



ITU ROVER YAZILIM ALT EKIBİ

Uygulama Ödevi

Feyza Nur Keleş

August 2024

Contents

1	ROS'un Temel Öğeleri	3
1.1	ROS Node:	3
1.1.1	ROS Master:	3
1.2	ROS Package:	3
1.3	ROS Topic:	3
1.4	Publisher ve Subscriber:	3
1.4.1	Publisher:	3
1.4.2	Subscriber:	3
1.5	ROS Launch:	3
1.6	ROS Service ve ROS Client:	4
1.6.1	ROS Service:	4
1.6.2	ROS Client (Service Client):	4
1.7	ROSBAG:	4
1.8	ROS2 ile ROS1 Arasındaki Farklar:	4
1.9	Gazebo ve RViz Nedir:	4
1.9.1	Gazebo:	4
1.9.2	RViz:	5
2	Sensörler	5
2.1	LiDAR Sensörü	5
2.2	IMU Sensörü	5
2.3	GNSS Sensörü	5
2.4	FPV Kamera	5
2.5	Derinlik Kamerası	5

2.6	IP Kamera	6
2.7	Enkoder	6
3	Otonom Sürüş	6
3.1	Path Planning, Motion Planning ve Trajectory Planning	6
3.1.1	Path Planning:	6
3.1.2	Motion Planning:	6
3.1.3	Trajectory Planning:	6
3.2	Costmap	7
3.3	Indoor ve Outdoor Otonom Sürüş Farklılıkları	7
3.3.1	Indoor otonom sürüş:	7
3.3.2	Outdoor otonom sürüş:	7
4	Robot Kol	7
4.1	İleri ve Ters Kinematik	7
4.1.1	İleri kinematik:	7
4.1.2	Ters kinematik:	7
4.2	Cartesian Space ve Joint Space Trajectory Planning	7
4.2.1	Cartesian Space Trajectory Planning:	7
4.2.2	Joint Space Trajectory Planning:	8
4.3	Serbestlik Derecesi	8
5	Kontrol Sistemleri	8
5.1	Kontrol Sistemi Nedir?	8
5.2	PID Kontrolü	8
5.3	Bir Kara Aracının Yokuşta Kaymadan Durabilmesi	8
5.4	State Machine	9
6	Lokalizasyon	9
6.1	Lokalizasyon Nedir?	9
6.2	Odometri Nedir?	9
6.3	Çöl Gibi Bir Arazide Lokalizasyon ve Odometri	9
7	SLAM	10
7.1	SLAM Nedir?	10
8	Görüntü İşleme	10
8.1	Görüntü İşleme Robotikte Ne İçin Kullanılabilir?	10
8.2	Fiducial Marker'lar Nedir?	10
8.3	Işık Koşullarının Değişmesi durumu:	10
9	Haberleşme	11
9.1	900MHz, 2.4GHz ve 5GHz Frekansları Arasındaki Farklar	11
9.2	Bant Genişliği Nedir?	11
10	Docker	11
10.1	Docker Nedir?	11
10.2	Docker'ın Avantajları	12
10.3	Docker kullanılmadığında	12

1 ROS'un Temel Öğeleri

1.1 ROS Node:

Birbirinden bağımsız çalışabilen, farklı görevleri yerine getiren bileşenlerdir. Birbirleriyle çeşitli yöntemlerle iletişim kurarak birlikte çalışırlar. Farklı node lar farklı dillerle de yazılabilir, farklı bilgisayarlarda da çalıştırılabilir. Node'ların ROS ağında birbirini bulmasını ve iletişimin koordine edilmesini ise ROS Master sağlar.

1.1.1 ROS Master:

Her Node'un adı, ağ adresi ve port bilgilerini kaydeder, aynı ve ya farklı yerlerde çalışan Node'lar arasındaki iletişimi sağlar. Aynı zamanda hangi Topic'in hangi Node tarafından yayınlandığını ve hangi Node'ların bu Topic'e abone olduğunu takip eder.

1.2 ROS Package:

ROS paketleri, belirli bir görevi yerinegetirmek için bir araya getirilmiş dosyalar ve kütüphanelerden oluşur. Bupaketler, robotların işlevselliğini modüler hale getirir.

1.3 ROS Topic:

Node'ların haberleşmesini sağlayan kanallardır. Belirli türde veri akışını sağlarlar. Veriyi alma ve okuma senkronize olmak zorunda değildir. Her Topic belirli bir mesaj türü ile tanımlanır ve bu Topic üzerinden yayınlanan tüm veriler bu türde olmalıdır. Hiyerarşik olarak düzenlenebilir ve farklı namespace'ler ile yayınlanır.

1.4 Publisher ve Subscriber:

1.4.1 Publisher:

Her publisher belirli türde veri üretir ve bunu yalnızca bir Topic'e iletir. Bir node birden fazla Publisher oluşturabilir.

1.4.2 Subscriber:

Her publisher belirli türde veri alabilir ve yalnızca bir Topic'e abone olur. Bir node birden fazla Publisher oluşturabilir. Her yeni mesaj geldiğinde callback function otomatik çağırılır.

1.5 ROS Launch:

XML tabanlı dosya formatıdır. Node'ların başlatılmasını izler ve terminal çıktısını ekranda gösterir. Birden fazla node'u başlatabilir ve aralarındaki bağlantıyı düzenleyebilir. ROS Node'larını başlatmak için kullanılan <node> etiketleri, ROS parametrelerini ayarlamak için <param> etiketleri, Topic adlarını yeniden haritalamak için <remap> etiketleri, Node'ları gruplayıp aynı anda başlatmak için <group> etiketleri ve diğer launch dosyalarını çağırmak için <include> etiketleri vb öğeleri bulundurabilir.

1.6 ROS Service ve ROS Client:

1.6.1 ROS Service:

ROS Service, senkronize bir iletişim mekanizması sağlar. Belirli bir hizmeti sunan ve client'tan gelen istekleri işleyen bir node'dur. Service tanımlamaları '.srv' dosyalarında yapılır ve bu dosyalar, istek ve yanıt mesajlarının formatını belirtir.

1.6.2 ROS Client (Service Client):

ROS Client, bir ROS Service'e istek gönderen ve yanıtı bekleyen node'dur. Hizmeti çağırmak için bir service proxy kullanır ve yanıt alındığında işlemi tamamlar. Bu süreç, senkronize bir şekilde çalışır, yani client, yanıtı alana kadar bekler.

1.7 ROSBAG:

ROSBAG, ROS'taki veri akışını kaydetmek ve yeniden oynatmak için kullanılan bir araçtır. ROS verilerini belirli bir süre boyunca toplamak ve bu veriler üzerinde analiz yapmak için kullanılır. ROS sistemlerinin performansını değerlendirmek ve test etmek için kullanılır.

1.8 ROS2 ile ROS1 Arasındaki Farklar:

ROS2, ROS1'in yeni özellikler eklenmiş bir sürümüdür.

- ROS2, iletişim altyapısı için DDS (Data Distribution Service) kullanır. ROS1 ise kendi iletişim protokolünü kullanır.
- ROS2, gerçek zamanlı sistemler için daha iyi destek sağlar. ROS1, gerçek zamanlı özelliklere sahip değildir.
- ROS2, Windows dahil olmak üzere daha geniş platform desteği sunar. ROS1, genellikle Unix tabanlı sistemlerle sınırlıdır.
- ROS2, node lifecycle yönetimi için geliştirilmiş özellikler sunar. ROS1'de bu tür bir destek mevcut değildir.
- ROS2, veri iletim kalitesini yönetmek için QoS (Quality of Service) politikaları sunar. ROS1'de böyle bir özellik bulunmamaktadır.

1.9 Gazebo ve RViz Nedir:

1.9.1 Gazebo:

Gazebo, ROS ile entegre olabilen bir robot simülasyon yazılımıdır. Robotların gerçek dünyadaki hareketlerini ve etkileşimlerini test etmeyi; robotlar, sensörler ve çevre ile ilgili 3D görselleştirmeyi sağlar. Robot modelleri ve çevreyi tanımlamak için çeşitli modüller ve bileşenler sunar. Ayrıca ROS topic'leri, services ve actions ile veri alışverişi yapabilir.

1.9.2 RViz:

RViz, ROS'un verilerini görselleştirmek için kullanılan bir araçtır. ROS topic'lerinden gelen verileri 2D ve 3D olarak görselleştirir. Kullanıcıların farklı veri türlerini ve görselleştirme seçeneklerini dinamik olarak değiştirmesine olanak tanır. Robot hareketlerini ve verilerini analiz etmek için çeşitli araçlar ve eklentiler sunar ve ROS'un verilerini doğrudan kullanarak görselleştirme yapar.

2 Sensörler

2.1 LiDAR Sensörü

LiDAR (Light Detection and Ranging), mesafe ölçümü yapılmasını sağlar. Yüksek hassasiyetlidir ve ışık darbelerini kullanan bir optik sensördür. Hedefe doğru ışık dalgaları gönderir ve bir objeye çarpıp geri dönme süresini ölçererek mesafe bilgisini hesaplar. Bu teknolojiye "Time of Flight" (TOF) denir. ROS (Robot Operating System) ile tam uyumluluk gösterir ve RVIZ gibi araçlar için hazır yapılandırmalar sunar. Livox MID40 gibi modeller, farklı FOV ve yansıtıcılık oranlarına göre 90 m ila 260 m arasında değişen algılama mesafeleri sunar.

2.2 IMU Sensörü

Inertial Measurement Unit (IMU), bir nesnenin hızlanma, açısal hız ve bazen manyetik alan gibi hareketle ilgili parametrelerini ölçen bir sensördür. IMU, genellikle üç eksen (X, Y, Z) çalışan ivmeölçer ve jiroskop bileşenlerinden oluşur. IMU verileri, sensör füzyon algoritmalarıyla işlenerek cihazın doğrultusu ve konumu hesaplanır.

2.3 GNSS Sensörü

Global Navigation Satellite System (GNSS), dünya üzerinde bir konum belirlemek için uydu aracılığıyla çalışan bir sistemdir. Çeşitli küresel konumlandırma sistemlerinden gelen sinyalleri alır. Bu sinyallerin farklı uydulardan ulaşma sürelerini ölçerek, kullanıcının üç boyutlu konumunu (enlem, boylam, yükseklik) belirler. Hassas konum belirleme, navigasyon ve zamanlama hizmetleri sunar.

2.4 FPV Kamera

First-Person View (FPV) kamera, gerçek zamanlı olarak video görüntülerini bir monitöre veya gözlüğe ileterek kullanıcının aracı ilk kişi perspektifinden kontrol etmesini sağlar. Düşük gecikme süreleri ve geniş görüş açısına sahiptir.

2.5 Derinlik Kamerası

Derinlik kamerası, bir sahnedeki nesnelerin uzaklığını ölçerek derinlik bilgisi sağlayan bir sensördür. Bu kameralar, genellikle kızılötesi ışık projeksiyonları ve stereo görüntüleme teknikleri kullanarak derinlik haritaları oluşturur.

2.6 IP Kamera

Internet Protocol (IP) kamera, video görüntülerini dijital olarak yakalayıp internet üzerinden ileten bir kameradır. IP kameralar, yerel ağ (LAN) veya geniş alan ağı (WAN) aracılığıyla uzaktan izleme ve kayıt yapma imkanı sunar. Yüksek çözünürlük, iki yönlü ses ve hareket algılama gibi özelliklere sahip olabilir.

2.7 Enkoder

Mekanik bir hareketi elektrik sinyallerine dönüştüren bir cihazdır. Pozisyon, hız veya dönüş miktarı gibi verileri dijital veya analog sinyaller olarak çıkış verir. Optik, manyetik veya kapasitif sensörler kullanarak çalışır.

3 Otonom Sürüş

3.1 Path Planning, Motion Planning ve Trajectory Planning

3.1.1 Path Planning:

Bir robotun başlangıç noktasından hedef noktasına çarpışmalardan kaçınarak en uygun yolu bulmasını sağlayan süreçtir.

- **A* (A Star):** Graf tabanlı bir algoritma olup, belirlenen hedefe en kısa ve en uygun yolu bulmaya çalışır.
- **Dijkstra:** A* algoritmasına benzer, ancak her yol maliyetini eşit kabul eder.
- **Rapidly-exploring Random Trees (RRT):** Robotun hareket edebileceği alanları rastgele keşfederek hedefe ulaşır.

3.1.2 Motion Planning:

path planning ile belirlenen yolu takip ederek robotun hareketini planlama sürecidir.

- **Pure Pursuit:** Robotun belirlenen yolu takip etmesi için basit ve etkili bir yöntemdir. Yol üzerindeki belirli bir noktaya odaklanarak rotayı takip eder.
- **Dynamic Window Approach (DWA):** Robotun anlık hız, yön ve çarpışma risklerini hesaplayarak uygun hareket komutlarını oluşturur.
- **Model Predictive Control (MPC):** Gelecekteki hareketlerini belirlemek için robotun dinamik modelini ve çevresel faktörleri dikkate alır.

3.1.3 Trajectory Planning:

Motion planning sürecinin bir parçası olarak, robotun belirli bir süre boyunca nasıl hareket edeceğini belirler. Amacı, robotun hareketini güvenli ve verimli bir şekilde zamanlamak ve kontrol etmektir.

3.2 Costmap

Bir robotun çevresindeki alanı maliyet değerleri ile temsil eden bir haritadır. Bu harita, robotun güvenli bir şekilde hareket etmesi için farklı bölgelerin geçiş maliyetlerini gösterir. Maliyetler, genellikle robotun çarpışma riskini, engellerin yoğunluğunu ve geçiş zorluklarını yansıtır.

Global Costmap, robotun tüm çevresini kapsayan geniş bir haritadır ve genellikle uzun vadeli planlama için kullanılır. Global costmap, statik engelleri ve geniş çevresel bilgileri içerir.

Local Costmap ise robotun yakın çevresini kapsayan daha küçük bir haritadır ve kısa vadeli planlama ve engellerden kaçınma için kullanılır. Local costmap, dinamik engeller ve anlık çevresel değişiklikleri içerir.

3.3 Indoor ve Outdoor Otonom Sürüş Farklılıkları

3.3.1 Indoor otonom sürüş:

Kapalı alanlarda gerçekleşir ve genellikle GPS sinyallerinin zayıf olduğu ortamlarda gerçekleştirilir. Bu nedenle, lidar, kamera ve ultrasonik sensörler gibi lokal sensörler yoğun bir şekilde kullanılır. Kapalı alanlarda daha fazla manevra kabiliyeti ve hassasiyet gereklidir, zira alanlar daha dar ve engeller daha yoğundur.

3.3.2 Outdoor otonom sürüş:

Açık alanlarda gerçekleşir ve GPS gibi global navigasyon sistemlerinden faydalanır. Bu tür sürüş, genellikle daha uzun mesafelerde ve değişken çevresel koşullarda yapılır. Bu nedenle, haritalama ve çevresel algılama sistemlerinin daha geniş bir kapsama alanı ve daha yüksek dayanıklılık sunması gerekir.

4 Robot Kol

4.1 İleri ve Ters Kinematik

4.1.1 İleri kinematik:

Bir robot kolunun eklemlerinin belirli pozisyonlarına göre uç efektörün (end-effector) pozisyonunu ve yönelimini hesaplama işlemidir. Bu işlem, robotun istenen bir pozisyonda işlem yapabilmesi için eklem açılarını belirlemek amacıyla kullanılır.

4.1.2 Ters kinematik:

Uç efektörün belirli bir pozisyona ulaşması için gerekli olan eklem açılarını hesaplama işlemidir. Ters kinematik, robot kolunun belirli bir hedefe ulaşmasını sağlamak amacıyla kullanılır ve genellikle ileri kinematiğe göre daha karmaşıktır.

4.2 Cartesian Space ve Joint Space Trajectory Planning

4.2.1 Cartesian Space Trajectory Planning:

Robot kolunun uç efektörünün kartezyen koordinat sisteminde (X, Y, Z) belirli bir yolu izlemesi için hareket planlaması yapar. Bu yöntem, genellikle doğrusal hareket ve belirli

bir yolu takip etmek için kullanılır. Avantajı, uç efektörün hassas bir şekilde kontrol edilmesini sağlar; dezavantajı ise, robotun eklem limitlerini dikkate almak zor olabilir.

4.2.2 Joint Space Trajectory Planning:

Robot kolunun her bir eklemının açısal hareketlerini planlar. Bu yöntem, genellikle robotun eklem hareketleri üzerinde doğrudan kontrol sağlar ve robotun tüm çalışma alanını etkili bir şekilde kullanmasını mümkün kılar. Avantajı, eklem limitlerine ve dinamik kısıtlamalara daha uygun olmasıdır; dezavantajı ise, uç efektörün izlemesi gereken yolu doğrudan kontrol etmek zor olabilir.

4.3 Serbestlik Derecesi

Serbestlik derecesi (DoF), bir robot kolunun bağımsız hareket edebileceği eksenlerin sayısını ifade eder. Bir robot kolunda her bir eklem, bir serbestlik derecesi sağlar. Örneğin, 6 serbestlik derecesine sahip bir robot kolu, 3D uzayda herhangi bir noktaya ulaşabilir ve herhangi bir yönde hareket edebilir.

5 Kontrol Sistemleri

5.1 Kontrol Sistemi Nedir?

Bir sistemin istenilen bir performans seviyesinde çalışmasını sağlamak için kullanılan düzenleme mekanizmalarıdır.

5.2 PID Kontrolü

PID (Proportional-Integral-Derivative) kontrolü, yaygın olarak kullanılan bir geri besleme kontrol yöntemidir. PID kontrolü, bir sistemin hata değerini minimize etmek için üç farklı bileşeni kullanır:

- **Proportional (P)** bileşeni, mevcut hataya orantılı bir tepki verir.
- **Integral (I)** bileşeni, zamanla biriken hatayı düzeltmek için tepki verir.
- **Derivative (D)** bileşeni, hatanın değişim hızına tepki verir ve sistemin aşırı tepki vermesini engeller.

PID kontrolü, sistemlerin hassasiyetini artırır ve kararlı bir performans sağlar. Özellikle robotik sistemlerde, hız, pozisyon ve basınç kontrolünde sıkça kullanılır.

5.3 Bir Kara Aracının Yokuşta Kaymadan Durabilmesi

Öncelikle tekerleklerin yol tutuşu yüksek olmalıdır. Çok iyi tutuşu varsa ani frenle de kaymadan durulabilir. Yine de kaygan/buzlu zeminlerde ani fren kaymaya sebep olabileceğinden hareketine bağlı olarak fren miktarını otonom ayarlayacak bir sistem kullanılması mantıklı olacaktır. Bu durum için de PID destekli fren sistemi kullanılabilir. Aracın eğim açısı ve kayma eğilimi sürekli izlenerek gerekli fren kuvvetini otomatik olarak uygulamasını sağlar.

5.4 State Machine

Bir sistemin farklı durumlar (states) arasında geçiş yapmasını sağlamak için oluşturulan bir modeldir. Her durum, belirli bir işlevi temsil eder ve istenilem etkenlere bağlı olarak bir durumdan diğerine geçiş yapar. Karmaşık işlemleri yönetmek ve sistemin mantıksal akışını kontrol etmek için kullanılır.

6 Lokalizasyon

6.1 Lokalizasyon Nedir?

Lokalizasyon, bir robotun kendi konumunu ve yönelimini çevresine göre belirleme işlemidir. Bu, robotun engelleri algılaması, belirli görevleri yerine getirmesi ve çevresel değişikliklere adapte olabilmesi için gereklidir. Farklı sensörlerden elde edilen verileri kullanarak robotun pozisyonunu belirleyecek yöntemler vardır. Slam ile çok karıştırdığım bi başlık gerçekten. Monte Carlo + G. Kalman Filtresi gibi algoritmalar kullanıldığını okudum ama araştırmalarımnda hep daha çok mapping ile ilgili sonuçlar aldım. Harita varsa haritadaki eğer yoksa çevresine göre konumunu belirler kısaca.

Mutlak Lokalizasyon: Bu yöntemde, robotun konumu, belirli bir referans noktasına veya harita üzerindeki kesin bir pozisyona göre belirlenir. Özellikle açık alanlarda ve geniş ölçekli haritalarda kullanılır. Bu yöntem için kullanılan başlıca GPS (uydulardan gelen sinyaller ile belirler) ve Manyometre (dünya üzerindeki manyetik alanını ölçerek yönelimini belirler) kullanılır.

Dead Reckoning: Bu yöntem, robotun kendi hareket bilgilerini kullanarak, bir önceki pozisyonuna göre yeni konumunu tahmin eder. Bu nedenle "göreceli lokalizasyon" olarak da adlandırılır. Ancak hata birikimi yaşanabileceğinden uzun süreli veya karmaşık hareketlerde doğruluğunu kaybedebilir. Dead reckoning için Odometri Sensörleri ve IMU ile robotun anlık hareketlerini daha hassas bir şekilde tahmin etmeye olanak tanır.

6.2 Odometri Nedir?

Bir robotun hareketini ölçmek için kullanılan bir tekniktir. Bu yöntem, robotun tekerleklerinin veya diğer hareketli parçalarının dönüşünü izleyerek kat edilen mesafeyi hesaplar.

Odometri'nin iki ana türü vardır:

- **Wheel Odometry:** Robotun tekerleklerinin dönüşü izlenir ve bu bilgi kullanılarak robotun kat ettiği mesafe hesaplanır.
- **Visual Odometry:** Kamera görüntüleri kullanılarak robotun hareketi izlenir ve bu görüntülerden çıkarılan bilgilerle robotun konumu hesaplanır.

6.3 Çöl Gibi Bir Arazide Lokalizasyon ve Odometri

Burada mümkünse GPS gibi sistemlerden yararlanarak lokalizasyon yapmak bence daha mantıklı olur çünkü kumlu ve birbirine çok benzeyen bir ortam olacak ve rüzgara vs bağlı olarak da hep değişeceği için görsel olarak işlenmesi daha zor olur. GPS ve wheel odometry kullanılarak konum takibi ve mapping daha mantıklı olur.

7 SLAM

7.1 SLAM Nedir?

SLAM (Simultaneous Localization and Mapping), bir robotun aynı anda hem kendi konumunu hem de çevresinin haritasını oluşturmalarını sağlayan bir tekniktir. SLAM, robotun çevresini algılayarak bir harita oluştururken, bu haritayı kullanarak robotun konumunu günceller. Bu süreç, lidar, kamera, ve ultrasonik sensörler gibi çeşitli sensör verilerini birleştirir.

Mesela GPS sistemine erişimi imkanı olmayan iç mekanlarda SLAM ile sorun çözülmeye çalışılmaktadır. Mono/stereo kameralar, lidar, lazer veya sonar sensörler ile robotun hareketine bağlı olarak mapping yapması mümkündür. Sadece lazer veya sonar ile yapılan yöntemlerde robotun ortamın görsel özelliklerini öğrenmesi ve aynı/farklı ortamları birbirinden ayırması zorlaşabilmektedir. Büyük ortamlarda tercih edilmez. Görsel işlemede ise mono kamera derinliği anlamaya yeterli olmayacağından stereo kullanılır ve iki görüntü arasındaki açı farkı hesabından derinlik bulunur. Ama rgb değerlerine göre yapılacak bu hesaplama çok ağır ilerler lidar sensörle yapılabileceği kıyasla. Yine de doğruluk açısından daha iyi diye anladım. Bu sitede çok güzel özetlenmiş.

8 Görüntü İşleme

8.1 Görüntü İşleme Robotikte Ne İçin Kullanılabilir?

Robotikte **görüntü işleme**, robotların çevresini algılaması, tanıması ve bu çevrede hareket etmesi için kullanılır. Görüntü işleme, robotun kameralar aracılığıyla aldığı görüntüleri analiz ederek, nesneleri tanımasına, engelleri algılamasına ve belirli görevleri yerine getirmesine yardımcı olur. Örneğin, robotların insan yüzlerini tanıması, nesneleri sınıflandırması, ve çevresel değişikliklere tepki vermesi için görüntü işleme algoritmaları kullanılır.

8.2 Fiducial Marker'lar Nedir?

Robotların, çevresindeki nesneleri tanımasına ve konumlarını belirlemesine yardımcı olan özel işaretlerdir. Bu marker'lar, genellikle belirli bir desen veya şekle sahiptir ve robotun kamerası tarafından kolayca tanınabilir. Fiducial marker'lar, robotların navigasyon ve nesne tanıma görevlerinde kullanılır. Örneğin, bir robot, fiducial marker'ları kullanarak belirli bir noktaya gitmek veya belirli bir nesneyi tanımak için bu marker'lardan faydalanabilir.

8.3 Işık Koşullarının Değişmesi durumu:

Düşük ışık durumunda problem olmaması için gece görüş desteği gibi teknolojiler kullanılabilir. Markerların daha reflektif olması sağlanabilir. Robota ışık kaynağı konabilir. Renk tonu problemi yaşamamak adına siyah beyaz markerlar tercih edilebilir.

9 Haberleşme

9.1 900MHz, 2.4GHz ve 5GHz Frekansları Arasındaki Farklar

Bu üç frekans bandı, farklı kablosuz iletişim uygulamalarında kullanılır ve her birinin kendine özgü özellikleri vardır:

- **900MHz:** Daha uzun menzilli ve engelleri aşmada daha etkili bir frekans bandıdır. Ancak, veri iletim hızı daha düşüktür ve bant genişliği sınırlıdır. Genellikle uzun mesafeli haberleşme ve düşük veri hızına ihtiyaç duyulan uygulamalarda kullanılır.
- **2.4GHz:** Orta menzilli bir frekans bandıdır ve daha yüksek veri iletim hızı sunar. Wi-Fi ve Bluetooth gibi birçok kablosuz iletişim protokolü bu frekansta çalışır. Dezavantajı, bu bantta çok fazla cihaz çalıştığı için parazit olasılığının yüksek olmasıdır.
- **5GHz:** Daha yüksek veri iletim hızı ve daha geniş bant genişliği sunan bir frekans bandıdır. Ancak, menzili daha kısa ve engelleri aşma kapasitesi daha düşüktür. Yoğun veri gerektiren uygulamalar için idealdir, ancak kısa mesafelerde kullanılır.

9.2 Bant Genişliği Nedir?

Bir iletişim kanalının belirli bir süre içinde iletebileceği veri miktarını ifade eder. Bant genişliği, iletişim hızını belirleyen önemli bir faktördür; geniş bir bant genişliği, daha yüksek veri iletim hızlarına izin verir. Bant genişlikleri arasındaki farklar, veri iletim hızını, sinyalin kalitesini ve mesafeyi etkileyebilir. Örneğin, dar bir bant genişliği daha düşük hız ve daha az veri kapasitesi sunarken, geniş bir bant genişliği daha yüksek hız ve daha fazla veri kapasitesi sağlar.

10 Docker

10.1 Docker Nedir?

Docker, yazılım uygulamalarını ve bu uygulamaların çalışması için gereken bağımlılıkları **container** adı verilen izole bir ortamda çalıştıran bir uygulamadır. Docker'ı, sıfırdan temiz bir bilgisayar olarak düşünebilirsiniz.

Docker'ın Bileşenleri

- **Docker Daemon:** Docker işlemlerini yöneten arka plan bileşenidir.
- **Container:** Docker Daemon tarafından izole olarak çalıştırılan process'lerdir.
- **Image:** Container'lar, belirli uygulamaları veya işletim sistemlerini içeren katmanlardan oluşur.
- **Docker Registry:** Docker Image'ların saklandığı ve paylaşıldığı yerdir.
- **Docker CLI:** Docker komutlarını çalıştırmak için kullanılan komut satırı arayüzüdür.
- **Docker Compose:** Birden fazla container'ı yönetmek için kullanılan bir araçtır. YAML dosyaları ile uygulama servislerini yapılandırır ve başlatır.

10.2 Docker'ın Avantajları

Docker'ın en büyük avantajı, izole bir ortam sağlayarak uygulamaların farklı bilgisayarlarda aynı şekilde çalışmasını sağlamasıdır. Ayrıca, Docker container'ları *image* haline getirilip taşınabilir, bu da yazılımın dağıtımını kolaylaştırır.

10.3 Docker kullanılmadığında

Uygulamanın bir bilgisayarda çalışıp diğerinde çalışmaması gibi sorunlar yaşanabilir. Bu, bağımlılık ve versiyon uyumsuzluklarından kaynaklanabilir. Başka bir bilgisayarda programı çalıştırmak daha problemlidir. Sanal makineye kıyasla da daha az kaynak tüketimi vardır.