5 && pw_data<PwTh2+5 )
        if (flag_u==0)
            sound(ch_u,Fs);
            flag_u = 1;
        end
        if (dataHoldUD == 2)
            char_data = 'u';
            char_array = [char_array, char_data];
            flag_t1 = 0;
            flag_t = 0;
            flag_u = 0;
        end
    elseif ( pw_data >=PwTh2+5 && pw_data<PwTh3+5 )
        if (flag_v==0)
            sound(ch_v,Fs);
            flag_v = 1;
        end
        if (dataHoldUD == 2)
            char_data = 'v';
            char_array = [char_array, char_data];
            flag_t1 = 0;
            flag_t = 0;
            flag_u = 0;
            flag_v = 0;
        end
    elseif ( pw_data>=PwTh3+5 && pw_data<PwTh4+5 )
        % Yeni karakterler icin ayrılmıřtır.

    end

elseif ( min_data>LowerNegTh && min_data<=UpperNegTh)

    pw_data = sum(dataBufferUD<Th_min);

    if (pw_data<PwTh1-5)      % Darbe oluřtuđu anda harfi seslendir.

```

```

        if (flag_y1==0)
            sound(ch_y,Fs);
            flag_y1 = 1;
        end
        if (dataHoldUD == 2)
            flag_y1 = 0;
        end

elseif ( pw_data >=PwTh1-5 && pw_data<PwTh1+5 )
    if (flag_y==0)
        %sound(ch_y,Fs);
        flag_y = 1;
    end
    if (dataHoldUD == 2)
        char_data = 'y';
        char_array = [char_array, char_data];
        flag_y1 = 0;
        flag_y = 0;
    end
elseif ( pw_data >=PwTh1+5 && pw_data<PwTh2+5 )
    if (flag_z==0)
        sound(ch_z,Fs);
        flag_z = 1;
    end
    if (dataHoldUD == 2)
        char_data = 'z';
        char_array = [char_array, char_data];
        flag_y1 = 0;
        flag_y = 0;
        flag_z = 0;
    end
elseif ( pw_data >=PwTh2+5 && pw_data<PwTh3+5 )
    % Yeni karakterler icin ayrilmistir.

elseif ( pw_data>PwTh3+5 && pw_data<PwTh4+5 )
    % Yeni karakterler icin ayrilmistir.

end

end

end

if (dataHoldUD == 2)
    char_array_disp = char_array ;
    dataBufferUD = [];
    dataHoldUD = 0;
end

end

end

```