

RoboCup RRL Manevra 2 Alanı için Otonom Gezinim Algoritması

Autonomous Navigation Algorithm for RoboCup RRL Maneuvering 2 Field

Mustafa Burak Dilaver, Furkan Çakmak, Erkan Uslu, Nihal Altuntaş, M. Fatih Amasyalı, Sırma Yavuz
Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul, Türkiye
mbdilaver@gmail.com {furkan, erkan, nihal, mfatih, sirma}@ce.yildiz.edu.tr

Özetçe —Bu çalışmanın amacı, dünyanın en önemli robotik yarışmalarından biri olan RoboCup yarışmasının Real Rescue Robot Ligi'nin MAN (Maneuvering) 2 adlı parkurunu otonom olarak tamamlayabilecek bir robot platformunun geliştirilmesidir. Sözü edilen parkur yer robotları içindir. Bu parkurda yarışan robotlar tekerlerinin ve paletlerinin genişliği sadece 5 cm'lik temas alanına sahip olduğu bir ray üzerinde gitmelidirler. Robotlar, ray üzerinde 5 cm sağa veya sola kaydığında raydan düşebilmektedir. Bu durum yarışma kuralları gereği robotların alanın başlangıç noktasına gitmelerine ve o zamana kadar bu alanda kazandıkları puanı kaybetmelerine sebep olmaktadır. Yarışmacılar rayların zeminden farklı renkte olmasından faydalanabilmektedir. Bu çalışmada kullanılan robot platformu, ROS (Robot Operating System) çatısı üzerine kurulmuştur. Robot üzerinde bulunan birden fazla kameradan alınan görüntüler kullanılarak hareket komutları üretilmektedir. Bu görüntülerin işlenebilmesi için OpenCV (Open Computer Vision) kütüphanesinden faydalanılmıştır. Bir raydan diğerine geçişin yapıldığı orta platformda ise kameradan alınan görüntülere ek olarak atalet duyargası (IMU - Inertial Measurement Unit) kullanılmıştır. Çalışma sonucunda ortaya çıkan yazılım elemanları benzer duyargaları barındıran ROS kurulu bir robot platformu üzerinde herhangi bir ekstra geliştirmeye gerek olmadan kullanılabilir.

Anahtar Kelimeler—RoboCup RRL, ROS, Manevra 2, Otonom Robot, Görüntü İşleme.

Abstract—The goal of this project is to develop a robot platform which can autonomously accomplish the track named Maneuvering 2 of the Real Rescue Robot League of one of the most important robotics contest of the world, RoboCup. The mentioned track is made for ground robots. Robots that compete in this track should move on rays which have a contact area that has 5 cm width with their wheels and tracks. Robots may fall when they slide left or right for 5 cm. Because of competition rules, this situation causes competitors to go to the starting point and lose the score that they've earned in this field until that time. Competitors may use the color difference between rays and grounds. The robot platform which is used in this study, is built on ROS framework. Movement commands are produced using the images that are taken by multiple cameras which are on the robot. OpenCV library is used to process these images. In addition to images that are taken by camera, IMU is used in the middle platform where the pass between two rays is made. The software components that generated as a result of this study can be used in a ROS installed robot platform, which had the similar sensors on it, without making any extra development.

Keywords—RoboCup RRL, ROS, Maneuvering 2, Autonomous

Robot, Image Processing.

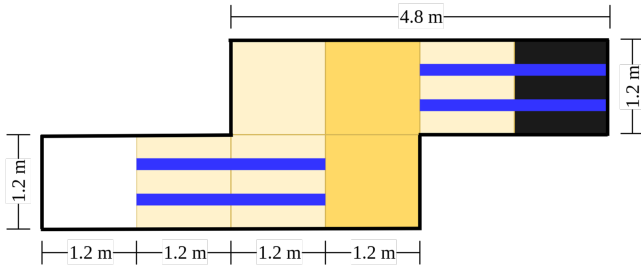
I. GİRİŞ

RoboCup, robotik ve yapay zeka çalışmalarını desteklemek üzere 1997 yılından beri her yıl yapılan uluslararası bir yarışmadır. RoboCup Soccer, RoboCup Rescue, RoboCup@Home, RoboCup@Work ve RoboCup Junior olmak üzere 5 ana kategoriden oluşmaktadır. Bu çalışmada, RoboCup Rescue kategorisinin Real Robot Ligi'nde (RRL) bulunan Manevra 2 (Maneuvering 2 - MAN2) alanını otonom olarak tamamlayabilecek bir robot platformu yazılım ve donanım elemanlarıyla gerçekleştirilmiştir. Laboratuvar ortamında MAN2 alanının benzeri kurulmuş ve testler bu alan içerisinde yapılmıştır.

II. ROBOCUP RRL MAN2 ALANI

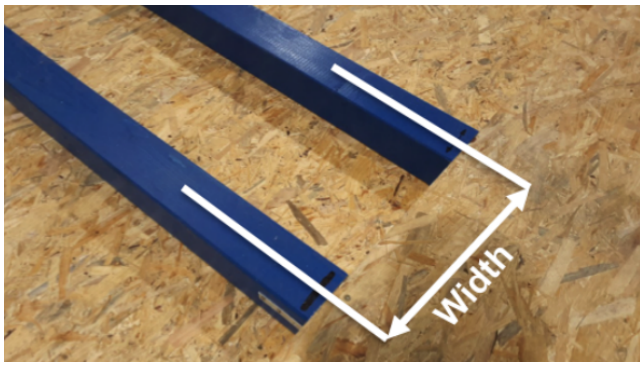
RoboCup yarışmasının Rescue kategorisinin Real Robot Ligi'nde toplam 20 adet parkur bulunmaktadır. Bu parkurlar Maneuvering, Mobility, Dexterity ve Exploration olmak üzere 4 gruba ayrılmıştır. Her grupta 5 parkur bulunmaktadır. Tüm MAN alanlarında yarışmacıların yarışmak için 20 dakika süreleri mevcuttur. Bu sürenin ilk 15 dakikası takımlar robotlarını elle teleop (tele-operated) kontrol etmektedir. Her parkurun bir başlangıç ve bir bitiş noktası vardır. Yarışmacılar 20 dakikalık süre içerisinde bu iki nokta arasında ne kadar çok gidip gelme hareketi yaparlarsa o kadar çok puan kazanmaktadır. Robotun operatör tarafından sürüldüğü 15 dakikalık süre içerisinde bitişten başlangıça dönerken robotun geri geri hareket etmesi gerekmektedir ve kalan 5 dakikalık zaman dilimi sadece otonom hareket eden robotlara verilmektedir. Bu da yarışmacılar için hem genel kategoride hem de otonom kategoride aldıkları puanı arttırmak için bir fırsattır.

Parkur planı Şekil 1 üzerinde görülebilmektedir. Parkurun etrafı 1.2 m yüksekliğinde sunta plakalar ile çevrilidir. Beyaz alan başlangıç, siyah alan bitiş alanıdır. Raylar mavi renk ile gösterilmiştir. Ortada bulunan koyu sarı alan orta alandır. Başlangıç alanı, raylar ve orta alan, diğer alanlardan yaklaşık 10 cm daha yüksektir. Başka bir deyişle açık sarı ve siyah alanlar diğer alanlardan 10 cm daha derindedir. Rayların genişliği de 10 cm'dir. Yarışacak robotların teker ve paletlerinin dış noktası rayların ortasına gelecek şekilde iki ray arası mesafe ayarlanmaktadır. Şekil 2 üzerinde bu ayarlama görülebilmektedir.



Şekil 1: Çalışmada da kullanılan MAN2 parkur planı

Yarışmacılar başlangıç noktasına vardıklarında robotları için 180 derece dönebilecek alan mevcut iken bitiş alanında geri gönebilmekleri bir alan mevcut değildir. Bu sebeple bitiş noktasından tekrar orta alana ulaşabilmek için geri yönde hareket etmeleri gerekmektedir.



Şekil 2: İki ray arası mesafenin robotun teker ve palet genişliğine göre ayarlanması

III. ROBOT PLATFORMU

Bu çalışmada kullanılan robot 4 tekerli ve diferansiyel sürüşlüdür. Ön, arka ve sağ tarafında bulunan 3 adet web kamera ile görüntü alınmaktadır. Üzerinde bulunan iki adet Arduino Uno'dan biri tekerleri döndüren motorları kontrol ederken diğeri kameranın altında bulunan servo motorları kontrol etmektedir. Yarışma sırasında işlemler robot üzerinde bulunan bir laptop üzerinde yapılmaktadır. Bu çalışmada ise robot platformunda içerisinde Raspbian Jessie işletim sistemi yüklü olan bir Raspberry Pi 3 bulunmaktadır. Raspberry Pi 3'ün bu çalışmadaki kullanımı duyargalardan alınan verileri işlem gücü daha yüksek bir uzak bilgisayara aktarmaktır. ROS aynı ağa bağlı bilgisayarlar arasında gerekli tüm veri akışını sağladığı için bu konuya ekstra vakit harcanması gerekmemektedir. Robot üzerinde bir adet atalet duyargası bulunmaktadır. Bu duyarga sayesinde robotun üç eksenle açısal pozisyonu bulunabilmektedir. Robota ait bir görsel Şekil 3 ile verilmiştir.

A. Donanım

Robot platformu iskelet, Raspberry Pi 3, 2 adet Arduino Uno, 2 adet motor sürücü, 4 adet DC motor, 2 adet servo motor, atalet duyargası, 3 adet web kamera, 2 adet 14.8V batarya ve gerekli regülatörlerden oluşmaktadır. Başlıca donanım elemanları alt başlıklar halinde incelenmiştir.



Şekil 3: Çalışmada kullanılan robot platformu

1) *Raspberry Pi 3*: Raspberry Pi 3, kredi kartı boyutunda bir bilgisayardır. 1.2 GHz 4 çekirdekli ARM işlemciye ve 1 GB RAM kapasitesine sahiptir. Kamera görüntülerinin işlenmesi ve hareket komutlarının üretilmesi Raspberry Pi3 üzerinde yapılmaktadır. Üzerinde Raspbian Jessie işletim sistemi ve ROS çalıştırılmaktadır.

2) *Arduino Uno*: DC motorların ve servo motorların kontrolü için birer adet Arduino Uno kullanılmıştır.

3) *Microstrain 3DM-GX25*: Robot üzerinde kullanılan atalet duyargasının modeli Microstrain 3DM-GX3-25'tir. 1000Hz frekansta çalışabilmektedir. Hareketliken 2 derece, sabitken 0.5 derece hata payına sahiptir. ROS ile uyumlu olarak çalışabilmektedir.

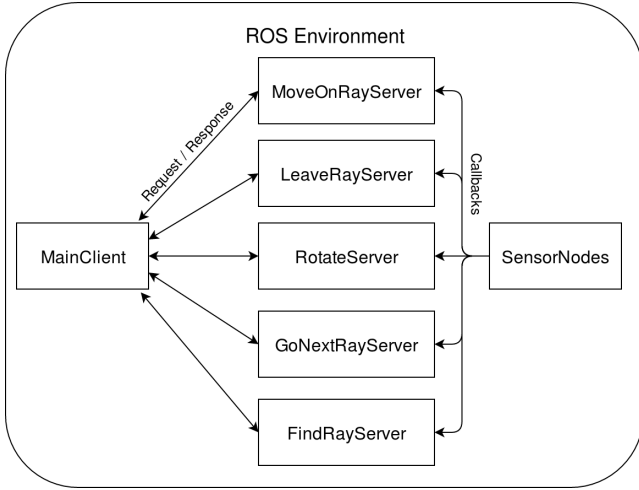
4) *Kameralar*: Kullanılan web kameraların tercih edilmesi için öne çıkan herhangi bir spesifik özellik bulunmamaktadır. Kameralar 15 FPS ve 320x240 çözünürlükte çalıştırılmıştır. Bu sebeple herhangi bir web kamera ile aynı işlemler yapılabilir. Özel olarak belirtmek gerekirse çalışmada kullanılan ön web kameranın modeli Logitech C270'tir ve 30 FPS ile 1280x720 çözünürlükte görüntü verebilmektedir [5]. Sağ ve arka kameralar Everest V69B marka ve modelidir.

5) *Robot iskeleti*: Robot iskeleti 4 tekerlidir ve diferansiyel sürüşlüdür. Tüm donanım elemanlarını taşıyabilecek alanlara sahip olduğu için seçilmiştir.

B. Yazılım

Robotun yazılım sistemi ROS üzerine kuruludur. ROS içerisinde kullanılan program parçalarına node denmektedir. Bu çalışmada her duyarga için o duyarganın verisini okuyan ayrı bir node kullanılmıştır. Her işlem adımı için bir action server oluşturulmuştur. Action server'lar kendilerine gelen isteği karşılık duyargalardan veri okur ve hareket komutu üretirler. Bu çalışmada aynı anda sadece bir server aktif olarak çalışmaktadır. Server kendisine istek gelmediğinde veya bir isteğe cevap gönderdiğinde artık pasif duruma geçer ve duyargalardan veri okumayı bırakır. Örneğin ray üzerinde hareket MoveOnRayServer adında bir server tarafından yapılırken, robotun kendi eksen etrafında dönüşünü RotateServer adında bir server yapmaktadır. Şekil 4 üzerinde çalışmada kullanılan

ROS ortamı şematize edilmiştir. Her server bir veya birden fazla duyarga verisi kullanmaktadır.



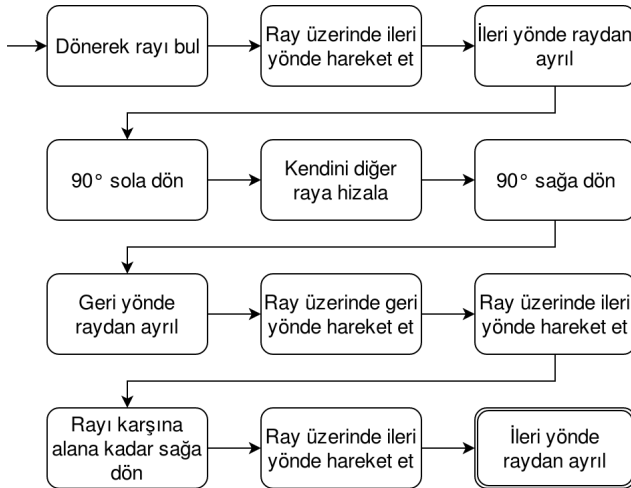
Şekil 4: Çalışmada kullanılan ROS ortamı

IV. KULLANILAN YÖNTEMLER

Bu bölümde MAN2 alanının başarıyla gezilebilmesi için kullanılan veri tipleri ve bu veriler üzerinde uygulanan işlem adımları incelenmiştir.

A. İşlem Adımları

Parkurun tamamlanması için izlenen işlem adımları Şekil 5 üzerinde görülebilmektedir. Robot bu adımları başarıyla tamamladığında bir tam tur atıp başlangıç noktasına dönmüş demektir. İlerleyen başlıklarda işlem adımlarının nasıl çalıştığı açıklanmıştır.

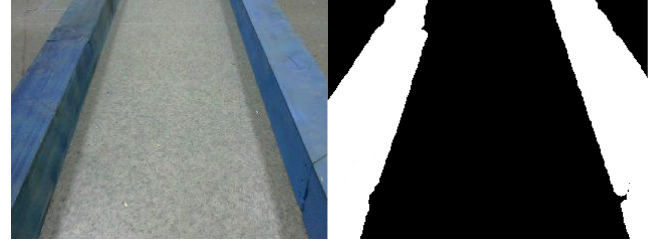


Şekil 5: Parkurda izlenen işlem adımları

B. Veri

Robot üzerinde bulunan kameralar 320x240 çözünürlükte ve 15 FPS (Frame Per Second) frekansında çalıştırılmıştır. Bu değerler deneysel olarak belirlenmiştir. Kameradan alınan

görüntüye filtreleme ve morfolojik işlemler uygulanmaktadır. Filtreleme aralığı el ile belirlenmektedir. Şekil 6 üzerinde işlenmemiş ve işlenmiş görüntüler görülebilmektedir.



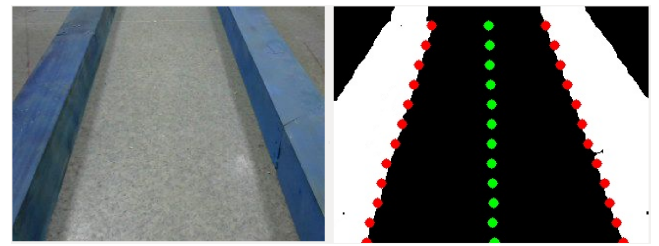
Şekil 6: Kameradan alınmış görüntü ve işlenmiş görüntü

Atalet duyargası robotun quaternion değerlerini verebilmektedir. Quaternion değerleri kullanılarak duyarganın ne kadar döndüğü ve dünyanın manyetik kuzeyi referans alınarak hangi açıda durduğu öğrenilebilmektedir. Bu çalışmada atalet duyargası 100 Hz frekansında çalıştırılmıştır ve her okumada 4 adet float tipinde değer işlenmektedir.

C. Ray Üzerinde İleri ve Geri Hareket

Ray üzerinde ileri gitmek için robotun önünde bulunan kameradan, geri gitmek için robotun arkasında bulunan kameradan alınan görüntülerden faydalanılmıştır.

Rayın hizasını bulabilmek için ekranın ortasında en alttan başlayarak 20 piksel aralıklarla görüntüden birer satır alınır. Ortadan başlayarak her satırın sol ve sağ beyaz değerleri bulunur. Bu noktanın koordinatı bir dizide kaydedilir. Bu dizide tutulan noktalar rayların iç yüzey noktalarıdır. Bu noktalar işaretlenir. Her satırda sol ve sağ noktanın orta noktası bulunur. Bu orta nokta toplulukları referans noktaları olarak kullanılır. Bu orta noktaların içerisinde bir adet referans noktası seçilir. Referans noktasının sütun değeri belirlenmiş bir aralığın içerisindeyse (görüntünün genişliği / 2 +/- eşik değeri) robota düz gitme komutu verilir. Nokta bu aralığın sağındaysa sağa, solundaysa sola dönme komutu üretilmektedir. Sözü edilen noktaların nasıl bulunduğu Şekil 7 üzerinde görülebilmektedir.



Şekil 7: Kameradan alınan görüntüden üretilen ray kontörlerini kullanarak (yeşil) referans noktalarının bulunması

D. Raydan Ayrılma

Robot raydan orta alana az bir mesafe kala ileri giderken ön kamerasında, geri gelirken arka kamerasında rayı görememektedir. Bunun sebebi ön ve arka kameraların dik bir şekilde yere bakmayıp açılı bir şekilde karşıya bakmasıdır. Bu sebeple robotun raydan ayrılırken hizasını koruyabilmesi

için ileri giderken arka kamerasından, geri giderken ön kamerasından faydalanılmaktadır. Burada bir üst başlıkta verilen komut üretme algoritması kullanılmaktadır. Robotun ilerlediği yönün tersindeki kamera kullanıldığından düz gitme komutu kullanılan kameranın tersi istikamette verilmektedir.

E. Dönme Hareketi

Robot orta alana çıktığında diğer raya ulaşması gerekmektedir. Bir raydan orta alana gelindiğinde robot 90 derece sola dönmeli, diğer raya kendini hizalayana kadar düz gitmeli ve ardından 90 derece sağa dönmelidir. Robotun dönme hareketi yapması gerektiğinde mevcut quaternion değerleri atalet duyargasından alınır. Bu değerler ile dönme miktarının quaternion karşılığı çarpılır. Bu işlem sonucunda robotun dönme işlemi bittiğinde sahip olması gereken quaternion değerleri bulunmuş olur. Dönme miktarı pozitif ise sola, negatif ise sağa dönülür. Buradaki dönme işlemi robotun olduğu yerde kendi eksenini etrafında dönmesi şeklindedir. Robot platformu diferansiyel sürüş dinamiğine sahip olduğu için kendi eksenini etrafında dönme hareketi yapabilmektedir.

F. Robotu Diğer Raya Hizalamak

Robot raydan ayrılıp orta platforma geldiğinde diğer raya gidebilmek için sola döndükten sonra düz gitme hareketi yapmaktadır. Robotun kendini diğer raya hizalayabilmesi için sağ tarafında bulunan kameradan alınan görüntülerden faydalanılmaktadır. Robotu raya hizalı götürmek için kullanılan algoritma burada da kullanılmaktadır. Robot kendini diğer raya hizayana kadar düz gitme hareketi yapmaktadır. Robotun kendini hizalamaya çalıştığı rayın orta noktası görüntünün ortasına geldiği zaman durulmakta ve 90 derece sağa dönme hareketi yapılacaktır.

V. UYGULAMA

Çalışmada kullanılan parkur yarışmada kullanılan parkur ile aynı ölçülere sahiptir. Test parkurunun planı Şekil 1 üzerinde görülebilmektedir. Bu çalışmada robot üzerindeki Raspberry Pi sadece verinin duyargalardan okunması ve uzak bir bilgisayara gönderilmesi için kullanılmıştır. Verinin işlenmesi bu uzak bilgisayarda yapılmıştır. ROS bu iki bilgisayar arasındaki veri iletişimini üstlenmektedir. Kullanıcılarına arka plandaki işlemleri yansıtmamaktadır.

VI. DENEYSEL SONUÇLAR

Çalışma sonucunda RoboCup Real Rescue Robot Ligi'nde bulunan MAN2 alanını otonom olarak gezen bir robot platformu oluşturulmuştur. Robotun ileri-geri ve dönme hızları belirtilerek elde edilen sonuç Tablo I ile verilmiştir. Burada verilen süre robotun bir turu tamamlama süresidir. Robotun başlangıç noktasından başlayarak tekrar başlangıç noktasına dönmesi bir tur olarak ifade edilmektedir.

Deney tablosu	Parametreler ve Sonuç		
	İleri-geri hız (m/s)	Dönme hızı (rad/s)	Süre (dd:ss)
	0.3	0.3	01:36

Tablo I: Deney parametreleri ve elde edilen süre

VII. SONUÇLAR VE DEĞERLENDİRME

Bu çalışmanın amacı RoboCup RRL içerisinde düzenlenen MAN2 alanını başarıyla tamamlayan bir robot platformunun geliştirilmesidir. Çalışma sonucunda alanı otonom olarak başarıyla gezen bir algoritma geliştirilebilmiştir. Bu çalışmada daha kısa sürede parkuru bitirmek gibi hedef belirlenmemiştir. İleride yapılacak çalışmalarda parkurun daha hızlı gidilmesini ve dolayısıyla yarışmada daha fazla puan alınmasını sağlayacak çalışmalar yapılabilir.

YAZAR KATKILARI

Mustafa Burak Dilaver çalışmanın sorumlu yazarı olup çalışmanın geliştirilme aşamalarından sorumludur. *Furkan Çakmak* algoritmanın geliştirilmesine katkı sağlamış, deney düzeneğinin kurulumunu ve bildirinin son düzenlemelerini yapmış ve *Erkan Uslu* ise algoritmanın geliştirilmesine katkı sağlamıştır.

TEŞEKKÜR

Bu çalışma, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu TÜBİTAK (EEEAG-113E212 Projesi ve BİDEB-2211/E Programı) ve Yıldız Teknik Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü YTÜ-BAPK (2015-04-01-GEP01 ve 2015-04-01-KAP01) tarafından desteklenmiştir.