International Marmara Sciences Congress (Spring 2021)

21 – 22 May 2021 KOCAELI

PROCEEDINGS
E-BOOK

Organizer

Sponsors

This congress proceedings book is published as an electronic format as e-book.

All rights reserved.

Editorial Board

Assoc.Prof.Dr. Arif ÖZKAN
Assoc.Prof.Dr. Serdar BİROĞUL
Lect. Dr. Ömer GÜNGÖR
Lect. İsmail KILIÇASLAN
Lect. Kazım KAHRAMAN
Research Asist. Çağatay TAŞDEMİRCİ

ISBN: 978-605-70762-0-5

Address Kocaeli Üniversitesi Rektörlüğü, Umuttepe Yerleşkesi Kocaeli, TÜRKİYE
Phone +90 262 303 1000
E-mail info@imascon.com
Web www.imascon.com

All legal and ethical responsibility of the studies belongs to the authors. All rights reserved. The copyright of this proceedings book belongs to Kocaeli University. It may not be copied or reproduced without permission.

TÜRKİYE'DE ALINAN YASAL TEDBİRLER VE MEVZUAT DÜZENLEMELERİNİN KAZA VE ÖLÜM ORANLARI ÜZERİNDEKİ ETKİSİ 249

CİVATA ÇEKME DENEYİNDE KARŞILAŞILABİLECEK SORUNLAR VE ETKİLERİ 257

DÜŞÜK KARBONLU ÇELİKLERE UYGULANAN SERT DOLGU KAYNAĞINDA KAYNAK AKIMI OPTİMİZASYONU. 265

KAYMA KİPLİ KONTROL VE PI KONTROL YÖNTEMLERİ İLE TEK FAZLI TAM KÖPRÜ EVİRİCİNİN AKIM KONTROLÜ 272

MOBİL KONTROLLÜ VE HELİKS DİZİLİMLİ DÖNEL TİP FOTOTERAPİ CİHAZI TASARIMI 278

İZNİK GÖLÜ (BURSA)'NÜN DÜNÜ, BUGÜNÜ VE GELECEĞİ İLE İLGİLİ BAZI ÖNGÖRÜLER 286

ARAMA MOTORLARI İÇİN YENİ BİR WEB SAYFASI SIRALAMA ALGORİTMA YAKLAŞIMI 294

YENİLEBİLİR BİTKİ ŞEVKETİ BOSTAN (*SCOLYMUS HİSPANİCUS* L.): BESLEYİCİ, SAĞLIK ÖZELLİKLERİ VE KULLANIM ALANLARI 302

KOYUN VE KEÇİ AĞIZ SÜTÜNDEN BAKTERİYOSİN ÜRETİCİSİ LAKTİK ASİT BAKTERİLERİNİN İZOLASYONU VE MOLEKÜLER KARAKTERİZASYONU 308

KALIN CİDARLI HDPE MALZEMELERİN PLASTİK ENJEKSİYON YÖNTEMİYLE KALIPLANMASINDA SOĞUTMA TASARIMININ PARÇA KALİTESİ ÜZERİNE ETKİSİ 317

METİLPREDNİZOLON VE METİRAPON TEDAVİLERİNİN TRAVMA SONRASI STRES BOZUKLUĞUNDA NÖROKİMYASAL ETKİLERİ 331

UZAKTAN İZLENEBİLEN KUVÖZ TASARIMI 337

AN INVESTIGATION INTO THE SERVICE LIFE OF SEAL COATED ROADS IN TURKEY 344

SENTIMENT ANALYSIS ON COVID-19 VACCINES TWEETS 349

SOSYAL MEDYA PLATFORMLARINDA VERI GÜVENLIĞI ÜZERINE DEĞERLENDIRME: TWITTER ÖRNEĞI 358

BLOK ZİNCİRİ TEKNOLOJİSİ YAPISAL İNCELEMESİ VE GELİŞİM SÜRECİ 364

REVERS OSMOZ TESİSİNDE ÜRETİLEN SAF SU KALİTESİNİN UZAKTAN KONTROLÜ 371

TEDARİK ZİNCİRİNDE OTOMATİK TANIMA VE VERİ TOPLAMA SİSTEMLERİ 377

TÜRKİYE'DE DENİZCİLİK EĞİTİMİ VEREN MESLEK YÜKSEKOKULU VE FAKÜLTELERDE KULLANILAN KÖPRÜÜSTÜ SİMÜLASYON SİSTEMLERİNİN VERİMLİLİĞİNİN İNCELENMESİ 381

AKILLI TELEFONLA SENSÖRLERİN İZLENMESİ VE KONTROLÜ 388

30MNVS6 ÇELİĞİNİN RECKWALLS İŞLEMİ İLE SICAK DÖVÜLEBİLİRLİĞİNİN İNCELENMESİ 395

ÖRÜMCEK ROBOT İLE GÖRÜNTÜ İŞLEME VE NESNE TAKİBİ 403

VRF VE FAN COİL SİSTEMLERİNİN KULLANIM AMAÇLARI, AVANTAJLARI , DEZAVANTAJLARI VE SİSTEM KARŞILAŞTIRMALARI 414

BEŞİNCİ ALAN OLARAK SİBER UZAYDA TÜRKİYE'NİN HARP KABİLİYETİ VE HAZIRLIK DURUMU 422

ELEKTRİK SİGORTALARININ KONUMUNUN SIM800 GSM MODÜLÜ İLE UZAKTAN İZLENMESİ 427

GENETİK ALGORİTMA KULLANILARAK DOĞALGAZ ÇEVRİM SANTRALLERİNDE YÜK FREKANS KONTROLÜ 435

SANAL GERÇEKLİK TABANLI PEDİATRİK FİZİK TEDAVİ UYGULAMASI TASARIMI 443

INCONEL 625 SÜPERALAŞIMININ ALUMİNİD KAPLANMASI ÜZERİNE SICAKLIK ETKİSİ 443

İÇME SUYU DAĞITIM SİSTEMLERİNDE MEYDANA GELEN ARIZALARIN ANALİZİ 454

EŞİT KANAL AÇISAL PRESLEME VE YAŞLANDIRMA İŞLEMLERİ UYGULANAN AA 7075 ALAŞIMININ TALAŞLI İMALATINDA KESME KUVVETLERİNİN İNCELENMESİ 464

DÖKÜM CORSON ALAŞIMLARININ KOROZYON DAVRANIŞI ÜZERİNE ALAŞIM ELEMENTLERİNİN ETKİSİ 471

IMASCON SESSION 3.3 / 17:00 – 18:15	**Hall : 3**	**21 MAY 2021**
Remote Access Link:		
SESSION CHAIR : Assoc.Prof.Dr. Serdar BİROĞUL		
Assoc.Prof.Dr. Lyudmyla SYMOCHKO	Structural And Functional Successions Of Soil Microbiome In The Forest Ecosystems	
Irmak YILDIRIM	Sustainability In Waste Management	
Tuğsem SÖNMEZ	Formation Of An Urban Heat Island And The Effects Of Plants On Urban Climate	
Tuğsem SÖNMEZ	Effects Of Climate Parameters On Trees	
Tuğsem SÖNMEZ	Effects Of Climate Change On Arid-Semi Arid Areas And Afforestation Techniques	
Tuğçe Nur BACAK	Impact of the COVID-19 Event on PM10 Air Pollution in Istanbul and Ankara	
IMASCON SESSION 3.4 / 17:00 – 18:15	**Hall : 4**	**21 MAY 2021**
Remote Access Link:		
SESSION CHAIR : Assist.Prof.Dr. Merve DANDAN DOGANCI		
Tuğçe YÜKSEL	Preventing Silicate Deposite In Reinjection Wells In Geothermal Power Plants	
Tuğba DUMAN	Enzymatic Conversion Of Glycerin To Carbonate Ester	
Güler HASIRCI	Investigation Of The Use Of Catalytic Membranes In Chemical Reactions	
Murad RAHIMOV	Catalytic Treatment of Metal Cutting Fluid by Supercritical Water Oxidation	
Hakan SEVİNÇ	Investigation of Properties of Al_2O_3 Added Superhydrophobic Epoxy Films on Different Surfaces	
Discussion		
ORAL PRESENTATIONS		
21 May 2021	**SESSION 4**	**Time 18:30 – 19:45**
IMASCON SESSION 4.1 / 18:30 – 19:45	**Hall : 1**	**21 MAY 2021**
Remote Access Link:		
SESSION CHAIR : Assist.Prof.Dr. İbrahim MUTLU		
Muhammed SELMAN EREL	Performance Analysis Of Photovoltaic Systems In South East Region Cities Of Turkey Based On Meteorological Data	
Merve Yetimoğlu	Legal Measures Taken In Turkey And The Impact Of Legislative Regulations On Accident And Death Rates	
Muhammed ALİZADA	Monitoring And Control Of Sensors With Smart Phone	
Berat Eren TERZİOĞLU	Image Processing And Object Tracking Applications With Spider Robot	
Şuayp Karakaş	Energy Efficiency Applications in Chemical Fertilizer Factories	
Dr. Aykan Akça	N-N And N-H Bond Activations Of Hydrazine Molecule On Fe Embedded Graphene Surface: A Dft Study	

ÖRÜMCEK ROBOT İLE GÖRÜNTÜ İŞLEME VE NESNE TAKİBİ

Berat Eren TERZİOĞLU
Kocaeli Üniversitesi / Mühendislik Fakültesi / Elektronik ve Haberleşme Mühendisliği
beraterenterzi@gmail.com, ORCID: 0000-0002-9797-0235

Serhat YILMAZ
Kocaeli Üniversitesi / Mühendislik Fakültesi / Elektronik ve Haberleşme Mühendisliği
serhaty@kocaeli.edu.tr, ORCID: 0000-0001-9765-7225

Özet
Bu proje kapsamında Örümcek Robot ile Görüntü İşleme ve Nesne Takibi Uygulamalarının yapılması amaçlanmıştır. Amaç kapsamında robotun otonom hareketi, nesne tanıma ve nesne takibi uygulamalarıyla yapılmıştır. Görüntü işleme uygulaması Raspberry Pi 3 B+ kartıyla yapılmıştır. Görüntü işleme algoritması Python dilinde OpenCV kütüphanesi kullanılarak yapılmıştır. Motor kontrolleri Arduino geliştirme kartı ile yapılmıştır. Hareket kontrolü, Raspberry Pi kartından gelen nesne tespit verilerinin Arduino kartına aktarılmasıyla yapılmıştır. İki kartın haberleşmesi için UART haberleşme protokolü kullanılmıştır. Robot iskeleti, 4 kol üzerinde bulunan servo motorların belli bir sırayla çalışmasıyla ileri veya geri PWM sinyalleriyle birlikte hareket eder. PWM sinyalleri Raspberry Pi kartından Arduino kartına aktarılacak verilerle oluşturulur. Nesneyi tespit eden program servo motorları konumlandırarak harekete başlar. Takip edilecek nesnenin, kamera ekranındaki konumuna göre motor hareketleri şekillendirilir. Proje kapsamında kırmızı bir nesnenin tespit edilmesi ve takip edilmesi amaçlanmıştır. Bu kapsamda görüntü işleme algoritması kırmızı renkte cisimlerin tespiti ve takibini yapacaktır.

Anahtar Kelimeler: Görüntü işleme, OpenCv, Raspberry Pi, Arduino, PWM, Örümcek Robot, Servo motor, UART.

IMAGE PROCESSING AND OBJECT TRACKING APPLICATIONS WITH SPIDER ROBOT

Abstract
Within the scope of this project, it is aimed to perform Image Processing and Object Tracking Applications with Spider Robot.Within the scope of the purpose, the autonomous movement of the robot was made with object recognition and object tracking applications. Image processing application was made with Raspberry Pi 3 B + board. The image processing algorithm was made using the OpenCV library in Python language. Motor controls are made with Arduino development board. The motion control is done by transferring the object detection data coming from Raspberry Pi board to Arduino board. UART communication protocol is used for the communication of the two cards. The robot skeleton moves forward or backward with PWM signals, with the servo motors on the 4 arms operating in a certain order. PWM signals are created with the data to be transferred from the Raspberry Pi board to the Arduino board.The program that detects the object starts the motion by positioning the servo motors. Motor movements are shaped according to the position of the object to be tracked on the camera screen. Within the scope of the project, it was aimed to detect and track a red object.

Keywords: Image processing, OpenCv, Raspberry Pi, Arduino, PWM, Spider Robot, Servo-motor, UART.

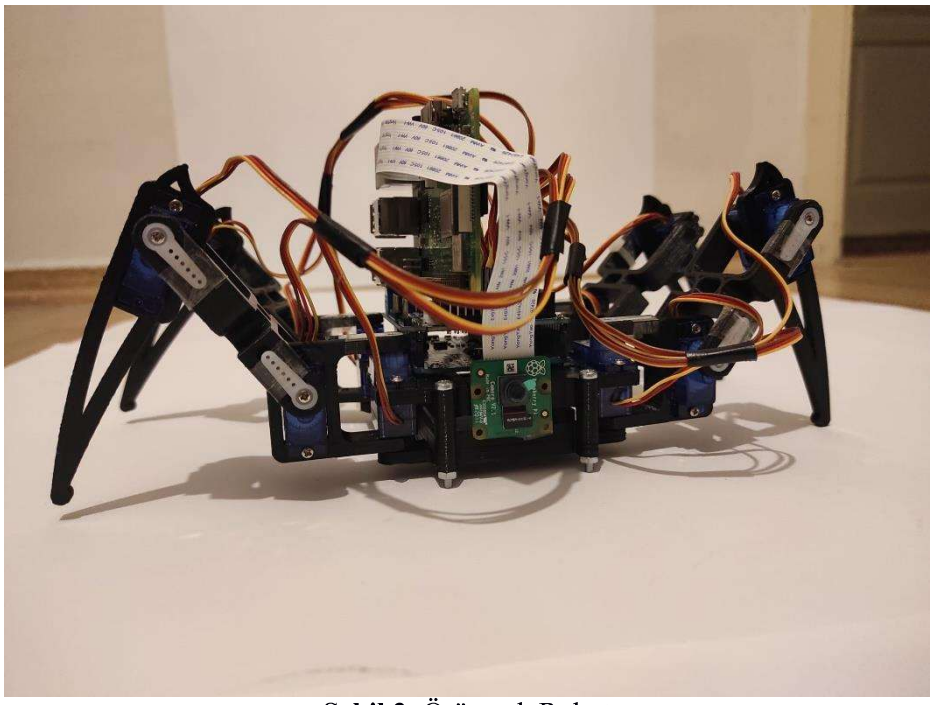
1. GİRİŞ

İnsansız araçlar, günümüzde endüstride ve günlük yaşamımızda yaygın bir şekilde kullanılmaktadır. Endüstride hızlı ve kesin sonuç almak için kullanılan insansız araçlar günlük yaşantımızı daha kolay bir hale getirmek için geliştirilmiştir. Buradan yola çıkarak temel anlamda otonom hareket edebilecek ve yapay zeka uygulamaları geliştirmeye müsait olan 4 kollu örümcek robot tasarımı uygun görülmüştür.

Bu projede Raspberry Pi 3 Model B+ kartı ve Arduino Uno geliştirme kartı ile 4 kollu örümcek robotun otonom olarak hareket etmesi ve bu hareketi kameradan alınan görüntüyü işleyerek gerçekleştirilmesi amaçlanmıştır. Otonom hareket sistemi bulanık mantık algoritması kullanılarak çalıştırılmıştır. Örümcek robot dört adet bacak ve bacakların bağlandığı bir adet gövdeden oluşmaktadır. Her bir bacak üç adet parçadan oluşmaktadır. Bacağı oluşturan parçalar birbirlerine servo motorlar ile ve her bir bacak gövdeye bir adet servo motor ile bağlanmaktadır. Servo motorlar örümcek için eklem görevi görmekte ve örümceğin hareketini sağlamaktadır. Her bir bacak için örümceğin gövdesine bağlı olan servo motora X, sonraki servo motora Y ve en uç kısımda bulunan servo motora Z isimleri verilmiştir. Örümceğe tepeden bakıldığı zaman sağ tarafta ve geride bulunan bacağa 1. Bacak, sağ tarafta ve ileride bulunan bacağa 2. Bacak olarak isimlendirilmiştir. Sol tarafta ve geride bulunan bacağa 3. Bacak ve sol tarafta ve ileride bulunan bacağa 4. Bacak olarak isimlendirilmiştir. Robot sensörün sol yanında bulunan 2. bacağını kaldırarak ileri hareket ettirir daha sonra bu hareketi diğer üç bacak tekrar eder. Bu hareket dizini tamamlandığında robot bir adım ilerlemiş olacaktır. Örümceğin hareketi mesafe sensörlerinden gelecek veriye göre şekillenmektedir. Engele olan uzaklığına göre göreceli olarak hızını değiştirmektedir.

2. MATERYAL VE METOD

Proje için Örümcek Robot dış donanımı, 12 adet servo motor, Raspberry Pi 3 B+, Arduino Uno, 2 Adet 3.7 Volt Li-on Pil, Arduino Shield, USB-Micro USB kablo, jumper kablo materyalleri kullanılmışır. Kartlar 7.4 Volt besleme gücüyle çalışır. Piller direkt olarak Arduino'ya bağlıdır. Raspberry Pi gücü Arduino üzerinde USB girişten alır.



Şekil 2: Örümcek Robot

2.1. Hareket Mekanizması

2.1.1 İleri Yönlü Hareket Mekanizması

Bacakta bulunan Y isimli servo motorun açısı belirli bir miktar azaltılır, Z isimli servo morun açısı belirli bir miktar azaltılır ve örümceğin bacağının yer ile olan teması kesilir. X isimli servo motorun açısı belirli bir miktar arttırılarak örümceğin bacağı ileriye doğru çevrilir. Y ve Z isimli servo motorların açısı başlangıç konumuna getirilerek bacağın yer ile olan teması sağlanır. Böylece örümceğin bir bacağı için bir adım tamamlanmış olur. Örümceğin ileriye doğru hareket etmesi için bacakların koordineli bir şekilde adım atmaları gerekir. İleri yönlü hareket için bacakların hareket sırası; 2 numaralı bacak, 1 numaralı bacak, 3 numaralı bacak ve 4 numaralı bacak şeklindedir.

2.1.2 Geri Yönlü Hareket Mekanizması

Bacakta bulunan Y isimli servo motorun açısı belirli bir miktar azaltılır, Z isimli servo morun açısı belirli bir miktar azaltılır ve örümceğin bacağının yer ile olan teması kesilir. X isimli servo motorun açısı belirli bir miktar arttırılarak örümceğin bacağı geriye doğru hareket ettirilir. Y ve Z isimli servo motorların açısı başlangıç konumuna getirilerek bacağın yer ile olan teması sağlanır. Böylece örümceğin bir bacağı için bir adım tamamlanmış olur.

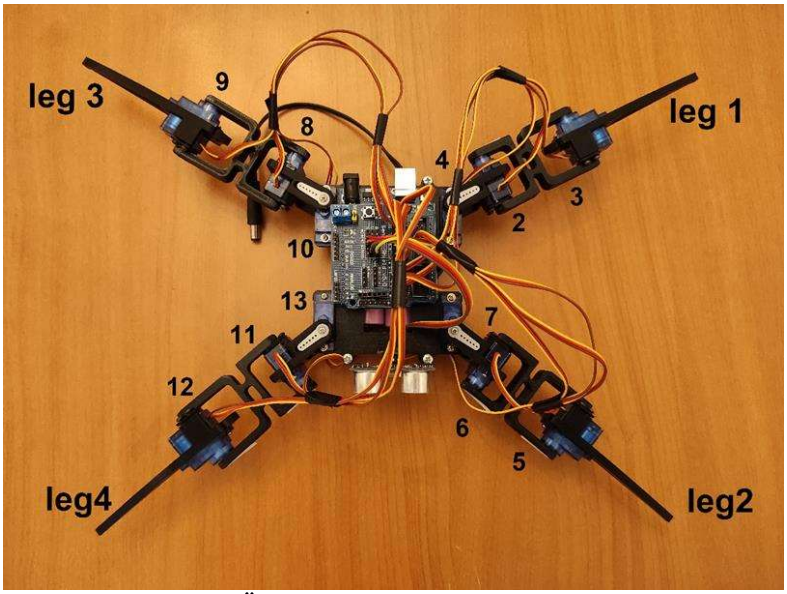
Örümceğin geriye doğru hareket etmesi için bacakların koordineli bir şekilde adım atmaları gerekir. Geri yönlü hareket için bacakların hareket sırası; 4 numaralı bacak, 3 numaralı bacak, 1 numaralı bacak ve 2 numaralı bacak şeklindedir.

2.1.3 Sağ Yönlü Hareket Mekanizması

Bacakta bulunan Y isimli servo motorun açısı belirli bir miktar azaltılır, Z isimli servo motorun açısı belirli bir miktar azaltılır ve böylece bacağın yer ile olan bağlantısı kesilir. X isimli servo motorunun saat yönünde dönecek şekilde açısı arttırılır. Y ve Z isimli servo motorları başlangıç açılarına getirilir. Bu sayede bir bacağın sağa dönüşü tamamlanmış olur. Örümceğin tam sağa doğru dönebilmesi için bu işlemler her bacak için üçer kez tekrarlanır. Sırasıyla hareket ettirilecek bacaklar; 2 numaralı bacak, 4 numaralı bacak, 3 numaralı bacak ve 1 numaralı bacaktır.

2.1.4 Sol Yönlü Hareket Mekanizması

Bacakta bulunan Y isimli servo motorun açısı belirli bir miktar azaltılır, Z isimli servo motorun açısı belirli bir miktar azaltılır ve böylece bacağın yer ile olan bağlantısı kesilir. X isimli servo motorunun açısı saat yönünün tersine doğru azaltılır. Y ve Z isimli servo motorları başlangıç açılarına getirilir. Bu sayede bir bacağın sola dönüşü tamamlanmış olur. Örümceğin tam sola doğru dönebilmesi için bu işlemler her bacak için üçer kez tekrarlanır. Sırasıyla hareket ettirilecek bacaklar; 4 numaralı bacak, 2 numaralı bacak, 1 numaralı bacak ve 3 numaralı bacaklardır.



Şekil 3: Örümcek Robot Motor Numaraları

2.2. Görüntü İşleme Uygulamaları

2.2.1 OpenCv Kütüphanesi Kurulumu

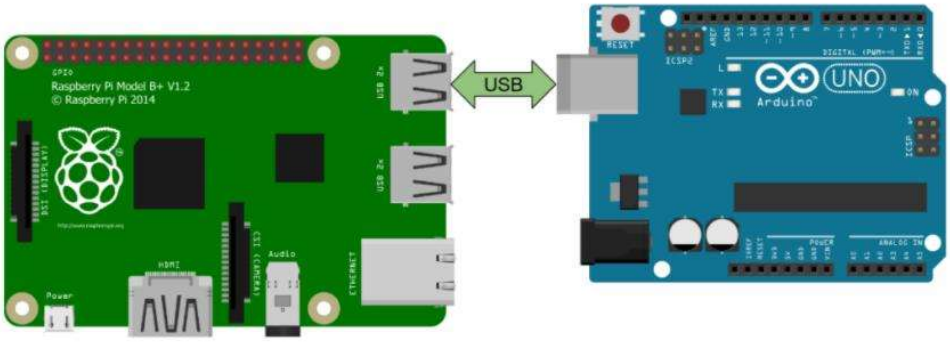
Görüntü işleme uygulamalarını gerçekleştirmek için öncelikle OpenCv kütüphanesi kurulması gereklidir. Raspberry Pi configuration ekranından OpenCv kütüphanesinin indirilmesi için gereken kodlar yazılmalıdır.

2.2.1.2 Raspberry Pi-Arduino Haberleşmesi İçin Gerekli Yazılımların Kurulumu

Raspberry Pi ve Arduino Uno Seri Haberleşme Yöntemi kullanılarak haberleştirilmiştir. Bu yöntemin kulanılabilmesi için Raspberry Pi configuration ekranında aşağıdaki kodların yazılması gerekir.

- sudo apt-get install arduino
- sudo apt-get install python-serial mercurial
- python3 -m pip install pyserial
- sudo apt-get install python3-pip

Kodlar yazıldığında Raspberry üzerinde Arduino için gerekli yazılım kurulmuş olacaktır. Donanımsal olarak kartların bağlantısı Şekil 3'te verildiği gibi kurulmalıdır.



Şekil 4: Raspberry Pi-Arduino Uno Bağlantısı

2.2.2 Yazılım Geliştirme

Yazılım geliştirme aşamasında görüntü işleme uygulaması için takip edilmesi gereken adımlar aşağıda verilmiştir. Örümcek Robot Projesi'nde amaç, renk skalası daha önceden belirlenmiş olan bir nesnenin kamera ekranındaki yerini tespit etmek ve bu nesnenin bulunduğu konuma göre servo motorlara PWM sinyalleri göndererek nesne takibini sağlamaktır.

Algoritma akışı aşağıdaki şekilde ilerleyecektir.

- Görüntüyü elde etme
- Ön görüntü işleme
- Nitelik tespiti
- Nesne sınıflandırılması
- Nesne takibi

2.2.2.1 Görüntünün Elde Edilmesi

Görüntü işlemeye başlamadan önce ilk olarak görüntünün yakalanması gerekmektedir. Herhangi bir kamera yardımıyla görüntü elde edilebilir. Görüntünün kalitesi, görüntü işleme operasyonunun daha temiz ve problemsiz olmasını sağlayacaktır. Bu proje kapsamında görüntüyü elde etmek için Raspberry Pi'ın kendi kamera modülü kullanılmıştır.

2.2.2.2 Ön Görüntü İşleme

Yakalanan görüntünün yapılması istenen çalışmaya uygun duruma getirilmesi için uygulanan yöntemlere ön işleme denir. Genel olarak belirli bir formülasyonu olmamakla beraber, deney ve test yoluyla gerçekleştirilir. Alınan görüntüdeki herhangi bir bozulma, kontrast dengesizliği, bulanıklık, veri kaybı gibi etmenleri ortadan kaldırmak için ön işleme yapmak gerekir. Görüntü en uygun koşullara gelene kadar ön işleme aşaması devam eder.Ön işleme tek bir yöntemle değil, birden fazla yöntemle yapılabilir. Görüntünün bozulmasına bağlı olarak birden fazla yöntem aynı anda kullanılabilmektedir.

2.2.2.3 Nitelik Tespiti

Ön işlemeden geçirilen görüntü üzerinde istenen özelliklere sahip özelliğin belirgin hale getirilmesi işlemine öznitelik çıkarımı denir. Önceden belirgin olmasını istediğimiz özelliğin otomatik olarak tespitini yapmak için bu işlemi uygularız. Bu proje kapsamında baktığımızda, robotun kırmızı renkli bir cismi tespit etmesi gerekmektedir. Kırmızı renkli olma durumu, burada cismin niteliği, özelliğidir. Kırmızı renk dışındaki bütün veriler göz ardı edilmiştir.

2.2.2.4 Nesne Sınıflandırması

Görüntülenen cisimlerin belirli özellikleri bulunmaktadır. Bu özelliklere göre işleyeceğimiz görüntüyü sınıflandırma işlemine tabii tutabiliriz. Bu sınıflandırma işlemi yapılırken nesnenin kendine özgü özellikleri kullanılmaktadır. Bu özelliklerin bir takım sayısal değerleri bulunmaktadır. Bu sayısal değerleri kullanarak, nesneleri sınıflandırabiliriz.

2.3 Görüntü İşleme Uygulamasının Python Dilinde Kodlanması

Görüntü İşleme Uygulamaları kodlama aşamasında algoritma oluşturulurken yukarıdaki adımlar izlenmiştir. Kod dizini OpenCv kütüphanesinin, numpy kütüphanesinin, ve nesne merkezini depolayacak veri tipinin kütühanelerinin import edilmesiyle başlar. Sonrasında nesne merkezini depolayacak veri tipi tanımlanır. Ardından kırmızı renk için RGB sklasında kırmızıya denk gelen en düşük ve en yüksek aralık tanımlanır. Kameradan görüntü alınması için OpenCy kütüphanesinden kamera kaydını başlatacak fonksiyon çağırılır. While döngüsü başlatılıp True değeri döndükçe while döngüsü içindeki işlemler yapılır. Bir if blok yapısı içinde blurlama, hsv, renk için maske oluşturma, maske etrafındaki görüntüleri slime işlemi yapılır. Sonrasında kontür adımına geçilir. Bu adıma geçildiğinde aynı if bloğu içinde kalınmalıdır. Kontür büyüklüğü sıfırdan büyük olduğunda cismi dikdörtgen olarak algılaması için OpenCv kütüphanesinden komut çağırılır. Cisim etrafına kutucuk oluşturması için kullanılan komut çağırılır. Cismin merkezini belli edecek komut çağırılır. Cismin merkezine pembe renkte bir nokta atanır. Kontür çizdirilir. Ve konum bilgisi output ekranına yazdırılır. Bu adımdan sonra cismin ekrandaki hareket bilgisini Arduino Uno kartına aktaracak kod bloğuna geçilir. Bu işlemlerden sonra uygulama Arduino üzerinden devam edecektir. Python kodlarına EK A' da yer verilmiştir. Algoritma akış şeması EK B'de verilmiştir.

2.4 Arduino Uno ile Motor Kontrolü Algoritması

Servo motorların sürülmesi Arduino Uno aracılığıyla sağlanmıştır. Öncelikle motorlara PWM sinyallerinin gönderilmesi için Raspberry Pi kartından nesne tespiti sinyallerinin gönderilmesi gerekmektedir. Öncelikle motorlar ve giriş pinleri tanımlanmalıdır. Sonrasında Raspberry Pi ile haberleşme için kullanılacak kodlar yazılmalıdır. Raspberry Pi'den gelen sayısal değerler bir değişkene atanmalı ve PWM sinyallerini bu değişkene göre oluşturacak fonksiyon tanımlamaları yapılmalıdır. Bu fonksiyonlar oluşturulurken dikkat edilmesi gereken hususlar Bölüm 2.1 'de verilen hareket mekanizmasının uygulanması işlemidir. Yön kontrolleri farklı PWM sinyalleri ile sağlanacağından her yön için farklı bir fonksiyon oluşturulmalıdır. Yön fonksiyonları oluşturulurken motor hızları da cismi takip edebilecek miktarda olmalıdır. Tüm fonksiyon blokları oluşturulurken if-else bloğu içinde olmalıdır. Kodlar derlenip karta yüklendikten sonra gerekli donanımsal bağlatılar robot üzerinde yapılır ve sistem çalışır hale getirilmiştir. Yatay düzlemde, takibi yapılan cismin kamera ekranı ara yüzünde konum bilgileri Raspberry Pi tarafından gönderilmiştir. Cisim merkezi yatay eksende 0-550 piksel aralığında değer alır. Buradan yola çıkarak kamera ekranı ara yüzünün merkezi referans noktası 0 değeri olarak belirlenir. Üyelik fonksiyonu ise -275 ve 275 aralığında değerler alır. Anlık konum bilgisinin 275 değerinden çıkarılmasıyla hata fonksiyonunu üretiriz. Hata fonksiyonu değerinin negatif veya pozitif bir değer çıkmasıyla cismin Örümcek Robot'a göre yönü tayin edilir. Negatif bir değer üretildiğinde cisim robota göre sağda, pozitif bir değer üretildiğinde cisim robota göre soldadır.

3. SONUÇ

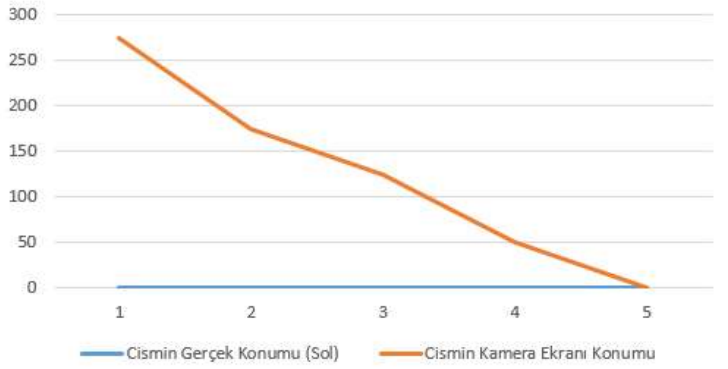
Örümcek Robot donanım parçaları adım adım monte edilerek birleştirilmiş ve teste hazır hale getirilmiştir. Görüntü işleme uygulamasının daha sağlıklı olması için kamera açısındaki bölgeler beyaz karton ile kaplanmıştır. Beyaz bölgelerle kaplanan alan içine uzaktan kırmızı bir cisim tutularak nesne algılama ve takibi işlemleri gerçekleştirilmiştir. Deney sırasında Raspberry Pi kartında yaşanılan ısınmadan ötürü çok zorlanmaması tavsiye edilir. Güç kaynağı olarak kullanılan 3.7V Li-on pillerin seri bağlanması ile elde edilen 7.4V'luk gerilim bu sorunun kaynağı olarak görülmektedir. Nesne Tanıma ve nesne takibi uygulamaları başarıyla sonuçlanmıştır. Uygulanan FIS algoritmalarına göre nesneye yönelim hareketleri gözlenmiştir. Nesne takibi sırasında servo motorların çalışmasında bazı aksaklıklar yaşanmıştır. Bu aksaklıkların Robot iskeleti üzerine düşen ağırlıktan ötürü olduğu düşünülmektedir. Deneyler az pürüzlü ortamda yapılmalıdır. Görüntü işleme uygulaması Raspberry Pi üzerinde başarılı bir şekilde yürütülmüştür. Şekil deney aşaması verilmiştir. Görüntünün daha net algılanması adına arka planda beyaz bir karton kullanılmıştır.

3.1 DENEYSEL SONUÇLAR

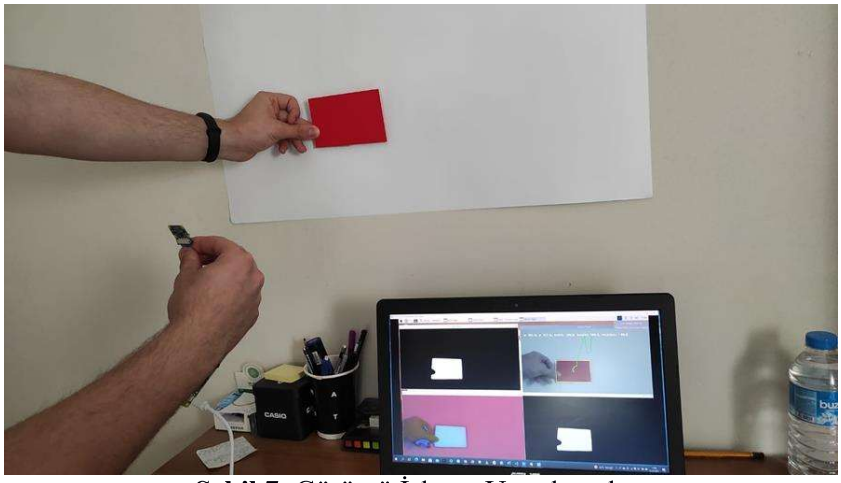
Raspberry Pi kartının çalışma yavaşlığından ötürü cismin yüksek hızda yer değiştirmesini takip etmekte zorluklar yaşanmıştır. Yavaş veya kesikli zaman aralıklarında yapılan yer değiştirmelerde takip sağlanmıştır. Robot kasasının ve motorların ağırlıktan ötürü zorlandığı tespit edilmiştir. Şekil 4 ve Şekil 5'te Örümcek Robot'un Cisime Zamana göre yaklaşım grafikleri verilmiştir.



Şekil 5: Zamana Göre Yatay Eksende Yaklaşım Grafiği (Cisim Sağdaysa)



Şekil 6:Zamana Göre Yaklaşım Grafiği (Cisim Soldaysa)



Şekil 7: Görüntü İşleme Uygulamaları

4. TARTIŞMA

Günümüzde İnsansız Araçların önemi günden güne artmaktadır. Gelişen teknoloji ile birlikte İnsansız Araçlar hem maliyeti düşürmesiyle hem de riskleri en aza indirmesiyle çok rağbet görmektedir. Bu

çalışmada temel olarak bir robotun otonom olarak çalışması amaçlanmıştır. Görüntü işleme uygulamaları ile birlikte robot kontrolü çok daha rahat bir şekilde yapılabilmektedir.

KAYNAKLAR

YILMAZ, S.(2020), KILCI, B.(2020), 6 Serbestlik Dereceli Sualtı Aracı ve Manipülatör Sistemi İle Görüntü İşleme Uygulamaları, *İleri Mühendislik Çalışmaları ve Teknolojileri Dergisi,*1(2), 63-79

REN, S. (2015), HE, K. (2015), GIRSICK, R. (2015), SUN, J. (2015), Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *Cornell University*

MALLICK, S. (2016), *Histogram of Oriented Gradients explained using OpenCV, 10 Mayıs 2021 tarihinde https://learnopencv.com/histogram-of-oriented-gradients/ adresinden alındı.*

YILMAZ S. (2007),” Bulanık Mantık ve Mühendislik Uygulamaları”, *KOU Yayınları*, Yayın No: 289

EK A

```python
import cv2
from picamera.array import PiRGBArray
from picamera import PiCamera
from imutils.video import VideoStream
import numpy as np
from collections import deque

# nesne merkezini depolayacak veri tipi
buffer_size = 16
pts = deque(maxlen = buffer_size)

# kirmizi renk aralığı HSV
redLower = (84,  98,  0)
redUpper = (179, 255, 255)

# capture
cap = cv2.VideoCapture(0)
cap.set(3,960)
cap.set(4,480)

while True:

    success, imgOriginal = cap.read()

    if success:

        # blur
        blurred = cv2.GaussianBlur(imgOriginal, (11,11), 0)

        # hsv
        hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)
        cv2.imshow("HSV Image",hsv)

        # kirmizi için maske oluştur
        mask = cv2.inRange(hsv, redLower, redUpper)
        cv2.imshow("mask Image",mask)
        # maskenin etrafında kalan gürültüleri sil
        mask = cv2.erode(mask, None, iterations = 2)
        mask = cv2.dilate(mask, None, iterations = 2)
        cv2.imshow("Mask + erozyon ve genisleme",mask)
```

Şekil 8: Görüntü İşleme Python Kodları 1. Kısım

```
        # kontur
        (contours,_) = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        center = None

        if len(contours) > 0:

            # en buyuk konturu al
            c = max(contours, key = cv2.contourArea)

            # dikdörtgene cevir
            rect = cv2.minAreaRect(c)

            ((x,y), (width,height), rotation) = rect

            s = "x: {}, y: {}, width: {}, height: {}, rotation: {}".format(np.round(x),np.round(y),np.round(width),np.round(height),np.round(rotation))
            print(s)

            # moment
            M = cv2.moments(c)
            center = (int(M["m10"]/M["m00"]), int(M["m01"]/M["m00"]))

            #haberleşme

            ser = serial.Serial('/dev/ttyUSB0',9600)
            kamera = x, g = width

            if g<150 :

                ser.write(b'1')

            elif g<300 and g>=150:

                if (kamera<200) :
                    ser.write(b'2')

                elif (kamera<=400 and kamera>200) :
                    ser.write(b'3')

                elif (kamera>400) :
                    ser.write(b'4')

            else:
                ser.write(b'5')
            # kutucuk
            box = cv2.boxPoints(rect)
            box = np.int64(box)
```

Şekil 9:Görüntü İşleme Python Kodları 2. Kısım

```
                # kutucuk
                box = cv2.boxPoints(rect)
                box = np.int64(box)

                # konturu çizdir: sarı
                cv2.drawContours(imgOriginal, [box], 0, (0,255,255),2)

                # merkeze bir tane nokta çizelim: pembe
                cv2.circle(imgOriginal, center, 5, (255,0,255),-1)

                # bilgileri ekrana yazdır
                cv2.putText(imgOriginal, s, (25,50), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255,255,255), 2)

                #b = "x: {}, y: {}".format(np.round( (int(M["m10"]/M["m00"]))),np.round(int(M["m01"]/M["m00"])))

            # deque
            pts.appendleft(center)

            for i in range(1, len(pts)):

                if pts[i-1] is None or pts[i] is None: continue

                cv2.line(imgOriginal, pts[i-1], pts[i],(0,255,0),3) #

            cv2.imshow("Orijinal Tespit",imgOriginal)

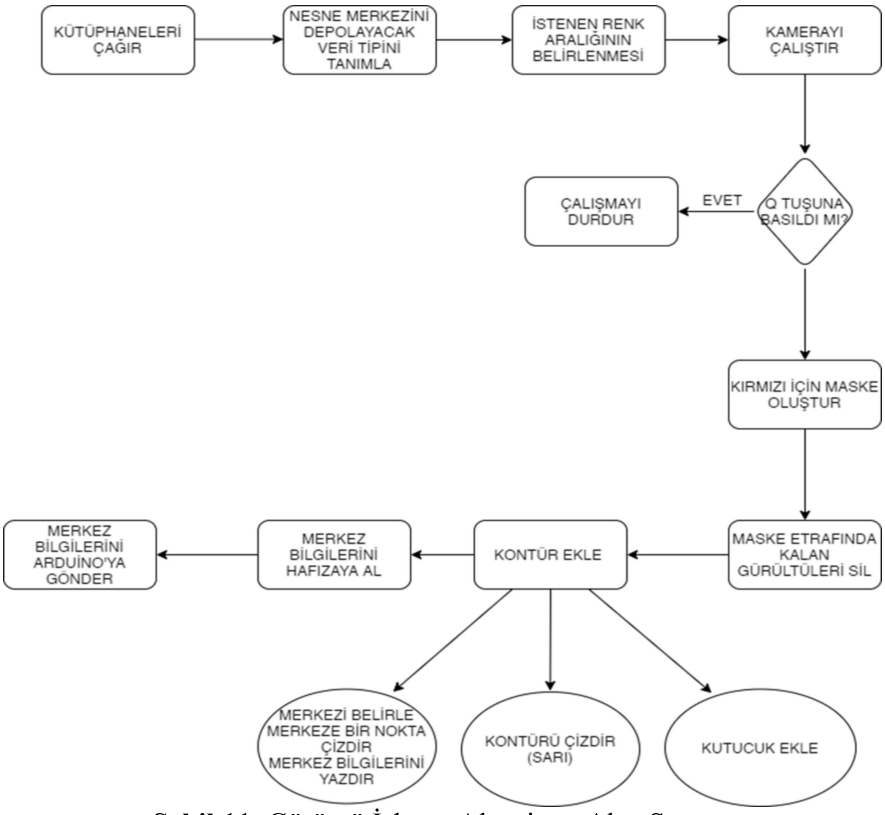
        if cv2.waitKey(1) & 0xFF == ord("q"): break

    cap.release()

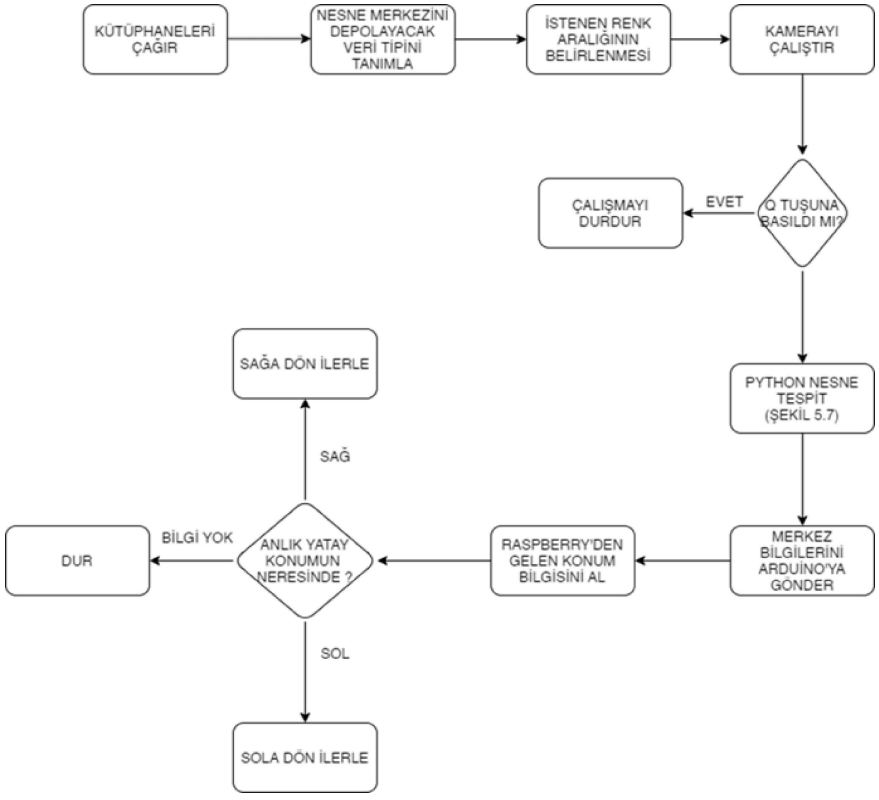

    cv2.destroyAllWindows()
```

Şekil 10: Görüntü İşleme Python Kodları 3. Kısım

EK B



Şekil 11: Görüntü İşleme Algoritma Akış Şeması



Şekil 12: Hareket Algoritması Akış Şeması

EK C

```
void loop()
{
    if(Serial.available())          //From RPi to Arduino
    {
      x = (Serial.read() - '0');    //conveting the value of chars to integer
      Serial.println(x);
    }

      if(x==1)
      {
        step_forward(4);
      }

    else if(x==3)
      {
        sit();

      }

    else if(x==4)
      {
        turn_left(3);
      }

    else if(x==2)
      {
        turn_right(3);
      }

    else if(x==5)
      {
        step_back(4);
      }
      delay(1000);
}
```

Şekil 13: Arduino Hareket Komut Kodları