

Hizmet Sektöründe Kafe veya Restoran için Yerleştirme ve Masaüstü Entegre GUI Uygulamasına Sahip Yapay Zeka Güdümlü Servis Robotu

Umair YOUNAS¹, Barış KOKULU², Aysel Banu TEKDAMAR², Bahtiyar Erener KURT², Akif DURDU¹

¹Elektrik Elektronik Mühendisliği Bölümü
Konya Teknik Üniversitesi
e178121001012@ktun.edu.tr
umair.ciitad@gmail.com

²Bilgisayar Mühendisliği Bölümü
Konya Teknik Üniversitesi
bariskokulu@gmail.com

²Bilgisayar Mühendisliği Bölümü
Konya Teknik Üniversitesi
abanu.tekdamar@gmail.com

²Bilgisayar Mühendisliği Bölümü
Konya Teknik Üniversitesi
bahtiyarkurt42@gmail.com

¹Elektrik Elektronik Mühendisliği Bölümü
Konya Teknik Üniversitesi
adurdu@ktun.edu.tr

Özet

Yapay Zeka ve Robotik Otomasyonun konaklama sektörüne entegrasyonu, daha yüksek verimlilik ve doğruluk, tekrar eden görevler için daha düşük işgücü maliyeti, iyileştirilmiş müşteri deneyimi, hijyen ve güvenlik ve teknoloji meraklısı müşterileri çekmek için teknolojik yenilikler gibi çeşitli faydalar sunmaktadır. Teknoloji, restoranların dijital dönüşümünde gelişmeye devam ederken, konaklama endüstrisinde yapay zeka güdümlü mobil robotların kapsamı önemlidir. Bu makale, bir kafede yemek dağıtım hizmetleri sunan masaüstü (GUI) entegre yapay zeka güdümlü bir mobil servis robotunun tasarımına ve uygulamasına odaklanmaktadır. Sunulan servis robotu, müşterilerin masaüstü GUI menüsünden yiyecek, içecek veya tatlı seçerek kolayca sipariş vermelerini sağlar. İlgili sipariş için seçilen ürünler, fiyat etiketi, tarih saati ve masa numarası ile birlikte makbuz oluşturulur. Servis robotu, ilgili masaya gitmek ve yiyecekleri FIFO sırasına göre servis etmek için ROS navigasyon yığını kullanır. Buna ek olarak, servis robotu, düşük pil seviyesini tespit ettiğinde robotun kendisini şarj istasyonuna yerleştirmesini sağlayan mekanizma ile donatılmıştır. Proje uygulamasında Jetson Nano ve RP-Lidar ile donatılmış TurtleBot Kobuki burger modeli kullanılmış, Ubuntu işletim sistemi, Python dil yorumlayıcısı kullanılmış ve sensör ve motorlarla ROS entegrasyonu sağlanmıştır.

Anahtar Kelimeler:

AI, Servis Robotu, Robotik İşletim Sistemi, Hizmet Endüstrisi, Navigasyon Yığını, Yerleştirme, GUI

1. Giriş

Yapay Zeka (AI) ve robotik otomasyonun entegrasyonu, 4. sanayi devriminde baskın bir rol oynamakta ve imalat, konaklama, sağlık, ulaşım, tarım ve finans gibi çeşitli sektörleri dönüştürmektedir. Yapay zeka ve robotik teknolojilerinin kullanılmasının konaklama sektörü üzerinde büyük bir etkisi vardır çünkü bir robot, tekrar eden bir görevi insan personelden daha verimli, daha hızlı ve tutarlı bir şekilde yerine getirebilir. [1], [2]. Ayrıca robotlar, Covid-19 pandemisinde yaşandığı gibi fiziksel teması en aza indirerek insanlar için güvenlik sunuyor [3]. Robotik otomasyonun yüksek üretkenlik, tutarlılık, işgücü dökümünün azaltılması ve insan güvenliğinin sağlanması gibi ödülleri, konaklama sektörüne entegrasyonuna yönelik gelişmeye olan ilgiyi artırmıştır. [4]. Yazarlar [5] restoran hizmetleri için navigasyon yığını kullanan bir insansı robot önermiş ve karesel programlama yaklaşımını kullanarak doğruluğu artırmıştır. Ancak, GUI arayüzü ve yerleştirme bu araştırmanın kapsamına dahil edilmemiştir. Yazarlar [6] mobil robotun kafe gibi karmaşık bir ortamda kendi kendini konumlandırmasını sağlamak için görüntü tanıma yöntemini değerlendirmiştir. Kamera ve lidar gibi çoklu sensörlerin kombinasyonu daha hızlı yanıt ve daha iyi doğruluk sağlar ancak zaman karmaşıklığını da artırır. Benzer navigasyon yığını tabanlı araştırma yazarlar tarafından [7] Bir fast-food restoranı için servis robotu tasarladı. Ancak, birden fazla sensörün entegrasyonu karmaşıklığı artırmakta ve GUI

arayüzü dikkate alınmamaktadır. Literatürde, servis robotu uygulamalarında önemli gelişmeler kaydedilmiştir. Hizmet endüstrisinde servis robotlarının umut verici ilerlemesine rağmen, servis robotunda tam otonom navigasyon, gerçek zamanlı ortamın karmaşıklığı nedeniyle zorlu bir görev olmaya devam etmektedir.

Sunulan hizmet robotu sistemi, sahte bir kafede otonom navigasyon, Gmapping ile haritalama, AMCL ile lokalizasyon, A* global planlayıcı, DWA yerel planlayıcı kullanarak masalar arasında navigasyon ve pil azaldığında şarj istasyonuna ulaşmak için otomatik kenetlenme uygulamak için bir TurtleBot ve ROS kullanır. Buna ek olarak, çalışmanın özgünlüğünü artıran GUI ve yerleştirme mekanizması ekler. Makalenin geri kalanı aşağıdaki şekilde düzenlenmiştir.

Bölüm 2'de sunulan araştırma metodolojisi açıklanmakta ve navigasyon yığını, masaüstü GUI ve kenetlenme mekanizması vurgulanmaktadır. Simülasyon kurulumu bölüm 3'te sunulmakta ve bunu bölüm 4'teki deneysel kurulum takip etmektedir. Sonuçların tartışılması bölüm 5'te gerçekleştirilmiştir. Son olarak, bölüm 6 sonuç tartışması, temel katkılar, sınırlamalar ve gelecekteki çalışmalar için gelecekteki yönergeler ile makaleyi sonuçlandırmaktadır.

2. Sunulan Araştırma Metodolojisi

Önceki araştırmalardan da anlaşılacağı üzere, konaklama endüstrisinde robot teknolojisinin ortaya çıkışı ve gelişimi göz önüne alındığında, sunulan araştırma, bir kafede masa hizmetleri sunan bir servis robotunun otonom davranışını daha da geliştirmek için bir adımdır. Sunulan araştırma, müşterilerden sipariş almak için bir masaüstü GUI ekleyerek yeni bir yaklaşım gerçekleştirmiş ve düşük pil seviyesini algıladığında kendisini şarj istasyonuna yerleştirmek için yerleştirme mekanizması da oluşturulmuştur.

2.1. Sunulan Sistem Mimarisi

Sunulan araştırmada, Turtlebot2 Kobuki kafe için bir hizmet robotu olarak, Python ROS1 çerçeve düğümlerini geliştirmek için, jetson nano geliştirici kiti (4GB), RP-Lidar AIM8 ve Logitech Gamepad F710 joystick olarak kullanılmıştır. Bu bölüm her bir unsurun açıklamasıyla devam etmektedir (bkz. Şekil 11)

2.1.1. ROS Çerçevesi

Bu çalışmada, modüler bir navigasyon sistemi oluşturmak için ROS1 Melodic ara yazılımı kullanılmıştır. Robotları kontrol etmek için yeni paketler geliştirerek veya mevcut olanları kullanarak genişletilebilecek bir temel yazılım seti sağlar [8]. Lazerler, kameralar, sonar, kızılötesi, çarpma sensörleri, lidar, radar, RGB kamera, derinlik kamerası, IMU ve GPS dahil olmak üzere çeşitli sensörler ROS tarafından desteklenmektedir. Bu sensörler paketi, robot navigasyonunu destekleyen bu sensörler için standart mesaj formatları sağlar.

2.1.2. Nvidia Jetson nano Geliştirici Kiti

Nvidia Jetson Nano Geliştirici Kiti (4GB) ROS melodik ile uyumludur ve kurulumunu destekler. Jetson nano, ROS'un robotik navigasyon algoritmalarını (python) çalıştırmasını ve GUI arayüzünü desteklemesini sağlayan bilgi işlem yeteneklerine sahiptir (ve etkili paralel işleme için [9]). Jetson nano'nun bu özellikleri onu konaklama sektöründeki otonom robotik uygulamaları için uygun hale getirmektedir.

2.1.3. ROS Çerçevesi için Python

Python, robotik uygulama tasarlamak, navigasyon algoritmalarını uygulamak ve bir GUI arayüzü oluşturmaya yardımcı olmak için programlama dili olarak kullanılır, bu nedenle Nvidia Jetson Nano modülünün ROS ile entegrasyonunda önemli bir rol oynar [10]. Python, sensör verilerini yayınlamak ve abone olmak ve kafedeki servis robotumuzda haritalama, yerleştirme, planlama ve kontrol yeteneklerini gerçekleştirmek için robot düğümlerinde hata ayıklamak için kullanılmıştır. Python programlama kullanılarak kontrol edilir.

2.1.4. Haritalama

Haritalama, sahte kafe ortamının statik 2D doluluk ızgara haritasını oluşturmak için bir lazer sensörü (RP-Lidar) kullanılarak gerçekleştirilmiştir. Harita oluşturma sırasında robotların hareketi klavye kullanılarak manuel olarak kontrol edilmiş ve deneme Logitech F710 joystick ile de test edilmiştir. Harita hem gazebo simülasyonu hem de sonuçlar ve tartışma bölümünde açıklanan pratik gerçek robot kullanılarak oluşturulmuştur.

1.6. Lokalizasyon

Lokalizasyon, lazer tarama ve statik harita verilerini kullanan bir robot navigasyonu için olasılıksal bir lokalizasyon tekniği olan AMCL'yi kullanarak robotun önceden oluşturulmuş bir harita içindeki konumunu tahmin etme işlemidir [11]. Lazer taramaları ve lazer haritaları ile çalışır, ancak işlevselliğini gerçekleştirmek için bir odometri mesajı gereklidir. amcl_pose düğümü, robotun eşlenen ortam içindeki konumunu belirlemeye yardımcı olur.

1.7. Navigasyon

ROS move_base paketi, harita ve yerleştirilmiş duruş üzerinde küresel ve yerel yol planlayıcısını yönetir, engellerden kaçınır ve algılama, yerleştirme ve hareketi birbirine bağlar [12]. Bu çalışmada, robotun mevcut konumundan hedef konumuna (restoran masası) yüksek seviyeli bir yol oluşturan küresel planlayıcı olarak A* kullanılmıştır. DWA ise düşük seviyeli engellerden kaçınma işlemlerini gerçekleştiren ve robotun navigasyon sırasında küresel yolu takip etmesini sağlayan yerel planlayıcı olarak kullanılmıştır [13].

2.2. Masaüstü GUI Entegre Uygulaması

Sunulan çalışmada, masaüstü GUI entegre uygulaması, söz konusu kafe ortamının yeniliklerinden biridir. Bu kafe sipariş uygulaması, Şekil 1'de gösterildiği gibi müşterilerin menü kenar çubuğu seçeneklerini görüntülemesine, siparişleri özelleştirmesine ve satın alma işlemi yapmasına olanak tanıyan Python ve Tkinter kullanılarak oluşturulmuştur [14].



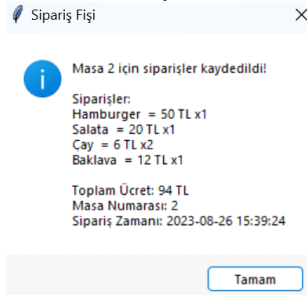
Şekil 1: RacLab Caf  için Python ve Tkinter tabanlı Sipariş Yerleştirme GUI Uygulaması.

Python Imaging Library (PIL) modülü görüntüleri işlemek için yüklenirken, datetime ve messagebox modülleri sırasıyla tarih ve iletişim kutusu açılır pencereleri sağlar.



Şekil 2: RacLab Cafe'nin GUI tabanlı menüsü

Özel işlevler menü kenar çubuğunu değiştirme, sipariş düğmeleri oluşturma, sepete seçim ekleme, toplam fiyatı hesaplama, sepeti temizleme ve siparişleri kaydetme gibi görevleri yerine getirir. GUI, tıkladığında sipariş edilecek öğeleri gösteren kategorize edilmiş bir menü görüntüler. Müşteriler sepete eklemek ve sipariş vermek için ürün düğmelerine tıklar. Ana Tkinter penceresi ve olay döngüsü, sipariş oluşturmak için etkileşimleri izler. Genel olarak, uygulamanın etkileşimli Python Tkinter tabanlı GUI arayüzü, müşterilerin menü kategorilerini ve öğelerini görüntülemesine, bir sipariş oluşturmaya ve kolayca satın almasına olanak tanır. Menü düğmesine tıkladığında, müşteriler yukarıdaki Şekil 2'de gösterildiği gibi Yiyecek, İçecek ve Tatlılardan oluşan yiyecek kategorilerine yönlendirilecektir. Müşteri siparişini bitirip Satın Al düğmesine bastığında, GUI Şekil 3'te gösterildiği gibi bir makbuz oluşturur



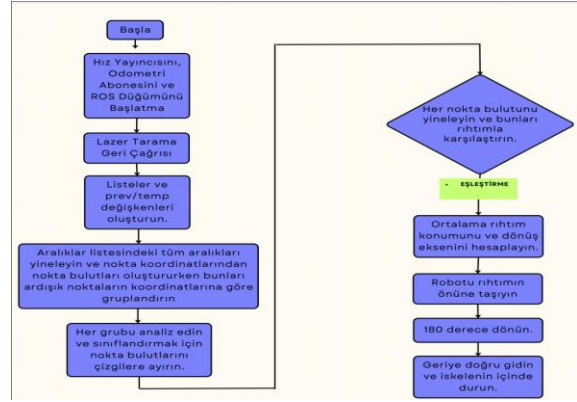
Şekil 3: Müşteri sipariş seçimine dayalı oluşturulan makbuz.

ve /cafe_service başlığı üzerinden siparişi ve masa numarasını belirten bir mesaj gönderir. Robot sipariş hazır olana kadar bekler, ardından siparişi alır ve /move_base düğümü üzerinden hedefleri kullanarak siparişi masaya getirir. Siparişi servis ettikten sonra kontrol istasyonuna geri döner.

2.3. Şarj İstasyonu için Otomatik Kenetlenme

Robotlarda otomatik kenetlenme, önemli bir görev olan düşük pil seviyesini algıladıktan sonra robotu şarj istasyonuna otonom olarak geri döndürme işlemidir [15] ve sunulan

araştırmanın iki yeniliğinden ikincisidir. Kenetlenme sürecindeki ana adımlar (a) yer işareti tasarımı, (b) Lidar veri toplama, (c) nokta bulutu sınıflandırması ve (d) bölme ve birleştirme algoritmasıdır. Algoritmanın akış şeması Şekil 4'te sunulmuştur.



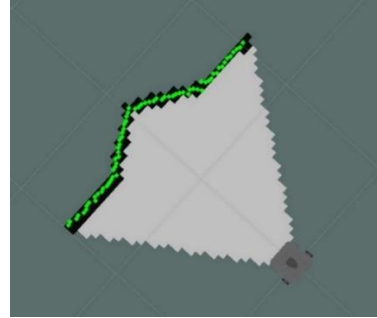
Şekil 4: Kenetlenme algoritması akış şeması

2.3.1. Şarj İstasyonunun Simgesel Tasarımı

Kenetlenme için kullanılan yer işareti sırasıyla 50, 58, 58 ve 50 cm genişliğinde dört kenardan oluşmaktadır. Merkez açı 120 derece, diğer iki açı ise 150 derecedir. İskelenin toplam uzunluğu yaklaşık 2 metredir.

2.3.2. Lazer Tarama Verilerinin Toplanması

Tarama konusundan alınan RP-Lidar sensör verileri, Şekil 5'te gösterildiği gibi dönüş noktası şeklimizi oluşturan çizgileri ayırarak rihthim tanımlamak için kullanılır. Geometrik analiz, rihthimin nereye ve nasıl baktığını belirlemek için kullanılır. Elde edilen konumu kullanarak robot, şarj istasyonuna mükemmel bir şekilde hizalanmak ve inmek için hareketleri hazırlar ve yürütür.

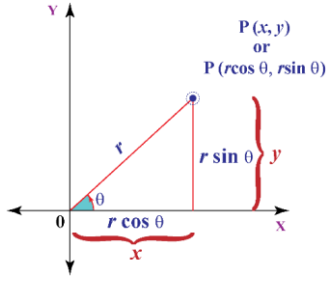


Şekil 5: Rihthim istasyonunda lazer noktalarının toplanması

2.3.3. Nokta Bulutu Sınıflandırması

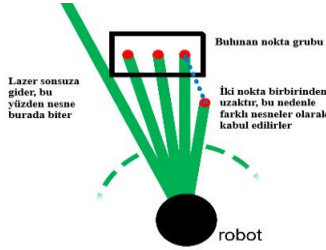
RP-Lidar sensörü /scan başlığından elde edilen menzil verilerini çıkarır ve menzilden lazer uç noktaları, Şekil 6'da gösterildiği gibi kutupsal koordinatlardan Kartezyen koordinatlara [16] trigonometrik dönüşümler gerçekleştirildikten sonra hesaplanır. Kutupsal koordinatlar açı ve menzil içerir. Lidar sensöründen menzil alınırken, lazerin açısı robotun dönüşü artı lazer indeksi çarpı lazerler arasındaki açı artışıdır. Bu açılar ve aralıklar ve robotun odometrisinden alınan robotun kendi koordinatları yardımıyla lazer uç

noktalarının küresel koordinatları bulunur. Yakındaki uzamsal konumlar sensör verilerinde ardışık noktalarla temsil edildiğinden, yakındaki tahmini lidar uç noktaları **Şekil 7**'de gösterildiği gibi bir araya getirilir.



Şekil 6: Kutupsal koordinatlardan Kartezyen koordinatlara trigonometrik dönüşüm

Süresizlik, bir sınır değerinden daha büyük bir mesafe ile gösterilir ve noktalar farklı gruplar halinde kümelenir.



Şekil 7: Lazer verilerinin gruplandırılması ve kümelenmesi

Bu gruplar farklı nesnelere veya yüzeylere ait noktalaradır.

2.3.4. Böl ve Birleştir Algoritması

Her nokta grubu analiz edilir ve Böl ve Birleştir algoritması kullanılarak doğrulara ayrılır. (a) En uzak iki nokta hesaplanır ve AB çizgisi çizilir, (b) [AB] çizgisinden en uzak nokta [P] belirlenir, (c) Noktadan çizgiye olan mesafe belirtilen eşikten büyükse, (d) [AP] ve [BP] çizgileri oluşturulur. Her iki çizgi için de ikinci adıma (b) gidin. Noktadan çizgiye olan mesafe belirtilen sınır değerinden küçükse, döngüyü kırın [17]. Son olarak, bulunan çizgiler ve aralarındaki açılar, gerçek veya simüle edilmiş bir ortamda anahtarın şekli tarafından tanımlananlarla karşılaştırılır.

2.3.5. İstasyona Yanaşma

Robot üç aşamada kenetlenir, (a) normal eksenine doğru döner ve kenetlenme yönüne hizalanır, (b) istenen kenetlenme mesafesine tamamen geri hareket eder ve (c) iniş hedefine ulaştığında yavaşça yavaşlar. Kenetleme dizisi, robotu şarj istasyonuna tamamen indirmek için algılanan konuma dayalı koordinat dönüşümlerini ve orantılı kontrolü kullanır.

3. Simülasyon Kurulumu

Şekil 8'de gösterildiği gibi 3D robotik Gazebo simülatörü kullanılarak sahte bir restoran ortamının simülasyon prototipi oluşturulmuştur. Gazebo simülasyonu, masalar, insanlar ve diğer rutin öğeler gibi statik nesnelere varlığında haritalama,

yerleştirme ve otonom navigasyonu test etmek için fizibilite sağlar [18]. Bu çalışmada, sandalye, çöp kovası veya saksı vb. üç ekstra engel göz önünde bulundurulmuş ve bu öğeler haritalama sürecinde statik engeller olarak işaretlenmiştir. Sunulan hizmet robotu tüm statik nesnelere kaçınır ve belirli bir hedefe doğru düzgün bir şekilde ilerler.



Şekil 8: RacLab kafenin Gazebo simülasyon modeli

Simülasyon ortamımızda altı masa (T1 ila T6) yerleştirilmiştir ve servis robotu hedef yol noktasına göre ilgili masalara doğru ilerlemektedir. Simülasyon, servis robotunun belirli bir hedefe yönelik isabetli navigasyonu ile sonuçlanır. Ayrıntılı açıklama sonuçlar ve tartışma bölümünde açıklanmıştır.

4. Deneysel Kurulum

Deney düzeneği, Şekil 9'da gösterildiği gibi laboratuvar ortamında prototip bir ortam oluşturularak kurulmuştur. prototip kafe 6 masa ile donatılmıştır. Her masa bir ahşap tahta ile temsil edilmektedir. Gazebo'da, insan görüntüleri statik engeller olarak yerleştirilirken, ahşap tahtalar masa olarak işaretlenir ve gerçek zamanlı deneyde haritalama işlemi sırasında statik engelleri almak için sinyal ışığı örnekleri yerleştirilir.



Şekil 9: Laboratuvar ortamında prototip bir ortam oluşturularak deney düzeneği kurulur.

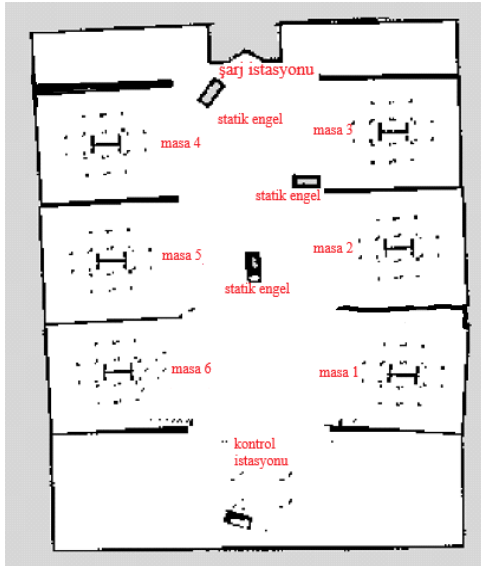
Sunulan deney, bir masaüstü GUI ve otomatik yerleştirme ekleyerek yeni bir yaklaşım sundu.

5. Sonuçlar ve Tartışma

Sunulan hizmet robotunun haritalama, konum belirleme, navigasyon, GUI tasarımı, yer işareti tasarımı ve kenetlenme gibi ön kavramları daha önce açıklanmıştır. Bu bölümde, öncelikle simülasyon sonuçları tartışılacak ve bunu deneysel sonuçlar takip edecektir. Hizmet robotunun otonom navigasyonunda, temel olarak kullanılan sensörler ve konular hem simülasyonda hem de gerçek zamanlı robot deneyinde neredeyse aynıdır. Örneğin, robotun konumu ve yönelimi /odom başlığından, lazer tarama veya RP-Lidar verileri /scan başlığından, doğrusal ve açısal komut hızları /cmd_vel başlığından (gerçek robotta cmd_vel_mux/input/navi idi) çıkarılır, /tf başlığı koordinat dönüşümü sunar, /move_base/goal başlığı robot için navigasyon hedefi belirler, /amcl_pose robotun tahmini konumunu yayınlar. Bu benzerlik, gerçek zamanlı deneyden önce Gazebo simülasyonunun iyi bir pratik olmasını sağlar.

5.1. Gazebo Simülasyon Sonuçları

Gazebo simülasyonunda, yukarıda belirtilen ön adımlar gerçekleştirildikten sonra, ana amaç servis robotunu kafe haritası içinde otonom olarak gezdirmektir. Gazebo simülasyonunda haritalama işlemi sırasında Şekil 9'da belirtilen ortam haritası oluşturulur ve Şekil 11'de gösterilir.



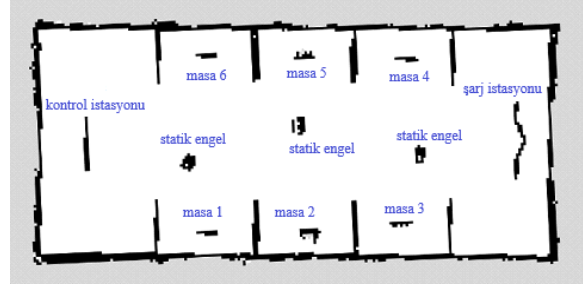
Şekil 10: Gazebo dünyasını kullanan RacLab kafe haritası.

Başlangıçta, konumu tahmin etmek ve robotun navigasyonunu gözlemlemek için harita RViz'e yüklenir. RViz aracılığıyla tüm masa konumları için bireysel manuel navigasyon hedefi sağlanır ve /odom ve /amcl_pose konuları kullanılarak konum ve yön çıkarılır ve bir listede saklanır. ROS'taki Move Base Action Client kütüphanesi kullanılarak, navigasyon hedeflerini (masalar) temsil eden ara noktaların listesi /move_base/goal içinde yayınlanır. Her hedef Move Base sunucusu tarafından incelenir ve yanıt olarak yerel ve küresel planlayıcı kullanılarak ilgili hedefe veya ara noktaya ulaşmak için bir yol oluşturulur. Gazebo'da, sahte kafeye üç kişi yerleştirilmiştir (bkz. Şekil. 9), bunlar bilgisayar kontrol istasyonu ve masaların konumu ile birlikte statik engeller olarak kabul

edilmiştir. Simülasyon testi birçok kez gerçekleştirilmiş ve robot A* global planlayıcı ve DWA yerel planlayıcı kullanarak hedef masaya yönlük isabetli bir yol oluşturmuştur. Hedefe (masa numarası) ulaşıktan sonra robot, yiyecek veya içeceklerin teslim süresi olan 10 saniye bekler ve ardından bir sonraki göreve veya hedefe geçer. Batarya azaldığında robot otomatik olarak şarj istasyonuna bağlanır. İstenilen batarya seviyesine ulaşıldığında hedef konumlarına göre navigasyonuna devam eder.

5.2. Gerçek Robot Üzerinde Pratik Deneysel Sonuçları

Bir kafenin gerçek bir temsilini oluşturmak için, laboratuvarımızın içine bir RacLab prototip kafe inşa edilmiştir (bkz. Şekil 10). Masalar ve kontrol bölümü ahşap tahtalarla temsil edilir ve bir sandalye, bir çöp kutusu veya bir saksı vb. olabilecek üç sinyal ışığı, haritada gösterildiği gibi statik engeller olarak eşleştirilir. Şekil 12.



Şekil 11: Gerçek zamanlı robot kullanarak harita oluşturma.

Gerçek zamanlı uygulama, Gazebo simülasyonlarından farklı olarak referans noktalarını seçmek için otomatik bir strateji kullanır. Müşteriler, gerçek zamanlı modelde yerleşik bir GUI masaüstü uygulamasını kullanarak menü siparişleri verebilirler. Daha sonra, seçilen ürünler için fiyat etiketleri görüntülenir, ardından kullanıcı bir ürün seçer ve "siparişi satın al" düğmesine tıklar. Bu olaydan sonra, ürünler, maliyetleri, masa numarası ve siparişin oluşturulduğu zaman hakkında tam bilgi içeren bir fiş otomatik olarak oluşturulur. ROS'taki Move Base Action Client modülü sayesinde /move_base/goal konusu, navigasyon hedeflerini (masa numaraları) temsil eden FIFO tabanlı bir ara nokta listesi alır. Bir masa numarasıyla belirtilen her hedef, robotun ara noktalara veya ilgili masaya ulaşmak için izlemesi gereken bir yol oluşturan Move Base sunucusu tarafından da değerlendirilir.

Laboratuvar testi sırasında, robot masaya ulaşıktan sonra, yemek dağıtım hizmeti süresi olarak adlandırılan 10 saniyelik bir bekleme süresi seçilir. Bu bekleme süresinin ardından robot bir sonraki sipariş masasına geçer ve süreci tekrarlar. Test aşamasında robot, müşterinin siparişine bağlı kalır ve her masada hassas bir şekilde gezinir, hem haritalanmış ortamdaki statik engellerden hem de yolunda karşılaşılan dinamik olarak görülebilen engellerden etkili bir şekilde kaçınır. Servis robotu düşük bir pil seviyesi tespit ettiğinde, robot otomatik olarak şarj istasyonuna hareket edecek ve kendisini istenen seviyeye şarj etmek için kenetlenecektir.

6. Sonuç ve Geleceğe Yönelik Öneriler

Hizmet endüstrisinde robotik otomasyonun önemi göz önüne alındığında, bu araştırma, müşterilere insan müdahalesi olmadan yiyecek, içecek, tatlı vb. servis etmek için bir

TurtleBot Kobuki servis robotu önerdi. Hedefe ulaşmak için hem simülasyon hem de deneysel kurulum oluşturulmuş ve başarıyla test edilmiştir. Sunulan servis robotu, masaüstü GUI uygulamasını entegre eder ve araştırmanın yenilikçiliğini artıran otomatik bir yerleştirme mekanizması oluşturur. Sonuçlar, müşterilerin GUI uygulaması aracılığıyla kolayca sipariş verebildiğini ve robotun ilgili masaya yemek servisi yapmak için FIFO kuralını izlediğini göstermektedir. Deneysel sonuçlar robotun yolundaki statik ve dinamik engellerden düzgün bir şekilde kaçınabildiğini ve hedef masaya başarıyla ulaşmak için navigasyon yığınının yerel ve küresel planlayıcısını doğru bir şekilde kullanabildiğini göstermektedir. Sunulan araştırma, artan hız ve doğruluk sayesinde daha yüksek üretkenlik, tekrarlayan görevler için daha düşük işçilik maliyeti, gelişmiş müşteri deneyimi, hijyen ve güvenlik ve teknoloji meraklısı müşterileri çekmek için teknolojik yenilikler gibi çeşitli faydalar sunmaktadır. Gelecekte birden fazla robotu entegre etmek için çalışmalar yapılabilir çünkü tek bir robot pil seviyesinin düşük olduğunu tespit ettiğinde şarj istasyonuna geçecek ve müşterilerin beklemesi gerekecektir. Ek olarak, sistem karmaşıklığını artıracak ancak doğruluğu da artıracak sensör füzyonu dahil edilebilir.

Referanslar

- [1] A. Khaliq, A. Waqas, Q. A. Nisar, S. Haider, and Z. Asghar, "Application of AI and robotics in hospitality sector: A resource gain and resource loss perspective," *Technol. Soc.*, vol. 68, no. November 2021, p. 101807, 2022, doi: 10.1016/j.techsoc.2021.101807.
- [2] W. & B. Inanov, "Adoption of robots and service automation by tourism and hospitality companies. Revista Turismo & Desenvolvimento," no. May, pp. 1501–1517, 2017.
- [3] Y. K. Dwivedi *et al.*, "Impact of COVID-19 pandemic on information management research and practice: Transforming education, work and life," *Int. J. Inf. Manage.*, vol. 55, no. July, p. 102211, 2020, doi: 10.1016/j.ijinfomgt.2020.102211.
- [4] K. H. Seo and J. H. Lee, "The emergence of service robots at restaurants: Integrating trust, perceived risk, and satisfaction," *Sustain.*, vol. 13, no. 8, 2021, doi: 10.3390/su13084431.
- [5] K. Hu, C. Ott, and D. Lee, "Online human walking imitation in task and joint space based on quadratic programming," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 3458–3464, 2014, doi: 10.1109/ICRA.2014.6907357.
- [6] A. C. Hernández, C. Gómez, J. Crespo, and R. Barber, "Object detection applied to indoor environments for mobile robot navigation," *Sensors (Switzerland)*, vol. 16, no. 8, 2016, doi: 10.3390/s16081180.
- [7] C. S. Chen, C. J. Lin, and C. C. Lai, "Non-Contact Service Robot Development in Fast-Food Restaurants," *IEEE Access*, vol. 10, pp. 31466–31479, 2022, doi: 10.1109/ACCESS.2022.3155661.
- [8] D. Herath and D. St. Onge, *Foundations of robotics : a multidisciplinary approach with Python and ROS.* .
- [9] R. Gandhinathan and L. Joseph, *ROS Robotics Projects: Build and control robots powered by the Robot Operating System, machine learning, and virtual reality*, 2nd Editio. Packt Publishing, 2020.
- [10] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS A Practical Introduction to the Robot Operating System*, vol. 53. 2015.
- [11] M. A. Chung and C. W. Lin, "An Improved Localization of Mobile Robotic System Based on AMCL Algorithm," *IEEE Sens. J.*, vol. 22, no. 1, 2022, doi: 10.1109/JSEN.2021.3126605.
- [12] K. Zheng, "ROS Navigation Tuning Guide," in *Studies in Computational Intelligence*, vol. 962, 2021.
- [13] A. Filotheou, E. Tsardoulis, A. Dimitriou, A. Symeonidis, and L. Petrou, "Quantitative and Qualitative Evaluation of ROS-Enabled Local and Global Planners in 2D Static Environments," *J. Intell. Robot. Syst. Theory Appl.*, vol. 98, no. 3–4, 2020, doi: 10.1007/s10846-019-01086-y.
- [14] I. Ahmad, Y. Iqbal, I. Haq, and S. U. Jan, "Pizza Ordering Management System," *SSRN Electron. J.*, 2023, doi: 10.2139/ssrn.4355330.
- [15] D. A. P. Wardhana, D. C. Happyanto, E. Purwanto, G. E. A. Akbar, and K. T. Putra, "AutoDock-IPS: An Automated Docking for Mobile Robot Based on Indoor Positioning System," *J. Electr. Technol. UMY*, vol. 5, no. 1, 2021, doi: 10.18196/jet.v5i1.12310.
- [16] S. Klinaku, "Galilean transformation in polar coordinates and Doppler effect," *Results Phys.*, vol. 31, 2021, doi: 10.1016/j.rinp.2021.104885.
- [17] Y. Zhu *et al.*, "Wheelchair Automatic Docking Method for Body-Separated Nursing Bed Based on Grid Map," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3084620.
- [18] A. Maryanto and C. Kesuma, "Sistem Informasi Pemesanan Makanan Dan Minuman Berbasis Web Pada Rumah Makan Gazebo Purwokerto," 2017.