



로봇운영체제를 이용한 보트의 자율운항 알고리즘 개발

조현재¹·김정현¹·김수림¹·우주현²·박종용^{3,†}
부경대학교 마린융합디자인공학과¹
한국해양대학교 조선해양시스템공학부²
부경대학교 조선해양시스템공학과³

Development of Autonomous Algorithm for Boat Using Robot Operating System

Hyun-Jae Jo¹·Jung-Hyeon Kim¹·Su-Rim Kim¹·Ju-Hyun Woo²·Jong-Yong Park^{3,†}
Department of Marine Design Convergence Engineering, Pukyong National University¹
Naval Architecture and Ocean Systems Engineering, Korea Maritime and Ocean University²
Department of Naval Architecture and Marine System Engineering, Pukyong National University³

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.

According to the increasing interest and demand for the Autonomous Surface Vessels (ASV), the autonomous navigation system is being developed such as obstacle detection, avoidance, and path planning. In general, autonomous navigation algorithm controls the ship by detecting the obstacles with various sensors and planning path for collision avoidance. This study aims to construct and prove autonomous algorithm with integrated various sensor using the Robot Operating System (ROS). In this study, the safety zone technique was used to avoid obstacles. The safety zone was selected by an algorithm to determine an obstacle-free area using 2D LiDAR. Then, drift angle of the ship was controlled by the propulsion difference of the port and starboard side that based on PID control. The algorithm performance was verified by participating in the 2020 Korea Autonomous BOAT (KABOAT).

Keywords : Robot Operating System(ROS, 로봇운영체제), Light Detection And Ranging(LiDAR, 라이다), Proportional Integral Derivation Control(PID 제어), Autonomous driving(자율주행), Obstacle avoidance(장애물 회피), Safety zone(안전구역)

1. 서론

최근 EU MUNIN(Maritime Unmanned Navigation through Intelligence in Networks) 프로젝트의 무인 원격제어 벌크선, Rolls-Royce의 무인 컨테이너 선박, DnV-GL의 컨테이너 선박과 같이 사람의 개입 없이 자율적으로 선박을 제어하기 위한 노력이 이루어지고 있다. 국내에서는 현대중공업의 충돌 회피 지원시스템 HiCASS, 삼성중공업의 선박 모니터링 시스템 VPS를 개발하여 자사 건조 선박에 적용을 시작하였다. 위와 같이 자율운항선박에 대한 관심과 기술개발이 이루어지고 있어 국제해사기구(IMO)에서는 자율화 등급에 따라 자율운항선박에 대하여 정의하고 있다.

자율운항선박은 유인 선박 및 장애물을 회피하는 기술을 필수적으로 보유하고 있어야 한다. Statheros et al. (2008)은 장애물 회피에 대한 이해와 이론적 배경을 제시하고, 국제해사기구(IMO)가 제정한 국제 해상충돌 예비규칙(International regulations for preventing collision at sea, COLREGs)에 대하여 기술한 바 있다.

장애물 인식에 주로 사용되는 센서는 레이더(RADAR), 라이다(LiDAR), 초음파, 카메라 등이 있다. Polvara et al. (2018)은 환경에 따른 장애물 인식에 대한 다양한 센서의 장·단점과 인식된 장애물을 기반으로 충돌 회피 경로 계획에 대한 연구를 수행한 바 있다. LiDAR 센서는 2, 3차원 센서로 나뉜다. 3차원 센서의 경우 좁은 공간에서는 데이터의 정밀도가 높고 견고했지만 움직이는 장애물을 실시간으로 예측할 때에는 데이터의 잡음이 커지게

된다 (Hong et al., 2012). 따라서 실외에서 운항하는 선박의 경우에는 3D LiDAR보다는 2D LiDAR를 통해 장애물을 정확하게 계측하는 것이 적합하다. Song et al. (2019)은 저가형 2D LiDAR 센서와 퍼지(fuzzy) 제어를 사용하여 좌·우현의 추진기 속력을 조절하며 장애물을 회피하는 시스템을 제안한 바 있다.

충돌 회피 경로 계획은 장애물 지도를 이용하여 경로를 생성하는 전역 경로 계획(global path planning) 기법과 장애물을 실시간으로 계측하여 경로를 생성하는 지역 경로 계획(local path planning) 기법이 있다. 대표적으로 Larson et al. (2007)의 2차원에서의 장애물 충돌 회피 경로 계획에 대한 연구, Kuwata et al. (2014)의 국제 해상충돌 예비규칙과 속도 장애물을 고려한 충돌 회피 경로 생성에 대한 연구가 수행된 바 있다. 이외에 Simmons and Henriksen (1996)의 Morphin 알고리즘, Casalino et al. (2009)의 경계박스(Bounding Box, BB) 개념 알고리즘 등이 있다.

본 연구에서는 로봇운영체제를 이용하여 다양한 센서가 통합된 자율운항 알고리즘의 구성 및 검증은 목표로 한다. 보트는 2D LiDAR를 이용하여 실시간으로 장애물이 없는 안전구역을 선정 후, PID 제어를 바탕으로 보트의 좌·우현 추진기의 회전 수 차이로 보트의 방위각을 제어하였다. 최종적으로, 2020 자율운항보트 경진대회(Korea Autonomous BOAT, KABOAT)에 참가하여 자율운항 알고리즘의 성능을 검증하였다.

2. 자율운항보트 시험준비

2.1 자율운항보트 및 탑재장비

본 연구에 사용된 보트는 Fig. 1의 부경대학교 PASS Mk. I (Pukyong Autonomous Surface Ship Mark I)이다. 보트는 유리섬유(Glass Fiber Reinforced Plastic, GFRP)와 아크릴로 제작된 좌우대칭의 쌍동선(catamaran) 형식이다. 보트의 제원은 Table 1과 같고, 좌·우현에 배치된 2개의 추진기로 보트의 변속 및 조종을 수행한다.

본 연구의 자율운항보트에는 Fig. 2와 같이 보트의 자세 및 위치를 측정하는 관성측정장치(Inertial Measurement Unit, IMU)와 GPS 센서, 장애물 인식 및 거리 측정을 위한 LiDAR(Light Detecting And Ranging)와 각 센서의 데이터를 처리 및 연산하는 메인 프로



Fig. 1 PASS Mk. I

세서와 서브 프로세서인 라즈베리파이(Raspberry Pi) 등을 탑재하였다. 자율운항보트의 동력원은 14.8V의 리튬폴리머(Lithium Polymer, Li-Po) 배터리를 사용하였다.

Table 1 Principal dimensions of PASS Mk. I

Measurement	Value	Unit
Length	1.2	m
Width	0.64	m
Height	0.44	m
Draft	0.08	m

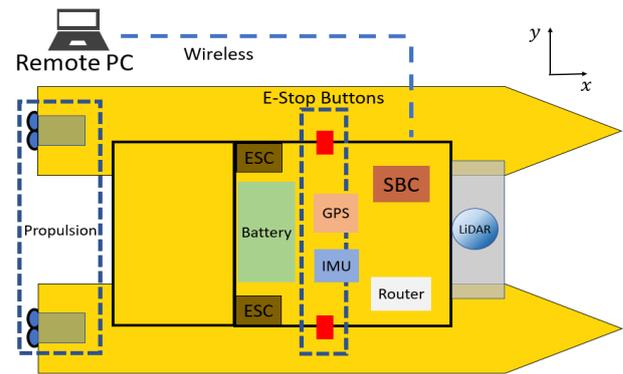


Fig. 2 Plot plan of PASS Mk. I

2.2 좌표계

본 연구에서 사용된 좌표계는 Fig. 3과 같다. 수평면 운동을 표현하기 위하여 지구고정좌표계 $O_0-x_0y_0$ 와 보트고정좌표계 $O-xy$ 가 사용되었다. 지구고정좌표계 x_0 축과 보트고정좌표계 x 축이 이루는 각을 방위각 ψ 로 정의하였다. ψ 는 시계방향을 양의 방향으로, LiDAR 데이터의 번호 i 는 반시계방향을 양의 방향으로 정의하였다.

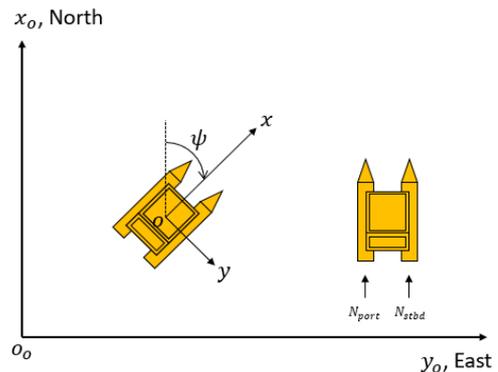


Fig. 3 Coordinate system

3. 자율운항 알고리즘

3.1 로봇운영체제

본 연구에서 각 센서의 데이터를 개별적으로 처리 후, 통합

하여 자율운항보트를 제어하기 위해 로봇운영체제 (Robot Operating System, ROS)를 사용하였다. 로봇운영체제는 로봇 소프트웨어를 개발하기 위한 소프트웨어 프레임 워크이며, 센서 및 장비 제어, 프로세스 간 통신 등을 위한 라이브러리가 포함된 오픈소스이다. 또한, 각 센서와 장비의 데이터를 개별적으로 처리할 수 있고, 복수의 프로그램 언어를 지원한다. 로봇운영체제는 Fig. 4에서 볼 수 있듯이 ‘로스 마스터(ROS master)’라는 관리자와 ‘노드(node)’라는 계산이 이루어지는 프로세스로 구성되어 있다. 로스 마스터에 Publisher 노드 및 Subscriber 노드를 등록하면 등록된 노드끼리 메시지가 담긴 토픽을 주고받을 수 있다. 즉, 로스 마스터와 노드는 각각 보트에 탑재된 프로세서와 센서이며, 메인 프로세서에 의해 센서의 데이터가 처리되고 보트를 제어하게 된다.

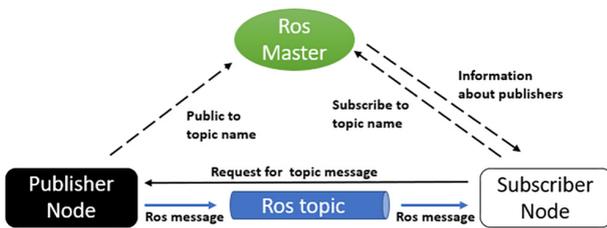


Fig. 4 Robot Operating System, ROS

3.2 제어 알고리즘

본 연구에서는 식 (1)과 (2)와 같은 PID 제어를 바탕으로 추진기의 회전수 차이를 이용하여 보트의 방위각을 제어하였다. ψ_d 는 목표 방위각, ψ 는 현재 방위각, e 는 방위각 오차, n 은 추진기의 회전수를 의미한다. 상수 k_p , k_i , k_d 는 각각 비례인수, 적분인수, 미분인수를 의미한다. 시행착오를 통해 PID 제어의 각 인수의 값을 Table 2와 같이 선정하였다.

$$e = \psi_d - \psi \tag{1}$$

$$n = k_p e + k_i \int e dt + k_d \dot{e} \tag{2}$$

좌·우현 추진기의 회전수 차이는 식 (3)과 식 (4)을 바탕으로 결정된다. n_{stbd} 는 우현 추진기의 회전수, n_{port} 는 좌현 추진기의 회전수를 의미한다. n_0 는 전진 및 후진을 위한 추진기의 기본 회전수를 의미하며 보트와 장애물 사이의 거리에 따라 결정된다.

$$n_{stbd} = n_0 \pm n \tag{3}$$

$$n_{port} = n_0 \mp n \tag{4}$$

Table 2 PID Parameter

Parameter	Value
k_p	0.45
k_d	0.2
k_i	0.01

3.3 경유점 추종 알고리즘

본 연구에서 GPS로 측정된 보트의 위치 데이터와 경유점의 위치 데이터를 이용하여 Fig. 5와 같이 경유점을 추종하기 위한 목표 방위각을 유도하였다. 보트와 경유점의 경도와 위도를 UTM-K(Universal Transverse Mercator-Korea) 좌표계로 변환한 후, 식 (5)와 같이 보트의 실시간 위치와 경유점 사이의 각도를 목표 방위각으로 계산하였다. (x_s, y_s) 는 GPS로부터 측정된 보트의 UTM-K 좌표를, (x_{wp}, y_{wp}) 는 경유점의 UTM-K 좌표를 의미한다. Fossen (2002)에 의거하여 목표 경유점으로부터 보트 전진길이의 약 2배인 반경 2.5m내 진입 시 도착이라고 판단하였다.

$$\psi_d = \tan^{-1} \left(\frac{y_{wp} - y_s}{x_{wp} - x_s} \right) \tag{5}$$

경유점 추종 알고리즘에서 정확한 위치에 도착하기 위해서 GPS의 정확도가 중요하다. GPS의 정확도를 측정하기 위해 GPS의 데이터 측정주기를 10Hz로 설정하여 부경대학교 총무관 옥상에서 실험을 진행하였다. 실험 결과 GPS 오차가 심해 식 (6)과 식 (7)과 같은 저주파 통과 필터(Low Pass Filter, LPF)를 통해 센서의 오차를 억제하였다.

$$\hat{x}_{s,k} = \alpha \hat{x}_{s,k-1} + (1-\alpha)x_{s,k} \tag{6}$$

$$\hat{y}_{s,k} = \alpha \hat{y}_{s,k-1} + (1-\alpha)y_{s,k} \tag{7}$$

위 첨자 “ $\hat{}$ ”는 추정값으로 정의하였다. k 는 데이터의 순서를 의미하는 지수, $x_{s,k}$, $y_{s,k}$ 는 측정된 위치, $\hat{x}_{s,k}$, $\hat{y}_{s,k}$ 는 저주파 통과 필터를 통하여 추정된 위치를 의미한다. 상수 α 는 저주파 통과 필터의 설계 파라미터(parameter)이며, 본 팀은 시행착오를 통하여 0.8로 적용하였다.

고정된 위치에서 GPS를 이용하여 약 15분간 계속 및 추정된 위치 데이터를 도시한 결과는 Fig. 6의 그래프와 같다. 파란색 실선은 GPS로 측정된 위치 데이터이며, 빨간색 점선은 저주파 통과 필터로 추정된 위치 데이터이다. 실험 결과 GPS로 측정된

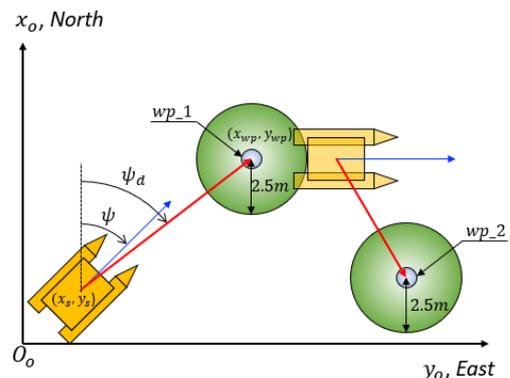


Fig. 5 Autopilot algorithm

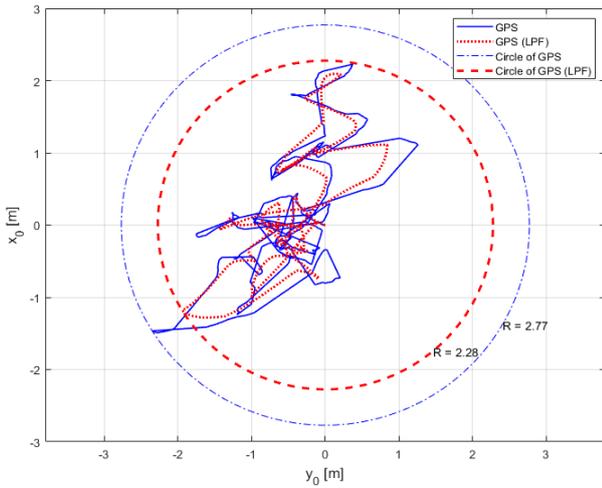


Fig. 6 GPS accuracy test

위치 데이터는 시작점을 기준으로 반경 약 2.8m 이내의 오차가 있었으나, 필터를 통해 약 2.3m 이내로 오차가 줄었음을 확인할 수 있었다.

3.4 장애물 회피 알고리즘

장애물 회피 알고리즘은 LiDAR 센서로 장애물을 인식하고 회피할 수 있는 목표 방위각을 유도하였다. LiDAR 센서의 데이터는 식 (8)과 같이 계측된다.

$$\alpha_0 = 2\pi/N, \alpha_i = i\alpha_0 \quad \text{where, } i \in [1, N] \quad (8)$$

α_0 는 LiDAR 센서 데이터 간의 사잇각, N 은 데이터의 총 개수, i 는 데이터의 번호를 의미한다.

장애물 인식은 Fig. 7에서 도시된 방법으로 구현되었다. LiDAR 데이터 개수와 거리의 파라미터를 각각 η_1 과 η_2 로 정의한다. LiDAR 센서에서 계측된 장애물의 데이터 번호 st 부터 ed 까지의 평균거리를 \bar{d} 로 정의한다. 이때, 데이터 번호 st 와 ed 사이의 데이터 개수는 η_1 보다 많아야 한다. 동시에, st 부터 ed 까지 모든 거리가 장애물의 평균거리 \bar{d} 로부터 $\pm\eta_2$ 이내에 존재한다면 장애물로 인식한다. Fig. 7의 오른쪽 상황의 경우, 데이터 개수는 η_1 보다 많았으나 모든 거리가 평균거리 \bar{d} 로부터 $\pm\eta_2$ 이내에 있지 않았기에 장애물로 인식되지 않는다. 인식된 장애물들은 식 (9)와 같이 Ob_j 로 저장된다.

$$Ob_j = [\alpha_{st}, \alpha_{ed}, d] \quad (9)$$

Ob_j 는 LiDAR 센서로 계측된 j 번째 장애물의 시작각 α_{st} , 끝각 α_{ed} , 거리 d 를 의미한다.

$$Safety\ zone_k = [\theta_{st}, \theta_{ed}, d_{st}, d_{ed}] \quad (10)$$

식 (10)의 안전구역(safety zone)은 LiDAR 센서로 계측된 모든 장애물을 제외한 구역을 의미하며 Fig. 8에 도시되어 있다.

안전구역은 시작각 θ_{st} , 끝각 θ_{ed} , 시작각 방향 장애물의 거리 d_{st} , 끝각 방향 장애물의 거리 d_{ed} 로 구성된다. k 는 안전구역 중 가장 넓은 구역의 번호이다. 목표 방위각은 안전구역 좌우에 존재하는 장애물 거리를 기중치로 하여 더 멀리 있는 장애물에 가깝도록 식 (11)과 같이 계산된다.

$$\psi_d = \frac{\theta_{st} d_{ed} + \theta_{ed} d_{st}}{d_{st} + d_{ed}} \quad (11)$$

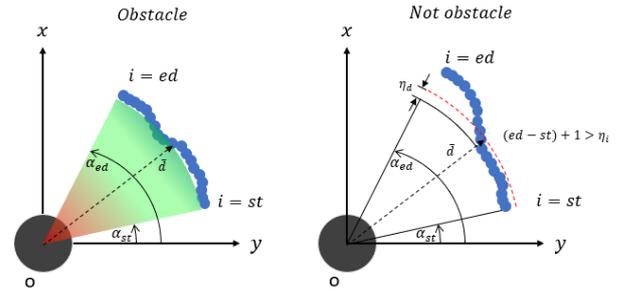


Fig. 7 Obstacle recognition algorithm

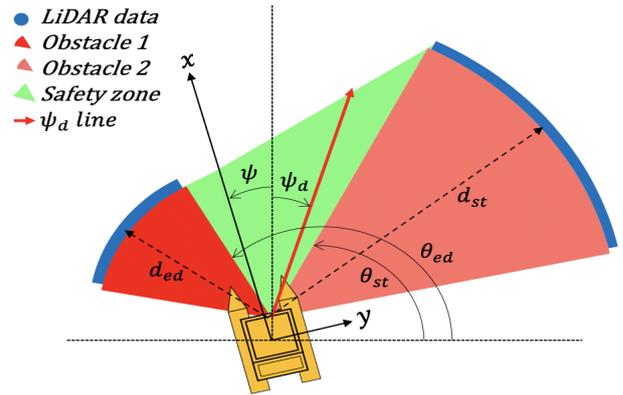


Fig. 8 Obstacle avoidance using safety zone method

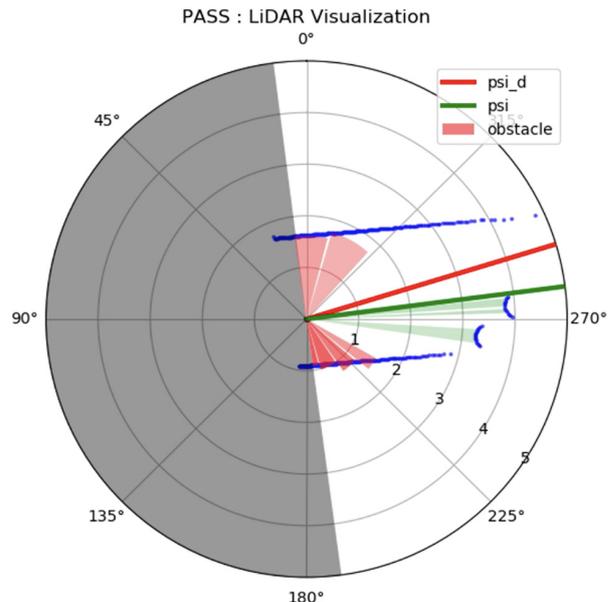


Fig. 9 Visualization of autonomous algorithm

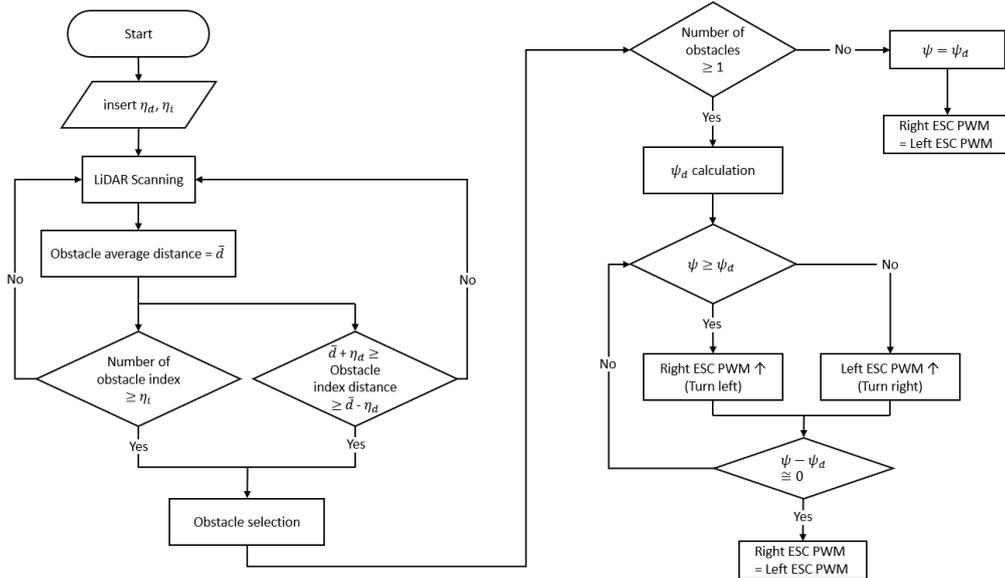


Fig. 10 Overview of the autonomous algorithm

장애물 회피 알고리즘의 시각화(Visualization)는 Fig. 9와 같다. 빨간색 실선은 장애물 회피 알고리즘으로 계산된 목표 방위각, 초록색 실선은 보트의 현재 방위각, 파란색 점은 LiDAR 센서로부터 계속된 데이터이다. 인식된 장애물의 거리에 따라 부채꼴의 색상이 빨간색 혹은 초록색으로 결정된다. 장애물 회피 자율운항 알고리즘의 개요는 Fig. 10과 같다.

4. 자율운항시험 결과 및 분석

본 연구의 자율운항 알고리즘 성능을 확인하기 위해 부경대학교 예인 수조에서 본 시험을 수행하였다. 부경대학교 예인 수조의 제원은 Table 3과 같다.

자율운항보트의 제어 알고리즘 성능을 확인하기 위해 제자리 선화시험을 진행하였다. Fig. 11은 보트의 방위각을 0°, 목표 방위각을 90°로 설정한 제자리 선화시험 결과를 나타내고 있다. 오버슈트는 약 25%, 상승 시간(rise time)은 약 2초, 5%의 정정 시간(settling time)은 14.39초였다. 시험결과를 보면, 좌·우현 추진력의 차이로 설정한 목표 방위각을 추종하는 것을 확인할 수 있다. Table 4는 제자리 선화시험의 시간 응답 요소를 표로 정리한 것이다.

장애물 회피 알고리즘 시험에 앞서 보트의 궤적을 기록하기 위해 보트의 위치 데이터를 계속하였다. GPS의 특성상 실내에서 보트의 위치를 계속할 수 없으므로, 실내·외에서 초음파를 발사하여 위치를 계속할 수 있는 마블마인드(Marvelmind)사의 비콘(Beacon)을 사용하였다. 비콘을 보트와 예인 수조 벽면에 부착하여 10Hz의 계속주기로 보트 및 장애물 위치를 계속하였다. 계속된 데이터의 잡음을 억제하기 위해 3.3절의 저주파 통과 필터를 적용하였다. 보트의 속도는 계속된 보트의 위치 데이터를 이용하여 식 (12)로 추정하였다. “ \hat{U} ”는 보트 전속도의 추

정값으로 정의하였고, $x_{s,k}$, $y_{s,k}$ 는 계속된 위치, dt 는 계속된 데이터 간의 시간을 의미한다.

Table 3 Dimensions of PKNU towing tank

Measurement	Value	Unit
Length	40	m
Width	2.5	m
Depth	1.5	m

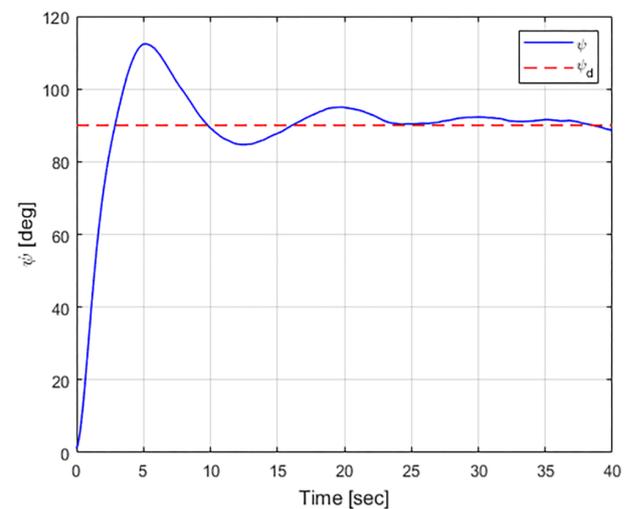


Fig. 11 Turning test by PID control

Table 4 Time response specification of turning test

Measurement	Value	Unit
Overshoot	25.37	percent
Rise time	1.99	sec
5% Settling time	14.39	sec

$$\hat{U}_k = \sqrt{\left(\frac{x_{s,k} - x_{s,k-1}}{dt}\right)^2 + \left(\frac{y_{s,k} - y_{s,k-1}}{dt}\right)^2} \quad (12)$$

본 연구에서 비콘을 이용하여 계측한 보트 위치 데이터를 바탕으로 Table 5와 같이 모터 추진력의 백분율당 보트의 추정 속도를 나타내었다.

장애물 회피 시험은 case 1, case 2로 나누어 시험했으며, case 1의 경우 선수를 동쪽으로, case 2는 서쪽으로 두고 자율 운항하였다. 장애물은 8m의 간격을 두고 수조의 좌우 벽면에 설치하였다. 각각의 시험을 통해 계측한 보트의 추정 위치, 현재 방위각 및 목표 방위각, 방위각 오차, 속도, 좌·우현 추진기 회전수의 백분율을 Fig. 12와 Fig. 13에서 볼 수 있다. 시험결과

보트의 궤적을 나타낸 그래프를 보면, 자율운항 알고리즘을 기반으로 장애물 인식 및 회피기능을 수행하는 것을 확인할 수 있다. 또한 비콘으로 계측된 위치 데이터에 저주파 통과 필터를 적용하여 잡음을 줄였으나, 잡음이 완전히 억제되지 않았음을

Table 5 The boat's speed corresponding to the percentage of motor propulsion

Speed [m/s]	Percentage [%]
1.0	30
1.2	40
1.5	60
1.8	80

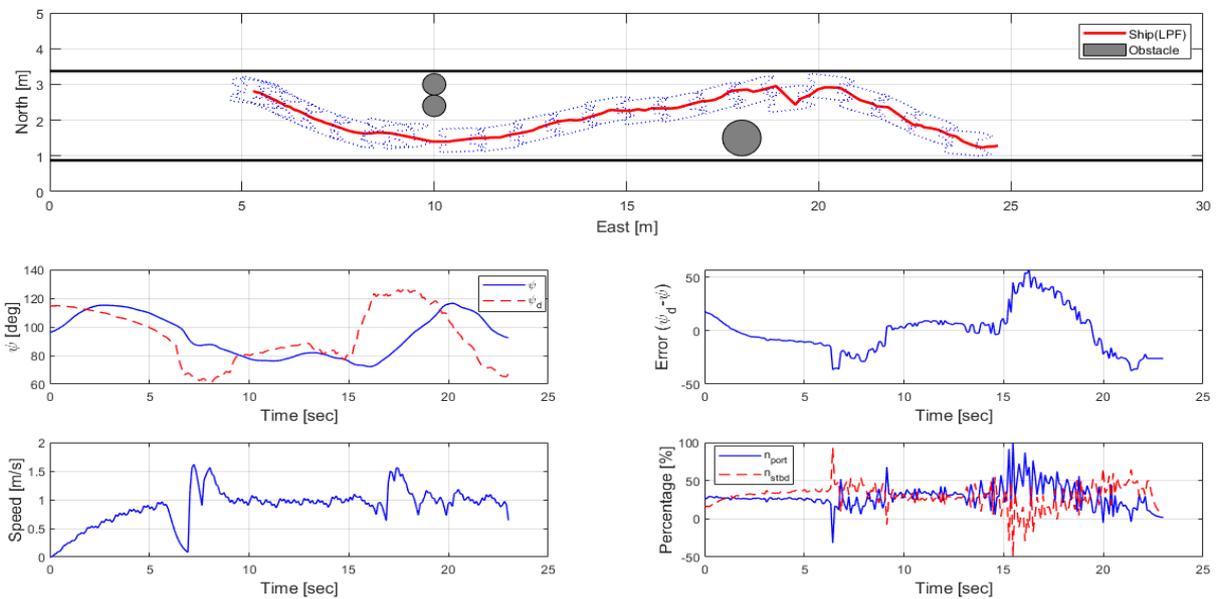


Fig. 12 Autonomous algorithm test results (case 1)

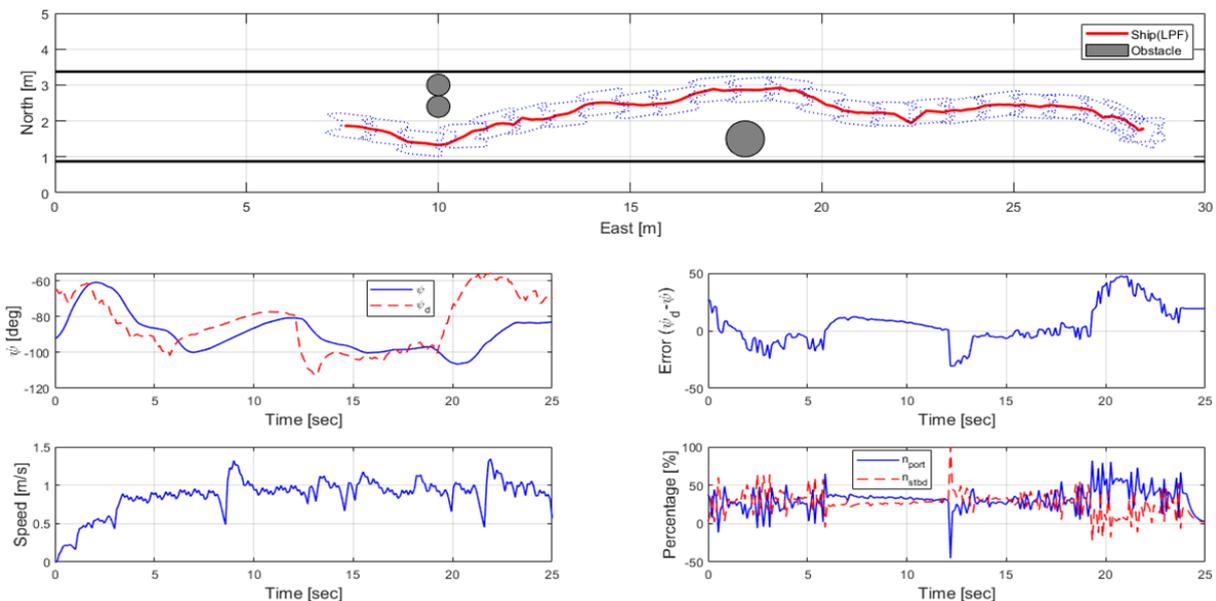


Fig. 13 Autonomous algorithm test results (case 2)

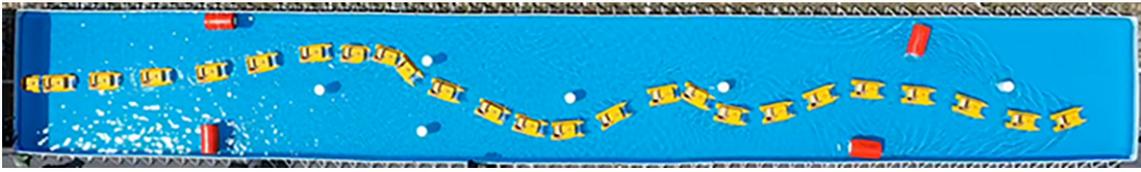


Fig. 14 Video snapshot during autonomous preliminary task in KABOAT 2020



Fig. 15 Video snapshot during of autonomous final task in KABOAT 2020

볼 수 있다. 이는 장애물 가까이에서 비콘 간의 초음파 송수신 문제로 발생한 것으로 판단된다. 장애물이 인식되지 않을 경우, 방위각 오차가 0에 가까워지는 것을 확인할 수 있다. 반면, 장애물을 인식한 경우, 좌·우현의 추진기 회전수 차이로 보트의 방위각을 제어하여 장애물을 회피하는 것을 확인할 수 있다.

작성된 자율운항 알고리즘을 활용하여 2020 자율운항보트 경진대회에 참가하여 실외에서의 알고리즘 성능을 검증하였다. Fig. 14와 Fig. 15는 각각 드론으로 촬영한 자율운항보트 경진대회의 자율운항임무 예선과 결선의 동일한 주기 스냅샷 사진이다. 장애물 회피 알고리즘에 따라 장애물을 완벽히 회피하는 것을 확인할 수 있다. 또한, 예선의 스냅샷 사진에서 장애물과 가까울 때 보트가 겹쳐 보이는 것은 감지된 장애물과 거리에 따라 보트의 속도가 제어되기 때문이다.

5. 결론

본 논문에서는 보트의 자율운항 알고리즘의 개발 및 성능 검증을 위해 시험을 수행하고 그 결과를 서술하였다. 자율운항보트에 사용된 로봇운영체제로 센서의 데이터를 처리 및 통합하였다. 예측된 데이터로부터 장애물을 감지하고 안전구역을 선정하여 자율운항보트의 장애물 인식 및 회피 운동을 구현하였다.

본 연구의 한계는 다음과 같다. 첫째, 정적 장애물을 대상으로 작성된 장애물 회피 알고리즘이므로 동적 장애물 회피에 대해 확신할 수 없다. 둘째, 보트가 횡동요 혹은 종동요 상황일 때 2차원 라이다로는 정확한 장애물 탐지에 어려움이 있었다.

향후 움직이는 장애물에 대한 회피 알고리즘을 구현하고, 카메라 이미지 프로세싱을 이용한 장애물 회피 및 도킹에 대해 연구할 예정이다.

후 기

본 논문은 2020년도 해양수산부 및 해양수산과학기술진흥원

연구비 지원으로 수행된 '자율운항선박 기술개발사업(20200615, 지능형 항로 의사결정 기능을 갖는 자율운항 시스템 개발)' 및 부경대학교 신입교원 학술지원사업(C-D-2018-0934)의 지원으로 연구되었습니다.

References

- Casalino, G., Turetta, A. & Simetti, E., 2009. A Three-layered architecture for real time path planning and obstacle avoidance for surveillance USVs operating in harbour fields. *Oceans 2009*, Bremen, Germany, pp.1-8.
- Fossen, T.I., 2002. *Marine control systems: guidance, navigation and control of ships*. Rigs and Underwater Vehicles, Marine Cybernetics.
- Hong, S.C., Park, I.S., Heo, J., & Choi, H.S., 2012. Indoor 3D modeling approach based on terrestrial LiDAR. *Journal of the Korean Society of Civil Engineers*, 32(5D), pp.527-532.
- Kuwata, Y., Wolf, M.T., Zarchitsky, D. & Huntsberger, T.L., 2014. Safe maritime autonomous navigation with COLREGS, using velocity obstacles. *IEEE Journal of Oceanic Engineering*, 39, pp.110-119.
- Larson, J. et al., 2007. Autonomous navigation and obstacle avoidance for unmanned surface vehicles. *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp.17-20.
- Polvara, R., et al., 2018. Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles. *The Journal of Navigation*, 71(1), pp.241-256.
- Statheros, T., Howells, G., & McDonald-Maier, K. 2008. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *The Journal of Navigation*, 61, pp.129-142.

Simmons, R. & Henriksen, L. 1996. Obstacle avoidance and safeguarding for a lunar rover. *AIAA Forum on Advanced Developments in Space Robotics, Madison, WI.*

Song, H.W., Lee, K.K., & Kim, D.H., 2019. Implementation of an obstacle avoidance system based on a low-cost LiDAR sensor for autonomous navigation of an unmanned ship. *The transactions of The Korean Institute of Electrical Engineers*, 68(3), pp.480-488.

