



T.C.  
**BURSA ULUDAĞ ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**MAKİNE MÜHENDİSLİĞİ BÖLÜMÜ**



**DÖRT TEKERLEKLİ ARACA OTURTULMUŞ  
ROBOTİK KOLLU SİSTEMİN TASARIMI VE  
ÇALIŞMA PRENSİBİ**

**BURHAN CAN ÖZEN 031920556  
ALPEREN İMAMOĞLU 031820552  
TAYYİP FATİH GÜNEŞ 031920528**

**BİTİRME PROJESİ**

BURSA 2023  
T.C.  
ULUDAĞ ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
MAKİNE MÜHENDİSLİĞİ BÖLÜMÜ

DÖRT TEKERLEKLİ ARACA OTURTULMUŞ  
ROBOTİK KOLLU SİSTEMİN TASARIMI VE  
ÇALIŞMA PRENSİBİ

BURHAN CAN ÖZEN 031920556  
ALPEREN İMAMOĞLU 031820552  
TAYYİP FATİH GÜNEŞ 031920528

Projenin danışmanı: DOÇ. DR. ELİF ERZAN TOPÇU  
Jüri Üyesi:  
Jüri Üyesi:

## ÖZET

Fabrika ortamlarında taşıma ve nakliye alanlarında çevrim süresini azaltmak ve otomasyona geçiş yapabilmek için tasarladığımız bu projeye başlangıç için ilk olarak robotik tasarımlar ve örnek robotlar incelenmiştir. İncelenen örnek robotlar doğrultusunda robotik kolumuzun temasına karar verilmiştir.

Sonrasında sistemin SolidWorks ortamında robotun ve aracın tekerlekleri , uzuvları ve ana kasası tasarlanmıştır. Tasarlanan sistemin robot kolu ve tekerlekleri internet ortamındaki kütüphanelerden çekilmiştir. Sonrasında sistem montajlanarak SolidWorks ortamında hazır hale getirilmiştir. Robot kol ve tekerlekler STL dosyası olarak kaydedilerek UltiMaker Cura programına aktarılmıştır. Bu program yardımıyla 3D yazıcı ile üretilmiştir.

Yapılan tasarım Matlab Simulink ortamına aktarılmıştır. Simulink ortamında Robot kolun analizleri yapılarak gerekli tork değerleri bulunmuştur. Simulink ve SolidWork ortamında çalıştırılan sistemin sonrasında elektronik ve yazılımsal kısımlarına başlanmıştır.

Elektronik kısım için robotun muadili ve piyasada bulunan robotlar incelenmiş, Robotistan firmasından destek alınmıştır. Sistemin elektronik şeması Fritzing programı ile çizilmiş ve bir devre oluşturulmuştur. Bu devre robotik sistemlerde çokça kullanılan ve esnek olan arduino kart sistemleri ile tasarlanmıştır.

Sistemin yazılımsal kısmı ise Robotistanın REX robot kol yazılımıdır. Bu yazılım 4 tekerlekli ve 4 uzuvlu kıskaklı robot kol mekanizmaları için tasarlanmıştır. Yazılım arduinoya yüklenikten sonra sistemin montajına geçilmiştir.

Sistemin montajı için ana kasa olarak MDF sunta seçilmiştir. Seçilen sunta CNC makinesinde işlenerek istenilen forma getirilmiştir. CNC makinesinde işlemek için gövde tasarımı Catia ile tekrar yapılmıştır. Catia CAM modülü yardımıyla CNC makinesinde işlenmiştir. MDF sunta kasasının içine tekerelerler ve step motorlar montajlanarak aracın gövde kısmı oluşturulmuştur. Gövdenin içine arduino, Bluetooth modülü, Batarya ve elektronik donanımlar yerleştirilmiştir. Sonrasında kasadan robot kola geçiş yapacak ana parça takılarak bu parça servo motorlar ile desteklenmiştir. Sonrasında ki uzuv bağlantıları yapılp servo motor ve mikro servo motorlar kullanılmıştır. Sistemin montajlanmasında Freudenberg firmasının bakım bölümünden destek alınmıştır.

## ABSTRACT

We started the project by examining robotic designs and sample robots in order to reduce cycle time and transition to automation in factory environments for transportation and handling purposes. Based on the analyzed sample robots, we decided on the robotic arm for our system.

Next, the wheels, limbs, and main chassis of the robot and vehicle were designed in SolidWorks. The robot arm and wheels were sourced from online libraries. The system was assembled and prepared in SolidWorks. The designs of the robot arm and wheels were saved as STL files and transferred to the UltiMaker Cura program for 3D printing. The parts were then printed using a 3D printer.

The designed system was transferred to the MathWorks Simulink environment. In Simulink, analyses of the robot arm were conducted to determine the required torque values. After running simulations in Simulink and SolidWorks, we proceeded to work on the electronic and software components.

For the electronic part, we examined similar robots and existing robots in the market, and we received support from Robotistan. The electronic schematic of the system was created using the Fritzing program, and a circuit was built. This circuit was designed using Arduino boards, which are commonly used and flexible in robotic systems.

The software part of the system is based on Robotistan's REX robot arm software. This software is specifically designed for 4-wheeled and 4-limbed gripper robot arm mechanisms. After loading the software onto the Arduino, we proceeded with the assembly of the system.

For the assembly of the system, we chose MDF plywood as the main chassis material. The selected plywood was processed using a CNC machine to achieve the desired shape. The body design was recreated in Catia for CNC machining purposes. With the help of the Catia CAM module, the body was machined using the CNC machine. The wheels and step motors were mounted inside the MDF plywood chassis to create the body of the vehicle. Arduino, Bluetooth module, battery, and electronic components were placed inside the body. Then, the main part connecting the chassis to the robot arm was attached and supported by servo motors. The limb connections were made using servo motors and micro servo motors. Freudenberg's maintenance department provided support during the assembly of the system.

## İçindekiler

1. GİRİŞ .....	2
1.1 PROJEMİZİN AMACI.....	3
2. KAYNAK ARAŞTIRMASI.....	4
3. MATERİYAL VE YÖNTEM.....	8
3.1 ROBOTİGE GİRİŞ .....	8
3.2 ROBOT KOL SINIFLANDIRILMASI.....	9
3.3 ROBOTİK KOLUN HAREKET DÜZLEMLERİ .....	10
3.4 MECHANUM TEKERLEKLERİN ÇALIŞMA PRENSİBİ .....	13
3.5 MOTOR TİPLERİ.....	15
3.6 PROJEMİZDE KULLANACAĞIMIZ MOTORLAR:.....	21
3.7 ARDUINO NEDİR: .....	21
4. ARAŞTIRMA SONUÇLARI .....	24
4.1 SOLIDWORKS ORTAMINDA TASARIM VE MONTAJ.....	24
4.2 ROBOT KOL SİSTEMİNİN MATLAB SIMULINK ORTAMINA TAŞINMASI .....	30
4.3 FRITZING ŞEMASI VE SİSTEMİN 2. ELEKTRİKSEL TEMSİLİ ŞEMASI .....	34
4.4 ÜRETİM AŞAMASI.....	34
4.4.1 3D YAZICI AŞAMASI .....	34
4.4.2 ARAÇ GÖVDESİNİN ÜRETİM AŞAMASI .....	36
4.5 MONTAJ AŞAMASI .....	40
4.6 YAZILIM.....	44
4.7 MALİYET ANALİZİ .....	66
5. Kaynakça .....	67

## ŞEKİLLER DİZİNİ

Şekil 1 Seri Robot Kol.....	9
Şekil 2 Paralel Robot Kol.....	9
Şekil 3 Robotik Kol X-Y Düzlemindeki Hareketi.....	10
Şekil 4 Robotik Kolun Z Düzlemindeki Hareketi.....	10
Şekil 5 Robotik Kolun X-Y-Z Düzlemindeki Hareketi.....	11
Şekil 6 Tasarımda Mikro Sevo Motor Gösterimi .....	11
Şekil 7 Tutucu İçin Kullanılan Mikro Sevo Motor .....	12
Şekil 8 Tutucu.....	12
Şekil 9 Araç İskeleti.....	13
Şekil 10 Tekerlekler .....	13
Şekil 11 Mehcanum Tekerleklerin Hareketi.....	14
Şekil 12 Servo Motor Örneği.....	15
Şekil 13 Servo Motor Elemanları.....	16
Şekil 14 Servo Motor Elemanları.....	16
Şekil 15 Step Motor Örneği .....	18
Şekil 16 Step Motor Örneği .....	19
Şekil 17 Step Motor Devre Şeması.....	20
Şekil 18 Arduino Mega Kart Şeması .....	23
Şekil 19 Fritzig Programı.....	23
Şekil 20 Mechannum Teker Montajı .....	24
Şekil 21 Mechannum Teker ve Step Motor Montajı .....	24
Şekil 22 Gövde ve Tekerleklerin Montajı.....	25
Şekil 23 Servo Motorların SolidWorks Ortamındaki Görüntüsü .....	25
Şekil 24 Servo Motorlar ile Robot Gövdesi Montajı.....	26
Şekil 25 1. Uzvun Eklenmesi .....	26
Şekil 26 Servo Motor ve 2. Uzuv .....	27

Şekil 27 3. Uzuv ve Mikro Servo Motor .....	27
Şekil 28 Gripper ve Mikro Servo Motor.....	28
Şekil 29 Dişliler İçin Gereken Mikro Servo Motor .....	28
Şekil 30 Tasarımın Son Hali.....	29
Şekil 31 Simulink Ortamında Robot Kol.....	30
Şekil 32 Simulink Ana Şeması.....	31
Şekil 33 Simülasyon Ortamında Robot Kol .....	31
Şekil 34 Mafsal 1 İçin 30° Hareket Sonucu .....	32
Şekil 35 Mafsal 2 İçin 30° Hareket Sonucu .....	32
Şekil 36 Mafsal 3 İçin 30° Hareket Sonucu .....	33
Şekil 37 Mafsal 4 İçin 30° Hareket Sonucu .....	33
Şekil 38 Fritzing Şeması Ve Sistemin 2. Elektriksel Temsili Şeması .....	34
Şekil 39 Ultimaker Cura Programında İçe Aktarma.....	34
Şekil 40 Ultimaker Cura Programında Parametre Girdisi Örneği.....	35
Şekil 41 3D Yazıcı Baskı Anı .....	35
Şekil 42 3D Yazıcı Baskı Anı .....	36
Şekil 43 CATIA Tasarım Aşaması .....	36
Şekil 44 CATIA Ortamında Çapın Eklenmesi .....	36
Şekil 45 CATIA Ortamında Cıvata Deliği Eklenmesi .....	37
Şekil 46 CATIA Ortamında Tasarımın Son Hali .....	37
Şekil 47 Programların Ürün Ağacındaki Görüntüsü.....	38
Şekil 48 CAM Programlarının Önizlemesi .....	38
Şekil 49 CNC Makinesinde Sıfırlama.....	38
Şekil 50 CNC Makinesinde İşlenme Anı .....	39
Şekil 51 Yan Gövde Son Hali .....	39
Şekil 52 Tekerlek Montaj Görseli .....	40
Şekil 53 Step Motor Bağlanması .....	40
Şekil 54 Tekerlek ve Step motor birleştirilmesi .....	41

Şekil 55 PCB'nin Lehimlenmesi .....	41
Şekil 56 PCB ve Arduino Bağlantılarının Yapılması.....	42
Şekil 57 Uzuvların Eklenmesi .....	42
Şekil 58 Tutucu Montajı.....	43
Şekil 59 Robotun Montajlanmış Hali .....	43

## 1. GİRİŞ

Seri üretimin hayatımıza girişiyle birlikte üretim proseslerinde çevrim süreleri fabrikalarda ve üretim tesislerinde çok önemli bir önem arz etmektedir. Ürünlerin depolanması, istiflenmesi ve bir parçanın bir konumdan başka bir konuma taşınması fabrikalarda genel anlamda insan gücü (forklift, transpalet) ile sağlanmaktadır. Üretimin çok yoğun olduğu dönemlerde insan gücü ile yapılan taşımalarda hata payları ve çevrim süreleri otomatik sistemlere göre daha fazladır.

Fabrikalarda robotların kullanımı, çeşitli taşıma işlerinde oldukça yaygındır. Özellikle, otomatik üretim hatlarında parça taşıma, malzeme depolama ve dağıtım gibi işler için robotlar kullanılır. Robotların bu işleri yapması, çeşitli avantajlar sağlar. Ana avantajlardan biri, robotların insan gücüne göre daha verimli ve hassasiyetli çalışabilmeleridir. Bu sayede, robotlar hacimce küçük parçaları veya hassas malzemeleri de kolaylıkla taşıyabilirler. Diğer bir avantaj ise, robotların yorulmaması ve yıpranmamasıdır. Robotlar, sürekli aynı işleri yaparken bile yıpranmazlar ve yorulmazlar. Bu sayede, taşıma işlemleri sürekli olarak gerçekleştirilebilir ve üretim süreci de sürekli olarak devam edebilir. Ancak, robotların kullanımının bir dezavantajı da vardır. Bu dezavantaj, robotların kurulum ve bakım maliyetlerinin yüksek olmasıdır. Ayrıca, robotların programlanması ve yapılandırılması da zor olabilir. Bu nedenlerden dolayı, fabrikalarda robotların kullanımı kararı verilirken, maliyetler ve işlem zorlukları da göz önünde bulundurulmalıdır.

Günümüzde gelişmiş fabrika ortamlarında çoğu sistem endüstrinin gelişimiyle birlikte insan gücünden ziyade robotik sistemler ile yapılmaya çalışılmaktadır. Bu çalışmalar yoğun bir ar-ge ve araştırma gerektirmektedir. Yapılan birçok araştırmaya göre robotlaşan sistemlerde hata paylarının ve çevrim sürelerinin herhangi bir arıza durumu oluşmadığı zamanlarda operatörlerin yükünü yaklaşık %60 oranında azalttığı ve yaralanmaların önemli ölçüde düşürdüğü gözlenmiştir.

Bu proje kapsamında yapacağımız çalışma esasen prototip olup direkt üretim alanında kullanma amacı bulunmamaktadır. Projenin önemli değeri ise üretim alanındaki senaryonun uygulanabilmesidir. Projedeki uzuv ve ekipmanlar geliştirilebilirse üretim alanında direkt

faaliyet gösterebilir. Projenin asıl amacı otomatik ve uzaktan belirli bir operatörün kontrolü ile taşınması gereken veya depolanması gereken ürünün hatasız bir şekilde konumlandırılmak istenen noktaya iletilemesidir. Projede yapılacak olan sistem step motorlar ile x ve z düzleminde hareket eden 4 tekerlekli bir araca oturtulmuş 6 serbestlik derecesine sahip hareketi servo motorlar ile sağlanan robotik bir koldan oluşmaktadır. Sistem arduino yardımıyla çalıştırılacaktır ve yazılımı açık kaynak ara yüzlerden geliştirilecektir. Sistemin girdileri ise uzaktan kontrol ile yapılacaktır. Bu uzaktan kontrol uygulaması MIT App Inventor Online sistemiyle kurulacaktır. Sistemin tutamaç kısmı mikro servo motorlar ile desteklenecektir ve bir kıskaç hareketi ile yapılacaktır.

## 1.1 PROJEMİZİN AMACI

Projemizin amacı,

- Üretimde yapılacak taşımalar, depolama ve istifleme aşamasındaki insan gücünü azaltmak ve robotlaştmaktır. Bu robotlaştırma ile firmalar uzun vadede yüksek bir kar elde edilebilir.
- Üretim proseslerinde büyük firmaların çevrim sürelerinde yapmak istediği geliştirmeler için projemiz büyük bir katkı verecektir. İnsan gücü ile yapılan taşımalarda oluşabilecek problemlere karşı robot sistemler net ve belirli çevrim sürelerinde çalışmaktadır.
- Üretim alanındaki diğer robotlara kıyasla aracımız dar ve sıkışık alanlarda dahi yüksek hareket esnekliği ile dar alanlarda taşımaya yardımcı olmaktadır.

## **2. KAYNAK ARAŞTIRMASI**

Robotlar hemen hemen birçok alanda kullanılmaya başlamasıyla birlikte Nükleer atık temizliğinden, arama kurtarma çalışmalarına, su altı araştırmalarından, uzay araştırmalarına, hasta refakatinden, sağlık sektörüne kadar birçok alanda kullanılmaya başlanılmıştır. Hareketli robotlar kullanım yerlerine göre günümüzde çok sayıda üretilmiştir. Mobil robotlar insanlar için rutin, tehlikeli ve sağlık açısından riskli ortamlarda en etkili şekilde kullanılabilmektedirler. Bu nedenlerle mobil robotlar insan hayatını daha da kolaylaştırıcı rol oynamaktadırlar. Hareketli robotlar görev aldıkları yerlerde ortam bilgilerini toplayıp, cihazı kullanıcısı veya tasarlanan program ile bu veriler analiz edilerek istenilen işlem gerçekleştirilebilmektedir. Daha önce ulaşılması güç veya ulaşılamamış yerlerin keşfi bu robotlarla gerçekleştirilebilmektedir.

Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü bünyesinde Kaan KUZER tarafından yapılan “AC SERVO MOTORLAR VE SÜRÜCÜ DEVRELERİ”larındaki araştırmada çalışmada endüstride oldukça yaygın bir şekilde kullanılmaya başlanan AC servo motorlar ve sürücü devreleri teorik olarak incelenmiştir. Servo motor mantığının sistemler için uyarlanması, bir servo kontrolde kullanılan terimler, geri beslemeli kontrol sistemleri, sistemlerin sınıflandırılması, bir servo kontrol sistemin başlıca elemanları, uygulamaya yönelik servo örnekleri ve servo motorların sınıflandırılması hakkında çalışmalar bulunmaktadır. (Kaan, 2006)

T.C. FIRAT ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ kapsamında OZAN ÇAKMAK tarafından yapılan “ADIM MOTORLARININ MİKROADIM SÜRME TEKNİĞİ İLE KONTROLÜ” çalışmasında mikro adımlama tekniği, adım motorunun fazlarındaki akımın ölçülecek tork kontrolü yapılması ile ilgili bilgiler bulunmaktadır. (Çakmak, 2016)

NASA Uzay araştırma çalışmalarında kullanımı için tasarlanmış olan Rocky7, Shrimpan and Octopus isimli hareketli, hafif ve küçük robot platformları tekerlekli-bacaklı konfigürasyonunu kullanmaktadır. 8 tekerlekli Rocky7, Mars araştırmaları için tasarlanan Sojourner hareketli robot platformunun bir üst sürümüdür. 61x49x31 cm ölçülerinde ve 11,5 kg ağırlığında olan bu robot 6 serbestlik derecesine sahiptir. Shrimp, 6 DC motor ve 6 tekerlek kullanmaktadır. 60x35x23 cm ölçülerinde ve 3,1 kg ağırlığındadır. Bu robotun önünde ve arkasında manevra yapabilen ikişer tekerlek ve orta kısmında birer tekerlege bulunmaktadır. (Ark., 1991)

İsviçre Federal Teknoloji Enstitüsünde gerçekleştirilen bir çalışmada labirent içinde hareket eden mobil mini bir robot tasarlanmıştır. Robot tasarımında PIC16CS4 mikroişlemci kullanılarak robot kontrolü gerçekleştirilmiştir. Robot genel geri beslemede

infrared tabanlı olarak uzaktan kontrol sağlanmıştır. Kullanıcı operatör tarafından labirent içindeki bazı noktalarda istenirse kontrol ele alabilmektedir. Robot hareketleri yerel geri besleme tekniği ve buna bağlı sensörlerle otomatik olarak yapmaktadır. (Buchi, 1996)

Bu çalışmada yol üzerine çizilen şeritleri kamera modülü ile izleyerek yönünü bulan bir robot geliştirmiştir. Tasarlanan algoritma ile yol üzerinde sapmaya neden olan nesneler olması durumunda alternatif şerit oluşturarak güzergâhtan çıkmayı önlemektedir. Sistem çalışması sınırlı düzeyde olup, yapay sinir ağları ile sonraki adımları tahmin etme üzerine tasarlanmıştır. (Baluja, 1997)

Bu çalışmada bilgisayar destekli kontrol yöntemi ile üç boyutlu uzayda hareket edebilen üç serbestlik dereceli robot kol tasarıımı, kontrolü ve simülasyonu yapılmıştır. (Albayrak, 1997)

Bu çalışmada görüntü ile uzaktan kontrol edilen bir robotun kontrolünü gerçekleştirmiştir. Bir veya birkaç kameradan alınan ortamın görüntüleri işlenerek robotun kontrolü kamera kalibrasyonu yapılmadan gerçekleştirilmiştir. Sistem birkaç aşamada ele alınmıştır. Bunlar; görsel bilgilerin işlenmesi, robotun ve kapalı devre kontrol sisteminin tasarıımı, kamera yeri bilinmediğinde doğruluk analizinin yapılmasıdır. (Victor, 1998)

Bu çalışmada otonom taşıtların yön bulma davranışları üzerine çalışma gerçekleştirilmiştir. Özellikle bu çalışmada net belirgin yollar, hareketli hedefler ve çevre özelliklerini üzerinde yoğunlaşmışlardır. Bu bağlamda birçok pozisyon belirleme teknikleri kullanılmışlardır. Tasarlanan robota Romeo adını verilmiş olup, robot taşıt üzerinde montajı yapılmıştır. (Ollero, 1999)

“Araştırma Amaçlı Modüler Bir Hareketli Robot Platformu Tasarımı”, isimli çalışmada hareketli robot uygulamalarında araştırmacıların bekleyicilerine çözüm üretmek amacıyla, farklı uygulama alanlarında kullanılabilecek hareket edebilen bir robot platformu tasarlanmıştır. Bu makalede günümüzde aktif kullanılan mobil robotlar hakkında kapsamlı bir inceleme sunularak, daha sonra tasarımı tasarlanan robot hakkında bilgi verilmiştir. (Bayar, 2006)

“Genel Amaçlı Web Tabanlı Mobil Robot” (SUNAR) isimli çalışmada, mühendislik eğitim derslerinde ve bilimsel çalışma alanlarında kullanılmak üzere tasarımı gerçekleştirilen mobil robot, web tabanlı olarak tasarlanmıştır. Mobil robot platformu internet üzerinden kontrol edilebilen, yeniden programlanabilen ve kablosuz olarak haberleşebilen bu robot, üzerindeki kamera ile ortam hakkında görüntüleri izleme imkânı

sunmaktadır. Bu çalışma gerçek zamanlı uygulamalarda kullanılmak üzere tasarlanmıştır. (Nihat Yılmaz, 2005)

Bu çalışmada, gezgin robot çalışmalarında kullanılmak üzere, Z80 mikroişlemci tabanlı bir robot modülü tasarımı gerçeklenmiştir. Gezgin robot platformu dört tekerlekli olarak tasarlanmış olup, robot hareketi adım motorları ile sağlanmaktadır. Bu robot platformu üzerine konulan bir kamera ile çizgi takibi uygulaması yaptırılmıştır. (A. Çetiner, 2003)

“Bir Mobil Robot Kolun Tasarımı ve Görüntü Destekli Kontrolü” isimli Doktora çalışmasında, hareket edebilen mobil robot platformu üzerine monte edilmiş bir robot koldan oluşan otonom bir görüntü sistemi kullanarak bir hedefi bulup takip edebildiği gibi istenilen nesneye kol yardımı ile müdahale edebilecek bir araç tasarlanmıştır. Bu çalışmanın ilk bölümünde, bir mobil robot kol tüm alt sistemleri ile birlikte tasarlanıp imal edilmiştir. Çalışmanın ikinci bölümünde, görüntü tabanlı kontrol yöntemleri yardımı ile iki farklı özellikle ki mekanik sistem için iki ayrı kontrol yöntemi geliştirilmiştir. Çalışmanın son aşamasında, bu iki sistem hareketi önce ardışık olarak tekrar ettirilerek robot kontrolü için kullanılmış, ikinci aşamada ise sistemin kinematik özellikleri göz önüne alınarak robot kol hareketinin iki alt sisteminin beraber çalıştırılması sağlayan bir yöntem geliştirilmiştir. (Çetin, 2088)

“Mobil Bomba İmha Robotlarının İncelenmesi ve Prototip Robot Tasarımı”, isimli çalışmada, kullanılmakta olan bomba imha robotları işlev ve yapı olarak incelenmiş ve donanımsal tüm özellikler çalışma içerisinde sunulmuştur. Operatörün robot kontrolünü rahat ve hatasız şekilde yapabilmesi için gerekli yöntemler incelenmiş ve bu incelemeler sonucunda robot kumanda ve kamera kontrolleri üzerine iyileştirme önerileri yapılmıştır. (Ünlütürk, 2012)

MYROBOT (Kablosuz Kontrol Edilebilen Mobil Araştırma Robotu), isimli çalışmada, tasarımı gerçekleştirilen robot kol bir bilgisayar yazılımı ile kullanıcı tarafından girilen komut bilgileri uygun şekilde kodlandıktan sonra komut koduna robot tanımlayıcı kimlik bilgisi eklenerek kablosuz (RF) modül aracılığıyla robota gönderilir. Robot üzerinde bulunan RF alıcı modüle gelen bilgiler güvenlik eşleştirilmesi yapıldıktan sonra kodlar ilgili mikroişlemciye ilettilir. Bu işlem bir if zinciri ile komut çözülür ve robot istenilen işlemi gerçekleştirir. (Şahin İ., 2011)

“Seri Porttan Kablosuz Ağ ile Haberleşebilen Kameralı Araç Kontrolü”, isimli yüksek lisans çalışmasında, DC motorlu bir mobil aracın RF iletişim aracılığıyla bir bilgisayar tarafından kontrolü gerçekleştirilmiştir. Proje kapsamında, bilgisayarın seri portu RS232 ile haberleşerek RF vasıtası ile kontrol edilebilmekte ve üzerinde bulunan kamera ile görüntü

aktarabilen bir araç tasarlanmıştır. Ara yüz programından gelen verilileri seri porttan alarak alıcı kart da bulunan antene iletmek için haberleşme kartı tasarlanmıştır. Ardından ise mikrodenetleyici ve ilgili entegreler birleştirilerek alıcı kartı tasarımını gerçekleştirmiştir. Çalışmada esnasında hızı ve esnekliği göz önünde bulundurularak kullanıcı ara yüz programında Delphi 10. 0 yazılımı tercih edilmiştir. Mikrodenetleyici de ise PIC-C programlama dili kullanılmıştır . (Önal, 2011)

Huang ve Ark (2004), “Alma ve Yerleştirme İşlemleri İçin Yeni Bir İki serbestlik Derecesine Sahip Paralel Robotun Kavramsal Tasarımı ve Boyutsal Sentezi” başlıklı çalışmaları vardır. Bu çalışmaları, alma ve yerleştirme işlemleri için iki serbestlik derecesine sahip paralel robotun kavramsal tasarımını ve en uygun şartlarda boyutsal sentezi ile ilgilidir. Tek serbestlik derecesine sahip besleme mekanizması ile birleştirerek, biz düzlemde nesnelerin çok yüksek hızda ve nispeten yavaş ya da adım adım fakat uzun mesafe hareketlerini normal bir şekilde taşımak için hibrit bir robot oluşturulabilir kanısına varmışlardır. (Huang, 2004)

Juang ve Lum (2013) 2 DC motor, Arduino mega, tek eksen jiroskop ve 2 eksen ivme ölçer kullanarak kendini dengede tutan düşük maliyetli bir robot geliştirmiştir. Bu geliştirilen robot arduinoların basit robotikte kullanımı hakkında detaylar içermektedir. (Lum, 2013)

Yukarıda bahsedilen tez ve makale çalışmalarında da görüldüğü üzere değişik amaçlara hizmet edebilecek birçok hareketli robot platformu tasarlanıp üretilmiştir. Yapılan çalışmalarda da görüldüğü üzere hareketli robotlar tekerlekli, paletli ya da bacaklı olarak tasarlanmıştır. Hareketli robot platformlarının yukarıda bahsedilen koşullarda kendi başına ya da bir yardımcı ile birlikte görev alması düşünüldüğünde hareket etmenin yanında ölçülerinin ve ağırlığının da mutlaka belirli sınırlar içerisinde olması, tasarım esnasında mutlaka göz önünde bulundurulması gereken, önemli kriterlerdendir. Yapılan çalışmamızda incelemiş olduğumuz çalışmalar neticesinde mobil robot tasarımlarının karmaşık yapısı en aza indirilerek, bu alanda yapılacak çalışmalara örnek olabilecek mekanum tekerlekli bir araç üzerinde mobil robot kol tasarımını gerçekleştirmiştir.

### **3. MATERİYAL VE YÖNTEM**

Bu bölümde, geliştirdiğimiz araçta bulunan robotiğe giriş ve robot kol sınıflandırılmasına yer verilmiştir. Tutucu kol mekanizmasındaki motor çeşitlerinin seçilme sebepleri açıklanmış olup, bu motorların açıklamaları kendi başlıklarında incelenmiştir. Robotik kolun, düzlemlerdeki hareketlerinin nasıl sağlandığı açıklanmıştır. Dört tekerlekli aracın araç iskeleti, ve en önemli parçası olan mecanum tekerleklerinin çalışma prensibi konuları yer almaktadır.

#### **3.1 ROBOTİĞE GİRİŞ**

Robotik, Makine Mühendisliği, Elektrik ve Elektronik Mühendisliği ve Bilgisayar Mühendisliği disiplinlerinin ortak çalışma alanıdır. Robotun, Amerikan Robot Enstitüsü tarafından yapılan tanımı, "malzemelerin, parçaların ve araçların hareket ettirilebilmesi için tasarlanmış olan çok fonksiyonlu ve programlanabilir manipülatör veya farklı görevleri yerine getirebilmek için değişken programlı hareketleri gerçekleştirebilen özel araç" şeklinde dir. Robot kavramının temelleri eski tarihlere dayansa da ilk olarak "Robot" kavramı 1922 yılında Çek yazar Karel Čapek'in piyesinde bir tiplemesinin ismine Robot demesiyle ortaya atılmıştır. Bu tarihten sonra robot bilimi üzerinde bilimsel çalışmalar yoğunlaşmıştır. (Alp, 2012)

Robot kelimesi ilk olarak 1920 yılında kullanılmış olsa da, robotlara ait ilk kavramlar ve robot benzeri ilk makinelere ait bilgiler M.Ö. 3000 yıllarına kadar uzanmaktadır. Örneğin:

- M.Ö. 800 Homeros İlyada adlı eserinde hareketli üçayaklılardan bahseder.
- M.Ö. 350 Aristo insanların isteklerini anlayıp itaat eden mekanizmalar öngörür.

Bilgisayar destekli tasarım ve buna paralel olarak bilgisayar destekli üretimin gelişmesiyle endüstriyel robot ihtiyacı ve kullanımı artmıştır. Robotları diğer makinelere ayıran en önemli özellik robotların birden fazla programlanabilmeleri ve mekanik sistemlerinin yanında kontrol sistemi içermesidir. Kısacası robot bilgisayar kontrollü endüstriyel manipülatördür. Robot bilimi elektrik, elektronik, makine, bilgisayar ve matematik mühendisliğinin içine girdiği disiplinler arası bir bilim dalıdır. Makine mühendisliği robotların dinamik ve statik yapısıyla, bilgisayar mühendisliği robotun kontrolünde kullanılacak programla ve elektrik mühendisliği de robotta kullanılacak algılayıcı tasarımlıyla ilgilenir. (Özcan, 2009)

Robotlar ve robotik sistemler, özellikle sanayide ve üretim alanında, iş otomasyonu uygulamalarında kullanılmaktadır. Temel olarak robotlar belli bir zeka veya zeka sistemine sahip değildirler. Yeni nesil robotlar ise çevrelerini algılamaya ve hareketlerini planlamaya yönelik yapay zeka yetenekleri ile geliştirilmeye çalışılmaktadır. Tüm robotlar yapmakla yükümlü oldukları işleri, üzerinde yer alan algoritmalar yardımıyla yürütürler. Görev değişikliğinde bu algoritmanın ve dolayısıyla kodun değiştirilmesi gerekmektedir. Robotik cerrahi sistemleri, ilk kullanılmaya başlanmasından günümüze kadar büyük bir gelişim

göstermişlerdir. Bu sistemlerin yaygın kullanımı ile birlikte gelecekte robotların yeteneklerinin arttırılması ve daha fazla uygulama alanı bulması hedeflenmektedir. Robotların cerrahi, bomba imha, kimyasal süreçler gibi tehlikeli ve kritik alanlarda uzaktan erişim ile kontrol edilmesi gerekmektedir. (Yüzgeç, 2016)

### **3.2 ROBOT KOL SINIFLANDIRILMASI**

Robotlar temel olarak seri ve paralel robotlar olmak üzere iki gruba ayrılır:

Seri robotlar bir dizi eklemler ve bu eklemleri birbirine bağlayan uzuvlardan oluşur. Seri robotlar geniş bir çalışma uzayına ve az sayıda mekanik parçaya sahiptir.

Paralel robotlar ise ana çerçeve ile uç işlevcisi arasında birden fazla paralel uzuvlardan bir araya gelmiştir. Paralel robotlar seri robotlara göre çok sağlam bir mekanik yapıya sahiptir. Bu iki robot türü taşıyacakları kütlenin mekanik yapılarının kütlesine oranı bakımından karşılaşıldıklarında bu oran seri robotlarda çok küçük, paralel robotlarda ise çok büyüktür. Kısacası seri robotlar daha küçük kütleli işlerde, paralel robotlar ise büyük kütleli işlerde kullanılır. (Özcan, 2009)



*Şekil 1 Seri Robot Kol*



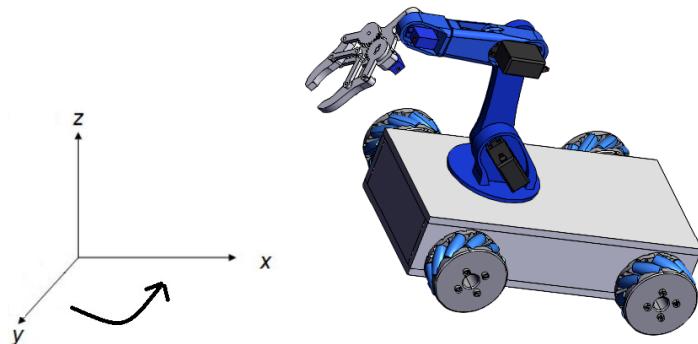
*Şekil 2 Paralel Robot Kol*

Geliştirdiğimiz sistemde seri robot sistemi kullanılmıştır.

### 3.3 ROBOTİK KOLUN HAREKET DÜZLEMLERİ

#### 3.3.1 X-Y DÜZLEMİNDE HAREKET

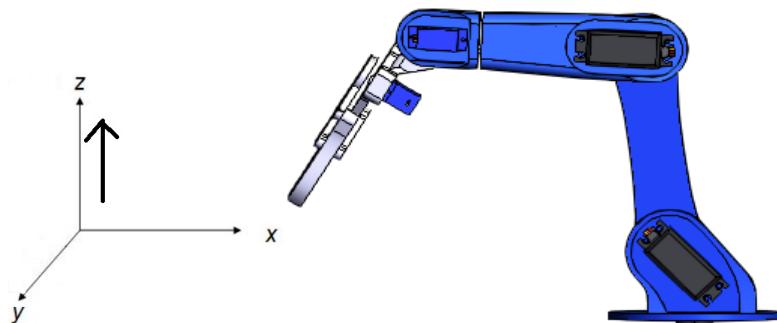
Alt tabanda bulunan, robot kolun oturduğu plakanın içinde bulunan servo motorumuz sayesinde, robot kolumuzun üstten bakış altında saat yönünde veya saat yönünün tersi yönde hareketi sağlanmıştır. Bu hareket yönyle, taşınacak olan yükün yatay doğrultuda hareket edebilmesi amaçlanmıştır.



*Şekil 3 Robotik Kol X-Y Düzlemindeki Hareketi*

#### 3.3.2 Z DÜZLEMİNDE HAREKET

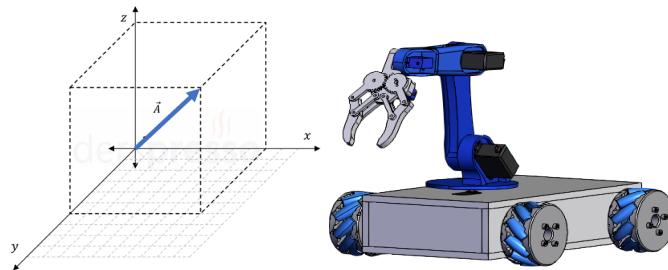
Robotik kolumuzun eklemlerine konumlandırdığımız servo motorlarımızın hareketleri sayesinde, yükü taşıyacak olan uzvumuzun z düzleminde hareketinin sağlanması amaçlanmıştır. Yükün, alt konumdan üst konuma taşınması, bu hareket ile birlikte mümkün kılınmıştır. Düşey doğrultudaki bu hareket, yükün alçak bölgelerden alınıp, yüksek bölgeye taşınmasını sağlamaktadır.



*Şekil 4 Robotik Kolun Z Düzlemindeki Hareketi*

### 3.3.3 X-Y-Z DÜZELMİNDE EŞ ZAMANLI HAREKET

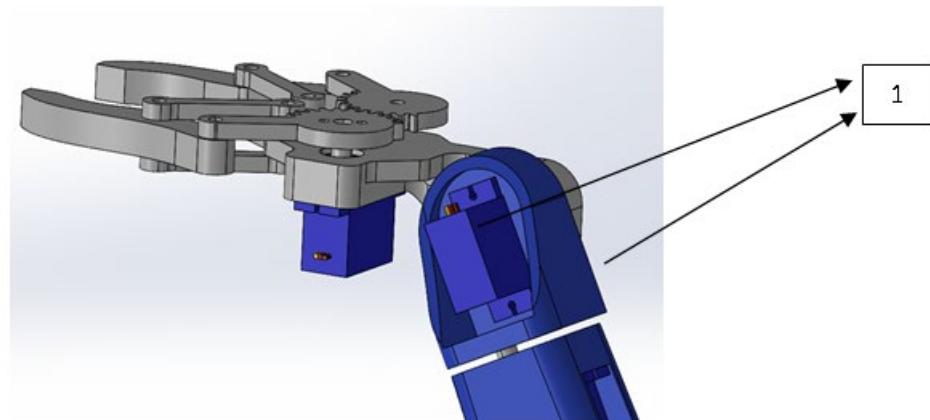
Kolumuzda bulunan motorlarımızın, eş zamanlı hareketleriyle birlikte 3 düzlemdede aynı anda hareket edebilmesi, yapılmış olan tasarımla mümkün kılınmıştır. Üretimdeki çevrim süresini azaltmaya yönelik geliştirilmiştir. 3 düzlemdede eş zamanlı hareketi ile, kol boş iken hareket eden aracımızın, önüne çıkan engellerden kolayca sıyrılabilmesi mümkündür. Aracın yükü almak için durabileceği alan kısıtı ortadan kaldırılmıştır. Bu gibi faktörlerle birlikte, üretimde yaşanılacak zaman kayıpları en aza indirilmeye çalışılmıştır.



Şekil 5 Robotik Kolu X-Y-Z Düzlemindeki Hareketi

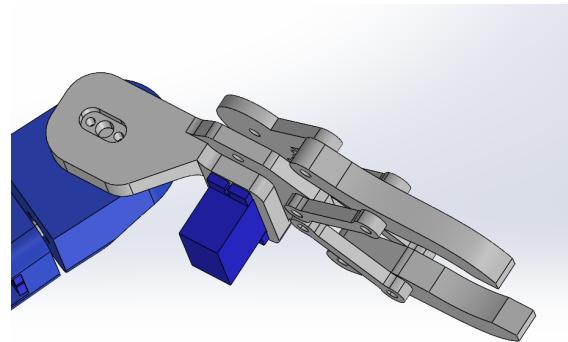
### 3.3.4 TUTUCU HAREKETİ

Tutucu uzvumuzda mikro servo motorlar kullanılmıştır. Şekil Tutucu uzvun hareketi mikro servo motor hareketi vasıtayıyla sağlanmıştır. Sağlanan bu hareketin kalıcı olması ve kilit rolünü üstlenmesi amacıyla robot kolumuzdaki tutucu uzvumuzun gerekli parçaları dişli şeklinde tasarlanmıştır. Yük kaldırıldığında, olabilecek iş kazalarının, bu sayede önlenmesi hedeflenmiştir.



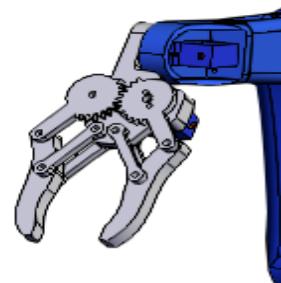
Şekil 6 Tasarımda Mikro Sevo Motor Gösterimi

Şekil 7'de gösterilen, kullanılan motorlardan bir tanesi, tutucu kısmın ekleminde bulunmaktadır. Burada bulunan mikro servo motorun görevi, tutucuyu z düzleminde hareket ettirmektir. Bu ekstra hareket, yükün bırakılacağı yere konumlandırılırken büyük bir kolaylık sağlayacaktır.



*Şekil 7 Tutucu İçin Kullanılan Mikro Sevo Motor*

Diğer motor (Şekil 8) ise, tutucunun kıskaçlarını hareket ettirmektedir. Kıskaçlardan biri sağa açılırken diğeri de sol tarafa açılır. Bu sayede taşınacak yük, kıskaçların arasına alınır. Motor, ilk yaptığı hareketin aksi yönde çalıştırılınca, yük kıskaçlar yardımıyla sıkışip taşınmaya hazır hale gelecektir.



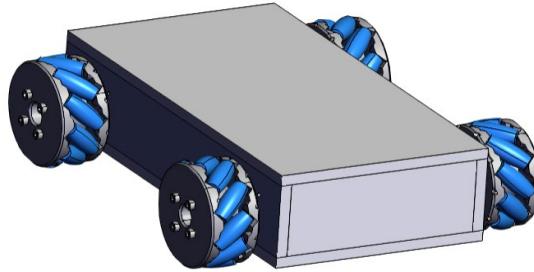
*Şekil 8 Tutucu*

### **3.3.5 HAREKETLİ ARAÇ**

Bu bölümde aracın gövde ve tekerleklikleri ile ilgili bilgilere yer verilmiştir. Tekerlek ve gövde seçim nedenlerinden ve çalışma prensiplerinden bahsedilmiştir.

### 3.3.6 ARAÇ İSKELETİ

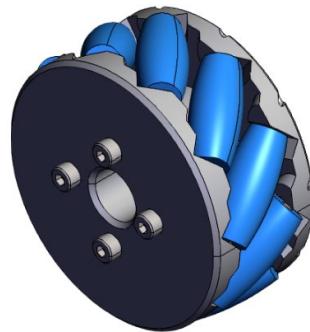
Yaptığımız tasarımın araç bölümünde gövde ile ilgili herhangi bir aerodinamik beklentiye girmediği için basit bir dikdörtgen tasarım tercihi yapılmıştır. Bununla birlikte aracın birçok ekipmanı gövde içinde bulunacağından aracın dış gövdesi tamamen kapalı bir kutu şeklinde olacaktır.



*Şekil 9 Araç İskeleti*

### 3.3.7 TEKERLEKLER

Yapılan tasarımda en önemli özelliklerden biri mecanum tekerleklerdir. Bu tekerleklerin en büyük avantajı, aracın gideceği yöne doğru konum almasına gerek duymadan bulunduğu yerden hareketine başlayabilmesidir.



*Şekil 10 Tekerlekler*

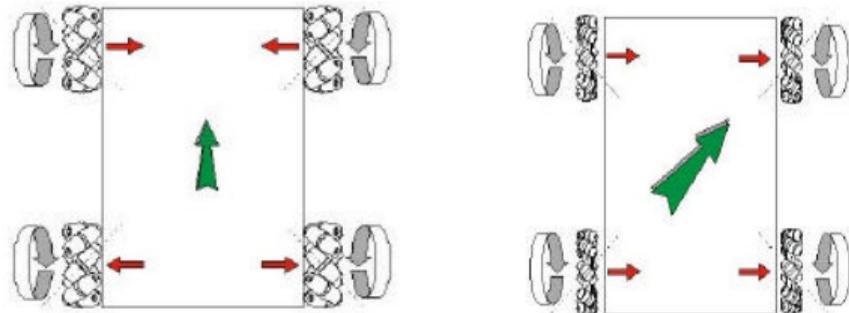
## 3.4 MECHANNUM TEKERLEKLERİN ÇALIŞMA PRENSİBİ

Mechannum tekerlekler, bir tür özel tekerlektir ve döner hareket etme özelliğine sahiptir. Bu sayede, mechannum tekerlekler sayesinde bir robot veya araç çeşitli yönlerde ilerleyebilir. Mechannum tekerlekler genellikle dört tanesi olur ve her biri birbirlerinden açılı

olarak yerleştirilir. Bu sayede, araç veya robot yüksek derecede esneklik sağlar ve dört yönlü hareket etme özelliğine sahip olur. Mechanum tekerleklerin çalışma prensibi, her bir tekerlek üzerinde bulunan küçük dişliler sayesindedir. Bu dişliler, çevrim sırasında döner ve bu hareket sayesinde araç veya robot ileri veya geri hareket ettirilebilir. Ayrıca, mechanum tekerleklerin birbirlerinden açılı olarak yerleştirilmesi sayesinde, araç veya robot yana da hareket ettirilebilir.

Bir Mechanum tekerliği, çevresine silindirler takılı bir tekerlektir. Bu silindirler, tekerliğin dönme eksenine çapraz veya 45 derecelik açı yapacak şekilde konumlandırılmıştır. Bu, tekerliğin ileri geri hareket ederken çapraz yönde kuvvet uygulamasını sağlar.

Böylece tekerlekleri belirli bir düzende döndürerek bu çapraz kuvvetleri kullanız ve böylece robot herhangi bir yönde hareket edebilir.



*Şekil 11 Mechanum Tekerleklerin Hareketi*

Burada ayrıca, genellikle solak ve sağlam mechanum tekerlekleri olarak adlandırılan iki tür Mechanum tekerlegine ihtiyacımız olduğunu da belirtmeliyiz. Aralarındaki fark, silindirlerin yönüdür ve robotta belirli konumlara monte edilmeleri gereklidir. Her tekerliğin üst silindirinin dönme ekseni, robotun merkezine bakmalıdır.

Geleneksel tasarımda doğrudan sürtünmeye neden olan ön sağ ve arka sol silindirler artık yalnızca sürtünmeye neden olmakla kalmıyor, aynı zamanda çapraz hareket hızına katkıda bulunmak için tekerlekler de döndürülebiliyor.

Ustaca bir sistem olmasına rağmen, geleneksel Mechanum tekerlekleri enerji kullanımında biraz verimsizdir. Mechanum tekerleklerinin çevresel silindirlerini kilitleme özelliğinin eklenmesi, ileri yönde hareket ederken bu verimliliği artırır. Hareket yönüne en iyi uyacak şekilde çevresel silindirlerin açısını dinamik olarak ayarlama yeteneğinin eklenmesi, geleneksel tasarımın doğasında var olan verimsizlikleri büyük ölçüde çözer. Sıkışık bir fabrika ortamında hareket edebilir ve karmaşık görevleri gerçekleştirebilir. Bu yetenek, malzeme transferi ve taşıma sırasında paha biçilmezdir.

### 3.5 MOTOR TİPLERİ

Pozisyon belirleme sistemlerinde kullanılan motorlar, sistemin gerektirdiği hareket ve doğruluğa göre seçilir. Örneğin, hafif ve düşük doğruluk gerektiren sistemlerde düşük güçlü ve ucuz motorlar kullanılabilir, ancak yüksek doğruluk ve yüksek hızlı hareket gerektiren sistemlerde daha güçlü ve hassas motorlar tercih edilebilir.

Popüler motor tipleri şunlardır:

**DC motorlar:** Doğrulukları genellikle düşüktür ve hareketlerini kontrol etmek için sensörler ve dijital devreler gereklidir.

**Stepper motorlar:** İleri geri hareketlerini belirli adımlarla yapabilirler ve bu nedenle pozisyon kontrolü için popülerdirler. Ancak, güç tasarrufu ve hız açısından diğer motorlara göre daha düşüktürler.

**Servo motorlar:** Düşük hızlı hareketler için hassas pozisyon kontrolü sağlar, hız ve güç açısından diğer motorlardan daha iyidirler. Ancak, maliyetleri diğer motorlara göre daha yüksektir.

#### 3.5.1 SERVO MOTORLAR

Hataları önlemek için geri bildirimler alan, sürekli kendini denetleyen ve düzenleyen, kapalı devre bir motor sistemidir. Servo motorlar kendi aralarında DC servo motor, AC servo motor olmak üzere iki ana gruba ayrılmaktadır.



Şekil 12 Servo Motor Örneği

Servo motorlar bazen kontrol motorları olarak da adlandırılırlar. Servo motorlar açısal konum, ivmelenme ve hız açısından sisteme hassas bir kontrol sağlar. (Kaan, 2006)

Servo motorların güçleri birkaç watt'tan birkaç yüz watt'a kadar olabilir. Servo motorlar yüksek hız tepkisine sahiptirler. Bu özellik ise servo motorların düşük rotor ataletine sahip olmalarını gerektirir. Bu motorlar daha küçük çaplı ve daha uzundurlar. Servo motorlar normal olarak düşük veya sıfır hızda çalışırlar ve bundan dolayı moment veya güç değerleri aynı olan klasik motorlara göre boyutları daha büyüktür. Servo motorlar, robotlar, radarlar, bilgisayar, takım tezgahları, izleme ve yol gösterme sistemleri ve işlev denetleyiciler olarak kullanılabilir. (Kaan, 2006)

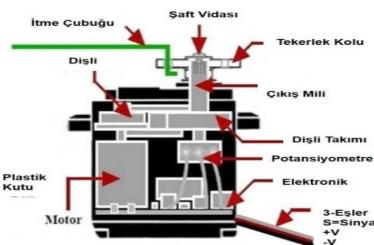
**1.AC Servo Motor:** Hızlı bir motor çeşidi olmasıyla beraber, zorlu çalışma koşullarında da oldukça etkili olarak çalışmaktadır. Hareket mekanizması olarak, alternatif akım elektrik motorları ile aynı özelliklere sahip olmasına rağmen AC Servo motorlar, AC motorlardan farklı olarak encoder kısmına sahiptir. Encoder yardımı ile açısal hassasiyette dönme kabiliyeti kazandırılmış olur. (Aksa, 2021)

AC servo motor yüksek güç gereksinimi ve frekans değeri ihtiyacı olan projelerde tercih edilmektedir. Yüksek enerjili manyetik bir alana sahip olan AC servo motorlar kullanıldığı alanlarda verim elde edebilmesi açısından büyük avantaj sağlamaktadır. Hareket kontrolü gerektiren uygulamalarda hızlı ve seri bir şekilde konumlanmasına olanak tanımaktadır. (Şahin, 2021)



Şekil 13 Servo Motor Elemanları

**2.DC Servo Motor:** Hareket kontrolü gerektiren uygulamalarda hızlı ve seri bir şekilde konumlanmasına olanak tanımaktadır. Yüksek tork değerlerine kısa süre çıkabilmektedir. Buna bağlı olarak da çalışma verimi artar. Düşük hızlarda da yüksek enerji üretebilmektedir. Çalışma esnasında bobinler sabit kalır ve manyetik alan hareket halindedir. Geri beslemeli sistemi sahiptir. PWM ile çalışan DC motorlar analog veya dijital olmaktadır. (Şahin, 2021)



Şekil 14 Servo Motor Elemanları

### **3.5.2 SERVO MOTORUN ÇALIŞMA PRENSİBİ**

Servo motorlar akım ve voltajla bağlı olarak tork ve hız üretirler. Servo motor, geribildirim cihazı kullanımı ile bir servo kontrolcü tarafından yönetilerek tork ve hız sağlayan bir kapalı döngü sisteminin bir parçası olarak çalışır. Geri besleme cihazı akım, hız veya pozisyon gibi bilgileri servo sürücüye sunar. Bu sürücü de komut verilen parametrelere göre motor eylemini gerçekleştirir. Bir servo motor; bir motor, bir geribildirim cihazı ve kontrol elektronikleri olmak üzere üç ana elemandan oluşur. Motor AC veya DC, fırçalı veya fırçasız, döner veya lineer ve herhangi bir boyutta olabilir.

Geribildirim cihazı bir potansiyometredir, Hall etkili cihaz, devir sayacı, çözücü, kodlayıcı, lineer transdüser veya uygun olan herhangi bir sensör olabilir. Servo sistemi tamamlamak, motora güç veren ve servo motorun komut verildiği üzere çalıştığını doğrulamak için geribildirim verileri ile komut referansını karşılaştırılan kontrol elektroniğidir. Örneğin, araç yavaşlarsa sensör bu veriyi elektroniklere ileter ve ayarlanan hızı yükselmek için motora gaz verir, basit bir kapalı döngü sistemidir. (Kollmorgen, 2020)

### **3.5.3 SERVO MOTORUN KULLANIM ALANLARI**

Servo motorların kullanım alanları oldukça genişdir. Aşağıda servo motorların kullanım alanlarına bazı örnekler verilmiştir.

- Hassas Tezgahlar
- Pozisyon Belirleme Sistemleri
- Dijital Kontrol Makineleri
- Pompalar
- Radarlar
- Klimalar
- Robotik kol

### **3.5.4 STEPPER MOTORLAR**

Step motorlar elektrik enerjisini dönme hareketi ile fiziksel enerjiye çeviren elektromekanik aygıtlardır. İsimlerinden de anlaşıldığı üzere adım adım hareket eden motorlardır. Biraz daha detaylı olarak açıklayacak olursak, girişlerine uygulanan puls sinyallerine karşı analog dönme hareketi çıkışı üreten, bu dönme hareketini adım adım ve çok hassas kontolle sağlayan sabit mıknatıs kutuplu motorlardır. Step motorların yapıları rotor, stator ve rulmanlardan oluşmaktadır. Rulmanlar, rotora bağlı şaftın rahat hareket etmesini sağlarlar. Statorun birden fazla kutbu vardır. Kutup sayısı motordan motora değişmektedir ancak genellikle bu sayı sekizdir. Kutupların polaritesi elektronik anahtarlar vasıtıyla sürekli değişir. Rotorun mıknatılılığı ya sabit mıknatıs ile ya da dış uyarı teknikleri ile meydana getirilir. Step motorun en büyük avantajı hassas pozisyon ve hız kontrolü sağlaması ve aynı zamanda düşük devirde yüksek tork üretebilmesidir. Fakat verim ve geribildirim

mekanizması barındırmadığından harici konum limiteye gerek duymaktadır. (Rehberiniz, 2011)



*Şekil 15 Step Motor Örneği*

Stepper motorlar 5 tipte incelenebilir:

**Sabit mıknatıslı step motorlar (PM):** PM (Permanent Magnet Stepper Motors) motorlarının yapılarında rotorda sabit mıknatıslar bulunur. Stator bobinlerine uygulanan gerilime ve akımın yönüne göre motorun dönmesi gerçekleşir. Bu tip motorlardaki rotorlarda bulunan mıknatıslar, manyetik akının artmasını sağlarlar. Stator bobinlerindeki akım arttıkça manyetik alan, buna bağlı olarak tork da artar. (Semiz, 2018)

**Değişken relüktanslı step motorlar (VR):** Değişken relüktanslı step motorlar en temel ve en basit step motor tipidir. Sabit mıknatıslı step motorlarda olduğu gibi bu motorlarda da en az dört kutuplu stator bulunmaktadır. Sabit mıknatıslı değil, mıknatıslanabilen çok kutuplu malzemelerden yapılmış rotorlar bulunur. (Semiz, 2018)

**Hibrit step motorlar (HB):** Hibrit step motorların da rotorlarında sabit mıknatıs bulunur. Bu motorlara hibrit denmesinin sebebi çalışma prensiplerinin sabit mıknatıslı step motorlar ve değişken relüktanslı step motorların birleşiminden meydana gelmesidir. Statorlarının nüve yapıları değişken relüktanslı step motorlara çok benzese de sargı bağlantıları daha farklıdır. Bu motorlar adım kararlılığı, hız ve tork gibi özellikleri kıyaslanarak sabit mıknatıslı step motorlardan daha iyi performans gösterirler. (Semiz, 2018)

**Hidrolik step motorlar:** Bir diğer adları da elektro-hidrolik step motorlardır. Elektro-hidrolik step motor, bir hidrolik motorun sunduğu büyük gücü kontrol etmek için küçük bir elektrikli step motor kullanan cihazdır. Yapısında step motor, hidrolik motor, valf, translatör ve elektronik konektör bulunur. (Semiz, 2018)

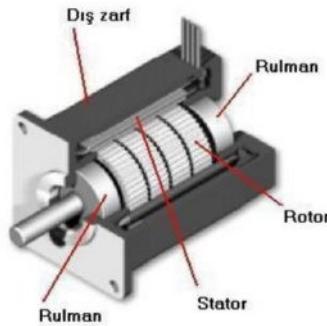
**Lineer step motorlar:** Yüksek devir yerine yüksek tork tercih edilen uygulamalarda, motorun miline bağlanan bir dişli seti sayesinde örnek olarak çıkış hızını 30'da 1'e düşürecek, fakat elde edilen torkun (tork: dönmeye kuvveti) teorik olarak 30 katına çıkması sağlanabilir. (Semiz, 2018)

### 3.5.5 STEPPER MOTORUN ÇALIŞMA PRENSİBİ:

Step motorların çalışma prensibi, bir rotoru elektromanyetik kuvvetler kullanarak hareket ettirmeyi amaçlar. Rotor, step motorun çevirme eksenini oluşturan parçadır ve normal olarak bir mıknatıс bloğu veya parçalarından oluşur. İçinde bulunan stator ise, rotorun etrafındaki manyetik alanı oluşturan elektromanyetik bobinlerden oluşur.

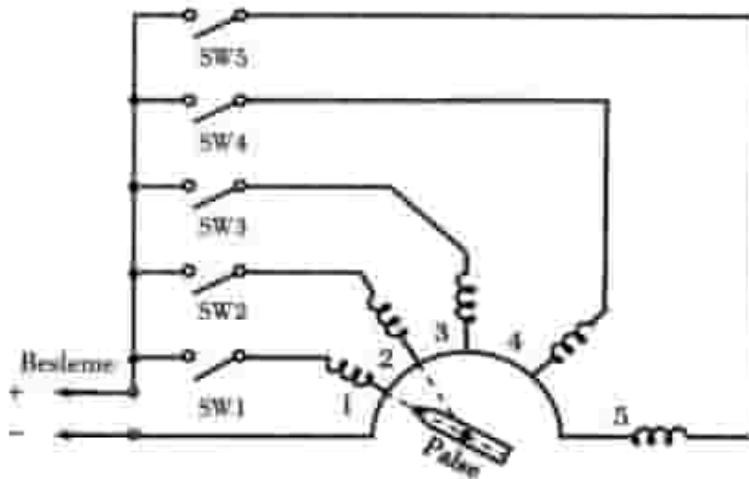
Step motorlar, genellikle hareket kontrolü için kullanıldıklarında, çalışma prensibi üzerine bir kontrol sistemi kullanılarak rotorun hareketini kontrol etmek için kullanılır. Bu sistemler, genellikle bir bilgisayar veya mikroişlemci tarafından yönetilir ve step motorun hareketlerini belirli bir şekilde kontrol etmek için kullanılır.

Step motorların yapısı genel anlamda rotor, stator ve de rulmanlardan oluşmaktadır. Burada rulmanlar, rotora bağlı olan şaftın rahat bir şekilde hareket etmesini sağlar. Statorun birden çok kutbu var olup kutup sayısı ayrıca motordan motora da değişmektedir. Fakat genellikle sayı sekiz olmaktadır. Kutupların polaritesi elektronik olan anahtarlar vasıtası ile sürekli olarak değişir. Rotorun mıknatılılığı sabit mıknatıс ile veya dış uyarım teknikleriyle meydana getirilir. (Robocombo, 2021)



Şekil 16 Step Motor Örneği

Öncelikle elektronik anahtarlar vasıtası ile bobinlere enerji verilir ve de rotor, üzerinde enerji bulunan bobinin karşısına geçerek durmaktadır. Motorun ne kadar fazla dönmesi isteniyor ise bobinlere sırası ile de o kadar puls sinyali verilmektedir. Bu dönmeye açısı genelde step motorda değişken olmaktadır. (Robocombo, 2021)



*Şekil 17 Step Motor Devre Şeması*

### 3.5.6 STEP MOTORUN ADIM AÇISININ ve ADIM SAYISININ HESAPLANMASI:

360° dönen step motorlarda gerekli adım açısının ( $Q_s$ ) bulunması için faz sayısını ( $N_s$ ) ve motordaki rotorun çıkışlı kutup sayısını ( $N_r$ ) bilmek gereklidir. Adım sayısının ( $S$ ) hesaplanması için bir adımın açısını ( $Q_s$ ) bulmak yeterlidir.

Step motorlar içerisinde bulunan mıknatıslar ve sarımlarla hareketi sağlar. Bu hareketin her bir adımına step denmektedir. Bir step'in tamamlanması için stesp motorda bulunan 2 artı ve iki eksi kutbun sırasıyla pulse değeri alarak 4 faz tamamlanır. Bu durum tamamlandığında 1 step yani bir adım atılmış olur. Daha önce bahsedildiği gibi step motor 360 dereceyi tamamlamak için 200 adım atması gerekmektedir.  $360/200=1,8$  derecelik açı ile bu durum sağlanmaktadır. Step motorlar verdigimiz adım sayısına göre hareket etmektedir. İstersek 360 derece istersek 10 ya da sonsuz bir şekilde döndürebiliriz. Step motora elektrik akımı geldiğin motor elektrik akımından dolayı tork oluşturmaktadır. Bu durum ise motorun tutma kuvvetini artırmaktadır. İşlem bitse bile tork sağlanmış olur fakat elektrik akımı kesilirse tork ortadan kalkar. (Rehberiniz, 2011)

### 3.5.7 STEP MOTORUN KULLANIM ALANLARI

Step motorlar, hareket kontrolünde yüksek hassasiyete sahip olmaları nedeniyle, birçok farklı uygulamada kullanılır. Örnek olarak:

- Otomatik endüstriyel makineler
- Robotik sistemler
- Uzaktan kumandalı araçlar

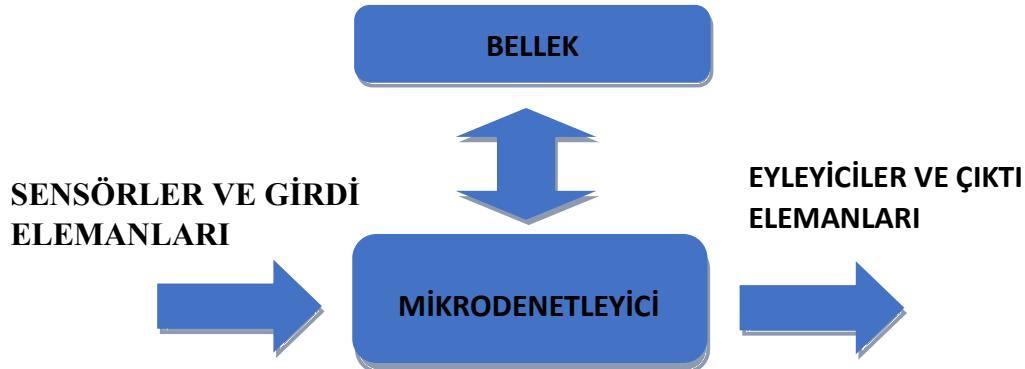
### **3.6 PROJEMİZDE KULLANACAĞIMIZ MOTORLAR:**

Projemizde düzlem üzerinde hata payı daha az bir hareket sağlamak amacıyla her tekerleğe 1 adet olmak üzere toplam 4 adet step motor kullanılacaktır. Bunun yanında robot kolun hareket mekanizmalarını sağlayacak olan kısmda ise literatürde daha kolay programlanabilmesi ve geribildirim verebilmesinden ötürü robot kol kısmında ise toplamda 3 adet servo motor, 3 adet mikro servo motor kullanılacaktır.

Robot kolun, x ve y düzleminde hareket etmesi için 1 adet, z düzleminde 1 ve 2 numaralı uzun hareket edebilmesi için iki adet servo motor kullanılmıştır. Tutucunun z düzleminde hareketi için 1 adet ve kışkaçların hareketlerini sağlamak amacıyla 2 adet olacak şekilde toplam 3 adet mikro servo motor kullanılmıştır.

### **3.7 ARDUINO NEDİR:**

Arduino, bir elektronik gömülü sistemdir. Farklı tanımlar bulunsa bile gömülü sistemler belirli bir işi gerçekleştirmek için tasarlanmış donanım, yazılım ve bazen mekanik bileşenlerden oluşturulmuş ve çoğu zaman kendisinden büyük bir sistemin içine gömülü olarak çalışan sistemlerdir. Bu sistem, bir mikroişlemci üzerinde çalışan bir yazılımla birlikte, fiziksel objelerle etkileşim kurmaya ve bunları otomatikleştirmeye yardımcı olmak için tasarlandı. (Taşdemir, 2017)



Arduino, elektronik projelerin tasarımını ve geliştirilmesi için basit ve kullanışlı bir araçtır. Örneğin, bir sensörün çıktısını kullanarak bir motoru kontrol etmek veya bir internet bağlantısı olan bir cihazın ısısını ölçmek gibi birçok farklı uygulamaya uygun olarak tasarlanmıştır. (Taşdemir, 2017)

Arduino'nun kullanımının popülerleşmesinin sebeplerine bakacak olursak:

- Geliştirme ortamının ve sürücülerinin kolay kurulumu.
- Geniş kütüphane destegine sahip olması.

- Birlikte çalışabileceği birçok shield destegine sahip olması. Arduino'ya bağlanamayan hemen hemen hiçbir sensör tipi bulunmamaktadır.
- Projelerde maliyet açısından uygunluğu. Ortalama bir arduino kartı maliyeti 30-40\$ civarında bulunmaktadır.
- Açık kaynaklı bir donanım olması.

### **3.7.1 ARDUINO ÇEŞİTLERİ:**

Arduino kartlarda birçok çeşit bulunmaktadır. Temel olarak bütün kartlarda benzer bileşenler olmakla beraber mikrodenetleyici modelleri farklı özelliklere sahiptir.

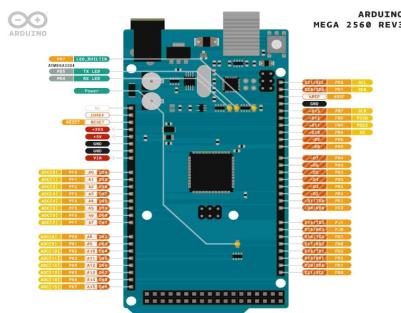
Aşağıdaki tabloda bazı arduino mikrodenetleyicilerinin modelleri ve özelliklerini görebilirsiniz.

AURDUINO	ATMEGA 2560	ATMEGA 328	ATMEGA 168
<b>ÇALIŞMA GERİLİMİ</b>	5V	5V	3.3V VEYA 5V
<b>DİJİTAL GİRİŞ/ÇIKIŞ</b>	54 (14 ADET PWM)	14 ( 6 ADET PWM)	14 (6 ADET PWM)
<b>ANALOG GİRİŞ PINLERİ</b>	16	6	6
<b>HER BİR G/Ç PINİ BAŞINA AKIM</b>	40 mA	40 mA	40 mA
<b>3.3V GERİLİM ÇIKIŞ AKIMI</b>	50 mA	50 mA	50 mA
<b>FLASH BELLEK</b>	256 Kb	32 Kb	16 Kb
<b>SRAM</b>	8 Kb	2 Kb	1 Kb
<b>EEPROM</b>	4 Kb	1 Kb	512 Bayt
<b>SAAT HIZI</b>	16 MHz	16 Mhz	8 Mhz 3.3V 16 Mhz 5V

Arduino Mega çalışma prensibi olarak Arduino Uno'ya benzeyen ve ondan sonra en ünlü olan bir diğer karttır. Arduino Mega yeterli sayıda giriş/çıkış pinine sahiptir. Arduino Mega diğer arduinolara göre daha büyük tasarımlı ve daha çok elektronik komponentin bağlanabildiği, daha fazla pine ihtiyaç duyulan projeler için kullanılmaktadır. Sistemimizde kullanacağımız motor sayısının çeşitliliğinden dolayı ve kullanacağımız elektronik komponent sayısının fazla olmasından dolayı Arduino Mega tercih edilmiştir.

Projemizde temel olarak Arduino Mega kullanılcaktır. Arduino Mega; 2560 ATmega2560 çipi üzerine kurulmuş bir mikro kontrolcü kartıdır. Üzerinde 54 dijital giriş/çıkış pini (15 tanesi pwm kapasiteli), 16 analog giriş, 4 UART(donanımsal seri port), 16MHz kristal osilatör, Usb bağlantısı, güç girişi, ICSP pinleri ve bir adet reset butonu içerir. (Taşdemir, 2017)

Arduino Mega kartının şeması:

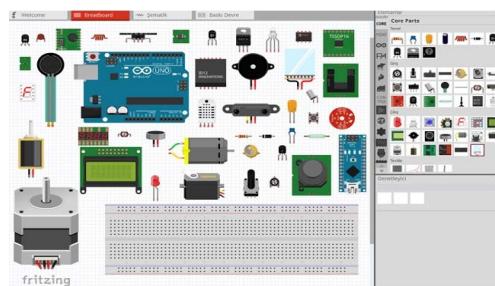


*Şekil 18 Arduino Mega Kart Şeması*

### 3.7.2 ARDUİNO İLE YAZILACAK KOD:

Arduino yazılımı bir geliştirme ortamı (IDE) ve kütüphanelerden oluşur. IDE, Java dilinde yazılmıştır ve Processing adlı dilin ortamına dayanmaktadır. Kütüphaneler ise C ve C++ dillerinde yazılmıştır ve AVR-GCC ve AVR Libc. ile derlenmiştir.

Yapılacak olan devre için öncelikli olarak ücretsiz bir sürüm olan ve Arduino devresi çizmek için kullanılan Fritzing programı kullanılacaktır. Fakat Fritzing programında sadece komponentlerin şeması kurulabilmektedir, simülasyon özelliği bulunmamaktadır.



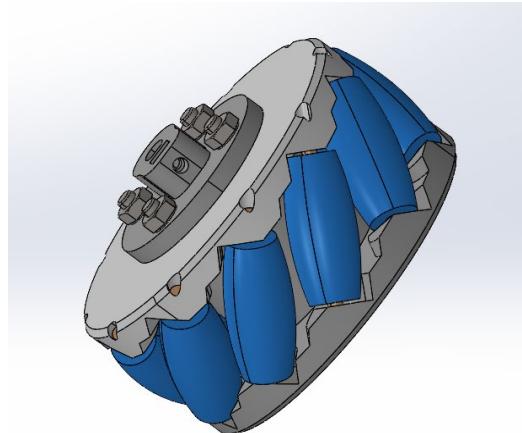
*Şekil 19 Fritzing Programı*

Sonrasında ise toplam kullanılacak olan motor sayıları, tipleri ve Bluetooth bağlantılarına göre aralarında uyumu sağlayacak bir iletişim yazılımı yapılması gerekmektedir. Bu yazılım Arduino'nun açık kaynak yazılımları arasında bulunmaktadır. Projemizde kullanacağımız yazılım genel manada açık kaynak yazılımlarının ve senkronizasyonu paylaşılmış yazılımların üzerinde çalışılmış versiyonları olacaktır.

## 4. ARAŞTIRMA SONUÇLARI

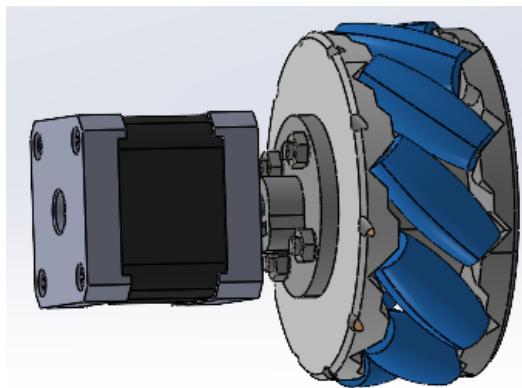
### 4.1 SOLIDWORKS ORTAMINDA TASARIM VE MONTAJ

SolidWorks ortamında çizilen ve GrabCAD sitesinden alınan çizimler birleştirilmiştir. Birleştirme işlemine mechannum tekerleklerin montajı ile başlanmıştır. Her tekerlekte 10 adet silindirik (roller) eleman bulunmaktadır.



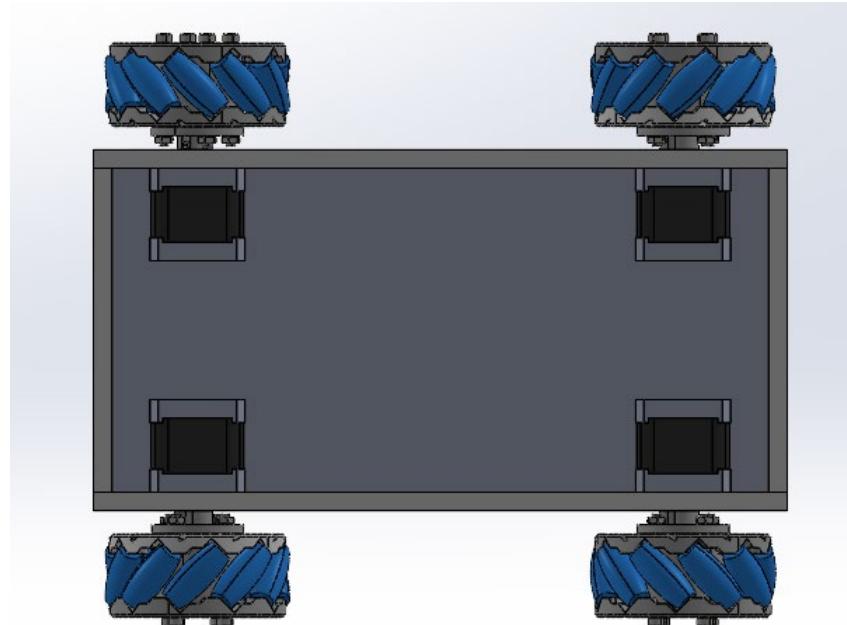
*Şekil 20 Mechannum Teker Montajı*

Sonrasında Step motor çizimi GrabCAD ortamından alınarak tekerlek ile montajlanmıştır.



*Şekil 21 Mechannum Teker ve Step Motor Montajı*

Montajlanan mehanik tekerlek ve step motorlardan 4 adet kopyalanarak gövde tasarımları oluşturulmuş ve tasarıma yerleştirilmiştir.



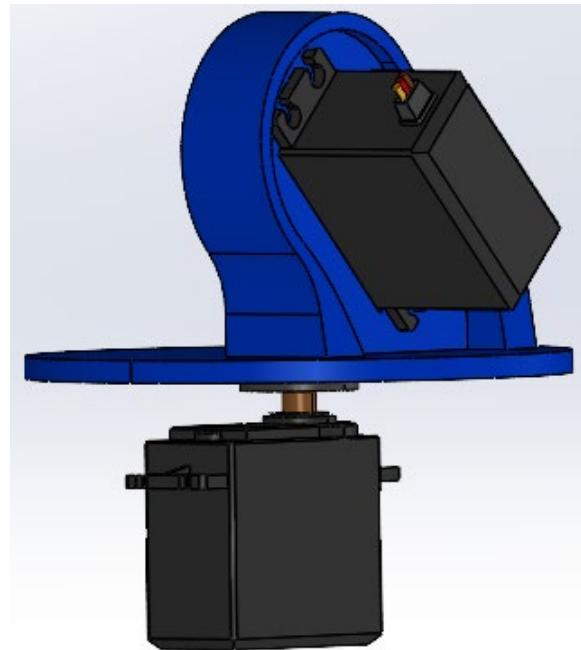
*Şekil 22 Gövde ve Tekerleklerin Montajı*

Tekerlek, gövde ve step motorlarının montajından sonra robot kola sıra gelmiştir. Robot kol tasarımı için GrabCAD ortamından servo ve mikro servo motor tasarımları alınmıştır.



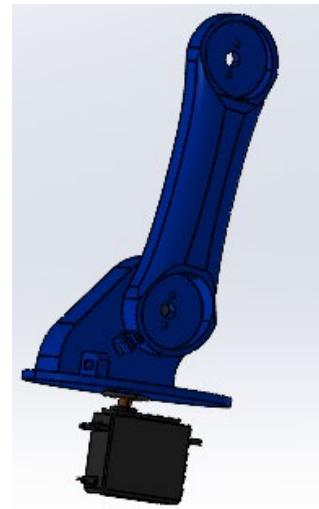
*Şekil 23 Servo Motorların SolidWorks Ortamındaki Görüntüsü*

Sonrasında robotun kasaya bağlanacağı ana gövde uzvu eklenerek uzva servo motor ve 2. Uzva bağlanacak servo motor eklenmiştir.



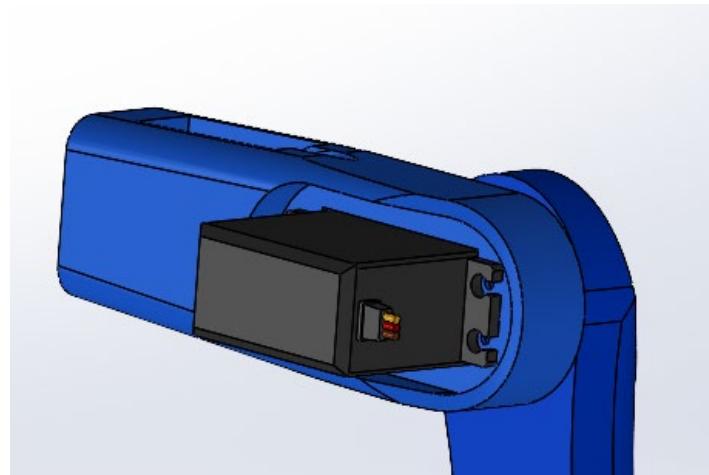
*Şekil 24 Servo Motorlar ile Robot Gövdesi Montajı*

1. uzuv diğer servo motora montajlanarak sisteme dahil olmuştur.



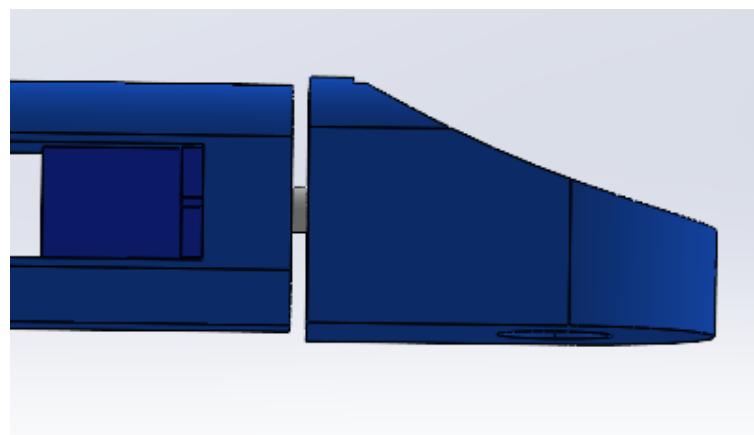
*Şekil 25 1. Uzvun Eklenmesi*

2. uzuv eklenerek üçüncü servo motor robot kola montajlanmıştır.



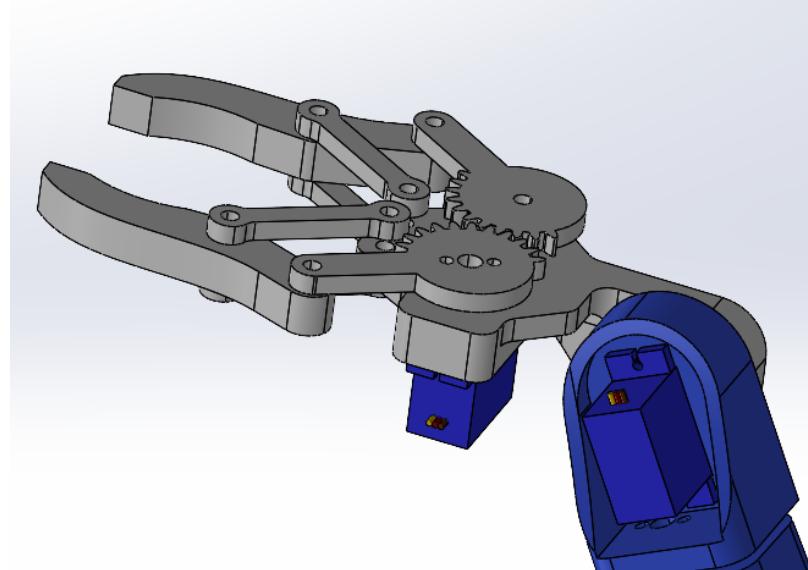
Şekil 26 Servo Motor ve 2. Uzuv

İlk mikro servo motor ve üçüncü robot kol parçası eklenmiştir.



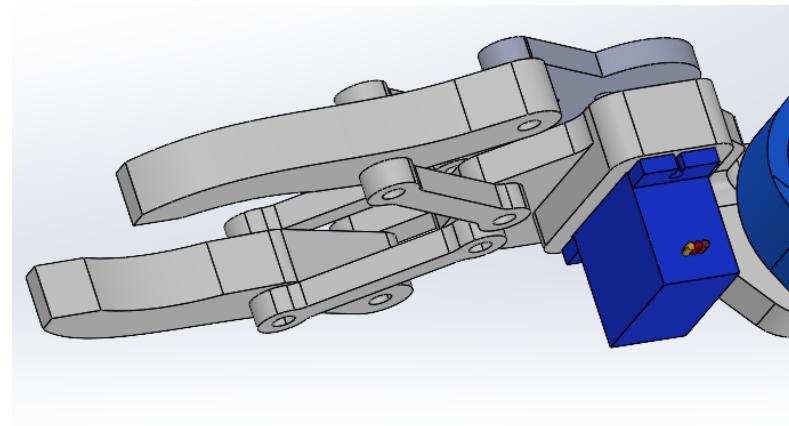
Şekil 27 3. Uzuv ve Mikro Servo Motor

Tutucu bölümünü yukarı ve aşağı hareketini sağlayacak olan mikro servo motor bağlanarak robot kolun tutucu ile bağlantısı sağlanmıştır.



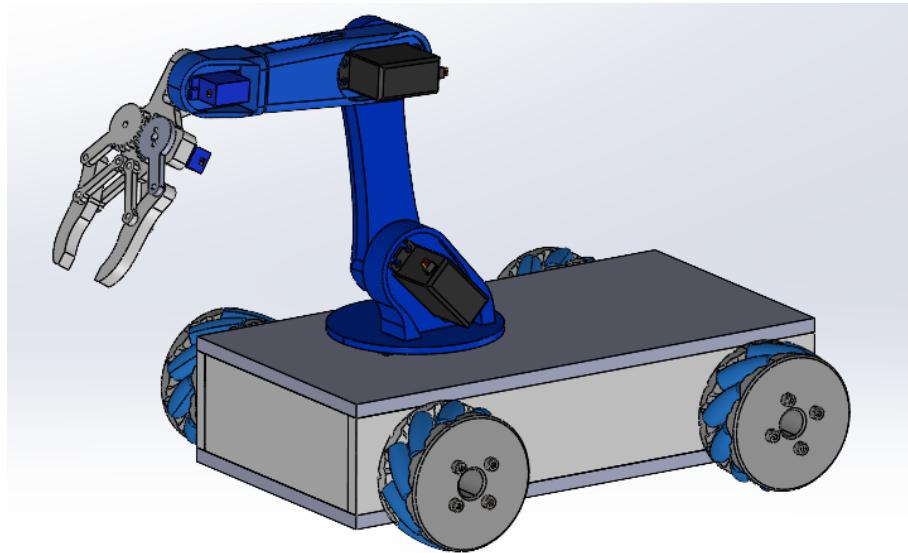
Şekil 28 Gripper ve Mikro Servo Motor

Son olarak kıskacın dişlilerine döndürme hareketini verecek olan mikro servo motor eklenmiştir.



Şekil 29 Dişliler İçin Gereken Mikro Servo Motor

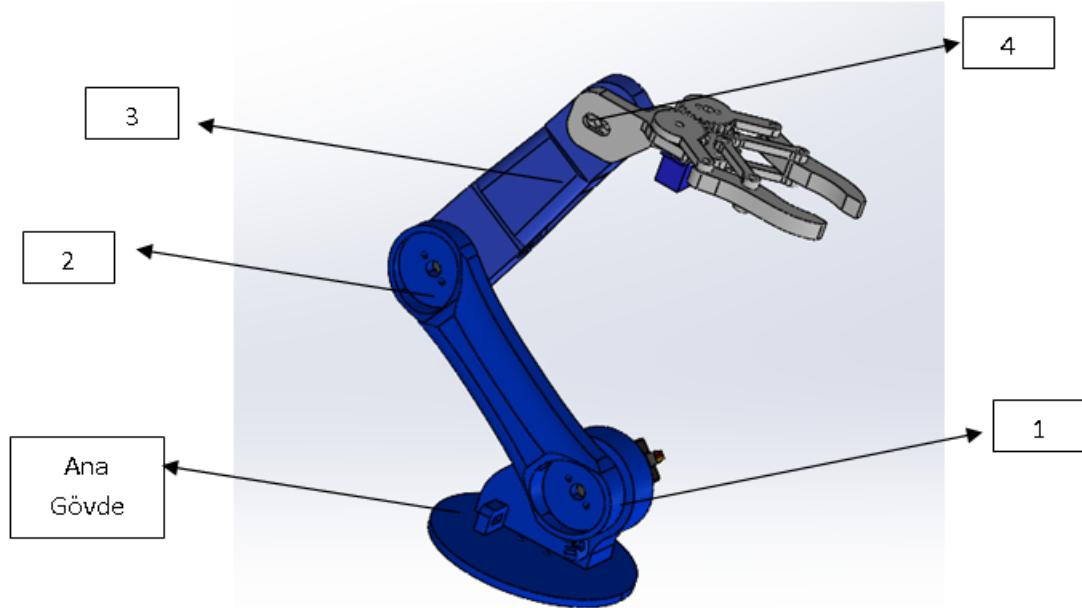
Robot kol gövdeye montajlanarak tasarım tamamlanmıştır.



*Şekil 30 Tasarımın Son Hali*

## 4.2 ROBOT KOL SİSTEMİNİN MATLAB SIMULINK ORTAMINA TAŞINMASI

Robot kol mekanizmasında sistemin analizini ve gerekli tork kuvvetlerini bulabilmek için Sistem Öncelikli olarak SolidWorks programında çizildi ve montajlandı. Montajlanan çizimin Eksen doğrultuları Matlab ortamında kullanılan standart eksen tipine göre ayarlandı ve tamamlandı.



*Şekil 31 Simulink Ortamında Robot Kol*

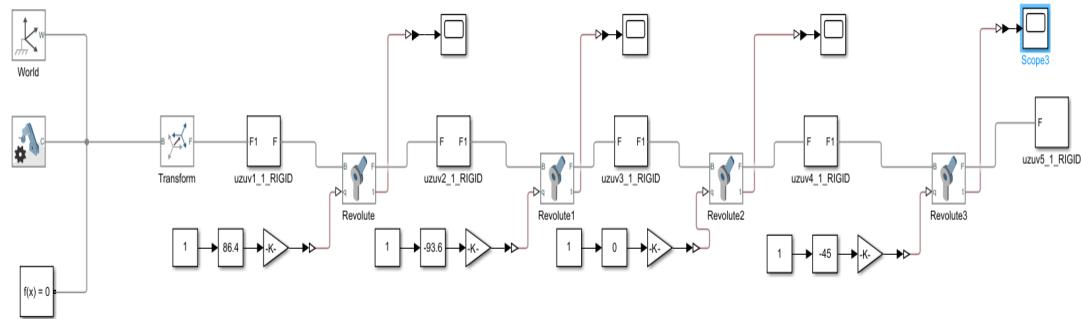
Sonrasında Matlab fonksiyonu olan ve SolidWorks programına bir ekleni olarak gelen Simscape MultiBody Link ile sistemin montaj özellikleri ve montaj ilişkileri Matlab Simulink ortamında şema haline getirildi. Getirilen şemada mafsal olarak 4 adet mafsal atanmıştır. Bu mafsallar numaralara göre:

- 1 Numara: Kolun ana gövdesinden sonraki öne ve geriye hareketi sağlayacak olan uzvu bağlayan mafsaldır. Bir servo motor ile hareketi desteklenecektir
- 2 Numara: Bu bölümdeki mafsal kolun yüksekliğini ve öne geriye ulaşabilmesini sağlayacak olan uzvu hareket ettirecek döner mafsaldır. Servo motor ile desteklenecektir.
- 3 Numara: Bu bölgedeki mafsal sistemin kıskacının bağlı olduğu uzvu 360 derece döndürecek olan döner mafsaldır. Burası mikro servo motorlar ile desteklenecektir.
- 4 Numara: Kıskaç mekanizmasını yukarı ve aşağı z ekseninde hareket kapasitesi kazandıracak olan mikro servo ile desteklenecek bölümdür.

Ekstra ile gösterilmiş bölge ise sistemin analizinde bulunmamaktadır. Burası Robotun ana sunta gövdesinde 360 derece hareketini sağlayacak olan servo motor ile desteklenecektir.

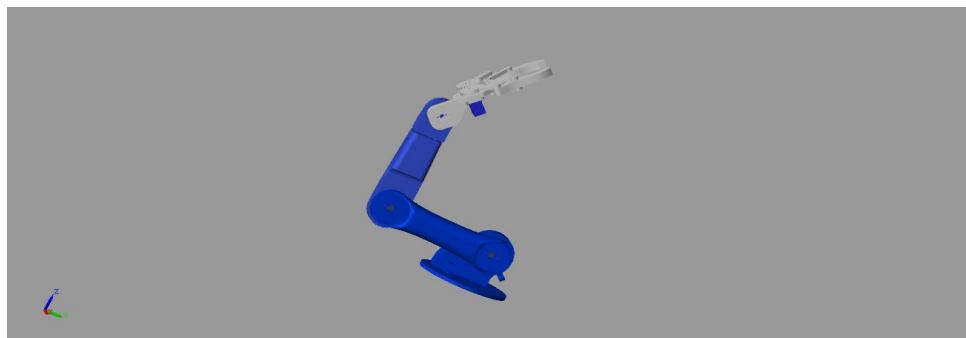
Bu bölgenin Matlabda bulunmama sebebi kendi ana gövdesi olmamasından dolayı eksenel yönlerde sıkıntı çıkarmasıdır.

Sonraki aşama olarak gelen Simulink ana şeması alt görsel şeklindedir. Bu görsele eklenen özellikler:



Şekil 32 Simulink Ana Şeması

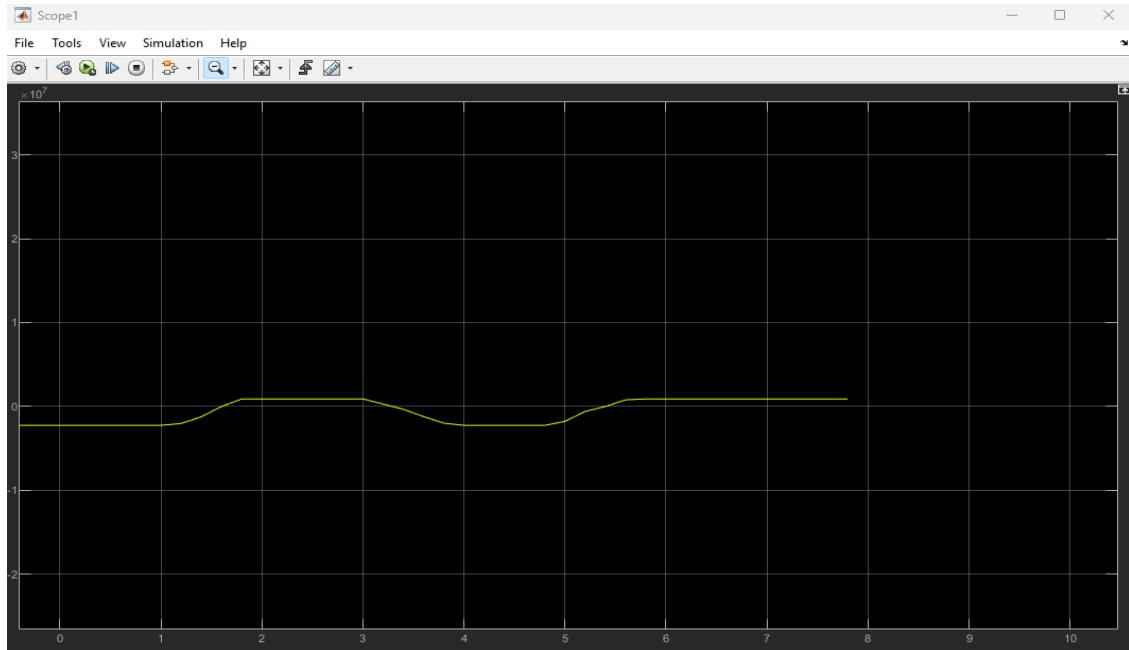
Mafsalların alt bölgelerinden gelen girdi verileri ve scope sistemidir. Bu veriler için öncelikle bir adet sabit eklendi. Bu sabitin sonrasında -180 ve 180 derece arasında sınırlı sabite ekleme özelliği eklenerek sistemin tam şekilde hareketi sağlandı. Ve radyanın dereceye dönüşmesi için çevirici eklendi. Son olarak bu eklenen verileri matlab ortamından fiziki ortama aktarmak için Simulink PS Converter eklendi. Bu eklenen Converteri mafsala eklemek için mafsaların girdilerini otomatikten manuele olarak sistemi elle kontrole açık hale getirildi. Aynı aşamalar bütün mafsallar için yapıldı. Sonrasında sistemin çıktısını görebilmek için mafsalların sensing bölmelerinden Tork aktivasyonu sağlanarak çıktıları görmek üzere bu sefer PS-Simulink Converter eklenerek sonrasında Scope eklendi. Son aşamada sistemin belirli aralıkta değil tam olarak görülmesi için model simülasyon süresi sonsuza eklenecek anlık verilerin görülebileceği bir hale getirildi.



Şekil 33 Simülasyon Ortamında Robot Kol

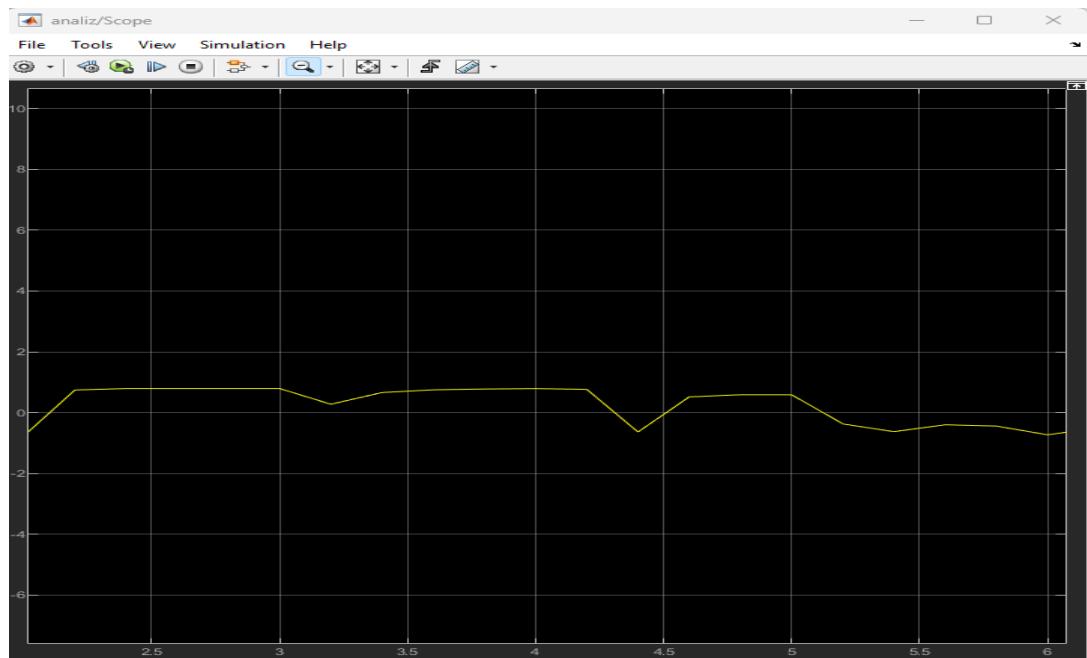
Sistemin simülasyonu sonrası her mafsalda girilen hareket sonrası çıkış hareketinde gerekli kuvvet ve tork değerlerinin grafikleri:

Mafsal 1 için 30 derecelik hareket sonucu:



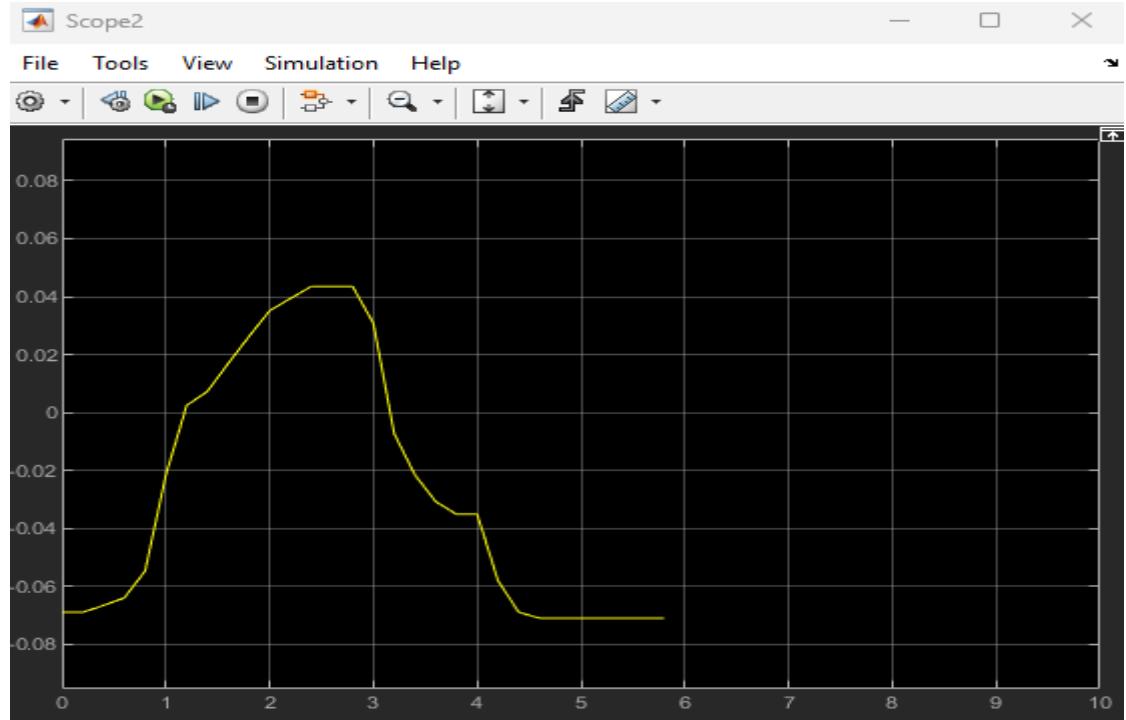
Şekil 34 Mafsal 1 İçin  $30^\circ$  Hareket Sonucu

Mafsal 2 için analiz sonucu:



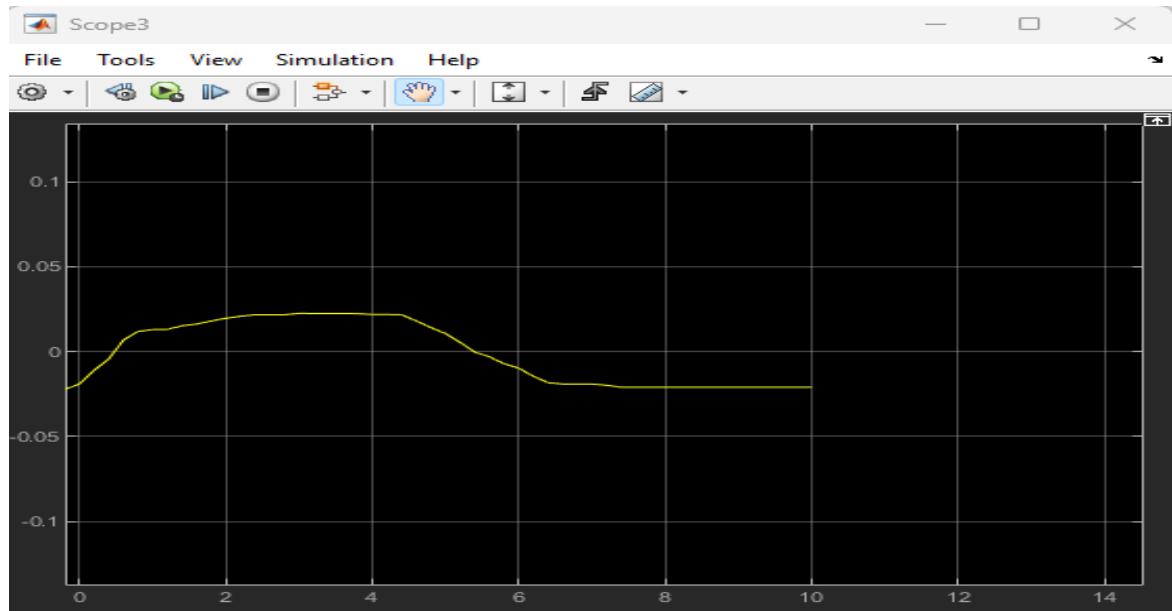
Şekil 35 Mafsal 2 İçin  $30^\circ$  Hareket Sonucu

Mafsal 3 için analiz sonucu:



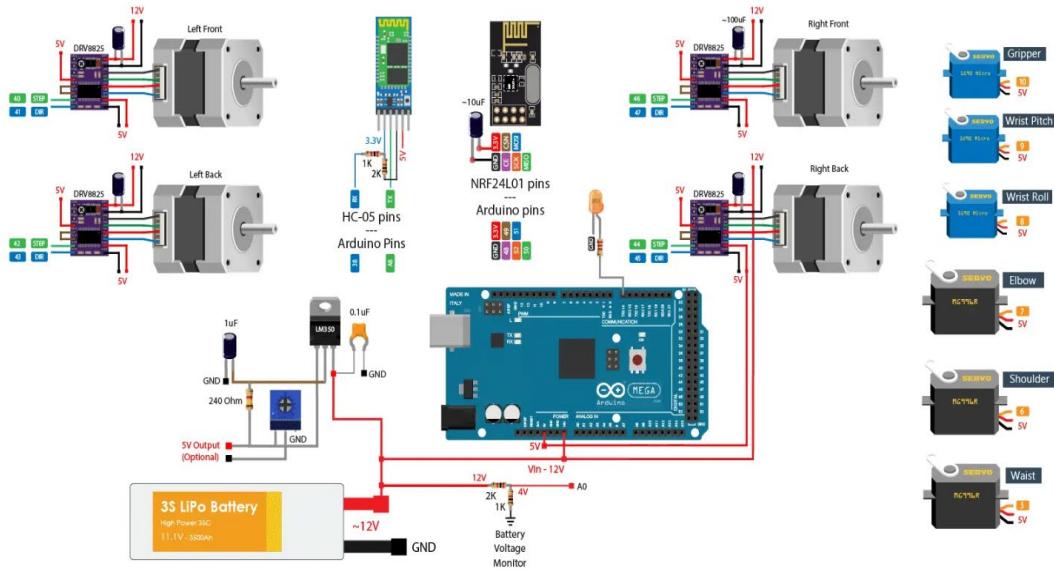
Şekil 36 Mafsal 3 İçin  $30^\circ$  Hareket Sonucu

Mafsal 4 için analiz sonucu:



Şekil 37 Mafsal 4 İçin  $30^\circ$  Hareket Sonucu

### 4.3 FRITZING ŞEMASI VE SİSTEMİN 2. ELEKTRİKSEL TEMSİLİ ŞEMASI



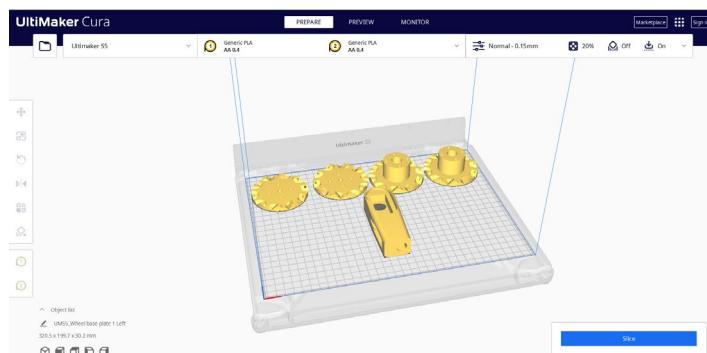
Şekil 38 Fritzing Şeması Ve Sistemin 2. Elektriksel Temsili Şeması

Sistemi örnek olarak aldığımız fonksiyon olarak aynı görevde olan Robotistan firması tarafından yapılan modelden örnek alınmıştır. Fritzing programı ücretli bir program olmasından dolayı ve ücretsiz versiyonuyla yapılmasından dolayı gerekli şemalar tam olarak gösterilememiştir. Yalnızca ana hatlar ve ana elemanlar bulunmaktadır.

### 4.4 ÜRETİM AŞAMASI

#### 4.4.1 3D YAZICI AŞAMASI

3D modleme yazılımı olan SolidWorks programında oluşturulan tasarım, UltiMaker Cura programına .stl formatında aktarıldı.

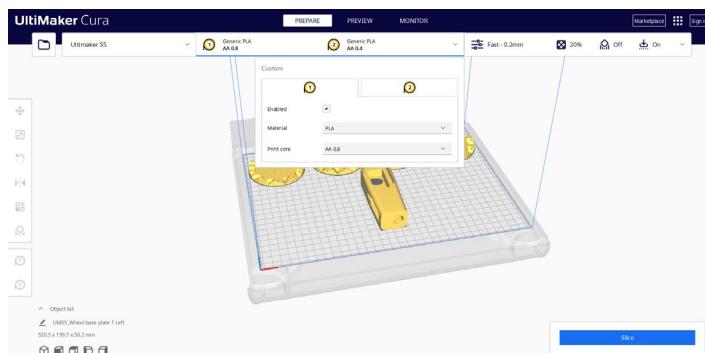


Şekil 39 Ultimaker Cura Programında İçe Aktarma

Model içeri aktarıldıktan sonra baskı ayarları yapıldı. Baskı malzemesi, tabaka kalınlığı, dolgu yoğunluğu, hız ve destek malzemesi gibi parametreler girildi.

Baskı malzemesi olarak PLA filament tercih edilmiştir. Bunun sebepleri aşağıdaki şekilde sıralanmıştır:

- PLA filamentin yükseklik toleransı  $\pm 0.03$  mm ve genişlik toleransı  $\pm 0.05$  mm'dir. Bu tolerans değerleri, baskıdıraki hataları minimize etmektedir.
- Pürüzsüz bir yüzey görüntüsü sunar.
- Maliyeti ucuzdur.



*Şekil 40 Ultimaker Cura Programında Parametre Girdisi Örneği*

Tüm bu işlemler tamamlandıktan sonra baskı işlemine başlandı.



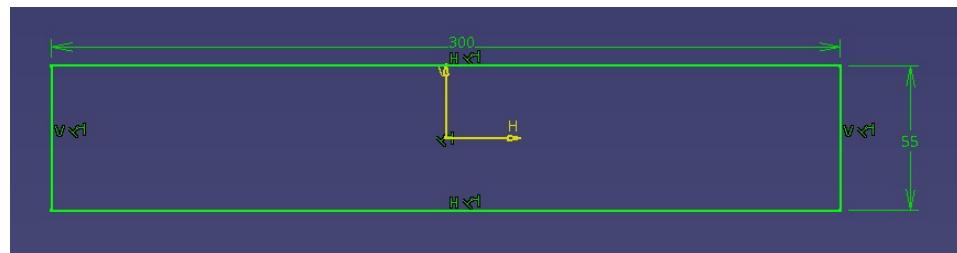
*Şekil 41 3D Yazıcı Baskı Anı*



Şekil 42 3D Yazıcı Baskı Anı

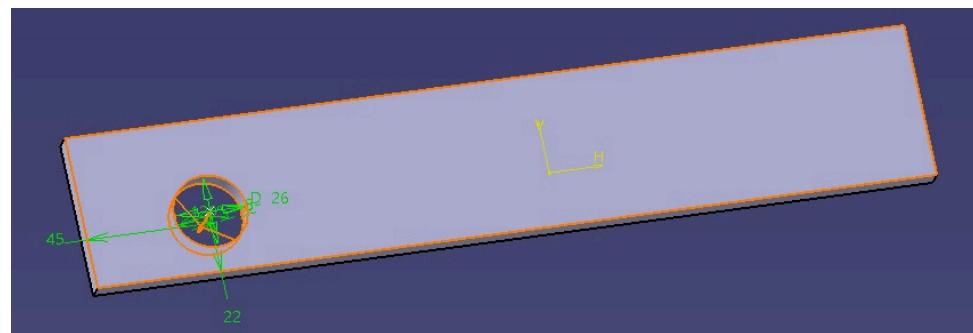
#### 4.4.2 ARAÇ GÖVDESİNİN ÜRETİM AŞAMASI

Araç gövdesinin üretim aşamasına, CATIA programında üretilcek olan gövde parçasının tasarımını ile başlandı.



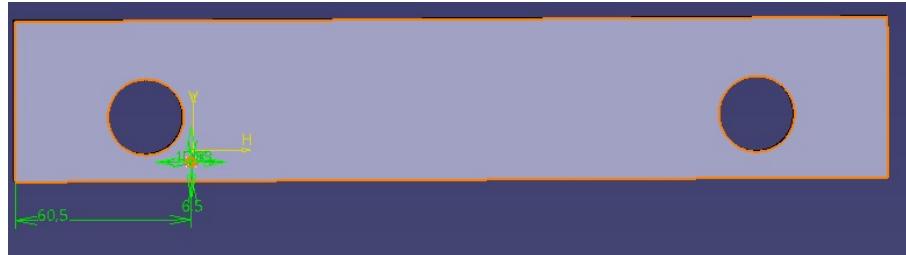
Şekil 43 CATIA Tasarım Aşaması

Çizim ölçülendirdikten sonra katı hale getirildi. Aracın tekerleklerinin geleceği kısmın delikleri eklendi.



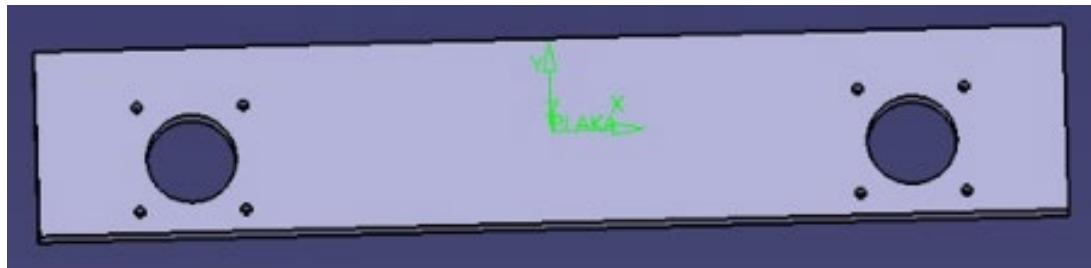
Şekil 44 CATIA Ortamında Çapın Eklenmesi

İlk tekerliğin geleceği çap ölçüldürülüp eklendikten sonra yerleşecek olan diğer tekerliğin çapı eklendi. Sonraki aşamada gövdenin içinde bulunan step motorların montajında gerekli olan civata delikleri ölçüldürülüp eklendi.



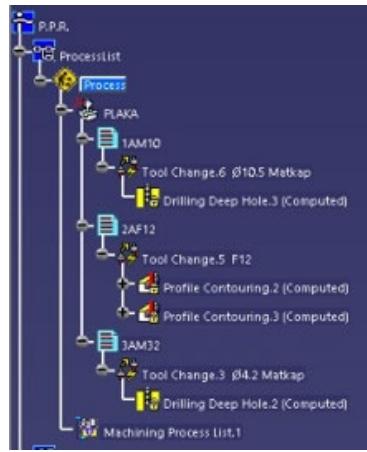
*Şekil 45 CATIA Ortamında Civata Deliği Eklentimesi*

Eklenen ilk civata delığının ardından, diğer civata deliklerinin geleceği alanlar hesaplandı ve ilk civata deliği bu alanlara çoğaltıldı.

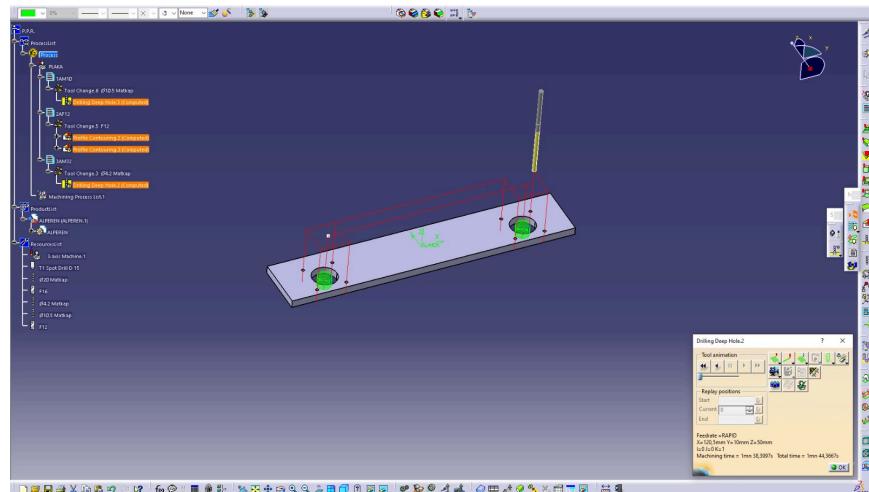


*Şekil 46 CATIA Ortamında Tasarımın Son Hali*

Bu aşamadan sonra CATIA programının CAM modülü kullanılarak CNC makinesi için gereken kodlar oluşturuldu. Öncelikle tekerleklerin geleceği çaplar için, ön delik programı yapıldı. Bu programın yapılmasındaki amaç, hem dairesel hareket yapılırken oluşacak talaşın rahat bir şekilde boşaltılmasını sağlamak hem de dairesel harekette kullanılacak takımın kesici elmaslarını daha az alanda kullanılarak hasar görme olasılığını azaltmaktadır. Daha sonra büyük çapları oluşturmak için, kesici takımın dairesel hareket yapmasını sağlayan program hazırlandı. En son aşamada ise, civataların delikleri için bir delik programı hazırlandı.



Şekil 47 Programların Ürün Ağacındaki Görüntüsü



Şekil 48 CAM Programlarının Önizlemesi

Önce, CNC makinesine bağlanan yan gövde parçası sıfırlandı.



Şekil 49 CNC Makinesinde Sıfırlama

Parça sıfırlandıktan sonra, hazırlanan programlar yardımıyla kesici takımların hareketleri sağlandı.



*Şekil 50 CNC Makinesinde İşlenme Anı*

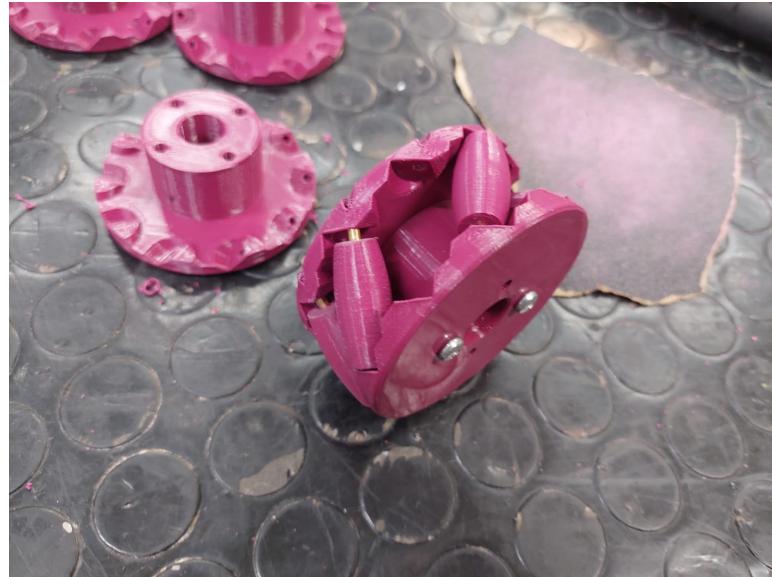
Tüm bu aşamalardan sonra, parça son halini aldı.



*Şekil 51 Yan Gövde Son Hali*

#### 4.5 MONTAJ AŞAMASI

3D yazıcıdan çıkan tekerlek parçaları M4 vidalar ve 3 mm bakır ile tel tasarımda öngörüldüğü şekilde montajlanmıştır. Toplamda 40 silindir 4 adet tekerleğe eşit dağılmıştır.



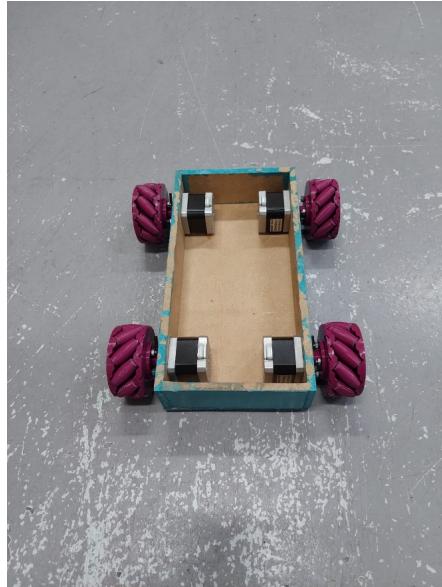
Şekil 52 Tekerlek Montaj Görüsü

Step motorlar araç gövdesine montajlanmıştır.



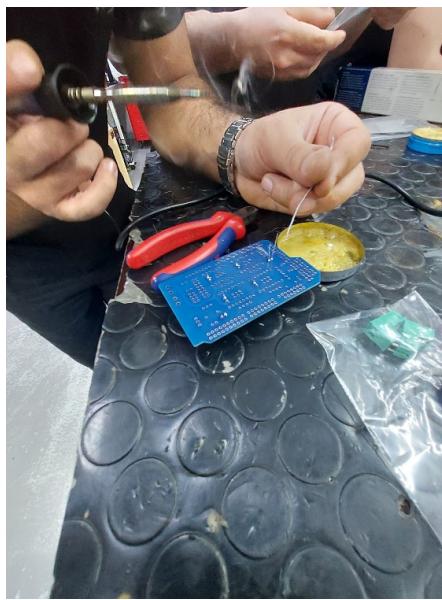
Şekil 53 Step Motor Bağlanması

Monte edilen step motorlara tekerlekler takılmıştır.



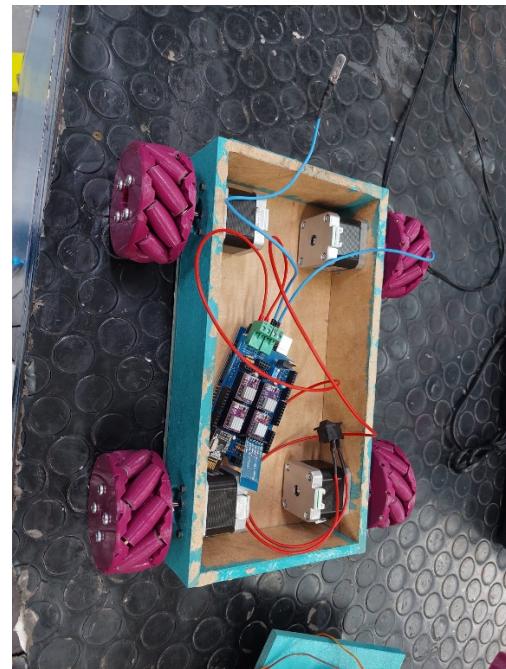
*Şekil 54 Tekerlek ve Step motor birleştirilmesi*

PCB entegreleri (direnç, regülatör, trimpot, jumper soketleri vb.) entegreler tek tek lehimlenip PCB entegresi oluşturulmuştur.



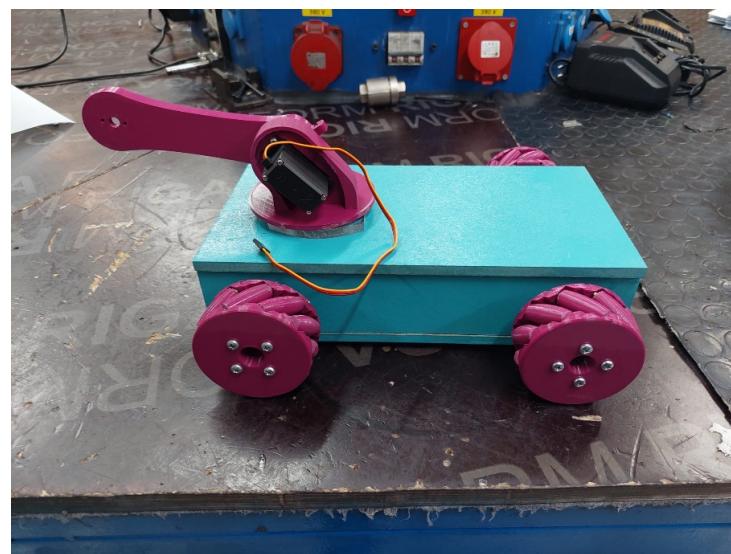
*Şekil 55 PCB'nin Lehimlenmesi*

Arduino step motorlara bağlanmıştır.



Şekil 56 PCB ve Arduino Bağlantılarının Yapılması

3D yazıcıdan çıkarılan robot kol parçaları birbirine tasarımda öngörüldüğü şekilde montajlanıp servo motor bağlantıları yapılmıştır.



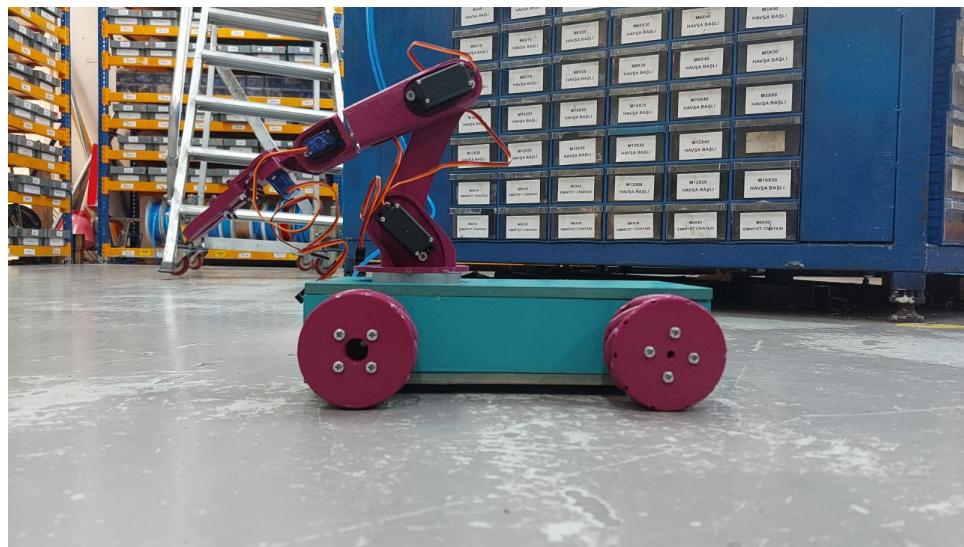
Şekil 57 Uzuvların Eklenmesi

Ana uzuvarlar eklendikten sonra Robotun tutucu kısmı birleştirilmiştir.



*Şekil 58 Tutucu Montajı*

Sonrasında robotun PCB bağlantıları ve elektronik devreleri lehimlenerek gerekli servo motor bağlantıları Arduino kartına bağlanmıştır. Tutucu ve uzuvarlar monte edilerek robotun son haline ulaşılmıştır.



*Şekil 59 Robotun Montajlanmış Hali*

## 4.6 YAZILIM

Yazılım için öncelikle Robotistan firmasının REX Chassis Serisi Arduino 4WD Robot Kollu Pro Robotunun yazılımı örnek olarak alınmıştır. Bunun yanında 2bndy5 adlı GİTHUB kullanıcısının tutucu yazılımları da kullanılmıştır.

Önce 6 servo, 4 step motor ve Bluetooth iletişimini sağlamak için bir bölüm oluşturuldu. Kurulum bölümünde steplerin maksimum hızını ayarlıyoruz, servo motorların bağlı olduğu pimleri tanımlıyoruz, Bluetooth iletişimini başlatıyoruz ve robot kolunu başlangıç konumuna getiriyoruz. Bu bölüm:

```
#include <SoftwareSerial.h>
#include <AccelStepper.h>
#include <Servo.h>
```

**Servo** servo01; (ana gövde servo motoru)

**Servo** servo02; (1. uzuv servo motoru)

**Servo** servo03; (2. uzuv servo motoru)

**Servo** servo04; (3. uzuv mikro servo motoru)

**Servo** servo05; (4. uzuv mikro servo motoru)

**Servo** servo06; (Gripper mikro servo motoru)

**SoftwareSerial** **Bluetooth(A8, 38);**

Step motorlarını ve kullanılacak pimleri tanımlanan bölüm:

**AccelStepper LeftBackWheel(1, 42, 43); Stepper1 için**

**AccelStepper LeftFrontWheel(1, 40, 41); Stepper2 için**

**AccelStepper RightBackWheel(1, 44, 45); Stepper3 için**

**AccelStepper RightFrontWheel(1, 46, 47); Stepper4 için**

```
int wheelSpeed = 1500;
```

```
int lbw[50], lfw[50], rbw[50], rfw[50];
```

konumları ve adımları depolamak için diziler:

```
int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos;
```

Anlık pozisyonlar

```
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos, servo6PPos;
```

```
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50],  
servo06SP[50];
```

```
int speedDelay = 20;
```

```
int index = 0;
```

```
int dataIn;
```

```
int m = 0;
```

```
void setup() {
```

*Stepler için başlangıç değerlerini ayarlama*

```
LeftFrontWheel.setMaxSpeed(3000);
```

```
LeftBackWheel.setMaxSpeed(3000);
```

```
RightFrontWheel.setMaxSpeed(3000);
```

```
RightBackWheel.setMaxSpeed(3000);
```

```
pinMode(led, OUTPUT);
```

```
servo01.attach(5);
```

```
servo02.attach(6);
```

```
servo03.attach(7);
```

```

servo04.attach(8);
servo05.attach(9);
servo06.attach(10);
Bluetooth.begin(38400);
Bluetooth.setTimeout(5);
delay(20);
Serial.begin(38400);

```

Robot kolu Başlangıç pozisyonuna alma kısmı:

```

servo1PPos = 90;
servo01.write(servo1PPos);
servo2PPos = 100;
servo02.write(servo2PPos);
servo3PPos = 120;
servo03.write(servo3PPos);
servo4PPos = 95;
servo04.write(servo4PPos);
servo5PPos = 60;
servo05.write(servo5PPos);
servo6PPos = 110;
servo06.write(servo6PPos);
}

```

Ardından döngü bölümünde, gelen veriler kontrol edilir. Bu bölüm:

```
void loop() {
```

Gelen datayı inceleme ve şart koşma için kısım:

```
if (Bluetooth.available() > 0) {  
    dataIn = Bluetooth.read();  
  
    if (dataIn == 0) {  
        m = 0;  
    }  
    if (dataIn == 1) {  
        m = 1;  
    }  
    if (dataIn == 2) {  
        m = 2;  
    }  
    if (dataIn == 3) {  
        m = 3;  
    }  
    if (dataIn == 4) {  
        m = 4;  
    }  
    if (dataIn == 5) {  
        m = 5;  
    }  
    if (dataIn == 6) {  
        m = 6;  
    }  
}
```

```
    if (dataIn == 7) {  
        m = 7;  
    }  
    if (dataIn == 8) {  
        m = 8;  
    }  
    if (dataIn == 9) {  
        m = 9;  
    }  
    if (dataIn == 10) {  
        m = 10;  
    }  
    if (dataIn == 11) {  
        m = 11;  
    }  
    if (dataIn == 12) {  
        m = 12;  
    }  
    if (dataIn == 14) {  
        m = 14;  
    }  
    if (dataIn == 16) {  
        m = 16;  
    }  
    if (dataIn == 17) {  
        m = 17;  
    }
```

```
    if (dataIn == 18) {  
        m = 18;  
    }  
    if (dataIn == 19) {  
        m = 19;  
    }  
    if (dataIn == 20) {  
        m = 20;  
    }  
    if (dataIn == 21) {  
        m = 21;  
    }  
    if (dataIn == 22) {  
        m = 22;  
    }  
    if (dataIn == 23) {  
        m = 23;  
    }  
    if (dataIn == 24) {  
        m = 24;  
    }  
    if (dataIn == 25) {  
        m = 25;  
    }  
    if (dataIn == 26) {  
        m = 26;  
    }
```

```

if (dataIn == 27) {
    m = 27;
}

```

Tekerleklerin ilerlemesi için girdi bölümü:

```

if (m == 4) {
    moveSidewaysLeft();
}

if (m == 5) {
    moveSidewaysRight();
}

if (m == 2) {
    moveForward();
}

if (m == 7) {
    moveBackward();
}

if (m == 3) {
    moveRightForward();
}

if (m == 1) {
    moveLeftForward();
}

if (m == 8) {
}

```

```

    moveRightBackward();

}

if (m == 6) {
    moveLeftBackward();
}

if (m == 9) {
    rotateLeft();
}

if (m == 10) {
    rotateRight();
}

}

if (m == 0) {
    stopMoving();
}

```

Tekerlerin hız ayarı için:

```

if (dataIn > 30 & dataIn < 100) {
    wheelSpeed = dataIn * 20;
}

```

Robot kolunu kontrol etmek için aynı yöntemi kullanıyoruz. Girdiye göre robot kol hareketini sağlamak için gerekli komutlar

Servo 1 için

```

while (m == 16) {
    if (Bluetooth.available() > 0) {
        m = Bluetooth.read();
    }
}

```

```

    }

    servo01.write(servo1PPos);

    servo1PPos++;

    delay(speedDelay);

}

while (m == 17) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    servo01.write(servo1PPos);

    servo1PPos--;

    delay(speedDelay);

}

```

Servo 2 için

```

while (m == 19) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    servo02.write(servo2PPos);

    servo2PPos++;

    delay(speedDelay);

}

while (m == 18) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }
}

```

```

    servo02.write(servo2PPos);

    servo2PPos--;
    delay(speedDelay);

}

```

Servo 3 için

```

while (m == 20) {
    if (Bluetooth.available() > 0) {
        m = Bluetooth.read();
    }
    servo03.write(servo3PPos);
    servo3PPos++;
    delay(speedDelay);
}

while (m == 21) {
    if (Bluetooth.available() > 0) {
        m = Bluetooth.read();
    }
    servo03.write(servo3PPos);
    servo3PPos--;
    delay(speedDelay);
}

```

Servo 4 için

```

while (m == 23) {
    if (Bluetooth.available() > 0) {
        m = Bluetooth.read();
    }
}

```

```

    }

    servo04.write(servo4PPos);

    servo4PPos++;

    delay(speedDelay);

}

while (m == 22) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    servo04.write(servo4PPos);

    servo4PPos--;

    delay(speedDelay);

}

```

Servo 5 için

```

while (m == 25) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    servo05.write(servo5PPos);

    servo5PPos++;

    delay(speedDelay);

}

while (m == 24) {

    if (Bluetooth.available() > 0) {

```

```

    m = Bluetooth.read();
}

servo05.write(servo5PPos);
servo5PPos--;
delay(speedDelay);
}

```

Servo 6 için

```

while (m == 26) {
    if (Bluetooth.available() > 0) {
        m = Bluetooth.read();
    }
    servo06.write(servo6PPos);
    servo6PPos++;
    delay(speedDelay);
}

while (m == 27) {
    if (Bluetooth.available() > 0) {
        m = Bluetooth.read();
    }
    servo06.write(servo6PPos);
    servo6PPos--;
    delay(speedDelay);
}

if (dataIn > 101 & dataIn < 250) {
    speedDelay = dataIn / 10;
}

```

Hareket kayıt sistemini oluşturmak için yapılan hareketin kaydedilmesi:

```

if (m == 12) {
    if (index == 0) {
        LeftBackWheel.setCurrentPosition(0);
        LeftFrontWheel.setCurrentPosition(0);
        RightBackWheel.setCurrentPosition(0);
        RightFrontWheel.setCurrentPosition(0);
    }
    lbw[index] = LeftBackWheel.currentPosition();
    lfw[index] = LeftFrontWheel.currentPosition();
    rbw[index] = RightBackWheel.currentPosition();
    rfw[index] = RightFrontWheel.currentPosition();
    servo01SP[index] = servo1PPos;
    servo02SP[index] = servo2PPos;
    servo03SP[index] = servo3PPos;
    servo04SP[index] = servo4PPos;
    servo05SP[index] = servo5PPos;
    servo06SP[index] = servo6PPos;
    index++;
    m = 0;
}

if (m == 14) {
    runSteps();

    if (dataIn != 14) {

```

```

stopMoving();

memset(lbw, 0, sizeof(lbw));
memset(lfw, 0, sizeof(lfw));
memset(rbw, 0, sizeof(rbw));
memset(rfw, 0, sizeof(rfw));
memset(serv01SP, 0, sizeof(serv01SP));
memset(serv02SP, 0, sizeof(serv02SP));
memset(serv03SP, 0, sizeof(serv03SP));
memset(serv04SP, 0, sizeof(serv04SP));
memset(serv05SP, 0, sizeof(serv05SP));
memset(serv06SP, 0, sizeof(serv06SP));
index = 0;

}

}

}

LeftFrontWheel.runSpeed();
LeftBackWheel.runSpeed();
RightFrontWheel.runSpeed();
RightBackWheel.runSpeed();

```

Monitörde batarya pilini görebilmek için gerekli kod

```

int sensorValue = analogRead(A0);
float voltage = sensorValue * (5.0 / 1023.00) * 3;
if (voltage < 11) {
    digitalWrite(led, HIGH);
}
else {
    digitalWrite(led, LOW);
}

```

```

    }
}
```

Sistemin hareketinin

```

void moveForward() {
    LeftFrontWheel.setSpeed(wheelSpeed);
    LeftBackWheel.setSpeed(wheelSpeed);
    RightFrontWheel.setSpeed(wheelSpeed);
    RightBackWheel.setSpeed(wheelSpeed);
}

void moveBackward() {
    LeftFrontWheel.setSpeed(-wheelSpeed);
    LeftBackWheel.setSpeed(-wheelSpeed);
    RightFrontWheel.setSpeed(-wheelSpeed);
    RightBackWheel.setSpeed(-wheelSpeed);
}

void moveSidewaysRight() {
    LeftFrontWheel.setSpeed(wheelSpeed);
    LeftBackWheel.setSpeed(-wheelSpeed);
    RightFrontWheel.setSpeed(-wheelSpeed);
    RightBackWheel.setSpeed(wheelSpeed);
}

void moveSidewaysLeft() {
    LeftFrontWheel.setSpeed(-wheelSpeed);
    LeftBackWheel.setSpeed(wheelSpeed);
    RightFrontWheel.setSpeed(wheelSpeed);
    RightBackWheel.setSpeed(-wheelSpeed);
```

```
}
```

```
void rotateLeft() {
```

```
    LeftFrontWheel.setSpeed(-wheelSpeed);
```

```
    LeftBackWheel.setSpeed(-wheelSpeed);
```

```
    RightFrontWheel.setSpeed(wheelSpeed);
```

```
    RightBackWheel.setSpeed(wheelSpeed);
```

```
}
```

```
void rotateRight() {
```

```
    LeftFrontWheel.setSpeed(wheelSpeed);
```

```
    LeftBackWheel.setSpeed(wheelSpeed);
```

```
    RightFrontWheel.setSpeed(-wheelSpeed);
```

```
    RightBackWheel.setSpeed(-wheelSpeed);
```

```
}
```

```
void moveRightForward() {
```

```
    LeftFrontWheel.setSpeed(wheelSpeed);
```

```
    LeftBackWheel.setSpeed(0);
```

```
    RightFrontWheel.setSpeed(0);
```

```
    RightBackWheel.setSpeed(wheelSpeed);
```

```
}
```

```
void moveRightBackward() {
```

```
    LeftFrontWheel.setSpeed(0);
```

```
    LeftBackWheel.setSpeed(-wheelSpeed);
```

```
    RightFrontWheel.setSpeed(-wheelSpeed);
```

```
    RightBackWheel.setSpeed(0);
```

```
}
```

```
void moveLeftForward() {
```

```
    LeftFrontWheel.setSpeed(0);
```

```

LeftBackWheel.setSpeed(wheelSpeed);
RightFrontWheel.setSpeed(wheelSpeed);
RightBackWheel.setSpeed(0);
}

void moveLeftBackward() {
    LeftFrontWheel.setSpeed(-wheelSpeed);
    LeftBackWheel.setSpeed(0);
    RightFrontWheel.setSpeed(0);
    RightBackWheel.setSpeed(-wheelSpeed);
}

void stopMoving() {
    LeftFrontWheel.setSpeed(0);
    LeftBackWheel.setSpeed(0);
    RightFrontWheel.setSpeed(0);
    RightBackWheel.setSpeed(0);
}

void runSteps() {
    while (dataIn != 13) {
        for (int i = 0; i <= index - 2; i++) {
            if (Bluetooth.available() > 0) {
                dataIn = Bluetooth.read();
                if (dataIn == 15) {
                    while (dataIn != 14) {
                        if (Bluetooth.available() > 0) {
                            dataIn = Bluetooth.read();
                            if (dataIn == 13) {

```

```

        break;

    }

}

}

}

if(dataIn > 100 & dataIn < 150) {
    speedDelay = dataIn / 10;
}

if(dataIn > 30 & dataIn < 100) {
    wheelSpeed = dataIn * 10;
    dataIn = 14;
}

LeftFrontWheel.moveTo(lfw[i]);
LeftFrontWheel.setSpeed(wheelSpeed);
LeftBackWheel.moveTo(lbw[i]);
LeftBackWheel.setSpeed(wheelSpeed);
RightFrontWheel.moveTo(rfw[i]);
RightFrontWheel.setSpeed(wheelSpeed);
RightBackWheel.moveTo(rbw[i]);
RightBackWheel.setSpeed(wheelSpeed);

while (LeftBackWheel.currentPosition() != lbw[i] & LeftFrontWheel.currentPosition()
!= lfw[i] & RightFrontWheel.currentPosition() != rfw[i] &
RightBackWheel.currentPosition() != rbw[i]) {
    LeftFrontWheel.runSpeedToPosition();
    LeftBackWheel.runSpeedToPosition();
}

```

```
    RightFrontWheel.runSpeedToPosition();

    RightBackWheel.runSpeedToPosition();

}

if (servo01SP[i] == servo01SP[i + 1]) {

}

if (servo01SP[i] > servo01SP[i + 1]) {

    for ( int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {

        servo01.write(j);

        delay(speedDelay);

    }

}

if (servo01SP[i] < servo01SP[i + 1]) {

    for ( int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {

        servo01.write(j);

        delay(speedDelay);

    }

}

if (servo02SP[i] == servo02SP[i + 1]) {

}

if (servo02SP[i] > servo02SP[i + 1]) {

    for ( int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {

        servo02.write(j);

        delay(speedDelay);

    }

}
```

```

if (servo02SP[i] < servo02SP[i + 1]) {
    for ( int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {
        servo02.write(j);
        delay(speedDelay);
    }
}

if (servo03SP[i] == servo03SP[i + 1]) {
}
if (servo03SP[i] > servo03SP[i + 1]) {
    for ( int j = servo03SP[i]; j >= servo03SP[i + 1]; j--) {
        servo03.write(j);
        delay(speedDelay);
    }
}
if (servo03SP[i] < servo03SP[i + 1]) {
    for ( int j = servo03SP[i]; j <= servo03SP[i + 1]; j++) {
        servo03.write(j);
        delay(speedDelay);
    }
}

if (servo04SP[i] == servo04SP[i + 1]) {
}
if (servo04SP[i] > servo04SP[i + 1]) {
    for ( int j = servo04SP[i]; j >= servo04SP[i + 1]; j--) {
        servo04.write(j);
    }
}

```

```
    delay(speedDelay);
}

}

if (servo04SP[i] < servo04SP[i + 1]) {
    for ( int j = servo04SP[i]; j <= servo04SP[i + 1]; j++) {
        servo04.write(j);
        delay(speedDelay);
    }
}

if (servo05SP[i] == servo05SP[i + 1]) {
}

if (servo05SP[i] > servo05SP[i + 1]) {
    for ( int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {
        servo05.write(j);
        delay(speedDelay);
    }
}

if (servo05SP[i] < servo05SP[i + 1]) {
    for ( int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {
        servo05.write(j);
        delay(speedDelay);
    }
}

if (servo06SP[i] == servo06SP[i + 1]) {
}

if (servo06SP[i] > servo06SP[i + 1]) {
```

```
for ( int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {  
    servo06.write(j);  
    delay(speedDelay);  
}  
}  
  
if (servo06SP[i] < servo06SP[i + 1]) {  
    for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {  
        servo06.write(j);  
        delay(speedDelay);  
    }  
}  
}  
}  
}
```

#### 4.7 MALİYET ANALİZİ

Sistemin elektronik kısmı için 4 adet NEMA 17 step motor, 3 adet MG996R servo motor, 3 adet SG90 mikro servo motor, DRV8825 step motor sürücü kartı, nRF24L01, HC-05 Bluetooth Modülü, 3S lipo batarya, Arduino Mega Board, kablo ve dirençler , PCB sürücü kartı dahil olmak üzere 3470 Türk Lirası tutmuştur.

ELEKTRONİK PARÇA LİSTESİ	ADET	BİRİM FİYAT	TOPLAM FİYAT
STEP MOTOR - NEMA 17	4	₺400,00	₺1.600,00
MG996R SERVO MOTOR	3	₺115,00	₺345,00
SG90 MİKRO SERVO MOTOR	3	₺50,00	₺150,00
DRV8825 Step Motor Sürücü Kart	4	₺55,00	₺220,00
nRF24L01	1	₺20,00	₺20,00
HC-05 Bluetooth Module Pinout	1	₺190,00	₺190,00
Arduino Mega Board	1	₺375,00	₺375,00
3S Lipo Batarya	1	₺450,00	₺450,00
Kablo ve Direnç	1	₺35,00	₺35,00
PCB Sürücü Kartı	1	₺85,00	₺85,00
TOPLAM		₺3.470,00	

Bunun yanı sıra mekanik kısımlar için ise 1 kg mor ABG marka PLA filament ve Yarım metrekare MDF sunta kullanılmıştır. Filamentin kg bazında birim fiyatı 350 Türk Lirası ve MDF sunta için ise 90 Türk Lirası verilmiştir. Bu fiyatlarında eklenince prototip robotun toplam fiyatı 3910 TL olarak belirlenmiştir.

## 5. Kaynakça

- A. Çetiner, H. E. (2003). LABİRENT ROBOTU TASARIMI VE UYGULAMASI. *SAU Fen Bilimleri Enstitüsü Dergisi*, 103-109.
- Aksa, O. (2021, ARALIK 7). AKSA OTOMASYON. <https://www.aksautomasyon.com/servo-motor-nedir-cesitleri-nelerdir/> adresinden alındı
- Albayrak, M. (1997). ÜÇ BOYUTLU UZAYDA HAREKET EDEBİLEN ROBOTUN BİLGİSAYAR DESTEKLİ KONTROLÜ. ANKARA, TÜRKİYE.
- Alp, O. E. (2012, NİSAN). GENEL AMAÇLI ROBOT KOL TASARIMI. İZMİR, TÜRKİYE.
- Ark., L. v. (1991). Robot Motion Planning: A Distributed. *INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH*, 628-649.
- Baluja. (1997). Dynamic relevance: vision-based focus of attention using artificial neural networks. *Artificial Intelligence*, 381-395.
- Bayar, G. (2006, NİSAN 28). ARAŞTIRMA AMAÇLI MODÜLER BİR ROBOT PLATFORMU. *TİMAK-Tasarım İmalat Analiz Kongresi*. ANKARA: ODTÜ.
- Buchi. (1996). CONTROL OF A MOBILE ROBOT. *Seventh International Symposium on Micro Machine and*, 203.
- Çakmak, O. (2016, EKİM). ADIM MOTORLARININ MİKROADIM SÜRME TEKNİĞİ İLE KONTROLÜ. *ADIM MOTORLARININ MİKROADIM SÜRME TEKNİĞİ İLE KONTROLÜ*.
- Çetin, L. (2088). DESIGN AND VISION BASED CONTROL OF A. *9 EYLÜL ÜNİVERSİTESİ DERGİSİ*.
- Huang. (2004). Alma ve Yerleştirme İşlemleri İçin Yeni Bir İki serbestlik Derecesine Sahip Paralel Robotun Kavramsal Tasarımı ve Boyutsal Sentezi. *MIT TECHNOLOGY REVIEW*.
- Kaan, K. (2006). AC SERVO MOTORLAR VE SÜRÜCÜ DEVRELERİ. İSTANBUL.
- Kollmorgen. (2020, EKİM 21). <https://www.kollmorgen.com/tr/> <https://www.kollmorgen.com/tr/tr/blogs/servo-motorlar-nasil-calisir/> adresinden alındı
- Lum, J. v. (2013). Arduino ve Wifi Temelli Çok Sensörlü Robot Tasarımı ve Denetimi. *BİLİŞİM TEKNOLOJİLERİ DERGİSİ*.
- Nihat Yılmaz, Ş. S. (2005). *MOBİL ROBOT: SUNAR*. ANKARA: GAZİ ÜNİVERSİTESİ.
- Ollero, A. (1999). Control Engineering Practice. *Artıbilim: Adana Bilim ve Teknoloji Üniversitesi Fen Bilimleri Dergisi*, 1291.
- Önal, Ö. (2011). SERİ PORTTAN KABLOSUZ AĞ İLE HABERLEŞEBİLEN KAMERALI ARAÇ KONTROLÜ. KAHRAMANMARAŞ, TÜRKİYE.

- Rehberiniz, E. (2011, HAZİRAN 09). *ELEKTRİK REHBERİNİZ*. ELEKTRİK REHBERİNİZ:  
<https://www.elektrikrehberiniz.com/elektrik-motorlari/step-motor-nedir-543/> adresinden alındı
- Robocombo. (2021, EKİM 17). *ROBOCOMBO*. ROBOCOMBO:  
<https://www.robocombo.com/blog/icerik/step-motor-nedir-ne-ise-yarar-nasil-calisir#:~:text=Sargilar%C2%B1belirli%20bir%20sirayla%20adim,step%20motorlar%C4%B1n%20%C4%CAlismaya%20prensibini%20olu%C5%9Fturur.> adresinden alındı
- Semiz, T. Y. (2018, ŞUBAT 6). *ROBOTİSTAN*. ROBOTİSTAN: <https://maker.robotistan.com/step-motor-nedir/> adresinden alındı
- Şahin, İ. (2011). KABLOSUZ KONTROL EDİLEBİLEN MOBİL ARAŞTIRMA ROBOTU. *International Computer & Instructional Technologies*.
- Şahin, R. (2021). *ŞAHİN RULMAN*. <https://www.sahinrulman.com/blog/ac-ve-dc-servo-motor-nedir-ozellikleri-ve-arasindaki-farklar> adresinden alındı
- Taşdemir, C. (2017). *AURDINO*. İSTANBUL: Dikey Eksen.
- Ünlütürk, A. (2012). Uzaktan Kontrollü Prototip Mobil Bomba İmha Robotu Tasarımı ve Gerçekleştirilmesi. *ELEKTRİK MÜHENDSİLİĞİ SEMPOZYUMU* (s. 700-703). BURSA: ELECO YAYINLARI.
- Victor, J. (1998). Robotics and Autonomous Systems. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 221.
- Yüzgeç, U. (2016). Uzaktan Kontrol Edilebilen Robot Kol Tasarımı ve Gerçeklenmesi Remote Controlled Robot Arm Design and Implementation. *International Conference on Computer Science and Engineering* (s. 457-461). TEKİRDAĞ: UBMK.