GAZİ ÜNİVERSİTESİ
Gazi
1926

OTONOM SU ALTI ARAÇLARDA ROBOT İŞLETİM SİSTEMİ TABANLI YOL PLANLAMA

Emre GÖZÜTOK

YÜKSEK LİSANS TEZİ
BİLİŞİM SİSTEMLERİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

AĞUSTOS 2024

OTONOM SU ALTI ARAÇLARDA ROBOT İŞLETİM SİSTEMİ TABANLI YOL PLANLAMA

(Yüksek Lisans Tezi)

Emre GÖZÜTOK

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Ağustos 2024

ÖZET

Su altı ortamları, sınırlı veri erişimi ve fiziksel koşullarından dolayı otonom araç navigasyonu için büyük zorluklar sunar. Bu çalışma, uzun menzilli otonom denizaltı araçlarının (LRAUV'ler) rotalarının planlanmasında geleneksel kontrolcüler yerine Derin Q-Öğrenme (DQN) ve Çift Gecikmeli Derin Belirleyici Politika Gradienti (TD3) algoritmalarının uygulanmasını araştırmaktadır. Su altı aracının Gazebo simülasyon ortamındaki kontrolü Robot İşletim Sistemi (ROS) kullanılarak yönetilmektedir. Elde edilen sonuçlar, bu algoritmaların geleneksel kontrolcülere kıyasla birçok avantaj sağlayabileceğini göstermektedir. Çalışma hem DQN hem de TD3 algoritmalarının su altı araçlarının rotalarının planlanmasında etkin bir şekilde kullanılabileceğini ve eğitimlerin simülasyon ortamında etkili bir şekilde gerçekleştirilebileceğini vurgulamaktadır. Bu araştırmanın amacı, su altı araçları için pek çok geleneksel kontrol yöntemine kıyasla, pek çok öğrenmeye dayalı otonom rotalama algoritmalarının etkinliğini ve potansiyel yararlarını araştırmaktır. Çalışmada Gazebo simülatör üzerinde modellenen bir su altı simülatörü kullanılmıştır. Simülasyon tabanlı öğrenmeyi gerçekleştirmek amacı ile geliştirilen Robot İşletim Sistemi tabanlı ve simülasyon sistemi arasında bir iletişim mimarisi tasarlanmıştır. Simülasyon tabanlı öğrenmede aktif olarak konuşlandırılmış bir su altı aracı modeli kullanılmıştır. Bulgular, DQN ve TD3 yöntemlerinin su altı araçlarının rota planlamasında etkin olduğunu ve algoritma performansı üzerinde hiperparametrelerin kritik etkisini vurgulamaktadır. Sonuç olarak, DQN ve TD3 yöntemlerini kullanarak otonom su altı araçlarının rota planlamasının mümkün olduğu ve bu alanda gelecekte önemli ilerleme potansiyeli taşıdığı açıkça ortaya görülmüştür.

Bilim Kodu	:	92432
Anahtar Kelimeler	:	Robotik, otonom su altı araçlar, robot işletim sistemi simülasyon, yol planlama
Sayfa Adedi	:	53
Danışman	:	Doç. Dr. Fecir DURAN

PATH PLANNING IN UNDERWATER AUTONOMOUS VEHICLES BASED ON ROBOTIC OPERATING SYTEM

(M. Sc. Thesis)

Emre GÖZÜTOK

GAZİ UNIVERSITY

INSTITUTE OF INFORMATICS

August 2024

ABSTRACT

Underwater environments present significant challenges for autonomous vehicle navigation due to limited data access and physical conditions. This study investigates the application of Deep Q-Learning (DQN) and Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithms in the route planning of long-range autonomous underwater vehicles (LRAUVs) instead of traditional controllers. The control of the underwater vehicle in the Gazebo simulation environment is managed using the Robot Operating System (ROS). The results obtained indicate that these algorithms can provide many advantages compared to traditional controllers. The study emphasizes that both DQN and TD3 algorithms can be effectively used for route planning of underwater vehicles and that training can be conducted efficiently in a simulation environment. The aim of this research is to explore the effectiveness and potential benefits of many learning-based autonomous routing algorithms compared to traditional control methods for underwater vehicles. A underwater simulator modeled on the Gazebo simulator was used in the study. To achieve simulation-based learning, a communication architecture between the Robot Operating System-based simulation system and the simulator was designed. An actively deployed underwater vehicle model was used in simulation-based learning. The findings highlight that both DQN and TD3 methods are effective in route planning for underwater vehicles and underscore the critical impact of hyperparameters on algorithm performance. Consequently, it is evident that autonomous underwater vehicle route planning using DQN and TD3 methods is feasible and holds significant potential for future advancements in this field.

Science Code	:	92432
Key Words	:	Robotic, autonomous underwater vehicles, robot operating system, simulation, path planning
Page Number	:	53
Supervisor	:	Assoc. Prof. Dr. Doç. Dr. Fecir DURAN

TEŞEKKÜR

Bu çalışmanın tamamlanmasında sağladığı değerli destek ve rehberlik için danışman hocam Doç. Dr. Fecir Duran'a en içten teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xi

SİMGELER VE KISALTMALAR iv

1. GİRİŞ 1

2. OTONOM SU ALTI ARAÇLARINDA YOL PLANLAMA YÖNTEMLERİNİN İNCELENMESİ 5

2.1. Graf Tabanlı Yol Planlama 5

2.2 Bulanık Mantık Tabanlı Yol Planlama 6

2.3 Genetik Algoritma Tabanlı Yol Planlama 7

2.4 Parçacık Optimizasyonu Tabanlı Yol Planlama 8

2.5. Otonom Su Altı Yol Planlamasında Pekiştirmeli Öğrenme (Reinforcement Learning) Algoritması Tabanlı Yaklaşımlar 10

2.6. Otonom Su Altı Yol Planlamasında Hibrit Yaklaşımlar 11

3. MATERYAL VE METOD 13

3.1 Robot İşletim Sistemi (ROS) 13

3.1.1 ROS düğümleri (nodes) 13

3.1.2 ROS konuları (topics) 13

3.1.3 ROS Master 14

3.1.4 ROS-Gazebo Köprüsü 14

3.2. Gazebo 15

Sayfa

3.3 Tethys Su Altı Simülatörü 15
3.3.1 Tethys su altı aracı 16
3.3.2 Sensörler 16
3.3.3 Aktüatörler 17
3.4. Yapay Sinir Ağları 17
3.4.1 Regülarizasyon 18
3.4.2 Aktivasyon fonksiyonları 19
3.4.3 Batch öğrenmesi ve normalizasyon 20
3.4.4 Evrişimli sinir ağları 20
3.4.5 Kayıp fonksiyonu 21
3.5 Pekiştirmeli Öğrenme 22
3.5.1 Markov Karar Süreci 22
3.5.2 Bellman Denklemi 23
3.5.3 Zamansal Fark Öğrenmesi (TD Learning) 23
3.5.4 Sarsa 24
3.5.5 Q-Öğrenmesi 24
3.5.6 Çevrim içi öğrenme 25
3.5.6 Keşif stratejileri 25
3.6. Derin Q-Öğrenmesi 26
3.7. TD3 27
3.8 Geliştirilen Simülasyon Kontrol Mimarisi 28
3.9 Geliştirilen ROS İletişim Mimarisi 31
3.10 Geliştirilen Derin Q-Öğrenmesi Algoritması 32
3.11 Geliştirilen Derin TD3 Algoritması 34
4. DENEYSEL ÇALIŞMALAR 37
4.1. Kullanılan Donanım 37

Sayfa

4.2. Kullanılan Deney Düzeneği 37
4.2.1. Derin Q-Öğrenmesi ile Su Altı Yol Planlama Deneyi 40
4.2.2 TD3 ile Su Altı Yol Planlama Deneyi 43
5. SONUÇLAR VE TARTIŞMA 47
KAYNAKLAR 49
ÖZGEÇMİŞ 53

ÇİZELGELERİN LİSTESİ

Çizelge Sayfa

Çizelge 3.1. Tethys UMOSA Teknik Özellikleri 16

Çizelge 3.2. ROS-Gazebo Pekiştirme Öğrenmesi tabanlı kontrol 30

Çizelge 3.3. Derin Q-Öğrenmesi algoritması hiperparametre değerleri 33

Çizelge 3.4 TD3 algoritması hiperparametre değerleri 35

Çizelge 4.1 Simülasyon ortamı hidrodinamik parametreleri 38

Çizelge 4.2 Simülasyon ortamı fiziksel özellikleri 39

ŞEKİLLERİN LİSTESİ

Şekil Sayfa

Şekil 3.1. ROS master iletişim yapısı 14

Şekil 3.2. ROS - Gazebo köprüsü çalışma prensibi 14

Şekil 3.3. Tethys UMOSA modeli simülasyon görseli 15

Şekil 3.4. Genel YSA yapısı 17

Şekil 3.5. ROS tabanlı Pekiştirmeli Öğrenme ile otonom su altı yol planlama mimarisi 28

Şekil 3.6. ROS rqt_graph yazılım paketi ile görüntülenen iletişim grafiği 31

Şekil 3.7. Geliştirilen Derin Q-Öğrenmesi algoritması mimarisi 32

Şekil 3.8. Derin Q-Ağları YSA mimarisi 33

Şekil 3.9. Geliştirilen TD3 algoritması mimarisi 35

Şekil 3.10 Aktör ağı YSA mimarisi 36

Şekil 3.11 Kritik ağları YSA mimarisi 36

Şekil 4.1 Tethys su altı simülasyon ortamı 38

Şekil 4.2 Simülasyon senaryosu 39

Şekil 4.3. Takip edilen deney metotları 40

Şekil 4.4. Derin Q-Öğrenmesi deneyi ortalama durum zaman grafiği 41

Şekil 4.5. Derin Q-Öğrenmesi deneyi ödül, zaman grafiği 41

Şekil 4.6. Derin Q-Öğrenmesi deneyi batarya doluluk yüzdesi grafiği 42

Şekil 4.5. Derin Q-Öğrenmesi batarya şarj kapasitesi zamana bağlı değişim grafiği 42

Şekil 4.7. Derin Q-Öğrenmesi deneyi hedefe uzaklık zaman grafiği 42

Şekil 4.8. TD3 deneyi ortalama durum zaman grafiği 43

Şekil 4.9. Derin TD3 deneyi ödül, zaman grafiği 43

Şekil 4 10. TD3 deneyi batarya doluluk yüzdesi grafiği 44

Şekil 4.11. TD3 deneyi batarya doluluk yüzdesi grafiği 44

Şekil 4.12. TD3 deneyi hedefe uzaklık zaman grafiği 44

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
A	Amper
h	saat
hz	Hertz
kg	Kilogram
m	Metre
s	saniye
W	Watt

Kısaltmalar	**Açıklamalar**
DE	Diferansiyel Evrimleme)
DQN	Derin-Q Ağları (Deep Q-Networks)
GPP	Genetik Yol Planlayıcı (Genetic Path Planner)
IMU	Ataletsel Ölçüm Birimi
IPP	Bilgilendirici Yol Planlama (Informative Path Planning)
OSA	Otonom Su Altı Araçları (Autonomous Submarine Vehicles)
MAE	Ortalama Mutlak Hata
MSE	Ortalama Kare Hata
RF	Radyo Frekansı
ROS	Robot İşletim Sistemi (Robot Operating System)
TD3	Çift Gecikmiş Derin Belirleyici Politika Gradyanı
UMOSA	Uzun Menzilli Otonom Su Altı Aracı

1. GİRİŞ

Su altı tabanı, dünya okyanuslarının yalnızca %5'inin haritalandığı ve keşfedildiği göz önüne alındığında, çok az bilinmektedir ve büyük ölçüde belirsizdir. [1]. Otonom su altı araçları (OSA), denizaltı keşiflerinden arama-kurtarma operasyonlarına kadar geniş bir uygulama yelpazesinde kullanılmaktadır. Ancak su altı ortamının fiziksel ve çevresel özellikleri, bu araçların etkinliğini ve verimliliğini önemli ölçüde zorlaştırmaktadır. Geleneksel kontrol yöntemleri, denizaltı ortamlarının dinamik doğası nedeniyle sınırlı kalabilmektedir. Bu ortamda etkili bir şekilde görev yapabilmek için araçların karmaşık ve sürekli değişen koşullara uyum sağlayabilen gelişmiş kontrol sistemlerine ihtiyaçları vardır.

Su altı ortamında karşılaşılan zorlukların yanı sıra, mevcut algılayıcılar (sensors) sınırlamaları da önemli bir problem teşkil etmektedir. Genellikle kullanılan algılayıcılar arasında sonar, Ataletsel Ölçüm Birimi (IMU), basınç algılayıcıları, manyetometreler ve kamera sistemleri bulunmaktadır. Ancak bu algılayıcılar, su altındaki ışık koşulları, optik özellikler ve suyun iletkenliği gibi çevresel etmenlerden etkilenmektedir. Örneğin ışığın azalması ve suyun optik özellikleri kameraların veri toplama yeteneğini sınırlamaktadır. Ayrıca, RF tabanlı algılayıcılar suyun iletkenliği nedeniyle sınırlı menzil ve veri kalitesi sorunları yaşayabilmektedir.

Uzun menzilli su altı araçlarının kontrolü ise kinematik yapılarından dolayı daha zordur. Bu araçların uzun menzilli operasyonları, kontrol ve yönlendirme açısından karmaşıklık yaratmaktadır. Su altı ortamının 3 boyutlu yapısı da su altı araçlarının kontrolünü daha da zorlaştırmaktadır; bu üç boyutlu ortam, hareketlerin ve yönlendirmelerin yönetilmesini karmaşıklaştırmaktadır. Ayrıca, su tarafından uygulanan çeşitli fiziksel etkiler, araçların stabilitesini ve hareket kabiliyetini etkileyerek kontrolünü daha da güçleştirmektedir. Otonom kontrol sistemlerinin bu zorlukları aşabilmesi için daha gelişmiş ve dinamik adaptasyon yeteneklerine sahip olması gerekmektedir.

Tüm bu nedenlerden dolayı OSA’larda yol planlama problemi oldukça karmaşık bir hale gelmekte olup, su altı ortamının fiziksel ve çevresel zorlukları, yol planlamanın sadece klasik yöntemlerle çözülmesini güçleştirmektedir.

Bu tez çalışmasında, bu problemlerden yola çıkarak OSA'larda yol planlama problemi ele alınmıştır. Su altı ortamının karmaşıklığı göz önünde bulundurularak, düşük veri erişimine sahip olunan ortamlarda yol planlamanın gerçekleştirilmesi amaçlanmaktadır.

Su altında yol planlamanın gerçekleştirilmesi son zamanlarda literatürde daha sık işlenmeye başlanan bir konudur. Bu alanda yapılan araştırmalar, OSA'ların karmaşık ve dinamik su altı ortamlarında etkili bir şekilde yol alabilmesi için çeşitli yöntemleri incelemektedir [2-5].

Su altı araçlarının farklı çevresel koşullarda etkili bir yol planlaması sağlayabilmesi için Pekiştirmeli Öğrenme tabanlı bir yaklaşımın uygulanabilirliği araştırılmıştır. Bu kapsamda, Derin-Q Ağları (DQN) ve TD3 algoritmaları kullanılarak su altı araçlarının gerçek zamanlı olarak kontrol edilmesi ve yönlendirilmesi amaçlanmıştır.

Çalışmada, Players ve diğer araştırmacılar tarafından geliştirilen Tethys Uzun Menzilli Otonom Su Altı Aracı'nı (UMOSA) modelleyen ve su altı ortamını simüle eden Tethys simülatörü kullanılmıştır [6]. Geliştirilen algoritma simülasyon ortamında test edilmiş olup, algoritma ROS çerçevesinde geliştirilerek gerçek sistemlere uygulanabilirliğin artırılması amaçlanmıştır.

Çalışmada OSA'lar için geliştirilen algoritmaların maliyet etkin bir şekilde test edilebilmesi amacı ile simülasyon tabanlı bir yol planlama algoritmasının geliştirilmesi amaçlanmıştır. Bu tez çalışması, otonom su altı araçları (OSA) için yol planlama problemini ele alarak, bu alandaki literatürü ve yaklaşımları değerlendirip, çeşitli yenilikçi katkılar sunmayı hedeflemektedir. Çalışmanın ana katkıları aşağıda sıralanmıştır.

- Otonom Su Altı Araçları (OSA) için yol planlama problemini ele alarak, mevcut literatürdeki öne çıkan bilimsel çalışmaları incelemek ve analiz etmek,
- Simülasyon tabanlı öğrenme metodu geliştirerek, OSA'larda yol planlamanın maliyet etkin geliştirilmesi ve test edilmesini sağlamak,
- Pekiştirmeli öğrenme tabanlı algoritmaların OSA'larda yol planlama problemindeki etkinliğini araştırmak,
- Pekiştirmeli Öğrenme yöntemlerinin simülasyon tabanlı ortamlarda kullanılabilirliğini araştırmak,
- Kontrolü zor olan UMOSA modellerinde yol planlaması için kullanılabilecek bir yöntem önermek,

- Gerçek zamanlı simülasyon ortamlarında Robot İşletim Sistemi ile entegre olarak kullanılabilecek bir iletişim mimarisi geliştirmek,
- Gazebo simülasyon ortamında yapılan testlerde, geliştirilmiş pekiştirmeli öğrenme modelinin performansını iyileştirmek ve doğrulamaktır.

Bu tez çalışmasında Bölüm 2'de OSA'larda yol planlama için farklı yol planlama yöntemlere dair literatür taraması verilmiş, Bölüm 3'te çalışmada kullanılan materyaller ve metotlara yer verilerek gerçekleştirilen simülasyon iletişim mimarisi ve Pekiştirmeli Öğrenme algoritmaları detaylandırılmıştır. Bölüm 4, elde edilen deneysel sonuçların bir özetini vermektedir. Son olarak, Bölüm 5'te sonuçlar tartışılmış, elde edilen bulgular değerlendirilmiş ve gelecekte yapılabilecek çalışmalara dair önerilerde bulunulmuştur.

2. OTONOM SU ALTI ARAÇLARINDA YOL PLANLAMA YÖNTEMLERİNİN İNCELENMESİ

Su altı otonom araçlarında navigasyon ve yol planlama konuları ile ilgili farklı yol planlama algoritmalarını ve farklı otonom su altı aracı türünü konu alan çeşitli çalışmalar bulunmaktadır.

Su altı otonom araçlarında navigasyon ve yol planlama, derin deniz keşfi, bilimsel araştırmalar, kurtarma operasyonları ve çevresel izleme gibi birçok alanda kritik bir öneme sahip olduğu bilinmektedir. Bu araçlar, genellikle zorlu ve dinamik su altı görev yapmaktadır ve hedeflerine güvenli ve etkin bir şekilde ulaşmalarını sağlamak için doğru ve akıllıca planlanmış bir yol izlemeleri gerekmektedir.

Yol planlama yöntemleri, su altı otonom araçlarının başarılı bir şekilde görevlerini gerçekleştirmesinde kritik bir rol oynamaktadır. Bu yöntemler, farklı keşif görevlerine, operasyonel gereksinimlere ve ortam koşullarına uygun olarak çeşitlilik göstermektedir. Literatürde graf tabanlı algoritmalar, derin öğrenme teknikleri, genetik algoritmalar ve hibrit yaklaşımlar gibi çeşitli yöntemler, su altı araçlarının yol planlama yeteneklerini artırmak için kullanılmaktadır. Bu literatür taraması, su altı otonom araçlarında yol planlama yöntemlerini kapsamlı bir şekilde incelemeyi amaçlamaktadır.

2.1. Graf Tabanlı Yol Planlama

Carroll ve arkadaşları tarafından yürütülen çalışmada, Otonom Denizaltı Araçları (ODA'lar) için graf tabanlı bir yol planlama yöntemi geliştirilmiştir. Bu çalışmada, yol planlayıcı bir A* algoritması kullanarak ODA'nın rotasını belirlenmiştir. Kararlar, yol planlayıcı tarafından batimetri (deniz tabanı haritası), erişime kapalı bölgeler, engeller ve okyanus akıntısı gibi çeşitli veri tabanlarına dayanılarak yapılmıştır. Özellikle, ODA'nın güvenli bir şekilde seyahat edebileceği derinlik bilgisi, batimetri veri tabanı tarafından sağlanmıştır. Tehlikeli alanların belirlenmesi ve bu alanlardan uzak durulması, hariç tutma bölgeleri ve engeller veri tabanları tarafından gerçekleştirilmiştir. Ayrıca, ODA'nın seyahat rotası belirlenirken akıntının etkisi okyanus akıntısı veri tabanı tarafından dikkate alınmış ve uygun bir rota planlanmıştır [7].

Çalışmada, belirli durumlarda, özellikle yüzlerce küçük, yan yana düğümün (node) geçilmesi gerektiğinde, farklı hızlardaki yol bölümlerinin (segments) sayısının çok fazla olması nedeniyle, elde edilen avantajlar hesaplama süresi tarafından aşılmıştır. Bu durum, A* tabanlı yol planlama algoritmalarında hız değişikliklerinin kontrol edilmesi gerektiğini ve yol planlama sürecinin optimize edilmesi gerektiğini göstermektedir.

Li ve Zhang tarafından yürütülen çalışmada graf tabanlı yol planlama yöntemleri kullanılmıştır. Bu araştırmada, Otomatik Denizaltı Araçları (ODA'lar) için üç boyutlu çevre modeli oluşturulmuş ve grid yöntemine dayalı olarak Dijsktra ve A* algoritmaları kullanılarak üç boyutlu yol araması gerçekleştirilmiştir. Simülasyon sonuçları, her iki algoritmanın da arama süresinin kısa olduğunu göstermektedir, ancak A* algoritmasının düğüm sayısı ve yol uzunluğu açısından Dijsktra algoritmasından daha iyi olduğunu göstermektedir. Düğüm arama sayısını azaltmak ve arama yolunu daha da kısaltmak için çok yönlü bir A* algoritması tasarlanmıştır. AUV'ler için üç boyutlu yol planlamasının yüksek verimlilik ve gerçek zamanlı performans sergilediği gösterilmiştir [8].

2.2 Bulanık Mantık Tabanlı Yol Planlama

Fang ve ekibinin yürüttüğü çalışmada, otonom denizaltı araçları için engel kaçınma amacıyla kendini ayarlayan bulanık denetleyici kullanılarak basit bir arama tekniği önerilmiştir. Denizaltı aracının hidrodinamik katsayılarını belirlemek için Planar Hareket Mekanizması (PMM) testi kullanılmış ve bu veriler denetim sistemi için önemli veri girişleri olarak kullanılmıştır. Ardından, denizaltı araçlarının itişini komuta etmek için kendini ayarlayan bulanık denetleyici benimsenmiştir. Engelden kaçınma fonksiyonu, sualtı görüntü tespit yöntemiyle birlikte BK üçgen alt ürünü kullanılarak belirlenmiştir. Yapılan simülasyonlar, dört tür hareket kontrol faktörünün kendini ayarlayan kontrol, görünürlük, güvenlik ve akıntı etkisi- manevra performansını incelemek için kullanıldığını göstermektedir [9]. Çalışma, BK üçgen alt ürününe dayalı görüntü tespit tekniği ile birleştirilmiş kendini ayarlayan bulanık denetleyicinin, AUV'lerin engel kaçınması için etkili bir arama tekniği olduğunu ortaya koymaktadır.

Liu ve ekibinin gerçekleştirdiği çalışmada, otonom su altı araçlarının başlangıç konumu ile istenen rota arasındaki önemli sapmaların olduğu senaryolarda, bir yol takibi kontrol stratejisi geliştirilmiştir. Bu strateji ile UMOSA'ların mevcut konumu ile istenen rotadaki gelecekteki konumu dikkate alarak her zaman diliminde yerel bir istenen rota yeniden

oluşturur ve kontrol çabası bu yerel rota temel alınarak hesaplanır. Bu yaklaşım, özellikle büyük takip hatası girişi olduğunda ortaya çıkan ciddi titremeyi azaltmayı amaçlamaktadır. Simülasyonlar, yeni kontrol stratejisinin etkinliğini değerlendirmek için kullanılmıştır. Sonuçlar, bu yaklaşımın kontrol çıkışındaki titremeyi önemli ölçüde azalttığını ve daha pürüzsüz bir kontrol sinyali sağladığını göstermektedir. Ayrıca, stratejinin, büyük okyanus akıntıları ve ölçüm gürültüsü gibi dış etkenler altında bile tatmin edici performans sergilediği gözlemlenmiştir [10].

Bu çalışma, yüksek hassasiyet yerine daha uzun işletme sürelerini önceliklendiren uygulamalar için alternatif bir yaklaşım sunmaktadır. Ayrıca, UMOSA'ların başlangıçta sapma gösterdiği senaryolarda, kontrol çıkışında yaşanan titreme sorununu çözmek için bulanık mantık temelli bir kontrol stratejisinin etkinliğini göstermektedir.

2.3 Genetik Algoritma Tabanlı Yol Planlama

Geleneksel yol izleme algoritmalarındaki verimsizlik, Bresciani ve ekibi tarafından araştırmada ele alınmıştır. Kontrol edilen Otonom Su Altı Araçları (USOMA'lar) tarafından, genellikle bilgi toplama amaçlarıyla önceden belirlenmiş yollar takip edilir. Ancak, bu yolların genellikle bilgi içeriği ve enerji tüketimi açısından verimsiz olduğu görülmektedir. Bu problem, Bilgilendirici Yol Planlama (IPP) ile ele alınmış; toplanan bilgiyi maksimize eden bir yol tanımlanmıştır. Bu amaçla, Genetik Algoritma tabanlı bir IPP stratejisi olan Genetik Yol Planlayıcı (GPP) sunulmuştur. GPP, aynı anda toplanan bilgiyi ve incelenen alanın kapsamını maksimize etmeyi hedeflemektedir [11].

Önerilen yöntem, gerçek deniz verileri kullanılarak çevrimdışı olarak test edilmiştir ve Posidonia Oceanica (PO) izleme ve denetim uygulamaları için üç farklı coğrafi alanda değerlendirilmiştir. PO varlığı hakkında apriori bilgi, gerçek deniz verileri üzerinde eğitilmiş bir Gauss Süreci (GP) kullanılarak olasılıksal terimlerle modellenmiştir. Ardından, GP tahmini, ilgi alanlarında her bir konumun bilgi içeriğini elde etmek için kullanılmıştır [11]. Önerilen algoritmanın performansını değerlendirmek için Sıralı Bölümler (S-B) algoritması ve Rastgele planlayıcı dahil olmak üzere diğer iki IPP yaklaşımıyla karşılaştırmalı bir çalışma yapılmıştır.

Strateji, genetik işlemler kavramlarına dayanmakta olup incelenen alanın bilgilendiriciliği ve kapsamı açısından optimal bir yol oluşturmak için kullanılmıştır. Karşılaştırmalı analiz,

3 farklı alanda gerçekleştirilmiş ve test sonuçları, GPP'nin genellikle diğer IPP stratejilerine göre daha iyi performans gösterdiğini ortaya konmuştur.

Hao ve arkadaşları tarafından yapılan çalışmada, deniz altında üç boyutlu Otonom Denizaltı Araçlarının (OSA'ların) global yol planlama ve dinamik engel önleme performansını artırmak amacıyla bir yöntem önerilmektedir. Bu yeni yöntem, adaptif genetik algoritma (AGA) kullanarak OSA'ların etkili ve yüksek kaliteli global yollar oluşturmasını sağlamaktadır. Geleneksel genetik algoritma optimizasyonları ile birlikte, çarpışma tespit mekanizmasına dayalı bir yol optimizasyon stratejisi benimsenmiştir. Ayrıca, vektör yapay potansiyel alan yöntemi (VAPF) kullanılarak bir yerel engel önleme stratejisi geliştirilmiştir. Bu strateji, uzay vektör yöntemini kullanarak hesaplama yöntemini iyileştirmekte ve OSA'ların gerçek zamanlı olarak düşük maliyetle engellerden kaçınmasını sağlamaktadır [12].

Çalışmada, OSA'ların global yol planlama ve yerel engel önleme yöntemlerini iyileştirmek için AGA ve VAPF algoritmalarını benimseyerek, etkili bir yaklaşım sunulmaktadır. Simülasyon sonuçları, önerilen yöntemin OSA'ların su altında düşük maliyetle güvenli bir şekilde seyahat etmesini sağladığını göstermektedir. Ancak, mekanik kısıtların ve engellerin boyutunun ve şeklinin dikkate alınmaması gibi bazı eksiklikler bulunmaktadır. Bu nedenle, gelecekteki çalışmaların bu hususları ele alarak yöntemi daha da iyileştirmesi beklenmektedir.

2.4 Parçacık Optimizasyonu Tabanlı Yol Planlama

Lim ve arkadaşları tarafından yapılan çalışmada, otonom su altı aracı (OSA) yol planlama senaryosu, sert kısıtların ve yumuşak kısıtların kombinasyonuyla sınırlı bir optimizasyon problemi olarak sunulmaktadır. Yol planlayıcının amacı, önceden bilinen engeller ve 2D ve 3D'de düzensiz akıntılarla karakterize okyanus ortamında bir OSA'yı güvenli bir şekilde yönlendiren optimum yolu oluşturmaktır. Yol planlayıcı, diferansiyel evrimleme (DE) ile hibridleştirilmiş Quantum PSO (SDEQPSO) ve Uyarlamalı PSO (SDEAPSO) olmak üzere 2 farklı varyantı kullanılmıştır. Farklı kısıtlar kullanılarak yapılan yol planlayıcılarının performansı, çözüm kaliteleri, kararlılıkları ve hesaplama verimlilikleri açısından kapsamlı Monte Carlo simülasyonları ve Varyans Analizi (ANOVA) prosedürleri aracılığıyla analiz edilmiştir. Simülasyon sonuçlarına göre, sert sınırlama koşulları için sınırlandırılmış bir ayar

ile yumuşak engel önleme için SDEQPSO yol planlayıcısının diğer algoritmalardan daha yüksek verimlilikle pürüzsüz ve uygun bir OSA yol oluşturabildiği tespit edilmiştir [13].

Bu çalışma, farklı türde kısıtlama ayarlarında bir OSA yol planlayıcısının performansını değerlendirmektedir. Sert sınırlı engel önleme ile ayarlanmış SDEQPSO yol planlayıcısının, yüksek çözüm kalitesi ve hesaplama verimliliği ile en iyi performansı sunduğu görülmüştür. Sert sınırlı engel önleme ile yol planlayıcılarının önemli ölçüde daha yüksek hesaplama gereksinimi olduğu bulunmuştur. Bu nedenle, yol planlayıcının engel önleme ayarı için yumuşak kısıtlama ayarı önerilmekte olup, her yineleme sonunda sert sınırlı engel önleme ile yolun güvenliği ve geçerliliği garanti altına alınmaktadır. Önerilen yol planlayıcısı, REMUS 100 OSA için uygun ve güvenli bir yol oluşturmayı başarmıştır, bu da OSA dinamik modelinin simülasyonuyla doğrulanmıştır.

Simülasyon, minimum yetenekli yol planlayıcıyı temsil etmek için önceden bilinen bir çevre varsaymış olsa da bu planlayıcının gösterdiği yüksek hesaplama verimliliği sayesinde, dinamik ortamlarda yol yeniden planlama gibi hesaplama yoğun problemleri çözmek için uygun olabileceği düşünülmektedir. Gelecek çalışmalarda, bu stokastik algoritmanın gerçekçi operasyonel koşullara nasıl uyum sağlayabileceği araştırılacaktır.

Huang ve Jin tarafından yapılan çalışmada, 2D sualtı ortamında otonom denizaltı aracı (AUV) için hızlı yol planlama ve etkili engelden kaçınma sorunlarını çözmek amacıyla bir yol planlama algoritması önerilmektedir. Bu amaçla, bir takviye öğrenme mekanizması ve parçacık sürü optimizasyonu (PSO) mekanizmasını birleştiren RMPSO tabanlı bir yol planlama algoritması önerilmektedir. PSO algoritmasının yakınsama hızını ve adaptif yeteneğini artırmak için takviye öğrenme mekanizmasının geri bildirim mekanizması RMPSO algoritmasına entegre edilmiştir. Daha sonra, RMPSO, AUV için okyanus akıntılarının etkisini ortadan kaldırmak ve enerji tasarrufu sağlamak için hız sentezi yöntemini Bezier eğrisi ile bütünleştirmektedir. Son olarak, RMPSO kullanılarak yol hızla geliştirilir ve engeller etkili bir şekilde kaçınılır. Simülasyon ve deney sonuçları, önerilen yöntemin geleneksel yöntemlere göre üstünlüğünü göstermektedir [14].

Önerilen RMPSO yol planlama algoritması, MAKLINK yönsüz ağ grafiğinde engellerden etkili bir şekilde kaçınabilme yeteneğini göstermektedir. RMPSO ile çeşitli algoritmalar karşılaştırıldığında, önerilen algoritmanın yakınsama hızının çok daha hızlı olduğu ancak kararlılık açısından bazı dezavantajları olduğu gözlemlenmiştir. Gelecek araştırmalarda, RMPSO algoritmasının su altı arama, kurtarma ve araştırma alanlarında pratik

uygulamalarda kullanılabileceği belirtilmiştir. İstatistiksel analiz, RMPSO'nun üstünlüğünü kanıtlamak ve algoritmanın kararlılığını artırmak için kullanılabileceği, böylece su altı 3D yol planlama problemlerinin çözümüne daha etkili bir şekilde bulunabileceği belirtilmiştir.

2.5. Otonom Su Altı Yol Planlamasında Pekiştirmeli Öğrenme (Reinforcement Learning) Algoritması Tabanlı Yaklaşımlar

Ma ve meslektaşları tarafından yapılan çalışmada, otonom su altı araçlarının (AUVs) hassas navigasyonu için derin öğrenmeye dayalı bir adaptif navigasyon algoritması önerilmiştir. Özellikle mikroelektromekanik sistemlerin (MEMS) algılayıcılarının ölçüm sapması gibi faktörlerin konum belirleme doğruluğunu etkilediği bu algoritma tarafından belirtilmiştir. Bu sorunu çözmek için derin öğrenme yeni bir yöntem olarak kullanılmaktadır, ancak pratik uygulamalarda hesaplama döngüsü ve sağlamlık açısından bazı zorluklarla karşılaşılabilir. Zorlukların ele alınması ve doğru navigasyonun gerçekleştirilmesi amacıyla derin öğrenme ile adaptif bir navigasyon algoritması önerilmektedir. Algoritma, navigasyon sisteminin hata birikimini düzeltmek için derin öğrenme kullanır ve DVL ölçümünün başarısız olup olmadığını belirlemek için $\chi 2$ kuralı kullanmaktadır. Ayrıca, ölçüm varyansı ile aynı anda navigasyon bilgisini tahmin etmek için Değişken Bayes (VB) yöntemine dayalı adaptif bir filtre kullanılmaktadır. Deneyler, önerilen algoritmanın navigasyon performansını ve konum doğruluğunu önemli ölçüde artırdığını göstermektedir [15].

Liu ve ekibinin çalışmasında, otonom su altı araçları (OSA) için Deep Deterministic Policy Gradient (DDPG) tabanlı bir kontrolcü önerilmiştir. Önerilen algoritma ile, OSA'nın iç algılayıcıları tarafından ölçülen bilgiler kullanılarak görevlerin yerine getirilmesi için kontrol komutları sağlanmaktadır. Büyük karmaşık model gereksinimi olmaksızın, önerilen algoritma sadece OSA'nın bazı giriş parametrelerine ihtiyaç duyar ve OSA'nın uygulama gereksinimlerini karşılamak için bir kontrol stratejisi öğrenmektedir. Bu algoritma, kontrol hassasiyeti gereksinimlerini ve operasyonel kısıtlamaların etkisini göz önünde bulundurarak bir ödül fonksiyonu önererek geliştirilmiştir. DDPG algoritması temel alınarak geliştirilen bu kontrol sistemi, vektörlü pervaneli OSA'nın alt katman hareket kontrolünü gerçekleştirmek üzere tasarlanmıştır. Gerçek denizaltı uygulama ortamını simülasyon ortamıyla değiştirmenin maliyet tasarrufu sağladığı ve verimliliği artırdığı düşünülmektedir. Bu bağlamda, derin öğrenme yöntemini kullanarak OSA kontrol çalışmalarının uygulama alanını genişletme konusunda çalışmanın belirli iyileştirmeler elde ettiği belirtilmiştir [16].

Ancak, mevcut çalışmanın bazı sınırlamaları bulunmaktadır. Önerilen algoritma altında yapılan simülasyonlar ideal koşullar altında gerçekleştirilmiştir, bu nedenle önerilen yöntemin doğruluğunu ve uygulanabilirliğini doğrulamak için gerçek deneylerin tamamlanması gerektiği belirtilmiştir. Ayrıca, algılayıcılar, aktüatörler ve kontrolcüler arasındaki zaman gecikmesi belirsizliği gibi daha fazla etki faktörünün göz önüne alınması gerekmektedir.

2.6. Otonom Su Altı Yol Planlamasında Hibrit Yaklaşımlar

Tian ve arkadaşları tarafından yapılan araştırmada, otonom su altı araçlarının (AUV'ler) rotayı takip etme problemi çözümlemek için kurulmuş kinematik ve dinamik modellere dayanan bir bulanık denetleyici sunulmuştur. Bulanık denetleyicinin iyi performans sergilemesi için, bulanık denetleyici için çoklu optimizasyon performans göstergeleri olarak rota uzunluğu, düzgünlük ve çapraz iz pozisyon hatası seçilmiştir. Üyelik fonksiyonlarının parametrelerini belirlemek için parçacık sürü optimizasyonu (PSO) algoritması önerilmiştir. Önerilen algoritmanın performansını test etmek için düz çizgi, sinüs eğrisi, yarım ay şekli, Arşimed spirali ve pratik rotaları içeren farklı senaryolar sunulmuştur.

Ayrıca, PSO algoritması kullanılarak üyelik fonksiyonlarının parametrelerinin belirlenmesiyle elde edilen bulanık denetleyicinin performansının test edildiği çeşitli senaryolar sunulmuştur. Çalışmada, bulanık denetleyicinin çoklu performans endekslerinin optimizasyonu ile OSA rota takip etme probleminin çözümleyebileceğini gösterilmiştir. Sonuç olarak, çoklu performans endekslerinin optimizasyonu ile bulanık denetleyicinin hibrit olarak kullanılması ile etkin performans alınabileceği belirtilmiştir [17].

Jian ve arkadaşları tarafından yapılan araştırmada, OSA için geliştirilen bir planlayıcı ile büyük ölçekli bilinen statik bir ortamların yanı sıra, dinamik engellerin bulunduğu ortamlarda da çarpışma olmadan bir yol belirleyebilmesi amaçlanmıştır. Bu çalışmada, dinamik engellerin bulunabileceği ortamlarda OSA'lar için hem yerel iz hem de global yol planlamak için değiştirilmiş ve birleştirilmiş Dinamik Pencere Yaklaşımı (DWA) ve Hızlı Keşfetme Rastgele Ağaç (RRT*) yöntemleri kullanılmıştır. Önerilen hibrit yaklaşımda ortamda dinamik engellerin bulunması durumunda planlayıcı otomatik olarak çarpışma riskini değerlendirmekte ve gerekirse RRT* ile DWA arasında geçiş yapmaktadır. Daha sonra, çarpışma riski ortadan kalktıktan sonra planlayıcı tekrar global yol planlama için Hızlı Keşfetme Rastgele Ağaç yöntemine geçiş yapmaktadır. Bu şekilde, iki algoritma arasında

geçiş yaparak gerçek zamanlı hesaplama dengesi ve küresel olarak optimal çözüm elde edilmektedir. Önerilen hibrit planlama stratejisinin etkinliği simülasyon ortamında test edilmiştir [18].

Çalışmada OSA'lar için önerilen hibrit bir planlayıcı, otonom çarpışma önleme ve hedef nokta takibini gerçekleştirmektedir. Bu planlayıcı, global yol planlama algoritması RRT* ve yerel iz planlama algoritması DWA'yı birleştirir. RRT* ve DWA'nın birleşimi, küresel optimum ve hesaplama maliyeti arasında denge sağlayabilir. Bu sayede, AUV, gerçek zamanlı yerel iz planlama ile dinamik engellerden kaçınabileceği ve aynı zamanda global yol planlama ile nispeten optimal bir yolda hedef noktaya ulaşabileceği gösterilmiştir. RRT* yönteminin verimliliğinin iyileştirilebileceği ve belirli bir frekansta sürekli planlama için uygun hale getirilebileceği belirtilmiştir.

3. MATERYAL VE METOD

Bu bölümde tez çalışması kapsamında oluşturulan simülasyon senaryosu ve simülasyon kontrol mimarisi açıklanmış ve geliştirilen Pekiştirmeli Öğrenme yöntemlerinin detayları anlatılmıştır.

3.1 Robot İşletim Sistemi (ROS)

ROS, robotik sistemlerin geliştirilmesi ve test edilmesi için esnek ve genişletilebilir bir çerçeve sağlayan bir açık kaynak robotik yazılım (middleware) platformudur. Robotik yazılımların geliştirilmesinde, donanımın soyutlanması, düşük seviyeli cihaz kontrolü ve süreçler arası mesaj iletimi gibi çeşitli servislerin ROS tarafından sağlandığı belirtilmiştir [19]. ROS çalışma prensibinin temelinde düğümler (topics), servisler (services) ve mesajlar bulunmaktadır.

Yapılan çalışmada, su altı aracı kontrol algoritmasının geliştirilmesinde açık kaynaklı ROS2 Humble sürümü kullanılmıştır.

3.1.1 ROS düğümleri (nodes)

ROS içerisinde, belirli işlevlerin gerçekleştirilmesi düğümler tarafından bağımsız programlar olarak yapılmaktadır. Her düğüm, belirli bir görevi yerine getirmek için tasarlanmıştır.

Düğümler arasında iletişim kurulmasında, ROS tarafından sağlanan konular (topics) ve servisler (services) gibi mekanizmalar kullanılmaktadır. Bu sayede, her bir düğüm diğer düğümlerin varlığından habersiz çalışabilir [20]. ROS düğümlerinin bağımsız olarak çalışabilmesi sistem modülerliğini ve esnekliğini artırmaktadır. Bu sayede ROS robotik sistemlerin dinamik ve değişken ortamlarda etkili bir şekilde çalışabilmesini sağlamaktadır.

3.1.2 ROS konuları (topics)

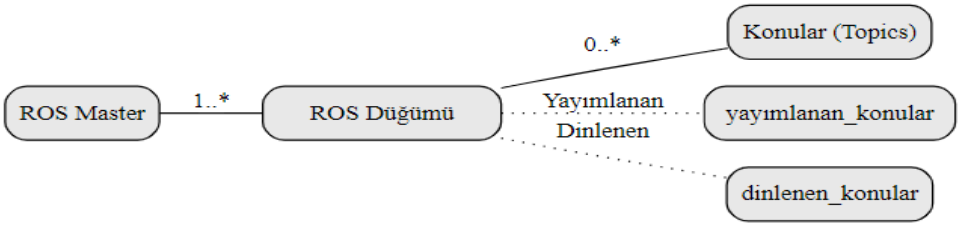
Konular ROS iletişiminin temel unsurlarından biridir. Konular, düğümler arasında asenkron veri alışverişi için yayınla/abone ol modelinde çalışırlar. Her konu iletilen verinin yapısını

tanımlayan belirli bir mesaj türü ile ilişkilidir [21]. Düğümler, bir konuya mesaj gönderebilir veya bir konudan mesaj almak üzere abone olabilir.

3.1.3 ROS Master

ROS Master, ROS'un merkezi bileşenlerinden biridir ve ROS ağındaki iletişim altyapısını yönetir. ROS Master düğümünün diğer ROS düğümleri ile iletişim yapısı Şekil 3.1.'de görüldüğü gibidir.

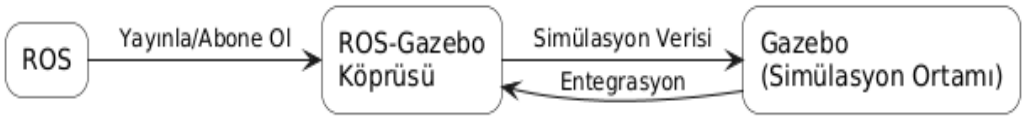
ROS Master, düğümlerin birbiri ile iletişim kurmasını ve veri alışverişinde bulunmasını sağlar. Düğümler, ihtiyaç duydukları diğer düğümleri ROS Master aracılığıyla keşfederler.



Şekil 3.1. ROS master iletişim yapısı

3.1.4 ROS-Gazebo Köprüsü

ROS-Gazebo köprüsü, ROS ile Gazebo simülasyon ortamı arasında sağlanan entegrasyon ve iletişim mekanizmasıdır. Bu köprü, Gazebo ortamında geliştirilen simülasyon yazılımlarının ROS tabanlı yazılımlar ile doğrudan entegrasyonunu mümkün kılmaktadır. ROS – Gazebo Köprüsü çalışma prensibi Şekil 3.2.'de görüldüğü gibidir. ROS-Gazebo köprüsü ile ROS, düğümler arası iletişimde kullanılan konu ve servis yapılarını Gazebo ortamına taşımaktadır [22]. Bu sayede, simülasyon içindeki farklı bileşenler arasında ROS standardıyla iletişim kurulabilir.



Şekil 3.2. ROS - Gazebo köprüsü çalışma prensibi

3.2. Gazebo

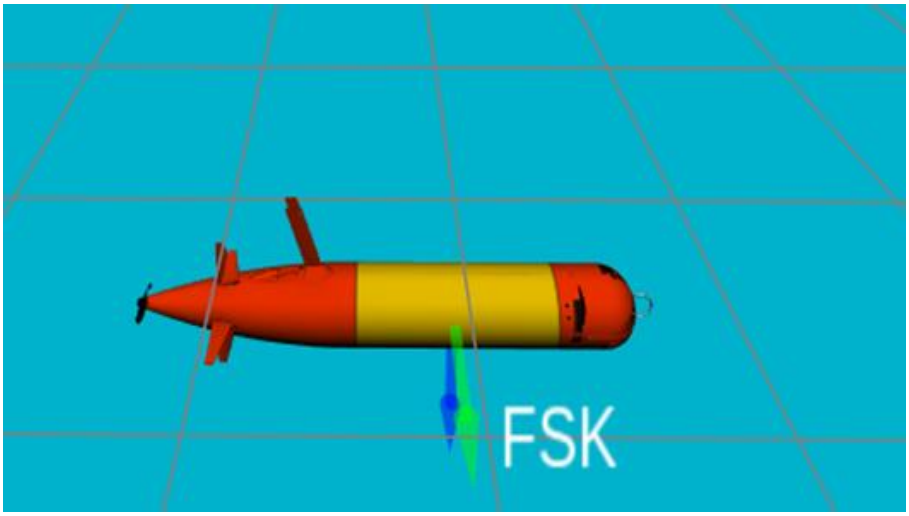
Gazebo robotik sistemlerin gerçekçi 3-boyutlu çevrelerde modellenmesini ve test edilebilmesini sağlayan fizik tabanlı bir simülasyon platformudur [22].

Gazebo, algılayıcıların, aktüatörlerin ve karmaşık robotik mekanizmaların simülasyonunu destekleyerek, robotik alanında geliştirilen algoritmaları ve bu algoritmaların davranışlarını fiziksel robotlara uygulamadan önce doğrulamalarını sağlamaktadır [22].

Yapılan çalışmada su altı ortamının simüle edilmesi için açık kod kaynaklı Gazebo Garden sürümü kullanılmıştır.

3.3 Tethys Su Altı Simülatörü

Çalışmada Monterey Araştırma Enstitüsü tarafından geliştirilen Tethys Su Altı Simülatörü kullanılmıştır [23]. Tethys Su Altı Simülatörü, Gazebo uygulaması içerisinde geliştirilmiş üç boyutlu bir su altı simülatörüdür [6]. Simülatör içerisinde su altı ortamının simüle edilebilmesi için yapılandırılmış edilmiş bir su altı simülasyon ortamı ve Tethys UMOSA modeli bulunmaktadır. Simülasyon ortamındaki Tethys su altı aracının ait görsel Şekil 3.3'te görüldüğü gibidir.



Şekil 3.3. Tethys UMOSA modeli simülasyon görseli

3.3.1 Tethys su altı aracı

Simülasyonda kullanılan Tethys modeli, Monterey Araştırma Enstitüsü tarafından geliştirilmiş Tethys sınıfı bir UMOSA'dır. Tethys UMOSA, oşinografik araştırmalar, deniz tabanı haritalama, çevre izleme, istihbarat, keşif ve denizaltı savunma harbi gibi çeşitli kullanım alanlarına sahiptir.

Çizelge 3.1. Tethys UMOSA Teknik Özellikleri [23]

Özellik	**Açıklama**
Uzunluk	2,3 m
Çap	0.30 m
Ağırlık	110 kg
Dalma Derinliği	1500 m
Menzil	1000 – 2000 m (güç moduna bağlı olarak)
Güç Tüketimi (Yüksek Güç Modu)	~ 8 W
İletişim	Uydu bağlantısı

3.3.2 Sensörler

Çalışmada su altı ortamındaki çevresel nicelikleri ölçümleyebilmek için çeşitli algılayıcılar kullanılmıştır. Kullanılan algılayıcılar şu şekildedir:

- Manyetometre Algılayıcısı: Manyetik alanın ölçümü için kullanılan bir cihazdır. Tethys UMOSA modelinde, bu algılayıcı çevresel manyetik alanları ölçmek ve konum belirlemek amacıyla kullanılır.
- IMU Algılayıcısı: hız, ivme ve açısal hız gibi hareket verilerini ölçmek amacıyla kullanılmaktadır. Tethys UMOSA modelinde IMU, denge sağlama ve yön tayini için kullanılır.
- Navigasyon Uydu Algılayıcısı: uydu sinyallerini alarak konum belirleme ve navigasyon amaçları için kullanılır. Tethys UMOSA modelinde bu algılayıcı, hassas konum belirleme ve rota takibi için kritik öneme sahiptir.
- Batarya Algılayıcısı: Batarya gerilimi, akımı ve sıcaklığını ölçmek için tasarlanmıştır. Tethys UMOSA modelinde bu algılayıcı, enerji tüketimini izlemek ve pil güvenilirliğini sağlamak amacıyla kullanılmaktadır.

3.3.3 Aktüatörler

Çalışmada Tethys su altı aracında bulunan aktüatörler kullanılarak aracın kontrolü hedeflenmiştir. Kullanılan aktüatörler şu şekildedir:

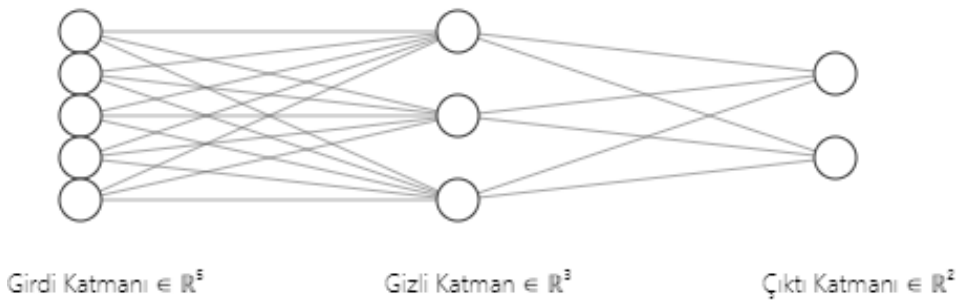
- İtici: Su altında hareket amacıyla kullanılır. İlerleme ve manevra yeteneklerini sağlar.
- Yatay Kanatlar: Yüzeyde dengeyi sağlamak ve yönlendirmek için kullanılır. Ayrıca dalış ve açılma işlemlerini kontrol etmek amacıyla kullanılır.
- Dikey Kanatlar: Dikey hareketi sağlamak ve dalış/açılma işlemlerini kontrol etmek için kullanılır. Derinlik kontrolü ve manevra yeteneklerini artırmak için kullanılır.
- Kaldırma Kuvveti Motoru: Sualtı yüzeyinde kalma ve yüzmeyi sağlamak için kullanılır. Ayrıca dalış ve açılma işlemlerini kontrol etmek ve su üstünde kalma yeteneğini sağlamak amacıyla kullanılır.

3.4. Yapay Sinir Ağları

Yapay Sinir Ağları (YSA), insan beyninin işleyişinden esinlenerek geliştirilen ve birçok karmaşık problemin çözümünde kullanılan bir yapay zeka modelidir. Bu model, birçok yapay sinir hücresinden oluşan bir ağ yapısına dayanmaktadır.

Yapay sinir ağları genellikle üç tip katmandan oluşur: giriş katmanı, gizli katman(lar) ve çıkış katmanı. Giriş katmanı, dış dünyadan gelen verilerin ağa girişini sağlamaktadır. Gizli katman(lar), bu girdileri işler ve ağın öğrenmesini sağlamaktadır. Son olarak, çıkış katmanı, ağın sonucunu üretir ve istenilen çıktıyı sağlamaktadır.

YSA'ların genel yapısı Şekil 3.4.'te ifade edilmiştir. Her bir yapay sinir hücresi, girdileri alır, onları ağırlıklarla çarpar, bir aktivasyon fonksiyonuna sokar ve çıktıyı üretir. Bu işlem, ağın içindeki her katmanda tekrarlanır ve ağ, verilen bir girdiye bağlı olarak belirli bir çıktı üretir.



Şekil 3.4. Genel YSA yapısı

YSA işleyişi içerisinde elde edilen modeller, veriye dayalı olarak kendilerini ayarlar ve belirli bir görevi yerine getirmek için optimize edilirler. YSA'ların öğrenme süreci, gerçek veriyle modelin tahmini arasındaki farkın minimize edilmesini içermektedir [23].

Yapay sinir ağlarının öğrenme süreci, gerçek veri ile modelin tahmini arasındaki farkın minimize edilmesini içermektedir.

3.4.1 Regülarizasyon

Yapay sinir ağlarında, aşırı uyum (overfitting) riskini azaltmak için kullanılan bir teknik olan regülarizasyon, modelin karmaşıklığını kontrol etmeyi amaçlamaktadır. Aşırı uyum, modelin eğitim verilerine çok iyi uyması ancak genelleme yeteneğinin düşük olması durumunu ifade etmektedir. Regülarizasyon, genellikle modelin parametrelerinin büyüklüğünü sınırlayarak veya modelin karmaşıklığını azaltarak aşırı uyumu önlemeye çalışmaktadır [24].

Genellikle L2 veya L1 düzenlemesi kullanılarak yapılan regülarizasyon, kayıp fonksiyonuna bir ceza terimi eklenerek gerçekleştirilir. Bu ceza terimi, modelin parametrelerinin büyüklüğünü kontrol eder ve aşırı uyumu azaltır. Genel olarak, regülarizasyon formülü Eş. 3.1'de ifade edilmiştir:

$$\mathcal{L}(\theta) = L\left(\theta, Y, \hat{Y}\right) + \lambda\Omega(\theta) \tag{3.1}$$

Regülarizasyon ağın eğitim sürecinde kullanılırken, orijinal kayıp fonksiyonuna regülarizasyon terimini ekleyerek, modelin genelleme yeteneğini artırmak ve aşırı uyumu önlemek için kullanılır. Regülarizasyon, modelin aşırı uyumu azaltarak daha iyi genelleme yapmasını sağlar ve genellikle eğitim veri setinin ötesinde daha iyi performans göstermesini sağlamaktadır.

L2 düzenlemesi, modelin parametrelerinin büyüklüğünü sınırlamak için kullanılan bir yöntemdir. Bu yöntem, kayıp fonksiyonuna bir ceza ekleyerek büyük parametre değerlerini engellemektedir. L2 düzenlemesi, genellikle ağın tüm parametrelerinin karesinin toplamıyla Eş. 3.2'deki gibi hesaplanmaktadır.

$$J_{\text{regularized}}(\theta) = J(\theta) + \lambda \sum_{i=1}^{n} \theta_i^2 \text{ (3.2)} \tag{3.2}$$

L1 düzenlemesi, L2 düzenlemesi gibi parametrelerin büyüklüğünü sınırlamak için kullanılmakta, ancak farklı bir ceza fonksiyonu kullanılmaktadır. L1 düzenlemesi, Eş. 3.3'te olduğu gibi parametrelerin mutlak değerlerinin toplamıyla hesaplanmaktadır.

$$J_{\text{regularized}}(\theta) = J(\theta) + \lambda \sum_{i=1}^{n} |\theta_i| \tag{3.3}$$

3.4.2 Aktivasyon fonksiyonları

Sinir hücrelerinin çıkışını belirleyen aktivasyon fonksiyonları, sinir ağlarının matematiksel modelinde önemli bir rol oynamaktadır. Her bir sinir hücresi, girdi özelliklerini ağırlıklarla çarparak bir çıktı üretir [25]. Bu çıktı, bir aktivasyon fonksiyonuna geçirilerek son çıkış elde edilir. Genellikle, sinir hücresinin çıkışı Eş. 3.4'teki şekilde ifade edilir.

$$a = \text{activation}\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{3.4}$$

Sigmoid fonksiyonu, giriş değerini 0 ile 1 arasında bir değere dönüştüren sürekli ve türevlenebilir bir fonksiyondur. Bu fonksiyon genellikle aralıklı veya olasılık değerlerini tahmin etmek için kullanılır. Sigmoid fonksiyonu Eş. 3.5'teki gibi tanımlanır.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.5}$$

ReLU fonksiyonu, giriş değeri negatifse 0, pozitifse giriş değerini koruyan basit bir doğrusal fonksiyondur. ReLU fonksiyonu matematiksel olarak Eş. 3.6'daki şekilde ifade edilir.

$$\text{ReLU}(x) = \max(0, x) \tag{3.6}$$

Tanh fonksiyonu, sigmoid fonksiyonuna benzer ancak çıkış aralığı -1 ile 1 arasındadır. Bu fonksiyon, simetrik bir çıkış aralığı sağlar ve ağın daha dengeli bir eğitim almasını sağlar. Matematiksel olarak tanh fonksiyonu şu Eş. 3.7'deki gibidir.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.7}$$

3.4.3 Batch öğrenmesi ve normalizasyon

Batch öğrenmesi, büyük veri kümelerini daha küçük parçalara (batch'ler) bölerek yapay sinir ağlarını eğitme yöntemidir. Her bir batch, belirli sayıda eğitim örneğinden oluşur ve modelin ağırlıkları her batch işlendikten sonra güncellenir. Bu yöntem, ağın bellek ve hesaplama kaynaklarını daha verimli kullanmasına olanak tanır [26]. Örneğin, bir batch içindeki m sayıda eğitim örneği için, ağırlık güncellemesi Eş. 3.8'deki gibi yapılır:

$$\theta = \theta - \eta \cdot \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta J\left(\theta; x^{(i)}, y^{(i)}\right) \tag{3.8}$$

Normalizasyon, yapay sinir ağlarında eğitim verilerinin dağılımını dengeler ve ağın daha hızlı ve güvenilir bir şekilde öğrenmesine yardımcı olmaktadır. Bu işlem, verilerin ortalamasını sıfır ve standart sapmasını bir yaparak gerçekleştirilir.

$$\widehat{x_i} = \frac{x_i - \mu_i}{\sigma_i} \tag{3.9}$$

3.4.4 Evrişimli sinir ağları

Evrişimsel Sinir Ağları, verileri katmanlar halinde işleyerek ve her katmanda belirli özellikleri öğrenerek çalışmaktadır. Bu ağlar, tamamen bağlı (fully connected) katmanlardan oluşan geleneksel yapay sinir ağlarından farklı olarak, evrişim (convolution) ve havuzlama (pooling) katmanlarını içermektedir [27]. Bu yapı, Evrişimsel Sinir Ağlarının çok çeşitli veri türlerini, özellikle de yüksek boyutlu verileri (örneğin, zaman serileri, sinyaller) etkili bir şekilde işlemesini sağlamaktadır.

Evrişim katmanı, giriş verisine uygulanan filtreler (kernels) ile çalışmaktadır. Her filtre, belirli bir özellik haritası (feature map) üretmektedir. Matematiksel olarak, evrişim işlemi Eş. 3.10'daki şekilde ifade edilir.

$$(f * g)(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} f(i, j) \cdot g(x - i, y - j) \tag{3.10}$$

Havuzlama katmanı, özellik haritalarının boyutunu azaltarak modelin hesaplama yükünü ve parametre sayısını düşürmektedir. En yaygın kullanılan havuzlama türü maksimum havuzlama (max pooling) olup, belirli bir bölgede en yüksek değeri seçmektedir [28].

Matematiksel olarak maksimum havuzlama işlemi Eş. 3.11'deki gibi ifade edilmektedir. Havuzlama, önemli özelliklerin korunarak veri boyutunun küçültülmesini sağlamaktadır.

$$h(x,y) = \max_{i,j \in R} f\,(x+i, y+j) \tag{3.11}$$

3.4.5 Kayıp fonksiyonu

Kayıp fonksiyonu (loss function), yapay sinir ağlarının eğitiminde modelin performansını değerlendirmek ve iyileştirmek için kullanılır. Kayıp fonksiyonu, modelin tahminleri ile gerçek değerler arasındaki farkı hesaplayarak, bu farkı minimize etmek amacıyla modelin ağırlıklarını güncellemektedir. Bu süreç, modelin doğruluğunu ve genelleme kabiliyetini artırmayı hedeflemektedir [29].

En yaygın kullanılan kayıp fonksiyonlarından biri olan ortalama kare hatası, modelin tahminleri ile gerçek değerler arasındaki ortalama kare farkını hesaplamaktadır. MSE, özellikle regresyon problemlerinde yaygın olarak kullanılır. MSE Eş. 3.12'deki gibi ifade edilmektedir.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{3.12}$$

Ortalama Mutlak Hata (MAE), regresyon modellerinin performansını değerlendirmek için kullanılan temel bir kayıp fonksiyonudur. Ortalama Mutlak Hata, modelin tahminleri ile gerçek değerler arasındaki ortalama mutlak farkı hesaplamaktadır.

Bu kayıp fonksiyonu, her bir tahmin hatasının mutlak değeri alınarak hesaplandığı için, hataların büyüklüğüne karşı duyarlı olmakta ve uç değerlere karşı daha dayanıklı hale gelmektedir.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \tag{3.13}$$

Huber kaybı (Huber loss), ortalama kare hatası ve ortalama mutlak hata fonksiyonlarının avantajlarını birleştiren bir kayıp fonksiyonudur. Huber kaybı, özellikle veri kümesinde hem küçük hem de büyük hataların bulunduğu durumlarda daha dengeli ve güvenilir sonuçlar elde etmek için kullanılır. Küçük hatalar için MSE gibi davranırken, büyük hatalar için MAE gibi davranarak, her iki fonksiyonun da dezavantajlarını minimize etmektedir [30].

Kayıp fonksiyonları, modelin öğrenme sürecinde merkezi bir rol oynar. Modelin performansını optimize etmek için, kayıp fonksiyonu minimize edilir. Bu süreçte, yaygın olarak kullanılan optimizasyon algoritmalarından biri olan gradyan inişi (gradient descent) algoritması, kayıp fonksiyonunun türevini kullanarak modelin ağırlıklarını güncellemektedir.

Gradyan inişi algoritmasının güncelleme kuralı Eş 3.14'teki şekilde hesaplanır.

$$\theta = \theta - \eta \nabla_{\theta} J(\theta) \tag{3.14}$$

3.5 Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme, bir ajanın bir hedefe ulaşmak için çevresindeki en iyi stratejileri öğrenmesini sağlayan bir yaklaşımdır. Bu süreç, ajanın aldığı ödüller ve cezalar doğrultusunda davranışlarını optimize etmesini içermektedir. Temel olarak, ajanın çevresiyle olan etkileşimleri matematiksel olarak modellenir. Bu modelleme, ajanın öğrenme sürecinde çevresine yönelik stratejilerini geliştirmesine yardımcı olmaktadır.

3.5.1 Markov Karar Süreci

Markov karar süreci, pekiştirmeli öğrenmede kullanılan bir matematiksel çerçeve olarak bilinmektedir. Markov karar süreci bir ajan ve çevre arasındaki etkileşimi açıklamak ve modellemek amacı ile kullanılmakta olup çeşitli bileşenlerden oluşmaktadır:

- Durum: Bir ajanın içinde bulunduğu çevrenin anlık durumunu temsil eder. Durum ajanın mevcut ortamda incelediği ve kararlarını temel aldığı bilgileri içermektedir.
- Aksiyon: Bir ajanın içerisinde bulunduğu çevrede gerçekleştirebileceği bir eylemi tanımlamaktadır. Yapılan aksiyon sonucunda ajan içerisinde bulunduğu çevre ile etkileşime girer ve çevrenin durumu değiştirebilir.
- Geçiş olasılığı: Geçiş olasılığı ajanın bir durumdan bir diğer duruma geçiş olasılığını ifade etmektedir.
- Ödül: Ajanın içinde bulunduğu durumda aldığı aksiyon için çevreden aldığı geri bildirim değeri olarak ifade edilebilir. Her bir durum, aksiyon çifti için verilen geri bildirim ile ajanın içinde bulunduğu güncel durum için hangi aksiyonu seçeceğine dair etkide bulunur.
- İndirim Faktörü: İndirim faktörü belirli bir durum aksiyon çifti için alınan çevreden alınan ödülün önemine atıfta bulunur. İndirim faktörü, ajanın gelecekteki ödüller yerine anlık ödülleri tercih etmesini sağlamaktadır [31].

3.5.2 Bellman Denklemi

Bellman denklemi, dinamik programlama ve pekiştirmeli öğrenme alanlarında merkezi bir rol oynamaktadır. Bu denklem, optimal politika ve değer fonksiyonlarını hesaplamak için kullanılır. Bellman denklemi, gelecekteki ödülleri maksimize eden kararların nasıl alınacağını belirlemektedir [32].

Bellman denklemi, Markov Karar Süreçleri çerçevesinde optimal kontrol sorunlarını çözmek için kullanılır. Her durum için optimal değer fonksiyonunu hesaplayarak, her durum için hangi eylemin alınması gerektiğini belirlemektedir. Bellman denklemi Eş. 3.15'teki şekilde ifade edilmektedir.

$$V^*(s) = \max_a(R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^*(s')) \quad (3.15)$$

Denklem, bir durumun optimal değerini, o durumdan başlayarak tüm olası eylemleri göz önünde bulundurarak hesaplamaktadır. Bir durumun optimal değeri, o durumdan alınabilecek tüm eylemlerin değerlerinin maksimumu olarak tanımlanır. Bu, Bellman denkleminin öz yinelemeli yapısının temelini oluşturmaktadır [33].

Denklemde yer alan ödül fonksiyonu ise, bir durumda belirli bir eylemin gerçekleştirildiğinde alınan anlık ödülü ifade etmektedir ve Bellman denkleminde, durumların optimal değerlerinin hesaplanmasında önemli bir rol oynamaktadır.

3.5.3 Zamansal Fark Öğrenmesi (TD Learning)

Zamansal Fark Öğrenmesi temel pekiştirmeli öğrenme yöntemlerinden biridir. Zamansal Fark Öğrenmesi, dinamik öğrenme ve Monte Carlo yöntemlerinin bir kombinasyonu olarak değerlendirilmektedir.

Monte Carlo yöntemlerinden farklı olarak Zamansal Fark Öğrenmesi yönteminde tahminler bölüm sonunda alınan toplam ödül (G) ile değil, bir sonraki duruma bağlı olarak güncellenmektedir. Bu ise, Zamansal Fark algoritmasının bölüm sonunu beklemeden dinamik ortamlarda tahminde bulunabilmesini sağlamaktadır [34].

Zamansal Fark öğrenmesi algoritmasının temelinde belirli bir zaman için, o ana bağlı alınan ödül ve durum tahminine göre elde edilen zamansal farkın elde edilmesi yatmaktadır. Zamansal fark algoritması, Eş. 3.16'daki şekilde hesaplanmaktadır. Zamansal Fark

Öğrenmesi algoritmasında öncelikle zamansal fark hatası elde edilir ve ajan tahminini çevreden gelen ödüller ve bir sonraki durum tahminine göre güncellemektedir.

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{3.16}$$

Elde edilen zamansal fark, öğrenme oranına bağlı olarak değer fonksiyonunun güncellenmesinde kullanılmaktadır. Zamansal Fark Öğrenmesi algoritması için değer fonksiyonunun yenilenmesi Eş. 3.17'daki şekilde ifade edilmektedir.

$$V(s_t) \leftarrow V(s_t) + \alpha \cdot [r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)] \tag{3.17}$$

Zamansal Fark Öğrenmesi, temel bir pekiştirmeli öğrenme algoritması olup, çeşitli pekiştirmeli öğrenme algoritmaları bu sınıfta değerlendirilmektedir. Q-Öğrenmesi ve Sarsa algoritmaları Zamansal Fark Öğrenmesi algoritması sınıfında tanımlanmaktadırlar. Bununla birlikte, Zamansal Fark Öğrenmesi Aktör-Kritik yöntemlerin de temelini oluşturmaktadır.

3.5.4 Sarsa

SARSA, Q-Öğrenmesi gibi pekiştirmeli öğrenme problemi çözümünde kullanılan bir algoritmadır. Her iki algoritma da bir ajan ve bir ortam arasındaki etkileşim ile optimal politikaları öğrenmeyi amaçlamaktadır. SARSA, mevcut politikaya göre ajan eylemlerini seçen ve öğrenen politika tabanlı bir algoritma olarak bilinmektedir [35]. Bu algoritmada Q değeri, o anda gerçekleştirilen eyleme ve ondan sonra seçilen eyleme göre güncellenir. SARSA algoritması için Q-değer fonksiyonu Eş. 3.18'deki şekilde güncellenir.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{3.18}$$

3.5.5 Q-Öğrenmesi

Q-Öğrenme, sonlu bir Markov karar süreci için optimal eylem seçim politikasını belirlemek için kullanılan modelden bağımsız bir pekiştirmeli öğrenme algoritmasıdır. Q-Öğrenme algoritmasında tahmini değer ile gözlemlenen değer arasındaki farka dayalı olarak değer fonksiyonu güncellenmekledir [36]. Bu bağlamda Q-Öğrenme algoritması temporal farklılık öğrenme algoritma sınıfında değerlendirilmektedir.

Q-Öğrenme'nin temelinde Q(s,a) olarak gösterilen Q-değer fonksiyonu bulunmaktadır. Bu fonksiyon, belirli durum-eylem çifti için izlenen optimal politika sonrasında elde edilmesi beklenen ödülü tahmin etmektedir.

$$(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q\,(s',a') - Q(s,a)\right] \tag{3.19}$$

Q-değer fonksiyonu işlevi, gelecekteki adım için en iyi politikaya dayalı olarak belirli bir eylemin genel ödülünü göstermektedir. Q değer fonksiyonu bir ajanın farklı durumlardaki eylemleri için gelecekteki ödüllerin tahmin edilmesinde önemli rol oynamaktadır [37]. Q-Öğrenme algoritması bu değerin belirlenmesi ve güncellenmesi için geçmiş deneyimleri kullanır ve ajanın durumlar arasında hareket ederken daha yüksek ödüle yol açan en uygun stratejileri öğrenmesine yardımcı olmaktadır.

3.5.6 Çevrim içi öğrenme

Çevrimiçi öğrenme, yeni veri akışını sürekli işleyerek yapay zeka modelini güncellemeyi sağlayan bir makine öğrenimi yöntemidir. Model veri girişini aldıktan sonra, kendisini hemen güncellemek ve bir sonraki veri noktasını işlemek için bu bilgileri kullanmaktadır. Bu işlem, modelin gerçek zamanlı olarak yeni verilerle etkileşime girdiği anlamına gelmektedir [38].

Çevrimiçi öğrenme özellikle büyük veri akışları ve dinamik ortamlar için uygundur. Bu yöntemin amacı, modellenmesi zor olan dinamik ortamda çevrimiçi öğrenme yöntemini kullanarak çevreden alınan geri bildirimleri kullanarak yapay zeka yetiştirmektir.

Bu yöntemin avantajlarından biri, modelin en son verilere dayanarak sürekli geliştirilmesidir. Ancak, veriler değişirse veya dağıtım değişirse, modelin performansı düşebilir. Bu nedenle, çevrimiçi öğrenme uygulamalarında, modeli gerektiği gibi sürekli izlemek ve güncellemek önemlidir.

3.5.6 Keşif stratejileri

Pekiştirmeli Öğrenme'de ajanın çevre ile etkileşimi sonucu aldığı ödül, ajanın çevrenin durumuna dair yaptığı tahminlerinde ve aldığı aksiyonlarda önemli rol oynamaktadır. Bu nedenle, çevrenin dinamik koşullarına bağlı olarak bir öğrenmenin gerçekleşebilmesi için

ajanın çevre ile etkileşime geçmesi ve tecrübe edinmesi gerekmektedir. Ajanın çevre ile etkileşime girerek elde ettiği geri bildirim sonucu çevreye dair bir strateji geliştirmesi süreci keşif (exploration) olarak adlandırılmaktadır. Yapılan keşif sonucu ajanın elde ettiği tecrübeyi kullanarak bir aksiyon seçmesi ise sömürü (exploitation) olarak adlandırılmaktadır.

Pekiştirmeli öğrenmede ajanın tecrübe ajanın elde ettiği tecrübe ile bir aksiyon alması beklenmektedir. Ancak bu tecrübenin elde edilmesi için de çevrenin keşfedilmesi gereklidir. Bu ikilem keşif ikilemi olarak adlandırılmaktadır [39].

Ajanın çevre ile etkileşiminde kullanılan keşif stratejisi önemli bir yer tutmaktadır. Kullanılan keşif stratejisi ile keşif ve sömürü arasında elde edilecek denge ajanın etkili şekilde en iyi en iyi aksiyonları bulmasını sağlayabilir veya aksiyonları keşfetmek için çok fazla zaman harcamasına neden olabilir.

Pekiştirmeli Öğrenme'de en sık kullanılan keşif stratejisi epsilon-greedy keşif stratejisidir. Epsilon-greedy keşif stratejisi, pekiştirmeli öğrenmede sıkça kullanılan bir keşif stratejisidir. Bu strateji ajanın hem keşif hem de sömürü yapabilmesini sağlayarak keşif ikilemini ele almaktadır.

Epsilon-greedy yaklaşımı Eş. 3.20'deki gibi ifade edilmektedir.

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A(s)|} & \text{eğer } a^* = \text{argmax}_a Q(s,a) \\ \frac{\epsilon}{|A(s)|} & \text{diğer durumlarda} \end{cases} \quad \textbf{(3.20)}$$

3.6. Derin Q-Öğrenmesi

Derin Q-Öğrenmesi, Derin Öğrenme ve Q-Öğrenme yöntemlerini birleştiren bir pekiştirmeli öğrenme (Reinforcement Learning) algoritmasıdır. Bu yöntem, tıpkı Q-Öğrenmesi gibi bir ajan ile çevresi arasındaki etkileşimi en iyi öğrenerek, en iyi eylem politikasını belirlemeye çalışmaktadır. Derin Q-Öğrenmesi, Q-Öğrenmesi yönteminden farklı olarak durum-aksiyon çiftleri için Q değerlerinin tutulduğu bir tablo yerine derin sinir ağlarını (Deep Neural Networks) kullanmaktadır [40]. Bu yaklaşımda, Q-değerleri, bir sinir ağı tarafından tahmin edilir. Sinir ağı durum (s), ve eylem (a) girdilerini alır ve Q-değerlerini üretmektedir.

Hedef Q-değerleri ağ tarafından Q-değerlerinin tahmin edilmesi ve kayıp fonksiyonu minimizasyonu ile güncellenir. Hedef Q-değer Eş. 3.21'deki gibi hesaplanır [41]. Hedef Q

değerleri, ağın öğrenme sürecinde gerçek Q değeri ile karşılaştırılır ve ağın parametreleri (θ) bu farkı en aza indirecek şekilde güncellenir. Bu süreç, ağın en iyi eylem politikasını öğrenmesini amaçlamaktadır.

$$y_i \coloneqq r_i + \max_{a_{i+1}} \gamma\, Q(s_{i+1}, a_{i+1}; \theta) \tag{3.21}$$

Derin Q-Öğrenmesi yönteminde ağın parametrelerinin, tahmin edilen Q değerleri ile gerçek Q değerleri arasındaki farkı en aza indirecek şekilde güncellemek için bir optimizasyon süreci mevcuttur. Bu süreç içerisinde kayıp fonksiyonu verili parametrelerle tahmin edilen Q değerleri ile gerçek Q değerleri arasındaki farkı ölçer. Bu fark, ağın eylem politikasını iyileştirmek için kullanılmaktadır. Ağın parametreleri güncellenirken, kayıp fonksiyonundan elde edilen değerin en aza indirilmesi amaçlanır. Bu sayede ağ, daha doğru Q değerlerini tahmin edebilir ve daha iyi bir eylem politikası oluşturabilir. Derin Q-Öğrenmesi'nde ağ parametrelerinin güncellenme yöntemi Eş. 3.22'deki şekilde hesaplanmaktadır.

$$\theta \leftarrow \theta - \alpha \cdot \nabla_\theta \mathrm{Loss}(\theta) \tag{3.22}$$

3.7. TD3

TD3, pekiştirmeli öğrenme alanında kullanılan bir aktör-kritik metottur. TD3 algoritması, bir başka pekiştirmeli öğrenme algoritması olan DDPG algoritmasının temellerine dayanmaktadır. TD3 belirli değişiklikler ile daha DDPG'ye göre daha verimli bir öğrenme süreci amaçlamaktadır [42]. Bir aktör-kritik metot olarak TD3, kritik ve aktör bileşenlerini barındırmaktadır.

TD3'te aktör, genelliklere derin sinir ağları ile elde edilen bir politika ağıdır. Bu ağ, ajanın verilen bir durumda optimal bir aksiyon seçmesini hedeflemektedir. Bir politika ağı olarak ağ parametrelerine bağlı olarak aksiyonları Eş.21'deki gibi üretmektedir.

$$a_t = \pi_\phi(s_t) \tag{3.23}$$

Kritik ise bir veya birden fazla Q-değerini tahmin eden derin sinir ağıdır. Bu sayede Q-değerlerinin tahmininin iyileştirilmesi ve aşırı tahmin hatalarının azaltılması amaçlanmaktadır.

TD3'te aktör ve kritik ağları kullanılarak elde edilen hedef Q-değeri Eş. 3.24'teki şekilde hesaplanmaktadır.

$$y_t = r_t + \gamma \min_{i=1,2} Q_{\theta_i'}\left(s_{t+1}, \pi_{\phi'}(s_{t+1})\right) \tag{3.24}$$

TD3 aktör ve kritik ağların yanı sıra ajanın geçmiş deneyimlerini depolamak için kullanılan bir tekrar oynatma belleğine (replay memory) ihtiyaç duymaktadır [43]. Bu sayede TD3 ajanın deneyimlerini depolar ve verimli bir öğrenme için gerekli olan veri tekrarını sağlamaktadır. Tekrar oynatma belleği genellikle Eş. 3.25'teki şekilde ifade edilir.

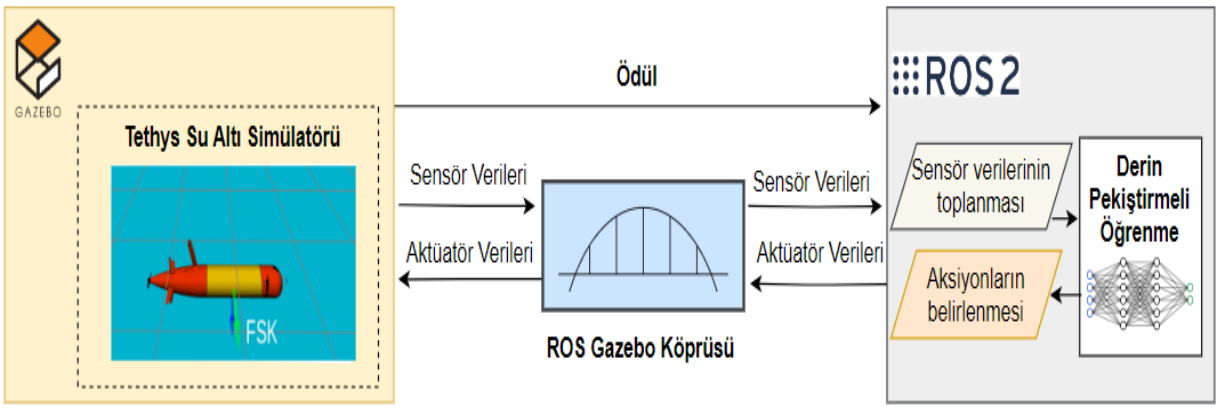
$$\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1}, d_t)\} \tag{3.25}$$

Tekrar oynatma belleğinde yer alan veriler mini-batchler olarak örneklenir. Mini-batch içerisindeki Q-değerlerinin her bir örneklemdeki değeri Eş. 3.26'daki gibi hesaplanır [44].

$$y_t^i = r_t^i + \gamma(1 - d_t^i) \min_{j=1,2} Q_{\theta_j'}\left(s_{t+1}^i, \pi_{\phi'}(s_{t+1}^i) + \epsilon\right) \tag{3.26}$$

3.8 Geliştirilen Simülasyon Kontrol Mimarisi

Geliştirilen simülasyon mimarisi ROS ve Gazebo simülatörü arasında çift taraflı olarak veri akışı gerçekleştirilerek elde edilmiştir. İletişimde çift taraflı veri akışının sağlanmasında ROS Gazebo Köprüsü kullanılmıştır [45].



Şekil 3.5. ROS tabanlı Pekiştirmeli Öğrenme ile otonom su altı yol planlama mimarisi

Tethys UMOSA modeli ve su altı fiziksel ortamı Gazebo içerisinde simüle edilmektedir. ROS Gazebo Köprüsü, Gazebo'dan gelen algılayıcı verilerini ROS mesajlarına

dönüştürmektedir. ROS uygulaması içeresinde oluşturulan düğümler ile bu mesajlar okunmakta ve algılayıcı verileri toplanmaktadır. Toplanan veriler ile Tethys UMOSA'ya ait durum uzayı elde edilmektedir. Elde edilen durum uzayı Derin Pekiştirmeli Öğrenme algoritması için girdi olarak kullanılmakta ve algoritma tarafından hedefe oluşmak için gerekli aksiyonlar belirlenmektedir. Bu aksiyonlar, ROS düğümleri aracılığı ile aktüatör verilerine dönüştürülmektedir. Aktüatör verilerinin ROS Gazebo Köprüsü ile Gazebo simülatörüne aktarılması ile ajanın hareketi sağlanmaktadır. Tethy UMOSA'nın hareket ettiği her bir kontrol döngüsü sonunda ise hedeflenen konum ve tercih edilen aksiyonun ajanı bu konuma yakınlaştırması veya uzaklaştırılması durumuna göre ödül verilmektedir. Verilen ödül ile Derin Pekiştirmeli Öğrenme algoritmasına geri bildirim verilmektedir.

Çizelge 3. 2. ROS-Gazebo Pekiştirme Öğrenmesi tabanlı kontrol

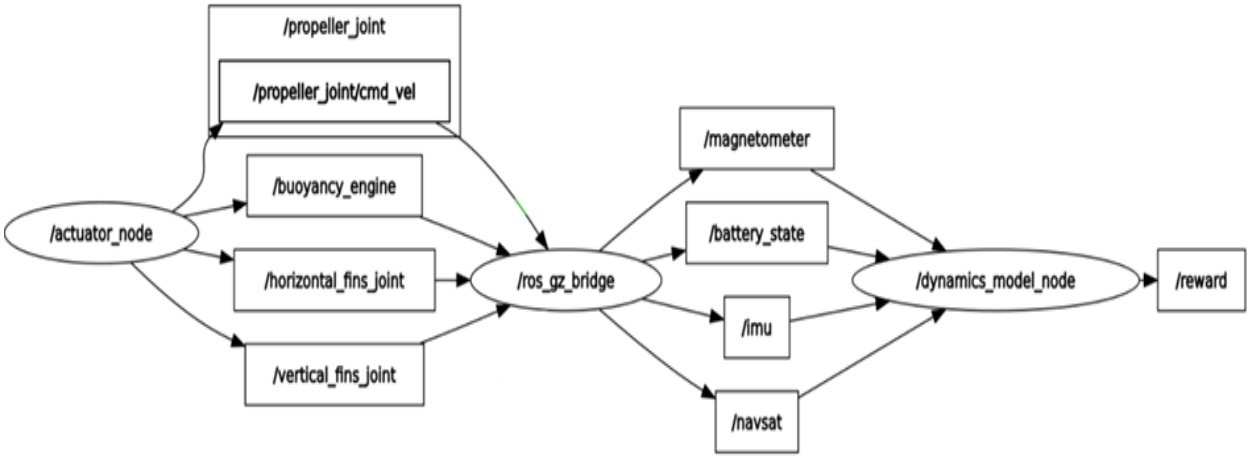
```
Gazebo simülasyonunu Tethys UMOSA modeli ile başlat
ROS-Gazebo köprüsü ile Gazebo ve ROS arasındaki veri akışını başlat
Simülasyon içerisinde rastgele bir noktayı hedef olarak belirle.
Pekiştirmeli Öğrenme algoritmasını başlat
    Durum (state) ve eylem (action) boyutları belirlenir
    Hiperparametreler (öğrenme oranı, indirim faktörü) ayarlanır
Bölüm sonlanana kadar ve sonlanma koşulları karşılaşana kadar:
    Ortamdan mevcut durumu gözlemle
        Algılayıcı verilerini vektörel olarak kaydet ve sonraki durumu oluştur
    Hedef noktaya olan mesafedeki değişime bağlı olarak ortamdan ödül al
    Derin Q-Ağları algoritmasını (durum, eylem, ödül, sonraki durum) demeti ile güncelle.
    Sonlanma koşullarını kontrol et:
    Eğer sonlanma koşulları sağlandıysa veya bölüm uzunluğu aşıldıysa:
        Yeniden Başla
```

3.9 Geliştirilen ROS İletişim Mimarisi

ROS iletişim mimarisi Tethys UMOSA üzerinde bulunan ve çalışma kapsamında kullanılan algılayıcı ve aktüatörlerin ROS düğümleri olarak yansıtılması ve karşılık gelen ROS mesaj tipinin iletişim ağında yayınlanması veya paylaşılması ile elde edilmektedir. Kullanılan her aktüatör ve algılayıcı için ilgili mesaj tipleri tanımlanmıştır.

Su altı aracının hareket kontrolü geliştirilen "Aktüatör Kontrol Modülü" düğümü tarafından sağlanmaktadır. Bu modül, kontrol döngüsü içerisinde Derin Pekiştirmeli Öğrenme algoritmasının ürettiği aksiyonları paylaşmaktan sorumludur.

Su altı aracının su altı ortamından gelen verileri toplaması ve işleyebilmesi için ise, bir "Dinamik Model" düğümü geliştirilmiştir. Bu düğüm algılayıcı verilerine abone olarak simülasyon kontrol döngüsü içinde durum uzayını oluşturmakta ve girdi olarak Derin Pekiştirme Öğrenmesi algoritmasına iletmektedir. Ayrıca, hedeflenen konum ve üretilen aksiyon ile bu konuma yaklaşma / uzaklaşma durumuna bağlı olarak pozitif veya negatif ödülün üretilmesini sağlayarak kontrol algoritmasında geri bildirim mekanizmasını sağlamaktadır. Oluşturulan sistemde ROS Gazebo arasındaki iletişimin ve ROS konularının düğümler arasındaki akışı ROS rqt_graph yazılım paketi kullanılarak Şekil 3.6.'daki şekilde görüntülenmiştir [45].

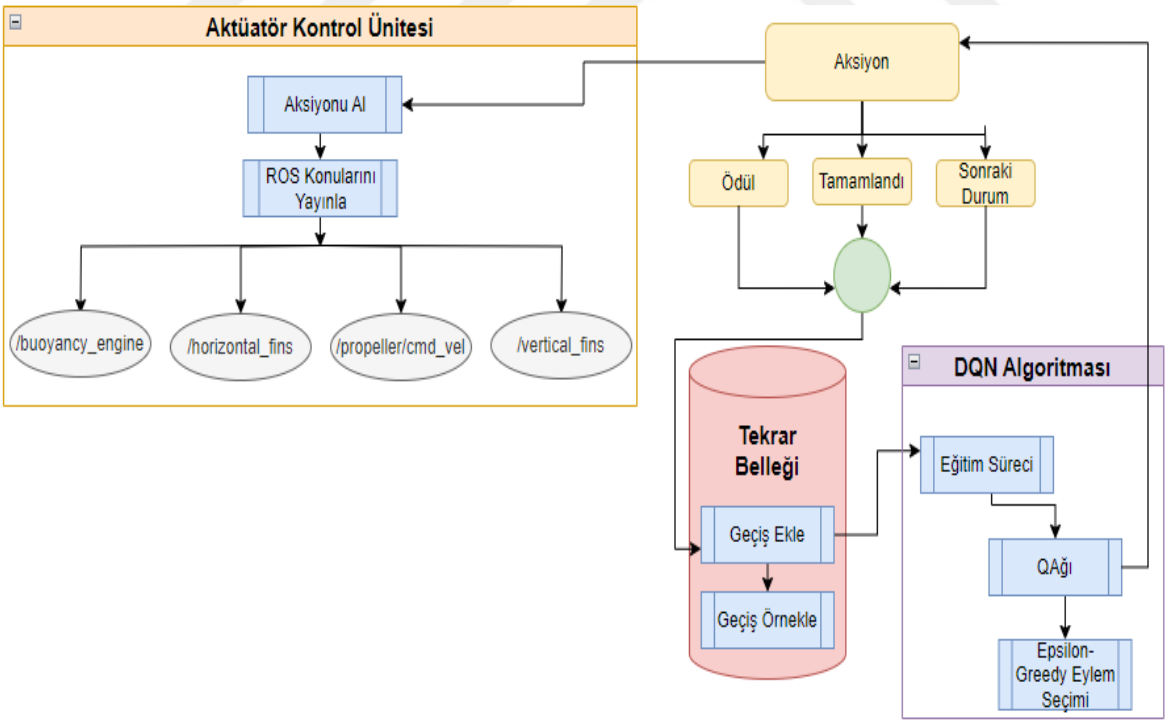


Şekil 3.6. ROS rqt_graph yazılım paketi ile görüntülenen iletişim grafiği

3.10 Geliştirilen Derin Q-Öğrenmesi Algoritması

Geliştirilen Derin Q-Öğrenmesi algoritmasında tekrar belleği kullanılarak aşırı uyum (overfitting) ihtimalinin en aza indirilmesi amaçlanmıştır. Tekrar belleğinde örneklenen ödül, durum ve hedefe ulaşılma durumu bilgisi Derin Q-Ağları algoritmasının eğitilmesinde kullanılmıştır. Derin Q-Ağlarında Epsilon-Greedy stratejisi ile elde edilen aksiyon ROS düğümleri ile yayınlanarak su altı aracının hareket etmesi sağlanmıştır.

Bu hareketin sonucunda, sisteme ödül ile geri bildirim verilmiştir. Ödülün hesaplanmasında simülasyon için kullanılan hedef noktasına uzaklık göz önünde bulundurulmuş ve hedefe yakınlaşma ve uzaklaşma durumuna göre pozitif, negatif ödül verilmiştir. Bu şekilde bir kontrol döngüsü elde edilmiş olup, kontrol döngüsü içerisinde geri bildirim sağlanması amaçlanmıştır. Her bir kontrol döngüsünde önceden belirlenmiş olan simülasyon sonlanma şartları kontrol edilerek bölümün devam etmesi veya sonlanması sağlanmıştır. Geliştirilen Derin Q-Öğrenmesi mimarisi Şekil 3.7.'daki gibidir.



Şekil 3.7. Geliştirilen Derin Q-Öğrenmesi algoritması mimarisi

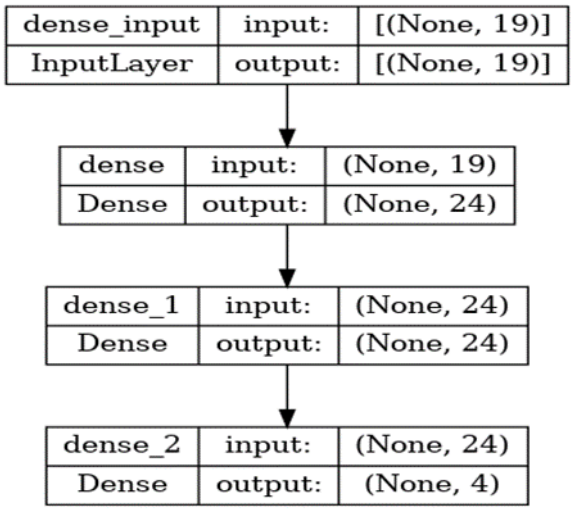
Tethys UMOSA'nın kontrolü için geliştirilmiş olan Derin Q-Öğrenmesi algoritması hiperparametreleri Çizelge 3.3'teki gibidir. Simülasyon başında belirlenen keşif oranı 1,0 olup, keşif azalma oranı 0,99 olarak belirlenmiştir. Keşif azalma oranı ile Epsilon Greedy stratejisi çerçevesinde ajanın simülasyon başlangıcında çevresini tanıyarak tecrübe ettikten sonra kendi geçmiş tecrübesine dayanarak aksiyon alması amaçlanmıştır.

Geliştirilen Derin Q-Öğrenmesine yönteminde kullanılan hiperparametreler Çizelge 3.3'teki gibidir.

Çizelge 3.3. Derin Q-Öğrenmesi algoritması hiperparametre değerleri

Hiperparametre	**Değer**
Durum Boyutu	19
Eylem Boyutu	4
Öğrenme Oranı	0,1
İndirim Faktörü	0,9
Keşif Oranı	1,0
Keşif Azalma Oranı	0,99

Geliştirilen Derin Q-Öğrenmesi metodunda kullanılan Q-ağlarına ilişkin YSA mimarisi şekil 3.8.'deki gibidir.



Şekil 3.8. Derin Q-Ağları YSA mimarisi

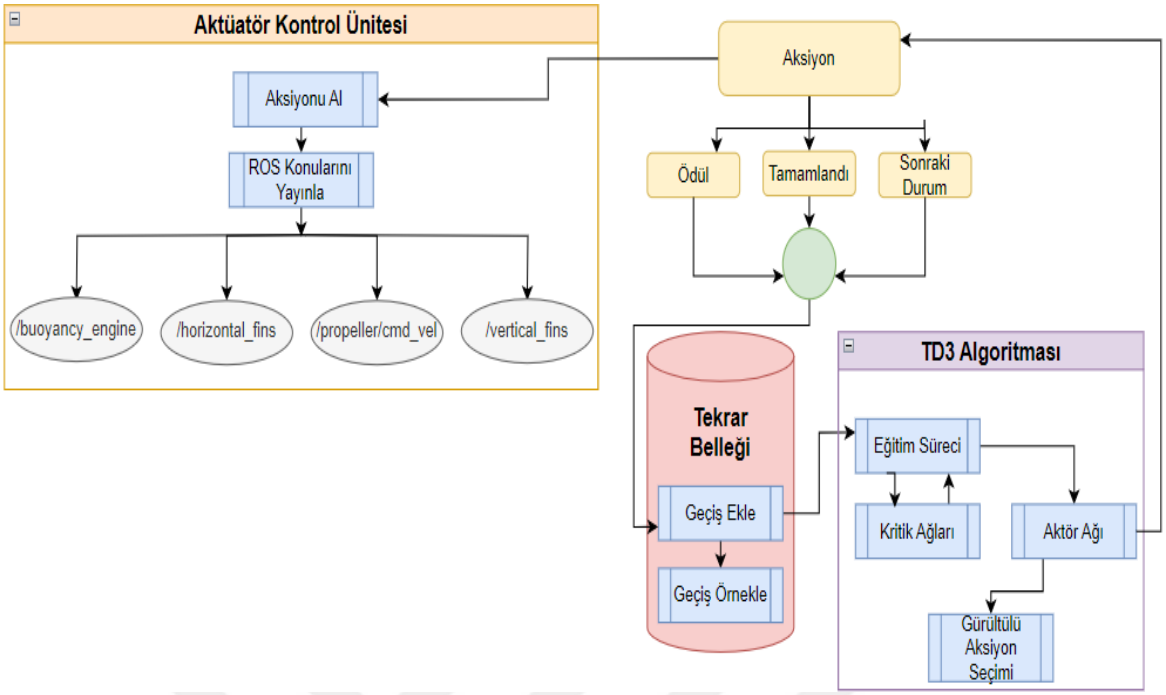
3.11 Geliştirilen Derin TD3 Algoritması

Geliştirilen algoritma akışında ajanın çevreden aldığı durum bilgisine dayanarak bir aksiyon seçmesi ve bu aksiyonu ortama uygulaması sağlanmıştır. Bu eylem sonucunda elde edilen yeni durum, ödül ve tamamlama bilgisi tekrar belleğine eklenir. Tekrar belleğinde saklanan rastgele örnekler alınarak kritik ağları güncellenmektedir. Kritik ağları aktör ağı tarafından seçilen eylemin değerini tahmin ederek, verilen durum ve eylem kombinasyonunun ne kadar uygun olduğunu belirlemektedir. Kritik ağları, gerçek değerlerle tahmin edilen değerler arasındaki farkı (zamansal hata) minimize ederek eğitilmektedir.

Keşif stratejisini belirlemek için ise, aktör ağ tarafından üretilen aksiyonlara gürültü eklenerek ajanın seçeceği eylemlere rastgele bir varyasyon eklenerek eylem uzayında çeşitliliği sağlanması amaçlanmıştır.

Elde edilen aksiyonlar ile yüzdürme motoru, yatay yüzgeçler, pervane ve dikey yüzgeçlere komut gönderilerek ve hareket kontrolü gerçekleştirilmiştir.

TD3 algoritması kullanılarak yürütülen deneylerde 128 batch ile eğitim gerçekleştirilmiştir. TD3 algoritmasının geliştirilmesinde Fujimoto ve diğerleri tarafından geliştirilmiş TD3 uygulaması dikkate alınarak YSA mimarisi oluşturulmuştur [42].



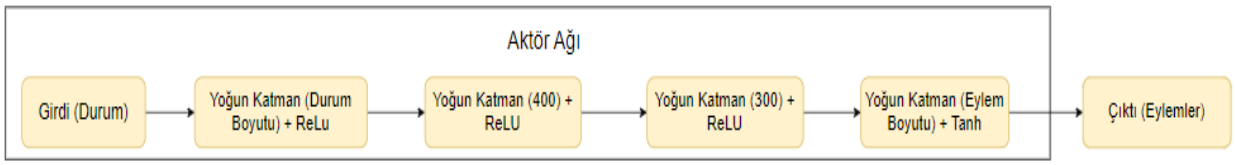
Şekil 3.9. Geliştirilen TD3 algoritması mimarisi

Geliştirilen TD3 algoritmasında kullanılan hiperparametreler Çizelge 3.4'teki gibidir.

Çizelge 3.4. TD3 algoritması hiperparametre değerleri

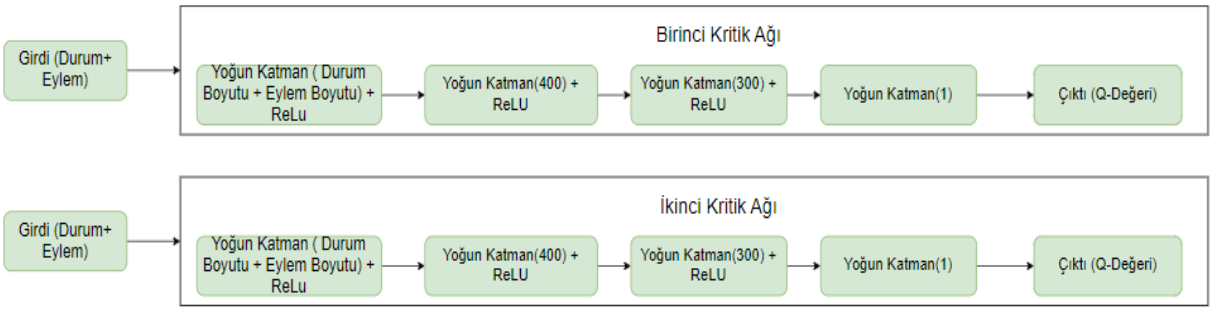
Hiperparametre	**Değer**
Durum Boyutu	19
Eylem Boyutu	4
İndirim Faktörü	0,99
Yumuşak Güncelleme Faktörü	0,005
Öğrenme Oranı -Aktör Ağ	0,003
Öğrenme Oranı – Kritik Ağ	0,003
Keşif Gürültüsü Standart Sapması	0,1

TD3 algoritmasında kritik ağının eğitiminde Huber kayıp fonksiyonu kullanılmıştır. Aktör ağına ait YSA mimarisi Şekil 3.10'daki gibidir. Aktör ağı için oluşturulan YSA 4 katmandan oluşmaktadır.



Şekil 3.10 Aktör ağı YSA mimarisi

Kullanılan kritik ağlarına ait YSA mimarisi Şekil 3.11'deki gibidir. İki kritik ağı kullanılmakta olup, toplamda 4 katmandan oluşan bir YSA yapısı oluşturulmuştur.



Şekil 3.11 Kritik ağları YSA mimarisi

4. DENEYSEL ÇALIŞMALAR

Bu bölümde, tez çalışması kapsamında geliştirilen yöntemlerin deneysel bulguları sunulmaktadır.

4.1. Kullanılan Donanım

Deneysel çalışmalar, Intel(R) Core(TM) i7-10510U işlemci, 16 GB RAM, NVIDIA GeForce MX250 GPU ve 64 bit Windows 11 işletim sistemine sahip proje bilgisayarında gerçekleştirilmiştir.

Deney ortamının oluşturulması için bir Ubuntu / Linux işletim sistemine sahip bir sanal makine kullanılmıştır.

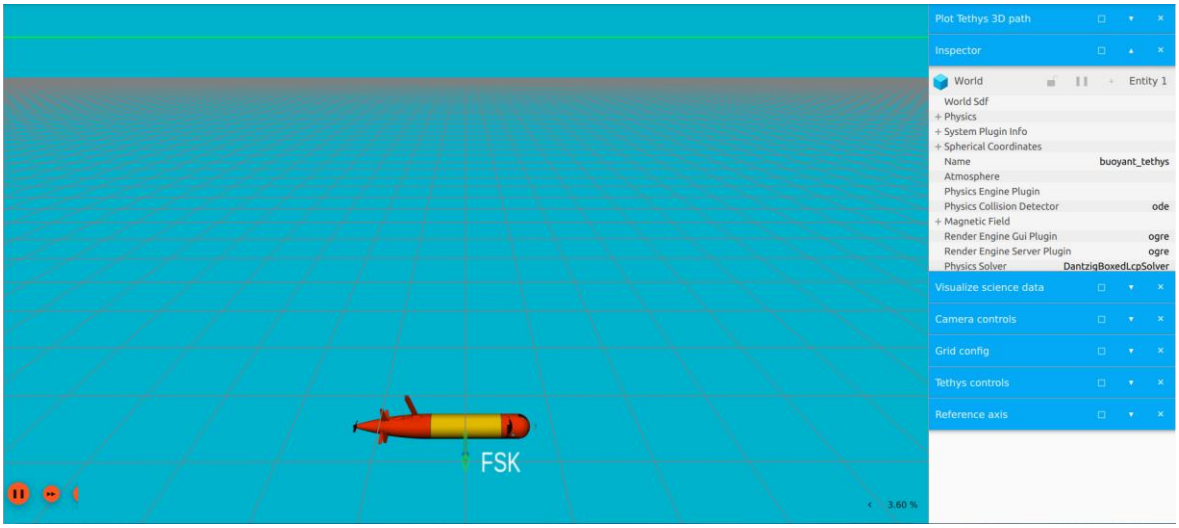
4.2. Kullanılan Deney Düzeneği

Bölüm 3'te açıklanan Pekiştirmeli Öğrenme modelleri Python Keras kütüphanesi yardımı ile oluşturulmuştur. Deney verilerinin kaydedilmesinde Tensorflow kütüphanesine ait veri görselleştirme araçları kullanılmıştır. Elde edilen deney düzeneğinde Pekiştirmeli Öğrenme algoritması kullanılarak elde edilen kontrol döngüsü sağlanmış ve simülasyon sonlanma koşulları incelenerek içinde bulunan bölümün sonlandırılıp sonlandırılmamasına karar verilmiştir. Simülasyon sonlanması durumunda elde edilen ağ parametreleri kaydedilerek kazanılan deneyimin bir sonraki bölümde kullanılması amaçlanmıştır.

Kaydedilen YSA parametreleri bir sonraki simülasyon bölümünde kullanılarak modelin dinamik çevreden aldığı sonuçlara bağlı olarak eğitilmesi amaçlanmıştır. Simülasyon sonlanmasının ölçümünde kullanılan kurallar:

- batarya seviyesinin 10 % altında olması,
- son 60 sn içerisinde hareket gözlemlenmemesi,
- hedefe ulaşılmasıdır.

Kullanılan simülasyon ortamı şekil 4.1'deki gibidir.



Şekil 4.1. Tethys su altı simülasyon ortamı

Oluşturulan deney düzeneğinde kullanılan simülasyon senaryosu şekil 4.1'de görülmektedir. Simülasyonda zaman faktörü gerçek zaman ölçeğinde ve 1:1 oranında kullanılmıştır. Kullanılan simülasyon ortamı için su altı zemininin bulunmadığı, rüzgarlı bir açık deniz ortamı simüle edilmiştir. Simülasyon konfigürasyonunda Gazebo Hidrodinamik eklentisi kullanılarak belirlenen ve araca etki eden su altı kuvvetleri, Çizelge 4.1'de görülmektedir.

Çizelge 4.1 Simülasyon ortamı hidrodinamik parametreleri

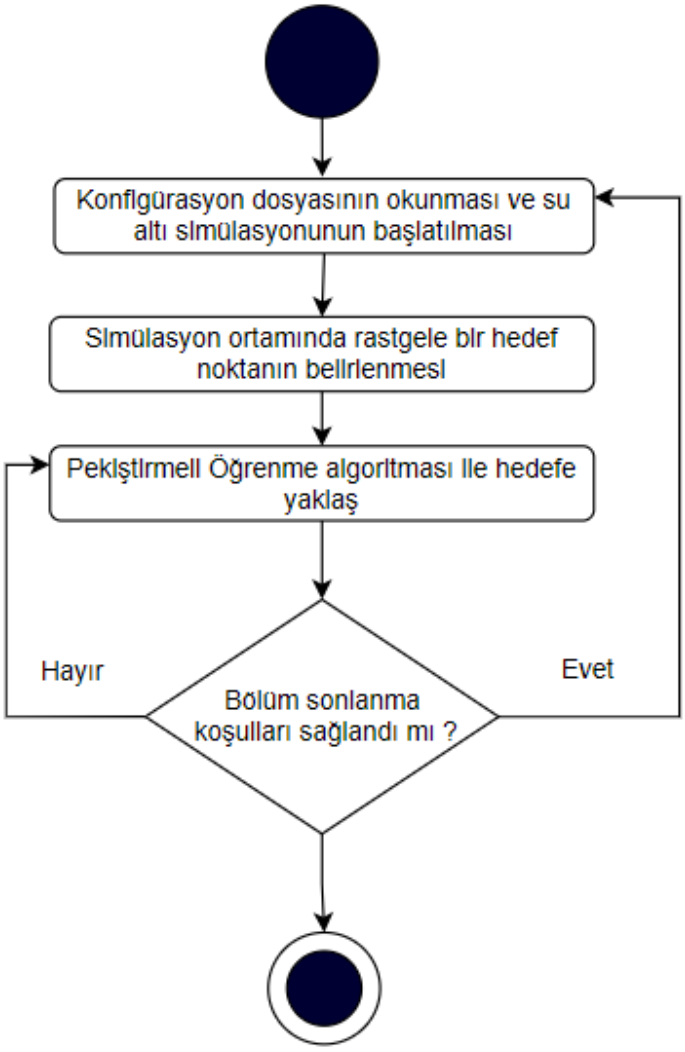
Açıklama	**Değer**	**Birim**
İleri hareket kütle artışı (Surge)	-4,88	kg
Yanal hareket kütle artışı (Sway)	-126,325	kg
Dikey hareket kütle artışı (Heave)	-126,325	kg
Yuvarlanma kütle artışı (Roll)	0	kg·m²
Yan yatma kütle artışı (Pitch)	-33,5	kg·m²
Dönme kütle artışı (Yaw)	-33,5	kg·m²

Simülasyon ortamına ait fiziksel özellikler Çizelge 4.2'de görülmektedir.

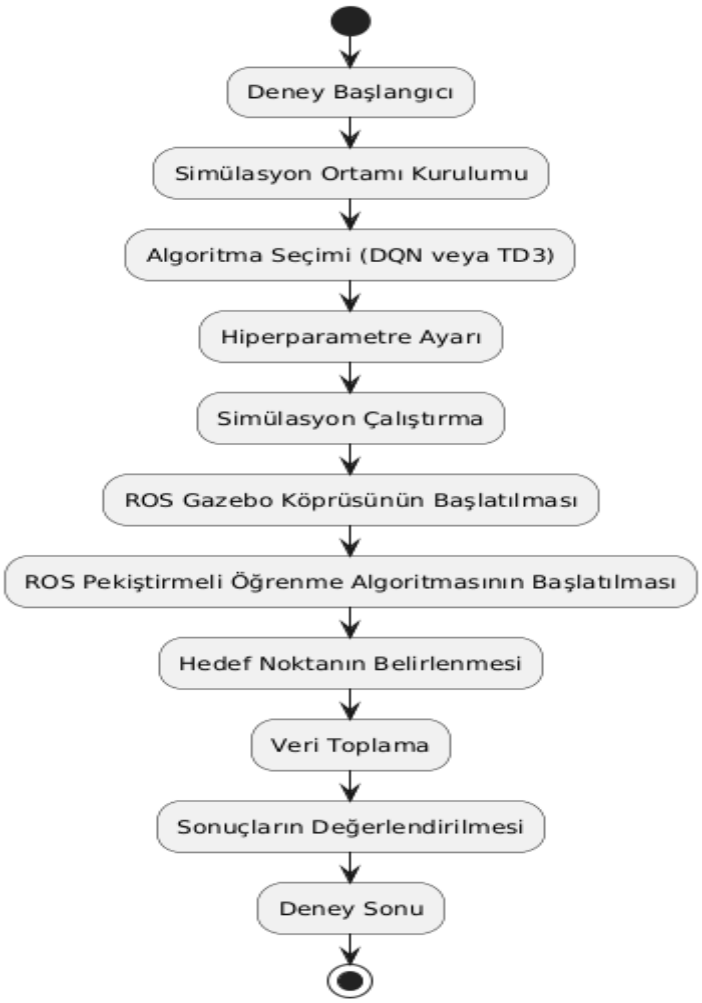
Çizelge 4.2. Simülasyon ortamı fiziksel özellikleri

Açıklama	**Değer**	**Birim**
Su Yoğunluğu	1025	kg/m^3
Rüzgar Hızı	1	m/s
Rüzgar Yönü	Doğu	-

Oluşturulan simülasyon senaryosunda simülasyon konfigürasyondaki hidrodinamik ve fiziksel parametreler ile başlatılmakta ve rastgele bir hedef nokta seçilmektedir. Bölüm sonlanma koşullarının sağlanması ile simülasyon ortamı yeniden başlatılır ve yeni bir hedef nokta belirlenir. Simülasyon senaryosu Şekil 4.2'de görülmektedir.



Şekil 4.2 Simülasyon senaryosu



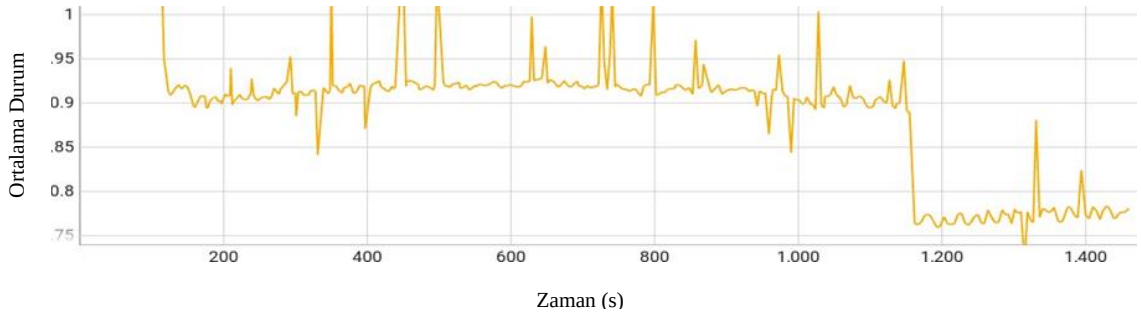
Şekil 4.3. Takip edilen deney metotları

4.2. Deneysel Sonuçlar

Bu bölümde, gerçekleştirilen deneylerin ayrıntıları, elde edilen sayısal veriler ve literatürle yapılan karşılaştırmalar sunulmuştur.

4.2.1. Derin Q-Öğrenmesi ile Su Altı Yol Planlama Deneyi

Derin Q-Öğrenmesi kullanılarak yapılan otonom yol planlama deneyinde, algılayıcı verilerinden alınan veriler toplanarak, 0-1 aralığında normalleştirilmiştir. Normalleştirilen ortalama durum değerlerinin zamana göre gösterdiği değişim Şekil 4.2'de gösterildiği gibidir.



Şekil 4.4. Derin Q-Öğrenmesi deneyi ortalama durum zaman grafiği

Simülasyon başlangıcında ortalama durum değerlerinin kabaca 0,9 – 1,0 aralığında değiştiği görülürken, simülasyon sonunda 0,75 düzeyinde değerler gözlemlenmiştir.

Simülasyon süresince ajanın seçtiği aksiyonlara bağlı olarak ödüller geri bildirim olarak kontrol sistemine verilmiş olup, Derin Q-Öğrenmesi algoritması kullanılarak elde edilen ödüllerin zamana bağlı değişimi şekil 4.3'te görülmektedir.

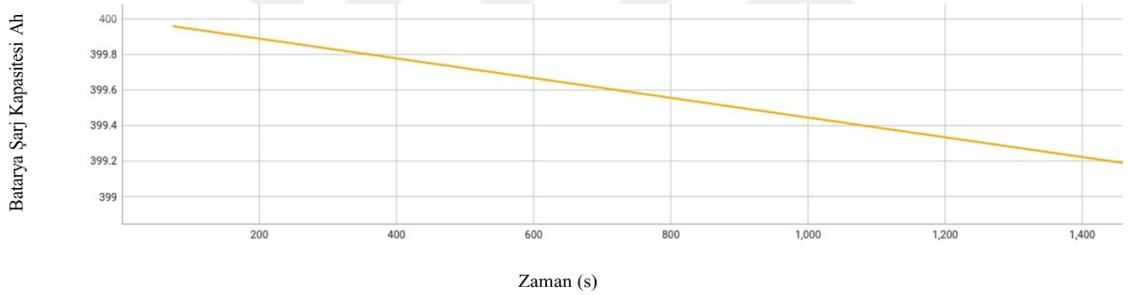
Şekil 4.5. Derin Q-Öğrenmesi deneyi ödül, zaman grafiği

Epsilon-Greedy, stratejisi bağlamında ajanın başlangıçta rastgele eylemler seçtiği ve bu eylemlerin aldığı ödüllerin negatif olduğu gözlenmektedir. Simülasyon akışı içerisinde hedefe olan uzaklığa orantılı olarak verilen ödüllerin (-20, 50) aralığında olduğu gözlemlenmiştir. Keşif oranının keşif azalma oranına bağlı olarak azalması ile ajan daha önceki eylemler sonucu elde ettiği tecrübeyi kullanmaya başlamaktadır. Ajanın elde ettiği tecrübeye bağlı olarak, alınan ödüller en yüksek seviyeye çıktığı ve simülasyon sonlanmadan önce alınan ödüllerin 0 seviyesine yakınsadığı görülmektedir.

Derin-Q algoritması kullanılarak yapılan deneylerde batarya kapasitesinin zamana bağlı olarak lineer düşüş gösterdiği gözlemlenmişir. 1400 saniyelik bir simülasyon sonucunda 400Ah olan batarya kapasitesi 399.2 Ah seviyesine düşmüştür.

Simülasyon sonucunda batarya seviyesinde gözlemlenen düşüş 0.2% seviyesindedir. Sırası ile batarya doluluk oranının ve batarya şarj kapasitesinin zamana bağlı değişim Şekil 4.4 ve Şekil 4.5'te görüldüğü gibidir.

Şekil 4.6. Derin Q-Öğrenmesi deneyi batarya doluluk yüzdesi grafiği



Şekil 4.7. Derin Q-Öğrenmesi batarya şarj kapasitesi zamana bağlı değişim grafiği

Hedefe uzaklığın zamana bağlı değişimi Şekil 4.6'da görüldüğü gibidir.

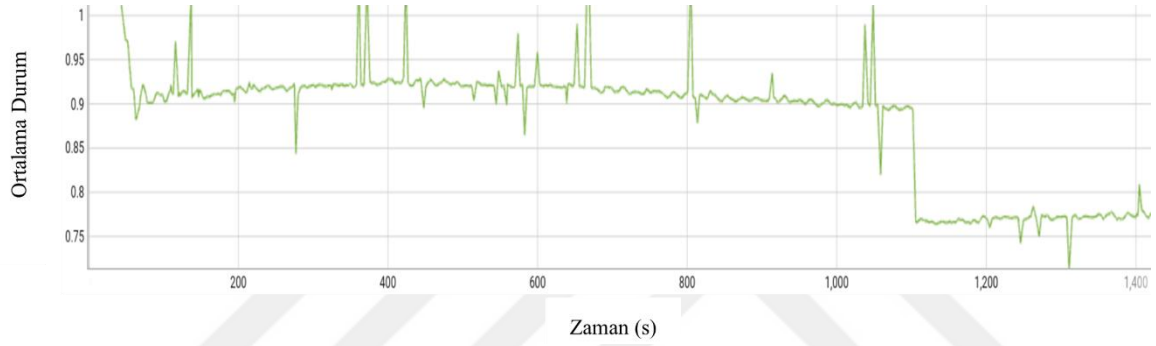
Şekil 4.8. Derin Q-Öğrenmesi deneyi hedefe uzaklık zaman grafiği

Simülasyon başlangıcında ajanın tecrübe edindiği süre içerisinde hedefe olan uzaklıkta artış gözlemlenmektedir. Simülasyon sürecinde elde edilen tecrübe ile hedefe yakınlaşa hızında artış gözlemlenmektedir.

Simülasyon sonunda hedefe 100 m uzaklığa kadar yaklaşıldığı ve ajanın hareketsizliğine bağlı olarak bölüm sonlanma koşullarının tamamlandığı ve simülasyonun sonladığı gözlenmiştir.

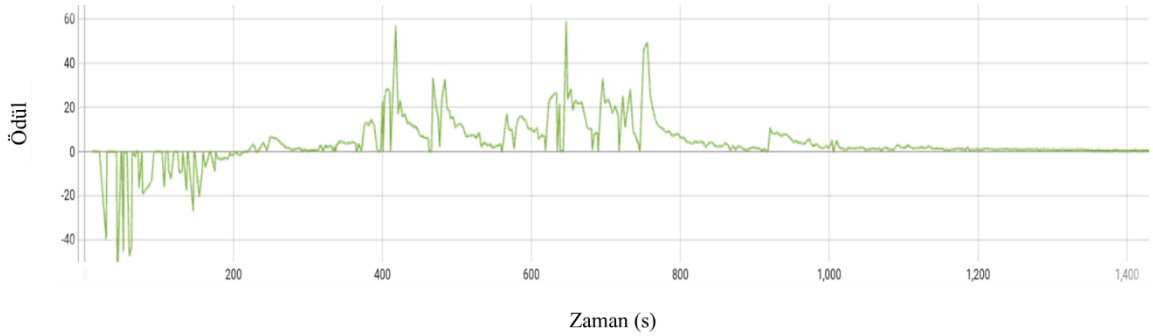
4.2.2 TD3 ile Su Altı Yol Planlama Deneyi

TD3 algoritması ile yapılan deneylerde, ortamdan alınan algılayıcı verileri 0-1 aralığında normalleştirildiğinde ortalama durumun zamana bağlı olan değişimi Şekil 4.7'deki gibidir.



Şekil 4.9. TD3 deneyi ortalama durum zaman grafiği

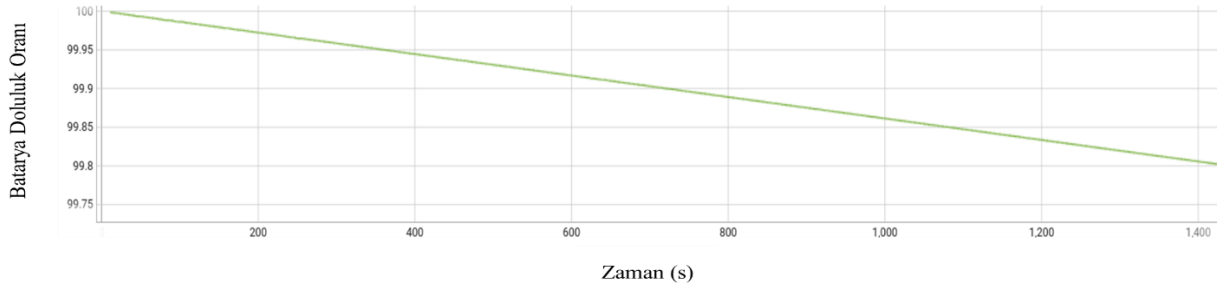
TD3 algoritması kullanılarak yapılan deneylerde zamana bağlı olarak elden edilen ödüller Şekil 4.8'deki gibidir.



Şekil 4.10. Derin TD3 deneyi ödül, zaman grafiği

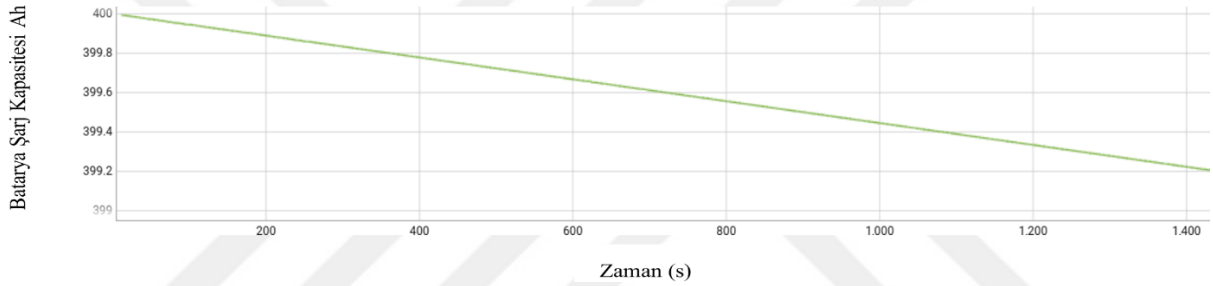
Ödül zaman grafiğinde, 200. saniyeye kadar negatif ödüllerin alındığı ancak 200. saniyeden sonra pozitif ödüller alınarak hedefe yaklaşıldığı gözlemlenmiştir. Yaklaşık 1000. saniyeden sonra ise, alınan ödüllerde ciddi bir değişim gözlemlenmemiş ve ajanın hareketsizliğine bağlı olarak simülasyon sonlanma koşullarının sağlandığı ve simülasyonun sonlandığı gözlemlenmiştir.

Batarya doluluk oranının zamana bağlı değişimi Şekil 4.9'da görüldüğü şekilde olup, batarya doluluk oranında yaklaşık 0,2 % azalma gözlemlenmiştir.



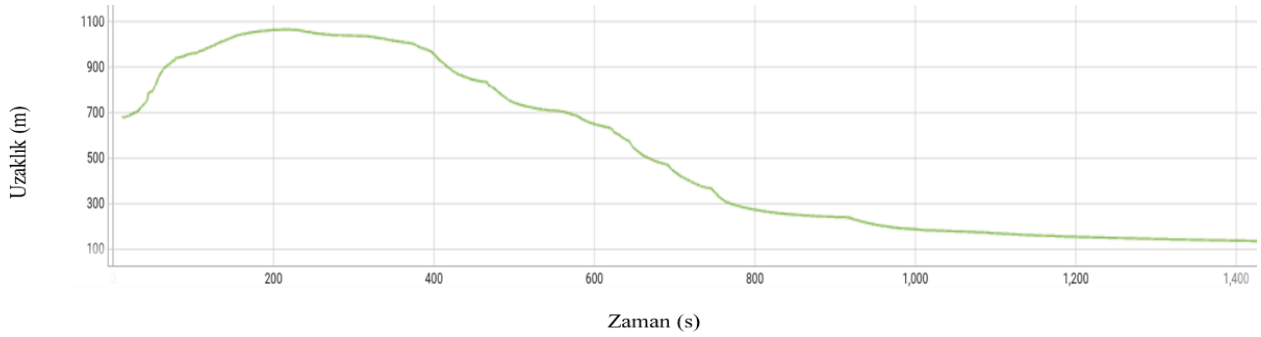
Şekil 4 10. TD3 deneyi batarya doluluk yüzdesi grafiği

Batarya kapasitesinin zamana bağlı olarak değişimi Şekil 4.10'da görüldüğü gibi olup, 0,8 Ah'lık bir düşük gözlemlenmiştir.



Şekil 4.11. TD3 deneyi batarya doluluk yüzdesi grafiği

Su altı aracının hedefe uzaklığının zamana bağlı olarak değişimi Şekil 4.11'deki gibidir. İlk 200 saniyede hedefe olan uzaklıkta artış gözlemlenirken, ajanın çevreden edindiği tecrübe ile hedefe yaklaşma hızında 200 – 400 sn arasında artış gözlemlenmiştir. Hedefe ise yaklaşık 100 m kadar yaklaşılmış ve hareketsiz kalma koşulundan dolayı simülasyon sonlandırılmıştır.



Şekil 4.12. TD3 deneyi hedefe uzaklık zaman grafiği

Çizelge 4.3 Derin Q-Öğrenmesi ve TD3 algoritmalarının performans karşılaştırması

Metrik	**DQN**	**TD3**	**Birim**
Toplam Ödül	3038	**3577**	-
Ortalama Ödül	2,170	**2,555**	-
Toplam Bölüm Sayısı	240	**110**	-
Ortalama Durum Değeri	0,778	**0,864**	-
Hedefe Yakınsama Başlangıcı	400 s	**200 s**	s
Hedefe Yaklaşma Mesafesi	**100**	**100**	m
Batarya Doluluk Oranı	**99,2**	**99,2**	%

Yapılan deneyler sonucunda, Derin Q-Öğrenmesi ve TD3 algoritmalarının performansları karşılaştırılmış olup, elde edilen sonuçlar ve karşılaştırma Çizelge 4.2'de gözlemlenmektedir. Elde edilen sonuçlar oluşturulan ödül algoritması ile TD3 algoritması ile elde edilen ortalama ve toplam ödülün Derin Q-Öğrenmesi metoduna göre daha yüksek olduğunu göstermektedir.

Ayrıca, TD3 algoritmasının eğitilmesi gereken süre 110 bölüm iken Derin Q-Ağları algoritmasının eğitilmesi için gereken süre 240 bölüm olarak gözlemlenmiştir.

Her iki algoritma ile yapılan deneylerde aracın hedefe 100 m kadar yaklaştığı gözlemlenmiş, ancak TD3 algoritması ile elde sonuçlarda algoritmanın Derin Q-Ağları algoritmasına göre daha erken bir sürede hedefe yakınsadığı saptanmıştır. Bu sonuçlar ölçülen ortalama durum değeri ile de paralellik göstermekte olup TD3 algoritmasında daha yüksek ortalama durum değeri gözlemlenmiştir. Bu sonucun elde edilmesinde aracın daha yüksek ivme ile kontrol edilmesine bağlı olarak IMU algılayıcısında okunan açısal ve çizgisel hız değerlerindeki yükseklik olduğu düşünülmektedir.

5. SONUÇLAR VE TARTIŞMA

Bu tez çalışması kapsamında Otonom Su Altı Araçlarında (OSA) yol planlama problemine ilişkin çözüm önerileri geliştirilmiştir. Literatürde yapılan benzer OSA yol planlama çalışmalarından farklı olarak, pekiştirmeli öğrenme yöntemleri kullanılarak veri akışının kısıtlı olduğu bir ortamda aktif olarak görev yapan bir UMOSA modeli ile gerçek zamanlı yol planlamanın gerçekleştirilmesi amaçlanmıştır.

Yol planlama algoritmasının geliştirilmesi ve test edilmesinde, ROS ve Gazebo arasında iletişimi sağlayan bir iletişim mimarisi oluşturulmuştur. Elde edilen sonuçlar aşağıdaki şekilde özetlenebilir:

Bir dinamik yol planlama metodu olarak pekiştirmeli öğrenme algoritmaları kullanıldığında, veri akışının görece az olduğu ortamlarda yüksek kontrol becerisi gerektiren bir otonom su altı aracı türünün kontrolünün mümkün olduğu gözlemlenmiştir. Bu bağlamda, devamlı aksiyon uzayını destekleyen bir aktör-kritik metot olan TD3 algoritmasının, Derin Q-Öğrenmesi algoritmasına kıyasla daha başarılı olduğu ortaya konulmuştur.

Simülasyon tabanlı öğrenme ile maliyet etkin şekilde su altı yol planlama algoritmalarının geliştirilmesi ve test edilmesi mümkün olmuştur. Ayrıca, pekiştirmeli öğrenme tabanlı modellerin eğitiminde kullanılan ödül politikasının alınan verime doğrudan etki ettiği gözlemlenmiştir. Ödül mekanizmasında negatif ödüllerin tasarlanmasının veya simülasyon sonlanma kurallarının doğru belirlenmesinin öğrenmede yüksek öneme sahip olduğu saptanmıştır. Negatif ödül doğru tasarlanmadığında, ajan negatif ödül almamak için eylemsizliği tercih edebilmektedir. Bu nedenle, simülasyon tabanlı eğitimde eylemsizliğe yol açabilecek durumların belirlenerek bu durumlar için bölümün sonlandırılmasının etkili olabileceği belirlenmiştir.

Sonuçların daha iyi hale getirilmesi ve gelecekte yapılacak çalışmalar için öneriler aşağıdaki gibidir:

İlk olarak, farklı simülasyon koşulları ve test ortamlarında çapraz doğrulama yapılarak modelin genelleme yeteneğinin artırılması üzerinde çalışılabilir. Eğitim sırasında su altı ortamının çeşitli koşullarını simüle etmek, aracın farklı durumlara uyum yeteneğini

geliştirebilir. Bu amaçla, simülasyon ortamına farklı su altı akıntıları, engeller ve derinlikler gibi değişkenler eklenebilir.

Ayrıca, simülasyon ortamındaki testlerin yanı sıra fiziksel ortamda testler yapılarak simülasyon tabanlı öğrenmede iyileştirmeler sağlanabilir. Modelin beklenmedik durumlarda verebileceği hataları ve alınan aksiyonlardaki anomalileri tespit etmek, aracın güvenliğini ve çevreye uyum yeteneğini artırabilir. Simülasyon tabanlı eğitim algoritmaları, barındırdıkları karmaşıklık düzeyi ve farklı yazılım bileşenlerinin kullanımı nedeniyle yüksek işlem gücü gerektirmektedir. Bu durum, simülasyonun hızlandırılarak daha hızlı sonuçlar elde edilmesini zorlaştırmaktadır. Bu nedenle, simülasyon sistemlerinde farklı simülasyonların asenkron olarak farklı hiperparametreleri ve politikaları deneyebileceği bir yapı oluşturulabilir.

KAYNAKLAR

1. İnternet: National Centers for Environmental Information (NCEI), National Oceanic and Atmospheric Administration (NOAA) Ocean Exploration: Mapping. URL: https://oceanexplorer.noaa.gov, Son Erişim Tarihi: 01.06.2024

2. Phillips, A. B., Templeton, R., Roper, D., Morrison, R., Pebody, M., Bagley, P. M., ... & Furlong, M. (2023). Autosub Long Range 1500: A continuous 2000 km field trial. *Ocean Engineering*, *280*, 114626.

3. Sun, Y., Abudula, A., Yang, H., Chiang, S. S., Wan, Z., Ozel, S., ... & Onal, C. D. (2021). Soft mobile robots: A review of soft robotic locomotion modes. *Current Robotics Reports*, *2*, 371-397.

4. Theja, V. B., Yachameni, P., Shakeera, S., & Venkataraman, H. (2022, February). Integration of Gazebo and ROS for Underwater Vehicle Environment. In *OCEANS 2022-Chennai* (pp. 1-6). IEEE.

5. Watanabe, T., Neves, G., Cerqueira, R., Trocoli, T., Reis, M., Joyeux, S., & Albiez, J. (2015, October). The rock-gazebo integration and a real-time auv simulation. In *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)* (pp. 132-138). IEEE.

6. Player, T. R., Chakravarty, A., Zhang, M. M., Raanan, B. Y., Kieft, B., Zhang, Y., & Hobson, B. (2023, May). From Concept to Field Tests: Accelerated Development of Multi-AUV Missions Using a High-Fidelity Faster-than-Real-Time Simulator. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3102-3108). IEEE.

7. Carroll, K. P., McClaran, S. R., Nelson, E. L., Barnett, D. M., Friesen, D. K., & William, G. N. (1992, June). AUV path planning: an A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. In *Proceedings of the 1992 symposium on autonomous underwater vehicle technology* (pp. 79-84). IEEE.

8. Li, M., & Zhang, H. (2020, November). AUV 3D path planning based on A* algorithm. In *2020 Chinese Automation Congress (CAC)* (pp. 11-16). IEEE.

9. Fang, M. C., Wang, S. M., Mu-Chen, W., & Lin, Y. H. (2015). Applying the self-tuning fuzzy control with the image detection technique on the obstacle-avoidance for autonomous underwater vehicles. *Ocean Engineering*, *93*, 11-24.

10. Liu, X., Zhang, M., & Rogers, E. (2019). Trajectory tracking control for autonomous underwater vehicles based on fuzzy re-planning of a local desired trajectory. *IEEE Transactions on Vehicular Technology*, *68*(12), 11657-11667.

11. Bresciani, M., Ruscio, F., Tani, S., Peralta, G., Timperi, A., Guerrero-Font, E., ... & Costanzi, R. (2021). Path planning for underwater information gathering based on genetic algorithms and data stochastic models. *Journal of Marine Science and Engineering*, *9*(11), 1183.

12. Hao, K., Zhao, J., Li, Z., Liu, Y., & Zhao, L. (2022). Dynamic path planning of a three-dimensional underwater AUV based on an adaptive genetic algorithm. *Ocean Engineering*, *263*, 112421.

13. Lim, H. S., Fan, S., Chin, C. K., Chai, S., Bose, N., & Kim, E. (2019). Constrained path planning of autonomous underwater vehicle using selectively-hybridized particle swarm optimization algorithms. *IFAC-PapersOnLine*, *52*(21), 315-322.

14. Huang, H., & Jin, C. (2021). A novel particle swarm optimization algorithm based on reinforcement learning mechanism for AUV path planning. *Complexity*, *2021*(1), 8993173.

15. Tian, Q., Wang, T., Song, Y., Wang, Y., & Liu, B. (2023). Autonomous Underwater Vehicle Path Tracking Based on the Optimal Fuzzy Controller with Multiple Performance Indexes. *Journal of Marine Science and Engineering*, *11*(3), 463.

16. Jian, X., Zou, T., Vardy, A., & Bose, N. (2020, September). A hybrid path planning strategy of autonomous underwater vehicles. In *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)* (pp. 1-6). IEEE.

17. Ma, H., Mu, X., & He, B. (2021). Adaptive navigation algorithm with deep learning for autonomous underwater vehicle. *Sensors*, *21*(19), 6406.

18. Jian, X., Zou, T., Vardy, A., & Bose, N. (2020, September). A hybrid path planning strategy of autonomous underwater vehicles. In *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)* (pp. 1-6). IEEE.

19. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). Robot operating system 2: Design, architecture, and uses in the wild. *Science robotics*, *7*(66), eabm6074.

20. Scholl, P. M., Brachmann, M., Santini, S., & Van Laerhoven, K. (2014). Integrating wireless sensor nodes in the robot operating system. *Cooperative Robots and Sensor Networks 2014*, 141-157.

21. Cousins, S. (2010). Welcome to ros topics [ros topics]. *IEEE Robotics & Automation Magazine*, *17*(1), 13-14.

22. Qian, W., Xia, Z., Xiong, J., Gan, Y., Guo, Y., Weng, S., ... & Zhang, J. (2014, December). Manipulation task simulation using ROS and Gazebo. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)* (pp. 2594-2598). IEEE.

23. Hobson, B. W., Bellingham, J. G., Kieft, B., McEwen, R., Godin, M., & Zhang, Y. (2012, September). Tethys-class long range AUVs-extending the endurance of propeller-driven cruising AUVs from days to weeks. In *2012 IEEE/OES autonomous underwater vehicles (AUV)* (pp. 1-8). IEEE.

24. Nusrat, I., & Jang, S. B. (2018). A comparison of regularization techniques in deep neural networks. *Symmetry*, *10*(11), 648.

25. Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

26. Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). Understanding batch normalization. *Advances in neural information processing systems*, *31*.

27. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

28. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, *37*(9), 1904-1916.

29. Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

30. Xu, H., Caramanis, C., & Mannor, S. (2008). Robust regression and lasso. *Advances in neural information processing systems*, *21*.

31. Zanini, Elena. "Markov decision processes." (2014).

32. Kubalík, J., Derner, E., Žegklitz, J., & Babuška, R. (2021). Symbolic regression methods for reinforcement learning. *IEEE Access*, *9*, 139697-139711.

33. Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society*, *60*(6), 503-515.

34. Yu, H., Mahmood, A. R., & Sutton, R. S. (2018). On generalized bellman equations and temporal-difference learning. *Journal of Machine Learning Research*, *19*(48), 1-49.

35. Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, *3*, 9-44.

36. Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*, 279-292.

37. Pieters, M., & Wiering, M. A. (2016, December). Q-learning with experience replay in a dynamic environment. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1-8). IEEE.

38. Shalev-Shwartz, S. (2012). Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, *4*(2), 107-194.

39. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237-285.

40. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

41. Van Hasselt, H., Guez, A., & Silver, D. (2016, March). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).

42. Fujimoto, S., Hoof, H., & Meger, D. (2018, July). Addressing function approximation error in actor-critic methods. In *International conference on machine learning* (pp. 1587-1596). PMLR.

43. Sun, B., Song, M., Li, A., Zou, N., Pan, P., Lu, X., ... & Kong, X. (2023). Multi-objective solution of optimal power flow based on TD3 deep reinforcement learning algorithm. *Sustainable Energy, Grids and Networks*, *34*, 101054.

44. Qiu, C., Hu, Y., Chen, Y., & Zeng, B. (2019). Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications. *IEEE Internet of Things Journal*, *6*(5), 8577-8588.

45. Gözütok, E., & Duran, F. (2024). Su Altı Otonom Araçlarda Derin Q-Ağları Algoritması Kullanılarak ROS Tabanlı Yol Planlama. *Gazi University Journal of Science Part C: Design and Technology*, 12(2)

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı :GÖZÜTOK, Emre

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Bilişim Sistemleri	Devam Ediyor
Lisans	Atılım Üniversitesi / Elektrik Elektronik Mühendisliği Mühendisliği	2020

İş Deneyimi

Yıl	Yer	Görev
2022-2024	ASAP Engineering GMBH	Yazılım Mühendisi

Yabancı Dil

İngilizce, Almanca

Yayınlar

Gözütok, E., & Duran, F. (2024). Su Altı Otonom Araçlarda Derin Q-Ağları Algoritması Kullanılarak ROS Tabanlı Yol Planlama. Gazi University Journal of Science Part C: Design and Technology, 12(2)

Hobiler

Yüzme

GAZİLİ OLMAK AYRICALIKTIR.