



**T.C.  
GEBZE TEKNİK ÜNİVERSİTESİ**

**Bilgisayar Mühendisliği Bölümü**

## **4 Çekişli Mobil Robot**

**Yasin AÇIKGÖZ**

**Danışman  
Prof. Dr. Erkan Zergeroğlu**

**Ocak 2018  
Gebze, KOCAELİ**





**T.C.  
GEBZE TEKNİK ÜNİVERSİTESİ**

**Bilgisayar Mühendisliği Bölümü**

## **4 Çekişli Mobil Robot**

**Yasin AÇIKGÖZ**

**Danışman  
Prof. Dr. Erkan Zergeroğlu**

**Ocak 2018  
Gebze, KOCAELİ**



Bu çalışma 04/01/2018 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Prof. Dr. Erkan Zergeroğlu	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Prof. Dr. İbrahim Soğukpınar	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

# İÇİNDEKİLER

İÇİNDEKİLER .....	VI
ŞEKİL LİSTESİ.....	VII
KISALTMA LİSTESİ .....	VIII
SEMBOL LİSTESİ.....	IX
ÖZET .....	X
1. GİRİŞ .....	1
2. ARAÇ KİNEMATİĞİ .....	2
2.1 KONTROL PRENSİBİ .....	5
2.2 ARAÇ DİNAMİĞİ .....	5
3. METOT.....	9
3.1 KULLANILAN DONANIM BİRİMLERİ.....	9
3.1.1 Raspberry Pi .....	9
3.1.2 Arduino Mega .....	9
3.1.3 L298N Motor Sürücü .....	9
3.1.4 Lipo Pil.....	9
3.1.5 MPU6050 İvme Ölçer .....	9
3.1.6 25GA DC Motor .....	10
3.1.7 Tekerlek .....	10
3.2 ARAÇ ŞAŞİSİNİN OLUŞTURULMASI VE MONTAJI .....	11
3.3 BİRİMLER ARASI HABERLEŞME .....	12
3.4 ENCODERDAN VERİ OKUNMASI .....	13
3.5 ANDROID UYGULAMASI .....	13
3.5.1 Uygulama Arayüzü .....	14
3.5.2 Mobil Robotun Kontrolü.....	14
3.5.3 Grafik Çizimleri .....	15
3.6 TEKERLEKLERE AİT ÇİZGİSEL HIZIN ELDE EDİLMESİ.....	16
4. TARTIŞMA VE SONUÇ .....	18
KAYNAKLAR .....	19
EKLER .....	20

## ŞEKİL LİSTESİ

Şekil 1 Ackermann Hareket Prensibi .....	2
Şekil 2 Diferansiyel Hareket Prensibi .....	3
Şekil 3 Dört Çekişli Mobil Robot Diferansiyel Hareket Şeması .....	4
Şekil 4 Dört Çekişli Diferansiyel Hız Kontrolüne Ait Kinematik Model .....	5
Şekil 5 DC Motor ve Tekerleğin Görünümü .....	10
Şekil 6 Aracın Genel Görünümü .....	11
Şekil 7 Modüller Arası Haberleşme .....	12
Şekil 8 Android Uygulamasıyla Araç Kontrolü .....	14
Şekil 9 Tekerleklerle Ait Çizgisel Hız Grafikleri .....	15
Şekil 10 Charts Ekranına Ait Görünüm .....	15

## KISALTMA LİSTESİ

**GPIO** : General Purpose Input/Output

**DC** : Direct Current

**ICR** : Instantaneous Centers of Rotation (Anlık Dönme Merkezi)

**RPM** : Revolutions Per Minute



## SEMBOL LİSTESİ

$\omega$	: Açısal Hız
$v$	: Çizgisel Hız
$F$	: Kuvvet
$\text{cm}$	: santimetre
$s$	: saniye

## ÖZET

Bu çalışmada Raspberry Pi mini bilgisayar kullanılarak geliştirilen dört çekişli bir mobil robotun sisteme kurulan Xenomai yaması kullanılarak gerçek zamanlı olarak çalıştırılması, kontrol yönteminin geliştirilmesi, kinematik modelinin tanımlanması ve bununla beraber farklı platformlarda çalışmakta olan kontrol birimlerinin birbiriyle senkron şekilde çalışması amaçlanmıştır. Araç Android işletim sistemine sahip bir mobil cihaz üzerinde çalışan uygulama yardımıyla kumanda edilmiştir. Araç hareketi için diferansiyel hareket prensibi kullanılmıştır. Kullanıcı tarafından araç kumanda edilirken aracın yaptığı harekete dair çeşitli donanım birimlerinden anlık olarak elde edilen bilgiler kullanılarak aracın yatay ve düşey çizgisel hızı elde edilmiştir. Bu hız bilgileri ile beraber 4 çekişli mobil robot için diferansiyel hareket prensibine kullanılarak elde edilen formüller yardımıyla araç hareketinin geliştirilen Android uygulaması üzerinde haritalanması amaçlanmıştır.

## 1. GİRİŞ

Son yıllarda mobil robotik alanına olan ilgi giderek artmaktadır. Araştırmacılar mobil robotların hareketine dair kinematik modellerin tanımlanması ve kontrol mekanizmalarının gerçekleştirilmesi konularına yoğunlaşmaktadır. Bu projede de bu çalışmalara benzer bir çalışma yapılmıştır. Temel olarak 4 çekişli bir mobil robot tasarlanmıştır. Mobil robotun hareketine dair anlık olarak bilgi sahibi olabilmek için encoderlı DC motorlar kullanılmıştır. Tasarlanan mobil robotun kontrolü için 3 adet farklı kontrol birimi kullanılmıştır. Bunlar projenin Arduino, Raspberry Pi ve Android tarafını ifade etmektedir.

Projenin Arduino tarafında, araç hareketi için kullanılan DC motorlar üzerinde bulunan encoderlardan pulse değeri okunmuş, bu değerler anlık olarak Raspberry Pi tarafına gönderilmiştir. Motorların hareketini kontrol edebilmek ise için Xenomai sistem yaması kurulu bir Raspberry Pi mini bilgisayar kullanılmıştır. Bu yama sayesinde sistem gerçek zamanlı olarak çalışmaktadır. Raspberry Pi üzerinde çalışan sunucu programı socket programlama teknikleri ile kablosuz ağ bağlantısı üzerinden Android tarafındaki uygulama ile haberleşmektedir.

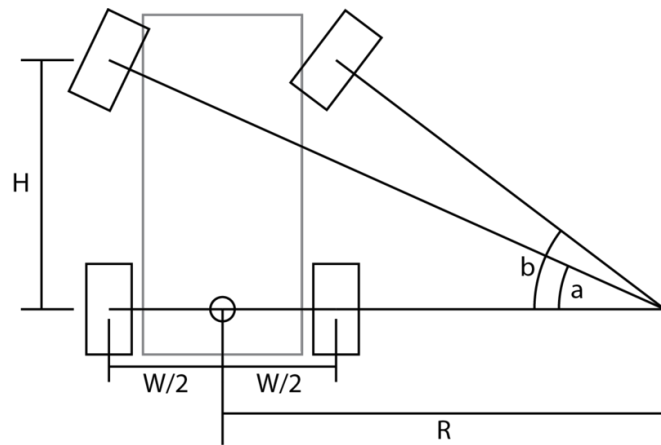
Tasarlan dört çekişli mobil robot diferansiyel hareket prensibiyle kontrol edilmiştir. Bu prensip kullanılarak Android uygulamasında kullanıcı tarafından Raspberry Pi'a gönderilen hareket komutları kullanılarak aracın sağ ve sol tekerlekleri farklı hızlarda döndürülmüş ve araç yönü değiştirilmiştir. Araç hareketi için kullanılan DC motorlara bağlı tekerleklerin çevre uzunluğu kullanılarak aracın  $\Delta t$  zamandaki  $\Delta x$  yer değiştirmesi bulunmuş, buradan her bir tekerleğe ait çizgisel hız bilgisi elde edilmiştir. Dört çekişli mobil robotun kinematik modelinin tanımlanması sonucu elde edilen bazı formüller elde edilmiştir [1]. Sağ ve sol tekerleklere ait hızlar bu formüllerde yerine koyularak tüm araç ile tekerlek hızı arasındaki ilişki bulunmuştur. Bu ilişki kullanılarak aracın yaptığı hareket haritalanmış şekilde Android uygulaması üzerinde anlık olarak gösterilmiştir.

## 2. ARAÇ KİNEMATİĞİ

Mobil robotların kinematik modellerinin tanımlanması için tasarıma başlamadan önce tasarıma dair çeşitli kararlar verilmelidir. Aracın hareketi için kaç adet motor kullanılacağı, dönme hareketinin gerçekleşmesi için nasıl bir yöntem izleneceği bu bilgilere örnek olarak verilebilir. Bu çalışmada birbirinden bağımsız dört adet DC motor kullanılarak dört çekişli bir mobil robot tasarlanmıştır. Aşağıda tasarlanan aracın kontrol mekanizması ve motorlara farklı yönlerde ve şekillerde uygulanan kuvvetin araç hareketinde meydana getirdiği değişiklikler incelenecektir.

### 2.1 KONTROL PRENSİBİ

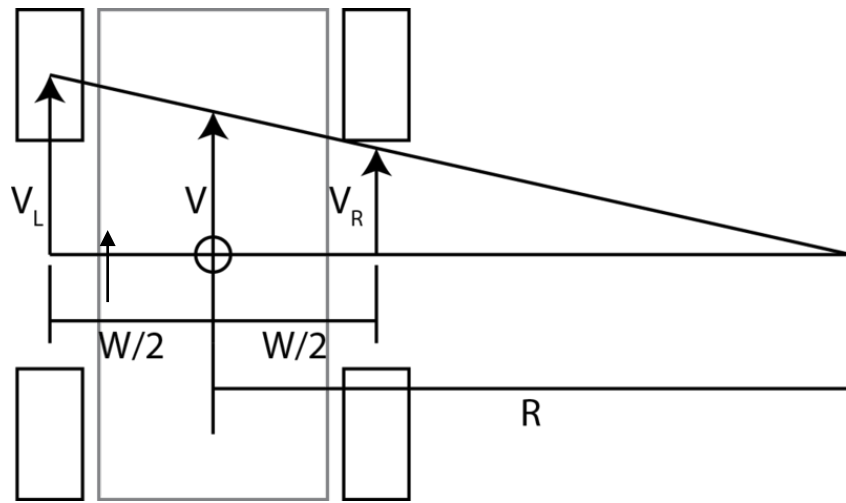
Mobil robotların hareket kontrol yöntemleri için birden fazla prensip vardır. [1] Bunlardan birisi Ackermann prensibidir. Bu prensipte aracın çekiş şeklinden bağımsız olarak ön tekerleklere bağlı olan aks sistemi kullanılır. Ackermann geometrisinden elde edilen formüllere bakıldığında aracın dönüş yarıçapını belirleyen değişkenler tekerleklerin dönüş açıları, aracın aks aralığı ve iz genişliğidir [2]. Bu parametrelerden aks aralığı belirlenirken taşıtın ağırlığı, ağırlık merkezinin yeri ve aksların taşıma kapasitesi gibi parametreler birincil rol oynamaktadır.



Şekil 1 Ackermann Hareket Prensibi

Ackermann prensibinin yanı sıra mobil robotların hareket kontrol yöntemlerinden biri de diferansiyel sürüş yöntemidir. Aracın sağ ve sol tekerlekler arasındaki diferansiyel hızlar ile yönlendirildiği bu yöntem, mobil robotların hareket kontrolünde yaygın olarak kullanılmaktadır. Bu projede tasarlanan 4 çekişli mobil robota da diferansiyel hareket prensibi uygulanmıştır.

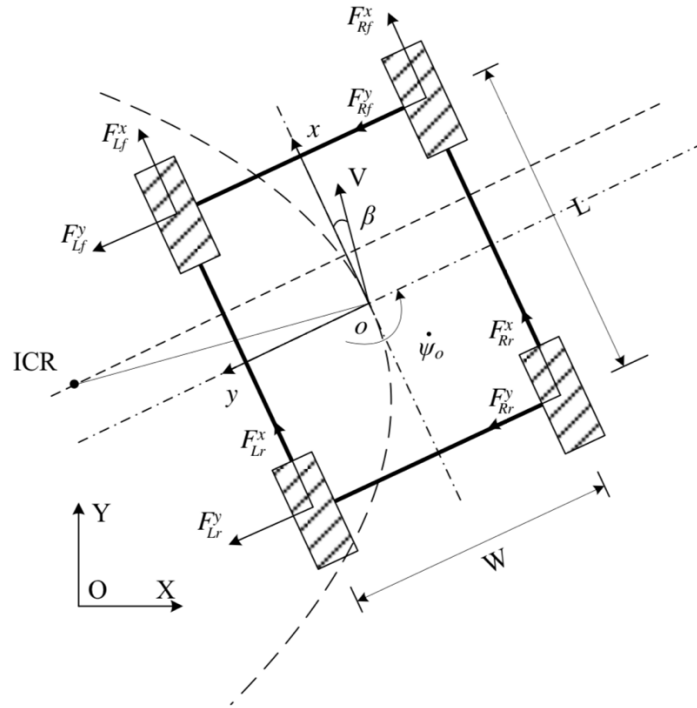
Diferansiyel hareket mekanizması kullanılarak sağ ve sol tekerlekleri farklı hızlarda döndürerek aracın yönü değiştirilebilir. Bu yöntem aracın hareket kontrolünü mekanik olarak sağlam ve basit hale getirir. Diferansiyel hız yönlendirme prensibinde dönüş yönü daha yüksek hızdaki taraf tarafından belirlenir. Tekerlekler arası açısal hız farkını  $\Delta\omega$  olarak varsayalım.  $\Delta\omega = \omega_R - \omega_L$ 'yi olur. Araç  $\Delta\omega > 0$  olduğunda sola dönecek ve  $\Delta\omega < 0$  olduğunda araç sağa dönecektir. İstenen hız farkını elde etmek için, kontrol stratejisi üç yöntemle uygulanabilir. Birinci yöntem dönüş yönüne göre dışarıdaki tekerleği hızlandırmak, ikinci yöntem ise dönüş yönüne göre içerideki tekerleği yavaşlatmaktır. Son seçenek ise aynı anda dönüş yönüne göre içerideki tekerleği yavaşlatıp dışarıdaki tekerleği hızlandırmaktır. Kararlı bir dönüş performansı için aracın sağ ve sol tekerleklerinin hız değişimleri orantılı olmalıdır. Ayrıca sürüş güvenliği açısından bakacak olursak dönüş hareketi sırasında aracın hızının yavaşlaması veya direksiyon hareketi sırasında sabit olması gereklidir. Bu nedenle projede ikinci yöntem kullanılacaktır.



Şekil 2 Diferansiyel Hareket Prensibi

## 2.2 ARAÇ DİNAMİĞİ

Tekerlek hızlarını araç hızlarına haritalayan diferansiyel hareket prensibiyle çalışan bir mobil robotun kinematik modeli, kontrol stratejisinin geliştirilmesi sırasında çok önemlidir. Diferansiyel hız direksiyonunun dinamik modellerini matematiksel olarak analiz etmek için, Şekil 3'te ICR etrafında sabit hızda hareket eden dört çekişli araç şeması gösterilmektedir. [1]



Şekil 3 Dört Çekişli Mobil Robot Diferansiyel Hareket Şeması

Aracın kütle merkezi, gövde çerçevesinin geometrik merkezinde varsayılmaktadır.  $F_x$  ve  $F_y$  uzunlamasına kuvvet ve yanal kuvveti gösterir.  $F$ 'in eklerinde, ilk harf sola veya sağa, ikincisini ön veya arka tekerleği simgelemektedir. Örneğin,  $F_{Lf}$  sol ön tekerleğe,  $F_{Rr}$  ise sağ arka tekerleğe etki eden kuvveti ifade eder.  $W$  aracın genişliği,  $L$  ise uzunluğudur. Şemadaki  $\beta$  işareti ise gövde eksenini ve aracın hareket yönü arasındaki açıyı belirtir. Aynı zamanda lastiğin seyir yönü ve lastiğin yanal kayma açısı olarak da düşünülebilir.

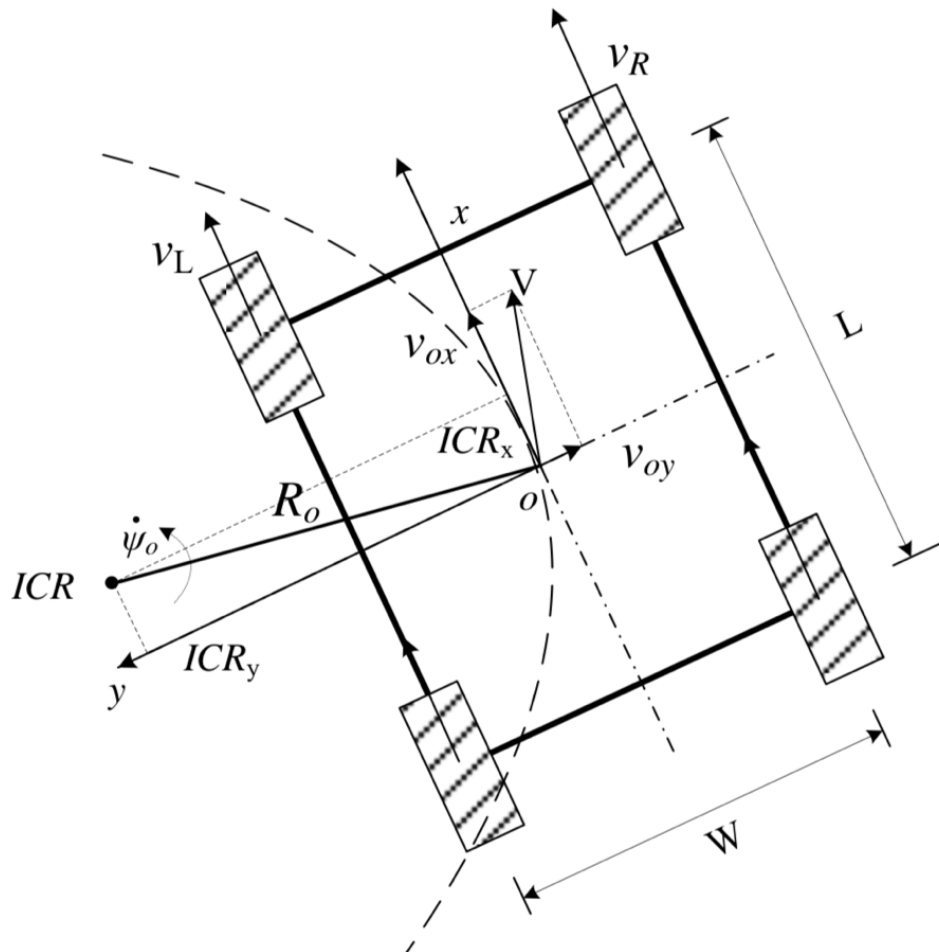
ICR, araç gövdesinin anlık dönüş merkezini Şekil 3'te göstermektedir. Referans çalışmalara dayanarak [3], [4], [5], ICR'nin y eksenine paralel bir çizgiye konulduğuna inanılmaktadır. Dikey konum  $ICR_x$  ve düşey konum  $ICR_y$  aşağıdaki denklemler kullanılarak bulunur:

$$ICR_x = -\frac{v_{oy}}{\psi_o}$$

$$ICR_y = \frac{v_{ox}}{\psi_o}$$

(2.1)

Burada  $v_{ox}$  dikey,  $v_{oy}$  ise yatay yönde aracın kütle merkezinin hızıdır.



**Şekil 4 Dört Çekişli Diferansiyel Hız Kontrolüne Ait Kinematik Model**

Şekil 4'te görüldüğü üzere  $v_R$  ve  $v_L$  sırasıyla ağ ve sol tekerleklerin çizgisel hızları olsun.

$$\begin{aligned} v_{ox} &= \frac{v_L + v_R}{2} \\ \psi_o &= \frac{v_L - v_R}{W} \\ ICR_y &= \frac{W}{2} \frac{v_L + v_R}{v_L - v_R} \end{aligned} \quad (2.2)$$

Dönme açısı (3.2)'deki formüller kullanılarak üçgende açı bulma mantığına benzer şekilde bulunabilir.

$$R_o = \sqrt{ICR_x^2 + ICR_y^2} \quad (2.3)$$

Denklem (3.1) ve (3.2) kullanılarak tüm araç ile tekerlek hızı arasındaki kinematik ilişki şu şekilde özetlenebilir:

$$\begin{pmatrix} v_{ox} \\ v_{oy} \\ \psi_o \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{ICR_x}{W} & -\frac{ICR_x}{W} \\ \frac{1}{W} & \frac{1}{W} \end{pmatrix} * \begin{pmatrix} v_L \\ v_R \end{pmatrix} \quad (2.4)$$

Tekerleklerdeki kayma, diferansiyel hareket prensibi için kritik bir rol oynamaktadır. Araç dönerken iki tekerleğin boylamasına kayması farklı olacaktır. Dönüş hareketi sırasında dış tekerlekteki dönüş hızı, ivme nedeniyle tekerlek çizgisel hızından daha büyüktür. Aynı zamanda iç tekerlek için, tekerlek dönüş hızı, yavaşlama nedeniyle tekerlek çizgisel hızından daha küçüktür. Bu nedenle kayma durumu iç tekerlekte sıfırdan büyük iken dış tekerleğin sıfırdan daha az olacaktır. [1]



Diferansiyel hız yönlendirme sırasındaki dönüş yarıçapı her iki taraftaki tekerlek hızlarıyla kuvvetli bir şekilde birleştirildiğinden, direksiyon performansı aracın anlık hızına duyarlıdır. Araç hareketi sırasında basit bir hız varyasyonu ile aracın ivme kazanması veya yavaşlaması, hızlanmasına etki edecektir. Kararlı dönüş performansını korumak için, sağ ve sol tekerleklerde hız değişimi, orantılı bir ilişkiyi sağlamalıdır. [6] Tekerlek hızının her iki taraftaki değişimini  $\Delta x$  ve  $\Delta y$  olarak tanımlarsak, diferansiyel hız yönlendirme sırasında dönüş yarıçapını korumak için aşağıdaki denklem yerine getirilmelidir:

$$\frac{\omega_R + \Delta x + \omega_L + \Delta y}{\omega_R + \Delta x - \omega_L - \Delta y} = \frac{\omega_R + \omega_L}{\omega_R - \omega_L} \quad (2.5)$$

Yapılan sadeleştirme işleminden sonra, tekerlek hızının  $\Delta x$  ve  $\Delta y$  değişimi denklem (3.6)'daki orantılı ilişkiyi sağlıyorsa, dönme yarıçapının aynı kalabileceği görülmektedir.

$$\frac{\Delta x}{\Delta y} = \frac{\omega_R}{\omega_L} \quad (2.6)$$

### **3. METOT**

#### **3.1 KULLANILAN DONANIM BİRİMLERİ**

##### **3.1.1 Raspberry Pi 3**

Projede geliştirilen sunucunun gerçek zamanlı olarak çalışması için Xenomai yaması kurulu bir Raspberry Pi mini bilgisayar kullanıldı. GPIO pinleri kullanılarak sisteme bağlı olan dört adet DC motor, iki adet L298N motor sürücü kartı yardımıyla sürüldü. Ayrıca yine Raspberry Pi üzerinde çalışan program sayesinde araç kontrolü için gerekli olan Android uygulaması ile motorların hareketi ve grafikler için çift yönlü bir bağlantı kuruldu.

##### **3.1.2 Arduino Mega**

Araç hareketi için kullanılan dört adet DC motor üzerindeki encoderlardan motor hareketi sırasında pulse değeri okumak için kullanılmıştır. Ayrıca aracın matematiksel hesaplamalar sonucu yaptığı hareketlerin gerçekteki hareketleriyle paralel olup olmadığını kontrol edebilmek için bir adet ivme ölçer sisteme dahil edilerek aracın sürüş anındaki hareketi sürekli olarak izlendi.

##### **3.1.3 L298N Motor Sürücü**

DC motorların kontrolü için 2 adet L298N motor sürücü devresi kullanıldı. Bu devrelerden her biri 2 adet motor sürebilmektedir. Motor başı 2A akım verebilirken 24 volta kadar gerilim uygulanabilmektedir.

##### **3.1.4 Lipo Pil**

Motorların çalışması için gerekli olan enerji 11.1 V, 3 hücreli Lipo pil tarafından sağlandı.

##### **3.1.5 MPU6050 İvme Ölçer**

Sisteme dahil edilen ivme ölçer yardımıyla araç kumanda edilirken sağ veya sola dönüş sırasında ortaya çıkan açı ile hesaplanan dönüş açısı arasındaki ilişkiyi ortaya koyarak bir nevi sistemin sağlamanın yapılması sağlanacaktır.



**Şekil 5 DC Motor ve Tekerleğin Görünümü**

### **3.1.6 25GA DC Motor**

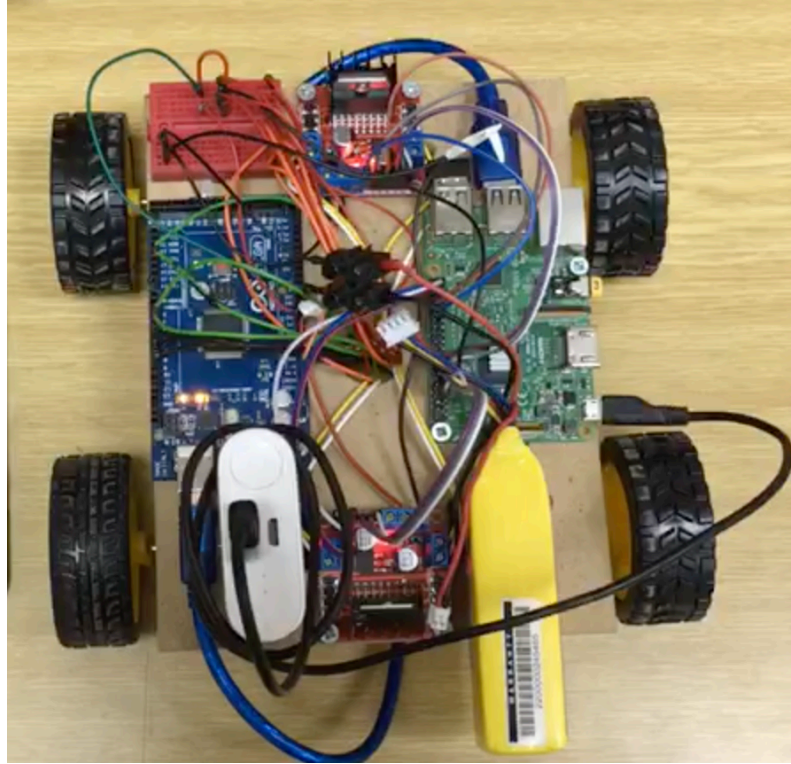
Araç hareketi için 4 adet encoderlı DC motor kullanıldı. Bu motorlar 24 volt gerilimde 500 rpm devirle dönebilmektedir. Mil şaftı 4 mm, mil uzunluğu 10 mm'dir. Ağırlığı 98 gram olan bu motorun toplam uzunluğu ise 77 mm'dir.

### **3.1.7 Tekerlek**

Motorlara monte edilmek üzere 65 mm çapında, 30 mm genişliğinde 4 adet tekerlek kullanıldı.

### 3.2 ARAÇ ŞASİSİNİN OLUŞTURULMASI VE MONTAJI

Araç şasisine motorların ve Raspberry Pi, Arduino, motor sürücü kartı, ivme ölçer gibi donanımsal birimlerin monte edilmesi için 15x20x0,8 cm boyutlarında bir tahta blok kestirildi. Motorlardan gelen kabloların üst kısımda RPi ve Arduino'yla olan bağlantılarının kolaylıkla sağlanabilmesi için tahtanın köşegenlerinin birleştiği noktaya 3 cm çapında bir delik açıldı. Ardından projede kullanılacak olan 4 adet motorun sabitlenmesi için tahtanın 4 köşesine ikişer adet delik açıldı. Açılan bu delikler kullanılarak motorlar cırt kelepçe yardımıyla iyice sıkıldı araç şasisine sabitlendi. Bu aşamada tekerlekler ve motorların tümünün aynı hizada olması için ekstra özen gösterildi.



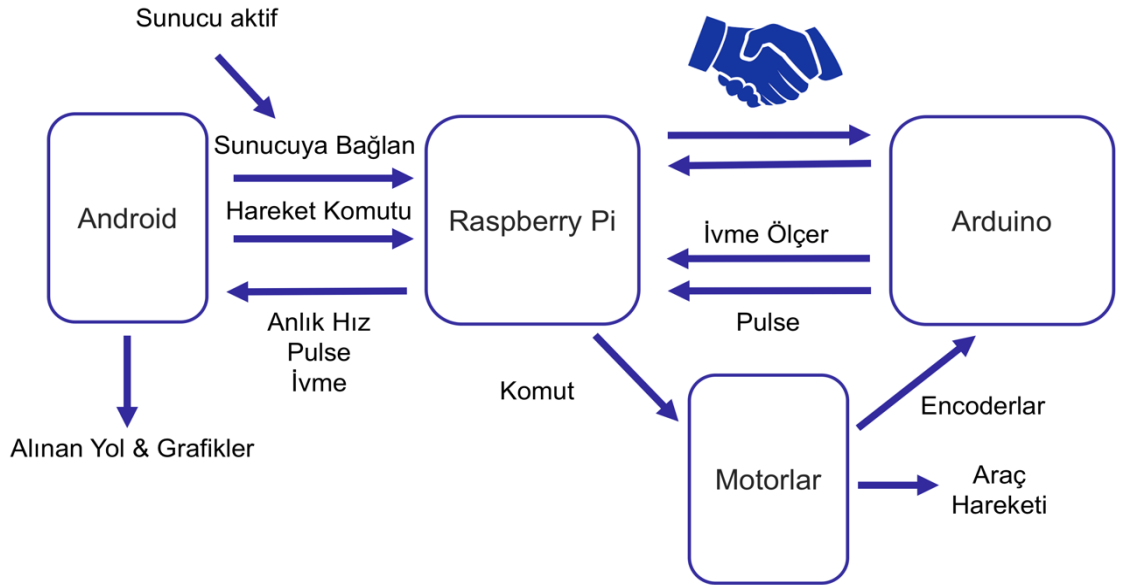
Şekil 6 Aracın Genel Görünümü

Ardından DC motorların kontrolü için kullanılacak olan Raspberry Pi ve motor sürücüler ile encoderlardan veri okumak için kullanılacak olan Arduino Mega, araç şasisinin üst kısmına vidalanmak suretiyle monte edildi. Kabloların bir arada durması için bir adet breadboard da şasinin üst kısmına yapıştırıldı. Montajın ardından Lipo pil de sisteme dahil edilerek tüm kablo bağlantıları yapıldı ve sistemin çalışır hale getirilmesi sağlandı.

### 3.3 BİRİMLER ARASI HABERLEŞME

Projede Android uygulaması ile Raspberry Pi üzerindeki sunucuyu haberleştirmek için kablosuz ağ bağlantısı ve soket programlama teknikleri kullanıldı. Raspberry Pi ile Arduino mikro denetleyici arasındaki haberleşme ise seri port üzerinden sağlandı.

İlk olarak sistemin çalışacağı sunucu aktif hale getirildi. Aynı kablosuz ağ üzerinde bulunan Android cep telefonu tarafından Raspberry Pi'a ait IP ve haberleşmenin sağlanacağı port bilgileri girilerek iki birim arasındaki bağlantı sağlandı ve ardından encoderların kalibre edilmesi için Raspberry Pi ve Arduino arasındaki haberleşmeye başlandı. Kalibrasyon işlemi başarıyla tamamlandıktan sonra Arduino tarafından okunan değerleri Android uygulamasına göndermek ve Android uygulamasından gelen hareket komutlarını işleyerek motorları hareket ettirebilmek amacıyla iki adet thread oluşturularak arka planda sürekli olarak çalışmaları sağlandı. Söz konusu threadler çalışmaya başladıktan sonra kullanıcının Android uygulamasında start butonuna basmasıyla beraber gelen hareket komutu Raspberry Pi üzerinde işlenerek motorların hareketi sağlandı. Motorlar tarafından yapılan bu harekete dair pulse değerleri encoder kullanılarak Arduino tarafından okundu ve ivme ölçerden okunan ivme bilgisiyle beraber serial port aracılığıyla Raspberry Pi'a gönderildi. Bu bilgiler kullanılarak Raspberry Pi tarafından tekerleklerin dönüş hızı hesaplandı ve Android uygulamasına gönderildi.



Şekil 7 Birimler Arası Haberleşme

### **3.4 ENCODERDAN VERİ OKUNMASI**

Encoder, temel olarak bir milin dönme ya da ötelenme hareketine karşılık, dijital bir elektrik sinyali (kare dalga) üreten elektromekanik bir cihazdır. Projede kullanılan DC motorlara bağlı olan encoderlar arttırımsal encoderlardır. Bu enkoderlardan iki adet kare dalga sinyal çıkar. Bunlar A ve B sinyalleri olarak isimlendirilir. Bu sinyaller arasında 90 derece faz farkı vardır. A sinyali Arduino üzerindeki analog pinlere, B sinyali ise dijital pinlere bağlanmıştır. Enkoderlardan gelen verileri okuyan Arduino bu iki kare dalgayı karşılaştırarak motorun hangi tarafa ve ne büyüklükteki bir hızla döndüğünü tespit edebilmektedir.

Sistem aktif olduğunda ilk olarak motorlar ileri ve geri yönde üç saniye döndürülür, bunun sebebi encoderların her çalışmada aynı pulse değerlerini vermemesidir. Bu üç saniyelik çalışmanın ardından encoderların okuduğu maksimum ve minimum pulse değerleri bulunur. Bu değerler kullanılarak araç hareketi sırasında oluşan pulse değerlerinin ölçeklendirmesi yapılır. -100 ile +100 değerleri arasına ölçeklenen pulse değerleri her 200 milisaniyede bir sunucu tarafına gönderilir. Bu işlem araç hareket halindeyken sürekli olarak devam etmektedir.

### **3.5 ANDROID UYGULAMASI**

Android uygulamasında Raspberry Pi üzerinde sunucu ile haberleşmek için socket programlama teknikleri kullanılmıştır. Uygulama Android 5.0 ve daha ileri sürümlerde çalışacak şekilde geliştirilmiştir. Uygulamanın görsel yapısı için ise Fragment mimarisi kullanılmıştır.

#### **3.5.1 Uygulama Ara Yüzü**

Program başladığında ilk olarak Raspberry Pi üzerindeki sunucuya bağlanmak için gerekli olan IP ve port numaraları kullanıcı tarafından girilir, ardından Connect butonu kullanılarak sunucuyla bağlantı gerçekleştirilir.

Bağlantı sağlandıktan sonra kullanıcının karşısına gelen ara yüz içerisinde 3 adet farklı ekran bulunmaktadır. Araç hareket ederken “Home” ekranında tekerleklerinin anlık çizgisel hız bilgileri yazılmaktadır. Ayrıca yine bu ekranda araç hareketini yönetmek için

gerekli olan ileri-geri ve durdur-devam et butonları bulunmaktadır. “Charts” ekranında encoderlardan okunan pulse değerlerinin gösterildiği grafikler gösterilmektedir. Bu grafikler encoderlardan okunan verilerle senkron çalışması için 200 milisaniye sıklıkla verilerini güncellemektedir. Son olarak “Path” ekranında ise aracın gittiği yolun haritalanmış gösterimi bulunmaktadır.

### 3.5.2 Mobil Robotun Kontrolü

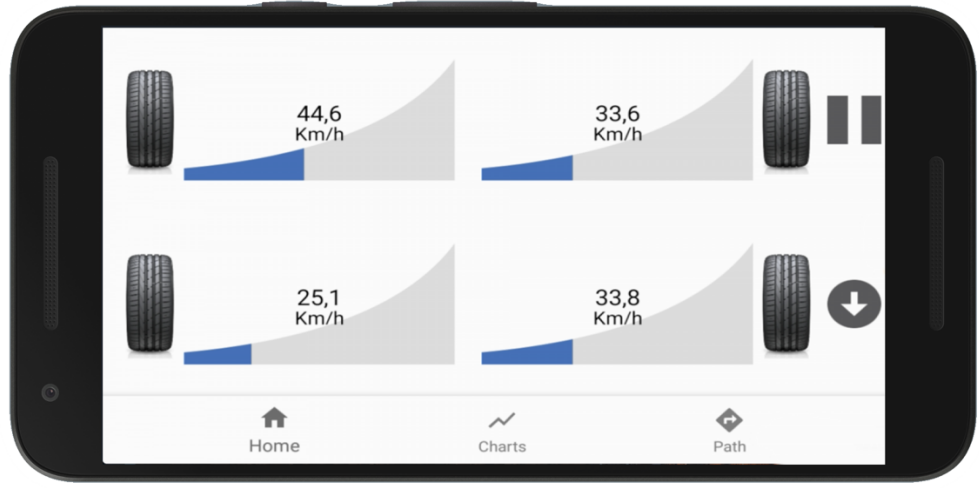
Proje kapsamında geliştirilen olan Android uygulaması ile kullanıcının verdiği komutlara göre aracın istenilen yönde hareket etmesi sağlandı. Android telefon bir direksiyon gibi düşünülerek sağ ve sol yöndeki ivmelenmesi araç hareketi için kullanıldı. Telefonun sağ veya sol yöndeki ivmesinin büyüklüğüne göre aracın dönme hareketini daha keskin şekilde yapması sağlandı. Kararlı dönüş performansı için, sağ ve sol tekerleklerde hız değişimi, orantılı bir şekilde yapıldı.



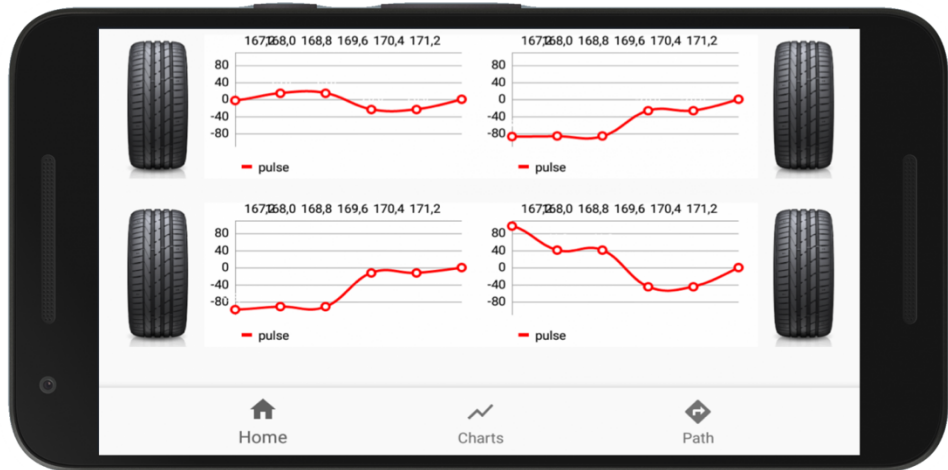
Şekil 8 Android Uygulamasıyla Araç Kontrolü

### 3.5.3 Grafik Çizimleri

Geliştirilen Android uygulamasında, araç kontrolü sırasında DC motorlar üzerindeki encoderlar kullanılarak elde edilen pulse değerlerinin grafik olarak ekranda gösterilmiştir. Bu grafiklerin gösterimi için Android geliştirme ortamı için sunulan açık kaynak kodlu bir grafik kütüphanesi [7] kullanılmıştır. Arduino tarafında -100 ile +100 arasına ölçeklenen pulse değerleri zaman ekseninde sürekli olarak güncellenerek anlık olarak kullanıcıya gösterilebilmektedir.



Şekil 9 Tekerleklere Ait Çizgisel Hız Grafikleri



Şekil 10 Charts Ekranına Ait Görünüm



### 3.6 TEKERLEKLERE AİT ÇİZGİSEL HIZIN ELDE EDİLMESİ

Tekerleklerin çizgisel hızını elde edebilmek için ilk olarak tekerleğin birim zamandaki yer değiştirmesi bulunacaktır. Araç hareketi için kullanılan tekerleğin çevre uzunluğu  $c_{teker}$  olsun.

$$c_{teker} = 2\pi r \quad (3.1)$$

Kullanılan tekerleğin yarıçapı 3,25 cm olduğundan değerler denklem (4.1)'de yerine yazıldığında tekerleğin çevre uzunluğu yaklaşık olarak 20,4 cm olarak bulunur. Bir başka deyişle her tekerlek 1 tam tur ( $360^\circ$ ) döndüğünde yanal ve dikey kayma olmadığı varsayıldığında 20,4 cm yol almaktadır.

$x$  yer değişimi,  $v$  çizgisel hız,  $t$  ise geçen zaman olsun. Bir cismin birim zamanda aldığı yol ve hız değişimi ile ilgili formüller aşağıda verilmiştir.

$$x = v \cdot t$$
$$\Delta v = \frac{\Delta x}{\Delta t} = \frac{x_1 - x_0}{t_1 - t_0} \quad (3.2)$$

Encoderlardan okunan pulse değerleri kullanılarak yapılan testlerde, tekerleklerin 1 tur dönmesi için gerekli olan eşik pulse değerinin yaklaşık olarak 1700 birim olduğu tespit edilmiştir. Arduino üzerinde çalışan program encoderlardan her 200 milisaniyede bir pulse değişimini okumaktadır. Yani bu işlem saniyede 5 kez yapılmaktadır.

1 saniye içerisindeki toplam pulse değişimi  $\Delta p$ , tekerleğin  $360^\circ$  dönmesi için gerekli olan pulse eşiği  $k$  olsun. Denklem (4.1) ve (4.2) kullanılarak tekerleğin çizgisel hızını  $cm/s$  cinsinden elde edebilmek için aşağıdaki denklemler elde edilir.

$$v_{teker} = \frac{\Delta p \cdot c_{teker}}{k} \quad (3.3)$$

$c_{teker}$  ve  $k$  deęerleri yerine yazıldığında denklem (4.3) sadeleşerek aşığıdaki şekle gelir.

$$v_{teker} = \Delta p * 0,012 \quad (3.4)$$

Böylelikle anlık olarak okunan pulse deęerleri kullanılarak her bir tekerleęe ait çizgisel hız  $cm/s$  cinsinden bulunmuş olur.

#### 4. TARTIŞMA VE SONUÇ

Bu projede temel olarak diferansiyel hız kontrol prensibiyle sürülen 4 çekişli bir mobil robotun kinematik incelenmesinin yapılması, yapılan inceleme sonucunda araç hareketine ait matematiksel bir takım verinin elde edilmesine bağlı olarak araç hızı ve konumuna dair çıkarımlar yapılması amaçlanmıştır.

Bu amaç ışığında yapılan çalışma değerlendirildiğinde aracın hareketi sırasında zemine tutunma problemi nedeniyle tekerleklerin kaydığı, bu sebeple de dönüş hareketi sırasında hesaplanan değerler ile aracın gerçek dönüş açısı arasında farklar olduğu görülmüştür. Bununla beraber kullanılan DC motorlara bağlı olan encoderların stabil çalışmaması tekerlek hızlarının bulunması konusunda birtakım sorunlar ortaya çıkarmıştır. Okunan değerlerin anlık olarak çok değişken olması hız tespiti konusunda yanıltıcı sonuçlar verebilmektedir.

Bununla beraber sistemin gerçek zamanlı olarak çalışması için Raspberry Pi üzerine kurulan Xenomai yaması kablosuz ağ bağlantısını desteklemediği için kullanılamadı.

Sonuç olarak, 4 Çekişli Mobil Robot projesinde aracın verilen komutlara göre stabil bir şekilde hareket etmesi sağlanmış, tekerleklerin çizgisel hızları belirli bir yakınsaklıkta hesaplanmıştır. Encoderlardan pulse değerleri başarıyla okunmuş, Android tarafında ise okunan bu pulse değerlerine ait grafiklerinin anlık olarak gösterimi ve tekerleklerle ait çizgisel hız bilgisi gösterilmiştir.

## KAYNAKLAR

- [1] Wu, Xiaodong, Min Xu, and Lei Wang. "Differential speed steering control for four-wheel independent driving electric vehicle." *Industrial Electronics (ISIE), 2013 IEEE International Symposium on*. IEEE, 2013.
- [2] J. Ackermann and W. Sienel, "Robust yaw damping of cars with front and rear wheel steering." *IEEE Transactions on Control Systems Technology*, vol. 1, no. 1, pp. 15-20, 1993.
- [3] J. Y. Wong and C. F. Chiang, "A general theory for skid steering of tracked vehicles on firm ground," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 215, no. 3, pp. 343-355, 2001
- [4] B. Shamah, "Experimental Comparison of Skid Steering Vs. Explicit Steering for a Wheeled Mobile Robot," Master Thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh Pennsylvania, USA, 1999.
- [5] J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya and J. Liu. "Kinematic Modeling and Analysis of Skid-Steered Mobile Robots With Applications to Low-Cost Inertial-Measurement-Unit-Based Motion Estimation," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1087-1097, 2009
- [6] K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 4, pp. 477-496, 2004
- [7] PHIL, JAY, MPAndroidChart Library [Online], <https://github.com/PhilJay/MPAndroidChart>, [Kasim 2017]

## **EKLER**

A. Lisans Bitirme Projesi Konusu Bildirme Formu

<b>T.C. GEBZE TEKNİK ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ LİSANS BİTİRME PROJESİ KONUSU BİLDİRME FORMU</b>	
<b>ÖĞRENCİ ADI SOYADI :</b>	
<b>ÖĞRENCİ NO :</b>	<b>İMZA:</b>
<b>PROJE KONU BAŞLIĞI:</b>	
<b>PROJENİN AMACI :</b>	
<b>FAYDALANILACAK KAYNAKLAR :</b>	
<b>PROJE DANIŞMANI:</b>	<b>İMZA:</b>
<b>BÖLÜM BAŞKANI:</b>	<b>İMZA:</b>

- Bu form bilgisayar ortamında 2 nüsha olarak düzenlenecek, bir nüsha bölüm başkanlığına ve bir nüsha proje danışmanına verilecektir.