



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0004915
(43) 공개일자 2015년01월13일

(51) 국제특허분류(Int. Cl.)
G05D 1/10 (2006.01)
(21) 출원번호 10-2014-7033799
(22) 출원일자(국제) 2013년04월30일
심사청구일자 없음
(85) 번역문제출일자 2014년12월01일
(86) 국제출원번호 PCT/US2013/038769
(87) 국제공개번호 WO 2014/018147
국제공개일자 2014년01월30일
(30) 우선권주장
61/640,249 2012년04월30일 미국(US)

(71) 출원인
더 트러스티스 오브 더 유니버시티 오브 펜실바니아
미국 19104 펜실베니아주 필라델피아 슈트 200 케스트너트스트리트 3160
(72) 발명자
쿠마르 비제이
미국 19104 펜실베니아주 필라델피아 타운 109 사우스 33알디 스트리트 220
쿠칠레예프 알렉산드르
미국 19146 펜실베니아주 필라델피아 룸 201 빌딩 192 그레이스 페리 에비뉴 3401
멜링어 다니엘
미국 19146 펜실베니아주 필라델피아 룸 201 빌딩 192 그레이스 페리 에비뉴 3401

(74) 대리인
장수길, 양영준, 안국찬

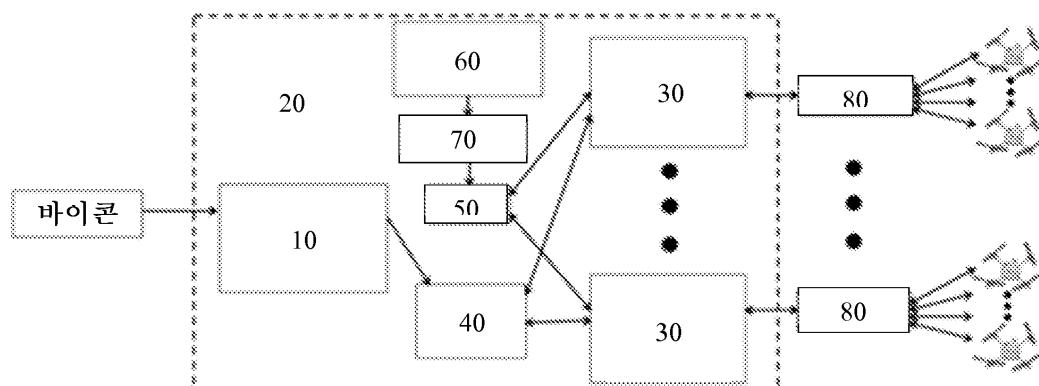
전체 청구항 수 : 총 35 항

(54) 발명의 명칭 큐드로터 팀의 3차원적 조정

(57) 요약

본 발명은 목표 위치를 향하여 적어도 2개의 비행체의 비행 궤적을 제어하는 시스템 및 방법에 관한 것이다. 이 시스템은 적어도 2개의 비행체와, 적어도 2개의 비행체의 배향, 각속도, 위치 및 선형 속도를 확인하고 갱신하기 위한 온보드 관성 측정 유닛, 적어도 2개의 비행체 각각의 현재 위치와 속도를 감지하기 위한 모션 캡쳐 시스템, 및 모션 캡쳐 시스템 및 복수개의 비행체들과 통신하는 베이스 컨트롤러를 포함한다. 베이스 컨트롤러는 미리 정해진 시간 간격으로 구간별 매끄러운 다항 함수를 이용하여 최적 궤도 경로를 계산한다. 베이스 컨트롤러는 또한 계산된 최적 궤도 경로를 기반으로 각각의 비행체에게 그들의 상태(state)를 개별적으로 제어하도록 명령을 보내어, 적어도 2개의 비행체가 충돌을 피하면서 계산된 최적 궤도 경로를 따르게 한다.

대 표 도 - 도7



특허청구의 범위

청구항 1

목표 위치 및 배향을 향하여 적어도 2개의 비행체의 상태를 제어하기 위한 궤적 생성 방법이며,
 비행체의 배향 및 각속도를 판단하는 단계와,
 비행체의 적어도 하나의 모터를 제어하여 비행체의 배향 및 각속도를 제어하는 단계와,
 각각의 비행체의 현재 위치 및 속도를 판단하는 단계와,
 원하는 배향 및 각속도와 적어도 하나의 모터로부터 요구되는 순 추력을 지정하여 각각의 비행체의 위치 및 속도를 제어하는 단계와,
 각각의 비행체에 대해, 구간별 매끄러운 다항 함수를 사용하고 가중치를 적용하고 중첩 제약을 실행하여 최적 궤적 경로를 미리 정한 시간 간격들로 계산하는 단계와,
 계산된 최적 궤적 경로에 기초하여, 각각의 비행체에 명령을 보내 비행체들의 상태를 개별적으로 제어하여 비행체들이 충돌을 피하면서 계산된 최적 궤적 경로를 따르도록 하는 단계와,
 각각의 비행체의 현재 위치 및 속도를 갱신하는 단계를 포함하는
 궤적 생성 방법.

청구항 2

제1항에 있어서, 비행체의 각각의 상태는 비행체의 배향 및 각속도와 위치 및 선형 속도를 포함하는
 궤적 생성 방법.

청구항 3

제1항에 있어서, 배향 여러가 예측되고 각각의 비행체 상에서 배향이 제어되는
 궤적 생성 방법.

청구항 4

제1항에 있어서, 적어도 2개의 비행체에 각각 적용된 가중치는 서로 상이한
 궤적 생성 방법.

청구항 5

제1항에 있어서, 상기 궤적 생성 방법은 상기 복수의 미리 정한 시간 간격들로 반복적으로 실행되는
 궤적 생성 방법.

청구항 6

제1항에 있어서, 정수 제약을 사용하여 장애물과의 충돌 제약을 실행하고 상기 적어도 2개의 비행체에 최적으로 목표 위치를 할당하는 단계를 더 포함하는

궤적 생성 방법.

청구항 7

제1항에 있어서, 각각의 비행체에 대해 최적 궤적 경로를 계산하는 단계는 지정된 시간(t_w)에 n_w 개의 원하는 경유점을 통해 매끈하게 트랜지션하는 궤적을 생성하면서, 다음의 식에 따라 n_q 개의 쿼드로터에 대해 위치의 k_r 차 도함수 제곱값의 적분을 최소화하고,

$$\begin{aligned} & \min \sum_{q=1}^{n_q} \int_{t_0}^{t_{n_q}} \left\| \frac{d^k r_{Tq}}{dt^k} \right\|^2 dt \\ \text{s.t. } & \mathbf{r}_{Tq}(t_w) = \mathbf{r}_{wq}, \quad w = 0, \dots, n_w; \forall q \\ & \frac{d^k x_{Tq}}{dt^k}|_{t=t_w} = 0 \text{ or free, } w = 0, n_w; j = 1, \dots, k_x; \forall q \\ & \frac{d^k y_{Tq}}{dt^k}|_{t=t_w} = 0 \text{ or free, } w = 0, n_w; j = 1, \dots, k_y; \forall q \\ & \frac{d^k z_{Tq}}{dt^k}|_{t=t_w} = 0 \text{ or free, } w = 0, n_w; j = 1, \dots, k_z; \forall q \end{aligned}$$

여기서 $\mathbf{r}_{Tq} = [x_{Tq}, y_{Tq}, z_{Tq}]$ 는 큐드로터(q)의 궤적을 나타내고, \mathbf{r}_{wq} 는 큐드로터(q)의 원하는 경유점을 나타내는 궤적 생성 방법.

청구항 8

제1항에 있어서, 비행체를 직사각형 프리즘으로서 모델링함으로써 적어도 2개의 비행체들 사이의 충돌 회피를 제공하는 단계를 더 포함하고, 직사각형 프리즘은 비행체가 프리즘 내에 머무르고 임의의 각도로 롤, 퍼치, 및 요 운동하기에 충분한 길이인 변 길이(l_x , l_y 및 l_z)를 갖는 전역 프레임에 의해 배향되는

궤적 생성 방법.

청구항 9

제8항에 있어서, n_o 개의 볼록한 장애물을 갖는 환경을 통해 프리즘을 운항하는 단계를 더 포함하고, 각각의 볼록한 장애물(o)은 $n_f(o)$ 개의 면을 가진 구조 공간 내의 볼록 영역으로 표현되고, 각각의 면(f)에 대해 시간 t_k 에서의 비행체의 원하는 위치, $\mathbf{r}_{Tq}(t_k)$ 가 장애물(o)의 외부에 있는 조건은 다음과 같이 표시되며,

$$n_{of} \cdot \mathbf{r}_{Tq}(t_k) \leq s_{of}$$

여기서, n_{of} 는 구조 공간 내의 장애물(o)의 면(f)에 대한 법선 벡터이고, s_{of} 는 스칼라이고, 시간 t_k 에서의 비행체 위치에 대한 식이 면들 중 적어도 하나에 대해 만족되면, 직사각형 프리즘, 그리고 그에 따라 비행 기계가 장애물과 충돌하지 않는

궤적 생성 방법.

청구항 10

제9항에 있어서, 비행 기계(q)가 시간 t_k 에서 장애물(o)과 충돌하지 않는 조건은 다음과 같이 2차 변수(b_{qofk})에 의해 실시되고,

$$\begin{aligned} n_{of} \cdot \mathbf{r}_{Tq}(t_k) &\leq s_{of} + M b_{qofk} \quad \forall f = 1, \dots, n_f(o) \\ b_{qofk} &= 0 \text{ or } 1 \quad \forall f = 1, \dots, n_f(o) \\ \sum_{f=1}^{n_f(o)} b_{qofk} &\leq n_f(o) - 1 \end{aligned}$$

여기서 M 은 큰 양수인

궤적 생성 방법.

청구항 11

제10항에 있어서, 제10항의 식은 경유점을 사이의 n_k 개의 중간 시간 단계들에서 모든 장애물에서 n_q 개의 비행 기계 전부에 대해 제7항의 식에 도입되는

궤적 생성 방법.

청구항 12

제7항에 있어서, 다음의 제약 세트에 의해 비행체 1 및 2에 대해 수학적으로 나타낼 수 있는 경유점을 사이의 n_k 개의 중간 시간 단계들에서 제약을 실행함으로써 비행체의 궤적 경로 상의 경유점을 사이에서 트랜지션할 때 비행체들은 서로 안전 거리에 유지되고,

$$\begin{aligned} \forall t_k : & \quad x_{T_1}(t_k) - x_{T_2}(t_k) \leq d_{x12} \\ \text{or} \quad & x_{T_2}(t_k) - x_{T_1}(t_k) \leq d_{x21} \\ \text{or} \quad & y_{T_1}(t_k) - y_{T_2}(t_k) \leq d_{y12} \\ \text{or} \quad & y_{T_2}(t_k) - y_{T_1}(t_k) \leq d_{y21} \end{aligned}$$

여기서 d는 비행체 1과 2 사이의 안전 거리를 나타내는

궤적 생성 방법.

청구항 13

제12항에 있어서, 상기 비행체들은 축방향으로 대칭이고 $d_{x12}=d_{x21}=d_{y12}=d_{y21}$ 인

궤적 생성 방법.

청구항 14

제6항에 있어서, 상기 정수 제약들은 각각의 큐드로터(q) 및 목표(g)에 다음의 정수 제약들을 적용함으로써 비행체들의 최적 목표 할당을 찾는데 사용되고,

$$\begin{aligned} x_{T_q}(t_{n_w}) &\leq x_g + M\beta_{qg} \\ x_{T_q}(t_{n_w}) &\geq x_g - M\beta_{qg} \\ y_{T_q}(t_{n_w}) &\leq y_g + M\beta_{qg} \\ y_{T_q}(t_{n_w}) &\geq y_g - M\beta_{qg} \\ z_{T_q}(t_{n_w}) &\leq z_g + M\beta_{qg} \\ z_{T_q}(t_{n_w}) &\geq z_g - M\beta_{qg} \end{aligned}$$

여기서 β_{qg} 는 최적 목표 할당하는데 사용되는 2차 변수인

궤적 생성 방법.

청구항 15

제14항에 있어서, 적어도 n_g 개의 큐드로터가 원하는 목표 위치에 도달하는 것을 보장하도록 다음의 제약을 적용하는 단계를 더 포함하는

$$\sum_{q=1}^{n_g} \sum_{g=1}^{n_g} \beta_{qg} \leq n_g n_q - n_g$$

궤적 생성 방법.

청구항 16

제1항에 있어서, 복수의 그룹으로 비행체들을 조작하는 단계-복수의 그룹은 각각 독립적으로 조정됨-, 및

복수의 그룹 각각에 대한 궤적을 생성하여 목표 위치들을 집단화하는 단계를 도 포함하는

궤적 생성 방법.

청구항 17

제16항에 있어서, 비행체의 환경은 nr개의 볼록한 서브 구역으로 구획되고, 여기서 각각의 서브 구역은 동일한 수의 비행체 출발 및 목표 위치를 포함하고, 각각의 서브 구역 내부에서 비행체들에 대해 별개의 궤적들이 생성되어, 비행체들이 비행체들의 위치에 대한 선형 제약을 사용하여 각자의 서브 구역 내부에 머무르도록 요구되는

궤적 생성 방법.

청구항 18

목표 위치를 향하여 적어도 2개의 비행체의 비행 궤적을 제어하기 위한 시스템이며,

적어도 2개의 비행체의 배향, 각속도, 위치 및 선형 속도를 판단하고 갱신하기 위한 온-보드(on-board) 관성 측

정 유닛을 갖는 적어도 2개의 비행체와,

적어도 2개의 각각의 비행체의 현재 위치 및 속도를 탐지하는 모션 캡쳐 시스템과,

모션 캡쳐 시스템과 통신하고 복수의 비행체와 통신하는 베이스 컨트롤러를 포함하고,

상기 베이스 컨트롤러는 각각의 비행체에 대해, 구간별 매끄러운 다향 함수를 사용하고 가중치를 적용하고 중첩 제약을 실행하여 최적 궤적 경로를 미리 정한 시간 간격들로 계산하고,

계산된 최적 궤적 경로에 기초하여, 각각의 비행체에 명령을 보내 비행체들의 상태를 개별적으로 제어하여, 상기 적어도 2개의 비행체가 충돌을 피하면서 계산된 최적 궤적 경로를 따르도록 하는

시스템.

청구항 19

제18항에 있어서, 적어도 2개의 비행체는 동종(homogenous)의 운행체인

시스템.

청구항 20

제18항에 있어서, 적어도 2개의 비행체는 이종의(heterogeneous)의 운행체인

시스템.

청구항 21

제18항에 있어서, 비행체의 각각의 상태는 비행체의 배향과 각속도와 위치 및 선형 속도를 포함하는

시스템.

청구항 22

제18항에 있어서, 각각의 비행체에 온 보드 마이크로컨트롤러를 더 포함하고, 마이크로컨트롤러는 배향 에러를 예측하고 관련된 비행체의 배향을 제어하는

시스템.

청구항 23

제20항에 있어서, 적어도 2개의 비행체에 각각 적용된 가중치는 서로 상이한

시스템.

청구항 24

제18항에 있어서, 상기 베이스 컨트롤러는 정수 제약을 사용하여 장애물 및 다른 운행체와의 충돌 제약을 실행하고 상기 적어도 2개의 비행체에 최적으로 목표 위치를 할당하는

시스템.

청구항 25

제18항에 있어서, 상기 베이스 컨트롤러는 지정된 시간 t_w 에서 n_w 개의 원하는 경유점을 통해 매끈하게 트랜지션되는 궤적을 생성하면서, 다음의 식에 따라 n_q 개의 큐드로터에 대한 위치의 k_r 차 도함수 제곱값의 적분을 최소화 함으로써 각각의 비행체에 대해 최적 궤적 경로를 계산하고,

$$\begin{aligned} & \min \sum_{q=1}^{n_q} \int_{t_0}^{t_{n_w}} \left\| \frac{d^{k_r} r_{Tq}}{dt^{k_r}} \right\|^2 dt \\ \text{s.t. } & r_{Tq}(t_w) = r_{wg}, \quad w = 0, \dots, n_w; \forall q \\ & \frac{d^j r_{Tq}}{dt^j}|_{t=t_w} = 0 \text{ or free}, \quad w = 0, n_w; j = 1, \dots, k_r; \forall q \\ & \frac{d^j r_{Tq}}{dt^j}|_{t=t_w} = 0 \text{ or free}, \quad w = 0, n_w; j = 1, \dots, k_r; \forall q \\ & \frac{d^j r_{Tq}}{dt^j}|_{t=t_w} = 0 \text{ or free}, \quad w = 0, n_w; j = 1, \dots, k_r; \forall q \end{aligned}$$

여기서 $\mathbf{r}_{Tq} = [x_{Tq}, y_{Tq}, z_{Tq}]$ 는 큐드로터(q)에 대한 궤적을 나타내고, \mathbf{r}_{wq} 는 큐드로터(q)의 원하는 경유점을 나타낸다.

시스템.

청구항 26

제18항에 있어서, 상기 베이스 컨트롤러는 비행체를 직사각형 프리즘으로서 모델링함으로써 적어도 2개의 비행체들 사이의 충돌 회피를 더 제공하고, 직사각형 프리즘은 비행체가 프리즘 내에 머무르고 임의의 각도로 롤, 페치, 및 요 운동하기에 충분한 길이인 변 길이(l_x , l_y 및 l_z)를 갖는 전역 프레임에 의해 배향되는 시스템.

청구항 27

제26항에 있어서, 상기 베이스 컨트롤러는 n_o 개의 불록한 장애물을 갖는 환경을 통해 프리즘을 운항하고, 각각의 불록한 장애물(o)은 $n_f(o)$ 개의 면을 가진 구조 공간 내의 불록 영역으로 표현되고, 각각의 면(f)에 대해 시간 t_k 에서의 비행체의 원하는 위치, $\mathbf{r}_{Tq}(t_k)$ 가 장애물(o)의 외부에 있는 조건은 다음과 같이 표시되며,

$$\mathbf{n}_{of} \cdot \mathbf{r}_{Tq}(t_k) \leq s_{of}$$

여기서, \mathbf{n}_{of} 는 구조 공간 내의 장애물(o)의 면(f)에 대한 법선 벡터이고, s_{of} 는 스칼라이고, 시간 t_k 에서의 비행체 위치에 대한 식이 면들 중 적어도 하나에 대해 만족되면, 직사각형 프리즘, 그리고 그에 따라 비행 기계가 장애물과 충돌하지 않는

시스템.

청구항 28

제27항에 있어서, 베이스 컨트롤러는 비행 기계(q)가 시간 t_k 에서 장애물(o)과 충돌하지 않는 조건을 다음과 같이 2차 변수(b_{qofk})에 의해 실행하고,

$$\begin{aligned} n_{of} \cdot \mathbf{r}_{Tq}(t_k) &\leq s_{of} + M b_{qofk} \quad \forall f = 1, \dots, n_f(o) \\ b_{qofk} &= 0 \text{ or } 1 \quad \forall f = 1, \dots, n_f(o) \\ \sum_{f=1}^{n_f(o)} b_{qofk} &\leq n_f(o) - 1 \end{aligned}$$

여기서 M 은 큰 양수인

시스템.

청구항 29

제28항에 있어서, 베이스 컨트롤러는 제28항의 식을 경유점들 사이의 n_k 개의 중간 시간 단계들에서 모든 장애물에서 n_q 개의 비행 기계 전부에 대해 제25항의 식에 도입하는

시스템.

청구항 30

제25항에 있어서, 베이스 스테이션은 다음의 제약 세트에 의해 비행체 1 및 2에 대해 수학적으로 나타낼 수 있는 경유점들 사이의 n_k 개의 중간 시간 단계들에서 제약을 실행함으로써 비행체의 궤적 경로 상의 경유점들 사이에서 트랜지션할 때 비행체들을 서로 안전 거리에 유지시키고,

$$\begin{aligned} \forall t_k : \quad &x_{T1}(t_k) - x_{T2}(t_k) \leq d_{x12} \\ \text{or} \quad &x_{T2}(t_k) - x_{T1}(t_k) \leq d_{x21} \\ \text{or} \quad &y_{T1}(t_k) - y_{T2}(t_k) \leq d_{y12} \\ \text{or} \quad &y_{T2}(t_k) - y_{T1}(t_k) \leq d_{y21} \end{aligned}$$

여기서 d는 비행체 1과 2 사이의 안전 거리를 나타내는 시스템.

청구항 31

제30항에 있어서, 상기 비행체는 축방향 대칭이고 $d_{x12}=d_{x21}=d_{y12}=d_{y21}$ 인 시스템.

청구항 32

제24항에 있어서, 베이스 스테이션은 상기 정수 제약들을 사용하여 각각의 큐드로터(q) 및 목표(g)에 다음의 정수 제약들을 적용함으로써 비행체들의 최적 목표 할당을 찾고,

$$\begin{aligned} x_{T_q}(t_{new}) &\leq x_g + M\beta_{qg} \\ x_{T_q}(t_{new}) &\geq x_g - M\beta_{qg} \\ y_{T_q}(t_{new}) &\leq y_g + M\beta_{qg} \\ y_{T_q}(t_{new}) &\geq y_g - M\beta_{qg} \\ z_{T_q}(t_{new}) &\leq z_g + M\beta_{qg} \\ z_{T_q}(t_{new}) &\geq z_g - M\beta_{qg} \end{aligned}$$

여기서 β_{qg} 는 최적 목표 할당하는데 사용되는 2차 변수인 시스템.

청구항 33

제32항에 있어서, 상기 베이스 컨트롤러는 적어도 n_g 개의 큐드로터가 원하는 목표 위치에 도달하는 것을 보장하도록 다음의 제약을 더 적용하는

$$\sum_{q=1}^{n_g} \sum_{g=1}^{n_g} \beta_{qg} \leq n_g n_g \dots n_g$$

시스템.

청구항 34

제18항에 있어서, 상기 베이스 컨트롤러는 복수의 그룹으로 비행체들을 조작하고-복수의 그룹은 각각 독립적으로 조정됨-, 그리고 복수의 그룹 각각에 대한 궤적을 생성하여 목표 위치들을 집단화하도록 더 프로그램되는 시스템.

청구항 35

제34항에 있어서, 베이스 컨트롤러는 비행체의 환경을 nr개의 볼록한 서브 구역으로 구획하고, 여기서 각각의 서브 구역은 동일한 수의 비행체 출발 및 목표 위치를 포함하고, 각각의 서브 구역 내부에서 비행체들에 대해 별개의 궤적들이 생성하여, 비행체들이 비행체들의 위치에 대한 선형 제약을 사용하여 각자의 서브 구역 내부에 머무르도록 요구하는

시스템.

명세서

기술분야

[0001] 관련 출원에 대한 상호 참조

[0002] 본 출원은 2012년 4월 30일자로 출원된 미국 가출원 제61/640,249호의 이익을 주장한다. 상기 출원의 내용은

본 명세서에 참고로 통합되어 있다.

[0003] 본 발명은 멀티-로터 공중 운행체 분야에 관한 것이다. 보다 구체적으로, 본 발명은 충돌 회피를 실행하도록 혼합 정수 이차 프로그램(MIQPs)을 사용하고 정수 제약(integer constraints)을 사용하여 최적의 궤적을 발생시키는 것에 의한, 복잡한 장애물에 대한 쿼드로터 팀의 3차원적 조정에 관한 것이다.

배경기술

[0004] 지난 10년에는 크기가 1m 보다 작고 질량이 1kg 또는 그 미만인 마이크로 공중 로봇 및 자동 공중 운행체에 있어서 급격한 진보가 있었다. 날개형 비행기로는 고정 날개형 운행체 내지 플래핑-날개형(flapping-wing) 운행체가 있을 수 있으며, 후자는 대개 곤충 비행에 의해 영감을 받은 것이다. 헬리콥터, 동축 로터 기구, 덕트 팬(ducted fan), 쿼드로터 및 헬사로터를 포함하는 로터 기구(rotor crafts)는, 로봇 연구소의 가장 통상적으로 사용되는 공중 플랫폼인 쿼드로터에 의해 보다 완성도가 있게 됨이 입증되었다. 이러한 유형의 장치에 있어서, 55cm의 날개 끝부터 끝까지의 날개폭과, 8cm의 높이와, 리튬 폴리머 배터리를 포함하여 약 500 그램의 질량을 가지며, 약 75 와트를 소비하는, Ascending Technologies, GmbH에 의해 판매되는 허밍버드 쿼드로터(Hummingbird quadrotor)가 현저하게 우수하고 견고한 플랫폼이다.

[0005] 멀티-로터 공중 운행체는 그 기계적 단순성과, 동적 특성과, 실내 및 실외 환경 모두에 대한 적격성으로 인하여, 점점 인기있는 로봇식 플랫폼이 되었다. 특히, 4개의 로터를 구비하는 로터기구인 쿼드로터의 설계, 제어 및 계획에 있어서 최근 많은 진보가 있었다. 이하에서 설명하는 바와 같이, 본 발명은 장애물이 있는 환경에서 도 1에 도시되어 있는 것과 유사한 이종의 쿼드로터 팀에 대해 최적의 궤적을 생성하는 방법에 관한 것이다.

[0006] 마이크로 공중 로봇은 많은 실용적인 적용예에서 극복하기 어려운 기본적인 유효 탑재량 제한을 갖고 있다. 그러나, 더 큰 유효 탑재량이 그리퍼 또는 케이블을 사용하는 다수의 UAVs에 의해 조정 및 운반될 수 있다. 다수의 센서로부터 넓은 영역 또는 형상을 커버하여야 하는 감시 또는 조사 및 구조와 같은 적용예는 각각 상이한 센서를 갖는 다수의 UAVs를 조화시킴으로써 처리될 수 있다.

[0007] 쿼드로터가 신속하고 정확하게 따를 수 있는 궤적은 위치의 3차 도함수(또는 C3)까지 연속적이어야 한다. 그 이유는, 쿼드로터에 있어서, 측방향 가속의 불연속은 률 및 피치 각도에 있어서의 순간적인 변화를 필요로 하고, 측방향의 저크(jerk)의 불연속은 각속도에 있어서의 순간적인 변화를 필요로 한다. C3 궤적을 찾는 데에는, 도달성(reachability) 알고리즘, 충분적 탑색 기법, 또는 LOR-트리-기반 탑색을 사용하는 방법에 실행될 수 없는 고차원 탑색 공간에서의 플래닝을 필요로 한다. 문제는 다수의 운행체를 계획할 때에 악화되는데, 그 이유는 탑색 공간의 크기가 더 넓어지기 때문이다.

[0008] 본 발명은 매우 소형의 마이크로 UAV를 개발하도록 쿼드로터 플랫폼을 축소하는 문제를 다루고 있다. 크기 축소의 가장 중요하고 명백한 이점은 간결한 형태로 매우 제한된 환경에서 작동되는 쿼드로터의 능력이다. 쿼드로터의 유효 탑재 능력은 급격하게 떨어지는 한편, 이러한 한계를 극복하도록 협동하는 다수의 쿼드로터를 배치 할 수 있다. 또한, 소형의 크기는 더 작은 운행체가 큰 운행체보다 매우 근접하여 작동될 수 있기 때문에 유리하다. 소형화의 또 다른 흥미로운 이점은 민첩성이다. 더 작은 쿼드로터는 더 큰 가속도를 가져서, 교란에 대해 보다 신속한 적응이 가능해지고 안정성이 더 높아질 수 있다.

[0009] 발명자에 의한 선행 작업은 쿼드로터의 동적 모델이 미분적으로 플랫(differentially flat)하다는 것을 보여주었다. 발명자는 이러한 사실을 이용하여, 궤적의 매끄러움에 대한 요건을 충족시키면서, 원하는 위치, 속도, 가속도, 급작스런 운동 및 입력에 대한 제약을 자연적으로 포함하게 하는 궤적 생성 알고리즘을 유도하였다. 발명자는 본 발명에 따른 방법을 다수의 쿼드로터와 장애물을 포함하도록 확장시켰다. 상기 방법은 상이한 크기, 용량, 및 상이한 쿼드로터 사이의 변화하는 동적 효과를 허용한다. 발명자는 2차 프로그램(QP)을 혼합 정수 이차 프로그램(MIQP)으로 변환시키는 정수 제약을 사용하여 충돌 회피를 실행한다.

[0010] 또한, 발명자에 의한 선행 작업은 혼합 정수 선형 프로그램(MILPs)과, 쇼우베나스(Schouwenaars) 등의 "강력하게 안전 보장되는 다수의 항공기의 분산된 협동 궤적 플래닝(Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees)"[Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, August 2004]으로부터의 궤적 플래닝에 대한 적용에 대한 광범위한 문헌으로부터 얻어진다. 본 명세서에 기재되어 있는 상기 방법은 이러한 작업을 토대로 한 것이다.

발명의 내용

[0011] 본 명세서에 기재되어 있는 방법 및 시스템은 고정 날개형 공중 운행체 및 로터 기구 모두에 있어서 유사한 궤적 플래닝 문제에서 정수 제약의 파워 및 유연성을 예시하고 있다. 이러한 해결 방안에 있어서의 핵심적인 차이는 플랫 출력 공간에서의 궤적을 합성하기 위한 구간별(piece-wise) 매끄러운 다항 함수의 사용이다. 구간별 매끄러운 다항 함수를 사용하면, 임의의 원하는 위치의 도함수까지 경유점들 사이에서 연속성을 실행하게 할 수 있다. 본 명세서에 기재되어 있는 바와 같은 궤적 생성에 있어서의 다른 차이점은 혼합 정수 선형 프로그램과 반대되는 혼합 정수 이차 프로그램으로 귀착되는 2차 비용 함수의 사용이다.

[0012] 예시적인 실시예에 있어서, 목표 위치를 향하는 적어도 2개의 비행 운행체의 비행 궤적을 제어하기 위한 시스템이 기재되어 있다. 시스템은 적어도 2개의 비행 운행체의 배향, 각속도, 위치 및 선형 속도를 결정하고 개선하기 위한 온보드(onboard) 관성 측정 유닛을 구비하는 적어도 2개의 비행 운행체와, 적어도 2개의 비행 운행체 각각의 현재 위치 및 속도를 탐지하기 위한 모션 캡쳐 시스템과, 상기 모션 캡쳐 시스템과 복수 개의 비행 운행체와 통신하는 베이스 컨트롤러를 포함하고 있다. 베이스 컨트롤러는 각각의 비행 운행체에 대해서 미리 정해진 시간 간격으로 구간별 매끄러운 다항 함수를 사용하고, 가중치를 적용하며, 중첩 제약을 실행하여, 최적의 궤적 경로를 계산한다. 베이스 컨트롤러는 또한, 계산된 최적의 궤적 경로에 기초하여, 개별적으로 그 상태를 제어하도록 각각의 비행 운행체에 명령을 보내어, 적어도 2개의 비행 운행체가 충돌을 피하면서 계산된 최적의 궤적 경로를 따르게 한다.

[0013] 또한, 본 발명은 목표 위치와 배향을 향하여 적어도 2개의 비행 운행체의 상태를 제어하기 위한 궤적 생성 방법을 포함하고 있다. 예시적인 실시예에 있어서, 상기 방법은 비행 운행체의 배향 및 각속도를 결정하는 단계와, 비행 운행체의 적어도 하나의 모터를 제어함으로써 비행 운행체의 배향 및 각속도를 제어하는 단계와, 각각의 비행 운행체의 현재 위치 및 속도를 결정하는 단계와, 원하는 배향 및 각속도와, 적어도 하나의 모터에 필요한 순 추력(net thrust)을 지정함으로써, 각각의 비행 운행체의 위치 및 속도를 제어하는 단계를 포함하고 있다. 각각의 비행 운행체에 있어서, 미리 정해진 시간 간격으로, 구간별 매끄러운 다항 함수를 사용하고, 가중치를 적용하며, 중첩 제약을 실행함으로써 최적의 궤적 경로가 계산된다. 이어서, 계산된 최적의 궤적 경로에 기초하여, 개별적으로 그 상태를 제어하도록 각각의 비행 운행체에 명령을 보내어, 이러한 비행 운행체가 충돌을 피하면서 계산된 최적의 궤적 경로를 따르게 한다. 각각의 비행 운행체의 개선된 현재 위치 및 속도가 제공되고, 프로세스는 복수 개의 미리 정해진 시간 간격에서 반복적으로 실행된다. 예시적인 실시예에 있어서, 비행 운행체의 각각의 상태는 그 배향 및 각속도와, 위치 및 선형 속도를 포함하고 있다. 배향 애러가 예측될 수 있으며, 배향은 각각의 비행 운행체 자체에서 제어된다.

[0014] 최적의 궤적 경로의 계산 동안에, 적어도 2개의 비행 운행체 각각에 적용되는 가중치는 다른 비행 운행체와 상이할 수 있다. 또한, 장애물 및 다른 운행체와의 충돌 제약을 실행하고 적어도 2개의 비행 운행체의 목표 위치를 최적으로 할당하도록 최적의 궤적 경로의 계산 동안에 정수 제약이 사용될 수 있다. 각각의 비행 운행체에 대한 최적의 궤적 경로의 계산은, 아래의 식에 따라 n_q 개의 큐드로터에 대한 위치의 k_r 차 도함수 제곱값의 적분을 최소화하면서, 정해진 시간 t_w 에서 원하는 경유점 n_w 를 통해 매끄럽게 트랜지션되는 궤적을 생성시키는 것을 포함할 수 있는데,

$$\begin{aligned} & \min \sum_{q=1}^{n_q} \int_{t_0}^{t_w} \left\| \frac{d^{k_r} r_{Tq}}{dt^{k_r}} \right\|^2 dt \\ \text{s.t. } & \quad r_{Tq}(t_w) = r_{wq}, \quad w = 0, \dots, n_w; \quad \forall q \\ & \quad \frac{d^j r_{Tq}}{dt^j}|_{t=t_w} = 0 \text{ or free, } w = 0, n_w; \quad j = 1, \dots, k_r; \quad \forall q \\ & \quad \frac{d^j r_{Tq}}{dt^j}|_{t=t_w} = 0 \text{ or free, } w = 0, n_w; \quad j = 1, \dots, k_r; \quad \forall q \\ & \quad \frac{d^j r_{Tq}}{dt^j}|_{t=t_w} = 0 \text{ or free, } w = 0, n_w; \quad j = 1, \dots, k_r; \quad \forall q \end{aligned}$$

[0015] [0016] 여기서, $\mathbf{r}_{Tq} = [x_{Tq}, y_{Tq}, z_{Tq}]$ 은 큐드로터 q 의 궤적을 나타내고, r_{wq} 는 큐드로터 q 의 원하는 경유점을 나타낸다. 적어도 2개의 비행 운행체 사이의 충돌 회피는, 비행기가 임의의 각도로 롤, 피치, 및 요 운동하고, 프리즘 내에 체류할 수 있도록 충분히 큰 측면 길이 l_x , l_y , l_z 를 갖는 전역 프레임(world frame)에 배향된 직사각형 프리즘으로서 비행 운행체를 모델링함으로써 제공될 수 있다. 프리즘은 $nf(o)$ 개의 면을 가진 구조 공간 내의 볼록 영역에 의해 표현되며, 각각의 면(f)에 있어서, $\mathbf{rk}, \mathbf{r}_{Tq}(\mathbf{rk})$ 시간에서의 비행 운행체의 원하는 위치가 장애물의 외부에 있는 조건은 아래의 식으로 표현되는데,

$$\mathbf{n}_{of} \cdot \mathbf{r}_{Tq}(tk) \leq s_{of}$$

[0017] 여기서, \mathbf{n}_{of} 는 구성 공간 내에서 장애물 (o)의 면(f)에 대한 법선 벡터이고, s_{of} 는 스칼라이며, 이에 의해 시간 tk 에서의 비행 운행체의 위치에 대한 식은 적어도 하나의 면에 대해 충돌되며, 그 후 직사각형 프리즘과 비행기는 장애물과 충돌하지 않게 된다. 비행기(q)가 시간 tk 에서 장애물(o)과 충돌하지 않는 조건은 2차 변수 b_{qofk} 에 의해 실행될 수 있고,

$$\begin{aligned} \mathbf{n}_{of} \cdot \mathbf{r}_{Tq}(tk) &\leq s_{of} + Mb_{qofk} \quad \forall f = 1, \dots, n_f(o) \\ b_{qofk} &= 0 \text{ or } 1 \quad \forall f = 1, \dots, n_f(o) \\ \sum_{f=1}^{n_f(o)} b_{qofk} &\leq n_f(o) - 1 \end{aligned}$$

[0019]

[0020] 여기서, M은 큰 양의 숫자이다.

[0021] 예시적인 실시예에 있어서, 후자의 식은 경유점들 사이의 n_k 중간 시간 단계에서 모든 장애물에 대한 모든 n_q 개의 비행 운송기를 위한 이전의 식에 도입된다. 아래의 제약에 의해 비행 운행체 1 및 2에 대해 수학적으로 표현될 수 있는 경유점들 사이의 n_k 개의 중간 시간 단계에서 제약을 실행함으로써, 비행 운행체는 비행 운행체의 궤적 경로 상의 경유점들 사이에서 트랜지션될 때에 서로로부터 안전한 거리에 유지될 수 있고,

$$\begin{aligned} \forall t_k : \quad &x_{T_1}(t_k) - x_{T_2}(t_k) \leq d_{x12} \\ \text{or} \quad &x_{T_2}(t_k) - x_{T_1}(t_k) \leq d_{x21} \\ \text{or} \quad &y_{T_1}(t_k) - y_{T_2}(t_k) \leq d_{y12} \\ \text{or} \quad &y_{T_2}(t_k) - y_{T_1}(t_k) \leq d_{y21} \end{aligned}$$

[0022]

[0023] 여기서, 용어 d는 비행 운행체 1 및 2 사이의 안전 거리를 나타낸다. 비행 운행체가 축방향으로 대칭이면, $d_{x12}=d_{x21}=d_{y12}=d_{y21}$ 이다.

[0024] 추가의 예시적인 실시예에 있어서, 정수 제약은 각각의 큐드로터(q) 및 목표(g)에 대해서 이하의 정수 제약을 적용함으로써 비행 운행체를 위한 최적의 목표 할당을 찾기 위해 사용될 수 있으며,

$$\begin{aligned} x_{Tq}(t_{n_w}) &\leq x_g + M\beta_{qg} \\ x_{Tq}(t_{n_w}) &\geq x_g - M\beta_{qg} \\ y_{Tq}(t_{n_w}) &\leq y_g + M\beta_{qg} \\ y_{Tq}(t_{n_w}) &\geq y_g - M\beta_{qg} \\ z_{Tq}(t_{n_w}) &\leq z_g + M\beta_{qg} \\ z_{Tq}(t_{n_w}) &\geq z_g - M\beta_{qg} \end{aligned}$$

[0025]

[0026] 여기서, β_{qg} 는 최적 목표 할당을 실행하기 위해 사용되는 2차 변수이다. 이하의 제약은 적어도 ng 개의 큐드로터가 원하는 목표 위치에 도달하는 것을 보장하도록 추가로 적용될 수 있다.

$$\sum_{q=1}^{ng} \sum_{g=1}^{ng} \beta_{qg} \leq n_g n_g - n_g$$

[0027]

[0028] 본 발명의 방법의 다른 예시적인 실시예에 있어서, 방법은 비행 운행체를 복수 개의 그룹으로 편성하는 단계로서, 복수 개의 그룹 각각이 독립적으로 조정되는, 편성 단계와, 목표 위치를 분류하기 위해 각각의 복수 개의 그룹에 대한 궤적을 생성시키는 단계를 포함하고 있다. 이러한 실시예에 있어서, 비행 운행체의 환경은 nr개의 볼록 서브 구역으로 분할되며, 여기서 각각의 서브 구역은 동일한 개수의 비행 운행체 출발 및 목표 위치를 포함하며, 개별적인 궤적들이 각각의 서브 구역 내의 비행 운행체에 대해 생성되며, 이에 의해 비행 운행체는 비행 운행체의 위치 상의 선형 제약을 사용하여 그 자체의 서브 구역 내에 머무르도록 요구된다.

도면의 간단한 설명

[0029] 본 발명의 전술한 특징들 및 다른 특징들과 장점들은 후술하는 도면의 상세한 설명으로부터 명확해질 것이다.

도 1은 비행중인 20개의 마이크로 큐드로터의 포메이션을 도시한다.

도 2는 마이크로 쿼드로터의 예시적 실시예를 도시한다.

도 3은 고도 컨트롤러 성능 데이터를 도시하며, (a)는 피치 앵글 스텝 입력 반응을 도시하며, (b)는 플립핑 조작을 위한 데이터를 도시한다.

도 4는 (a) 호버링 동안의 x, y, z 에러 및 (b) x(위) 및 z(아래)에서 위치 컨트롤러에 대한 스텝 입력 반응을 도시한다.

도 5는 기준 프레임 및 프로펠러 넘버링 방식(convention)을 도시한다.

도 6은 3개의 그룹으로 조직된 쿼드로터의 팀을 도시한다.

도 7은 쿼드로터의 그룹에 대한 소프트웨어 인프라스트럭처를 도시한다.

도 8은 4개의 쿼드로터 궤적을 따르는 포메이션을 도시하며, (a)는 4개의 운행체 각각에 대한 원하는 궤적 및 실제 궤적을 도시하며, 각각의 쿼드로터에 대한 포메이션 에러는 (b)에 도시되어 있다.

도 9는 (a)에서 그리드 포메이션의 20개의 쿼드로터에 대한 x, y, z에 대한 평균 표준 편차를 도시하며, (b)는 도 8의 패턴을 따르는 16개의 쿼드로터를 도시한다.

도 10은 윈도우를 통해 비행하는 4개의 쿼드로터의 4개의 그룹을 도시한다.

도 11은 평면형 그리드로부터 3차원 나선(위) 및 피라미드(아래)로 트랜지션하는 16개의 운행체의 팀을 도시한다.

도 12는 kQuad65(위), Asctec Hummingbird(중간), kQuad1000(아래) 쿼드로터의 비행을 도시한다.

도 13은 기준 프레임 및 프로펠러 넘버링 방식을 도시한다.

도 14는 솔리드 박스인 4개의 장애물이 있는 환경을 운항하는 단일 쿼드로터에 대한 궤적을 도시하며, 궤적은 실선으로 도시되며, 충돌 체크가 실행되는 n_k 개의 중간 시간 단계에서의 쿼드로터의 위치는 시간이 지남에 따라 더 어두워지는 좀 더 밝은 음영 상자로 도시되며, (a)는 시간-스텝 중첩 제약이 실행되지 않을 때의 궤적을 도시하며, (b)는 시간-스텝 중첩 제약이 실행될 때이다.

도 15는 (a)에서 다양한 속도에서 30차례 시험에 대한 RMSE를 도시하며, (b)-(d)는 단일 주행에 대한 데이터 [(a)에서 상자 안의 데이터]를 도시하며, 여기서 상자들은 도 16의 스냅 사진에 대응하는 실험 동안 특정 시간에서의 쿼드로터 위치를 나타내며 실선은 상기 주행에 대한 실제 쿼드로터 궤적을 나타내고 점선은 원하는 궤적을 나타낸다.

도 16은 3개의 쿼드로터 실험의 스냅 사진이며, 후프는 캡을 나타낸다.

도 17은 (a)에서 다양한 속도에서 11차례 시험에 대한 RMSE를 도시하며, (b)-(d)는 단일 주행에 대한 데이터 [(a)에서 상자 안의 데이터]를 도시하며, 여기서 상자들은 도 18의 스냅 사진에 대응하는 실험 동안 특정 시간에서의 쿼드로터 위치를 나타내며 실선은 상기 주행에 대한 실제 쿼드로터 궤적을 나타내고 점선은 원하는 궤적을 나타낸다.

도 18은 kQuad1000 쿼드로터 및 AscTec Hummingbird 쿼드로터를 이용한 실험의 스냅 사진이며, 후프는 수평 캡을 나타낸다.

도 19는 (a) 동질(homogeneous) 및 (b) 이질(heterogeneous) 쿼드로터 팀에 대한 포메이션 재구성을 위한 궤적을 도시하며, 여기서 상자들은 궤적 동안의 중간 시간에서의 쿼드로터 위치를 나타내고 실선은 실제 쿼드로터 궤적을 나타내며 점선은 원하는 궤적을 나타낸다.

도 20은 출발 시점(위), 중간 시점(중간), 최종 시점(아래)에서 라인 포메이션 내의 4개의 쿼드로터 트랜지션의 스냅 사진을 도시한다.

발명을 실시하기 위한 구체적인 내용

본 발명의 다양한 실시예의 철저한 이해를 제공하기 위해, 도 1 내지 도 20에 대한 특정한 구체적인 상세사항이 후술하는 설명에 제공된다. 그러나 본 발명의 다양한 실시예가 필요 없이 불분명해지는 것을 방지하기 위해, 특정한 잘 알려진 사항들은 후술하는 설명에 제공되지 않는다. 해당 기술분야의 당업자는 후술된 하나 이상의 상세한 설명 없이도 본 발명의 다른 실시예들을 실시할 수 있다는 것을 이해할 것이다. 또한, 후술하는 설명에

서 다양한 방법들이 단계 및 순서를 참고하여 설명되었더라도, 그 기재사항은 본 발명의 실시예의 명확한 트랜지션을 제공하는 것으로 의도되며, 단계 및 단계의 순서는 본 발명의 실시를 위해 요구되는 것으로 간주되지 않아야 한다.

[0031] 마이크로 쿼드로터

호버 상태로부터 선형 및 각 가속도를 생성시키는 쿼드로터의 능력을 분석하기 위해 단순 물리 모델을 개발하는 것이 유용하다. 특성 길이가 L 이라면, 로터 반경(R)은 L 에 대해 선형으로 비례한다. 질량은 L^3 에 비례하고, 관성 모멘트는 L^5 에 비례한다. 반면, 로터로부터의 양력 또는 추력(F) 및 항력(D)은 단면적 및 블레이드-팁 속도(v)의 제곱에 비례한다. 블레이드의 각 속력이 $\omega = v/L$, $F \sim \omega^2 L^4$, 및 $D \sim \omega^2 L^4$ 인 경우, 선형 가속도(a)는 $a \sim \omega^2 L^4 / L^3 = \omega^2 L$ 에 비례한다. 로터로부터의 추력을 모멘트 아암(L)을 갖는 모멘트를 생성한다. 따라서, 각 가속도 $a \sim \omega^2 L^5 / L^5 = \omega^2$ 이다.

모터가 작아질수록 양력과 동일한 방식으로 또한 비례하는 항력을 저항으로 인해 최대 속력을 제한하는 더 적은 토크를 생성시키기 때문에, 로터 속력(ω)은 길이에도 비례한다. 공중 운행체의 스케일링을 연구하기 위한 2가지 통상적으로 수용되는 접근법이 있다. 마하 스케일링(Mach scaling)은 압축성 유동을 위해 사용되며 본질적으로 팁 속도가 $\omega \sim 1/R$ 로 유도되는 상수라 가정한다. 프라우드 스케일링(Froude scaling)은 비압축성 유동을 위해 사용되며 유사한 비행기 형상의 경우 프라우드 수(v^2/L_g)가 상수라 가정한다. 여기서, g 는 중력으로 인한 가속도이다. 이는 $\omega \sim 1/(R)^{1/2}$ 을 유도한다. 그러나 프라우드 또는 마하 수의 유사성은 모터 특성은 물론 배터리 성능을 고려하지 못한다. 모터 토크가 길이에 따라 증가하는 반면, 로터에 대한 작동 속력은 모터의 토크-속력 특성을 프로펠러의 항력 대 속력 특성에 대해 매칭시켜 결정된다. 또한, 모터 토크는 요구되는 전류를 공급하는 배터리의 능력에 의존한다. 작은 길이 규모에서는 선택권이 거의 없기 때문에, 이러한 모든 변수들은 더 작은 설계를 위해 밀접하게 결합된다. 결국, 프로펠러 블레이드가 강성이라는 가정은 틀렸을 수 있으며 블레이드의 성능은 작은 규모에서 매우 달라질 수 있고, 속력에 관한 양력의 이차 스케일링은 정확하지 않을 수 있다. 그럼에도, 이러한 두 가지 경우는 기구(craft)의 조작성에 있어 유의미하다.

프라우드 스케일링은 각 가속도는 $a \sim L^{-1}$ 에도 불구하고 가속도는 길이에 의존하지 않는다고 제안한다. 반면, 마하 스케일링은 $a \sim L^{-2}$ 에도 불구하고 $a \sim L$ 이라는 결론을 도출한다. 쿼드로터는 병진을 위해 반드시 회전(각 가속도를 나타냄)해야 하기 때문에, 더 작은 쿼드로터가 훨씬 더 민첩하다. 쿼드로터 형상을 나타내는 두 가지 설계 포인트가 있다. Ascending Technologies의 Pelican 쿼드로터는 센서(대략 2kg의 총중량, 0.75m 직경, 호버링시 5400rpm 공청 로터 속력)를 장착하고 있으며 대략 400W의 전력을 소비한다. Ascending Technologies의 허밍버드 쿼드로터(500 그램 총중량, 대략 0.5m 직경, 호버링시 5000rpm 공청 로터 속력)는 추가적인 센서 없이 약 75W를 소비한다. 본 발명자들은 여기서 허밍버드의 크기의 대략 40%, 그 질량의 15%이며 호버링을 위해 대략 20%의 전력을 소비하는 쿼드로터를 위한 설계의 개요를 설명한다.

본 발명에 따른 쿼드로터의 예시적 실시예가 도 2에 도시된다. 이것의 봄(boom)은 하부의 커스텀 모터 컨트롤러 보드와 상부의 메인 컨트롤러 보드 사이에 개재된 탄소섬유 로드(rod)로 이루어진다. 양력을 생성하기 위해, 운행체는 8cm 직경을 갖는 4개의 고정-피치 프로펠러를 사용한다. 운행체의 프로펠러 팁-프로펠러 팁 거리는 21cm이며 그 중량은 배터리를 제외하고 50그램이다. 호버링 시간은 23그램인 2-셀 400 mAh Li-Po 배터리로 대략 11분이다.

그 작은 크기에도 불구하고, 도 2의 운행체는 온보드 센서의 완전한 세트(full suite)를 포함한다. 72MHz에서 작동하는 ARM Cortex-M3 프로세서가 메인 프로세서로서 작용한다. 운행체는 3축 자력계, 3축 가속도계, 롤 및 피치 축에 대한 2축 2000도/초 속도의 자이로, 및 요 축에 대한 단축 500도/초 속도의 자이로를 포함한다. 운행체는 고도 변화를 감지하는데 사용될 수 있는 기압계를 또한 포함한다. 통신을 위해, 운행체는 900MHz 또는 2.4GHz에서 작동할 수 있는 2개의 Zigbee 송수신기를 포함한다.

100Hz에서 각 운행체의 위치를 감지하기 위해, 바이콘(Vicon) 모션 캡쳐 시스템이 사용된다. 이 데이터는 기가비트 Ethernet 네트워크를 통해 데스크톱 베이스 스테이션으로 스트리밍된다. 100Hz에서 각 쿼드로터로 명령을 보내는 베이스 스테이션 상의 MATLAB에서 하이 레벨 제어 및 플래닝이 이루어진다. 쿼드 로터의 포메이션 팀을 제어하기 위한 소프트웨어는 이하에서 도 7과 관련하여 설명된다. 로우 레벨 예측 및 제어 루프는 600Hz의 속도에서 온보드 마이크로프로세서 상에서 실행된다.

[0038] 각각의 큐드로터는 900Mhz 및 2.4GHz에서 작동하는 2개의 독립적인 라디오 송수신기를 갖는다. 베이스 스테이션은 커스텀 라디오 모듈을 통해 배향, 추력, 각도 변화율 및 고도 컨트롤러 이득을 포함하는 원하는 명령을 개별 큐드로터로 보낸다. 온보드 변화율 자이로 및 가속도계를 사용하여 기구의 배향 및 각속도를 예측한다. 메인 마이크로프로세서는 아래쪽 고도 컨트롤러를 작동시키고 완전한 속도(full rate)에서 (600Hz) 4개의 모터 컨트롤러의 각각에 대해 원하는 프로펠러 속력을 송신한다.

[0039] 온보드 고도 컨트롤러에 대한 몇몇 성능 데이터가 도 3에 도시되어 있다. 운행체의 작은 관성 모멘트는 운행체가 큰 각 가속도를 생성하게 해준다. 도 3의 (a)에 도시된 바와 같이, 고도 제어는 0.2초 미만의 정정 시간(settling time)으로 대략적으로 임계적으로 감쇠되도록 설계된다. 이는 AscTec Hummingbird를 위한 고도 컨트롤러에 대한 정정 시간보다 두 배 빠른 것임을 주목해야 한다. 플립핑 조작을 위한 데이터가 도 3의 (b)에 나타나 있다. 여기서, 운행체는 약 0.4초에 y축 주위에서 완전한 플립을 완료하고 1850 도/초의 최대 각속도에 도달한다.

[0040] 후술하는 위치 컨트롤러는 운행체의 x 및 y 위치를 제어하기 위해 롤 및 퍼치 각도를 사용한다. 이러한 이유로, 강성(stiff) 위치 제어를 위해 강성 고도 컨트롤러가 요구된다. 측방향 및 수직방향으로의 스텝 입력에 대한 응답은 도 4의 (b)에 도시되어 있다. 도 4의 (a)에 도시된 호버링 성능 데이터의 경우, x 및 y에 대한 에러의 표준 편차는 약 0.75cm이며 z에 대해서는 약 0.2cm이다.

역학 및 제어

[0042] 마이크로 큐드로터에 대한 역학적 모델 및 제어는 본 발명자들의 기존 연구의 접근법을 기초로 한다. 도 5에 도시된 바와 같이, 본 발명자들은 큐드로터의 주축[단위 벡터 (b)]과 정렬된 본체-고정 프레임(B) 및 단위 벡터(a)를 갖는 관성 프레임(A)을 고려한다. B는 질량 중심(C) 및 회전 매트릭스(R)에 대한 위치 벡터(r)에 의해 A로 표현된다. 파라미터표현(parameterization)과 관련된 특이점을 피하기 위해, 본 발명자들은 배향을 표현하도록 완전 회전 매트릭스(full rotation matrix)을 사용한다. 본체 프레임 내의 큐드로터의 각속도(ω)는 $\dot{\omega} = \mathbf{R}^T \mathbf{R}$ 로 주어지며, 여기서 $\hat{\cdot}$ 는 벡터의 비대칭 매트릭스 형태를 지칭한다.

[0043] 도 5에 도시된 바와 같이, 4개의 로터는 1-4로 넘버링되며, 홀수 번호의 로터는 짝수 번호 로터에 반대되는 퍼치를 갖는다. 로터의 각 속력은 ω_i 이다. 결과적인 양력(F_i) 및 반응 모멘트(M_i)는 다음과 같으며,

$$F_i = k_F \omega_i^2, \quad M_i = k_M \omega_i^2$$

[0045] 여기서, 상수 k_F 및 k_M 은 실험적으로 결정된다. 마이크로 큐드로터에 있어서, 모터 역학은 10 밀리초 미만의 시간 상수를 가지며 강체 역학 및 공기역학의 시간 스케일보다 훨씬 빠르다. 따라서, 본 발명자들은 역학을 무시하고 F_i 및 M_i 가 즉각적으로 변화할 수 있다고 가정한다. 따라서, 시스템에 대한 제어 입력(u)은 다음과 같이

$$\mathbf{u}_1 = \sum_{i=1}^4 F_i$$

b3 방향으로의 순 추력(net thrust), 및 B에서의 모멘트, $[u_2; u_3; u_4]^T$,에 의해 구성되며,

$$\mathbf{u} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \quad (1)$$

[0046]

[0047] 여기서, L은 프로펠러의 회전축으로부터 큐드로터의 중심까지의 거리이다.

[0048] 운동의 뉴턴-오일러 식은 다음과 같으며,

$$m\ddot{\mathbf{r}} = -mg\mathbf{a}_3 + u_1\mathbf{b}_3 \quad (2)$$

$$\dot{\omega} = \mathcal{I}^{-1} \left[-\omega \times \mathcal{I}\omega + \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} \right] \quad (3)$$

[0049] 여기서 \mathcal{I} 는 \mathbf{b}_i 를 따르는 관성 모멘트 매트릭스이다.

[0051] 본 발명자들은 시간-파라미터화된 위치 벡터 및 요 각도를 사용하여 원하는 궤적을 구체화한다. 궤적,

$\sigma(t) : [0, t_f] \rightarrow \mathbb{R}^3 \times \text{SO}(2)$, 을 가정하면, 컨트롤러는 위치 및 속도 에러를 기초로 입력(u_i)을 유도하며,

$$u_1 = (-K_p \mathbf{e}_p - K_v \mathbf{e}_v + mg\mathbf{a}_3) \cdot \mathbf{b}_3 \quad (4)$$

[0052] 여기서, $\mathbf{e}_p = \mathbf{r} - \mathbf{r}T$ 이고, $\mathbf{e}_v = \dot{\mathbf{r}} - \dot{\mathbf{r}}T$ 이다. 나머지 3개의 입력은 원하는 회전 매트릭스을 연산하여 결정된다. 본 발명자들은 추력 벡터($u_1 \mathbf{b}_3$)를 식(4)의 $(-K_p \mathbf{e}_p - K_v \mathbf{e}_v + mg\mathbf{a}_3)$ 과 정렬시키기를 원한다. 본 발명자들은 또한 요 각도가 구체화된 요[T(t)]를 따르기를 원한다. 이러한 두 가지 정보로부터, 본 발명자들은 R_{des} 및 다음에 따른 회전 에러를 연산할 수 있으며,

$$\mathbf{e}_R = \frac{1}{2} (R_{des}^T R - R^T R_{des})^\vee$$

[0054] 여기서 \vee 는 $so(3)$ 의 요소를 \mathbb{R}^3 로 취하는 브이 맵(vee map)을 나타낸다. 원하는 각속도는 R 에 대한 표현을 미분하여 연산되며, 원하는 모멘트는 배향 에러(e_R) 및 각속도 에러(e_ω)의 함수로서 표현될 수 있고,

$$[u_2, u_3, u_4]^T = -K_R \mathbf{e}_R - K_\omega \mathbf{e}_\omega, \quad (5)$$

[0055] 여기서, K_R 및 K_ω 는 대각선 이득 매트릭스이다. 최종적으로, 본 발명자들은 식(1)을 인버팅하여 원하는 u 를 달성하기 위해 원하는 로터 속력을 계산한다.

그룹에 대한 제어 및 계획

[0056] 본원의 주요 요지는 많은 팀의 큐드로터를 조직화하는 것이다. 상태 공간 차원수의 성장으로 인한 복잡성을 관리하고 라벨링된 운행체 사이의 상호작용으로 생성되는 조합 급증을 제한하기 위해, 발명자들은 팀 아키텍처를 고려하고 팀은 라벨링된 그룹으로 조직되며 각 그룹은 라벨링된 운행체를 갖는다. 정식적으로, 발명자들은 단일 과제를 완료하기 위해 동시에 일하는 에이전트의 집합체로서 한 그룹의 에이전트를 형성할 수 있다. 둘 이상의 그룹은 다중적으로 병행되는 부과제를 동시에 완료할 것이 요구되는 과제를 완료하기 위해 하나의 팀으로서 기능한다. 발명자들은 그룹 내의 운행체가 낮은 잠복기의 높은 데이터율로 통신할 수 있고, 그룹을 교차하는 조정에 대한 통신 요구 사항은 상당히 엄격하다고 가정한다. 그룹 내의 운행체는 라벨링되는 것이 매우 중요하다. 그룹의 크기를 작게 하여 발명자는 형상, 통신 토플로지, 및 개별 민첩 로봇의 상대 위치에 대한 글로벌 보증을 제공하는 컨트롤러 및 플래너를 설계할 수 있다.

[0057] 이러한 접근법은 수많은 에이전트가 필요한 참분권화 접근법(truly decentralized approach)과는 반대이다. 항공 로봇들을 모으되 리더가 없는 집합체 모델은 로봇 커뮤니티에서 논의되었으나, 여기서 로봇들 사이의 라벨링된 상호 작용을 열거하는 것은 통계적 분배로서 이러한 집합체 기술의 포메이션을 제어하는 것으로 우회한다. 이러한 방법은 형상 또는 토플로지에 대한 보증을 제공할 수 없다. 상호 충돌 회피 알고리즘은 최종 목적으로 로봇을 조종하는 잠재성을 갖지만, 일시적 성능에 대해 보증할 수 없고 수렴 증거를 얻을 수 없다.

[0058] 한편, 항공 로봇에 대한 몇몇 실험 결과를 얻을 수 있음에도, 3차원 강성 구조체를 트레킹하는 궤도에 대해 분권화된 컨트롤러를 설계하는 문제점은 이제 적절하게 이해된다. 여기서 프레임워크는 그룹 내 이러한 강성 구조체의 유지보수를 허용한다.

[0059] 포메이션 비행은 많은 팀의 운행체에 대한 궤적을 생성하는 복잡성을 감소시켜 단일체에 대한 궤도를 생성한다. 컨트롤러가 잘 설계되는 경우, 운행체 사이에 충돌 회피를 명시적으로 포함시킬 필요가 없다. 시간(t)에서의 큐드로터(q)에 대한 위치 에러는 다음과 같이 표현될 수 있다.

$$e_{pq}(t) = e_f(t) + e_{Iq}(t) \quad (6)$$

[0060] 여기서, $e_f(t)$ 는 예정 궤도로부터 그룹의 위치 에러를 설명하는 포메이션 에러이고, $e_{Iq}(t)$ 는 그룹의 포메이션의 큐드로터(q)의 로컬 에러이다. 발명자들이 이하에서 설명하는 바와 같이, 포메이션 에러가 상당히 클 수 있음에도 로컬 에러는 난폭 궤도에 대해 전형적으로 상당히 작다.

[0061] 포메이션 비행의 주요 단점은 강성 포메이션이 포메이션 간극을 통해서만 고정될 수 있다는 것이다. 이것은 각 그룹이 독립적으로 간극을 지날 수 있도록, 팀의 포메이션 형상을 변경하거나 팀을 작은 그룹으로 나눔으로서

해결될 수 있다.

[0066] 궤도 생성 문제의 복잡성을 감소시키는 다른 방법은 모든 운행체가 동일 팀 궤도를 쫓아가지만 몇몇 시간 증가 만큼 구분되는 것을 요구한다. 발명자들은 큐드로터(q)에 대한 궤도를 아래와 같이 정의한다.

$$\mathbf{r}_{Tq}(t) = \mathbf{r}_{TT}(t + \Delta t_q) \quad (7)$$

[0068] 여기서, \mathbf{r}_{TT} 는 팀 궤도이고 Δt_q 는 몇몇 공통 시작(t)으로부터 큐드로터(q)에 대한 시간 이동이다. 팀 궤도가 교차하지 않거나 자체의 불안전한 거리 내에 있으면, 운행체는 안전한 시간 간격으로 서로를 단순히 따르는 것이 필요하다. 많은 수의 운행체는 두개의 운행체가 동일 시간에서 어느 교차점에도 존재하지 않도록 시간 간격 (t_q)이 선택되는 경우에, 자체 교차하는 팀 궤도를 따를 수 있다. 이하에서 팀 궤도를 교차시키는 시험을 설명 한다.

[0069] 발명자들은 공지된 3차원 환경 만족 사양을 통해 많은 운행체에 대한 경유점 상에 원활하고 안전한 궤도를 생성하는 방법을 설명할 것이다. 장애물 및 다른 운행체와의 충돌 제약을 실행하고 선택적으로 목표 위치를 최적으로 할당하기 위해 정수 제약이 사용된다. 이 방법은 계획되어 혼합 정수 선형 프로그램(MILPs) 및 이들의 궤적 플래닝에의 적용에 대한 쇼우베나스(Schouwenaars) 등의 방대한 문헌으로부터 유래된다. 최적화는 특정 시간 (t_w)에서 n_w 개의 원하는 경유점을 통한 원활한 트랜지션의 궤적을 생성하는데 사용될 수 있다. n_q 개의 큐드로터에 대한 k_r 차 도함수 제곱값의 적분을 최소화하면서 이 문제를 해결하는 최적화 프로그램이 이하에 설명된다.

$$\begin{aligned} & \min \sum_{q=1}^{n_q} \int_{t_0}^{t_{n_w}} \left\| \frac{d^k \mathbf{r}_{Tq}}{dt^k} \right\|^2 dt \\ \text{s.t. } & \mathbf{r}_{Tq}(t_w) = \mathbf{r}_{wq}, \quad w = 0, \dots, n_w; \forall q \\ & \frac{d^j \mathbf{r}_{Tq}}{dt^j} \Big|_{t=t_w} = 0 \text{ or free}, \quad w = 0, n_w; j = 1, \dots, k_r; \forall q \\ & \frac{d^j \mathbf{r}_{Tq}}{dt^j} \Big|_{t=t_w} = 0 \text{ or free}, \quad w = 0, n_w; j = 1, \dots, k_r; \forall q \\ & \frac{d^j \mathbf{r}_{Tq}}{dt^j} \Big|_{t=t_w} = 0 \text{ or free}, \quad w = 0, n_w; j = 1, \dots, k_r; \forall q \end{aligned} \quad (8)$$

[0070]

[0071] 여기서, $r_{Tq} = [x_{Tq}, y_{Tq}, z_{Tq}]$ 은 큐드로터(q)에 대한 궤도를 나타내고, r_{wq} 는 큐드로터(q)에 대한 원하는 경유점을 나타낸다. 발명자들은 t_1, \dots, t_{n_w-1} 에서의 r_{Tq} 의 1차 k_r 차 도함수의 연속성을 실행한다. 구간별 다항 함수로서 궤도를 작성하는 것으로 궤도는 이차 프로그램(또는 QP)으로 작성되고 여기서 결정 변수는 다항식 계수이다.

[0072] 궤드로터에 대해, 입력(u_2 및 u_3)은 위치의 4차 도함수로 나타내지기 때문에, 발명자들은 스냅의 노음(norm)의 제곱의 적분을 최소화하는 궤도를 생성한다(가속도의 2차 도함수, $k_r=4$). 큰 차수의 다항식은 중간 경유점에 의해 명확하게 구체화되지 않는 추가 제약을 장애물 회피로서 만족시키는데 사용된다.

[0073] 충돌 회피에 대해, 발명자들은 측면 길이(l_x , l_y 및 l_z)를 갖는 전역 프레임을 사용하여 배향된 직사각형 프리즘으로서 큐드로터를 모델링한다. 이 길이는 충분히 커서 큐드로터는 임의의 각도로 둘, 피치, 및 요 운동하여 프리즘 내에 머무를 수 있다. 발명자들은 n_o 볼록 장애물을 갖는 환경을 통해 이 프리즘을 운행하는 것을 고려하였다. 각 볼록 장애물(o)은 $nf(o)$ 면을 갖는 구성 공간에 볼록 영역으로 표현될 수 있다. 각 면(f)에 대해 시간(tk), $\mathbf{r}_{Tq}(tk)$ 에서 장애물(o) 외측에 있는 큐드로터의 원하는 위치는 다음과 같이 작성될 수 있다.

$$nof \cdot \mathbf{r}_{Tq}(tk) \leq sof. \quad (9)$$

[0074] 여기서, nof 는 구성 공간에서 장애물(o)의 면(f)에 대한 벡터이고, sof 는 평면의 위치를 결정하는 스칼라이다. 식(9)가 적어도 하나의 면에 대해 만족하는 경우, 이후 직사각형 프리즘, 그리고 이에 따른 큐드로터는 장애물과 충돌하지 않는다. 큐드로터(q)가 시간(tk)에서 장애물(o)과 충돌하지 않는 조건은 2차 변수(b_{qofk})로 다음과 같이 실행될 수 있다.

$$\begin{aligned} n_{of} \cdot \mathbf{r}_{Tq}(tk) &\leq s_{of} + Mb_{qofk} \quad \forall f = 1, \dots, n_f(o) \quad (10) \\ b_{qofk} &= 0 \text{ or } 1 \quad \forall f = 1, \dots, n_f(o) \\ \sum_{f=1}^{n_f(o)} b_{qofk} &\leq n_f(o) - 1 \end{aligned}$$

[0076]

[0077] 여기서, M은 큰 양수이다. b_{qofk} 가 1인 경우, 이후 면(f)에 대한 부등식은 항상 만족된다. 식(10)에서 마지막 부등식은 비충돌 제약이 장애물의 적어도 하나의 면에 대해 만족되는 것을 요구하고, 이는 프리즘이 장애물과 충돌하지 않는 것을 내포한다. 발명자들은 이후에 경유점들 사이의 n_k 개의 중간 시간 단계에서 모든 장애물이 없는 경우 모든 n_q 쿼드로터에 대해 식(10)을 식(8)에 도입할 수 있다. 이차 프로그램에 정수형 변수를 추가하여 이 최적화 문제는 혼합 정수 이차 프로그램(MIQP)이 된다.

[0078] 경유점들 사이에서 트랜지션할 때, 쿼드로터는 서로에 대해 안전한 거리로 떨어져 머물러야 한다. 발명자들은 경유점들 사이에서 n_k 개의 중간 시간 단계에 이러한 제약을 실행하고, 이는 쿼드로터(1 및 2)에 대해 다음 세트의 제약에 의해 수학적으로 표현될 수 있다.

$$\begin{aligned} \forall t_k : & \quad x_{T_1}(t_k) - x_{T_2}(t_k) \leq d_{x12} \\ & \text{or} \quad x_{T_2}(t_k) - x_{T_1}(t_k) \leq d_{x21} \\ & \text{or} \quad y_{T_1}(t_k) - y_{T_2}(t_k) \leq d_{y12} \\ & \text{or} \quad y_{T_2}(t_k) - y_{T_1}(t_k) \leq d_{y21} \end{aligned} \quad (11)$$

[0079]

[0080] 여기서, d 용어는 축방향 대칭 운행체($d_{x12}=d_{x21}=d_{y12}=d_{y21}$)에 대해 안전 거리를 나타낸다. 쿼드로터는 악조건에서의 불안정에도 트래킹 성능의 감소로 인해 서로에 대한 세류(downwash)를 회피해야한다는 점을 발명자들은 실험적으로 알았다. 따라서, 발명자들은 운행체가 서로 밑에서 비행하는 것을 허용하지 않는다. 최종적으로, 발명자들은 식(10)을 식(8)에 통합하는 것과 같은 방식으로 모든 n_q 개의 쿼드로터 사이에서 식(11)의 제약을 통합했다.

[0081] 많은 예에서, 임의의 운행체가 진행하는 것보다는 특정 쿼드로터는 특정 목표로 진행하는 것을 고려할 것이다. 발명자들은 운행체에 대한 최적 목표 할당을 찾기 위해 정수 제약을 사용하는 방법을 설명한다. 이에 따라, 최적화 문제에 많은 자유도가 있기 때문에 고정된-목표 할당 및 가끔씩 빠른 계획 시간과 비교하면 최종 비용이 낮아진다. 각 쿼드로터(q) 및 목표(g)에 대해 발명자들은 정수 제약을 도입한다.

$$\begin{aligned} x_{T_g}(t_{n_w}) &\leq x_g + M\beta_{qg} \\ x_{T_g}(t_{n_w}) &\geq x_g - M\beta_{qg} \\ y_{T_g}(t_{n_w}) &\leq y_g + M\beta_{qg} \\ y_{T_g}(t_{n_w}) &\geq y_g - M\beta_{qg} \\ z_{T_g}(t_{n_w}) &\leq z_g + M\beta_{qg} \\ z_{T_g}(t_{n_w}) &\geq z_g - M\beta_{qg} \end{aligned} \quad (12)$$

[0082]

[0083] 여기서, β_{qg} 는 최적 목표 할당을 실행하는데 사용되는 2차 변수이다. β_{qg} 가 0인 경우, 이후 쿼드로터(q)는 t_{n_w} 에서 목표(g)에 있어야 한다. β_{qg} 가 1인 경우, 이후 이들 제약은 쿼드로터(q)의 임의의 최종 위치에 대해 만족한다. 적어도 n_g 개의 쿼드로터가 원하는 목표에 도달하는 것을 보증하기 위해, 발명자들은 다음 제약을 도입한다.

$$\sum_{q=1}^{n_g} \sum_{g=1}^{n_g} \beta_{qg} \leq n_g n_q - n_g \quad (13)$$

[0084]

[0085] 이 접근법은 목표보다 쿼드로터가 더 많은 경우 또는 그 반대의 경우 용이하게 채용될 수 있다.

[0086] MIQP의 풀이 시간은 MIQP로 도입되는 2차 변수의 개수에 의해 지수적으로 증가한다. 따라서, 이러한 방법의 직접적인 사용은 많은 팀에 대해 확대되지 않다. 여기서 발명자들은 이러한 접근법이 많은 팀의 운행체에 대해 사용될 수 있는 두 개의 완화법을 제공한다.

[0087] 도 6에 도시된 바와 같이, 많은 팀의 운행체가 작은 그룹으로 나뉠 수 있다. 발명자들은 MIQP 방법을 사용하여 그룹 목표 위치에 대한 운행체의 트랜지션 그룹의 케도를 생성한다. 이는 모든 n_q 개의 운행체에 대한 케도를 계획하는 대신 발명자들이 단순히 그룹에 대한 케도를 계획하기 때문에 MIQP의 복잡성을 감소시킨다. 물론, 발명자들은 쿼드로터가 개별적으로 움직이는 유연성을 허용하지 않는 손실을 적용했다.

[0088] 많은 경우, 환경은 nr개의 볼록 서브 영역으로 분할될 수 있고, 각 서브 영역은 같은 수의 쿼드로터 출발 및 목표 위치를 포함한다. 환경을 분할한 후, MIQP 케도 생성 방법이 각 영역 내부의 운행체에 대해 사용될 수 있다. 여기서 발명자들은 운행체의 위치에 대한 선형 제약을 사용하여 자신의 영역 내부에 쿼드로터가 머무르

는 것을 요구한다. 이러한 접근법은 충돌이 없는 궤도를 보증하고 큐드로터에 개별적으로 움직이게 하는 유연성을 허용한다. 참된 최적화 솔루션은 큐드로터가 영역 경계부를 가로지르는 것을 요구하는 반면 이러한 완화 버전은 요구하지 않기 때문에 발명자들은 최적 비용으로 다루기 쉬운 성능을 얻는다. 또한, 실현 가능한 궤도가 서브 영역 내부에 존재하지 않지만, 영역 경계부를 가로지르는 실현 가능한 궤도가 존재할 수 있다. 그럼에도, 이러한 접근법은 많은 시나리오에 잘 작동하고, 발명자들은 16개 운행체의 팀에 대한 포메이션 트랜지션에 적용한다.

[0089] 큐드로터 동역학에 대한 모델

전역 프레임(W), 물체 프레임(B) 뿐 아니라 큐드로터에 대한 프로펠러 계수 컨벤션(propeller numbering convention)을 포함하는 좌표 시스템이 도 13에 도시된다. 발명자들은 또한 Z-X-Y를 사용한다. 오일러 각은 국부좌표시스템으로서 롤, 피치, 및 요 각도(ϕ , θ 및 ψ)를 형성한다. B로부터 W까지의 회전 매트릭스는 ${}^W_R {}^C R_B$ 로 제공되고 ${}^W_R C$ 는 중간 프레임(C)으로의 요 회전을 나타내고 여기서 ${}^C R_B$ 는 롤 및 피치의 효과를 나타낸다. 로봇의 각속도는 물체 프레임 내의 요소(p, q 및 r)와 함께 프레임(W)에서의 프레임(B)의 속도를 나타내는 ω_{BW} 로 나타낸다. 이들 값은 롤, 피치, 및 요 각도의 미분값에 직접 관련될 수 있다.

[0091] 각각의 로터는 각속도(ω_i)를 갖고, 아래 식에 따라 힘(F_i) 및 모멘트(M_i)를 생성한다.

$$F_i = k_F \omega^2, M_i = k_M \omega^2.$$

[0093] 실제로, 모터 역학은 강체 역학 및 공기역학에 비해 비교적 신속하다. 따라서, 컨트롤러 개발을 위해, 본 발명자는 이들이 순간적으로 달성될 수 있다고 가정한다. 따라서, 시스템에 대한 제어 입력은 u로서 표현될 수 있으며, 여기서, u_1 은 순 물체력이고, u_2, u_3, u_4 는 물체 모멘트이고, 로터 속도에 따라 다음과 같이 표현될 수 있다.

$$\mathbf{u} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \quad (1)$$

$$(14)$$

[0095] 여기서, L은 프로펠러의 회전축으로부터 큐드로터의 중심까지의 거리이다. 전역 프레임에서 질량 중심의 위치 벡터는 r로 표시된다. 시스템의 힘들은 $-z_W$ 방향의 중력과 z_B 방향으로의 로터 각각으로부터의 힘들의 합이다. 질량 중심의 가속에 관한 뉴턴 운동 식은 다음과 같다.

$$m\ddot{\mathbf{r}} = -mg\mathbf{z}_W + u_1 \mathbf{z}_B. \quad (15)$$

[0097] 오일러 식에 의해 결정된 각 가속도는 다음과 같다.

$$\dot{\omega}_{BW} = \mathcal{I}^{-1} \left[-\omega_{BW} \times \mathcal{I} \omega_{BW} + \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} \right], \quad (16)$$

[0099] 여기서, \mathcal{I} 는 $x_B - y_B - z_B$ 축을 따른 질량 중심에 관한 관성 모멘트 매트릭스이다. 시스템의 상태는 질량 중심의 위치 및 속도와 배향(오일러 각도에 의해 국소적으로 파라미터화됨) 및 각속도에 의해 다음과 같이 주어지거나,

$$\mathbf{x} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, p, q, r]^T$$

[0101] 질량 중심의 속도 및 위치와 회전 매트릭스(${}^W_R B$) 및 각속도(ω_{BW})에 의한 파라미터화 없이 주어진다.

[0102] 제어

[0103] 궤적을 따르기 위한 컨트롤러는 각각 시간($r_T(t), \psi T(t)$)의 함수로서 위치 및 요 각도로서 정의될 수 있다. 위치 및 속도의 오차들은 다음과 같이 정의될 수 있다.

[0104] $\mathbf{e}_p = \dot{\mathbf{r}} - \dot{\mathbf{r}}_T, \mathbf{e}_v = \dot{\mathbf{r}} - \dot{\mathbf{r}}_T.$

[0105] 다음에, 본 발명자는 컨트롤러와 원하는 물체 프레임 z 축을 위한 원하는 힘 벡터를 연산한다.

[0106] $\mathbf{F}_{des} = -K_p \mathbf{e}_p - K_v \mathbf{e}_v + mg \mathbf{z} W + m \ddot{\mathbf{r}} T,$

[0107] 여기서, K_p 및 K_v 는 양정치 이득 매트릭스이다. 여기서 본 발명자는 $\|\mathbf{F}_{des}\| \neq 0$ 으로 가정한다는 것을 주의하여야 한다. 다음에, 본 발명자는 큐드로터를 위한 원하는 힘과 제1 제어 입력을 연산하기 위해 원하는 힘 벡터를 실제 물체 프레임 z축 상에 투영한다.

[0108] $u_1 = \mathbf{F}_{des} \cdot \mathbf{z}_B.$

[0109] 다른 세 개의 입력을 결정하기 위해, 회전 오차를 고려하여야 한다. 먼저, 원하는 \mathbf{z}_B 방향은 원하는 추력 벡터를 따르는 것으로 판찰되었다.

[0110] $\mathbf{z}_{B,des} = \frac{\mathbf{F}_{des}}{\|\mathbf{F}_{des}\|}.$

[0111] 원하는 가속도 및 선택된 요 각도로부터, 전체 원하는 배향을 찾을 수 있다. 배향 오차는 원하는 회전 매트릭스(R_{des})와 실제 회전 매트릭스(R_B)의 함수이다.

[0112] $\mathbf{e}_R = \frac{1}{2}(R_{des}^T {}^W R_B - {}^W R_B^T R_{des})^\vee$

[0113] 여기서, V 는 $so(3)$ 내지 \mathbb{R}^3 의 원소를 갖는 브이 맵을 나타낸다. 오일러 각도의 편차가 이 매트릭스에 대한 근사치로서 사용될 수 있다는 것을 주의하여야 한다. 각속도 오차는 단순히 물체 프레임 좌표에서 원하는 각속도와 실제 각속도 사이의 편차이다.

[0114] $\mathbf{e}_\omega = {}^B[\omega_B] - {}^B[\omega_B]_{,T}.$

[0115] 이제, 원하는 모멘트와 3개의 나머지 입력이 다음과 같이 연산된다.

[0116] $[u_2, u_3, u_4]^T = -K_R \mathbf{e}_R - K_\omega \mathbf{e}_\omega, \quad (5) \quad (17)$

[0117] 여기서, K_R 및 K_ω 는 대각선 이득 매트릭스이다. 이는 롤, 피치, 및 요 각도 트래킹을 위해 고유한 이득을 사용할 수 있게 해준다. 마지막으로, 본 발명자는 식 (14)을 반전시킴으로써 원하는 u 를 달성하기 위한 원하는 로터 속도를 연산한다.

단일 큐드로터 궤적 생성

[0119] 본 부분에서, 본 발명자는 먼저 장애물을 공식에 포함시키는 르장드르 다항 함수를 사용한 기초 큐드로터 궤적 생성 방법을 설명한다. 구체적으로, 본 발명자는 중간 경유점 상의 사양을 충족시키는 공지된 3D 환경을 통해 매끄럽고 안전한 궤적을 생성하는 문제를 해결한다.

[0120] 지정된 시간에 n_w 개의 경유점을 통해 항공기를 운항하는 문제를 고려한다. 이를 제약을 충족시키는 통상적 궤적은 직선을 사용하여 경유점을 사이를 보간한 것이다. 그러나, 이 궤적은 비효율적이며, 그 이유는 큐드로터가 경유점에서 무한의 곡률을 가지고, 이러한 무한의 곡률은 큐드로터가 각 경유점에서 정지하는 것을 요구하기 때문이다. 본 명세서에 설명된 방법은 주어진 시간에 경유점을 통해 매끄럽게 전이되는 최적의 궤적을 생성한다. 이 문제를 해결하면서 동시에 위치의 k_i 차 도함수 제곱값의 적분을 최소화하는 최적화 프로그램은 다음과 같이 나타난다.

$$\begin{aligned}
 & \min \int_{t_0}^{t_{n_w}} \left\| \frac{d^{k_r} \mathbf{r}_T}{dt^{k_r}} \right\|^2 dt \\
 \text{s.t. } & \mathbf{r}_T(t_w) = \mathbf{r}_w, \quad w = 0, \dots, n_w \\
 & \frac{d^j x_T}{dt^j} \Big|_{t=t_w} = 0 \text{ or free}, w = 0, n_w; j = 1, \dots, k_r \\
 & \frac{d^j y_T}{dt^j} \Big|_{t=t_w} = 0 \text{ or free}, w = 0, n_w; j = 1, \dots, k_r \\
 & \frac{d^j z_T}{dt^j} \Big|_{t=t_w} = 0 \text{ or free}, w = 0, n_w; j = 1, \dots, k_r
 \end{aligned} \tag{18}$$

[0121]

[0122] 여기서, $\mathbf{r}_T = [x_T, y_T, z_T]^T$ 및 $\mathbf{r}_i = [x_i, y_i, z_i]^T$ 이다. 본 발명자는 $t_1, \dots, t_{n_w}-1$ 에서 \mathbf{r}_T 의 최초 k_r 도함수의 연속성을 적용한다. 다음에, 본 발명자는 다항 기초 함수($P_{pw}(t)$)를 사용하여 n_w 개의 시간 간격에 걸쳐 차수 n_p 의 구간별 다항 함수로서 궤적을 기재한다.

$$\mathbf{r}_T(t) = \begin{cases} \sum_{p=0}^{n_p} \mathbf{r}_{Tp1} P_{p1}(t) & t_0 \leq t < t_1 \\ \sum_{p=0}^{n_p} \mathbf{r}_{Tp2} P_{p2}(t) & t_1 \leq t < t_2 \\ \vdots \\ \sum_{p=0}^{n_p} \mathbf{r}_{Tpn_w} P_{pn_w}(t) & t_{n_w-1} \leq t \leq t_{n_w} \end{cases} \tag{7}$$

[0123]

[0124] 이는 본 발명자가 상수 $\mathbf{r}_{TpW} = [x_{TpW}, y_{TpW}, z_{TpW}]^T$ 를 $3n_p n_p \times 1$ 결정 변수 벡터 c 로서 기록함으로써 이차식 프로그램(또는 QP)로서 문제를 공식화할 수 있게 한다.

$$\begin{aligned}
 & \min c^T H c + f^T c \\
 \text{s.t. } & A c \leq b \\
 & A_{eq} c = b_{eq}
 \end{aligned} \tag{20}$$

[0125]

[0126] 본 명세서에 설명된 시스템에서, 입력 u_2 및 u_3 가 위치의 4차 도함수의 함수로서 나타나기 때문에, 본 발명자는 스냅(snap)(가속도의 2차 도함수, $k_r = 4$)의 노름(norm)의 제곱의 적분을 최소화하는 궤적을 생성한다. 식 (19)의 기초는 본 발명자가 더 고차의 다항식으로 진행할 수 있게 하며, 이는 본 발명자가 중간 경유점에 의해 명시적으로 지정되지 않은 장애물 회피로서 이런 추가적 궤적 제약을 만족시킬 수 있게 한다.

[0127]

비록, 이 문제 공식이 임의의 집단의 신장 다항 기초 함수($P_{pw}(t)$)에 대해 유효하지만, 선택은 해답자의 수치적 안정성에 영향을 준다. 기초 함수의 열악한 선택은 식 (20)의 매트릭스(H)가 큰 차수의 다항식에 대해 불량 조건화되게 할 수 있다. H 를 대각선화하고, 양호 조건 매트릭스가 되는 것을 보증하기 위해, 본 발명자는 여기서 위치의 k_r 차 도함수를 위한 기초 함수로서 르장드르 다항식을 사용한다. 르장드르 다항식은 -1로부터 1까지의 간격의 직교 다항식의 생성 집합이다.

$$\int_{-1}^1 \lambda_m(\tau) \lambda_n(\tau) d\tau = \frac{2}{2n+1} \delta_{nm}$$

[0128]

[0129] 여기서, δ_{nm} 은 크로네커 델타이고, 무차원 시간이다. 그후, 본 발명자는 이를 르장드르 다항식을 t_{w-1} 로부터 t_w 까지 간격으로 직교하도록 변위시키고, 본 발명자는 이를 $\lambda_{pw}(t)$ 라고 부른다. 본 발명자는 이 변위 르장드르 다항식을 위치 함수($P_{pw}(t)$)를 위한 최초 $n_p - k_r$ 기초 함수의 k_r 차 도함수를 나타내기 위해 사용한다. 이를 최초 $n_p - k_r$ 다항식은 다음을 충족하여야 한다.

$$\frac{d^{k_r} P_{pw}(t)}{dt^{k_r}} = \lambda_{pw}(t) \quad p = 1, \dots, (n_p - k_r)$$

[0130]

[0131] 본 발명자는 최종 k_r 한정 다항식 기초 함수를 $P_{pw}(t) = (t - t_{w-1})^p = (np - k_r + 1), \dots, n$ 으로서 규정한다. 이들 최종 k_r 다항식 기초 함수는 비용 함수에 어떠한 영향도 갖지 않으며 그 이유는 그 k_r 차 도함수가 0이기

때문이다. 이 연구에서, 본 발명자는 $k_r=4$ 로 취하고, Δ_p 는 일반적으로 9와 15 사이이다.

[0132] 충돌 회피를 위해, 본 발명자는 큐드로터를 측면 길이 I_x , I_y 및 I_z 를 갖는 전역 프레임으로 배향된 직사각 프리즘으로서 모델링하였다. 이를 길이는 큐드로터가 임의의 각도로 롤, 퍼치, 및 요 운동할 수 있고 프리즘 내에 머무를 수 있도록 충분히 크다. 본 발명자는 볼록한 장애물을 갖지 않는 환경을 통해 이 프리즘을 운항하는 것으로 고려한다. 각 볼록 장애물(o)은 $n_f(o)$ 면을 갖는 구성 공간에서 볼록 영역에 의해 표현될 수 있다. 각 면(f)에 대하여, 시간 t_k , $r_T(t_k)$ 에서 큐드로터의 원하는 위치가 장애물(o)의 외부에 있는 조건은 다음과 같이 기재될 수 있다.

$$\mathbf{n}_{of} \cdot \mathbf{r}_T(t_k) \leq s_{of}, \quad (21)$$

[0133] 여기서, \mathbf{n}_{of} 는 구성 공간의 장애물(o)의 면(f)에 대한 법선 벡터이고, s_{of} 는 평면의 위치를 결정하는 스칼라이다. 식 (21)이 적어도 하나의 면에 대해 충돌되는 경우, 이때, 직사각형 프리즘과, 이에 따라, 큐드로터는 장애물과 충돌하지 않는다. 프리즘이 시간 t_k 에 장애물(o)과 충돌하지 않는 조건은 2차 변수(b_{ofk})로 적용될 수 있다.

$$\begin{aligned} \mathbf{n}_{of} \cdot \mathbf{r}_T(t_k) &\leq s_{of} + M b_{ofk} \quad \forall f = 1, \dots, n_f(o) \\ b_{ofk} &= 0 \text{ or } 1 \quad \forall f = 1, \dots, n_f(o) \\ \sum_{f=1}^{n_f(o)} b_{ofk} &\leq n_f(o) - 1 \end{aligned} \quad (22)$$

[0135] 여기서, M 은 큰 양수이다. b_{ofk} 가 1인 경우, 이때, 면(f)의 부등식이 항상 충족된다. 식 (22)의 마지막 부등식은 장애물의 적어도 하나의 면에 대해 비충돌 제약이 충족되는 것을 필요로 하며, 이는 프리즘이 장애물과 충돌하지 않는다는 것을 의미한다. 이때, 본 발명자는 경유점들 사이의 n_k 개의 중간 시간 단계에서 모든 장애물에 대해 식 (22)을 (20)에 도입할 수 있다. 이차 식 프로그램에 정수 변수를 추가하면 이 최적화 문제는 혼합 정수 이차 식 프로그램(MIQP)이 된다.

[0137] 이 공식은 임의의 볼록 장애물에 대해 유효하지만, 본 발명자는 단순성을 위해 본 명세서에서는 단지 직사각 장애물만을 고려한다. 이 공식은 식 (22)에서 \mathbf{n}_{of} 를 $\mathbf{n}_{of}(t_k)$ 로 치환하고 s_{of} 를 $s_{of}(t_k)$ 로 치환함으로써 이동하는 장애물로 쉽게 확장된다. 또한, 이 프레임워크(framework) 내에서 볼록하지 않은 장애물도 효율적으로 모델링 될 수 있다.

[0138] 식 (20)은 연속적 시간 최적화를 나타낸다. 본 발명자는 시간을 불연속화하고 n_k 개의 시간 지점에 대해 식 (22)의 충돌 제약을 기록한다. 그러나, n_k 불연속 시간에서 충돌 제약은 궤적이 시간 단계 사이에서 충돌이 없는 상태가 되는 것을 보증하지는 않는다. 얇은 장애물에 대하여, 최적 궤적은 도 14a에 도시된 바와 같이 충돌 제약이 장애물 통과 직전 및 직후에 충족되도록 큐드로터가 장애물을 통해 이동할 수 있게 한다. 이 문제는 충돌 점검이 시간 단계 k 에서 적용되는 직사각형 프리즘이 시간 단계 $k+1$ 을 위한 대응 프리즘과 유한 교차를 갖는 것을 요건으로 함으로써 교정될 수 있다.

$$\begin{aligned} |x_T(t_k) - x_T(t_k + 1)| &\leq l_x \quad \forall k = 0, \dots, n_k \\ |y_T(t_k) - y_T(t_k + 1)| &\leq l_y \quad \forall k = 0, \dots, n_k \\ |z_T(t_k) - z_T(t_k + 1)| &\leq l_z \quad \forall k = 0, \dots, n_k \end{aligned} \quad (23)$$

[0139] 이러한 추가적인 시간-단계 중첩 제약은 도 14(b)에 도시된 바와 같이 궤적이 장애물을 지나가는 것을 방지한다. 시간-단계 오버랩을 실행하는 것은 시간 단계 사이의 평균 속도 제약을 실행하는 것과 동등하다. 물론, 솔루션이 실행 가능하도록 충분한 시간 단계가 이용되어야 한다. 궤적은 시간 이산화(time discretization)로 인해 코너를 여전히 잘라낼 것이다. 발명자는 충돌 검사가 실행되는 프리즘 및 장애물의 크기를 대략적으로 과장시킴에 의하여 이를 해결한다. 궤적이 찾아진 후, 발명자는 전체 궤적에 걸쳐 실제 큐드로터 형상이 임의의 장애물과 교차하지 않음을 보장하기 위해 충돌 검사를 실시한다.

[0141] 안전성과 공격성(aggressiveness) 사이의 트레이드오프(tradeoff)를 위해 시간적 스케일링을 이용할 수 있다. 발명자가 경유점을 운항할 시간을 예를 들어 $\alpha=2$ (궤적이 2배의 시간 동안 시행되게 함)의 계수로 변화시킬 경우, 시간 스케일링(time-scaled)된 문제에 대한 종래 솔루션의 버전은 단지 종래 솔루션의 시간 스케일링된 버전이 된다. 따라서, 발명자는 MIQP를 풀 필요가 없다. α 가 증가할수록 플랜은 실행하는데 더 오래 걸리고 더 안전해진다. α 가 무한에 가까워질수록 각속도, 위치 및 요 각도의 모든 도함수는 0에 가까워지며, 이는 극한에서 식(16) 및 식(17)의

$$\mathbf{u}(t) \rightarrow [mg, 0, 0, 0]^T,$$

[0142] 이 된다. 충분히 크게 만듬으로 인해, 발명자는 작은 피치 및 롤의 가정에 따라서 큐드로터를 위해 생성되는 임의의 모션 플랜을 달성할 수 있다. 역으로, 크기가 줄어들수록, 궤적은 시행되는데 더 적은 시간이 걸리고, 위치의 도함수는 증가하며, 궤적은 더 공격성이 되어 제로 피치 및 제로 롤 구성으로부터 큰 익스커션(excursion)으로 될 것이다.

다중 큐드로터 궤적 생성

[0143] 이 섹션에서, 발명자는 동일한 환경에서 지정된 목표 지점까지 각각 명시된 경유점을 갖고 종종 아주 근접하게 운항하는 n_q 개의 이종 큐드로터를 포함하도록 방법을 확장시킨다. 이는 결정 변수가 모든 n_q 개의 큐드로터의 궤적 계수인 식(20)의 확대 버전을 풀음으로써 수행된다. 충돌 회피 제약에 관하여, 각각의 큐드로터는 l_x , l_y , l_z 의 고유 값으로 명시된 바와 같이 상이한 크기일 수 있다. 또한, 발명자는 상대 비용 부여 및 상호-큐드로터 충돌 회피와 관련하여 이종 텁(term)을 고려한다.

[0144] 개별적으로 운항하는 큐드로터의 팀은 충돌로 이끄는 상충을 해결하고 3차원 공간을 "공유"해야 한다. 따라서, 이들은 환경을 운항하는 이들의 개별 궤적을 변경하고 서로를 피해야 한다. 모든 큐드로터가 동일한 타입일 경우, 이는 이들이 중첩 해결의 부담을 동등하게 나눈다는 것을 의미한다. 그러나, 이종 운행체의 팀에 대하여, 일부 큐드로터가 다른 큐드로터보다 상대적으로 쉬운 궤적을 따르거나, 사용자 선호도에 기초하여 큐드로터를 우선순위를 매기는 것이 바람직할 수 있다. 이는 큐드로터의 비용을 그에 따라서 부여함으로써 달성될 수 있다. 큐드로터(q)가 상대 비용(μ_q)를 갖는 경우, 식(20)의 다중-큐드로터 버전에서 이차 비용 매트릭스(H_m)가 다음과 같이 서술될 수 있다:

$$H_m = \text{diag}(\mu_1 H_1, \mu_2 H_2, \dots, \mu_{n_q} H_{n_q}) \quad (24)$$

[0145] [0147] 큐드로터에 더 큰 가중치를 적용하는 것은 이의 출발 및 목표 지점 사이에서 더 직접적인 경로를 취하게 한다. 더 작은 가중치를 적용하는 것은 하나의 큐드로터가 그 궤적을 변경하게 하고 다른 큐드로터가 더 큰 가중치를 갖게 한다. 이러한 성능은 느리고 큰 큐드로터를 위한 궤적은, 더 작고 더 민첩한 큐드로터를 위한 동일한 궤적보다 더 높은 비용을 할당받을 수 있으므로, 민첩한 큐드로터와 느린 큐드로터 양자의 팀에 특히 가치가 크다. 큰 큐드로터는 동일한 좁은 캡을 지나 비행하기 위해 작은 큐드로터보다 더 우수한 트래킹 정확도를 요구하여, 이는 또한 이러한 상황에서 더 큰 큐드로터에 더 높은 비용을 할당하는 데 유용하다.

[0148] [0149] 큐드로터는 서로로부터 안전 거리로 떨어져서 머물러야 한다. 발명자는 이러한 제약을 경유점들 사이의 n_k 개의 중간 시간 단계에 실행하며, 이는 큐드로터(1 및 2)에 다음의 제약 세트에 의해 수학적으로 나타내질 수 있다:

$$\begin{aligned} \forall t_k : & x_{1T}(t_k) - x_{2T}(t_k) \leq d_{x12} \\ \text{or } & x_{2T}(t_k) - x_{1T}(t_k) \leq d_{x21} \\ \text{or } & y_{1T}(t_k) - y_{2T}(t_k) \leq d_{y12} \\ \text{or } & y_{2T}(t_k) - y_{1T}(t_k) \leq d_{y21} \\ \text{or } & z_{1T}(t_k) - z_{2T}(t_k) \leq d_{z12} \\ \text{or } & z_{2T}(t_k) - z_{1T}(t_k) \leq d_{z21} \end{aligned} \quad (25)$$

[0150] 여기에서 용어 d 는 안전 거리를 나타낸다. 축대칭 운행체는 $d_{x12} = d_{x21} = d_{y12} = d_{y21}$ 이다. 실험상, 발명자는 트래킹 성능의 저하 및 더 나쁜 경우에는 심지어 불안정의 이유로 큐드로터가 유사-크기의 또는 더 큰 큐드로터의 세류에서 비행하는 것을 피해야 한다는 사실을 발견하였다. 그러나, 더 큰 큐드로터는 더 작은 큐드로터의 아

래에서 비행할 수 있다. 발명자는 더 큰 퀘드로터는 공중 작류 플랫폼으로써 기능하기 위해 작은 퀘드로터 아래에서 충분히 안정적으로 비행할 수 있음을 설명하였다. 따라서, 퀘드로터(1 및 2)와 동일한 탑입일 경우, $d_{z12} = d_{z21}$ 이 된다. 그러나, 퀘드로터(1)가 퀘드로터(2) 보다 훨씬 더 큰 경우, 퀘드로터(2)는 더 큰 퀘드로터 아래 어느 큰 거리(d_{z12})에서 비행해야 하고, 퀘드로터(1)는 퀘드로터(2)의 아래에서 더 작은 거리(d_{z21})로 나타내지는 가까운 거리에서 비행할 수 있다. 이러한 안전 거리의 정확한 값은 퀘드로터 탑입 사이의 상이한 이격 거리에 대한 트래킹 성능을 실험적으로 측정함에 의해 찾아질 수 있다. 최종적으로, 발명자는 모든 n_q 개의 퀘드로터 사이의 식(25)의 제약을 통합하고 동일한 방식으로 식(22)을 식(20)의 다중 퀘드로터 버전으로 통합한다.

[0152] 여기에서 발명자는 식(20, 22, 및 25)에 의해 형성된 3차원 내비게이션 문제에 대한 이러한 공식에 의하여 생성된 MIQP의 복잡성을 분석한다. 이러한 문제에서, 연속 변수 n_c 의 수는 기껏해야 다음과 같다:

$$n_c = 3n_w n_p n_q. \quad (26)$$

[0153] 일부 연속 변수는 균등 제약을 제거함으로써 MIQP로부터 삭제될 수 있다. 계산 시간을 결정하는 강한 요인은 소개되는 2차 변수(n_b)의 수이다. 3차원 내비게이션 문제에 대한 2차 변수의 수는 다음과 같다:

$$n_b = n_w n_k n_q \prod_{o=1}^{n_o} n_f(o) + n_w n_k \frac{n_q(n_q - 1)}{2} 6 \quad (27)$$

[0154] [0155] 수식(27)의 제1 항(term)은 장애물 회피 제약을 설명하고 제2 항은 상호-퀘드로터 안전 거리 실행을 나타낸다. 발명자는 MIQP를 풀기 위하여 분기 한정 솔버(branch and bound solver)를 이용한다. 최악의 경우, 분석할 트리(tree)의 2^{nb} 가지(leaves)가 존재한다. 따라서, 이는 많은 수의 로봇에 대하여 스케일링(scale)을 잘 조정하는 방법이 아니지만 이는 작은 팀(본원에서는 4 퀘드로터까지)에 최적의 궤적 및 적은 장애물을 생성시킬 수 있다. 이러한 기술에 의한 일 이점은 안전을 보장하는 차선의 실현가능한 해결책이라는 점이며, 가능한 계산 예산이 적은 경우 상충 해결이 매우 빨리 찾아질 수 있다(하기 표의 T_1 및 T_{opt} 을 비교).

실험 결과

[0156] 예시적인 퀘드로터의 구조는 매우 실질적인 이유로 중요하다. 퀘드로터의 큰 팀에 관하여, 모든 바이콘 데이터를 받고, 명령을 계산하고, 충분히 빠른 속도로 각각의 퀘드로터와 통신할 수 있는 단일 루프를 실행하는 것이 불가능하다. 도 7에 도시된 바와 같이, 각각의 그룹은 독립 스레드로 실행되는 데스크탑 베이스 스테이션(20)의 전용 바이콘 소프트웨어 노드(10)에 의하여 제어된다. 제어 노드(30)는 공유 메모리(40)를 통해 바이콘 노드(10)로부터 운행체 포즈(pose) 데이터를 받는다. 바이콘 노드(10)는 바이콘 트래킹 시스템에 연결되고, 각각의 물체의 마커 위치를 수신하고, 마커 데이터에 6D 포즈 팟을 수행하고, 속도 예측을 위해 추가적인 처리를 한다. 최종적으로, 처리된 포즈 예측은 Boost C++ 라이브러리를 이용하여 공유 메모리(40)로 재생된다. 공유 메모리(40)는 상호-처리 통신의 가장 빠른 방법이며, 이는 시간-임계적(time-critical) 데이터의 가장 적은 대기 시간을 보장한다.

[0157] Matlab에 설치된 제어 노드(30)는 직접 공유 메모리(40)로부터 포즈 데이터를 판독하고 상술된 조절기에 기초하여 다수의 퀘드로터의 명령화된 방향 및 순 추력(net thrust)을 계산한다. 비-시간-임계적 데이터 공유를 위해, 발명자는 Inter Process Communication (IPC)(50)을 이용한다. 예를 들어, 바람직한 운행체 위치와 같은 높은 레벨 사용자 명령(60)은 IPC(50)를 통하여 Matlab 제어 노드(30)로 전송된 운행체의 궤적을 계산하는 플래너(70)로 전송된다. IPC(50)는 유연한 메세지 전달을 제공하고 처리 사이에 데이터를 전송하는 데 TCP/IP 소켓을 이용한다.

[0158] 각각의 Matlab 제어 노드는 900 MHz 및 2.4 GHz Zigbee 송수신기를 포함하는 라디오 모듈과 연관되며, 이는 그룹 내의 모든 차량과 통신하는 데 이용된다. 라디오 모듈(80)은 다수의 운행체, 실시예에서는 5개 까지의 운행체에 제어 명령을 보낸다. 각각의 운행체는 개별 채널에서 작동하고, 라디오 모듈(80)은 각각의 퀘드로터의 주파수 사이에서 도약(hop)하고, 각각의 운행체에 100Hz로 명령을 보낸다. 또한, 라디오 모듈(80)은 2개의 독립 송수신기를 이용하여 운행체으로부터 높은 대역 피드백을 동시에 수용한다.

[0159] 도 8에서, 발명자는 궤적을 따르는 4개의 퀘드로터의 팀을 위한 데이터를 포메이션으로 나타낸다. 그룹 포메이션 에러는 국소적 에러보다 상당히 크다. 국소 x 및 y 에러는 항상 3 cm보다 작고, 포메이션 x 에러는 11cm만

큼 크다. 모든 운행체는 공칭 이득이므로 이러한 데이터는 모든 포메이션 궤적을 따르는 데이터를 대표한다. 따라서, 바람직한 궤적으로부터의 탈선이 크더라도, 그룹 내의 상대적 위치 에러는 작다.

[0162] 도 9(a)는 도 1에 도시된 그리드 포메이션에서 비행하는 20대 운행체의 평균 에러 데이터를 나타낸다. 이러한 실험에서, 운행체는 다수의 큐드로터-중심에서 큐드로터-중심까지의 그리드 간격에서 적어도 30초동안 1.3 미터의 높이에서 호버링하도록 제어된다. 모든 20대 운행체의 세류로부터 생성된 공기 디스터번스는 상당하며 이러한 포메이션의 임의의 운행체에 대하여 정체 공기에서의 개별 운행체에서보다 트래킹 성능을 악화시킨다. 그러나, 도 9(a)에 도시된 바와 같이, 이격 거리는 호버링하는 비행에 어떠한 영향도 갖지 않는다. 35cm 그리드 간격에서 프로펠러 팀 사이의 공칭 거리는 약 14cm이다.

[0163] 도 9(b)는 순환 형상 8자 패턴을 따르는 16대 운행체의 팀을 도시한다. 전체 사이클을 완성하는 시간은 t_c 이며 운행체는 궤적을 따라 $t_c/16$ 의 시간 증가율로 시간에서 균등하게 이격된다. 교차 지점에서 충돌 없는 궤적을 보장하기 위해, 운행체는 궤적의 하나의 루프에서 $15/32 t_c$ 가 걸리며 다른 루프에서는 $17/32 t_c$ 가 걸린다. 이러한 시간 제약을 만족시키고 교차 지점에서 (0 가속도와 저크를 갖는) 어떤 특정 속도를 갖는 궤적은 단일 운행체를 위한 최적화-베이스 방법을 이용하여 생성된다.

[0164] 또한, 발명자는 MIQP 궤적 생성 문제를 해결하는 데 분기 한정 솔버를 이용한다. MIQP에 대한 해결 시간은 이 항 제약의 수 및 또한 환경의 기하학적 복잡성의 지수 함수이다. 본원에서 출발된 예시에 대하여, 3.4 Ghz Corei7 Quad-Core 테스크탑 기계에서 제1 해답은 몇 초 안에 산출되지만, 실제 최적 해답 및 씨티피케이트 최적성을 찾는 데에는 적어도 20분이 걸린다.

1) 팀 내의 그룹을 위한 플래닝:

[0166] 도 10은 캡의 한 측에서 다른 측으로 이동하는 4대의 큐드로터의 4개의 그룹에 대한 실험으로부터의 스냅샷을 도시한다. 이러한 예시에서 최적 목표 과제는 그룹-레벨에서 수행된다는 점을 주목하여야 한다.

2) 서브 구역을 위한 플래닝:

[0168] 도 11은 평면 그리드로부터 3차원 헬릭스 및 피라미드로 트랜지션하는 16대 운행체를 이용한 실험으로부터의 스냅샷을 도시한다. 16대 운행체를 위한 궤적을 생성하는 데 MIQP 접근을 직접 이용하는 것은 유용하지 않다. 따라서, 양자의 실험에서 공간은 두 개의 지역으로 나뉘어지고 각각 8대 운행체의 개별 MIQP가 포메이션의 좌측 및 우측 상의 운행체를 위한 궤적을 생성하는 데 이용된다. 일반적으로, 포메이션은 대칭적일 필요는 없으나, 본원에서 발명자는 대칭을 이용하여 이러한 예시에 대하여 오직 8대 운행체를 위한 단일 MIQP를 푼다. 최적 목표 과제는 운행체이 총 비용을 최소화시키는 목적을 집합적으로 선택하는 데 이용된다.

[0169] 본원에서 출발된 실험은 Ascending Technologies Hummingbird 큐드로터 그리고 내부 개발된 kQuad65 및 kQuad1000 큐드로터에 의해 수행되었고, 각각 457, 65, 및 962 그램의 하중이 나가고, 55, 21, 및 67대의 블레이드 팀 길이에 대한 블레이드 팀을 갖는다. 이러한 큐드로터들은 도 12에 도시되어 있다. 발명자는 큐드로터의 위치 및 속도를 예측하는 데 바이콘 모션 캡쳐 시스템을 이용하고 방향 및 각속도를 예측하는데 온보드 IMU를 이용한다. 소프트웨어 기반구조는 2010년 9월, N Michael, D Mellinger, Q Lindsey, 및 V. Kumar의 "파지 다중 마이크로 uav 테스트베드 (*The grasp multiple micro uav testbed*)"에서 설명된다.

[0170] 발명자의 이전 연구에서, 방향 에러 텀(term)은 모션 캡쳐 시스템에 의해 측정된 방향을 사용하여 운행체 오프-보드에서 계산되었다. 이러한 오프-보드 계산은 복수의 큐드로터들을 사용할 때 중요한 제어 루프에서의 가변 시간 지연을 도입시킨다. 시간 지연은 고도 컨트롤러의 성능을 제한한다. 발명자는 연성 오프-보드 비선형 고도 컨트롤러를 대신하여 강성 온-보드 선형 고도 컨트롤러를 대신 이용하는 것을 선택한다. 발명자는 CPLEX 소프트웨어 패키지의 MIQP 솔버로 모든 문제를 푼다.

[0171] 이러한 실험은 장애물이 구비된 평면 시나리오에서의 3대 운행체를 위한 플래닝을 입증한다. 3대 동종 허밍버드 큐드로터는 좁은 캡의 한 측에서 출발하고 대향 측의 목표 지점으로 지나가야 한다. 궤적은 10차 다항식을 이용하는 본원에서 설명된 방법을 이용하여 그리고 2개의 경유점들 사이의 11 중간 시간 단계에서 충돌 제약을 실행하여 구해진다($n_p = 10$, $n_k = 11$, $n_w = 1$). 큐드로터는 이후 캡을 나타내도록 환경에 위치된 후프로 30번의 시도에 대하여 다양한 속도로 이러한 궤적을 따르도록 명령된다. 이러한 실험을 위한 데이터 및 이미지는 도 15 및 16에 도시되어 있다. 도 15(a)는 이러한 시도 각각에 대한 오차의 제곱 평균 제곱근(RMSE)를 나타낸다. 더 큰 가속도, 저크, 및 스냅이 구비된 궤적은 (예상되는 바와 같이) 더 큰 에러를 야기하며, 성능은 매우 유연

하게 저하된다. 단일 운행(single run)에 대한 데이터는 도 15(b) 내지 15(d)에 보여진다.

[0172] 실험은 kQuad1000(쿼드로터 1) 및 허밍버드(쿼드로터 2)가 캡 아래의 위치로부터 캡 위로 방의 반대편 위치까지 운항하는 것을 보여준다. 이 문제는 13차 다항식을 사용하고 2개의 경유점들 사이의 9개의 중간 시간 단계들에서 충돌 제약을 실시하여 3-D 궤적 생성 문제로서 공식화된다($n_p=13$, $n_k=9$, $n_w=1$). 문제 공식화를 위해, 4개의 3차원 직사각형 프리즘 형상 장애물이 사용되어 단일 3-D 캡을 생성하고, 쿼드로터들은 목표에 도달하기 위해 이 단일 3-D 캡을 통과하여야 한다. 이들 실험을 위한 데이터 및 이미지는 도 17 및 도 18에 도시되어 있다. 더 큰 쿼드로터는 캡을 통과하기 위해 더 엄격한 허용오차를 갖기 때문에, 발명자들은 그것의 비용함수에 허밍버드보다 10배의 가중치를 주도록 선택한다. 이것은 도 17의 쿼드로터 2가 취한 더 우회하는 루트로부터 관찰될 수 있다. 또한, 이것은 쿼드로터 2에서의 더 큰 오차에 의해 관찰될 수 있는데, 이는 쿼드로터 2가 더 큰 속도 및 가속을 요구하는 더 어려운 궤적을 따르기 때문이다. 마지막으로, 더 큰 쿼드로터는 캡을 통해 더 작은 쿼드로터를 따르는 것에 주목하여야 하는데, 그 이유는 더 작은 쿼드로터 아래에서 비행하는 것은 허용되지만 그 반대는 허용되지 않기 때문이다.

[0173] 이 실험은 4개의 쿼드로터가 이루는 팀의 재구성을 보여준다. 이 문제는 9차 다항식을 사용하고 2개의 경유점들 사이의 9개의 중간 시간 단계들에서 충돌 제약을 실시하여 3-D 궤적 생성 문제로서 공식화된다($n_p=9$, $n_k=9$, $n_w=1$). 쿼드로터들을 임의의 위치들 사이에서 소정의 3차원 포메이션으로 또는 완전히 상이한 포메이션으로 원활하고 신속하게 트랜지션하는 궤적들이 생성된다. 발명자들은 도 19 및 도 20에서 몇 가지 재구성 및 라인 포메이션 내에서의 단일 트랜지션을 제시한다. 발명자들은 4개의 허밍버드 및 2개의 허밍버드(하나는 kQuad65이고 다른 하나는 kQuad1000)로 구성된 이종 팀을 사용하여 실험을 진행하였다. 이종 그룹에서, 발명자들은 kQuad65의 비용에 다른 쿼드로터보다 10배 큰 가중치를 주었는데, 그 이유는 그것이 최소로 민첩하고 당장은 적당히 과감한 궤적만을 따르기 때문이다. 도 19(b)에서 kQuad65이 어떻게 가장 짧은 궤적을 취하는지 주목하자. 도 19(a)에 도시된 동종 실험에서, 쿼드로터들은 동일한 평면 상에 머무르는데, 그 이유는 설명된 것과 같이 이들이 서로의 아래에서 비행하는 것이 허용되지 않기 때문이다. 도 19(b)에 도시된 이종 실험에서 최적 해법은 z 성분을 포함하는데, 그 이유는 더 큰 쿼드로터들은 더 작은 것들 아래에서 비행하는 것이 허용되기 때문이다.

[0174] 몇몇 문제의 상세 및 본원에서 해결되는 MIQP 각각에 대한 계산 시간이 아래에 표에 표시된다.

Fig.	n_g	n_p	N_k	n_b	T_1 (s)	T_{opt} (s)
14(b)	1	15	16	208	0.42	35
17	2	13	9	270	0.62	1230
15	3	10	11	300	0.21	553
19(a)	4	9	9	324	0.11	39
19(b)	4	9	9	324	0.45	540

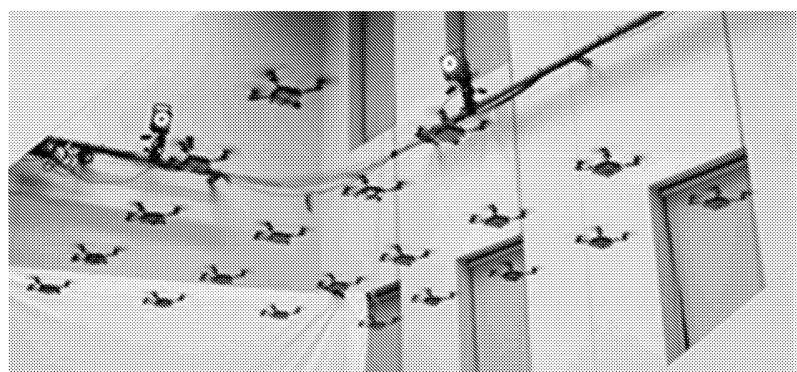
[0175]

[0176] 2.66GHz 인텔 코어 2 듀오 프로세서를 가진 맥북 프로 랩탑에서 CPLEX MIQP 솔버를 사용하여 모든 계산 시간이 열거된다. 특정 문제들은 최적의 해법을 찾고 최적으로 입증하는데 긴 시간이 걸리지만, 제1 해법을 찾는 것은 항상 제2 해법보다 짧다. 솔버는 제1 실현 가능한 답을 찾은 후에는 어느 때고 중단되고 하위 최적 해법으로 복귀할 수 있다.

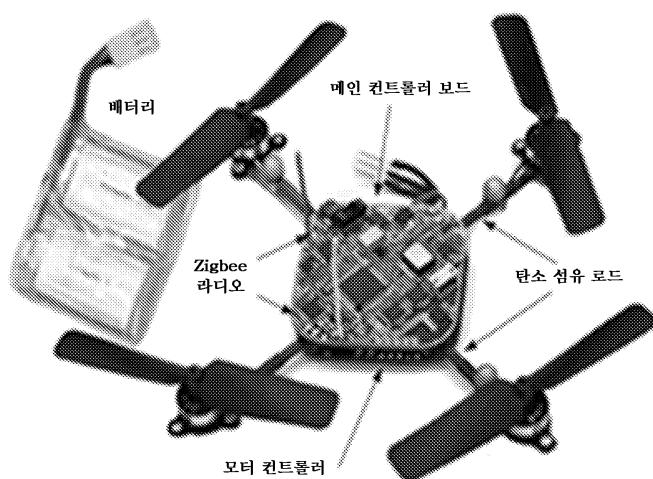
[0177] 통상의 기술자는 본 발명의 새로운 교시 및 이점을 실질적으로 벗어나지 않고 많은 추가적인 수정 및 시나리오가 가능하다는 것을 용이하게 이해할 것이다. 따라서, 그러한 어떠한 수정도 이하의 청구범위에 의해 정의된 본 발명의 범위 내에 포함되는 것이 의도된다.

도면

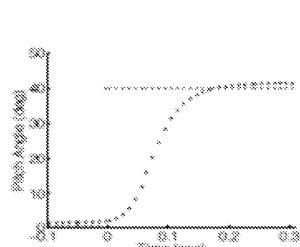
도면1



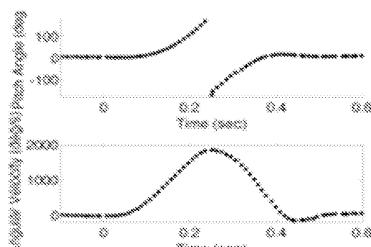
도면2



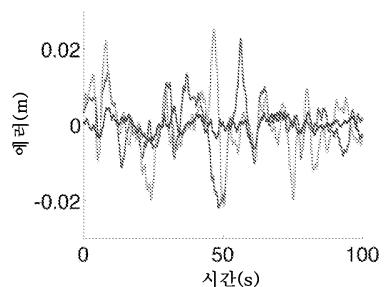
도면3



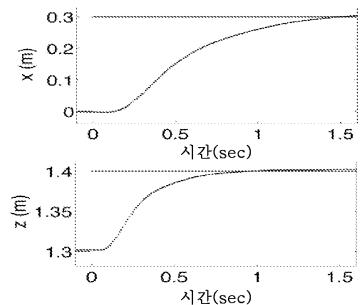
(a) 괴치 각도 단계 입력 반응



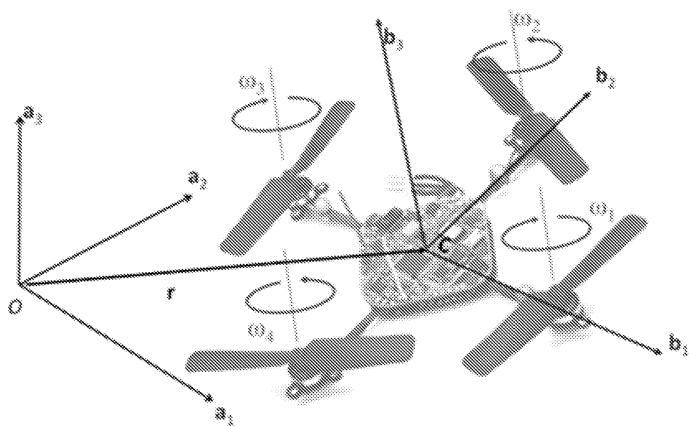
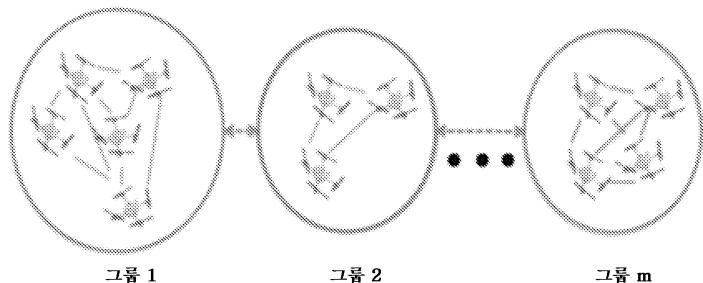
(b) 플립핑 조작을 위한 데이터

도면4

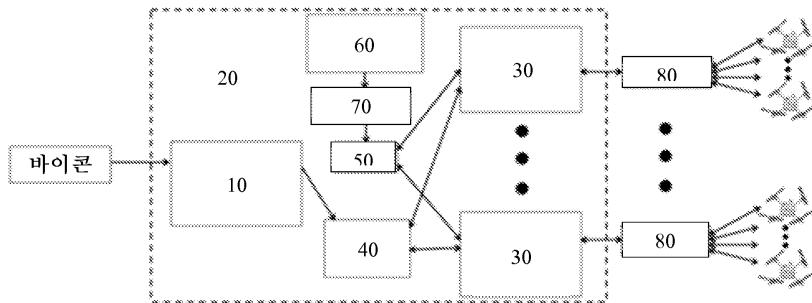
(a) 위치 에러



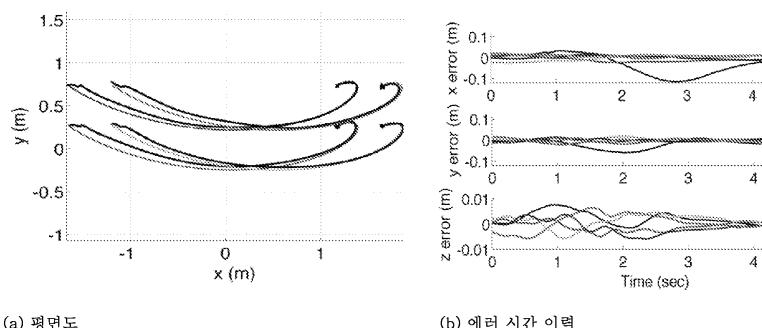
(b) 위치 단계 입력 응답

도면5**도면6**

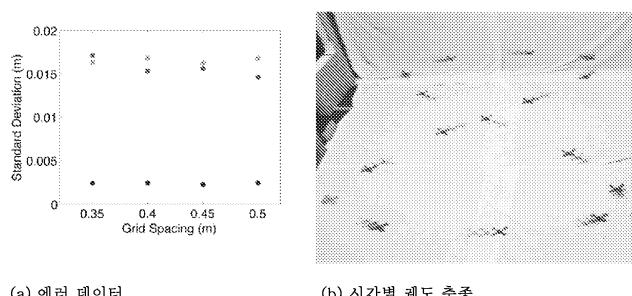
도면7



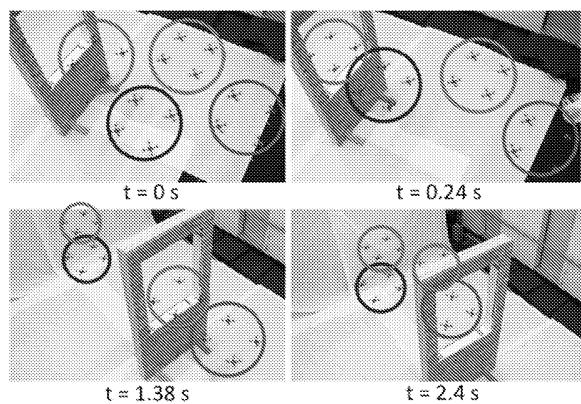
도면8



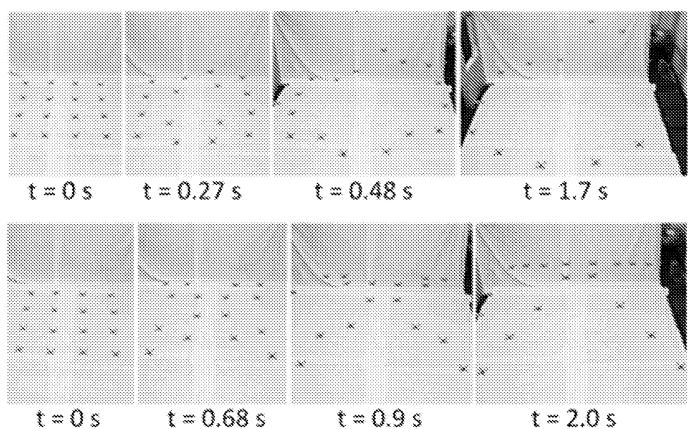
도면9



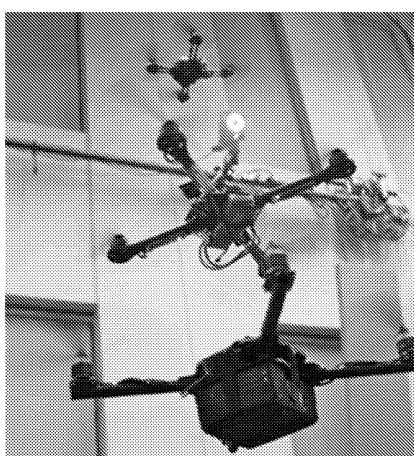
도면10



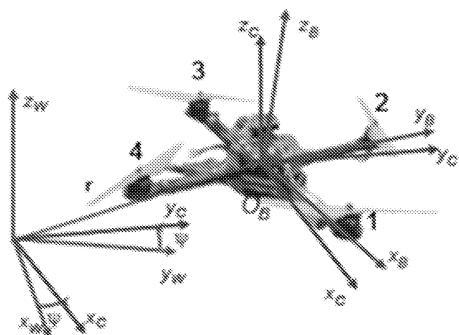
도면11



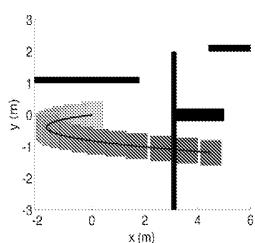
도면12



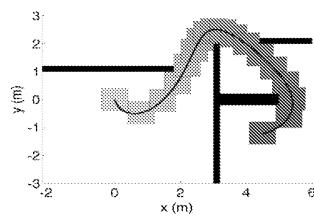
도면13



도면14

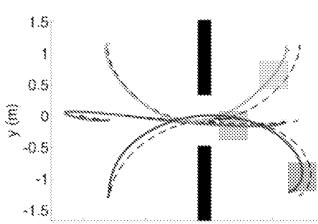
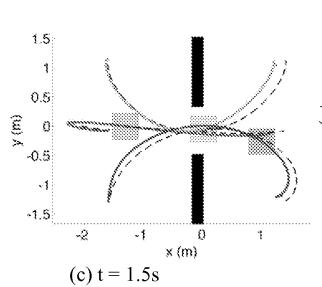
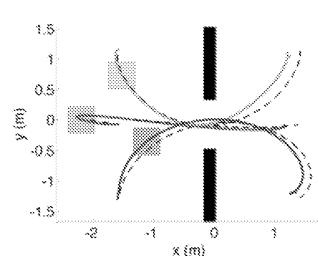
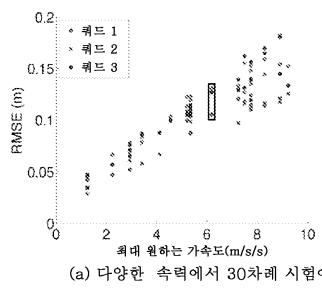


(a) 시간 단계 중첩 제약이 실행되지 않음

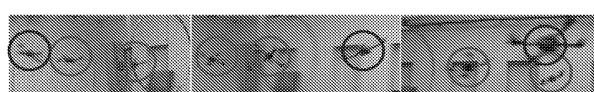


(b) 시간 단계 중첩 제약이 실행됨

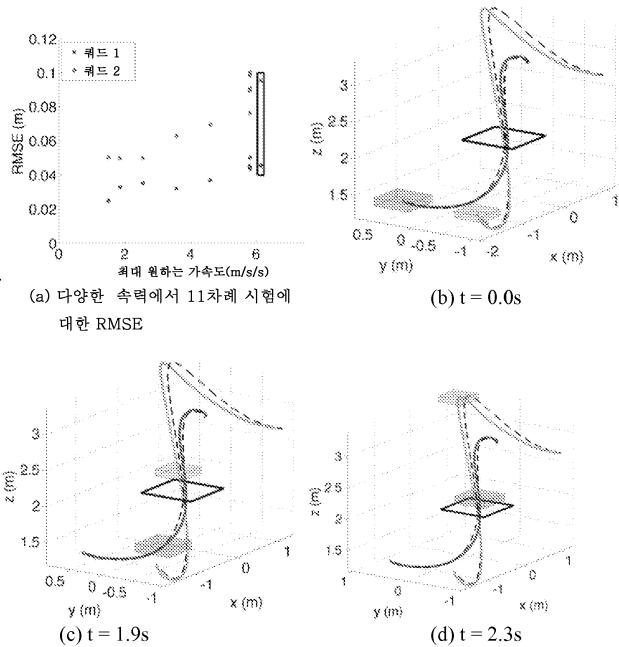
도면15



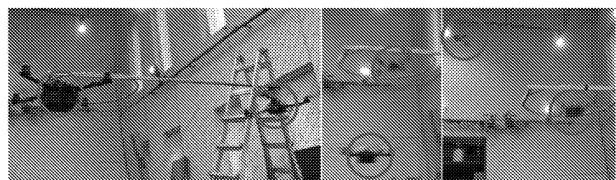
도면16



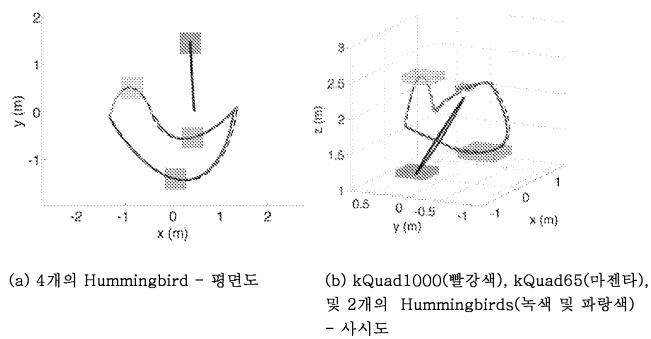
도면17



도면18



도면19



도면20

