

Aruco Marker를 사용한 드론 정밀 착륙 시스템

강 호 현*, 신 수 용°

Precise Drone Landing System Using Aruco Marker

Ho Hyun Kang*, Soo Young Shin°

요 약

본 논문은 기존 드론 시스템에서 발생하는 착륙 오차를 마커를 사용하는 방법을 통해서 줄이는 것을 목적으로 한다. 정밀 착륙 시스템은 주로 선박이나 병원 건물 옥상과 같은 특수한 목적을 위하여 사용되는 지점에 사용된다. 이를 위하여 GPS와 더불어 드론의 측위를 위하여 Aruco Marker라는 마커를 착륙지점에 부착하고 이를 기반으로 좌표계를 형성한다. 이후에 좌표계와 GPS를 동시에 사용하여 이를 구현하였고 GPS만을 이용한 기존 방법과 착륙지점 오차, 수행 시간을 비교한다. 이를 구현하고 GPS와 제안하는 기법의 착륙지점 오차, 수행 시간을 측정하고 비교한다. 제안하는 기법의 착륙지점 오차는 약 5cm 미만으로 측정되는 것을 확인하였다.

키워드 : 드론, 자동착륙, 범지구위치결정시스템, Aruco Marker

Key Words : UAV (Unmanned Aerial Vehicle), Auto landing, GPS (Global Positioning System), Aruco Marker

ABSTRACT

The purpose of this paper is to reduce landing errors occurring in existing drone systems through a method of using markers. Precision landing systems are mainly used at points used for special purposes, such as ships and rooftop of hospital buildings. To this end, a marker called Aruco Marker is attached to the landing point for the position of the drone along with GPS, and a coordinate system is formed based on this. After that, it was implemented using the coordinate system and GPS at the same time, and the landing point error and execution time were compared with the existing method using only GPS. This is implemented and the GPS and the landing point error and execution time of the proposed technique are measured and compared. It was confirmed that the landing point error of the proposed technique was measured to be less than about 5 cm.

I. 서 론

다양한 영역에서 드론이 활용성이 높아지고 있고 그 중에 상당수가 인력을 사용하기 힘든 상황 (재난, 구조, 탐색, 운송, 정찰 및 보안, 지리탐사 등)으로 많은 연구 개발이 진행되고 있다^[1]. 현재 드론은 방송이나 촬영, 취미 등으로 주로 활용되기 때문에 무인 운

영보다는 자격증이 있는 사람들이 직접 제어하고 있다. 그러나 여러 대의 드론을 동시에 운영하거나 장시간 사용하게 되는 경우가 발생한다. 또한, 사람의 시야로 확인하지 못하는 곳에서 특정한 임무를 수행하는 경우가 있고 카메라를 통한 시야 확보로 운영 중이다. 이러한 다양한 상황이나 문제를 해결하기 위하여 드론 자율비행을 필요한데, 이를 위하여 경로생성^[2],

※ “본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성(Grand ICT연구센터) 사업의 연구결과로 수행되었음” (IITP-2021-2020-0-01612)

• First Author : Kumoh National Institute of Technology, 20216106@kumoh.ac.kr, 학생회원

° Corresponding Author : Kumoh National Institute of Technology, wdragon@kumoh.ac.kr, 종신회원

논문번호 : 202108-217-D-RN, Received August 26, 2021; Revised November 8, 2021; Accepted November 14, 2021

장애물 회피^[3], 자동착륙^[4] 등의 여러 가지 기술들이 필요하다. 특히, 드론은 배터리를 통해 운영할 수 있기 때문에 이에 따라 충전 시스템도 필요하다^[5]. 제한된 환경에서 드론의 운영 시간을 늘리고 충전 시스템과 정확하게 안착될 수 있는 정밀 착륙 시스템이 요구된다.

이처럼 정밀착륙 시스템을 구성하기 위해서는 GPS만 사용해서는 착륙지점과의 오차가 평균 140cm 정도로 크기 때문에 불가하다^[6]. 또한, GPS 신호 불가 지역에서는 좌표계에 대한 정보가 아예 없기 때문에 착륙지점을 탐색하기 어렵다. 그러므로 마커를 활용한 카메라 기반 좌표계를 사용하는 방법을 통해 이를 해결할 수 있다. 본 논문에서는 Aruco Marker라는 영상처리 기법을 통해 착륙지점에 가상의 좌표계를 생성하고 이를 기반으로 현재 드론 측위를 수행하여 정밀 착륙하는 시스템을 제안한다.

논문 구성은 다음과 같다. 2장에서는 본 논문의 실험에서 이용된 시스템 모델 및 사용된 ROS, Aruco Marker, 알고리즘에 대하여 정리하고 3장에서는 실험 환경 및 실험방법을 설명하고 각 오차를 측정 및 비교하여 분석한다. 그리고 4장에서 논문의 결론을 맺는다.

II. 본 론

2.1 시스템 모델

본 논문에서는 그림 1과 같은 시스템 모델을 구성하였다. GCS(Ground Control Station)와 드론은 Wi-Fi로 연결되며 이는 드론의 영상을 GCS으로 송신하고 이에 알고리즘 연산을 수행하고 이에 따른 드론 제어 명령을 GCS에서 송신한다. 처음에는 GPS 네비게이션을 사용하여 목표지점까지 도달하고 이후에는 착륙패드에서 Aruco Marker를 삽입하고 이를 내장된

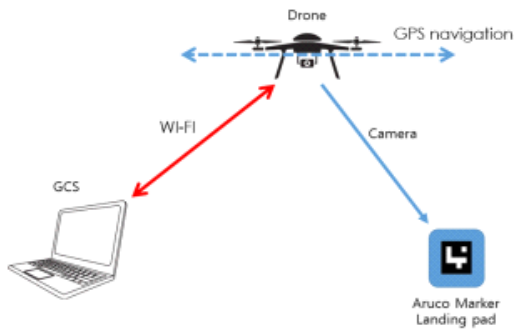


그림 1. 시스템 모델
Fig. 1. System model

카메라로 인식하여 드론의 측위를 진행한다.

2.2 ROS

ROS (Robot Operation System)은 로봇 운영 시스템으로 운영 체제가 아닌 메타 운영 체제로서 다중종간에 통신으로 제어를 가능하게 한다. 또한, 다른 언어로 작성된 코드를 토픽과 노드로 구성함으로써 이종 호환 문제, 운영 체제 호환 문제없이 서로 간의 통신을 할 수 있게 한다^[7]. ROS를 통하여 드론을 제어하면 서로 다른 언어로 작성된 알고리즘을 결합하는 것이 쉬워지므로 자유로운 제어가 가능하다. 본 논문에서는 이러한 목적으로 ROS를 통해 영상처리와 제어를 동시에 수행할 수 있도록 구현한다.

2.3 Aruco Marker

Aruco Marker는 사각형 형태를 가진 마커를 카메라로 위치를 예측하기 위한 OpenCV 기반 오픈소스 라이브러리이다. 카메라를 사용하여 드론의 측위를 진행하고 이를 기준으로 착륙패드와의 거리 및 위치를 계산하여 정밀 착륙을 하도록 제어한다^{[8][9][10]}.

2.4 알고리즘

본 논문에서 제안하는 알고리즘은 GPS만을 사용한 네비게이션과 Aruco Marker를 사용하여 정밀 착륙을 수행하도록 크게 두 부분으로 나뉘어진다.

그림 2는 GPS만을 사용하여 착륙지점까지 자율비행하고 해당 지점의 좌표를 기준으로 고도만 변경하여 착륙하는 기존 방식이다. 그림 3은 제안하는 알고리즘으로 기존 방식으로 착륙지점까지 GPS으로 이동하고 카메라로 착륙패드가 인식이 되는 지점부터 Aruco Marker를 이용한 정밀 착륙을 수행하는 방식이다. GPS로 도달한 목표지점이 실제 목표지점과 오차가 발생할 수 있기 때문에 카메라 기반 위치 추정 알고리즘을 수행하고 마커를 기준으로 드론이 정중앙에 위치하도록 좌표를 수정한다. 그런 다음 마커가 중앙의 지정된 영역 안에 올 때 드론의 고도를 조금씩 낮춰준다. 이 영역의 크기는 세 단계로 나누어 첫 단계에서는 고도를 빨리 낮추고 마지막 단계에서 정밀성을 위주로 고도를 낮춘다. 이때 드론과 착륙 패드 간의 거리 D 가 θ 보다 작을 때 드론이 착륙하는 것으로 알고리즘은 종료된다. 단 드론과 착륙 패드 간의

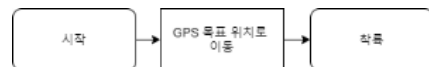


그림 2. GPS 기반 자동착륙 알고리즘
Fig. 2. GPS land algorithm

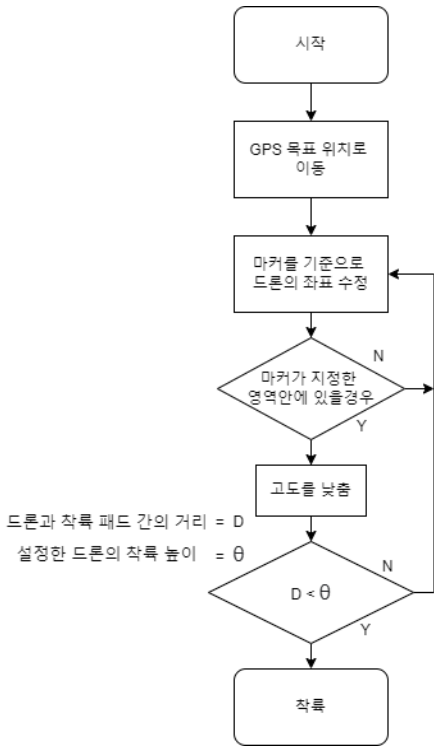


그림 3. GPS+Marker 자동착륙 algorithm
Fig. 3. Marker algorithm

거리 D 가 θ 보다 높을 때는 드론의 위치를 계속 수정하고 고도를 낮춰준다. 드론의 착륙 높이는 사용하는 플랫폼에 따라 변경이 가능하고 30cm로 지정하여 사용한다.

2.5 ROS 노드 & 토픽

ROS 노드란 코드를 작성하여 노드화 시킨 것이며 다른 노드들과 토픽으로 서로간의 소통을 한다. 이 같

은 방식을 사용함으로써 다른 언어로 작성된 코드들과의 소통도 가능하게 한다.

본 논문에서 사용된 ROS 노드는 /bebop 노드, /example_node 노드, /aruco_single 노드로 크게 세 가지로 구성되어 있으며, 노드와 토픽의 연결구조는 그림 4와 같다. /bebop 노드 안에 있는 토픽들은 parrot사의 bebop 2 드라이버에서 지원하는 노드와 토픽들이며 Control 네모 안의 토픽들은 드론을 제어하는데 사용되는 핵심 토픽들이다. /bebop/camera_control 토픽은 bebop 2 드론에 장착된 카메라 짐벌을 제어할 때 사용되는 토픽이고, /bebop/land 토픽은 드론을 착륙할 때 사용되는 토픽이며, /bebop/cmd_vel 토픽은 bebop 드론의 움직임을 제어할 때 사용되는 토픽이고, /bebop/takeoff 토픽은 드론을 이륙할 때 사용되는 토픽이다. /example_node는 본 논문에서 실험을 진행하기 위해 코드를 작성하여 노드화 시킨 것이며 GPS 이동과 착륙, Aruco Marker 착륙을 한 번에 진행하기 위해 사용되었다. 우측의 /bebop/image_raw 토픽은 드론에 장착된 카메라에서 이미지를 가져올 때 사용되는 토픽이며 /bebop/fix 토픽은 드론에 장착된 GPS로부터 GPS 값을 받아올 때 사용되는 토픽이다. 맨 좌측의 /aruco_single과 이어진 토픽들은 Aruco Marker로부터 카메라 기반 위치 추정 알고리즘을 수행하여 좌표값을 받아올 때 사용되는 토픽들이다. 본

표 1. 실험 환경
Table 1. Experimental Environment

System	GCS Laptop	Parrot	OS	Etc.
Model name	I7-10750H	Bebop drone 2	Ubuntu 18.04(use Docker)	ROS, python

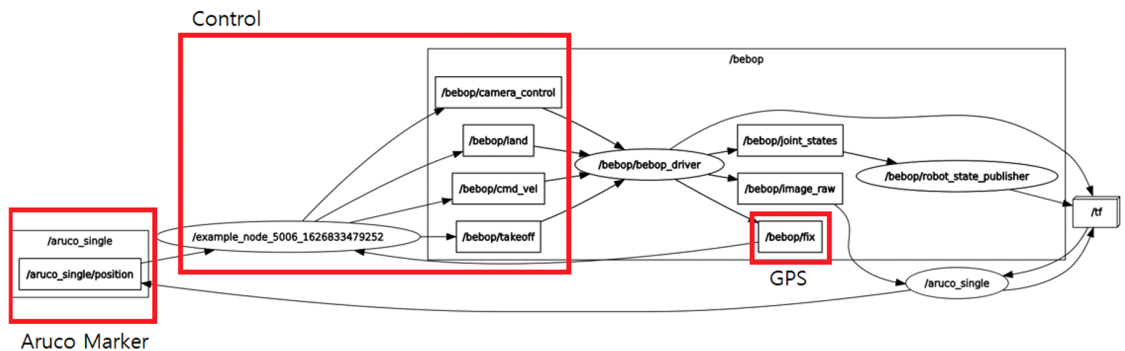


그림 4. 사용된 ROS 노드 & 토픽
Fig. 4. ROS Node & Topic

논문에서는 이 ROS 노드들과 토픽들을 사용하여 드론의 움직임을 제어하고 자동착륙하는 시스템을 구축하였다.

III. 실험

3.1 실험 장비 및 실험 환경

자동착륙 정확성 실험은 GPS만 사용한 착륙과 Aruco Marker를 같이 사용한 착륙 두 가지 방법으로 나누어 진행하였다. 드론은 parrot사의 bebop 2 드론을 사용하였으며, 착륙 패드는 Aruco Marker 1번을 20x20cm 크기로 인쇄하여 제작하였다. 드론을 제어할 GCS는 i7-10750H laptop을 사용하였고 드론 플랫폼으로 사용한 bebop 2의 드라이버가 Ubuntu 16.04 까지만을 지원함으로써, 이를 호환성 있게 접근하도록 Docker를 사용한다. 결과치를 구할 착륙지점의 GPS 목표 좌표는 드론의 내장된 GPS를 사용하여 지정하였다. 실험 진행 순서는 드론이 착륙지점(착륙 패드)에서부터 일정 거리가 떨어진 곳에서 비행을 시작하여 사전에 지정해둔 목표지점(GPS 좌표인 착륙 패드가 있는 지점)으로 이동한 뒤 각각 자동착륙을 진행한다. 이때 착륙 패드 중앙점에서 드론 중앙점까지 얼마나 떨어졌는지 착륙 정확성을 각각 10번씩 측정한 평균값과 이동한 경로, 착륙에 걸린 시간을 비교하였다. 아래의 표 1은 본 논문에 사용된 시스템을 정리한 표이다.

3.2 GPS 기반 자동착륙

3.2.1 실험방법

먼저 착륙 패드에 드론을 놔두고 목표지점 GPS 좌표를 측정한다. 그리고 드론이 일정 거리가 떨어진 위치에서 목표지점인 GPS 좌표로 이동하고 착륙을 하는 GPS 알고리즘을 실행하고 이동 경로와 걸린 시간을 측정하고, 지정된 목표지점 중앙점에서 드론 중앙점의 거리로 오차를 측정한다.

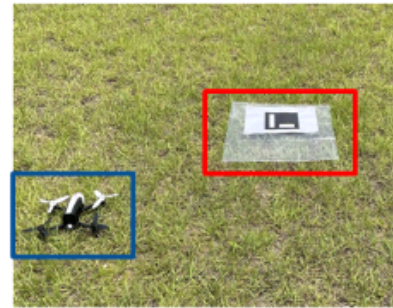
표 2. GPS 착륙 오차
Table 2. GPS land error

Count	Error(cm)	Time(sec)
1	84	26
2	97	41
3	206	28
4	115	37
5	247	24

6	245	23
7	90	43
8	70	29
9	130	35
10	220	32

3.2.2 실험결과

표 2는 GPS를 사용한 자동착륙을 하는 경우 착륙 지점과 착륙한 위치까지의 거리 오차와 걸린 시간을 측정하여 표시한 표이다. 목표 착륙지점과 드론까지의 거리가 최소 70cm 거리부터 최대 247cm 거리까지 오차가 발생하였고, 시간은 최소 23초, 최대 43초의 시간이 걸렸다. 그림 5에서 GPS만 사용하여 드론이 착륙한 지점을 확인해 볼 수 있다.



□ Drone
□ Goal

그림 5. GPS를 사용한 결과
Fig. 5. GPS used result

3.3 GPS+Aruco Marker 기반 자동착륙

3.3.1 실험방법

Aruco Marker 역시 GPS 기반 자동착륙과 같이 진행한다. 하지만 드론이 목표 GPS 지점에 도착하였을 경우 장착된 카메라를 사용하여 Aruco Marker 자동착륙을 하는 알고리즘을 실행시켜 이동 경로와 걸린 시간을 측정한다. 이후 착륙 패드 중앙점에서 드론 중앙점의 거리로 착륙 정확성을 측정한다.

3.3.2 실험결과

표 3은 Aruco Marker를 사용한 자동착륙을 하는 경우 착륙지점과 착륙한 위치까지의 거리 오차와 걸린 시간을 측정하여 표시한 표이다. 목표 착륙지점과

표 3. GPS + Aruco Marker 착륙 오차
Table 3. GPS + Aruco Marker land error

Count	Error(cm)	Time(m:s)
1	2	1:45
2	3	2:12
3	8	2:05
4	2	1:38
5	5	2:07
6	3	2:12
7	1	1:53
8	6	1:33
9	5	2:20
10	5	1:32

드론까지 최소 2cm, 최대 8cm 거리까지 오차가 발생하였다.



그림 6. 자동착륙 중인 드론
Fig. 6. drone on auto landing



 Drone
 Goal

그림 7. Aruco Marker를 사용한 결과
Fig. 7. Aruco Marker used result

그림 6에서 Aruco Marker를 사용하여 자동착륙을 하는 모습을 볼 수 있으며, 그림 7에서 착륙이 완료된 모습을 확인해 볼 수 있다.

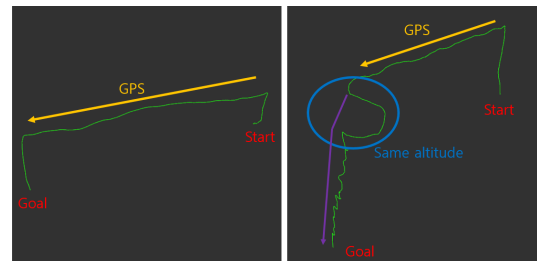
3.4 비교 및 성능 분석

GPS만 사용한 자동착륙과 GPS + Aruco Marker를 사용하여 자동착륙을 한 오차와 걸린 시간을 표 4에 정리하였다. GPS만 사용하여 걸린 시간은 평균적으로 31초, Aruco Marker를 사용한 자동착륙은 평균적으로 1분 55초로써 Aruco Marker가 정밀착륙을 하는 것에 비하여 시간이 오래 걸린다. 하지만 무선충전 등 정밀착륙을 요구할 때에는 시간이 걸리더라도 정확히 착륙하는 것이 의미가 있다. GPS만 사용하고 측정한 오차 평균값은 149.8cm로서 기존 연구 140.2cm와 비슷한 결과를 보여주었으며, Aruco Marker를 사용하여 측정한 오차 평균값은 4cm로 GPS 자동착륙과 비교하면 매우 높은 정확성을 보여주었다.

그림 8의 a 그림 GPS 궤적과 b 그림 Aruco Marker를 사용한 궤적을 서로 확인하여 볼 수가 있다. 그림 a의 GPS만 사용하여 자동착륙을 할때는 직선으로 이동 후 바로 수직으로 착륙을 하게 된다. 그래서 걸리는 시간이 짧지만 그림 b의 경우 GPS로 이동 이후 Aruco Marker 위에 정확히 착륙하기 위해 중심점 위치를 찾고, 천천히 위치를 조정하면서 착륙하기 때문에 GPS만 사용하여 착륙하는 것에 비하여 더 많이 복잡하고 시간이 걸리는 것을 확인해 볼 수 있다.

표 4. GPS & Aruco Marker 성능 비교
Table 4. GPS & Aruco Marker Performance Comparison

	GPS(cm)	Marker(cm)
Average	149.8	4
STD	70	2.1
Time	31sec	1m 55sec



a. GPS 경로 b. GPS + Aruco Marker 경로

그림 8. GPS & Aruco Marker 경로 비교
Fig. 8. GPS & Aruco Marker trajectory comparison

IV. 결 론

본 논문에서는 GPS만을 사용하여 착륙할 시에 발생하는 오차를 줄이기 위한 정밀 착륙 시스템을 제안하고 구축하였다. 또한 자율 비행을 수행하도록 함으로써 무인 드론의 활용성을 보여준다. 실제 실험결과를 비교했을 때, 제안하는 알고리즘이 적용된 경우에 오차가 5cm 미만으로 나온 것을 확인하였다. 본 논문에서는 특수한 목적을 기준으로 정밀한 착륙을 자율 비행으로 수행하는 것으로 연구하였다. 추후에는 신속한 착륙과 더불어 범용적인 목적을 수행하는 드론 제어 및 착륙 알고리즘에 대한 연구를 진행할 예정이다.

References

[1] S.-I. Oh, "A case study civilian drone," in *Proc. Korean Soc. Broadcast Eng. Conf.*, vol. 7, pp. 315-318, 2015.

[2] J.-H. Park, "Method for generating optimization paths of drone flight in user-defined zones," in *Proc. IEIE Fall Conf.*, pp. 860-861, 2017.

[3] S.-H. Lee, "A collision avoidance algorithm for autonomous drones," in *Proc. Symp. KICS*, pp. 428-429, Nov. 2016.

[4] E.-H. Sun, T. H. Luat, and Y.-T. Kim, "A study on the image-based automatic landing system of mini drone," *J. Korean Inst. Intell. Syst.*, vol. 25, no. 2, pp. 91-92, 2015.

[5] D.-N. Kim, "Research on a contact-type battery charging station for continuous mission performance of rotary wing drone," *J. KIIT*, vol. 19, no. 1, pp. 63-70, Jan. 2021.

[6] J. Kim, "An evaluation on accuracy of UAVs autonomous landing using GPS sensor module," *KIISE*, pp. 1473-1475, 2015.

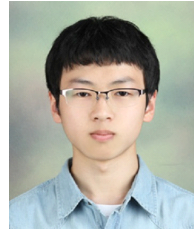
[7] Stanford Artificial Intelligence Laboratory, et al., *Robotic Operating System*, 2018, Retrieved from <https://www.ros.org>.

[8] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. Softw. Tools*, 2000.

[9] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, pp. 38-47, 2018.

[10] S. Garrido-Jurado, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, vol. 51, pp. 481-491, 2016.

강 호 현 (Ho Hyun Kang)



2021년 2월 : 국립금오공과대학교 전자IT융합과 졸업
 2021년 2월~현재 : 국립금오공과대학교 IT융복합공학과 석사과정
 <관심분야> 드론 응용, 로봇프로그래밍

[ORCID:0000-0003-3601-3690]

신 수 용 (Soo Young Shin)



1999년 2월 : 서울대학교 전기공학부 졸업
 2001년 2월 : 서울대학교 전기공학부 석사
 2006년 2월 : 서울대학교 전기공학부 박사
 2010년~현재 : 국립금오공과대학교 전자공학부 교수

<관심분야> 5G/B5G 무선 접속 기술, 드론 응용, 혼합 현실, 블록체인, 머신러닝 및 딥러닝
 [ORCID:0000-0002-2526-2395]