T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ALTI SERBESTLİK DERECESİNE SAHİP BİR ROBOT KOLUNUN SES İLE KONTROLÜ

Musa AYDIN

YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR VE KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Vedat TOPUZ
Yrd. Doç. Dr. Ayça AK

İSTANBUL, 2012

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ALTI SERBESTLİK DERECESİNE SAHİP BİR ROBOT KOLUNUN SES İLE KONTROLÜ

Musa AYDIN
(522110007)

YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR VE KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Vedat TOPUZ
Yrd. Doç. Dr. Ayça AK

İSTANBUL, 2012

TEŞEKKÜR

Yüksek Lisans tez çalışma sürecinde beni yönlendiren, karşılaştığım zorlukları bilgileri ve tecrübeleri ile aşmamda yardımcı olan desteklerini ve yardımlarını hiçbir zaman esirgemeyen değerli danışman hocalarım Yrd. Doç. Dr. Vedat TOPUZ ve Yrd. Doç. Dr. Ayça AK' a en içten teşekkürlerimi sunarım.

Ayrıca her zaman yanımda olan, maddi ve manevi desteklerini hiçbir zaman esirgemeyen aileme teşekkürlerimi, sevgi ve saygılarımı sunarım.

Temmuz, 2012 **Musa AYDIN**

İÇİNDEKİLER

SAYFA NO

TEŞEKKÜR ..... **i**

İÇİNDEKİLER ..... **ii**

ÖZET ..... **iv**

ABSTRACT ..... **v**

SEMBOLLER ..... **vi**

KISALTMALAR ..... **vii**

ŞEKİLLER ..... **viii**

TABLOLAR ..... **x**

BÖLÜM I **1**

I.1 GİRİŞ **1**

I.2 AMAÇ **4**

I.3 LİTERATÜR **5**

BÖLÜM II **7**

II.1 ROBOT KOLU **7**

II.1.1 Robotların Tanımlama Parametreleri 8

II.1.2 Bir Robotu Oluşturan Parçalar **9**

II.1.3 Robotların Eklem Yapıları **9**

II.1.4 Robot Çeşitleri **10**

II.1.4.1 Serbestlik Derecesine Göre Robotlar **10**

II.1.4.2 Kontrol Yöntemlerine Göre Robotlar **15**

II.1.4.3 Kullandıkları Güç Kaynaklarına Göre Robotlar **15**

II.1.4.4 Kesinlik Derecelerine Göre Robotlar **16**

II.1.5 İleri Yön Kinematiği ve Ters Kinematik **16**

II.1.5.1 İleri Yön (Düz) Kinematik **16**

II.1.5.2 Ters Yön Kinematiği **21**

II.2 KONUŞMA TANIMA **23**

II.2.1 Konuşma Tanıma Sistemine Genel Bir Bakış **27**
II.2.2 Ses Tanıma Süreci **28**
II.2.2.1 Sesin Sayısallaştırılması **28**
II.2.2.2 Sesli İfade Tanımada Akustik Analiz ve Öznitelik Çıkarımı 29
II.2.3 Ses Tanımada Sınıflandırma Süreci **35**
II.2.3.1 Saklı Markov Modeli (Hidden Markov Model - HMM) **35**
II.2.3.2 Yapay Sinir Ağları **35**
II.2.3.3 Dinamik Zaman Bükmesi (Dynamic Time Warping - DTW) **36**
BÖLÜM III 37
III.1 EASYVR SES TANIMA KARTI 38
III.2 MSP 430 MİKRODENETLEYİCİ 42
III.2.1 MSP430G2231 ve MSP430G2553 Mikrodenetleyicisi **43**
III.2.1.1 LaunchPad ve Üzerindeki Bağlantılar **44**
III.2.2 Code Composer Studio Programlama editörü 45
III.3 MİCRO MAESTRO 6 KANAL SERVO MOTOR SÜRÜCÜ 46
III.4 ROBOTİK KOL 48
III.4.1 Servo Motorlar 50
BÖLÜM IV 52
DENEYSEL ÇALIŞMA 52
BÖLÜM V 58
SONUÇLAR VE TARTIŞMA 58
BÖLÜM VI 60
DEĞERLENDİRMELER VE ÖNERİLER 60
KAYNAKLAR 61
EKLER 64
EK 1. MSP430 MİKRODENETLEYİCİ VE EASYVR ARASINDAKİ İLETİŞİM PROGRAMI 64
EK 2. MSP430 VE MOTOR SÜRÜCÜ KARTI ARASINDAKİ İLETİŞİM PROGRAMI 71
ÖZGEÇMİŞ 82

ÖZET

ALTI SERBESTLİK DERECESİNE SAHİP BİR ROBOT KOLUNUN SES İLE KONTROLÜ

Günümüz bilgisayar teknolojisinin gelişmesi ile birlikte, insan makine etkileşimi artmıştır. İnsanların makinelerle olan ilişkisinde, ses komutu ile bir sistemin kontrol edilmesi, oldukça popüler uygulamalar arasında yer almaktadır.

Bu tez çalışmasında, beş serbestlik dereceli, beş dönel ekleme ve bir uç işlevcisine sahip bir robot kolunun eklem hareketlerini ses ile kontrol edebilen bir sistemin oluşturulması amaçlanmıştır.

Ses ile robotik kolun kontrolü aşamasında, öncelikle robot kolun kontrolünde kullanılması düşünülen komutlar, EasyVr ses işleme kartı üzerinde tanımlanmıştır. Robot kolun kontrolünde EasyVr ses modülünde 17 adet komut tanımlanmıştır. EasyVr ses modülünde bir komut tanımlamak için, ses tanıma kartının bilgisayar ortamındaki grafiksel ara yüz programı kullanılmıştır. Komutlar tanımlandıktan sonra, kullanıcıdan alınan ses bilgisi ses tanıma kartında işlenmiş ve daha önceden tanımlanmış olan ses bilgisi ile karşılaştırılmıştır. Karşılaştırma sonucunda herhangi bir eşleşme gerçekleşmiş ise bu bilgi ses tanıma kartı ve motor sürücü kartı arasındaki kontrolü sağlayan bir mikrodenetleyiciye gönderilmiştir. Mikrodenetleyici kendisine gelen bilgi doğrusunda robot kolun hareketi için gerekli olan açı bilgisini, servo motor sürücü kartına göndermiş ve robot kol istenilen hareketi gerçekleştirmiştir.

Temmuz, 2012 **Musa AYDIN**

ABSTRACT

CONTROLLING OF A SIX AXIS ROBOTIC ARM WITH VOICE

The interaction between human beings and machines has been increasing in conjuction with the development in computer technology. Controlling a system with Voice based commands is one of the most popular application in this area. In this thesis the main goal is to build a system which is able to control an robotic arm comprised of six controllable axes with certain voice commands.

The process of controlling the robotic arm starts with matching of sounds taken from the user and sounds which were processed in voice recognition card and were defined by voice recognition card. After first process if there is a positive matching a data is sent to a microcontroller which provides a communication between voice recognition module and servo motor drive card by the system. Finally the microcontroller calculates the required angles by using the data provided by previous process and sends this data to the servo motor drive card in order to realize the robotic arm action.

July, 2012 **Musa AYDIN**

SEMBOLLER

$^{i-1}_{i}T$: Komşu iki eklem arasındaki ilişki bir dönüşüm matrisi.

a_{i-1} : İki eksen arasındaki uzuv uzunluğu.

α_{i-1} : İki komşu eksen arasındaki eksen açısı.

d_i : Üst üste çıkan bağlar arasındaki eklem kaçıklığı.

θ_{i-1} : İki komşu uzuv arasındaki eklem açısı.

c : Robot kolun uç noktası ile başlangıç ekseni arasındaki mesafe.

w : Açısal frekans (rad/san)

c_j : Kepstral katsayılar.

N : Filtre bankasındaki kanal sayısı.

A : Durum geçişi olasılık dağılımları matrisi.

P_i : Mikrodenetleyici port numarası.

KISALTMALAR

TC **:** Train Command

PH **:** Phase

DCT **:** Discrete Cosine Transform

DTW **:** Dynamic Time Warping

FFT **:** Fast Fourier Transform

MFCC **:** Mel-Frequency Cepstral Coefficients

HMM **:** Hidden Markov Model

I2C **:** İnter-Integrated Circuit

SPI **:** Serial Peripheral Interface

USART **:** Universal Asynchronous Receiver/Transmitter

ADC **:** Analog Digital Convertor.

RISC **:** Reduced Instruction Set Computing.

TTL **:** Transistor Transistor Logic.

USB **:** Universal Serial Bus.

PWM **:** Pulse Width Modulation.

ŞEKİLLER

SAYFA NO

Şekil II. 1 Robot kolu 8
Şekil II. 2 Döner Eklemler..... 10
Şekil II. 3 Kayar Eklemler..... 10
Şekil II. 4 Prensip yapıları, kinematik yapıları ve çalışma alanları ile robot çeşitleri [21] 11
Şekil II. 5 Kartezyen Robot Kolun Çalışma Alanı [16] 12
Şekil II. 6 Silindirik Robot Kolun Çalışma Alanı [16]..... 13
Şekil II. 7 Scara Robot Kolun Hareket Alanı [16] 14
Şekil II. 8 Küresel robot kolun çalışma alanı [16]. 14
Şekil II. 9 Mafsallı kol konfigürasyonlu robotlar [22]. 15
Şekil II. 10 Dönme Eksenlerinin Belirlenmesi..... 17
Şekil II. 11 Dönme Eksenlerine Z,Y ve X Eksenlerinin Yerleştirilmesi..... 18
Şekil II. 12 $\boldsymbol{ai-1}$ Uzuv uzunluğu 18
Şekil II. 13 $\boldsymbol{di}$ Eklem Kaçıklığı..... 19
Şekil II. 14 $\boldsymbol{\alpha i-1}$ Eksen Açısı..... 19
Şekil II. 15 $\boldsymbol{\theta i}$ Eklem Açısı 20
Şekil II. 16 Ters Kinematik Çift Çözüm 22
Şekil II. 17 Eklem Açılarının Çözümü..... 22
Şekil II. 18 Konuşma ve işitme süreçlerine ilişkin şematik yapı. 25
Şekil II. 19 İnsanın ses mekanizmasının şematik görünümü 25
Şekil II. 20 Ses oluşumunun lineer modeli..... 26
Şekil II. 21 İnsan kulağının şematik görünümü. 27
Şekil II. 22 Örnek Bir Ses Tanıma Sistemi Modeli..... 28
Şekil II. 23 Analog Sinyalin Sayısallaştırma Süreci 29
Şekil II. 24 Akustik analiz için genel blok diyagramı..... 30
Şekil II. 25 Bir sesli ifadenin dalga şekli: orijinal ses sinyali(mavi renkli) ve 1. dereceden yüksek geçirgen filtre sonrası(kırmızı renkli)..... 31
Şekil II. 26 Pencereleme işleminin bir çerçevedeki sinyal üzerinde gösterimi..... 32

Şekil II. 27 Davis ve Mermelstein tarafından Fourier tranformunu Mel skalasına dönüştürmek için (1980 yılında) ileri sürülen tipte üçgen filtreler 33
Şekil III. 1 Robotik bir sistemin ses ile kontrolünün akış diyagramı 37
Şekil III. 2 EasyVr Ses tanıma kartı 39
Şekil III. 3 EasyVr modülünün bilgisayar ara yüz programı 40
Şekil III. 4 Komut tanımlama(eğitme) ekranı 41
Şekil III. 5 EasyVr ses tanıma kartı ve MSP430 Mikrodenetleyici arasındaki haberleşme 41
Şekil III. 6 Msp430 Mikrodenetleyici ve diğer donanım birimleri arasındaki bağlantı. 42
Şekil III. 7 MSP430G2231 genel yapısı 43
Şekil III. 8 LaunhPad ve üzerindeki bağlantılar. 44
Şekil III. 9 Code Composer Studio çalışma ekranı 46
Şekil III. 10 Micro Maestro 6 kanal servo motor sürücü kartı 47
Şekil III. 11 Micro Maestro 6 kanal Servo Motor Sürücü kartı ve MSP430G2231 Mikrodenetleyici arasındaki Haberleşme. 48
Şekil III. 12 Tez çalışmasında kullanılan robotik kol 49
Şekil III. 13 İleri yön kinematik hesabı simülasyonu 50
Şekil III. 14 0,5ms ‘lik modülasyon genişliği ve motor milinin aldığı açı değeri 51
Şekil III. 15 1,5ms ‘lik modülasyon genişliği ve motor milinin aldığı açı değeri 51
Şekil III. 16 2,5ms ‘lik modülasyon genişliği ve motor milinin aldığı açı değeri 51
Şekil IV. 1 EasyVr ses tanımlama kartı ara yüz programı tanımlanan komut listesi 53
Şekil IV. 2 Robot kolu deney düzeneği 54
Şekil IV. 3 4 eksenin hareketi sonrasındaki robot kolun geldiği konumu 55
Şekil IV. 4 2 eksenin hareketi sonrasındaki robot kolun konumu 56
Şekil IV. 5 2 eksenin hareketi sonrasındaki robot kolun konumu 57

TABLOLAR

SAYFA NO

Tablo II. 1 D-H Değişkenleri 21
Tablo III. 1 MSP430G2231, MSP430G2553 Mikrodenetleyici Özellikleri [29] 43
Tablo III. 2 Mikrodenetleyiciden motor sürcü kartına gönderilen paket yapısı 48
Tablo IV. 1 Kullanılan komutların listesi. 52
Tablo IV. 2 Robot kolun ses ile kontrolünde komutların söyleniş sırası 53
Tablo IV. 3 4 eksenin hareketi için gerekli komutlar 55
Tablo IV. 4 2 eksenin hareketi için gerekli komutlar 56
Tablo IV. 5 3 eksenin hareketi için gerekli komutlar 56
Tablo V. 1 Test Sonuçları 59

BÖLÜM I

GİRİŞ, AMAÇ VE LİTERATÜR

I.1 GİRİŞ

“Robotik” kelimesi ilk olarak 1941 yılında, bilimkurgu yazarı Isaac Asimov tarafından kullanılmıştır. Isaac Asimov, bu bilimin ileride çok daha fazla gelişeceği ve güçlü bir robot endüstrisinin doğacağı öngörüsünde bulunmuştur. Şu ana kadar gerçekleşen gelişmeler göz önünde bulundurulduğunda özellikle endüstrideki robot kullanımında yaşanan patlama, bu öngörünün oldukça başarılı olduğunu göstermektedir. Bütün bu gelişmeler ışığında “Robotik” kelimesi, günlük yaşamda da yerini alarak robotlarla ilgili her tür teknolojiyi kapsayan bir kelime haline gelmiştir.

Amerikan Robot Enstitüsü, robot kavramını şu şekilde ifade etmektedir: "Robot, çeşitli görevlerin gerçekleştirilmesi için, malzeme, parça, takım ya da değişken programlanmış hareketler aracılığıyla, özel parçaları hareket ettirmek amaçlı tasarlanmış, çok fonksiyonlu, yeniden programlanabilir manipülatördür". Robotun her bir ekleminin hareketi motorlarla gerçekleştirilir. Bu durum, robotun verilen bir görevi gerçekleştirmesinin kontrolünü sağlar.

İnsan hayatını kolaylaştırmak ve yeni imkânlar sunmak amacıyla günden güne artan bilgisayar teknolojisindeki araştırmaların büyük bir kısmı, insan müdahalesi olmadan kendi kendine hareket edebilen akıllı makineler geliştirmekle ilgilidir. Bu araştırmalar arasında yer alan görüntü ve ses işleme algoritmalarının robot gibi akıllı makinelerle birlikte kullanılması, insan yaşamını daha da kolaylaştırdığı için günümüz teknolojik gelişmeleri arasında önemli bir yer tutmuştur. Robotlar, endüstri, sağlık, savunma sanayi ve haberleşme olmak üzere birçok sektörde çeşitli

görevleri yapmak için kullanılmaktadır. Robotlar doğrudan bir operatörün kontrolünde çalışabildikleri gibi bağımsız olarak bir bilgisayar programının kontrolünde de çalışabilir. Robot deyince insan benzeri makineler akla gelse de robotların çok azı insana benzer. Günümüzde robotların en büyük kullanım alanı endüstriyel üretimdir. Özellikle otomotiv endüstrisinde çok sayıda robot kullanılır. Bunların çoğu kol şeklindeki robotlardır. Bunlar parçaları monte eden, birleştiren, kaynak ve boya yapan robotlardır [2].

İnsanlar fiziksel yapılarından dolayı bedensel olarak bütün işleri yapma imkânına sahip olmadıkları için, gücünün yetmediği yerlerde kullanmak üzere değişik makineler geliştirmiştir. İlk önceleri insan yardımı ile çalışan bu makineler, zamanla geliştirilerek ve çeşitli çevre birimlerini de beraberinde kullanarak insana ihtiyaç duymadan otomatik olarak çalışır hale getirilmiştir [1].

Otomasyon uygulamalarının görünen yüzü endüstriyel robotlardır. Günümüzde robotlar laboratuar uygulamalarından uzay çalışmalarına, endüstriden hizmet sektörüne kadar pek çok alanda kullanılmaktadır. Robotlar genellikle, üretim maliyetini düşürmek ve daha kaliteli üretim yapmak için kullanılmaktadır. Ayrıca insan sağlığının zarar görme riskinin olduğu işlerde ve insan elinin ulaşamayacağı yerlerde robotlar kullanılmaktadır. Bir üretim robotunun yapısının anlaşılabilmesi için iki ana unsurun bilinmesi gerekmektedir. Bu unsurların ilki hareket mekanizmasıdır. Pozisyon kontrolü otomasyon sistemlerinde çok yaygın olarak kullanılmaktadır.

Konuşma insanların en önemli iletişim yoludur, dolayısıyla ses tanıma sistemleri insan-makine arası iletişimi hızlı ve etkin bir şekilde gerçekleştirmeyi amaçlar. Bu amaçla, ses tanıma teknolojilerini geliştirmek için son yıllarda çeşitli çalışmalar yapılmaktadır [3]. Konuşma tanıma alanındaki çalışmalardan bazıları; web sayfalarında dolaşımı sağlayabilecek, klavyeye dokunmadan e-posta veya metin yazmaya imkan sağlayacak, programları açma kapamayı sağlayabilecek, haber veya bilgi içerikli internet sayfalarının özürlü insanlar tarafından takip edilebilmesine imkan sağlayabilecek uygulamaları içeren insan-bilgisayar iletişimi [4,5,6], konuşma tanıma sistemlerinin eğitim alanında kullanılması [7], robotların sesle kontrolü [8], çocukların dil gelişiminde otomatik dil tanıma sistemlerinin kullanılması [9] gibi uygulamalar olarak gösterilebilir. Konuşma tanıma sistemleri sayesinde telefonda kredi kartı numarası tuşlamak yerine numarayı sesli söyleyebiliyor veya arabada sesli komutlarla bilgiye ulaşılabilmektedir [10].

Sesi işlemlerde ara yüz olarak kullanmak yapay zekâdaki gelişmelerle birlikte daha fazla önem kazanmıştır [11]. Yapay Zekâ (Artificial Intelligence), Makine Öğrenmesi (Machine Learning), matematik, istatistik, kavrama (Cognition) ve dilbilimi (Linguistics) gibi farklı disiplinlerin iç içe geçmiş olarak kullanıldığı Konuşma Tanıma Teknolojisi (KTT), bahsi geçen disiplinlerdeki önemli gelişmelerle birlikte, elektronikteki ilerlemelerden (mikrofon, ses kartı teknolojileri, işlemci hızları v.b.) de önemli ölçüde yararlanmıştır [12,13,14]. Dağıtık sistemlerin de gelişmesiyle beraber yerel alan ağlarında veya internette ses işleme modülleri sıklıkla kullanım alanı bulurken Dağıtık Konuşma Tanıma (DKT) modülleri kişisel bilgisayarlara, cep telefonlarına ve avuç içi bilgisayarlara entegre edilmektedir [15].

İnsanların kendi aralarında iletişim kurarken kullandıkları doğal yolun, yani konuşarak iletişim kurabilmenin insan-makine etkileşimine taşınması, insanların makineleri daha basit olarak kullanılmasını mümkün kılmıştır. Geliştirilen konuşma tanıma sistemlerinin bilgisayarlara veya elektro-mekanik cihazlara entegrasyonu ile bu cihazlara konuşarak komut göndermek mümkün hale gelmiştir. Örneğin, bu sistemler sayesinde bilgisayardaki bir kelime işlemci programında klavyeyi kullanmadan konuşarak yazı yazmak veya bir makineyi kontrol etmek mümkün hale gelmiştir. Konuşma tanıma sistemleri ayrıca, zaman ve emek tasarrufu sağlaması açısından da önem arz etmektedir.

I.2 AMAÇ

Günümüzde görsel teknolojilerin olduğu kadar artık ses teknolojilerin de kullanıldığı alanlar artmakta ve bu teknolojide yaşanan her türlü gelişmeler devam etmektedir. Başlangıçta sadece makineyi konuşturmaya yönelik çalışmalar yapılırken, modern anlamda ilk bilgisayarların ortaya çıkısından itibaren bu çalışmalar da giderek insan makine etkileşimi yönüne doğru kaymış ve *"konuşma tanıma"* (speech recognition) teknolojisi doğmuştur. Yapay Zekâ (Artificial Intelligence), Makine Öğrenmesi (Machine Learning), Matematik, İstatistik, Kavrama (Cognition) ve Dilbilimi (Linguistics) gibi farklı disiplinlerin iç içe geçmiş olarak kullanıldığı Konuşma Tanıma teknolojisi, bahsi geçen disiplinlerdeki hatırı sayılır gelişmelerle birlikte, elektronikteki ilerlemelerden (mikrofon, ses kartı teknolojileri, işlemci hızları v.b.) de önemli ölçüde yararlanmıştır.

Bilgisayarlar, insan sesini tanıyabilmekte, belirlenen komutları anlayabilmekte, kendi kelime haznelerindeki (yüzlerce, hatta binlerce) sözcüklerden oluşan cümleleri dikte edebilmekte ve kısmen de olsa, kurulan cümlelerdeki anlamları algılayabilmektedir. Ayrıca bu gelişmelere göre bilgisayarlara ses ile komutlar verebilmekte, evde çeşitli elektronik eşyalara bağlanan bir bilgisayar tarafından tüm ev ışıkları, klima ve televizyon kumanda edilebilmektedir.

Bu tez çalışmasında bir robotik sistemin çalışması sesli komutlar ile gerçekleştirilmiştir. Öncelikli olarak sesli komutlar mikrofonla alındıktan sonra yazılımsal/donanımsal konuşma tanıma işlemleri gerçekleştirilmiştir ve sistemin anlayacağı forma dönüştürülmüştür. Bu işlemden sonra kullanılan robotik sistemin hareketi için gerekli olan açısal bilgiler gönderilmiştir ve robotik kolun hareketi ses komutu ile sağlanmıştır.

Bölüm II' de konuşma tanıma sistemleri ve robot kollara ilişkin genel yaklaşım verilmiştir. Daha sonra sesli ifade tanıma sürecinin aşamaları, robot kolların çeşitleri ve çalışma yapıları açıklanmıştır.

Bölüm III' te yapılan tez çalışmasında kullanılan donanımsal bileşenlerin çalışması ile ilgili bilgiler verilmiştir. Bu donanımsal bileşenleri birbirleri ile olan bağlantısına değinilmiştir. Daha sonra yapılan deneysel çalışma anlatılmıştır.

Bölüm IV' te bu tezde yapılan çalışma değerlendiriliştir.

I.3 LİTERATÜR

Peter, X “Voice Based Robot Control” isimli çalışmada mobil bir robotun kontrolünü yapmak için, ses komutlarını kullanan bir sistem geliştirilmiştir. Sistem 2 ana bölümden oluşmaktadır. Konuşma tanıma motoru ve Robotik kontrol ünitesi. Konuşma tanımlama motorunda, ses tanımlamada ana metot olarak Saklı Markov Modeli (Hidden Markov Model -HMM) kullanılmıştır. Robotik kontrol ünitesi, robot üreticisi tarafından sunulan bir ara yüz altında geliştirilmiştir [26].

M, Z, Özdermircan “Robot Control with Voice Command” isimli yüksek lisans çalışmasında sesli komut ile robot kontrolü gerçekleştirilmiştir. Sesli komutlar mikrofonla bilgisayar ortamına alınmış, Mel Frequency Cepstral Coefficients (MFCC) yöntemi kullanılarak bu sesli komutların özellikleri çıkartılmış ve Yapay Sinir Ağları kullanılarak komut tanınmıştır. Son olarak tanınan bu komutlar robotun anlayacağı forma dönüştürüp robota gönderilmiş ve robotun hareketi sağlanmıştır [30].

F, Baltacıoğlu “Uzaktan Sesle Kontrol Edilebilen Robot Tasarımı”, çalışmasında bir robotun sesle ve uzaktan kablosuz olarak kontrol edilmesini amaçlamıştır. Kablosuz robot, kontrolcüsünden uzakta hareket edeceği için, robota kameradaki görüntüyü aktarma özelliği eklenmiştir. Görüntü aktarımı ve kontrol uygulaması bilgisayar kullanılarak gerçekleştirilmiştir. Hem klavye kullanılarak, hem de ses ile yön kontrolü yapılabilmektedir. Kontrolün ses komutları ile yapılmasıyla daha hızlı, hatasız ve güvenilir bir kontrol işlemi gerçekleştirmek hedeflenmektedir. Tasarlanan ve gerçekleştirilen sistem, üç ana bölümden oluşmaktadır: kullanıcı kontrol yazılımı, haberleşme kartı ve mikrodenetleyici kontrol kartı. Haberleşme için RF kablosuz iletişim yöntemi kullanılmış ve görüntü aktarımı için IP kamera robot üzerine monte edilmiştir. Yapılan bu çalışma sayesinde robot mayınlı arazide, enkaz üzerinde kısacası insan sağlığını veya güvenliğini olumsuz etkileyebilecek, insanların ulaşmasında sakınca bulunan bölgelerin keşfedilmesinde kullanılabilecektir. Sonuç olarak insan sağlığı ve güvenliği için tehlikeli bölgelere erişimde kablosuz kontrol edilebilen robotların kullanılabilirliği gösterilmekte ve robotun ses ile kontrol edilebilmesi bir özellik olarak eklenmektedir. Robota eklenebilecek

mekanik/donanımsal özellikler yardımıyla robotun hareket kabiliyeti arttırılarak kullanım alanının genişletilmesi mümkün gözükmektedir [31].

M, Nishimori “Voice Controlled Intelligent Wheelchair”, isimli çalışmasında, fiziksel engelli kişilere yardımcı olmak amacıyla, bir ses kontrollü tekerlekli sandalye geliştirmiştir. Kullanıcı ses komutlarını kullanarak tekerlekli sandalyeyi kontrol edebilmektedir. Bu işlemi gerçekleştirmek için bir adet laptop bilgisayar kullanılmaktadır. Ses komutları kullanıcıya ekranda gösterilmektedir. Komutların yanlış eşleştirilmesini önlemek için kullanıcıdan komutları onaylaması istenmektedir. Kontrol sinyali bilgisayardan mikrodenetleyiciye gönderilir ve mikrodenetleyici tekerlekli sandalyedeki, tekerlere bağlı motorları sürmek için bir kontrol sinyali üretir. Geliştirilen sistemin yazılımı için Kyoto Üniversitesi tarafından geliştirilen “Julian” adlı açık kaynak kodlu yazılım kullanılmıştır [32].

K, Ishikawa “Development of self-propelled Robot that Recognizes Voice Instruction” isimli çalışmasında, ses komutlara göre hareket eden ve engelden kaçınan bir robot geliştirmiştir. Ses komutları olarak ileri git, geri git, sağa dön, sola dön, fonksiyon değiştir komutları kullanılmıştır. Öznitelik çıkarma Levinson Durbin’s algoritması ve LPC katsayıları kullanılmıştır. Bu yöntem ile ses komutlarının tepe frekansları her ses komutu için belirlenir. Test işleminde ise bu elde edilen tepe frekanslarına benzerlik oranına göre karar verilir. Bilgisayar ortamında alınan ses bilgileri, öznitelik çıkartma ve karar verme süreçlerine tabi tutulur. Karar bilgisi ise radyo frekansı ile robota iletilir [33].

Kelebekler, E “*Otomobil İçindeki Cihazların Sesle Kontrolüne Yönelik Konuşma Tanıma Sisteminin Gerçek Zamanlı Laboratuvar Uygulaması*” isimli çalışmasında, konuşmacıdan bağımsız konuşma tanıma sisteminin gerçek zamanlı uygulamasının yapılması amaçlanmıştır. Konuşma tanıma siteminde kullanılan veri tabanı için 10 kadın 10 erkek konuşmacının telaffuzları kullanılarak oluşturulmuştur. Sitemin gerçek zamanlı laboratuar uygulaması MATLAB 6.5 sürümü ile gerçekleştirilmiştir [34].

BÖLÜM II

ROBOT KOLU VE KONUŞMA TANIMA

Bu bölümde robot kollar ve konuşma tanıma konularına değinilmiştir. İlk olarak robot kolların genel yapısı, çeşitleri, kullanım alanları, eklem yapıları, ileri yön kinematiği ve ters yön kinematiği hesaplamaları anlatılmıştır. Daha sonra konuşma tanıma süreci, sesin sayısallaştırılması, sesten öznitelik çıkarımı ve sesin sınıflandırılması konularına değinilmiştir.

II.1 ROBOT KOLU

Robot kolu; yapacağı işe göre dönel ya da bulunduğu eksende düz hareket yapabilen eklemlere sahip, tutma mekanizmaları olan, mikroişlemci, mikrodenetleyici veya diğer bilgi işlem elemanları ile kontrol edilebilen elektromekanik bir sistemdir. Eklem hareketleri, pnömatik veya hidrolik sistemlerle veya elektrik motorları ile sağlanır. Robot kolları, endüstriyel manipülatör olarak da tanımlanabilir. Manipülatör; eklemlerle birbirine bağlanmış sert veya katı cisimlerin açık uçlu kinematik zinciri olarak kabul edilir. Sekil II.1’de örnek bir endüstriyel manipülatör (robot kolu) verilmiştir.



Şekil II. 1 Robot kolu

Robot kollarının üretim bantlarında yer almaya başlaması ile insan hataları azalarak, üretim zamanının kısalması ve üretim performansının artması sağlanmış, daha az çalışan sayısı ile daha fazla üretimin sağlanabilmesi robot kollarının geliştirilmesine duyulan ihtiyacı artırmıştır. Endüstride kullanılan robotları değişik şekillerde sınıflandırmak mümkündür [16].

II.1.1 Robotların Tanımlama Parametreleri

Robotlar tanımlanırken belli başlı bazı parametrelerden yararlanılmaktadır. Bunlardan bazıları aşağıdaki gibidir [17].

1. Eksen Sayısı: Bir düzlemde herhangi bir noktaya ulaşmak için iki eksen gereklidir; uzayda herhangi bir noktaya ulaşmak için üç eksen gereklidir. Robotun uç elemanının yön kontrolü için üç ya da daha fazla eksene ihtiyaç vardır.
2. Serbestlik Derecesi: Robotların serbestlik derecesi genellikle eksenler ile aynı sayıdadır. Robotun ucunun uzayda istenilen pozisyonda konumlanması

için üç, istenilen yönde konumlanması için ise üç yani toplam altı serbestlik derecesine ihtiyaç vardır.

3. Çalışma Alanı: Robotun ulaşabildiği her noktanın oluşturduğu bölgedir.

4. Kinematik: Robotun yapabileceği hareketleri belirleyen, robot üzerindeki katı cisimlerin ve eklemlerin düzenlemesi, sıralaması.

5. Taşıma Kapasitesi: Robotun taşıyabileceği maksimum yük.

6. Hız: Robotun ucunu konumlandırdığı maksimum hızdır.

7. İvme: Robotun bir ekleminin maksimum ivmesidir.

8. Doğruluk: Robotun konumlanması istenilen noktaya ne kadar yaklaşabildiğini gösterir.

9. Tekrarlama Kabiliyeti: Robotun daha önce ulaştığı bir noktaya dönebilmesi ile ilgili ölçüdür.

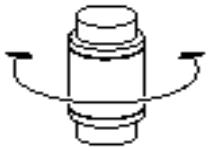
II.1.2 Bir Robotu Oluşturan Parçalar

Bir robot dört ana kısımdan oluşur [18]. Bir mekanik yapı ya da eklemlerle birbirine bağlanmış sıralı katı cisimlerden (uzuvlardan) oluşan manipülatör; serbestliği sağlayan bir kol, el becerisi sağlayan bir bilek ve robotun yapması gereken görevi tamamlayan uç elemanından oluşmaktadır. Eklemlerin hareketlenmesiyle manipülatörün hareketini sağlayan hareketlendiriciler için genellikle elektrikli motorlar, hidrolik ve pnömatik devindiriciler kullanılır. Manipülatörün durumunu gözleyen algılayıcılar ve gerekirse çevrenin durumunu gözleyen algılayıcılar, manipülatör hareketini denetleyen ve yöneten bir denetim sistemi oluşturur.

II.1.3 Robotların Eklem Yapıları

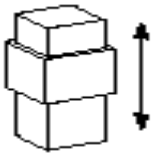
Robotlar genelde bir küme eklemin birbirine bağladığı bir küme rijit kol içerir. Robot kollarını birbirine bağlamak için genelde döner, kayar, helikal, silindirik, küresel ve düzlemsel eklemler kullanılır. Robotikte en çok kullanılanlar, döner ve kayar eklemlerdir [19].

1. Döner (Revolute – R) Eklemler: Şekil II.2’de görüldüğü gibi iki uzuv arasında dönme hareketini sağlayan eklemlerdir.



Şekil II. 2 Döner Eklemler

2. Kayar (Prismatic – P) Eklemler: İki uzuv arasında doğrusal hareketi sağlayan eklemlerdir. Şekil II.3'de bu eklemlerin genel görünümü verilmiştir.



Şekil II. 3 Kayar Eklemler

II.1.4 Robot Çeşitleri

Robotlar, genel itibariyle seri ve paralel olmak üzere iki temel gruba ayrılır. Bir dizi eklemlerden ve bu eklemleri birbirine birleştiren bağlardan oluşan seri robotlar, geniş çalışma uzayına, az sayıda mekanik parçaya ve paralel robota göre daha basit kinematik denklemlere sahiptirler. Kendi ağırlıkları taşıdıkları ağırlıklara oranla çok büyüktür. Paralel robotlarda ise iki eklem birbirine birden çok bağ ile bağlanır. Bu özellik sayesinde paralel robotlar kendi ağırlıklarına oranla çok fazla yük taşıyabilirler. Fakat mekanik yapılarının karmaşık olması sebebiyle kinematik problemleri de seri robotlara göre oldukça karmaşıktır [20].

Robotlar bu iki temel gruplandırma dışında, serbestlik derecelerine, kontrol yöntemlerine, kullandıkları güç kaynaklarına göre robotlar ve kesinlik derecelerine göre de sınıflandırılırlar.

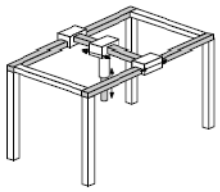
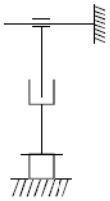
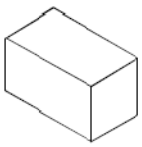
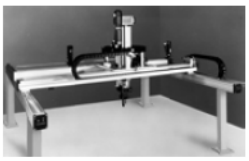
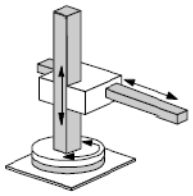
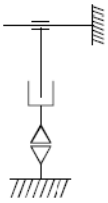
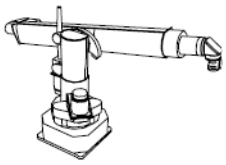
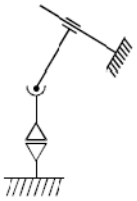
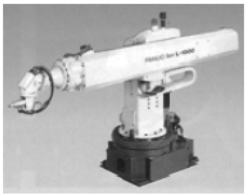
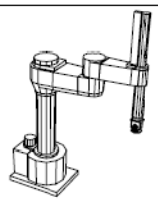
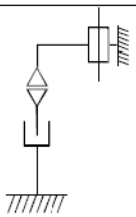
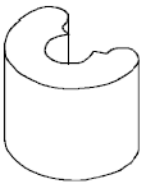
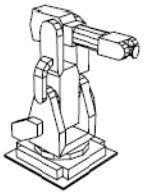
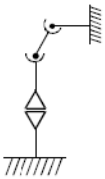
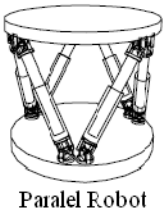
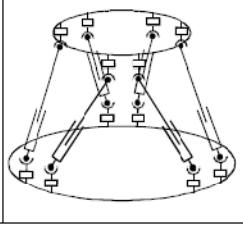
II.1.4.1 Serbestlik Derecesine Göre Robotlar

Robotlar serbestlik derecesine göre ilk üç ekleminin yapısı göz önünde bulundurularak aşağıdaki gibi sınıflandırılırlar:

1. Kartezyen robot kolları,
2. Silindirik robot kolları,
3. SCARA (Selective Compliant Assembly Robot Arm) konfigürasyonlu robot kolları,

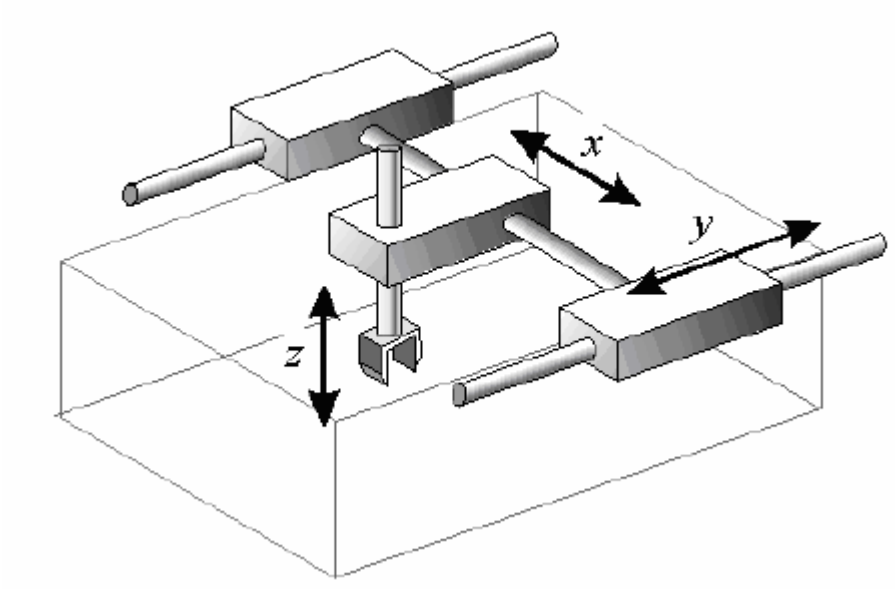
4. Küresel robot kolları,
5. Mafsallı robot kollar.

Şekil II.4’ de prensip yapıları, kinematik yapıları, çalışma alanları ve örnek fotoğraflarıyla birlikte temel robot örnekleri gösterilmektedir.

ROBOT	EKSENLER		ÖRNEK
Prensip Yapı	Kinematik Yapı	Çalışma alanı	Örnek Fotoğraf
 Kartezyen Robot			
 Silindirik Robot			
 Küresel Robot			
 SCARA Robot			
 Mafsallı Robot			
 Paralel Robot			

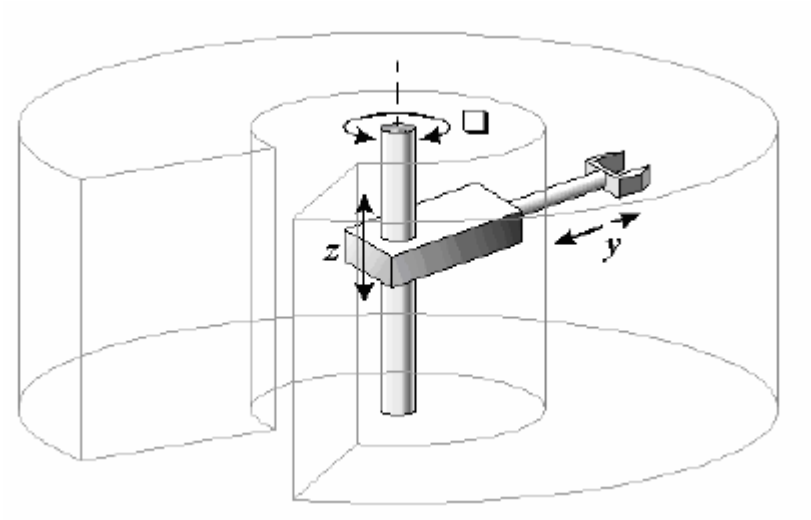
Şekil II. 4 Prensip yapıları, kinematik yapıları ve çalışma alanları ile robot çeşitleri [21]

Kartezyen Robot Kolları: Sadece tutma ve taşıma yeteneği olan bu robot tipi X,Y,Z, eksenlerinde doğrusal olarak hareket etme yeteneğine sahiptirler. Basit bir yapıya sahip oldukları için hareketlerin planlanması çok kolaydır. Bu tür robotlarda; pozisyon hesaplamaları, robot uç elemanının bulunduğu pozisyon, mafsalların o anda olduğu yerde bulunduğundan çok kolaydır. Çalışma alanları Şekil II.5' de görüldüğü gibi robotun yapısından daha küçüktür. Eğilme ve bükülme işlemlerini gerçekleştiremez. Çalışma alanları kare veya dikdörtgen pirizma şeklindedir. Yük taşıma kapasitesi diğer robot türlerine göre daha büyüktür. İnsan gücünün taşıma kapasitesini aşan yüklerin taşınmasında kullanılır. Bu nedenle genellikle yük araçlarına, yükleme ve boşaltma işlerinde, fabrikalar da ağır yükleri taşımak amacı ile fabrikaların tavanlarına monta edilerek kullanımı yaygındır. Islak, nemli, rutubetli çalışma ortamlarında kullanılabilir. Küçük güçte olanları pnömatik olarak tahrik sistemine sahiptir. Büyük güç gereken yerlerde hidrolik tahrikli olan kartezyen robotlar kullanılır. Bunların yağ sızdırma problemleri olduğu için temizliğin önem arz ettiği ortamlarda pnömatik tahrikli olanlar tercih edilir. Hava tahrikli olan robot tipinde basınçlı hava ve havanın kontrolüne ihtiyaç olduğu için yatırım maliyetleri daha ucuz olup işletim maliyetleri de düşüktür. Büyük güçte yapılan kartezyen robotların tahrik sistemleri elektrik motorları veya hidrolik tahrik sistemleri ile sağlanmaktadır [16].



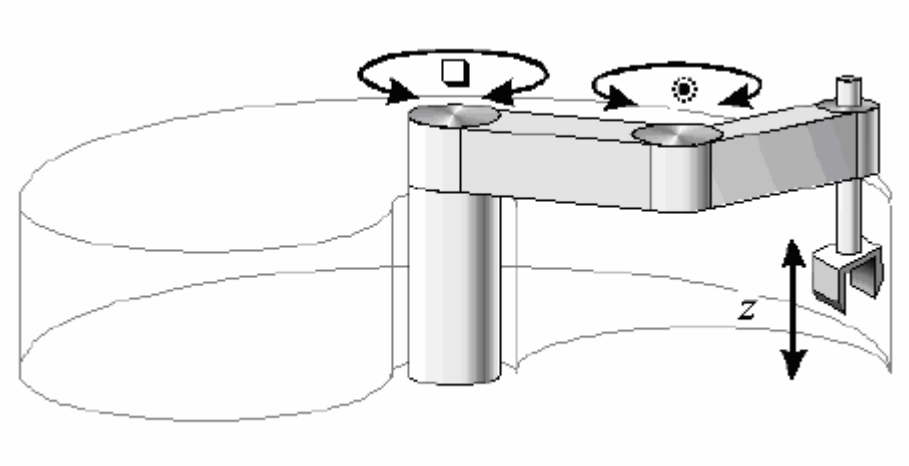
Şekil II. 5 Kartezyen Robot Kolun Çalışma Alanı [16]

Silindirik Robot kolları: Silindirik robot kollar da kendi etrafında dönebilen bir mafsal ve bunun üzerinde bulunan X,Y,Z düzleminde doğrusal hareket edebilen kollardan oluşmaktadır. Şekil II.6‘ da görüldüğü gibi esnek olmayan silindirik bir koordinat sistemine sahiptirler. Kartezyen robot kola göre hareket serbestiyesi daha geniştir. Çalışma alanı içindeki noktalara ulaşımı çok iyidir. Hareket kabiliyetinin az olmasından dolayı programlanması kolaydır. Robot kolun çalışma alanı silindirik koordinat sisteminde hareket eden kolların uzunluğuna bağlı olarak değişmektedir. Robotun kullanım alanı ve yük taşıma kapasitesine göre hidrolik, pnömatik veya elektrik tahrikli olarak kullanılmaktadır. Silindirik robot kollar nemli, rutubetli ve tozlu ortamlarda, deniz altı, uzay gözlem araçlarında ve nokta kaynağı işlerinde yaygın olarak kullanılmaktadır [16].



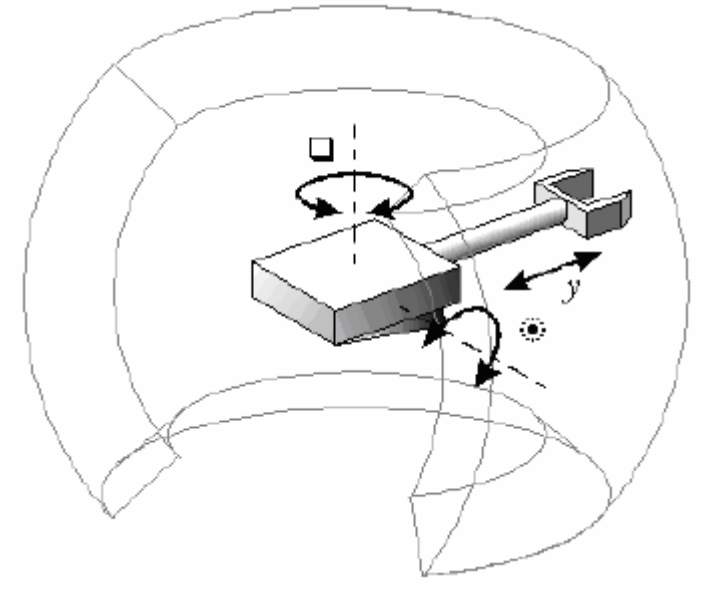
Şekil II. 6 Silindirik Robot Kolun Çalışma Alanı [16].

Scara (Selective Compliant Assembly Robot Arm) tip robot kolları: İki eklem yerinde elektrik motoru ve aşağı yukarı hareket edebilen pnömatik koldan oluşmuştur. Eklemlerdeki elektrik motorları eksenlerin kendi etrafında dönmesini sağlamaktadır. Tutucu ağzın bulunduğu kol pnömatik tahrikli olup Z ekseninde hareket etme kabiliyetine sahiptir. Bu da robot kola esnek hareket imkânı sağlamaktadır. Hız ve konum performansı çok iyi olduğundan dolayı bu robot kol en çok elektronik sanayinde, elektronik kartlara malzemelerin montajını gerçekleştirmek için kullanılmaktadır. Tutma ve taşıma işlerinde maliyetinin ucuz olmasından ve programlanmasının kolay olmasından dolayı şu anda sanayide en çok kullanılan robot olmuştur. Bu robot kol Şekil II.7 den çalışma alanı görüldüğü gibi, kolun altında bulunan parçaların taşınmasında kullanılır [16].



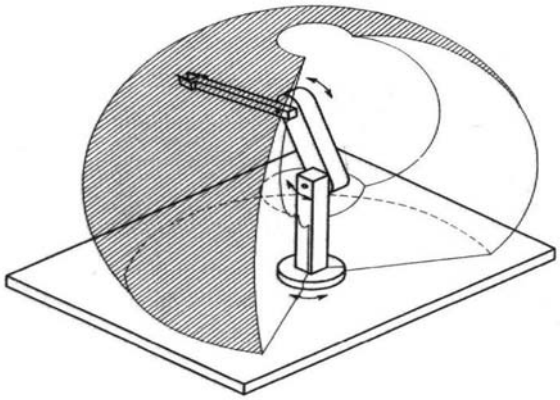
Şekil II. 7 Scara Robot Kolun Hareket Alanı [16]

Küresel robot kolları: Bel, omuz ve dirsek mafsallarından oluşan bir yapıya sahiptirler. Bel ve omuz mafsalı kendi etrafında dönme hareketi yapabilirken, dirsek mafsalı kola uzama ve kısalma hareketi yaptırmaktadır. Hareket alanı Şekil II.8' de görüldüğü gibi silindirik bir koordinat sistemine sahiptir. Kol yapılarından dolayı eklemli robot kollarına benzemektedirler. Kinematik yapıları kartezyen ve silindirik robot kollara göre daha karmaşıktır. Çalışma şeklinin zihinde canlandırılması zor olduğu için programlama ve kontrolü de zordur. Çalışma alanının büyüklüğü kolların büyüklüğüne bağlıdır. Hidrolik tahrik sistemine sahip olan küresel robot kollar eğme, bükme işlerinde, kameralı izleme işlerinde kullanılmaktadır. Küresel koordinatlı robotlar hassas işlerde diğer robot çeşitlerine nazaran daha çok tercih edilirler [16].



Şekil II. 8 Küresel robot kolun çalışma alanı [16].

Mafsallı kol konfigürasyonlu robot: Mafsallı (eklemli) kol konfigürasyonlu robotlarda, insan kol yapısı esas alınmıştır. Robot kolunun tüm eklemleri döner yapıya sahiptir. Uç kısmına cisimleri tutabilmesi için robot eli takılabilir. Şekil II.9'da görüldüğü üzere, çalışma uzayları küresel ya da silindiriktir. Endüstride geniş kullanım alanına sahiptir. Tıbbi araç-gereç yapımında, otomobil sanayinde kullanılırlar.



Şekil II. 9 Mafsallı kol konfigürasyonlu robotlar [22].

II.1.4.2 Kontrol Yöntemlerine Göre Robotlar

Kontrol yöntemlerine göre robotlar noktasal kontrol edilenler ve sürekli yörüngede hareket edenler olmak üzere iki kısma ayrılır.

Noktasal Kontrol Edilen Robotlar: Noktasal kontrol edilen robotların çalışmasında hedeflenen robot uç işlevcisinin belli noktalara gitmesidir. Dolayısıyla robot kolunun gerek takip ettiği yörünge gerekse hızı önemli değildir. Genellikle taşıma işlemlerinde kullanılır.

Sürekli yörüngede hareket eden robotlar: Bu tür robotların çalışmasında, noktasal kontrol edilen robotlara nazaran robot eklemlerinin belirli bir yörüngeyi takip etmesi zorunludur. Her bir eklem hızının değişken olmasında sakınca yoktur.

II.1.4.3 Kullandıkları Güç Kaynaklarına Göre Robotlar

Kullandıkları güç kaynaklarına göre elektrik, hidrolik ve pnömatik olmak üzere üçe ayrılır.

DC Servo ve adım motoru kullanan robotlar: DC Servo motorlar, düşük gerilimle yüksek tork ürettikleri için tercih edilir. Adım motorlar ise daha basit uygulamalarda kullanılırlar. Her iki motor tipi de hassas çalışma sağlar [23].

Hidrolik robotlar: Bu tür robotlar düşük güç ile yüksek tork elde ettikleri için daha çok ağır endüstride kullanılırlar [23].

Pnömatik robotlar: Tasarımları çok basittir. Hareket eden pistonların ataletini hızla ortadan kaldırabilecek hava basıncı üretilemediği için kontrolleri zordur. Basit uygulamalarda kullanılırlar [23].

II.1.4.4 Kesinlik Derecelerine Göre Robotlar

Kesinlik derecelerine göre robotlar, çözünürlük, doğruluk ve yinelenebilirlik olmak üzere üç kısımda incelenir.

Çözünürlük: Çok küçük bir yer değiştirmeyi gerçekleştirme yeteneğidir. Teknolojik gelişmelerle birlikte çözünürlük oranı da gün geçtikçe artmaktadır [20].

Doğruluk: Bir robotun hareket edebilmesi için yazılan programın, uç işlevci tarafından gerçekleştirme derecesidir. Daha çok çevrimdışı uygulamalarda kullanılan bir özelliktir [20].

Yenilenebilirlik: Robotun uç işlevcisinin birçok işlemi gerçekleştirdikten sonra tekrar aynı noktaya gelebilme özelliğidir. Günümüzde kullanılan robotların tekrar edebilirliği 0.05 ile 0.005 inç arasındadır [20].

II.1.5 İleri Yön Kinematiği ve Ters Kinematik

II.1.5.1 İleri Yön (Düz) Kinematik

Bir robot ana çerçevesinden araç çerçevesine doğru birbirine prizmatik veya döner eklemlerle bağlanmış seri uzuvlardan oluşur. İki uzuv arasındaki ilişki bir homojen dönüşüm matrisiyle açıklanır. Eklem dönüşüm matrislerinin ard arda çarpılmasıyla ana çerçeve ile araç çerçeve arasındaki ilişki tanımlanır. Bu ilişki manipülatörün araç çerçevesinin konumunu ve yönelimini ana çerçeveye göre belirtir. Kısaca ileri yön kinematiği eklem değişkenleri ile uç işlevcisinin konumu ve yönelimini ana çerçeveye göre hesaplar diyebiliriz.

Her bir ekleme bir koordinat sistemi yerleştirilse komşu iki eklem arasındaki ilişki bir ${}^{i-1}_{i}T$ dönüşüm matrisi elde edilir. İlk ekleme ait dönüşüm matrisi ilk eklem ile ana çerçeve arasındaki ilişkiyi tanımlarken, son ekleme ait dönüşüm matrisi uç işlevcisi ile son eklem arasındaki ilişkiyi ifade eder. Arka arkaya sıralanan bu eklem dönmüşüm matrisleriyle ana çerçeve ile araç çerçevesi arasındaki ilişki tanımlanır.

Bu ilişkiye de ileri kinematik denir. Ana çerçeve ile araç çerçevesi arasındaki ilişki denklem II.1'deki gibi tanımlanır.

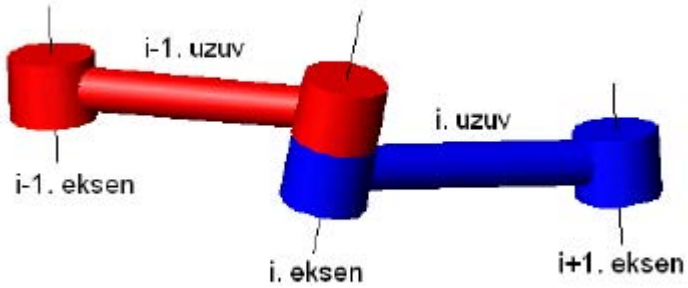
$$ {}^{0}_{N}T = {}^{0}_{1}T\,{}^{1}_{2}T \dots\dots\dots {}^{N-1}_{N}T \qquad \text{(II.1)} $$

Eklem Değişkenlerinin Belirlenmesi:

Robotların eklem değişkenlerinin belirlenmesi için birçok kinematik yöntem belirlenmiştir. Kinematik problemler kartezyen üç boyutlu ve kartonom dört boyutlu olmak üzere iki farklı uzayda gerçekleştirilir. Kartezyen uzayda üstel yöntem, Pieper- Roth yöntemi ve Denavit-Hartenberg yöntemi kullanılır. Ancak en fazla tercih edilen yöntem Denavit-Hartenberg yöntemidir [24].

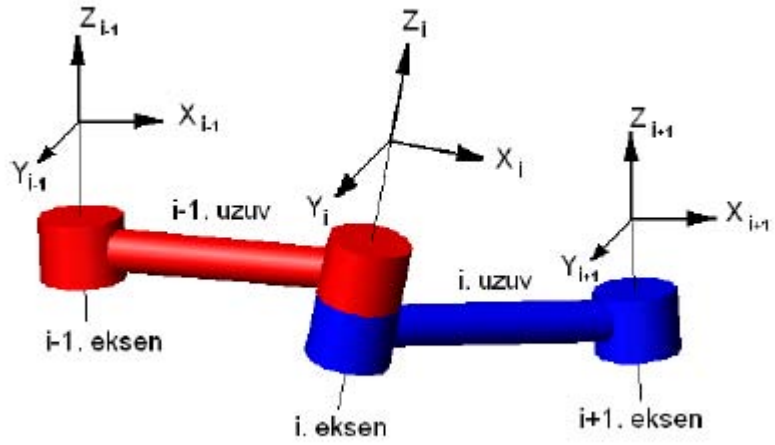
Denavit-Hartenberg Yöntemi:

Bu yöntemde dört ana değişken kullanılarak robot kinematiği çıkarılır. Bu değişkenler iki eksen arasındaki uzuv uzunluğu a_{i-1} , iki komşu eksen arasındaki eksen açısı α_{i-1} , üst üste çıkan bağlar arasındaki eklem kaçıklığı d_i ve iki komşu uzuv arasındaki eklem açısı θ_{i-1} 'dir. Bu dört değişkende D-H değişkenidir. Bu değişkenleri belirlemek için öncelikli olarak şekilden görüldüğü gibi robotun dönme eksenleri belirlenir ve dönme eksenleri uzuvlardan bir fazla olacak şekilde numaralandırılır.



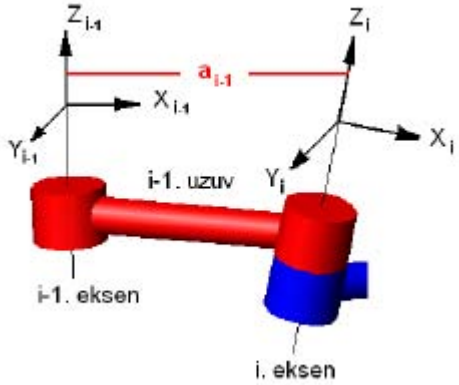
Şekil II. 10 Dönme Eksenlerinin Belirlenmesi

Daha sonra bu eksenlerin her birine bir koordinat sistemi yerleştirilir ve uzuv dönme ekseni Şekil II.11'de görüldüğü gibi koordinat sisteminin Z ekseni olarak kabul edilir.



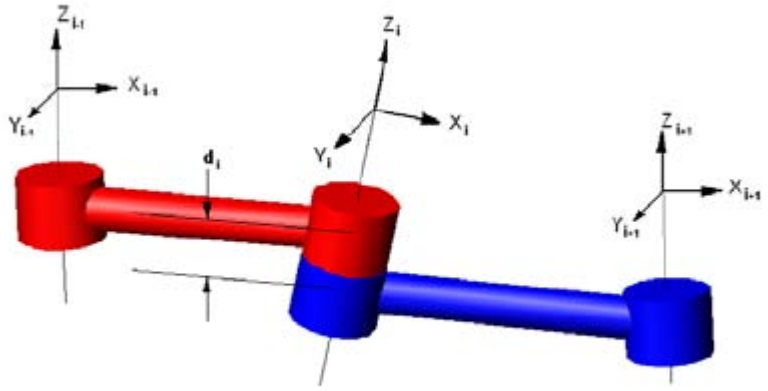
Şekil II. 11 Dönme Eksenlerine Z,Y ve X Eksenlerinin Yerleştirilmesi

Şekil II.12' deki gibi X_{i-1} yönünde uzanan Z_{i-1} ile Z_i eksenleri arasındaki dik uzaklığa a_{i-1} uzuv uzunluğu denir.



Şekil II. 12 a_{i-1} Uzuv uzunluğu

Şekil II.13' deki gibi Z_i yönünde uzanan X_{i-1} ile $X_i$eksenleri arasındaki dik uzaklığa d_i eklem kaçıklığı denir.



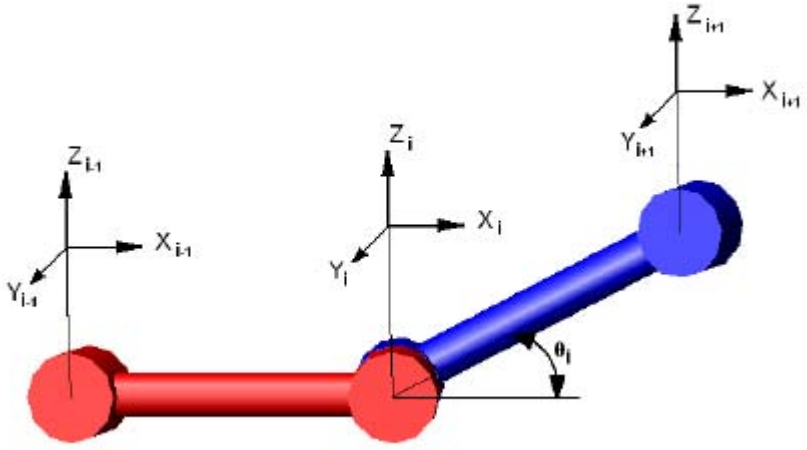
Şekil II. 13 d_i Eklem Kaçıklığı

Şekil II.14‘ deki gibi Z_{i-1} ekseni ile Z_i ekseni arasındaki X_i boyunca ölçülen açıya α_{i-1} eksen açısı denir.



Şekil II. 14 α_{i-1} Eksen Açısı

Şekil II.15’ deki gibi X_{i-1} ekseni ile X_i ekseni arasındaki Z_i boyunca ölçülen açıya θ_i eklem açısı denir.



Şekil II. 15 θ_i Eklem Açısı

Koordinat Sistemlerinin Eklemlere Yerleştirilmesi:

Koordinat sistemleri eklemlere yerleştirilirken sırasıyla aşağıdaki işlemler gerçekleştirilir.

Öncelikle eklem eksenlerinin dönme ve kayma yönleri belirlenerek bu eksene paralel bir doğru çizilir.

Eklem eksenleri döner eklemler için dönme yönü Z, prizmatik eklerler için kayma yönü Z ekseni olarak belirlenir.

Z eksenine dik ve uzuv boyunca uzanan eksen X ekseni olarak belirlenir. Z ve X eksenleri belirlendikten sonra Y ekseni sağ el kuralına göre belirlenir.

Arka arkaya gelen iki eklemin dönme ve kayma yönleri aynı ise Z ekseni belirlendikten sonra kol boyunca X ekseni belirlenir ve sağ el kuralına göre Y ekseni belirlenir. Sıfır ve birinci eksenler üst üste kabul edilir.

Bir robotun eklemlerine koordinat sistemleri yerleştirilirken birinci eksenin dönme yönü Z ekseni olarak belirlendikten sonra bu ekleme X ekseni döndürüldüğünde komşu iki Z ekseni üst üste çakışacak şekilde bir X ekseni yerleştirilir.

Koordinat sistemleri eklemlere yerleştirildikten sonra D-H değişkenleri bulunur ve aşağıdaki tabloya yazılır. Robotun hareket etmesiyle değişmeyen parametreler a_{i-1} uzuv uzunlukları ve α_{i-1} eksen açılarıdır. Değişen parametreler eklem döner ise θ_{i-1} denklem açısı eğer eklem prizmatik ise d_{i-1} dir.

Eksen No	D-H Değişkenleri				Eklem Değişkeni
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i veya θ_i
1	α_0	a_0	d_1	θ_1	$d_1\ veya\ \theta_1$
2	α_1	a_1	d_2	θ_2	$d_{2\ veya}\ \theta_2$
3	α_2	a_2	d_3	θ_3	$d_3 veya\ \theta_3$

Tablo II. 1 D-H Değişkenleri

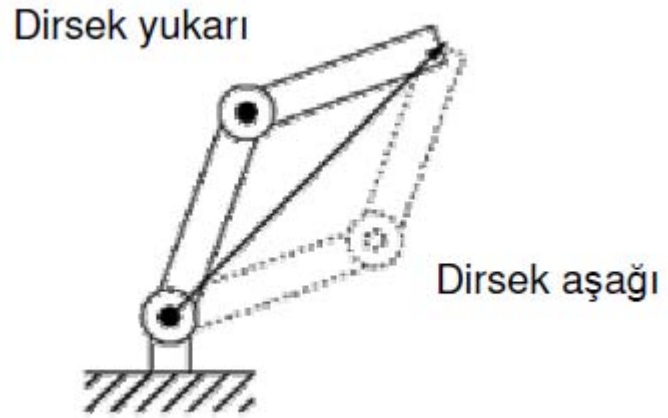
$$
{}^{i-1}_{i}T = R_x(\alpha_{i-1})D_x(a_{i-1})R_Z(\theta_i)D_{Zx}(d_i)
$$

$$
= \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & a_{i-1} \\ sin\theta_i \cos\alpha_{i-1} & cos\theta_i cos\ \alpha_{i-1} & -sin\ \alpha_{i-1} & -\sin\alpha_{i-1}\ d_i \\ sin\theta_i \sin\alpha_{i-1} & cos\theta_i sin\ \alpha_{i-1} & \cos\ \alpha_{i-1} & \cos\alpha_{i-1}\ d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(II.2)}
$$

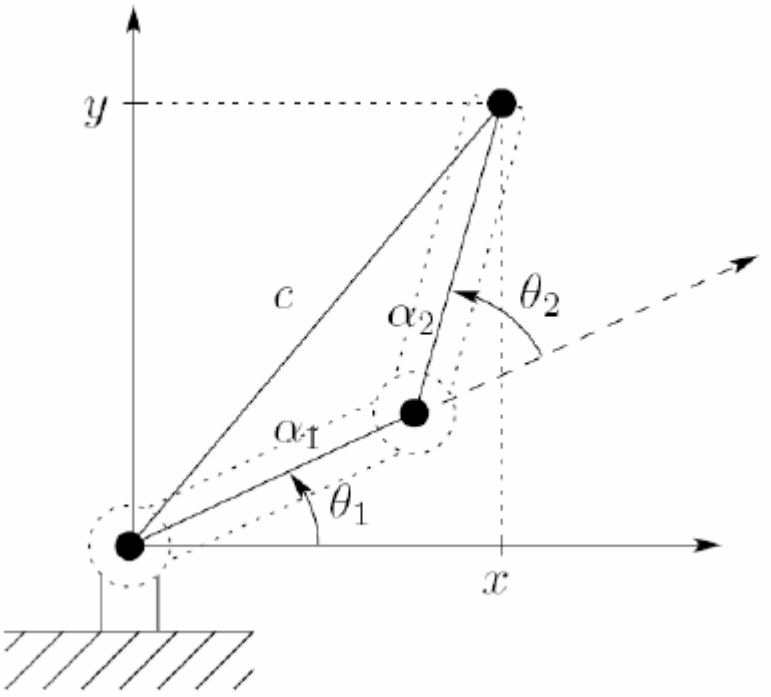
Her bir ekleme ait genel dönüşüm matrisi elde edilir.

II.1.5.2 Ters Yön Kinematiği

Robot kolu ters kinematigi; robot kolunun uç noktasının koordinat bilgilerinden hareketle şekilde verilen robot kolunun θ_1 ve θ_2 açılarının bulunması seklinde ifade edilebilir Sekil II.16. Robot kolunun uç noktasının verilen koordinat bilgileriyle açısının bulunmasında, şekilde de görüleceği üzere iki farklı çözüm bulunabilir. Bunlara dirsek yukarı ve dirsek aşağı çözümleri diyebiliriz. Açıların bulunmasında robot kolunun eklem sayısına göre, bazen bir çözüm veya sonsuz çözüm de bulunabilir [27].



Şekil II. 16 Ters Kinematik Çift Çözüm



Şekil II. 17 Eklem Açılarının Çözümü

Şekil II.17'de θ_2 açısının bütünler açısına kosinüs teoremi uygulanırsa:

$$\cos(180 - \theta_2) = \frac{a_1^2 + a_2^2 - c^2}{2a_1a_2} = -D \qquad \text{(II.3)}$$

$$\cos(180 - \theta_2) = \cos(180)\,.\cos(\theta_2) + \sin(\theta_2).\sin(180) = \cos(\theta_2) \qquad \text{(II.4)}$$

Denklem (II.3) ile (II.4) kullanılarak θ_2 açısı:

$$\theta_2 = cos^{-1}(D) \quad \text{(II.5)}$$

olarak elde edilir. Fakat daha genel bir çözüm için:

$$cos^2\theta_2 + sin^2\theta_2 = 1 \quad \text{(II.6)}$$

Denklemden yararlanarak θ_2 'nin sinüs ile ifadesi:

$$\sin(\theta_2) = \pm\sqrt{1 - D^2} \quad \text{(II.7)}$$

Elde edilir.
Dolayısı ile θ_2 açısı:

$$\theta_2 = tan^{-1}\left(\frac{\pm\sqrt{1-D^2}}{D}\right) \quad \text{(II.8)}$$

Şeklinde ifade edilebilir. Bu yaklaşımın sonucu olarak, dirsek yukarı ve dirsek aşağı çözümleri; pozitif ve negatif işaretlerin seçilmesiyle elde edilir. Günümüzde bilgisayar ortamında modellemeler yapılarak, robot kolunun tüm eklemlerinin yapabileceği azami açıların girilmesi ile robot kolunun hareket kabiliyetini görmemiz mümkündür. Bu ise 6 veya daha üzeri serbestlik derecesi olan bir robot kolunun yorucu kinematik hesaplamalarına göre büyük avantaj sağlamaktadır.

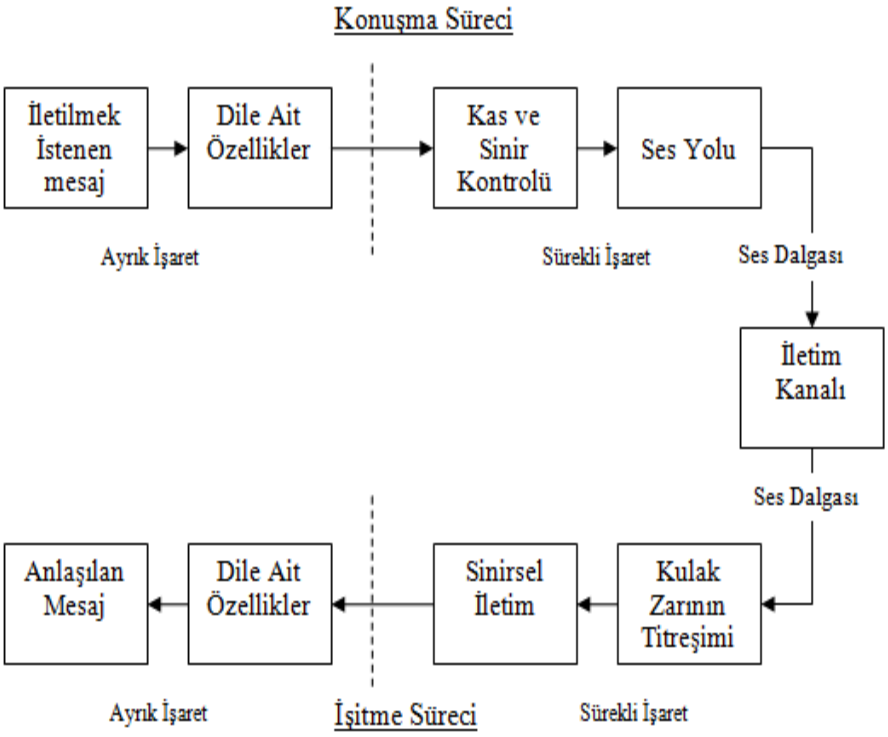
II.2 KONUŞMA TANIMA

Konuşma süreci, konuşmacının dinleyiciye iletmek istediği mesajı beyninde oluşturmasıyla başlar. Bir sonraki aşamada, oluşturulan mesaj konuşulan dile ait

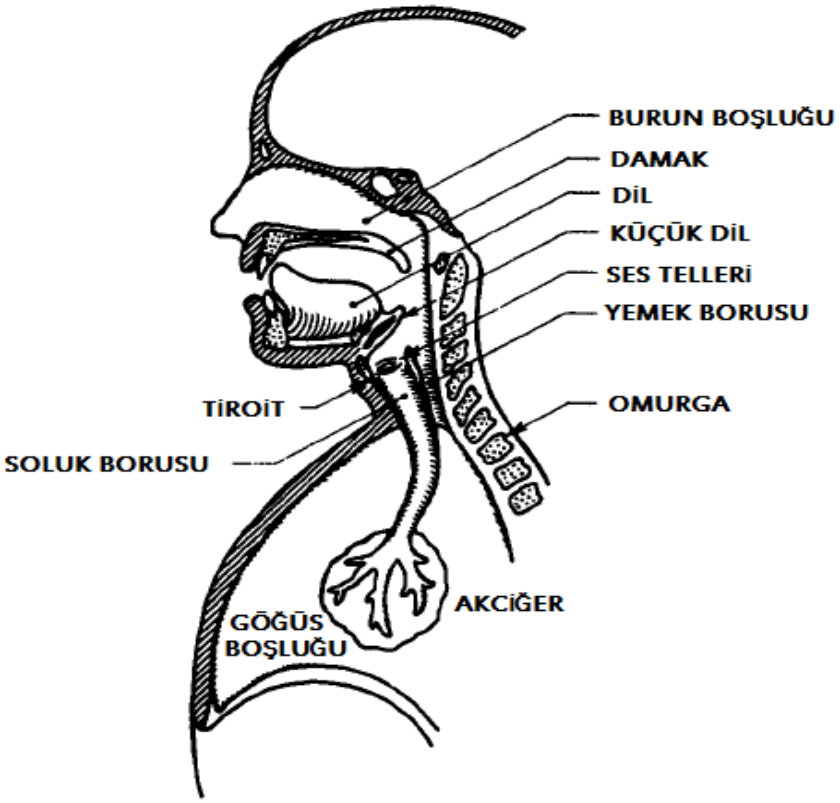
özellikler çerçevesinde o dilin koduna dönüştürülür. Sinirsel sinyaller kasları uyararak ses mekanizmasının uygun şekilde çalıştırılmasını sağlar ve iletilmek istenen mesajı oluşturan ses dalgaları ortaya çıkar. Böylece konuşma sürecinin sonunda bir akustik sinyal üretilmiş olur. Bu akustik sinyal (yani ses dalgası), hava aracılığıyla yayılır [25].

İşitme süreci ise, konuşma sinyalinin üretilip dinleyicinin kulağına ulaştığı anda başlar. İlk olarak, dinleyiciye ulaşan akustik sinyal kulak zarının titreştirir ve iç kulakta gelen sinyalin spektrum analizi yapılır. Sonraki aşamada spektral sinyaller, işitsel sinirleri uyaracak şekilde sinirsel aktivitelere dönüştürülür. Detayları halâ çok iyi bilinmeyen bir şekilde, işitsel sinirlerdeki sinirsel aktiviteler, beynin işleme ile ilgili daha yüksek seviyedeki merkezlerinde dil koduna dönüştürülür. Son aşamada mesajın anlaşılması gerçekleşir. Konuşma ve işitme süreçlerini tasvir eden şema Şekil II.18'da görülmektedir.

İnsanın ses mekanizmasına ilişkin şematik diyagram Şekil II.19 'de görülmektedir. Bu mekanizmanın çalışması şu şekilde gerçekleşir. Normal nefes alma mekanizması vasıtasıyla akciğerler hava ile dolar. Akciğerlerdeki hava, soluk borusu yoluyla dışarı atılır ve böylece bu hava akışı gırtlaktaki ses tellerinin titremesi sağlar. Hava akışı sözde periyodik darbeler haline getirilir. Bu darbeler yutak, ağız boşluğu ve burun boşluğundan geçerken frekans modülasyonu gerçekleşir. Çeşitli artikülatörlerin (mesela çene, dil, damak, dudaklar, ağız) pozisyonlarına bağlı olarak farklı sesler üretilir.

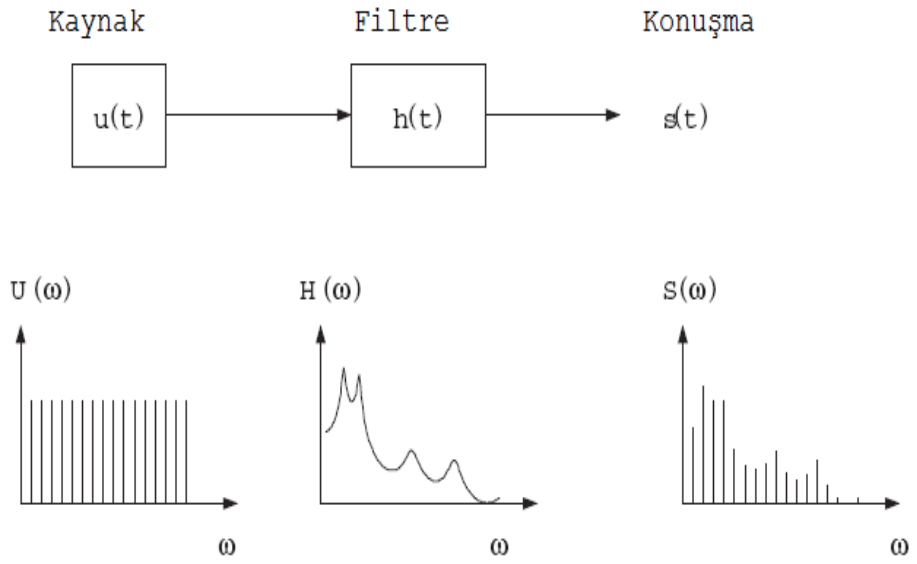


Şekil II. 18 Konuşma ve işitme süreçlerine ilişkin şematik yapı.



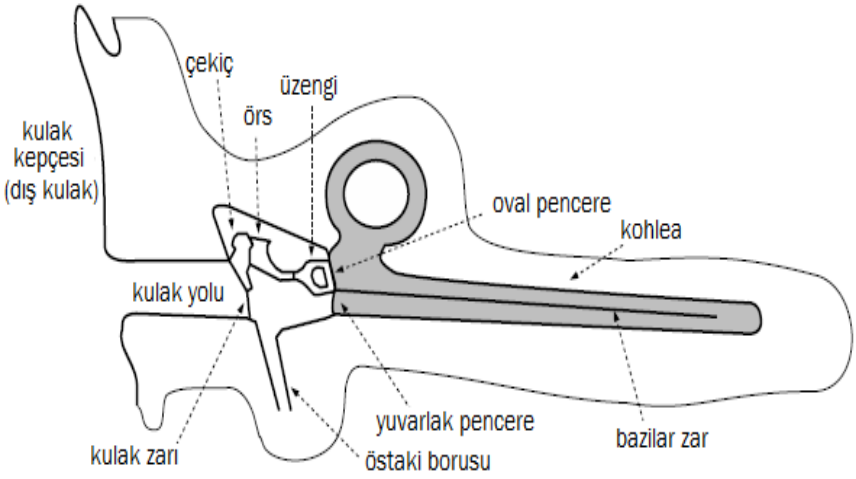
Şekil II. 19 İnsanın ses mekanizmasının şematik görünümü

Akustik teoriye göre, ses oluşumu sağlayan sistemi iki bağımsız alt sisteme ayırmak gerekir. Buna göre, ses yoluna ilişkin transfer fonksiyonu, bir lineer filtre olarak modellenebilir. Filtrenin girişi kaynak sinyalidir ve çıkışı da ses sinyalidir (yani konuşmadır). Ses oluşumunu sağlayan sistemin modeli Şekil II.20'de görülmektedir. Burada u(t): kaynak, h(t): filtre ve s(t): ses sinyalidir. Bu bileşenlerin spektrumları ise sırasıyla U(ω), H(ω) ve S(ω) dır.



Şekil II. 20 Ses oluşumunun lineer modeli.

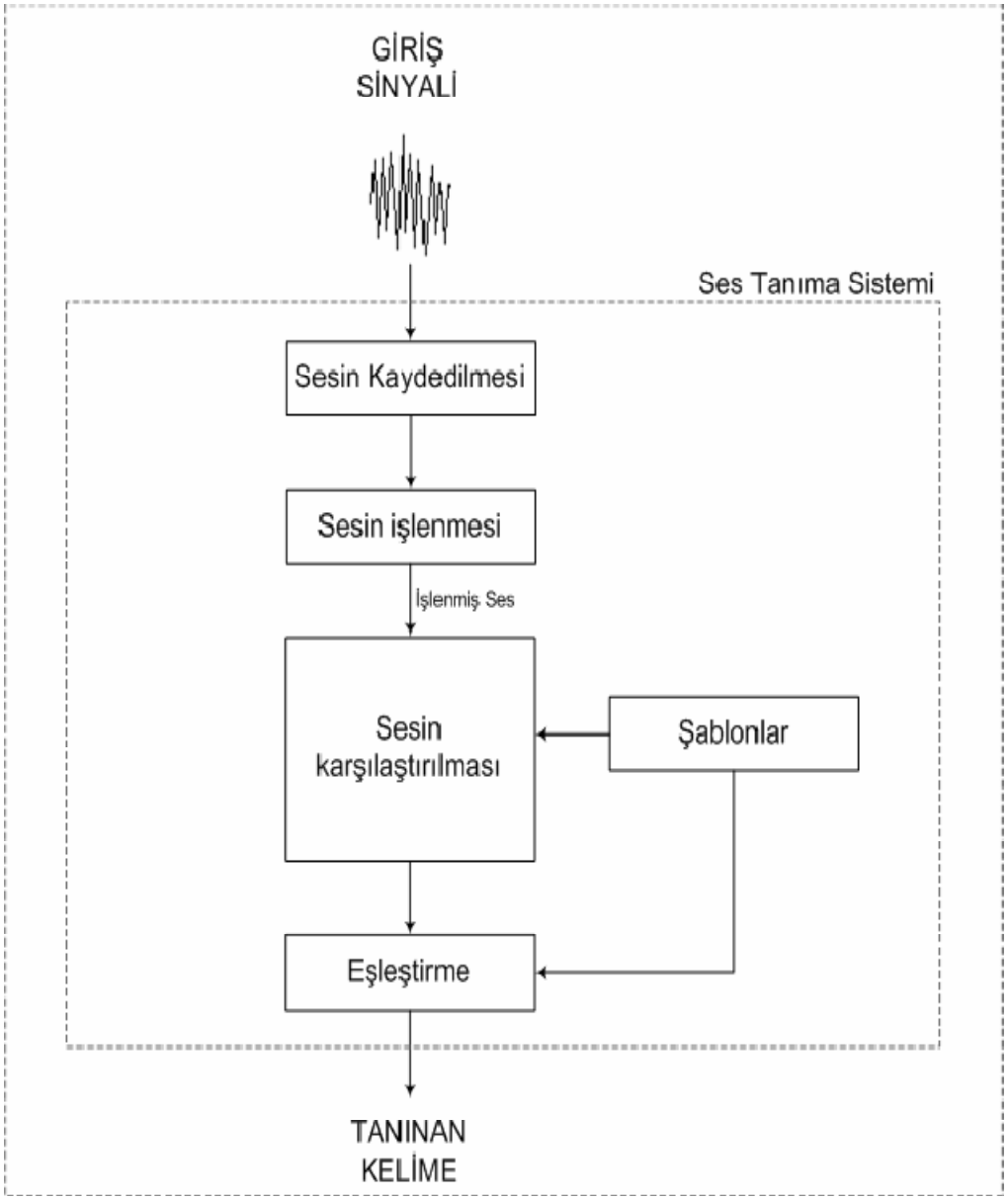
İnsan kulağının şematik görünümü Şekil II.21 'da verilmiştir. Dış kulak ortamdan gelen ses dalgalarını toplayarak kulak yoluna iletir. İşitme kanalından geçerek kulak zarına gelen ses dalgaları, buradaki çekiç, örs ve üzengi kemikleri tarafından güçlendirilerek oval pencere üzerinden kohleaya kadar iletilir. Kohlea şekilde gri renkli olarak görünen kısımdır ve kolay anlaşılması açısından kohleanın sarmal yapısı açılarak düz halde çizilmiştir. Kohlea yaklaşık 35 mm uzunluğunda iki tane zarı (bazilar ve Reissner) içinde barındırır. Bazilar zar üzerinde Korti organları bulunur ve bu organlar çok sayıda kılcal reseptör hücreler içerir. Bu hücreler üzerlerine baskı uygulandığında küçük elektrik sinyalleri üretirler. Bazilar zarın genişliğinin ve sertliğinin kohlea boyunca azalması, kohleanın bir ucunda düşük frekans, diğer ucunda yüksek frekans duyarlılığı sağlar. Son olarak üretilen elektrik sinyalleri sinirler tarafından kulaktan beyne taşınır ve burada yoğun bilinçdışı işlemler gerçekleşir.



Şekil II. 21 İnsan kulağının şematik görünümü.

II.2.1 Konuşma Tanıma Sistemine Genel Bir Bakış

Ses tanıma sistemlerinin çalışması, insanlar arası sesli iletişim sürecinde dinleyicinin yaptığı işlevleri yapay olarak gerçekleştirme prensibine dayanmaktadır. Dinleyici ve ses tanıma sistemlerinde basit bir eşleştirme yapılırsa; sırasıyla dinleyicideki işitme sisteminin yerini, sesin kaydedilmesi alır; sinirsel dönüşümlerin yerini, alınan sesin islenmesi; dilsel kodlamanın yerini, işlenmiş sesin dilsel karşılıkların belirlenmesi veya mevcut şablonlarla karşılaştırılması; konuşmanın anlaşılmasının yerini ise eşleştirme alır. Ses tanıma sistemi için örnek bir ses tanıma modeli, Şekil II.22'de verilmektedir.



Şekil II. 22 Örnek Bir Ses Tanıma Sistemi Modeli

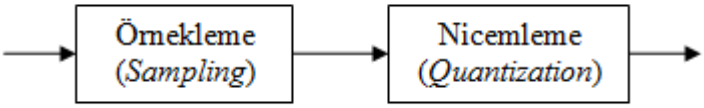
II.2.2 Ses Tanıma Süreci

Sesli ifade tanıma süreci, mikrofon tarafından algılanan analog ses dalgalarının sayısallaştırılmasıyla başlar. Sayısallaştırılan sinyal üzerinde çeşitli ön işlemler gerçekleştirilerek, sesli ifade sinyalinin öznitelik vektörü çıkarılır. Oluşturulan öznitelik vektörü, sınıflandırma sürecinden geçirilir. Sesli ifade tanıma süreci genel şekliyle bu adımlardan oluşmaktadır.

II.2.2.1 Sesin Sayısallaştırılması

Konuşurken ortama yayılan ses dalgaları, mikrofon tarafından algılanarak elektriksel sinyale dönüştürülür. Elde edilen analog sinyalin bir bilgisayar

aracılığıyla analiz edilebilmesi için öncelikle sayısallaştırılması gerekmektedir. Her analog sinyalde olduğu gibi, sesli ifade sinyalinde de, düzenli zaman aralıklarıyla analog sinyalin seviyesi yakalanır veya diğer bir ifadeyle örneklenir. Örnekleme, belli zaman aralıklarıyla sinyalden örnek alma işlemidir. Bir analog sinyalin sayısallaştırılması için minimum örnekleme oranının ne olması gerektiği, temel sinyal işleme kavramı olan, Nyquist-Shannon Örnekleme Teoremi ile belirlenmiştir [28]. Bu teoreme göre bir analog sinyal, içerdiği en yüksek frekanslı bileşenin en az iki katı frekansta örneklenmelidir. Aksi takdirde, sayısallaştırılan sinyalde sahte frekans bileşenleri görülür ve bu sayısal sinyalden, tekrar orijinal analog sinyali elde etmek için dönüşüm yapıldığında, sinyalde bozulmalar meydana geldiği görülür. Şekil II.23'te, analog sinyalin sayısallaştırılması için gerçekleştirilen işlemler görülmektedir. Nicemleme işlemi ise, sürekli değerlerin ayrık değerlere eşlenmesi işlemidir.



Şekil II. 23 Analog Sinyalin Sayısallaştırma Süreci

II.2.2.2 Sesli İfade Tanımada Akustik Analiz ve Öznitelik Çıkarımı

Akustik analizin amacı, sesli ifade sinyalindeki dilsel mesajla ilgili özellikleri vurgulamak ve fonetik anlamla ilişkisi olmayan, yankı ve gürültü, vb. etkenleri bertaraf etmektir. Şekil II.24'te verilen diyagramda, akustik analizdeki temel adımlar görülmektedir.

Ses sinyali spektrumu, aşağı doğru bir eğilim gösterir. Yani spektrumun üst kısmındaki yüksek frekanslı bileşenlerin genlikleri düşüktür [25]. Sağlıklı bir analiz gerçekleştirmek için öncelikle, ses sinyali spektrumunu düzgünleştirmek gerekmektedir. Bu işlemi gerçekleştirmek için genellikle birinci dereceden yüksek geçirgen filtre kullanılır. Kullanılan filtre, yüksek frekans bileşenlerini vurgulamaya yarar. Bu filtrenin transfer fonksiyonu Denklem II.9'da verilmiştir.

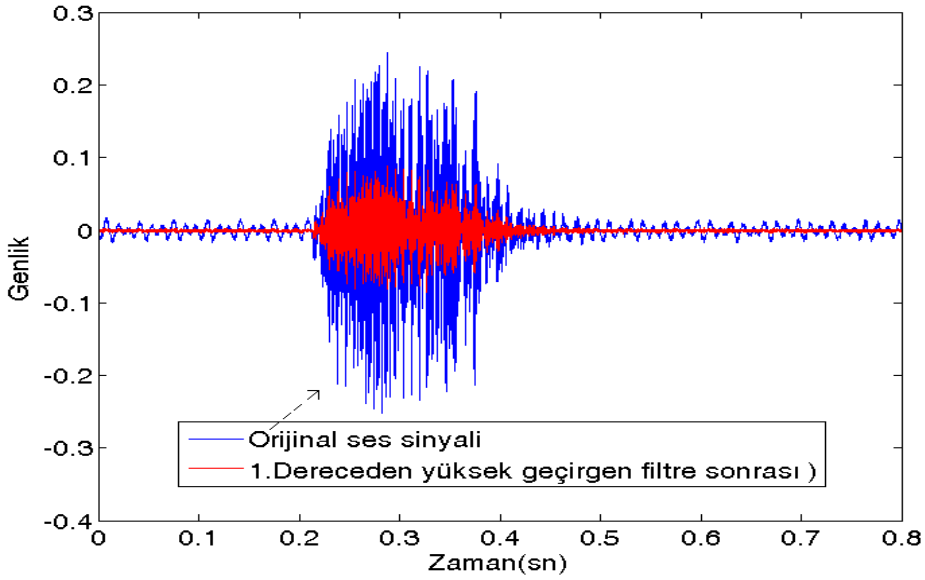
$$H(z) = 1 - \alpha z^{-1} \tag{II.9}$$

Burada α, bir sabittir.(Genellikle $0.9 < \alpha < 1.0$ seçilir.) Filtreleme işleminin sonucu Şekil II.25'de görülmektedir. Bu örnekte α'nın değeri 0.95 olarak seçilmiştir.



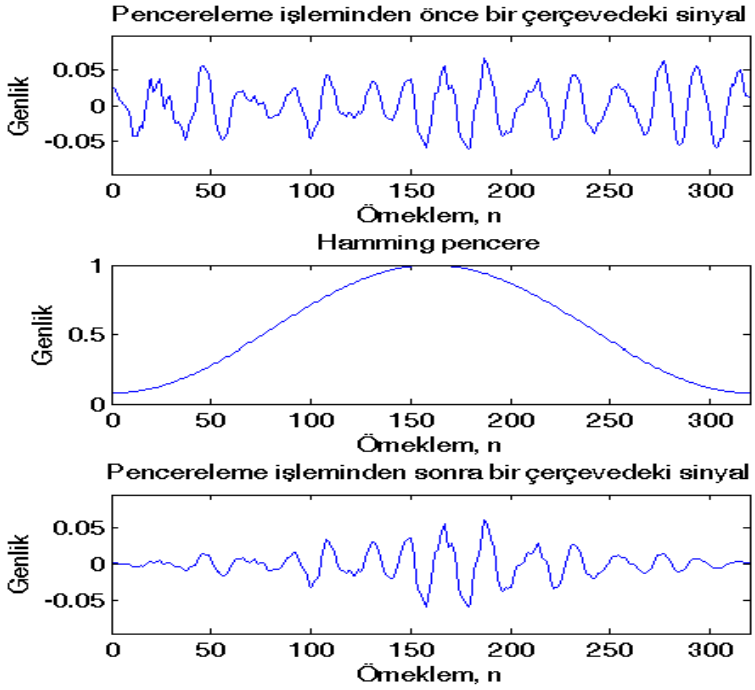
Şekil II. 24 Akustik analiz için genel blok diyagramı

Fiziksel yapısından kaynaklanan nedenlerden dolayı ses yolunun biçimi oldukça yavaş değişim gösterir. Yani kısa zaman aralıklarında (örn. 10-20 ms) sabit kalma eğilimi sergiler. Analiz için mantıklı yaklaşım, ses sinyalini eşit uzunluktaki parçalara bölmektir. Parçalama sonucunda, ardışık ses çerçeveleri ortaya çıkar. Her hangi bir frekans analizi yapmadan önce, ses sinyalinin parçalarını barındıran bu çerçeveler, konik yapıya sahip (genellikle 'Hanning' veya 'Hamming') bir "pencere" ile çarpılır. Yapılan bu pencereleme işlemi, ele alınan ses parçalarının kenarlarındaki süreksizlikleri azaltmaya yarar. Yapılan bu pencereleme işlemi Şekil II.26'daki grafiklerle [25] gösterilmiştir.



Şekil II. 25 Bir sesli ifadenin dalga şekli: orijinal ses sinyali(mavi renkli) ve 1. dereceden yüksek geçirgen filtre sonrası(kırmızı renkli)

Aksi takdirde daha sonra yapılacak olan frekans analizinde, sahte (yani aslında var olmayan) yüksek frekanslı bileşenler spektrumda ortaya çıkar. Analiz penceresinin boyu, gereken zaman çözünürlüğünü verecek kadar kısa olmalıdır ve aynı zamanda yeterli frekans çözünürlüğünü sağlayacak kadar uzun olmalıdır. Genel uzlaşma, 20-32 ms'lik bir pencere kullanmaktır [25]. Her analiz penceresi, ele alınan ses sinyali üzerinde pencere boyunun %50 si (yani 10-16 ms) kadar veya daha az ilerletilir. Oluşturulan bu analiz pencereleri, akustik analiz genel diyagramında bir sonraki blok tarafından kullanılmaktadır.



Şekil II. 26 Pencereleme işleminin bir çerçevedeki sinyal üzerinde gösterimi

Şekil II.24'de görülen üçüncü blok, önceki blok tarafından oluşturulan sıralı analiz pencereleri üzerinde hızlı Fourier dönüşümü (*Fast Fourier Transform -FFT*) yapar. Fourier donüşümü, analiz pencerelerindeki değerleri, farklı frekans bileşenlerine ayrıştırma işini yapar. FFT, sinyallerin Ayrık Kosinüs dönüşümünü (*Discrete Cosine Transform - DCT*) hesaplamak için kullanılan etkili bir algoritmadır. Kaliteli ses kaydı elde etmek için örnekleme oranı oldukça yüksek tutulur. Bunun sonucunda, elde büyük miktarlarda işlenecek veri olur. FFT algoritması, aynı miktardaki veri için sonucu daha kısa sürede hesaplar.

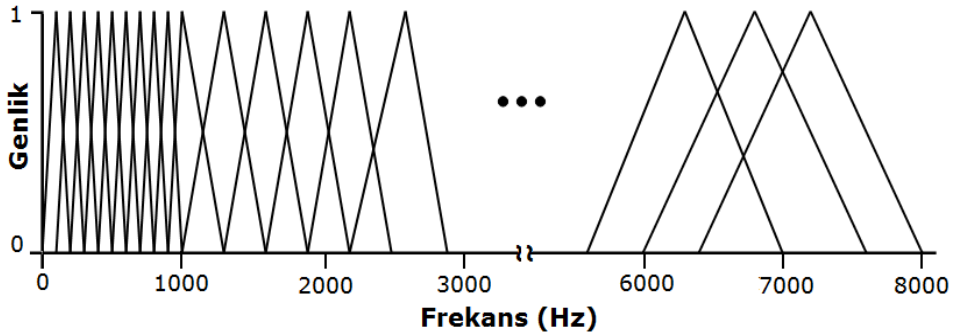
Analiz penceresinin boyu 20-32 ms arasında bir değerdedir. Bu kısa zaman periyodunda spektrumun faz bileşeni göz ardı edilebilir varsayımından hareketle, sadece kısa-vadeli güç spektrumu hesaplanır. Bu güç spektrumu, üçüncü bloktan gelen kompleks FFT katsayılarının büyüklüklerinin kareleri alınarak hesaplanır [25].

Psikofiziksel çalışmalar insan kulağının frekans içeriğini algılamasının lineer bir skalada gerçekleşmediğini göstermektedir. Çalışmalar göstermiştir ki bu skala, 1 KHz e kadar doğrusal, 1 Khz'in üzerinde ise logaritmik bir yapıdadır. Bu yapıya karşılık gelen ve frekans eksenini, Mel-kepstral işlemeye uygun hale döndüren fonksiyon [25] Denklem II.10'daki gibidir.

$$\Omega(\omega) = 2595 \log_{10}\left[1 + \frac{\omega}{1400\pi}\right] \qquad \text{(II.10)}$$

Burada ω, (raydan/saniye) olarak frekansı göstermektedir.

Şekil II.24'deki altıncı blokta yapılan işlem, her bir kritik bant kanalı için sinyaldeki enerjiyi hesaplamaktır. İnsan kulağının frekans ayırma gücü ile ilgili psikofiziksel çalışmalardan hareketle bu filtre kanallarının bant genişlikleri ve aralıkları frekansla artan bir yapıda kullanılır. Şekil II.27' de Davis ve Mermelstein tarafından Fourier tranformunun sonucunu Mel skalasına dönüştürmek için ileri sürülen tipte üçgen filtreler görülmektedir. 4kHz bant genişliğindeki ses sinyalleri genellikle için 20 adet kanal kullanılır [25]. Daha yüksek bant genişliğine sahip sinyaller için birkaç tane kanal ilave edilebilir.



Şekil II. 27 Davis ve Mermelstein tarafından Fourier tranformunu Mel skalasına dönüştürmek için (1980 yılında) ileri sürülen tipte üçgen filtreler

Şekil II.24'deki akustik analiz genel diyagramındaki altıncı blok, sıkıştırma modülüdür. Bu adımda, kritik bant güç spektrumunun doğrusal olmayan dönüşümü hesaplanmak suretiyle spektral tepeler arasındaki farklar azaltılır [25]. Mel-kepstral analizde, kritik-bant güç spektrumunun logaritması alınarak, bahsedilen sıkıştırma işlemi yapılmaktadır.

Ayrık Kosinüs dönüşümü, işitsel spektrum benzeri yapıyı gerçek katsayılara dönüştürür. Bu katsayılar, sıkıştırılmış spektrumun bileşenlerine ayrıldığında ortaya çıkan Kosinüs terimlerinin katsayılarını belirtir. Mel kepstrum işleminde bu katsayılar, logaritmik spektrumun Fourier bileşenlerinin genliklerine karşılık gelmektedir [25].

Şekil II.24'deki akustik analiz genel diyagramının dokuzuncu modülü, adından da anlaşılacağı gibi spektrumu düzgünleştirme işini gerçekleştirir. Spektrumu düzgünleştirme işinin bir kısmı, kritik-bant entegrasyonu aşamasında spektrumun

detaylarının önemli bir kısmı bastırılarak yapılmaktadır. Buna ek olarak bu aşamada ayrı bir entegrasyon işlemi yapılarak, konuşma sinyalinde bulunan dilsel mesajla ilgili olmayan faktörlerin ektisini azaltmak amaçlanmaktadır. Bu işlem genellikle, 20 tane veya daha fazla filtreden elde edilen genlik değerlerinden, ilk 12 veya 14 bileşenin dışındakilerin budanmasıyla yapılmaktadır [25].

Şekil II.24'deki akustik analiz genel diyagramının son modülü, "liftering", İngilizce'deki "filtering" kelimesinin harflerin yerlerinde oynama yapılarak elde edilen bir kelimedir. "Liftering" işlemi, cepstrum üzerinde yapılan bir filtreleme işlemi olarak bilinmektedir. Bu modül, bir önceki adımda elde edilen cepstral parametreleri (örneğin, n:cepstral indeks ve α:değeri 0 ile 1 arasında olan bir parametre olmak üzere n^{α} gibi) basit bir fonksiyonla çarpar [25].

Mel frekans cepstrum yöntemiyle elde edilen katsayılara, mel frekans cepstral katsayılar (Mel-frequency cepstral coefficients-MFCCs) denir. Bu katsayılar Denklem II.11'deki ifadeyle DCT dönüşümü yapılarak hesaplanmaktadır.

$$c_j = \sqrt{\frac{2}{N}} \sum_{i=1}^{N} A_i \cos\left(\frac{\pi j(i-0.5)}{N}\right) \quad 0 \le j < N \qquad \text{(II.11)}$$

Burada c_j cepstral katsayı, A_i ise N adet kanal için logaritmik genlik olarak verilmiştir. Bu denklemdeki N filtre bankasındaki kanal sayısını ifade etmektedir. Genellikle bu cepstral katsayılardan ilk 12 tanesi alınır ve buna enerji terimi de eklenerek 13 adet statik parametre elde edilir. Sesli ifade tanıma sistemlerinin sağlamlığını ve kararlılığını arttırmak amacıyla, art arda gelen analiz pencereleri arasındaki zamansal bağımlılıkla ilgili bilgiyi dâhil etmek için dinamik özellikler işin içine sokulur. Bunun için statik parametrelerin peşine, hız katsayıları olarak bilinen 'delta' (birinci mertebeden zaman türevi) katsayıları ve ivme katsayıları olarak bilinen 'delta-delta' (ikinci mertebeden zaman türevi) katsayıları eklenir. Zaman türevi katsayıları, orijinal statik özelliklerdeki rasgele dalgalanmalara duyarlı olduğundan gürültülü olma eğilimi gösterir. Daha sağlam değişim ölçümü yapabilmek için genellikle analiz çerçeveleri üzerinde doğrusal yordama (*regression*) uygulanır. Sonuçta statik parametrelerin (MFCCs) peşine, delta ve delta-delta katsayıları eklenerek, 39 parametreli öznitelik vektörü oluşturulur.

II.2.3 Ses Tanımada Sınıflandırma Süreci

Modern ses tanıma sistemlerinde genel olarak kullanılan mimari, ses sinyalinden, olması muhtemel kelimeler serisi oluşturan yazılım mimarisi seklindedir. Bu mimarideki en popüler algoritmalar istatistiksel metotlara dayanır.

Karşılaştırma ve eşleştirme aşamasında en çok kullanılan teknikler; HMM (Hidden Markov Model), DTW (Dynamic Time Warping) ve Yapay Sinir Ağlarıdır.

II.2.3.1 Saklı Markov Modeli (Hidden Markov Model - HMM)

HMM, ayrık zamanlı serilerin, gözlenmiş veri örneklerinin sınıflandırılması için güçlü bir istatistiksel yöntemdir. Ses tanıma sistemlerinde çok sıklıkla kullanılır. Bu kadar popüler olmasının nedeni ise, zengin bir matematiksel yapıya sahip olması ve uygun olarak uygulandığında başarılı sonuçlar elde edilmesini sağlamasındandır.

Ses tanıma sistemlerinde, HMM'deki gözlem serileri, ses sinyalini simgeleyen özellik vektörleridir. Modelde görülen durumlardan kendisine geçişler de, fonem tabanlı bir ses tanıma sistemi için, bir fonemin uzun seslendirilmesinde aynı durumun tekrarlarını ifade etmek içindir. Kısa bir deyişle, ses tanıma sistemlerinde HMM'deki durumlar fonemlere karşılık gelir.

Ses tanıma sistemlerinde kullanılan, HMM'in amacı da bilinmeyen (saklı) durum dizisini gözlemlere dayanarak bulmaktır.

II.2.3.2 Yapay Sinir Ağları

Yapay sinir ağlarının kullanım alanı oldukça geniştir. Kullanım prensibi ise bilinmeyen bir fonksiyonun modellemesini yapmaya dayanır. Bunu yapabilmek için ilk olarak yapay sinir ağı modeli oluşturulmaktadır. Bilinmeyen fonksiyona ait gözlenen giriş ve çıkışlar, veri serisi olarak yapay sinir ağında eğitim amacıyla kullanılır. Yapay sinir ağının eğitimi tamamlandıktan sonra amaç, bu bilinmeyen fonksiyonun çıkışlarını üretebilme yeteneğine kavuşturulmasıdır. Yapay sinir ağına temel olarak insan beyninin yapısı, nöronlar ve aralarındaki bağlar alınmıştır. Yapay sinir ağlarına, insan beyninin basit bir modelidir denebilir.

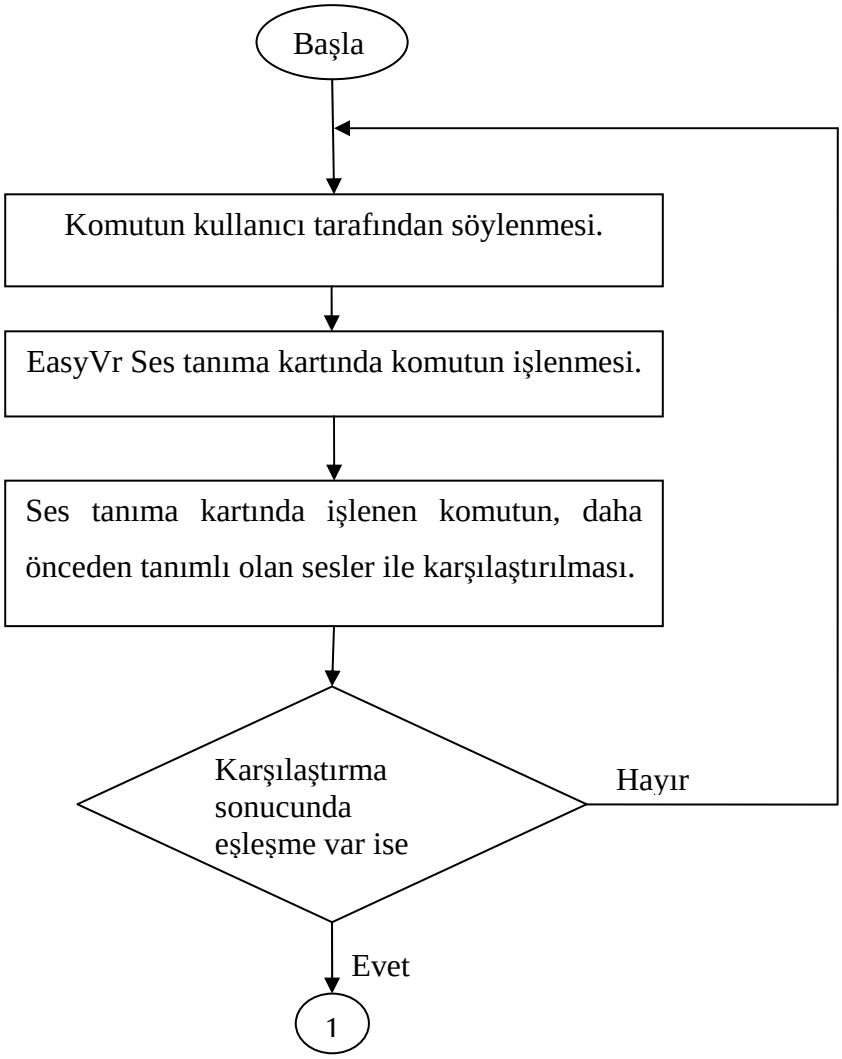
II.2.3.3 Dinamik Zaman Bükmesi (Dynamic Time Warping - DTW)

Belirli bir sözcüğün seslendirilmesi, aynı kişi seslendirse bile zaman içinde farklılık gösterebilmektedir. Aynı sözcüğün seslendirilmesi, bir seslendirmede uzun, bir seslendirmede ise daha kısa zamanda gerçekleştirilebilir. DTW algoritması yardımıyla, bu iki seslendirme, zaman içinde yayılarak ya da daraltılarak birbirine yaklaştırılmaya çalışılır. Yani bu iki seslendirmenin, zaman olarak örtüştürülmesi işlevi gerçekleştirilir. DTW algoritması, sözcük tabanlı ses tanıma sistemlerinde etkin ve sıkça kullanılan bir yöntemdir. Bu yaklaşımla, çalışma anında saptanan sözcük kesimlemesi, sistemde kayıtlı sözcük şablonları ile seslendirme zamanları örtüştürülerek karşılaştırılması gerçekleştirilebilir. Bu algoritmanın uygulanması dinamik programlama teknikleri kullanımı ile gerçekleştirilmektedir. İki ses sinyalinin arasındaki fark ve benzeşme çıkarılırken, büyük bir problem olan iki seri farkının hesaplanması, daha küçük bir problem olan seriden ikişer ikişer alınan örneklerin farklarının hesaplanması üzerine kurularak bu farklardan yararlanmaktadır.

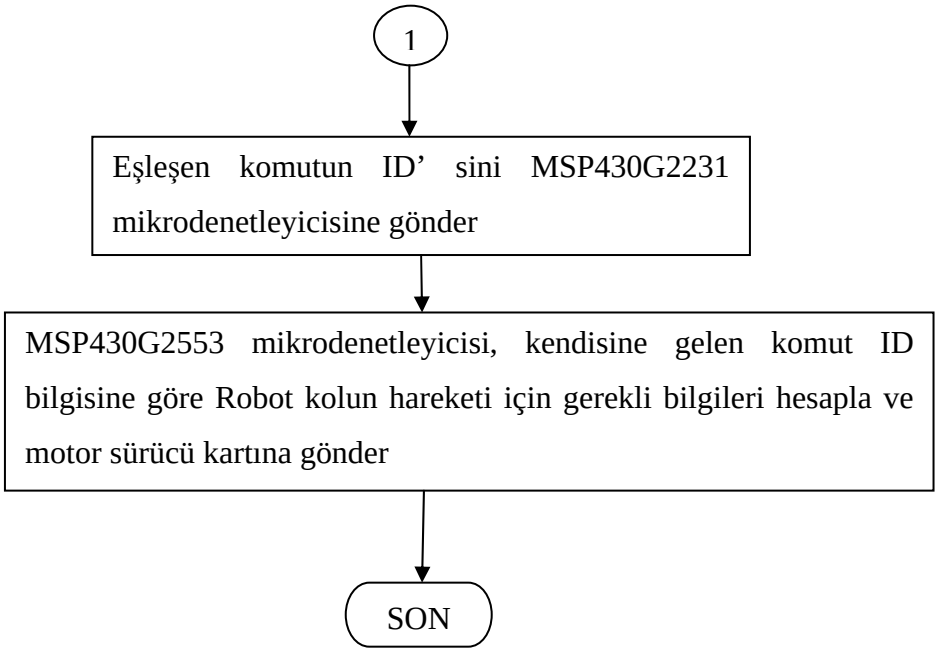
BÖLÜM III

GERÇEKLEŞTİRİLEN SES TANIMA VE KONTROL SİSTEMİ

Bu bölümde, yapılan tez çalışmasında kullanılan donanımsal bileşenlerin çalışması ile ilgili bilgiler verilecektir. Şekil III.1' de robotik bir sistemin ses ile kontrolünün akış diyagramı görülmektedir.



Şekil III. 1 Robotik bir sistemin ses ile kontrolünün akış diyagramı

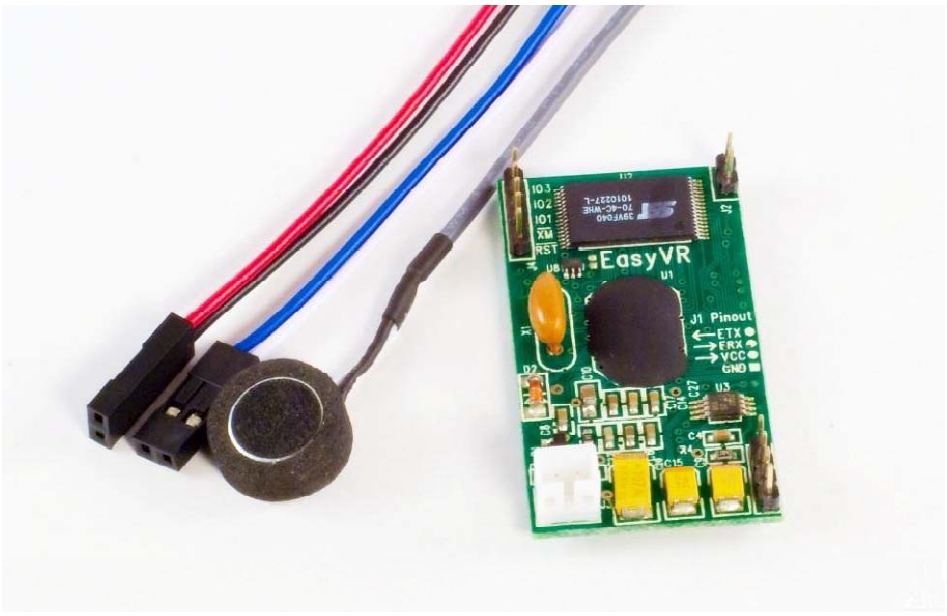


Şekil III. 1 Robotik bir sistemin ses ile kontrolünün akış diyagramı (devam)

Şekil III.1' deki akış diyagramında da görüldüğü gibi, öncelikli olarak ses bir mikrofon yardımı ile algılanıp, ses tanıma kiti ile kontrol edilmiştir. Algılanan ses bilgisi, daha önceden tanımlanmış bir ses ise, bu sesin hangi ses bilgisi olduğu kontrol edilip sonuç bilgisi mikrodenetleyiciye gönderilir. Mikrodenetleyici gelen bilgiye göre motor sürücü kartına, robot kolun hareketi için gerekli olan açı bilgisini gönderir. Bu işlem sonucunda robot kol ses ile kontrol edilmiş olur.

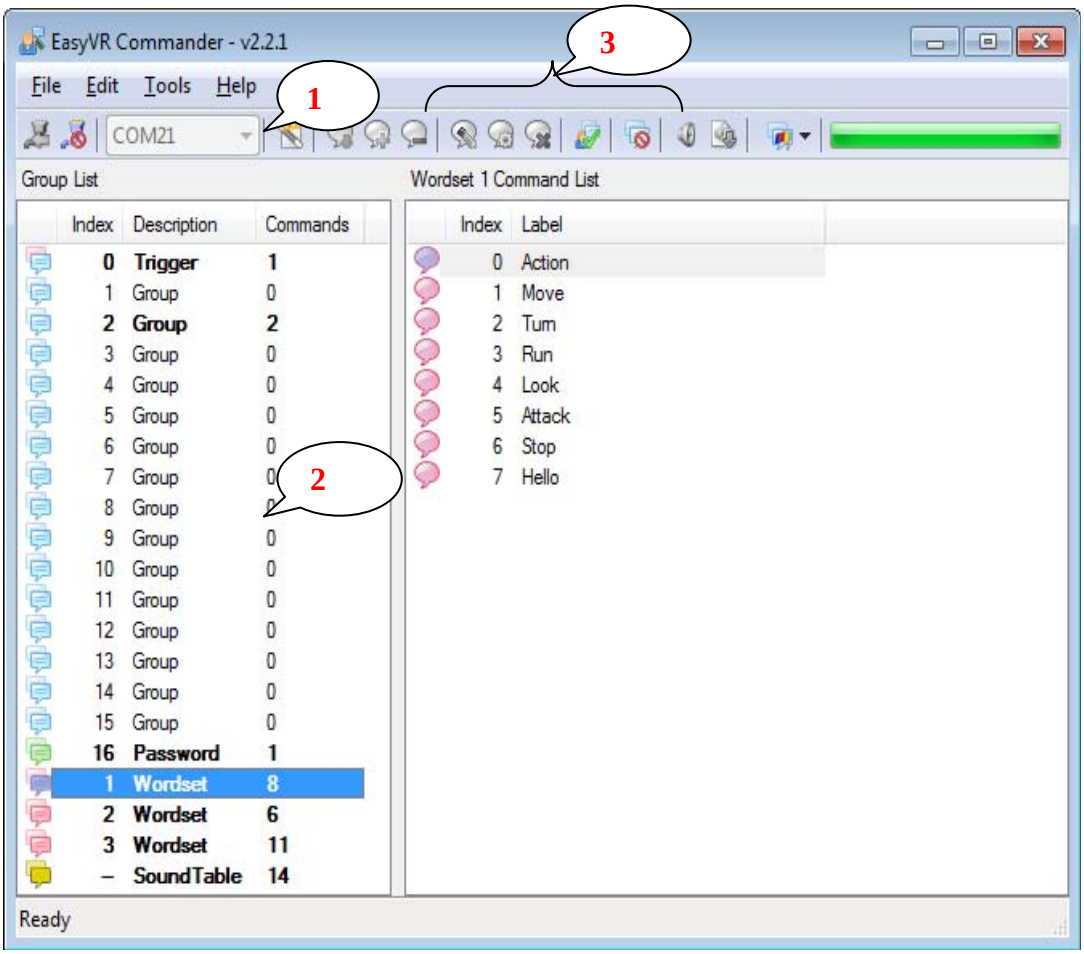
III.1 EASYVR SES TANIMA KARTI

Bilindiği üzere ses tanıma uygulamaları yüksek işlem gücü gerektiren karmaşık algoritmalardan oluşur. Genel amaçlı denetleyiciler ile gerçekleştirmesi pek mümkün değildir. EasyVr ses tanıma modülü sayesinde mikrodenetleyicili uygulamalarına kolaylıkla sesli kontrol özelliği eklenebilmektedir. Şekil III.2'de easyvr ses tanıma kartı görülmektedir.



Şekil III. 2 EasyVr Ses tanıma kartı

EasyVr modülünün genel olarak bahsedecek olursak; EasyVr modülü 26 konuşmacıdan bağımsız (SI-Speaker independent) ve 32 adet konuşmacı tanımlı (SD-Speaker Dependent) her biri maksimum 5 saniye olan sesli komuta sahiptir. SI komut seti İngilizce, Almanca, İtalyanca ve Japonca olmak üzere 4 dilde destek vermektedir. Kullanıcı tanımlı komutlar ise istenilen dilde tanımlanabilir. Modül ile haberleşmek için seri bağlantı gereklidir. EasyVr modülü seri haberleşmede 9600-11520 baud arası hızı desteklemektedir. Yeni ses komutu tanımlamak için Şekil III.3' te EasyVr modülünün basit bir ara yüzü görülmektedir. Seri port üzerinden ara yüze bağlanarak sesli komutlar istenilen grubuna kayıt edilebilir. Sonrasında yüklenen komutlar uygulamada mikrodenetleyici ile seri bağlantı ile kurarak kullanılabilir. Şekil III.5'te EasyVr ses tanıma modülü ile MSP430 mikrodenetleyicimiz arasındaki bağlantı görülmektedir. EasyVr modülü 3.3V – 5V arasındaki gerilim ile çalışmaktadır. Modüle bilgisayar ara yüzü yardımı ile konutlar tanımlandıktan sonra, bilgisayardan bağımsız olarak herhangi bir mikrodeneyleci ile bağlantı kurarak çalışabilmektedir.



Şekil III. 3 EasyVr modülünün bilgisayar ara yüz programı

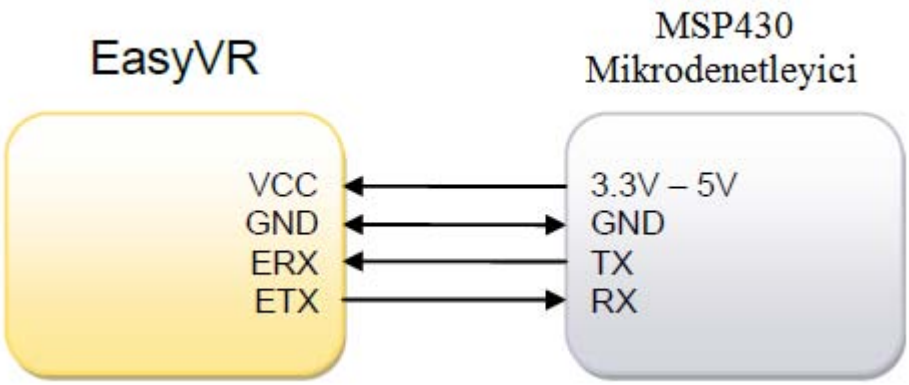
1. Şekil III.3'te 1 numaralı bölümde, ara yüz programı ile EasyVr modülünün bağlantısını gerçekleştirmek için öncelikle modülün bilgisayara hangi seri port'tan bağlandığını seçmek gereklidir, port ismi seçildikten sonra bağlantı işlemini gerçekleştirilmiştir.
2. Şekil III.3'te 2 numaralı bölümde, grup numaralarının olduğu bölüm, tanımlamak istediğimiz komutları gireceğimiz bölümdür. 15 adet gruptan hangisinde komut tanımlamışsa, mikrodenetleyici ile easyvr modülü arasında bağlantı kurulduğunda o grup numarası ile haberleşme yapılacağı belirtilmelidir.
3. Şekil III.3'te 3 numaralı bölümde yeni bir komut eklemek, eklenilen komuta isim vermek, ismini değiştirmek ve eklenen komutu easyvr modüle tanıtmak için kullanılan butonlar.

EasyVr modülünde eklemiş olan bir komutu modüle tanıtmak(eğitmek) için komutu seçtikten sonra Şekil III.3'teki 3.bölümde bulunan komut butonlarından TC butonuna tıklanır ve Şekil III.4'teki ekran gelir.



Şekil III. 4 Komut tanımlama(eğitme) ekranı

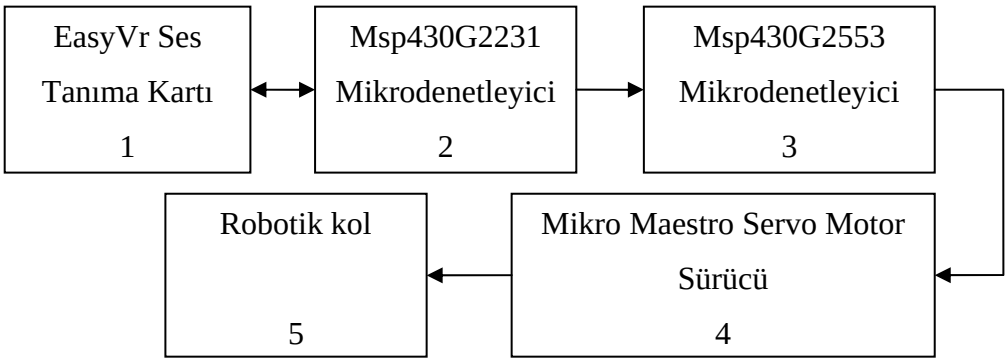
Komutu eğitmek için PH butonuna tıklanır ve komut için hangi kelimenin tanımlanması isteniyorsa o kelime söylenir.



Şekil III. 5 EasyVr ses tanıma kartı ve MSP430 Mikrodenetleyici arasındaki haberleşme

III.2 MSP 430 MİKRODENETLEYİCİ

Bu tez çalışmasında kullanılan msp430 mikrodenetleyicisi, ses tanıma kitinden alınan bilgiyi, robot kolun hareketi için gerekli olan açı bilgisine çevirmekte, gerekli olan eksen bilgisi ile birlikte motor sürücü kartına göndermektedir. Şekil III.6'da Msp430 mikrodenetleyicisinin diğer birimlerle olan haberleşmesinin blok şeması görülmektedir. 1 numaralı blokta ses tanıma kartında tanınan ses bilgisinin ID' si G2231 mikrodenetleyicisine gönderilmiştir. 2 numaralı blokta gelen ID bilgisi G2553 mikrodenetleyicisine gönderilmiştir. 3 numaralı blokta G2553 mikrodenetleyicisi ID bilgisinden motor sürücü kartına gönderilmesi gereken açı bilgisini hesaplayıp, motor sürücü karta göndermiştir. 4 numaralı blokta motor sürücü kart, gelen açı bilgisini, servo motor için gerekli olan elektrik sinyaline çevirip robotik kola göndermiştir.



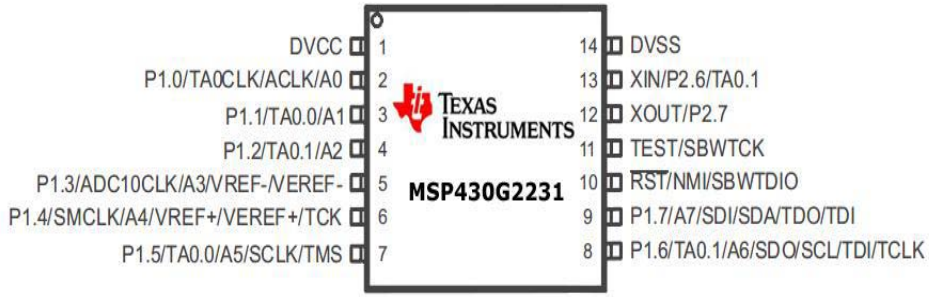
Şekil III. 6 Msp430 Mikrodenetleyici ve diğer donanım birimleri arasındaki bağlantı.

MSP430 Texas Instrument firmasının ürettiği çok düşük güç tüketimiyle ön plana çıkan bir mikrodenetleyicidir. 16 bit RISC mimarisine sahiptir. İçerisinde I2C, SPI, USART, ADC gibi klasik haline gelmiş birçok modülü barındırmaktadır. MSP430'u programlamak için IAR Embedded Workbench ya da Code Composer Studio kullanılmaktadır. Bu iki program da Texas Instrument sitesinden ücretsiz

indirilerek kullanılabilmektedir. Çalışmamızda Code Composer Studio'yu kullandık. Çalışmamızda kullandığımız mikrodenetleyici olarak MSP430G2231 ve MSP430G2553 modelleri seçilmiştir.

III.2.1 MSP430G2231 ve MSP430G2553 Mikrodenetleyicisi

MSP430G2231 mikrodenetleyicisi giriş seviyesinde, birçok işlevi yerine getiren bir mikrodenetleyicidir. G2231 denetleyicisinin genel yapısı [29] Şekil III.7'te görülmektedir.



Şekil III. 7 MSP430G2231 genel yapısı

	MSP430 G2553	MSP430 G2231
Program Memory(kB)	16	2
SRAM(Bytes)	512	128
I/O Pins	16	10
16-bit Timers	2	1
Capture/Compare Registers	3	2
USI:I2C/SPI	1	1
ADC Channels	8	8
ADC Resolution(Bits)	10	10

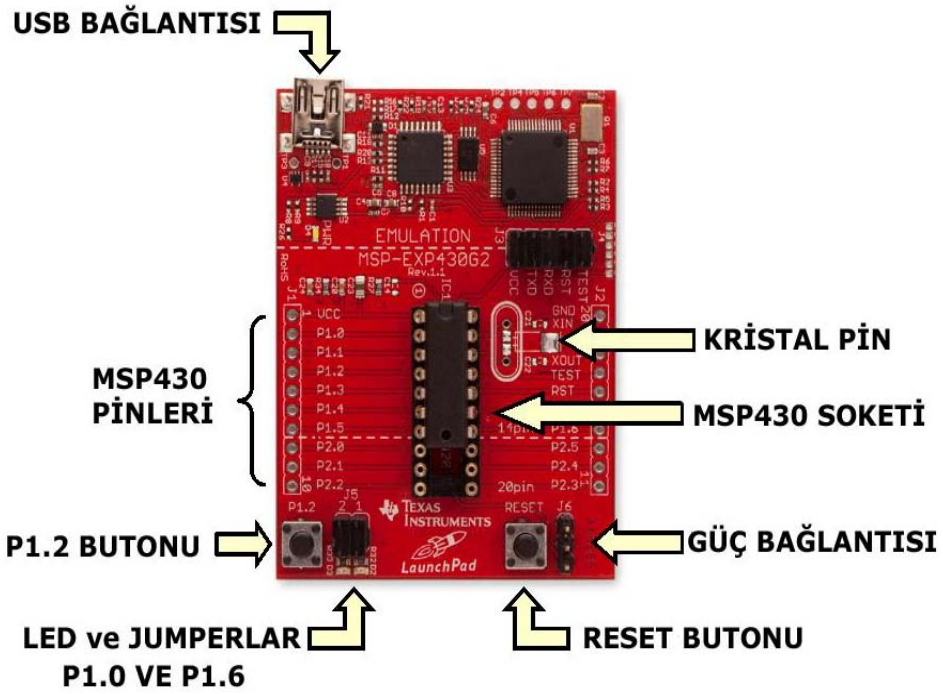
Tablo III. 1 MSP430G2231, MSP430G2553 Mikrodenetleyici Özellikleri [29]

G2231 mikrodenetleyicisi P1 ve P2 olmak üzere 2 adet giriş çıkış portu bulunmaktadır. Bu portlardan P1 portunun 8 adet giriş çıkışı bulunmaktadır, P2 portunun ise 2 adet giriş çıkışı bulunmaktadır. G2553 mikrodenetleyicisinde, P1 ve P2 portu bulunmaktadır. P1 portunun 8 adet giriş çıkışı bulunmakta, P2 portunun ise

6 adet giriş çıkış bulunmaktadır. Texas Instrument MSP430 mikrodenetleyicileri programlayabilmek için LaunchPAD isimli bir deney kartını sunmaktadır. MSP430G2231 mikrodenetleyicisi LaunchPAD deney kartı ile birlikte satışa sunulmaktadır [29].

III.2.1.1 LaunchPad ve Üzerindeki Bağlantılar

Şekil III.8'te LaunchPad deney kartının üzerindeki giriş-çıkışlar ve bağlantı birimleri görülmektedir [29].



Şekil III. 8 LaunhPad ve üzerindeki bağlantılar.

USB bağlantısı ile, bilgisayar ortamında yazmış olduğumuz programlarımızı launchpad üzerinde bulunan mikrodenetleyimize aktarmak için bilgisayar-launchpad arasındaki iletişimi sağlamaktadır.

MSP430 pinleri, launchpad üzerinde takılı bulunan mikrodenetleyicinin, bacak bağlantılarıdır ve bu bağlantılar kullanarak deney kartı üzerinden mikrodenetleyicinin portlarına bilgi gönderip alınabilir.

P1.2 butonu, mikrodenetleyicinin 1.portunun 2.bacağına bağlanmıştır. Eğer buton gerektiren bir uygulama geliştirilirse bu buton kullanılabilir.

Led ve jumperlar, mikrodenetleyicinin 1.portunun 0 ve 6 numaralı bacaklarına bağlı olan ledlerdir. Geliştirmiş olduğumuz bir uygulamada bir led'e ihtiyaç duyarsak, dışarıdan bir led bağlamadan bu ledleri kullanabiliriz.

Reset(yeniden başlat) butonu, mikrodenetleyicimiz çalışırken, çalıştırmış olduğu programı yeniden başlatmak için kullanılır.

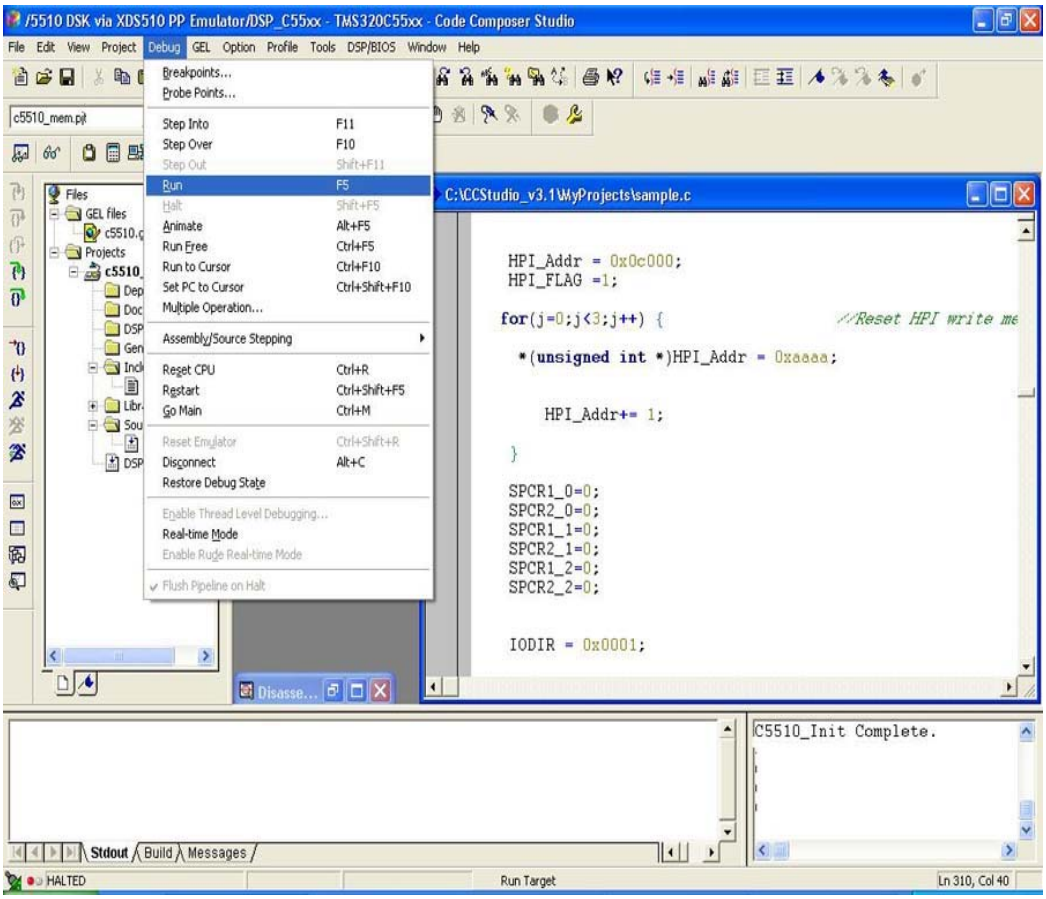
Güç bağlantısı, launchpad bilgisayara bağlı olduğunda mikrodenetleyici için gerekli olan enerjiyi USB kablosu üzerinden sağlamaktadır. Eğer bilgisayara bağlı değilse gerekli olan enerji güç bağlantısı üzerinden verilmektedir.

MSP430 soketi, lauchpad deney kartı birden fazla MSP430 mikrodenetleyici modelini programlayabilmektedir. Kullanılacak mikrodenetleyici bu sokete yerleştirilir.

Kristal Pin, MSP430 mikrodenetleyicilerine eğer dışarıdan bir osilatör kristal takılacaksa bu pinleri kullanarak kristal takılır.

III.2.2 Code Composer Studio Programlama editörü

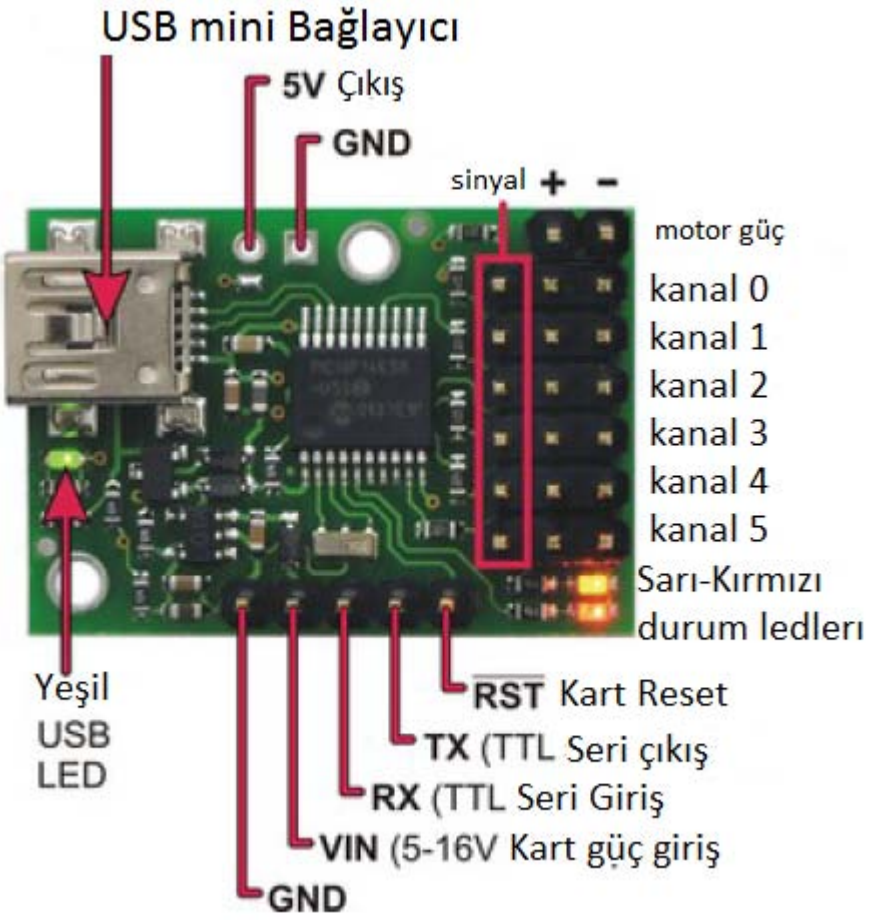
Code Composer Studio (CCS) Texas Instruments firmasının bütün işlemci aileleri için geliştirdiği genel bir yazılım geliştirme, derleme ve deneme ortamıdır. CCS'de yazılım geliştirirken kullanılan programlama dili C'dir. CCS Texas Instruments şirketinin internet sitesinden indirilip kurulmaktadır. Şekil III.9'da CCS'nin çalışma ekranı görülmektedir.



Şekil III. 9 Code Composer Studio çalışma ekranı

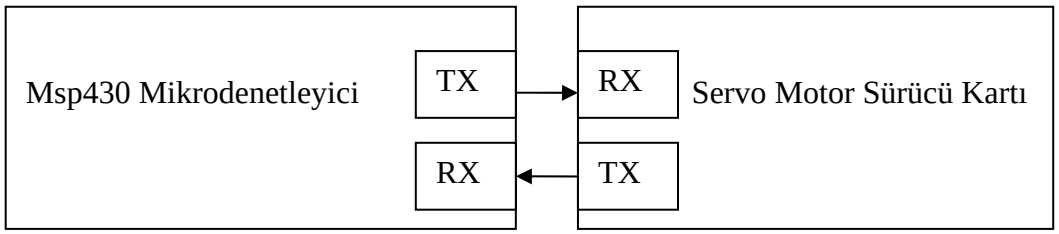
III.3 MİCRO MAESTRO 6 KANAL SERVO MOTOR SÜRÜCÜ

Servo motor kontrolü için Altı-kanal Micro Maestro motor sürücü kartı kullanılmıştır. Micro Maestro son derece küçük boyutlarda (2.16 x 3.05 cm), çok yönlü servo denetleyicisi ve genel amaçlı I / O kartıdır. Üç farklı kontrol yöntemini destekler. Bir bilgisayara doğrudan bağlantı için USB, gömülü sistemlerde kullanmak için TTL serial, ve ana bilgisayar olmaksızın çalışabilen uygulamalar için dahili komut dosyası. Şekil III.10'da servo motor kontrol kartının şekli görülmektedir.



Şekil III. 10 Micro Maestro 6 kanal servo motor sürücü kartı

Şekil III.10'da servo motor sürü kartı, kullanım şekillerine göre 3'e ayrılmaktadır. Birinci yöntem olarak USB bağlantısı ile bilgisayara bağlanıp bilgisayarda kart için yazılmış olan bir ara yüz ile kart kontrol edilmektedir. İkincisi 6 kanal servo sürücü kartını üreten firma tarafından oluşturulmuş bir programlama dili ile kart üzerindeki mikrodenetleyiciye program yazılarak, bilgisayardan bağımsız olarak çalıştırılabilmektedir. Üçüncü olarak bir mikrodenetleyici ile motor sürücü kartı arasında seri haberleşme kurularak kontrol işlemi gerçekleştirilmektedir. Bu tez çalışmasında 3.yöntem kullanılmıştır. MSP430G2231 mikrodenetleyicisi ile motor sürücü kartı arasında seri haberleşme kullanılarak, mikrodenetleyiciden gönderilen açı bilgisi ile robot kolun hareketi gerçekleştirilmiştir.



Şekil III. 11 Micro Maestro 6 kanal Servo Motor Sürücü kartı ve MSP430G2231 Mikrodenetleyici arasındaki Haberleşme.

Şekil III.11'de mikrodenetleyici ve servo motor sürücü kartı arasındaki haberleşme gösterilmiştir. Mikrodenetleyicinin motor sürücü kartına gönderdiği paket yapısı Tablo III.2'de gösterilmiştir.

	1	2	3	4
Paket Bilgisi	0x84	0x00	0x00	0x00

Tablo III. 2 Mikrodenetleyiciden motor sürcü kartına gönderilen paket yapısı

1. İlk gönderilen bilgi motor sürücü kartı ile mikrodenetleyici arasındaki iletişim hızının eşitlenmesi için gönderilmektedir.
2. İkinci Byte ise motor sürücü kartı üzerinde 6 adet servo motor sürmek için çıkış bulunmaktadır, ikinci byte'ta servo motor numarası gönderilmektedir. Örneğin Tablo III.2'de 0 numaralı motora bilgi gönderilmektedir.
3. 3. ve 4. Byte'lar ise motorun hareket etmesi gereken açı bilgisini ifade etmektedir, Tablo III.2'de iki 3. Ve 4. Veri sıfır olduğundan açı bilgisi 0 derece olarak gönderilecektir.

III.4 ROBOTİK KOL

Robot kolu; yapacağı işe göre dönel ya da bulunduğu eksende düz hareket yapabilen eklemlere sahip, tutma mekanizmaları olan, mikroişlemci, mikrodenetleyici veya diğer bilgi işlem elemanları ile kontrol edilebilen elektromekanik bir sistemdir. Bu tez çalışmasında, beş serbestlik dereceli, beş dönel

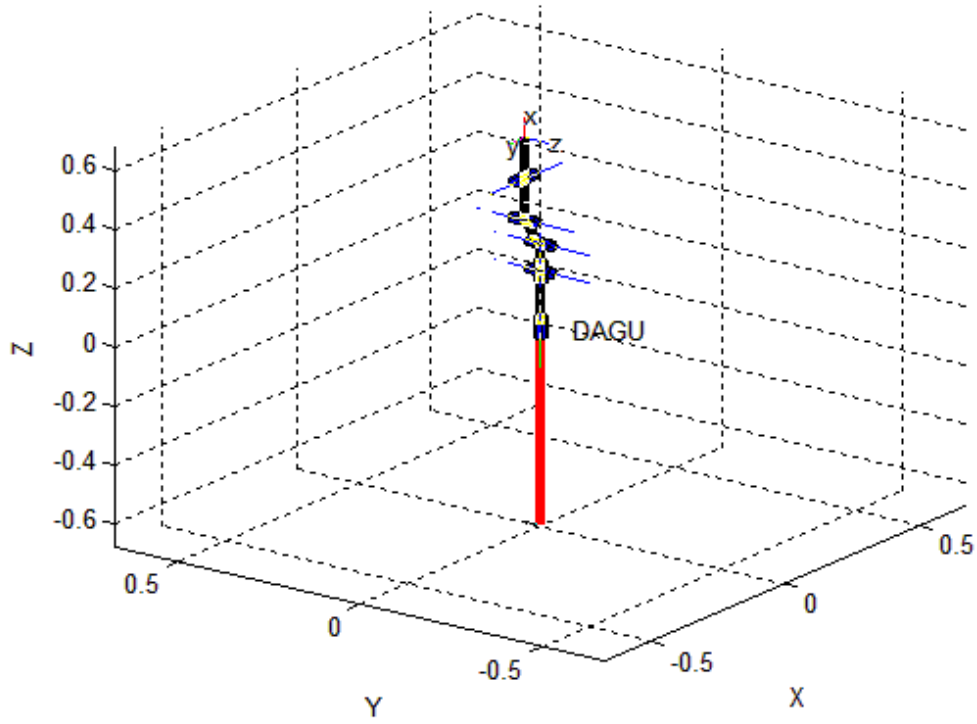
ekleme ve bir uç işlevcisine sahip bir robot kolunun eklem hareketlerini ses ile kontrol edebilen bir sistemin oluşturulması amaçlanmıştır. Şekil III.12'de çalışmamızda kullanmış olduğumuz robot kol görülmektedir.



Şekil III. 12 Tez çalışmasında kullanılan robotik kol

Şekil III.12'deki robotik kolda 6 tane servo motor bulunmaktadır, bu servo motorlardan 5 tanesi robot kolun eklem bileşenlerini hareket ettirmede kullanılmakta, 6. Servo motor ise uç bileşenini açıp kapatmak için kullanılmaktadır.

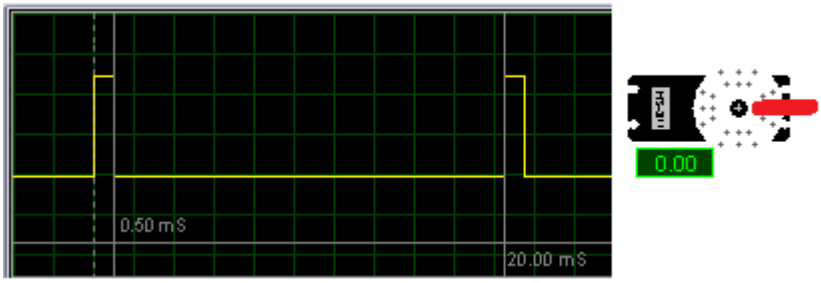
Tez çalışmasında kullandığımız robot kolun, MATLAB programında ileri yön kinematiği hesaplanmış ve simülasyon ortamında gösterilmiştir. Şekil III.13'te robot kolun simülasyon ortamındaki görüntüsü görülmektedir [35].



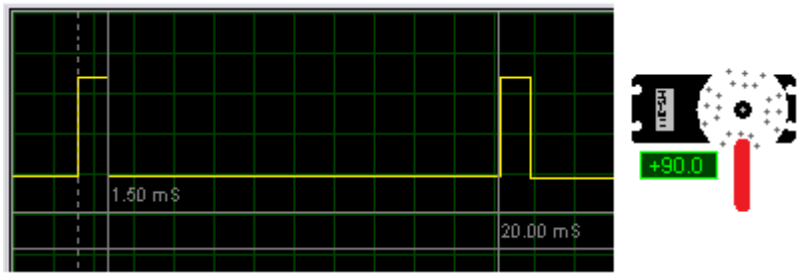
Şekil III. 13 İleri yön kinematik hesabı simülasyonu

III.4.1 Servo Motorlar

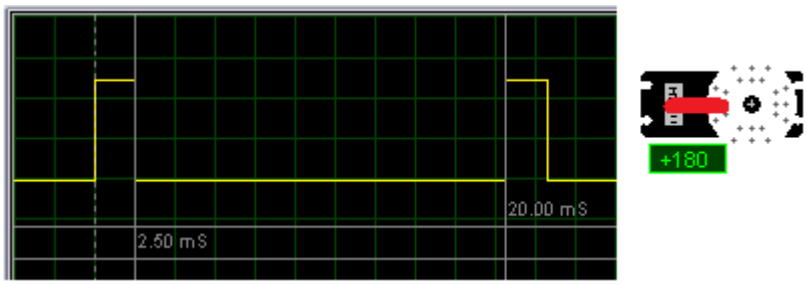
Servo motorların mili normal bir doğru akım veya alternatif akım motorunun mili gibi sürekli dönme hareketi yapmaz. Servo motorların üç tane bağlantı ucu bulunur, bunlar Besleme ucu, toprak ucu ve Mil pozisyonunu belirleyen kontrol sinyalinin uygulandığı Kontrol ucudur. Servo motorun mili kontrol ucuna Darbe genişlik modülasyonu (Pulse Width Modulation-PWM) tekniği ile uygulanan sinyale göre 0–180° arasında bir açı değeri alır ve kontrol sinyali değişmediği sürece mil bulunduğu konumu korur. Servo motorun sadece besleme uçlarına Doğru gerilim uygulanırsa herhangi bir hareket yapmaz. Servo motor milinin bir açı değeri alması için kontrol ucuna, modülasyon genişliği 0,5 ms ile 2,5 ms arasında değişen bir sinyalin uygulanması gerekir. Milin alacağı açı değeri 0,5–2,5 ms ‘lik modülasyon genişliği ile doğru orantılı olarak 0–180° arasında değişir [24]. Örneğin 0,5ms‘lik bir modülasyon genişliğindeki bir sinyal ile mil 0° Şekil III.14’te görülmektedir, 1.5ms ‘lik modülasyon genişliğinde 90° Şekil III.15’te görülmektedir ve 2,5 ms ‘lik modülasyon genişliğinde ise 180° açı değeri alır Şekil III.16’da görülmektedir.



Şekil III. 14 0,5ms ‘lik modülasyon genişliği ve motor milinin aldığı açı değeri



Şekil III. 15 1,5ms ‘lik modülasyon genişliği ve motor milinin aldığı açı değeri



Şekil III. 16 2,5ms ‘lik modülasyon genişliği ve motor milinin aldığı açı değeri

BÖLÜM IV

DENEYSEL ÇALIŞMA

Yapılan çalışmada ses ile robotik kol kontrolü için 16 adet komut tanımlanmıştır. Bu komutlar tablo IV.1'de görülmektedir.

Komut ID'si	Komut Adı
0	BİR
1	İKİ
2	ÜÇ
3	DÖRT
4	BEŞ
5	ALTI
6	SAĞ
7	SOL
8	YUKARI
9	AŞAĞI
10	ÇOK
11	AZ
12	BİRAZ
13	TAMAM
14	BAŞLA
15	SIFIRLA

Tablo IV. 1 Kullanılan komutların listesi.

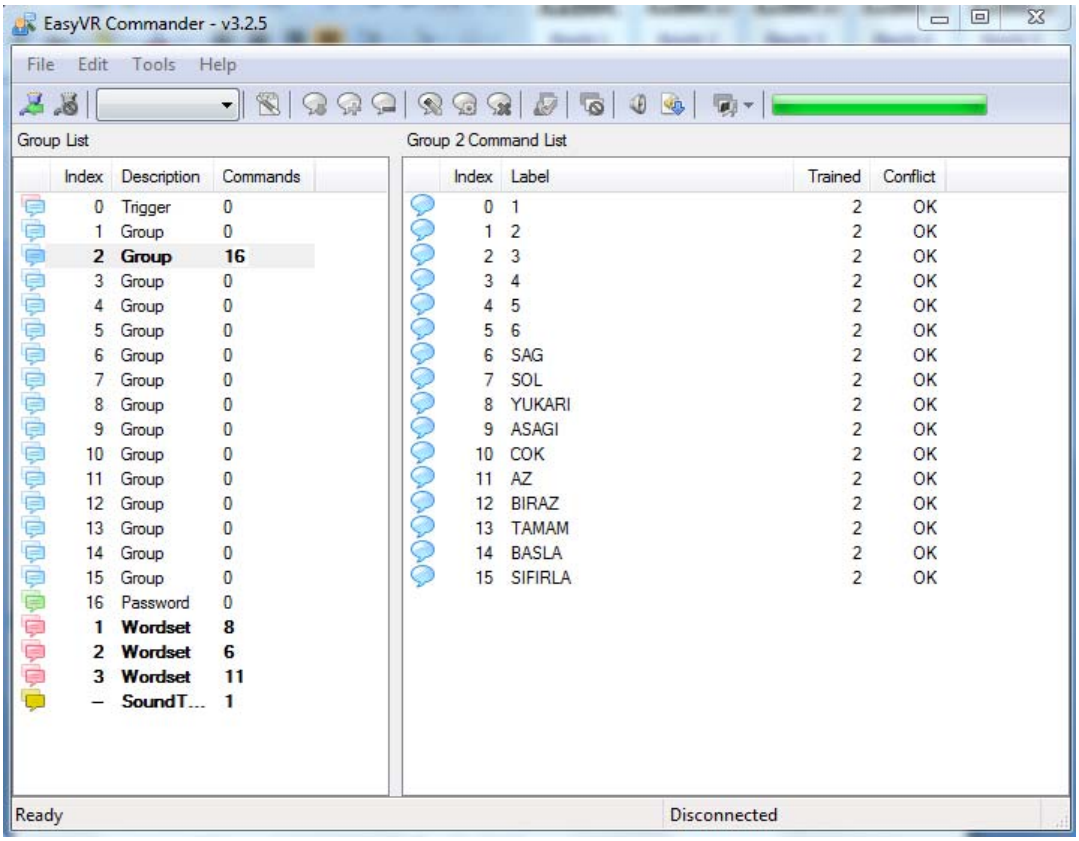
Tablo IV.1' deki komutlar ile robotik kolu ses ile kontrol etmek için, her bir eksenin hangi yöne hareket etmesi gerektiğini ve bu hareketin kaç derece olacağı bildirildi. Bu komutları ses tanıma kartına tanımlamak için EasyVr ses tanıma kartının bilgisayar ortamındaki ara yüz programını kullanıldı. Şekil IV.1' de

tanımlanmış olan komutların ara yüz programındaki listesi görülmektedir. Robot kolu ses ile kontrol etmek için, sırası ile verilmesi gereken komutlar Tablo IV.2' de görülmektedir.

Komutlar						
Eksen No	Yön	Miktar	Tamam	Eksen No	……	Başla

Tablo IV. 2 Robot kolun ses ile kontrolünde komutların söyleniş sırası

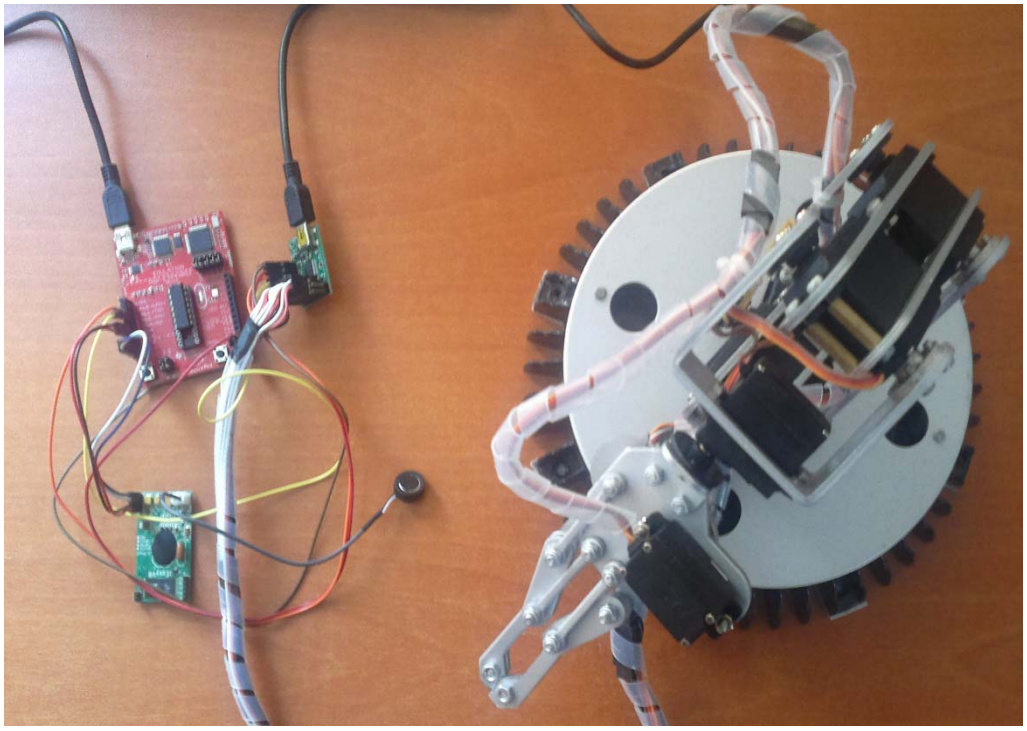
Kullanıcı robot kola ses ile komut verirken, Tablo IV.2' deki sırayı izlemelidir. Öncelikle hareket ettirilmek istenilen eksen numarası, daha sonra hangi yöne hareket etmesi gerektiğini, daha sonra hareket etmesi gereken miktar, daha sonra ise tamam komutu verilir. Bu işlem ses ile kontrol edilmek istenilen tüm eksenler için ayrı ayrı tekrarlanır. Ses ile kontrol edilmek istenilen eksenler için komutlar verildikten sonra istenilen hareketin gerçekleşmesi için başla komutu verilir ve eksenler istenilen hareketi gerçekleştirilir.



Şekil IV. 1 EasyVr ses tanımlama kartı ara yüz programı tanımlanan komut listesi

Ses ile robotik kolu kontrol edebilmek için öncelikle uygulamada kullanılması gereken sesleri EasyVr ses tanıma kartında tanımlandı. Bunun için Şekil IV.1' deki EasyVr ses tanıma modülünün bilgisayar ara yüzünü kullanıldı. EasyVr ses tanıma modülü bilgisayar ile seri olarak haberleşmektedir. Kullanılan bilgisayarda seri port bulunmadığından bu bağlantı bir USB seri dönüştürücü ile gerçekleştirildi.

Tablo IV.1'de, çalışmada kullanılan komutların listesi ve her bir komutun kendisine ait bir ID si bulunmaktadır. Bir ses komutu ses tanıma kartında tanımlandığında, mikrodenetleyiciye o komuta ait ID bilgisi gönderilmiştir. Mikrodenetleyici içersine yazılan programda ise kullanıcının hangi ekseni ne kadar hareket ettirmek istediği bilgisi motor sürücü kartına gönderilmiştir. Şekil IV.2'de robotik kol, mikrodenetleyici ve ses tanıma kartı deney düzeneği görülmektedir.



Şekil IV. 2 Robot kolu deney düzeneği

Robot kolu ses ile kontrol etmek için aşağıdaki adımların izlenmesi gerekmektedir.

1. Eksen numarasını söyle,
2. Hangi yöne hareket etmesi gerektiğini söyle,
3. Eksenin hareket etmesi gereken miktarı söyle,
4. Bir ve iki numaraları komutları söyledikten sonra tamam komutunu söyle.
5. İlk 3 adımı istenilen tüm eksen numaraları için söyledikten sonra başla komutunu söyle.

Robot kolun ses ile kontrolünde eksenlerin hareket miktarları için 3 ayrı seçenek bulunmaktadır;

1. Az
2. Çok
3. Biraz

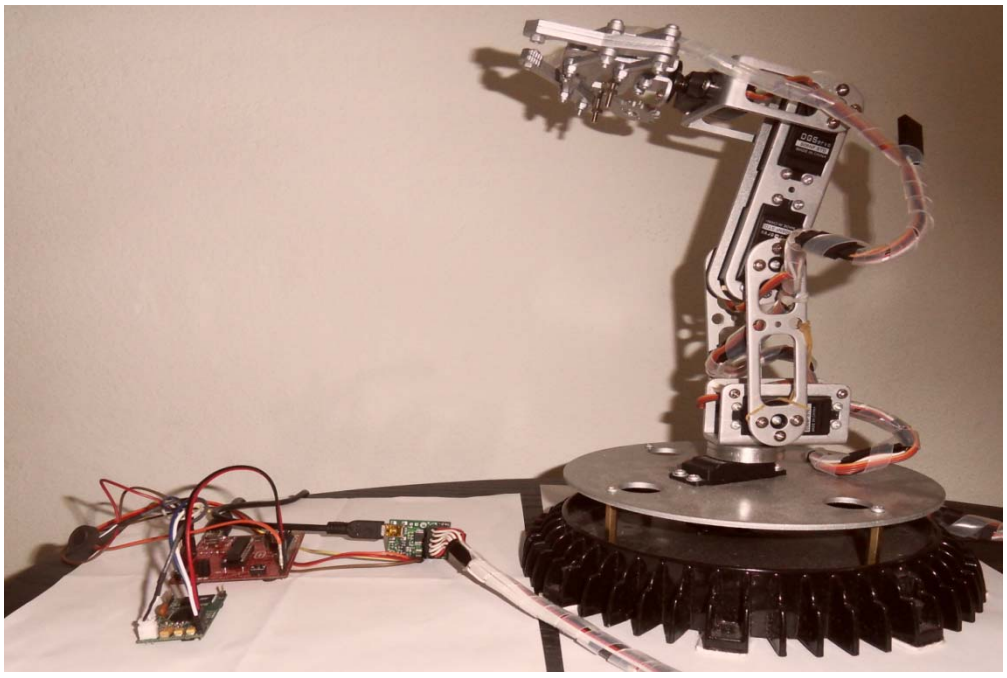
Kullanıcı robot kolun eksen hareket miktarı için bu üç seçenekten birini seçmelidir.

1, 2, 3, 5 numaralı eksenleri hareket ettirmek için gerekli olan komutlar Tablo IV.2'de görülmektedir. Bu komular söylendikten sonra başla komutunun verilmesi ile birlikte 1, 2, 3, 5 numaralı eksenler aynı anda istenilen hareketi gerçekleştirmiştir.

Eksen numarası	Hareket yönü	Hareket miktarı
1	Sağ	Biraz
2	Yukarı	Çok
3	Yukarı	Çok
5	Aşağı	Az

Tablo IV. 3 4 eksenin hareketi için gerekli komutlar

Robot kol, Tablo IV.3' deki komutların verilmesinden sonra, Şekil IV.3'deki konuma gelmiştir.



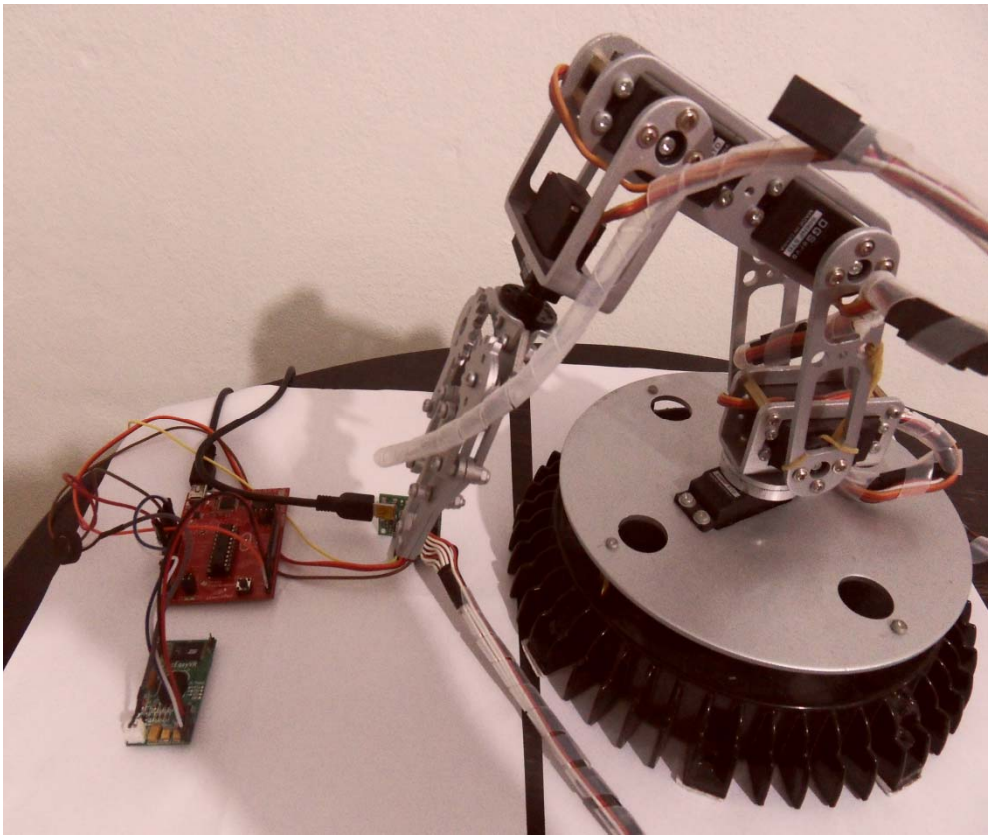
Şekil IV. 3 4 eksenin hareketi sonrasındaki robot kolun geldiği konumu

Robot kol Şekil IV.3'deki konumunda sabit durumda iken, 3 ve 4 numaralı eksenleri hareket ettirmek için Tablo IV.4'teki komutlar söylenmiştir.

Eksen numarası	Hareket yönü	Hareket miktarı
3	Aşağı	Çok
4	Aşağı	Az

Tablo IV. 4 2 eksenin hareketi için gerekli komutlar

Tablo IV.4'teki komutlar söylendikten sonra robot kol Şekil IV.4'deki konuma gelmiştir.



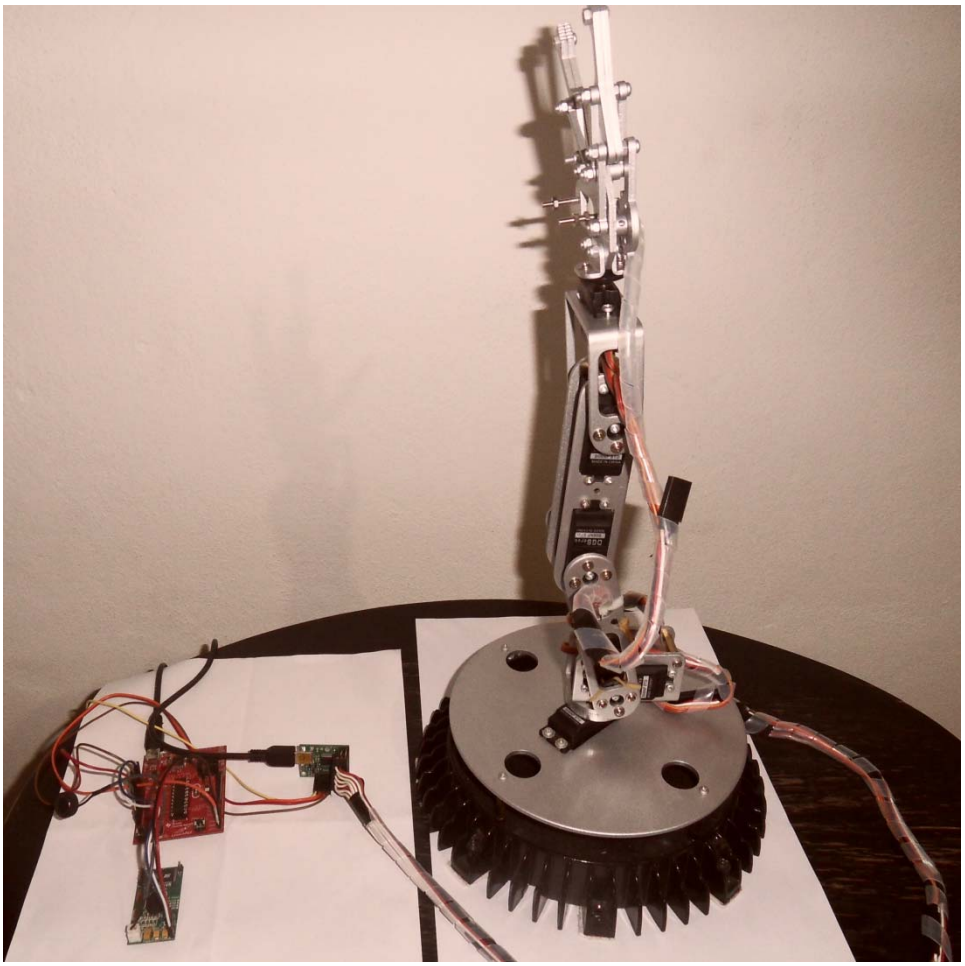
Şekil IV. 4 2 eksenin hareketi sonrasındaki robot kolun konumu

Robot kol Şekil IV.4'deki konumunda sabit durumda iken, 1, 3 ve 4 numaralı eksenleri hareket ettirmek için Tablo IV.5'deki komutlar söylenmiştir.

Eksen numarası	Hareket yönü	Hareket miktarı
1	Sol	Az
3	Yukarı	Çok
4	Yukarı	Çok

Tablo IV. 5 3 eksenin hareketi için gerekli komutlar

Tablo IV.4'deki komutlar söylendikten sonra robot kol Şekil IV.5'deki konuma gelmiştir.



Şekil IV. 5 2 eksenin hareketi sonrasındaki robot kolun konumu

BÖLÜM V

SONUÇLAR VE TARTIŞMA

Bu tez çalışmasında, ses komutlarını algılayıp tanımlayan ve bu komutlara uygun olarak robotik bir kolu istenilen şekilde hareket ettiren bilgisayardan bağımsız olarak çalışan bir sitem gerçekleştirilmiştir.

Ses ile robotik kolun kontrol edilmesi sisteminde, ses komutlarını tanımada EasyVr ses tanımlama modülü kullanılmıştır, robot kolun eksenlerinin hareket etmesi sağlamak için micro maestro servo motor sürücü kartı kullanılmıştır, EasyVr ses tanıma modülü ile micro maestro servo motor sürücü kartı arasındaki iletişimi sağlamak için ise MSP430 mikrokontroller kullanılmıştır.

Yapılan deneysel çalışmalarda, kullanılan komutlar birden fazla robot kol üzerinde denenmiş ve söylenen komutların robot kol için istenilen hareketi gerçekleştirip gerçekleştirmediği test edilmiştir. Tablo V.I' de robot kol üzerinde denenmiş olan komutlar verilmiştir. Robot kol üzerinde tanım olan her bir komut 30 defa test edilmiş bu test sonucunda %85,4 başarı oranı elde edilmiştir.

Komut Adı	Test Sayısı	Doğru Sayısı
BİR	30	29
İKİ	30	28
ÜÇ	30	25
DÖRT	30	28
BEŞ	30	25
ALTI	30	29
SAĞ	30	24
SOL	30	27
YUKARI	30	24
AŞAĞI	30	26
ÇOK	30	20
AZ	30	27
BİRAZ	30	25
TAMAM	30	21
BAŞLA	30	24
SIFIRLA	30	28

Tablo V. 1 Test Sonuçları

Tablo V.1'de görüldüğü gibi robot kolun ses ile kontrolünde kullanılan komutların tanınma doğrulukları verilmiştir. Robot kol üzerindeki doğruluk oranı ise, Tablo V.1'de örneğin yukarı komutu 30 kez test edilmiş bu 30 test içerisinden 24 tanesi doğru olarak bulunmuştur. Yapılan çalışmada kullanılan komutlar kullanıcı bağımlı olduğundan, robot kolun ses ile kontrolü sadece bir kişi tarafından mümkün olmaktadır. Bunun yanında test aşamasında komutların hatalı tanınmasında, çevre şartlarının büyük etkisi olmuştur. Ses komutu ile dış ortamdan gelen gürültüler, söylemiş olduğumuz komutun hatalı olarak tanınmasına neden olmuştur, bu da sistem başarımını düşüren etkenler arasında yer almaktadır.

BÖLÜM VI

DEĞERLENDİRMELER VE ÖNERİLER

Bu tez çalışmasında, ses komutu ile robotik bir kolun kontrolü, bir bilgisayara ihtiyaç duyulmadan, mikrodenetleyicili sistemler yardımı ile başarılı bir şekilde gerçekleştirilmiştir.

Bilgisayara bağlı olunmaması sistemin mobil olarak birçok ortamda kolayca kullanılabilmesine imkân tanımaktadır. Aynı zamanda bilgisayarlı bir alt yapı gerektirmemesi, sistemin kurulumunu, kullanımını ve maliyetini büyük oranda düşürmüştür.

Yapılan bu çalışmada kişiye bağımlı, ses komutu ile robotik bir kolun kontrolü sağlanmıştır. Gerçekleştirilen bu sistem kişiden bağımsız olarak çalışabilen yani fonem tabanlı bir sistem gerçekleştirilebilir.

Yapmış olduğumuz çalışmada robot kolun hareketi için her bir eksenin hareket bilgileri kullanıcı tarafından ses komutu olarak verilmektedir. Örneğin robot kola bir masa üzerindeki nesneyi aldırıp, başka bir yere taşıtmak istediğimizde her bir eksenin masa üzerindeki nesneye ulaşması için yapması gereken açı hareketini, tek tek söylüyoruz. Her bir eksene tek tek hareket bilgisini söylemek epeyce zaman almaktadır, bunun yerine video işleme tekniklerini kullanarak, robot kol üzerine yerleştirilmiş bir video kamera yardımı ile istenilen nesnenin sadece adını verip daha sonrasında robot kolun video kameradan o nesneyi tanıyıp ilgili nesneye hareket etmesini sağlayacak bir sistem geliştirilebilir.

Ses tanıma sitemlerindeki performansı etkileyen genel hatalardan biri olan, dış çevreden gelen gürültülerin, ses komutlarına karışmasını en az etkileyecek gürültü engelleyici algoritmalar sağlanabilir.

Sitemin performansını etkileyen bir diğer etken ise kayıt cihazlarının kalitesidir. Kullanılan mikrofonun kaliteli olması, performansı etkilediği görülmüştür.

KAYNAKLAR

[1] Yılmaz, M, " Step Motor ile İki Eksenli Robot Kol Tasarımı ", Yüksek Lisans Tezi, Van Yüzüncü Yıl Üni. F.B.E, Van,**(2007)**

[2] http://tr.wikipedia.org/wiki/Robot

[3] Furui, S., "Speech Recognition Technology in the Ubiquitous/Wearable Computing Environment",Acoustics-Speech and Signal Processing, ICASSP '00. Proceedings. 2000 IEEE International Conference, 6,3735-3738, **(2000)**

[4] Thong, J-M, vd., "SpeechBot: a Speech Recognition based Audio Indexing System for the Web", International Conference on Computer-Assisted Information Retrieval, RIAO, 106-115, **(2000)**

[5] Davis, C.D., "Automatic Speech Recognition and Access: 20 years, 20 Months, or Tomorrow?", Issue of Hearing Loss: The Journal of Self Help for Hard of Hearing People, 22, No. 4, 11-14, **(2001)**

[6] Wald, M., "Using Automatic Speech Recognition to Enhance Education for All Students: Turning a Vision into Reality", 34th ASEE/IEEE Frontiers in Education Conference, Savannah, GA, **(2004)**

[7] Li, M., and Okamura, A.M., "Recognition of Operator Motions for Real Time Assistance using Virtual Fixtures", Proc. of the11th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems, (HAPTICS'03), **(2003)**

[8] Russell, M., vd., "Applications of Automatic Speech Recognition of Speech and Language Development in Yough Children", British Crown Copyright/DERA, **(1996)**

[9] Hemphill, C. T., Agarwal, R., Muthusamy, Y. K., and Gong, Y. "Voice-Driven Information Access in the Automobile", *IEEE Vehicular Technology Society News,Augut,8,11,***(2000)**

[10] Ozdemircan, Z.M "Robot Control With Voice Command" Yıldız Technical University Faculty Of Electrıc And Electronics Department Of Computer Engineering Senior Project İstanbul ,**(2008)**

[11] M. Uysal, MS Visual C# .NET ile Yazılım Gelistirme, 1. Baskı, Beta Yayıncılık, İstanbul, **(2003)**

[12] M. Yanık MS Visual C# .NET, 2. Baskı, Seçkin Yayıncılık, Ankara**,(2004)**

[13] H. Palaz, TREN-Türkçe Konusma Tanıma Platformu, Tübitak- Ulusal Elektronik ve Kriptoloji Enstitüsü, **(2005)**

[14] B. Akçay, ”Yapay Sinir Ağları ile Türkçe Konuşma Tanıma” Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, **(1994)**

[15] ÇAKIR H, OKUTAN B, “Ses Kontrollü Web Tarayıcı” Bilgisayar Egitimi Bölümü, Gazi Üniversitesi, Bilişim Teknolojileri Dergisi, Cilt: 4, Sayı: 1, Ocak **(2011)** Ankara, Türkiye.

[16] B.Çengelci, H.Çimen, *“Endüstriyel Robotlar”*, Makine Teknolojileri Elektronik Dergisi **(2005)** (2) 69-78.

[17] Koca, H.: “Robot Manipülatör Denetimi”, *Yüksek Lisans Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye,**(2007)**

[18] Sciavicco, L.; Siciliano, B.: “Modeling and control of robot manipulators”, *The McGraw-Hill Companies, Inc*., New York, USA, **(1996)** 11-26

[19] Albayrak, A.: “Beş Eksenli Bir Robot Kolunun Simülasyonu ve Kontrolü”, *Yüksek Lisans Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye,**(2007)**

[20] Bingül Z.; Küçük, S.: *“Robot Teknigi-I-”*, Birsen Yayınevi, İstanbul,Türkiye, **(2005)** ,1-17, 23-59, 104-120

[21] S.Ögnir, *“Bilgisayar Kontrollü Kartezyen Robot”*, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, İstanbul, (**2011)**

[22] http://www2.omu.edu.tr/akademik_birimler/muhendislik/elektr/Lego_sitesi4/Manip.htm, **(2008)**

[23] Albayrak, A.: “Beş Eksenli Bir Robot Kolunun Simülasyonu ve Kontrolü”, *Yüksek Lisans Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye,**(2007)**

[24] Çetinkaya, Ö “ *Bir Robot Kolun Hareketini Takip eden Dört Dönel Mafsallı Robot Kolu Tasarımı ve Deneysel Araştırılması ”* Yüksek Lisans Tezi, Trakya Üniversitesi F.B.E Ocak, **(2009)**

[25] Yavuz, E.: “*Sesli İfade Tanıma Sistemi Gerçekleştirilmesi*” Yüksek Lisans Tezi, Marmara Üniversitesi F.B.E, İstanbul, **(2010)**

[26] Peter X. Liu, A. D. C. Chan, R. Chen, K. Wang, Y. Zhu.:“*Voice Based Robot control*” Proceedings of the 2005 IEEE International Conference on Information Acquisition June 27 - July 3 ,Hong Kong and Macau, China **(2005)**

[27] Yazar, N, M.: " Üç Eksenli Robot Kol Tasarımı ve Uygulaması" Yüksek lisans Tezi, Yüzüncü Yıl Üni. F.B.E, Van, **(2010)**

[28] Sinha, P.: "*Speech Processing in Embedded Systems*", Springer, New York, **(2010)** 12-15

[29] http://www.ti.com/product/msp430g2231

[30] M, Z, Özdemircan "*Robot Control with Voice Command*" Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Elektrik-Elektronik Fakültesi Bilgisayar Müh, İstanbul, **(2008)**

[31] F, Baltacıoğlu " Uzaktan Sesle Kontrol Edilebilen Robot Tasarımı", Yüksek Lisans Tezi, Sakarya Üniversitesi. F.B.E, Elektronik ve Bilgisayar Eğitimi, Sakarya ,**(2010)**

[32] M, Nishimori. T, Saitoh " *Voice Controlled Intelligent Wheelchair*", *SICE Annual Conference*, Kagawa University, Japan, **(2007)**

[33] K, Ishikawa. Y, Mori "*Development of self-propelled Robot that Recognizes Voice Instruction*", *SICE Annual Conference*, The University Electro-Communications, Japan, **(2008)**

[34] Kelebekler, E; İnal, M "*Otomobil İçindeki Cihazların Sesle Kontrolüne Yönelik Konuşma Tanıma Sisteminin Gerçek Zamanlı Labaratuar Uygulaması*", Politeknik Dergisi, **(2008)**

[35] Corke, P,I ; " Robotics Toolbox for MATLAB " Pullenvale, AUSTRALIA, **(2002)**

EKLER

EK 1. MSP430 MİKRODENETLEYİCİ VE EASYVR ARASINDAKİ İLETİŞİM PROGRAMI

```
#include  <msp430x20x3.h>
#include "stdbool.h"
#include "protocol.h"

#define LED2 0x01
#define LED1 0x40
#define TXD 0x02
#define RXD 0x04
#define Bit_time 104
#define Bit_time_5 52

unsigned char BitCnt;
unsigned int RXByte;
unsigned int TXByte;
bool isReceiving;
bool hasReceived;
unsigned int recog_result;
void Transmit(void);
char wake_up();
char v_sleep();
char v_setup();
unsigned int v_recognize(char cmd, char group);


void led1_blink(){
    if(P1OUT & LED1)
      P1OUT &= ~LED1;
    else
      P1OUT |= LED1;
      __delay_cycles(100000);
    if(P1OUT & LED1)
      P1OUT &= ~LED1;
    else
      P1OUT |= LED1;
```

```
    __delay_cycles(100000);
}
void led2_blink(){
    if(P1OUT & LED2)
      P1OUT &= ~LED2;
    else
      P1OUT |= LED2;
      __delay_cycles(100000);
    if(P1OUT & LED2)
      P1OUT &= ~LED2;
    else
      P1OUT |= LED2;
    __delay_cycles(100000);
}

void main (void)
{         WDTCTL = WDTPW + WDTHOLD;
          BCSCTL1 = CALBC1_1MHZ;
          DCOCTL = CALDCO_1MHZ;
          P1SEL |= TXD;
          P1IES |= RXD;
          P1IFG &= ~RXD;
          P1IE |= RXD;
          P1DIR |= TXD+LED1+LED2;
          isReceiving = false;
          hasReceived = false;
          __bis_SR_register(GIE);
       led1_blink();
       led1_blink();
       if(wake_up() == 0) {
       P1OUT |= LED1;
       __delay_cycles(10000);
       if(v_setup() == 0) {
       led1_blink();
       led1_blink();
       }}

      while(1)
      {
       if(wake_up() == 0)
       {
         P1OUT |= LED1;
       }
       do{
          __delay_cycles(100000);


       recog_result = v_recognize(CMD_RECOG_SD,2);

       if(recog_result == 0) // eksen_1
```

```
{
   led2_blink();
  TXByte = 0x01; Transmit();
}

if(recog_result == 1)// eksen_2
{
   led2_blink();
 TXByte = 0x02; Transmit();
}

if(recog_result == 2)//eksen_3
{
   led2_blink();
 TXByte = 0x03; Transmit();
}

if(recog_result == 3) // eksen_4
{
   led2_blink();
  TXByte = 0x04; Transmit();
}

if(recog_result == 4) // eksen_5
{
   led2_blink();
  TXByte = 0x05; Transmit();
}

if(recog_result == 5) // eksen_6
{
   led2_blink();
  TXByte = 0x06; Transmit();
}

if(recog_result == 6) // sag
{
   led2_blink();
  TXByte = 0x07; Transmit();
}

if(recog_result == 7) // sol
{
   led2_blink();
  TXByte = 0x08; Transmit();
}

if(recog_result == 8) // yukari
{
   led2_blink();
```

```
  TXByte = 0x09; Transmit();
 }

 if(recog_result == 9) // asagi
 {
    led2_blink();
  TXByte = 0x0A; Transmit();
 }

 if(recog_result == 10) // cok
 {
    led2_blink();
  TXByte = 0x0B; Transmit();
 }

 if(recog_result == 11) // az
 {
    led2_blink();
  TXByte = 0x0C; Transmit();
 }

 if(recog_result == 12) // biraz
 {
    led2_blink();
  TXByte = 0x0D; Transmit();
 }

 if(recog_result == 13) // tamam
 {
    led2_blink();
  TXByte = 0x0E; Transmit();
 }

 if(recog_result == 14) // Basla
 {
    led2_blink();
   TXByte = 0x0F; Transmit();
 }

 if(recog_result == 15) // sifirla
 {
    led2_blink();
   TXByte = 0x10; Transmit();
 }

 } while (recog_result != STS_TIMEOUT );
do{
      __delay_cycles(100000);          }while(v_sleep() != 0);
P1OUT &= ~LED1;
```

```
    while((hasReceived != 1) &&  (RXByte != STS_AWAKEN));
    P1OUT |= LED1;

      }}


//*********************************************************************
    char wake_up(){
    char i;
    for( i=0; i<9; i++){
        TXByte = CMD_BREAK; Transmit();
        while(!hasReceived);
        hasReceived=false;
        if(RXByte == STS_SUCCESS) return 0;
        __delay_cycles(1000);
    }
    return 1;
}


//*********************************************************************
char v_sleep(){
    TXByte = CMD_SLEEP; Transmit();
    TXByte = (ARG_ZERO + 1); Transmit();
    while(!hasReceived);
    hasReceived=false;
    if(RXByte == STS_SUCCESS) return 0;
    else return 1;
}

//*********************************************************************
 char v_setup(){
    TXByte = CMD_ID; Transmit();
    while(!hasReceived);
    hasReceived=false;
    if(RXByte != STS_ID) return 1;
    __delay_cycles(10000);
    TXByte = ARG_ACK; Transmit();
    while(!hasReceived);
    hasReceived=false;
    if(RXByte != ARG_ZERO) return 1;
    __delay_cycles(10000);
    TXByte = CMD_LANGUAGE; Transmit();
    TXByte = ARG_ZERO; Transmit();
    while(!hasReceived);
    hasReceived=false;
    if(RXByte != STS_SUCCESS) return 1;
    __delay_cycles(10000);
    TXByte = CMD_TIMEOUT; Transmit();
    TXByte = (ARG_ZERO+5); Transmit();
```

```
    while(!hasReceived);
    hasReceived=false;
    if(RXByte != STS_SUCCESS) return 1;
    return 0;
}


//*********************************************************************
unsigned int v_recognize(char cmd, char group) {
    unsigned int rslt;
    TXByte = cmd; Transmit();
    TXByte = (ARG_ZERO+group); Transmit();
    while(!hasReceived);
    hasReceived=false;
    rslt = RXByte;
    if((rslt == STS_RESULT) || (rslt == STS_SIMILAR)){
          __delay_cycles(10000);
      TXByte = ARG_ACK; Transmit();
    while(!hasReceived);
    hasReceived=false;
    return (RXByte-ARG_ZERO);
    }
    return rslt;
}

void Transmit(void)
{
        while(isReceiving);
        TXByte |= 0x100;
        TXByte = TXByte << 1;
        BitCnt = 0xA;
        CCTL0 = OUT;
        TACTL = TASSEL_2 + MC_2;
        CCR0 = TAR;
        CCR0 += Bit_time;
        CCTL0 = CCIS0 + OUTMOD0 + CCIE;
        while ( CCTL0 & CCIE );
}

#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void)
{
        isReceiving = true;
        P1IE &= ~RXD;
        P1IFG &= ~RXD;
        TACTL = TASSEL_2 + MC_2;
        CCR0 = TAR;
        CCR0 += Bit_time_5;
        CCTL0 = OUTMOD1 + CCIE;
        RXByte = 0;
```

```
        BitCnt = 0x9;
}
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A (void)
{
if(!isReceiving)
{
        CCR0 += Bit_time;
if ( BitCnt == 0)
{
        TACTL = TASSEL_2;
        CCTL0 &= ~ CCIE ;
}
        else
{
        CCTL0 |= OUTMOD2;
if (TXByte & 0x01)
        CCTL0 &= ~ OUTMOD2;
        TXByte = TXByte >> 1;
        BitCnt --;
 }}
        else
{
        CCR0 += Bit_time;
if ( BitCnt == 0)
{
        TACTL = TASSEL_2;
        CCTL0 &= ~ CCIE ;
        isReceiving = false;
        P1IFG &= ~RXD;
        P1IE |= RXD;
if ( (RXByte & 0x201) == 0x200)
{
        RXByte = RXByte >> 1;
        RXByte &= 0xFF;
        hasReceived = true;
}
__bic_SR_register_on_exit(CPUOFF);
}
else
{
        if ( (P1IN & RXD) == RXD)
        RXByte |= 0x400;
        RXByte = RXByte >> 1;
        BitCnt --;
        }}}
```

EK 2. MSP430 VE MOTOR SÜRÜCÜ KARTI ARASINDAKİ İLETİŞİM PROGRAMI

```
#include<msp430g2553.h>
#include "stdbool.h"
void UARTInit(void);
#define led1 BIT0
#define led2 BIT1
#define led3 BIT2
#define led4 BIT3
#define led5 BIT4
#define LED1 BIT0

unsigned int RXByte;
void led1_blink(){
    if(P1OUT & LED1)
      P1OUT &= ~LED1;
    else
      P1OUT |= LED1;
      __delay_cycles(1000000);
    if(P1OUT & LED1)
      P1OUT &= ~LED1;
    else
      P1OUT |= LED1;
    __delay_cycles(1000000);
}

void main (void)
{
        bool eksen1 = false,eksen2 = false,eksen3 = false,eksen4 = false,eksen5 =
false,eksen6 = false,durum_sag = false,durum_sol = false;
        bool eksen1_bsl = false,eksen2_bsl = false,eksen3_bsl = false,eksen4_bsl =
false,eksen5_bsl = false,eksen6_bsl = false;
   bool durum_yukari = false,durum_asagi = false;

        const unsigned char eksen_1[]={0x84,0x00,0x00,0x00};
        const unsigned char eksen_2[]={0x84,0x01,0x00,0x00};
        const unsigned char eksen_3[]={0x84,0x02,0x00,0x00};
        const unsigned char eksen_4[]={0x84,0x03,0x00,0x00};
        const unsigned char eksen_5[]={0x84,0x04,0x00,0x00};
        const unsigned char eksen_6[]={0x84,0x05,0x00,0x00};
   char get_pos1 = 0x2f, get_pos2 = 0x2f,get_pos3 = 0x2f, get_pos4 =
0x2f,get_pos5 = 0x2f, get_pos6 = 0x2f;


          WDTCTL = WDTPW + WDTHOLD;
          BCSCTL1 = CALBC1_8MHZ;
```

```
                DCOCTL = CALDCO_8MHZ;
                UARTInit();
                P1DIR |= 0x41;
                P1OUT = 0x00;
        P2DIR = 0x3f;
        P2OUT = 0x00;
           __bis_SR_register(GIE);
            __enable_interrupt();
    int i,ek1=0,ek1_1
=0,ek2=0,ek2_2=0,ek3=0,ek3_3=0,ek4=0,ek4_4=0,ek5=0,ek5_5=0;

          while(1)
          {

//============================================================
                if (RXByte == 0x01)// eksen 1 komutu geldiyse eger
                {
                        eksen1 = true;
                        eksen1_bsl = true;
                }
                if (eksen1 == true)
                {
                 if (RXByte == 0x07) // sag
                 {
                          ek1 = get_pos1;
                          durum_sag = true;
                 }
                 if (RXByte == 0x08) // sol
                 {
                          ek1 = get_pos1;
                          durum_sol = true;
                 }
                 if (RXByte == 0x0b) // cok
                 {
                          if (durum_sag)
                            ek1_1 = 0x1a;
                          if (durum_sol)
                            ek1_1 = -0x1a;
                          durum_sag = false;
                          durum_sol = false;
                 }

                 if (RXByte == 0x0c)// az
                 {
                          if (durum_sag)
                             ek1_1= 0x0a;
                          if (durum_sol)
                                  ek1_1 = -0x0a;
                          durum_sag = false;
                          durum_sol = false;
```

```
                }
                if (RXByte == 0x0d) // biraz
                {
                        if (durum_sag)
                          ek1_1=0x10;
                        if (durum_sol)
                                ek1_1 = -0x10;
                        durum_sag = false;
                        durum_sol = false;
                }
                if (RXByte == 0x0e)
                        eksen1 = false;
        }

//=============================================================
                if (RXByte == 0x02)// eksen 2ye komut geldiyse eger
                {
                        eksen2_bsl = true;
                        eksen2 = true;
                }
                if (eksen2)
                {
                        if (RXByte == 0x09) // yukari
                        {
                          ek2 = get_pos2;
                          durum_yukari = true;
                        }
                        if (RXByte == 0x0a) // asagi
                        {
                          ek2 = get_pos2;
                          durum_asagi = true;
                        }
                        if (RXByte == 0x0b) // cok
                        {
                                if (durum_asagi)
                                        ek2_2 = 0x1a;
                                if (durum_yukari)
                                        ek2_2 = -0x1a;
                                durum_asagi = false;
                                durum_yukari = false;
                        }
                        if (RXByte == 0x0c)// az
                        {
                                if (durum_asagi)
                                        ek2_2= 0x0a;
                                if (durum_yukari)
                                        ek2_2 = -0x0a;
                                durum_asagi = false;
                                durum_yukari = false;
                        }
```

```
                    if (RXByte == 0x0d)//biraz
                    {
                            if (durum_asagi)
                                    ek2_2=0x10;
                            if (durum_yukari)
                                    ek2_2 = -0x10;
                            durum_asagi = false;
                            durum_yukari = false;
                    }
                    if (RXByte == 0x0e) // tamam komutu geldiyse eger
                      eksen2 = false;
            }

//============================================================
                    if (RXByte == 0x03)// eksen 3 komutu geldiyse eger
                    {
                            eksen3_bsl = true;
                            eksen3 = true;
                    }
                    if (eksen3)
                    {
                            if (RXByte == 0x09) // yukari
                            {
                                    ek3 = get_pos3;
                                    durum_yukari = true;
                            }
                            if (RXByte == 0x0a) // asagi
                            {
                                    durum_asagi = true;
                              ek3= get_pos3;
                            }
                            if (RXByte == 0x0b) // cok
                            {
                                    if (durum_asagi)
                                            ek3_3 = 0x1a;
                                    if (durum_yukari)
                                            ek3_3 = -0x1a;
                                    durum_asagi = false;
                                    durum_yukari = false;
                            }
                            if (RXByte == 0x0c)// az
                            {
                                    if (durum_asagi)
                                            ek3_3= 0x0a;
                                    if (durum_yukari)
                                            ek3_3 = -0x0a;
                                            durum_asagi = false;
                                    durum_yukari = false;
                            }
                            if (RXByte == 0x0d)//biraz
```

```
                    {
                        if (durum_asagi)
                            ek3_3=0x10;
                        if (durum_yukari)
                            ek3_3 = -0x10;
                        durum_asagi = false;
                        durum_yukari = true;
                    }
                    if (RXByte == 0x0e)
                      eksen3 = false;
                }
//=============================================================
                if (RXByte == 0x04)// eksen 4ye komut geldiyse eger
            {
                    eksen4_bsl = true;
                    eksen4 = true;
                }
                if (eksen4)
                {
                    if (RXByte == 0x09) // yukari
                    {
                  ek4 = get_pos4;
                  durum_yukari = true;
                    }
                    if (RXByte == 0x0a) // asagi
                    {
                      ek4 = get_pos4;
                      durum_asagi = true;
                    }
                    if (RXByte == 0x0b) // cok
                    {
                        if (durum_asagi)
                            ek4_4 = 0x1a;
                        if (durum_yukari)
                            ek4_4 = -0x1a;
                            durum_asagi = false;
                        durum_yukari = false;
                    }
                    if (RXByte == 0x0c)// az
                    {
                        if (durum_asagi)
                            ek4_4= 0x0a;
                        if (durum_yukari)
                            ek4_4 = -0x0a;
                        durum_asagi = false;
                        durum_yukari = false;
                    }
                    if (RXByte == 0x0d)//biraz
                    {
                        if (durum_asagi)
```

```
                    ek4_4=0x10;
                if (durum_yukari)
                    ek4_4 = -0x10;
                durum_asagi = false;
                durum_yukari = false;
            }
            if (RXByte == 0x0e)
            eksen4 = false;
        }

//=============================================================
            if (RXByte == 0x05)// eksen 4ye komut geldiyse eger
            {
               eksen5_bsl = true;
                    eksen5 = true;
            }
            if (eksen5)
            {
                if (RXByte == 0x07) // sag
                {
              ek5 = get_pos5;
              durum_sag = true;
                }
                if (RXByte == 0x08) // sol
                {
                  ek5 = get_pos5;
                  durum_sol = true;
                }
                if (RXByte == 0x0b) // cok
                {
                        if (durum_sag)
                    ek5_5 = 0x1a;
                        if (durum_sol)
                          ek5_5 = -0x1a;
                        durum_sag = false;
                        durum_sol = false;
                }
                if (RXByte == 0x0c)// az
                {
                   if (durum_sag)
                            ek5_5= 0x0a;
                   if (durum_sol)
                            ek5_5 = -0x0a;
                  durum_sag = false;
                  durum_sol = false;
                }
                if (RXByte == 0x0d)//biraz
                {
                        if (durum_sag)
                    ek5_5=0x10;
```

```
                        if (durum_sol)
                          ek5_5 = -0x10;
                        durum_sag = false;
                        durum_sol = false;

                    }
                    if (RXByte == 0x0e)
                    eksen5 = false;
                }
//=======================================================
            if (RXByte ==  0x10) // eger sifirla komutu algilandiysa
            {
                    ek1_1 = 0;
                    ek2_2 = 0;
                    ek3_3 = 0;
                    ek4_4 = 0;
                    ek5_5 = 0;


            }

            if (RXByte == 0x0f) // eger basla komutu geldiyse
             {
                    if (eksen1_bsl)// 1.eksen icin komut verildiyse eger
                    {
                      eksen1_bsl = false;
                     for (i = 0; i < 4; i++)
                     {
                        if (i<3)
                        {
                            while (!(IFG2&UCA0TXIFG));
                            UCA0TXBUF = eksen_1[i];
                        }
                        if (i == 3)
                        {
                            if ((ek1 + ek1_1) > 0x4e)
                            {
                                    while (!(IFG2&UCA0TXIFG));
                                     UCA0TXBUF = eksen_1[i] + 0x4d;
                                     get_pos1 = 0x4e;
                                     P2OUT |= led1;
                            }
                            else if ((ek1 + ek1_1) < 0x11)
                            {
                                    while (!(IFG2&UCA0TXIFG));
                                    UCA0TXBUF = eksen_1[i] + 0x11;
                                     get_pos1 = 0x11;
                                    P2OUT |= led1;
                            }
                            else
```

```
                    {
                      while (!(IFG2&UCA0TXIFG));
                      UCA0TXBUF = eksen_1[i] + ek1 + ek1_1;
                      get_pos1 = ek1+ek1_1;
                     P2OUT &= ~led1;
                    }
                }
            }
        }
//=============================================================
            if (eksen2_bsl)// 2.eksen icin komut verildiyse eger
            {
               eksen2_bsl = false;
              for (i = 0; i < 4; i++)
              {
                    if (i<3)
                    {
                          while (!(IFG2&UCA0TXIFG));
                          UCA0TXBUF = eksen_2[i];
                    }
                    if (i == 3)
                    {
                          if ((ek2+ek2_2)>0x4e)
                          {
                                while (!(IFG2&UCA0TXIFG));
                                UCA0TXBUF = eksen_2[i] + 0x4d;
                                get_pos2 = 0x4e;
                               P2OUT |= led2;
                          }
                          else if((ek2+ek2_2)<0x11)
                          {
                                while (!(IFG2&UCA0TXIFG));
                                UCA0TXBUF = eksen_2[i] + 0x11;
                                get_pos2 = 0x11;
                               P2OUT |= led2;
                          }
                          else
                          {
                                while (!(IFG2&UCA0TXIFG));
                      UCA0TXBUF = eksen_2[i] + ek2+ek2_2;
                                get_pos2 = ek2 + ek2_2;
                               P2OUT &= ~led2;
                          }


                    }
              }
            }
//=============================================================
            if (eksen3_bsl)// 3.eksen icin komut verildiyse eger
```

```
{
  eksen3_bsl = false;
 for (i = 0; i < 4; i++)
 {
        if (i<3)
        {
              while (!(IFG2&UCA0TXIFG));
              UCA0TXBUF = eksen_3[i];
        }
        if (i == 3)
        {
             if ((ek3 + ek3_3) > 0x4e)
             {
          while (!(IFG2&UCA0TXIFG));
              UCA0TXBUF = eksen_3[i] + 0x4d;
              get_pos3 = 0x4e;
              P2OUT |= led3;
           }
             else if((ek3 + ek3_3) < 0x11)
             {
                  while (!(IFG2&UCA0TXIFG));
                  UCA0TXBUF = eksen_3[i] + 0x11;
                  get_pos3 = 0x11;
                  P2OUT |=led3;
             }
             else
             {
                  while (!(IFG2&UCA0TXIFG));
            UCA0TXBUF = eksen_3[i] + ek3 + ek3_3;
                   get_pos3 = ek3 + ek3_3;
                   P2OUT &=~led3;
             }
        }
 }
}
//=========================================================
if (eksen4_bsl)// 4.eksen icin komut verildiyse eger
{
  eksen4_bsl = false;
        for (i = 0; i < 4; i++)
        {
             if (i<3)
             {
                   while (!(IFG2&UCA0TXIFG));
                   UCA0TXBUF = eksen_4[i];
             }
             if (i == 3)
             {
                  if((ek4 + ek4_4) > 0x4e)
                  {
```

```
                                        while (!(IFG2&UCA0TXIFG));
                                    UCA0TXBUF = eksen_4[i] + 0x4d;
                                        get_pos4 = 0x4d;
                                        P2OUT |= led4;
                                    }
                                    else if ((ek4 + ek4_4) < 0x11)
                                    {
                                        while (!(IFG2&UCA0TXIFG));
                                    UCA0TXBUF = eksen_4[i] + 0x11;
                                        get_pos4 = 0x11;
                                        P2OUT |= led4;
                                    }
                                    else
                                    {
                                while (!(IFG2&UCA0TXIFG));
                                    UCA0TXBUF = eksen_4[i] + ek4 + ek4_4;
                                        get_pos4 = ek4 + ek4_4;
                                        P2OUT &=~led4;
                                    }
                                }
                            }
                        }
//=============================================================
                        if (eksen5_bsl)// 5.eksen icin komut verildiyse eger
                        {
                            eksen5_bsl = false;
                            for (i = 0; i < 6; i++)
                            {
                        if (i<3)
                            {
                                while (!(IFG2&UCA0TXIFG));
                                UCA0TXBUF = eksen_5[i];
                            }
                            if (i == 3)
                            {
                                if ((ek5 + ek5_5) > 0x4e)
                                {
                                        while (!(IFG2&UCA0TXIFG));
                                        UCA0TXBUF = eksen_5[i] + 0x4d;
                                        get_pos5 = 0x4d;
                                        P2OUT |= led5;
                                }
                                else if ((ek5 + ek5_5) < 0x11)
                                {
                                        while (!(IFG2&UCA0TXIFG));
                                        UCA0TXBUF = eksen_5[i] + 0x11;
                                    get_pos5 = 0x11;
                                    P2OUT |=led5;
                                }
                                else
```

```
                                        {
                                        while (!(IFG2&UCA0TXIFG));
                                            UCA0TXBUF = eksen_5[i] + ek5 + ek5_5;
                                             get_pos5 = ek5 + ek5_5;
                                             P2OUT &= ~led5;
                                        }
                                    }
                                }
                        }
//=============================================================
        }
  }
}

void UARTInit(void)
{
        P1SEL |= (BIT1 + BIT2);
        P1SEL2 |= (BIT1 + BIT2);
        UCA0CTL1 = UCSSEL_2;
        UCA0BR0 = 52;
        UCA0BR1 = 0;

        UCA0MCTL = 0x11;
        UCA0CTL1 &= ~UCSWRST;
        IE2 |= UCA0RXIE;
}
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR_MASTER(void)
{
  while (!(IFG2&UCA0RXIFG));              // USCI_A0 TX buffer ready?
  RXByte = UCA0RXBUF;
  IFG2 &= ~UCA0RXIFG;
}
```

ÖZGEÇMİŞ

Musa AYDIN, 1987 yılında Giresun ‘da doğdu. 2005 yılında Bulancak Teknik Lisesi’nin Bilgisayar bölümünden mezun oldu.

2006 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar-Kontrol Öğretmenliği Bölümü’ne girdi. 2010 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar-Kontrol Öğretmenliği Bölümü lisans eğitimini tamamladı. 2010-2011 Öğretim yılında Marmara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar-Kontrol Eğitimi programında yüksek lisans öğrenimine başladı. 2011 yılından itibaren İstanbul Aydın Üniversitesi’nde Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümünde araştırma görevlisi olarak çalışmaktadır.