



실시간 항공영상 기반 UAV-USV 간 협응 유도·제어 알고리즘 개발

김도균¹·김정현¹·손희훈¹·최시웅¹·김동한²·여찬영³·박종용^{1,2,†}
국립부경대학교 마린융합디자인공학과 조선해양공학전공¹
국립부경대학교 조선해양시스템공학과²
한화오션(주) 해양제품연구팀³

A Study on a Real-Time Aerial Image-Based UAV-USV Cooperative Guidance and Control Algorithm

Do-Kyun Kim¹·Jeong-Hyeon Kim²·Hui-Hun Son¹·Si-Woong Choi¹·Dong-Han Kim²·Chan Young Yeo³·Jong-Yong Park^{1,2,†}
Department of Marine Design Convergence Engineering Pukyong National University¹
Department of Naval Architecture and Marine System Engineering Pukyong National University²
Hanwha ocean R&D Institute³

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper focuses on the cooperation between Unmanned Aerial Vehicle (UAV) and Unmanned Surface Vessel (USV). It aims to develop efficient guidance and control algorithms for USV based on obstacle identification and path planning from aerial images captured by UAV. Various obstacle scenarios were implemented using the Robot Operating System (ROS) and the Gazebo simulation environment. The aerial images transmitted in real-time from UAV to USV are processed using the computer vision-based deep learning model, You Only Look Once (YOLO), to classify and recognize elements such as the water surface, obstacles, and ships. The recognized data is used to create a two-dimensional grid map. Algorithms such as A* and Rapidly-exploring Random Tree star (RRT*) were used for path planning. This process enhances the guidance and control strategies within the UAV-USV collaborative system, especially improving the navigational capabilities of the USV in complex and dynamic environments. This research offers significant insights into obstacle avoidance and path planning in maritime environments and proposes new directions for the integrated operation of UAV and USV.

Keywords : UAV(Unmanned Aerial Vehicle, 무인항공기), USV(Unmanned Surface Vessel, 무인수상선), ROS(Robot Operating System, 로봇 운영체제), YOLO(You Only Look Once, 딥러닝 모델), Path planning(경로 계획)

1. 서론

최근 무인기동체의 관심이 높아짐에 따라 무인기동체의 충돌 회피, 경제적 운항, 자동 도킹, 이기종간의 협업에 대한 연구가 활발히 수행되고 있다. USV(Unmanned Surface Vessel)와 UAV(Unmanned Aerial Vehicle)의 협업은 다양한 미션과 환경에서 작업 효율성과 범위를 확장시키는 중요한 역할을 한다. 해상과 공중이라는 서로 다른 영역에서 동시에 작업을 수행함으로써, 넓은 지역을 신속하게 모니터링하고, 정밀한 데이터를 수집할 수 있다. 예를 들어, 해상 재난상황이 발생하였을 때 USV는 해상의 환경 모니터링이나 구조 작업을 수행할 수 있으며, 동시에 UAV

는 공중에서 더 넓은 시야를 제공하여 재난상황과 요구조건을 파악할 수 있다. 이러한 협업을 통해, 단일 무인기동체만으로는 수행하기 어려운 환경에서 신속하고 안전하게 임무를 수행할 수 있으며, 인간의 위험 노출을 최소화하고, 작업의 정확도를 향상시킬 수 있다. 결론적으로, USV와 UAV의 협업은 복잡하고 도전적인 임무 수행에 있어서, 효율성, 안전성, 유연성을 제공하는 필수적인 방법이다. 무인기동체의 협업에 관한 연구는 다음과 같다. 군집 USV와 UAV의 경로 계획 연구 (Hou et al., 2021), 시뮬레이션 상에서의 USV-UAV 시스템의 시각적 항법과 강화 학습 기반의 USV 제어 전략 개발에 관한 연구 (Wang et al., 2023), USV-UAV를 통한 협력 궤적 계획 알고리즘 개발에 관한 연구

(Huang et al., 2023), UAV, USV 두 이동체 모두 로봇운영체제를 이용하여 UAV가 임무 수행 후 USV에 착륙하는 연구 (Weaver et al., 2013) 등이 수행된 바 있다. Martinez (2022)은 호버링중인 UAV의 단안 카메라만을 통해 마커인식 기반 동적 USV의 자세추정 시스템을 설계하였다. Li et al. (2022a)은 외란이 있는 해상환경에서 UAV가 USV에 자율착륙 하기 위해 컴퓨터 비전 인식 기반 양방향 신경망인 LSTM(Bi-LSTM)으로 USV 자세를 추정하는 방식을 제안하였다. Shao et al. (2019)은 UAV, USV 간의 협력을 위한 새로운 플랫폼을 제시하며 UAV의 안정적인 착륙을 위한 경로생성 방법을 제시하였다. Wei et al. (2022)은 수중 목표 추적을 위해 UAV, USV, UUV(Unmanned Underwater Vehicle) 간의 협력을 위한 네트워크 시스템을 제안하였다. Liu et al. (2021)은 스마트 해양 네트워크에서 정보 수집 및 모니터링의 효율성을 개선하기 위해 UAV와 USV의 협력적 임무 할당 방안을 제시하였다. Li et al. (2023)은 3차원 매핑 가이던스와 적응형 퍼지 제어 알고리즘을 이용하여 USV와 UAV의 새로운 협력 설계 방안을 제시하였다. Dufek and Murphy (2016)은 색상 임계값과 윤곽 검출에 기반한 방법, 히스토그램 역투영 및 Camshift 알고리즘을 사용하여 시각적 위치 추정 솔루션을 제시하였다. Li et al. (2022b)은 동적 표면 제어(DSC), 기저함수 신경망(RBF-NNs) 그리고 트리거 기법을 활용하여 강인한 협력 제어 알고리즘을 설계하였다.

본 연구는 UAV-USV 협업을 통해 장애물 식별 및 최적 경로 생성에 중점을 둔 접근 방식을 제시한다. 기존 연구들은 주로 UAV와 USV의 궤적 계획, 자동 도킹 그리고 네트워크 시스템에 초점을 맞춘 반면, 본 연구는 실시간 항공 이미지를 활용하여 Redmon et al. (2016)이 제시한 컴퓨터 비전 기반의 딥러닝 모델인 YOLO(You Only Look Once)를 사용하여 장애물을 분류하고 인식하는 방법을 채택하였다. 인식된 장애물 데이터를 기반으로 복잡하고 동적인 해상 환경에서 USV의 운항 능력을 향상시키는 경로 계획과 유도, 제어 기법을 사용하였다. 특히, 본 방법은 해상에서 이동하는 선박, 부유물과 같은 동적 장애물에 효과적으로 대응할 수 있으며, 광범위한 지역 모니터링이 필요한 재난 구호, 환경 보호 등의 상황에서 빠르게 모니터링하고 정밀한 데이터를 수집할 수 있다. ROS(Robot Operating System)을 통해 USV와 UAV의 상호정보를 통신하였고, Gazebo를 사용하여 다양한 장애물 시나리오를 구축하여 이론적 모델과 실용적 응용 사이의 격차를 좁히고 UAV-USV 통합 운용에 있어 해상 환경에서의 자율 항해 시스템 발전에 새로운 방향을 제시한다. 본 연구에서는 실시간 이미지 처리와 컴퓨터 비전을 결합하여 무인 이동체 간 협업을 통한 장애물 인식 및 경로 계획 가능성을 탐구하였으며, 생성된 경로를 추종하는 유도, 제어 알고리즘을 검증하였다.

2. 시뮬레이션 환경 구성 및 동역학 모델

2.1 로봇운영체제(ROS, Robot Operating System)

본 연구에서 시뮬레이션 환경을 구축하고, 무인 이동체의 실시간 통신 및 유도, 제어를 위해 ROS의 최신 버전인 ROS2를 사용

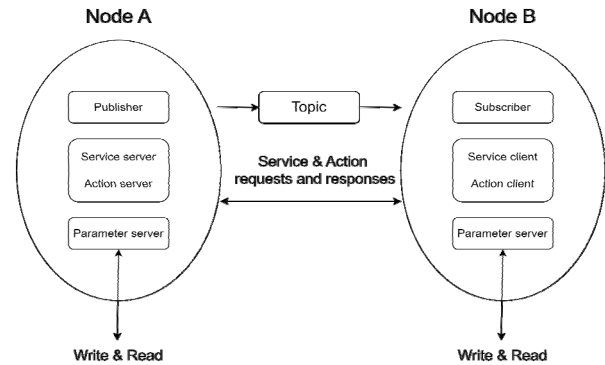


Fig. 1 ROS2 data communication structure

하였다. ROS는 오픈 소스 메타 운영체제로, 로봇 응용 소프트웨어 개발을 위한 프레임워크이다. 다양한 로봇 부품 및 다기종 디바이스 간의 통합과 효율적인 메시지 교환을 가능하게 하는 미들웨어 역할을 수행한다. ROS2는 기존의 ROS 마스터 시스템 대신 DDS (Data Distribution Service)를 통해 통신하여 실시간 데이터 전송, 불안정한 네트워크 환경에 대한 대응 및 보안 기능을 강화하였다. Fig. 1은 ROS2의 데이터 통신 구조를 가시화한 것이다.

2.2 Gazebo를 이용한 시뮬레이션 환경 구축

시뮬레이션 환경 구축을 위해 오픈소스 기반의 로봇 시뮬레이터인 Gazebo를 사용하였다. Gazebo는 OpenGL 기반의 고성능 3D 로봇 시뮬레이터로, 다양한 물리 엔진을 지원하고 복잡한 로봇 모델과 환경을 시뮬레이션할 수 있는 툴을 제공한다. Camera, LiDAR(Light Detection And Ranging), GPS(Global Positioning System), IMU(Inertia Measurement Unit) 등 다양한 가상 센서를 지원하여, 로봇의 동작을 효과적으로 시뮬레이션할 수 있다. ROS2를 사용하여 개발된 로봇 애플리케이션을 Gazebo에서 직접 테스트하고 평가할 수 있으며, 로봇 시스템의 개발과 테스트 과정을 효율적으로 만들어준다. Fig. 2는 Gazebo를 통한 시뮬레이션의 예시를 보여주는 그림이다.

본 연구에서는 해상 환경 구성과 로봇의 구현을 위해 VRX (Virtual RobotX)가 제공하는 오픈소스 패키지를 활용하였다. VRX는 다양한 해양 로봇 시뮬레이션을 위한 툴킷을 제공하며, 복잡한 해상 환경을 효과적으로 구현할 수 있다. VRX에서 제공하는 리소스를 사용하여 Fig. 4와 같은 해상 환경을 구축하였다. 항공기의 고도와 카메라의 수직 및 수평 시야각을 삼각법을 활용하여 비춰지는 영역의 가로 및 세로 길이를 추정하였다. Fig. 3은

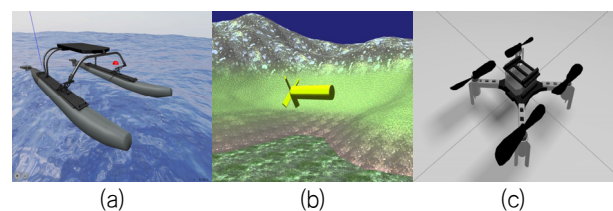


Fig. 2 Simulation example with Gazebo, USV marine simulation(a), UUV underwater simulation(b), UAV simulation(c)

Table 1 Parameters for calculating the size of the projected area

Parameter	Value
Altitude	100 [m]
Horizontal viewing angle	1.2 [rad]
Vertical viewing angle	0.693 [rad]

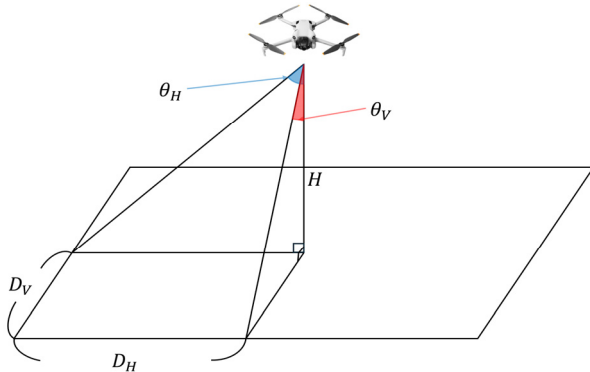


Fig. 3 Projection area of an image taken from a UAV

UAV를 통해 촬영된 이미지의 투영 영역을 나타낸다. 항공기의 고도(H), 카메라의 수직 시야각, 수평 시야각이 표시되어 있다. 삼각법을 이용하여 지면에 투영된 이미지의 수직 길이와 수평 길이를 계산할 수 있다. 수평 길이는 UAV 고도와 카메라 수평 시야각을 이용하여 식 1과 같이 계산되며 수직 길이는 고도와 수직 시야각을 이용하여 식 2와 같이 계산된다. Table 1은 UAV의 고도정보와 수평, 수직 시야각을 나타낸 것이다.

$$\tan(\theta_H) = \frac{D_H}{H} \tag{1}$$

$$\tan(\theta_V) = \frac{D_V}{H} \tag{2}$$

UAV 모델의 경우, PX4 Autopilot 시뮬레이션에서 사용되는 드론 모델을 채택하였다. PX4는 무인항공기를 위한 오픈 소스 비행 제어 소프트웨어 플랫폼으로, 다양한 드론 모델을 제공한다.

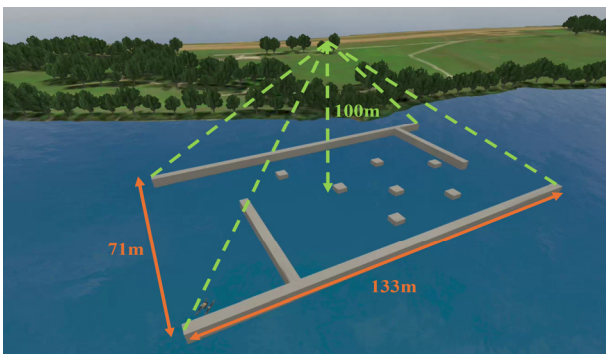


Fig. 4 Maritime simulation environment implemented with Gazebo

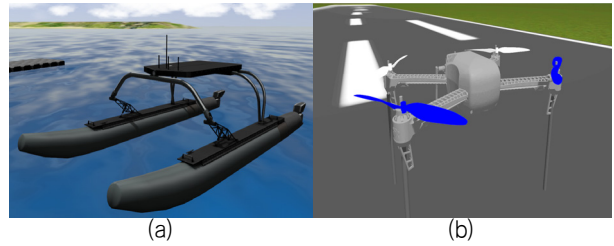


Fig. 5 Dynamics models used in simulation, WAM-V(a), IRIS+(b)

Table 2 Parameter of dynamics model(WAM-V)

USV parameter	Value
Hull length	4.9 [m]
Hull radius	0.213 [m]
Breadth	2.5 [m]
Draft	0.3 [m]
weight	151 [kg]
Moment of inertia	257 [kg·m]

Table 3 Parameter of dynamics mode(IRIS+)

UAV parameter	Value
Motor to motor dimension	0.55 [m]
Height	0.1 [m]
Weight	1282 [g]
Average flight time	16~22 [min]
Propellers	9.5 x 4.5 [cm]
Motors	920 [Kv]

USV 모델은 Woo et al. (2018)에서 사용된 WAM-V모델을 사용하였으며, UAV의 동역학 모델로는 IRIS+ 가 사용되었다. Fig. 5는 시뮬레이션 상에서 보여지는 WAM-V와 IRIS+이며 Table 2은 WAM-V 동역학 모델의 파라미터를 나타낸 것이고 Table 3는 IRIS+ 동역학 모델의 파라미터를 나타낸 것이다.

3. 협응 유도·제어 알고리즘

3.1 항공 영상을 활용한 장애물 인식

전역 경로 생성을 위해서는 인식된 장애물을 고려한 정확한 전역지도가 필수적이다. 이를 위해 사용되는 주요 방법으로는 LiDAR를 이용한 SLAM(Simultaneous Localization and Mapping) 알고리즘과 상공 이미지를 활용한 방법이 있다. 해양 환경 특성상, USV에 장착된 LiDAR를 사용하여 전역지도를 생성하는 것은 큰 파도나 물결 등의 영향으로 센서 데이터에 혼선을 초래할 수 있다. 특히 후에 동적 장애물을 추가할 경우, 이러한 혼선 문제를 해결하기 위해 별도의 데이터 처리가 필요할 수 있다. 따라서 본 연구에서는 UAV가 상공에서 호버링하며 촬영된

항공 영상을 기반으로 전역지도를 생성하였다. 해상 환경 내 장애물을 실시간으로 탐지하기 위해 원스테이지 방식의 딥러닝 모델인 YOLO를 활용하였다. 본 연구에서는 현재 가장 최신 버전인 Yolov8 모델을 사용하여 객체를 인식하였다. YOLO는 입력 이미지를 여러 그리드로 나누고, 각 그리드에 대한 객체의 경계박스(Bounding Box, Bbox)와 클래스 예측을 단 한 번의 신경망 패스를 통해 수행하는 원스테이지 방식의 딥러닝 모델이다. 객체 탐지 후 처리 과정에서는 IOU(Intersection Over Union)를 계산하여 예측된 경계박스와 실제 객체의 경계박스가 얼마나 겹치는지를 평가하고 NMS(Non-Maximum Suppression)를 적용하여 여러 겹쳐진 예측 상자 중 가장 높은 확률을 가진 상자를 유지하며 하나의 객체에 대해 여러 개의 경계상자가 생성되는 것을 방지한다. 이러한 과정을 통해 모델의 정확도를 높이며 장애물을 효과적으로 탐지할 수 있다. Fig. 6은 YOLO를 통해 객체를 검출하는 과정을 나타낸 그림이다.

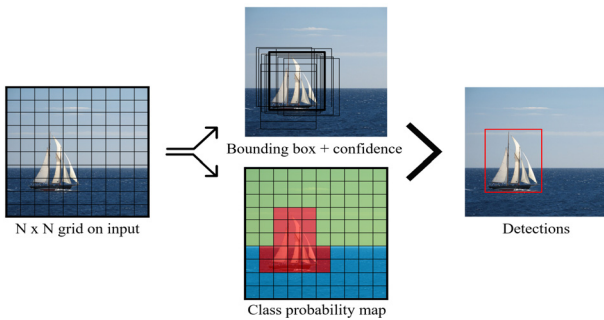


Fig. 6 The process of detecting objects with YOLO

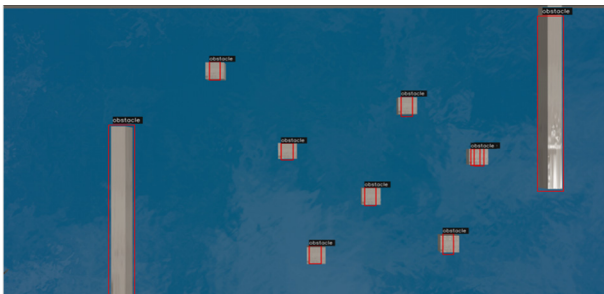


Fig. 7 Detect obstacles by receiving video from a UAV hovering over a Gazebo

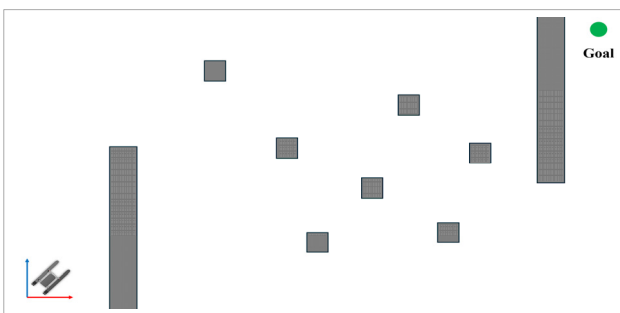


Fig. 8 Two-dimensional map based on detected obstacle coordinates

검출된 장애물의 픽셀상에서의 좌표와 크기를 바탕으로 2차원 그리드 맵을 생성하였다. Fig. 7은 해상 환경에서의 장애물을 검출한 모습을 나타낸 그림이고 Fig. 8은 검출한 장애물의 위치와 크기를 기반으로 2차원 맵을 나타낸 그림이다.

3.2 경로생성 알고리즘

본 연구에서는 장애물을 위한 경로계획을 위해 최적경로계획 알고리즘 중 Hart et al. (1968)이 제시한 그래프 탐색기반 A* 알고리즘과 Karaman and Frazzoli (2011)이 제시한 랜덤 샘플링 기반 RRT*(Rapidly-exploring Random Tree) 알고리즘을 선정하였다. A*알고리즘은 휴리스틱 함수를 통해 효율적인 탐색을 가능하게 하며, 최적의 경로를 보장한다는 장점이 있다. 이는 2D 그리드 맵과 같은 명확한 경로 탐색 문제에서 신뢰할 수 있는 결과를 제공한다. RRT*알고리즘은 고차원 공간 및 복잡한 환경에서 샘플링 기반 접근 방식을 통해 효과적으로 경로를 탐색하며, 시간이 지남에 따라 최적 경로로 수렴하는 특성을 지닌다. 또한 다양한 환경에서의 경로 탐색 문제를 효과적으로 해결할 수 있어 두 알고리즘을 선정하였다. 다음은 선정한 알고리즘에 대해 자세하게 기술하겠다. A* 알고리즘은 주어진 공간에 대해 그리드를 생성하고, 그리드 상에서 최단 경로를 탐색하여 결정한다. 현재 위치로부터 8가지 방향으로의 이동 가능성을 고려하여 각 방향에 대한 이동 비용을 계산한다. 장애물이 있는 그리드는 1로 표시되어 이동할 수 없음을 나타내고, 그리드가 0으로 표시되면 이동이 가능함을 나타낸다.

본 연구에서는 YOLO를 통해 인식된 장애물 좌표를 이용하여 생성된 그리드맵에 투영하였다. 구체적으로, 인식된 장애물의 좌표가 그리드에 포함되면 해당 그리드를 장애물이 있는 그리드로 표시하였다. 이를 통해 생성된 경로의 안정성을 보장하였다. 이러한 방식은 장애물의 정확한 위치를 반영함으로써 경로 계획 시 장애물 회피 능력을 향상시키고, USV가 안전하게 주행할 수 있도록 하였다.

식 (3)은 A* 알고리즘에서 사용하는 평가함수를 구하는 공식이다. $g(n)$ 은 현재 노드에서 바로 다음 노드 n 까지의 경로비용을 나타내고 $h(n)$ 은 노드 n 부터 목표노드까지의 추정비용을 나타낸다. 비용함수는 그리드의 거리공식으로 계산하며 계산된 비용을 기반으로 시작점부터 목적지까지의 최적 경로를 찾는다. 그리드의 크기는 해상도와 계산 복잡도 간의 트레이드오프를 나타내는데, 더 작은 그리드는 보다 정밀한 경로를 생성하지만 계산량이 증가하는 반면, 더 큰 그리드는 계산량이 줄어들지만 경로의 정밀도가 감소한다. Fig. 9는 A*알고리즘의 경로탐색과정을 가시화한 것이다.

$$f(n) = g(n) + h(n) \tag{3}$$

RRT 알고리즘은 로봇 및 무인기의 경로 계획에서 사용되는 탐색 알고리즘이다. 이 알고리즘은 시작 지점에서 목표 지점까지의 경로를 찾기 위해 무작위 샘플링을 통해 트리를 점진적으로

확장하는 방식으로 작동한다. 먼저, 시작 지점을 루트 노드로 하는 트리를 초기화하고, 작업 공간 내에서 무작위로 한 점을 샘플링한다. 이 샘플링된 점과 가장 가까운 트리 노드를 찾은 후, 해당 노드에서 샘플링된 점으로 향해 한 단계 전진하여 새로운 노드를 생성한다. 생성된 노드는 장애물과 충돌하지 않는다면 트리에 추가된다. 이러한 과정을 반복하여 트리를 확장해 나가며, 새로운 노드가 목표 지점에 도달하면 경로 탐색을 완료한다. 이러한 과정을 통해 RRT 알고리즘은 고차원 및 비구조화된 환경에서도 효율적으로 경로를 찾을 수 있으며, 확률적 완전성을 보장하여 시간이 충분하다면 항상 경로를 발견할 수 있다. 또한 이 알고리즘은 비교적 단순하고 구현하기 쉬워 여러 분야에서 활용된다. 본 논문에서는 RRT의 변형 알고리즘인 RRT*을 사용하였다.

RRT* 알고리즘은 RRT 알고리즘의 확장 버전으로, 최적 경로를 찾기 위해 부모 노드의 재선정과 트리의 재구성을 수행한다. RRT*는 기본적인 RRT의 무작위 샘플링과 트리 확장 방식을 따르지만, 새로운 노드를 추가할 때 단순히 가장 가까운 기존 노드와 연결하는 대신, 일정 반경 내의 기존 노드들을 탐색한다. 이 반경 내에서 새로운 노드와의 연결 비용을 계산하여 비용이

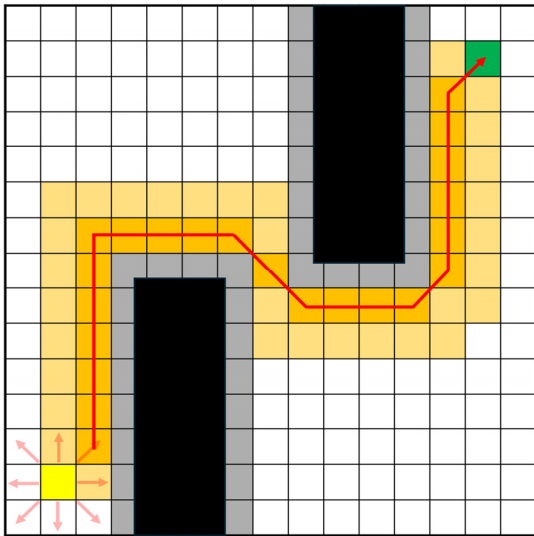


Fig. 9 Visualizing the path planning process with the A* algorithm

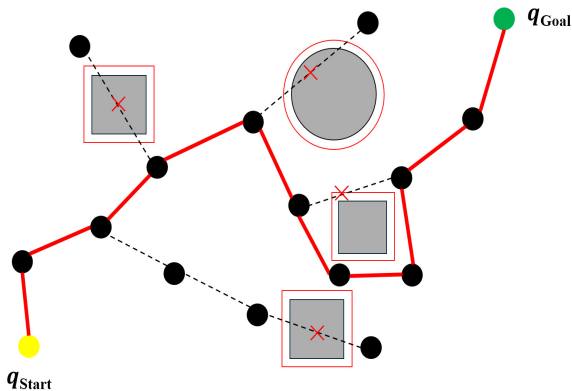


Fig. 10 Visualizing the path planning process with the RRT algorithm

최소가 되는 노드를 새로운 노드의 부모로 재선정한다. 이를 통해 경로의 비용을 줄이고, 더 효율적인 경로를 생성한다.

또한, 새로운 노드가 추가된 후에는 일정 반경 내의 기존 노드들이 새로운 노드를 통해 연결될 경우 비용이 더 낮아지는지 확인하고, 낮아진다면 기존 노드들의 부모를 새로운 노드로 변경한다. 이러한 과정을 반복하여 트리를 재구성함으로써 경로의 최적화를 지속적으로 수행한다. Fig. 10은 RRT 알고리즘을 통해 생성된 경로를 가시화한 것이며, Fig. 11은 RRT* 알고리즘의 부모 노드 재선정 과정과 트리의 재구성 과정을 그림으로 나타낸 것이다. RRT*는 이러한 추가적인 과정을 통해 최적의 경로를 찾도록 보장한다. RRT* 알고리즘의 전체적인 흐름도는 Fig. 12와 같다.

A*와 RRT* 알고리즘들은 점과 점 사이의 기하학적인 관계만을 활용하여 경로를 계획하기 때문에, 최적의 경로를 도출했다 하더라도 이동체가 해당 경로를 추종하는 과정에서 장애물과 충돌할 위험이 있다. 이동체가 안전하게 이동할 수 있도록 Fig. 8, Fig. 10과 같이 장애물 주변에 충분한 안전 거리를 설정하였다. 안전 거리는 USV의 반쪽으로 설정한 후 경로 계획을 수행하였으며 안전 거리 내에는 경로가 생성되지 않도록 함으로써 장애물 회피 능력을 향상시켰다.

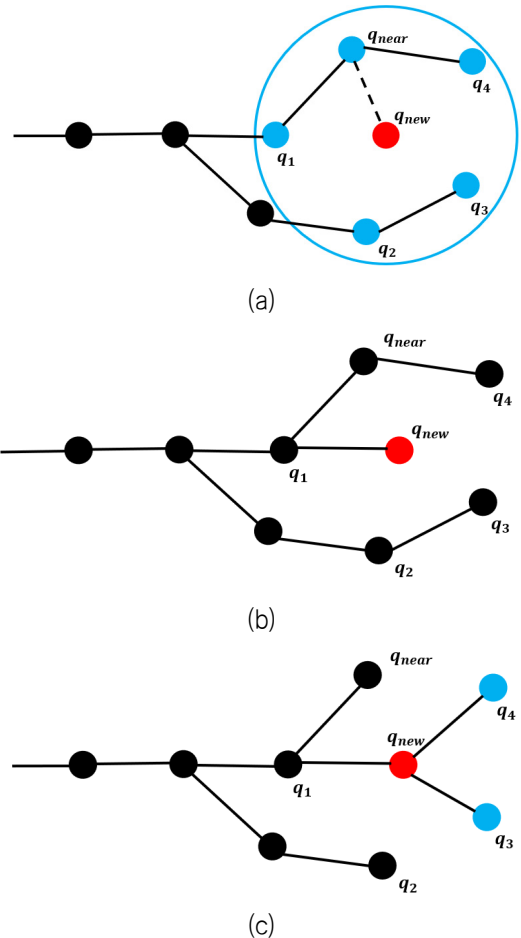


Fig. 11 Visualization of the process of detecting neighbors in the radius of the RRT* algorithm(a), reselecting the parent node(b), and reconstructing the tree(c).

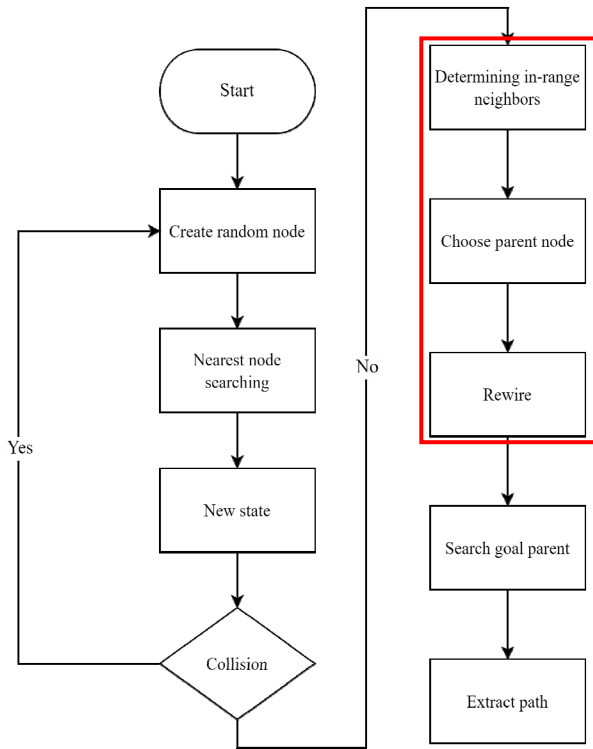


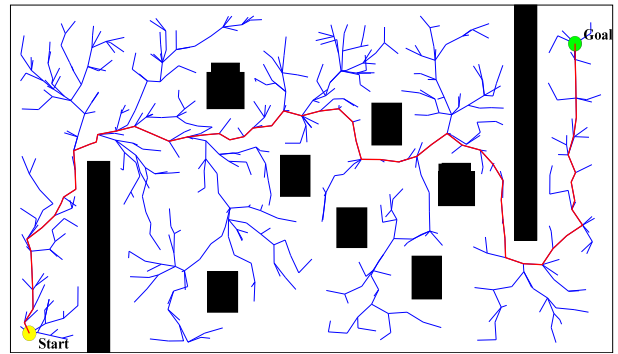
Fig. 12 Flowchart of the RRT* algorithm used (red square processes are processes that only exist in the RRT* algorithm and not in the RRT algorithm).

A*와 RRT* 알고리즘은 경로 계획을 수행할 때 각각의 파라미터 설정이 매우 중요하다. A* 알고리즘의 경우 그리드의 크기 (cell size) 결정은 알고리즘의 성능과 정밀도에 결정적인 영향을 미친다. RRT* 알고리즘의 경우 반복 횟수(max iteration), 노드 간 최대거리(step size), 목표 샘플링 확률(goal sample) 등의 파라미터 설정이 중요하다. 앞서 설명한 파라미터들은 특정 환경에 적합하게 조정되어야 하며, 경로의 품질과 계산 시간에 영향을 줄 수 있다.

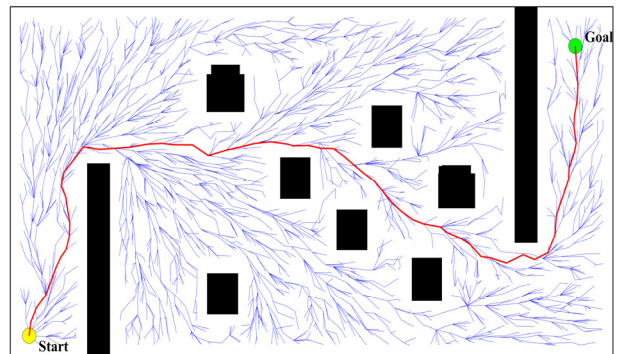
본 연구에서는 RRT* 알고리즘의 효율적인 실행을 위해 적절한 반복 횟수의 결정이 중요하다는 것을 강조한다. 반복 횟수에 따른 코스트의 변화를 분석하여, 코스트의 변화량에 대한 표준편차값을 기준으로 최적의 반복 횟수를 결정하는 방법을 도입하였다. 반복 횟수를 증가시킬 때 코스트가 수렴하는 경향을 확인하였으며, 이때의 코스트 값의 변화율을 통해 표준편차를 계산한다. 표준편차가 임계값보다 작아지는 지점의 반복 횟수를 최적값으로 선정하였으며, 이는 코스트 값의 변화가 작다는 것을 나타내는 지표로 활용되었다. Fig. 13은 주어진 환경에서 다양한 반복 횟수에 따른 경로 탐색 결과의 변화를 시각적으로 보여준다. 반복 횟수를 50,000으로 설정하고 전체 알고리즘을 50회 반복 실행하여 얻은 코스트의 평균값과 코스트의 변화율을 통한 표준편차를 Fig. 14에 그래프로 나타내었다. 반복 횟수가 증가함에 따라 코스트 값이 특정 값에 수렴하는 경향을 보이며 표준편차값 역시 반복횟수가 높을수록 감소하는 경향을 보인다. 특정 임계값

보다 낮아지는 지점을 빨간색 점으로 표시하였고, 해당 지점의 반복 횟수를 채택하여 경로 계획을 진행하였다. 결정된 경로계획 알고리즘의 파라미터 값은 Table 4와 같다.

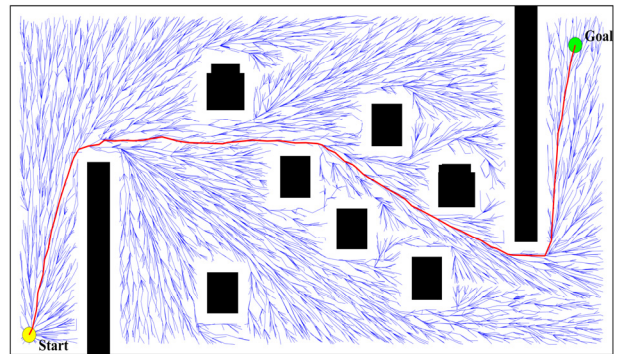
본 연구는 RRT* 알고리즘의 실행에 있어서 계산 효율성과 경로 최적화 사이의 균형을 찾는 데 중요한 기여를 하며, 결과적으로 탐색 공간에서의 더 나은 경로를 도출할 수 있음을 보여준다.



(a)



(b)



(c)

Fig. 13 RRT* algorithm shown after (a)1,500, (b) 5,000, (c) 15,000 iterations

Table 4 Path planning algorithm parameter

Algorithm	Parameter	Value
A*	Cell size	0.25
RRT*	Goal sample	0.1
	Step len	0.25
	Max iter	29900

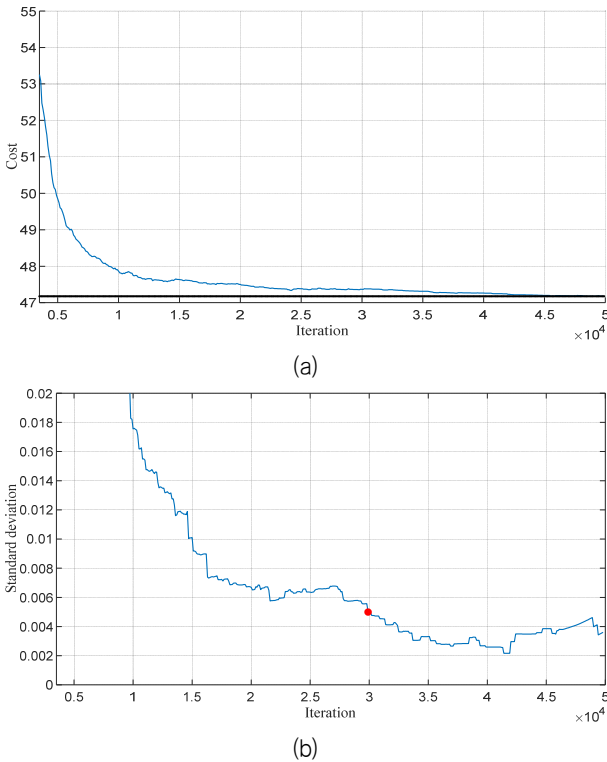


Fig. 14 In the given environment cluttered with obstacles, the result of applying the RRT* algorithm is presented, showing the average cost over 50 trials in (a), with the optimal cost indicated by a black dotted line. The variance is illustrated in (b).

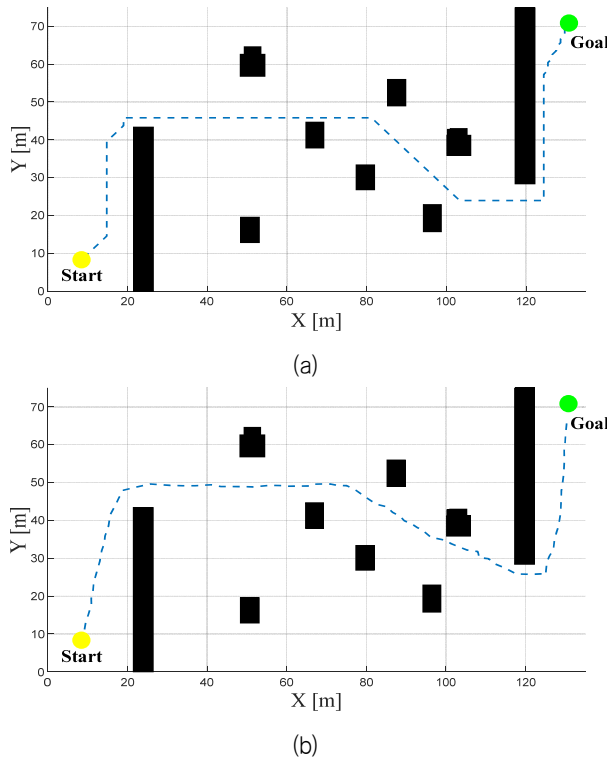


Fig. 15 Visualization of path planning in a given environment A*(a), RRT*(b)

본 연구에서는 Fig. 8과 같은 환경에서 RRT*, A* 2가지의 경로 생성 알고리즘을 구현하였다. Fig. 15은 주어진 환경에서 경로 생성 알고리즘을 통해 생성한 경로를 나타낸 그림이다.

3.3 경로 추종 알고리즘

본 연구에서는 USV에 장착된 GPS장치를 활용하여 위치 데이터를 수집하였다. GPS 데이터는 위도와 경도 형태로 제공되며, 연구 목적에 부합하게 하기 위해 UTM-K(Universal Transverse Mercator-Korea)좌표계로 변환하는 절차를 거쳤다. 변환 과정을 통해 얻어진 USV의 위치 데이터(x_s, y_s)는 목표 방위각(ψ_d)을 결정하는 데 중요한 역할을 하며, 이는 식 (4)를 통해 계산되었다.

경로 추종에 있어서는 LOS with CA(Line Of Sight with Circle Acceptance) 기법을 적용하였다. 이 기법은 목표 지점과 USV사이의 시야선을 기반으로 하여, USV의 주변에 설정된 반경 0.5미터의 가상 원을 활용한다. 경로와 원의 접점(x_w, y_w)에 따라 USV가 이동하도록 목표 방위각을 설정함으로써, USV가 경로를 효율적으로 추종할 수 있게 한다. Fig. 16는 USV가 생성된 경로를 추종하는 과정을 시각화한 것이다.

$$\psi_d = \tan^{-1} \left(\frac{y_w - y_s}{x_w - x_s} \right) \quad (4)$$

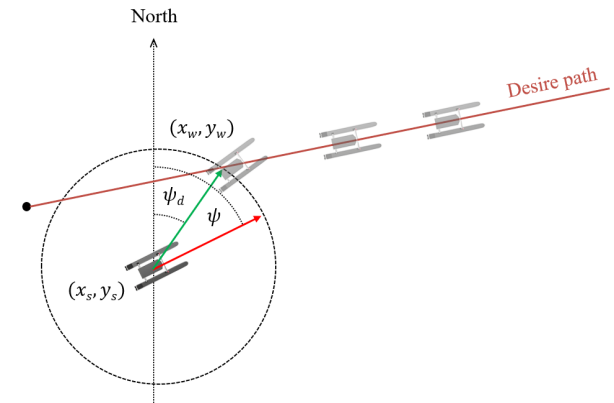


Fig. 16 Visualization waypoint tracking progress

3.4 제어 알고리즘

본 연구에서는 USV의 방향 제어를 위해 PD(Proportional-Derivative) 제어 기법을 사용하였다. 식 (5)에서 ψ_d 는 USV가 추종해야 할 목표 방위각을 나타내고, ψ 는 USV의 현재 상태 방위각을 나타낸다. 상태방위각과 목표방위각의 차이를 바탕으로, 식 (6)를 사용하여 K_p 와 K_d 를 통해 좌우현 추진기의 추력 차이를 계산하고, 방향 제어를 수행한다. PD제어의 게인 값은 다수의 시행착오를 통해 Table 5와 같이 선정하였으며 두 가지의 다른 경로 생성 방식의 효율성 비교를 위해 동일한 계인을 통해 시험을 진행하였다.

Table 5 Control gain of PD controller

Parameter	Value
K_p	5.0
K_d	10.0

$$e = \psi_d - \psi \tag{5}$$

$$\tau = K_p e + K_d \dot{e} \tag{6}$$

방향 제어를 위해 좌우현 추진기의 값을 식 (7),(8)을 사용하여 계산하였다. n_{stbd} 는 우현 추진기의 회전수를 나타내고 n_{port} 는 좌현 추진기의 회전수를 나타낸다. n_0 는 전진 및 후진을 위한 추진기의 기본 회전수를 나타낸다.

$$n_{stbd} = n_0 - \tau \tag{7}$$

$$n_{port} = n_0 + \tau \tag{8}$$

4. 알고리즘 결과 및 분석

본 연구에서 설계한 경로생성 알고리즘의 성능을 시뮬레이션 환경에서 비교 평가 하였다. 이를 위해 코스트(cost of length), 연산시간(computation time), 도달시간(time to reach), 곡률(curvature)등 다양한 지표들을 사용하여 경로의 성능을 평가하였다. 곡률은 곡선이나 경로의 방향 변화율을 측정하는 지표로, 경로가 직선으로부터 얼마나 벗어나는지를 나타내는 수치로 해석될 수 있다. 곡률이 높은 지점은 경로가 급격히 방향을 변경하는 부분을 의미하며, 낮은 곡률값은 경로가 상대적으로 평탄하거나 직선에 가까움을 나타낸다. 곡률 계산은 식 (9)에 따라, x', y' 는 각각 x, y 의 첫 번째 도함수, x'', y'' 는 두 번째 도함수를 나타낸다. Fig. 17는 제안된 경로 생성 알고리즘에 의해 생성된 경로의 곡률 변화를 시각화한 것으로, 임계 곡률 값을 초과하는 부분은 경로 상에서 빨간 점으로 표시되어 있다. 이러한 분석을 통해, 알고리즘의 성능을 종합적으로 평가하고, 특히 경로의 효율성 및 안정성 측면에서의 장단점을 식별할 수 있었다. Table 6에는 제안된 경로 생성 알고리즘의 성능을 평가하기 위해 여러 지표들을 포함시켰으며, RRT* 알고리즘은 특성상 연산 시간이 알고리즘의 랜덤성에 의해 달라질 수 있으므로, 연산 시간 지표는 상대적인 비교를 통해 순위 방식으로 표현하였다.

$$k = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{3/2}} \tag{9}$$

Fig. 18은 두 가지의 경로생성 알고리즘과 유도·제어 알고리즘을 통해 시험을 진행하였다. 각각의 시험을 통한 보트의 위치,

현재 및 목표 방위각, 방위각 오차, 속도, 좌·우현 추진기 회전수를 Fig. 18을 통해 확인할 수 있다.

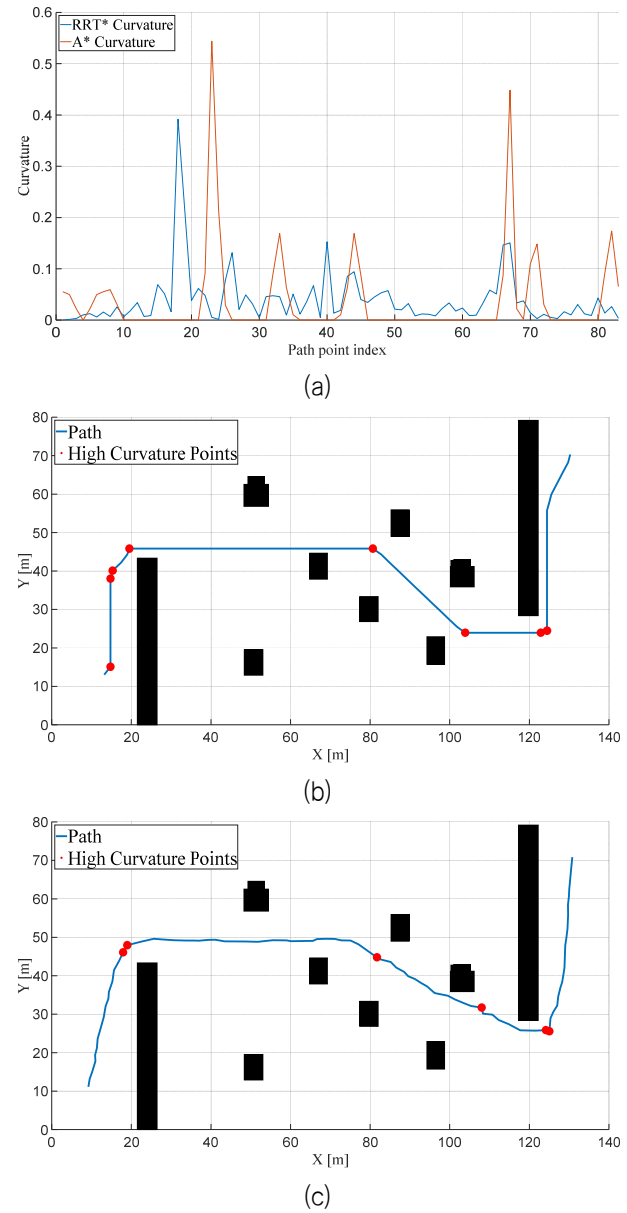


Fig. 17 Plot showing curvature for path planning algorithms, graph showing the magnitude of curvature per path(a), graph showing the bins where curvature is greater than a threshold for each path planning algorithm A*(b) & RRT*(c)

Table 6 Quantitative comparison of various path generation algorithms

Method	Cost of length [m]	Computation rating	Time to reach [sec]	Curvature (>0.1)
A*	217.74	1	651.0	8
RRT*	213.29	2	562.1	6

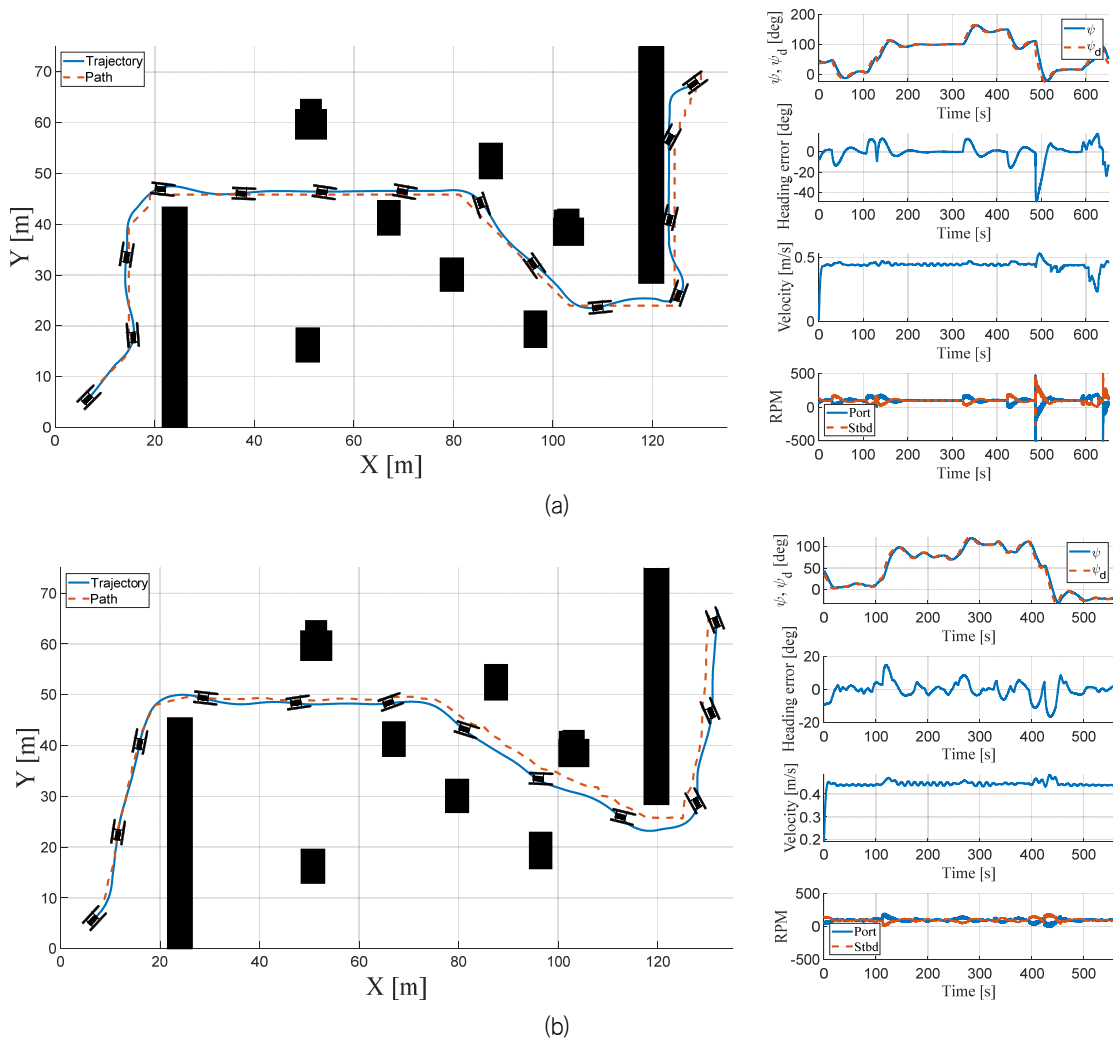


Fig. 18 The result of guiding and controlling the generated paths, (a) A*, (b) RRT*

5. 결론

본 연구에서는 Gazebo 시뮬레이션 툴을 사용하여 해상 환경을 구축하고, ROS2를 통해 USV의 실시간 통신 및 유도, 제어를 실행하였다. 장애물은 UAV의 항공영상을 실시간으로 수신하여 YOLO 딥러닝 모델을 통해 검출하였다. 경로 생성은 A*와 RRT* 알고리즘을 사용하여 수행되었다. 경로 추종에는 LOS with CA 기법을 적용하고, PD 제어 기법을 사용하였다. 본 연구에서 설계한 두 가지 경로계획 알고리즘의 성능은 코스트, 연산 시간, 도달 시간, 곡률 등의 정량적 지표를 통해 비교 분석되었다.

분석 결과, 각 알고리즘의 장단점과 특정 환경에서의 적응 가능성이 명확하게 드러났다. RRT* 알고리즘은 동적 환경에서의 적응성과 경로 최적화 측면에서 유리한 반면 A* 알고리즘은 계산 효율성과 예측 가능한 경로 생성에서 강점을 보였다. PD제어와 LOS with CA 기법의 적용은 USV의 경로 추종 성능을 효과적으로 개선하는 데 기여하였다.

본 연구에서 제시된 시스템은 동적 장애물과 정적 장애물 모

두에 대해 차별성과 효과를 발휘할 수 있다고 생각한다. 이번 논문에서는 정적 장애물에 대한 경로 생성과 추종을 잘 구현했지만, 향후 연구에서는 실제 환경에서 실시간 항공영상을 받아 동적 장애물과 정적 장애물을 모두 인식하고 회피할 수 있는 경로 생성 알고리즘을 개발할 계획이다. 또한 LiDAR와 카메라를 활용한 회피 알고리즘을 유도하고 제어하여 그 성능을 비교, 분석할 예정이다. 이를 통해 시스템의 실용성을 더 높이고자 한다.

후기

이 논문은 국립부경대학교 자율창의학술연구비(2022년)에 의하여 연구되었습니다.

References

Dufek, J. and Murphy, R., 2016. Visual pose estimation of USV from UAV to assist drowning victims recovery, in:

2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Lausanne, Switzerland, pp.147–153.

Hart, P., Nilsson, N. and Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4, pp.100–107.

Hou, Y., Tao, H., Gong, J., Liang, X. and Zhang, N., 2021. Cooperative path planning of USV and UAV swarms under multiple constraints. *Chinese Journal of Ship Research*, 16(1), pp.74–82.

Huang, T., Chen, Z., Gao, W., Xue, Z. and Liu, Y., 2023. A USV-UAV cooperative trajectory planning algorithm with hull dynamic constraints. *Sensors*, 23, 1845.

Karaman, S. and Frazzoli, E., 2011. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30, pp.846–894.

Li, W., Ge, Y., Guan, Z. and Ye, G., 2022a. Synchronized motion-based UAV-USV cooperative autonomous landing. *Journal of Marine Science and Engineering*, 10, pp.1214.

Li, J., Zhang, G. and Li, B., 2022b. Robust adaptive neural cooperative control for the USV-UAV based on the LVS-LVA guidance principle. *Journal of Marine Science and Engineering*, 10, 51.

Li, J., Zhang, G., Shan, Q. and Zhang, W., 2023. A novel cooperative design for USV-UAV systems: 3-D mapping guidance and adaptive fuzzy control. *IEEE Transactions on Control of Network Systems*, 10, pp.564–574.

Liu, J., Su, Z. and Xu, Q., 2021. UAV-USV Cooperative task allocation for smart ocean networks, in: *2021 IEEE 23rd Int Conference on High Performance Computing & Communications; 7th Int Conference on Data Science & Systems; 19th Int Conference on Smart City; 7th Int Conference on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, Haikou, Hainan, China, pp.1815–1820.

Martinez, M. A., 2022. *USV attitude position estimation by a hovering UAV using monocular images of deck-mounted lights*. Ph.D. University of California.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: unified, real-time object detection, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779–788.

Shao, G., Ma, Y., Malekian, R., Yan, X. and Li, Z., 2019. A novel cooperative platform design for coupled USV-UAV systems. *IEEE Transactions on Industrial Informatics*, 15, pp.4913–4922.

Wang, Y., Liu, W., Liu, J. and Sun, C., 2023. Cooperative USV-UAV marine search and rescue with visual navigation and reinforcement learning-based control. *ISA Transactions*, 137, pp.222–235.

Weaver, J. N., Frank, D. Z., Schwartz, E. M. and Arroyo, A. A., 2013. UAV performing autonomous landing on USV utilizing the robot operating system. *Proceedings of the ASME District F-Early Career Technical Conference*, 12, pp.119–124.

Wei, W., Wang, J., Fang, Z., Chen, J., Ren, Y. and Dong, Y., 2022. 3U: Joint design of UAV-USV-UUV networks for cooperative target hunting. *IEEE Transactions on Vehicular Technology*, 72, pp.4085–4090.

Woo, J.H., Park, J.Y., Yu, C.W. and Kim, N.W., 2018. Dynamic model identification of unmanned surface vehicles using deep learning network. *Applied Ocean Research*, 78, pp.123–133.

