

# Dual Quaternion 기반 관성항법 시뮬레이터

## SDINS Simulator Based On Dual Quaternions

심주영\*, 이형근(한국항공대학교),  
이한성, 박찬국(서울대학교), 유명중(국방과학연구소)

### 1. 요약

이체 쿼터니언(dual quaternion)은 선형 미분방정식으로 계산하는 쿼터니언을 회전운동과 병진운동을 동시에 취급하는 이체수(dual numbers) 체계로 표현한 형태이다. 성능향상을 위해 이체 쿼터니언을 관성항법 알고리즘에 적용시킨 논문이 새로이 소개되었다. 기존 알고리즘과 새로운 알고리즘을 MATLAB으로 시뮬레이터를 작성하였으며, 정확도와 실행 속도 측면에서 성능을 평가하고자 한다.

### 2. 명명법

- $f$  : 비력(specific force)
- $g$  : 중력 가속도(gravitational acceleration)
- $\omega_{C_1 C_2}$  :  $C_1$ 좌표에서  $C_2$ 좌표로의 각속도
- $q_{C_1}^{C_2}$  :  $C_1$ 좌표에서  $C_2$ 좌표로의 쿼터니언
- $\hat{q}_{C_1}^{C_2}$  :  $C_1$ 좌표에서  $C_2$ 좌표로의 이체쿼터니언
- $\hat{\omega}$  : 회속도
- $\hat{\sigma}$  : 스크류 벡터
- : 쿼터니언 곱셈

### 3. 서론

기존의 SDINS(Strapdown Inertial Navigation System) 알고리즘은 각속도 및 비력 측정치를 활용하여 항체의 회전과 병진운동을 순차적으로 고려하여 위치, 속도, 및 자세정보를 계산하는 구조를 가진다. 이를 위하여 항체의 운동은 관성 좌표계(Inertial frame; I-frame), 지구좌표계(Earth frame; E-frame), 지역수평 항법좌표계(Navigation frame; N-frame), 그리고 동체좌표계(Body frame; B-frame) 등에 의하여 회전운동과 병진운동으로 나누어 표현된다. 널리 알려진 바와 같이, 항체의 자세 계산에 있어

서는 연산량이 적고, 선형 미분방정식에 기인하여 수치적인 안정성이 우수한 쿼터니언이 SDINS에서 많이 활용되고 있다.

회전운동과 병진운동을 별도로 고려하는 기존 SDINS 알고리즘의 개념적 번거로움과 수식 표현의 간편화를 위하여 근래에 들어 이체수 체계와 쿼터니언의 장점을 활용하는 이체 쿼터니언 기반 SDINS 알고리즘이 새로이 소개되었다[1]. 기존의 연구에 의하여 알려진 바와 같이 쿼터니언은 항체의 회전을 선형 미분방정식으로 계산하는 장점을 가진다. 이에 덧붙여, 이체수 체계는 회전운동량과 병진운동량을 표현하는 이체수와 회전축과 모멘트축을 표현하는 이체벡터로 각 좌표간의 회전과 병진운동을 동시에 표현하여 주는 장점이 있다.

외견상 이체 쿼터니언 기반 알고리즘은 항체의 회전 및 병진운동을 선형 미분 방정식에 의하여 동시에 묘사하여 주므로 SDINS 성능향상에 기여하리라 예상된다. 이를 확인하기 위하여 본 논문에서는 기존 SDINS 알고리즘과 이체 쿼터니언에 의한 SDINS 알고리즘을 MATLAB의 SIMULINK로 SDINS 시뮬레이터를 각각 작성하고, 이를 기반으로 두 알고리즘의 정확도 및 실행 속도를 비교하고자 한다.

### 4. 이체 쿼터니언

#### 4.1. 이체 쿼터니언

이체 쿼터니언은 식 (1)과 같이 실수부의 기존의 회전운동을 나타내는 쿼터니언과 이체부의 회전운동과 병진운동이 조합된 쿼터니언으로 구성된다.  $t$ 는 병진운동에 해당하며, 위치 또는 속도로 표현된다. 이체부의 쿼터니언에서 병진운동 부분은 식 (2)에서 보는 바와 같이 실수부의 쿼터니언과 이체부의 쿼터니언의 곱으로 표현된다.[1]

$$\begin{aligned}\hat{q} &= q + \epsilon q' \\ &= q + \epsilon \frac{1}{2} t^O \circ q = q + \epsilon \frac{1}{2} q \circ t^N\end{aligned}\quad (1)$$

$$t^O = 2q' \circ q^*, t^N = 2q^* \circ q' \quad (2)$$

$$\hat{q} = \cos\left(\frac{\hat{\theta}}{2}\right) + \sin\left(\frac{\hat{\theta}}{2}\right)\hat{n} \quad (3)$$

식 (1)은 이체수와 이체벡터간의 관계에서 유도되며, 식 (3)은 식(1)에서 유도되며, 형태는 기존의 쿼터니언과 동일하다.[1]

식 (1)로부터 쿼터니언 미분방정식을 유도하면 다음과 같이 나타낸다.[1]

$$\dot{\hat{q}}_{ON}^O(t) = \frac{1}{2}\hat{q}_{ON}^O(t) \circ \hat{\omega}_{ON}^N(t) \quad (4)$$

$$\hat{\omega}_{ON}^N = \omega_{ON}^N + \epsilon(\dot{t}^N + \omega_{ON}^N \times t^N) \quad (5)$$

식 (4)의 쿼터니언 미분방정식은 기존의 형태와 동일하다. 식 (5)의 회속도는 실수부의 각속도 성분과 이체부의 회전운동과 병진운동이 조합된 성분으로 구성된다. 이체부는 코리올리 방정식의 형태로 기준 좌표에서의 속도 또는 가속도를 나타낸다.

#### 4.2. 이체 쿼터니언에 의한 SDINS 방정식

이체 쿼터니언은 회전과 병진운동을 동시에 다루므로, 회전에 덧붙여 병진을 포함한 새로운 좌표계의 개념이 선행되어야 한다. 추력 속도 좌표계(Thrust velocity frame; T-frame), 중력 속도 좌표계(Gravitational velocity frame; G-frame), 위치 기반 좌표계(Updated position frame; U-frame)는 이체 쿼터니언 기반 SDINS 알고리즘을 위하여 새로이 정의된 일종의 확장 좌표계들로서 실수부에는 동체 좌표계, 중력속도 좌표계, 그리고 지구좌표계에 대한 회전 성분이 나타나며, 이체부는 비력, 누적 중력 가속도, 항체 운동 속도에 대한 정보가 포괄적으로 포함되어 있다.

이체 쿼터니언을 활용하여 항체의 자세, 속도, 그리고 위치를 다루기 위해서는 기존의 알고리즘과 다른 개념으로 속도 성분을 분해하여야 한다. 새로운 개념의 추력 속도(thrust velocity)와 중력 속도(gravitational velocity)를 활용하여 항체의 속도 및 위치를 표현하면 다음과 같다.

$$v_t^I = \int_0^t f^I(\tau) d\tau \quad (6)$$

$$v_g^I = r^I(0) + \int_0^t g^I(\tau) d\tau \quad (7)$$

$$r^I = r^I(0) + \int_0^t (v_t^I(\tau) + v_g^I(\tau)) d\tau \quad (8)$$

식 (6)과 식(7)에 나타난 바와 같이 추력 속도  $v_t^I$ 는 비력(specific force)을 적분한 속도 성분을 나타내며 중력 속도  $v_g^I$ 는 중력 가속도 (gravitational acceleration)를 적분한 속도 성분을 나타낸다.  $v_t^I$ 와  $v_g^I$  두 속도의 합으로 항체 운동의 속도 성분을 나타내며 이를 다시 적분하여 위치를 계산한다.

자세, 속도, 그리고 위치 정보의 갱신을 위해 이체 쿼터니언의 미분방정식을 활용하기 위한 회속도  $\hat{\omega}$ 가 정의되어야 한다. 각 좌표계의 이체 쿼터니언을 표현하면 다음과 같다.

$$\begin{aligned}\dot{\hat{q}}_{IT}^I(t) &= \frac{1}{2}\hat{q}_{IT}^I(t) \circ \hat{\omega}_{IT}^T(t), \\ \hat{\omega}_{IT}^T &= \omega_{IB}^B + \epsilon f^B, v_t^I = 2q_{IT}^{I'} \circ q_{IT}^{I*}\end{aligned}\quad (9)$$

$$\begin{aligned}\dot{\hat{q}}_{IG}^I(t) &= \frac{1}{2}\hat{q}_{IG}^I(t) \circ \hat{\omega}_{IG}^G(t), \\ \hat{\omega}_{IG}^G &= \omega_{IE}^E + \epsilon g^E, v_g^I = 2q_{IG}^{I'} \circ q_{IG}^{I*}\end{aligned}\quad (10)$$

$$\begin{aligned}\dot{\hat{q}}_{IU}^I(t) &= \frac{1}{2}\hat{q}_{IU}^I(t) \circ \hat{\omega}_{IU}^U(t), \\ \hat{\omega}_{IU}^U &= \omega_{IE}^E + \epsilon(q_{IU}^{I*} \circ (v_t^I + v_g^I) \circ q_{IU}^I) \\ v^N &= q_{IU}^{I*} \circ (v_t^I + v_g^I) \circ q_{IU}^I + r^N \times \omega_{IE}^E \\ r^E &= r^U = 2q_{IU}^{I*} \circ q_{IU}^{I'}\end{aligned}\quad (11)$$

자세정보에 대해서만 기존 좌표계와 확장좌표계를 비교하면, I-frame은 동일하며, T-frame는 B frame과 일치하며, G-frame과 U-frame은 E frame과 일치한다. 위의 세 개의 미분방정식으로 인해 이체 쿼터니언 기반의 항법시스템에서 정보를 갱신할 수 있다.

위치정보 위도, 경도, 고도는  $r^E$ 로부터 구할 수 있다. 자세정보 roll, pitch, yaw는 식(9), (10)에서의 회전성분과 위도와 경도 정보로 알 수 있는 지구좌표계(E-frame)와 지역수평 항법 좌표계(N-frame)의 회전성분으로부터 구한다.

#### 4.3. 이산 시스템에서의 이체 쿼터니언 갱신

일반적인 이체 쿼터니언 갱신은 다음과 같다.

$$\hat{q}(t + \Delta t) = \hat{q}(t) \circ \hat{q}(\Delta t) \quad (12)$$

$\hat{q}(\Delta t)$ 는 현재 구간에서 갱신되는 이체 쿼터니언이며, 스크류 벡터의 함수로 표현된다.

$$\hat{q}(\Delta t) = \left[ \cos\left(\frac{\hat{\sigma}}{2}\right), \frac{\hat{\sigma}}{\sigma} \sin\left(\frac{\Delta t}{2}\right) \right] \quad (13)$$

회속도와 스크류 벡터의 관계는 각속도와 회전 벡터의 관계와 동일하다.[1] 갱신구간에서는 회속도와 스크류 벡터를 선형관계로 가정한다.

### 5. 시뮬레이션

시뮬레이션은 MATLAB 6.5의 SIMULINK를 이용하여 작성하였다. 전체적인 구성은 경로 생성 부분, 센서 부분, 항법컴퓨터 부분, 그리고 오차 계산 부분으로 나뉜다.

경로 생성 부분에서는 연속 시간 시스템이며, 시간에 따른 경로를 입력되면서, 비력( $f^B$ )과 각속도( $\omega_{IB}^B$ )값을 출력한다. 자세(roll, pitch, yaw), 속도(north, east, down), 위치(latitude, longitude, height)의 참값도 출력한다. 각 자세, 속도, 위치의 미분 방정식은 MATLAB에서 제공하는 수치적분(ode45; 4th or 5th Runge-Kuta Method)을 사용하였다.

센서 부분은 센서의 주파수에 맞추어 구간의 적분값을 출력한다.

항법 컴퓨터 부분은 이산 시간 시스템이며, 실행 주파수에 따라 센서 데이터를 입력받아 해당 알고리즘으로 위치, 속도, 자세를 계산하여 출력한다.

오차 부분은 항법컴퓨터 부분에서 생성된 자세, 속도, 위치 데이터와 경로 생성 부분에서 생성된 참값의 차인 오차를 계산하고 저장한다.

센서의 데이터 출력 주파수는 100Hz이며, 항법 컴퓨터의 실행 주파수는 50Hz이다. 코닝, 스킵링 보정은 2샘플 보정을 사용한다. 코닝 모션은 코닝 주파수 1Hz, 코닝 각도는 0.5로 설정하였다. 센서 출력값에 노이즈와 초기 오차가 없는 상태로 실행하였으며, 초기 위치는 위도 37.5°, 경도 127°로 설정하였다. 경로는 정지 상태를 유지하다가 120~150초 구간에서 2m/s<sup>2</sup>로 가속하고, 150~420초 구간에서는 S턴을 한다.

실행 속도 측정은 MATLAB에서 제공하는 프로파일러를 사용하였고, 이 부분은 항법 컴퓨터 부분의 실행시간 데이터를 근거로 평가한다.

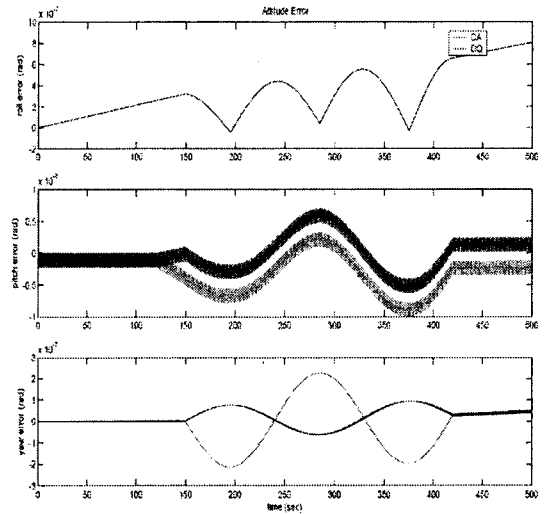


그림 1. 자세 오차

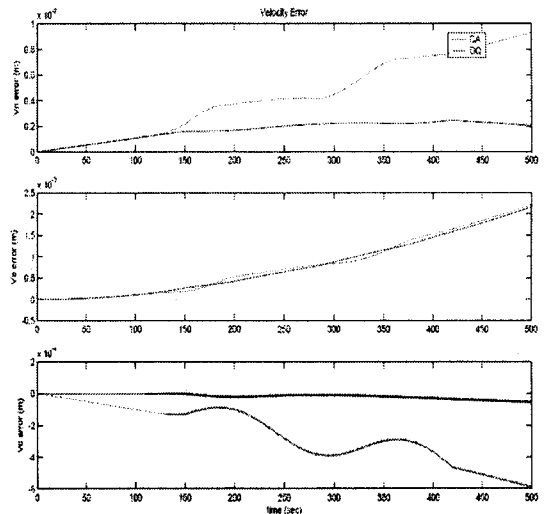


그림 2. 속도 오차

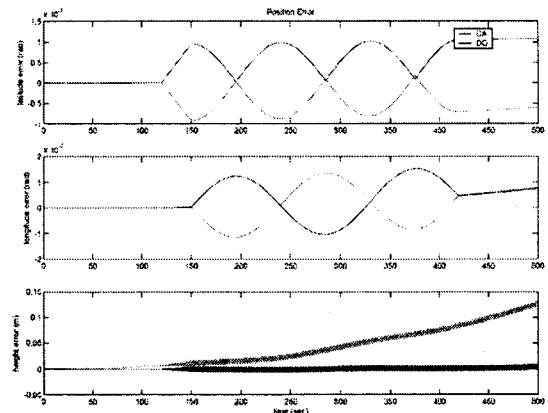


그림 3. 위치 오차

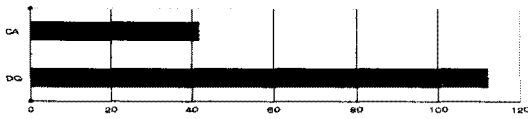


그림 4. 실행 시간

그림 1, 2, 그리고 3의 오차 그래프들은 x축이 경과 시간을 나타내며, y축은 오차량을 나타낸다. 그림 1의 자세 오차 부분에서 롤(roll) 오차는 전체적으로 증가하는 경향을 보이고, 피치(pitch)와 요(yaw) 오차는 주기적인 형태를 보인다. 이것은 코닝 모션에 의한 것으로, 이산 시간 시스템에서 갱신하면서 휘속도와 스칼라 벡터를 선형관계로 가정하면서 생긴 것이다. 그림 1에 나타난 두 알고리즘의 미소한 차이는 속도와 위치 오차에 기인한 것으로 보인다.

그림 2와 그림 3에서 북쪽과 고도 방향 속도 오차와 고도 오차가 이체 쿼터니언 기반 알고리즘에서 향상된 성능이 보인다. 이유는 이체 쿼터니언 알고리즘에서 병진운동 부분도 회전운동과 결합된 쿼터니언 형태로 계산되기 때문이다.

실행 시간은 프로파일러 기능을 활성화시키고, 동일한 경로와 시간 설정으로 알고리즘을 실행할 때, 각각 항법 컴퓨터 부분이 실행된 시간을 나타낸 것이다. 이체 쿼터니언 알고리즘 쪽이 약 2.7배 정도 더 걸린 것을 확인할 수 있다.

## 6. 결론

본 논문에서는 새로이 소개된 이체 쿼터니언 기반의 관성항법 알고리즘에 대해 시뮬레이션을 작성하고, 성능을 평가해 보았다. 위의 결과에서 알 수 있듯이, 이체 쿼터니언 알고리즘이 기존 알고리즘에 비하여 다소 향상된 성능을 보인다. 이는 자세, 속도, 위치 정보를 갱신할 때, 이체 쿼터니언은 회전운동과 병진운동을 선형 미분 방정식에 의하여 동시에 갱신함에 기인한다. 그러나, 이체 쿼터니언 알고리즘은 기존 알고리즘에 비해 실행 시간이 긴 단점이 있음을 확인할 수 있었다. 이와 관련해서는 추가적인 연구가 필요하리라 예상된다.

## 참고문헌

- [1] Wu, Y. X., "Strapdown Inertial Navigation System Algorithms Based on Dual Quaternions," *IEEE Transactions on Aerospace and Electronic Systems*, 41 (Jan.

2005), 110–132.

[2] Savage, P. G., "Strapdown inertial navigation integration algorithm design, Part 1: attitude algorithms," *Journal of Guidance, Control, and Dynamics*, 21, 1, (Jan.–Feb. 1998), 19–28.

[3] Savage, P. G., "Strapdown inertial navigation integration algorithm design, Part 2: velocity and position algorithms," *Journal of Guidance, Control, and Dynamics*, 21, 2, (Mar.–Apr. 1998), 208–221.

[4] Keler, M. L., "On the theory of screws and the dual method," In *Proceedings of A Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of "A Treatise on the Theory of Screws,"* University of Cambridge, Trinity College, July 9–11, 2000.

[5] Miller, R., "A new strapdown attitude algorithm," *Journal of Guidance, Control, and Dynamics*, 6, 4 (July–Aug. 1983), 287–291.

[6] Ignagni, M. B., "Duality of optimal strapdown sculling and coning compensation algorithms," *Navigation: Journal of The Institute of Navigation*, 45, 2, (Summer 1998), 85–95.