T.C.
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GERÇEK ZAMANLI İŞLETİM SİSTEMİ İLE ÇALIŞAN SUMO ROBOT

Ahmet Akif KÖSE

Danışman
Dr.Öğr.Üyesi Mehmet ALBAYRAK

YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
ISPARTA- 2019

TEZ ONAYI

Ahmet Akif KÖSE tarafından hazırlanan **"Gerçek Zamanlı İşletim Sistemi ile Çalışan Sumo Robot"** adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü **Elektronik-Bilgisayar Eğitimi Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak başarı ile savunulmuştur.

Danışman	**Dr.Öğr.Üyesi Mehmet ALBAYRAK** Isparta Uygulamalı Bilimler Üniversitesi
Jüri Üyesi	**Prof.Dr. Tuncay AYDOĞAN** Isparta Uygulamalı Bilimler Üniversitesi
Jüri Üyesi	**Doç.Dr. Veysel DEMİRER** Süleyman Demirel Üniversitesi

Enstitü Müdürü **Doç.Dr. Şule Sultan UĞUR**

TAAHHÜTNAME

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Ahmet Akif KÖSE

İÇİNDEKİLER

ÖZET

Yüksek Lisans Tezi

GERÇEK ZAMANLI İŞLETİM SİSTEMİ İLE ÇALIŞAN SUMO ROBOT

Ahmet Akif KÖSE

Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik-Bilgisayar Eğitimi Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Mehmet ALBAYRAK

Bu tez çalışmasında; sumo robot mekaniği üzerinde gerçek zamanlı işletim sistemi kütüphanesi kullanılarak robotun kontrol edilmesi sağlanmıştır. Sumo robotun gerçek zamanlı işletim sistemi komutlarını kullanarak karar vermesi ve mümkün olan en hızlı şekilde komutu uygulaması, robotun istenilen davranışları en kısa sürede karar vererek yapması ve rakiplerine karşı üstünlük sağlaması amaçlanmıştır.

Mevcut sumo robot uygulamalarında geçerli ölçü olan 20cm*20cm boyutlarına sığabilecek dört tekerli bir robot mekaniği hazırlanmıştır. Bu mekanikte kullanılan motorlar 1000d/dk üzerinde ve yüksek güç üreten olması esas alınmıştır. Bunların yanı sıra tabanda kullanılan kontrast sensörleri de tepkisi hızlı olan türden seçilmiştir. Ayrıca farklı robot yarışmalarında da rakip robotları algılamada kullanılanlara benzer olan mesafe sensörleri de kızılötesi çalışacak yapıda seçilmiştir. Kontrol kartında kullanılan mikro denetleyici/mikroişlemci seçiminde gerçek zamanlı işletim sistemi komutlarını destekleyecek ve mümkün olan en yüksek frekansta çalışabilecek işlemci tercih edilmiştir. Hızlı algılama ve hızlı karar verme sonucunda motorların devirlerinin de yüksekliği dikkate alınırsa hızlı hareket (çabuk tepki verme) sağlanmıştır.

Bu çalışmayı önceki çalışmalardan farklı yapan unsur donanımsal parçaların yüksek değerlere (hız, güç, dayanıklılık v.b.) sahip olmasının yanı sıra GZİS komutları ile yazılımsal olarak üstünlük elde edilmeye çalışılmıştır.

Anahtar Kelimeler: Sumo robot, Gerçek zamanlı işletim sistemi (GZİS), Kontrol

2019, 57 sayfa

ABSTRACT

M.Sc. Thesis

SUMO ROBOT WORKING WITH REAL-TIME OPERATING SYSTEM

Ahmet Akif KÖSE

Süleyman Demirel University
Graduate School of Applied and Natural Sciences
Department of Electronics and Computer Education

Supervisor: Asst. Prof. Dr. Mehmet ALBAYRAK

In this thesis; on the sumo robot mechanics, real-time operating system library is used to control the robot. It is aimed that Sumo robot can decide by using real-time operating system commands and execute the command as fast as possible, the robot will make the desired behaviors in the shortest time and make superiority to its competitors.

A four-wheeled robot mechanics has been prepared which can fit 20cm*20cm which is the current dimension in current sumo robot applications. The motors used in this mechanics are based on 1000rpm and have high power generating. In addition, the contrast sensors used in the base are selected from the fast response type. In addition, distance sensors, which are similar to those used to detect competing robots in different robot competitions, have been chosen to operate in infrared. In the selection of the microcontroller / microprocessor used in the control card, the processor that supports real-time operating system commands and can operate at the highest possible frequency is preferred. Rapid movement (quick response) is provided if the speed of the motors is also taken into account as a result of rapid detection and quick decision making.

What makes this study different from the previous studies is that the hardware parts have high values (speed, power, durability, etc.), and the RTOS commands have been tried to gain software advantage.

Keywords: Sumo robot, Real-time operating system (RTOS), Control

2019, 57 pages

TEŞEKKÜR

Çalışmalarımın her aşamasında beni yönlendiren, bilgi, yardım ve desteklerini esirgemeyen danışmanım Sayın Dr.Öğr.Üyesi Mehmet ALBAYRAK'a, geliştirme sürecindeki desteklerinden ötürü Samed ORHAN, Zafer BAHAR'a, her saniye desteğini hissettiğim sevgili eşim Ümran TOMBURLAK KÖSE'ye, her yorulduğum anda varlıkları ile moral tazelediğim canım kızlarım Ecenaz ve Elfin KÖSE'ye teşekkürlerimi sunarım.

1130-YL-05 No`lu Proje ile tezimi maddi olarak destekleyen Süleyman Demirel Üniversitesi Bilimsel Araştırma Projeleri Yönetim Birimi Başkanlığı'na teşekkür ederim.

Robotun imalat aşamasındaki desteklerinde dolayı Uluborlu Mesleki ve Teknik Anadolu Lisesi personellerine teşekkür ederim.

Tezimin her aşamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Ahmet Akif KÖSE
Isparta, 2019

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

A	Amper
ABD	Amerika Birleşik Devletleri
AC	Alternatif akım (Alternative current)
AREF	Analog referans (Analog reference)
Cm	Santimetre (Centimeter)
DC	Doğru akım (Direct current)
d/dk	Devir/Dakika (rpm)
EEPROM	Elektriksel olarak silinebilir programlanabilir salt okunur bellek (Electrically erasable programmable read only memory)
GND	Toprak (Ground)
g	Gram
GZÇS	Gerçek zamanlı çalışan sumo robot
GZİS	Gerçek zamanlı işletim sistemi
I2C	Bütünleşik devre (Inter-integrated circuit)
ICSP	Devre seri programlayıcı (In-circuit serial programming)
IN A	Giriş A (Input A)
IN B	Giriş B (Input B)
IR	Kızılötesi (Infrared)
KB	Kilobyte
KHz	Kilohertz
Kg	Kilogram
LED	Işık yayan diyot (Light emitting diode)
mA	Mili amper
mAh	Mili amper saat (Mili amper per hour)
MISO	Ana giriş uydu çıkış hattı (Master in slave out)
mm	Milimetre
MOSI	Ana çıkış uydu giriş hattı (Master out slave in)
MRDS	Microsoft Robotics Devoloper Studio
nm	Newtonmetre
PCB	Baskı devre kartı (Printed circuit board)
PVC	Polivinilklorür (Poli vinil clorur)
PVDF	Alüminyum sandiviç panel (Polivinilidin florid)
PWM	Sinyal genişlik modülasyonu (Pulse width modulation)
QTR	Kontrast sensörü (Reflectance sensor)
RIA	Robot Endüstri Birliği (Robot Industry Association)
RTOS	Gerçek zamanlı işletim sistemi (Real time operating system)
R.U.R.	Rossumun evrensel robotları (Rossum's universal robot)
RX	Alıcı X (Receive X)
SCK	Seri saat (Serial clock)
SCL	Şube yapılandırma dili (Substation configuration language)
SDK	Yazılım geliştirme kiti (Software development kit)
SS	Uydu seçici (Slave select)
SPI	Seri çevre arayüzü (Serial peripheral interface)
SRAM	Durağan rastegele erişim belleği (Static random access memory)
TTL	Transistör Transistör Lojik (Transistor transistor logic)
TX	Gönderici X (Transmit X)

UART	Evrensel asenkron alıcı verisi (Universal asynchronorus receiver transmitter)
USB	Evrensel seri veri yolu (Universal serial bus)

1. GİRİŞ

Robot otonom veya önceden programlanmış görevleri yerine getirebilen elektro-mekanik bir cihazdır. Robotlar doğrudan bir operatörün kontrolünde çalışabildikleri gibi bağımsız olarak bir bilgisayar programının kontrolünde de çalışabilirler.

Robotik bilimiminin temelleri çok eskilere dayanır. Ktesibius robotik biliminin babası sayılır. Bu alanda çağının ötesinde çalışmalarda bulunmuştur. Sibernetik ve robotik biliminin kurucusu olmuştur. Ktesibius dan sonra en büyük sibernetik uzmanlarından biri Ebu'l iz El Cezeri' dir. Yaşadığı zamanın çok ilerisinde robotlar yapmıştır. El Cezeri otomatik makineler tarihinden çağının doruğuna erişmiş büyük mühendis İbni Razzaz Cesari adıyla anılır.

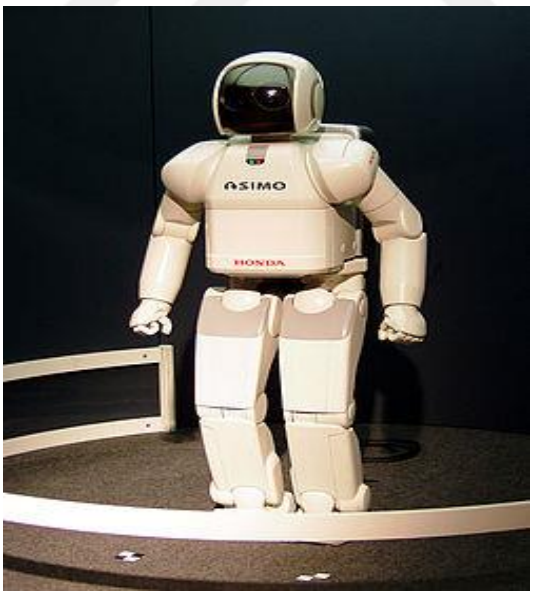
Katipoğlu (2013), robot deyince insan benzeri makineler akla gelsede robotların çok azı insana benzer. Naughton (1984), robot kelimesi ilk olarak Karel Capek'in 1920 yılında yazdığı "R.U.R. – Rossum's Universal Robots" adlı eserinde yer almıştır. Karel Capek'in R.U.R. adlı oyununda mekanik ve otonom, ama insana duygulardan yoksun yaratıklar olarak kullanılan robot daha sonra birçok bilim kurgu romanına konu olmuştur. Isaac Asimov ünlü robot serisiyle teknolojik açıdan tutarlı bir robot kavramı ortaya koymaktadır. Ayrıca, robotların amacının insan hayatını kolaylaştırmak ve insana hizmet etmek olduğunu, bir robotun kendi amaçlarını insanların amaçlarına hiçbir zaman tercih edemiyeceğini üç robot kuralı ile belirler. Asimov (1992), bu üç kural aşağıdaki gibidir.

i. Bir robot bir insana zarar vermez veya pasif kalmak sureti ile zarar görmesine izin vermez.
ii. Bir robot kendisine insanlar tarafından verilen emirlere birinci kural ile çelişmediği sürece itaat etmek zorundadır.
iii. Bir robot birinci ve ikinci kurallar ile çelişmediği sürece kendi varlığını korumak zorundadır

Albayrak vd. (2003), robot teknolojisi, insan örnek alınarak fiziksel kapasite ve duyulara sahip cihazlar üretmeyi amaçlar. Robot Teknolojisi bilgisayar destekli

kontrol sistemleri ile gelişmiş ve yapay zekâ alanındaki gelişmelerle ilerlemektedir. Geçmişten günümüze kadar robotik alanında çok çeşitli robotlar geliştirilmiştir. Bunlar değişik çalışma şekillerine ve ortamlara uygun olarak imal edilmişlerdir. Örneğin insanlar için elverişsiz ortamlarda (yüksek basınç, sıcaklık vb.) çalışacak şekilde tasarlanmış veya fabrikalarda robot kollar şeklinde tasarlanmış robotlar imalat sanayiinde verimliliğe çok büyük katkıda bulunmaktadır.

Robotlar günümüzde birçok alanda ve birçok formda karşımıza çıkmaktadırlar. Geçmişten günümüze çok fazla form değiştirmişlerdir. İlk önce endüstride kullanılan robotlar sabit olarak düşünülmüş ve belli hareketleri tekrar edecek şekilde tasarlanmışlardır. Bugün robotların gerçekleştirdikleri görevleri yerine getirirken hareketli olmaları hızlı olmaları ve bunun yanında hata yapmamaları amaçlanmaktadır. Günümüz robotlarının hareketlerinin çeşitliliği bakımından insansı ilk robot Şekil 1.1.'de örnek olarak ilk modern robot olarak da tanımlayabileceğimiz Hondanın üretmiş olduğu Asimo'dur.



Şekil 1.1. İlk modern insansı robot Honda Asimo

Asimo insan hareketlerine çok benzer hareket etmektedir. Örneğin merdiven çıkabilir, el sallayabilir veya tek ayağı üzerinde durabilmektedir.

Robotun bir başka tanımıda; Amerikan robot endüstrisi birliği (RIA "The American Robot Industry Association") tarafından, programlanmış değişik hareketlerle, malzeme, parça, takım veya özel araçlar taşıyabilen, yeniden programlanabilme özelliğine sahip, çok amaçlı bir manipülatör olarak tanımlanmaktadır. Üretim açısından daha basit bir tanım vermek gerekir ise, programlanabilen, otomatik olarak hareketi tekrar edebilen ve endüstriyel ortamda iş yapabilen makina bir endüstriyel robottur.

Yiğiter (2010), çalışmasında robotların endüstriyel kullanım alanlarındaki gelişimindeki dönüm noktalarını aşağıdaki şekilde özetlemiştir.

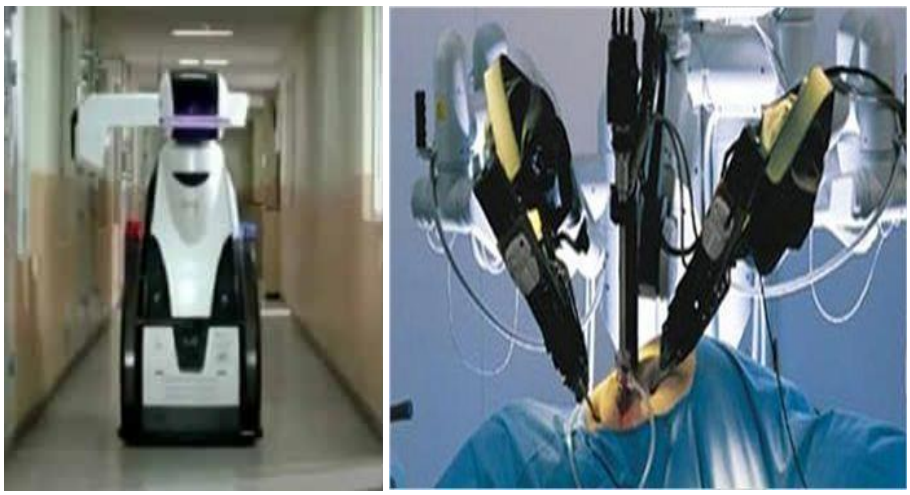
- 1801: Programlanabilir dokuma tezgâhı,
- 1830: Otomatik çıkrık,
- 1893: Ayakları ile yürüyen araç,
- 1945: Radyoaktif malzemeyi tutmak için teleoperatör,
- 1953: Servo denetimli freze tezgâhı,
- 1954: İlk programlanabilir endüstriyel robot,
- 1959: İlk ticari robot,
- 1974: Mini bilgisayar kullanan ilk ticari endüstriyel robot.

Son onbeş yılda verimliliği arttırmak için yoğun çalışmalar yapılmış ve değişik otomasyon yöntemleri geliştirilmiştir. Bu yöntemlerden birisi de programlanabilir otomasyon yöntemidir. Günümüzde robotlar bu tip uygulamalar için kullanılmaktadır. Örnek olarak kaynak, boyama veya döküm işleri verilebilir. Bu tür bir otomasyonda robotlann bilgisayar programlan değiştirilerek, değişik işleri yapabilmeleri sağlanabilmektedir.

Robotların imalatta kullanılması ile verimlilik önemli ölçüde artmaktadır. Örneğin, bir kaynak işlemi, robot kullanılarak İkİ veya üç kat daha verimli olmaktadır. Kaynak hatalannın azalmış olması da gözönüne alınırsa verimliliğin daha da artacağı görülür. Bir başka konu da bir robotun, kaliteyi düşürmeden çok sayıda parçayı, devamlı çalışarak aynı kalitede imal edebilmesidir. Kaynağın robot tarafından yapılmasının getirdiği bir diğer avantaj da çalışan kişinin kaynak

gazlarına ve zararlı ışınlara maruz kalmamasıdır. Robot kullanarak insanlann ulaşamayacağı yerlerde kaynak yapmak da mümkün olmaktadır.

Günümüzde robotların kullanım alanları çok fazla çeşitlilik kazanmıştır. Şekil 1.2.' de gardiyan ve cerrah robotları görülmektedir. ABD'deki hapishanelerde, gardiyan robotlar saatte dokuz kilometre hızla dolaşarak video ve radar görüntüleriyle 360 derecelik bir açı ile cezaevini kontrol ediyorlar. Bu durumda iş gücünden ve maliyetten tasarruf edilmektedir. Normal de aynı işi yapmak için birden çok insana ihtiyaç duyulmaktadır. Buna benzer örneklerden biri de sigorta şirketleri hastanelerin sağlık harcamalarından şikâyet etmekte ve azaltılması yönünde tedbirler üzerinde çalışmaktadırlar. Bunun için yapılan çalışmalardan biri de ameliyathanedeki işlemlerin otomasyona bağlanmasıdır. Gelecekte rutin prosedürler için robotlar kullanılacak ve hem ameliyat maliyetleri düşecek hemde cerrahların yükü azalacaktır. Ameliyatlarda insanın zayıf yönlerinden kaynaklı risklerde ortadan kalkmış olacaktır. Robotların kullanım alanları ile ilgili örnekleri çoğaltmak mümkün. Bugün robot teknolojisinin gelişim hızı baş döndüren bir hızla ilerlemekte ve gelişmektedir. Bu sebeple robotların yaptıkları işlerde basit düzeyde de olsa insiyatif almaları en azından yapılan işin işlem basamakları arasında öncelik sıralaması konusunda karar verebilmeleri daha da önem kazanmaktadır.



Şekil 1.2. Gardiyan ve cerrah robotlar

Boston Dynamics (2019)’ in üretmiş olduğu Atlas isimli robot, robotların insanların hareketlerini taklit edebilme yeteklerinde geldikleri son noktayı gösterme açısından iyi bir örnektir. Atlas, gelişmiş insansı robotlar serisinin sonuncusudur. Atlas'ın kontrol sistemi, tüm vücut hareketlerini elde etmek için kolların, gövdenin ve bacakların hareketlerini koordine eder, bu sayede robot hareket kabiliyetini ve çalışma alanını büyük ölçüde genişletir. Atlas'ın görevleri yerine getirirken dengede durma yeteneği, sadece yere basan ayakları sayesinde küçük bir alan kaplarken büyük hacimlerde çalışmasına izin verir.

Şekil 1.3.’ de de görüldüğü üzere Atlas donanımı üretilirken ağırlıktan ve yerden tasarruf etmek için 3D yazıcılardan faydalanılmıştır, bu da parçaların hafifi olması sebebiyle ağırlık avantajına ve çok büyük yeteneklere sahip olağanüstü bir kompakt robot üretilmesini sağlamıştır. Çift görüntüleme teknolojisi, mesafe algılama ve diğer sensörler Atlas'a ortamındaki nesneleri değiştirme ve engebeli arazide hareket etme yeteneği kazandırmıştır. Atlas, sarsıldığında veya bastırıldığında dengesini korumakta ve devrilirse bile ayağa yardım almadan kalkabilmektedir.

Şekil 1.3. Boston Dynamics’in robotu Atlas

Robotik alanındaki çalışmalardaki örneklerden de gördüğümüz üzere

- Endüstriyel robotik
- Operasyonel robotik
- Tıp ve sağlıkta robotik
- Sibernetik
- Oyuncak robotlar
- Hobi amaçlı robotlar olarak kategorize edilmiştir.

Bu tez çalışmasında hobi amaçlı robotlar kategorisine dâhil olan sumo robot konusunda çalışılmıştır.

Albayrak ve Yüksel (2007), insanlar hobi ve oyun amaçlı küçük ölçekte robotlar yapmaktadır. Bunlardan birisi çalışma konusu olan sumo robotlardır. Sumo robotlar dünya çapında çok ilgi görmektedir ve sumo robotlarla ilgili ulusal ve uluslarası ödüllü turnuvalar düzenlenmektedir. Bu turnuvalar 1980' li yılların sonunda ortaya çıkmıştır özellikle Amerika ve Kanada'da büyük ilgi görmüştür. Sumo robot orijinal olarak 1980 lerin sonunda Japonya da Fuji yazılım şirketinin başkanı Hiroshi Nozawa tarafından icat edilmiştir. İlk gösteri oyunu Ağustos 1989 da ilk resmi turnuvada 1990 yılında gerçekleştirilmiştir.

Sumo robotları rakiplerine karşı başarılı kılan mekanik kısımlarının güçlü ve sağlam oluşundan ziyade sensörlerin algılma hızının yüksek olması ve işletim sisteminin kararlı çalışmasıdır. Bu çalışmada robotun işletim sistemi gerçek zamanlı işletim sistemi seçilmiştir. Gerçek zamanlı çalışan sumo robot adından da anlaşılacağı üzere gerçek zamanlı bir işletim sistemine sahiptir. Bu diğer sumo robotlara nazaran gerçek zamanlı çalışan sumo robotu ayrıcalıklı kılan en önemli özellikleriden birisidir. Bu özellik sayesinde yerine getireceği görevleri daha hızlı yapabilmekte ve görevlerin öncelik sırasını belirleyebilmektedir.

Bu çalışma ile gerçek zamanlı işletim sistemi ile çalışan sumo robot ile gerçek zamanlı işletim sisitemi ile çalışmayan sumo robot arasındaki farkları deneysel olarak ortaya koymak amaçlanmıştır.

2. KAYNAK ÖZETLERİ

MRDS (2013), Microsoft Robotics Developer Studio web tabanlı ve otonom uygulama senaryolarının her ikisini de desteklemektedir. Uzaktan bağlama senaryosu, bilgisayardan robota seri port, bluetooth veya 802.11 teknolojileri ile haberleşmeyi sunar. Program ayrıca Microsoft Windows'la çalışan, bilgisayar tabanlı robotların üzerinde yerel çalışabilir. Bu yapı tamamen otonom çalışmayı desteklemektedir. Seung Han Kim ve Jae Wook Jeon çalışmalarında LEGO Mindstorms NXT2' i hem LSOFT programı hem de Microsoft Robotics Developer Studio'nun Visual Programming Language ile programlanmasını sunmuştur.

C# Web Servisleri (2013), MRDS.NET Framework üzerinde çalışır dolayısıyla robot uygulamaları Microsoft Visual Studio da geliştirilmek üzere, .NET Framework' un desteklediği herhangi bir dil ile çalışmak mümkündür. MRDS içindeki birçok örnek uygulamaları ve yardımcı örnekleri C#, Visual Basic.NET ya da C++ yazılmış olarak bulanabilir. MRDS' de geliştirme için kullanılan dillerin arasında Python script dili de bulunur. Python gerçekten etkili bir dildir yorumlanabilir derlenebilir bir dildir ve açık kaynak lisansla desteklenen object-oriented bir dil olup genişletilebilirdir. Microsoft Python dilinin Iron Python versiyonun .NET platformu için çalışan bir versiyonunu açık kaynak olarak sunmaktadır. MRDS, VPL adında grafiksel bir programlama dili sunar, VPL (Visual Programming Languge)'in anlamı görsel programlama dilidir ve tecrübesiz programcı ya da öğrenciler robotik uygulamalar geliştirmek istediklerinde bu dilin sunduğu basit sürükle bırak teknikleriyle uygulamaları için görsel bir ara yüzden faydalanabilir. Gerçekten VPL çok daha tecrübeli programcılar için de çok kullanışlıdır ve hızlı bir biçimde robotik uygulamalar geliştirmeye olanak verir.

Morgan (2008), Moreno projelerinde Microsoft Robotics Developer Studio yazılımı kullanarak otonom hareket edebilen robot yazılımı geliştirme hakkında bir proje gerçekleştirmişlerdir.

Kim (2009) çalışmasında, geliştirilen uygulamalar, Microsoft Robotics Developer Studio ile uygulama geliştirme imkânı sağlayan donanımlarda

gerçekleştirilmiştir. Microsoft Robotics Developer Studio ile programlanabilen LEGO MINDSTORMS NXT, Traxter+e-Box+Serializer, KHR, Irobot gibi donanımlar Microsoft firmasının geliştirdiği servisler ile çalışan modellerdir. Bu modellerden Traxter+e-box+Serializer kiti sunulmaktadır. Microsoft Robotics Developer Studio' nun desteklemediği donanımlarla çalışmak gerektiğinde servis programlama ve config yazma işlemlerini yazılımcıların kendileri yapmaları gerekmektedir.

Johns ve Taylor (2008), MRDS, VPL isimli görsel kod üretim aracı içerir. VPL hem amatör hem de tecrübeli programcılar için uygundur. Kullanıcı program yapmak için, ara yüzdeki servisleri veya aktiviteleri temsil eden, sürükle bırak bloklarını kullanır. VPL iki çeşit aktivite sunar. Birincisi; veri girişi, hesaplamalar ve değişkenleri içeren basit aktiviteler, ikincisi ise MRDS ile gelen yerleşik servisleri içeren runtime servislerdir.

Ulukuş (2010), bir makineye robot diyebilmek için en önemli şart karar verme mekanizmasının olmasıdır. Dış dünyadan gelen uyarıları algılamalı, algıladığı her şeyi kural tabanında yorumlayabilmelidir. Sonrasında yorumladığı sorgulara göre işlemler gerçekleştirebilmelidir. Toplayacak olursak bir robot yapısı 3 ana bölümden oluşur.

1. Çevre hakkında gerçek-zamanlı bilgi elde etmek için kullanılan alıcılar
2. Karar vermeyi ve kontrolü sağlayan kural tabanı
3. Verilen kararların uygulamasını sağlayan motorlar ve hareket sistemleri.

Ulukuş (2010), bugün robotlar ev işlerinde bile insanlara yardımcı olabiliyorlar. Bunun iyi bir örneği Arok adlı robot verilebilir. Yerleri süpürmek, köpeği gezdirmek, ağır eşyaları kaldırmak gibi 36 işlev gerçekleştirebiliyor. Hero adlı robot çocuklarla oyun oynayabiliyor, bekçilik yapabiliyor. Robotlar fiziksel engellilere de yardımcı oluyor. Palo Alto'da geliştirilen ve sesle komut verilen hizmetçi robotun yapabildiklerinden bazıları yemek hazırlamak, servis yapmak, masadaki dosyaları getirmek, çizim yapmak ve kitap sayfası çevirmek" olarak verilmiştir.

Zurawski (2007), “Embedded Systems Handbook, Second Edition: Embedded Systems Desing and Verification” isimli kitapta ağ ortamındaki gömülü sistemlerden, gömülü sistemlerin haberleşmesinden ve güvenlik protokollerinden bahsedilmektedir. Ethernet veya bluetooth gibi farklı bağlantı türleri incelenmiştir.

Zurawski (2007), “Embedded Systems Handbook, Second Edition: Embedded Systems Desing and Verification” isimli kitapta gömülü işletim sistemlerini tasarımı ve kontrolü açıklanmıştır. Farklı işletim sistemlerinde tasarımlar yapılmış, yapılan tasarımlar ayrıntılı örneklerle gösterilmiştir.

Qian vd. (2009), “Embedded Software Development with C” isimli kitapta gömülü sistemlerin yazılımsal olarak programlanmasını açıklanmış ve örneklendirilmiştir.

Lima vd. (2013), Alan yönteminin bir uygulaması olarak, hareketli hedef ve statik engel dâhil olmak üzere dinamik ortamlarda özerk sumo robot sınıfı navigasyonu sunulmuştur. Kullanılan yöntem, mümkün olan en kısa sürede hedefinize ulaşmak için robotun aldığı hızı ve yönünü planlamayı amaçlamaktadır. Tasarım hızı, robot ve hedef arasındaki bağıl hızlardan ve aralarındaki bağıl konumlardan belirlenir. Bu çalışma, rakibi arenadan dışarı çıkarmak için tasarlanan bir sumo robot sınıfının hızını ve konumunu belirlemek için bir karma yöntem kullanmaktadır. Bu hibrit yöntem, bulanık mantığı olan potansiyel alanı uygular ve Mamdani modelini kullanarak robot tarafından takip edilecek hız ve yönü belirlemek için gerekli değişkenleri elde eder. Hedefi bulmak için gereken süreyi belirlemek için MATLAB'da iki simülasyon yapılmıştır. Elde edilen sonuçlar, hibrid yöntemin statik engelli herhangi bir sabit veya dinamik ortamdaki yerel asgari sorunun üstesinden geldiğini göstermektedir.

Chojnowski vd. (2018), MegaSumo robotlarında hareket modellemesi, ivmelerin genellikle 32g standart ivmeölçer aralığını aştığı darbeler nedeniyle zordur. Robotlar nispeten küçük alanlarda yüksek hızlarda hareket eder. Modeli mümkün olan en yüksek frekansta güncellenmeye zorlar. MegaSumo

robotlarında rota planlama, maç kazanma şansını artırabilecek farklı başlangıç stratejileri oluşturmaya izin verir. Her yarışta Dohyo üretimindeki farklılıklar nedeniyle hareketin sadece önceden kodlanmış parametrelerinin kullanılmasının sağlanması imkânsızdır. Makale, hareket modellemeye üç farklı yaklaşım sunmaktadır. İlk olarak, her bir tekerleğin kat ettiği mesafeyi belirlemek. İkinci modelleme, robotun doğrusal hızını ve açısal hızını belirlemek için 1 eksenli ivmeölçer ve 1 eksenli jiroskop kullanır. Sonuncusu, 3 eksenli jiroskop ve 3 eksenli ivmeölçer kullanan 3B alanda modeldir. Her modelin avantajları ve dezavantajları bu makalede sunulmuştur.

Yiğiter (2010), Mobil robot tasarımı, disiplinler arası çalışma gerektiren en kapsamlı çalışma alanlarından birisidir. Mobil robotlar, sabit bir noktada çalışmanın aksine, sürekli yer değiştiren ve bu sırada kontrol edilmesi gereken mekanik bir ana yapıyı içerir. Robotun sürekli hareket etmesi, değişen ve birçok belirsizlikle dolu bir ortamda bulunması ve dolayısı ile denetlenebilmesinin zorlaş ması anlamına gelmektedir.

3. MATERYAL VE METOD

Günümüzde Sumo robotlar Japonya'da doğmuş olasalar bile, birçok ülkede kabul görmüş ve standart ölçüler dâhilinde çok çeşitli şekillerde yapılmışlardır. Burada sumo robotların tarihini çeşitlerini ve bir sumo robotu oluşturan bileşenlerin tanıtımı yapılmıştır.

3.1. Sumo Robotun Tanımı ve Çeşitleri

Sumo robotlar adından da anlaşılacağı gibi Japon sumo güreşlerin robotiğe uyarlanarak meydana gelmiştir. Sumo güreşlerinde güreşçiler birbirlerini ringin dışına çıkarmak için mücadele ederler. Sumo robot yarışmalarında da birtakım kurallar vardır fakat temelde robotlar birbirlerinin özel olarak hazırlanmış pistin dışına iterek çıkaran robot galip gelir. Şekil 3.1.' de güreş yapan sumo robotlardan bir örnek görülmektedir.

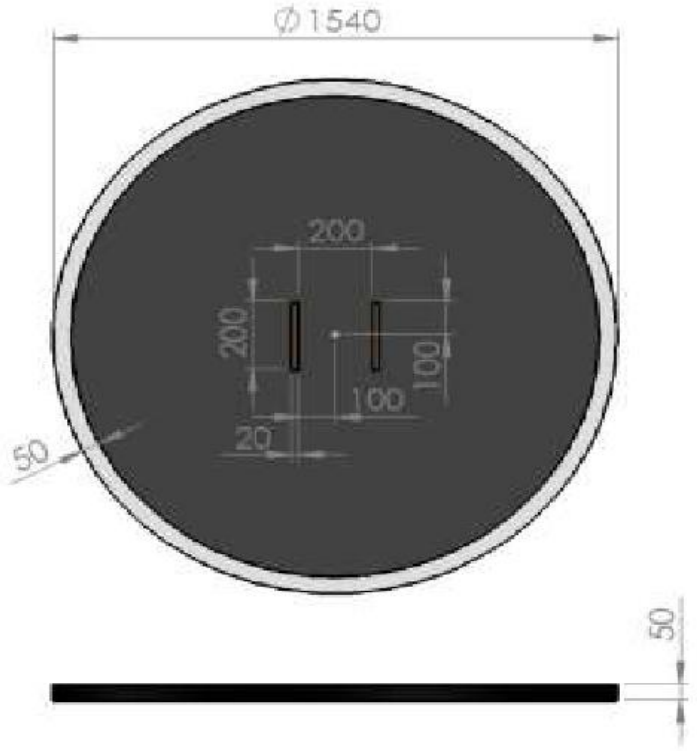
Şekil 3.1. Sumo güreşi yapan robotlar

Dohyo, sumo robot turnuvalarının üzerinde yapıldıgı, içi siyah renkte, dışında beyaz serit şeklinde çizgi bulunan halka seklinde özel olarak hazırlanan ringin adıdır.

Çizelge 3.1. Sumo robot kategorileri

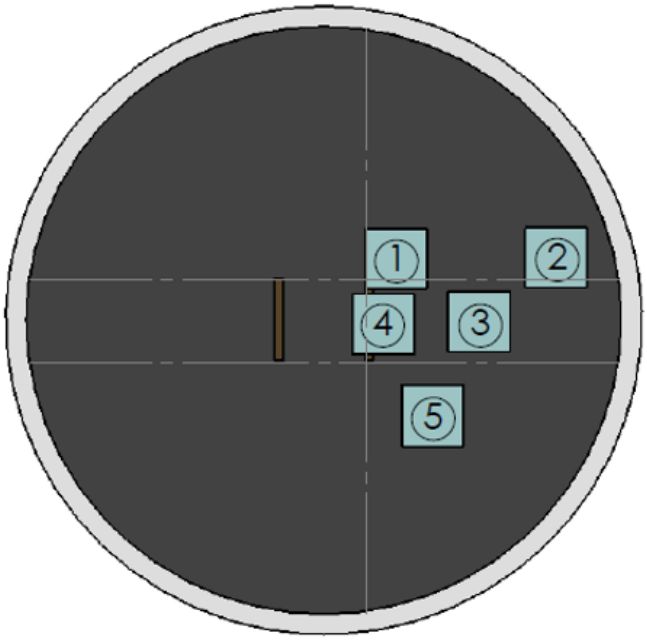
Sınıf	Yükseklik (cm)	En (cm)	Boy (cm)	Ağırlık (g)
Ağır Japon Sumo	Sınırsız	30	30	10.000
Sumo	Sınırsız	20	20	3.000
Mini Sumo	Sınırsız	10	10	500
Micro Sumo	5	5	5	100
Nano Sumo	2,5	2,5	2,5	25

Şekil 3.2.' de görüldüğü gibi Dohyo özellikleri şöyle olmalıdır. Dohyo, yerden 5cm yükseklikte, daire seklinde, dıstaki 5cm kalınlıgındaki beyaz çizgi dâhil olmak üzere 154cm çapında olmalıdır. Robotların başlama yerleri arası 20cm 'dir. Bu yerler 2cm kalınlıgında kahverengi çizgilerle işaretlenmelidir. Dohyo 'nun dısındaki 100cm 'lik mesafe boş ve beyaz dısında bir renk olmalıdır. Bu 100cm 'lik mesafe kurallara aykırı olmamak şartıyla herhangi bir maddeden yapılmış veya herhangi bir şekilde olabilir.



Şekil 3.2. Dohyo ölçüleri

Şekil 3.3.’ de yarışmalarda kullanılan doğru ve yanlış sumo robotların yerleşimleri gösterilmektedir.



Şekil 3.3. Dohyo üzerine sumo robot yerleşimi (1, 2, 3 doğru 4, 5 yanlış)

Çizelge 3.2’ de farklı sumo robot yarışma kategorilerine ait dohyo ebatları ve kullanılabilecek robotların teknik özellikleri gösterilmektedir.

Çizelge 3.2. Sumo robot dohyo özellikleri

Sumo Robot Kategorisi		Dohyo Ebatları (cm)					
Sumo Cinsi	Sumo Ağırlığı (g)	Çap (cm)	Beyaz Kenar (cm)	Yükseklik (cm)	Başlama Çizgileri Aralığı (cm)	Başlama Çizgileri Boyu (cm)	Başlama Çizgileri Kalınlığı (cm)
Ağır Japon Sumo	10.000	227	10	10	30	30	2
Sumo	3.000	154	5	5	20	20	2
Mini Sumo	500	77	2,5	2,5	10	10	1
Micro Sumo	100	38,5	1,25	1,25	5	5	0,5
Nano Sumo	25	19,25	0,625	0,625	2,5	2,5	0,25

3.2. GZÇS Sumo Robotu Oluşturan Bileşenler

Yapılan çalışmada tasarımı ve imalatı bakımından gerçekleştirilen robot bir sumo robot olmakla birlikte, üzerinde çalışan işletim sistemi bakımından gerçek zamanlı işletim sistemi (GZÇS) tercih edilmiştir. GZÇS' nin tercih edilme sebepleri altta yer alan kısımlarda açıklanmıştır.

GZÇS Sumo robotu oluşturan bileşenleri üç ana başlık altında ele alınmıştır. Bunlar mekanik, elektronik ve yazılım bölümleridir. Bir sumo robotun yapılış amacı rakibini dohyonun dışına itmek olduğudan mekaniğin (gövde) sağlam ve dayanıklı olması önemlidir. Mekanik aksamda bir diğer dikkat edilmesi gereken motorlar ve tekerleklerdir. Motorların rakibine üstünlük sağlayacak şekilde yüksek devirli ve yüksek torka sahip seçilmelidir. Elektronik devre elemanları kontrol kartı, sensörler ve bataryayı kapsamaktadır. Sumo robotu oluşturan bileşenlerden sonuncusu da yazılımdır. Yazılımda gerçek zamanlı işletim sistemidir.

3.3. GZÇS Robotun Mekanik Bileşenleri

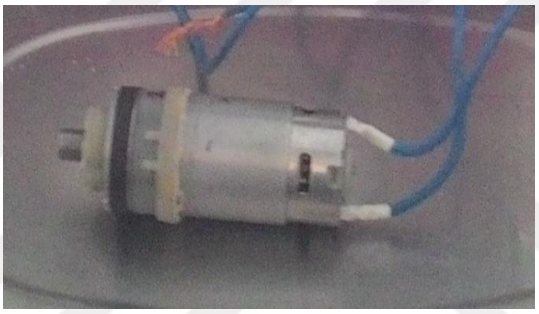
Gerçek zamanlı işletim sistemine sahip sumo robot temelde sumo robot yapısına sahiptir. Fakat imal edilirken mekanik tasarımın yapılmasında, parçaların seçiminde en iyi parçaların seçilmesine dikkat edilmiştir. Bu bölümde tasarım sürecinde yapılan işlemler ve GZÇS' nin bileşenleri hakkında bilgi verilmiştir.

3.3.1. Mekanik tasarım

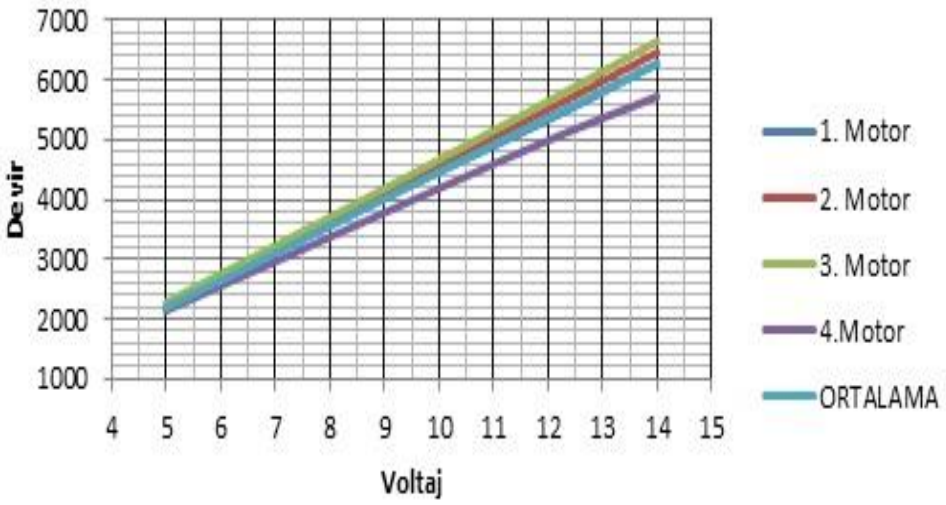
GZÇS robotun mekanik tasarımı gerçekleştirilirken öncelikle motorların seçimiyle başlanılmıştır. Motorları belirlenmesinden sonra motorlara uygun tekerleklerin üretilmesi ile güç aktarım sağlayacak bileşenleri tamamlamıştır. Mekanik tasarımda son olarak seçilen motorlar ve üretilen tekerleklere uygun gövdenin tasarımı gerçekleştirilmiştir.

3.3.2. Motorlar ve seçimi

GZÇS robot iki adet sağda ve iki adet solda olmak üzere dört adet motora sahiptir. Şekil 3.4.' de görülen motorlar 2500d/dk' nın üzerinde devire sahip olması sebebiyle ilk etapta tercih edilmiştir. Torkları ise 50nm dir. Motorlar boşta çalışır iken 7500d/dk ile dönmektedirler bu hız kontrol kontrol kartının motorları kararlı bir şekilde kontrol edebilmesi için uygun değildir. Bu sebeple motorların istediğimiz hız değerinde çalışmalarını sağlamak için millerine akuple redüktör kullanılmış ve devirleri 1/2 oranında azaltılmıştır. Motorların devirleri farklı voltaj değerlerinde turmetre ile ölçülmüş ve Şekil 3.5.' de yer alan grafik çıkarılmıştır.

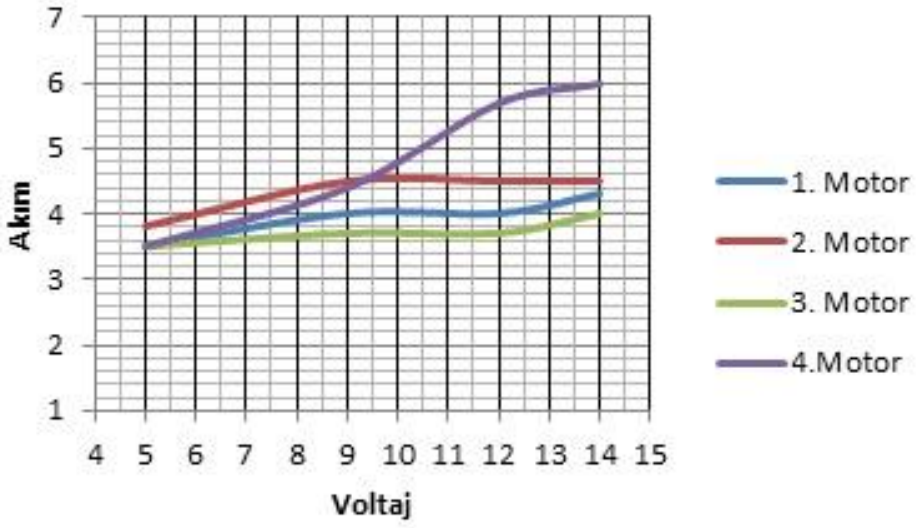


Şekil 3.4. İlk seçilen DC motor ve redüktörü



Şekil 3.5. Motorlara ait devir grafikleri

Motorların devirlerini ½ oranında azaltılması için redüktör dişlilerinin değiştirlmesi gerekmiştir. Bunun için redüktörlerin iki katmandan oluşan dişli grubu bir katmanı sökülerek kalan dişliler yerlerine geri yerleştirlmiş içlerine yağ konularak redüktör kapağı kapatılmış ve redüktör motorun mil çıkışının olduğu tarafa sabitlenmiştir. Redüktörlerin orijinal dişli gurubunun yapısı değiştirldiği için dört motorun farklı voltaj değerlerindeki çektikleri akım değrleri farklılık göstermiştir. Motorların farklı voltaj değerlerinde çalıştırılırken çektikleri akım değerleride ölçülmüştür. Bunlar Şekil 3.6.' da grafik olarak gösterilmiştir. Redüktörde yapılan değişikler motorların aynı devirde çalışmalarını olumsuz etkilemiş Şekil 3.5.' de görüleceği üzere motorlar farklı devirlerde çalışmışlardır. Bu durum istenilmeyen bir durumdur.



Şekil 3.6. Motorların Akım – Gerilim grafiği

Bu sebeple motorlar değiştirilip Şekil 3.7.'de yer alan motor ve redüktör kombinasyonuna geçilmiştir. Bu motor ve redüktörler ile istenilen değerlere ulaşılmıştır.

Şekil 3.7. Son seçilen DC motor ve redüktörü

3.3.3. Tekerlekler ve üretilmesi

Tekerlekler seçilirken mukavemetinin yüksek olması ve ağırlık bakımından hafif olması göz önünüde bulundurulmuştur. Motor ve redüktörlere uygun olacak şekilde yeniden tasarlanması ve üretilmesi gerekmiştir. Tekerlekler için iki tip malzeme kullanılmıştır ilk olarak PVC (Polivinilklorür) malzemden 50mm çapında 50mm genişliğinde kesilerek tornada işlenmiş ve son şekli verilmiştir. Şekil 3.8.' de görülebilir.

Şekil 3.8. PVC (Polivinilklorür) tekerleklerin tornada işlenmesi.

Bir diğer malzeme olarak 60mm çapında ve 4mm et kalınlığı olan metal boruyu 50mm genişliğinde kesilerek ucu kapatılmıştır. Şekil 3.9.' da görüldüğü üzere tornaya bağlanarak merkezlenmiş ve dıştan freze ile 2 mm kadar silinmiştir. Bu işlem dört teker içinde uygulanmıştır.

Şekil 3.9. Metal tekerlerin torna ile işlenmesi

Metal tekerlekler oluşturulduktan sonra motor millerine bağlayabilmek için merkezlerine elektrot kaynağı ile somun kaynatılmıştır. Motorların yüksek devir ve yüksek torka sahip olmaları sebebi ile PVC (Polivinilklorür) tekerleklerin motor millerine bağlandıkları noktalarda bağlantı sorunları ile karşılaşılmış, metal tekerleklerde ağırlık proplemi başta olmak üzere motorlar değiştiği içinde uyumsuzluk proplemi orataya çıkmıştır. Şekil 3.10.' da tornada imal edilen metal tekerlekler görülmektedir.

Şekil 3.10. Metal teker ağırlıkları

İlk olarak test edilen motorlardan ve redüktörlerden istenen değerler elde edilememiş ve kararlı çalışması sağlanamdığı için motorlar değiştirimiş üretilen tekerlerin kullanılmasından vazgeçilmiştir. Bunların yerine Şekil 3.11.' de yer alan alüminyum tekerler seçilmiş ve silikon kaplamalar ile kaplanmıştır.

Şekil 3.11. Alüminyum tekerin silikon kaplanması

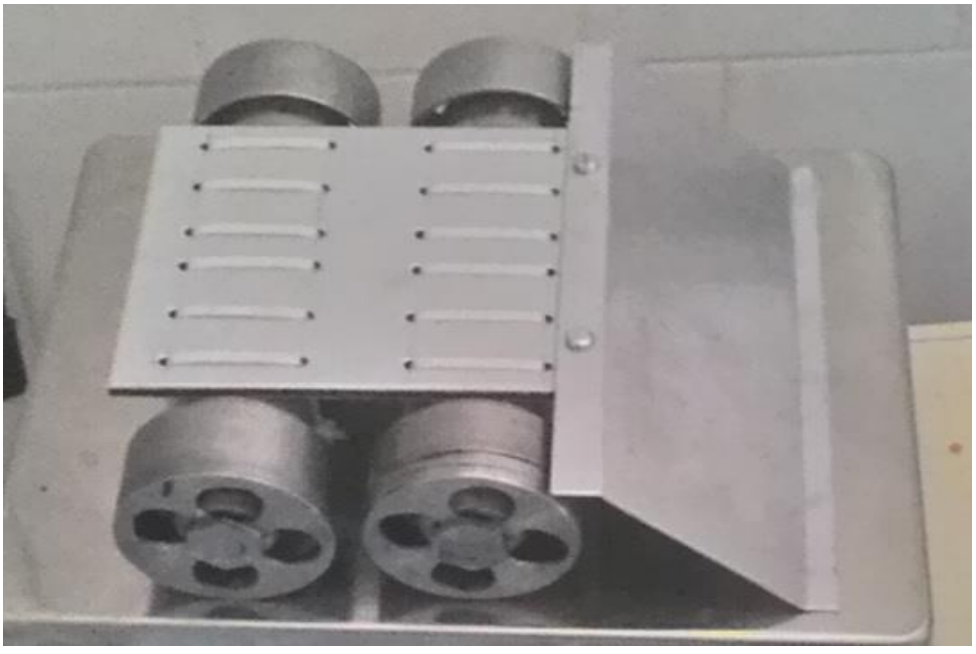
Teker alüminyum malzemeden üretilmiş iç Delik çapı 3mm'dir. Tekerlek motor miline setuskul vidasıyla mekanik olarak bağlanmıştır. Dış çap: 33mm, genişlik: 21mm dir. Ağırlık: 44gr (lastik: 8gr, jant: 36gr) dır. Tekerin dışında lastik malzemesi shore 20 sertliğinde silikon ve poliüretan katkılıdır. Değerlerdende anlışılacağı gibi bu tekerler daha sağlam ve hafiftir.

3.3.4. Gövdenin üretilmesi

GZÇS Robotun yarış amacı rakibini dohyo dışına itip alt etmek olduğundan gövde kısmının sağlam ve dayanıklı olması önemlidir. Yapım aşamısına geçmeden yarış kategorisi kurallarına uygun olarak 20x20cm boyutlarında ve 3kg ağırlığı geçmeyecek şekilde tasarlanmalıdır. Kurallarda verilen ağırlık üst sınırına uygun olması için alminyum, çelik, pleksiglas gibi malzemeler denenmiş ve hafif ve dayanıklı malzeme olan aliminyum ve PVDF sandwich aliminyum panel tercih edilmiştir. Şekil 3.12.' de görüldüğü üzere sandwich panel, her iki yüzü boyalı galvaniz sac veya boyalı gofrajlı alüminyum levhalar arasına muhtelif kalınlıkta ve 38-40 kg/m3 poliüretan enjekte edilerek üretilen bir sanayi ürünüdür.

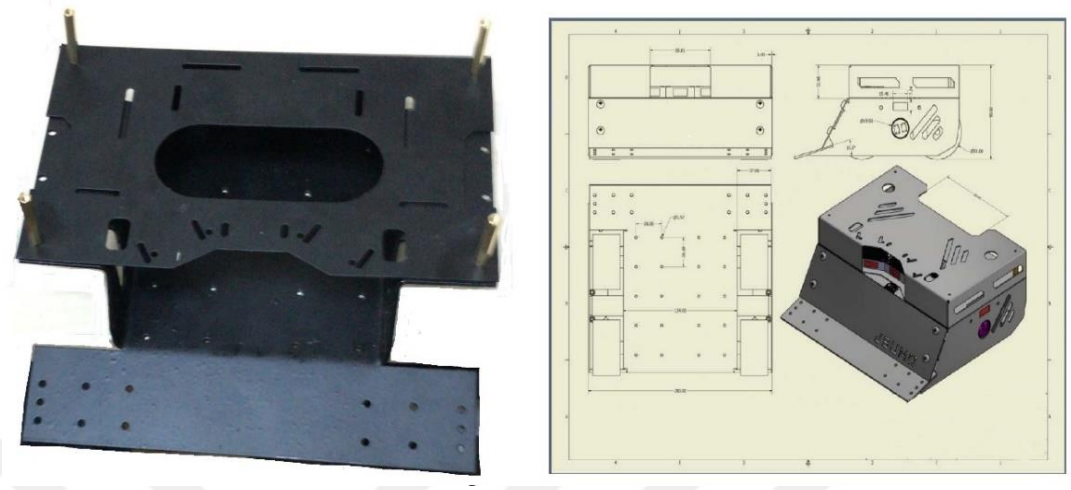
Şekil 3.12. PVDF sandwich alüminyum panel

Robotun taşıyıcı şasesi için PVDF sandwich alümiyum panel kullanılmıştır. Bu ana şaseye motorlar, ön rampa, sensörlerin bağlandığı alümiyum tabla bağlanmıştır. Ön tarafta kullanılan rakip altına girip aşırtmak için eğimli şekilde yerleştirilmiş rampa ve sensörlerin bağlı olduğu tabla için 1,5mm kalınlığında alümniyum levha kullanılmıştır. Şekil 3.13.' de motorların ve ön rampanın ana şaseye bağlantısı görülmektedir.



Şekil 3.13. Motorların ve rampanın şaseye bağlantısı.

Denemeler esnasında bu şasede bükülmeler olmuş ve yapısal olarak bozulmuştur. Bunun yerine Şekil 3.14.' de görülen şase ye geçilmiştir.



Şekil 3.14. Robot şasesi

Ana gövde 2mm demir plakadan yapılmıştır. Robotun kaplama malzemesi 1mm sacdan lazer kesim yönetemi ile yapılmıştır. Yeni tercih edilen motor ve redüktörler ile tam uyumlu olup sağlam bir yapıya sahiptir.

3.4. GZÇS Robotun Elektronik Bileşenleri

GZÇS Robotun elektronik bileşenleri bataryalar, sensörler, Motor sürücü devreleri ve kontrol kartından oluşmaktadır. Bu bölümde bataryanın seçilmesinden ve seçilen bataryanın özellikleri, sensörlerin teknik özellikleri robota bağlanırken yapılan elektronik düzenlemeleri içermektedir. Ayrıca robot un diğer elektronik bileşenlerinden olan kontrol kartı ve motor sürücü devresi tanıtılmış bu kartları çalışmada kullanırken dikkat edilmesi gereken hususlardan söz edilmiştir.

3.4.1. Bataryalar ve seçimi

Bataryalar robotun ihtiyacı olan gücü yeterli süre içersinde sağlarlar. Bu şartı yerine getirirken robotun sağlaması gereken ağırlık ve mekanik ölçüler içerisinde olmasına dikkat edilmiştir. Bataryalar küçük boyutta ve yüksek akım sağlamaları için litium–polimer (LiPo) batarya tercih edilmiştir. Bataryalar kapasite olarak 2600mAh ve 11.1 Volt seçilmiştir. Batarya şekil 3.15.' de görülebilir.

Batarya seçiminde LiPo türü tercih edilmesinin bir sebebide anlık çok yüksek deşarj akımı (dört motorun yüklü durumda çektiği 4x12A=48A) sağlayabilmesidir. Bu durum LiMh şarjlı piller ile sağlanamamasıdır.

Şekil 3.15. Batarya

3.4.2. Sensörler ve seçimi

Sensörler robotun hızlı tepki verebilmesinde rol oynayan en önemli bileşenlerdendir. Bu bakımdan tepki süreleri en düşük olan sensörler tercih edilmiştir. Robot da iki tip sensör kullanılmıştır. Şekil 3.16.' da yer alan sensörler kızıl ötesi mesafe sensörüdür. Kızılötesi mesafe sensörü, sumo ve mini sumo projelerinde tercih edilen sensörlerdir. Boyutunun küçük olması, hassasiyetinin yüksek ve tepki süresinin az olması sensörün avantajlarındandır. Sensör üzerindeki trimpot ile mesafe ayarı yapılarak 10cm ile 100cm arasında algılama mesafesi değiştirilmiştir. Sensör seçiminde altta yer alan teknik özelliklerdede görüleceği üzere tepki süresinin 1 ms gibi hızlı bir değere sahip olması rakipleri algılama adına üstünlük sağlamaktadır. Kızılötesi çalışma sistemine sahip olmasına rağmen farklı sensörlerde bu değer 5ms nin üzerine çıkmaktadır.

Şekil 3.16. Kızılötesi mesafe sensörü

Teknik Özellikler:

- Çalışma Gerilimi: 12-24V DC
- Çektiği Akım: 100mA
- Görme Mesafesi: Ayarlanabilir 10-100cm
- Tepki Süresi: 1ms
- Sensör Boyutları
    - Uzunluk: 20mm
    - Yükseklik: 33.2mm
    - Genişlik: 10.8mm
    - Kablo Uzunluğu: 180cm

Sensörün 3 bağlantı kablosu bulunmaktadır. Kahverengi kablo dan 12-24 VDC besleme gerilimi verilmesi gerekmektedir. Mavi kablo GND (Toprak) bağlantı kablosudur. Siyah kablo ise sensör çıkışıdır.

Robot da kullanılan diğer sensör QTR kontrast sensörüdür. Şekil 3.17.' de yeralan sensör tek bir kızıl ötesi LED ve bir foto-transistör çifti bulundurur. Üzerindeki foto-transistör bir kapasitör boşalma devresini kullanarak dijital giriş çıkış hattını mikro denetleyici üzerinde kullanarak yansıyan IR analog değerlerini kapasitörün boşalma süresini ölçerek hesaplamaktadır. Akım ihtiyaçları bazı mikro işlemci I/O hatları ile yapılmasına izin vererek, sensörlerin I/O hattı üzerinden güç saklaması çalıştırılması ve durdurulmasını sağlar.

Şekil 3.17. QTR Kontrast sensörü

Bu modül, tek bir kurulum vida deliği bulundurmaktadır. PCB'nin bu kısmı üniteyi daha da küçük yapmak için topraklanabilir. Şekil 3.18.' de yer alan kontrast sensörü 3-pin bağlantı soketine sahiptir ve bu da geleneksel dikey konumla ya da paralel bir şekilde PCB'ye yerleştirilebilir.

Şekil 3.18. QTR Kontrast sensörü PCB ye dikey bağlantı şekli

Teknik Özellikler:

- 5VDC çalışma voltajı
- 25mA besleme akımı
- Dijital I/O uyumlu
- Optimal algılama mesafesi: 0.125" (3mm)

Sensör Boyutlar:

- 7.62x 12.7x 2.54mm

3.4.3. Motor sürücü devresi

Robot da kullanılan motorların yüksek güç ve torka sahip olmaları bu gücün kontrol edilmesi içinde uygun kontrol kartı gerektirmektedir. Şekil 3.19.' de yer alan VNH2SP30 motor sürücü kartı teknik özelliklerinin motorlara uyumlu olması açısından tercih edilmiştir. Bu kontrol kartı 5.5V-16V arasında çalışan yüksek güç gerektiren motorların sürülmesi ve kontrol edilebilmesi için tasarlanmış bir karttır. Ters voltaj, yüksek voltaj, düşük voltaj, yüksek sıcaklık ve yüksek akım koruma özellikleri olan bir motor sürücüsüdür. Bu motor sürücü devresinin tercih edilmesinde en önemli sebeplerden birisi motorların (sağ iki, sol iki) kilitli rotor ve/veya tam yükte çalışması durumunda 2x12A=24A anlık akım çekebilmesi, buna karşılık sağ ve sol yönler için motor sürücü devresinin 30A lik değerinin bu akımı karşılayabilmesi, ayrıca rakip algılandığında PWM de kullanılarak farklı hızlarda hareketin sağlanabilmesdir.

Özellikleri:

- Motor Sürücüsü: VNH2SP30
- Motor Kanal Sayısı: 1
- Minimum Çalışma Gerilimi: 5.5V
- Maksimum Çalışma Gerilimi: 16V
- Sürekli Çıkış Akımı: 14A
- Anlık Maksimum Akım Çıkışı: 30A
- Maksimum PWM Frekansı: 20 kHz
- Ters Voltaj Koruması: Var

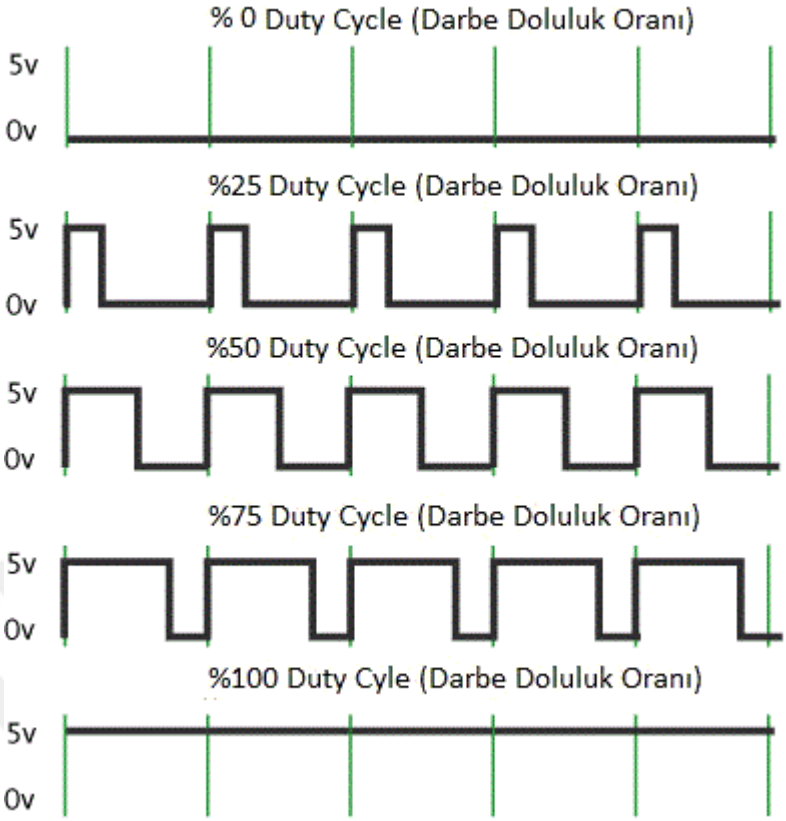
Boyutları:

- 3,81 x 2,99 cm

Şekil 3.19. VNH2SP30 Motor sürücü kartı

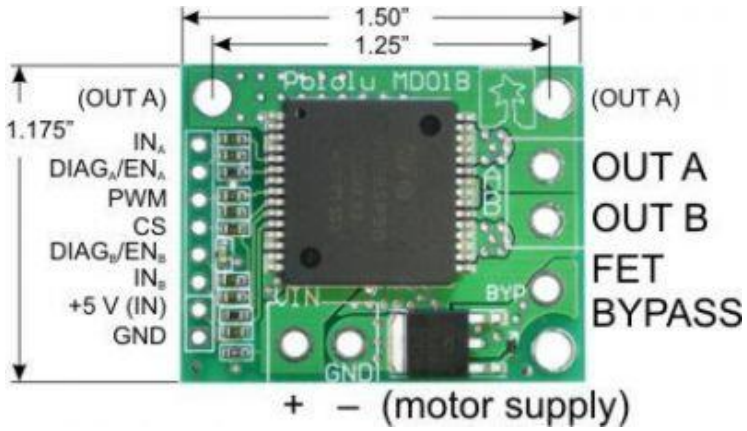
Yarı iletkenlerinin iletimde ve kesimde kalma sürelerini değiştirerek DC motorunun endüvi uçlarına uygulanan darbeli gerilimlerin ortalama değerleri ayarlanabilir. Yarı iletken elemanlarla doğru gerilimden darbeli gerilim elde edilmesi ve bu darbeli gerilimin sinyal oranının yarı iletkenlerin iletimde veya kesimde kalma süreleri ayarlanması yöntemine PWM (Pulse Width Modülation-Darbe Genişlik Modülasyonu) denir. PWM yöntemi diğer kontrol yöntemlerine oranla daha az kayıp ile daha hızlı olarak gerilim kontrolü sağlanmaktadır. Bu nedenle PWM yöntemi daha hızlı ve verimli bir kontrol işlemi gerçekleştir. Burada anahtar belirli oranlarla açılıp kapandığında, DC motorun endüvi uçlarına uygulanan gerilimin değeri değişecektir. PWM ile üretilebilecek olan çeşitli sinyal oranları Şekil 3.20.' de verilmiştir. Darbe doluluk Oranı (Duty Cycle) sinyal oranlarını alttaki formül ile hesaplanabilir.

Darbe doluluk oranı X Genlik Tepe Değeri = Etkin Değer



Şekil 3.20. PWM ile üretilebilecek olan çeşitli sinyal oranları

Sürekli olarak 14A verebilen VNH2SP30 motor sürücü kartı anlık olarak 30A'e kadar akım verebilir. 20KHz' e kadar olan PWM sinyallerini destekler. VNH2SP30 motor sürücü kartının giriş ve çıkış bağlantı noktaları şekil 3.21.' de ki gibidir.



Şekil 3.21. VNH2SP30 Motor sürücü kartı giriş – çıkış bağlantı noktaları

3.4.4. Kontrol kartı

Robot için Şekil 3.22.' de görülen Arduino Mega 2560 kontrol kartı olarak seçilmiştir. Bu kontrol kartı Atmega2560 temelli bir mikrodenetleyici kartıdır. Üzerinde 54 adet dijital giriş/çıkış pini (15 tanesi PWM çıkışı olarak kullanılabilir), 16 analog giriş, 4 UART (donanımsal seri port), 16Mhz kristal, usb soketi, güç soketi, ICSP konektörü ve reset tuşu bulundurmaktadır. Bilgisayara usb kablosu üzerinden bağlanabilir, adaptör veya pil ile çalıştırılabilir. Arduino Mega 2560 program yüklemek ve bilgisayar haberleşmesi yapmak için üzerinde usb-seri dönüştürücü bulundurmaktadır.

Şekil 3.22. Arduino Mega 2560 kontrol kartı

Teknik Özellikler:

- Mikrodenetleyici Atmega2560
- Çalışma Gerilimi 5V
- Giriş Gerilimi (önerilen) 7-12V
- Giriş Gerilimi (limit) 6-20V
- Dijital I/O Pinleri 54 (15 tanesi PWM çıkışı)
- Analog Giriş Pinleri 16
- Her I/O için Akım 40mA
- 3.3V Çıkış için Akım 50mA

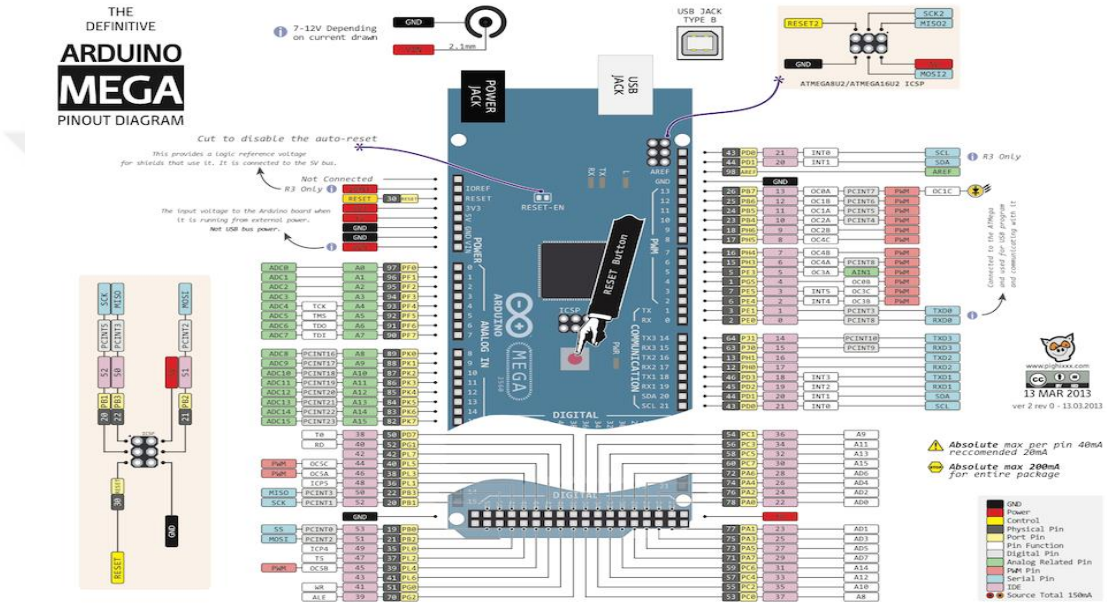
- Flash Hafıza 256KB (Atmega2560) 8KB kadarı bootloader tarafından kullanılmaktadır.
- SRAM 8KB (ATmega2560)
- EEPROM 4KB (ATmega2560)
- Saat Hızı (Frekans) 16MHz
- Uzunluk 101.6mm
- Genişlik 53.4mm
- Ağırlık 36g

Arduino Mega gücünü usb üzerinden veya harici güç kaynağından alabilir. Harici güç kaynağı AC-DC adaptör olabileceği gibi bataryada olabilir. Adaptör kart üzerindeki 2.1mm merkez-pozitif güç soketinden bağlanabilir. Batarya kart üzerindeki GND ve Vin pinleri üzerinden bağlanabilir. Kartın çalışması için sürekli olarak usb'nin bağlı olması şart değildir. Kart sadece adaptör veya batarya ile çalıştırılabilir. Bu sayede kart bilgisayardan bağımsız olarak çalıştırılmıştır.

Harici güç kaynağı olarak 6-20V arası kullanılabilir. Ancak bu değerler limit değerleridir. Kart için önerilen harici besleme 7-12V arasıdır. Çünkü kart üzerinde bulunan regülatör 7V altındaki değerlerde stabil çalışmayabilir. 12V üstündeki değerlerde de aşırı ısınabilir. Mega kartının üzerindeki mikrodenetleyicinin çalışma gerilimi 5V'dur. Vin pini veya güç soketi üzerinden verilen 7-12V arası gerilim kart üzerinde bulunan voltaj regülatörü ile 5V'a düşürülerek karta dağıltılmıştır.

Güç pinleri hakkında bilgi verecek olursak; Vin harici güç kaynağı kullanılırken 7-12V arası gerilim giriş pinidir. 5V pin; regülatörden çıkan 5V çıkışı verir. Eğer kart sadece usb (5V) Üzerin den çalışıyor ise usb üzerinden gelen 5V doğrudan bu pin üzerinden çıkış olarak verilir. Eğer karta güç Vin (7-12V) veya güç soketi (7-12V) üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir. 3V3 pin; kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış sağlar. GND pinleri toprak pinleridir. Arduino mega2560 256 KB'lık flash belleğe sahiptir (8KB kadarı bootloader tarafından kullanılmaktadır). 8KB SRAM ve 4KB EEPROM'u bulunmaktadır.

Şekil 3.23.' de Arduino Mega üzerindeki 54 adet dijital pin görülmektedir. Bu pinlerin hepsi giriş veya çıkış olarak kullanılabilir. 16 tane analog giriş pinide bulunmaktadır. Bu analog giriş pinleride aynı şekilde dijital giriş ve çıkış olarak kullanılabilir. Yani kart üzerinde toplam 70 tane dijital giriş çıkış pini vardır. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maksimum 40mA giriş ve çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir.



Şekil 3.23. Ardiuno Mega giriş – çıkış pinleri

Seri Haberleşme, Serial: 0 (RX) ve 1 (TX), Serial1: 19 (RX) ve 18 (TX), Serial2: 17 (RX) ve 16 (TX), Serial3: 15 (RX) ve 14 (TX): TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Pin 0 ve 1 doğrudan kart üzerinde bulunan Atmega16u2 usb-seri dönüştürücüsüne bağlanmıştır. Yani bilgisayardan karta kod yüklerken veya bilgisayar-konrol kartı arasında karşılıklı haberleşme yapılırkende bu pinler kullanılır. O yüzden karta kod yüklerken veya haberleşme yapılırken hata olmaması için mecbur kalınmadıkça bu pinlerin kullanılmamıştır.

Harici Kesme, 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), 21 (interrupt 2): Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir. PWM, 2-13 ve 44-46: 8-bit

çözünürülükte PWM çıkış pinleri olarak kullanılabilir. SPI, 53 (SS), 51 (MOSI), 50 (MISO), 52 (SCK): Bu pinler SPI haberleşmesi için kullanılır.

LED 13; Atmega üzerinden 13. pine bağlı olan dâhili bir led bulunmaktadır. Pin HIGH yapıldığında led yanacak, LOW yapıldığında led sönecektir. Analog, A0-A15; Mega 16 tane 10-bit çözünürlüğünde analog giriş pinine sahiptir. Bu pinler dijital giriş ve çıkış içinde kullanılabilir. Pinlerin ölçüm aralığı 0-5V'dur. AREF pini ve analog Reference() foksiyonu kullanılarak alt limit yükseltilip, üst limit düşürülebilir. I2C, 20 veya SDA pini ve 21 veya SCL pini: Bu pinler I2C haberleşmesi için kullanılır. Aref; Analog giriş için referans pinidir. Reset; Mikrodenetleyici resetlenmek istendiğinde bu pin LOW yapılır. Reset işlemi kart üzerinde bulunan Reset Butonu ile de yapılabilir.

Arduino Mega'nın bilgisayarla, başka bir arduino veya mikrodenetleyici ile haberleşmesi için birkaç farklı seçenek vardır. Atmega2560, 4 tane donanımsal UART TTL (5V) seri haberleşme imkânı sunar. Kart üzerinde bulunan Atmega16u2 usb-seri dönüştürücüde bilgisayarda sanal bir com port açarak Atmega2560 ile bilgisayar arasında bir köprü kurar. Arduino bilgisayar programı içerisinde barındırdığı seri monitör ile arduino ile bilgisayar arasında text temelli bilgilerin gönderilip alınmasını sağlar. Usb-seri dönüştürücü ile bilgisayar arasında usb üzerinden haberleşme olduğu zaman kart üzerinde bulunan RX ve TX ledleri yanacaktır.

Arduino Mega üzerinde donanımsal olarak 4 adet seri port bulunmaktadır. Ancak Software Serial kütüphanesi ile bu sayı yazılımsal olarak arttırılabilir. Atmega2560 aynı şekilde I2C ve SPI portlarıda sağlamaktadır. Arduino bilgisayar programı ile gelen Wire kütüphanesi I2C kullanımını, SPI kütüphanesi de SPI haberleşmesini sağlamak için kullanılır.

Arduino Mega kartı Arduino bilgisayar programı (Arduino IDE) ile programlanır. Programda Tools> Board sekmesi altında Arduino Mega'yı seçip programlamaya başlayabilirsiniz. Arduino Mega üzerindeki Atmega2560 üzerine bootloader denilen özel bir yazılım yüklü gelir. Bu sayede kartı programlarken ekstra bir

programlayıcı kullanmanıza gerek yoktur. Haberleşme orjinal STK500 protokolü ile sağlanır. Şekil 3.24.' de örnek bir kod verilmiştir

Şekil 3.24. Ardiuno IDE ile yapılan örnek bir kod

Arduino Mega üzerinden bulunan resetlenebilir sigorta bilgisayarın usb portunu kısa devrelerden veya aşırı akım tüketimi durumlarından korumaktadır. Kart bilgisayar usb portu üzerinden 500mA' den fazla akım çektiğinde kart otomatik olarak usb'den aldığı gücü koruma amacıyla kesmektedir. Fazla akım durumu veya kısa devre ortadan kaldırıldığında sigorta normal konuma döner ve tekrar bağlantı kurulur.

Kullanılılmış olan Arduino Mega 2560 kontrol kartının çalışma frekansı 16Mhz değerindedir. Bu frekans robotun kontrolü için yeterli bir frekans olmasına rağmen donanım ve yazılımsal yapı tercih edilemesi durumuda bu değer daha yüksek frekanslarda çalışmaya imkân veren kontrol kartları ile uygulanabilir.

3.5 GZÇS Robotun Yazılımsal Bileşenleri

GZÇS Robotun yazılımsal bileşenleri robotun en öne çıkan özelliğidir. Robot bu yazılım sayesinde otonom olarak hareket edebilmekte değişken durumlara göre pozisyon alabilmektedir. Ayrıca yazılımın gerçek zamanlı olması robotun birçok işlevi hızlı ve aynı zamanda gerçekleştirebilmesini sağlamaktadır.

3.5.1. Gerçek zamanlı işletim sistemi kavramı

Gerçek zamanlı işletim sistemi (GZİS) çoklu görevleri hızlı ve doğru bir şekilde gerçekleştirebilen uygulamalar için tasarlanmış bir işletim sistemidir. Gerçek zamanlı işletim sistemi günümüzde birçok alanda ve çeşitli cihazlarda kullanım alanı bulmuştur. GZİS Belli bi zaman dilimi içerisinde mantıksal olarak doğru sonuç elde edilmesi gereken gerçek zaman uygulamalarında ve gömülü sistemlerde kullanılır. GZİS zamana bağlı olarak kaynakları verimli kullanmasıdır. Gerçek zamanlı işletim sistemi iki bölümden oluşmaktadır. Bunlar: Gerçek zaman ve işletim sistemidir.

Gerçek zaman gerçekleştirilen işlemlerin yanıt verme süresinin belli bir değer aralığında olmasıdır. Gömülü sistemlerde bu değerin önemi artmaktadır. Bu nedenle değerin öneminin yüksek olduğu sistemlerde gerçek zaman olgusunun garanti altına almak için bazı mekanizmalar kullanılır. Bu pencereden bakıldığında GZİS gerçek zamanı garanti eden bir mekanizmadır.

Gerçek zamanlı işletim sistemini oluşturan bölümlerden bir diğeri işletim sistemidir. İşletim sistemi uygulamayı gerçekleştiren programlar ile donanım arasında bir arayüz oluşturan programdır. Bu program oluşturduğu katman ile kullanıcıların donanımlara günümüzde geçerli yazılım dilleri ile kolayca ulaşmalarını sağlar. GZİS temelde bir işletim sistemi olduğundan gömülü sistemlerle kullanıldığında tasarımcılara işletim sisteminin sağladığı kolaylıkları kullanmasına imkân tanır.

İşletim sisteminin birinci görevi uygulama programlarının ihtiyaçlarını karşılamak amacı ile donanım kaynaklarının yönetilmesini sağlamaktır. Bu amacı aşağıdaki özellikleri sayesinde gerçekleştirir.

- Çoklu Görev (Multitasking)
- Senkronizasyon (Syncronzation)
- Kesme ve Olay Yönetimi (Interrupt and Event Handling)
- Giriş / Çıkış (Input / Output)
- İç Görev İletişimi (Inter Task Communication)
- Zamanlama ve saat (Timers ve Clocks)
- Bellek Yönetimi (Memory Management)

Günümüzde bu özellikleri sağlayan işletim sistemleri yaygınlaşmıştır; vxWorks, Integrity, LinuxRT bunlar arasında sayılabilir. İşletim sistemleri "Görev" (task) ve "Öncelik" belirlemelerine ilişkin altyapı içermekte ve çeşitli görev sıralaması mekanizmalarını sağlamaktadır. Bu tür mekanizmalar çalışma zamanının başında belirlenerek yazılımın çalıştığı süre boyunca uygulanabildiği gibi, yazılım çalışırken de değiştirilebilmektedir. Yazılımsal olarak iş paylaştırma ve koordineli çalışma algoritmaları Microsoft Windows işletim sistemi türevlerin üzerinde özellikle robot kontrolü içinde kullanılan Microsoft Developer Studio (MRDS) yazılımında çekirdek (Kernel) içerisinde yer alan "Concurrency and Coordination Runtime" (CCR) ve "Disturubuted Software Services" (DSS) alt yapıları sağlanmaktadır. Kullanıcı tarafından çok işlemcili ve çok çekirdekli donanımsal yapılar kullanıldığında Windows ortamında bu işlem işletim sistemi çekirdeğinde anılan yazılımsal yapılarla sağlanmakta ve kullanıcı müdahele edememektedir. Windows platformunda manuel olarak birden çok işlemci ya da çekirdeğe iş paylaşımı yaptırılmak istenirse kullanılacak yazılımların bu yapıya uygun çalışması için yazılımcı tarafından bu işlemlerin yapılması gerekmektedir. Gerçek zamanlı çalışan sumo robot gerçek zamanlı işletim sisteminin çalışma mantığına göre ana işi iş parçacıklarına ayırabilmek ve/veya çoklu görev yönetimi için Arduino Mega kontrol kartını destekleyen FreeRTOS kütüphanesi tercih edilmiştir.

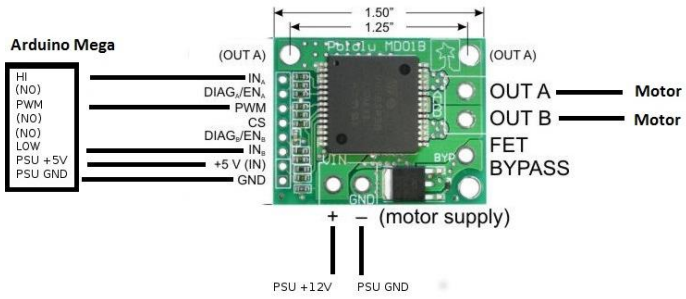
FreeRTOS kütüphanesi diğer gerçek zamanlı işletim sistemlerinin (Windows, Linux, Unix vs.) yapıları temel alınarak hazırlanmış kod bloğudur. Diğer işletim sistemlerinde RTOS sistemde gömülü olarak bulunmaktadır. Arduino Mega RTOS gömülü değildir. FreeRTOS kütüphanesi yardımı ile robotun donanımlarına özel kodlamayı yapabilmemizi sağlar. FreeRTOS kütüphanesinde kullanılan görevin önceliği ve önceliği seçilen görevin ne kadar süre çalışması gerektiği bölümüdür.

3.5.2. Gerçek zamanlı işletim sistemi kodlarının kontrol yazılımı ile birleştirilmesi

Robot üzerinde kullanılan FreeRTOS kütüphanesi robotun donanımlarına özel kodlanabilir olduğu ve kodlama yapısında esneklik sağlamasıdan dolayı tercih edilmiştir. Gerçek zamanlı işletim sistemi kodlarının kontrol yazılımı ile birleştirilmesi belli bir işlem sırasına göre gerçekleştirilir. Gerekli donanımların eşzamanlı çalışabilmesi için her birine özgün görevler tanımlanır. Bu donanımların task (görev) tanımlanırınında robotun çalışma yapısına uygun şekilde öncelikler tanımlanır. Tanımlanan bu görevlere stack size (Yığın boyutlandırılması) tanımlaması yapılır. Yığın boyutlandırılması donanımların çalışma yoğunluğuna göre belirlenir.

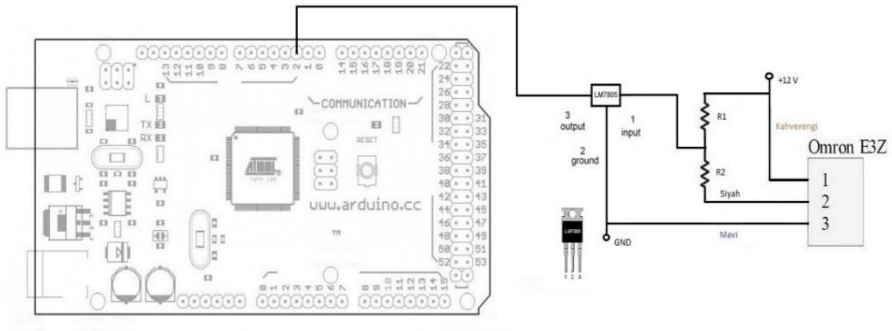
3.6. Donanımsal ve Yazılımsal Bileşenlerin Entegrasyonu

Robotun yapısında kontrol kartı, motor sürücü devreleri ve sensörler için RTOS dan farklı temel kodlama ile yazılım geliştirilir. Herbir donanımın teknik özelliklerine bakılarak, kontrol kartı üzerindeki yazılımda uygulanacak yöntemler belirlenir. Bu işlem motor sürücü devresi için incelenirse motor sürücü kartının üzerinde ki giriş ve çıkış portlarının hangi değerleri alacağı kataloğudan incelenerek yazılım üzerinde Şekil 3.25.' de görüldüğü gibi PWM sinyali, Motor kontrol girişleri besleme ve şase bağlantı noktaları ayarlanır. IN A ve IN B girişleri üzerinden motorun devir yönü kontrolü sağlanmıştır.



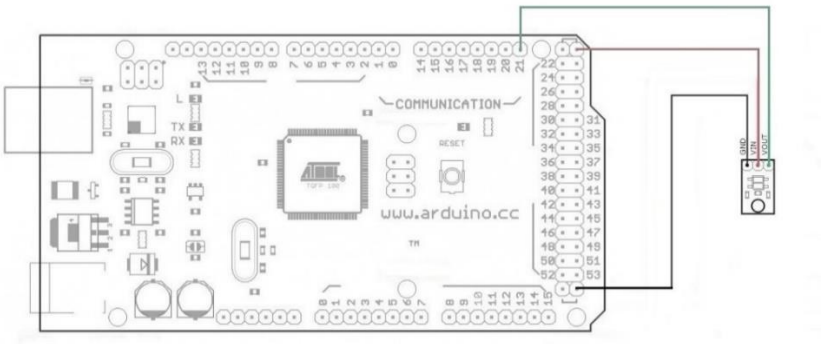
Şekil 3.25. Motor kontrol kartı ile kontrol kartı bağlantı şeması

Kızılötesi Sensörler için bağlantı uçları hangi değerleri alacağı teknik özellikleri incelenerek Şekil 3. 26.' da görüldüğü gibi 5V' luk sistemlerle kullanımda bir 7805 gerilim regülatörü ile kullanılarak bağlanmıştır.



Şekil 3.26. Kızılötesi mesafe sensörü ile kontrol kartı bağlantı şeması

Kontrast sensörleri için bağlantı uçları hangi değerleri alacağı teknik özellikleri incelendiğinde QTR kontrast sensörünün 3 bağlantı pini olduğu görülmüştür. Siyah kablo GND (Toprak) bağlantı kablosu, kırmızı kablodan (Vin) 3.7-5 Volt DC besleme verilmesi gerekir ve yeşil kablo Vout sinyal çıkışı alınmaktadır. QTR kontrast sensörü Arduino mega kartına Şekil 3.27.' deki gibi bağlanmıştır.



Şekil 3.27. QTR Kontrast Sensörü ile Kotrol Kartı Bağlantı Şeması

4. ARAŞTIRMA BULGULARI

Gerçek zamanlı işletim sistemine sahip sumo robotu hem gerçek zamanlı işletim sistemi ile hemde ardiuno komut sistemi ile belirli görevleri yapması için programlanmıştır. Bunun için her iki programlamada sonucu somut olarak gözlemleyebileceğimiz görevler seçilmiştir. Bu görev seçme testinde FreeRTOS kütüphanesi kullanılarak ve kullanılmadan aynı işlemlerin (hesaplamaların) karşılaştırılabilmesi adına Çizelge 4.1.' de yer alan görevler tanımlarak uygulanmış ve bu işlemleri yerine getirme süreleri yazılımsal olarak ölçülmüştür. Bunlara örnek olarak sistem boşta, belirli pinden değer okuma, belirli pine değer yazma vb. görevlerdir. Bu görevler kontrol algoritmasında farklı sensörlerden veri alma, gerekli kontrol sinyallerini üretme ve motorların hareketinde kullanılan sinyallerdir. Kullanılan sistem zaman saatini hazır fonksiyon (millis) ile bir değişkene aktararak görevlerin başlangıç ve bitiş zamanları kaydedilmiş ve bu kayıtlar arasındaki farktan yararlanılarak o görevdeki çalışma süresi ölçülmüştür. Bu ölçümler sonucunda Çizelge 4.1.' de ki değerler elde edilmiştir.

Çizelge 4.1.' de birinci sütun çalıştırlan test görevlerini, ikinci sütun tek görev ile RTOS çalıştırmada ölçülen süreler üçüncü sütun ise RTOS kütüphanesi kullanılmadan ölçülen süreleri göstermektedir.

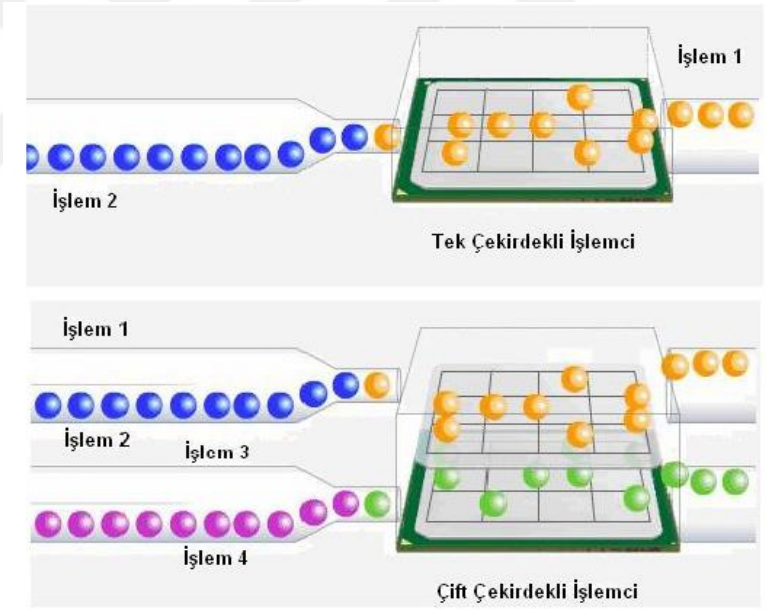
Çizelge 4.1. Görevlerin gerçekleştirme süreleri

DENEME	RTOS tek Task (us)	RTOSSUZ(us)
nop	0,063	0,063
digitalRead	6,355	6,287
digitalWrite	7,21	7,145
pinMode	4,599	4,53
multiply byte	0,632	0,632
divide byte	5,54	5,535
add byte	0,569	0,569
multiply integer	1,388	1,386
divide integer	14,477	14,467
add integer	0,884	0,883
multiply long	6,36	6,355
divide long	38,912	38,887
add long	1,766	1,765
multiply float	7,747	7,742
divide float	83,537	83,462
add float	9,617	9,607
itoa()	12,962	12,957
ltoa()	126,187	125,987
dtostrf()	81,012	80,887
random()	93,187	93,137
y |= (1<<x)	0,569	0,569
bitSet()	0,569	0,569
analogReference()	-0,013	-0,013
analogRead()	111,987	111,987
analogWrite() PWM	8,817	8,867
delay(1)	1008,487	1007,987
delay(100)	99999,984	100024,984
delayMicroseconds(2)	0,758	0,757
delayMicroseconds(5)	3,779	3,776
delayMicroseconds(100)	99,487	99,337

Bu değerlerden de anlaşılacağı üzere; görevler için gereken zamanlar aşağı yukarı eşit çıkmıştır. Burada gerçek zamanlı işletim sistemi olmayan programlamada görevler sırayla yapılmakta ve bir görev bitmeden diğerine geçilmemektedir.

Gerçek zamanlı işletim sisteminde de görev yöneticileri ayrı ayrı olmakla beraber kullanmış olduğumuz Arduino Mega kontrol kartı tek çekirdekli bir işlemciye sahiptir. Şekil 4.1.' de görüldüğü üzere çift çekirdekli işlemci, tek bir fiziksel işlemci içinde aynı frekansta çalışan iki tam yürütme biriminden oluşur. Çift çekirdekli işlemcili bir kontrol kartı, kontrol kartının daha yüksek kapasite ve eşzamanlı bilgi işlemeyi artırdırdığı için yeni imkânlar sunar.

Gerçekleştirilen yazılımda çoklu görev yönetimi yapısı önceliklendirme kullanılarak oluşturulmuştur. Kullanılmış olan kontrol kartı tek işlemci ve tek çekirdek yapısında olduğu için, iş parçacıklarına ayırıp ayırmama arasında süre bakımından farklılık oluşması görülmemiştir. Oluşturulan yazılımın ana işi rakip robotu algılayarak en kısa sürede dohyo dışına çıkarmaktır. Buradan hareketle her bir sensörden gelen veri motorlara kontrol algoritması ile işlenip aktarılması ana görevdir. Donanımsal yapıda tek işlemci ve tek çekirdek yapısının bulunması normalde her giriş ve her çıkış için iş parçacıkları olarak tanımlanması gerekirken kontrol kartının bu donanımsal kısıtı sebebi ile alt iş parçacıkları alt görevler şeklinde çalışmakta, görev yönetimide ancak önceliklendirme ile sağlanmıştır. Fakat yine bu durum Şekil 4.1.' de görüleceği üzere donanımsal bir kısıtlılık sebebi ile görevlerin önceliklendirilmesininde kontrol algoritmesında çok fazla bir avantaj sağlamadığı tez sonuçlarından görülmüştür.



Şekil 4.1. Tek çekirdekli işlemci ile çift çekirdekli işlemci yapısı

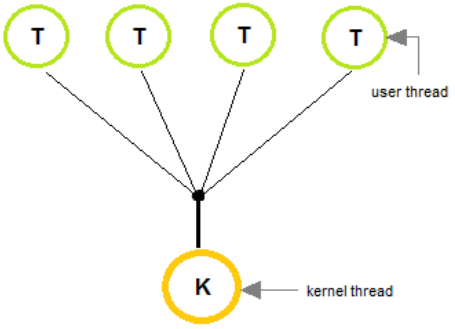
Bu projede GZÇS robot da GZİS de kullanılan algoritma modeli alt kısımda açıklanan çoklu iş parçası yönetimi (multithreading) modellerine benze şekilde açıklanabilir. Benzer şekilde çoklu görev yönetimi kullanımı (Multitasking) yapısının yazılımlarda kullanılma amacı algoritmalarda öncelik atama yapılmasına rağmen asıl önemli husus çalışmaya hazır olan çok sayıda görev arasında hızlı geçişin sağlanması ve tüm görevlerin en kısa zamanda

bitirilebilmesi için işlemci çekirdeğinin optimum paylaşım sürelerinde kullanılmasıdır (Agyp & Rubin, 2012; Lin,2013).

Çoklu iş parçası (multithreading) çalışma modelleri üç tiptir.

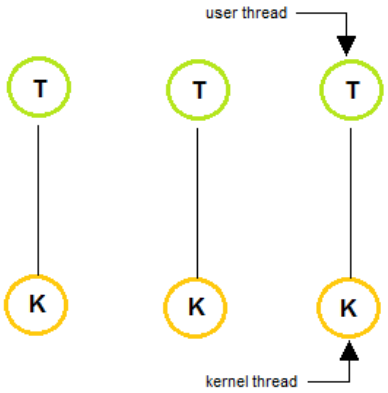
- Many to one
- One to one
- Many to many

Şekil 4.2.' de görülen "many to one" modelinde birçok görevi tek bir işlem kaynağına yönlendirir. Bir kaynak birçok görevi kontrol etmekle uğraşır.



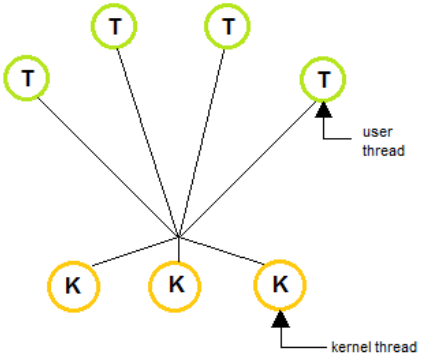
Şekil 4.2. Many to one modeli (Techtud, 2019)

Bir diğer model de "one to one multithreading" dir. Bu modelde her işlemci ayrı bir kayanağa sahiptir. Bu modelde kaynak sayısı kadar iş yapılır. Bu model şeması Şekil 4.3.' de gösterilmiştir.



Şekil 4.3. One to one modeli (Techtud, 2019)

Üçüncü model ise “many to many” dir. Bu modelde birçok kaynak birçok işlemciye yönlendirilir. Bu model şeması Şekil 4.4.’de gösterilmiştir.



Şekil 4.4. Many to many modeli (Techtud, 2019)

Bu iş parçacığı modeli ve çoklu çekirdek yapısı kullanıldığında, işlemcinin içindeki çekirdek adedinin iki katına çıkartılarak kontrol kartının yeteneklerinin ve bilgi işlem kaynaklarının önemli ölçüde artırılması ve daha hızlı tepki süresi, daha yüksek çok kademeli işlem kapasitesi ve paralel bilgi işlem özellikleri sağlanabilecektir. Benzer şekilde çift çekirdekli işlemci tabanlı bir kontrol kartı ile birden çok görev aynı anda gerçekleştirilebilir.

Çalışmada oluşturulan yazılım Ardiuno Mega kontrol kartının donanımsal kısıtlarıda göz önüne alındığında “many to one” modelinin çalışma yapısına benzemektedir. Sensörlerden gelen herbir veri kullanıcıya ait iş parçaları (user thread) yapısına benzerlik göstermektedir. Kontrol kartının donanımsal kısıtları bu yapıda yer alan çoklu iş parçacığı yapısından zıyade öncelikli görev yönetimi yapısınada benzetilebilir. Görevlerin önceliklendirilmesi, tam anlamıyla gerçekleşmediğinden, görevleri belirli öncelik sırasına göre yerine getirmekte, bu da işlemlerin daha hızlı gerçekleştirilmesine engel teşkil etmiştir.

SONUÇ VE ÖNERİLER

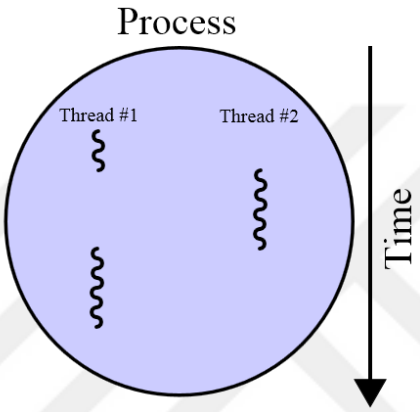
Bu çalışmada; gerçek zamanlı işletim sistemi ile bir sumo robot tasarımı ve uygulaması gerçekleştirilmiştir. Araştırma bulgularından da görüldüğü üzere gerçek zamanlı işletim sistemi ile çalışan sumo robotun gerçek zamanlı çalışmayan sumo robota göre daha hızlı olduğunu deneysel olarak gösterilebilmesi, uygun çoklu iş parçacığı modeli (GZİS desteği) ve kullanılan kontrol kartının teknik özellikleri ile sınırlılık göstermektedir. Gerçek zamanlı işletim sistemine sahip sumo robota verilen görevlerin öncelik sırası ve gerçekleştirilme zamanı yazılımsal olarak değiştirilebildiği için, gerçek zamanlı işletim sistemine sahip olmayan sumo robota göre yazılımsal anlamda daha esnek özelliklere sahip olduğu görülmüştür. Bu durum daha ayrıntılı açıklanırsa; gerçek zamanlı işletim sistemi ile istenilen iş istenilen zamanda, istenilen süre ile çalıştırabilir. İşlere öncelik sırası verilerek, önce çalışması gereken işi daha önce çalıştırabilir.

Çoklu iş parçacığı yönetimi (multithreading), bir görevi birden fazla parçaya bölüp, bu bölümleri birbirinden bağımsız olarak çalıştırabilen aynı zamanda kaynağı yönlendirerek birden fazla görevi çalışmasını sağlayan bir modeldir.

Öncelikli çoklu görev yönetimi donanımda yer alan 5+2 sensörden alınan giriş verileri üzerinde uygulanmış, fakat bu yedi ayrı görevin (yedi ayrı sensör giriş verisi) önceliklendirildiğinde yazılımda promlemler yaşanmış, kontrol kartının sınırlılıkları bu veri işleme hızına cevap verememiştir. Bu durum robotun yapması gereken işlemleri yerine getiremediğini göstermiştir. Bu problem görev önceliklendirilmesinin dört sensör verisinin öncelikli olarak işlenmesi ile (öncelik sayısının dörde düşürülmesi ile) çözülmüştür. Bu çözüm sonrası robotun yapması beklenen doğru hareketleri yerine getirdiği görülmüştür.

Eğer bu model tek işlemci veya tek çekirdekli yapı ile değil de birden fazla işlemci veya birden fazla çekirdek içeren donanımsal yapılar ile denenmiş olsaydı daha hızlı bir işlem yapma yeteneğini kazanması sağlanabileceği görülmektedir.

Bu tez çalışmasında asıl amaç hızlı algılama ve karar verme için çoklu iş parçacığına (many to one modeli) benzer bir öncelikli görev yönetimi kullanılarak, bir işlemcili tek çekirdekli sistem üzerinde, kontrast sensörleri ve cisim algılamada kullanılan mesafe sensörlerinin önceliğini belirleyerek robotun dohyo sınırlarını algılaması ya da rakibi algılaması arasındaki öncelik belirlenmesi sağlanmıştır. Aynı zamanda da bu görevlerin ne kadar süre ile çalışmaları gerektiği belirlemiştir. Şekil 5.1.' de Çoklu iş parçacığı yönetimi modeli gösterilmiştir.



Şekil 5.1. Çoklu iş parçacığı yönetimi modeli

Kullanılacak kontrol kartının çok çekirdekli veya çok işlemcili seçilmesi halinde gerçek zamanlı işletim sistemi olan sumo robount görevleri "Multithreading Many to Many" modeli ile daha hızlı gerçekleştirebilir. Kontrol kartında kullanılacak işlemcinin çalışma frekansının yükseltilmesi de birim zamanda yapılacak iş miktarını arttıracak ve böylelikle daha hızlı karar verme ve harekete geçme sağlanabilecektir. Fakat farklı kontrol kartları veya işlemci seçimi yapıldığında yazılımsal uyumluluklar, robotun fiziksel sınırlılıkları ve maliyetler de göz önünde bulundurulmalıdır.

Farklı daha yoğun hesaplama ve iş yükü gerektirecek robotik uygulamalarında birden çoklu işlemci yönetimi (multiprocessing) ya da çoklu çekirdek üzerinde uygun iş parçacığı modeli (multithreading) birlikte de tercih edilebilir. Benzer uygulamalar farklı yazılım uygulamalarında, sunucular ve iş istasyonlarında üzerine yük dengeleme yazılım desteği de eklenerek farklı işletim sistemleri ile

kullanılabilir. Bu durum pek çok endüstriyel otomasyon ve robotik uygulamasında görülmektedir. Bunun yanı sıra koordinasyon ve dağıtık yazılım servisleri de kullanıldığında yazılımsal yeni esneklikler sağlanabilir.

Çalışmanın devamında yapılacak çalışmalar için yazılımsal geliştirmelerin yanı sıra, fiziksel sınırılıklar da göz önünde tutularak çok çekirdekli ve daha yüksek frekansa sahip işlemcili kontrol kartları ve tepki süresi, devir sayısı ve gücü daha yüksek motorlar seçilerek farklı sumo robot uygulamaları gerçekleştirilrbilir. Fakat bu durum donanımsal maliyet olarak bir anda çok yüksek rakamlara çıkmasına sebep olabilmektedir.

KAYNAKLAR

Albayrak, M., Çakır, A., Coşkun, A., (2003). Sumo Robotları İçin Bulanık Mantıkla Hareket Optimizasyonu Uygulaması. International Twelfth Turkish Symposium on Artificial Intelligence & Neural Networks, 2003, Çanakkale.

Albayrak, M., & Yüksel, Y., (2007). 8051 Mikrodenetleyicili Bir Sumo Robot Tasarımı ve Uygulaması. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 11(1), 96-102.

Asimov, I., (1992). Ben Robot, Altın Kitaplar yayınevi, 45.

Boston Dynamics, (2018). Erişim Tarihi: 20.05.2019 https://www.bostondynamics.com/atlas

Chojnowski, K., Wasilewski, P., & Grądzki, R., (2018). Movement modeling of mobile robot in MegaSumo category. In *AIP Conference Proceedings* (Vol. 2029, No. 1, p. 020010). AIP Publishing.

İTÜ Bilgi işlem Daire Başkamlığı, (2013). Erişim Tarihi:05.10.2018 http://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/06/%C3%A7ift-%C3%A7ekirdekli-i%C5%9Flemci

Katipoğlu, G., (2013). Üç serbestlik dereceli robot kolunun pozisyon kontrolü (Doctoral dissertation, DEÜ Fen Bilimleri Enstitüsü), 2-3.

Lima, C. E. O., de Araújo, F. M. A., da Silva, M. B., Rocha Filho, A. E., Rabêlo, R. D. A. L., da Silva, T. A. R., & de Oliveira Alves, A. J., (2013). Velocity and direction planning in a sumo robot type using the method of potential field with fuzzy systems. In 2013 16th International Conference on Advanced Robotics (ICAR) (pp. 1-7). IEEE.

Morgan, S., (2008). Programming Microsoft® Robotics Studio, 1st ed. Microsoft Press, USA

Qian K., Haring D., Cao L., (2009). Embedded Software Development with C, 184-200.

Robotistan.com, (2018). Arduino Mega 2560, Erişim tarihi: 13.04.2018 https://www.robotistan.com/orjinal-arduino-mega-2560-r3-yeni-versiyon-1

Schultz, C., Schreyoegg, J., & von Reitzenstein, C. (2013). The moderating role of internal and external resources on the performance effect of multitasking: Evidence from the R&D performance of surgeons. *Research Policy*, *42*(8), 1356-1365.

Trevor T., Kyle J., (2013). Professional Microsoft Robotics Developer Studio, 223-325.

Techtud, (2019). Multitheread modelleri, Erişim tarihi: 20.04.2019 https://www.techtud.com/short-notes/multithreading-models-many-one-model-one-one-model-many-many-model

Ulukuş, S., Süzen A., (2011). Robot Programlama. Kodlab Yayın Dağıtım Yazılım ve Eğitim Hizmetleri San.ve Tic. Ltd. Şti., 272. İstanbul.

Yiğiter E., (2010). Mobil keşif robotu tasarımı. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans tezi, 19. İstanbul

Zurawski R., (2007). Embedded Systems Handbook, Second Edition, 13-17.
Naughton, J. D. 1984. Futurology and robots: Karel Čaper's RUR. Culture, Theory and Critique, 28(1), 72-86.

EKLER

EK A. Kod Bloğu

RTOSsuz Zaman Deneme Kod Bloğu Örnek Parçası:

```
//Seri portun konfigürasyonu//
void setup()
{
 Serial.begin(9600);

 //I/O aktivasyonu//
 pinMode(9, OUTPUT);
 pinMode(10, INPUT);

 Serial.println("Speed Test will begin momentarily.");
 Serial.println("");

delay(4000);

}

//Değişken tanımlamaları//
void loop()
{
 register int i,j;
 volatile unsigned char c1,c2;
 volatile int v;
 volatile long l1,l2;
 volatile float f1,f2;
 int p,q;
 long int r;
 unsigned long m,n;
 float d, overhead;
 char buffer[30];

 Serial.println(F(""));
 Serial.println(F("Speed test"));
 Serial.println(F("----------"));

 Serial.print(F("F_CPU = "));
 Serial.print(F_CPU,DEC);
 Serial.println(F(" Hz"));
 Serial.print(F("1/F_CPU = "));
 Serial.print((1000000.0/(float)F_CPU),4);
 Serial.println(F(" us"));

 delay(800);    // Allow the Serial text to be transmitted
```

```
Serial.print(F("  nop                    : "));
delay(70);     // Allow the Serial text to be transmitted
m=millis();
for (i=0; i<100; i++)
{
  for (j=0; j<10000; j++)
  {
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
  }
}
n=millis();
d = ((float)n - (float)m) / ((float)i * (float)j);
d *= 1000.0;          // in micro seconds
// Calculate overhead with 'nop' instruction per loop in microseconds
overhead = d - (20.0 * (1000000.0/(float)F_CPU));
d -= overhead;
d /= 20.0;            // per instruction
Serial.print (d,3);
Serial.println (F(" us"));

Serial.print(F("  digitalRead            : "));
delay(70);     // Allow the Serial text to be transmitted
m=millis();
for (i=0; i<2; i++)
{
  for (j=0; j<10000; j++)
  {
    digitalRead(10);
    digitalRead(10);
```

```
    digitalRead(10);
    digitalRead(10);

    digitalRead(10);
    digitalRead(10);

    digitalRead(10);
    digitalRead(10);

    digitalRead(10);
    digitalRead(10);

    digitalRead(10);
    digitalRead(10);

    digitalRead(10);
    digitalRead(10);

    digitalRead(10);
    digitalRead(10);

    digitalRead(10);
    digitalRead(10);

    digitalRead(10);
    digitalRead(10);
  }
}
n=millis();
d = ((float)n - (float)m) / ((float)i * (float)j);
d *= 1000.0;
d -= overhead;
d /= 20.0;
Serial.print (d,3);
Serial.println (F(" us"));
```

EK B. Kod Bloğu

RTOS Kullanılarak Test Kod Bloğu Örnek Parçası:

```
#include <Arduino_FreeRTOS.h>

void Task1( void *pvParameters );

void setup() {

 xTaskCreate(Task1, "K1 ve K2 aynı anda beyaz", 150, NULL, 1, NULL);
 Serial.begin(9600);

 pinMode(9, OUTPUT);
 pinMode(10, INPUT);
 pinMode(7,INPUT);

 Serial.println("Speed Test will begin momentarily.");
 Serial.println("");

delay(4000);


}

void loop() {
  // put your main code here, to run repeatedly:
}
void Task1(void *pvParamaters) {
 (void) pvParamaters;
 //loop here
while(1){

 register int i,j;
 volatile unsigned char c1,c2;
 volatile int v;
 volatile long l1,l2;
 volatile float f1,f2;
 int p,q;
 long int r;
 unsigned long m,n;
 float d, overhead;
 char buffer[30];

 Serial.println(F(""));
 Serial.println(F("Speed test"));
 Serial.println(F("----------"));

 Serial.print(F("F_CPU = "));
```

```
Serial.print(F_CPU,DEC);
Serial.println(F(" Hz"));
Serial.print(F("1/F_CPU = "));
Serial.print((1000000.0/(float)F_CPU),4);
Serial.println(F(" us"));

delay(800);    // Allow the Serial text to be transmitted

Serial.print(F("  nop                    : "));
delay(70);     // Allow the Serial text to be transmitted
m=millis();
for (i=0; i<100; i++)
{
  for (j=0; j<10000; j++)
  {
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
    asm volatile ("nop");
  }
}
n=millis();
d = ((float)n - (float)m) / ((float)i * (float)j);
d *= 1000.0;          // in micro seconds
// Calculate overhead with 'nop' instruction per loop in microseconds
overhead = d - (20.0 * (1000000.0/(float)F_CPU));
d -= overhead;
d /= 20.0;            // per instruction
Serial.print (d,3);
Serial.println (F(" us"));

Serial.print(F("  digitalRead            : "));
delay(70);     // Allow the Serial text to be transmitted
```

```
m=millis();
for (i=0; i<2; i++)
{
 for (j=0; j<10000; j++)
 {
  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);

  digitalRead(10);
  digitalRead(10);
 }
}
n=millis();
d = ((float)n - (float)m) / ((float)i * (float)j);
d *= 1000.0;
d -= overhead;
d /= 20.0;
Serial.print (d,3);
Serial.println (F(" us"));
```

EK C. Kod Bloğu

RTOS Kullanılan Sumo Robot Kod Bloğu Örneği:

```
#include <Arduino_FreeRTOS.h>

void forward();
void backward();
void stop();
void turnLeft();
void turnRight();

int sen_1 = 30;
int sen_2 = 32;
int sen_3 = 34;

int senL = A1;
int senR = A15;

int led=13;

int engineLeftA = 10;
int engineLeftB = 9;
int engineLeftP = 8;

int engineRightA = 7;
int engineRightB = 6;
int engineRightP = 5;

void gorev_1 (void *pvParameters);
void gorev_2 (void *pvParameters);
void gorev_3 (void *pvParameters);
void gorev_L (void *pvParameters);
void gorev_R (void *pvParameters);

void setup()
{
 pinMode(engineLeftA, OUTPUT);
 pinMode(engineLeftB, OUTPUT);
 pinMode(engineLeftP, OUTPUT);
 pinMode(engineRightA, OUTPUT);
 pinMode(engineRightB, OUTPUT);
 pinMode(engineRightP, OUTPUT);

 pinMode(led, OUTPUT);

 pinMode(sen_1, INPUT);
 pinMode(sen_2, INPUT);
```

```
 pinMode(sen_3, INPUT);
 pinMode(senL, INPUT);
 pinMode(senR, INPUT);

 Serial.begin(9600);

 xTaskCreate(gorev_1, "gorev_1", 100, NULL, 1, NULL);
 xTaskCreate(gorev_2, "gorev_2", 100, NULL, 1, NULL);
 xTaskCreate(gorev_3, "gorev_3", 100, NULL, 1, NULL);


 xTaskCreate(gorev_L, "gorev_L", 100, NULL, 0, NULL);
 xTaskCreate(gorev_R, "gorev_R", 100, NULL, 0, NULL);
}


void loop()
{
 // DO nothing
}



static void gorev_1(void* pvParameters)
{
 while(1)
 {
  int sen_1 = digitalRead(sen_1);
  if(sen_1==HIGH){
            do{turnLeft(50);}
            while(sen_3==HIGH);
            Serial.println(F("gorev_1"));
  }


  vTaskDelay(100/portTICK_PERIOD_MS);
 }
}

static void gorev_2(void* pvParameters)
{
 while(1)
 {
  int sen_2 = digitalRead(sen_2);
  if(sen_2==HIGH){
            do{turnRight(50);}
            while(sen_3==HIGH);
            Serial.println(F("gorev_2"));
  }
```

```
   vTaskDelay(100/portTICK_PERIOD_MS);
  }
}

static void gorev_3(void* pvParameters)
{
  while(1)
  {
    int sen_33 = digitalRead(sen_5);
     if(sen_33==HIGH){
       forward(70);
       digitalWrite(led,HIGH);
       }
       else
        {stop();
        digitalWrite(led,LOW);
        }
   //vTaskDelay(100/portTICK_PERIOD_MS);
  }
}

static void gorev_L(void* pvParameters)
{
  while(1)
  {
    int degerL  = analogRead(senL);
    if(degerL>=500){
                stop();
                Serial.println(F("gorevL"));
    }


    vTaskDelay(100/portTICK_PERIOD_MS);
  }
}

static void gorev_R(void* pvParameters)
{
  while(1)
  {
    int degerR  = analogRead(senR);
    if(degerR>=500){
                stop();
                Serial.println(F("gorevL"));
    }
```

```
   vTaskDelay(100/portTICK_PERIOD_MS);
  }
}

void forward(int a){
        digitalWrite(engineLeftA, HIGH);
        digitalWrite(engineLeftB, LOW);
        analogWrite(engineLeftP, a);
        digitalWrite(engineRightA, HIGH);
        digitalWrite(engineRightB, LOW);
        analogWrite(engineRightP, a);
 }

void backward(int a){
        digitalWrite(engineLeftA, LOW);
        digitalWrite(engineLeftB, HIGH);
        analogWrite(engineLeftP, a);
        digitalWrite(engineRightA, LOW);
        digitalWrite(engineRightB, HIGH);
        analogWrite(engineRightP, a);
 }
void stop(){
        digitalWrite(engineLeftA, LOW);
        digitalWrite(engineLeftB, LOW);
        analogWrite(engineLeftP, 0);
        digitalWrite(engineRightA, LOW);
        digitalWrite(engineRightB, LOW);
        analogWrite(engineRightP, 0);
 }
void turnLeft(int a){
        digitalWrite(engineLeftA,LOW);
        digitalWrite(engineLeftB,HIGH);
        analogWrite(engineLeftP,a);
        digitalWrite(engineRightA,HIGH);
        digitalWrite(engineRightB,LOW);
        analogWrite(engineRightP,a);
 }
void turnRight(int a){
        digitalWrite(engineLeftA,HIGH);
        digitalWrite(engineLeftB,LOW);
        analogWrite(engineLeftP,a);
        digitalWrite(engineRightA,LOW);
        digitalWrite(engineRightB,HIGH);
        analogWrite(engineRightP,a);
 }
```

ÖZGEÇMİŞ

Adı Soyadı : Ahmet Akif KÖSE

Doğum Yeri ve Yılı : Isparta, 1978

Medeni Hali : Evli

Yabancı Dili : İngilizce

E-posta : aakose@live.com

Eğitim Durumu

Lise : Isparta Teknik Lisesi, 1995

Lisans : Gazi, Teknik Eğitim Fakültesi, Elektrik Öğretmenliği, 1999

Mesleki Deneyim

Senirkent Endüstri Meslek Lisesi	1999-2001
Senirkent İşitme Engelliler Meslek Lisesi	2002-2008
Uluborlu Mesleki ve Teknik Anadolu Lisesi	2008-…..... (halen)