creative commons

COMMONS DEED

저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

조선대학교
CHOSUN UNIVERSITY

[UCI]I804:24011-200000267262

2019년 2월

석사학위논문

장애물 탐지 및 회피를 위한 VTOL 개발

조선대학교 대학원

항공우주공학과

양 유 영

조선대학교
CHOSUN UNIVERSITY

장애물 탐지 및 회피를 위한 VTOL 개발

VTOL Development for Obstacle Awareness and Collision Avoidance

2019년 2월 25일

조선대학교 대학원

항공우주공학과

양 유 영

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

장애물 탐지 및 회피를 위한 VTOL 개발

지도교수 이 현 재

이 논문을 공학석사학위 신청 논문으로 제출함

2018년 10월

조선대학교 대학원

항공우주공학과

양 유 영

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

양유영의 석사학위논문을 인준함

위원장 조선대학교 교수 김 태 규 (인)

위 원 조선대학교 교수 이 창 열 (인)

위 원 조선대학교 교수 이 현 재 (인)

2018년 11월

조선대학교 대학원

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

목차

I. 서론 1

II. 쿼드로터 구성 4

1. 시스템 구성 4

2. 비행원리 5

1) 수직 이착륙 및 제자리비행 6

2) Roll 운동 6

3) Pitch 운동 7

4) Yaw 운동 7

3. 구성 요소 8

1) 기체 8

2) 구동기 9

3) FCC 14

4) 센서 16

5) 배터리 19

6) RC 수신기 20

4. 임베디드 시스템 21

1) GPIO 21

2) UART 22

3) I2C 23

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

4) SPI 25
5) PWM 26
III.쿼드로터 모델링 **29**
1.좌표계 29
2.회전행렬 30
3.운동학 35
4.쿼드로터의 비행동역학 36
5.힘과 모멘트 38
IV.쿼드로터 제어기 **41**
1.PID 제어기 42
1) 비례 제어기 43
2) 적분 제어기 44
3) 미분 제어기 44
2.제어기 설계 45
V. 장애물 탐지 및 회피 **48**
VI.시뮬레이션 **50**
1.위치 및 자세제어 시뮬레이션 조건 50
2.위치 및 자세 시뮬레이션 결과 51
3.장애물 회피 시뮬레이션 조건 57
4.장애물 회피 시뮬레이션 결과 57
5.비행시험 64
VII결론 **67**
[참고문헌] **68**

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

ABSTRACT

VTOL Development for Obstacle Awareness and Collision Avoidance

Yang, You Young
Advisor : Prof. Henzeh Leeghim, Ph.D.
Department of Aerospace Engineering,
Graduate School of Chosun University

This thesis introduces a technique of quad-rotor for detecting and avoiding obstacles to reduce the possibility of collisions. The configuration of the drone considered in this thesis is based on a simple four-rotor system. So it is simple to design and manufacture. The design starts by selecting the appropriate combined component in the system configuration. Mechanical design shall be considered in the choice of propulsion systems as a combination of propellers, motors and batteries that deal with thrust generation. On the other hand, structural design should be robust and the weight of each component should be taken into account.

To become an unmanned system, flight control systems require high-speed onboard computers to process and control reliable and reliable data from proven, expensive GPS/INS sensors. So in this paper, we built our own flight control computer.

In this work, modeling of quad-rotors and designing controls are considered. Modeling is designed using two coordinate systems and Newton's law, while controls are designed using PID controllers. Since modelling and controller performance analysis and verification are required, the performance of the controller is verified by using numerical simulations.

In this thesis, collision-avoidance algorithm for drones based on Lidar sensors are also briefly investigated. The collision avoidance function shall ensure that the drone does not collide with the obstacle up to the target point. An collision-avoidance algorithm is developed by estimating the time of collision and relative speed between the drone and the obstacle. The obstacle detection and avoidance algorithms suggested in

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

this work are demonstrated by using numerical simulation. Finally, flight performance of the quad-rotor is demonstrated by flight test.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

I. 서론

최근 전 세계적으로 가장 이슈 중 하나인 드론은 과거에 군사용으로 개발 되었다. 최근에는 다양한 분야로의 활용 가능성이 높아지면서 산업 및 민간용 시장으로 빠르게 확산 되고 있으며 취미 · 레저용으로 점차 대중화 · 보편화되고 있다.[1] 드론이라 불리는 무인 항공기(Unmanned Aerial Vehicle, UAV)는 크게 고정익(Fixed-wing)방식과 회전익(Rorary-wing)방식으로 나눌 수 있다. 고정익은 비행속도, 비행거리가 회전익보다 길지만 회전익은 수직 이 · 착륙(Vertical Take-off and Landing) 및 제자리비행(Hovering)이 가능하여 공간의 제약을 받지 않는 장점이 있다.[2–4]

무인 항공기는 공중을 날아다니는 시스템이기 때문에 드론과의 충돌에 의한 부상이나, 조정 거리 이탈로 인한 추락 · 충돌 등이 있으며, 한국 소비자원 보고에 따르면 드론에 의한 안전사고 비율 중 50% 이상이 드론 충돌에 의한 사고이고 국내외에서 드론 충돌 사고 사례가 빈번히 보고되고 있다. 유명한 사례 중 이탈리아 두오모 성당에 쿼드로터가 충돌하였으며, 미국 백악관에 쿼드로터 충돌하는 사고가 있었다. 따라서 사고의 가능성을 줄이기 위해 산업 및 민간용 안전성과 신뢰성 확보가 우선되어야하며 원활한 임무수행을 위해서는 충돌 회피 기술이 필요로 하다.[5, 6]

최근 해외의 드론 업체에서 비행 중 발생하는 충돌 회피등에 대한 기술 개발 및 출시가 이루어지고 있으나 대부분 단거리의 장애물을 인식하는 레저용 드론이다. 국내에서는 충돌 회피에 관련된 연구가 증가하고 있지만 대부분이 지상 기반 기술이다.[7] 지상에서 비행체나 지상에서 관측된 데이터를 기반으로 회피 명령을 송신하는 형태이기 때문에 장애물 인식과 회피율이 높지 않고, 충돌 위험에 대한 대처가 빠르게 이루어지지 않는다는 단점이 있다. 산업용 드론의 경우 회피 등의 기능이 들어가 있지 않은 경우가 많고, 도심이나 장애물이 많은 지역에서 드론이 활용되기 위해서 안전에 대한 부분은 반드시 해결해야 하는 문제이다. 다양한 드론의 임무수행 중에 원활한 임무수행을 위해서는 충돌회피 기술은 필요하다. 이를 위해서 무인 항공기에서 얻을 수 있는 데이터를 기반으로 충돌을 예측하고 회피하는 알고리즘의 개발이 필요하다.[8] 장애물을

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

인식 하고 충돌을 회피하며 비행하는 방법으로는 크게 두 가지방법으로 나눌 수 있다.

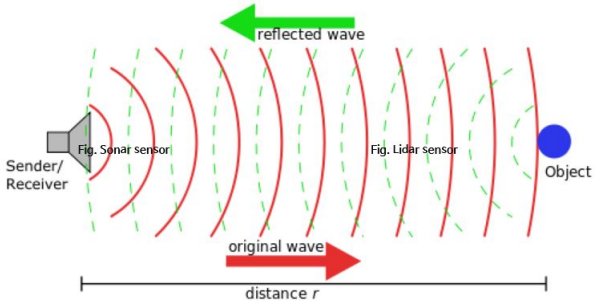


Figure. 1 Sonar sensor measurement method

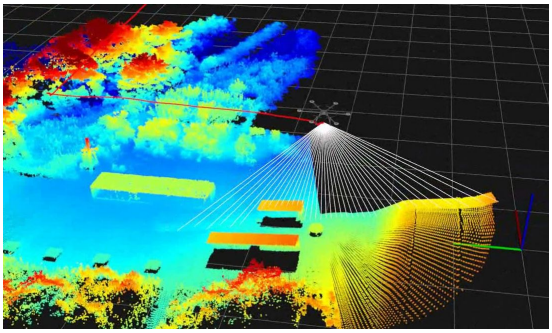


Figure. 2 Lidar sensor measurement method

Figure.1과 같이 초음파, IR 센서와 같은 간단한 센서를 사용하여 비행체를 단순히 장애물과 반대 방향으로 조종하여 충돌을 회피하는 방법이 있다.[9] 이 방법은 적은 양의 데이터 처리와 빠른 처리가 가능하지만 주변 환경 인식에 대한 신뢰도가 낮고 여러 상황에 필요한 임무 수행에 어려움이 있다. Figure.2와 같이 라이다(Lidar), 깊이 카메라(Depth Camera)와 같은 고성능 센서를 사용한 SLAM(Simulation Localization and Mapping) 기반의 방법이다.[10, 11] SLAM 기반의 방법은 주변 환경 인식에 대한 신뢰도가 높고 임무 수행에 제한은 줄어들지만 알고리즘의 복잡성 때문에 상당한 메모리와 계산 능력이 필요하다.

본 논문에서는 기존 연구들의 단점인 주변 환경 인식의 낮은 신뢰도와 고성능의 계산능력과 메모리가 필요를 없애고 비행 할 수 있는 장애물 회피 시스템을 제안한다. 제안하는 장애물

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

회피 시스템에는 2D-Lidar 센서를 사용한다. 2D-Lidar 센서의 원리는 레이저를 사용하여 일정 범위를 스캔하여 물체와의 거리를 측정하기 때문에 초음파, IR 센서들 보다 주변 환경을 인식할 때 신뢰도가 높고 3D 환경 데이터를 측정하는 3D-Lidar, 깊이 카메라보다 적은 양의 데이터를 처리하기 때문에 빠른 처리가 가능하다.

본 논문에서는 2D-Lidar 센서 기반으로 드론의 충돌 회피 알고리즘 연구 및 개발을 진행하였으며, 수직 이·착륙 및 제자리비행이 가능하여 공간의 제약을 받지 않는 회전익을 선택하였다. 회전익은 로터의 개수에 따라 트라이로터(Tri-rotor), 쿼드로터(Quad-rotor), 헥사로터(Hexa-rotor) 등으로 나뉜다.[12] 멀티로터 종류 중 가장 흔히 사용되며 상대적으로 동작원리가 간단하고 직관적인 쿼드로터를 사용하였다. 충돌 회피 기능은 드론이 목표 지점에 이르기까지 장애물과의 충돌이 발생하지 않도록 보장한다. 이를 위해서는 크게 두 가지 문제에 대한 방법을 결정해야한다. 첫 번째로는 드론의 주변 환경을 감지하고 장애물 회피에 대한 유용한 정보를 추출하는 방법을 결정해야 하고, 두 번째는 어떻게 충돌회피가 수행되는지, 그리고 충돌 회피 단계를 언제 시작하고 중지할지 결정해야한다. 이러한 문제를 기반으로 충돌 회피는 크게 충돌 회피와 기동으로 두 가지 구성요소로 나눌 수 있다.

본 논문에서는 드론과 장애물에 대한 정보기반으로 장애물과의 충돌 예상 시간 및 상대속도를 이용한 회피 알고리즘을 개발하였다. 회피 알고리즘에서 드론은 장애물과의 충돌까지 걸리는 시간 및 장애물과의 거리를 기준으로 탐지모드와 추적모드, 회피모드로 나누어 비행한다.

본 논문에서는 다음과 같이 구성되어 있다. 먼저 2장에서는 쿼드로터의 전체적인 시스템과 비행원리 그리고 구성되어 있는 구성요에 대해 소개한다. 3장에서 쿼드로터를 표현하는 좌표계와 뉴턴-오일러 방정식(Newton-Euler Equation)을 사용하여 쿼드로터의 동역학적 모델링에 대해 소개한다. 4장에서는 PID 제어기의 대한 설명과 PID 제어기를 사용하여 쿼드로터의 위치, 자세 제어기를 설계한다. 그리고 5장에서 2D-Lidar 센서 기반으로 쿼드로터의 장애물 충돌 회피 알고리즘에 설계한다. 6장에서는 앞에서 설계한 PID 제어기와 충돌 회피 알고리즘에 대한 수치 시뮬레이션을 통해서 확인하고 자체적으로 제작한 비행 제어 컴퓨터(Flight Control Computer, FCC) Uribee에 검증한 알고리즘을 넣어 실제 비행시험을 통해 성능을 검증하고 6장에서 결론으로 마무리한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

II. 쿼드로터 구성

1. 시스템 구성

쿼드로터 시스템 구성도는 Fig.3와 같이 구성되어 있다:

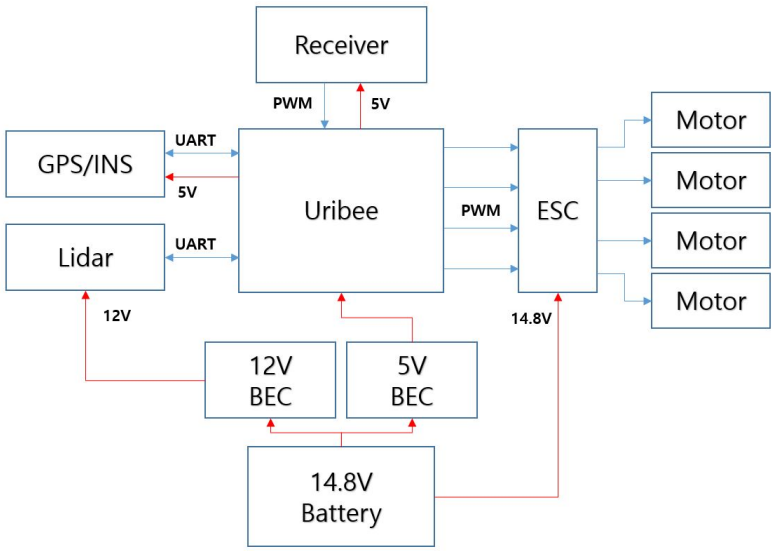


Figure. 3 System configuration

- 1개의 자체 제작한 Uribee(FCC), 온보드 처리 시스템이며 5V의 동작전압이 공급된다.
- BLDC 모터(Brushless DC Motor), 전자속도제어기(Electronic Speed Controller, ESC) 프로펠러 각각 4개의 구동기로 이루어져있다. ESC에 14.8V의 동작전압이 공급된다.
- 1개의 GPS/INS(GPS/Inertial Navigation System) 통합 항법 시스템, 3축 가속도계, 3축 자이로스코프, 3축 자력계 및 기압계 그리고 위치, 속도 및 방향 데이터를 제공한다. 5V의 동작전압이 공급된다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

- 1개의 Lidar(Light Detection and Ranging) 센서, 8개의 세그먼트로 개체를 탐지한 데이터를 제공한다. 12V의 동작전압이 공급된다.
- 지상에서 조이스틱 명령을 수신하기 위한 1개의 RC 수신기
- 배터리에서 공급하는 전압은 14.8V이며 5V, 12V의 레귤레이터를 사용하여 보드 및 센서에 전원을 공급한다.

2. 비행원리

본 논문에서 사용한 쿼드로터는 Fig.4와 같이 이 가장 널리 사용되는 형태인 X형 쿼드로터를 사용하였다.[13] 앞뒤로 2개씩의 프로펠러가 있고, 그 가운데 X형태의 기체가 있다. X형 쿼드로터는 대칭이며, 간단히 모터의 속도만 바꾸면 3차원 이동 및 회전을 제어할 수 있고 단순한 비행부터 다양한 임무를 수행할 수 있다는 장점을 가지고 있다. 쿼드로터는 시계방향(Clockwise, CW)으로 회전하는 1,3 번 로터와 반시계방향(Counterclockwise, CCW)으로 회전하는 2, 4번 로터, 총 4개의 로터 추력으로 3차원 공간에서 직선운동과 회전운동을 갖는 6자유도 운동을 한다.[14]

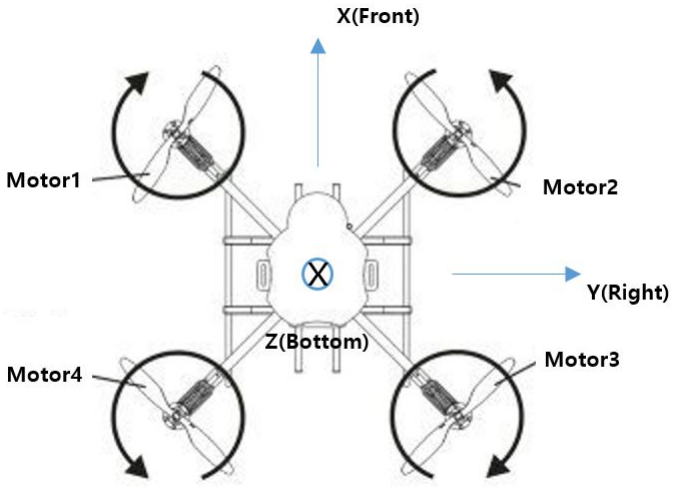


Figure. 4 Motor loacation and rotation

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

1) 수직 이착륙 및 제자리비행

쿼드로터의 운동은 프로펠러의 회전수에 따라 결정 되며, 수직 이착륙 및 제자리 비행은 Z축 기준으로 운동을 하며 4개의 모터의 속도가 동일하게 회전해야한다. 제자리 비행(Hovering)하는 동안 양력과 중력, 추력과 항력은 모두 평형을 이룬다. 모든 프로펠러가 동일한 속도로 회전하여 추력을 증가시켜 양력이 중력보다 크게 되므로 상승비행을 하고, 반대로 추력을 감소시키면 하강비행을 한다.

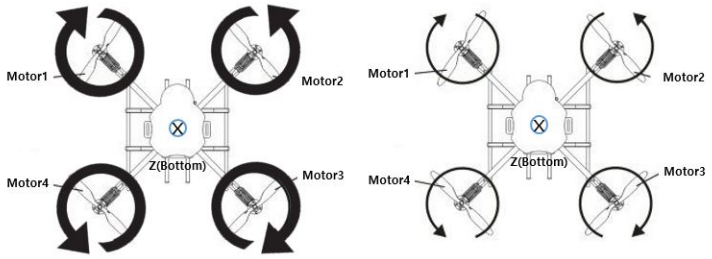


Figure. 5 VTOL and Hovering

2) Roll 운동

Roll 운동은 X축을 기준으로 1,4번 모터와 2,3번 모터의 속도를 조정하여 좌우로 비행을 한다. 1,4번 모터의 회전속도를 증가하고 2,3번 모터의 회전속도를 감소하면 기체가 우측으로 비행하고 반대로 1,4번 모터의 회전속도를 감소하고 2,3번 모터의 회전속도를 증가하여 좌측으로 비행한다.

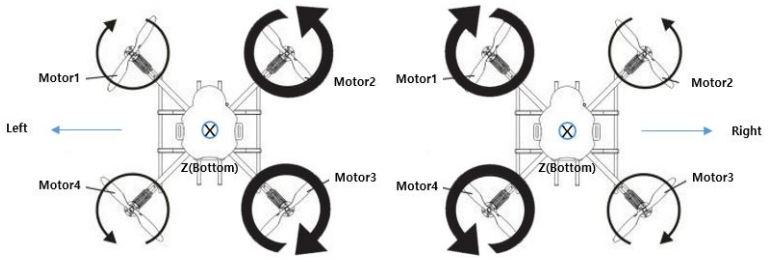


Figure. 6 Roll motion

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

3) Pitch 운동

Pitch 운동은 Y축을 기준으로 1,2번 모터와 3,4번 모터의 속도를 조정하여 전진, 후진 비행을 한다. 3,4번 모터의 회전속도를 증가하고 1,2번 모터의 회전속도를 감소하면 기체가 전진비행을 하고 반대로 3,4번 모터의 회전속도를 감소하고 1,2번 모터의 회전속도를 증가하여 후진비행을 한다.

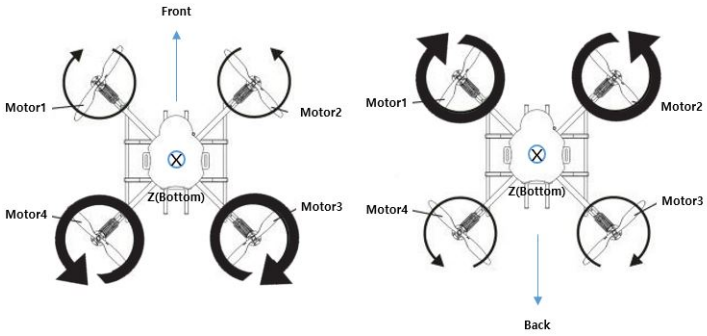


Figure. 7 Pitch motion

4) Yaw 운동

Yaw 운동은 Z축을 기준으로 시계방향으로 회전하는 1,3번 모터와 반시계방향으로 회전하는 2,4번 모터의 속도를 조정하면 토크의 균형이 깨지며 Yaw 모멘트가 발생한다. 1,3번 모터의 회전속도를 증가하고 2,4번 모터의 회전속도를 감소하면 기체가 반시계방향으로 회전하면서 비행하고 반대로 1,3번 모터의 회전속도를 감소하고 2,4번 모터의 회전속도를 증가하면 시계방향으로 회전하면서 비행한다.

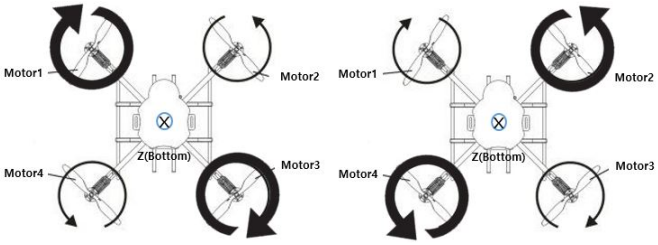


Figure. 8 Yaw motion

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

3. 구성 요소

본 논문에서 제작한 쿼드로터의 구성요소로는 기체와 BLDC 모터(Brushless DC Motor), 전자속도제어기(Electronic Speed Controller, ESC), 프로펠러는 각각 4개씩으로 이루어져 있으며, 비행제어컴퓨터(Flight Control Computer, FCC) 1개와 GPS/INS, Lidar 센서로 이루어져있다. 배터리에서 공급하게 되는 전압은 14.8V(4-cell)이고 12V와 5V로 변환해주는 BEC(Battery Eliminator Circuit)를 사용하여 FCC와 센서에 전원을 공급한다. 마지막으로 쿼드로터에 명령을 수신하기 위한 RC 수신기로 구성되어있다.

1) 기체

멀티로터 기체의 종류로는 트라이로터(Tri-rotor), 쿼드로터(Quad-rotor), 헥사로터(Hexa-rotor)등이 있다. 기체선택은 임무와 용도에 맞게 선택을 한다. 본 논문에서는 선택한 쿼드로터 기체는 S500 프레임 모델을 사용하였다. 기체 프레임은 지름은 520 mm, 앞 모터와 뒤 모터 사이 거리는 330 mm, 랜딩기어 높이는 250 mm이며, 프레임의 무게는 454 g이고 탄소섬유 재질로 되어있다. S500 프레임 모델은 Fig.9 와 같고 제원은 Table.1 와 같다.

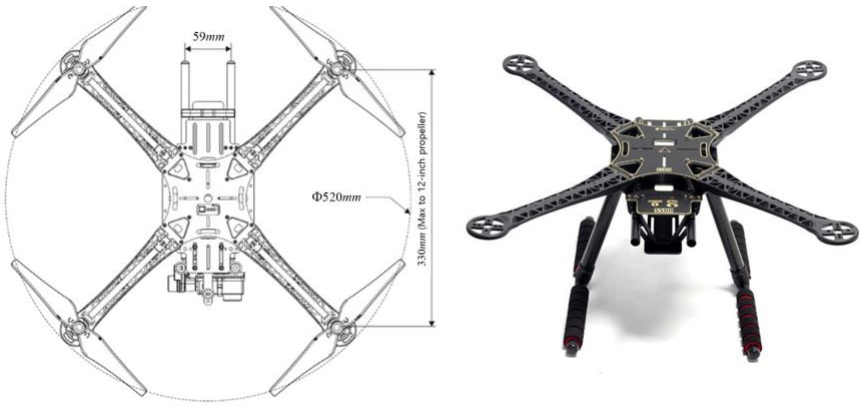


Figure. 9 S500 frame

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Table 1: S500 frame specifications

Properties	Value
Weight	454 g
Wheelbase	520 mm
Height	250 mm

2) 구동기

쿼드로터의 구동기에는 모터, ESC, 프로펠러로 구성되어있다. 구동기는 비행성에 가장 큰 영향을 좌우하며, 제자리 비행을 하거나 원하는 방향으로 비행할 수 있게 속도에 대한 조절을 수행하는 구성요소들이다. 모터는 영구 자석과 전기가 전선을 통과할 때 발생하는 자력 사이의 당기는 힘과 미는 힘을 이용해 회전하는 장치이며 자석을 사용하기 때문에 전력 밀도와 효율성이 상대적으로 높다. 멀티로터에 사용되는 모터는 크게 브러시 모터(Brushed motor)와 브러시리스 모터(Brushless motor) 두 가지 종류가 있다. Brush의 역할은 모터가 한번 회전하고 반대방향으로 회전했을 때 전류의 흐름을 전환 시켜 동일한 방향으로 회전을 계속할 수 있게 만들어 주는 역할을 한다. 모터 내부의 전자석에 전기를 공급하는 부분에 브러시 여부에 따라 브러시 모터와 브러시리스 모터로 나뉜다. 브러시 모터는 기계적 접점이 있기 때문에 전기불꽃(Spark), 마찰에 의한 에너지 손실, 회전 소음, 마모 등의 수명 문제가 있다. BLDC 모터는 홀센서로부터 모터의 속도를 정확히 알 수 있으므로 일정한 속도제어 및 가변속 제어가 용이하며 모터 수명에 영향을 주는 브러시가 없기 때문에 반영구적인 수명을 가지고 있고 고속회전에 문제가 없으며 모터의 소형화와 경량화가 가능하다. BLDC 모터는 쿼드로터와 같은 드론에 많이 사용된다.[15]

적절한 모터를 선택하기 위해서는 RPM, 토크, 중량을 고려해야 한다. Fig. 10은 일정한 인가전압을 가진 DC모터의 일반화된 토크대 속도 그래프이다.[16] 모터가 완전히 제동될 때 최대 토크가 발생하고 모터가 최대 회전일 때 토크가 없다. DC모터는 기계적 전력을 가장 많이 생성할 때 최고의 성능을 발휘한다. 전력은 토크와 모터의 회전속도의 곱으로 Eq.1 와 같다.

$$\text{Power} = \text{Torque} \times \text{Rotational speed} \tag{1}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

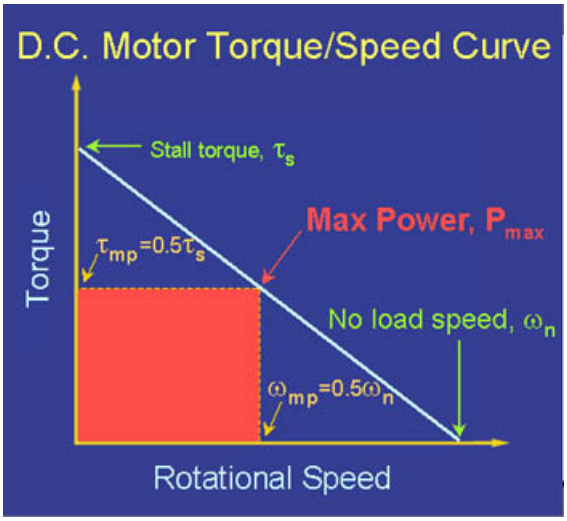


Figure. 10 Relationship between the motor speed and torque

모터의 최대 출력을 찾기 위해서는 최대 출력이 발생하는 회전 속도를 찾는 것이 필요하다. Fig. 10에서 일정한 전압이 주어지면 모터의 최대 성능의 절반인 토크및 회전 속도에서 최대 전력이 달성된다. 최대 전력 출력은 모터의 최상의 성능범위를 항공기의 가장 일반적인 비행 모드와 일치시키는 바람직하기 때문에 중요하다. 쿼드로터의 경우 일반적인 비행모드는 호버링이다. 호버링은 프로펠러,모터가 한 세트로 항공기의 중량과 같은 추력을 생성 할 때 발생하며 배터리의 전압의 절반에 해당하는 모터의 최대 성능의 50%에서 발생한다. DC모터는 KV(rpm/v)로 정격되므로 최대 전력이 달성되는 회전 속도는 Eq.2 과 같다. 원하는 RPM은 최대 전압에서 모터의 최대 RPM의 1/4에서 발생한다. 원하는 RPM은 프로펠러 및 모터 선택에서 사용된다. 그러나 적절한 프로펠러와 모터를 선택하기 전에 항공기의 질량을 추정해야한다. 모터의 추력은 KV, 배터리의 전압, 프로펠러에 따라 정해진다. 여기서 KV는 모터의 성능을 나타내는 지수로써 주어진 전압 대비 회전율 즉, 1V당 회전수(RPM)를 의미한다.[17]

$$\mathrm{RPM}_{\mathrm{maxPower}} = \frac{\mathrm{KV} \times 0.5 \times \text{Battery Volts}}{2} \tag{2}$$

본 논문에서 사용한 모터는 T-Motor사의 AIR 2213/920KV 제품으로 Fig.11와 같고 제원은 Table.2 와 같다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Figure. 11 T-motor AIR 2213/920KV BLDC motor

Table 2: BLDC motor specifications

Properties	**Value**
KV	920
Idle Current(10V)	0.5 A
No.of Cell(Lipo)	3-4 S
Max Continuous Current	18 A
Max Continuous Power	230 W
Max. Efficiency Current	(3-10A)>83%
Weight	54 g
Dimension	Φ27.5×30 mm

BLDC 모터의 회전 속도를 제어하기 위해서는 전자속도제어기(ESC)가 필요하며 변속기라 말한다. 멀티로터에 사용되는 모터는 주로 3개의 선이 달린 Outrunner BLDC 모터이다. 변속기는 배터리에서 전원 인가 받아 FCC로부터 입력 받은 펄스 폭 변조(Pulse Width Modulation, PWM) 신호에 따라 3상 주파수를 발생시켜 모터의 속도 제어를 한다. 모터는 회전 명령을 받는 데로 프로펠러를 돌린다. 모터에 어떤 무리한 부하가 발생하더라도 신호에 따라 회전하려고 한다. 발생한 부하 때문에 많은 양의 전류가 필요하며 전류량이 변속기의 허용하는 양보다 크면 변속기는 고장 난다. 그래서 모터가 필요로 하는 최대 전류량보다 큰 변속기를 선택해야하고 변속기의 허용전류보다 큰 용량을 소화할 수 있는 배터리를 선택해야한다. 배터리의 경우는 일

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

정 전압 이하로 떨어지면 수명이 단축되거나 망가질 수 있기 때문이다. 따라서 변속기를 선택을 위해서 가장 중요한 부분은 허용 가능한 전류량이다.

BLDC 모터는 14.8V(4-cell)와 9.5x4.5 inch 프로펠러를 사용하였다. 스로틀이 100%일 때 전류는 14.3 A, RPM은 10,200 RPM이고 추력은 1,084 g이다. 수치들은 모터 제작사에서 제공한 정보이다. 본 논문에서 사용한 변속기는 AIR 20A ESC 제품이며 Fig.와 같다. 선택한 변속기는 14 g이며, 최대 621 Hz의 신호 주파수를 지원한다.

Figure. 12 T-motor AIR 20A ESC

Table 3: ESC specifications

Properties	**Value**
Max Frequency	621 Hz
Current	20 A
Peak Current(10s)	30 A
BEC	No
No.of Cell(Lipo)	3-4 S
Weight	14 g
Dimension	52.4×21.5×7 mm

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

프로펠러는 가볍고 적은 전력을 소모하면서 추력을 발생시키기에 가정 적합한 형태이다. 프로펠러는 2개 또는 3개의 블레이드 형태가 있고 회전은 일반적으로 시계방향(Clockwise, CW)과 반시계 방향(Counter Clockwise, CCW)로 구성되어있다. 블레이드의 길이는 작을수록 관성모멘트가 작아 기동성이 높은 드론에 적합하다. 프로펠러에 의해 생산되는 추력은 공기밀도, 프로펠러의 RPM, 반경, 면적 그리고 피치에 의해 좌우되며 프로펠러의 효율은 받음각(Angle of attack)에 따라 달라지며 입력과 출력의 비율로 정의된다. 멀티로터의 프로펠러는 일반적으로 XXYY또는 XX × YY로 표현하고 Fig.13와 같이 나타낼 수 있다. XX는 프로펠러 전체 길이를 나타내고, YY는 프로펠러의 피치 즉 프로펠러가 한 바퀴를 돌았을 때 전진하는 길이를 나타낸다. 프로펠러의 길이가 길어지면 큰 양력을 얻을 수 있고, 피치길이가 길어지면 작은 회전수에도 전진 거리가 길어지기 때문에 빠르게 양력을 얻을 수 있다. 성능으로 판단하면 크고 높은 피치의 프로펠러가 이상적으로 판단되지만 커질수록 프로펠러의 무게도 증가하여 모터 부하가 증가하게 되어 비행시간이 단축되므로 사양에 맞는 적당한 프로펠러를 선정해야한다. 본 논문에서는 사용된 프로펠러는 T9545로 Fig.14 와 같다. 프로펠러 길이 9.5 inch에 피치 길이 4.5 inch를 가지고 13.6 g이고 제원은 Table.4 와 같다.

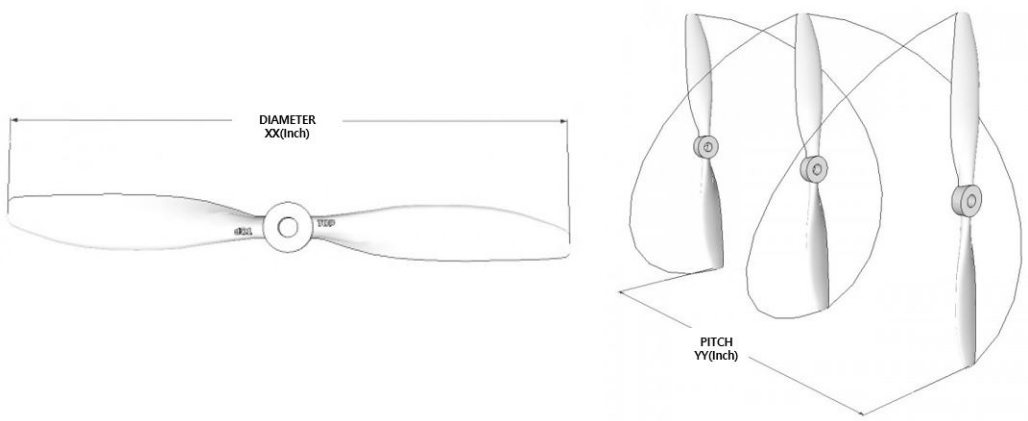


Figure. 13 Propeller inch and pitch

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Figure. 14 T-motor AIR T9545 propeller

Table 4: Propeller specifications

Properties	**Value**
Weight	12 g
Dimension	9.5×4.5 inch

3) FCC

비행제어컴퓨터(Flight Control Computer, FCC)는 수신기로 수신된 데이터 명령을 받아서 장착된 센서 데이터를 분석하여 제어 알고리즘을 통해 안정적인 자세를 유지하도록 변속기에 PWM 신호를 처리하는 역할을 하고 있다. FCC는 마이크로프로세서, 입출력 단자등으로 구성되어있다.

본 논문에서는 자체적으로 제작한 FCC를 사용하였으며, 제작된 FCC의 이름은 본 연구에서 Uribee라고 명명하였으며, Fig.15,16 와 같다. Uribee에는 마이크로프로세서 STM32F767VIT6를 부착되어있고, 총 84개의 핀이 있다. UART, I2C, SPI, ADC, PWM이 가능하며 Uribee의 제원은 Table.5 와 같다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

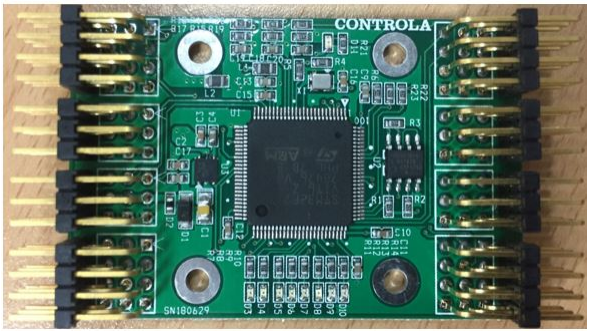


Figure. 15 The Developed Fight Controller(Uribee)

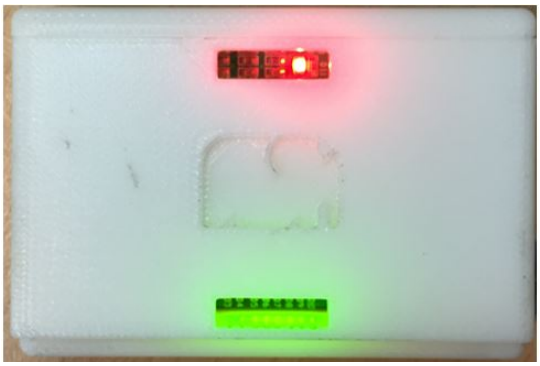


Figure. 16 Casing a Uribee

Table 5: Uribee Specifications

Properties	**Value**
Microcontroller	STM32F767VIT6
Data Bus Width	32 bit
Maximum Frequency	216 MHz
Flash Memory	2 MB
RAM	512 KB
GPIO Count	48pin GPIO
Operating Voltage	5V
Weight(include case)	18 g(39g)
Dimension	60×40×10 mm

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

4) 센서

드론이 비행을 위해서는 기본적으로 자세, 위치, 고도, 속도, 방향, 장애물 등의 정보가 필요하다. 드론을 조종하기 위해서도 필요하지만 안정적으로 비행 및 호버링 하는데도 중요한 정보이다. 드론의 방향과 자세를 측정하기 위한 센서로는 3축 가속도(3-axis accelerometer), 3축 자이로스코프(3-axis gyroscope), 3축 지자계(3-axis magnetometer)가 있고, 위치 및 고도를 측정하기 위해서는 기압계(Barometer), GPS, 초음파, 라이다(Lidar) 등이 있다.

본 논문에서는 가속도, 자이로, 지자계, 기압계, GPS 센서가 통합되어있는 Fig.17와 같이 VECTORNAV사의 VN-200와 Fig.18와 같이 LeddarTech사의 LeddarVu 라아다 센서를 사용하였다. VN-200은 200Hz로 GPS/INS 칼만필터를 실행하는 내장형 마이크로프로세서를 가지고 있으며 제원은 Table.6 와 같다. LeddarVu는 라이다센서로 Light Detection and Ranging이라 하며 8개의 세그먼트로 동시에 펄스 레이저를 여러 목표물에 방출하고 빛이 돌아오기까지 걸리는 시간 및 강도를 측정해 거리, 방향, 속도, 온도 등을 감지하는 기술이다. 제원은 Table.7 와 같다.

Figure. 17 VECTORNAV VN-200

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Table 6: VN-200 Specifications

Navigation		GPS	
Properties	**Value**	**Properties**	**Value**
Static Accuracy (Heading, Magnetic)	2.0°RMS	Receiver Type	±2.5 Gauss
Static Accuracy (Pitch/Roll)	0.5°RMS	Time to First Fix	Cold/Warm Start: 36s Hot Start: <1s
Dynamic Accuracy (Heading, True Inertial)	0.3°RMS	Tracking & Navigation Sensitivity	-159 dBm
Dynamic Accuracy (Pitch/Roll)	0.1°RMS	Time pulse Signal Accuracy	30 nsec RMS
Velocity Accuracy	±0.05 m/s		

	Accelerometer	**Gyroscope**	**Magnetometer**
Range	±16 g	±2000°/s	±2.5 Gauss
Linearity	< 0.1% FS	< 0.5° FS	< 0.1%
Noise Density	$< 0.14\text{mg}/\sqrt{\text{Hz}}$	$0.0035°/s\sqrt{\text{Hz}}$	$140\mu\text{Gauss}/\sqrt{\text{Hz}}$
Alignment Error	±0.05°	±0.05°	±0.05°
Resolution	< 0.5 mg	$< 0.02°/s$	1.5milliGauss

Properties	**Value**
Operating Temp	$-40°C$ to $80°C$
Digital Interface	512 KB
GPIO Count	Serial TTL, RS-232
Input Voltage	3.3V to 17V
Weight	16 g
Dimension	36×33×9.5 mm

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Figure. 18 LeddarTech LeddarVu

Table 7: LeddarVu Specifications

Properties	**Value**
Horizontal FOV	100°
Vertical FOV	0.3°
Retro Reflector Range	61 m
White Target Range	12 m
Gray Target Range	7 m
Accuracy	5 cm
Data refresh rate	5 Up to 100 Hz
Power supply	12VDC
Weight	128.5 g
Dimension	73×40×65 mm

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

5) 배터리

드론에 사용되는 배터리는 리튬 계통 배터리 일부 사용되지만 대부분 리튬폴리머(Lithium polymer, LiPo)를 사용한다. LiPo 배터리는 무게에 비해 용량이 크고, 방출률이 높다. LiPo 배터리의 각 셀은 3.7V의 전압을 가지며 각 셀이 3.0V에 도달 할 때 완전히 방전된다. 배터리의 용량(Battery Capacity)은 밀리 암페어(mAh) 단위를 사용하며 용량이 클수록 비행시간이 길어지지만 배터리 무게도 커진다. 배터리의 방전율(Battery Discharge)은 C로 표시되고 배터리 용량 대비 공급할 수 있는 전류의 비율을 뜻한다. 드론에 사용되는 모터는 높은 전류를 소모하므로 높은 방출률의 배터리를 사용한다. 비행 시간을 계산하여 적절한 배터리를 결정한다. 비행 시간은 배터리 용량과 직접적으로 비례한다. 비행 시간 Eq.3은 다음과 같다.

$$\text{Flight time} = \frac{\text{Battery Capacity} \times \text{Battery Discharge}}{\text{Average Amp Draw}} \times \left(\frac{60\,\text{min}}{\text{hour}}\right) \tag{3}$$

일반적으로 배터리는 20%이하로 사용하지 않기 때문에 80%를 사용한다고 가정했을 때 Battery Discharge는 0.8이다. 평균 전류 감소(Average Amp Draw)은 비행을 할 때 필요한 전류의 양을 뜻하며 드론 전체 중량과 그 중량을 띄울 수 있는 모터의 소비 전류 값을 사용한다.

본 논문에서는 Fig.19 와 같이 Sky-holic에 14.8V(4-cell), 3,200 mAh, 40C의 리튬배터리를 사용하였으며 제원은 Table.8 과 같다.

Figure. 19 Sky-holic 3,200mAh 14.8V Battery

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Table 8: Battery Specifications

Properties	Value
Voltage	14.8V(4-cell)
Discharge	40-70 C(max)
Weight	378 g
Dimension	146×50×31 mm

6) RC 수신기

조종기에서 전달된 명령이 라디오 전파(Radio Frequency, RF)로 수신기에 전달되고 수신된 신호는 FCC로 보내진다. 수신기에서 전달되는 신호 전달 방식으로는 PWM(Pulse Width Modulation), PPM(Pulse Position Modulation)등이 있으며 Fig.20 와 같다. PWM방식은 최소 4-5개의 채널을 지원하며, 하나의 핀 당 하나의 채널이 할당된다. PWM은 펄스 폭 변조를 나타내며 일정 간격 동안 신호가 들어오는 시간을 검출하는 방식이다. 신호 펄스의 길이는 일반적으로 1000us 2000us사이에서 변한다. PPM 수신기는 일정 간격동안 어느 위치에 신호가 들어오는지를 검출하는 방식이며, CPPM 또는 PPMSUM이라고 하고 장점은 여러 개의 선 대신 여러 채널에 하나의 신호선만 필요하다. 기본적으로 PWM 신호가 동일한 선에서 하나씩 전송되는 PPM 신호이지만 신호는 다르게 변조된다.

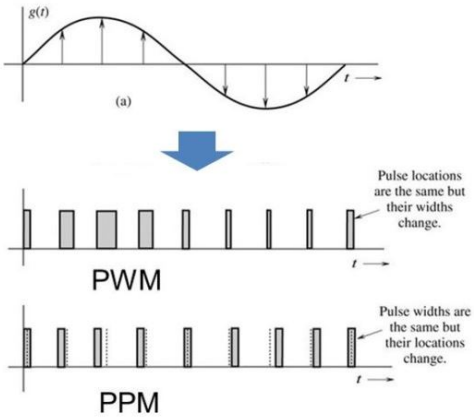


Figure. 20 PWM and PPM

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

본 논문에서 사용한 수신기는 Graupner사에 GR-16L이며 Fig.21 와 같다. 수신기는 PWM 방식을 사용하며 PWM을 포함한 출력타입이 총5개가 있다. 제원은 Table.9 과 같다.

Figure. 21 Graupner GR-16L

Table 9: Receiver Specifications

Properties	Value
Frequency	2400 MHz
Modulation	2.4 GHz FHSS
Range	4000 m
Operation voltage	3.6-8.4V
Weight	12 g
Dimension	46×21×14 mm

4. 임베디드 시스템

1) GPIO

GPIO(General Pupose Input Output)는 마이크로프로세서가 주변장치와 통신하기 위해 범용으로 사용되는 입출력 포트를 말한다. 마이크로 컨트롤러의 핀을 입력 또는 출력 용도로 사용할 수 있게 하는 기능이며 핀을 통해 0(0V)과 1(5V)의 디지털 출력 신호를 보낸다. 간단하게 GPIO를 이용해서 LED 출력 제어나 스위치 ON/OFF 입력 확인을 할 수 있다. GPIO는 STM32의 경우 RCC, GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_BSRR 레지스터 설정을 통해 사용할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

2) UART

UART(Universal Asynchronous Receiver/Transmitter)는 컴퓨터에서 다른 주변장치와 데이터 교환하기 위한 통신방법 중에 가장 널리 쓰이는 통신 방식이며 병렬 데이터의 형태를 직렬 방식으로 전환하여 데이터를 전송한다. UART는 비동기(Asynchronous) 통신이기 때문에 별도의 클럭이 없어서 통신하는 양쪽 기기의 속도를 동일하게 맞춰야하고 1:1 통신만 가능하다. 통신 속도는 기기에 따라 다르며, 일반적으로 115,200bps을 사용한다. 통신거리는 5m이하이고 통신을 위해 TX, RX라인 있다. TX는 전달하려는 데이터를 다른 기기로 보내는 선이고, RX는 다른 기기로부터 데이터를 받는 선이다. UART 인터페이스 다이어그램 Fig.22 와 같고, 두 기기의 TX, RX를 교차가 되도록 연결을 한다.

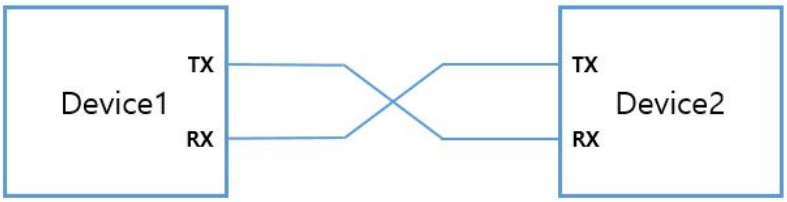


Figure. 22 UART Interface Diagram

UART의 프로토콜(protocol)의 시작을 의미하는 Start bit은 신호를 High에서 Low상태로 1bit 길이를 유지한다. 보내는 데이터의 길이는 5 9bit이며, 한번 길이를 정하면 계속 같은 크기의 데이터로 송수신해야한다. 데이터 전송이 완료되면 오류 검출을 위해 패리티 비트(Parity bit)를 보내지만 생략가능하다. Stop bit는 신호를 High로 하여 완료한다. Stop bit의 길이는 1bit, 1.5bit, 2bit가 있다. 일반적으로 데이터 길이는 8bit, 패리티 비트는 생략, Stop bit는 1bit를 사용하고 Fig.23 와 같이 나타낸다. STM32에서 UART를 사용하기 위해서는 USART_CRx, USART_BRR, USART_RDR, USART_TD 레지스터 설정을 통해 사용할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

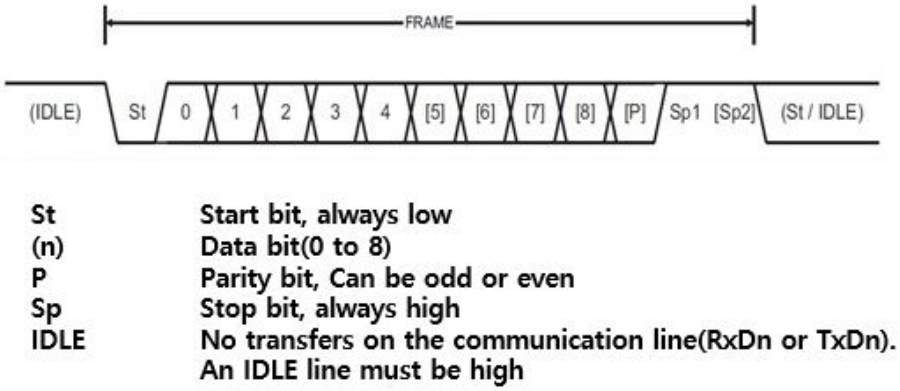


Figure. 23 UART Protocol

3) I2C

I2C(Inter Integrated Circuit)는 직렬 버스이며 빠른 속도를 요구하지 않는 간단하고 저비용의 기기들에 적합하다. 일반적으로 통신 속도는 100kbps, 400kbps, 3.4Mbps이다. 비동기식 통신인 UART와 달리 I2C는 동기식 통신으로 마스터와 슬레이브의 클럭을 맞춰야하고 N개의 마스터, 슬레이브 기기와 통신이 가능하다. I2C의 가장 장점은 SCL, SDA 2개의 라인으로 여러 기기와 연결하여 통신이 가능하며 Fig.24와 같다. SCL은 데이터를 전달하기 위한 클럭을 전달하는 신호선이며 단방향이다. SDA는 전달하고자하는 데이터의 비트 정보를 전달하는 신호선이며 양방향이다. 마스터와 슬레이브에 각각 SCL와 SDA를 바로 연결하게 되면 신호가 High도 아니고 Low도 아닌 플로팅 현상으로 불안정하게 변하기 때문에 풀업저항(Pull up register)을 연결해 두 개의 양방향 오픈 컬렉터 라인을 사용한다. 버스 커패시턴스(Bus Capacitance) 값에 따라 풀업저항의 최댓값이 제한된다.

I2C의 프로토콜 구조는 UART와 SPI보다 복잡하며 UART와 SPI에서 문제가 되는 부분들을 소프트웨어적으로 처리한다. I2C는 주소(Address)프레임과 데이터프레임 2개로 구분할 수 있다. 주소프레임은 마스터에서 전송하는 데이터가 원하는 슬레이브에 전송하는지 확인하기 위한 프레임이다. 데이터프레임은 전송하고자하는 데이터가 담긴 프레임이다. 데이터 통신의 시작을 위해서는 마스터에서 SCL이 High상태일 때 SDA가 High에서 Low상태로 변경할 때

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

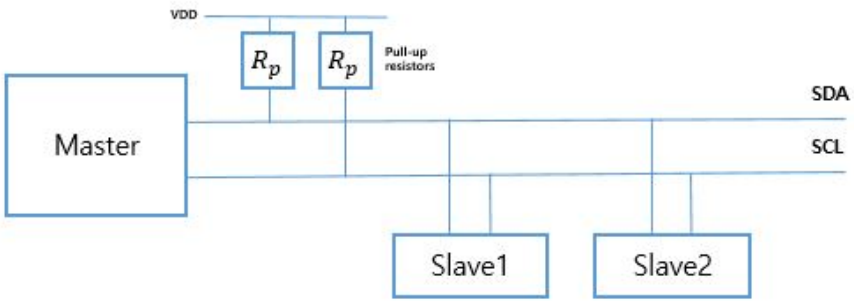


Figure. 24 I2C Interface Diagram

START conditions가 되며 이 신호는 모든 슬레이브 기기들에게 데이터를 전송할 준비가 되었다는 신호로 인식된다. 만약 회로에 여러 개의 마스터가 일 때는 먼저 SDA라인을 Low로 상태로 변경하는 마스터가 제어 권한을 갖는다. 주소프레임은 항상 모든 통신 데이터의 앞에 위치하며 처음 수신되는 1byte중 앞선 7bit 데이터가 Slave 주소(Address)를 나타낸다. 8번째 1bit(R/W)는 읽기(Read) 데이터 요청 또는 쓰기(Write) 데이터 전송을 의미한다. 9번째 bit는 NACK/ACK bit이며 마스터는 이 신호를 보고 성공인지 실패인지 판단한다. NACK/ACK bit는 주소프레임, 데이터프레임의 끝에 항상 위치한다. 주소프레임의 전송이 완료되면 데이터프레임의 전송이 시작된다. 마스터 장치는 SCL라인의 클럭 신호를 계속 생성하며 SDA라인에서 데이터를 양방향으로 전송한다. SDA라인의 데이터프레임은 Read, Write상태를 나타내는 bit 값에 따라 마스터 또는 슬레이브로 사용한다. 모든 데이터가 전송되면 마스터에서 종료를 위해 SCL이 High상태일 때 SDA가 Low에서 High상태로 변경할 때 STOP conditions가 되며 통신을 종료한다. 종료신호 오류방지를 위해 반드시 모든 기기는 SCL라인이 High인 상태에서 SDA라인의 상태를 변경해서는 안된다. I2C 프로토콜은 Fig.25 와 같이 나타낸다. STM32에서 I2C를 사용하기 위해서는 I2C_CRx, I2C_OARx, I2C_ISR, I2C_RXDR, I2C_TXDR 레지스터 설정을 통해 사용할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

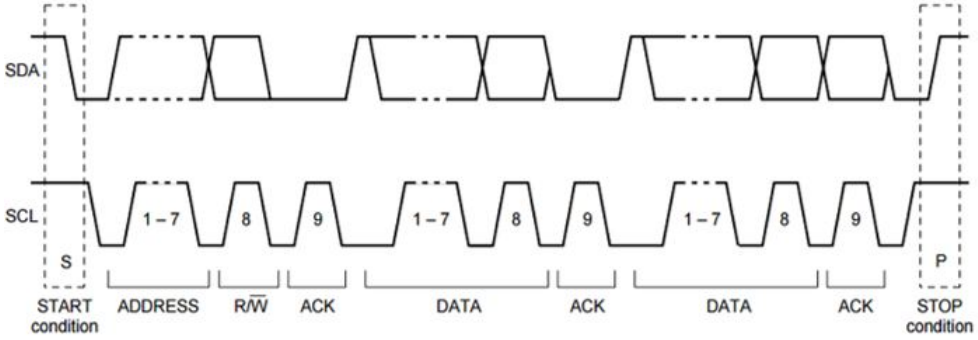


Figure. 25 I2C Protocol

4) SPI

SPI(Serial Peripheral Interconnect)는 전이중(Full duplex)이며 1:N 동기식 통신방식이다. 마스터, 슬레이브 방식으로 동작하며 마스터가 동기를 위한 클럭을 출력한다. 각 슬레이브 기기는 chip enable(/CE)입력을 가지고 있으며 입력이 활성화 되었을 때에만 동작한다. 마스터는 여러 개의 Slave select(SS)선을 슬레이브들의 /CE에 연결하고 하나의 슬레이브를 선택하는 방법을 사용하여 두 개 이상의 슬레이브를 구동할 수 있다. 따라서 I2C 슬레이브와 달리 SPI 슬레이브는 자체 주소를 가지고 있지 않아 주소 충돌의 문제가 없다. SPI는 기본적으로 SCLK, MOSI, MISO, SS 4개의 라인을 사용하며 마스터에 연결된 슬레이브의 수만큼 SS 라인이 증가한다. SCLK는 직렬 클럭으로 마스터가 출력하는 동기용 클럭선 이다. MOSI는 마스터에서 슬레이로 가는 데이터 선이고 MISO는 슬레이브에서 마스터로 가는 데이터선이다. SS는 마스터와 데이터를 송수신할 슬레이브를 선택하기 위한 신호선이다. SPI 인터페이스 다이어그램은 Fig.26와 같이 슬레이브가 1개일 때와 2개 이상일 때의 모습을 나타내고 있다.

SPI 통신을 시작하기 위해서 마스터는 원하는 슬레이브와 연결된 SS라인을 Low로 신호로 변경하여 해당 슬레이브를 활성화 시킨다. 마스터는 SLCK로 동기를 위한 클럭을 출력한다. 클럭에 맞추어 MOSI로 명령어와 주소 데이터를 한 비트씩 전송한다. 양방향 통신이기 때문에 마스터는 SCLK 클럭에 맞추어 동시에 MOSI로 데이터를 내보내고 MISO로는 데이터를 수신한다. 동일하게 슬레이브도 클럭신호에 맞추어 데이터를 송수신한다. 일반적으로 SPI통신은 8bit 단위로 데이터를 전송하지만 단위에 대한 제약이 없어서 12bit, 16bit, 24bit등 원하는 길이로 설

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

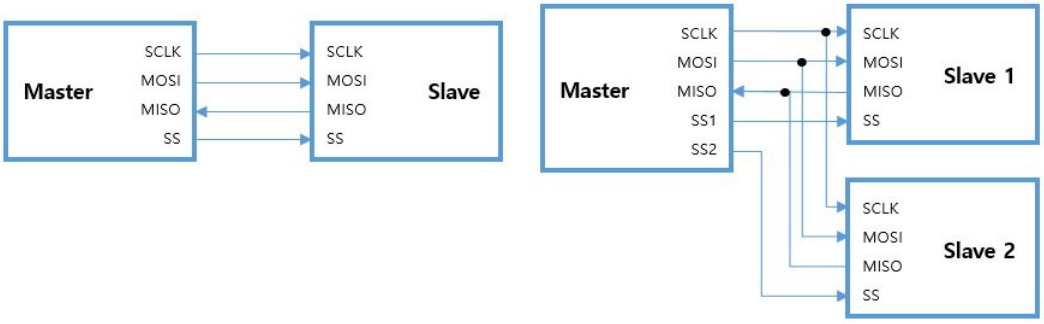


Figure. 26 SPI Interface Diagram

정하여 사용가능하다. 모든 데이터 전송이 끝나면 SS라인의 신호를 High로 변경하여 종료한다. SPI 통신 프로토콜은 Fig.27 와 같이 나타낼 수 있다. SPI는 마스터의 SCLK 클럭에 맞추어 마스터와 슬레이브가 데이터를 송수신하는 방법 이외에도 클럭의 극성(Polarity)와 위상(Phase)의 조합으로 총 4가지 방법으로 사용 가능하다. STM32에서 SPI를 사용하기 위해서는 SPI_CRx, SPI_SR, SPI_DR, SPI_RXCRCR, SPI_TXCRCR 레지스터 설정을 통해 사용할 수 있다.

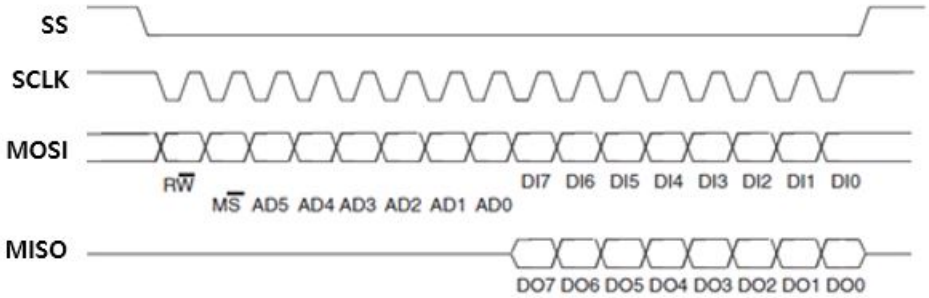


Figure. 27 SPI Protocol

5) PWM

PWM(Pulse Width Modulation)은 디지털 펄스를 주파수에 맞춰 주기적으로 출력한다. PWM은 출력 파형에서 주기(Period)와 듀티 비(Duty Cycle)를 조절 할 수 있으며 진폭(Amplitude)은 디지털 신호의 특성상 일반적으로 미리 설계된 값으로 정해진다. 듀티 비의 경우 전체

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

주기에서 신호가 있는 구간의 비율을 나타내며, PWM의 진폭은 STM32의 경우 TTL(Transistor Transistor Logic) 레벨(5V)로 출력된다. PWM 펄스 신호는 Fig.28 와 같이 나타낼 수 있다. PWM 제어는 듀티 비를 변화시켜 전체적인 평균값을 조절해서 LED의 밝기제어 DC모터의 속도제어, 서보모터의 각도제어에 사용된다.

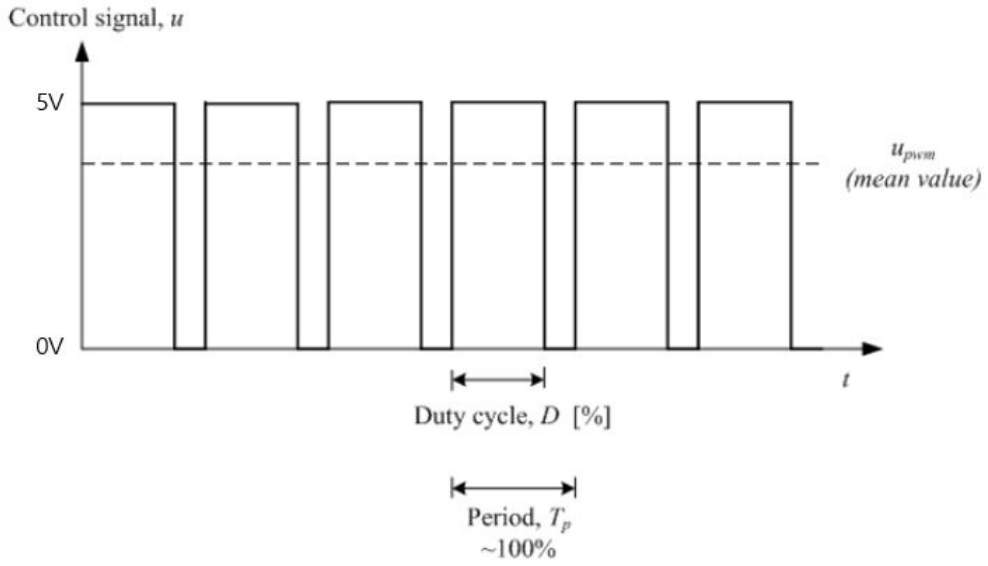


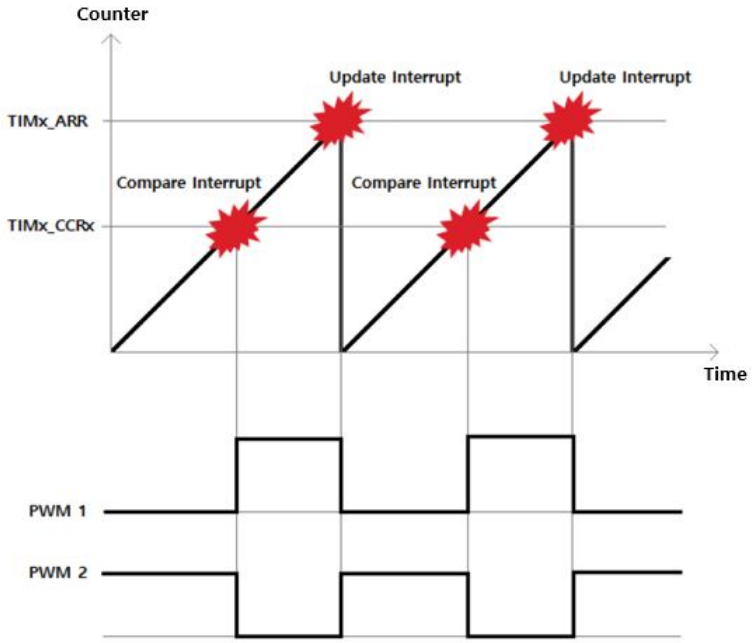
Figure. 28 PWM Waveform

PWM은 내부적으로 타이머 회로를 통해 작동된다. 일정한 주기로 입력되는 클럭 신호를 통해서 시간을 측정하고, 특정 시간마다 인터럽트를 발생시켜 출력 신호 비트를 1 또는 0으로 설정한다. 이때 인터럽트란, MCU 자체가 하드웨어적으로 그 변화를 체크하여 변화 시에만 일정한 동작하는 것을 말한다. STM32의 경우 Fast PWM와 Center aligned PWM 총 2가지의 PWM 모드가 제공된다. Figure.29 는 PWM 작동 방식 중 Fast PWM 모드를 나타낸 것이다. 클럭의 펄스시간에 따라 카운트 되는 레지스터의 값과 특정 레지스터와의 관계를 보여주고 있다. TIMx_CCRx에 따라 듀티 비가 조정되고, TIMx_ARR에 따라 PWM의 주파수 혹은 주기가 조정된다. Center aligned PWM모드는 카운터가 0-TIM_ARR까지 세었다가, 다시 0으로 되는 Fast PWM과 달리, 카운터가 0에서 TIM_ARR까지 도달하면 0으로 항하여 다운 카운트를 하는

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

방식이다. STM32에서 TIMER를 사용하기 위해서는 TIM_CRx, TIM_CCMRx, TIM_CCER, TIM_PSC, TIM_ARR, TIM_CCRx 레지스터 설정을 통해 사용할 수 있다.



$$duty\ cycle = \frac{TIM_CCRx}{TIM_ARR}$$

Figure. 29 PWM Fast Mdoe

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

III. 쿼드로터 모델링

1. 좌표계

쿼드로터의 동적 동작을 설명하기 위해서는 두 개의 기준 프레임 좌표계가 필요하며 좌표계로는 동체 좌표계(Body Frame, X_B, Y_B, Z_B)와 관성 좌표계(Inertial Frame, X_E, Y_E, Z_E)이다. 쿼드로터의 동체 좌표계는 실제 관찰자 또는 비행기의 중심에 고정되어 있는 좌표계이고, 관성 좌표계는 뉴턴의 제 1법칙이 성립하여 아무런 힘도 작용하지 않는 물체는 정지해 있거나 등속 직선 운동을 한다. 기본적으로 지구 중심을 영점으로 두는 좌표계이다. 쿼드로터 구조는 4개의 로터에 의해 추력, 각속도, 토크를 포함하여 Fig.30와 같이 나타낸다.[18–20]

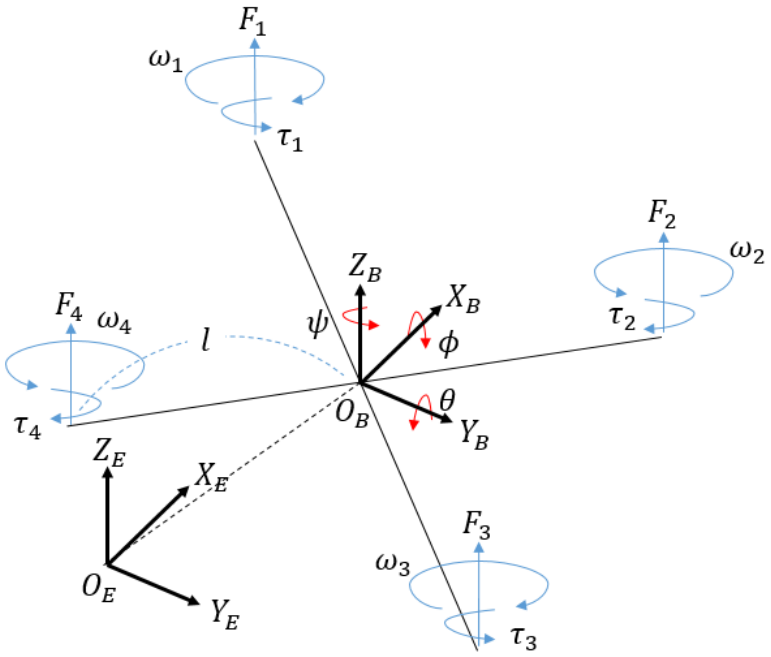


Figure. 30 Coordinate system of quad-rotor

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

여기에서 쿼드로터의 동적 동작은 쿼드로터의 12개 상태에서 나타나는 6 자유도로 설명할 수 있다. 쿼드로터의 위치는 관성프레임에서 X_E, Y_E, Z_E 축을 ξ로 정의된다. 자세 즉, 각도의 위치는 관성프레임에서 3개의 오일러 각도(Euler Angle) η로 정의한다. Pitch 각도 θ는 Y축을 중심으로 회전 하고, Roll 각도 ϕ는 X축을 중심으로 하고, Yaw 각도 ψ는 Z축을 중심으로 쿼드로터의 회전 운동을 한다. 동체프레임의 원점은 쿼드로터의 질량 중심에 있다. 동체프레임에서 선형속도는 V_B, 각속도는 ν로 정의한다[21].

$$\xi_E = \begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix}, \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{4}$$

$$V_B = \begin{bmatrix} v_{x,B} \\ v_{y,B} \\ v_{z,B} \end{bmatrix}, \quad \nu = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5}$$

2. 회전행렬

쿼드로터의 운동방정식을 유도하기 위해서는 하나의 좌표계에 관하여 기술해야 하므로 관성좌표계를 동체 좌표계로 변환하기 위한 좌표변환행렬 R_E^B을 통해 변환한다.[18] 본 논문에서의 좌표변환행렬 R_E^B은 3→2→1 변환 순서로 오일러각(Euler angle, $\psi \rightarrow \theta \rightarrow \phi$)을 사용하였으며, 항공우주분야에서 많이 사용되는 시퀀스 중 하나이다. Fig.31 와 같다.

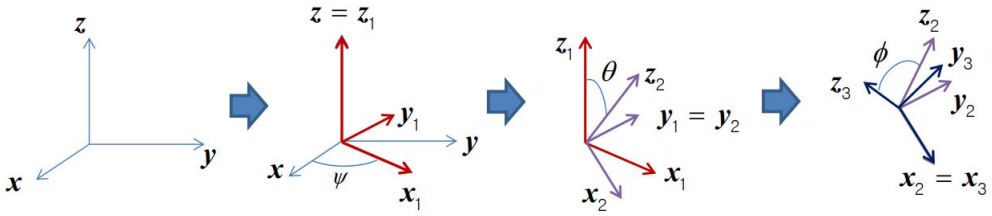


Figure. 31 $\psi \rightarrow \theta \rightarrow \phi$ Conversion Coordinate System

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

첫 번째 회전 z축에 대해 회전 ψ를 Yaw라 하고 Eq.6과 같이 나타낸다.

$$\begin{aligned} x_1 &= \cos\psi x + \sin\psi y \\ y_1 &= -\sin\psi x + \cos\psi y \\ z_1 &= z \end{aligned} \tag{6}$$

행렬의 행태로 변환하면,

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{7}$$

따라서,

$$R_E^{\psi} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

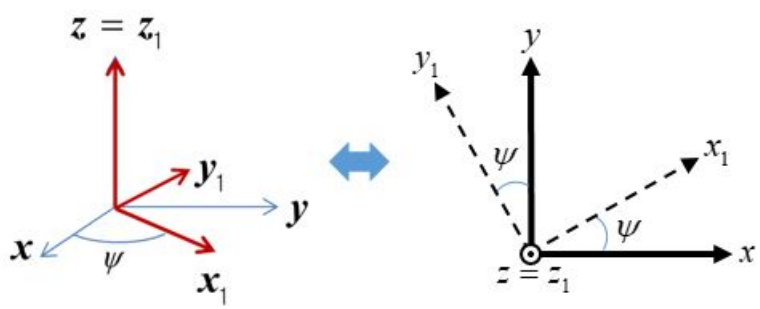


Figure. 32 Definition of Yaw-axis Rotation

두 번째 회전 y축에 대해 회전 θ를 Pitch라 하고 Eq.9과 같이 나타낸다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

$$
\begin{aligned}
x_2 &= \cos\theta x_1 - \sin\theta z_1 \\
y_2 &= y_1 \\
z_2 &= \sin\theta x_1 + \cos\theta z_1
\end{aligned}
\tag{9}
$$

행렬의 형태로 변환하면,

$$
\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}
\tag{10}
$$

따라서,

$$
R_{\psi}^{\theta} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}
\tag{11}
$$

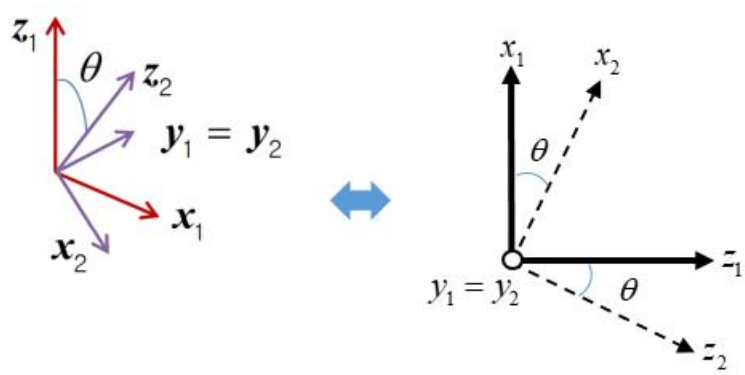


Figure. 33 Definition of Pitch-axis Rotation

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

세 번째 회전 x축에 대해 회전 ϕ를 Roll라 하고 Eq.12과 같이 나타낸다.

$$\begin{aligned} x_3 &= x_2 \\ y_3 &= \cos\phi y_2 + \sin\phi z_2 \\ z_3 &= -\sin\phi y_2 + \cos\phi z_2 \end{aligned} \tag{12}$$

행렬의 형태로 변환하면,

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \tag{13}$$

따라서,

$$R_\theta^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{14}$$

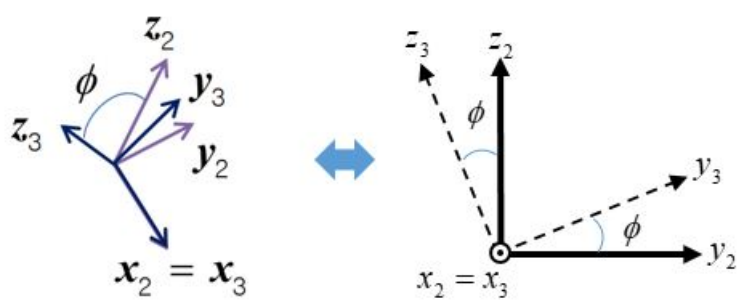


Figure. 34 Definition of Roll-axis Rotation

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

따라서 관성 좌표계 E에서 동체 좌표계 B의 변환을 다음과 같이 모든 행렬의 곱으로 나타낸다.

$$
\begin{aligned}
\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} &= R_\theta^B \cdot R_\psi^\theta \cdot R_E^\psi \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
&= \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\theta c\psi s\phi - s\psi c\phi & s\theta c\psi s\phi + c\psi c\phi & c\theta s\phi \\ s\theta c\psi c\phi + s\psi s\phi & s\theta c\psi c\phi - c\psi s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}
\end{aligned}
\tag{15}
$$

따라서, 좌표변환행렬 R_E^B은 다음과 같다.

$$
R_E^B = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\theta c\psi s\phi - s\psi c\phi & s\theta c\psi s\phi + c\psi c\phi & c\theta s\phi \\ s\theta c\psi c\phi + s\psi s\phi & s\theta c\psi c\phi - c\psi s\phi & \mathrm{c}\theta c\phi \end{bmatrix} \tag{16}
$$

여기서 s는 sin, c는 cos를 의미한다. R_E^B은 표준직교행렬(Orthonormal matrix)이므로 $R^{-1} = R^T$이다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

3. 운동학

상태변수 X_E, Y_E, Z_E는 관성 좌표계이고 속도 $v_{x,B}, v_{y,B}, v_{z,B}$는 동체 좌표계이다. 이러한 위치와 속도 사이의 관계의 관계식은 아래의 식과 같다.[22]

$$\begin{aligned}
\frac{d}{dt}\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} &= R_B^E \begin{bmatrix} v_{x,B} \\ v_{y,B} \\ v_{z,B} \end{bmatrix} \\
&= \left(R_E^B\right)^T \begin{bmatrix} v_{x,B} \\ v_{y,B} \\ v_{z,B} \end{bmatrix} \\
&= \begin{bmatrix} c\theta c\psi & s\theta c\psi s\phi - s\psi c\phi & s\theta c\psi c\phi + s\psi s\phi \\ c\theta s\psi & s\theta c\psi s\phi + c\psi c\phi & s\theta c\psi c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} v_{x,B} \\ v_{y,B} \\ v_{z,B} \end{bmatrix}
\end{aligned} \tag{17}$$

관성좌표계에서 오일러 각 ϕ, θ, ψ의 시간변화율과 동체좌표계의 각속도 p, q, r 관계에 대해 나타낼 수 있다. 따라서 관계식은 다음과 같다.[23]

$$\begin{aligned}
\begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_\theta^B \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_\theta^B(\phi) R_\psi^\theta(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\
&= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & \mathrm{c}\phi & \mathrm{s}\phi c\theta \\ 0 & -\mathrm{s}\phi & \mathrm{c}\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}
\end{aligned} \tag{18}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

위 식을 다시 정리하면,

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = W_E^B \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \qquad \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = (W_E^B)^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \qquad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{20}$$

4. 쿼드로터의 비행동역학

뉴턴-오일러 방정식을 사용하여 동역학을 설명할 수 있다. 동체좌표계에서 질량 가속에 필요한 힘 $m\dot{V}_B$와 원심력 $\nu \times (mV_B)$에 필요한 힘은 중력 $R^T G$와 로터의 총 추력 T_B와 동일하다.[24]

$$m\dot{V}_B + \nu \times (mV_B) \;=\; R^T G + T_B \tag{21}$$

관성좌표계에서 원심력은은 무시할정도로 작기 때문에 생략한다. 따라서, 추력의 중력의 힘, 크기, 방향만이 쿼드로터의 가속에 기여한다.

$$m\dot{V}_B = \; G + RT_B \tag{22}$$

다시 쓰면,

$$\begin{bmatrix} \dot{v}_{x,B} \\ \dot{v}_{y,B} \\ \dot{v}_{z,B} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} c\phi s\theta c\psi + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} \tag{23}$$

앞의 모델 Eq.23은 복잡한 동적 상호 작용의 단순화된 모델이다. 쿼드로터의 실제 동작을 강화하기 위해 공기 저항에 의해 생성 된 항력이 포함되어야 한다. 따라서 다음과 같이 나타낼수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

$$\begin{bmatrix} \dot{v}_{x,B} \\ \dot{v}_{y,B} \\ \dot{v}_{z,B} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} c\phi s\theta c\psi + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} - \frac{1}{m} \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} \tag{24}$$

여기서 A_x, A_y, A_z는 관성좌표계의 해당 방향에서의 속도에 대한 항력계수이다.

오일러 방정식으로 부터 회전운동(Rotational motion)에 적용하면 다음과 같습니다.

$$I\dot{\omega} + \omega \times (I\omega) \ = \ \tau \tag{25}$$

여기서 ω는 각속도 벡터, I는 관성모멘트, τ는 외부 토크의 벡터이다. 따라서 다음과 같이 다시 쓸 수 있다.

$$\dot{\nu} \ = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \ = \ I^{-1} \left(\tau - \nu \times (I\nu) \right) \tag{26}$$

쿼드로터의 관성 모멘트는 Eq.27 와 같이 표현되며 쿼드로터의 기체는 대칭이므로 $I_x x = I_y y$ 이다.

$$I \ = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{27}$$

따라서, 동체좌표계의 운동 방정식은 다음과 같이 얻어진다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

$$\dot{\nu} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 1/I_{xx} & 0 & 0 \\ 0 & 1/I_{yy} & 0 \\ 0 & 0 & 1/I_{zz} \end{bmatrix} \left(\begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \right) \tag{28}$$

정리하면,

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \tau_\phi {I_{xx}}^{-1} \\ \tau_\theta {I_{yy}}^{-1} \\ \tau_\psi {I_{zz}}^{-1} \end{bmatrix} - \begin{bmatrix} \frac{I_{yy}-I_{zz}}{I_{xx}} qr \\ \frac{I_{zz}-I_{xx}}{I_{yy}} pr \\ \frac{I_{xx}-I_{yy}}{I_{zz}} pq \end{bmatrix} \tag{29}$$

5. 힘과 모멘트

이 장의 목적은 쿼드로터에 작용하는 힘과 토크를 설명하는 것이다. 공기역학적인 양력이 없기 때문에 공기역학적 힘과 모멘트는 무시할 수 있다고 가정한다. 힘과 모멘트는 주로 중력과 네 개의 프로펠러에 영향이 있다.[25] Figure.30에 나타낸 것처럼, 각 모터는 힘 F과 토크 τ를 생성한다. 쿼드로터의 총 힘 F는

$$F = F_1 + F_2 + F_3 + F_4 \tag{30}$$

Roll 운동의 토크는 1,4번과 2,3번의 모터의 힘에 의해 생성된다.

$$\tau_\phi = l\,(F_1 + F_4 - F_2 - F_3) \tag{31}$$

Pitch 운동의 토크는 1,2번과 3,4번의 모터의 힘에 의해 생성된다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

$$\tau_\theta \ = \ l\,(F_1 + F_2 - F_3 - F_4) \tag{32}$$

뉴턴의 3 번째 법칙으로 인해 프로펠러가 회전하면 쿼드로터의 기체에 Yaw 운동의 토크가 생성된다. 토크의 방향은 프로펠러 운동의 반대 방향이다. 따라서 Yaw 운동 토크는

$$\tau_\psi \ = F_1 + F_3 - F_2 - F_4 \tag{33}$$

프로펠러에 의해 생성된 양력(Lift)와 항력(Drag)는 각속도의 제곱에 비례한다. 각속도는 모터로 전송된 펄스 폭 변조에 직접 비례한다고 가정한다. 따라서, 각 모터의 힘과 토크는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} F_i \ &= \ k\omega_i^2 \\ \tau_i &= b\omega_i^2 \ + \ I_M\dot{\omega}_i \end{aligned} \tag{34}$$

여기서 양력 상수는 k, 항력 상수는 b, 로터의 관성모멘트는 I_M, i는 1,2,3,4를 의미한다. $\dot{\omega}_i$의 효과가 작다고 간주되므로 생략한다.

로터의 결합된 합은 동체 Z축의 방향으로 추력 T를 생성한다. 토크 τ_B는 동체좌표계에서 $\tau_\phi, \tau_\theta, \tau_\psi$로 구성된다.

$$T \ = \ \sum_{i=1}^{4} F_i \ = \ k\sum_{i=1}^{4}\omega_i^2, \quad T_B \ = \ \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{35}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

$$\tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} lk\left(\omega_1^2 + \omega_4^2 - \omega_2^2 + \omega_3^2\right) \\ lk\left(\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right) \\ b\left(\omega_1^2 + \omega_3^2 - \omega_2^2 + \omega_4^2\right) \end{bmatrix} \tag{36}$$

여기서 l은 쿼드로터의 잘량중심과 로터중심의 거리이다.

따라서 쿼드로터의 힘과 토크를 Eq.37 과 같이 행렬 형태로 나타낼 수 있다.

$$\begin{bmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} k & k & k & k \\ lk & -lk & -lk & lk \\ lk & lk & -lk & -lk \\ b & -b & b & -b \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \triangleq \mathrm{M} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{37}$$

실제 모터 명령은 다음과 같다.

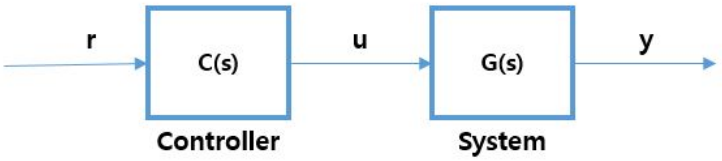
$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \mathrm{M}^{-1} \begin{bmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{38}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

IV. 쿼드로터 제어기

제어방식의 종류에는 크게 개루프(Open loop)제어 시스템과 폐루프(Closed loop)제어 시스템이 있다. 개루프 제어는 Fig.35 와 같다. 개루프 제어는 원하는 명령을 받아들여 제어기에서 처리하여 시스템에 적절한 입력(u)을 생성한다. 시스템에서는 제어기에서 받은 입력에 의해 출력(y)를 생성한다. 대표적인 개루프 제어 시스템으로 자판기가 있다.



r : reference, desired output
u : system input
y : system output

Figure. 35 Open loop control

개루프 제어 시스템의 전달함수는 다음과 같다.

$$\frac{Y(s)}{R(s)} = C(s)\,G(s) \tag{39}$$

폐루프 제어는 Fig.36 와 같다. 폐루프 제어는 개루프 제어와 다르게 출력(y)값을 센서로 측정하여 제어기에 반영한다. 다시 말해, 원하는 출력과 실제 출력 값의 차이가 적어지는 방향으로 제어기가 제어하는 원리이다. 대표적인 개루프 시스템으로 냉방기의 자동제어시스템이 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

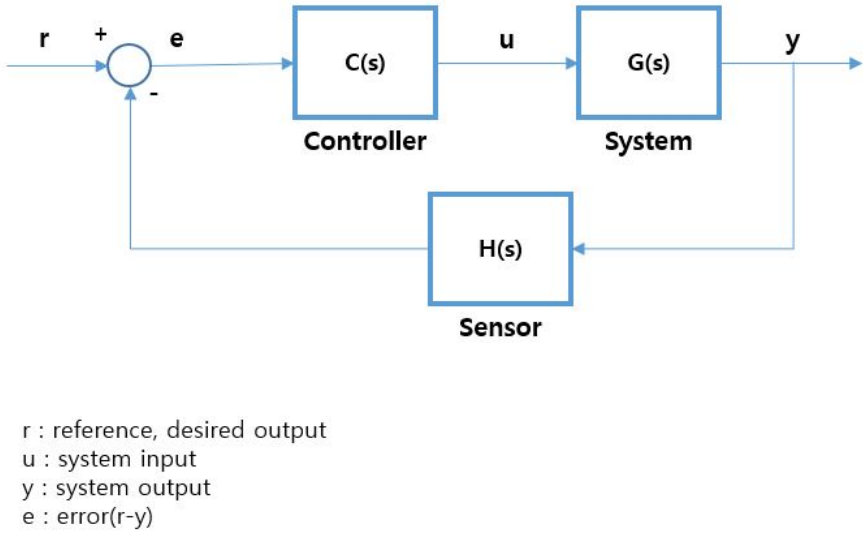


Figure. 36 Closed loop control

폐루프 제어 시스템의 전달함수는 다음과 같다.

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)} \tag{40}$$

1. PID 제어기

폐루프 제어 시스템의 제어기에서 가장 대표적인 제어기가 PID(Proportional Integral Derivative) 제어기이다. 각각 P는 비례, I는 적분, D는 미분을 의미한다. Eq.41는 가장 기본적인 PID 제어식이다.[26]

$$\begin{aligned} e(t) &= x_d(t) - x(t) \\ u(t) &= K_P e(t) + K_I \int_0^t e(\tau)\,d\tau + K_D \frac{de(t)}{dt} \end{aligned} \tag{41}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

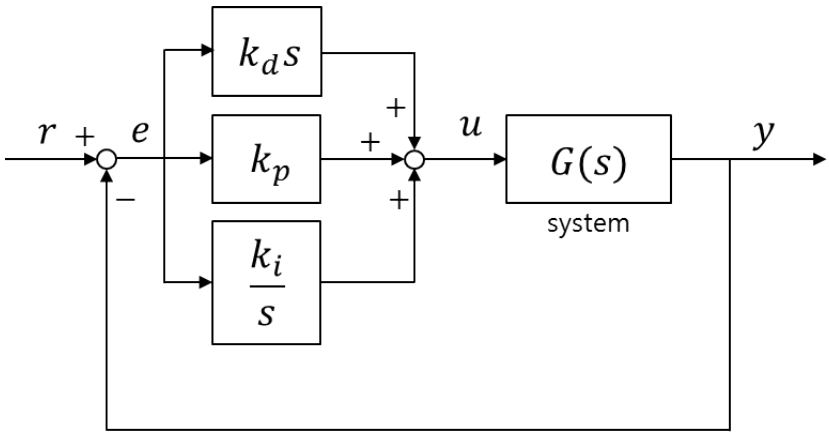


Figure. 37 Block diagram of PID control

Table 10: PID control parameter characteristics

	Rise time	**Overshoot**	**Settling time**	**Steady-state error**
K_P	↓	↑	Small change	↓
K_I	↓	↑	↑	Great reduce
K_D	Small change	↓	↓	Small change

1) 비례 제어기

비례(Proportional, P)제어기는 목표치에 가는 정도를 결정하는 상수이다. 비례 제어기는 다음과 같이 $error(x_d - x)$에 비례한 제어 입력을 생성한다.

$$u\,(t) = K_P e\,(t) \tag{42}$$

K_P는 비례 게인(Gain이라 하며 설계자가 설정하는 것이다. K_P가 클수록 같은 error에 대한 시스템의 입력(u)가 커지므로 출력값이 더 빠르게 원하는 값에 근접하게 수렴하고 응답속도가

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

빠르지만 오버슈트가 발생하여 오실레이션이 발생하게 된다. K_P가 너무 작으면 응답속도가 느리고 정상상태에 도달하지 못하게 된다. 수학적인 관점으로 보면 K_P가 커지면 Eq.39의 Pole의 복소수가 커진다는 의미이다.

2) 적분 제어기

적분(Integral, I)제어기는 목표치에 대한 정확도를 결정하는 상수이다. 비례제어기의 K_P가 아무리 크더라도 원하는 값에 끝까지 도달하지 못하고 아주 조금의 오차가 남게 된다. 원인은 Eq.42에서 찾을수 있으다. 목표값에 도달할수록 오차가 줄어들어 e의 값이 작아진다. 그리고 K_P는 고정값이기 때문에 목표값에 도달할수록 제어기 출력이 약해진다. 이를 해결하기 위해서 사용한 제어기가 바로 적분 제어기이다.

적분 제어기는 error를 적분하여 제어 입력 u를 생성한다. 즉, 0초부터 시작되서 나타난 error들을 모두 반영하여 시스템의 제어 입력값을 결정한다는 의이이다. 앞서 말했듯이, 적분 제어기의 장점은 정상 상태 오차(Steady state error)가 발생할 때 정상 상태 오차 구간을 적분하여 원하는 값을 끝까지 잘 추종할 수 있게 한다.

$$u\left(t\right) = K_I \int_0^t e\left(\tau\right) d\tau \tag{43}$$

3) 미분 제어기

미분(Derivative, D)제어기는 목표치에 대한 속도를 결정하는 상수이다. 응답속도를 빠르게 하기 위해서 K_P값을 높이다 보면 오버슈트(Overshoot)가 발생한다. 이때 K_D게인 값을 적절하게 반영해주면 오버슈트를 제거해주는 역할을 한다. 비례 제어기만 제어했을 때 error값은 음수값이다. 특히 error가 급격하게 변화할수록 K_D게인의 영향이 커진다. 따라서 미분 제어는 급격한 제어기의 출력을 완화하는 작용을 한다. 수학적인 관점으로 보면, K_D가 Fig.39에서 pole 복소근을 조절하는 역할을 한다.

$$u\left(t\right) = K_D \frac{de\left(t\right)}{dt} \tag{44}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

2. 제어기 설계

Eq.41와 같이 제어기를 설계한다. 위치 오류 ξ_e를 정의하면 다음과 같다.

$$\xi_e = \xi_c - \xi \tag{45}$$

여기서 벡터 $\xi_c = (X_c, Y_c, Z_c)^T$는 위치 명령 벡터이다. PID 제어식을 다음과 같이 구성한다.

$$U = K_P \xi_e + K_I \int \xi_e d\tau + K_D \dot{\xi}_e \tag{46}$$

가상 제어 벡터 $U = (U1, U2, U3)^T$로 정의한다[27]. Eq.22에서 $\dot{V}$에 U를 대체하여 쓰면,

$$U = -gz_e + \frac{1}{m} T_B R z_e \tag{47}$$

중력 가속 항목 $-gz_e$를 왼쪽 항으로 옮기고 양변에 R^T를 곱한다.

$$R^T (U + gz_e) = \frac{1}{m} T_B z_e \tag{48}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

다시 쓰면,

$$\begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\theta c\psi s\phi - s\psi c\phi & s\theta c\psi s\phi + c\psi c\phi & c\theta s\phi \\ s\theta c\psi c\phi + s\psi s\phi & s\theta c\psi c\phi - c\psi s\phi & \mathrm{c}\theta c\phi \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 + g \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{49}$$

다시 정리하면,

$$U_1 \mathrm{c}\theta \mathrm{c}\psi + U_2 \mathrm{c}\theta \mathrm{s}\psi - (U_3 + g)\,\mathrm{s}\theta = 0 \tag{50}$$

$$U_1 \left(\mathrm{s}\theta \mathrm{c}\psi \mathrm{s}\phi - \mathrm{s}\psi c\phi\right) + U_2 \left(\mathrm{s}\theta s\psi \mathrm{s}\phi + c\psi c\phi\right) + (U_3 + g)\, c\theta s\phi = 0 \tag{51}$$

$$U_1 \left(\mathrm{s}\theta \mathrm{c}\psi c\phi + \mathrm{s}\psi s\phi\right) + U_2 \left(\mathrm{s}\theta s\psi c\phi - c\psi s\phi\right) + (U_3 + g)\, c\theta c\phi = \frac{1}{m} T \tag{52}$$

위의 식을 풀면 다음과 같은 식이 구해진다.

$$\phi_c = \arcsin\left(\frac{U_1 \sin\psi_c - U_2 \cos\psi_c}{\sqrt{U_1^2 + U_2^2 + (U_3 - g)^2}}\right) \tag{53}$$

$$\theta_c = \arctan\left(\frac{U_1 \cos\psi_c + U_2 \sin\psi_c}{U_3 - g}\right) \tag{54}$$

$$T \;=\; m\left(U_1 \left(s\theta c\psi c\phi + s\psi s\phi\right) + U_2 \left(\mathrm{s}\theta s\psi c\phi - c\psi s\phi\right) + (U_3 + g)\, c\theta c\phi\right) \tag{55}$$

위의 식에서 ψ_c는 쿼드로터의 방향각(heading angle) 명령으로 다음과 같이 생성할 수 있으며 Fig.38 와 같다.[28]

$$\psi_c = \arctan\left(\frac{Y_d - Y}{X_d - X}\right) \tag{56}$$

제어기에 대해서 정리하면,

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

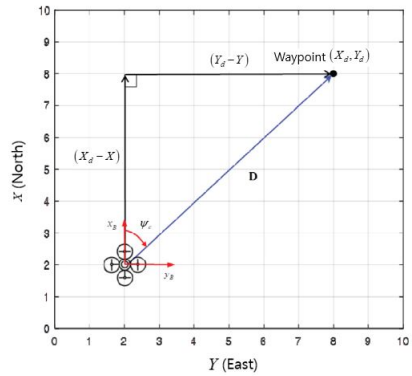


Figure. 38 Heading angle of quarrotor

$$
\begin{aligned}
U_1 &= K_{X,P}\left(X_d - X\right) + K_{X,I}\int\left(X_d - X\right) + K_{X,D}\left(\dot{X}_d - \dot{X}\right) \\
U_2 &= K_{Y,P}\left(Y_d - Y\right) + K_{Y,I}\int\left(Y_d - Y\right) + K_{Y,D}\left(\dot{Y}_d - \dot{Y}\right) \\
U_3 &= K_{Z,P}\left(Z_d - Z\right) + K_{Z,I}\int\left(Z_d - Z\right) + K_{Z,D}\left(\dot{Z}_d - \dot{Z}\right)
\end{aligned} \tag{57}
$$

$$
\begin{aligned}
\tau_\phi &= \left(K_{\phi,P}\left(\phi_c - \phi\right) + K_{\phi,I}\int\left(\phi_c - \phi\right) + K_{\phi,D}\left(\dot{\phi}_c - \dot{\phi}\right)\right) I_{xx} \\
\tau_\theta &= \left(K_{\theta,P}\left(\theta_c - \theta\right) + K_{\theta,I}\int\left(\theta_c - \theta\right) + K_{\theta,D}\left(\dot{\theta}_c - \dot{\theta}\right)\right) I_{yy} \\
\tau_\psi &= \left(K_{\psi,P}\left(\psi_d - \psi\right) + K_{\psi,I}\int\left(\psi_c - \psi\right) + K_{\psi,D}\left(\dot{\psi}_d - \dot{\psi}\right)\right) I_{zz}
\end{aligned} \tag{58}
$$

$$
T = m\left(U_1\left(s\theta c\psi c\phi + s\psi s\phi\right) + U_2\left(s\theta s\psi c\phi - c\psi s\phi\right) + \left(U_3 + g\right) c\theta c\phi\right) \tag{59}
$$

Figure.39는 설계한 제어기의 블록선도이다.

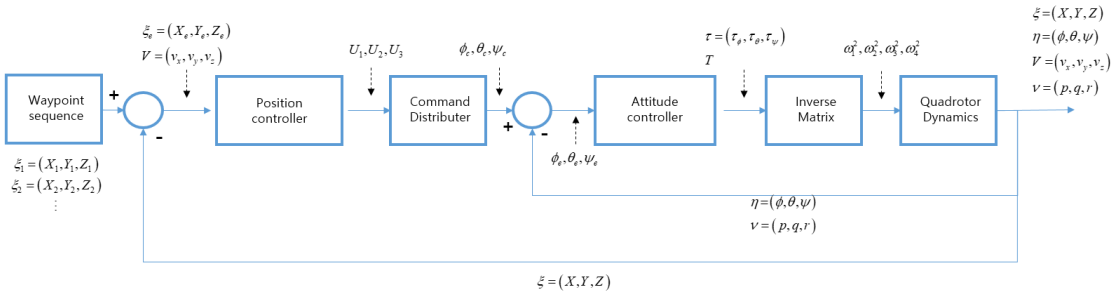


Figure. 39 Control block diagram

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

V. 장애물 탐지 및 회피

본 논문에서는 드론과 장애물에 대한 정보기반으로 장애물과의 충돌 예상 시간 및 상대속도를 이용한 회피 알고리즘을 개발한다. 회피 알고리즘에서 드론은 장애물과의 충돌까지 걸리는 시간 및 장애물과의 거리는 기준으로 탐지모드(Detection mode)와 추적모드(Tracking mode), 회피모드(Avoid mode)로 나누어 비행한다.

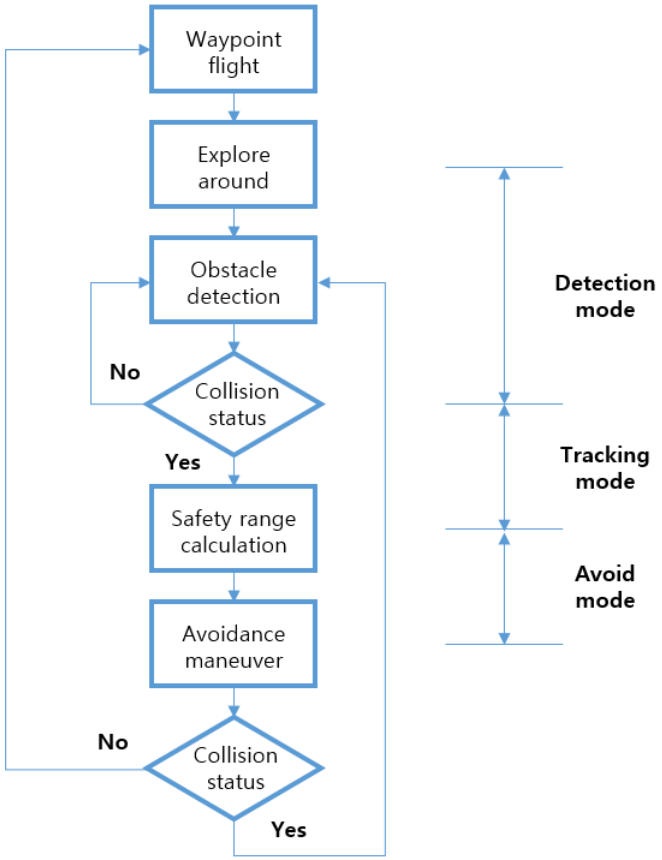


Figure. 40 Collision algorithm

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

탐지모드에서는 드론이 경로지점을 향해 항법시스템을 이용하는 비행하는 항행모드로 LIDAR를 이용하여 비행경로에 있는 장애물을 탐지한다. 탐지모드에서 특정 물체를 발견하였을 경우 추적모드로 전환한다.

추적모드에서는 특정 물체의 위치정보를 획득하여 드론과의 충돌위험을 인지한다. LIDAR를 이용하여 드론과 특정 물체와의 거리 및 드론의 속도 및 방향을 고려하여 예상 충돌시간을 계산한다. 특정 물체와의 충돌 예상시간 및 상대거리를 기반으로 장애물로 판단될 경우 회피모드 전환여부를 결정한다. 회피모드에서 수식 계산 및 시뮬레이션을 통해 회피 확률 및 효율적 회피가 가능하도록 전환 시기를 조정한다.

추적모드에서 계산된 장애물과의 충돌위험에 따라 회피모드로 전환된다. 회피모드로 전환되었을 때에는 장애물과의 안전거리와 추적모드에서 계산된 장애물의 위치 기반으로 장애물과 충돌 회피를 위한 안전반경을 계산한다. 계산된 안전반경으로 드론의 새로운 경로지점을 생성하여 기존의 경로지점을 수정하여 새로운 경로로의 비행을 통해 장애물 회피기동을 수행한다. 회피기동을 하는 도중에도 꾸준히 충돌위험을 계산하여 충돌위험이 계속될 경우 경로 지점 생성 및 수정을 통해 회피기동을 하고, 충돌위험이 사라지면 다시 탐지모드로 전환하여 비행을 계속한다.

첫 번째 회피 경로는 우선 비행경로의 수직인 방향으로 장애물과 안전거리만큼 떨어진다. 그 후 비행경로와 같은 방향으로 장애물과 충돌위험이 적어질 때까지 비행하고, 충돌위험이 없어지면 기존의 비행경로로 돌아와 기존의 임무를 수행한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

VI. 시뮬레이션

1. 위치 및 자세제어 시뮬레이션 조건

쿼드로터의 시뮬레이션을 위한 쿼드로터 초기 조건을 Table.11과 같이 설정했다. 그리고 본 논문에서 설계한 쿼드로터 위치 ξ 파라미터는 K_P 0.5, K_I 0.1, K_D 0.8이며 자세 ϕ, θ는 동일하게 K_P 40, K_I 1, K_D 10이고 ψ는 K_P 30, K_I 20, K_D 20으로 Table.13과 같으며, 궤적추정을 위한 위치 명령은 Table.12와 같이 설정하였다.

Table 11: Simulation conditions of quad-rotor system

Properties	**Value**	**Properties**	**Value**
Mass (m)	1.6 Kg	Moment of x-axis ($I_x x$)	$4.856{\cdot}10^{-3}\ kg \cdot m^2$
Length (l)	0.52 m	Moment of y-axis ($I_y y$)	$4.856{\cdot}10^{-3}\ kg \cdot m^2$
Gravity acc. (g)	9.81 m/s^2	Moment of z-axis ($I_z z$)	$8.801{\cdot}10^{-3}\ kg \cdot m^2$
Drag force (b)	$1.140 \cdot 10^{-7}$	Drag force for vel. (A_x)	0.25
Lift force (k)	$2.980 \cdot 10^{-6}$	Drag force for vel. (A_y)	0.25
Moment of rotor (I_M)	$3.357{\cdot}10^{-7} kg \cdot m^2$	Drag force for vel. (A_z)	0.25
Initial Position (ξ_E)	$[0, 0, 0]\ \ m$	Initial Velocity (V_B)	$[0, 0, 0]\ \ m/s$
Initial Angle (η_E)	$[0, 0, 0]\ \ rad$	Initial Angular Velocity (ν_B)	$[0, 0, 0]\ \ rad/s$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Table 12: Guidance commands for trajectory tracking

	Position X_d	**Position Y_d**	**Position Z_d**
$\xi_d(1)$	0 m	0 m	1.5 m
$\xi_d(2)$	1 m	2 m	1 m
$\xi_d(3)$	2 m	1 m	0 m

Table 13: Parameters of attitude and position controller

Attitude controller			
Gain	K_P	K_I	K_D
Roll	40	1	10
Pitch	40	1	10
Yaw	30	20	20
Position controller			
Gain	K_P	K_I	K_D
Value	0.5	0.1	0.8

2. 위치 및 자세 시뮬레이션 결과

PID 제어기를 설계하여 쿼드로터의 위치 및 자세 제어 시뮬레이션을 수행하였다. 제어 시뮬레이션에서 사용된 조건은 Table.11,12,13와 같이 정리되어있다. PID 제어기를 통한 위치 및 자세제어 시뮬레이션의 최종 결과는 Fig.41과 같이 controla.re.kr에서 무료로 제공하는 Dancing plot을 이용하여 나타내었다. 주어진 유도 명령에 대하여 안정적으로 잘 추종하는 것을 볼 수 있었다.

주어진 시뮬레이션 결과를 자세하게 분석하기 위해서 각 축의 대한 위치, 속도, 각도, 각속도 그리고 각 로터에 대한 RPM, 제어입력에 대하여 Fig.42-47에 도시하였다.

Figure.42을 살펴보면 PID 제어기를 통한 각축에 대한 위치제어 성능을 보여주고 있다. 위치 명령 당 15초가 걸리며 각 축의 위치 명령이 변경되는 시점에서 1-2 m의 오차를 보이지만 5초 후 최대 0.015 m의 오버슈트를 보이며 그리고 5초 정도 후 명령값에 수렴하여 오차값이 1×10^{-4} m

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

정도인거를 확인하였다. Figure.43을 살펴보면 PID 제어기를 통한 각축에 대한 자세제어 성능을 보여주고 있다. 각 축의 자세 명령이 변경되는 시점에서 Roll의 경우 2.5 degree를 보이지만 2초 후 최대 0.5 degree의 오버슈트를 보이고 8초 정도 후 명령값에 수렴하여 오차값이 1×10^{-4} degree 정도이며, Pitch는 자세 명령이 변경되는 시점에서 오차값이 8 degree이고 2초 후 최대 1.6 degree의 오버슈트를 보이고 8초 정도 후 명령값에 수렴하여 오차값이 1×10^{-4} degree 그리고 Yaw는 자세 명령값이 변경되는 시점에서 오차값이 최대 90 degree이고 4초 정도 후 명령값에 수렴하여 오차값이 1×10^{-4} degree 정도인거를 확인하였다. Figure.44-45는 위치 및 자세제어의 시간에 따른 속도와 각속도 그래프이다. 마지막으로 Fig.46-47은 각 로터의 RPM과 제어입력값이며 명령값이 변화는 시점에서 기동을 하면서 15초와 30초때 값이 변화하는걸 확인할 수있다.

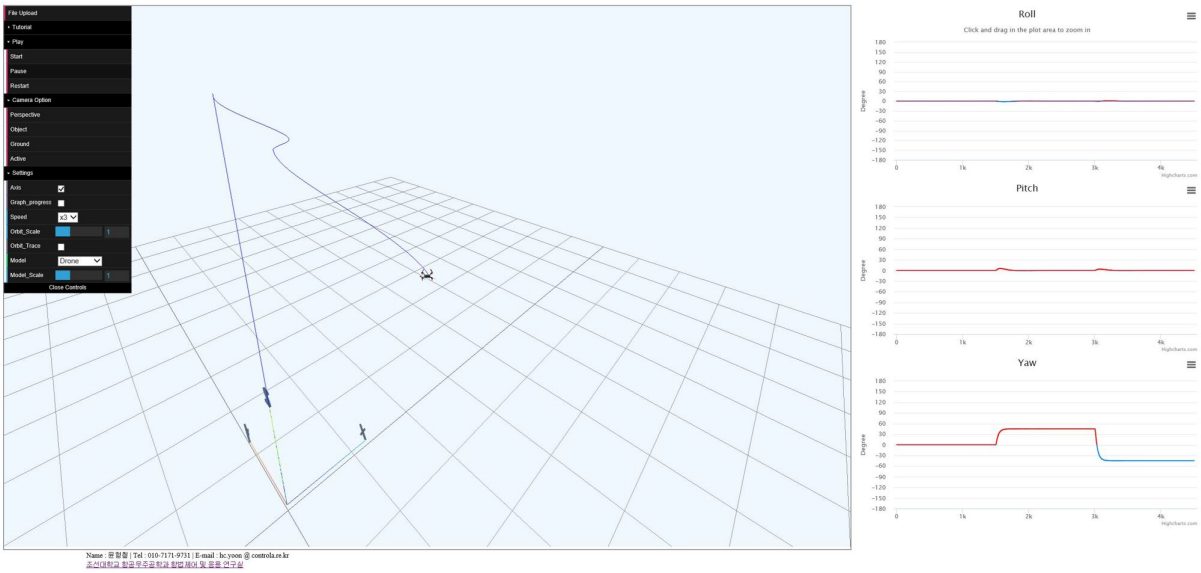


Figure. 41 Simulation using dancing plot

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

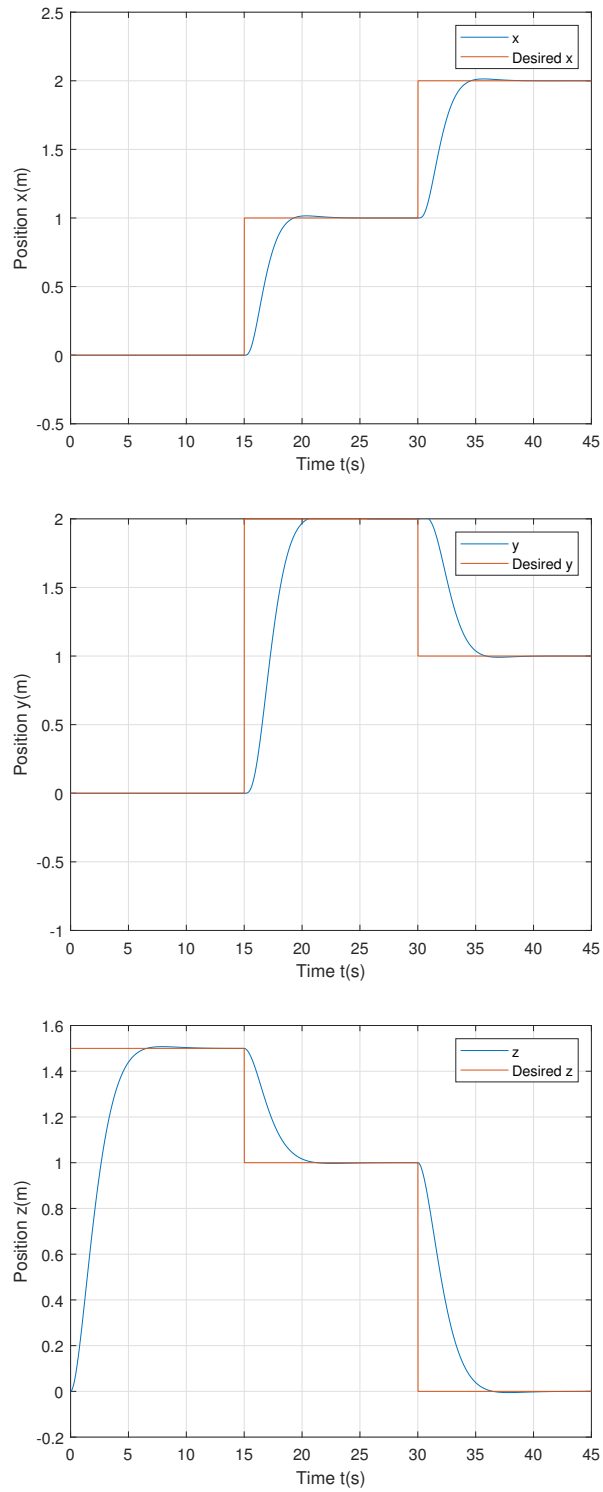


Figure. 42 Quad-rotor position history for a given position commands

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

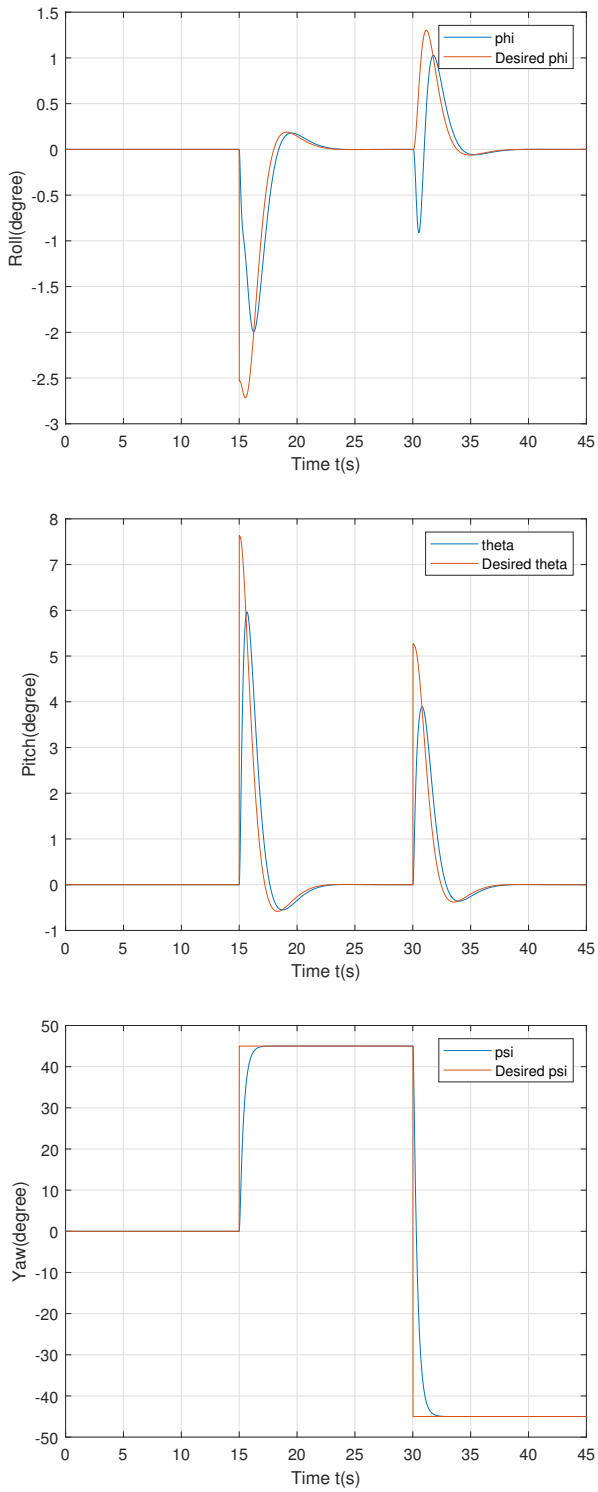


Figure. 43 Quad-rotor attitude histories for a given position commands

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

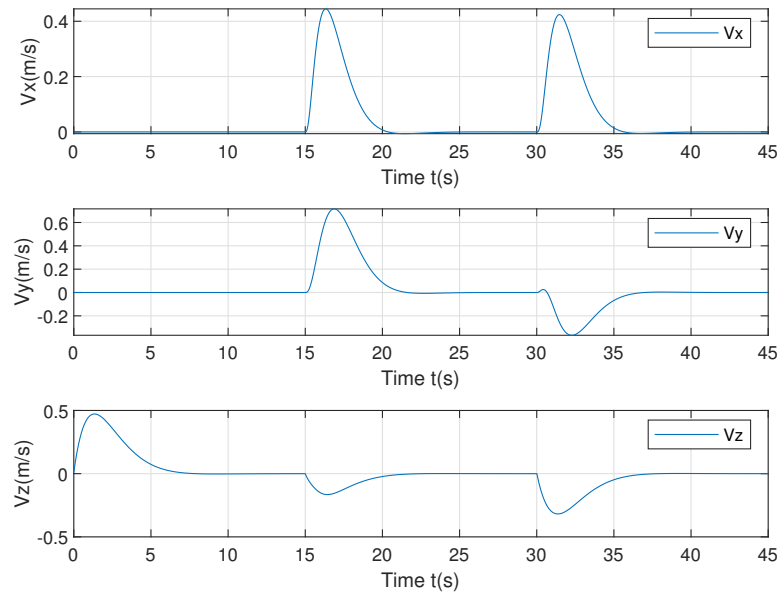


Figure. 44 Quad-rotor velocity responses for a given position commands

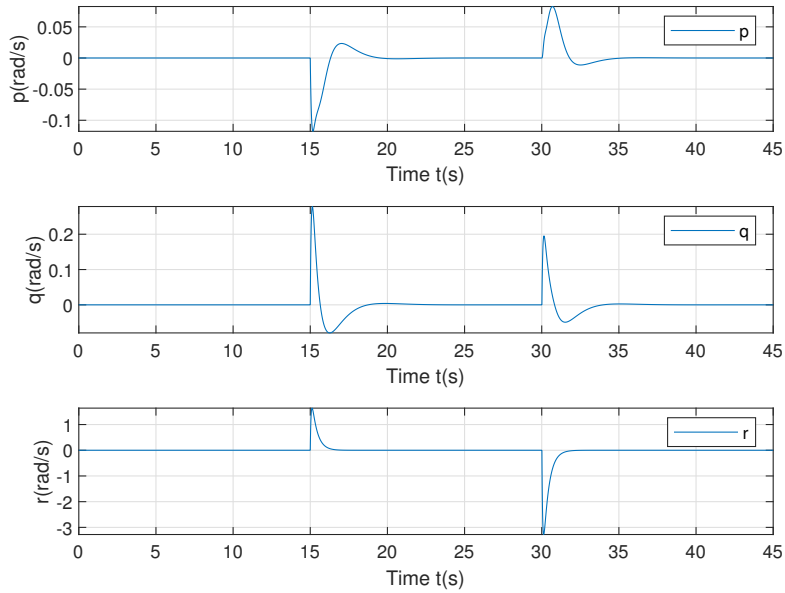


Figure. 45 Quad-rotor angular velocity history for a given position commands

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

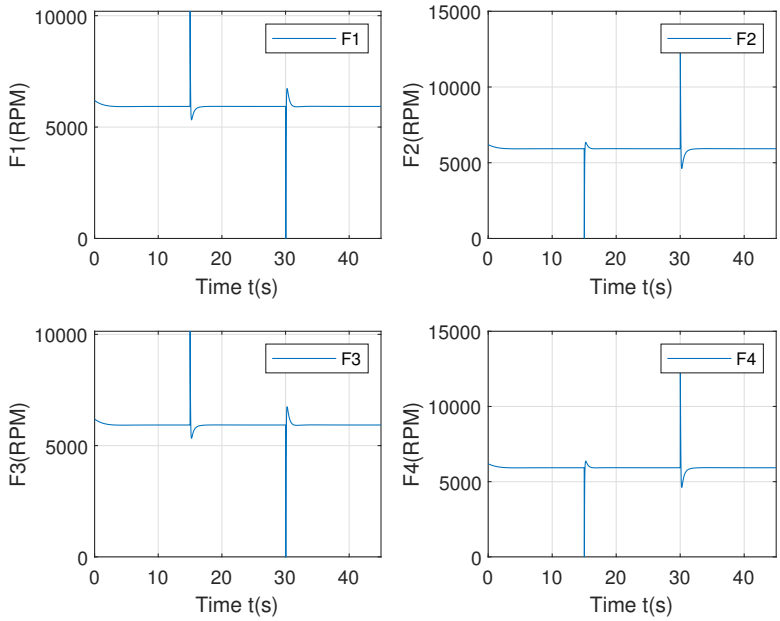


Figure. 46 Motor speed response for this simulation

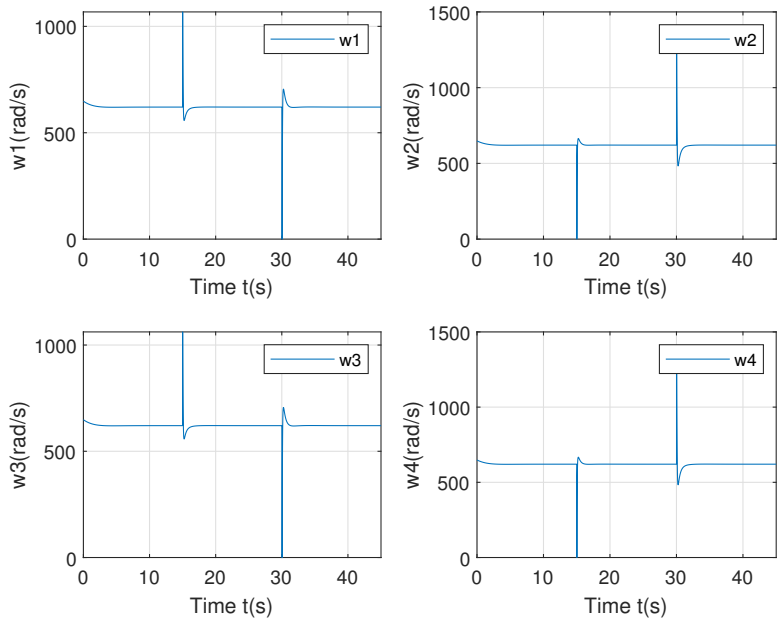


Figure. 47 Motor control input response for this simulation

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

3. 장애물 회피 시뮬레이션 조건

장애물 회피 시뮬레이션에서 쿼드로터의 초기 조건 및 파라미터는 Table.11, 13과 동일하다. Table.14는 초기 유도 명령이고, Table.15는 장애물 탐지 후 회피기동을 위해 새로운 경로지점을 설정한 유도명령이다.

Table 14: Initial guidance commands for trajectory tracking

	Position X_d	**Position** Y_d	**Position** Z_d
$\xi_d(1)$	0 m	0 m	1.5 m
$\xi_d(2)$	5 m	0 m	1.5 m
$\xi_d(3)$	5 m	0 m	0 m

Table 15: Guidance commands for trajectory tracking after obstacle detection

	Position X_d	**Position** Y_d	**Position** Z_d
$\xi_d(1)$	0 m	0 m	1.5 m
$\xi_d(3)$	2.0091 m	0 m	1.5 m
$\xi_d(3)$	2.0091 m	1 m	1.5 m
$\xi_d(4)$	3.5091 m	1 m	1.5 m
$\xi_d(5)$	3.5091 m	0 m	1.5 m
$\xi_d(6)$	5 m	0 m	1.5 m
$\xi_d(7)$	5 m	0 m	0 m

4. 장애물 회피 시뮬레이션 결과

위에서 실행한 위치 및 자세 시뮬레이션에서 장애물 회피 알고리즘을 추가하여 시뮬레이션을 수행하였다. 장애물 회피 시뮬레이션의 최종 결과는 Fig.48와 같이 Dancing plot으로 나타내었다. 장애물 탐지 후 회피를 위해 새로운 경로지점을 설정하여 새로운 유도명령을 통해 안전하게

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

회피하는 것을 확인할 수 있었다.

주어진 시뮬레이션 결과를 자세하게 분석하기 위해서 각 축의 대한 위치, 속도, 각도, 각속도 그리고 각 로터에 대한 RPM, 제어입력에 대하여 Fig.49-56에 도시하였다.

Fifgure.49을 살펴보면 초기 유도명령에 대하여 비행 중 장애물 탐지 후 장애물를 회피하여 초기 유도명령의 최종 목적지로 수렴하는지에 대하여 보여주고 있다. 장애물을 탐지하여 새로운 경로를 설정하는 시점에서 위치 X는 0.2 m, 위치 Z는 0.05 m의 오버슈트를 보이고 유도명령에 잘 수렴하는 것을 확인하였다. Figure.50-51는 X-Y평면상의 궤적추종 및 자세제어 결과와 3D상의 궤적추종 및 자세제어 결과이다. Figure.52은 각축에 대한 자세제어 성능을 보여주고 있다. Roll, Pitch의 경우 0.7-1 degree의 오버슈트를 보이고 자세명령에 잘 수렴하며, Yaw는 최대 1.4×10^{-3} degree의 오차를 보이며 잘 수렴하는 것을 확인하였다. Figure.53-54는 위치 및 자세제어의 시간에 따른 속도와 각속도 그래프이다. 마지막으로 Fig.55-56은 각 로터의 RPM과 제어입력 값이다.

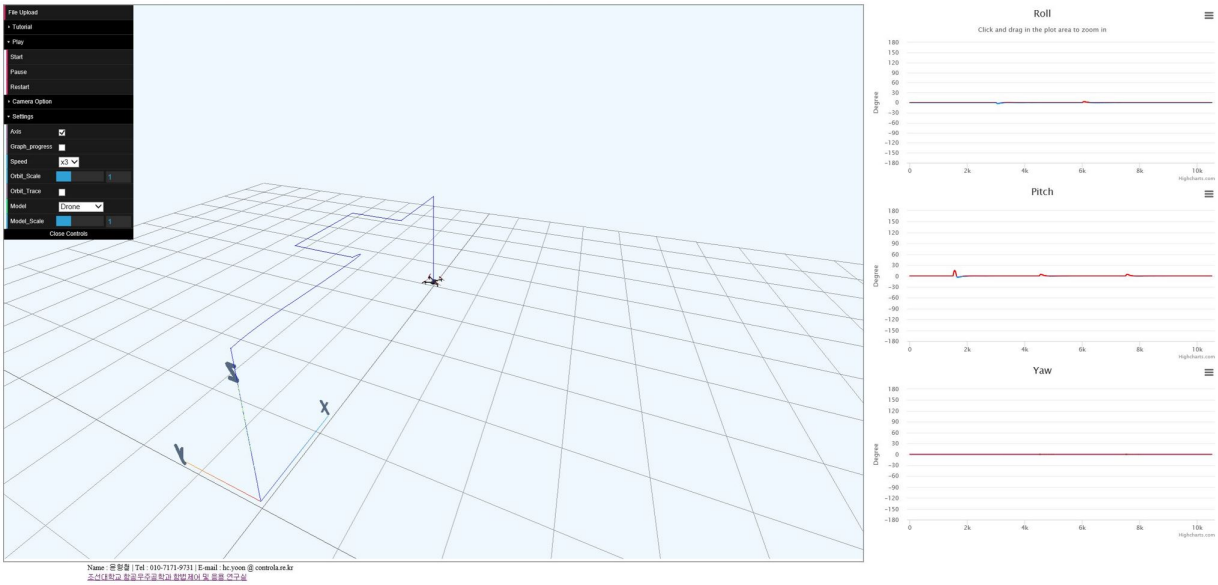


Figure. 48 Obstacle avoidance simulation using Dancing plot

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

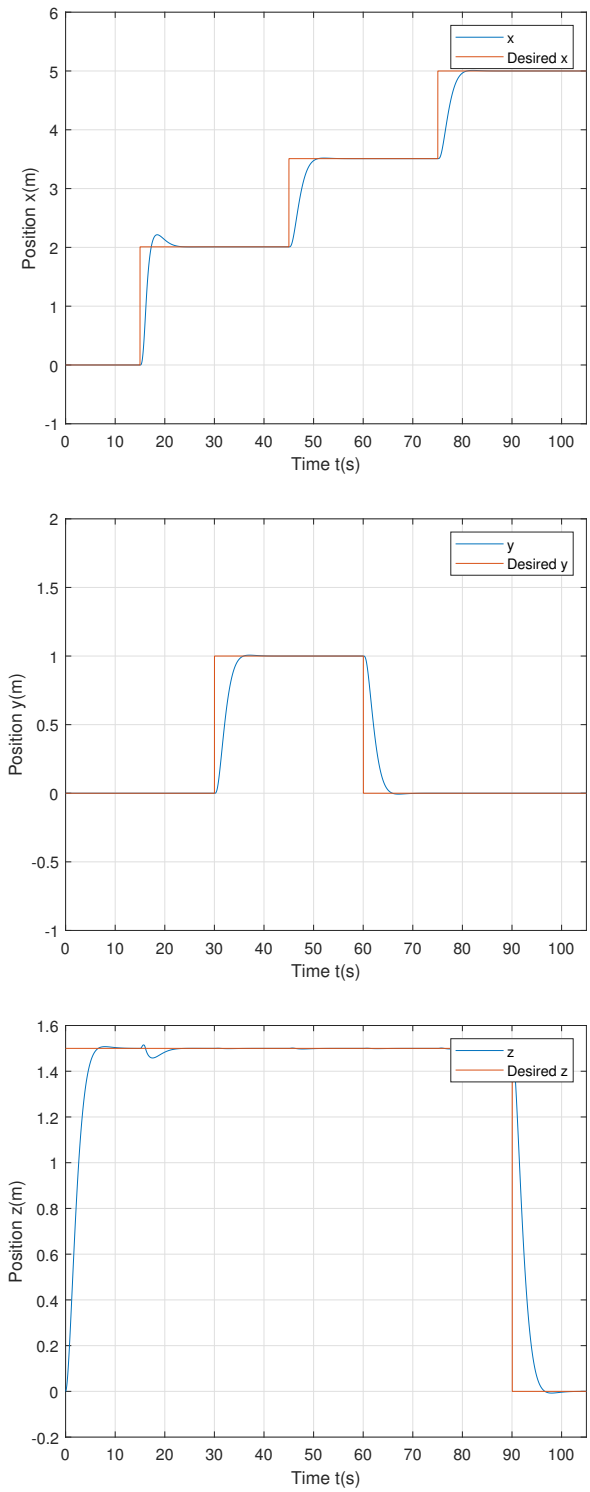


Figure. 49 Position history for the obstacle avoidance technique

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

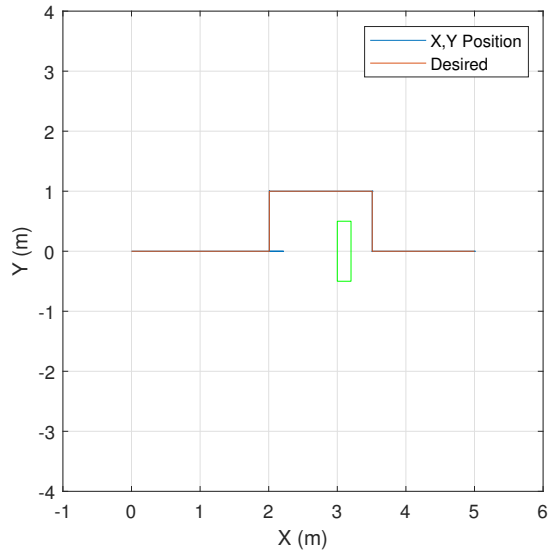


Figure. 50 Flight trajectory on the x-y plane for the obstacle avoidance technique.

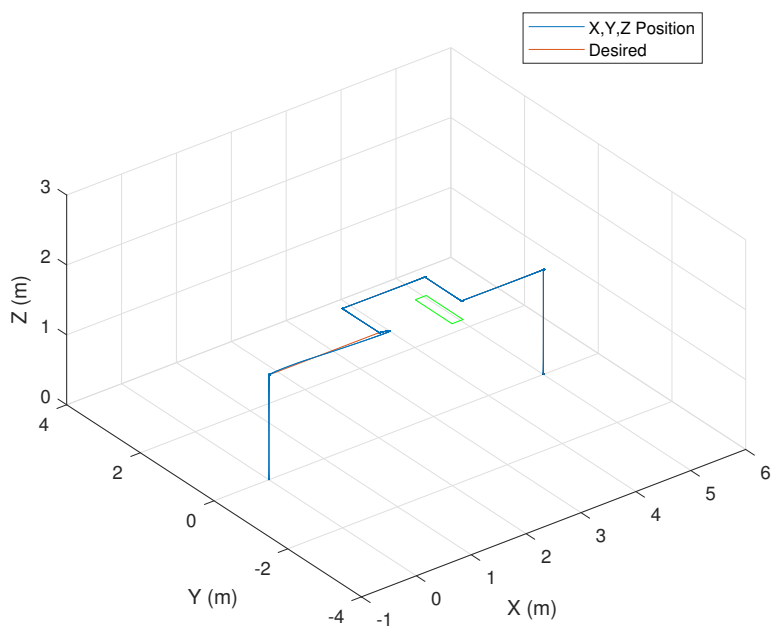


Figure. 51 Flight trajectory for the obstacle avoidance technique

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

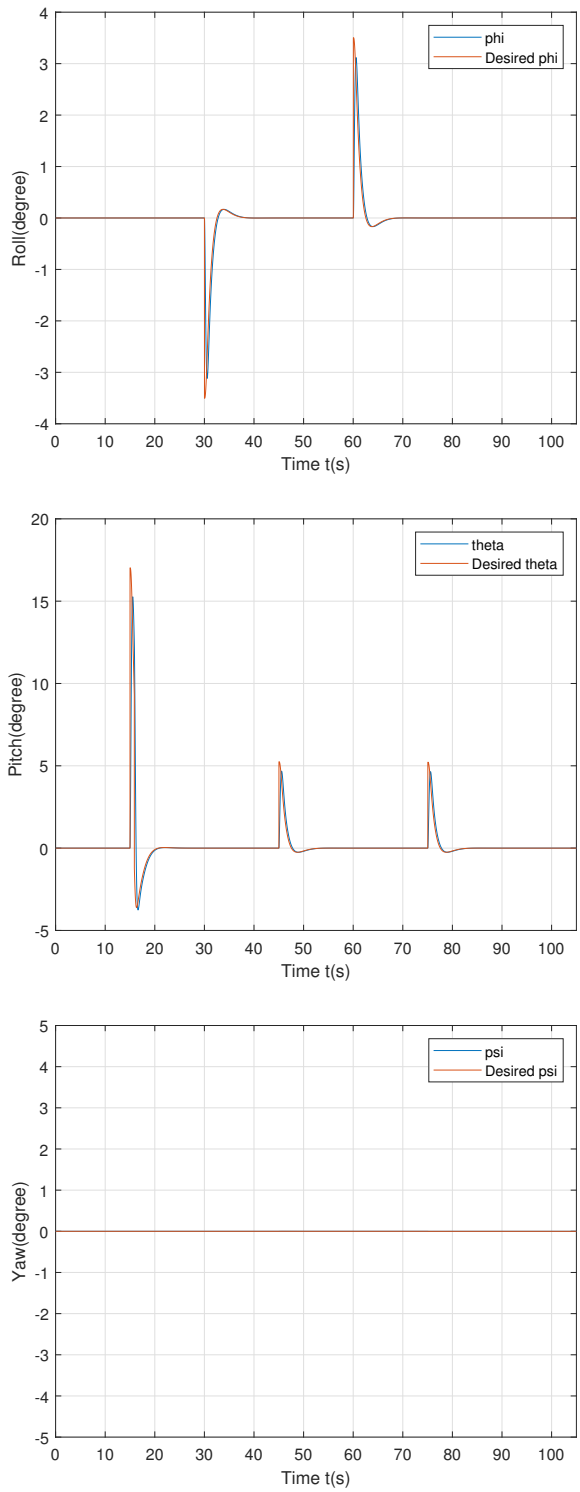


Figure. 52 Euler angle history for the obstacle avoidance technique

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

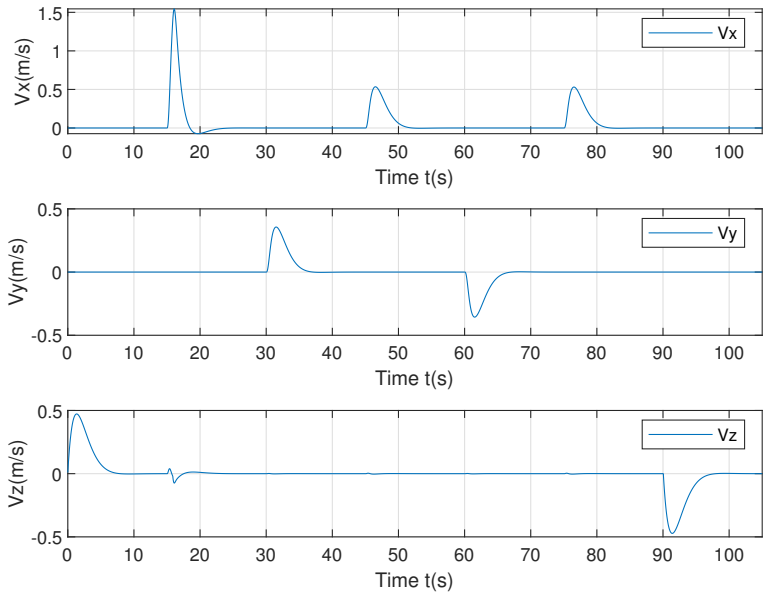


Figure. 53 Velocities history for the obstacle avoidance technique

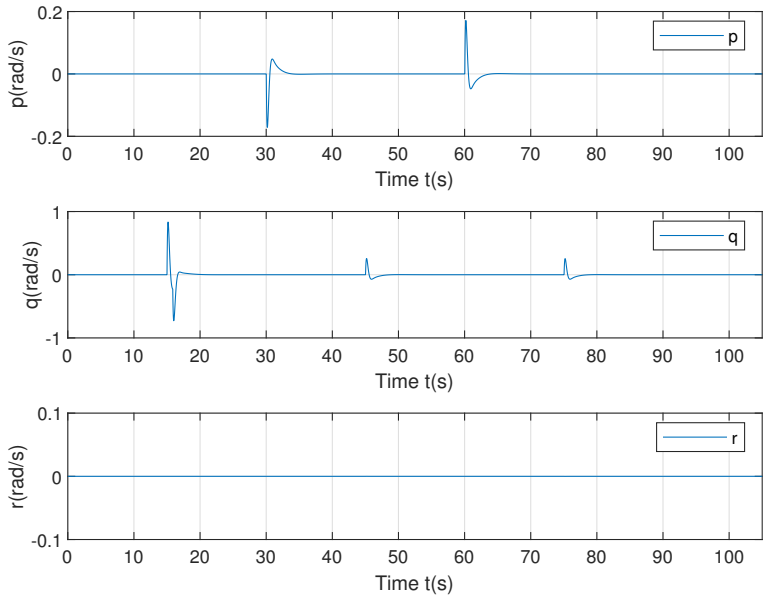


Figure. 54 Angular velocity history for the obstacle avoidance technique

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

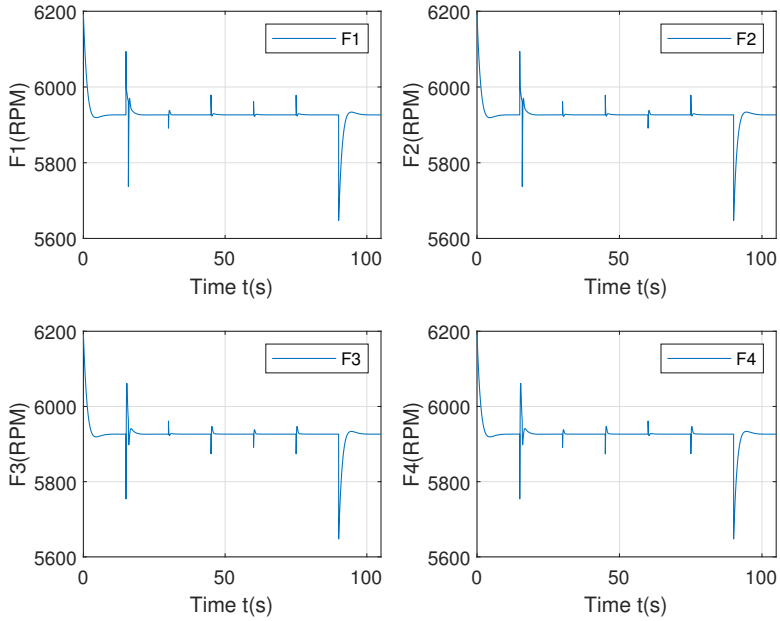


Figure. 55 Motor speed response for this simulation

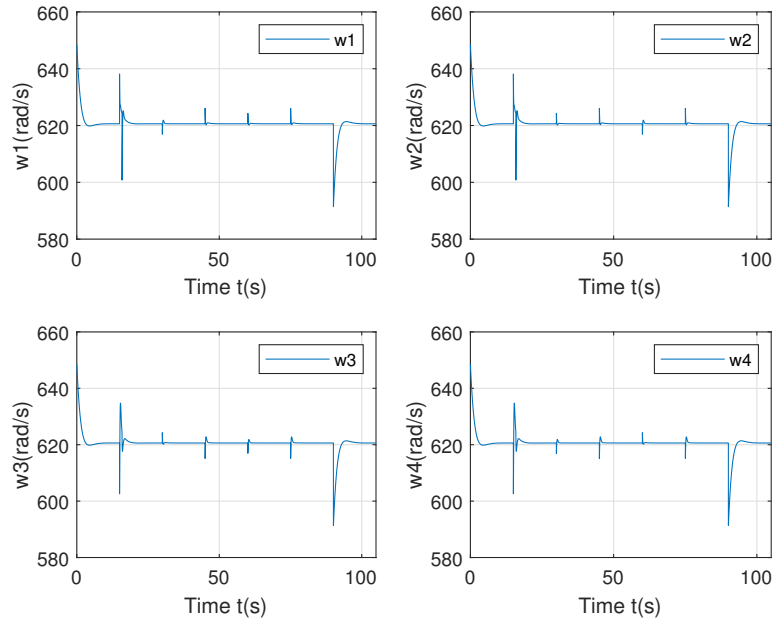


Figure. 56 Motor control input response for this simulation

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

5. 비행시험

Figure.57는 비행시험에 사용된 쿼드로터의 모습니다. 비행시험에서는 위치제어 및 자세제어의 성능을 확인하였다.

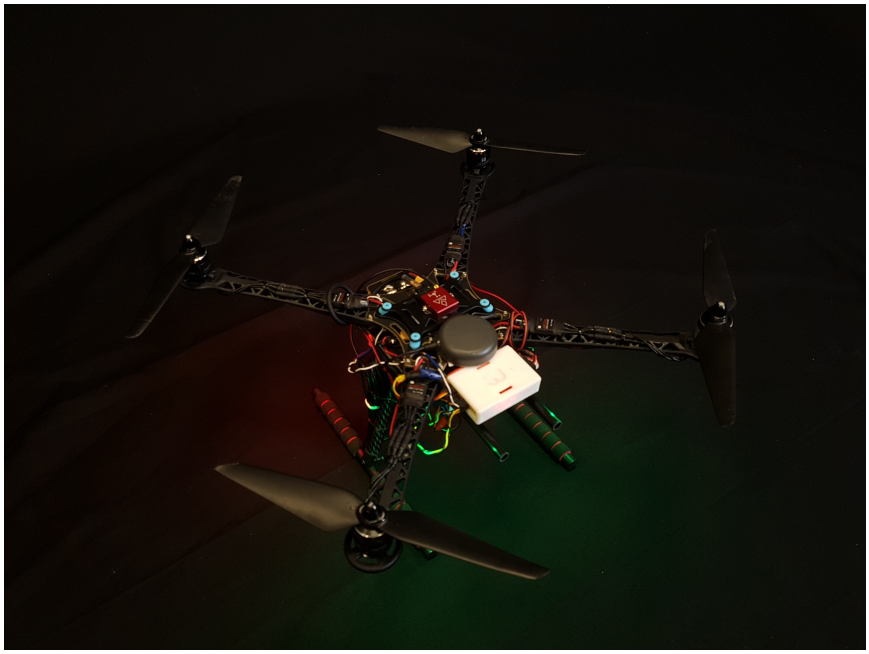


Figure. 57 Actual quad-rotor

비행시험 결과를 자세하게 분석하기 위해서 각 축의 대한 위치, 속도, 각도, 각속도에 대하여 Fig.58-61에 도시하였다. Figure.58을 살펴보면 고도제어 성능을 보여주고 있다. 고도 명령값 99.575 m으로 추종하는 것을 확인하였으며 최대 오차는 0.4 m이다. Figure.59는 속도에 대한 그래프이다. Figure.60을 살펴보면 자세제어 성능을 보여주고 있다. Roll, Pitch의 경우 2 degree의 오차를 가지며 명령값을 잘 추종하는 것을 볼 수 있었다. Figure.61을 살펴보면 각속도 r의 경우 최대 0.5 rad/s의 오차를 가지며 명령값을 잘 추종하는 것을 볼 수 있었다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

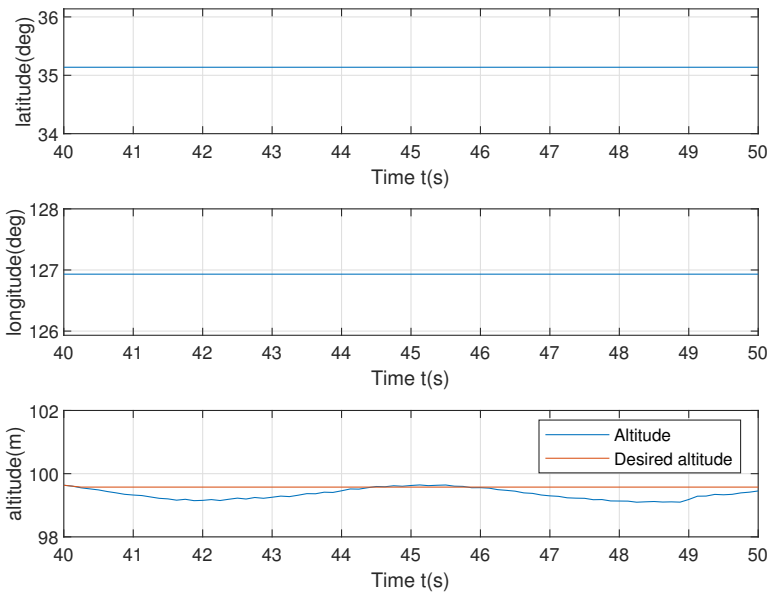


Figure. 58 Position history for the flight test

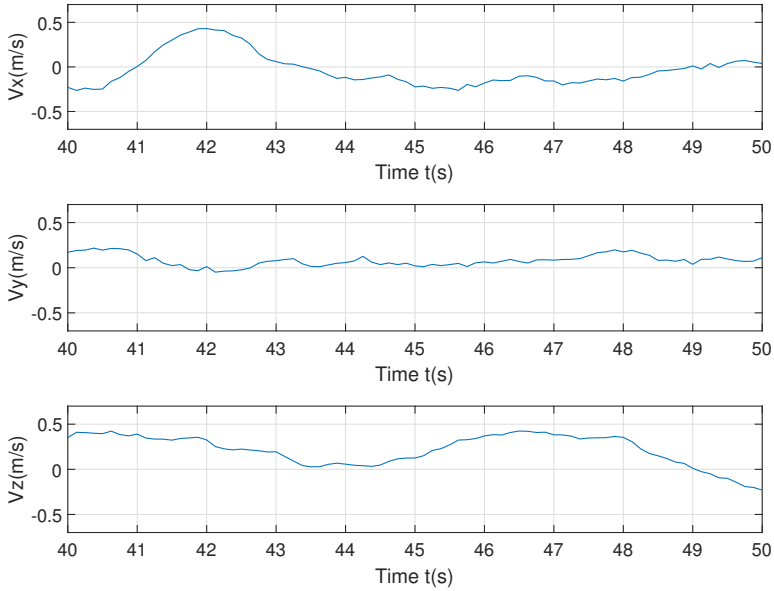


Figure. 59 Velocities history for the flight test

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

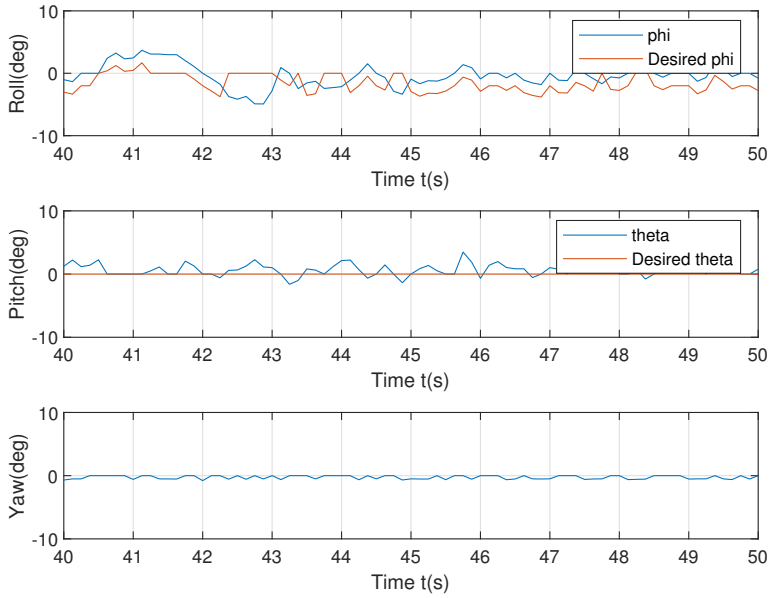


Figure. 60 Euler angle history for the flight test

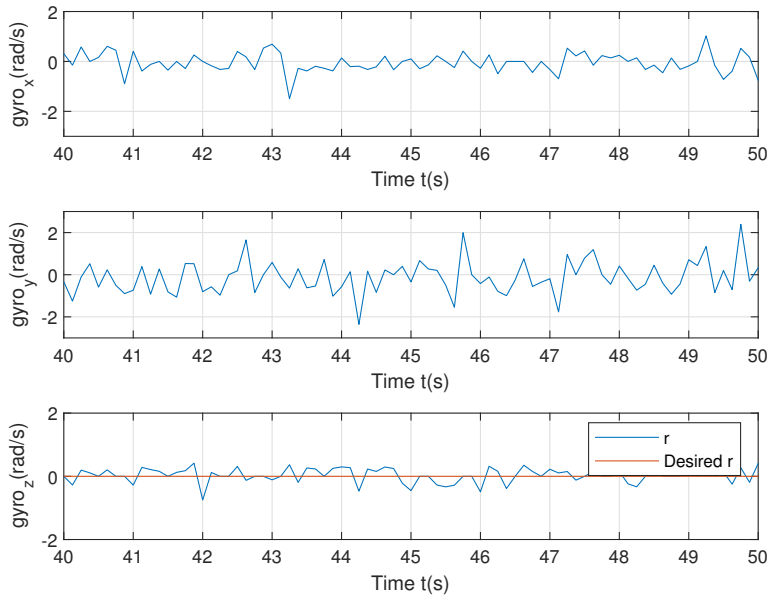


Figure. 61 Angular velocity history for the flight test

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

VII. 결론

본 논문에서는 PID 제어기를 쿼드로터의 동역학적 모델에 적용하여 위치 및 자세제어를 설계하여 수치 시뮬레이션을 통해 검증하며, Lidar센서 측정치 기반으로 장애물과 드론과의 충돌위험 인지 및 장애물 탐지 및 회피 알고리즘을 제안하고 장애물 탐지 및 회피 알고리즘 시뮬레이션을 통해 검증한다.

먼저, 유도명령이 주어지면 유도명령에 따라 위치제어를 하고 얻어진 가상벡터를 통해 자세 명령값을 갖는다. 이렇게 얻어진 6개의 명령값으로 6자유도 운동을 하며 최종 목적지를 추종하는 것을 수치 시뮬레이션을 통해 검증하였다.

충돌회피 기능은 드론이 목표 지점에 이르기까지 장애물과의 충돌이 발생하지 않도록 보장한다. 이를 위해서는 크게 두 가지 문제에 대한 방법을 결정해야한다. 첫 번째로 드론의 주변 환경을 감지하고 장애물 회피에 대한 유용한 정보를 추출하는 방법을 결정해야 하고, 두 번째는 어떻게 충돌회피가 수행되는지, 그리고 충돌 회피단계를 언제 시작하고 중지할지 결정해야한다. 이러한 문제기반으로 충돌회피와 기동 두 가지 구성요소로 나뉘고, 드론과 장애물의 대한 정보기반으로 장애물과의 충돌까지 걸리는 시간 및 장애물과의 거리를 기준으로 탐지모드, 추적모드, 회피모드로 나누어 비행한다. 따라서, 장애물 탐지 및 회피 알고리즘을 설계하고 수치 시뮬레이션을 통해 검증하였다.

최종적으로 실제 비행시험을 통해 위치 및 자세제어의 성능 검증을 하였다. 실제환경에서는 위성의 갯수에 따른 GPS의 오차와 외란으로 수치 시뮬레이션만큼의 성능이 나오지는 않았지만 적은 오차로 명령값에 잘 추종하는 것을 확인하였다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

참고문헌

[1] L. Areum, "Drone market and industry trend," *Convergence Research Policy Center*, 2017.

[2] H. H. Gabriel M. Hoffmann, "Quadrotor helicopter flight dynamics and control: Theory and experiment," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007.

[3] G. Hoffmann, S. Waslander, and C. Tomlin, "Distributed cooperative search using information-theoretic costs for particle filters, with quadrotor applications," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[4] J. M. B. Domingues, "Quadrotor prototype," *Instituto Superior Tecnico*, 2009.

[5] http://www.kca.go.kr/brd/m_32/view.do?seq=2281&multi_itm_seq=0.

[6] http://e-policy.kari.re.kr.

[7] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a uav with complementary low-cost sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.

[8] R. Gliarbudi, "Design and verification of autonomous hovering control system for quadrotor flying robot," *MasterThesis, Department ofAerospaceEngineering, KonkukUniversity*, 2011.

[9] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* IEEE, 2009, pp. 2878–2883.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[10] W. B. Slawomir Grzonka, Giorgio Grisetti, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, 2012.

[11] Y. Bi, H. Qin, M. Shan, J. Li, W. Liu, M. Lan, and B. M. Chen, "An autonomous quadrotor for indoor exploration with laser scanner and depth camera," in *Control and Automation (ICCA), 2016 12th IEEE International Conference on.* IEEE, 2016, pp. 50–55.

[12] Z. Fang, X.-y. Wang, and J. Sun, "Design and nonlinear control of an indoor quadrotor flying robot," in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on.* IEEE, 2010, pp. 429–434.

[13] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modelling and configuration stabilization for an x4-flyer." *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 217 – 222, 2002.

[14] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2004, pp. 2451–2456.

[15] X. Zhang, Z. Yang, T. Zhang, and Y. Shen, "An improved kalman filter for attitude determination of multi-rotor uavs based on low-cost mems sensors," in *Guidance, Navigation and Control Conference (CGNCC), 2016 IEEE Chinese.* IEEE, 2016, pp. 407–412.

[16] http://lancet.mit.edu/motor/motors3.html#torque.

[17] https://quadcopterproject.wordpress.com/.

[18] R. Beard, "Quadrotor dynamics and control rev 0.1," 2008.

[19] H. Lee, C. Hong, and H. Bang, "Nonlinear flight control based on model inversion using neural networks," in *39th JSASS Aircraft Symposium*, 2001.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[20] H.-W. Lim, M.-S. Jie, and W.-H. Choi, "Developing a safety flight assurance control (sfac) system using a quadcopter," 2016.

[21] A. Gibiansky, "Quadcopter dynamics, simulation, and control," *Andrew. gibiansky. com*, 2012.

[22] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, 2011.

[23] T. S. Alderete, "Simulator aero model implementation," *NASA Ames Research Center, Moffett Field, California*, 1995.

[24] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of control, Automation and systems*, vol. 7, no. 3, pp. 419–428, 2009.

[25] A. Tayebi and S. McGilvray, "Attitude stabilization of a vtol quadrotor aircraft," *IEEE Transactions on control systems technology*, vol. 14, no. 3, pp. 562–571, 2006.

[26] J. Liu *et al.*, "Advanced pid control and matlab simulation," *Beijing: Publishing House of electronics industry*, vol. 9, 2004.

[27] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET control theory & applications*, vol. 4, no. 11, pp. 2343–2355, 2010.

[28] A. Manjunath, "Path following by a quadrotor using virtual target pursuit guidance," 2016.

Collection @ chosun