



T.C.
KARADENİZ TEKNİK ÜNİVERSİTESİ
Mühendislik Fakültesi

Elektrik-Elektronik Mühendisliği Bölümü

RSSI Tabanlı Konum
Tespit Cihazı
(Bitirme Projesi)

365041 Furkan BİLGE
365036 Alipaşa ÖKSÜZ

Prof. Dr. Temel KAYIKÇIOĞLU

Ocak 2022
TRABZON



T.C.
KARADENİZ TEKNİK ÜNİVERSİTESİ
Mühendislik Fakültesi

Elektrik-Elektronik Mühendisliği Bölümü

RSSI Tabanlı Konum
Tespit Cihazı
(Bitirme Projesi)

365041 Furkan BİLGE
365036 Alipaşa ÖKSÜZ

Prof. Dr. Temel KAYIKÇIOĞLU

Ocak 2022
TRABZON

LİSANS BİTİRME PROJESİ ONAY FORMU

Furkan BİLGE ve Alipaşa ÖKSÜZ tarafından Prof. Dr. Temel KAYIKÇIOĞLU yönetiminde hazırlanan “RSSI Tabanlı Konum Tespit Cihazı” başlıklı lisans bitirme projesi tarafımızdan incelenmiş, kapsamı ve niteliği açısından bir Lisans Bitirme Projesi olarak kabul edilmiştir.

Danışman : Prof. Dr. Temel KAYIKÇIOĞLU

Jüri Üyesi 1 : Prof. Dr. İsmail KAYA

Jüri Üyesi 2 : Prof. Dr. Ali GANGAL

Bölüm Başkanı : Prof. Dr. Ayten ATASOY

ÖNSÖZ

Bu bitirme projesi Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü öğrencilerinden; Furkan BİLGE ve Alipaşa ÖKSÜZ tarafından Lisans Bitirme Projesi kapsamında hazırlanmıştır. Proje kapsamında bizlerden desteklerini esirgemeyen en başta Prof. Dr. Temel KAYIKÇIOĞLU' na teşekkürlerimizi sunarız.

Projenin gerektirdiği teknik bilgiler ve saha bilgileriyle bizi aydınlatan, yol gösteren değerli Prof. Dr. İsmail Hakkı ALTAŞ ve Doç. Dr. Gökçe HACIOĞLU hocamıza teşekkürlerimizi sunarız.

Benzer şekilde bu çalışmaya destekleyen Karadeniz Teknik Üniversitesi Rektörlüğüne, Mühendislik Fakültesi Dekanlığına ve Elektrik-Elektronik Mühendisliği Bölüm Başkanımız Prof. Dr. Ayten ATASOY hocamıza minnettarız.

İyi günümüzden kötü günümüze her daim yanımızda olan bizlerden desteklerini esirgemeyen tüm aile fertlerimize bize verdikleri emeklerden ötürü sonsuz minnettarım.

Ocak 2022
Furkan BİLGE
Alipaşa ÖKSÜZ

İÇİNDEKİLER

LİSANS BİTİRME PROJESİ ONAY FORMU	i
ÖNSÖZ	ii
İÇİNDEKİLER	iii
ÖZET	vi
SEMBOLLER VE KISALTMALAR	viii
SEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ		xi
1. GİRİŞ		1
1.1. Genel Bilgiler	1
1.2. Literatür Araştırması	1
1.3. Özgünlük	2
1.4. Yöntem	3
1.4.1. RSSI ile Mesafenin Kestirilmesi	3
1.4.2. SPI Haberleşmesi	4
1.5. Yaygın Etki	5
1.6. Standartlar	5
1.7. Çalışma Takvimi	6
1.8. İş Paketleri Organizasyonu ve Çalışma Yönetimi	10
2. TEORİK ALTYAPI		11
2.1. LoRa/LoRaWAN.....		11
2.2. LoRa ile Konum Tespitinde Kullanılabilecek Yöntemler.....		15
2.2.1. RSSI	15
2.3. Kablosuz Yayılım Modelleri	17
2.3.1. ITU Kapalı Alan İşaret Modeli	17
2.3.2. Serbest Uzay Yayılım Modeli	18
2.3.3. İki Işınlı Yayılım Modeli	19
2.3.4. Log Normal Gölgeleme Modeli	20
2.4. TOA	22
2.4.1. TOA için Senkronizasyon	23
2.5. TDOA	25

2.6.	PSO Algoritmasıyla Konum Tespiti	27
2.7.	Trilaterasyon Yöntemi ile Konumun Bulunması	29
3.	TASARIM	32
3.1.	Genel Bilgiler.....	32
3.2.	Boyutlandırmalar.....	33
3.3.	Sistem Bileşenleri ve Seçimleri	34
3.3.1.	Mikrodenetleyici.....	35
3.3.2.	Haberleşme Modülü.....	35
3.3.3.	Batarya.....	35
3.3.4.	Mikrodenetleyici Seçimi.....	35
3.3.5.	Haberleşme Modülü Seçimi.....	36
3.4.	Uyulanan Yöntemler.....	38
3.4.1	Devre Tasarım Yöntemleri.....	38
3.4.1.1.	Breadboard Devre Tasarımı.....	38
3.4.1.2.	PCB/Eagle Devre Tasarımı.....	39
3.4.2	Haberleşme Yöntemleri.....	39
3.4.2.1.	RSSI Hesaplama Yöntemi.....	39
3.4.2.2.	LoRa Haberleşme Yöntemi.....	40
3.5.	Yazılımlar.....	40
3.6.	Malzeme Listesi ve Ekonomik Analiz.....	41
3.7.	Hukuki Boyut.....	43
4.	SİMÜLASYON ÇALIŞMALARI	44
4.1.	Genel Bilgiler	44
4.2.	Yol Kayıpları Simülasyon Çalışmaları.....	44
4.2.1	Sistem Yol Kayıplarının Modellemesi.....	44
4.2.2	Sistem Yol Kayıplarının Simülasyon Yazılımı.....	45
4.2.3	Yol Kayıplarının Simülasyon.....	47
4.3.	PSO Algoritmasıyla Simülasyon Çalışmaları.....	49
4.3.1.	PSO Algoritmasıyla Koordinat Tespiti Simülasyon Modeli	49
4.3.2.	PSO Algoritmasıyla Konum Tespiti	51

Simülasyon Yazılımı	
4.3.3. PSO Algoritması Koordinat Tespiti Simülasyonu	54
5. DENEYSEL ÇALIŞMALAR	56
5.1. Genel Bilgiler	56
5.2. Yapılan Testler	76
6. SONUÇLAR	79
6.1. Genel Açıklamalar	79
6.2. Simülasyon Sonuçları	79
6.3. Deney Sonuçları	83
7. DEĞERLENDİRMELER	99
8. KAYNAKLAR	96
9. EKLER	98

ÖZET

Hayvansal gıdaların ve hayvansal ürünlerin elde edilmesine yönelik olarak yapılan hayvan yetiştirilmesi, beslenmesi ve bakımını kapsayan tarım kolu genel olarak hayvancılık olarak adlandırılır. Ülkemizde hayvancılık en fazla Doğu Anadolu bölgesinde yapılmaktadır. Hızla artan nüfus artışıyla beraber tüketimi dengelemek amacıyla artan hayvancılık sektörü de bölge halkının en önemli geçim kaynağı olmuş durumdadır. Bu sektör farklı alt dallarıyla beraber ekonomiye doğrudan ve dolaylı olarak birçok katkı sağlayarak ekonomide çok önemli bir yere sahiptir. Günümüzde birçok insan büyükbaş hayvan yetiştiriciliği yapmaktadır. Nüfus artışıyla beraber üreticiye olan talepte artmıştır. Ancak bu talep ekonomik nedenlerden ötürü sınırlandırılmıştır. Özellikle bu nedenlere bir de günümüzde köylülerin bu hayvanlarının çalınma, kaybolma korkusu eklenince üretici iyice bu işlere girişmekte çekinmektedir. Bu duruma çözüm olarak konum tespit cihazları geliştirilmiştir. Şimdiye kadar bu soruna çözüm olarak geliştirilen cihazlar genellikle Global Navigasyon Uydu Sistemleri (GNSS) tabanlı konum tespitini kullanmıştır. Ancak bu tür cihazlarda koordinatların iletimi için GSM ağı ile birleştirilmiş GPS alıcıları kullanılmaktadır. Bu cihazlarda kullanılan GPS alıcıları da normal çalışma modlarında 30 mA ile 50 mA arası tüketime sahiptir. Bu da taşınabilir bir alıcıda kullanılan pilin ömrünü birkaç günle sınırlandırılması anlamına gelmektedir. Oysaki bu tür IoT uygulamalarında kullanılacak alıcı ve vericilerden beklenen düşük maliyetli ve düşük güç tüketimli olmalarıdır. Teknolojinin gelişmesiyle maliyet düşümü sağlanırken, haberleşme teknolojilerindeki gelişmeyle de düşük güç tüketime dayalı yeni haberleşme protokolleri geliştirilmiştir. Bu gelişmelerle konum tespit uygulamalarında da farklı teknolojilerin gelişmesine fırsat doğmuştur. Bu projede kullandığımız RSSI tabanlı LoRa teknolojisi ise bu yeni teknolojilerden biridir. Yapılan bu projede LoRa ile ESP32 haberleşmesiyle çalışacak olan cihazdan en az 5 adet üretilecektir. 4'ünün koordinatı belli, sabit şekilde duracaktır. Konumu bilinmek istenen hayvana konulacak 5. cihaz sabit cihazlarla haberleşecektir. Bu haberleşme anında sinyalin diğer alıcılara ulaştığı andaki güç değerleriyle de içerisindeki algoritma sayesinde 5. cihazın konumu hesaplanacaktır. Bu sayede konumu bilinmiş olacaktır. Bu sistemi tasarlarken kullandığımız LoRaWAN teknolojisi hem maliyet hem de uzun ömürlülük açısından projeye en uygun protokoldür. LoRa haberleşmesi her ne kadar yeni bir teknoloji olsa da gittikçe yaygınlaşan ve IoT dediğimiz akıllı sistemlerin hayatımıza girmesiyle birlikte bu yükseliş ivmesini arttırmasını beklediğimiz bir teknolojidir.

SEMBOLLER VE KISALTMALAR

GPS	Küresel Konumlandırma Sistemi (Global Positioning System)
LoRa	Alıcı-verici anten modülü
LoRAWAN	Uzun Mesafeler için Kablosuz Telekomünikasyon Geniş Alan Ağı
IoT	Nesnelerin İnterneti
KB	Kilobayt
sn	Saniye
m	Mili
A	Amper
V	Volt
c	Işık Hızı (m/s)
P	Güç Birimi
C	Veri Hızı
Bps	Saniyede İletilen Bit Sayısı
B	Bant Genişliği
S	Watt Cinsinden İşaret Gücü
r	Yarıçap
N	Watt Cinsinden Gürültü Gücü
SNR	İşaret Gürültü Oranı
BER	Bitteki Hata Oranı
RF	Radyo Frekansı
V_{in}	Giriş Gerilimi
V_{out}	Çıkış Gerilimi
MHz	Megahertz
GSM	Mobil İletişim için Küresel Sistem
IMEI	Uluslararası Mobil Cihaz Kodu
Ohm	Direnç Birimi
W	Watt (Güç Birimi)
RTC	Real Time Clock
LCD	Sıvı Kristal Ekran
ADC	Analog Dijital Konvertör
SMD	Yüzeye Monte Edilebilen Eleman

hex	Hexadecimal
DC	Doğru Akım
AC	Alternatif Akım
ESP32	32 Bitlik Mikrodenetleyici
SPI	Serial Peripheral Interface
ARM	Acorn RISC Machine
RXD	Veri Al
TXD	Veri İlet
VCC	Besleme Gerilimi
ISO	International Organization for Standardization
TSE	Türk Standartları Enstitüsü

ŞEKİLLER DİZİNİ

Şekil 1.1.	SPI haberleşmesi blok diyagramı
Şekil 2.1.	Veri hızı ve enerji verimliliğine göre kablosuz haberleşme teknolojilerinin karşılaştırılması
Şekil 2.2.	LoRa protokol katman yığınları
Şekil 2.3.	RSSI ile konum tespiti
Şekil 2.4.	İki ışınlı yayılım modeli
Şekil 2.5.	Uç düğümler ve senkronizasyon düğümüyle kabaca sistemin modellenmesi
Şekil 2.6.	Üç alıcı ve hedef arasında oluşan hiperbolik hatlar
Şekil 2.7.	PSO yöntemi algoritma akış diyagramı
Şekil 2.8.	Üç boyutlu trilaterasyon
Şekil 3.1.	RSSI ile konum tespiti
Şekil 3.2.	Sistem şematiği-EagleCAD
Şekil 3.3.	Sistem kutu prototipi SketchUp
Şekil 3.4.	PCB üzerine kurulu sistem prototipi
Şekil 3.5.	ESP32-DEVKITC-32D mikrodenetleyicisi
Şekil 3.6.	E19-433M20S2 LoRa modülü
Şekil 3.7.	4 dBi SMA anten
Şekil 3.8.	Uzaklığa bağlı RSSI grafiği
Şekil 3.9.	Düğümün yerleştirilmesi
Şekil 3.10.	Konum tespiti için kullanılacak algoritmanın işleyişi
Şekil 4.1.	PSO yöntemiyle konum tespit algoritması akış diyagramı
Şekil 5.1.	ESP32-DEVKITC-32D ile LoRa haberleşme modülünün pin diyagramı
Şekil 5.2.	Konum tespitinde kullanılacak düğümlerin yalnızca bir tanesi
Şekil 5.3.	Gönderici LoRa modülüne ilişkin ekran görüntüsü
Şekil 5.4.	Alıcı LoRa modülüne ilişkin ekran görüntüsü
Şekil 5.5.	İlk test çalışmasının yapıldığı ortam
Şekil 5.6.	İkinci test çalışmasının yapıldığı ortam
Şekil 6.1.	Serbest Uzay Kayıp Modeli için Yol Kaybı-Mesafe Değişimi
Şekil 6.2.	Log distance kayıp modeli için yol kaybı-mesafe değişimi
Şekil 6.3.	Log normal gölgeleme kayıp modeli için yol kaybı-mesafe değişimi
Şekil 6.4.	Node 1 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi
Şekil 6.5.	Node 2 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi
Şekil 6.6.	Node 3 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi
Şekil 6.7.	Node 4 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi
Şekil 6.8.	Node 1 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi
Şekil 6.9.	Node 2 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi
Şekil 6.10.	Node 3 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi
Şekil 6.11.	Node 4 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi
Şekil 6.12.	Sistemin tamamının test edildiği alan
Şekil 6.13.	Teorik sonuçlar ve test sonuçlarının dağılımı
Şekil 6.14.	Hedef cihaz Node1 ile Node2 arasında olduğunda teorik ve test sonuçları
Şekil 6.15.	Hedef cihaz Node2 ile Node4 arasında olduğunda teorik ve test sonuçları
Şekil 6.16.	Hedef cihaz Node3 ile Node4 arasında olduğunda teorik ve test sonuçları

ÇİZELGELER DİZİNİ

Çizelge 1.1.	İş-Zaman Planı
Çizelge 1.2.	Risk Analizi ve B planları
Çizelge 1.3.	İş paketleri dağılımı
Çizelge 2.1.	LoRa haberleşmesinin özellikleri
Çizelge 2.2.	Farklı ortamlar için yol kayıp üsleri
Çizelge 2.3.	Düğümler arası zaman farkının tespiti
Çizelge 3.1.	ESP32-DEVKITC-32U donanım özellikleri
Çizelge 3.2.	E19-433M20S2 LoRa modülü donanımsal özellikleri
Çizelge 3.3.	4 dBi SMA antenin donanımsal özellikleri
Çizelge 3.4.	Tasarımda kullanılacak malzemelerin piyasa değerlendirmesi
Çizelge 3.5.	Tasarımda kullanılacak malzemelerin maliyet listesi
Çizelge 4.1.	Serbest uzay kayıpları için ortam kaybının Matlab modeli
Çizelge 4.2.	Log distance ve log normal gölgeleme için kayıpların Matlab modeli
Çizelge 4.3.	Yol kayıp modellerinin simülasyonu
Çizelge 4.4.	PSO algoritmasıyla konum tespiti için geliştirilen MATLAB yazılımı
Çizelge 4.5.	PSO algoritması sonucunda (xr,yr) koordinatlarına yakınsama değerleri
Çizelge 4.6.	Algoritmanın bulduğu değerler ile gerçek değerlerin karşılaştırılması
Çizelge 5.1.	Alıcı-Verici LoRa modüllerinin haberleşmesi için gerekli program
Çizelge 6.1.	Node 1 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri
Çizelge 6.2.	Node 2 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri
Çizelge 6.3.	Node 3 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri
Çizelge 6.4.	Node 4 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri
Çizelge 6.5.	Node 1 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri
Çizelge 6.6.	Node 2 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri
Çizelge 6.7.	Node 3 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri
Çizelge 6.8.	Node 4 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri
Çizelge 6.9.	Tüm sistemin testi sonucunda hedef cihazın X, Y ve Z bileşenleri
Çizelge 6.10.	Hedef cihazın Node1 ile Node2'nin ortasında olduğu durum
Çizelge 6.11.	Hedef cihazın Node2 ile Node4'nin ortasında olduğu durum
Çizelge 6.12.	Hedef cihazın Node3 ile Node4 arasında olduğu durum
Çizelge 6.13.	Hedef cihazın Node1 ile Node3'ün ortasında olduğu durum

1. GİRİŞ

1.1. Genel Bilgiler

Geniş alanlarda GPS'e alternatif geliştirilmesi için tasarlanacak olan cihaz, daha çok açık alanlarda nesnelerin takibi için kullanılması hedeflenmiştir. İlk kullanım amacı hayvancılıkla uğraşanların hayvanlarının takibi için düşünülmüştür. Bu sayede hayvanlarını salan kişi evinde rahat bir şekilde hayvanlarının konumunu izleyebilecektir. Aynı zamanda cihaza eklenecek farklı sensörler ile hayvan/hava üzerinden gerekli olan veriler çekilebilecektir. Bu sayede hızlı reaksiyon alınabilecektir.

Literatür taraması yapıldığında benzer amaçla yapılmış çalışmaları görmek mümkündür. Ancak mevcut çalışmaların çoğunda konum çözümlerinin, genellikle IoT sistemlerinin düşük maliyetli ve düşük güç tüketimi kısıtlamalarıyla uyumlu olmayan Küresel Konumlandırma Sistemine (GPS) dayandığı görülmektedir. GNSS, geniş kapsama alanı, kullanılabilirliği, yüksek doğruluğu ve birçok uygulamaya entegrasyonu açısından avantajlı olsa da GPS ile yapılan konum tespiti uygulamalarında daha karmaşık, enerjik ve maliyetli cihazlara ihtiyaç vardır.

Günümüz dünyasında sensör teknolojilerindeki gelişmeler yaşanan maliyet düşümleri ve düşük güçlü cihazların sayısındaki artış konum tespit uygulamalarında da farklı teknolojilerin gelişmesini sağladı. Projede kullandığımız LoRa teknolojisi ise bunlardan biridir. İşte bu proje düşük maliyeti ve yüksek enerji verimiyle benzer çalışmalardan ayrılmaktadır. Bu amaçla projede tasarlanması planlanan sistem baştan ucu düşük enerji kullanımı isteyen teknolojilerle desteklenmiştir. Projenin uygulanma aşamasında LoRa ile ESP32 haberleşmesiyle çalışacak olan cihazdan en az 5 adet üretilenektir. 4'ünün koordinatı belli, sabit şekilde duracaktır. Konumu bilinmek istenen hayvana konulacak 5. cihaz sabit cihazlarla haberleşecektir. Bu haberleşme anında sinyalin diğer alıcılara ulaştığı andaki RSSI değerleriyle de içerisindeki algoritma sayesinde 5. Cihazın konumu hesaplanacaktır. Bu sayede konumu sabit noktalara göre konumu belirlenmiş olacaktır.

1.2. Literatür Araştırması

Çin'de 3 araştırmacı RSSI kullanarak 3 boyutlu olarak açık alanda konum tespit üzerine çalışmıştır. Sun Yat-sen üniversitesinde insansız hava araçlarında kullanarak yapılan testlerde Monte-Carlo test simülasyonları ve gerçek testlerde PSO algoritmasıyla yaklaşık %2'lük hata payıyla başarılı şekilde testleri yapılmıştır [1].

Çin’de 3 araştırmacı RSSI tabanlı Arduino UNO-LoRa kullanarak lokalizasyon algoritmalarını simüle ve test etmiştir. K-Mean Kümeleme, yenilemeli iterasyon, minimum MBRE, Hassaslık tabanlı kümeleme, seçilmiş bağlantı düğümlerinin minimum MBRE, seçilmiş yaklaşıklık hedeflerin centroidi algoritmalarını kullanıp sırayla algoritmaların başarılarını test etmişlerdir. Bu çalışmada Minimum MBRE algoritması daha yüksek doğrulukla çalışmıştır [2].

Almanya’da 3 araştırmacı RSSI tabanlı lokalizasyon sistemi düşük güç üzerine çalışmışlar. Sistem aktiften maksimum anlık akım 30 mA, uyku durumunda 6.7uA güç tüketimini elde etmişler. Ağırlıklı LMS algoritmasını kullanan araştırmacılar kritik olmayan güç odaklı lokalizasyon sistemleri üzerine çalıştıkları için hata payları diğer çalışmalara göre doğruluk oranı düşüktür [3].

Gebze Teknik Üniversitesinde yapılan yüksek lisans tezinde RSSI tabanlı Log Normal Gölgeleme Modeli kullanarak kapalı alanlarda konum tespit üzerine çalışma yapılmıştır. Bluetooth kullanarak yapılan sistemde konumlandırma algoritması olarak farklı bir üçgenleme algoritması kullanılmış ve tüm testleri yapılmıştır. 9.7*8 metrelik alanda yapılan testlerde 40 cm’lik hata payı ile sonuçlanmıştır [4].

2019 yılında dergide yayınlanan bu makalede 3 araştırmacı konum belirleme algoritmalarını karşılaştırmıştır. PSO ile Yarasa algoritmasının karşılaştırıldığı çalışmada PSO algoritması daha başarılı algoritma olduğu sonucu elde edilmiştir [5].

1.3. Özgünlük

Konum tespit amacıyla yapılan çalışmalar incelendiğinde coğrafi konum bilgisi için genellikle GPS modüllerinin kullanıldığı görülmüştür. Ancak öncesinde de belirttiğimiz üzere IoT sistemleri düşük maliyetli ve düşük güç tüketim kısıtlamalarına sahip sistemler olduğu bilinmektedir. Bu amaçla bu projede yüksek enerji tüketimi, maliyetli ve karmaşık yapıyı beraberinde getiren GPS modüllerini kullanmak yerine konumu belli 4 adet LoRa modülleri üzerinden konumu bilinmeyen diğer modülün konumu tespit anlayışı üzerine kurulmuştur. Eğer bu çalışma GPS modülü üzerinden yapılacak olsaydı bu modül 40 mA ile 50 mA arasında bir akım çekecekti. Oysaki sistemin tasarımında kullanılacak olan ve dolaylı olarak konum belirlemeye yardımcı olacak 433 MHz bandında çalışacak LoRa modülleri uyku durumunda 0,2 uA, veri gönderirken 128 mA

ve veri alırken 14 mA tüketir. Görüldüğü üzere güç tüketimi çalışma modlarına göre değişmektedir. Bu sayede GPS' e göre enerji verimi daha yüksektir.

1.4. Yöntem

1.4.1. RSSI ile Mesafenin Kestirilmesi

Projede konumu bilinmeyen hayvanın konumunu tespit edebilmek için en az 4 adet konumu bilinen LoRa'lara ihtiyaç vardır. Diğer 5. LoRa modülü takip etmek istediğimiz canlıya yerleştirilir. LoRa modülleri birbirlerine periyodik olarak sinyaller gönderir. Gönderilen sinyalin gücü ortam parametrelerine bağlı olarak zamanla zayıflayarak alıcılara ulaşır. Sinyal gücünde meydana gelen zayıflama sinyalin kat ettiği mesafeyle doğru orantılı olarak artar. Buna bağlı olarakta sinyalin RSSI değeri azalır. Alınan sinyal gücü (RSSI), verici antenden yayılan sinyalin gücü, alıcı-verici anten kazançları, alıcı ile verici arasındaki mesafe, ortam kayıpları ve yayılan sinyalin frekans bilgisine bağlı olarak farklı yol modelleri için RSSI bilgisinde mesafe bilgisini çıkartacak eşitlikler bulunmaktadır.

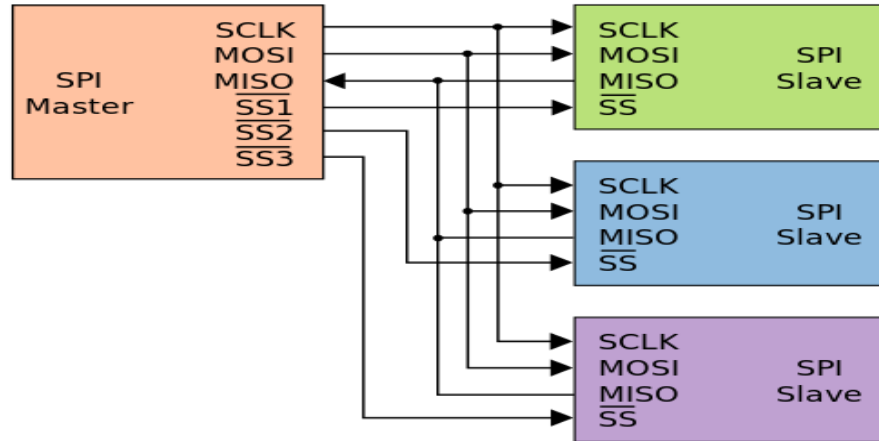
RSSI yöntemde ortam parametreleri bilindiği takdirde yapılması gereken tek şey sinyal gücünü ölçmektir. Bu da basit donanımlarla yapılabildiğinden bu yöntem konum tespitinde oldukça yaygın olarak kullanılmaktadır. Ancak bu yöntemin en büyük dezavantajı ortamdaki en küçük değişimden etkilenmesidir. Konumu tespit edilecek cihazın bu problemten etkilenmemesi için sürekli olarak değişen hava koşullarına göre ortam yol kayıplarının bulunması gereklidir. Bu problemi aşabilmek adına sisteme ilave bir adet konumu belli diğer 4 LoRa modülüne ek olarak konumu belli bir adet LoRa alıcısı eklenir. Bu alıcıyla sistemde konumu belli toplamda beş adet verici ve bir adet alıcı durumundaki hedef düğüm bulunur. Sonradan eklenen ve görevi sadece ortam kayıplarını bulmaya yarayan cihaz yayımladığı sinyal alıcıya ulaşmasıyla elde edilen RSSI ve konum değerlerinden yararlanarak eşitliklerdeki tek bilinmeyen ortam kayıpları değişen koşullar için güncel olarak elde edilir. Ardından bu kayıp değeri konumu bilinmeyen (hedef cihaz) sinyalin yaydığı ve alıcılarda alınan RSSI değeriyle öncesinden bulunan ortam kayıplarıyla basit bir denklem aracılığıyla mesafe bilgisi elde edilir. Bu yöntemle RSSI yönteminin en büyük dezavantajı olan değişken ortam şartlarına bağımsızlığı geliştirilmiş olur. Bu kısma **2.2.1** bölümünde detaylı olarak

değinilicektir. Bu aşamada ayrıca alınan sinyal gücünden mesafe bilgisini hesaplayacak bir mikrodnetleyiciye ihtiyaç vardır. Mikrodnetleyici olarak ESP32 kullanılacaktır. LoRa modülü ile ESP32 arasında SPI haberleşmesi uygun görülmüştür. Projeye sağladığı avantajlar ve dezavantajlarıyla 1.4.2 SPI Haberleşmesi başlığı altında detaylı olarak bahsedilmiştir.

1.4.2 SPI Haberleşmesi

Serial Peripheral Interface daha yaygın ismiyle SPI haberleşmesi birden fazla mikrodnetleyicinin master ve slave ilişkisi içerisinde birbirleriyle haberleşme imkânı sunan seri bir haberleşme protokolüdür. Master olan cihaz diğer çevresel aygıtları kontrol eder. SPI haberleşmede master ile slave arasında bağlantının kurulabilmesi için en az 4 adet veri yoluna ihtiyaç vardır. Bunlar:

- SCK: “Serial Clock” olarak bilinir. Master aygıt ile slave durumunda olan aygıtlar arasındaki senkronizasyonun sağlanması için kullanılan clock sinyalinin bulunduğu hattır. Master tarafından üretilir.
- MISO: “Master In Slave Out” ifadesinin baş harflerini kısaltmasıdır. Slave’deki verilerin Master’a aktırıldığı iletim hattını ifade eder.
- MOSI: Uzun ifadeye “Master Out Slave In” veri hattı Master cihazdan Slave cihazları gönderilecek verilerin yer aldığı iletim hattıdır.
- SS: Açılım “Slave Select”e karşılık gelen veri iletim yolu Master cihazın belirli Slave cihazları etkinleştirip diğerleri devre dışı bırakmasını yağılayan veri iletim yoludur.



Şekil 1.1 SPI haberleşmesi blok diyagramı

Tipik olarak bahsedilen bu veri iletim hatları Şekil 1.1.'te gösterilen Master-Slave yapısı üzerinde gösterilmiştir. Şekil 1.1.'te görüldüğü üzere Slave sayısı kadar Master cihazda SS hattı bulunur. Ayrıca her bir veri iletim hattıda tek yönlüdür. Bu yüzden Master ile Slave arasında da veri iletimi tek yönlü iki hat üzerinden yapılır. Veri iletimine başlanmadan önce Master aygıtta SS hattı üzerinede haberleşme yapılacak Slave seçilir. SS'in lojik 0 olduğu Slave seçilir. SCK sinyalinin her yükselen kenarında MOSI'deki veriler okunur. Benzer şekilde aynı durum MISO hattı içinde geçerlidir.

1.5. Yaygın Etki

LoRa haberleşmesi uzun mesafe, düşük güç tüketimiyle öne çıkan kablosuz bir haberleşme protokolüdür. Bu teknoloji henüz yeni gelişen bir teknoloji olmasına rağmen şimdiden çok geniş kullanım alanlarına sahiptir. Bu proje daha çok hayvancılık sektörüne yönelik RSSI tabanlı konum tespitini sağlar. Ancak bu çalışmayla araç takibinden, kayıp yaşlı ve çocuk takibine veya görme engelli bireylerin takibi ya da otonom hareket eden araçlara kadar konum tespitine dayalı bütün IoT projelerine entegre edilebileceği görüşü yaygındır. Bu proje tamamlandığında köylüler hayvanlarının çalınma, kaybolma korkusunu çekmeden takip edebilecektir. Ayrıca proje gelişime de açıktır. Öyle ki konumu tespit edilecek cihaza ilave olarak eklenen elektriksel şok etkisi veren bir cihazla hayvan kendisi için belirlenen sınırların dışına çıkarsa uyarılacak ve zamanla eğitilen hayvan uzaktan yönetilebilecektir. Bu proje benzer çalışmalara temel olması açısından ileriye yönelik bu alanda yayın çıkarma potansiyeli taşımaktadır.

1.6. Standartlar

Yapılan çalışmada uyulması gereken ve uyulan standartlar aşağıda verilmiştir.

- IEC 60559 Mikroişlemciler
- IEEE 2413 Nesnelerin internet (IoT) için mimari çerçeve standardı
- IEC 60086 Birincil nesil bataryalar
- IEC 61508 Programlanabilir cihazların güvenliği
- ISO/IEC 29182 Sensör ağ alanları
- ISO 19111 Koordinatlar ile mekânsal referans
- IEC 60748 Yarı iletken cihazlar ve entegre devreler

1.7. Çalışma Takvimi

Mühendislik Tasarımı ve Bitirme Projesi sürecinde izlenecek olan yol planlanarak iş- zaman grafiği oluşturulmuştur. Çizelge 1.1’de proje iş paketlerine bölünmüş ve her pakette yapılacak olan çalışmalar belirtilmiştir. Çalışmanın aksamaması için her iş paketinde tamamlanması gereken zaman aralıkları verilmiştir. Meydana gelebilecek beklenmedik durumlar için farklı yollar düşünülmüştür. Olası tehlikeler göz önünde bulundurularak Çizelge 1.2’de risk analizi yapılmıştır.

Çizelge 1.1. İş-Zaman Çizelgesi

İP No	İş Paketinin Adı ve Amacı	Kimler Tarafından Çalışıldığı	Zaman Aralığı	Projeye Katkısı
1	Proje fikrinin oluşturulması, literatür araştırması ve özgün fikrin ortaya çıkarılması	Alipaşa ÖKSÜZ(L) Furkan BİLGE	Ekim 2021	Proje fikri oluşturulup, özgünlük literatür araştırmasıyla desteklenir. Özgün fikrin oluşması sonraki adımlarda neler yapılacağını da netleştirir.
2	Uygulanacak yöntemlerin belirlenmesi ve ilgili teorik çalışmaların yapılması. Tasarım aşamasına geçilmesi	Alipaşa ÖKSÜZ Furkan BİLGE(L)	Ekim-Kasım 2021	Uygulanacak yöntemler teorik açıklamalarla desteklenir. Tasarıma devam edilebilmesi teorik bilgi yeterliliğine bağlıdır.
3	Tasarım hesaplamalarının ve çizimlerin tamamlanması, mali analiz yapıp bütçe oluşturulması ve hukuki sorumlulukların araştırılması	Alipaşa ÖKSÜZ(L) Furkan BİLGE	Kasım-Aralık 2021	Projenin tasarım hesapları ve teknik çizimleri yapılır. Masa, kasa, kutu vb bileşenlerin boyutlarına göre bağlantı diyagramları oluşturulur. Bağlantı diyagramları prototipin çalışma göstergesidir.
4	Simülasyon modelinin oluşturulup simülasyonların yapılması, sonuçların değerlendirilmesi ve Mühendislik Tasarımı kitabının yazılması	Alipaşa ÖKSÜZ Furkan BİLGE(L)	Aralık 2021 - Ocak 2022	Tasarlanan sistemin simülasyon modeli oluşturulup, paket programlar veya geliştirilecek yazılımlarla simülasyonu yapılır. Simülasyon prototipin çalışıp çalışmayacağını önceden bilmesini sağlayacağından önemlidir.
5	Dönemsonu sınavları ve Mühendislik Tasarımı sunumlarının yapılması,	Alipaşa ÖKSÜZ Furkan BİLGE(L)	Ocak 2022	Hazırlanan Mühendislik Tasarımı jüri tarafından sunumlar sırasında değerlendirilir. Mühendislik Tasarımı Bitirme Projesinin ön koşulu olduğundan sunumları başarıyla tamamlamak önemlidir.

Çizelge 1.1. İş-Zaman Çizelgesi (Devam)

İP No	İş Paketinn Adı ve Amacı	Kimler Tarafından Çalışıldığı	Zaman Aralığı	Projeye Katkısı
6	İlk Prototip hazırlanır ve düğümler haberleştirilir. Uzaklık tespit testleri yapılır.	Alipaşa ÖKSÜZ(L) Furkan BİLGE	Ocak 2022	Düğümmler arasında modüllerin kesin olarak haberleşmesi RSSI ile uzaklık tespiti hata payının 200 metre aşağında olması
7	Bütün düğümlerler aynı anda haberleştirilip konum tespiti yapılması, geliştirilmesi	Alipaşa ÖKSÜZ(L) Furkan BİLGE	Ocak 2022	Uygun sahada donanımlar hazırlanıp tam konum tespiti testleri yapılması Hata payının 100 metre civarında olması
8	Bulunan konum verisi internette güncel olarak takip edilecek olan ortamın gerçekleştirilmesi	Alipaşa ÖKSÜZ Furkan BİLGE(L)	Şubat 2022	Gönderilen verinin internet arayüzünde tam olarak okunması ve harita üzerinde nokta olarak gösterilmesi
9	Sistemin optimizasyonu ve sistemin geliştirilmesi	Alipaşa ÖKSÜZ Furkan BİLGE(L)	Mart-Nisan 2022	Hata payı düşürülmesi ve verimliliğin artırılması ile projenin geliştirilmesi
10	Baskı devre kartı üretilmesi, güvenli çalışması için cihaz kutusu basılması	Alipaşa ÖKSÜZ(L) Furkan BİLGE	Mayıs2022	PCB ve 3D ürünlerin çıkartılması Projenin son halini alması

Çizelge 1.2. Risk analizi ve B planları

İP No	Projeye Katkısı	Risk Analizi
1	Proje fikri oluşturulup, özgünlük literatür araştırmasıyla desteklenir. Özgün fikrin oluşması sonraki adımlarda neler yapılacağını da netleştirir.	Farklı bir projeye çok ortak nokta olması ya da yeterince yenilikçi ve zorlayıcı proje olarak görülmezse farklı proje fikirleri düşünülür.
2	Uygulanacak yöntemler teorik açıklamalarla desteklenir. Tasarıma devam edilebilmesi teorik bilgi yeterliliğine bağlıdır.	Konu hakkında bilgi yetersizliği varsa ek çalışmalar yapılarak giderilebilir. Projenin gerçekleştirilmesi üzerine teorik bir eksiklik olmadığından emin olunur.
3	Projenin tasarım hesapları ve teknik çizimleri yapılır. Masa, kasa, kutu vb bileşenlerin boyutlarına göre bağlantı diyagramları oluşturulur. Bağlantı diyagramları prototipin çalışma göstergesidir.	Çalışmanın bağlantı diyagramları teknik çizim programları kullanılarak çizilecektir. Üretilme sırasında kesin ölçüler önceden bilinecektir.
4	Tasarlanan sistemin simülasyon modeli oluşturulup, paket programlar veya geliştirilecek yazılımlarla simülasyonu yapılır. Simülasyon prototipin çalışıp çalışmayacağını önceden bilinmesini sağlayacağından önemlidir.	Projenin simülasyonu paket programlar kullanılarak yapılacaktır. Böylece proje üretilmeden önce karşılaşılabilecek problemler önceden bilinir ve değişikliklere gidilir.
5	Hazırlanan Mühendislik Tasarımı jüri tarafından sunumlar sırasında değerlendirilir. Mühendislik Tasarımı Bitirme Projesinin ön koşulu olduğundan sunumları başarıyla tamamlamak önemlidir.	Proje kitabının yazım kurallarına uygun hazırlanıp sunum yapılır. Sunum yapılmaması durumunda başarısız olunur.
6	Düğümmler arasında modüllerin kesin olarak haberleşmesi RSSI ile uzaklık tespiti hata payının 200 metre aşağıda olması	Basit haberleşme deneyi gerçekleştirilir. Gerçekleşmemesi durumunda modüllere daha araştırmaya yapılır ya da modül değişimi yapılır.
7	Uygun sahada donanımlar hazırlanıp tam konum tespiti testleri yapılması Hata payının 100 metre civarında olması	Uzaklıktan konum tespiti gerçekleştirilir. Hata payının simülasyondan fazla olması durumunda sistemde değişikliklere gidilir.
8	Gönderilen verinin internet arayüzünde tam olarak okunması ve harita üzerinde nokta olarak gösterilmesi	Tespit edilen konum bilgisi kullanıcın göreceği arayüze gönderilir. Eğer gerçekleştirilememesi durumunda farklı arayüz programlarına geçilir.
9	Hata payı düşürülmesi ve verimliliğin artırılması ile projenin geliştirilmesi	Sistem hataları üzerine geliştirilme yapılır.
10	PCB ve 3D ürünlerin çıkartılması Projenin son halini alması	Proje son halini alır. Bitirme sergisine katılmaya hazırdır.

1.8. İş Paketleri Organizasyonu ve Çalışma Yönetimi

Projenin planla bir şekilde ilerleyebilmesi için Çizelge 1.1.'de hazırlanan iş zaman çizelgesi projeden sorumlu öğrenciler arasında paylaştırılmıştır. Çizelge 1.1.'deki ilk 5 iş paketi tasarım çalışmasına ait iken son 5 iş paketi bitirme projesine aittir.

Proje yönetiminin kolaylaşması adına tüm iş paketleri biri Lider olmak üzere proje çalışanları arasında paylaştırılmıştır. Her öğrenci en az bir iş paketinde liderlik görevi yapmalı ve sorumluluk üstlenerek diğer öğrencilere neler yapmaları gerektiğini anlatarak ilgili iş paketinin bitmesini sağlamalıdır. Bu amaçla Çizelge 1.3.'de iş paket dağılımları oluşturulmuştur.

Çizelge 1.3. İş paketleri dağılımı

İş Paketi Lideri	Lider Olduğu İş Paketleri	Bulunduğu İş Paketi Sayısı
Alipaşa ÖKSÜZ	1,3,6,7,10	10
Furkan BİLGE	2,4,5,8,9	10

2. TEORİK ALTYAPI

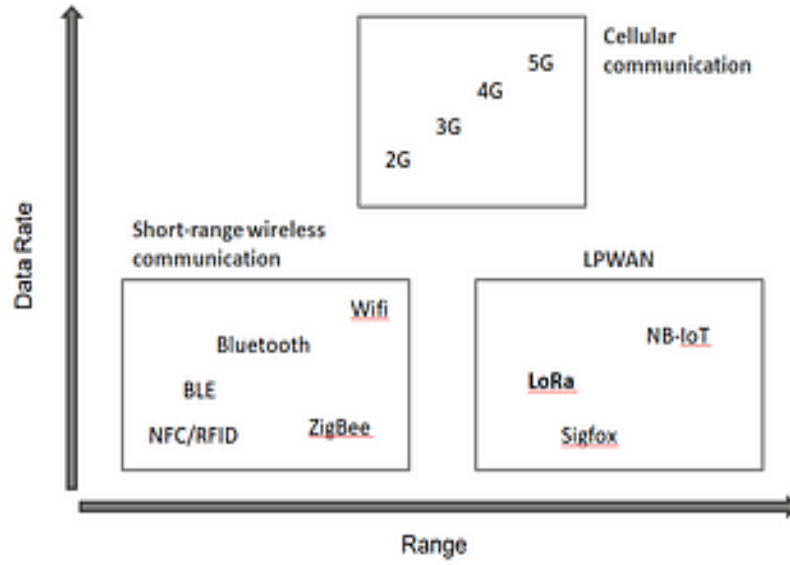
2.1. LoRa/LoRaWAN

Öncelikle LoRa teknolojisinin gelişim safhalarından bahsetmek ardından bu teknoloji neyin eksikliğini giderdiğini, hangi amaçlarla nerelerde kullanıldığını açıklamak ardından projede LoRa modüllerini nasıl, ne amaçla kullandığımızı açıklamak daha doğru olacaktır.

LoRa teknolojisi, özellikle IoT uygulamalarının yaygınlaşmasıyla sahada bulunan çok sayıda sensörlerden uzak mesafe düşük enerji tüketimiyle haberleşmesini sağlayan bir iletişim teknolojisidir. Sensörlerde bu teknolojiye ihtiyaç duyulmasının temelinde bu cihazların özgürleşme isteği yatmaktadır. Bu cihazlar sahada farklı verileri toplayabilen cihazlardır ve bunlar özgür olmak ister. Ancak bu durumda bu cihazları sınırlayan iki temel unsur vardır. Bunlardan biri enerji diğeri de haberleşmedir. Normalde her iki unsorda kablolarla yapılabilir. Ancak IoT uygulamalarında kullanılacak sensörler birbirlerinden bağımsız olmak istediklerinden dolayı kablolarla bu işlemler yapmak bir çözüm olmayacaktır. Bunun yerine haberleşme ihtiyacı kablosuz haberleşme teknolojileriyle, enerji ihtiyacı da pillerle sağlayabilecek alternatif çözüm arayışlarına girilmiştir. Bu durumda karşımıza uzun süre pille çalışabilecek kablosuz ağ teknolojilerini ve arayışlarını çıkıyor. Bu arayışların çıkış noktası ise uzun mesafelere sinyal gönderip bunu yaparken de düşük enerji tüketimiyle yapma isteğinden gelmektedir.

Literatür incelemesi yapıldığında aslında düşük enerji tüketimiyle kablosuz haberleşmeyi sağlayan farklı teknolojilerin olduğu görülmektedir. Şekil 2.1.'de IoT uygulamaları için en önemli kıstaslardan olan mesafe ve güç tüketimine göre bu teknolojiler gösterilmiştir.

Şekil 2.1.'de GSM şirketleri tarafından sağlanan 2G,3G,4G ve 5G gibi hizmetler yüksek veri iletimi imkânı sağlamaktadır. Ancak veri iletim hızı arttıkça verici güçleri de artmakta yani enerji tüketimi de artmaktadır. Bu da sahada çalışacak sensörlerin pillerinin iki veya üç günde değiştirilmesi anlamına gelmektedir. Diğer yandan Bluetooth, ZigBee kısa mesafelerde düşük enerji tüketimi sağlamakta ancak bunlarda bir veya iki engeli aşabilecek kısa mesafelerde iletim sağlıyor. WiFi haberleşmesi de yine kısa sayılabilecek mesafelere daha yüksek veri iletim hızını sağlamaktadır. Ancak IoT uygulamalarında bizim aradığımız çok yüksek veri iletim hızı değil uzun mesafelerde düşük veri iletimidir.



Şekil 2.1. Veri Hızı ve Enerji Verimliliğine Göre Kablosuz Haberleşme Teknolojilerinin Karşılaştırılması [7]

Şekil 2.1.’deki grafiğin sağ alt köşesi aranan kriterlere cevap veren bölgedir. Bu bölge üzerinde akademide uzun yıllar çalışmalar yapılmıştır. Nihayetinde aradığımız kriterleri sunan LoRa teknolojisi geliştirilmiştir. Bu teknoloji hem uzun mesafelere sinyal gönderirken hem de bunu düşük enerji tüketimiyle yapma fırsatı sağlamıştır.

LoRa teknolojisinin en yaygın kullanım alanı takip cihazlarıdır. Bu alanda özellikle Hindistan’da scooterları takibi için ülke genelinde yürütülen çalışmalar vardır. Yine Hindistan’da LoRa ağı içerisinde sokaklarda panic butonları kullanılmaktadır. Bu cihazlar olası tehlike durumlarında LoRa ağı üzerinden konum bilgisini güvenlik güçleriyle paylaşmaktadır. Farklı bir uygulama alanı da uzaktan sayaç okumalarıdır. Özellikle Fransa’da elektrik, su veya doğalgaz sayaçları uzaktan okuma için LoRa teknolojisini kullanmaktadır. Benzer şekilde akıllı tarım alanlarında nem sensörleriyle uzaktan sulama imkânı sağlayarak üretimde verimliliği artırmaktadır. Sıcaklık sensörleriyle orman yangınlarından haberdar olunabilmektedir. Tüm bu uygulamalarda sadece bir kalem pille on yıl gibi uzun sürelerde sahaya dağıtılmış bu sensörlerle LoRa teknolojisi sayesinde gerçekleştirilebilmektedir.

LoRa teknolojisinin pek çok özelliğiyle gelecekte yaygın kullanım alanına sahip olacağı şüphesizdir. Bu durumu destekleyen en önemlisi unsurda LoRa’ nın ISM bandını kullanmasıdır.

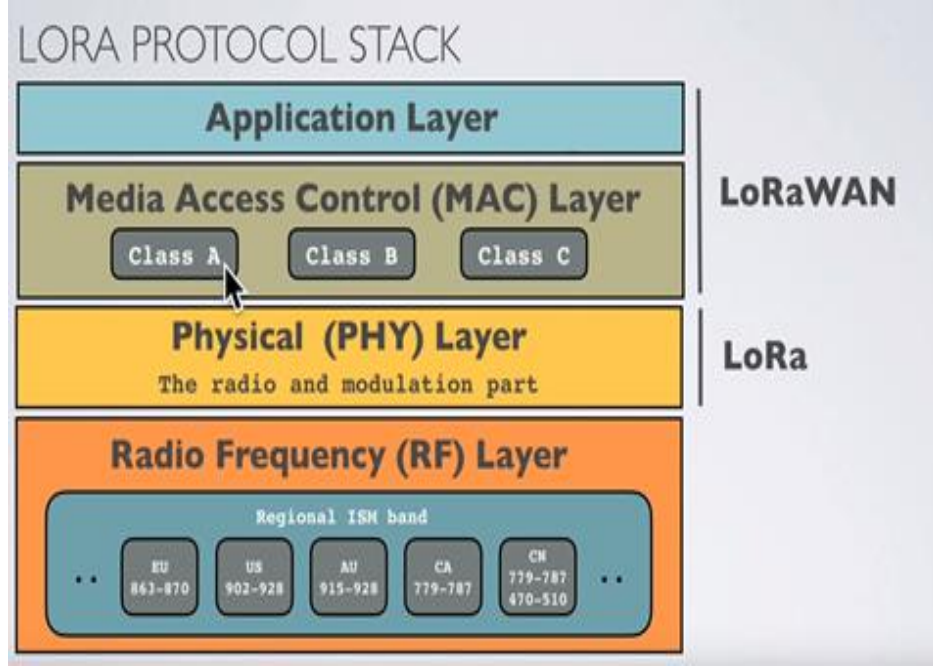
Bu bant içerisinde belirli çıkış güç sınırlamaları altında kurulacak LoRa ağının herhangi bir yasal izine, lisansa tabi olmamasıdır. Çizelge 2.1.’ de LoRa haberleşmesinin özellikleri gösterilmiştir.

Çizelge 2.1. LoRa haberleşmesinin özellikleri

Özellikler	LoRa (Aktarım Gücü: 20 dBm)
Modülasyon	Cıvıltı yayılma spektrumu (Chirp Spread Spectrum-CSS)
Frekans (Türkiye-Avrupa)	868 MHz-ISM Bandı
Kanal Bant Genişliği	125-500 KHz
Veri İletim Hızı	0.018 – 37.5 kbps
Link Budget	155-170 Db
Paket Boyu	11-242 byte
İletim Durumunda Güç Tüketimi	120 mA
Alıcı Durumunda Güç Tüketimi	14 Ma
Uyku Modunda Güç Tüketimi	4 μ A
İnterference Karşı Bağışıklık	Düşük giriş güçlerinde çalıştığından interference karşı yüksek bağışıklık

Çizelge 2.1.’e dikkat edilirse veri iletim hızları oldukça düşüktür. Aslında bu istenilen bir durumdur. Çünkü IoT uygulamalarında yüksek veri hızlarına ihtiyaç yoktur. Bu da güç tüketimini azaltıyor ve uzak mesafelere iletimi olanaklı kılıyor.

LoRa teknolojisi hakkında konuşulduğunda LoRa ve LoRaWAN genellikle birbiri ile karıştırılıyor. LoRa bu işin fiziksel katmanını temsil ederken, LoRaWAN dediğimiz bu işin ikinci katmanı (link layer) oluyor ve uygulamaları direk olarak bu katman üzerinde koşturulur. Şekil 2.2.'de LoRa protokolünün katman yapısı gösterilmiştir [8].



Şekil 2.2. LoRa protokol katman yığınları

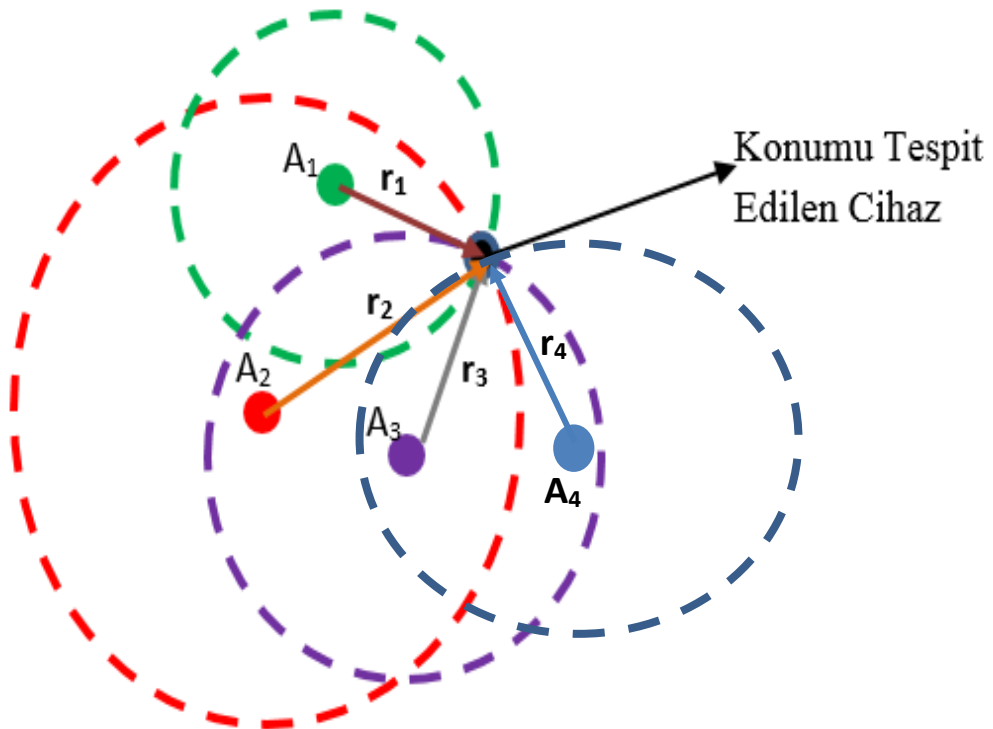
LoRa ve LoRaWAN hakkındaki bu açıklamalardan sonra projede LoRa modüllerinin hangi amaçla, nasıl kullanılacağı açıklanabilir. Öncelikle LoRa modülleri hem alıcı hem de verici durumunda çalışabilen haberleşme modülleridir. Bu cihazlar sadece kendi aralarında haberleşme kurarlar. Açık alanlarda 5 km kadar sağlıklı (0.3kbps için) iletişim mümkündür. Bu da hayvanlarını salan bir çoban için sürüsünü takip edebileceği yeterli bir menzildir. Sürü içerisinde bir hayvana takılan LoRa modülü menzil içerisindeki diğer LoRa modüllerine sinyaller gönderir. Bu sinyallerin ne olduğundan ziyade iletimi esnasında sinyal gücünde meydana gelen değişim bizim için önemlidir. LoRa modülünden alınan sinyal gücü LoRa ile SPI haberleşmesi yapan mikroişlemci üzerinde hesaplanarak iletimdeki yol kayıplarından modüller arasındaki mesafe tespit edilir. Bunu yapabilecek farklı konum tespit algoritmaları vardır. Bu algoritmalar temelde sinyalin farklı özelliklerini kullanarak mesafeyi tespit etmeye yarar.

2.2. LoRa ile Konum Tespitinde Kullanılabilecek Yöntemler

2.2.1 RSSI (Alınan Sinyal Gücü Göstergesi)

RSSI değeri alınan işaretin seviyesini belirten bir büyüklüktür. Her yonga üreticisi, farklı bir RSSI aralığı tanımlamaktadır. Bu değer aralığı 0 ile 255 arasında tanımlanmakta ve 0 alınan en iyi işareti simgelemektedir. RSSI göreceli bir değerdir ve hesaplamalarımızda kullanabilmemiz için dBm değerine çevrilmesi gerekmektedir. Bununla birlikte, genellikle donanım sürücülerini yongaya özel olarak geliştirildiği için, göreceli RSSI değeri ile birlikte dBm biriminde RF güç değerini de sunmaktadırlar [9].

RSSI yöntemiyle üç boyutlu konum tespitinin yapılabilmesi için en az 4 verici ve 1 alıcıya ihtiyaç vardır. Bu yöntemde sinyalin yayılacağı ortamın parametreleri ile yayılan sinyalin gücünün bilinmesi durumunda alıcıda ölçülen sinyal gücüne bakarak mesafeyi kestirebilmek mümkündür. Bu yöntemde konumunu tespit etmek istediğimiz hedef düğümün diğer dört vericiye olan doğrusal mesafelerini yarıçap kabul edecek çember çizildiğinde, çemberlerin tümünün kesiştiği nokta hedef cihazın konumu olmaktadır. Şekil 2.3.'te RSSI yöntemi için alıcıyla vericilerin konumunun belirlenmesine ait sistem diyagramı gösterilmiştir.



Şekil 2.3. RSSI ile konum tespiti

Şekil 2.3.'te ilgili sistemde alınan sinyalin gücü, verici antenin çıkışındaki sinyalin gücü, alıcı ve verici anten kazançları, yayılan sinyalin frekansı, alıcı-verici arasındaki mesafeye bağlı farklı yol kayıp modelleri için RSSI bağlı mesafe eşitlikler tanımlamak mümkündür.

RSSI yönteminde hedef düğüm ile alıcılar arasındaki mesafeyi tespit edebilmek için bilinmesi gereken bir diğer parametrede ortam kayıplarıdır. Ancak bu kısımda ortam kayıpları için tek eşitlik tanımlamak mümkün değildir. 2.3'de farklı yayılım modelleri için ortam kayıpları detaylı olarak ele alınacaktır. İlgili başlık altında görülecektir ki farklı her ortam için RSSI ve mesafede farklı eşitliklerle çıkarılacaktır.

RSSI yönteminde ($P_T - RSSI = L$) verici sinyalin gücü (P_T) ve RSSI değeri bilindiği takdirde ortam kayıpları (L) bulunabilir. RSSI değeri basit donanımlarla bulunabildiğinden bu yöntem konum tespitinde yaygın olarak kullanılabilir. Temelde alınan sinyal gücü (RSSI) ile verici anten çıkışındaki işaretin güçleri arasındaki fark ortam kayıplarına (L) eşit olacaktır. Ortam kayıpları da temelde mesafenin bir fonksiyonudur.

RSSI yöntemin en büyük dezavantajı ortamdaki değişimlerden etkilenmesidir. Bunun yanında hem alıcı hem de verici sabit noktalarda olsun yine de RSSI değeri sabit kalmayıp belli bir aralıkta değişim gösterebilir. Oysaki mesafe sabit ancak verici ile alıcı arasına bir nesne veya bir hareket olmasından etkilendiğinden RSSI değeri de değişebilir. Bu çalışmada bu gibi durumların etkisini en aza indirmek için en az 10 adet RSSI ölçümlerinin ortalaması alınıp elde edilen bu değer üzerinden Denklem (2.1) ifadesindeki ortam kayıpları elde edilebilir. Ardından hangi yayılım modeli için çalışılıyorsa o kanala ait ortam kayıp modeli üzerinden alıcı-verici arasındaki mesafe kestirilebilir.

$$P_T - RSSI = G_T + G_R - L \quad (2.1)$$

Denklem (2.1) ifadesinde P_T verici anten çıkışındaki gücü, RSSI alıcı antene ulaşan sinyalin gücü, G_T ve G_R sırasıyla verici ile alıcı anten kazançları, L ise ortam kayıplarıdır. Denklem (2.1) ifadesinde P_T , G_T , G_R biliniyor, RSSI bulunursa tek bilinmeyen mesafenin bir fonksiyonu olan L 'dir. Ortam kayıpları L için öncesinde de belirttiğimiz gibi tek bir tanım yoktur. 2.3 başlığı altında her kanal modeli için Denklem (2.1) ifadesi yeniden elde edilecektir.

2.3 Kablosuz Yayılım Modelleri

Bu çalışmada LoRa haberleşme teknolojisi üzerinden elde edilecek RSSI bilgisi ile kırsal alanlarda mesafe tespiti ardından da konum tespiti yapılacaktır. RSSI ifadesi dilimize alınan işaretin güç belirteci olarak çevrilir. Her yonga üreticisi, farklı bir RSSI aralığı tanımlamaktadır. Bu değer aralığı 0 ile 255 arasında tanımlanmakta ve 0 alınan en iyi işareti simgelemektedir. RSSI göreceli bir değerdir ve hesaplamalarımızda kullanabilmemiz için dBm değerine çevrilmesi gerekmektedir [9].

RSSI ifadesinden mesafe bilgisi çıkartılabilmesi için farklı kablosuz yayılım modelleri tanımlanmıştır. Bunlar sırasıyla bu başlık altında incelenecektir. Yaygın olarak kullanılan kablosuz yayılım modelleri:

- ITU Kapalı Alan İşaret Modeli,
- Serbest Uzay Yayılım Modeli,
- İki Işınlı Yayılım Modeli,
- Log Normal Gölgeleme Modeli

Olarak listelenebilir. Her modelin birbirine göre avantajları ve dezavantajları vardır.

2.3.1 ITU Kapalı Alan İşaret Modeli

ITU kapalı alan işaret modeli, mesafeye ve ortam parametrelerine bağlı olarak alıcıdaki işaret gücünü ortaya koyan, ofis, ev gibi farklı ortamlarda ve farklı çalışma frekanslarına göre parametreleri uyarlanabilen bir kapalı alan işaret modelidir. ITU kapalı alan işaret modeline göre oluşan güç kaybı (2.2) ifadesiyle verilmektedir [9].

$$L_{ITU} = 20 \times \log_{10}^{(f)} + N \times \log_{10}^{(d)} + L_f^{(n)} - 28dB \quad (2.2)$$

(2.2) ifadesinde L_{ITU} kapalı alan işaret modelindeki kayıp gücü, f çalışılan frekansı, N mesafe güç kayıp sabitini, d alıcı verici arasındaki mesafeyi, $L_f^{(n)}$ kat kayıp etkisi faktörünü ve n ifadesi de alıcı ile verici arasındaki kat sayısını temsil etmektedir. Bu durumda (2.2) denklemi LoRa modülünün 433 MHz'te çalıştığı düşünülürse aynı ifade (2.3) ifadesine dönüşür.

$$P_T - RSSI = 20 \times \log_{10}^{433} + N \times \log_{10}^{(d)} + L_f^{(N)} - 28dB \quad (2.3)$$

ITU modelinin Denklem (2.3)'de görülen halinde, N ve $L_f(0)$ parametreleri, çalışma frekansına ve alıcı ile verici arasındaki kat sayısına göre değişmektedir [10]. Bu parametrelerin 10-40 arasında değiştiği tüm kombinasyonlar için gerçek pozisyon ile hesaplanan pozisyon karşılaştırılarak ve en uygun parametre değerleri belirlenebilir. Konumu bilinmeyen bir verici için (2.3) ifadesinde bilinmeyen RSSI ve d kalır. RSSI değeri alıcıdan alınabilir. Bu durumda d mesafesi bilinmeyen tek parametre olur ve denklemden rahatlıkla elde edilebilir.

2.3.2 Serbest Uzay Yayılım Modeli

Bir diğer kablolu haberleşme kanal modeli de serbest uzay yayılım modelidir. Serbest uzay kayıpları (2.4) ifadesinde gösterilmiştir.

$$L_{fs} = 20 \times \log \left(\frac{4 \times \pi \times d}{\lambda} \right) \quad (2.4)$$

(2.4) ifadesinde L_{fs} serbest uzay kayıplarını, d alıcı-verici arasındaki mesafeyi, λ ise dalga boyunu ifade eder. Dalga boyu λ ise (2.5) ifadesiyle bulunur.

$$\begin{aligned} \lambda &= \frac{c}{f} \\ c &= 3 \times 10^8 \text{ m/s} \\ f &= 433 \times 10^6 \text{ Hz} \end{aligned} \quad (2.5)$$

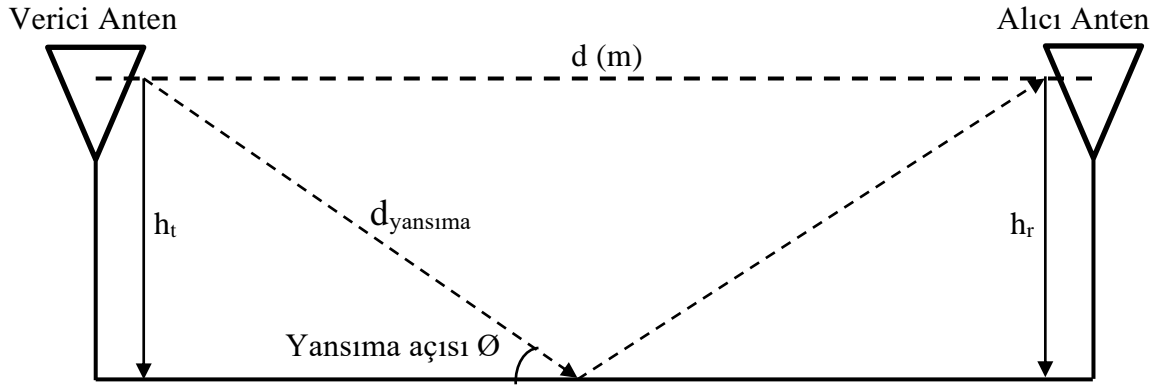
(2.5) ifadesinde dalga boyu λ hesaplanır. Bu durumda (2.1) ifadesi bu kanal modeli için (2.6) ifadesine dönüşür.

$$\begin{aligned} RSSI &= P_T + G_R + G_T - L_{fs} \\ RSSI &= P_T + G_R + G_T - 20 \times \log \left(\frac{4 \times \pi \times d}{\lambda} \right) \end{aligned} \quad (2.6)$$

Olarak elde edilir. Bu ifade de bilinmeyen tek parametre alıcı verici arasındaki d mesafesidir. Ancak çevresel değişkenleri ait parametreleri içermediğinden teorikte RSSI değeri gerçekteki RSSI değerinden çok farklı olacaktır.

2.3.3 İki Işınlı Yayılım Modeli

Bu modelde hedef cihazdan yayılan sinyalin zeminle yansımaya uğradığı ve antenler arasında doğrudan görüşün olduğu durumlar temsil etmekte başarılıdır. Bu çalışmada kırsal kesim için tasarlandığından anten arası görüşün olduğu kabul edilir. Bu şartlar altında bu modeli ilgili projeye uyarlayacak olursak Şekil 2.4.'te antenler arası görüşün olduğu yansımalı bir sistem modellenebilir.



Şekil-2.4. İki ışınımlı yayılım modeli

Şekil 2.4.'te yerden yansımalı ve doğrudan görüş açılı alıcı-vericilerden oluşan sistem gösterilmiştir. Bu sistemde ortam kayıpları iki ışınlı yayılım modeli ile açıklanır. İlgili modelde ortam kayıpları (L_{tr}), (2.7) ifadesiyle elde edilir.

$$L_{TR} = 20 \times \log \left(\frac{d^2}{h_t \times h_r} \right) \quad (2.7)$$

(2.7) ifadesinde çalışmada kullandığımız antenler özdeş olduğundan anten boyları $h_t = h_r$ olduğundan aynı ifade

$$L_{TR} = 20 \times \log \left(\frac{d^2}{h^2} \right) = 40 \times \log \left(\frac{d}{h} \right) \quad (2.8)$$

İfadesine dönüşür. Bu durumda RSSI dan mesafeyi kestirebileceğimiz (1) ifadesi bu model için (2.9) ile ifade edilir.

$$RSSI = P_T + G_R + G_T - 40 \times \log\left(\frac{d}{h}\right) \quad (2.9)$$

Olarak elde edilir. Bu ifade antenlerin kazançları ile boyları ve yayılan sinyalin gücü (P_T) bilindiğinden ve yine RSSI değeri alıcıdan okunduğunda (2.9) ifadesi tek bilinmeyenli denkleme dönüşür. Böylece alıcı ile verici arasındaki mesafe kestirilir. Bu işlem toplamda üç alıcı için yapılacak olursa konumu bilinmeyen cihazın koordinatları tespit edilebilir. Bu aşamada her bir alıcıda RSSI ölçümler en az 10 adet alınıp filtrelenerek yapılması durumunda ortamdaki değişimlerin etkisini en aza indireceğinden doğru konumu o kadar çok yaklaşılr.

2.3.4 Log Normal Gölgeleme Modeli

Log-normal gölgelendirme modeli elektromanyetik dalganın yansıma, kırınım ve saçılımlar sonucunda sinyalde meydana gelen sönümlemenin etkisini kablosuz algılayıcı ağlar için en iyi şekilde ifade eden yayılım modelidir [11]. Bu modelde elektromanyetik dalgada meydana gelen yol kaybı, genelde bir metreyi baz alarak ölçülen referans yol kaybı üzerine ortamın yol kayıp üssüyle doğru orantılı olarak mesafeye bağlı logaritmik kayıpları ekleyen RSSI tabanlı yayılım modelidir. Bu model ilave olarak ortalaması sıfır ve standart sapması ortama bağlı olarak değişen rastgele gölgeleme etkisini de modellemektedir. (2.10) ifadesinde Log-normal gölgeleme modeli için yol kayıplarını ifade eden eşitlik verilmiştir.

$$P_L(d)[dB] = P_L(d_0) + 10 \times n \times \log_{10}\left(\frac{d}{d_0}\right) + X(0, \sigma) \quad (2.10)$$

Denklem (2.10) ifadesinde $P_L(d)$ dB cinsinden toplam yol kaybını, $P_L(d_0)$ $d_0=1$ metre için referans yol kaybını, n yol kaybı üssünü, d alıcı-verici arasında metre cinsinden mesafeyi, $X(0, \sigma)$ ise sıfır ortalamalı σ standart sapmalı gauss gürültüsünü ifade eder. Bu eşitlikte toplam yol kaybını belirleyen en önemli parametre yol kaybı üssüdür. Bu değer ortama bağlı olarak değişmektedir. Çizelge 2.2’de farklı ortamlar için n değerleri verilmiştir.

Çizelge 2.2. Farklı ortamlar için yol kayıp üsleri [12]

<u>Çevresel Ortam</u>	<u>Yol Kayıp Üssü</u>
Serbest Uzay	2
Şehir içi Hücresel Radyo	2.7-3.5
Gölgelendirilmiş Şehir içi Hücresel Radyo	3-5
Görüş Açılı Bina içi	1.6-1.8
Bina İçinde Engelleri olan Ortam	4-6
Fabrika İçinde Engelleri olan Ortam	2-3

Çizelge 2.2.'de görüldüğü üzere yol kayıp üsleri ortamdan ortama değişmekte olup burada verilen değerlerin ortalama değerler olduğu bilinmelidir. Yol kayıpları alıcıda alınan sinyal gücü ile verici gücünün farkına eşittir. Bu şartlar altında (2.10) ifadesi (2.11)'de eşitlikle ifade edilebilir.

$$RSS [dBm] = P_T [dBm] - P_L(d) [dB] \quad (2.11)$$

Denklem (2.11) ifadesinde P_T verici gücü önceden bilindiği ve RSSI değerinin de bilinmesi durumunda yol kaybı $P_L(d)$ bulunabilir. Bulunan yol kaybı (2.10) ifadesine konulursa tek bilinmeyen d mesafesi eşitlikten çekilebilir. Böylece alınan sinyalin gücüne bağlı mesafeyi tespit edilebilir.

Uygulamada yol kayıp üssü ortamdan ortama değişebildiğinden Çizelge 2.2.'deki değerler çalışılan ortama en iyi şekilde tespit edemeyebilir. Dolayısıyla başlangıçta bilinen d mesafesi için (2.11) ifadesinden $P_L(d)$ bulunup ardından (2.10) ifadesinin bilinmeyen olan yol kayıp üssü (n) bulunabilir. Böylece n için daha doğru bir kestirim yapılmış olur. Sonrasında hedef düğüm bilinmeyen d mesafelerine taşınsa da n 'in bilindiği durum için (2.11) ifadesinde yol kayıplarını ardından (2.10) ifadesinde mesafe kestirilebilir.

2.4 TOA (Time of Arrival/Varış Zamanı)

Varış zamanı (TOA) yönteminde konumu bilinmek istenen hedef cihazdan yayılan sinyalin alıcıya ulaşana kadar geçen süreyi ifade eder. Bu sürenin tespit edilmesiyle alıcı ile verici arasındaki mesafenin tespiti basitçe sinyalin ortamdaki yayılma hızıyla çarpımına eşittir.

Mesafe bilgisi bulunduktan sonra üçgenleme metoduyla konum bilgisine ulaşılabilir. Bu yöntemle iki boyutlu konum tespiti yapabilmek için en az üç verici ve bir alıcıya ihtiyaç vardır.

Tek bir verici olsa hedef ile verici arasında d_1 mesafesi olurdu. Bu durumda hedef cihaz d_1 yarıçaplı çember üzerinde herhangi bir noktada olabilirdi. Eğer iki verici olsaydı hedef cihazın bu iki alıcılara uzaklıkları sırasıyla d_1 ve d_2 olurdu. Bu durumda hedef cihaz d_1 ve d_2 yarıçaplı çemberlerin kesişim noktasında olurdu ki bu da iki farklı koordinata karşılık gelirdi. Şayet üç verici olsaydı alıcının hedef cihazlara uzaklıkları sırasıyla d_1 , d_2 ve d_3 olurdu ve bu mesafeleri yarıçap kabul eden üç adet çember çizildiğinde bunların ortak olarak tek bir kesişim noktası olurdu. İşte bu nokta konumunu belirlemek istediğimiz cihazı gösterecektir. Anlaşıldığı üzere uzayda 2 boyutlu konum tespiti için en az üç adet vericiye ihtiyaç vardır. Her bir alıcı ile verici arasındaki varış zamanında (2.12)'deki gibi hesaplanabilir.

$$\sqrt{(x - a_i)^2 + (y - b_i)^2} = c \times (t_i - t_o) = d \quad (2.12)$$

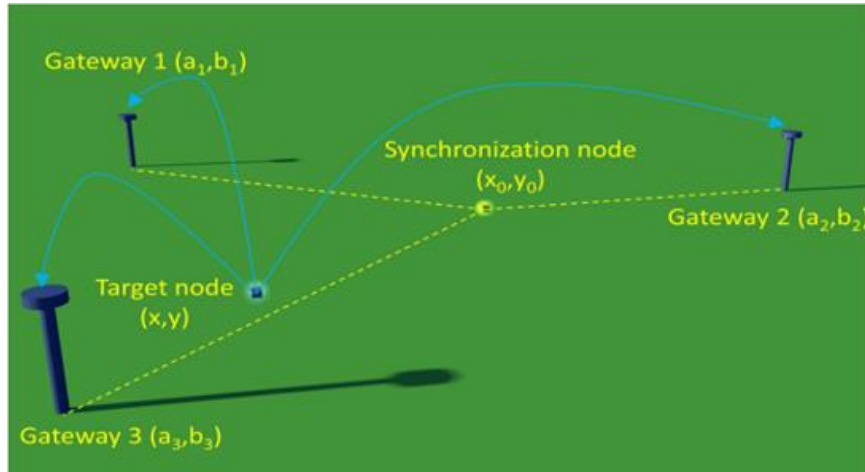
(2.12) ifadesinde (x,y) konumu tespit edilmek istenen hedef cihazın koordinatları, (a_i,b_i) konumu bilinen alıcıların koordinatlarını, t_i sinyalin alındığı anın zaman bilgisini, t_o sinyalin gönderildiği anın zaman bilgisini temsil etsin. Bu durumda iki zaman bileşeni arasındaki farka da varış zaman denilebilir. Bu durumda ışık hızı 'c' ile bu sürenin çarpımından hedef cihazın sabit noktaya olan uzaklığı 'd' hesaplanabilir

TOA yönteminde alıcıların birbiriyle senkronize çalışması gerekir. Yani alıcılar arasındaki zaman farkının bilinmesi gereklidir. Çünkü hedeften yayılan sinyal iki nokta arasında ileri-geri saat farkından etkilenmemesi gereklidir. Eğer alıcılarda senkronizasyon olmasa gerçek varış zamanında daha farklı süreler hesaplanacaktır.

Bu süreler küçük olsa bile sinyalin yayılma hızı ışık hızında olduğundan çok büyük konum hatalarının oluşmasına sebep olacaktır. Senkronizasyon gerektirmesi bu yöntemin en büyük dezavantajıdır.

2.4.1 TOA için Senkronizasyon

Konum tespitinde, senkronizasyon öncesinde de bahsedildiği üzere çoğunlukla harici veya dahili GNSS alıcılarıyla kullanılarak çözülmüştür. Çözümü gerçekten GNSS’ siz hale getirmek için tek görevi senkronizasyonu sağlamak olan ilave bir LoRa modülü daha kullanılır. Bu modül konumu öncesinden belirlenmiş diğer 3 modülün sınırları içerisinde bir noktada konumlandırılmıştır. Sistemi daha somutlaştırmak istersek konumu bilinen 3 cihazı bir üçgenin köşelerine, senkronizasyon düğümünü ise üçgenin tam orta noktasına benzetilebiliriz. Bu durumda senkronizasyon düğümlerinden köşelere olan uzaklıklar eşit olacaktır. Ayrıca sinyalin yayılma hızı da eşit olduğundan ideal durumda uç düğümlere senkronizasyon düğümünden yayılan sinyal eşit sürede ulaşacaktır. Sinyalin içinde zaman bilgisi sürekli olarak güncellenir ve alıcılar da alınan sinyal hassas şekilde işlenerek varış zamanı farkı elde edilir. Aynı fark diğer konumu belli düğümlerde de olacaktır. Böylece düğümlerin birbiri arasındaki zaman farkı mili/nano seviyelerinde, işlemcinin hızı ve enerji tüketimi arasındaki denge şartı korunarak elde edilmiş olacaktır. Şekil 2.5.’te senkronizasyon düğümü, konumu belli sabit üç adet alıcı ve hedef düğümün oluşturduğu sistem gösterilmiştir. Bu durumda ilgili sistem üzerinden senkronizasyonu bir örnek üzerinden aşağıdaki gibi açıklanabilir [13].



Şekil 2.5. Uç düğümler ve senkronizasyon düğümüyle kabaca sistemin modellenmesi

Konumu belli 4 düğüm arasındaki hesap matematiksel olarak ışık hızından elde edildiğini düşünelim ve zaman bilgisi içeren sinyalinde 16:30:-:- senkronizasyon düğümünden yayılsın. Düğümler arasındaki mesafe eşit ve 5:-:- dakikada mesafe süresi olsun.

Çizelge 2.3. Düğümler arası zaman farkının tespiti

LoRa Düğümleri	Düğümlerdeki Yerel Zaman	Sinyalin Ulaşma Zaman Farkı	Sinyalin Ulaştığı anda Düğümdeki Yerel Zaman	Düğümler Arasındaki Zaman İlişkisi
Node-1	16:35:-:-	00:05:-:-	16:40:-:-	$T + 00:02:-:-$
Node-2	16:33:-:-	00:05:-:-	16:38:-:-	T
Node-3	16:40:-:-	00:05:-:-	16:45:-:-	$T + 00:07:-:-$
Senkronizasyon Node	16:30:-:-	00:05:-:-		

Çizelge 2.3.'te görüldüğü üzere konumu belli düğümler arasındaki zaman farkı tespit edilmiş oldu. Bu durumda konumu bilinmeyen bir nesneden yayılan zaman bilgisi içeren sinyallerin düğümlere ulaştığı andaki düğümler arası zaman ilişkisi kullanarak düğümler arasındaki gerçek mesafe süresi hesaplanabilir. Sonuçta ışık hızıyla ve sürenin çarpımından sabit 3 düğüm arasındaki mesafe elde edilir. Mesafe bilgisi ise denklem (2.12) ile elde edilir.

Denklem(2.12)'de t_0 , hedef düğümün ilettime başladığı zamandır. Sinyalin yayılma zamanı ışık hızı olan c ile verilir. Bu lineer denklem sistemi x, y ve t_0 dan zaman ifadesinin yok edilmesiyle mesafe tespiti yapılır.

Çizelge 2.3.'te görüldüğü üzere zaman bilgisini çok hassas bir şekilde hesaplamak gerekir. Bunun için ek bir gerçek zamanlı sayaca (RTC) ihtiyaç vardır.

Bu sayaç öyle ki dış ortam şartlarından mümkün olduğunca etkilenmemelidir. Ayrıca RTC dışında seçilecek işlemcinin çalışma hızı da çok önemlidir. Aksi durumda konum tespitinde hata payı artarak kullanılmaz hale gelir.

2.5 TDOA (Varış Zaman Farkı)

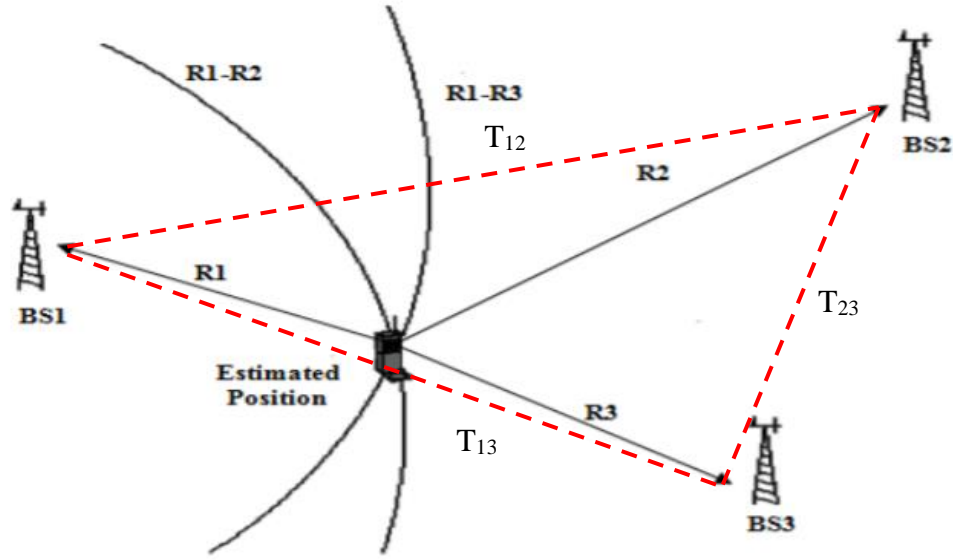
Farklı konumlandırılmış eş zamanlı alıcılara hedeften yayılan sinyal farklı zamanlarda ulaşmaktadır. Böylece alıcılar arasında bir zaman farkı oluşur. Bu zaman farkları ile hiperbolik konum hatları tanımlanmakta ve koordinatları aranan düğüm bu hatların kesişim noktası üzerinde olur.[1] Bu yöntemde alıcılar arası zaman farkından söz edilebilmesi için öncelikle iki alıcı ve bir vericinin durumunu incelenebilir.

Sistemde iki alıcı ve bir verici olması durumunda alıcılar arasında tek bir zaman farkı oluşur ve iki alıcı arasında tek bir hiperbol oluşur. Hedef düğüm, hiperbol üzerindeki herhangi bir noktada olabilir. Buradan da anlaşılacağı üzere hedef düğümün konumu tek bir hiperbol ile mümkün değil.

Sisteme ilave bir alıcı eklenirse yani iki alıcı bir verici olursa bu durumda hedef düğümünden yayılan sinyal üç farklı alıcıya üç farklı zaman farkıyla ulaşır. Her zaman farkı alıcı-verici arasında bir hiperbol tanımlar. Oluşan üç adet hiperbol en az bir noktada kesişir. Kesişimin tek nokta olması demek hedef düğümün koordinatlarının tam olarak bulunması anlamına gelir. Buradan da anlaşılacağı üzere TDOA yönteminde konum tespiti yapılabilmesi için en az üç alıcı ve bir vericiye ihtiyaç vardır. Dikkat edilirse en az bir nokta deniliyor. Çünkü alıcıların konumlandırılması esnasında bazı durumlarda üç hiperbol iki farklı noktada kesişebilir. Bu durumda ilave düğümler eklenebilir.

İlave edilen her alıcı yeni bir zaman farkı ve hiperbol tanımlar. Oluşan tüm zaman farkının hesaba katılmasıyla konum tespitin doğruluğu arttığı kadar konum tespitindeki algorithmadaki işlem yükü de bir o kadar artar.

Üç alıcı bir vericinin olduğu durumda oluşacak zaman farkları ve hiperbol hatları Şekil 2.6.'da gösterilmiştir.



Şekil 2.6. Üç alıcı ve hedef arasında oluşan hiperbolik hatlar

Şekil 2.6.'da BS1 ve BS2 alıcıları arasındaki zaman farkı T_{12} ile BS1 ve BS3 arasındaki zaman farkı T_{13} ile temsil edilirken BS2 ile BS3 alıcıları arasındaki zaman farkı ise T_{23} ile temsil edilmiştir. Oluşan zaman farkları ise (2.13)'teki gibi elde edilir.

$$T_{12} = \frac{R_1 - R_2}{c}$$

$$T_{13} = \frac{R_1 - R_3}{c}$$

$$T_{23} = \frac{R_2 - R_3}{c} \quad (2.13)$$

TDOA yönteminde TOA yöntemine göre alıcılarda senkronizasyon gerektirmemesi bir avantaj olutursa da TOA yöntemi kadar güvenilirliği yoktur. Bunun nedeni ise alıcıların konumuna bağlı bazı durumlarda TDOA da üç hiperbol iki farklı noktada kesişebilmektedir. Bu da konumun yanlış tespit edilmesine neden olacaktır.

2.6 PSO Algoritmasıyla Konum Tespiti

Parçaçık sürü optimizasyon yöntemlerinin geliştirilmesi Kennedy ve Eberhart tarafından sürü halinde hareket eden balık, kuş, arı ve karınca gibi canlıların yiyecek arayışlarında besin kaynağı ile sürü arasındaki en optimal yolu belirlemesi, sürü bireylerinin birbiriyle haberleşmesi ve güvenlik amaçlı sergiledikleri davranışların incelenmesiyle geliştirilmiştir [14]. Bu incelemeler sonucunda görülmüştür ki sürü halinde yaşayan canlıları besin ve su gibi ihtiyaçlarını karşılamada her bir bireyin sürüden bağımsız hareket etmesinin hedefe ulaşmada daha hızlı olduğu görülmüştür. PSO algoritması da sürü halinde yaşayan canlıların birbirleriyle haberleşmesinden esinlenerek geliştirilmiştir. Geliştirilen bu algorithmada sürüyü oluşturan canlılar hedefe ulaşabilecekleri en iyi çözümü bulmak adına arama yapan parçaçık adı verilen elemanlardan oluşur. Bu elemanların oluşturduğu topluluk ise sürü olarak adlandırılır. Parçaçıklar arama uzayında bir önceki arayışlarından ders çıkararak sürekli olarak güncellenen hız ve konum fonksiyonlarıyla hareket ederler. Parçaçığın bir sonraki konumunu elde etmek için (2.14) ve (2.15) eşitlikleri kullanılır [11].

$$V_{i,j}^{k+1} = w \times V_{i,j}^k + \mathcal{E}_1 r_1 (x_{best_{i,j}}^k - x_{i,j}^k) + \mathcal{E}_2 r_2 (x_{gbest_j}^k - x_{i,j}^k) \quad (2.14)$$

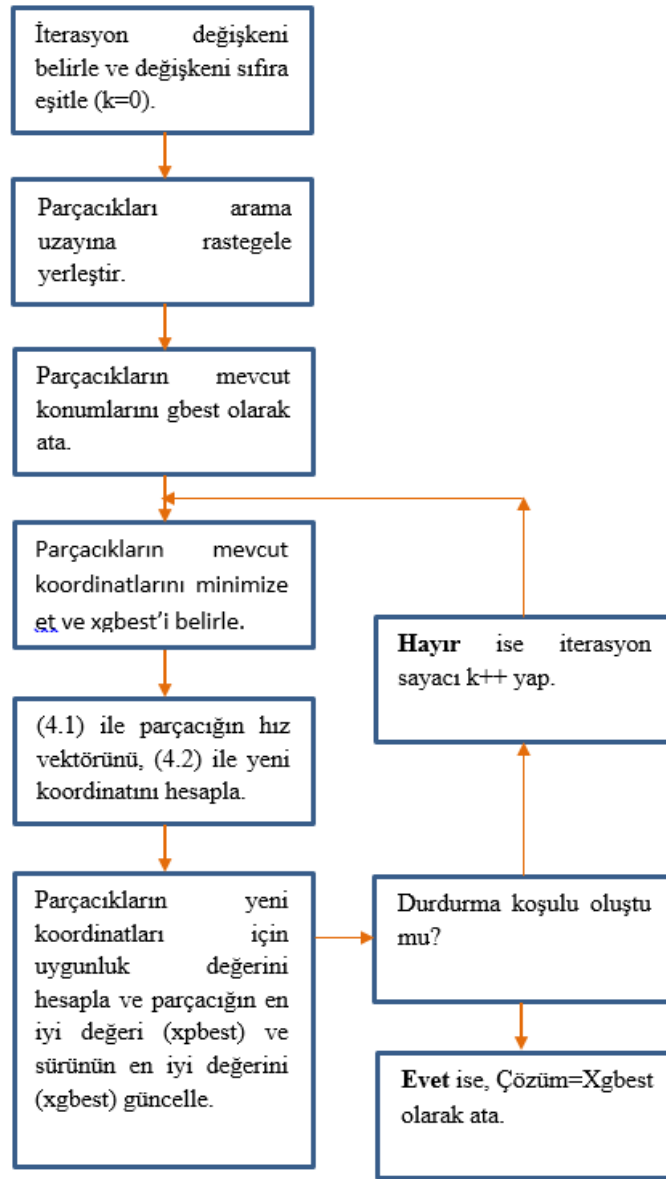
$$i = 1, 2, \dots, n; j = 1, 2, \dots, N$$

$$X_{ij} = X_{ij} + V_{ij} \quad (2.15)$$

(2.14) ifadesinde X_{ij} ve V_{ij} sırasıyla k.iterasyonun i.parçaçığının j.elemanını ifade eden konum ve hız vektörleridir. W parçaçığın eylemsizlik katsayısını ifader \mathcal{E}_1 ve \mathcal{E}_2 ise sırasıyla parçaçığın bilişsel katsayısı ile sürünün sosyal katsayısıdır. Benzer şekilde r_1 ve r_2 ise narmol dağılım rastgele değişkenleri $x_{best_{i,j}}^k$ ve $x_{gbest_j}^k$ ise sırasıyla parçaçığın en iyi konumu ile sürünün en iyi konumunu ifade eder. Parçaçık bulunduğu konuma göre en iyi konumuna oda sürünün en iyi konumuna ulaşınca kadar (2.14) ifadesinden bir hız vektörü türetilir. Hız vektörü parçaçığın birim süre içerisindeki yer değiştirmesine eşit olur. Bu durumda parçaçığın güncellenmiş konum vektörü (2.15) ifadesiyle elde edilir.

Denklem(2.15)'te ne zaman hız vektörü sıfıra eşit olur bu durumda parçaçık en iyi konumunu ulaşır Bu da ancak $x_{best_{i,j}}^k$ ve $x_{gbest_j}^k$ bulunduğu terimlerden sıfır değeri gelmesiyle oluşur. Böylece en optimal sonuca ulaşılmış olur.

PSO yönteminde görüldüğü üzere amaç parçacıkları arama bölgesinde bir v hızıyla hareket ettirmektir. Bu hareket bazı durumlarda alanın dışına çıkacak boyutta olabilir. Oysaki biz çalıştığımız alanın dışında bir optimal bir çözümün mümkün olmadığını biliyoruz. Bu durumun önüne geçebilmek için parçacığın hız vektörünün belirli aralıkta kalması sağlanır. Bu aralık $(V_{max}, -V_{max})$ şeklindedir. Bu açıklamalar doğrultusunda PSO yönteminin algoritması Şekil 2.7.'te ifade edildiği gibi oluşturulabilir.

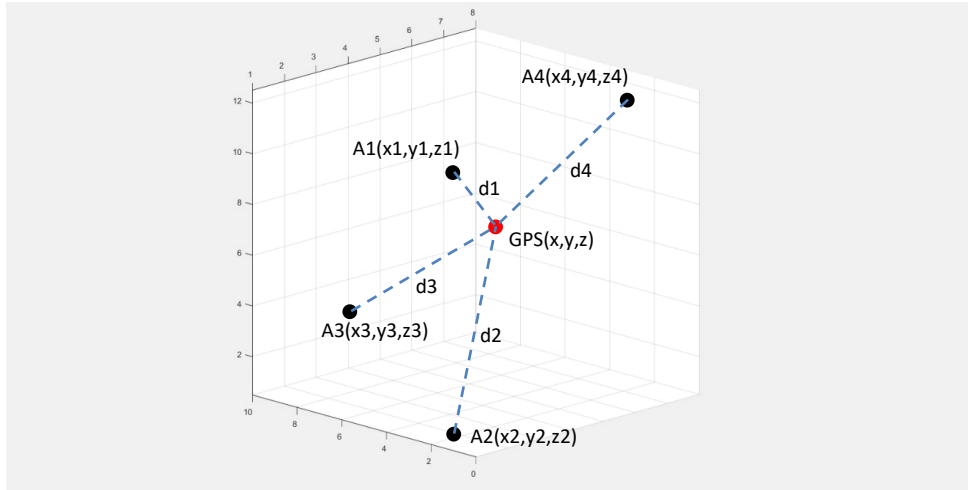


Şekil 2.7. PSO yöntemi algoritma akış diyagramı

Denklem (2.14)'te verilen eşitlikte w eylemsizlik katsayısı (0.9-0.4) aralığında alındığı durumda algoritmanın başarımlı kriteri artmıştır. Benzer şekilde yapılan çalışmalar sonucunda parçacık ve sürünün bilişsel katsayıları ise 2 değerini, dağıtılmış olarak yerleştirilen parçacık sayısı da 2-50 aralığına yakınsamaktadır.

2.7 Trilaterasyon Yöntemi ile Konumun Bulunması

Trilaterasyon yöntemi açık alanlarda veya kapalı alanlarda takip edilmek istenen cihazın konumunu belirlemede sıklıkla tercih edilen yöntemlerden biridir. GPS uyduları da arka planda herhangi bir nesnenin konumunu belirlemede bu yöntemi kullanırlar. Sistemin hedef cihazın konumunu başarılı bir şekilde bulabilmesi için bazı verilerin önceden bilinmesi gerekir. Bu veriler verici durumunda düğümlerin koordinatları ve alıcı (hedef) cihaz ile arasındaki mesafe bilgisidir. Eğer 2 boyutlu bir konum tespiti yapılacak ise en az üç adet verici düğümün konumuyla hedef cihaza olan uzaklıkları bilinmelidir. Benzer şekilde 3 boyutlu bir konum tespiti yapılacaksa da bu sefer en az dört verici cihazın konumu ve hedef cihaza olan uzaklığı bilinmelidir. Şekil 2.8'de verici düğümlerin koordinatları ve düğümler arası mesafeler bilinmesi durumunda hedef cihazın konum tespit gösterilmiştir.



Şekil 2.8. Üç boyutlu trilaterasyon

Bu yöntemde her bir düğümü merkez kabul eden ve hedef cihaza olan uzaklığında yarıçap kabul eden çemberlerin olduğu düşünülür. Her bir çember denklemi için (2.10), (2.11), (2.12) ve (2.13) eşitlikleri elde edilir.

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = d_1^2 \quad (2.10)$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = d_2^2 \quad (2.11)$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = d_3^2 \quad (2.12)$$

$$(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 = d_4^2 \quad (2.13)$$

İlgili denklem sistemi 3 bilinmeyenli lineer denklem sistemine indirgenebilir. (2.14), (2.15) ve (2.16)'da basitleştirilmiş halleri verilmiştir.

$$2(x_2 - x_1)x + 2(y_2 - y_1)y + 2(z_2 - z_1)z = (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) - (z_1^2 - z_2^2) \quad (2.14)$$

$$2(x_3 - x_1)x + 2(y_3 - y_1)y + 2(z_3 - z_1)z = (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) - (z_1^2 - z_3^2) \quad (2.15)$$

$$2(x_4 - x_1)x + 2(y_4 - y_1)y + 2(z_4 - z_1)z = (d_1^2 - d_4^2) - (x_1^2 - x_4^2) - (y_1^2 - y_4^2) - (z_1^2 - z_4^2) \quad (2.16)$$

(2.14), (2.15) ve (2.16) ifadelerinden oluşan denklem takımında bilinmeyen sayısı denklem sayısına eşit olduğundan Cramer yöntemiyle X, Y ve Z bileşenleri bulunabilir. Bu yöntemin uygulanması sonucunda X, Y ve Z bileşenlerini veren denklemler (2.17), (2.18) ve (2.19)'da verilmiştir.

$$\begin{aligned}
& (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) - (z_1^2 - z_2^2) \quad 2(y_2 - y_1) \quad 2(z_2 - z_1) \\
& (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) - (z_1^2 - z_3^2) \quad 2(y_3 - y_1) \quad 2(z_3 - z_1) \\
X = & \frac{(d_1^2 - d_4^2) - (x_1^2 - x_4^2) - (y_1^2 - y_4^2) - (z_1^2 - z_4^2) \quad 2(y_4 - y_1) \quad 2(z_4 - z_1)}{2(x_2 - x_1) \quad 2(y_2 - y_1) \quad 2(z_2 - z_1)} \quad (2.17) \\
& 2(x_3 - x_1) \quad 2(y_3 - y_1) \quad 2(z_3 - z_1) \\
& 2(x_4 - x_1) \quad 2(y_4 - y_1) \quad 2(z_4 - z_1)
\end{aligned}$$

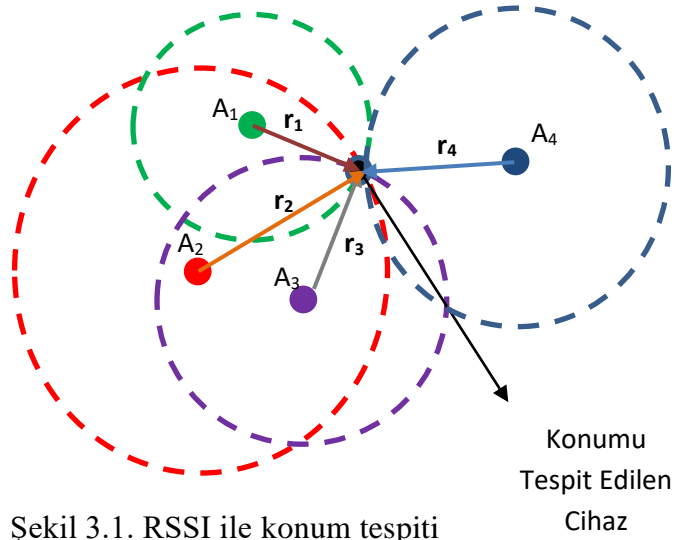
$$\begin{aligned}
& 2(x_2 - x_1) \quad (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) - (z_1^2 - z_2^2) \quad 2(z_2 - z_1) \\
& 2(x_3 - x_1) \quad (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) - (z_1^2 - z_3^2) \quad 2(z_3 - z_1) \\
Y = & \frac{2(x_4 - x_1) \quad (d_1^2 - d_4^2) - (x_1^2 - x_4^2) - (y_1^2 - y_4^2) - (z_1^2 - z_4^2) \quad 2(z_4 - z_1)}{2(x_2 - x_1) \quad 2(y_2 - y_1) \quad 2(z_2 - z_1)} \quad (2.18) \\
& 2(x_3 - x_1) \quad 2(y_3 - y_1) \quad 2(z_3 - z_1) \\
& 2(x_4 - x_1) \quad 2(y_4 - y_1) \quad 2(z_4 - z_1)
\end{aligned}$$

$$\begin{aligned}
& 2(x_2 - x_1) \quad 2(y_2 - y_1) \quad (d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) - (z_1^2 - z_2^2) \\
& 2(x_3 - x_1) \quad 2(y_3 - y_1) \quad (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) - (z_1^2 - z_3^2) \\
Z = & \frac{2(x_4 - x_1) \quad 2(y_4 - y_1) \quad (d_1^2 - d_4^2) - (x_1^2 - x_4^2) - (y_1^2 - y_4^2) - (z_1^2 - z_4^2)}{2(x_2 - x_1) \quad 2(y_2 - y_1) \quad 2(z_2 - z_1)} \quad (2.19) \\
& 2(x_3 - x_1) \quad 2(y_3 - y_1) \quad 2(z_3 - z_1) \\
& 2(x_4 - x_1) \quad 2(y_4 - y_1) \quad 2(z_4 - z_1)
\end{aligned}$$

3. TASARIM

3.1. Genel Bilgiler

Şekil 3.1.'deki sistemde konumu belli olan 4 (A_1, A_2, A_3, A_4) düğüm bulunacaktır. Bu düğümler sürekli olarak kimliklerini, konumlarını ve diğer bilgileri yayınlayacaktır. Bu düğümler ile konumunu tespit etmek istediğimiz düğüm haberleşecektir. Log Normal Gölgeleme Modelini kullanarak iki cihaz arasındaki uzaklık değeri bulunacaktır. 2.3.4 Log Normal Gölgeleme Modeli'nde bahsedilen (2.10) ve (2.11) formülleri ile uzaklık değeri bulunacaktır.

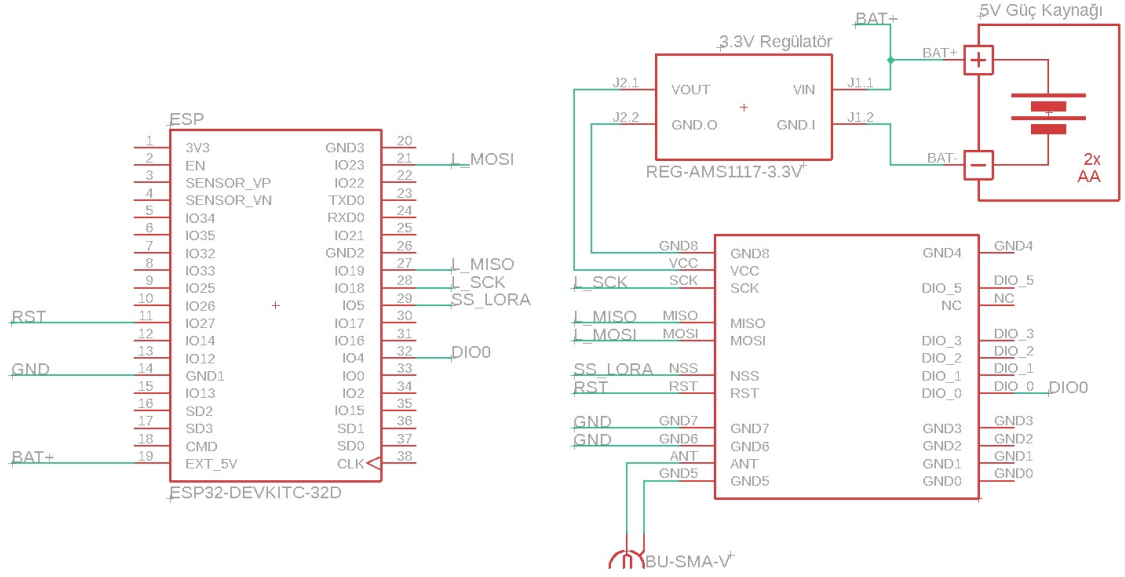


$$P_L(d)[dB] = P_L(d_0) + 10 \times n \times \log_{10} \left(\frac{d}{d_0} \right) + X(0, \sigma) \quad (2.10)$$

$$RSS [dBm] = P_T[dBm] - P_L(d)[dB] \quad (2.11)$$

Konumu tespit edilmek istenen düğüm, konumu sabit olan 4 düğümün konum ve uzaklık değerlerine sahip olduktan sonra Trilaterasyon Algoritmasıyla konumu hesaplayacaktır.

Şekil 3.2.'de ESP32 ve LoRa modüllerinin Eagle CAD üzerinde bağlantı şeması gösterilmiştir.

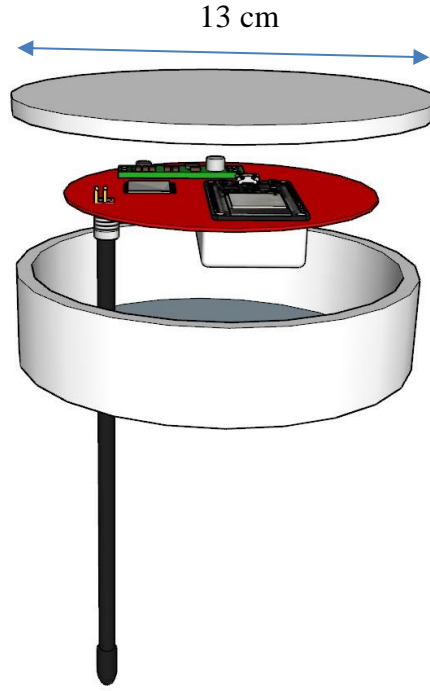


Şekil 3.2. Sistem şematiği-EagleCAD

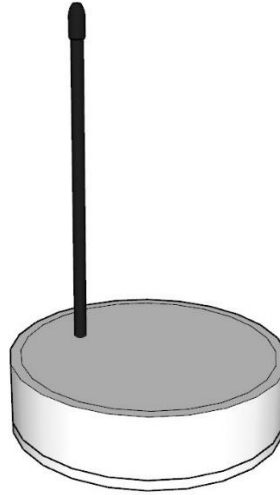
3.2. Boyutlandırılmalar

Projenin karmaşıklığını azaltmak ve güvenliğini artırmak için devre kartı ve cihaz kutusu üretilecektir. Devre kartı tasarımı EagleCAD programında yapılmıştır. Devre kartı şematiği Şekil 3.2'deki gibidir. Devre kartı DIP olarak planlandığından prototipleme ve bakım aşamasında kolaylık sağlayacaktır. Devre kartı boyutu Şekil 3.4'teki gibi 10 cm çapında yuvarlak olarak planlanmıştır.

Devre kartını ve pilleri dış etmenlerden ve sudan korumak için cihaz kutusu üretilecektir. Devre kartından sistem zorunluluğundan dolayı anten cihaz kutusuna dışarıdadır. 3D yazıcıdandan ABS filamentten üretilecek cihaz kutusu üzerinde açma-kapama düğmesi bulunacaktır. Cihaz kutusu ölçüleri çapı 13 cm, yüksekliği 5 cm ve kutu dışarısında kalan anten uzunluğu 11 cm'dir. Cihaz kutusu görünüşü Şekil 3.3'te gösterilmiştir.



Şekil 3.3. Sistem patlatılmış kutu prototipi SketchUp



Şekil 3.4. Sistem kapalı çalışırken prototipi

3.3. Sistem Bileşenleri ve Seçimleri

Yapılacak olan konum tespit cihazı ana bileşenler olarak elektronik devre kartı üzerinde bulunan mikrodenetleyici, haberleşme modülü ve bataryadan oluşuyor.

3.3.1 Mikrodenetleyici

Mikrodenetleyiciler otomatik kontrol amacıyla maliyeti ve düşük güç tüketimi gibi faydalarıyla tercih edilmektedir. Dışarıdan gelen bir veriyi içerisindeki hafızaya kaydeden, çalıştıran ve bir çıktı veren sistemin beyni olarak çalışan bir bilgisayardır. Yapısında CPU, RAM, ROM, I/O pinleri, seri ve paralel portlar ve bazı türlerinde ADC bulunur. Bu projede mikrodenetleyiciye gelen veriler her bir düğümden gelen RSSI değeri okuyup konumu tespit edecek algoritmada işleyerek konumu hesaplar ve çıktı verir. Böylece konum tespitini yapar.

3.3.2. Haberleşme Modülü

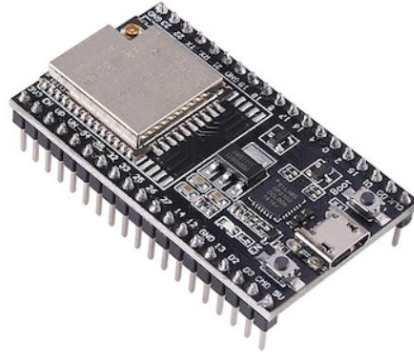
Haberleşme modülleri bağlı oldukları cihazların göndermek istedikleri verileri radyo dalgalarıyla birbirlerine yollayarak haberleşmesini sağlar. Sahip oldukları frekans ve modülasyon özellikleri sebebiyle birbirlerinden ayrılırlar. Çeşitli antenler takılarak sinyal iletme mesafeleri artırılabilir.

3.3.3. Batarya

Cihazda kullanılacak olan batarya sistemin taşınabilir olmasını ve istediğimiz yerde çalışmasını sağlar. Batarya kapasitesi arttıkça cihaz daha uzun süre çalıştırılabilir.

3.3.4. Mikrodenetleyici Seçimi

Mikrodenetleyici seçilirken ilk öncelik yeterli işlem gücü, kaynak bulunabilirliği, fiyat, erişilebilirlik ve piyasada kullanım oranı göz önüne alındı. Bunun sonucunda ESP32 ailesi seçildi. Bu aileden olan Şekil 3.5.'te gösterilen ESP32-DEVKITC-32U modülü seçildi. Çizelge 3.1.'de ise mikrodenetleyicinin özellikleri verilmiştir DIP Modülü seçilmesinin sebebi hem prototip gerçekleştirmesinin kolaylığı hemde piyasada ucuz fiyata hızlıca erişilebilmektir.



Şekil 3.5. ESP32-Wroom-32U mikrodnetleyicisi

Çizelge 3.1. ESP32-DEVKITC-32U donanım özellikleri

Mikrodnetleyici	ESP32-Wroom-32U
Çekirdek	ESPG2-DOWD
Paket Tipi	LQFP
Çalışma Gerilimi	3,3V- 5V
Analog Girişler	10 (12-bit)
Dijital IO Pinleri	37
Çalışma Frekansı	40 MHz
Flash Hafıza	4 MB
SRAM	3(16-bit), 1 (PWM)
Timers	3(16-bit), 1 (PWM)
Desteklenen İletişim Protokolleri	I2C, SPI, UART, CAN, USB (2x SPI, 3x USART, 2x I2C, 1x CAN)

3.3.5. Haberleşme Modülü Seçimi

Haberleşme modülü seçerken geniş alanda çalışacağımız için yüksek çıkış gücüne sahip, ulaşılabilirliği kolay, bol kaynak dosyasına sahip olmasına önem verildi. Ve bu şartları sağlayan Şekil 3.6.’te gösterilen “EBYTE LORA Module.SPI. 20dBm. 5km. SX1278. 410-525MHz” seçildi. Çizelge 3.2.’de bu modülün donanımsal bilgileri verilmiştir.



Şekil 3.6. LoRa E19 SX1278 modülü

Çizelge 3.2. E32-433T20DC LoRa modülü donanımsal özellikleri

Haberleşme Arayüzü	SPI
Besleme Gerilimi	3.3 V
Çalışma Frekansı	410-525 MHz
Güç Tüketimi	İletirken 128 mA Alırken 14 mA Kapalı 2uA
Maksimum TX Gücü	20 dBm
Alıcı Hassasiyeti	-147 dBm
Boyut	15*15 mm
Air data range	1.2 - 300 Kbps FSK 0.018 – 37.5 Kbps LoRaTM
TX Uzunluğu	256 Byte
Anten	SMA-K (50 Ohm Empedans)

Haberleşme modülüne takacağımız anten ise empedans olarak uyumlu ve kullanılacak yönteme göre çeşiti seçilmesi gerekmektedir. Antenlerin yönsüz olması daha uygun görülmüştür. Böylece gezici cihaz nerede olursa olsun haberleşmede kolaylık sağlanması amaçlanmıştır.

Bu sebeple Şekil 3.7.’da gösterilen “4.0 dBi SMA 433 MHz Anten TX433-JKD-20P” seçilmiştir. Çizelge 3.3.’de seçilen antenin donanımsal bilgileri verilmiştir.



Şekil 3.7. 4 dBi SMA anten

Çizelge 3.3. 4 dBi SMA antenin donanımsal özellikleri

Frekans	433 MHz
Kazanç	4 dBi
Polarizasyon	Yönsüz (yatay)
Empedans	50 Ohm
Güç Kapasitesi	20 W
Ağırlık	21 g

3.4. Uygulanan Yöntemler

3.4.1 Devre Tasarım Yöntemleri

3.4.1.1. Breadboard Devre Tasarımı

Breadboard ile devre tasarımı yöntemi en basit devre tasarım yöntemleri arasındadır. Breadboard yapısı dikine birbirine bağlı olan jumperlar ile birbirine bağlanabilen bir ekipmandır. Böylece DIP Modulleri kolay bir şekilde bir tahta üzerinde kullanabiliriz.

Bu yöntem bize hızlı prototipleme ve test imkânı sağlamaktadır. Ancak devre kolay bir şekilde bozulabilmesi, kısa devre ihtimali, karmaşıklık dezavantajları arasında yer almaktadır.

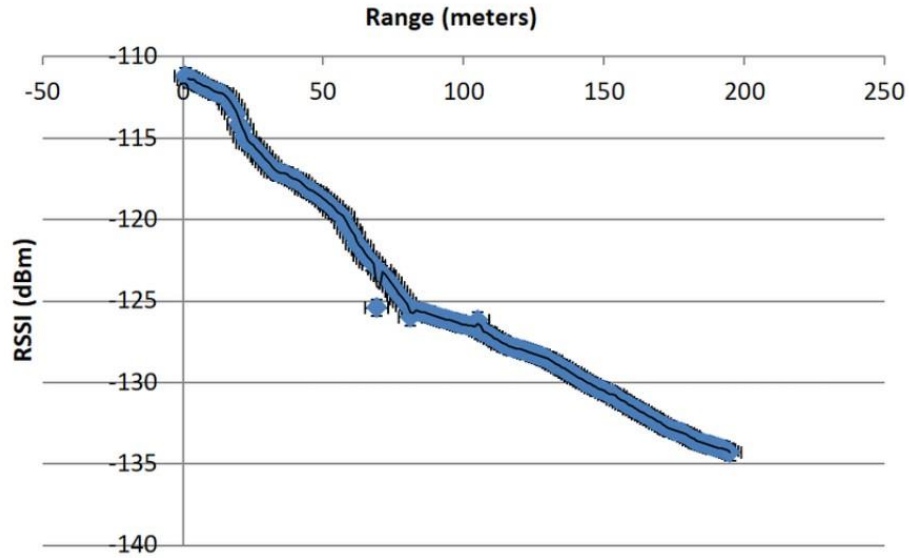
3.4.1.2. PCB/Eagle Devre Tasarımı

PCB devre tasarımı yöntemi breadboarda göre daha profesyonel ve gelişmiş bir yöntemdir. PCB, elektronik bileşenlerini bir arada tutan bir karttır. İçerisinde bulunan o sisteme özel elektrik yolları sayesinde karmaşıklık ortadan kalkmaktadır. Prototipleme aşaması temel olarak aşıldıktan sonra PCB devre kartı sayesinde daha az karmaşık ve güvenilir bir devre elde edilir. Ancak bu yöntemin üretim süresinin uzunluğu ve fiyat dezavantajları arasındadır.

3.4.2 Haberleşme Yöntemleri

3.4.2.1 RSSI Hesaplama Yöntemi

RSSI gelen sinyal gücünü gösteren bir değerdir. Her üretici kendi RSSI değerini 0 ile 255 arasında farklı şekilde kabul eder. RSSI hesaplanmadan önce dBm olarak hesaplanır. Sinyal gücü düştükçe RSSI'nin dBm değeri eksi yönde düşer. Haberleşme modüllerinde RSSI değerini hesaplanabilir olduğundan ek donanıma ihtiyaç yoktur. Bu yüzden konum bulma metodları için RSSI değerini kullanmak ekonomik bir yöntemdir. Şekil 3.8.'de mesafe arttıkça alınan sinyalin gücünde eksi yönlü bir değişim görülür [15].



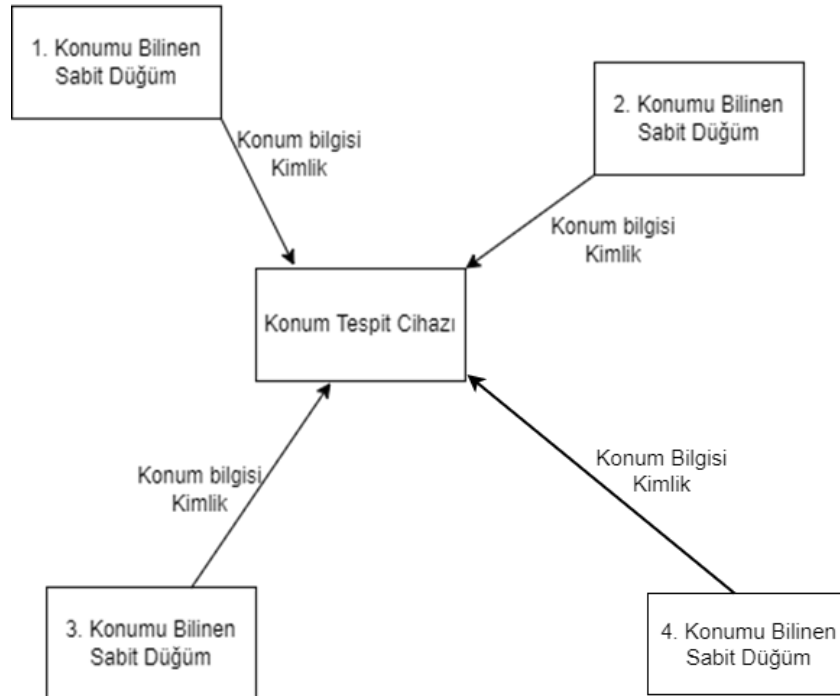
Şekil 3.8. Uzaklığa bağlı RSSI grafiği

3.4.2.2. LoRa Haberleşme Yöntemi

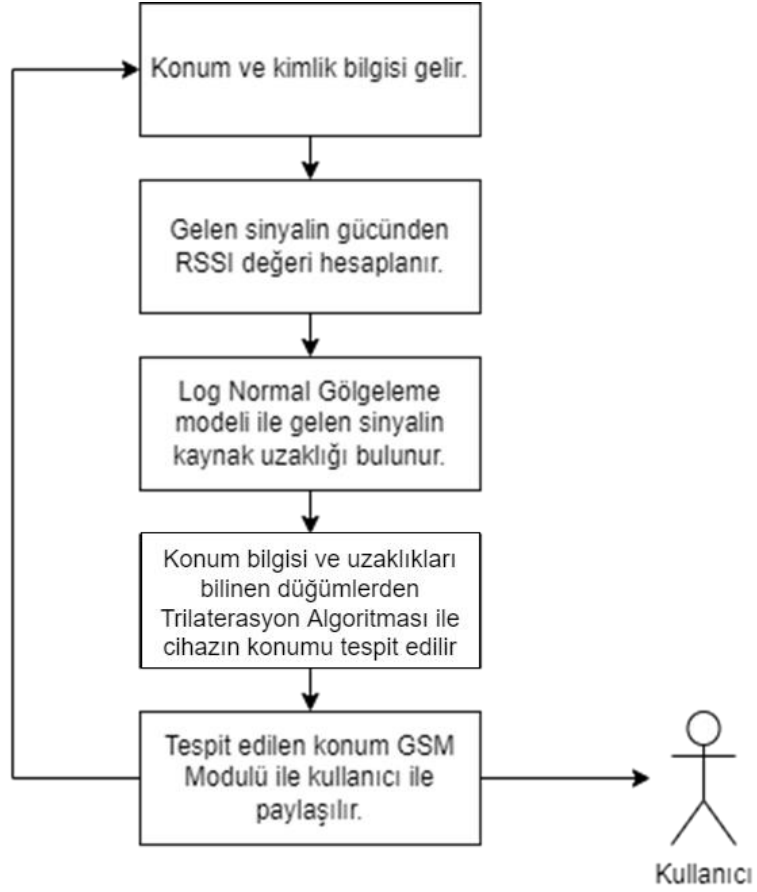
LoRa FSK modülasyonunun farklılaştırılmasıyla oluşturulan yeni bir modülasyon tekniğidir. Long Range (Uzun Mesafe) kelimelerinin ilk hecelerinden oluşuyor ve Wi-Fi, GSM ve kablolu haberleşme gibi rakip teknolojilere göre bazı avantajlara sahiptir. İnternet of Things teknolojilerinin gelişmesi ile uzak mesafeye veri aktarımı ve düşük güç tüketimi önemli bir sorun haline gelmiştir. LoRa, rakip teknolojilere göre çok daha uzun mesafede çift yönlü iletişime izin verir. Konum tespit cihazında güç tüketimi ve verimlilik sebebiyle tercih edilmiştir. LoRa haberleşme teknolojisinin en büyük avantajı pil ile çok uzun süre çalışan sensör çözümleri yapabilmeye imkân vermesidir.

3.5. Yazılımlar

Yazılım algoritmasına giriş yapmadan önce konumu tespit edilecek düğüme gelen olan veriler konum bilgisi ve bu bilginin geldiği düğüm kimliğidir. Sonrasında konum tespit cihazı algoritma işlemlerine başlayacaktır. Şekil 3.9.'de düğümlerin yerleştirilmesi, Şekil 3.10.'da konum tespiti için kullanılacak algoritmanın akış diyagramı gösterilmiştir.



Şekil 3.9. Düğümlerin yerleştirilmesi



Şekil 3.10. Konum tespiti için kullanılacak algoritmanın işleyişi

Konum tespit cihazının kullacağı algoritma şekildeki gibidir. Log Normal Gölgeleme modeli ve Trilaterasyon Algoritması ilgili başlıklarda detaylı şekilde anlatılmıştır. Çalışmanın simülasyonları MATLAB üzerinden yapılmıştır.

3.6. Malzeme Listesi ve Ekonomik Analiz

Projenin tasarım aşamasından gerçek hayatta uygulanabilir hale gelmesi için gerek iç donanımları olsun gerekse dış donanımları olsun tüm malzemeler hakkında piyasa araştırılması yapılmış olup proje için en uygun malzemeler Çizelge 3.4.'te artışıyla eksisiyle ele alınarak değerlendirilmiştir. Çizelge 3.5.'te ise projenin maliyet hesabı yapılmıştır.

Çizelge 3.4. Tasarımda kullanılacak malzemelerin piyasa değerlendirmesi

Malzemenin adı	Kullanım amacı	Olumlu Etkileri	Olumsuz Etkileri
ESP32-DEVKITC-32U	Mikrodenetleyici	-Yüksek işlem gücü -Sektörde oldukça yaygın olması -Profesyonel çalışmalarda kullanılabilirliği	- Fiyat
E19-433M20S2	Haberleşme Modülü	- LoRa teknolojisine sahip olması - Düşük güç tüketimi -IoT uygulamalarındaki popülerliği	-
4.0 dBi SMA 433 MHz Anten	Haberleşme Anteni	-Kullanılacak olan haberleşme modülü ile uyumluluğu - Yüksek kazanç	-
Samsung iForce Powerbank 10.000 mAh	Pil	- Yüksek güç kapasitesi - Boyut	-Fiyat
	Baskı Devre Kartı	- Daha düzenli ve güvenilir bağlantı kabiliyeti	-

Çizelge 3.5. Tasarımda kullanılacak malzemelerin maaliyet listesi

Malzemenin adı	Kullanım amacı	Birim fiyatı (TL)	Adedi	Fiyatı (TL)
ESP32-Wroom-32U	Mikrodenetleyici	180,53	5	902,65
E32-433M20S2	Haberleşme Modülü	171,83	5	859,15
4.0 dBi SMA 433 MHz Anten	Haberleşme Anteni	77,14	5	385,7
iForce Powerbank 10.000 mAh	Pil	150	5	750
	Baskı Devre Kartı	80	5	400
TOPLAM				3297,5

3.7. Hukuki Boyut

Projede 433 MHz uygun çıkış güçlerinde kullanılacağından hukuki olarak bir risk taşımamaktadır. GSM Modülü IMEI Kaydı olduğundan hukuki olarak risk taşımamaktadır [16], [17]

4. SİMÜLASYON ÇALIŞMALARI

4.1. Genel Bilgiler

Bağıl konum tespit çalışmalarında temelde çözülmesi gereken iki kısım vardır. Bunlardan ilki konum takibi yapılan cihazın, konumu bilinen cihazlara olan mesafeleridir. Diğerisi ise mesafe bilgilerinden koordinat eksenine geçilerek tam konumun tespitidir.

Bu kısımda alıcı ve verici arasındaki mesafenin alınan sinyal gücüne bağlı simülasyonu yapılmıştır. Bu amaçla farklı yol kayıp modelleri tanımlanmıştır. Her yol kayıp modeli için amaç sinyalin iletimi esnasında sinyalde meydana gelen zayıflamalardan mesafe bilgisini bulabilmektir. Biliyoruz ki sinyalin kat ettiği yol arttıkça sinyalde meydana gelen kayıplarda artacaktır. Literatürde çok sayıda yol kayıp modelleri tanımlanmıştır. Bu projede log normal gölgeleme modeli kullanılmıştır. Bu modele ait simülasyon çalışmaları yapılırken temelde mesafenin ve ortam kayıp üssünün etkisi incelenmiştir. Bu sebepten ötürü bu parametreler vektör formatında değişken olarak tanımlanmıştır.

4.2. Yol Kayıpları Simülasyon Çalışmaları

4.2.1. Sistem Yol Kayıplarının Modellemesi

Bu modelde elektromanyetik dalgada meydana gelen yol kaybı, genelde bir metreyi baz alarak ölçülen referans yol kaybı üzerine ortamın yol kayıp üssüyle doğru orantılı olarak mesafeye bağlı logaritmik kayıpları ekleyen RSS tabanlı yayılım modelidir. Bu model ilave olarak ortalaması sıfır ve standart sapması ortama bağlı olarak değişen rastgele gölgeleme etkisini de modellemektedir. (2.16) ifadesinde Log-normal gölgeleme modeli için yol kayıplarını ifade eden eşitlik verilmiştir [11].

$$P_L(d)[dB] = P_L(d_0) + 10 \times n \times \log_{10} \left(\frac{d}{d_0} \right) + X(0, \sigma) \quad (2.16)$$

Denklem (2.16) ifadesinde $P_L(d)$ dB cinsinden toplam yol kaybını, $P_L(d_0)$ $d_0=1$ metre için referans yol kaybını, n yol kaybı üssünü, d alıcı-verici arasında metre cinsinden mesafeyi, $X(0, \sigma)$ ise sıfır ortalamalı σ standart sapmalı gauss gürültüsünü ifade eder. Bu eşitlikte toplam yol kaybını belirleyen en önemli parametre yol kaybı üssüdür. Açık alanlarda yol kayıp üssü (n) 2 ile 3 arasında bir değer alır. $P_L(d_0)$ ise 1 metredeki referans yol kaybı olup yaklaşık olarak 48.6 dB olarak alınabilir.

Diğer yandan $X(0,\sigma)$ gauss gürültüsü ortamda anlık değişen bir parametredir. $X(0,\sigma)$ sıfır ortalamalı σ standart sapmalı gauss gürültüsünü ifade eder. Standart sapma $\sigma=3$ olarak alınırsa (2.16) ifadesi aşağıdaki gibi güncellenebilir.

$$P_L(d)[dB] = 48.6[dB] + 10 \times 2.7 \times \log_{10}(d) + X(0,3) \quad (2.17)$$

(2.17) ifadesinden mesafenin yol kaybını göre değişimini matlab üzerinde simüle etmek mümkündür. Gauss gürültüsü matlab üzerinde random fonksiyonu ile üretilebilmektedir. Mesafe bilgisi de belirli bir aralıkta belli değerlerde artan bir vektör gibi tanımlanırsa eşitlikte tek bilinmeyen yol kayıp değerleri olur. Bölümün ilerleyen kısmında matlab üzerinde log normal gölgeleme modeli için yol kayıpları simüle edilmiştir. Ayrıca bu modele ek olarak farklı yol kayıp modellerinin de değişimleri karşılaştırılmalı olarak konunun anlaşılabilirliğini arttırmak adına verilmiştir. İlgili kısımlarda bulunan grafikleri uygulamada konumu bilinen bir cihaz ile konumu bilinmeyen cihaz arasında yapılmaktadır. LoRa modüllerinden alınan işaretin sinyal gücü (RSS) doğrudan alınabilir. Ayrıca verici gücü P_T de bilindiğinden yol kaybı da bilinir. Simülasyon kısmındaki grafiklerden meydana gelen yol kaybından d mesafesinin karşılığı grafik üzerinde okunabilir.

Simülasyonun ikinci kısmını ise amaç RSSI değeri, sabit noktaların konumları ve alıcılar ile vericiler arasındaki mesafenin bilinmesi durumunda hedef cihazın koordinatının tespit edilebildiğini göstermektir. Konum tespiti için farklı yöntemler bulunmaktadır. Simülasyon çalışmalarında koordinat hesabı için PSO algoritması kullanılmıştır. Bu yöntem dışında konum tespiti için üç boyutlu trilaterasyon yöntemi de kullanılabilir. PSO yöntemde algoritmaya özgü parçacık hızı, parçacık sayısı, çalışılan alan aralığı dışında bir de (x,y,z) koordinatları bilinen dört düğüme ve her bir düğümlerle hedef cihaz arasındaki dört adet mesafe bilgisine ihtiyaç vardır. Bu bilgiler ışığında hedef cihazın konumu en düşük hata oranıyla simüle edilmeye çalışılacaktır.

4.2.2. Sistem Yol Kayıplarının Simülasyon Yazılımı

Çalışma kapsamında yol kaybının mesafeye bağlı değişimini simüle etmek için matlab üzerinde log normal gölgeleme modeli gerçekleştirilmiştir. Ancak farklı yol kayıp modellerinin mesafe etkisini görebilmek için serbest uzay kayıpları, log distance ve log

normal gölgeleme modeli bir arada yapılmıştır. Çizelge-4,5 ve 6’da çalışmanın matlab kodları verilmiştir.

Çizelge 4.1. Serbest uzay kayıpları için ortam kaybının Matlab modeli

PL_free.m
<pre>function PL=PL_free(fc,dist,Gt,Gr) lamda=3e8/fc; tmp=lamda./(4*pi*dist); if nargin>2, tmp=tmp*sqrt(Gt);end if nargin>3, tmp=tmp*sqrt(Gr);end PL=-20*log10(tmp);</pre>

Çizelge 4.2. Log distance ve log normal gölgeleme için kayıpların Matlab modeli

PL_logdist_or_norm.m
<pre>function PL=PL_logdist_or_norm(fc,d,d0,n,sigma) lamda=3e8/fc; PL=-20*log10(lamda/(4*pi*d0))+10*n*log10(d/d0); if nargin>4, PL=PL+sigma*randn(size(d)); end</pre>

Çizelge 4.3 Yol kayıp modellerinin simülasyonu

PL_GENEL.m
<pre>clear all, clf, clc fc=433e6; d0=1; sigma=3; distance=[1:2:15].^2; Gt=[1 1 4]; Gr=[1 0.5 4]; Exp=[2 2.5 3]; for k=1:3 y_logdist(k,:)=PL_logdist_or_norm(fc,distance,d0,Exp(k)); y_Free(k,:)= PL_free(fc,distance,Gt(k),Gr(k)); y_lognorm(k,:)=PL_logdist_or_norm(fc,distance,d0,Exp(1),sigma); end subplot(131) semilogx(distance,y_Free(1,:), 'k- o',distance,y_Free(2,:), 'b- ^',distance,y_Free(3,:), 'r-s</pre>

Çizelge 4.3.'ün devamı

```
grid on, axis([1 1000 40 110]), title(['Serbest Uzak  
Kayıp Modeli,f_c-',num2str(fc/1e6),'MHz'])  
xlabel('Mesafe[m'])  
ylabel('Yol Kaybı[dB]')  
legend('G_t=1, G_r=1','G_t=1, G_r=0.5','G_t=0.5,  
G_r=0.5')  
subplot(132)  
semilogx(distance,y_logdist(1,:), 'k-  
o',distance,y_logdist(2,:), 'b-  
^',distance,y_logdist(3,:), 'r-s')  
grid on, axis([1 1000 40 110]),  
title(['Log-distance Yol Kayıp Modeli, f_c-  
,num2str(fc/1e6),'MHz'])  
xlabel('Mesafe[m]'), ylabel('Yol Kaybı[dB]'),  
legend('n=2','n=3','n=6')  
subplot(133)  
semilogx(distance,y_lognorm(1,:), 'k-  
o',distance,y_lognorm(2,:), 'b-  
^',distance,y_lognorm(3,:), 'r-s')  
grid on, axis([1 1000 40 110]),  
title(['Log-normal Yol Kayıp Modeli, f_c-  
,num2str(fc/1e6),'MHz','\sigma-  
,num2str(sigma),'dB'])  
xlabel('Mesafe[m]'), ylabel('Yol Kaybı [dB]'),  
legend('Yol 1','Yol 2','Yol 3')
```

4.2.3. Yol Kayıplarının Simülasyon

Simülasyon için hazırlanan Çizelge 4.3.'deki ana program koşturulmadan önce Çizelge 4.1. ve Çizelge 4.2.'de sırasıyla serbest uzak kayıpları, log distance yayılım modeli ve log normal gölgeleme modeli için fonksiyonlar tanımlanmıştır.

PL_free matlab kodunda ilk “tmp” ifadesiyle alıcı ve verici antenin kazançlarının birim değerinde kabul ederek hesaplanmış ardından “if nargin” ifadesiyle fonksiyon giriş parametrelerinin ikiden fazla olduğu durumda alıcı ve verici antenin kazançlarını hesapa katan yeni bir “tmp” ifadesi hesaplanmıştır. Son olarakta PL ifadesiyle serbest uzak kayıpları hesaplanmıştır.

PL_logdist_or_norm matlab koduyla gölgeleme yol kaybının olduğu ve olmadığı durumlar için simülasyon yapılmıştır. Gölgeleme yol kaybı aslında gauss gürültüsünden başka bir şey değildir.

Matlab üzerinde sıfır ortalamalı bir standart sapmalı gauss gürültüsü `randn` komutuyla üretilir. Bunun σ ile çarpılması σ standart sapmalı gauss gürültüsünü ifade etmektedir. Fonksiyonun giriş parametreleri dört adet olduğunda gauss gürültüsü sıfır kabul edildiği log distance yayılım modeli için simülasyon yapılmıştır. Eğer fonksiyonun giriş parametreleri dörtten büyükse gölgeleme gauss gürültüsünün de işleme katıldığı durum log normal gölgeleme modeli için simülasyon koşturulmuştur. Ayrıca mesafe bilgisi vektör olarak tanımlanabileceğinden gölgeleme gauss gürültüsü de her farklı mesafe için üretilmesi adına `randn` komutu içerisine `size(d)` ifadesi eklenmiştir. Bu sayede her d için farklı gölgeleme gürültüsü üretilmiştir. “if” döngüsü ise fonksiyonun giriş parametresine göre log distance yol kaybı mı yoksa log normal gölgeleme yol kaybı mı olduğunu kontrol eder.

`PL_GENEL` matlab kodu simülasyonun ana kodudur. LoRa modülleri 433 MHz yayın yaptığından f_c taşıyıcı frekansı 433 MHz olarak ayarlanmıştır. Gölgeleme kaybını ifade etmek için $X(0,3)$ ifadesine uygun σ standart sapması 3 olarak tanımlanmıştır. Referans yol kaybı için d_0 mesafe bilgisi 1 m seçilmiştir. Alıcı-verici arasındaki d mesafesi ise vektör olarak tanımlanmıştır. G_T ve G_R ise sırasıyla alıcı ve verici antenler db cinsinden kazançlarıdır. Bu çalışmada kullanılan antenler 4 dB olduğundan anten kazançlarını ifade eden vektörde 4 dB ifadesi yazılmıştır. EXP ifadesi yol kaybını ifade etmektedir ve sırasıyla 2, 2.5 ve 3 olarak simüle edilmiştir. Biliyoruz ki yol kayıp üssü toplam yol kaybını logaritmik ifadeye bağlı doğrusal olarak artıracaktır. Bu değişimi logaritmik olarak görebilmek için `semilogx` komutuyla her bir yol kayıp modeli için mesafeye bağlı toplam yol kaybını değişimleri incelenmiştir.

4.3. PSO Algoritması Simülasyon Çalışmaları

4.3.1 PSO Algoritmasıyla Koordinat Tespiti Simülasyon Modeli

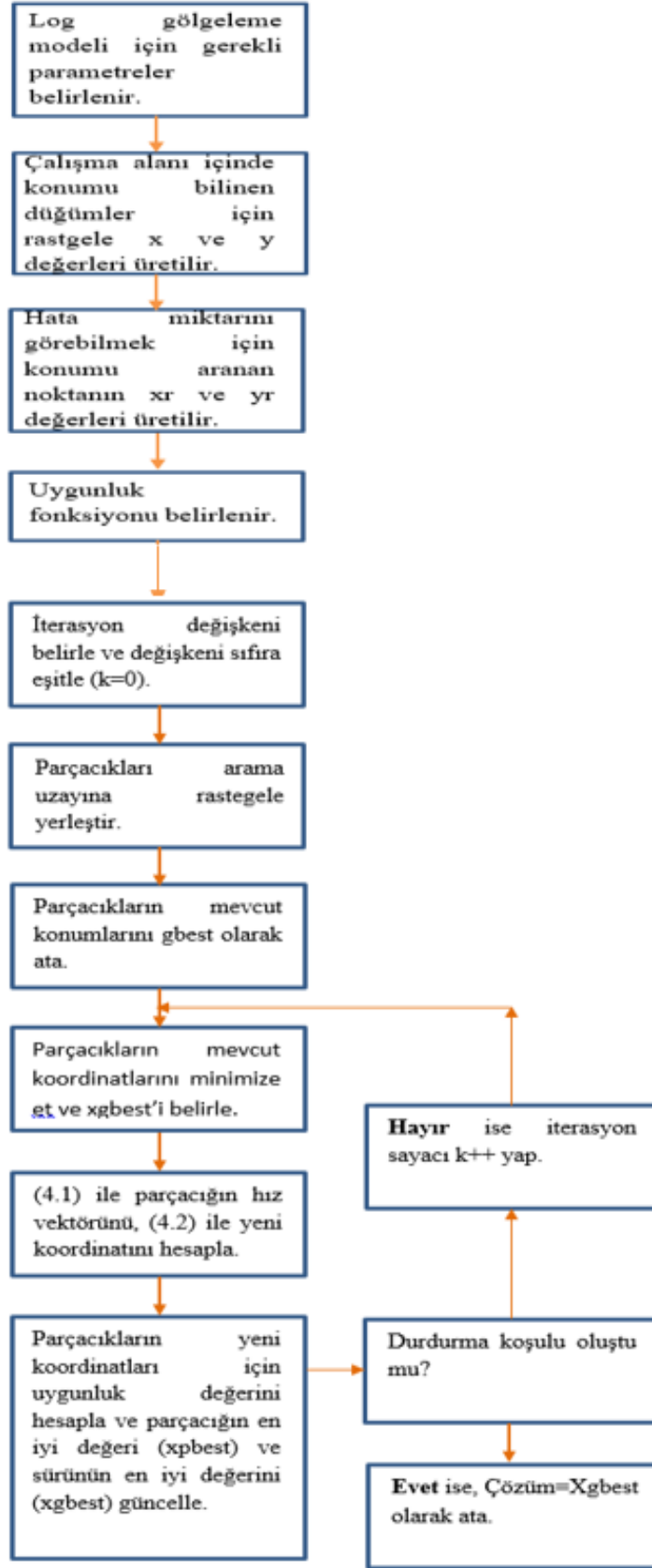
Simülasyon çalışmalarının bu kısmında iki boyutlu bir arama uzayında bulunan ve yol kayıplarının log normal gölgeleme modeliyle temsil edildiği üç adet verici ile bir adet konumu bilinmeyen alıcının koordinatlarını bulunması simüle edilmiştir. Koordinatların bulunması için PSO algoritmasından yararlanılmıştır. Bu yöntemde arama uzayında bulunan rastgele konumlandırılmış koordinata, durma koşulu sağlanıncaya kadar yaklaşarak en küçük hata ile koordinatı tespit edilmiştir. Bu amaç doğrultusunda ilk olarak konumu bilinmeyen cihazın rastgele X ve Y değerleri üretilmiştir. Ardından bir uygunluk fonksiyonu tanımlanmıştır. Bu fonksiyon iki boyutlu uzayda konumu bilinen noktalar ile konum tespiti yapılan nokta arasındaki mesafe olabilir. Sürüdeki her bir parçacık ve sürünün tamamı için uygunluk fonksiyonu sırasıyla (4.3) ve (4.4)'de ifade edilen eşitliklerle tanımlanabilir [14].

$$f_{parçacık}(X_P) = \sum_{j=2}^n (\widehat{x}_{1j} - x_{1jp})^2 \quad (4.3)$$

$$f_{sürü}(X_P) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n ((\widehat{x}_{ij} - x_{ijp}))^2 \quad (4.4)$$

Sırasıyla Denklem (4.3) ve (4.4)'te her bir iterasyon için parçacığın kendi en iyi değeri (xbest) ile sürünün en iyi değeri (xgbest) hesaplanarak bir sonraki iterasyon için (2.14) ifadesinden hız vektörü hesaplanır ve ardından (2.15) eşitliğinden yeni parçacığın konumu elde edilir. Bu işlem uygunluk fonksiyonunun en küçük değerine veya belirlenen iterasyon tamamlanıncaya kadar devam eder. Durma koşulu sağlandığı anda sürünün en iyi değeri (xgbest) aslında aradığımız noktaya en yakın yani uygunluk fonksiyonunun en küçük değerini aldığı koordinatı ifade eder.

PSO algoritmasının çalışabilmesi için bu yöntemde özgü bazı parametrelerin belirlenmesi gerekir. Denklem (2.14) eşitliğinde verilen bu parametrelerden w eylemsizlik katsayısı 0.7, parçacığın öğrenme katsayısı $\varepsilon_1=2.8$, sürünün öğrenme katsayısı $\varepsilon_2=2.05$, r_1 ve r_2 rastgele değişkenleri ise sırasıyla 0.2 ve 0.5 alınmıştır. İterasyon sayısı $k=100$ ve parçacık sayısı $n=50$ olarak belirlenmiştir. Şekil 4.4.'te bu şartlar altında algoritmanın akış diyagram modeli verilmiştir.



Şekil 4.1. PSO yöntemiyle konum tespit algoritması akış diyagramı

4.3.2 PSO Algoritmasıyla Koordinat Tespiti Simülasyon Yazılımı

Şekil 4.4.'te akış diyagramı paralel olarak bu kısmında öncelikle log gölgeleme modeli için gerekli parametreler (Pt, Pld0, fc, lamda, ortam kayıp üssü ve referans uzaklık) belirlenmiştir. Çalışma alanı içinde rastgele konumu bilinen üç noktanın xb ve yb değerleriyle konumu tespit edilecek noktanın xr ve yr değerleri oluşturulmuştur. Ardından PSO algoritması için gerekli parametreler (n=50 parçacık sayısı, k=100 iterasyon sayısı, parçacığın ve sürünün öğrenme katsayıları, w=0.7 eylemsizlik katsayısı ve r₁, r₂ normal dağılımlı rastgele değişkenler) belirlenmiştir. Bu şartlar altında Çizelge 4.4'teki MATLAB yazılımı geliştirilmiştir [18].

Çizelge 4.4. PSO algoritmasıyla konum tespiti için geliştirilen MATLAB yazılımı

```
function
[best,db,d,xr,yr]=psokonum(olcum_sayisi,sigma,xmax,ym
ax)
Pt=20;%verici gücü (dbm)
Pld0=48.6;
fc=433e6;
lamda=3e8/fc;
d0=1;
delta=2.5;%ortam parametresi
xb=150+10*rand(1,olcum_sayisi);%Konumu bilinen üç
noktanın x ve y değerleri oluşturuldu
yb=150+10*rand(1,olcum_sayisi);
xr=xmax*rand;%Çalışılan alanda konumu aranan
yr=ymax*rand;%cihazın x ve y değerleri
d=sqrt((xb-xr).^2+(yb-yr).^2);
gn=sigma*randn(1,olcum_sayisi)
rssi=(Pt-Pld0)-10*delta*log10(d/d0)+gn;
db=d0*10.^((Pt-Pld0-rssi)/(10*delta));;

%PSO algoritması geliştirilme kısmı
n=50;%parçacık sayısı
Num_iterations=100;%iterasyon sayısı

a1=0.2; b=0.5;%PSO parametreleri
wf=0.7;
c1=2.8; c2=2.05;
kordinat=2;
% -----
```

Çizelge 4.4.'ün devamı

```
best=zeros(Num_iterations,kordinat+1);    % Her bir
iterasyonda elde edilen en iyi x ve y değerleriyle
uygunluk fonksiyonun değerini oluşturulduğu matris

% n adet başlangıç parçacığının konumlarının
oluşturulması

xn=xmax*rand(n,kordinat);
xn(:,kordinat+1)=inf;

xo=xmax*rand(1,kordinat);
xo=[xo inf];

pb(1:n,1:kordinat)=xmax*rand(n,kordinat);
pb(:,kordinat+1)=inf;

pax(1:n,1:kordinat)=0;
ceza=1e10;
for l=1:Num_iterations
% 0 andaki en iyi konumun bulunması (xo,yo)

for j=1:n
dx=0;
    for jk=1:olcum_sayisi
        dx=dx+abs(db(jk)-sqrt(sum((xn(j,1:kordinat)-
[xb(jk) yb(jk)])).^2)));
    end
    if sum(xn(j,1:kordinat)<0)>0
        xn(j,kordinat+1)=dx+sum(xn(j,1:kordinat)<0)*ceza;
    end
    if sum(xn(j,1:kordinat)>xmax)>0

xn(j,kordinat+1)=dx+sum(xn(j,1:kordinat)>xmax)*ceza;
    end
    if sum(xn(j,1:kordinat)<0)==0 &&
sum(xn(j,1:kordinat)>xmax)==0
        xn(j,kordinat+1)=dx;
    end

if xn(j,kordinat+1)<pb(j,kordinat+1)

    pb(j,1:kordinat)=xn(j,1:kordinat);
    pb(j,kordinat+1)=xn(j,kordinat+1);

end
```

Çizelge 4.4.'ün devamı

```
if xn(j,kordinat+1)<xo(kordinat+1)
    xo(1:kordinat)=xn(j,1:kordinat);%sürünün en iyi
    değeri gbest
    xo(kordinat+1)= xn(j,kordinat+1);%sürünün en iyi
    değeri gbest
end

end

for j=1:n
    % xn(j,1:kordinat)=(1-
    b)*xn(j,1:kordinat)+b*xo(1:kordinat)+a1*(rand(1,kordi
    nat)-0.5);%APSO

    pax(j,1:kordinat)=(wf*pax(j,1:kordinat)+c1*rand(1,kor
    dinat).*(pb(j,1:kordinat)-
    xn(j,1:kordinat))+c2*rand(1,kordinat).*(xo(1:kordinat
    )-xn(j,1:kordinat)));
    %

    xn(j,1:kordinat)=(xn(j,1:kordinat)+pax(j,1:kordinat))
    ;

    %
    xn(j,xn(j,1:kordinat)<=0)=abs(xn(j,xn(j,1:kordinat)<=
    0));
    %
    % xn(j,xn(j,1:kordinat)>=xmax)=xmax;

end

% History
best(1,:)=xo;
% best(1,m+1)

end
```


4.3.3. PSO Algoritması Koordinat Tespiti Simülasyon

Simülasyon çalışmasının bu kısmında Çizelge 4.4'te geliştirilen MATLAB uygulaması koşturulmuştur. Ayrıca PSO algoritmasında iterasyon sayısı parçacıklar birbirine iyice yaklaşmaktadır. Çünkü PSO algoritmasında parçacıklar bir önceki hareketlerinden deneyim elde ederek bir sonraki hız vektörünü belirlemektedir. Bu durumda Çizelge 4.5'e bakılacak olursa iterasyon sayısı arttıkça parçacıklar hedef nokta etrafında yığılmaya başlıyor.

Çizelge 4.5. PSO algoritması sonucunda (xr,yr) koordinatlarına yakınsama değerleri

İterasyon Adımı (k)	Parçacığın xr,k değeri	Parçacığın yr,k değeri	Uygunluk Fonksiyon Değeri
1	31,36345791	187,6150914	32,03357515
5	30,15690291	147,5389639	30,65059512
10	32,58208308	182,1370747	29,71654553
15	32,01354704	177,4895008	29,57050484
20	32,01354704	177,4895008	29,57050484
25	32,01354704	177,4895008	29,57050484
30	32,01354704	177,4895008	29,57050484
35	32,01354704	177,4895008	29,57050484
40	32,01354704	177,4895008	29,57050484
45	32,01354704	177,4895008	29,57050484
50	32,01354704	177,4895008	29,57050484
55	32,01354704	177,4895008	29,57050484
60	32,01354704	177,4895008	29,57050484
65	32,01354704	177,4895008	29,57050484
70	32,01354704	177,4895008	29,57050484
75	32,01354704	177,4895008	29,57050484
80	32,01354704	177,4895008	29,57050484
85	32,01354704	177,4895008	29,57050484
90	34,11045157	186,3796652	29,27718067
95	34,11045157	186,3796652	29,27718067
100	34,70767183	188,5456563	29,18126686

Çizelge 4.5.'teki değerler çalışma alanı içindeki parçacıkların yakınsama değerlerini göstermektedir. PSO algoritmasının başarımını görebilmek adına program öncesinde (xr,yr) değerleri oluşturulmuştur. Çizelge 4.6'da noktanın gerçek değeri ile PSO algoritmasıyla bulunan sürünün en iyi değeri yani algoritmanın bulduğu x ve y değerleri birlikte verilmiştir.

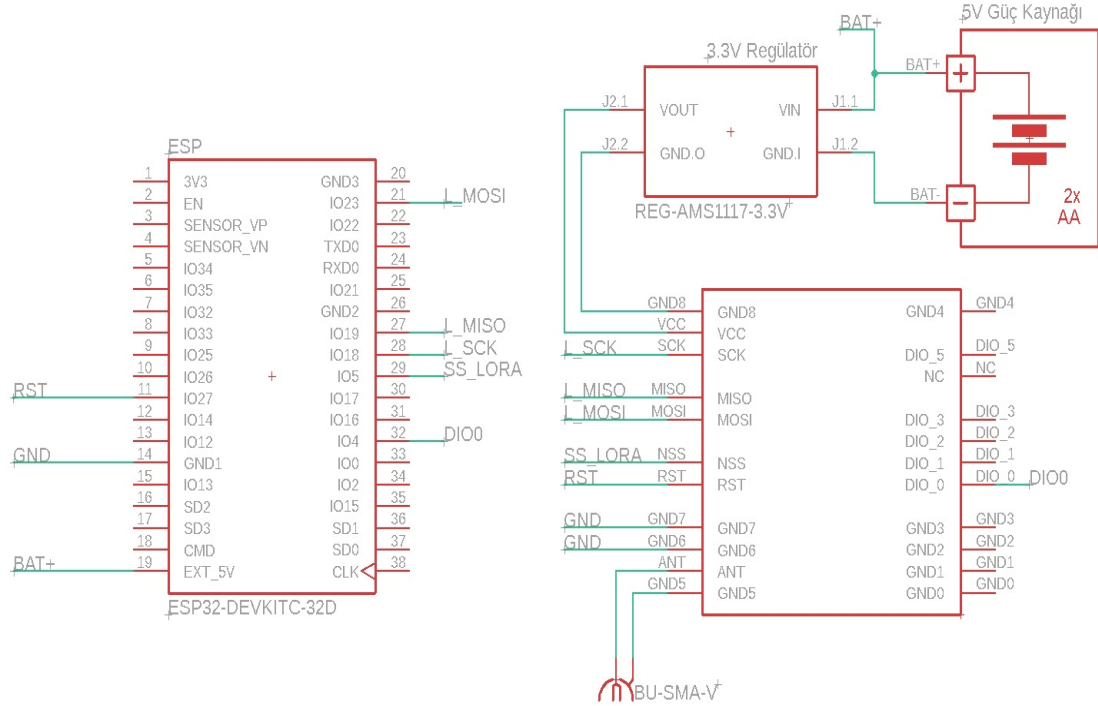
Çizelge 4.6. Algoritmanın bulduğu değerler ile gerçek değerlerin karşılaştırılması

xgbest	34.7077
Xr	13.7131
Ygbest	188.5457
Yr	189.9982

5. DENEYSEL ÇALIŞMALAR

5.1. Genel Bilgiler

Bu bölümde tasarım ve simülasyon çalışmaları sonucunda geliştirilmek istenen projenin elektronik entegrasyonu ve yazılımları yapılarak cihazın son kullanıcı formu oluşturulmuştur. Şekil 5.1.'de LoRa haberleşme modülleri ile ESP32 mikrodenetleyicinin pin bağlantı şeması verilmiştir.



Şekil 5.1. ESP32-DEVKITC-32D ile LoRa haberleşme modülünün pin diyagramı

Şekil 5.1.'deki bağlantı şeması tek bir düğümdeki bağlantı yapısını göstermektedir. Bu bağlantı yapısına sahip toplamda dört adet düğüm bulunmaktadır. Ayrıca sistemin 5 V'luk bir kaynak üzerinden beslenebilmesi için 3.3V regülatör tasarlanmıştır ve Şekil 5.1.'deki gibi sisteme entegresi yapılmıştır.

Şekil 5.1.'de gösterilen sistem açık ortamda doğa şartlarında kullanılacağından sistemi oluşturan elektronik bileşenleri bir arada kalması ve jumper kablolarından kaynaklı elektriksel kopuklukların önüne geçebilmek adına PCB hazırlanmış ve sistem üzerine entegre edilmiştir.

Şekil 5.1.'de tasarımı yapılan baskı devre kartı temin edildikten sonra diğer sistem bileşenlerinin kart üzerine lehimleri yapılarak Şekil 5.2.'te tek bir düğüme ait yapı elde edilmiştir.



Şekil 5.2 Konum tespitinde kullanılacak düğümlerden yalnızca bir tanesi

Konum tespitinin yapılabilmesi için Şekil 5.2.'te gösterilen yapıdan toplamda beş adet üretilmiştir. Bu aşamayla birlikte sistemin tüm elektronik entegrasyonları yapılmış oldu. Ancak LoRa modüllerinin birbiriyle haberleşebilmesi için modüllerin parametre ayarlarının (kanal, adres vs) yapılması gereklidir. Bunun için Çizelge 5.1.'de verilen kod yapısı hazırlanmıştır. Bu kod içerisinde RSSI değerinde mesafeyi ardından da trilaterasyon yöntemiyle hedef cihazın koordinatlarının X, Y ve Z bileşenleri hesaplanmıştır.

Çizelge 5.1. Alıcı-Verici LoRa modüllerinin haberleşmesi için gerekli program

<pre> LoRa Duplex communication #include <SPI.h> #include <LoRa.h> #define ss 5 #define rst 27 #define dio3 4 String hamveri = ""; byte msgCount = 0; byte localAddress = 0xBB; byte destination = 0xFF; long lastSendTime = 0; float MPwr1 = -37.0; float MPwr2 = -36.0; float MPwr3 = -37.0; float MPwr4 = -43.0; float N1 = 1.8; float N2 = 1.8; float N3 = 1.8; float N4 = 1.8; void loop() { String incoming = ""; onReceive(LoRa.parsePacket()); telemetriparcala(hamveri); </pre>	<pre> // Kullanılacak kütüphanelerin entegrasyonu // Select, Reset, DIO pinlerinin tanımlanması //Gelen verinin depolanacağı string //Yerel adres //Alıcı adres // Referans 1 metre için RSSI değeri // Ortam yol kayıp üssü //Gelen paketi “hamveri” stringi içine gömer. //Gelen paketi parçalayıp gerekli Node arrayının içine gömer </pre>
---	--

Çizelge 5.1.'in devamı

<pre> uzaklikolcum(); gpsX_Matrix(); gpsY_Matrix(); gpsZ_Matrix(); gpsX_det(); gpsY_det(); gpsZ_det(); //Node1_info(); //Node2_info(); //Node3_info(); //Node4_info(); Serial.println(hamveri); delay(50); } void onReceive(int packetSize) { if (packetSize == 0) return; int recipient = LoRa.read(); byte sender = LoRa.read(); byte incomingMsgId = LoRa.read(); byte incomingLength = LoRa.read(); String incoming = ""; while (LoRa.available()) { incoming += (char)LoRa.read(); } </pre>	<pre> //Gelen paket RSSI göre uzaklık ölçer. //Matrixleri oluşturur. //Determinantı hesaplar. //Test esnasında kontrol fonksiyonları //En son gelen veriyi yazdırır. //50 ms'lik gecikme // Sadece konumu tespit edilmek istenen cihazda bulunan ve gönderilen paketleri alan komut, eğer gelen paketin boyutu sıfır ise veri alma durumu geçer. // Alıcı adres okunur // Gönderen adres okunur // Gelen mesajın ID'si // Gelen mesajı uzunluğu </pre>
--	---

<pre> if (incomingLength != incoming.length()) {Serial.println("error: message length does not match length");return;} if (recipient != localAddress && recipient != 0xFF) {Serial.println("This message is not for me."); return;} /* Serial.println("Received from: 0x" + String(sender, HEX)); Serial.println("Sent to: 0x" + String(recipient, HEX)); Serial.println("Message ID: " + String(incomingMsgId)); Serial.println("Message length: " + String(incomingLength)); Serial.println("Message: " + incoming); Serial.println("RSSI: " +String(LoRa.packetRssi())); Serial.println("Snr: " + String(LoRa.packetSnr())); Serial.println(); */ </pre>	<pre> // if message is for this device, or broadcast, print details: </pre>
---	---

Çizelge 5.1.'in devamı

<pre> hamveri = incoming; Serial.println(hamveri); void telemetriparcala(String incoming){ int ID[1]; ID[0] = incoming.substring(0,1).toInt(); Serial.println(ID[0]); if (ID[0] == 1){ Node1[0] =LoRa.packetRssi(); Node1[1] =incoming.substring(1,5).toInt()-1000; Node1[2] =incoming.substring(5,9).toInt()-1000; Node1[3]=incoming.substring(9,13).toInt()-1000; Node1[4]=incoming.substring(13,20).toInt()-1000; } else if (ID[0] == 2){ Node2[0]=LoRa.packetRssi(); Node2[1]=incoming.substring(1,5).toInt()-2000; Node2[2]=incoming.substring(5,9).toInt()-2000; Node2[3]=incoming.substring(9,13).toInt()-2000; Node2[4]=incoming.substring(13,20).toInt()-2000; } else if (ID[0] == 3){ Node3[0]=LoRa.packetRssi(); Node3[1]=incoming.substring(1,5).toInt()-3000; Node3[2]=incoming.substring(5,9).toInt()-3000; Node3[3]=incoming.substring(9,13).toInt()-3000; Node3[4]=incoming.substring(13,20).toInt()-3000; } </pre>	<pre> // Node 1 için // RSSI Değeri // Enlem Bilgisi // Boylam Bilgisi // Yükseklik Bilgisi // Gelen Paketin Sayısı // Node 2 için // RSSI Değeri // Enlem Bilgisi // Boylam Bilgisi // Yükseklik Bilgisi // Gelen Paketin Sayısı // Node 3 için // RSSI Değeri // Enlem Bilgisi // Boylam Bilgisi // Yükseklik Bilgisi // Gelen Paketin Sayısı </pre>
---	--

Çizelge 5.1.'in devamı

<pre> else if (ID[0] == 4){ Node4[0]=LoRa.packetRssi(); Node4[1]=incoming.substring(1,5).toInt()-4000; Node4[2]=incoming.substring(5,9).toInt()-4000; Node4[3]=incoming.substring(9,13).toInt()-4000; Node4[4]=incoming.substring(13,20).toInt()- 4000; } else{ Serial.println("****Yanlış ID****"); } void uzaklikolcum(){ float ratio1 = (MPwr1 - float(Node1[0]))/(10 * N1); Node1[5] = (pow(10,ratio1)); float ratio2 = (MPwr2 - float(Node2[0]))/(10 * N2); Node2[5] = (pow(10,ratio2)); float ratio3 = (MPwr3 - float(Node3[0]))/(10 * N3); Node3[5] = (pow(10,ratio3)); float ratio4 = (MPwr4 - float(Node4[0]))/(10 * N4); Node4[5] = (pow(10,ratio4)); } void gpsX_Matrix(){ payX[0][0] = ((sq(Node1[5])-sq(Node2[5])) - (sq(Node1[1])-sq(Node2[1])) - (sq(Node1[2])- sq(Node2[2])) - (sq(Node1[3])-sq(Node2[3]))); </pre>	<pre> // Node 4 için // RSSI Değeri // Enlem Bilgisi // Boylam Bilgisi // Yükseklik Bilgisi // Gelen Paketin Sayısı // Node 1'den gelen paketin alınıp okunulmasıyla elde edilen RSSI değeri ve ortam yol kayıp üssü N1' den hedef cihaz ile düğüm arasındaki mesafe tespit edilir. Bu işlem sırasıyla Node2, Node3 ve Node4 arasında da yapılır. Ardından tanımlanmış 10 elemanlı dizilerin 5 indexli elemanına bu bilgi eşitleniyor. // Hedef cihazın koordinatlarında X'in hesaplanması için Denklem- 2.17'de gösterilen matrisel eşitlikler oluşturuluyor. </pre>
--	---

Çizelge 5.1.'in devamı

<pre> payX[0][1] = 2 * (Node2[2]-Node1[2]); payX[0][2] = 2 * (Node2[3]-Node1[3]); payX[1][0] = ((sq(Node1[5])-sq(Node3[5])) - (sq(Node1[1])-sq(Node3[1])) - (sq(Node1[2])- sq(Node3[2])) - (sq(Node1[3])-sq(Node3[3]))); payX[1][1] = 2 * (Node3[2]-Node1[2]); payX[1][2] = 2 * (Node3[3]-Node1[3]); payX[2][0] = ((sq(Node1[5])-sq(Node4[5])) - (sq(Node4[1])-sq(Node4[1])) - (sq(Node1[2])- sq(Node4[2])) - (sq(Node1[3])-sq(Node4[3]))); payX[2][1] = 2 * (Node4[2]-Node1[2]); payX[2][2] = 2 * (Node4[3]-Node1[3]); paydaX[0][0] = 2 * (Node2[1]-Node1[1]); paydaX[0][1] = payX[0][1]; paydaX[0][2] = payX[0][2]; paydaX[1][0] = 2 * (Node3[1]-Node1[1]); paydaX[1][1] = payX[1][1]; paydaX[1][2] = payX[1][2]; paydaX[2][0] = 2 * (Node3[1]-Node1[1]); paydaX[2][1] = payX[2][1]; paydaX[2][2] = payX[2][2]; } void gpsY_Matrix(){ payY[0][0] = 2 * (Node2[1]-Node1[1]); payY[0][1] = ((sq(Node1[5])-sq(Node2[5])) - (sq(Node1[1])-sq(Node2[1])) - (sq(Node1[2])- sq(Node2[2])) - (sq(Node1[3])-sq(Node2[3]))); payY[0][2] = 2 * (Node2[3]-Node1[3]); </pre>	<pre> // Hedef cihazın koordinatlarında Y'sinin hesaplanması için Denkle 2.18.'de gösterilen matrisel eşitlikler oluşturuluyor. </pre>
---	--

Çizelge 5.1.'in devamı

<pre> payY[1][0] = 2 * (Node3[1]-Node1[1]); payY[1][1] = ((sq(Node1[5])-sq(Node3[5])) - (sq(Node1[1])-sq(Node3[1])) - (sq(Node1[2])- sq(Node3[2])) - (sq(Node1[3])-sq(Node3[3])));; payY[1][2] = 2 * (Node3[3]-Node1[3]); payY[2][0] = 2 * (Node3[1]-Node1[1]); payY[2][1] = ((sq(Node1[5])-sq(Node4[5])) - (sq(Node4[1])-sq(Node4[1])) - (sq(Node1[2])- sq(Node4[2])) - (sq(Node1[3])-sq(Node4[3])));; payY[2][2] = 2 * (Node4[3]-Node1[3]); paydaY[0][0] = payY[0][0]; paydaY[0][1] = 2 * (Node2[2]-Node1[2]); paydaY[0][2] = payY[0][2]; paydaY[1][0] = payY[1][0]; paydaY[1][1] = 2 * (Node3[2]-Node1[2]); paydaY[1][2] = payY[1][2]; paydaY[2][0] = payY[2][0]; paydaY[2][1] = 2 * (Node4[2]-Node1[2]); paydaY[2][2] = payY[2][2]; } void gpsZ_Matrix(){ payZ[0][0] = 2 * (Node2[1]-Node1[1]); payZ[0][1] = 2 * (Node2[2]-Node1[2]); payZ[0][2] = ((sq(Node1[5])-sq(Node2[5])) - (sq(Node1[1])-sq(Node2[1])) - (sq(Node1[2])- sq(Node2[2])) - (sq(Node1[3])-sq(Node2[3])));; </pre>	<pre> // Hedef cihazın koordinatlarında Z'sinin hesaplanması için Denkle- 2.19'da gösterilen matrisel eşitlikler oluşturuluyor. </pre>
--	--

Çizelge 5.1.'in devamı

<pre> payZ[1][0] = 2 * (Node3[1]-Node3[1]); payZ[1][1] = 2 * (Node3[2]-Node3[2]); payZ[1][2] = ((sq(Node1[5])-sq(Node2[5])) - (sq(Node1[1])-sq(Node2[1])) - (sq(Node1[2]) -sq(Node2[2])) - (sq(Node1[3])-sq(Node2[3]))); payZ[2][0] = 2 * (Node4[1]-Node1[1]); payZ[2][1] = 2 * (Node4[2]-Node1[1]); payZ[2][2] = ((sq(Node1[5])-sq(Node2[5])) - (sq(Node1[1])-sq(Node2[1])) - (sq(Node1[2]) -sq(Node2[2])) - (sq(Node1[3])-sq(Node2[3]))); paydaZ[0][0] = payZ[0][0]; paydaZ[0][1] = payZ[0][1]; paydaZ[0][2] = 2 * (Node2[3]-Node1[3]); paydaZ[1][0] = payZ[1][0]; paydaZ[1][1] = payZ[1][1]; paydaZ[1][2] = 2 * (Node3[3]-Node1[3]); paydaZ[2][0] = payZ[2][0]; paydaZ[2][1] = payZ[2][1]; paydaZ[2][2] = 2 * (Node4[3]-Node3[3]); } void gpsX_det(){ gpsXpay=((payX[0][0]*payX[1][1]*payX[2][2]) +(payX[0][1]*payX[1][2]*payX[2][0])+ (payX[0][2]*payX[1][0]*payX[2][1])); </pre>	<pre> // Cramer kuralına göre oluşturulan gpsX matrisel ifadenin determinantı alınır. </pre>
--	--

Çizelge 5.1.'in devamı

<pre> void gpsX_det(){ gpsXpay=((payX[0][0]*payX[1][1]*payX[2][2])+(payX[0][1]*payX[1][2]*payX[2][0])+(payX[0][2]*payX[1][0]*payX[2][1])); gpsXpay=gpsXpay- ((payX[2][0]*payX[1][1]*payX[0][2])+(payX[2][1]*payX[1][2]*payX[0][0])+(payX[2][2]*payX[1][0]*payX[0][1])); gpsXpayda= ((paydaX[0][0]*paydaX[1][1]*paydaX[2][2])+(paydaX[0][1]*paydaX[1][2]*paydaX[2][0])+(paydaX[0][2]*paydaX[1][0]*paydaX[2][1])); gpsXpayda=gpsXpayda- ((paydaX[2][0]*paydaX[1][1]*paydaX[0][2])+(paydaX[2][1]*paydaX[1][2]*paydaX[0][0])+(paydaX[2][2]*paydaX[1][0]*paydaX[0][1])); if(gpsXpayda == 0){ gpsXpayda = 1; } gpsX = gpsXpay / gpsXpayda; Serial.println("*****"); Serial.println(gpsXpay); Serial.println(gpsXpayda); Serial.println(gpsX); return; } </pre>	<p>// Matrisel eşitlikten bulunan ifade hedef cihazın koordinatlarından X bileşenini verir ve ekrana yazdırılır.</p>
---	--

Çizelge 5.1.'in devamı

<pre> void gpsY_det(){ gpsYpay=((payY[0][0]*payY[1][1]*payY[2][2])+(payY[0][1]*payY[1][2]*payY[2][0])+(payY[0][2]*payY[1][0]*payY[2][1])); gpsYpay=gpsYpay- ((payY[2][0]*payY[1][1]*payY[0][2])+(payY[2][1]*payY[1][2]*payY[0][0])+(payY[2][2]*payY[1][0]*payY[0][1])); gpsYpayda=((paydaY[0][0]*paydaY[1][1]*paydaY[2][2])+(paydaY[0][1]*paydaY[1][2]*paydaY[2][0])+(paydaY[0][2]*paydaY[1][0]*paydaY[2][1])); gpsYpayda=gpsYpayda- ((paydaY[2][0]*paydaY[1][1]*paydaY[0][2])+(paydaY[2][1]*paydaY[1][2]*paydaY[0][0])+(paydaY[2][2]*paydaY[1][0]*paydaY[0][1])); if(gpsYpayda == 0){ gpsYpayda = 1; } gpsY = gpsYpay / gpsYpayda; Serial.println(gpsY); return; } void gpsZ_det(){ gpsZpay=((payZ[0][0]*payZ[1][1]*payZ[2][2])+(payZ[0][1]*payZ[1][2]*payZ[2][0])+(payZ[0][2]*payZ[1][0]*payZ[2][1])); gpsZpay=gpsZpay- ((payZ[2][0]*payZ[1][1]*payZ[0][2])+(payZ[2][1]*payZ[1][2]*payZ[0][0])+(payZ[2][2]*payZ[1][0]*payZ[0][1])); </pre>	<pre> // Cramer kuralına göre oluşturulan gpsY matrisel ifadenin determinantı alınır. // Matrisel eşitlikten bulunan ifade hedef cihazın koordinatlarından Y bileşenini verir ve ekrana yazdırılır. // Cramer kuralına göre oluşturulan gpsZ matrisel ifadenin determinantı alınır. </pre>
---	--

Çizelge 5.1.' in devamı

<pre> gpsZpayda=((paydaZ[0][0]*paydaZ[1][1]*paydaZ[2][2]) +(paydaZ[0][1]*paydaZ[1][2]*paydaZ[2][0])+ (paydaZ[0][2]*paydaZ[1][0]*paydaZ[2][1])); gpsZpayda=gpsZpayda- ((paydaZ[2][0]*paydaZ[1][1]*paydaZ[0][2]) +(paydaZ[2][1]*paydaZ[1][2]*paydaZ[0][0]) +(paydaZ[2][2]*paydaZ[1][0]*paydaZ[0][1])); if(gpsZpayda == 0){ gpsZpayda = 1; } gpsZ = gpsZpay / gpsZpayda; Serial.println(gpsZ); return; } void Node1_info(){ Serial.println("Node1 için RSSI degeri " + String(Node1[0],DEC)); Serial.println("Node1 uzaklığı " + String(Node1[5],DEC) + "m"); Serial.println("Node1 konumu (x,y,z)" + String(Node1[1],DEC) + "," + String(Node1[2],DEC) + "," + String(Node1[3],DEC)); Serial.println(); delay(10); } </pre>	<p>// Matrisel eşitlikten bulunan ifade hedef cihazın koordinatlarından Z bileşenini verir ve ekrana yazdırılır</p> <p>// Sistemin çalıştırılıp testinin yapılması durumunda Node1'e ait RSSI, uzaklık ve konum bilgileri ekrana yazdırılır.</p>
---	--

Çizelge 5.1.'in devamı

<pre> void Node2_info(){ Serial.println("Node2 için RSSI degeri " + String(Node2[0],DEC)); Serial.println("Node2 uzaklığı " + String(Node2[5],DEC) + "m"); Serial.println("Node2 konumu (x,y,z)" + String(Node2[1],DEC) + "," + String(Node2[2],DEC) + "," + String(Node2[3],DEC)); Serial.println(); delay(10); } void Node3_info(){ Serial.println("Node3 için RSSI degeri " + String(Node3[0],DEC)); Serial.println("Node3 uzaklığı " + String(Node3[5],DEC) + "m"); Serial.println("Node3 konumu (x,y,z)" + String(Node3[1],DEC) + "," + String(Node3[2],DEC) + "," + String(Node3[3],DEC)); Serial.println(); delay(10); } </pre>	<p>// Sistemin çalıştırılıp testinin yapılması durumunda Node2'ye ait RSSI, uzaklık ve konum bilgileri ekrana yazdırılır.</p> <p>// Sistemin çalıştırılıp testinin yapılması durumunda Node3'e ait RSSI, uzaklık ve konum bilgileri ekrana yazdırılır.</p>
---	--

Çizelge 5.1.'in devamı

<pre> void Node4_info(){ Serial.println("Node4 için RSSI degeri " + String(Node4[0],DEC)); Serial.println("Node4 uzaklığı " + String(Node4[5],DEC) + "m"); Serial.println("Node4 konumu (x,y,z)" + String(Node4[1],DEC) + "," + String(Node4[2],DEC) + "," + String(Node4[3],DEC)); Serial.println(); delay(10); } </pre>	<p>// Sistemin çalıştırılıp testinin yapılması durumunda Node4'e ait RSSI, uzaklık ve konum bilgileri ekrana yazdırılır.</p>
---	--

Çizelge 5.1.'de verilen yazılımla LoRa modüllerinin istenen koşullar altında birbiriyle sürekli olarak veri alış-verişi sağlanmıştır. Programın arka plandaki nasıl çalıştığını bilmek sürecin nasıl gerçekleştiğini anlamaya yardımcı olacaktır.

Öncelikle `#include <SPI.h>` ve `#include <LoRa.h>` fonksiyonlarıyla gerekli kütüphaneler eklenir. “ss, rst ve dio3” pinleri tanımlanır ve esp32 üzerinden hangi pinlere karşılık geldiği belirlenir. Sonrasında gönderilecek mesajın başlangıç msgCount, gönderilen adresi (localAddress=0xBB) ve hedef adresi (destination=0xFF) ifadeleri belirlenir. En son gönderilen mesajın t=0 anında gittiği düşünülür bunu da lastSendTime=0 şeklinde bir değişkenle tanımlanır. Her iki gönderim arasındaki zaman farkını belirlemek için bir başka değişken (interval=2000) ileride zaman sayacında kullanılmak üzere değer ataması yapılır. Bu sayede iki mesaj arasındaki zaman farkı ayarlanmış olacaktır. Ayrıca mesafenin hesaplanabilmesi için 1 metredeki RSSI

değerleri de test yapılarak ölçülmüş ve her bir node için MPwr1, MPwr2, MPwr3 ve MPwr4 değişkenlerine atanmıştır. İlave olarak mesafenin bulunması için bilinmesi gereken bir değişkende yol kayıp üssüdür. Yol kayıp üssü de test yapılan ortam için deneysel çalışmalar sonucunda $N=1.8$ olarak bulunmuştur.

Sabit konumunu bildiğimiz noktalarda bulunan LoRa modüllerinden gelen paketlerin açılıp okunduktan sonra pakete gömülmüş verilerin tutulacağı Node1, Node2, Node3 ve Node4 adlı diziler oluşturulmuştur. Ardından üçgenleme metoduyla hedef cihazın koordinatlarının X,Y ve Z bileşenleri için hesaplanması gereken matrisel işleme ait 3x3 boyutlarında matrisler oluşturulmuştur. Bu işlemler sonucunda hedef cihazın koordinatlarının tutulacağı gpsX, gpsY ve gpsZ değişkenleri oluşturulmuştur.

void setup() komutuyla birlikte program bir döngüye girer. Döngü içerisinde ilk olarak Serial.begin(115200) komutu ile serial portu açılır ve baud hızı 115200 baud/s olarak ayarlanır. Sonrasında portun doğru bir şekilde açılıp açılmadığını kontrol etmek için while (!Serial) döngüsüyle kontrol seti hazırlanır. Eğer Serial portu istenilen şekilde açılmışsa ekrana ‘LoRa Duplex’ yazdırır. Devamında kütüphanede tanımlı LoRa.setPins(ss, rst, dio3); komutuyla default olarak belirlenmiş CS, reset, and IRQ pinleri LoRa.setPins(ss, rst, dio3) komutuyla LoRa kütüphanesi üzerinden istenilen değerlerde ayarlanır. Sonrasında if (!LoRa.begin(433E6)) yapısıyla LoRa modülünün 433 MHz haberleşme frekansında başlatılıp başlatılmadığı kontrol edilir. Eğer LoRa başarılı bir şekilde 433 MHz’de çalıştırılırsa ekrana “LoRa init succeeded.” yazdırır yok eğer çalıştırılmazsa bu sefer de ekrana “LoRa init failed. Check your connections.” gibi bir ifadeyle geri dönüş verir.

void loop() ifadesiyle program döngünün içerisine girer. Döngüde gönderilen mesajın alıcı LoRa da alınabilmesi için kütüphanede bulunan onReceive(LoRa.parsePacket()) komutu kullanılır. Bu komutla gönderilen mesaj direkt olarak alınır. Sonrasında alınan paket telemetriparcala(hamveri) komutuyla paket içinde hangi enlem, boylam, yükseklikten gönderildiği ve kaçınıcı paket olduğu okunur. Devamında konumun bulunması için kullanılacak olan uzaklikolcum(), gpsX_Matrix(), gpsY_Matrix(), gpsZ_Matrix(), gpsX_det(), gpsY_det() ve gpsZ_det() fonksiyonları oluşturulur. Ayrıca sadece test aşamasında kullanılan ve dört farklı düğüme ait bilgi veren Node(ID)_info() fonksiyonu da oluşturulur.

`void send(Message(String outgoing)` ifadesiye başka bir döngü oluşturulur. `LoRa.beginPacket()` komutuyla gönderilecek olan paketin içeriği doldurulur. Bu işlemle alınan mesajın hangi adrese gönderilmek istendiği, hangi adresten gönderildiği, mesaj ID'si, yolladığımız mesajın uzunluğu yazdırılır ve sonrasında paketi bitirilip gönderilir. Her gönderme işleminden sonra da kaçınıcı paket olduğu bilgisini tutan `msgCount++` komutuyla bir arttırılır ve döngü tamamlanır. Aslında bu kısımla biz gönderilecek mesaja bir kimlik oluşturuz. Bu mesajın alınması aşamasında önemlidir. Çünkü ortamda 433 MHz yayın yapan başka cihazlarda olabilir. Alıcı ancak paket içinde yazılı bu bilgileri taşıyan paketleri tanır ve alır.

`void onReceive(int packetSize)` döngüsü sadece konumu tespit edilmek istenen sürekli alıcı durumunda olan cihazda bulunan ve gönderilen paketleri alan bir komuttur. `if (packetSize == 0) return;` döngüsüyle alınan paketin büyüklüğü kontrol edilir. Eğer paket büyüklüğü sıfıra eşitse bir mesaj alınmamıştır demektir. Bu durumda döngü şartı sağlanamadığından sürekli olarak `return` yapar. Varsayalım ki paket boyu sıfırdan farklı yani bilgi taşıyan bir mesaj gelmiş olsun. Bu aşamada gelen mesajın tanınıp doğruluğunun teyit edilebilmesi için öncelikle mesajın paket header'ları okunur. `LoRa.read();` komutuyla alınan hedef adres okunur `recipient'` e eşitlenir ve gönderici adres okunur `sender` ifadesine eşitlenir ve sonrasında `if (incomingLength != incoming.length())` komutuyla donanımsal olarak gelen mesaj ile gerçekte okunan birbirine eşit olup olmadığı kontrol edilir. Eğer bir uyumsuzluk varsa `Serial.println("error: message length does not match length");` yapısıyla mesaj boyutuyla ilgili bir hata olduğu ekrana yazdırılır.

`if (recipient != localAddress && recipient != 0xFF)` yapısıyla da gelen mesajın hedef adresiyle alınan LoRa'nın hedef adresi birbirini tutup tutmadığı kontrol edilir. Böylece bizim sistemimizin dışında etrafta bulunana farklı LoRa modüllerinden gelen mesajlar alınmamış olacaktır. Eğer bu aşamda bir hata varsa yani istenmeyen bir cihazdan mesaj gelmişse ekrana `Serial.println("This message is not for me.");` komutuyla bu mesaj benim değildir diye bir çıktı verecektir.

Doğru mesaj alınmış, mesaja ilişkin bilgiler okunmuş şimdi de okunan bilgilerin ekrana yazdırılması için `Serial.println()` fonksiyonu kullanılır. Bu kısımda alınan paketin hangi adresten hangi adrese gönderildiği, paket ID'si ve `Serial.println("RSSI: " + String(LoRa.packetRssi()));` komutuyla da alınan sinyalin RSSI değeri kütüphane

üzerinden elde edilip çıktı olarak ekrana bastırılır. Bu komut sayesinde alınan sinyalin RSSI değeri kütüphane üzerinden elde edilip çıktı olarak ekrana bastırılır. Sonrasında SNR değeri de yine kütüphanede tanımlı fonksiyonlarla çekilip ekrana yazdırılır.

Alınan paketin ID'sine göre paket içindeki bilgilerin programın girişinde tanımlanmış Node(ID)[9] dizisine kaydedilmesi gerekiyor. Bunun için öncelikle alınan paket void telemetriparcala(String incoming) komutuyla parçalanır ve program paketin ID'sine göre if döngüsü içindeki uygun kod bloğuna geçer. Bu kod bloğu içinde gelen paketin RSSI değeri Node(ID)[0] elemanında tutulurken, paketin geldiği cihazın enlem bilgisi Node(ID)[1] elemanında, boylam bilgisi Node(ID)[2] elemanında, yükseklik bilgisi Node(ID)[3] elemanında ve paket sayısına ait bilgi de Node(ID)[4] elemanında tutulur. Eğer paketin ID numarası 1,2,3 veya 4'ten biri değilse, bu mesaj bize ait olmayan bir mesaj olup program Serial.println("****Yanlış ID****") şeklinde bir ifade ekrana yazdırılır.

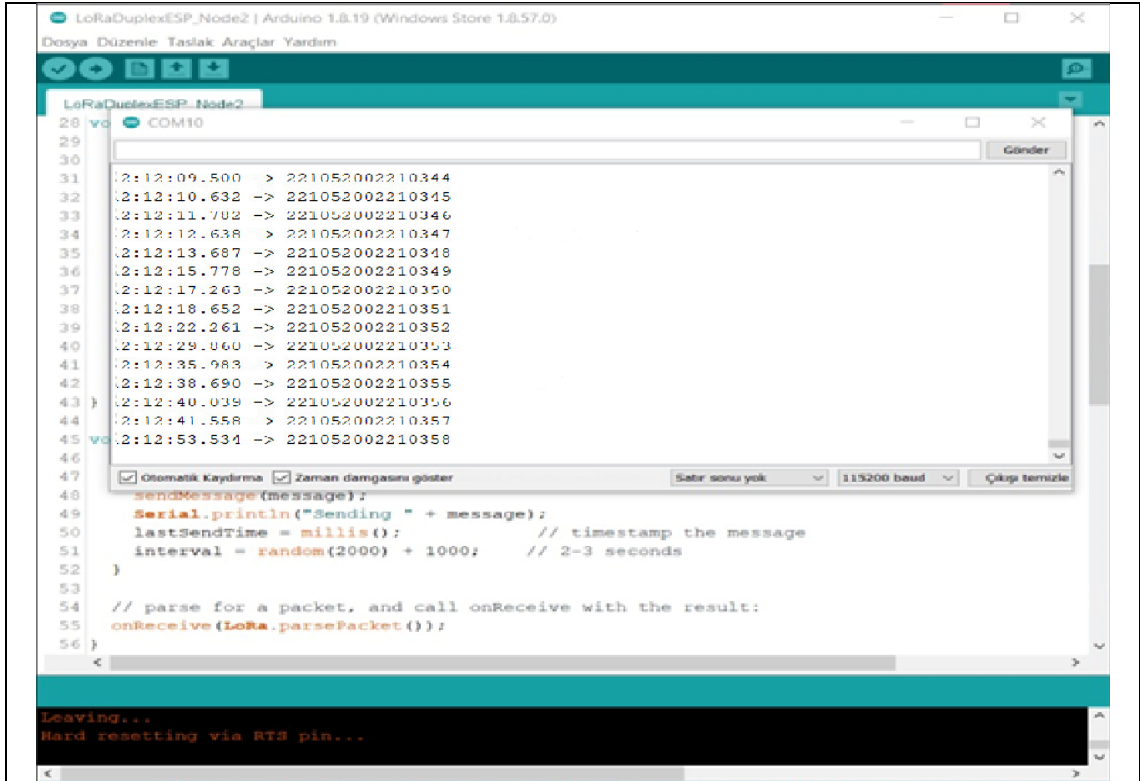
Düğümlerden gelen paketin alınıp okunmasıyla elde edilen RSSI ve ortam yol kaybı üssünden (N) hedef cihaz ile düğüm arasındaki mesafe Denklem (5.1)'deki eşitlikle hesaplanır. Denklemdaki MPwr ve N sabitleri kodun giriş kısmında her bir sabit düğüm ve hedef cihaz arasında yapılan testlerle belirlenmiştir. Bu işlem sırasıyla Node1, Node2, Node3 ve Node4 için yapılır ve ardından elde edilen değerler hangi paketle çalışılmışsa ona ait dizinin Node(ID)[5] elemanında tutulur.

$$ratio = (MPwr - RSSI)/(10^N)$$

$$Distance = 10^{ratio} = pow(10, ratio) \quad (5.1)$$

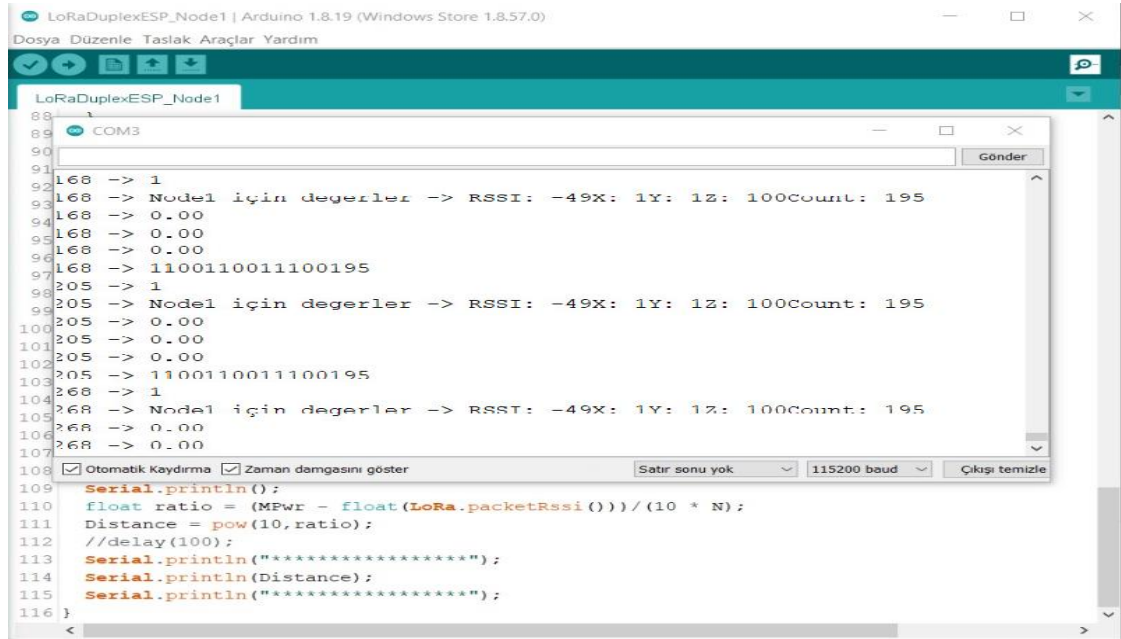
Son olarak gpsX_Matrix(), gpsY_Matrix() ve gpsZ_Matrix() fonksiyonlarıyla Denklem (X)'te gösterilen ifadeler yazılımsal olarak oluşturulur ve gpsX_det(), gpsY_det() ve gpsZ_det() fonksiyonlarıyla bu matrisel eşitliklerin determinantı alınarak hedef cihazın koordinatlarının X,Y ve Z bileşenleri elde edilir ve ekrana yazdırılır. Böylece program ilk olarak sabit düğümlerin RSSI bilgisinden hedef cihaz ile arasındaki mesafeyi ardından bu iki bilgiyi kullanarak cihazın koordinatlarına diğer dört cihaza bağlı olarak bulmuştur.

Çizelge 5.1.'de verilen program sadece bir alıcı-verici üzerinden koşturulursa çıktı olarak Şekil 5.1. ve Şekil 5.2.' de gösterilen ekran görüntüsü alınmıştır.



```
28 void setup() {
29   Serial.begin(115200);
30 }
31
32 // Send a message
33 void sendMessage(String message) {
34   Serial.println("Sending " + message);
35   lastSendTime = millis(); // timestamp the message
36   interval = random(2000) + 1000; // 2-3 seconds
37 }
38
39 // parse for a packet, and call onReceive with the result:
40 void onReceive(LoRa.packet_t packet) {
41   // ...
42 }
43
44 // ...
45 void loop() {
46   // ...
47   sendMessage("Leaving...");
48   // ...
49   Serial.println("Hard resetting via RTS pin...");
50 }
51
52 // ...
53
54 // ...
55
56 // ...
```

Şekil 5.3. Gönderici LoRa modülüne ilişkin ekran görüntüsü



```
88 void setup() {
89   Serial.begin(115200);
90 }
91
92 // Send a message
93 void sendMessage(String message) {
94   Serial.println("Sending " + message);
95   lastSendTime = millis(); // timestamp the message
96   interval = random(2000) + 1000; // 2-3 seconds
97 }
98
99 // parse for a packet, and call onReceive with the result:
100 void onReceive(LoRa.packet_t packet) {
101   // ...
102 }
103
104 // ...
105 void loop() {
106   // ...
107   sendMessage("Leaving...");
108   // ...
109   Serial.println("Hard resetting via RTS pin...");
110 }
111
112 // ...
113
114 // ...
115
116 // ...
```

Şekil 5.4. Alıcı LoRa modülüne ilişkin ekran görüntüsü

5.2. Yapılan Testler

Bu bölümde donanımsal ve yazılımsal olarak tamamlanan RSSI tabanlı konum tespit cihazının amacın uygun çalışıp çalışmadığını yorumlayabilmek için testler yapılmıştır. Cihazın kullanım alanı daha çok tarıma yönelik araziler olduğundan test ortamı olarak da daha çok düz, cihazlar arasında görüş açısının olduğu bir ortam seçilmiştir. Şekil 5.5 ve Şekil 5.6’ te testlerin yapıldığı ortamlar gösterilmiştir.



Şekil 5.5. İlk test çalışmasının yapıldığı ortam

Şekil 5.5 ve Şekil 5.6’te gösterilen test ortamlarında cihazın başarımını ölçebilmek için iki farklı test yapılmıştır. Bunlar:

1. Tek bir verici ile hedef cihaz arasında yapılan test,
2. Dört vericinin ve bir alıcının bulunması durumun hedef cihazın konumunun belirlenmesine yönelik testlerdir.



Şekil 5.6. İkinci test çalışmasının yapıldığı ortam

Bu aşamada her bir test düzeneğindeki alıcı ve vericiler arasındaki mesafe sırasıyla 1 metre, 5 metre, 10 metre ve 20 metre olarak belirlenmiştir. Ancak cihazın kapsama alanı kilometreler seviyesine rahatlıkla ulaşabilmektedir. Normal durumlarda kapsama alanı 3-5 km seviyesindeyken, gönderilen paketin iletim hızı düşürüldükçe 10 km kadar ulaşabilen kapsama alanı elde edilebilir.

İlk test çalışması sadece bir vericinin ve alıcının olduğu durum için yapılmıştır. Bu testin amacı tek bir vericiden gönderilen paketin hedef cihazın adresine doğru bir şekilde ulaşp ulaşmadığını görebilmek, eğer paket başarıyla hedefe ulaşmış ise alınan paketin doğru şekilde ayrıştırılıp RSSI değerinin elde edilmesi ve bu değer test yapılan ortama bağlı mesafeyle değişimi inceleyebilmektir. Ayrıca bu testle iki cihaz arasında belirlenen 1 metre referans uzaklığındaki MPwr değeri de belirlenecektir. Daha sonrasında bu değere bağlı olarak ortam yol kaybı üssü hesaplanacaktır. İkinci test çalışması ise doğrudan hedef cihazın konumunun tespitine yöneliktir. Bu çalışmayla ilk testten elde edilen değerler doğrultusunda ardışık alıcı ve vericiler arasındaki mesafenin 5 metre, 10 metre, 20 metre ve 50 metre değerlerinde olduğu karesel bir ortam belirlenmiş ve hedef cihaz bu alanın tam orta noktasına konumlandırılmıştır. Böylece düğümlere olan gerçek uzaklık belindiğinden cihazdan elde edilen mesafe ve konum bilgisinin doğruluğuda tespit edilebilecektir.

6. SONUÇLAR

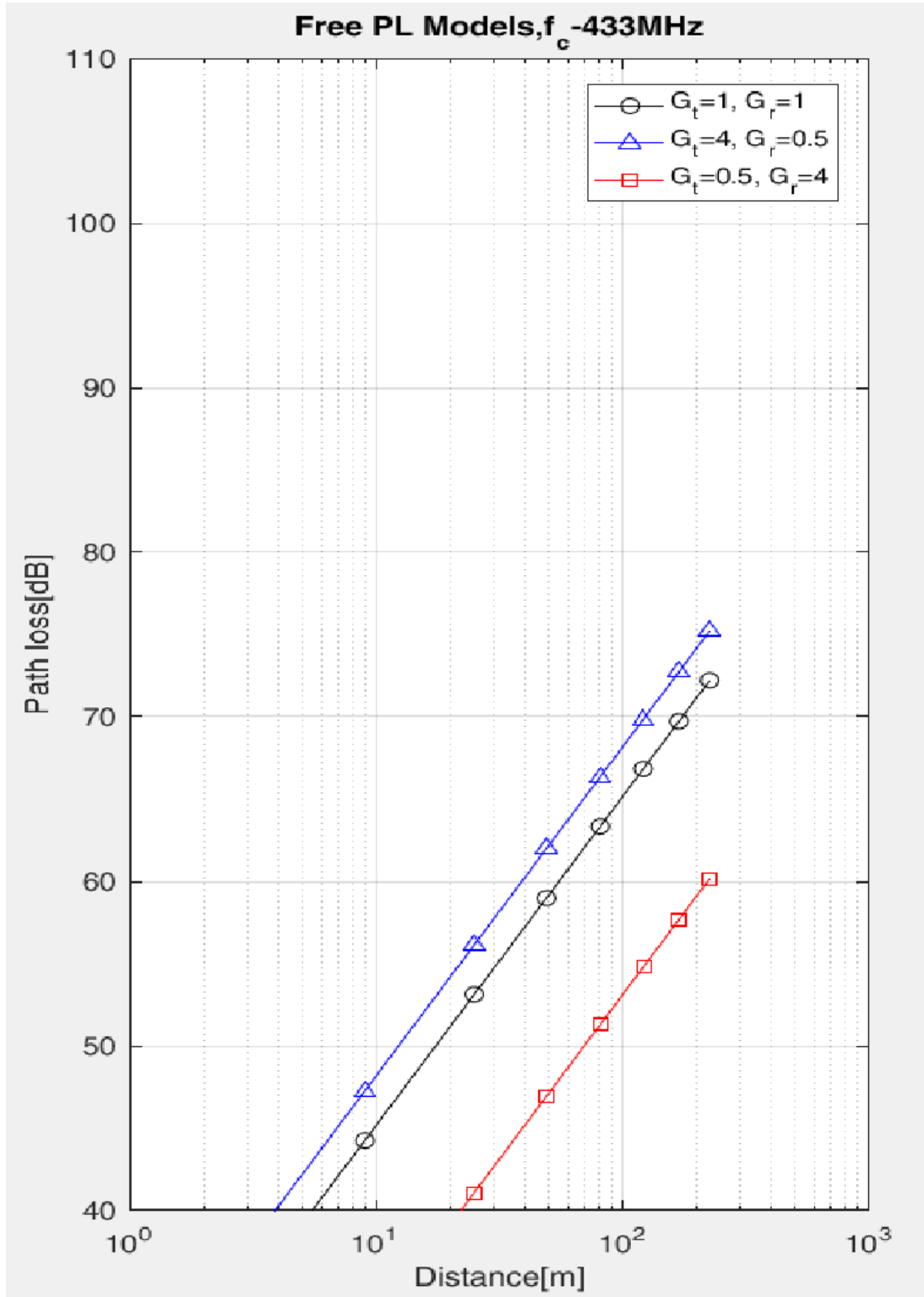
6.1. Genel Açıklamalar

Bu bölümde test ve simülasyon aşamasında yapılan teorik ve deneysel çalışmaların sonuçları verilerek, projenin ne düzeyde başarılı olduğu ele alınacaktır. Bu amaçla öncelikle olarak alıcı ve verici modüllerin bulunduğu çevresel ortamı tanımlayan parametreler belirlenerek sistemin simülasyon çalışmaları verilecektir. Akabinde yazılımsal ve donanımsal entegrasyon sonucunda konum tespit cihazının test sonuçları verilecektir. Her iki çalışma sonucunda sistemin ne ölçüde başarılı olduğu ve hata payının ne kadar olduğu da ölçümler sonucunda belirlenebilecektir.

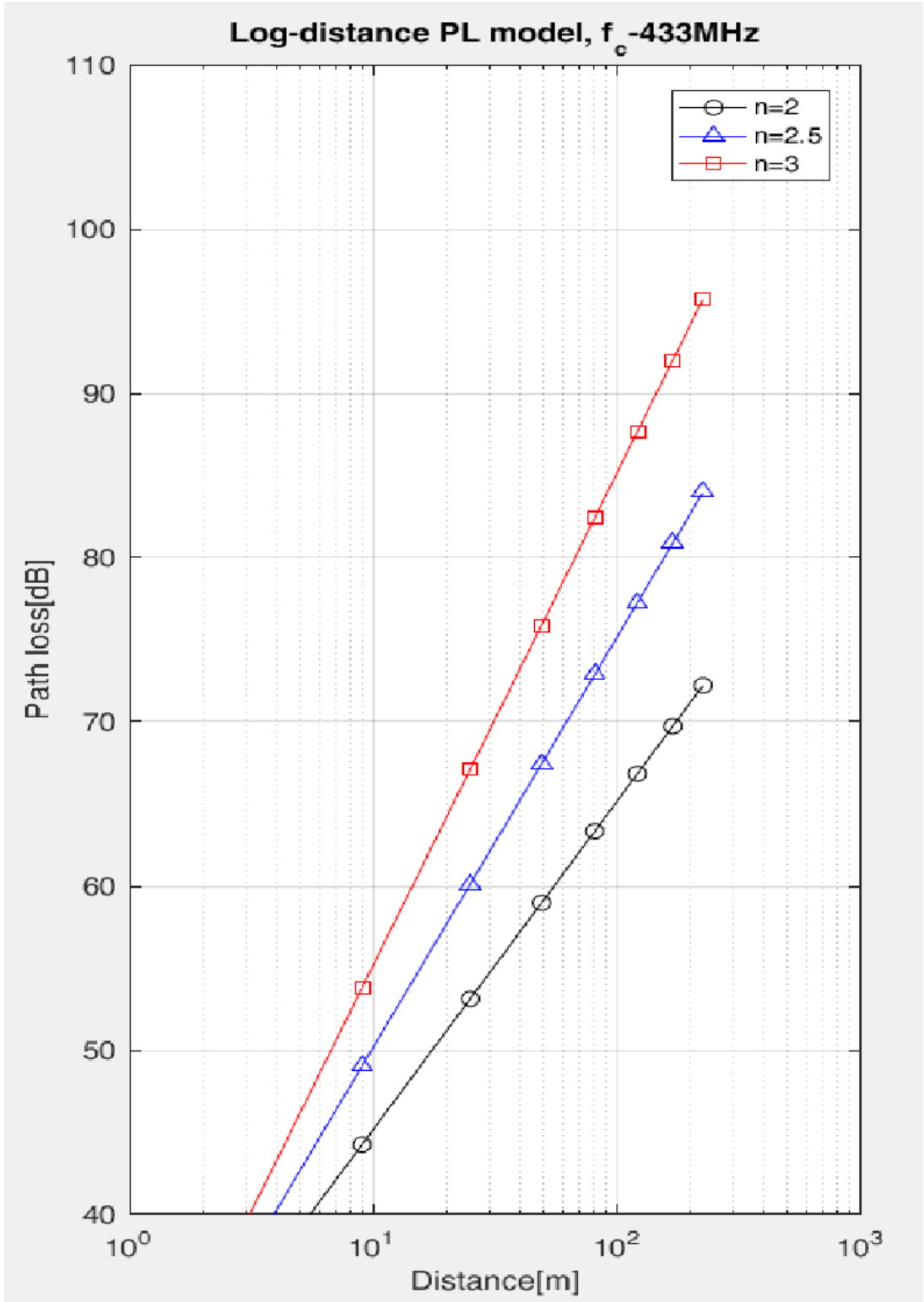
6.2. Simülasyon Sonuçları

Gerçekleştirilen simülasyon çalışmaları sonucunda çevresel şartların değişimine bağlı olarak alınan sinyalin gücünde de bir azalma görülmüştür. Simülasyon çalışmalarında kullanılan log normal gölgeleme modelinde alıcı ve verici arasındaki mesafe arttıkça alınan sinyalde zayıflamıştır. Ayrıyeten ortam kayıp üssündeki artışa bağlı olarak da sinyal gücü zayıflamıştır. Benzer şekilde simülasyon sonuçlarının gerçek hayattaki değerleri yansıtabilmesi için bir de gauss gürültüsü eklenmiştir. Gürültünün gücüne bağlı olarak yine RSSI değeri değişiklik göstermiştir.

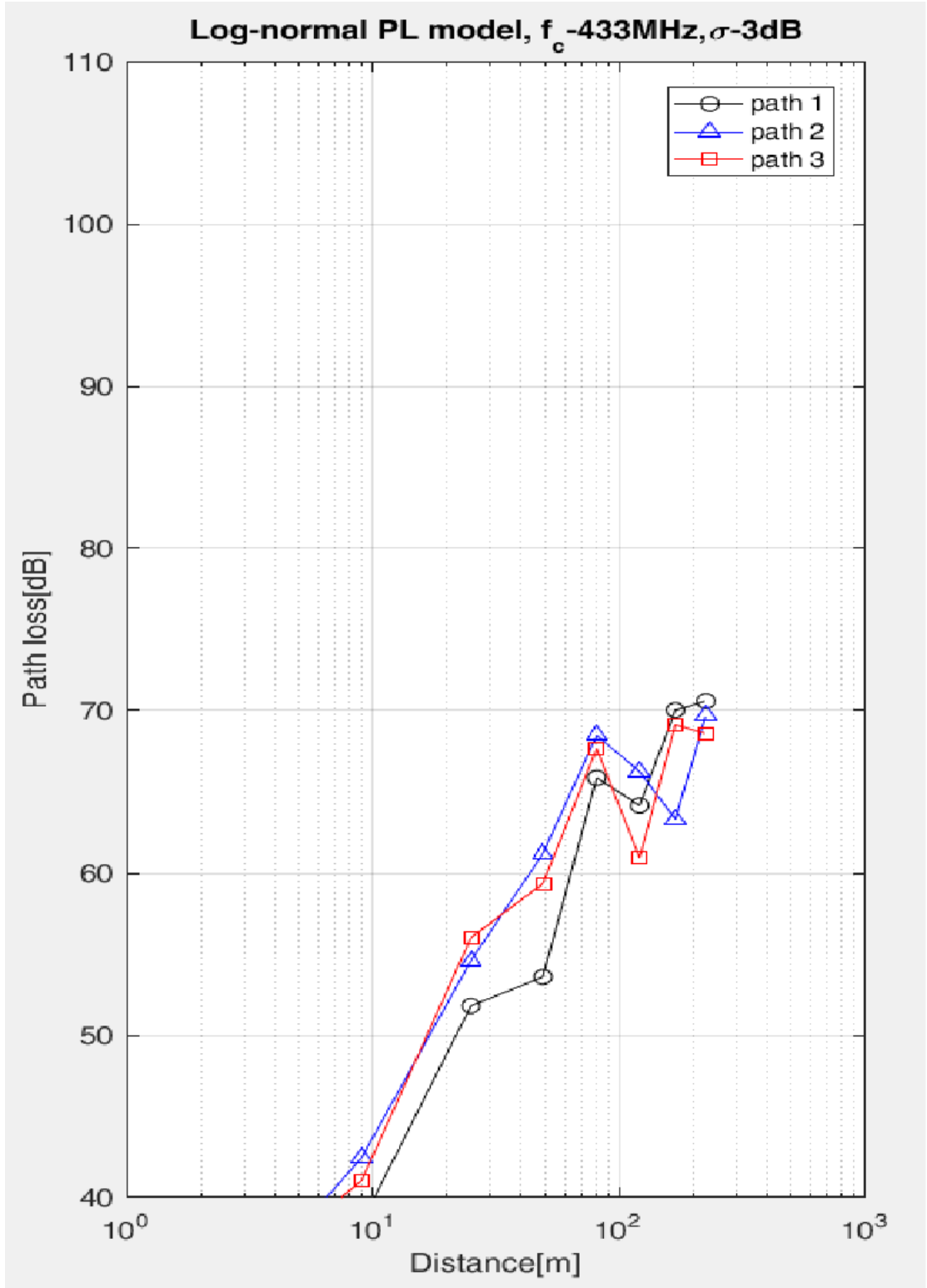
Simülasyon kısmında elde edilen grafiklerin tümünde ortak nokta mesafe arttıkça yol kayıplarının da artmasıdır. Şekil 4.1.'deki ilk değişim serbest uzay kayıplarını ifade etmektedir. Şekil 4.2.'de gölgeleme gürültüsü ilave edilmediğinden yol kayıpları lineer bir değişim göstermiştir. Bu grafikte görüldüğü üzere aynı mesafe için en çok yol kaybı $n=3$ için alınmıştır. Bu ifade yol kayıp üssü olarak ifade edilmekte olup, yol kaybıyla doğrusal bir değişime sahiptir. Şekil 4.3.'de ise bir öncekinden farklı olarak sadece rastgele ortama ilave edilen gölgeleme gauss gürültüsünden ötürü yol kaybı sürekli bir değişim göstermiştir. Tüm bu grafiklerde alıcı verici arasındaki mesafeye bağlı ne kadarlık yol kaybı olduğu gözlemlenebilmektedir.



Şekil 6.1. Serbest Uzay Kayıp Modeli için Yol Kaybı-Mesafe Değişimi



Şekil 6.2. Log distance kayıp modeli için yol kaybı-mesafe değişimi



Şekil 6.3. Log normal gölgeleme kayıp modeli için yol kaybı-mesafe değişimi

Simülasyon sonucunda alıcı ile verici düğümler arasındaki mesafeye bağlı olarak PSO algoritmasıyla uygunluk fonksiyonu olarak belirlenen iki nokta arası mesafe fonksiyonuyla hedef nokta arasındaki mesafeyi azaltacak şekilde parçacık hızları tayin edilmiştir. Sonuçta algortimanın durma şartı olan $k=100$ iterasyon gerçekleştiğinde hedef noktanın koordinatı (x_r, y_r) ' ye en yakın sürünün en iyi değeri olan $(34.7077, 188.5467)$ değeri bulunmuştur. Ancak hedef noktanın gerçek değeri $(13.7131, 189.9982)$ 'dir. İki nokta arasında görüldüğü üzere yaklaşık 20 m hata vardır. Ancak bu hata değeri çalışılan alanın kuş bakışı 200×200 'lük bir alan olduğu göz önüne alınırsa kabul edilebilir boyuttadır. Ayıryeten bu değer algoritma parametreleriyle bir miktar daha iyileştirilebilmektedir.

Sonuçta gerçekleştirilen simülasyon çalışmaları neticesinde gerekli entegrasyonun yapılması durumunda istenilen değerlere yakın değerler elde edilmiştir. Simülasyon çalışmaları sonucunda sistemin çalışabilmesi öngörülmektedir.

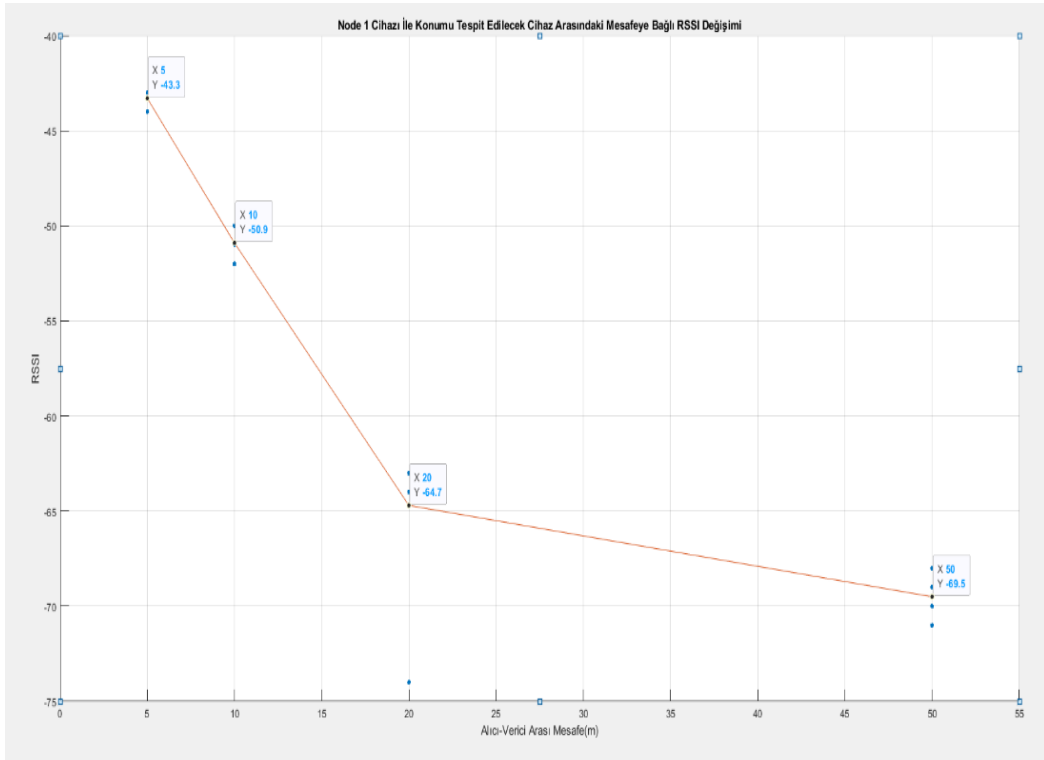
6.3. Deney Sonuçları

Sistemin değişen çevresel şartlar altındaki başarımını görebilmek için farklı ortamlarda test çalışmaları yapılmıştır. Bu bölümde farklı ortamlarda yapılmış iki farklı teste ait değerler tablolar ve grafikler halinde verilerek test sonuçları değerlendirilecektir. Tasarım aşamında yapılan simülasyon çalışmalarına olan benzerlikleri ele alınacak varsa farklılıklar nedenleriyle birlikte açıklanacaktır.

İlk çalışma Şekil 5.5'te verilen alanda gerçekleştirilmiştir. Bu alanda öncelikli olarak 1 m, 5 m, 10, ve 20 m mesafelerde sadece tek bir alıcı ve verici kullanılarak mesafeye bağlı RSSI değişimi elde edilmiştir. Bu işlem 4 adet verici için ayrı ayrı tekrarlanmıştır. Ölçüm sonuçları ise Çizelge 6.1'de verilmiştir. Akabinde çizelgede verilen değerler görselleştirilerek düğümlere ait mesafeye bağlı RSSI değişimleri verilmiştir.

Çizelge 6.1. Node 1 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri

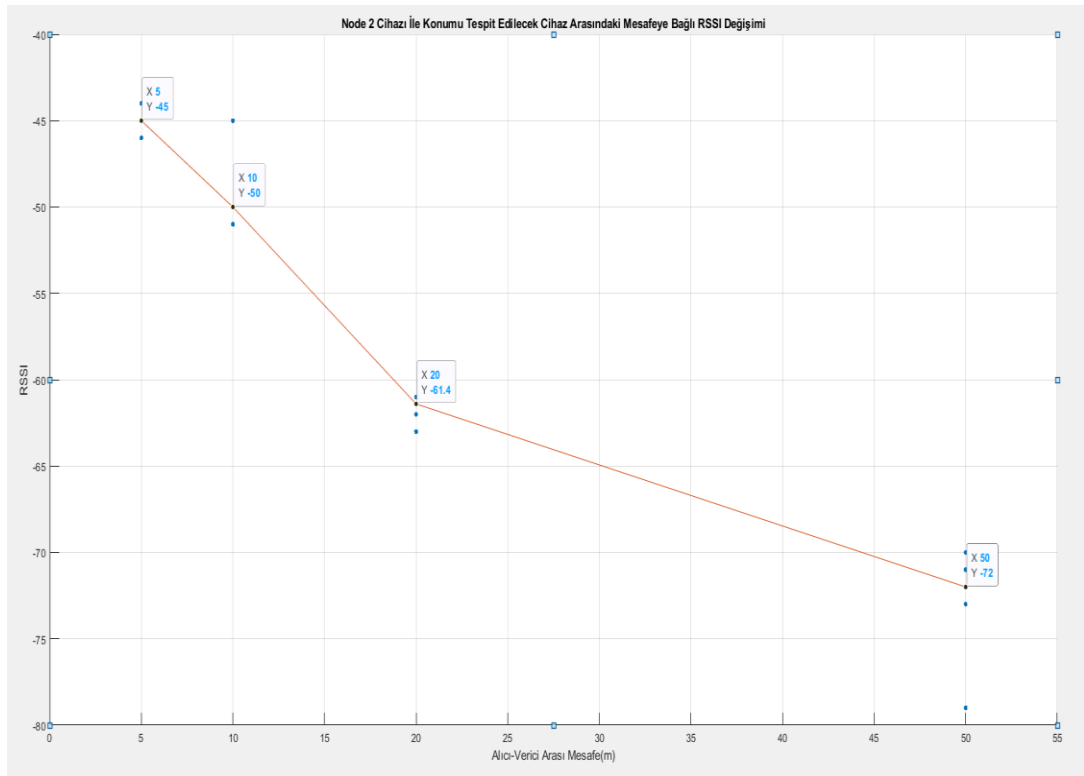
Gönderici Cihazın ID No	Paket Numarası	Verici ve Konumu Tespit Edilecek Cihaz Arasında Mesafeye (m) Bağlı RSSI Değeri			
		5 m	10 m	20 m	50 m
1	1	-43	-52	-74	-70
	2	-44	-51	-64	-69
	3	-43	-50	-63	-69
	4	-43	-50	-64	-69
	5	-43	-51	-64	-70
	6	-43	-52	-64	-71
	7	-44	-51	-63	-70
	8	-44	-51	-64	-69
	9	-43	-51	-63	-70
	10	-43	-50	-64	-68



Şekil 6.4. Node 1 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi

Çizelge 6.2. Node 2 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri

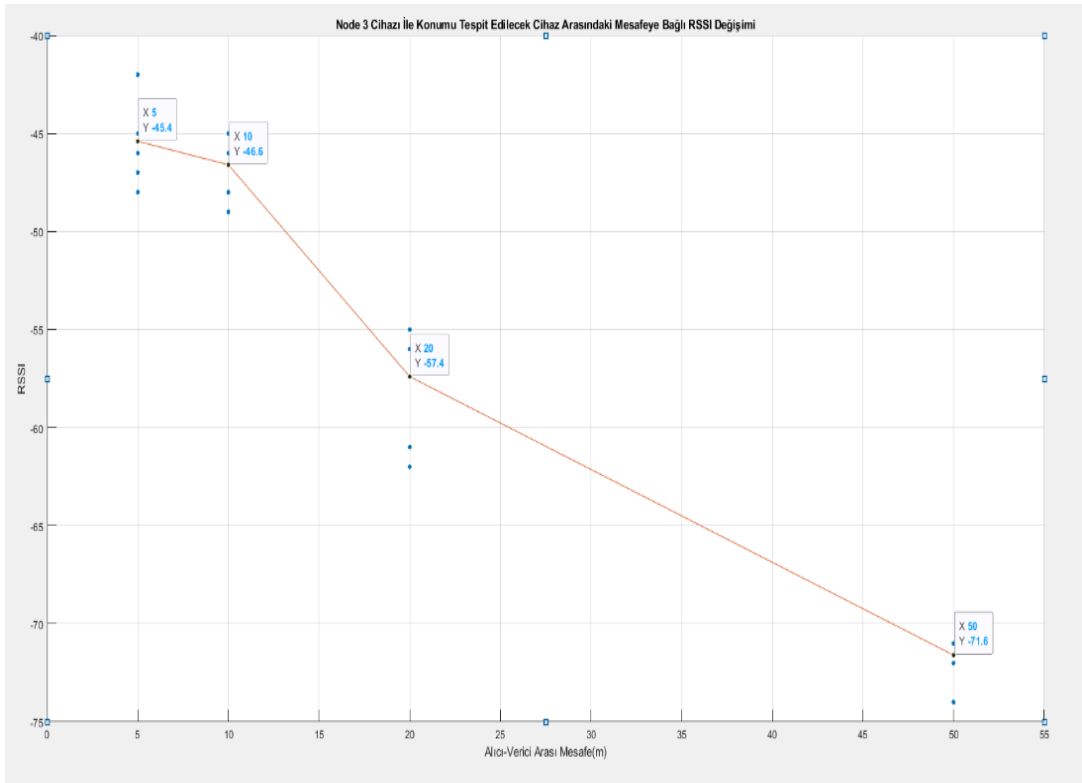
Gönderici Cihazın ID No	Paket Numarası	Verici ve Konumu Tespit Edilecek Cihaz Arasında Mesafeye (m) Bağlı RSSI Değeri			
		5 m	10 m	20 m	50 m
2	1	-45	-45	-61	-79
	2	-45	-51	-61	-73
	3	-45	-50	-61	-72
	4	-44	-50	-61	-72
	5	-44	-50	-62	-70
	6	-45	-50	-62	-70
	7	-45	-51	-63	-70
	8	-46	-51	-61	-72
	9	-46	-50	-61	-71
	10	-45	-51	-61	-71



Şekil 6.5. Node 2 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi

Çizelge 6.3. Node 3 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri

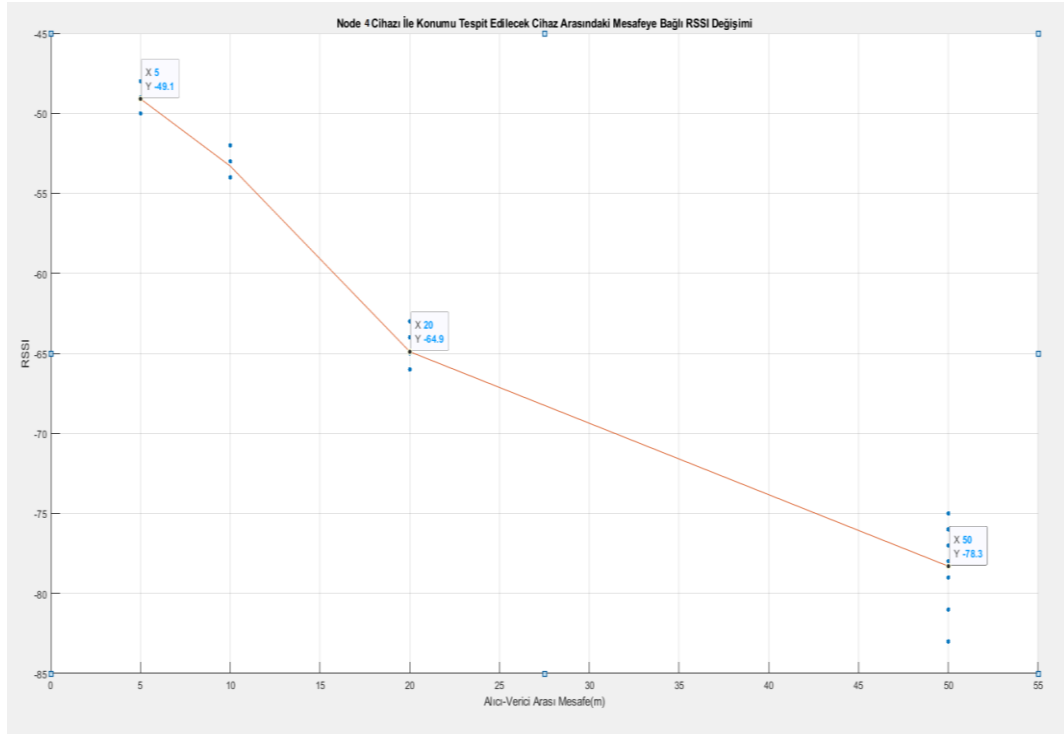
Gönderici Cihazın ID No	Paket Numarası	Verici ve Konumu Tespit Edilecek Cihaz Arasında Mesafeye (m) Bağlı RSSI Değeri			
		5 m	10 m	20 m	50 m
3	1	-47	-49	-55	-72
	2	-48	-49	-56	-71
	3	-48	-48	-56	-71
	4	-48	-46	-56	-71
	5	-46	-46	-55	-71
	6	-45	-46	-56	-71
	7	-46	-46	-56	-72
	8	-42	-45	-61	-71
	9	-42	-45	-61	-74
	10	-42	-46	-62	-72



Şekil 6.6. Node 3 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi

Çizelge 6.4. Node 4 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri

Gönderici Cihazın ID No	Paket Numarası	Verici ve Konumu Tespit Edilecek Cihaz Arasında Mesafeye (m) Bağlı RSSI Değeri			
		5 m	10 m	20 m	50 m
4	1	-50	-52	-66	-78
	2	-48	-54	-66	-81
	3	-49	-53	-66	-76
	4	-49	-54	-66	-78
	5	-49	-53	-64	-75
	6	-49	-54	-64	-77
	7	-49	-53	-63	-77
	8	-49	-54	-66	-83
	9	-49	-53	-65	-79
	10	-50	-53	-63	-79



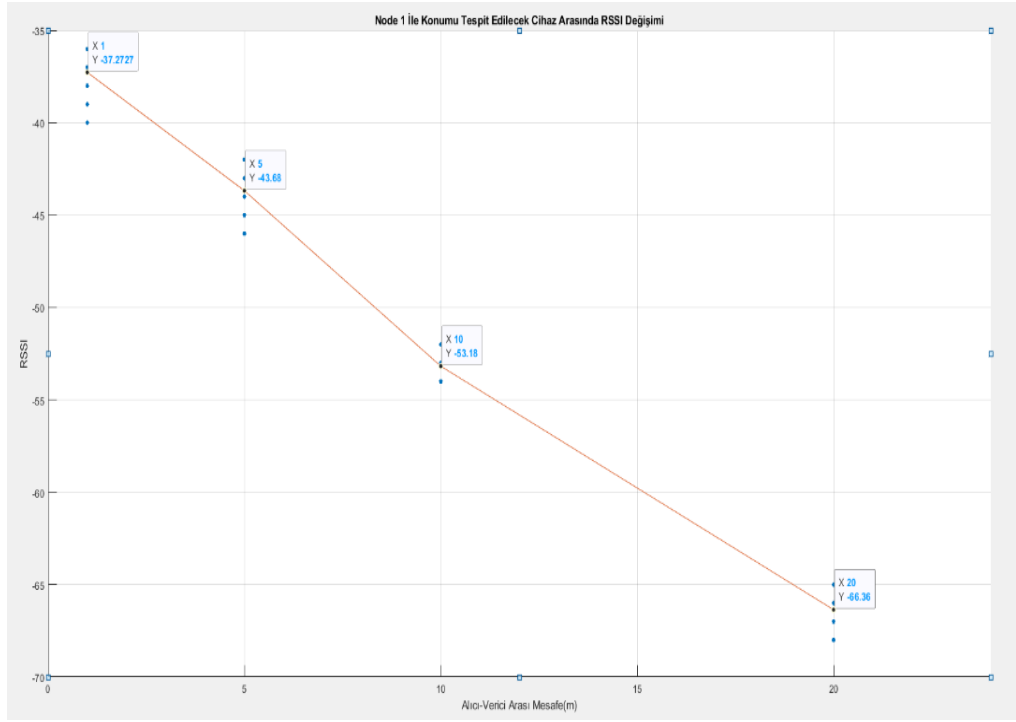
Şekil 6.7. Node 4 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi

İlk deneysel çalışmaya ait çizelge ve grafikler incelendiğinde her bir alıcı ve verici arasındaki RSSI değeri sürekli olarak değişkenlik gösterip belli bir aralıkta kümelenmiştir. Node 1 için 5 metredeki RSSI değeri -43 seviyelerindeyken, aynı mesafe için bu değer Node 2’de -45, Node 3’te -46 iken Node 4’ de ise -49 seviyelerindedir. Bu ölçümler daha da kırsal bir alanda yapılacak olursa alıcı ile verici arasındaki engeller azalacağından sinyal kalitesi de artarak RSSI değeri sıfıra doğru yaklaşacaktır. Diğer bir yandan cihazlar arasındaki mesafe arttıkça yol kayıpları artacağından alınan sinyalin gücü de zayıflayacak ve RSSI değeri de düşecektir. Node 1 için 5 metredeki RSSI=-43 iken 50 metredeki RSSI=-70 seviyelerine çıkmıştır. Bu durum beklenen bir durumdur.

Aynı deney düzeneği bu sefer de Şekil 5.6.’te gösterilen ortamda denenmiştir. Ancak bu sefer verici ile konumu tespit edilecek cihaz arasındaki mesafe 1 m, 5 m, 10 m ve 20 m ile sınırlandırılmıştır. Çizelge 6.5/6/7/8’te test sonuçlarına ait ölçüm sonuçları verilmiştir. Şekil 6.5/6/7/8.’ te de çizelgedeki değerler görselleştirilerek grafik üzerinde ifade edilmiştir.

Çizelge 6.5. Node 1 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri

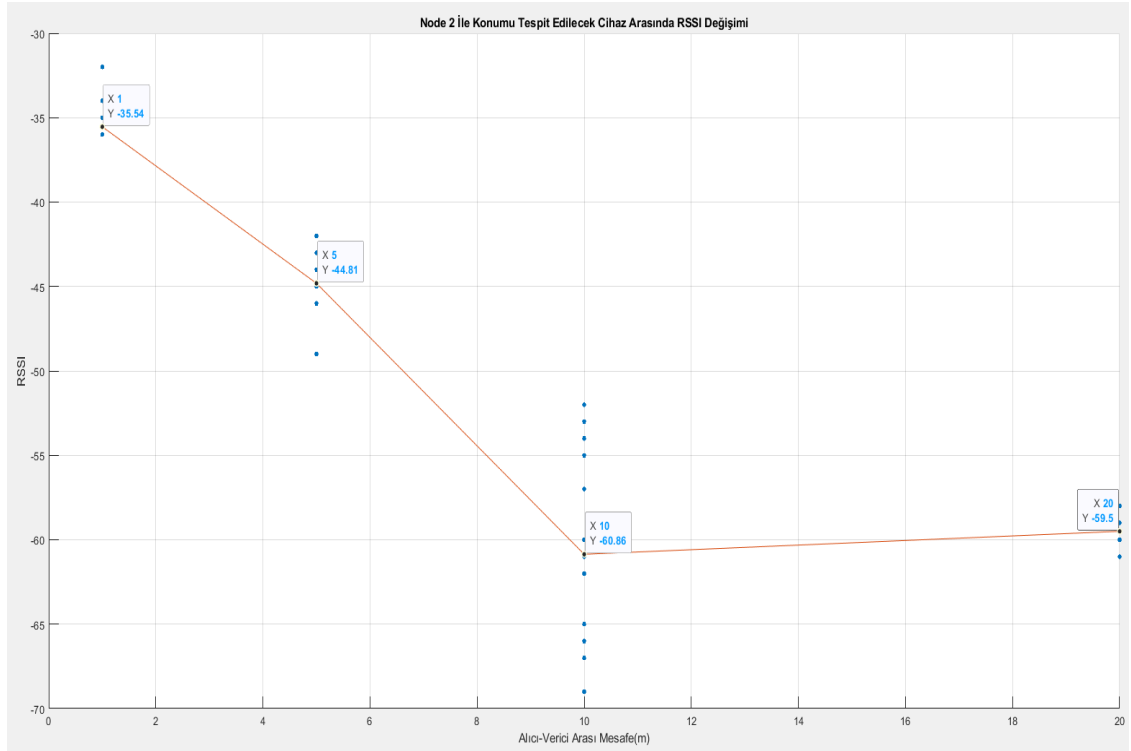
Gönderici Cihazın ID No	Paket Numarası	Verici ve Konumu Tespit Edilecek Cihaz Arasında Mesafeye (m) Bağlı RSSI Değeri			
		1 m	5 m	10 m	20 m
1	1	-38	-44	-54	-65
	2	-37	-43	-52	-66
	3	-38	-43	-53	-67
	4	-37	-45	-54	-68
	5	-39	-42	-53	-68
	6	-38	-42	-53	-68
	7	-40	-43	-53	-68
	8	-39	-44	-52	-66
	9	-38	-42	-53	-66
	10	-38	-44	-53	-66
	11	-38	-44	-53	-65
	12	-38	-46	-53	-65
	13	-37	-44	-54	-65
	14	-36	-44	-53	-65
	15	-36	-44	-53	-66
	16	-36	-45	-53	-67
	17	-36	-43	-54	-66
	18	-36	-42	-53	-68
	19	-36	-44	-53	-67
	20	-36	-44	-53	-67
	21	-37	-45	-54	-66
	22	-36	-44	-54	-65



Şekil 6.8. Node 1 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi

Çizelge 6.6. Node 2 ile konumu tespit edilecek cihaz arasında farklı mesafelerde edilen RSSI değerleri

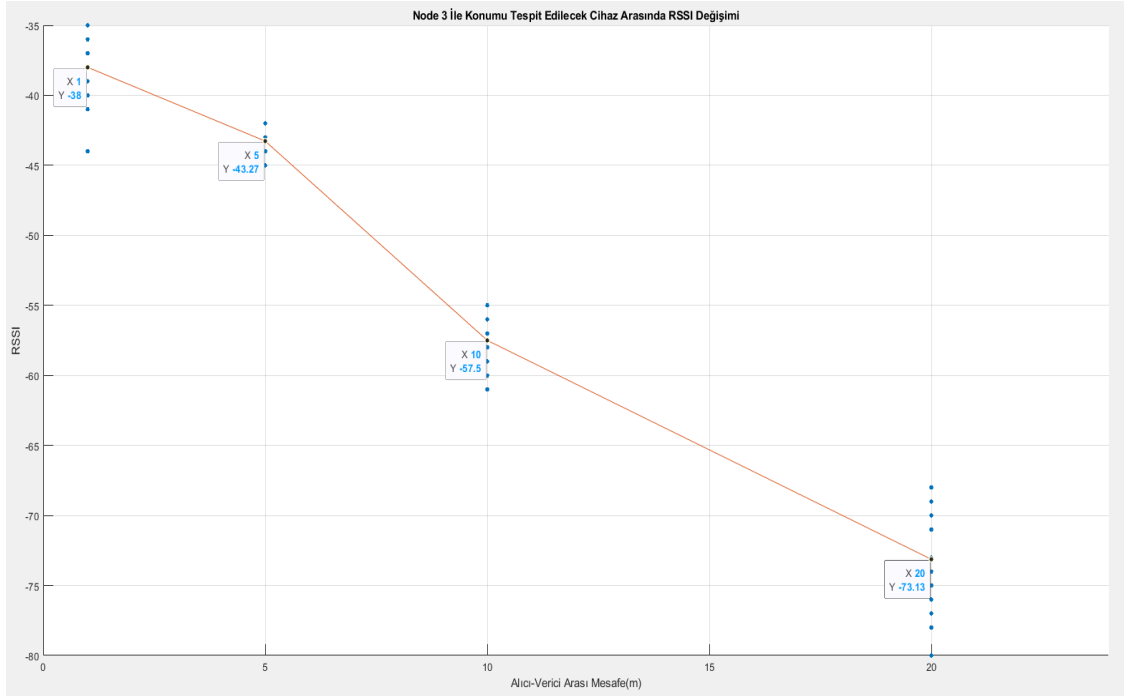
Gönderici Cihazın ID No	Paket Numarası	Verici ve Konumu Tespit Edilecek Cihaz Arasında Mesafeye (m) Bağlı RSSI Değeri			
		1 m	5 m	10 m	20 m
2	1	-35	-42	-53	-60
	2	-32	-45	-54	-60
	3	-34	-44	-54	-60
	4	-35	-44	-54	-58
	5	-36	-43	-52	-59
	6	-36	-45	-55	-60
	7	-36	-46	-57	-60
	8	-36	-49	-60	-59
	9	-36	-46	-65	-59
	10	-36	-45	-65	-60
	11	-36	-46	-66	-59
	12	-36	-46	-67	-59
	13	-36	-46	-66	-59
	14	-36	-45	-67	-60
	15	-36	-46	-69	-59
	16	-36	-45	-66	-59
	17	-36	-45	-61	-59
	18	-36	-46	-61	-60
	19	-36	-42	-62	-60
	20	-36	-45	-62	-60
	21	-35	-43	-62	-61
	22	-35	-42	-61	-61



Şekil 6.9. Node 2 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi

Çizelge 6.7. Node 3 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri

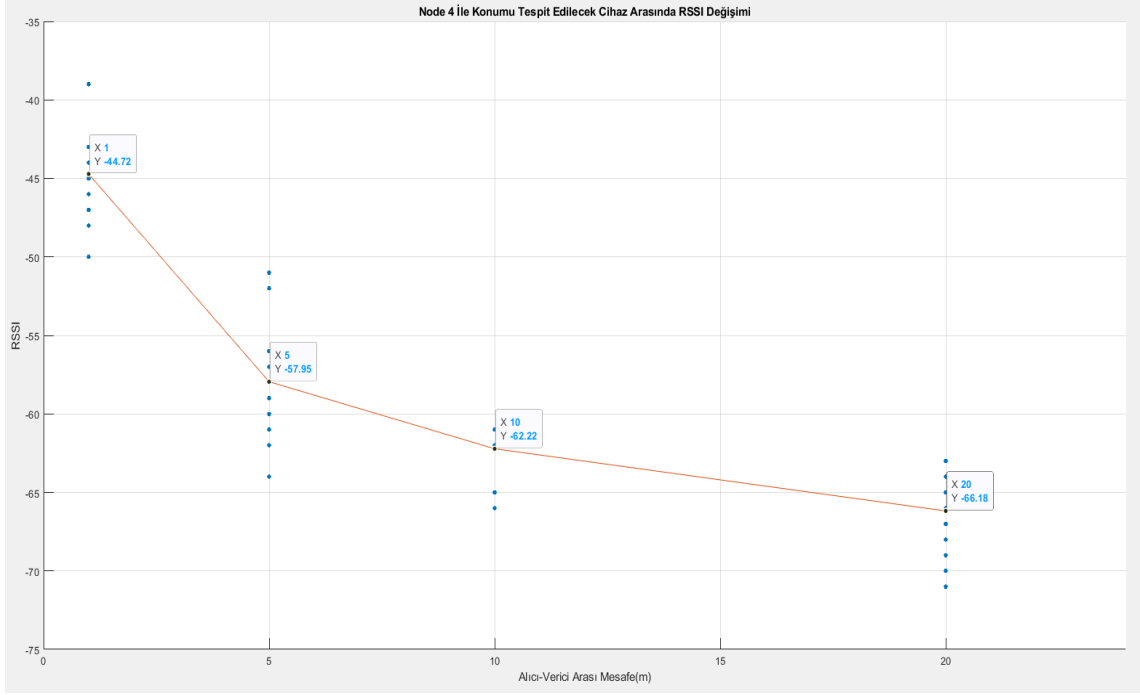
Gönderici Cihazın ID No	Paket Numarası	Verici ve Konumu Tespit Edilecek Cihaz Arasında Mesafeye (m) Bağlı RSSI Değeri			
		1 m	5 m	10 m	20 m
3	1	-40	-44	-56	-70
	2	-38	-43	-57	-68
	3	-37	-43	-56	-69
	4	-37	-42	-57	-71
	5	-37	-42	-57	-77
	6	-37	-44	-56	-73
	7	-39	-45	-56	-74
	8	-36	-44	-58	-74
	9	-36	-43	-58	-75
	10	-36	-44	-59	-75
	11	-37	-44	-59	-80
	12	-36	-44	-59	-78
	13	-36	-43	-60	-75
	14	-37	-43	-61	-74
	15	-35	-43	-60	-74
	16	-44	-43	-55	-74
	17	-41	-43	-56	-70
	18	-41	-44	-57	-68
	19	-40	-42	-57	-76
	20	-41	-43	-57	-73
	21	-38	-43	-57	-71
	22	-37	-43	-57	-70



Şekil 6.10. Node 3 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi

Çizelge 6.8. Node 4 ile konumu tespit edilecek cihaz arasında farklı mesafelerde elde edilen RSSI değerleri

Gönderici Cihazın ID No	Paket Numarası	Verici ve Konumu Tespit Edilecek Cihaz Arasında Mesafeye (m) Bağlı RSSI Değeri			
		1 m	5 m	10 m	20 m
4	1	-43	-52	-62	-67
	2	-46	-51	-61	-68
	3	-48	-51	-62	-64
	4	-50	-52	-62	-66
	5	-44	-59	-61	-64
	6	-44	-59	-62	-63
	7	-45	-60	-62	-63
	8	-47	-64	-62	-65
	9	-44	-61	-62	-65
	10	-45	-60	-62	-64
	11	-45	-62	-62	-65
	12	-46	-58	-62	-64
	13	-45	-57	-61	-65
	14	-44	-56	-62	-68
	15	-44	-60	-61	-67
	16	-45	-57	-61	-70
	17	-44	-61	-61	-69
	18	-44	-61	-62	-64
	19	-44	-59	-62	-66
	20	-39	-59	-66	-70
	21	-44	-59	-66	-71
	22	-44	-57	-65	-68



Şekil 6.11. Node 4 cihazı ile hedef cihaz arasındaki mesafeye bağlı RSSI değişimi

İkinci test ortamı için yapılan ölçümlerden görüldüğü üzere 5 m, 10 m ve 20 m değerleri için büyük oranda alınan işaretin gücünde zayıflama olmuştur. Örneğin Node 1 için 20 metredeki RSSI değeri -66.36 iken, Node 2’de bu değer -60, Node 3’te -73.13, Node 4’te ise -66.12 olarak ölçülmüştür. Diğer yandan ilk test ortamında Node 1’in 20 metredeki RSSI değeri -64.7 iken, Node 2’de bu değer -61.4, Node 3’te -57.4, Node 4’te ise -64.9 olarak bulunmuştur. Görüldüğü üzere genel olarak yol kayıplarından kaynaklı zayıflama ikinci ortamda daha fazladır. Bu zayıflama test ortamındaki yansıtıcı yüzeylerden veya alıcı-verici arasındaki engellerden dolayı kaynaklanabilir. Sinyali zayıflatan bu etkenler ikinci test ortamında daha fazladır. Özellikle etrafta bulunan cisimlerden saçılan dalgalar alıcıda fadinge neden olur.

Tüm bu testler simülasyon aşamasında yapay olarak oluşturulan çevresel ortamlardaki kayıplarla aynı yönde değişim göstermiştir. Ancak deneysel çalışmalarda yol kayıpları daha fazla gelmiştir. Bunun sebebi simülasyon aşamasında ne kadar rastgele gürültü eklenip, yol kayıp üssü deneysel değerlere yakın olsa da gerçekteki test ortamı sadece bu parametrelerden ibaret değildir. Bunlar dışında sisteme etkiyen bozucu etkenlerde göz önüne alındığında simülasyon sonuçlarıyla deneysel sonuçlar birbiriyle tutarlıdır.

Son olarak tüm verici cihazları 115 m – 50 m dikdörtgen şeklindeki Şekil 6.12’de gösterildiği gibi bir alanın köşelerine yerleştirildi ve konumu tespit edilecek cihaz bu karenin tam orta noktasına yerleştirildi. Hedef cihazın her bir düğüme olan uzaklığı teorik olarak bellidir. Bu uzaklıkta dikdörtgen alanın köşegen uzunluğunun yarısıdır. Diğer yandan hedef cihazın ölçümlere dayalı konumu Çizelge 6.9’ te verilmiştir.

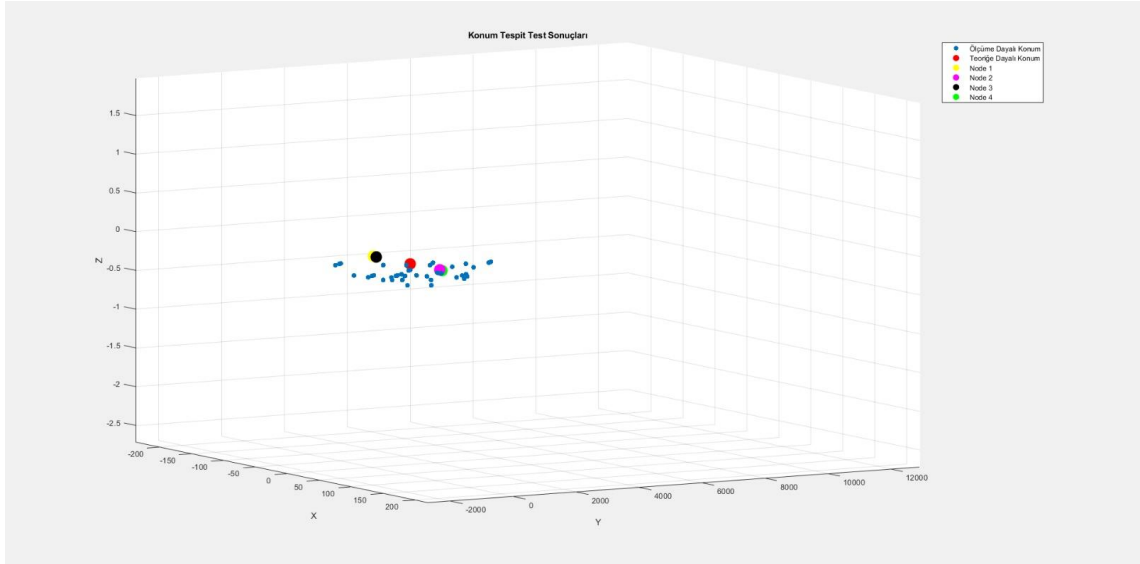


Şekil 6.12. Sistemin tamamının test edildiği alan

Çizelge 6.9. Hedef cihazın test alanının ortasında olduğu durum

Hedef Cihaz Test Alanının Tam Ortasında			
	X	Y	Z
Deneysel Sonuçlar	65.08	594.64	144.09
	94.8	605.87	147.28
	74.69	1447.91	-59.1
	140.78	-9.85	-2.98
	75.93	2147.16	-90.88
	72.45	2285.97	-97.28
Teorik Sonuç	57.5	25	50
Hata (m)	76.2		

Çizelge 6.9.'u oluştururken kullanılan veriler 3 boyutlu grafik haline getirilirse Şekil 6.13.'teki grafik elde edilir.



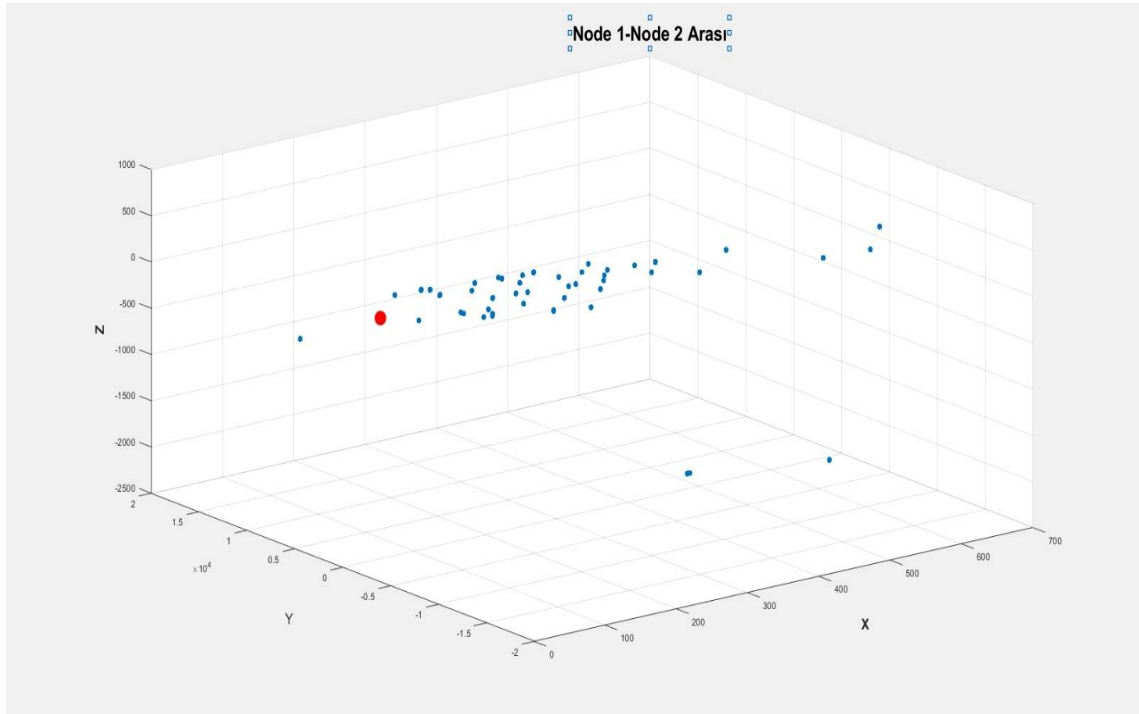
Şekil 6.13. Teorik sonuçlar ve test sonuçlarının dağılımı

Çizelge 6.9.'da yapılan işlemlerin aynısı hedef cihazın test alanı için farklı noktalarında tekrarlanabilir. Aynı işlemler hedef cihazın 1-2, 2-4, 3-4 ve 1-3 düğümleri arasında olduğu durum için tekrarlanabilir. Bu durumda aşağıda verilen çizelge ve grafikler elde edilir.

Çizelge 6.10. Hedef cihaz Node1 ile Node2'nin ortasında olduğu durum

Hedef Cihaz Node1 İle Node2 Arasında			
	X	Y	Z
Deneysel Sonuçlar	326.34	-42.38	-9.68
	327.39	28.29	-12.94
	319.73	199.94	-30.87
	350.65	-1146.35	31.2
	1.46	-17.24	0.55
	1.47	-13.92	0.53
Teorik Sonuç	53	1.5	101.5
Hata (m)	113		

Çizelge 6.10.'u oluştururken kullanılan veriler 3 boyutlu grafik haline getirilirse Şekil 6.14.'teki grafik elde edilir.

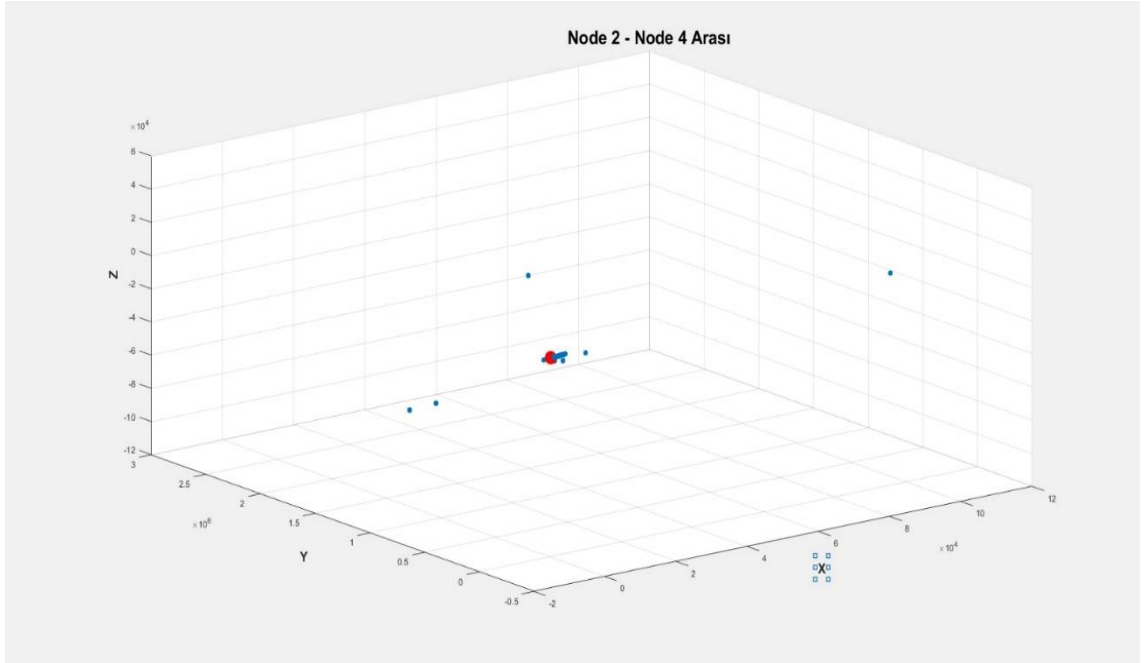


Şekil 6.14. Hedef cihaz Node1 ile Node2 arasında olduğunda teorik ve test sonuçları

Çizelge 6.11. Hedef cihaz Node2 ile Node4'nin ortasında olduğu durum

Hedef Cihaz Node2 İle Node4 Arasında			
	X	Y	Z
Deneysel Sonuçlar	1.7125	74.64	-3.97
	2.11	101.77	-5.2254
	13.16	56.04	-3.7
	20.98	54.55	-2.58
	102.45	110.9	-18.56
	10.15	124.33	-4.93
Teorik Sonuç	105.5	27	52.5
Hata (m)	593.94		

Çizelge 6.11.'u oluştururken kullanılan veriler 3 boyutlu grafik haline getirilirse Şekil 6.15.'teki grafik elde edilir.

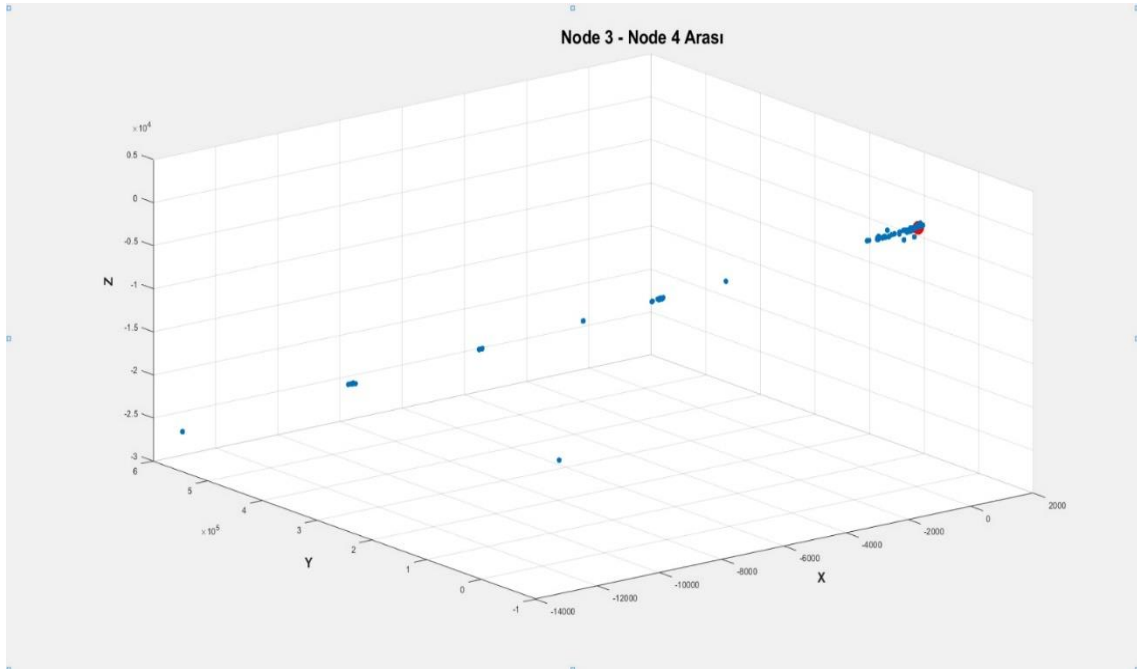


Şekil 6.15. Hedef cihaz Node2 ile Node4 arasında olduğunda teorik ve test sonuçları

Çizelge 6.12. Hedef cihazın Node3 ile Node4 arasında olduğu durum

Hedef Cihaz Node3 İle Node4 Arasında			
	X	Y	Z
Deneysel Sonuçlar	-11.3	57.1	-2.45
	-0.39	-4.56	0.25
	-4.03	85.68	-3.84
	-4.78	33.66	-1.62
	-2.32	18.07	-0.67
	-1.91	46.31	-1.88
Teorik Sonuç	54.5	52.5	2.5
Hata (m)	434.28		

Çizelge 6.12.'u oluştururken kullanılan veriler 3 boyutlu grafik haline getirilirse Şekil 6.16.'teki grafik elde edilir.

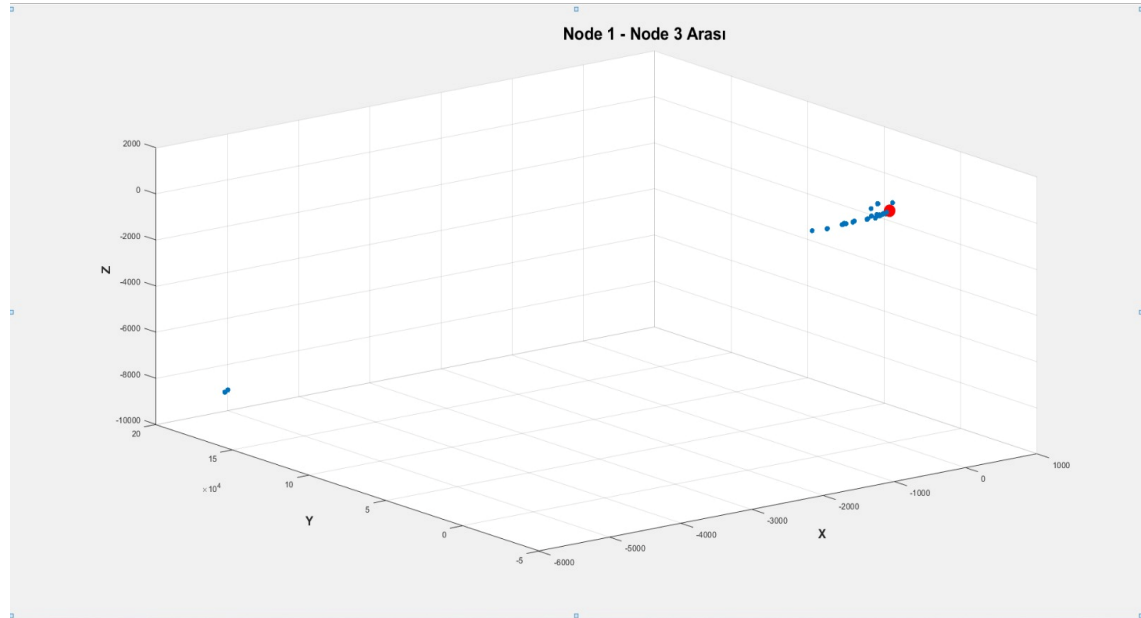


Şekil 6.16. Hedef cihaz Node3 ile Node4 arasında olduğunda teorik ve test sonuçları

Çizelge 6.13. Hedef cihaz Node1 ile Node3'ün ortasında olduğu durum

Hedef Cihaz Node1 İle Node3 Arasında			
	X	Y	Z
Deneysel Sonuçlar	-1.37	19.68	-0.73
	-1.77	35.85	-1.48
	-0.26	33.01	-1.5
	-0.25	32.67	-1.48
	-0.51	14.9	-0.66
	-0.05	27.37	-1.13
Teorik Sonuç	2	27	51.5
Hata (m)	319.68		

Çizelge 6.12.'u oluştururken kullanılan veriler 3 boyutlu grafik haline getirilirse Şekil 6.17.'teki grafik elde edilir.



Şekil 6.17. Hedef cihaz Node1 ile Node3 arasında olduğunda teorik ve test sonuçları

7. DEĞERLENDİRMELER

.Bu çalışma ile temel haberleşme ve IoT projelerine karşı deneyim kazanıldı. Teorilerin gerçek hayatta uygulanırken karşılaşılan gözardı edilen problemlerle başa çıkıldı. Ve böylece ileriki mühendislik hayatımızda gerçekleştireceğimiz projeler ile birebir çalışma imkânı bulunuldu. Bu çalışma gerekli eklemelerin yapılmasıyla hem tezde yazıldığı alanlarda hem de GPS ile konum tespitinin gerçekleşmesinin engellediği alanlarda/ sınır bölgelerinde kullanılabilir. Haberleşme modüllerinin kullanılmasından dolayı sadece konum tespiti için değil, konum tespiti yanında birçok IoT proje gerçekleştirilebilir.

Bu çalışma GPS sisteminin bulunmadığı ya da kullanılmadığı sivil alanlarda, çeşitli sınır bölgelerinde ve sürekli belirli bir alanda çalışacak otonom cihazlar için GPS'ten bağımsız alanlarda kullanılabilir. Müşterisi ise sivil ve askeri alanda, çeşitli IoT projeleri geliştiren şirketlerde ve diğer birçok alanda kullanılabilir.

Tasarımda hesaplanan bütçe ile tüm tasarım malzemeleri temin edildi. Ancak sonrasında donanımlarda çıkan beklenilmedik sorunlarda dolayı yenisi tedarik edildi. Bu yüzden gerçekleştirilen bütçe ile hesaplanan bütçe arasında farklılıklar oluştu. Bu problem sayesinde gerçek hayatta mühendislik ekonomisi gibi alanda deneyim kazanıldı.

KAYNAKLAR

- [1] W. Kuang, M. Zhang, W. Li, C. Chen, and M. Xia, "3D Outdoor Positioning Based on RSSI," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct. 2018, pp. 1–5, doi: 10.1109/WCSP.2018.8555599.
- [2] K.-H. Lam, C.-C. Cheung, and W.-C. Lee, "RSSI-Based LoRa Localization Systems for Large-Scale Indoor and Outdoor Environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11778–11791, Dec. 2019, doi: 10.1109/TVT.2019.2940272.
- [3] N. El Agroudy, N. Joram, and F. Ellinger, "Low power RSSI outdoor localization system," in *2016 12th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, Jun. 2016, pp. 1–4, doi: 10.1109/PRIME.2016.7519456.
- [4] N. N. SÜMER and A., "KAPALI ALAN KONUM TESPİTİNDE RSSI TABANLI DÜŞÜK ENERJİLİ BLUETOOTH KULLANIMI NİYAZI," in *5th International Conference on Computer Science and Engineering*, 2020, pp. 1–9.
- [5] A. S. Doğru, B. Temel, and T. Eren, "Comparison of Particle Swarm Optimization and Bat Algorithm Methods in Localization of Wireless Sensor Networks," *Uluslararası Muhendis. Arastirma ve Gelistirme Derg.*, vol. 11, no. 3, pp. 793–801, 2019, doi: 10.29137/umagd.668724.
- [6] H. Kaya, "UART Nedir, Çalışma Yapısı ve Seri Haberleşme," 2018. <https://www.hakankaya.kim/blog/uart-temelleri-uart-nedir-calisma-yapisi-ve-seri-haberlesme/> (accessed Jan. 02, 2022).
- [7] I. I. S. S.L., "Choose the right transmission protocol for your IoT solution," 2021. <https://www.inovum-solutions.com/choosing-the-right-technology-for-your-iot-solution-a-big-decision/> (accessed Jan. 02, 2022).
- [8] Priyanka Purbe, "LoRa, SigFox, and LTE-M Technologies for long-range communications," 2019. <https://volansys.com/selecting-right-low-power-wide-area-network-lpwan-technology/> (accessed Jan. 02, 2022).
- [9] T. Turkoral, O. Tamer, S. Yetis, E. Inanc, and L. Cetin, "Indoor distance estimation with using received signal strength indicator (RSSI) metric, Alinan isaret güç göstergesi (RSSI) metrigi kullanılarak kapali alan mesafe kestirimi," *2016 Natl. Conf. Electr. Electron. Biomed. Eng. ELECO 2016*, pp. 397–401, 2017, [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-85015417167&partnerID=MN8TOARS>.
- [10] J. S. Seybold, *Introduction to RF Propagation*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005.

- [11] S. ERHAN, “AKILLI ULAŞIM SİSTEMLERİ ve OTONOM TAŞIT TEKNOLOJİLERİ İÇİN YÜKSEK DOĞRULUKLU KONUMLANDIRMANIN SAĞLANMASI,” Doktora Tezi, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü, Trabzon, Türkiye, Kasım, 2019.
- [12] T. S. Rappaport, *Wireless communications : principles and practice*. Prentice Hall PTR, 1996.
- [13] R. Muppala, A. Navnit, D. Devendra, E. R. Matera, N. Accettura, and A. M. Hussain, “Feasibility of Standalone TDoA-based Localization Using LoRaWAN,” in *2021 International Conference on Localization and GNSS (ICL-GNSS)*, Jun. 2021, pp. 1–7, doi: 10.1109/ICL-GNSS51451.2021.9452311.
- [14] O. ÇAKIR, “VARIŞ ZAMANLARI FARKI TABANLI VE YÜKSEK DOĞRULUKLU YENİ KONUMLANDIRMA YÖNTEMLERİ,” Doktora Tezi, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü, Trabzon, Türkiye, Şubat, 2015.
- [15] N. BENKAHLA, H. TOUNSI, Y.-Q. SONG, and M. FRIKHA, “Enhanced ADR for LoRaWAN networks with mobility,” in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Jun. 2019, no. November, pp. 1–6, doi: 10.1109/IWCMC.2019.8766738.
- [16] E-Mevzuat, “FREKANS TAHSİSİNDEN MUAF TELSİZ CİHAZ VE SİSTEMLERİ HAKKINDA YÖNETMELİK,” Bilgi Teknolojileri ve İletişim Kurumu, Web [online]. (<https://www.resmigazete.gov.tr/eskiler/2018/11/20181127-18.htm>), Erişim tarihi: 22 Ocak 2022.
- [17] E-Mevzuat, “ELEKTRONİK HABERLEŞME KANUNU”, B. M. G. ve Y. G. Müdürlüğü, Web [online]. (<https://www.resmigazete.gov.tr/eskiler/2008/11/20081110M1-3.htm>), Erişim tarihi: 5 Kasım 2008.
- [18] X.-S. Yang, “Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics,” p. 160, 2008.

EKLER

EK-1. IEEE Etik Kuralları (Türkçe) ve IEEE Code of Ethics (İngilizce)



IEEE üyeleri olarak bizler bütün dünya üzerinde teknolojilerimizin hayat standartlarını etkilemesindeki önemin farkındayız. Mesleğimize karşı şahsi sorumluluğumuzu kabul ederek, hizmet ettiğimiz toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmayı söz verdiğimizizi ve aşağıdaki etik kuralları kabul ettiğimizi ifade ederiz.

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.



IEEE Code of Ethics



We, the members of the IEEE, in recognition of the importance of our technologies in affecting the quality of life throughout the world, and in accepting a personal obligation to our profession, its members and the communities we serve, do hereby commit ourselves to the highest ethical and professional conduct and agree:

1. to accept responsibility in making engineering decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;
2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;
3. to be honest and realistic in stating claims or estimates based on available data;
4. to reject bribery in all its forms;
5. to improve the understanding of technology, its appropriate application, and potential consequences;
6. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
7. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;
8. to treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin;
9. to avoid injuring others, their property, reputation, or employment by false or malicious action;
10. to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

Approved by the IEEE Board of Directors
August 1990

EK-2. Disiplinlerarası Çalışma

DISİPLİNLER ARASI ÇALIŞMA FORMU

Disiplinler arası çalışma	Açıklama
Bölüm Başkanlığınca organize edilen çalıştaylar veya ders	<p>Düzenlenen çalıştay programında günlük hayatta karşılaştığımız herhangi bir probleme karşı TRİZ yaklaşımıyla nasıl çözüm bulunabileceğinden bahsedildi. Triz yaklaşımıyla geliştirilen örnekler incelendi ve konu pekiştirildi. Çalışmalar sonucunda TRİZ'in bir problemi çözüm kümesinde nasıl arama yapılacağını, efektif çözümlere nasıl ulaşılacağını ve kılavuz niteliğindeki bu yöntemin nasıl kullanılacağı öğrenildi. Çalıştayın devamında, Elektrik/Elektronik Mühendisliği bölümü ve Bilgisayar Mühendisliği bölümünden öğrencileri bir araya gelerek bir problem tespit etti. Kısıtlı bir süre içerisinde belirlenen problemin gelişen ve zayıflayan yönleri belirlendi. Ardından TRİZ çelişki matrisinden yararlanarak olası çözüm kümesi tespit edildi ve en yüksek çözüm potansiyeli olan yaratıcılık ilkesi belirlendi. Belirlen ilkeyi temel alarak bu ilkeyi geliştirecek çözüm önerileri için takım arkadaşlarıyla beyin fırtınası yapıldı. En uygun çözüm önerisi belirlendi ve raporlandı.</p> <p>Disiplinler arası gerçekleştirilen bu çalıştay sonucunda ilgi alanları birbirinden farklı kişilerle takım olarak hareket etmenin önemi anlaşıldı. Takım olmanın sağladığı avantajlar TRİZ yöntemiyle daha da güçlendirildi. Problemin belirlenmesi aşamasında dahi görüş ayrıklarına, konuda dağılmalar olmasına rağmen çözüm aşamasında TRİZ yönteminin kullanılmasıyla hem takım arkadaşları arasında motivasyonun arttığı hem de çözüm fikirlerinde çeşitliliklerin sağlandığı görüldü.</p>

GİRİŞİMCİLİK ÇALIŞTAYI G29 -GRUBU RAPORU

Enes KURT 365028 --- Elektrik-Elektronik Mühendisliği Bölümü

Samet DOĞAN 365029 --- Elektrik-Elektronik Mühendisliği Bölümü

Beyza Gül ŞİMŞEK 365086 --- Elektrik-Elektronik Mühendisliği Bölümü

Münire Selin GÖKÇEBAŞ 365082 --- Elektrik-Elektronik Mühendisliği Bölümü

Furkan BİLGE 365041 --- Elektrik-Elektronik Mühendisliği Bölümü

Problemin Tanımı

Gelişen teknolojiyle birlikte elektronik cihazların enerji ihtiyacı artmaktadır. Bunun çözümüne yönelik günlük hayatta çoğunlukla kablolu şarj cihazları kullanılmaktadır. Bu cihazlarda malzeme dayanımının az olması nedeni ile kullanım ömrü az olmaktadır.

Teknik Çelişkilerin Formüle Edilmesi

Kullanılan malzemeye bağlı olarak fiziksel (uzunluk, ağırlık, tür, hacim, alan) özelliklerinde değişiklik yapılır.

Kullanılan malzemenin türüne göre dayanıklılık değişir.

Seçilen malzemeye göre üretim yöntemi değişeceğinden üretim kalitesi değişir. Buna bağlı olarak zaman kaybı ortaya çıkar.

Çelişki Matrisinden;

İyileşen özellik--→Dayanım

Kötüleşen özellik→Zaman Kaybı

Çelişki Matrisiyle Çalışma

Çelişki Matrisinden (14) satır ve (25) sütunun çaprazlanmasında, çelişkiyi çözmeyi amaçlayan ilkeler (29,3,28,10) olarak belirlenir.

Bu ilkelerden 3 numaralı ilkenin yaklaşımı sorunun çözümüne uygun görülmüştür.

3 numaralı ilkeye bakış:

Sonuç olarak bir cismin ya da dış ortamın/eylemin homojen yapısının heterojen bir yapıya dönüştürülmesi veya cismin farklı parçalarının farklı işlevler yerine getirmesini sağlama maddeleri temellendirmesiyle kompozit malzeme kullanımı ile sorun çözülür.

GİRİŞİMCİLİK ÇALIŞTAYI G3 -GRUBU RAPORU

- 365036 Alipaşa ÖKSÜZ
- 383257 Çağatay Ozan AKŞAN
- 383235 Sezgin SEFER
- 365042 Musa ÖNCÜ
- 365080 Melisa Eda HACIOĞLU
- 383241 Ersel Atakan KUMANDAŞ

SORUN: ARAÇLARDA YENİ LASTİKLERİN VERİMLİLİĞİ DÜŞÜK OLMASI

Yeni lastiklerde lastiğin yapısından dolayı yol tutuş ve buna bağlı olarak verimlilik düşük olmaktadır. Bundan dolayı yakıt israfı ve sürüş güvenlik zafiyeti oluşmaktadır. Bu riskler ortadan kaldırılmalıdır.

- TRIZ Yöntemine göre iyileştirilmek istenen yöntem: **Enerji kaybı**
- Bu yöntemi iyileştirirken olumsuz etkilenen durum: **Hareketli cismin alanı**

TRIZ yöntemine;

Dinamik hale getir

a) Cismin özelliğini, dış ortamı veya süreci, en uygun kullanım koşuluna göre değiştirin.

-Lastik dış yüzeyinin arttırılması

Taklit kullan

a) Kullanışsız, pahalı veya kırılğan bir cisim yerine basit ve ucuz kopyalarını kullanın

-Kullanıma daha uygun malzemeden üretilmiş lastikler ile kısa süre kullanılmasına rağmen yüksek verim ancak daha kısa kullanım ömrü

Boyut değiştir

a) Cismi iki veya üç boyutlu uzayda hareket ettirin

-Lastik dış yüzeyinin arttırılması

b) Cisimleri tek katlı yerine çok katlı olarak düzenleyin

Esnek kabuk veya ince film kullan

a) İnce filmler ve esnek kabuklar ile cismi dış çevreden yalıtın

-Daha ucuz ve kullanıma uygun malzeme ile lastiğin modifiye edilmesi

SONUÇ: Dinamik hale getir

Cismin özelliğini, dış ortamı veya süreci, en uygun kullanım koşuluna göre değiştirin.

-Lastik dış yüzeyinin arttırılması

Lastik dış yüzeyinin kullanıma göre değişmesini fabrikada üretilirken uygun hale getirilirse daha kullanıma uygun bir lastik üretimi gerçekleştirilmiş olur. Böylece yakıt israfı ve sürüş güvenlik zafiyeti ortadan kaldırılmış olur.

EK-3. Kısıtlar Formu

Kısıtlar	Açıklamalar
Tasarım Kısıtlamaları	Cihazın kullanım kalitesi açısından boyutu olabildiğince küçük ve tek parça olması gerekmektedir. Bu yüzden ona göre devre kartı ve cihaz kutusu üretilmesi gerekmektedir.
Prototip üretim kısıtlamaları	Sabit düğümler farklı cihazlardan dolayı enterferansa uğramaması için test edilmesi lazımdır. Çevredeki cihazlara bağlı olarak frekans değeri değiştirilmelidir.
Uygulama kısıtlamaları	Güç kullanımı ön planda tutularak geliştirilen projede yüksek sağlıklı yeterli kapasiteli piller kullanılacaktır. Li-İon piller tercih edildiğinden güvenlik tehlikelisi minimum seviyedir. Projede kullanılan cihaz kutusu anteni dışarı olacak şekilde su geçirmez olduğundan yağmur ve suya karşı güvenlik tehlikelisi oldukça azdır.
Toplumsal etki ve kısıtlamalar	Proje tamamlandığında hayvancılıkla uğraşan kişilerin sürülerini izlemesi mümkün olduğundan toplumsal açıdan olumlu bir etki oluşturur. Toplumsal ve sosyal yaşama yönelik doğrudan olumsuz bir etkisi tespit edilememiştir.
Sağlık kısıtlamaları	Proje tamamlandığında herhangi bir biyolojik unsura doğrudan çalışmada yer verilmediğinden dolayı doğrudan veya dolaylı olarak herhangi bir sağlık kısıtlaması tespit edilememiştir.
Çevresel etki ve kısıtlamaları	Cihaz içerisinde pil bulunduğundan cihaz kutusuna yüksek şiddetli darbe uygulanmamalıdır. Cihaz ömrünü tamamladığı zaman piller ve diğer donanımlar doğada bırakılmadan uygun şekilde geri dönüşüme atılmalıdır.
Güvenlik kısıtlamaları	Cihaz içerisinde pil bulunduğundan cihaz kutusuna yüksek şiddetli darbe uygulanmamalıdır. Küçük boyutlarda ve düşük güç seviyesinde çalışıldığı için güvenlik önlemine gerek yoktur.
Mali kısıtlamalar	Proje teori ve donanımlar geliştirme aşamasında olduğundan projeyi gerçekleştirmek için minimum sayıdaki donanım sayısına göre planlanmıştır. Proje geliştirilmesinin ilerleyen safhalarında projenin hata oranına göre donanım sayısı artırılabilir.
Hukuksal kısıtlamalar	Projenin imalatı ve uygulanması sırasında dikkate alınması gereken hukuki sorunlar açıklanmalıdır. Örneğin uyulması gereken yönetmeliklerin dikkate alınmaması durumunda oluşabilecek hukuki sorumluluklar açıklanabilir. Frekans Tahsisinden Muaf Telsiz Cihaz ve Sistemleri Hakkında Yönetmenlik, Resmi Gazete 30608 (27 Kasım 2018) Elektronik Haberleşme Kanunu, Resmi Gazete 27050 (10 Kasım 2008) Yukarıdaki yönetmeliklere uyulmaması halinde cihazları kullananlara hakkında ceza yaptırımı gerçekleştirilebilir.

EK-4. Özgeçmişler

Alipaşa Öksüz, 3 Ocak 1999'da Trabzon'da doğdu. Dedemin adı olan Alipaşa adı verildi. İlköğretimini Yomra Merkez İlkokulu& Ortaokulunda tamamladı. Sonrasında Akçaabat Fen Lisesinde lise öğretime devam edip Yomra Fen Lisesinde mezun oldu. Üniversite hayatına Karadeniz Teknik Üniversitesi %30 İngilizce Elektrik Elektronik Mühendisliği bölümünde devam etmektedir. Anadili Türkçe, yabancı dil olarak B2 seviye İngilizce bilmektedir. Teknik olarak Python, C, EagleCAD, SketchUp, Proteus'a hakimdir. 5 yıllık üniversite hayatında Elektrik Elektronik Mühendisliği Kulübü, Tenis Kulübü, Havacılık Kulübünde aktif olarak yer aldı. Şu an ise KTÜ Uzay Kulübü yönetim kurulu başkanlığını yürütmektedir. Aynı zamanda EMO Genç yönetim kurulu üyesidir. 2021 TEKNOFEST TURKSAT 6. Model Uydu Yarışmasında kaptanlığını yürüttüğü HUMA Model Uydu Takımı ile 3.lük derecesi elde etmiştir. Öğrenim hayatında üzerine yoğunlaştığı alanlar PCB, haberleşme sistemleri, uydular ve lisans bitirme projesiyle daha da üzerine düştüğü IoT ve lokalizasyon uygulamalarıdır.

Furkan Bilge, İstanbul'un Üsküdar ilçesinde doğdu. 2 yaşında ailesinin Giresun'a taşınmasıyla çocukluk ve eğitim hayatının bir kısmını burada geçirdi. Ailesinin Trabzona taşınmasıyla eğitim hayatını Ali Kemal Aktürk ilköğretim okulundan devam etti ve 2013 yılında mezun oldu. Aynı yıl Kanuni Anadolu Lisesini kazandı ve 2017 yılında sayısal bölümünden mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi Elektrik-Elektronik Mühendisliği bölümüne yerleşti. Bir yıl İngilizce hazırlık eğitimi aldıktan sonra 2018 yılında mühendislik eğitime başladı. Orta-üzeri İngilizce bilmektedir. Halen öğrenci olup 4.sınıfta okumaktadır

EK-5. Mühendislik Bitirme Teslim Koşulları Formu



1969'dan beri

Elektrik ve Elektronik Mühendisliği

<https://www.ktu.edu.tr/eee>

düşünden gerçeğine...

Bölümü

BİTİRME PROJESİ DOSYA TESLİM KOŞULLARI

ÖĞRENCİ BİLGİLERİ

TARİH: 02 / 06 / 2022

Numarası	Adı ve Soyadı	İmza
365041	Furkan BİLGE	
365036	Alipaşa ÖKSÜZ	

Projedeki öğrenci sayısına göre doldurulacak ve imzalanacaktır. Bu dosyayı imzalayıp teslim eden öğrenciler aşağıda sundukları bilgilerin doğruluğunu kabul etmiş demektir.

PROJE BİLGİLERİ

Proje Başlığı	:	RSSI Tabanlı Konum Tespit Cihazı
Proje Danışmanı	:	Prof. Dr. Temel KAYIKÇIOĞLU
		İmza:

PROJE DOSYASININ TESLİM EDİLEBİLMESİ İÇİN

Aşağıdaki yanıtların tümü EVET olmalıdır.

- Dosyanızda herhangi bir kaynaktan birebir alıntı yapılmış kısımlar %20'den AZ ve paragraf halinde bire bir alıntı sayısı sıfır mı? EVET ☒ Hayır ☐
- Dosyanızda önsöz, özet ve ana bölümler (Giriş, Teorik Altyapı, Tasarım, Simülasyon, Deneysel Çalışma, Sonuçlar, Değerlendirmeler) ve kaynaklar listesi var mı? EVET ☒ Hayır ☐
- Giriş bölümünde Giriş, Literatür Taraması, Özgün Değer, Yöntem, Yaygın Etki, Standartlar listesi ve Çalışma Takvimi var mı? İş paketlerinde liderlik paylaşımı ve B planları yapıldı mı? EVET ☒ Hayır ☐
- Tasarım bölümünün sonuna Mali analiz yorumu ve projeden kaynaklanabilecek hukuksal bir değerlendirme eklendi mi? EVET ☒ Hayır ☐
- Bütün denklemlere numara verildi, tüm şekillerde şekil alt yazısı ve tüm çizelgelerde üst başlık yazıldı mı? EVET ☒ Hayır ☐
- Bütün denklemlere şekillere, çizelgelere ve kaynaklara metin içerisinde atıf yapıldı mı? EVET ☒ Hayır ☐
- Tasarım, Simülasyon ve sonuçlar kısmındaki bütün şekiller size mi ait? EVET ☒ Hayır ☐
- Kaynaklar Bitirme Çalışması Yazım Klavuzuna uygun olarak sıralanmış ve 2 tanesi İngilizce olmak üzere en az 5 tane bilimsel makale, tez ve bildiri türü yayına atıfta bulunulmuşmu? EVET ☒ Hayır ☐
- IEEE Etik Kuralları (IEEE Code of Ethics) Ekler kısmına eklediniz mi? EVET ☒ Hayır ☐
- Disiplinler arası çalışma yaptınız ve bunu Ekler kısmında açıkladınız mı? EVET ☒ Hayır ☐

Proje dosyasının teslim edilebilmesi için yukarıdaki cevapların hepsinin EVET olması gerekir.

Bu formu Adobe Acrobat Reader kullanarak bilgisayarda doldurabilirsiniz.

EK-6. Turnitin Raporu

RSSI Tabanlı Konum Tespit Cihazı

ORJİNALLİK RAPORU

%8 BENZERLİK ENDEKSİ	%7 İNTERNET KAYNAKLARI	%3 YAYINLA R	%3 ÖĞRENCİ ÖDEVLERİ
-----------------------------------	-------------------------------------	---------------------------	-------------------------------

BİRİNCİL KAYNAKLAR

1	media.wiley.com İnternet Kaynağı	%2
2	www.emo.org.tr İnternet Kaynağı	%2
3	loranedir.com İnternet Kaynağı	%1
4	www.robotistan.com İnternet Kaynağı	<%1
5	akademi.robolinkmarket.com İnternet Kaynağı	<%1
6	Submitted to Karadeniz Teknik University Öğrenci Ödevi	<%1
7	Submitted to Escuela Politecnica Nacional Öğrenci Ödevi	<%1
8	www.hakankaya.kim İnternet Kaynağı	<%1
9	acikerisim.firat.edu.tr İnternet Kaynağı	<%1