



J. Korean Soc. Aeronaut. Space Sci. 48(9), 691-701(2020)

DOI:https://doi.org/10.5139/JKSAS.2020.48.9.691

ISSN 1225-1348(print), 2287-6871(online)

## 컨벡스 최적화 기법 기반 전기추진 수직이착륙 무인기의 추진 시스템 고장 대처를 위한 회전익 모드 믹서 설계

정연득<sup>1</sup>, 최형식<sup>2</sup>

### Actuator Mixer Design in Rotary-Wing Mode Based on Convex Optimization Technique for Electric VTOL UAV

Yeondeuk Jung<sup>1</sup> and Hyungsik Choi<sup>2</sup>

Korea Aerospace Research Institute

#### ABSTRACT

An actuator mixer design using convex optimization technique situation where the propulsion system of an electric VTOL UAV during vertical take-off and landing maneuvers is proposed. The attainable control set to analyze the impact from failure of each motor and propeller can be calculated and illustrated using the properties of the convex function. The control allocation can be defined as a convex function optimization problem to obtain an optimal solution in real time. The mixer is implemented using a convex optimization solver, and the performance of the control allocation methods is compared to the attainable control set. Finally, the proposed mixer is compared with other techniques with nonlinear six degree-of-freedom simulation.

#### 초 록

본 논문은 전기 추진 수직이착륙 무인기의 형상에 적합하도록 회전익 모드에서 추진시스템이 고장난 상황을 대처하기 위한 컨벡스 최적화 기법 기반의 믹서를 제시한다. 각 모터 및 프로펠러의 고장 영향성을 분석하기 위하여 컨벡스 함수 성질을 이용하여 가용 조종력 집합을 계산하였으며 이를 도시하는 방법을 기술하였다. 조종력 할당을 최적화 문제로 정의하고, 실시간으로 최적해를 구하기 위하여 컨벡스 함수로 문제를 재정의하였다. 컨벡스 최적화 솔버를 사용하여 수직 이착륙 비행 모드의 믹서를 구현하였으며 조종력 할당 기법들의 성능을 가용 조종력 범위 집합으로 비교하였다. 최종적으로 비선형 6자유도 시뮬레이션을 통하여 타기법(의사역행렬 기법, 재분배의사역행렬 기법)과 비교 분석하였다.

**Key Words** : Control Allocation(조종력 할당), eVTOL UAV(전기추진 수직이착륙 무인기), Convex Optimization(컨벡스 최적화)

#### 1. 서 론

무인항공기(Unmanned Aerial Vehicle, UAV)는 다

양한 형태로 다양한 국가에서 생산되고 있으며, 더욱 많은 국가에서 무인항공기가 활용되고 있다. 이와 더불어 제한적으로 군사적인 목적으로 활용되고 있던

† Received : May 7, 2020    Revised : July 28, 2020    Accepted : August 4, 2020

<sup>1</sup> Senior Researcher, <sup>2</sup> Principal Researcher

<sup>1</sup> Corresponding author, E-mail : jyd28@kari.re.kr

© 2020 The Korean Society for Aeronautical and Space Sciences

초기 무인항공기들은 현재에는 기반 시설 감시, 정찰, 물품 수송 등과 같이 광범위한 영역에서 많은 수가 운용되고 있다. 이러한 환경에서 무인항공기 연구 개발 방향은 초기에 비행에 관련된 기본적인 기능 개발에 중점을 두던 것과 달리 현재에는 안정성과 신뢰성 확보로 점차적으로 지향하고 있다.

많은 예산과 인력을 투자하여 개발하는 유인항공기와는 달리 상대적으로 적은 예산이 투입되는 무인항공기 분야에서는 유인항공기와 같은 안정성 및 신뢰도를 확보하기에는 어렵다. 따라서 고가의 신뢰도 높은 부품 사용보다는 중저가의 부품을 여러 개 사용하여 임무 신뢰도를 높이는 방법이 관심을 받고 있다. 이러한 흐름에서 가장 주목받는 무인항공기 형상은 다수의 프로펠러를 사용한 멀티콥터(Multi-copter) 비행체이다. 멀티콥터 비행체는 최소 3개 이상의 프로펠러로 구동되며, 임무 장비 탑재가 가능한 형상에서는 일반적으로 6개 이상의 프로펠러가 장착된다. 간단한 구조로 인하여 다양한 분야에 활용되고 있으며, 이에 힘입어 물품이 아닌 사람이 탑승할 수 있는 개인용항공기(Personal Aerial Vehicle, PAV)가 주목받기 시작하였다[1]. 현재 연구되는 대부분의 PAV는 멀티콥터 방식을 기본으로 하고 낮은 이동 속도 및 짧은 비행시간이라는 단점을 보완하는 형태로 개발되고 있다. 주로 멀티콥터 형상에서 프로펠러 배치 및 별도 작동기를 추가한 형태(Ehang 184, Blackfly, CityAirbus)와 비행모드별 전용 프로펠러가 있는 Lift-Cruise 형태(Kitty Hawk Cora, BRIEFLY, Uber eCRM003), 프로펠러 틸팅 매커니즘을 추가한 형태(Vahana, NEXUS)가 있다.

이러한 개발 사례들은 모두 과구동 시스템(Over-actuated system)을 채용하고 있으며, 이러한 시스템들은 원하는 조종력(추력, 3축 모멘트)을 생성하기 위해 수많은 작동기 출력 조합이 발생할 수 있다. 따라서 물리적인 한계 내에서 작동기가 고장이 나더라도 원하는 조종력을 생성할 수 있는 새로운 구동력 조합을 찾을 수 있다. 이러한 특성으로 인하여 멀티콥터 무인기 및 PAV들은 목표하는 추진시스템 신뢰도를 확보하기 위하여 여분의 작동기를 장착하고 고장 허용 제어(Fault-Tolerance Control, FTC) 기술을 적용하는 방향을 개발되고 있다[2-5]. FTC는 제어기법부터 조종력 할당까지 다양한 기법 및 접근 방법으로 연구되고 있으나, 고장에 대해서 근본적으로 대처할 수 있는 조종력 할당 기술은 FTC 중에서도 필수라고 여겨지고 있다.

조종력 할당은 요구되는 조종력을 생성하는 각 작동기의 구동력 조합을 계산하는 것으로, 의사역행렬(PseudoInverse, PI) 기법을 이용하여 주로 적용되었다[6,7]. PI 기법은 작동기의 임계점을 고려하지 못하여 특정 작동기가 포화 상태에 도달하면서 불안정해

지는 특성이 있다. 이러한 점을 보완하기 위하여 작동기의 한계 조건을 고려할 수 있도록 PI 기법을 다양한 방법으로 보완한 기법들이 발표되었다[8,9]. 이러한 기법들의 큰 줄기는 의사역행렬을 이용하여 필요 구동력을 계산하고 포화된 작동기에 분배된 구동력 초과분을 다시 재계산하여 재분배하는 것이다. 이러한 수정된 의사역행렬기법들은 비교적 단순한 구조와 적은 연산량으로 구현이 용이한 장점이 있지만, 비행체의 물리적인 한계점까지 모두 활용하지 못하는 단점이 있다. 반면에 direct allocation method[21] 및 linear/quadratic programming method[22]는 수치적인 방법으로 조종력 분배가 이루어지고, 상대적으로 연산량이 많지만 조종력 한계치까지 사용이 가능하다. 탑재 하드웨어가 많은 발전이 되었지만 불안정한 동특성을 안정화시키기 위해 빠른 주기로 구동이 필요한 믹서(Mixer) 루프에 온라인 최적화 연산은 한계가 있다. 이러한 한계로 인하여, [10]에서는 multi-parametric programming 기법으로 최적화 문제를 오프라인으로 해를 계산하여 데이터베이스화 시키고, 이를 piecewise affine function 형태로 불러 들여와 활용하였다. 이러한 접근 방법은 구축된 데이터베이스가 커지게 되면 필요한 메모리가 커지거나 런타임(Runtime) 시간이 늘어나게 된다.

컨벡스 최적화 기법은 목적 함수 및 부등식 조건을 컨벡스 함수로 정의하고 등식 조건을 affn 함수 형태로 정의하게 되면 가능해(Feasible solution)는 컨벡스 집합(Convex set)에 속하게 되는 성질로 최적해를 찾는다. 컨벡스 최적화 기법은 빠른 수렴 속도 및 수렴 보장성을 보장하기 때문에 다양한 영역에 널리 쓰이고 있다. 비행 제어 분야뿐만 아니라 회로 설계, 머신 러닝, 경제 등 다양한 분야에 잘 정립된 방법과 툴[11-15]을 통하여 적용되고 있다. 컨벡스 최적화 문제를 풀기 위해서는 in-house solver를 개발하거나 문제를 표준 형식(Standard form)으로 변형하여야 한다. 이러한 점들이 진입 장벽이 되었으나 domain-specific language(DSL)이 개발되면서 사용자가 인식하기 쉬운 상위 언어로 문제를 정의할 수 있어 비교적 간단히 알고리즘의 초기 성능을 확인할 수 있게 되었다. 이러한 특성을 지닌 DSL로는 MATLAB Toolbox 기반으로 개발된 YALMIP[16]와 CVX[15], C 코드 생성이 목적인 QCML[17], 파이썬 언어 기반인 PICOS[18]와 julia 기반인 Convex.jl[19]이 있다. 현재 가장 주목받고 있는 CVXPY[20]는 CVX를 기반으로 disciplined convex programming 개념을 도입하여 파이썬 언어로 개발되었다. 이와 같이 다양한 DSL이 개발되고 적용되고 있으나 빠른 연산 속도를 요구하는 분야에서는 아직 제한되는 부분이 많다. 특히나 항공 분야는 탑재되는 하드웨어의 무게에 많은 제약이 있어 충분한 연산 성능을 가진 임베디드 보드에 소프트웨어를 구동하지 못하는 경우가 많다.

본 논문에서는 분산 추진 방식의 전기추진 수직이착륙 무인기의 회전익 모드에서 추진시스템 고장에 대처하기 위한 실시간 컨벡스 최적화 기법 기반의 믹서를 제안한다. 비행체의 물리적인 한계까지 조종력을 사용하기 위하여 컨벡스 최적화 기법 기반의 믹서를 구현하였다. 최적화 기법은 가용 조종력 범위를 한계치까지 사용가능하나 최적 해를 계산하기 위해 많은 연산량이 필요하여 실시간 적용이 어렵다는 단점을 극복하기 위하여, 조종력 할당 문제를 컨벡스 함수 형태로 재정의하여 실시간 조종력 할당 믹서를 구현하였다. 또한 컨벡스 집합 성질을 이용한 가용 조종력 범위 변화를 통해 추진시스템 고장 영향성을 분석하는 방법을 소개한다.

본 논문에서는 먼저 전기추진 수직 이착륙 무인기에 대해서 설명하고 동역학 모델링을 통하여 조종력과 구동력간의 관계식과 컨벡스 집합의 특성을 이용하여 전기 추진 수직 이착륙 무인기의 조종력 범위에 대해서 분석 및 서술한다. 의사역행렬 기법, 재분배 의사역행렬 기법과 제안한 컨벡스 실시간 최적화 기법들의 성능을 가용 조종력의 범위로 비교한다. 3가지 기법(의사역행렬 기법, 재분배 의사역행렬 기법, 컨벡스 최적화 기법)들로 조종력 할당 믹서를 구현하여 요구되는 조종력과 생성한 조종력의 차이를 비교하고, 최종적으로 6자유도 비선형 시뮬레이션에서 추진시스템 고장 상황에서 각 기법들 간의 성능 차이를 서술한다.

## II. 본 론

### 2.1 문제 정의

#### 2.1.1 전기 수직 이착륙 무인기

대상 비행체는 본 연구원에서 개발 중인 Fig. 1과 같은 전기 수직 이착륙(Electric Vertical Take-off and Landing, eVTOL) 무인기의 형상적인 구성과 유사한 제어기 검증용 테스트베드이다. 테스트베드는 아래 그림과 같이 8개의 프로펠러를 주익을 기준으로 전후로 4개씩 장착하고, 고정익 형상을 지니고 있다. 주익을 따라서 프로펠러가 배치되어 있으며 동체와 간섭하지 않는 환경에서 주익 위 아래로 프로펠러가 배치되어 있다. 동체 중심을 기준으로 좌우 대칭하게 프

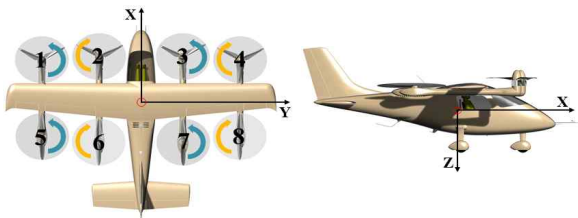


Fig. 1. Test-bed for eVTOL UAV

로펠러가 배치되어 있는 멀티콥터(Multicopter) 비행체와는 달리 항공기의 y축 방향으로 편중되어 배치된 것이 특징이다. 비행체 스케일은 프로펠러 13인치가 장착되며 이륙 중량 6.0kg이다.

#### 2.1.2 동역학 모델링

비행체의 6자유도 비선형 방정식은 비행체 질량( $m$ )에 대해 아래와 같이 표현된다.

$$m\dot{\mathbf{v}}_B + \boldsymbol{\omega} \times \mathbf{v}_B = \mathbf{F}_B + \mathbf{F}_u + m\mathbf{g} \quad (1)$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}_B + \mathbf{M}_u \quad (2)$$

여기서 속도( $\mathbf{v}_B$ ), 각속도( $\boldsymbol{\omega}$ )는 동체좌표계를 기준으로 하며,  $\mathbf{g}$ 는 중력가속도를 의미하며,  $\mathbf{J}$ 는 비행체의 관성모멘트를 의미한다. 비행체에 작용하는 외부 힘과 모멘트는  $\mathbf{F}_B, \mathbf{M}_B$ 이고, 비행체의 작동기(프로펠러)에 의해 생성되는 힘과 모멘트는  $\mathbf{F}_u, \mathbf{M}_u$ 이다. 여기서 eVTOL 비행체에서 조종력( $\mathbf{u} = [\mathbf{F}_u \ \mathbf{M}_u]^T$ )는 프로펠러 회전에서만 발생하며 프로펠러의 회전수와 제어유효행렬(Control effectiveness matrix)  $\mathbf{K}$ 와의 선형 관계식으로 표현할 수 있다.

$$\mathbf{u} = \begin{bmatrix} T_{cmd} & L_{cmd} & M_{cmd} & N_{cmd} \end{bmatrix}^T = \mathbf{K}\mathbf{w} \quad (3)$$

여기서  $T_{cmd}$ 는 8개의 프로펠러에서 생성하는 추력의 합이며,  $L_{cmd}, M_{cmd}, N_{cmd}$ 는 각각 프로펠러의 회전에 의해 발생하는 롤(Roll), 피치(Pitch), 요(Yaw)축 모멘트이고,  $\mathbf{w} = [w_1^2 \ \dots \ w_8^2]^T$ 는 각각의 프로펠러 회전수의 제곱 행렬을 의미한다. 각각의 프로펠러가 생성하는 추력( $F_i$ )과 반토크( $N_i$ )는 추력 상수( $C_i$ )와 반토크 상수( $C_p$ )로 아래와 같이 단순한 관계식으로 근사화 가능하다.

$$F_i = C_i w_i^2 \quad (4)$$

$$N_i = -C_p w_i^2 \text{sign}(\chi_i) \quad (5)$$

$\chi_i$ 는  $i$ 번째 프로펠러의 회전 방향을 의미한다. 위의 식을 바탕으로 제어유효행렬을 아래와 같이 정의할 수 있다.

$$\mathbf{K} = \begin{bmatrix} C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 & C_1 \\ l_1 C_1 & l_2 C_1 & -l_3 C_1 & -l_4 C_1 & l_5 C_1 & l_6 C_1 & -l_7 C_1 & -l_8 C_1 \\ d_1 C_1 & d_2 C_1 & d_3 C_1 & d_4 C_1 & -d_5 C_1 & -d_6 C_1 & -d_7 C_1 & -d_8 C_1 \\ C_p & -C_p & C_p & -C_p & C_p & -C_p & C_p & -C_p \end{bmatrix} \quad (6)$$

상수  $l_i$ 와  $d_i$ 는  $i$ 번째 프로펠러와 무게 중심과의 y축 및 x축 거리를 의미한다.

조종력과 프로펠러 회전수와의 관계식  $\mathbf{u} = \mathbf{K}\mathbf{w}$ 는 방정식보다 변수가 많은 과구동 시스템(Over-actuated system)이다. 따라서 주어진 조종력을 생성할 수 있

는 프로펠러 회전수는 무수히 많이 존재한다. 일반적으로는 하나의 관계식을 선정하여 회전수를 계산하지만, 고장과 같은 정상 상태에서 벗어난 경우에는 충분한 성능을 확보하기 힘들다. 따라서 추진시스템(프로펠러, 모터, 모터 컨트롤러)이 고장난 상황에서 주어진 조종력을 만족시키는 회전수를 구하는 방법이 필요하다. 본 논문에서는 최적해를 실시간으로 구할 수 있는 컨벡스 기반 실시간 최적화 방법을 제시하며, 의사역행렬 및 재분배 의사역행렬 기법과 그 성능을 비교하였다.

## 2.2 가용 조종력 집합

가용 조종력 집합은(Attainable Control Set, ACS)는 비행체가 생성할 수 있는 조종력( $\mathbf{u}$ )의 범위를 의미하며, 수직이착륙 비행체의 경우에는 추력  $T_{cmd}$ 와 톨, 피치, 요 축 모멘트( $L_{cmd}, M_{cmd}, N_{cmd}$ )로 구성된다. 일반적인 임무 상황에서는 ACS는 거의 일정하게 유지되나 고장과 같은 특수한 상황에서는 그 범위가 변하게 된다.

고정의 비행체와 비교하여 수직이착륙 비행체들은 상대적으로 외부 공기 유동에 의한 조종력보다는 프로펠러나 로터의 회전으로 발생하는 추진력이 주요한 구동력이다. 따라서 전기 수직 이착륙 무인기의 ACS는 각 프로펠러의 회전수를 기반으로 계산이 가능하며, 추진시스템 고장에 따른 영향성도 예측 및 분석이 ACS 기반으로 가능하다.

다면체(Polyhedron)  $R$ 은 유한한 선형 부등식과 등식의 교집합으로 정의되며, 컨벡스 집합(convex set)의 일종이며 간단히 다음과 같이 정의된다.

$$P = \{ \mathbf{x} \mid \mathbf{a}_i^T \mathbf{x} \leq \mathbf{b}_i, i = 1, \dots, m, \mathbf{c}_j^T \mathbf{x} = \mathbf{d}_j, j = 1, \dots, p \} \quad (7)$$

이를 이용하여 비행체의 장착되어 있는  $n$ 개의 프로펠러의 회전수 다면체( $Y$ )는 아래와 같이 표현할 수 있다.

$$\Psi = \{ \mathbf{w} \in \mathbb{R}^m \mid \mathbf{w}_{\min} \leq \mathbf{w} \leq \mathbf{w}_{\max} \} \quad (8)$$

$Y$ 는 각각 최소 및 최대값으로 제한된 공간이므로 초다면체(polytope)이며, 비행체가 생성할 수 있는 조종력은 모두  $Y$ 에 속한다. 따라서 컨벡스 집합인 ACS의 경계선(boundary)은  $Y$ 의 경계선에서 제어 유효행렬  $\mathbf{K}$ 를 이용한 선형 변환(Linear transformation)을 통해 계산된다. 선형 변환은 볼록성(Convexity)을 유지하는 연산이므로 결과 값인 ACS 또한 볼록 집합(Convex Set)이다.

초다면체를 표현하는 방법으로는 식 (7)과 같이 선형 부등식 및 등식으로 표현하는 half-space representation(H-representation)과 초다면체의 극점(Extreme point)들로 표현하는 vertex representation(V-representation)이 있다. H-representation로 표현된 초다면체를 V-

representation으로의 변환을 vertex enumeration이라 하며 reverse search, double-description method, primal-dual polytope method 등과 같은 방법으로 해를 구할 수 있다. 본 논문에서는 primal-dual polytope method를 기반으로 MATLAB의 “convexhull” 함수를 이용하여 다음과 같이  $Y$ 의 극점  $\mathbf{V}_\Psi$ 을 구하였다[23].

$$\mathbf{V}_\Psi = f_{lcon2vert}(\Psi) \quad (9)$$

여기서 함수  $f_{lcon2vert}(\cdot)$ 는 vertex enumeration problem의 해를 구하는 함수로서 초다면체를 나타내는 선형 부등식 및 등식을 인수로 받아 극점을 계산한다. 계산된 극점은 제어유효행렬을 사용하여 극점을 다음과 같이 계산할 수 있다.

$$\mathbf{V}'_\Omega = \mathbf{K} \cdot \mathbf{V}_\Psi \quad (10)$$

위의 극점은 실제로 계산되어야 하는 ACS  $\Omega$ 의 극점과는 다소 차이가 있다. 일반적으로 고차원 공간에서 저차원 공간으로 투사(projection)하게 되면 필연적으로 다수 점들이 하나의 점으로 투사되게 된다. 따라서 프로펠러 개수가 5개 이상인 비행체부터는  $Y$ 가 ACS에 비해 고차원 공간이 된다. 그러므로  $\mathbf{V}'_\Omega$ 는 ACS의 극점과 부수적인 나머지 점을 포함한 집합이 된다. 즉,  $\mathbf{V}'_\Omega$ 는 ACS의 극점을 포함하지만, 내부 공간의 점도 포함한다. 앞에서 언급한 듯이 선형 변환은 볼록성이 보존되므로  $\mathbf{V}'_\Omega$ 의 볼록 껍질(convex hull)을 계산하며 ACS의 극점  $\mathbf{V}_\Omega$ 를 구할 수 있다.

$$\mathbf{V}_\Omega = f_{convhull}(\mathbf{V}'_\Omega) \quad (11)$$

함수  $f_{convhull}$ 는 인수로 받은 포인트들로 볼록 껍질을 계산하는 함수이며 MATLAB 함수 및 참고문헌 [1]을 사용하였다.  $\mathbf{V}_\Omega$ 를 H-representation 방식으로 변환하면 가용 조종력 집합을 구할 수 있다.

$$\Omega = \{ \mathbf{u} \in \mathbb{R}^n \mid \mathbf{H} \cdot \mathbf{u} \leq \mathbf{k} \} \quad (12)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^T \\ \vdots \\ \mathbf{h}_M^T \end{bmatrix}, \mathbf{k} = \begin{bmatrix} k_1 \\ \vdots \\ k_M \end{bmatrix} \quad (13)$$

여기서  $\mathbf{h}_i \cdot \mathbf{u} \leq k_i$ 는 총  $M$ 개 중에서  $i$ 번째 반무한체(halfspace)를 의미한다. 즉,  $M$ 개의 조건들은 ACS의 경계선을 이루면서 초다면체를 형성하고, 초다면체 경계 및 내부 점들은 비행체가 생성할 수 있는 조종력의 가능 해(feasible solution)이다.

전기수직이착륙 무인기는 총 8개의 프로펠러를 장착하고 있으므로  $m=8$ 이고 회전 수 공간  $Y$ 는 총 16개의 선형 부등식으로 정의된다. 이를 식 (9)를 이용하여 256개의 극점을 구할 수 있으며 최종적으로 식 (12)을 이용하여 ACS의  $M=52$ 개의 선형 부등식



으로 얻어진다. 계산된 ACS  $\Omega$ 는 4차원 공간이므로 이를 그래프로 시각화하기 위해서는 최소 3차원으로 투영시켜야 한다. 투영시키는 방법으로는 다양하게 존재하지만, 본 논문에서는 직관적으로 물리적인 의미를 주입시키기 위해서는  $\Upsilon$ 와  $\Omega$ 에 선형 부등식 및 등식을 추가하여 시각화하였다.

ACS의 시각화를 위해서는 하나의 이상의 축에 제한 조건을 주어서 2, 3차원으로 투영하여야 한다. 일반적으로 회전익 항공기에서는 정지 비행 조건이 많이 고려되므로 추력 및 요 모멘트 제한 조건을 많이 채용한다. 요축 모멘트가 0인 상태에서의 ACS를 구하기 위하여 아래와 같은 선형 등식을 추가하여  $\Omega_{N=0}$ 를 다시 정의할 수 있다.

$$\Omega_{N=0} = \{ \mathbf{u} \in \mathbb{R}^n \mid \mathbf{H} \cdot \mathbf{u} \leq \mathbf{k}, \mathbf{C}^T \mathbf{u} = \mathbf{d} \} \quad (14)$$

또한, 아래 선형 등식과 같이 요축 모멘트 평형 조건에 정비 비행 추력 조건을 추가하면 롤-피치 축에 대한 2차원 ACS를 시각화할 수 있다.

앞에서는 시각화를 위한 제한 조건을 추가하는 방법을 설명하였고, 고장 영향성도 위와 유사한 방식으로 추가 가능하다. 본 논문에서는 고장 상황 대처를 고장이 발생한 추진시스템을 그 부정확성으로 인하여 격리시키는 방법으로 접근한다. 따라서 고장난 추진시스템은 정지시키는 개념으로 생각하면, 프로펠러 회전수 다면체에 등식 조건을 추가하여 표현할 수 있다. 즉, 식 (8)에  $w_i = 0$  조건을 추가하고, 식 (9)~(13)까지의 과정을 거치면 된다.

이와 같이 계산되는  $\Omega_{N=0}$ 는 Fig. 2와 같이 시각화된다. Fig. 2는 전기 수직 이착륙 무인기가 정상 상태일 때의 ACS를 시각화한 그래프이며, 요축 모멘트가 0인 상황에서 비행체가 생성할 수 있는 피치, 롤 모

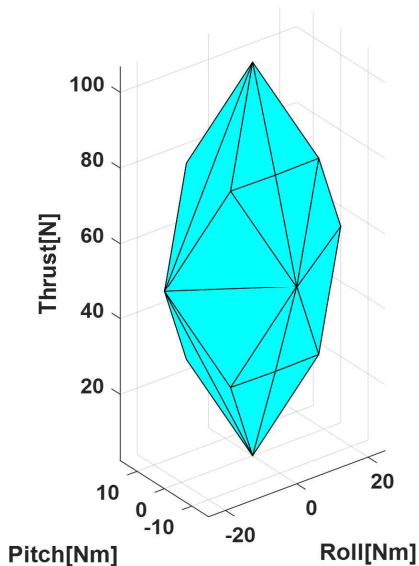


Fig. 2. ACS eVTOL when  $N_{cmd} = 0$

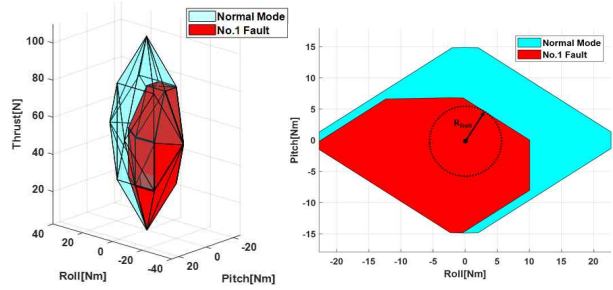


Fig. 3. Analysis of performance degradation for 1st motor failure

멘트와 추력이다. 추력 50% 지점을 기준으로 대칭인 형상을 가지며, 피치 축보다는 롤 축 가용 모멘트 범위가 넓다. 이는 주의 기준으로 모터가 배치되어 길어진 모멘트 양의 영향성을 받은 것이다.

Figure 3은 비행체가 정상인 상태와 1번 모터가 고장난 상태에서의 ACS를 시각화한 그래프이다. 좌측 그래프는 요 모멘트가 0인 조건이며, 1번 모터가 주의 가장자리에 배치되어 롤축 모멘트에 미치는 영향이 크며, 이를 상쇄하면서 요 모멘트를 0을 유지할 수 있는 공간이 매우 줄어드는 것을 확인할 수 있다. 또한 요구하는 추력이 높을수록 상황이 악화되는 것을 쉽게 추정할 수 있다. 오른쪽 그래프는 좌측 그래프에서 정지 비행의 추력 레벨에서 컷팅한 그래프이며, 요축 모멘트가 0이고 생성 추력이 자중인 조건에서 자세 제어를 위해 비행체가 생성할 수 있는 롤 및 피치 모멘트 범위를 나타낸다. 그래프에서 비행체 어법칙에서 가용할 수 있는 조종력이 50% 이상 저하된 것을 확인할 수 있다.

### 2.3 조종력 할당 기법

#### 2.3.1 의사역행렬(Pseudo Inverse Matrix, PI) 기법

일반적으로 믹서는 고정된 관계식을 사용하여, 단 하나의 해만 구하도록 구현되어 있다. 빠른 구동 속도를 요구하는 믹서의 특성상 실제 환경에서는 유용한 방법이나, 모터 및 프로펠러 고장 상황에서는 오히려 제어 불능 상태를 야기할 수도 있다. 단일 계산식은 계산된 모터 회전수에 의해 발생하는 제어 모멘트는 각 축에 대해서 독립적으로 영향을 미치는 것으로 가정으로 하지만, 고장 상황 시에는 이러한 가정이 부적합하다.

추력 및 제어 모멘트와 프로펠러 회전수와의 관계식에서 가장 간단한 방법은 역행렬을 이용한 방법이다. 다만 제어 유효 행렬이 정사행렬이 아니므로 의사역행렬을 다음과 같이 계산하여 적용한다.

$$\mathbf{w} = \mathbf{K}^+ \mathbf{u} = \mathbf{K}^T (\mathbf{K}\mathbf{K}^T)^{-1} \mathbf{u} \quad (15)$$

의사역행렬기법은 작동기의 포화상태를 고려할 수 없는 특성을 가진다. 충분한 조종력 여유가 있는 상

황에서는 성능저하가 발생하지 않으나 고장과 같은 조종력 여유가 제한된 상황에서는 작동기 포화가 발생하여 생성된 제어 모멘트의 각 축의 독립성이 유지가 되지 않아서 한계점을 보인다. 즉, 고도 상승 및 하강 속도 감속을 위하여 추가적인 추력이 요구되는 상황에서 각 프로펠러의 회전수가 증가할 때에 프로펠러 회전수가 최대가 되는 프로펠러가 있을 경우에는 각 축에 대해서 커플링이 생긴다. 즉, 추력 생성을 위한 회전수가 이미 해당 프로펠러의 최대 회전수가 되면 제어 모멘트 생성이 되지 않아 원하는 축의 모멘트가 발생하게 된다.

### 2.3.2 재분배 의사역행렬(Distributed Pseudo Inverse Matrix, DPI) 기법

의사역행렬기법의 단점을 보완하기 위해서 좀 더 넓은 ACS를 사용하기 위하여 다양한 방식으로 접근한 기법들이 많이 연구되었다. 이와 같은 연구 결과들의 핵심 목표는 의사역행렬기법보다는 좀 더 넓은 가용 조종력 범위를 사용하는 것이다. 따라서 여러 번의 반복 루프를 통하여 포화된 작동기를 식별하고, 다른 작동기에 구동력을 분배하는 방법을 채용한 기법을 구현하였다. 본 논문에서는 빠른 연산 속도를 가지며, 포화된 작동기를 고려하여 의사역행렬을 수정해가면서 조종력을 재분배하는 기법을 선정하여 구현하였다[9].

의사역행렬 재분배기법의 핵심은 1) 추력 및 제어 모멘트 생성에 필요한 프로펠러 회전수 분리와 2) 프로펠러 포화상태 고려이다. 알고리즘을 구현하여 비교할 때 관찰되는 가장 큰 차이점은 프로펠러 포화 상태에서 원하는 제어 모멘트 수치를 만족시키지는 못하지만, 작동기 포화 상태로 인해 원하지 않는 다른 축의 모멘트가 생성되지 않는다. 조종력  $\mathbf{u}$ 는 다음과 같이 나눌 수 있다.

$$\mathbf{u} = \mathbf{u}_\Delta + \mathbf{u}_0 \quad (16)$$

$\mathbf{u}_0$ 는 정지 비행을 위한 비행체 자중의 추력을 의미하고,  $\mathbf{u}_\Delta$ 는 비행체 안정화 및 추종을 위한 조종력을 의미한다. 여기서, 실제적으로 믹서에서 생성 가능한 추력 및 모멘트  $\mathbf{u}_{real}$ 는 다음과 같다.

$$\mathbf{u}_{real} = c\mathbf{u}_\Delta + \mathbf{u}_0 \quad (17)$$

$\mathbf{u}_{real}$ 은 어떠한 상황에서도  $\mathbf{u}$ 보다는 작아야 하므로,  $0 \leq c \leq 1$ 이다. 따라서,  $\mathbf{u}_\Delta$ 는 다음과 같이 정의할 수 있다.

$$\mathbf{u}_\Delta = [T_{cmd} - mg \quad L_{cmd} \quad M_{cmd} \quad N_{cmd}]^T \quad (18)$$

수정 의사역행렬 재분배 기법은 매 루프마다  $c$ 와  $\mathbf{w}$ 를 계산하고,  $c$ 의 최대값일 때 그 때의  $\mathbf{w}$ 를 최종값으로 정한다. 따라서,  $k$ 번째 루프에서의 제어 조종력은 다음과 같다.

$$\mathbf{u}_{real}(k) = c(k)\mathbf{u}_\Delta(k) + \mathbf{u}_0 \quad (19)$$

위의 식은 포화 상태인 프로펠러 회전수( $\mathbf{w}_{fix}$ )와 추가 구동력 생성이 가능한 프로펠러 회전수( $\mathbf{w}_{var}$ )로 분리할 수 있다.

$$\mathbf{u}_{real}(k) = \mathbf{K}_{var}(k)\mathbf{w}_{var}(k) + \mathbf{K}_{fix}(k)\mathbf{w}_{fix}(k) \quad (20)$$

$\mathbf{u}_\Delta$ 는 비행체를 원하는 상태로 추종하기 위한 추력 및 제어 모멘트이며, 비행체의 트림 상태를 유지하기 위한 성분( $\mathbf{w}_{var0}$ )도 포함하고 있다. 따라서  $\mathbf{w}_{var0}$ 를 포함한 식은 다음과 같다.

$$\mathbf{u}_{real}(k) = c(k)\mathbf{K}_{var}(k)\mathbf{w}_{var}(k) + \mathbf{K}_{var}(k)\mathbf{w}_{var0}(k) + \mathbf{K}_{fix}(k)\mathbf{w}_{fix}(k) \quad (21)$$

따라서  $\mathbf{w}_{var}$ 는 다음과 같이 정의할 수 있다.

$$\mathbf{w}_{var}(k) = c(k)\mathbf{w}_{var\Delta}(k) + \mathbf{w}_{var0}(k) \quad (22)$$

위의 두 식을 비교하면  $\mathbf{w}_{var0}$ 은 다음과 같은 관계식을 통하여 계산된다.

$$\mathbf{u}_\Delta(k) = \mathbf{K}_{var}(k)\mathbf{w}_{var\Delta}(k) \quad (23)$$

$$\mathbf{w}_{var0}(k) = \mathbf{K}_{var}^+(k)\mathbf{u}_\Delta(k) \quad (24)$$

$\mathbf{w}_{var0}$ 는  $\mathbf{u}_0$  관계식을 이용하여 다음과 같은 관계식을 가진다.

$$\mathbf{u}_0 = \mathbf{K}_{var}(k)\mathbf{w}_{var0}(k) + \mathbf{K}_{fix}(k)\mathbf{w}_{fix}(k) \quad (25)$$

$$\mathbf{w}_{var0}(k) = \mathbf{K}_{var}^+(\mathbf{u}_0 - \mathbf{K}_{fix}(k)\mathbf{w}_{fix}(k)) \quad (26)$$

$\mathbf{w}_{fix}$ 는 첫 번째 루프에서는 0으로 시작하며, 매 루프마다 아래 관계식을 통하여 계산된  $\mathbf{w}$ 에서 포화된 프로펠러가 발생하면 업데이트한다.

$$\mathbf{w}(k) = \mathbf{w}_{var\Delta}(k) + \mathbf{w}_{var0}(k) + \mathbf{w}_{fix}(k) \quad (27)$$

식 (23)과 프로펠러 회전수 제한을 고려하여 다음과 같이  $c$ 에 대한 관계식으로 표현할 수 있다.

$$\bar{w}_i - w_{var0,i} \leq c w_{var\Delta,i} \leq \bar{w}_i - w_{var0,i} \quad (28)$$

여기서  $c$ 는 가능한 한 큰 값을 가지는 것이 생성된 조종력과 주어진 조종력과의 차이를 줄어드는 것이므로 다음과 같이  $c$ 를 계산할 수 있다.

$$c_{\max} = \min_{i \in w_{var}} \left( \left[ \frac{\bar{w}_i - w_{var0,i}}{w_{var\Delta,i}^+} \right], \left[ \frac{\bar{w}_i - w_{var0,i}}{w_{var\Delta,i}^-} \right] \right) \quad (29)$$

$$c = \min(1, c_{\max})$$

### 2.3.3 컨벡스 최적화 기반 실시간 최적 조종력 분배 기법

전기수직이착륙 무인기는 여러 개의 프로펠러의 추진력으로 비행체 자세를 제어하는 조종력을 생성

한다. 조종사의 조종 입력을 적절히 분배하여 조종력을 생성하는 조종력 분배 모듈은 모터 컨트롤러의 제어 입력 수신 주기와 동일한 속도로 구동되어야 한다. 따라서 컨벡스 최적화 기법을 조종력 분배 문제에 적용하기 위해서는 최소한 200Hz 속도 이상으로 연산하여야 한다. 이를 만족시키는 DSL(Domain Specific Language)로는 컨벡스 최적화 문제 정의하여 C 코드 생성하는 CVXGEN[24]이 있다. CVXGEN은 상위 언어로 정의된 문제를 이차계획법(quadratic program, QP)으로 변환하여 별도로 요구하는 라이브러리가 없는 C 코드를 생성한다. 물론 CVXGEN은 소규모 문제에 대하여 연산 시간이 2.5 ms 이하가 가능하지만 적용가능한 문제가 QP로 제한된다는 단점도 있다. 그러나 전기 수직 이착륙 무인기의 조종력 분배 문제는 목적 함수를 QP로 정의 가능하고 KKT(Karush-Kuhn-Tucker) 행렬 크기도 소규모이므로 CVXGEN이 충분히 적용가능하다.

최적화 문제에서 목적 함수를 정의하는 것에 있어서 물리적인 의미를 반영하는 것이 매우 중요하다. 조종력 분배의 핵심은 현재의 제한 조건에서 비행제어법칙에서 요구하는 추력 및 제어 모멘트를 생성하는 것이다. 따라서 전기추진 수직이착륙 무인기의 조종력 분배 문제를 아래와 같이 정의하였다.

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} \quad \|\mathbf{G}(\mathbf{K}\mathbf{w} - \mathbf{u})\|_2^2 + \gamma \|\mathbf{w} - \mathbf{w}_p\|_2^2 + \chi \|\mathbf{w}\|_1 \\ \text{s.t.} \quad & \mathbf{w}_{\min} \leq \mathbf{w} \leq \mathbf{w}_{\max} \\ & w_{\text{fault},i} = 0 \end{aligned} \quad (30)$$

여기서  $\mathbf{w}_p$ 는 이전 스텝의 프로펠러 회전수,  $\gamma$ 와  $\chi$ 는 양의 상수이다.  $\mathbf{G} = \text{diag}(g_T, g_L, g_M, g_N)$ 는 가중치 대각 행렬이고,  $w_{\text{fault},i}$ 는 고장난  $i$ 번 모터의 회전수를 의미한다.

목적함수의 첫 번째 항은 조종력  $\mathbf{u}$ 를 비행제어법칙에서 요구하는 수치와 최대한 근사치로 생성하게 하는 항이며, 두 번째 항은 이전의 프로펠러 회전수와 현재의 회전수의 연속성을 보장하기 위한 항이다. 마지막 항은 에너지 효율성보다는 작동기 포화 현상 억제를 고려한 것이다. 본 논문에서는 조종력 할당 문제를 추진 시스템 일부의 고장 상황을 가정하기 때문에 조종력이 매우 제한된 상황이므로, 제어법칙에서 요구하는 값과 근사한 추력 및 제어 모멘트를 생성하는 것이 고려하여, 각 항들의 가중치( $\gamma, \chi$ )를 조정하여야 한다.

가중치 대각 행렬  $\mathbf{G}$ 는 각 축에 대한 우선순위를 반영하기 위한 변수이며 고장으로 인하여 충분한 조종력이 확보되지 않는 경우에는 비행체 안전에 관하여  $\mathbf{u}$ 의 구성 성분에 대한 우선순위가 발생한다. 롤 및 피치 축 모멘트는 당장의 추락을 막기 위한 최소한의 비행체 자세 유지에 필수적이므로 가장 우선순위가 된다. 요 축은 상대적으로 롤 및 피치 축에 비

하여 발산 속도가 느리고, 비행체 추락 상황에서 자세가 유지된 채 추락할 경우에는 상대적으로 충격 및 파손이 덜하므로 우선순위가 뒤쳐진다. 추력은 비행에 있어서 근본적인 요소지만 제한된 구동력 환경에서는 자세 유지에 비하여 중요성이 떨어지게 된다. 따라서 추력 레벨을 감소시켜 롤, 피치, 요 축 모멘트의 ACS의 넓이가 증가하는 것을 도모할 수 있다. 추력 가중치는 상대적으로 후순위가 되며, 추력 절대값이 큰 만큼 그 크기 또한 감소하여야 한다.

가중치 행렬  $\mathbf{G}$ 는 컨벡스 최적화 기법 기반과 타 기법간의 차이점을 발생하는 요소이다. 의사역변환 기반 조종력 분배 알고리즘들에서는 가용 조종력 범위를 초과하는 경우는 고려되지 않는 경우가 많아 비행체가 자세 유지를 하지 못해 추락하는 결과가 많이 발생한다. 또한 작동기의 포화를 고려한 알고리즘에서도 유사한 결과가 발생한다. 반면에 위의 목적 함수를 채용한 컨벡스 최적화 기법 기반 실시간 최적 조종력 분배기는 이와 같은 최악의 상황에서도 최소한의 안전을 위한 대처가 가능하다는 장점이 있다.

위에서 정의한 최적 조종력 분배 문제를 표준 형식(standard form)으로 변형하면 다음과 같이 표현된다.

$$\begin{aligned} & \underset{\mathbf{w}, \mathbf{t}_1, \mathbf{t}_2}{\text{minimize}} \quad \frac{1}{2} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{w} \end{bmatrix}^T \begin{bmatrix} 2\mathbf{I} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2\gamma\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{w} \end{bmatrix} + \begin{bmatrix} 0 \\ \chi \\ -2\gamma\mathbf{w}_p \end{bmatrix}^T \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{w} \end{bmatrix} + \gamma \mathbf{w}_p^T \mathbf{w}_p \\ \text{s.t.} \quad & \begin{bmatrix} -\mathbf{I} & 0 & \mathbf{G}\mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{G}\mathbf{u} \end{bmatrix} \\ & \begin{bmatrix} 0 & -\mathbf{I} & \mathbf{I} \\ 0 & -\mathbf{I} & -\mathbf{I} \\ 0 & 0 & -\mathbf{I} \\ 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{w} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ -\mathbf{w}_{\min} \\ \mathbf{w}_{\max} \end{bmatrix} \end{aligned} \quad (31)$$

여기서  $\mathbf{t}_1 \in \mathbb{R}^4$ 과  $\mathbf{t}_2 \in \mathbb{R}^8$ 는 각각 조종력과 프로펠러 회전수에 대한 최적화 문제를 위한 여유 변수(Slack Variables)이다. 위의 최적화 문제에서 고장에 대한 영향성은 부등식 조건에서 프로펠러 회전수의 최대 최소값을 0으로 선정하여 반영하였다.

$$w_{i,\min} = w_{i,\max} = 0 \quad (32)$$

위의 문제와 같은 표준 이차 문제(Standard Quadratic Problem)를 CVXGEN에서는 primal-dual algorithm 등으로 대표되는 interior point method에 Mehrotra correction 기법을 채용하여 해를 구한다. 연산 복잡도를 나타내는 KKT 행렬의 크기는  $88 \times 88$ 이며, solver 상태변수의 크기가 20이 된다. 이러한 문제 크기는 상당히 작은 문제가 되므로 충분한 연산 속도를 기대할 수 있다.

## 2.4 조종력 분배 기법 구현 및 성능 비교

조종력 분배 기법들의 목적은 결국 현재 상황에서

제어 기법에서 요구하는 조종력을 생성하는 것이다. 따라서 기법들의 성능은 얼마나 빨리 해를 계산하는지, 얼마나 넓은 범위의 조종력을 다룰 수 있는지로 판단할 수 있다. 연산 속도는 정확한 수치보다는 주어진 시간 안에 해를 계산하는지를 따지는 성능 지표이며, 반면에 가용 조종력 범위는 넓으면 넓을수록 좋은 성능 지표이다. 따라서 본 논문에서는 각 기법들의 가용 조종력 범위를 서로 비교하여 성능을 분석하고, 조종력 생성 시뮬레이션을 통해 검증하였다.

정지 비행 조건에서 정상 상태와 추진시스템 고장 상태에서의 롤-피치 ACS 넓이 기준으로 각 기법들의 성능을 비교하였다. Fig. 4와 같이 1번 모터가 고장난 상황에서의 L-M ACS에서는 각 기법들이 분배 가능한 영역이 상당히 차이가 발생하는 것으로 보인다. 정록색을 띤 영역은 비행체 고유의 ACS이며, 컨벡스 최적화 기법과 동일하다. ACS의 내접한 원의 반경 ( $R_{\text{fault}}$ )을 계산하면 각각 3.27(PI), 4.40(DPI), 5.71Nm이다. DPI가 사용 가능한 영역이 PI보다 상당히 넓고, 이는 포화된 작동기의 조종력을 다른 작동기에 재분배하는 루프가 동작하는 DPI의 강점으로 보인다. DPI 기법이 PI 기법보다는 넓은 영역이 활용 가능하지만, 비행체 고유의 ACS와는 매우 큰 차이가 보인다.

본 논문에서 제안하는 컨벡스 최적화 기반 실시간 최적화 기법은 이론적으로 비행체 고유의 ACS를 모두 활용할 수 있다. 이러한 장점에도 불구하고 실제로 최적화 기법을 조종력 분배 문제에 적용하지 못했던 점은 바로 연산 속도와 임베디드에서 구동 가능한 솔버가 부족하다는 점이었다. 그러나 컨벡스 함수를 이용하면서 연산 속도가 점차적으로 빨라지고, 문제를 단순화하여 적용하는 노력이 더해지면서 실시간 최적화 연구 사례가 발표되기 시작하였다.

기법들의 연산량은 가용할 수 있는 ACS 영역 넓이와 비례하는 경향이 있다. PI 기법이 연산량이 가

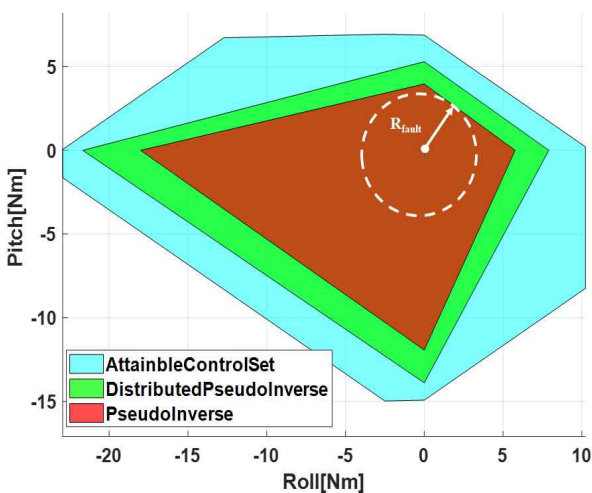


Fig. 4. ACS comparison for 1st motor failure

장 적으며, DPI는 최대 연산량이 대략 PI 기법의 약 4배에 달한다. DPI 기법까지는 믹서에서 요구하는 연산속도를 만족시키고 성공적인 적용 사례도 많이 발표되었다. 반면에 최적화 기법은 타 기법과 상대적인 비교가 무의미할 정도로 연산량이 많았다. 그러나 임베디드 기술이 발전하면서 소형 비행제어컴퓨터 (Flight Control Computer, FCC)의 성능이 점차적으로 좋아지고, C 언어 기반의 컨벡스 최적화 솔버인 CVXGEN이 발표되면서 실제적인 적용이 가능하게 되었다. 본 논문에서는 조종력 분배 문제를 단순화 및 일반화하여 연산 속도를 높였으며, CVXGEN 솔버를 통하여 실시간 조종력 분배 소프트웨어를 임베디드 보드에 구동되는 형태로 구현하였다.

각 기법들을 MATLAB/SIMULINK 환경에서 구현하였으며, 컨벡스 최적화 기법은 C언어 기반의 솔버를 "S-Function Builder"를 사용하여 구현하였다. 1번 모터가 고장난 상황에서 요구되는 조종력 대비 실제 생성하는 조종력을 비교한 결과가 Fig. 5이다. 요구하는 롤 및 피치 모멘트 크기가 커질수록 ACS의 가장 자리에 근접하게 되면서 조종력 생성 오차가 발생한다. 상대적으로 PI, DPI, 컨벡스 기법 순으로 근접한 모멘트를 생성한다. 이와 더불어 각 기법들의 최대 연산속도를 측정하였다. PI와 DPI 기법은 MATLAB의 tic/toc 함수로 측정하였으며, 컨벡스 최적화 기법은 C 코드 상에서 chrono 라이브러리를 사용하였다. 구동한 컴퓨터는 Intel i9-9900K CPU를 장착하고 있으며, PI 기법의 최대 연산 시간은  $3.6 \times 10^{-6}$ , DPI 기법은  $2.39 \times 10^{-5}$ , 컨벡스 최적화 기법은  $3.74 \times 10^{-5}$ 초이다. 컨벡스 기법은 C 언어로 구동하였으나 연산 속도가 상대적으로 느리지만, 비행체 안정화를 위한 조종력 할당 루프에는 충분한 연산 속도이다.

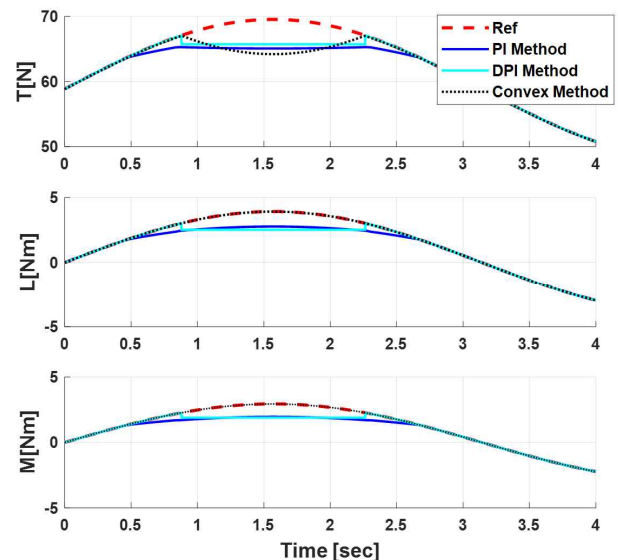


Fig. 5. Error of control force and moments when 1st motor failed



### 2.5 모터 고장 대처를 위한 6자유도 조종력 할당 시뮬레이션

앞 절에서는 3가지 조종력 할당 알고리즘을 구현하여 모터 고장 상황에서 요구되는 조종력 대비 생성되는 조종력을 비교하였다. 본 절에서는 테스트베드 비행체의 6자유도 모델에 조종력 할당 알고리즘을 적용하여 그 성능을 시뮬레이션을 통해 분석한 내용을 서술한다. 시뮬레이션은 MATLAB/SIMULINK를 기반으로 수행하였으며, 샘플링 시간(Sampling Time)은 오픈 소스 플랫폼(Pixhawk, Ardupilot)을 참고하여 2.5ms로 설정하였다. 또한 비행 제어 법칙은 일반적인 형태인 내·외부 루프로 위치 및 속도/자세 제어기 구조로 설계하였다.

시뮬레이션은 비행체가 원점을 중심으로 원형 기동을 하고, 1번 모터를 인위적으로 고장난 상황으로 묘사하여 각 알고리즘의 반응성을 확인하였다. 고장 주입은 시뮬레이션 시간으로 5초에 1번 모터에 주입하였으며, 고장 진단은 고장 발생 후 1초 뒤에 인식된다고 가정하였다. 1번 모터는 모멘텀암이 가장 긴 모터 중에 하나이므로 고장 주입 모터로 선정하였다. 고장을 인식한 후의 비행체의 위치, 자세 반응을 관찰하였으며, 생성 조종력의 오차와 각 모터의 입력 신호를 통해 각 알고리즘의 성능을 분석할 수 있었다.

Figure 6은 비행체의 원형 기동에 대한 위치 궤적이며, 고장 발생 지점을 확대한 그래프에서는 PI, DPI, 컨벡스 기법 순서로 기준 궤적에 가까운 것을 확인할 수 있다. 고장 인식한 후에 충분한 시간이 지나면, 3개의 알고리즘 모두 기준 궤적을 추종 할 수 있으나 급격한 기동이 필요한 고장 인식 시점에서는 차이가 발생하는 것을 볼 수 있다. 자세 반응도 비행체 궤적과 유사하게 컨벡스 기법이 가장 작은 오차가 발생하는 것을 Fig. 7에서 확인할 수 있다. Fig. 8과 Fig. 9는 조종력 할당 알고리즘에서 생성하는 조종력과 비행제어법칙에서 요구하는 조종과의 오차를 표시한 것이다. Fig. 8과 Fig. 9의 y축은 각각 롤과 피치 모멘트 오차를 의미하며, 오른쪽 그래프는 고장 인식 순간을 확대한 그래프이다. 두 그래프에서 모두 확연하게 컨벡스 최적화 기법과 타 기법들의 성능 차이를 확인할 수 있다. PI와 DPI는 알고리즘은 상당히 의미있는 크기의 모멘트 오차(약 5.5Nm, 2.5Nm)가

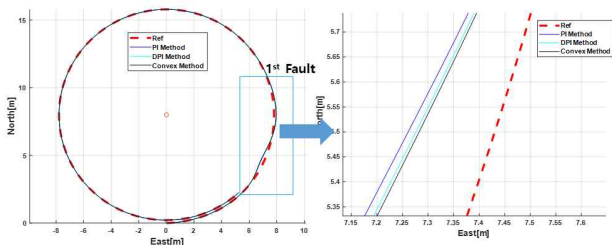


Fig. 6. Flight trajectory for 1<sup>st</sup> motor failure during circle maneuver

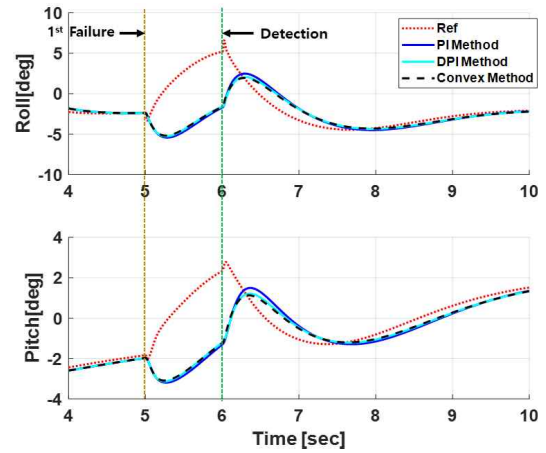


Fig. 7. Attitude response for 1<sup>st</sup> motor failure during circle maneuver

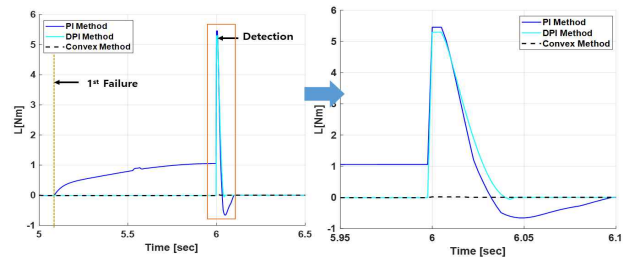


Fig. 8. Roll moment error for the three control allocation algorithms

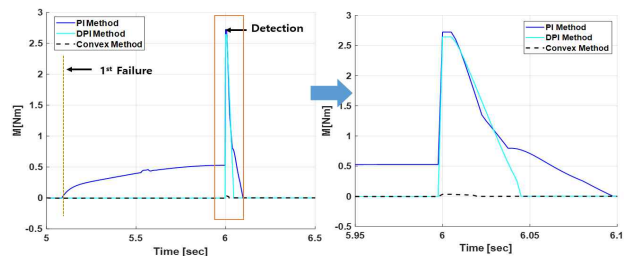


Fig. 9. Pitch moment error for the three control allocation algorithms

발생하며, 요구되는 조종력의 크기가 작아지면서 오차가 줄어들게 된다. 반면에 컨벡스 알고리즘은 고장 인식된 초기에 다른 알고리즘에 비해 큰 조종력을 발생시킬 수 있으므로 조종력 오차가 매우 작은 것을 확인할 수 있다.

Figure 10은 0 - 1 사이의 값으로 스케일된 각 모터 출력 그래프이다. 앞 절에서 언급한 바와 같이 이전 출력과 현재 출력과의 연속성을 고려한 항을 비용함수(cost function)에 추가하여 비연속적인 신호 생성 현상이 보이지 않는 것을 확인할 수 있다. 컨벡스 기법은 특히나 롤 축 균형을 위하여 1번 모터와 반대편에 배치된 4번, 7번 모터 명령이 다른 기법과는 달리 감소하는 구간을 발생시킨다. 이 구간이 롤축 모멘트 생성 오차가 타 기법과 확연히 작은 원인으로 보인다.

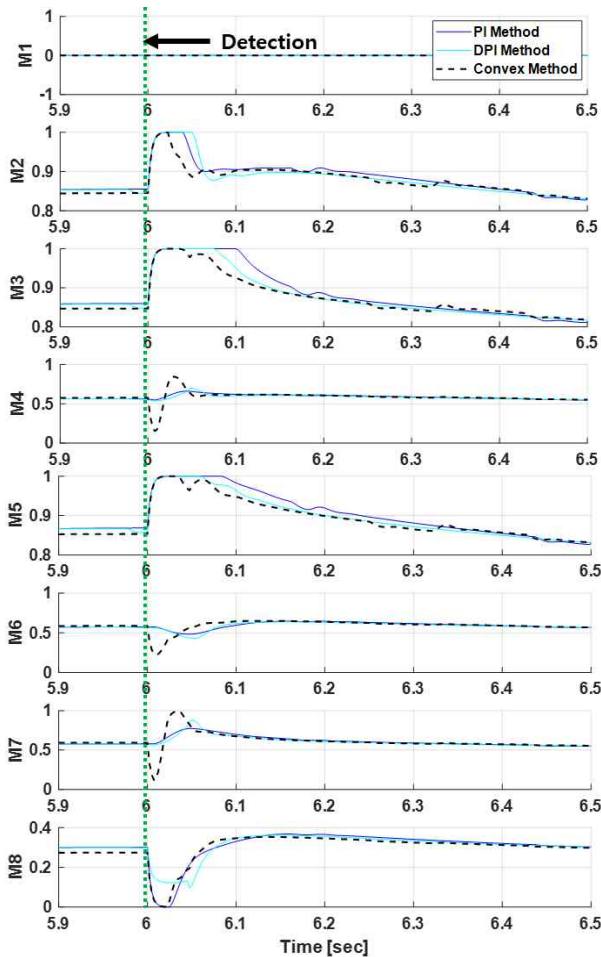


Fig. 10. Scaled motor outputs for the three control allocation algorithms

### III. 결 론

본 논문에서는 전기 추진 수직 이착륙 무인기의 추진시스템 고장에 대처하기 위하여 조종력 할당 알고리즘 설계 및 성능 분석에 대해 서술하였다. 제한된 임무 중량으로 인하여 경량의 비행제어시스템을 탑재해야하는 소형 무인항공기는 추진시스템 고장에 대처하기 위하여 주로 연산량이 작은 의사역변환행렬을 기반으로한 기법을 주로 적용하였다. 이러한 기법들은 연산량이 작고 구현이 쉬운 장점이 있으나 비행체 고유의 ACS를 모두 다 사용하지 못하고 제한된 범위만 사용한다. 반면에 컨벡스 함수의 특성을 이용한 최적화 기법은 물리적인 ACS를 모두 사용할 수 있어 고장 상황에서 비행제어법칙의 추종 성능이 향상된다. 본 논문에서는 기존에 주로 사용되던 PI 및 DPI 기법과 제안한 컨벡스 최적화 기법의 성능을 ACS 분석과 6자유도 시뮬레이션을 통하여 비교 검증하였다.

### 후 기

본 연구는 한국항공우주연구원 항공우주기반연구사업의 일환으로 “전기추진 수직이착륙 미래비행체 핵심기술 연구” 과제 지원으로 수행하였습니다.

### References

- 1) Whittle, R., “Air Mobility Bonanza Beckons Electric VTOL Developers,” *Vertiflite*, March-April 2017, pp. 14~21.
- 2) Marks, A., Whidborne, J. F. and Yamamoto, I., “Control Allocation for Fault Tolerant Control of a VTOL Octorotor,” *UKACC International Conference on Control*, Cardiff, September 3-5, 2012, pp. 357~362.
- 3) Alwi, H. and Edwards, C., “Fault Tolerant Control of an Octorotor Using LPV based Sliding Mode Control Allocation,” *American Control Conference (ACC)*, Washington, DC, USA, June 17-19, 2013, pp. 6505~6510.
- 4) Merheb, A., Nourra, H. and Bateman, F., “Active Fault Tolerant Control of Octorotor UAV using Dynamic Control Allocation,” *International Conference on Intelligent Unmanned Systems*, Montreal, Quebec, Canada, September 29 - October 1, 2014.
- 5) Du, G., Quan, Q. and Cai, K., “Controllability Analysis and Degraded Control for a Class of Hexacopters Subject to Rotor Failures,” *Journal of Intelligent Robotic Systems*, September 2014.
- 6) Horn, R. A. and Johnson, C. R., *Matrix Analysis*, Cambridge University Press, 1985.
- 7) Oppenheimer, M. W., Doman, D. B. and Sighthorsson, D. O., “Dynamics and Control of a Hovering Biometric Vehicle using Biased Wingbeat Forcing Function,” *Journal of Guidance, Navigation and Control*, Vol. 34, 2011, pp. 204~217.
- 8) Shi, J., Zhang, W. and Liu, X., “Research on Allocation Efficiency of the Redistributed Pseudo Inverse Algorithm,” *Science China Information Sciences*, Vol. 53, 2010, pp. 271~277.
- 9) Frangenberg, M., Stephan, J. and Fichter, W., “Fast Actuator Fault Detection and Reconfiguration for Multicopters,” *AIAA Guidance, Navigation, and Control Conference*, Florida, USA, January, 2015.
- 10) Schneider, T., Ducard, G., Rudin, K. and Strupler, P., “Fault-tolerant Control Allocation for Mulirotor Helicopters using Parametric Programm-

ing," International Micro Air Vehicle Conference and Flight Competition, Braunschweig, Germany, July, 2012.

11) Boyd, S. and Vandenberghe, L., "Convex optimization," *Cambridge University Press*, Cambridge, 2004.

12) Nesterov, Y. and Nemirovskii, A., "Interior point polynomial algorithms in convex programming," *SIAM*, Vol. 13, Philadelphia, 1994.

13) Ye, Y., *Interior point algorithms: theory and analysis*, Wiley, New York, 1997.

14) Nocedal, J. and Wright, S. J., *Numerical optimization* Springer, Berlin, 1999.

15) Grant, M. and Boyd, S., "CVX: Matlab software for disciplined convex programming (web page and software)," <http://www.stanford.edu/~boyd/cvx/>, July 2008.

16) Löfberg, J., "YALMIP: a toolbox for modeling and optimization in MATLAB," *In Proceedings of the CACSD conference*, Taipei, Taiwan, <http://control.ee.ethz.ch/~joloef/yalmip.php>, 2004.

17) Chu, E., Parikh, N., Domahidi, A. and Boyd, S., "Code generation for embedded second-order cone programming," *Proceedings of the European Control Conference*, 2013, pp. 1547~1552,

18) O'Donoghue, B., Chu, E., Parikh, N. and

24) Mattingley, J. and Boyd, S., "CVXGEN: a Code Generator for Embedded Convex Optimization," *Optimization and Engineering*, Vol. 13, No. 1, 2012, pp. 1~27.

Boyd, S., "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, 2016, pp. 1~27.

19) Udell, M., Mohan, K., Zeng, D., Hong, J., Diamond, S. and Boyd, S., "Convex optimization in Julia," *In Proceedings of the Workshop for High Performance Technical Computing in Dynamic Languages*, 2014, pp. 18~28.

20) Diamond, S. and Boyd, S., "CVXPY: A Python-Embedded Modeling Language for Convex Optimization," *Journal of Machine Learning Research*, Vol. 17, 2016, pp. 1~5.

21) Dyurham, W. C., "Constrained Control Allocation," *Journal Guidance, Control and Dynamics*, Vol. 16, 1993, pp. 717~725.

22) Petersen, J. A. M. and Bodson, M., "Constrained Quadratic Programming Techniques for Control Allocation," *IEEE Transactions on Control Systems Technology*, Vol. 14, pp. 91~98.

23) Matt, J., "Analyze N-dimensional Polyhedra in terms of Vertices or (In)Equalities," *MATLAB-CENTRAL*