

최적 스트랩다운 원추 보상 알고리즘

Optimal Strapdown Coning Compensation Algorithm

박 찬 국, 김 광 진, 이 장 규
(Chan Gook Park, Kwang Jin Kim and Jang Gyu Lee)

Abstract : In this paper, an optimal coning compensation algorithm for strapdown system is proposed by minimizing the coning error. The proposed algorithm is derived as a generalized form in that it contains the class of the existing coning algorithms and allows the design of optimal algorithm for various combinations of gyro samples. It is shown the magnitude of resulting algorithm errors depends mainly on the total number of gyro samples including present and previous gyro samples. Based on the results, the proposed algorithm enables the algorithm designers to develop the effective coning compensation algorithm according to their attitude computation specifications with ease. In addition, the multirate method which can efficiently implement the algorithm is presented.

Keywords : quaternion, rotation vector, coning motion, same distance effect, optimal attitude algorithm

I. 서론

스트랩다운 관성항법시스템(StrapDown Inertial Navigation System-이하 SDINS)은 자이로와 가속도계를 항체에 직접 부착시켜 항체의 움직임을 측정하므로 기준좌표계와 동체좌표계 사이의 자세계산이 필수적이다. 일반적으로 자이로의 출력은 항체의 증분각(incremental angle)을 나타내며, 정확한 항체의 자세를 알기 위해서는 항법 컴퓨터에 의하여 보다 빠르고 효율적으로 처리되어야만 한다. 그러나 고주파의 동적 운동에서는 계산상의 문제로 많은 자세오차가 발생하게 된다. 그러므로 스트랩다운 시스템의 효율적인 자세계산 알고리즘의 설계는 매우 중요한 문제이다.

일반적으로 사용되어지는 자세계산 알고리즘으로는 오일러 방법(Euler method), 방향 코사인 방법(direction cosine method), quaternion 방법 등이 있다. 이중 quaternion 방법은 4개의 변수를 가지고 있으며 비특이성(nonsingularity)과 계산 시간상의 장점으로 인하여 현재 가장 많이 사용되고 있다[1]. 이러한 quaternion 방법을 보다 효과적으로 계산하기 위하여 회전벡터(rotation vector)를 이용하게 되었다[2], [3]. 회전벡터를 이용한 quaternion 방법은 센서의 샘플링 사이에 존재하는 비교환 오차(noncommutativity error)를 효과적으로 보상할 수 있는 것으로 알려졌다[4].

SDINS에서 수행되는 자세계산 알고리즘의 이론적인 근거는 Bortz에 의하여 제시되었다[5]. Bortz는 방향 회전벡터의 미분 방정식을 측정에 의한 각속도 벡터와 계산에 의하여 결정되는 비교환 속도벡터의 합으로 표현하였다. 비교환 오차는 자세계산의 주요한 오차중의 하나로 컴퓨터의 처리 속도를 높여줌으로써 감소시킬 수 있으나 한계가 있기 때문에 이를 극복할 효율적인 알고리즘이 요구된다. 높은 주파수의 동적 환경에서 발생하는 원추운동(coning motion) 오차를 보상하기 위한 알고리즘 구성은 최적 자세계산 알고리즘의 토대가 되었다.

SDINS에서 원추운동 보상 알고리즘의 최적화 개념은 Jordan과 Miller에 의하여 처음으로 제시되고 적용되었다[2], [6]. Jordan은 각각의 갱신 루프에 두개의 자이로 샘플

을 사용하는 알고리즘을 제시하였다[6]. 이러한 개념은 그후에 3개와 4개의 자이로 샘플을 사용하는 알고리즘으로 확장되었다[2], [3]. 이전의 연구결과들은 Ignagni에 의하여 정리되어졌으며 온화한 환경(benign environment)에서 알고리즘의 성능이 해석되었다[7]. 이전 샘플(previous sample)을 이용하여 알고리즘의 정확도와 효율을 보다 향상시킨 방법이 Jiang과 Lin에 의하여 소개되었다[4].

지금까지 개발된 대부분의 알고리즘은 모두 Miller에 의하여 제시된 방법을 따르므로 유도 과정이 복잡하고 최적의 알고리즘을 찾기 위해서는 특정한 샘플수에 따라 이러한 과정을 반복해야 하므로 보다 일반적인 알고리즘이 요구되었다. 본 논문에서는 알고리즘을 일반화 시키기 위하여 원추 운동에 의하여 발생하는 부유오차(drift error) 보상식에 동일 거리 효과(same distance effect)를 적용하고, 샘플수에 관계없이 적용할 수 있는 일반화된 식을 제시한다. 일반화된 알고리즘의 증명을 위하여 지금까지 발표된 여러가지 알고리즘을 일반화된 방법으로 유도하여 이전까지의 결과와 비교하였으며, 지금까지 발표되지 않았던 새로운 알고리즘을 유도하고 오차해석을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 회전벡터와 원추운동에 관하여 설명하고 3장에서는 원추운동의 부유오차를 보상하기 위한 식을 유도하고 이 식에 동일 거리 효과와 이전 샘플을 적용하여 일반화된 알고리즘을 유도한다. 4장에서는 일반화된 알고리즘을 이용하여 이전에 발표된 다양한 알고리즘을 쉽게 유도하고 새로운 알고리즘을 제시한다. 마지막으로 5장에서 결론을 맺는다.

II. 회전벡터와 원추운동

스트랩다운 자세계산 알고리즘에서 가장 중요한 연산이 회전벡터와 quaternion을 효율적으로 갱신(update)하는 것이다[2]. Quaternion 갱신 $\bar{Q}(t+h)$ 는 다음의 quaternion 곱으로 정의된다.

$$\bar{Q}(t+h) = \bar{Q}(t) * \bar{q}(h) \tag{1}$$

위 식에서 $\bar{q}(h)$ 는 quaternion 갱신 주기 h 동안에 실제적으로 변화된 quaternion이며 회전벡터 ϕ 에 의하여 다음과 같이 주어진다.

$$\bar{q}(h) = \left(\cos \frac{\phi_0}{2}, \frac{\phi}{\phi_0} \sin \frac{\phi_0}{2} \right) \tag{2}$$

접수일자 : 1996. 4. 15., 수정완료 : 1996. 8. 12.
 박찬국, 김광진 : 광운대학교 제어계측공학과/신기술연구소
 이장규 : 서울대학교 전기공학부 자동제어특화연구센터
 ※ 본 연구는 서울대학교 자동제어특화연구센터의 지원으로 수행된 것입니다.

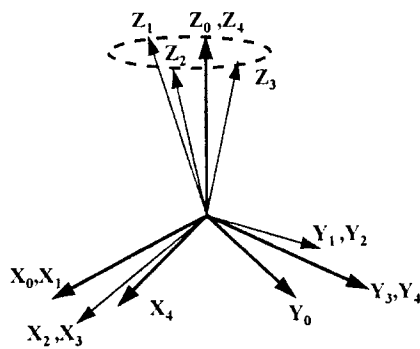
여기서, $\phi_0 = (\phi \cdot \phi)^{\frac{1}{2}}$ 이며 ϕ 는 입력각속도 벡터 ω 와 다 음과 같은 관계를 나타낸다[5], [8].

$$\dot{\phi} = \omega + \frac{1}{2} \phi \times \omega + \frac{1}{12} \phi \times (\phi \times \omega) \quad (3)$$

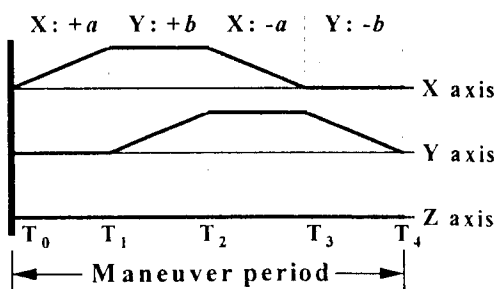
우변의 두번째와 세번째항은 비교환 오차 벡터성분이며 회전벡터를 이용한 스트랩다운 자세계산 알고리즘의 오차를 줄이기 위하여 효과적으로 이용할 수 있다[4].

일반적으로 항체는 운항중 여러 가지 운동을 함으로써 항 법 알고리즘에 오차를 유발하며 이중에서 자세오차에 영향을 미치는 대표적인 항체 운동으로 원추운동을 들 수 있다. 원추운동이란 두개의 위상차가 있는 정현파 운동이 강체의 수직된 두축에 인가되어 이와는 수직된 다른 한축이 원추모 양의 운동을 하는 것이다. 원추운동을 해석하기 위하여 X축 을 +a, Y축 +b, X축 -a, Y축 -b의 크기로 회전을 시키면 그림 1(a)와 같이 모형화 되며 이때의 적분 각속도는 그림 1(b)와 같다. 그림에서 X_0, Y_0, Z_0 는 동체 좌표계의 초기 위 치이며 X축 +a 회전에 의하여 변화된 축이 X_1, Y_1, Z_1 이면 X_0 는 X_1 과 동일하며 Y, Z의 위치가 변하게 된다. 주어진 회 전에 의하여 이 과정을 반복하면 마지막 축의 위치는 X_4, Y_4, Z_4 가 되며 그림 1(a)에서 X, Y축은 초기 위치와 달라져 있으나 Z축은 초기 위치 Z_0 와 동일하며 원추모양의 운동을 하는 것을 알 수 있다. 그림 1(b)에서 X축은 적분 각속도가 초기 위치에서부터 X_1 까지 증가한 후 X_3 에서는 0이 된다. 그리고 Y축 적분 각속도는 Y_1 에서 증가하여 Y_4 에서는 0이 되며 Z축에는 각속도가 입력되지 않는다.

그림 1과 같은 한 주기의 원추운동이 항법 컴퓨터의 계산 주기 이내에 발생하면 항체의 자세는 (a)와 같이 변화되어 있으나 항법 컴퓨터는 (b)와 같이 자세 변화가 없는 것으로 계산한다. 즉 항법 컴퓨터는 이 운동을 인식하지 못하게 되



(a) Motion of axes



(b) Integrated body rate

그림 1. 원추운동에 의한 부유오차.
Fig. 1. Drift error by the coning motion.

며 원추운동에 의한 자세 오차가 부유오차로 발생하게 된다 [9]. 이 오차는 계산시 자이로 출력만으로는 보상되지 못하고 계산을 통하여 보상되는 비교환 오차 성분으로 원추운동 은 자세계산에 가장 큰 오차를 발생시키는 운동이기 때문에 최적의 자세계산 알고리즘이라 함은 이 운동으로부터 유도 된다[4].

항체가 원추운동을 할 때 각속도 벡터식은 그림 1을 이용 하여 다음과 같이 정의할 수 있다.

$$\omega = a\Omega \cos \Omega\tau I + b\Omega \sin \Omega\tau J \quad (4)$$

여기서, ω = 동체 좌표계로 표시된 각속도 벡터

a, b = 동체의 수직된 두축의 각의 진폭

Ω = 원추운동의 각속도

I, J = 진동이 발생하는 두 축의 단위벡터

두 축 I, J 에 (4)의 입력이 들어가면 항체는 K 축을 중 심으로 원추(cone) 모양을 그리는 운동을 하게된다. 이때 원추운동의 회전각은 각속도 벡터의 적분에 의하여 다음과 같이 정의된다.

$$\begin{aligned} \alpha(\tau, \tau_{m-1}) &= \int_{\tau_{m-1}}^{\tau} \omega dt \\ &= a(\sin \Omega\tau - \sin \Omega\tau_{m-1})I \\ &\quad + b(\cos \Omega\tau_{m-1} - \cos \Omega\tau)J \end{aligned} \quad (5)$$

높은 주파수의 원추운동을 보다 정확하게 추정하고 보상 하기 위하여 원추운동 보상은 빠른 속도로 계산하고 자세변 수들의 보정은 상대적으로 느린 속도로 계산하므로써 효율 적인 알고리즘을 구현할 수 있다. 이 관계를 그림으로 표현 하면 그림 2와 같다. 그림 2에서 미소구간(subminor interval)이 자이로의 샘플링 구간이며 소구간(minor interval)은 회전벡터를 계산하는 구간이다. 주구간(major interval)은 M 번의 소구간 동안 계산된 회전벡터를 누적하여 quaternion을 갱신하는 구간이지만 소구간과 일치하여 quaternion을 갱신할 수도 있다.

자세갱신기간(attitude updating interval)인 t_{n-1} 부터 t_n 사이에 발생한 원추운동의 보정은 부유오차의 적분에 의하여 정의되며, 원추운동 보상 $\Delta\phi(n)$ 은 다음과 같다[7].

$$\Delta\phi(n) = \frac{1}{2} \int_{t_{n-1}}^{t_n} \alpha(t, t_{n-1}) \times \omega dt \quad (6)$$

여기서 ω 는 세개의 수직된 입력축을 가지는 자이로에 의하여 측정되는 각속도이다.

(6)은 Bortz[5]에 의하여 정의된 비교환 오차식과는 다른 원추운동에 의한 비교환 오차의 간략화된 식으로써 (3)의 두번째와 세번째항에 해당하는 부분이며 회전벡터와 다음과 같은 관계를 가진다[4], [7].

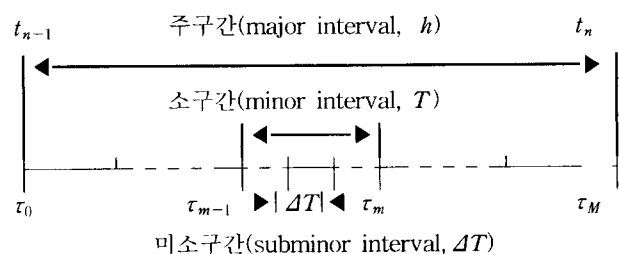


그림 2. 원추 보상 구간.

Fig. 2. Intervals associated with coning compensation.

$$\phi_n = \int_{t_{n-1}}^{t_n} \omega dt + \Delta\phi(n) \quad (7)$$

(6)과 같이 정의된 원추운동 보정은 미소구간 동안 얻은 자이로 출력에 이용하여 소구간이 끝날때마다 이루어지며, M 번의 소구간으로 이루어진 주구간이 지난 후 quaternion과 자세 변수들은 갱신된다. 즉, 주구간 동안의 원추운동 보정은 M 개의 소구간 동안 계산된 것의 합으로 구해진다. (6)을 그림 1에 표시된 소구간의 시간 간격으로 다시 표현하면 다음과 같다.

$$\begin{aligned} \Delta\phi(n) &= \frac{1}{2} \int_{t_{n-1}}^{t_n} \alpha(t, t_{n-1}) \times \omega dt \\ &= \sum_{m=1}^M \frac{1}{2} \int_{\tau_{m-1}}^{\tau_m} \alpha(t_{n-1} + \tau, t_{n-1}) \times \omega d\tau \\ &= \frac{1}{2} \sum_{m=2}^M \alpha_{m-1} \times \Delta\theta_m \\ &\quad + \frac{1}{2} \sum_{m=1}^M \int_{\tau_{m-1}}^{\tau_m} \alpha(\tau, \tau_{m-1}) \times \omega d\tau \end{aligned} \quad (8)$$

여기서, $\Delta\theta_m = \alpha(\tau_m, \tau_{m-1}) = \int_{\tau_{m-1}}^{\tau_m} \omega d\tau$,

$$\alpha_{m-1} = \int_{t_{n-1}}^{t_{n-1} + \tau_{m-1}} \omega dt = \sum_{i=1}^{m-1} \Delta\theta_i = \sum_{i=1}^{m-1} \int_{\tau_{i-1}}^{\tau_i} \omega dt$$

(8)에서 첫번째 항은 각속도 적분 자이로(rate-integrating gyro)의 출력으로 직접 구할 수 있으므로 오차에는 영향을 미치지 않으나, 두번째 항은 적분에 의하여 계산되는 항이므로 오차를 유발하게 된다. 그러므로 소구간이 지난후의 원추운동 보상 적분은 (8)의 두번째 항에 의하여 다음과 같이 정의된다.

$$\delta\phi_m = \frac{1}{2} \int_{\tau_{m-1}}^{\tau_m} \alpha(\tau, \tau_{m-1}) \times \omega d\tau \quad (9)$$

위 식이 오차 보상식을 결정하는 가장 중요한 식이다. 이제 원추운동 (4)와 (5)를 (9)에 대입하면 원추 보상식의 정확한 형태를 얻을 수 있다.

$$\delta\phi_m = \frac{ab\Omega}{2} \left(T - \frac{1}{\Omega} \sin\Omega T \right) K \quad (10)$$

(10)은 원추운동 보상식이 구간 시작 시간에 관계없이 소구간의 주기 T 와 원추운동의 각속도 Ω 에 의해서만 영향을 받고 있음을 나타내고 있다. 그러나 실제로는 원추운동의 각속도를 정확히 알 수 없으므로 (10)을 자세계산 알고리즘에 직접 사용할 수 없고 근사화된 오차 보상식이 요구된다. 또한 원추운동에 의하여 발생한 오차를 보상하지 않을 경우 그림 1과 같은 부유오차가 자세오차로 남게 되며 오차의 크기는 (10)과 같다.

근사화된 보상식을 만들기 위하여 (10)의 기하학적인 의미를 해석한다. (10)은 원추운동에 의한 부유오차를 소구간 동안 보상하는 식으로 순환운동(cyclic motion)을 하고있는 K 방향의 벡터가 발생시키는 넓이에 비례한다[10]. 보상 넓이의 근사화는 자이로 출력으로 삼각형의 면적을 계산하는 것으로서 이를 이용하여 원추운동 보상 알고리즘을 만들게 된다. k 번째의 자이로 출력은 다음과 같이 구해진다.

$$\begin{aligned} \Delta\theta(k) &= \alpha(t_k, t_{k-1}) = \int_{t_{k-1}}^{t_k} \omega dt \\ &= \int_{t_{k-1}}^{t_{k-1} + \Delta T} \Omega (a \cos \Omega t I + b \sin \Omega t J) dt \\ &= a(\eta \sin \Omega t_{k-1} + \nu \cos \Omega t_{k-1}) I \\ &\quad + b(\nu \sin \Omega t_{k-1} - \eta \cos \Omega t_{k-1}) J \end{aligned} \quad (11)$$

여기서, $\eta = \cos\lambda - 1$, $\nu = \sin\lambda$, $\lambda = \Omega\Delta T$.

i 번째와 j 번째의 서로 다른 자이로 출력의 외적을 계산하면 다음과 같다.

$$\begin{aligned} \Delta\theta(i) \times \Delta\theta(j) &= ab\{2\sin[(j-i)\lambda] \\ &\quad - \sin[(j-i+1)\lambda] - \sin[(j-i-1)\lambda]\} K \end{aligned} \quad (12)$$

(12)는 두개의 증분 각벡터의 곱이 정확한 오차보상식 처럼 상수이고 계산 주기 T 에만 관계된다는 것을 보여주고 있다. 이는 (12)의 변형에 의하여 (10)과 유사한 식을 만들 수 있다는 것을 의미하며 근사식의 토대가 된다. (12)를 이용하여 자이로 출력에 의한 근사화된 오차보상식을 정의하면 다음과 같다[4].

$$\delta\hat{\phi}_m = \sum_{i=1}^{m-1} \sum_{j=i+1}^m d_{ij} \Delta\theta_m(i) \times \Delta\theta_m(j) \quad (13)$$

여기서, $\delta\hat{\phi}_m = m$ 번째 소구간 후의 근사화된 알고리즘

$\Delta\theta_m(i) = m$ 번째 소구간의 i 번째 미소구간에서의 증분 각벡터

$n =$ 한 소구간에서의 자이로 출력 횟수

$d_{ij} =$ 자이로 곱의 상수

III. 알고리즘 일반화

앞에서 정의된 (13)은 현재 계산되고 있는 소구간의 자이로 출력만을 이용하여 근사화된 형태이나 이미 얻어진 과거 출력까지도 함께 이용하면 알고리즘의 성능을 보다 향상시킬 수 있다[4]. 이때 알고리즘의 계산량을 줄이고 보다 간략히 하기 위하여 동일 거리 효과를 적용한다. 동일 거리 효과란 자이로 출력의 곱에 의하여 보상되는 근사화된 알고리즘에서 시간 간격이 같은 자이로 출력의 외적들은 원추운동의 경우에 같은 값을 주기 때문에 대표적인 한가지 경우만을 사용하여 알고리즘을 구성해도 동일한 성능을 나타낸다는 것이다. 여기서 거리(distance)라함은 자이로 출력들의 시간 간격을 의미한다[3]. 그러나 유한한 다항식의 형태로 표시되는 온화한 환경에서는 동일거리 효과를 사용하여 간단하게 표현된 원추 보상 알고리즘이 모든 자이로 출력의 외적으로 표현된 알고리즘보다 성능이 다소 떨어질 수 있다[11].

가장 최근 샘플과의 거리(시간 간격의 차)만을 고려하여 m 번째 소구간에 대한 근사화된 원추 보상식 $\delta\hat{\phi}_m$ 은 n 개의 현재 샘플과 p 개의 이전 샘플들에 대하여 다음과 같이 간략히 표현할 수 있다[11].

$$\begin{aligned} \delta\hat{\phi}_m &= \left[\sum_{j=n-p+1}^n k_{2n-j} \Delta\theta_{m-1}(j) \right. \\ &\quad \left. + \sum_{i=1}^{n-1} k_{n-i} \Delta\theta_m(i) \right] \times \Delta\theta_m(n) \end{aligned} \quad (14)$$

여기서, k_{2n-j} , $k_{n-i} = \Delta\theta_m(n)$ 과 거리 $2n-j$, $n-i$ 인 상수 $\Delta\theta_{m-1}(j) = m-1$ 번째 소구간의 j 번째 소구간에서의 증분 각벡터

$\Delta\theta_m(i) = m$ 번째 소구간의 i 번째 미소구간에서의 증분 각벡터

원추운동 보상 알고리즘은 근사화된 (14)에 (12)를 대입하여 Taylor 급수를 전개한 후, 정확한 (10)의 Taylor 급수 전개한 식과 비교하여 상수 k_{2n-j} , k_{n-i} 를 구한다. 이렇게 구한 알고리즘은 동일 거리 효과를 사용하지 않는 것에 비하여 계산량을 크게 감소시킬 수 있으며 원추운동에서 최적의 성능을 나타낸다.

(14)를 전개한 후 (12)를 대입하면 다음과 같다.

$$\begin{aligned} \delta \hat{\phi}_m &= k_{n+p-1} \Delta \theta_{m-1}(n-p+1) \times \Delta \theta_m(n) \\ &+ \dots + k_n \Delta \theta_{m-1}(n) \times \Delta \theta_m(n) \\ &+ k_{n-1} \Delta \theta_m(1) \times \Delta \theta_m(n) \\ &+ \dots + k_1 \Delta \theta_m(n-1) \times \Delta \theta_m(n) \\ &= ab \{ k_{n+p-1} \{ 2\sin(n+p-1)\lambda \\ &\quad - \sin(n+p)\lambda - \sin(n+p-2)\lambda \} \\ &+ \dots + k_n \{ 2\sin n\lambda - \sin(n+1)\lambda - \sin(n-1)\lambda \} \\ &+ k_{n-1} \{ 2\sin(n-1)\lambda - \sin n\lambda - \sin(n-2)\lambda \} \\ &+ \dots + k_1 \{ 2\sin \lambda - \sin 2\lambda - \sin 0\lambda \} \} K \end{aligned} \tag{15}$$

(15)를 Taylor 급수 전개하면 아래와 같다.

$$\begin{aligned} \delta \hat{\phi}_m &= ab \{ [A_{11}k_1 + A_{12}k_2 + \dots \\ &+ A_{1j-1}k_{n+p-2} + A_{1j}k_{n+p-1}] \lambda^3 \\ &- [A_{21}k_1 + A_{22}k_2 + \dots \\ &+ A_{2j-1}k_{n+p-2} + A_{2j}k_{n+p-1}] \lambda^5 + \dots \\ &- (-1)^i [A_{i1}k_1 + A_{i2}k_2 + \dots \\ &+ A_{ij-1}k_{n+p-1} + A_{ij}k_{n+p-1}] \lambda^{2i+1} + \dots \} K \end{aligned} \tag{16}$$

여기서 A_{ij} 는 거리 i 인 k_i 의 Taylor 급수 전개후의 상수로서 (15)의 전개에 의하여 다음과 같은 일반식을 유도할 수 있다.

$$A_{ij} = \frac{(j+1)^{2i+1} - 2 \cdot j^{2i+1} + (j-1)^{2i+1}}{(2i+1)!} \tag{17}$$

(16)의 k_i 을 구하기 위하여 정확한 원추운동 보상식 (10)을 Taylor 급수 전개한다.

$$\begin{aligned} \delta \phi_m &= \frac{ab}{2} \left[\frac{(\Omega T)^3}{3!} - \frac{(\Omega T)^5}{5!} + \dots \right] K \\ &= ab \left[\frac{n^3}{3! \times 2} \lambda^3 - \frac{n^5}{5! \times 2} \lambda^5 + \dots \right] K \\ &= ab [c_1 \lambda^3 - c_2 \lambda^5 + c_3 \lambda^7 - c_4 \lambda^9 \\ &\quad + \dots - (-1)^i c_i \lambda^{(2i+1)} + \dots] K \end{aligned} \tag{18}$$

여기서 c_i 는 정확한 원추운동 보상식의 $2i+1$ 의 차수를 갖는 λ 의 상수항으로 다음과 같이 일반식으로 표현할 수 있다.

$$c_i = \frac{n^{(2i+1)}}{(2i+1)! \times 2} \tag{19}$$

전체 샘플수 $N = p + n$ 이라 정의하면 구하고자 하는 미지 상수 k_i 은 $N-1$ 개가 되고 (16)과 (18)을 정리하면 다음과 같은 선형 연립 방정식의 형태로 간략히 표현할 수 있다.

$$\sum_{i=1}^{N-1} \sum_{d=j=1}^{N-1} k_d \times A_{ij} = \sum_{i=1}^{N-1} c_i \tag{20}$$

즉, (17)과 (19)에서 A_{ij} 와 c_i 를 계산하고 이를 연립한 (20)에서 원하는 알고리즘의 미지 상수인 k_i 을 구한다. 이 값을 (14)에 대입하여 원하는 알고리즘을 구성한다.

(20)의 선형 연립 방정식은 행렬 형태로 바꾸어 표현될 수 있으며 (17)과 (19)에서 전체 샘플수가 7 이하인 경우에 대하여 행렬 형태로 표현하면 다음과 같다.

$N=2$	$N=3$	$N=4$	$N=5$	$N=6$	$N=7$
1	2	3	4	5	6
$\frac{1}{4}$	$\frac{3}{2}$	$\frac{19}{4}$	11	$\frac{85}{4}$	$\frac{73}{2}$
$\frac{1}{40}$	$\frac{23}{60}$	$\frac{289}{120}$	$\frac{283}{30}$	$\frac{667}{24}$	$\frac{4069}{60}$
$\frac{17}{12,096}$	$\frac{311}{6,048}$	$\frac{827}{1,344}$	$\frac{12,071}{3,024}$	$\frac{214,453}{12,096}$	$\frac{41,021}{672}$
$\frac{31}{604,800}$	$\frac{437}{100,800}$	$\frac{58,213}{604,800}$	$\frac{153,851}{604,800}$	$\frac{816,167}{120,960}$	$\frac{9,852,733}{302,400}$
$\frac{1}{760,320}$	$\frac{289}{1,140,480}$	$\frac{23,417}{228,096}$	$\frac{99,641}{570,240}$	$\frac{3,914,431}{2,280,960}$	$\frac{2,636,935}{228,096}$

$$\begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \end{bmatrix} \times \begin{bmatrix} \frac{n^3}{12} \\ \frac{n^5}{240} \\ \frac{n^7}{10,080} \\ \frac{n^9}{725,760} \\ \frac{n^{11}}{79,833,600} \\ \frac{n^{13}}{12,454,041,600} \end{bmatrix} = \tag{21}$$

(21)에서 구하고자 하는 알고리즘의 전체 샘플수에 해당하는 불려를 선정하여 k_i 값들을 구한 후 (14)를 이용하여 알고리즘을 유도할 수 있다.

알고리즘의 오차는 근사화된 보상식(16)과 정확한 원추오차 보상식(18)에서 구할 수 있으며 다음과 같이 정의한다.

$$\epsilon = \delta \hat{\phi}_m - \delta \phi_m \tag{22}$$

앞의 결과를 이용하여 (22)를 계산하면 $\lambda^{2(\rho+n)+1}$ 항부터 오차가 발생한다. 이것은 미지 상수 k_i 값들을 구하기 위해서 λ^3 항부터 $\lambda^{2(\rho+n)-1}$ 항까지 연립하였으므로 그 다음 항부터 오차가 발생하기 때문이다. 오차는 $\lambda^{2(\rho+n)+1}$ 항에서부터 무한대까지의 오차가 발생하며 가장 큰 영향을 미치는 오차항(dominant error)인 $\lambda^{2(\rho+n)+1}$ 항에서의 오차 ϕ_ϵ 은 다음과 같다.

$$\phi_\epsilon = \frac{n \times (\rho+n)!}{2^{(\rho+n+1)} \times \prod_{k=1}^{(\rho+n)+1} (2k-1)} ab \lambda^{2(\rho+n)+1} \tag{23}$$

$$= \frac{n^{-2(\rho+n)} \times (\rho+n)!}{2^{(\rho+n+1)} \times \prod_{k=1}^{(\rho+n)+1} (2k-1)} ab (\Omega T)^{2(\rho+n)+1} \tag{24}$$

(23)은 자이로 샘플 구간인 미소구간 ΔT 에 대하여 표현된 식이고, (24)는 소구간 T 가 일정할 때의 알고리즘 오차를 나타낸 것이다. 위 식에서 알고리즘의 오차는 사용되어지는 전체 샘플수에 의하여 주로 영향을 받는다는 것을 알 수 있다.

IV. 여러가지 알고리즘

회전백터를 갱신할 때 발생하는 비교환 오차는 보상시 사용하는 자이로의 샘플수에 의하여 좌우되므로, 소구간의 샘플수가 4, 5, 6인 경우의 이전 샘플들을 고려한 알고리즘을 구하고 그 오차를 분석한다. 알고리즘은 차례로 (17), (19), (20) 그리고 (14)의 순서로 결정되며 (23)과 (24)에서 오차를 계산한다.

4-샘플 알고리즘

알고리즘 4-0 4-샘플을 사용할 경우 $n=4, p=0$ 이므로

$N=4$ 가 된다. 이 값을 가지고 (21)에서 구한 미지 상수는 $k_1 = \frac{214}{105}$, $k_2 = \frac{92}{105}$, $k_3 = \frac{18}{35}$ 이 되며 알고리즘과 알고리즘 오차는 다음과 같다.

$$\begin{aligned} \delta \hat{\varphi}_m &= \frac{18}{35} \Delta \varrho_m(1) \times \Delta \varrho_m(4) \\ &+ \frac{92}{105} \Delta \varrho_m(2) \times \Delta \varrho_m(4) \\ &+ \frac{214}{105} \Delta \varrho_m(3) \times \Delta \varrho_m(4) \end{aligned} \quad (25)$$

$$\phi_\epsilon = \frac{ab}{315} \lambda^9 = \frac{ab}{82,575,360} (\Omega T)^9 \quad (26)$$

이 결과는 Lee et. al.[3]가 제시한 알고리즘과 동일하며 본 논문에서 제안된 식을 이용하여 간단하게 유도할 수 있다.

알고리즘 4-1 4-샘플에 1개의 이전 샘플을 가지는 알고리즘을 구한다. 이때 $N=5$ 가 되므로 4×4 의 정방행렬을 이용하여 미지 상수 k_l 들을 구하면 $k_1 = \frac{656}{315}$, $k_2 = \frac{262}{315}$, $k_3 = \frac{8}{15}$, $k_4 = -\frac{1}{315}$ 가 되며 이때 알고리즘과 오차를 구하면 다음과 같다.

$$\begin{aligned} \delta \hat{\varphi}_m &= -\frac{1}{315} \Delta \varrho_{m-1}(4) \times \Delta \varrho_m(4) \\ &+ \frac{8}{15} \Delta \varrho_m(1) \times \Delta \varrho_m(4) \\ &+ \frac{262}{315} \Delta \varrho_m(2) \times \Delta \varrho_m(4) \\ &+ \frac{656}{315} \Delta \varrho_m(3) \times \Delta \varrho_m(4) \end{aligned} \quad (27)$$

$$\phi_\epsilon = \frac{ab}{1386} \lambda^{11} = \frac{ab}{5,813,305,344} (\Omega T)^{11} \quad (28)$$

이 결과를 통하여 알 수 있듯이 알고리즘 오차의 차수는 11이 되어 (26)과 비교할 때 차수가 2 이상 증가 되었다. 즉 이전 샘플을 하나만 사용하여도 알고리즘의 성능은 기하 급수적으로 향상 된다는 것을 알 수 있다. 알고리즘 4-1은 Ignagni [11]의 결과와 동일하다.

5-샘플 알고리즘

알고리즘 5-0 5-샘플을 사용할 $n=5$, $p=0$ 이므로 $N=5$ 가 되며 (21)에서 k_l 를 구하면 다음과 같다.

$$k_1 = \frac{1375}{504}, k_2 = \frac{325}{252}, k_3 = \frac{25}{24}, k_4 = \frac{125}{252} \quad (29)$$

각각의 k_l 를 이용하여 알고리즘과 알고리즘 오차를 구하면 다음과 같다.

$$\begin{aligned} \delta \hat{\varphi}_m &= \frac{125}{252} \Delta \varrho_m(1) \times \Delta \varrho_m(5) \\ &+ \frac{25}{24} \Delta \varrho_m(2) \times \Delta \varrho_m(5) \\ &+ \frac{325}{252} \Delta \varrho_m(3) \times \Delta \varrho_m(5) \\ &+ \frac{1375}{504} \Delta \varrho_m(4) \times \Delta \varrho_m(5) \end{aligned} \quad (30)$$

$$\phi_\epsilon = \frac{5ab}{5544} \lambda^{11} = \frac{ab}{54,140,625,000} (\Omega T)^{11} \quad (31)$$

5-샘플 알고리즘은 (30)과 같으며 Ignagni [11]의 결과와 동일하다. 이때 오차의 차수는 (31)에서 알고리즘 4-1과 동일한 11차로써 전체 샘플수가 동일한 경우 알고리즘 오차의 차수 역시 동일함을 알 수 있다.

6-샘플 알고리즘

알고리즘 6-0 이제 소구간의 샘플수가 6일때의 알고리즘

표 1. 알고리즘 오차.

Table 1. Errors of algorithms.

샘플 횟수	이전 샘플수	알고리즘 오차	
		$\Delta T \rightarrow$ 일정	$T \rightarrow$ 일정
4	0	$\frac{ab}{315} \lambda^9$	$\frac{ab}{82,575,360} (\Omega T)^9$
	1	$\frac{ab}{1386} \lambda^{11}$	$\frac{ab}{5,813,305,344} (\Omega T)^{11}$
5	0	$\frac{5ab}{5544} \lambda^{11}$	$\frac{ab}{54,140,625,000} (\Omega T)^{11}$
6	0	$\frac{ab}{4004} \lambda^{13}$	$\frac{ab}{52,295,018,840,064} (\Omega T)^{13}$

을 일반화된 방법을 이용하여 유도하고 해석한다. 이 경우 $N=6$ 이 되며 (21)에서 k_l 를 구하면 다음과 같다.

$$\begin{aligned} k_1 &= \frac{15797}{4650}, k_2 = \frac{3917}{2310}, \\ k_3 &= \frac{608}{385}, k_4 = \frac{2279}{2310}, k_5 = \frac{463}{924} \end{aligned} \quad (32)$$

(32)에서 구한 k_l 를 이용하여 알고리즘과 알고리즘 오차를 구하면 다음과 같다.

$$\begin{aligned} \delta \hat{\varphi}_m &= k_5 \Delta \varrho_m(1) \times \Delta \varrho_m(6) \\ &+ k_4 \Delta \varrho_m(2) \times \Delta \varrho_m(6) + k_3 \Delta \varrho_m(3) \times \Delta \varrho_m(6) \\ &+ k_2 \Delta \varrho_m(4) \times \Delta \varrho_m(6) + k_1 \Delta \varrho_m(5) \times \Delta \varrho_m(6) \end{aligned} \quad (33)$$

$$\phi_\epsilon = \frac{ab}{4004} \lambda^{13} = \frac{ab}{52,295,018,840,064} (\Omega T)^{13} \quad (34)$$

일반화된 알고리즘을 이용하여 본 논문에서 새롭게 유도한 6-샘플 알고리즘은 (33)과 같으며 오차의 차수는 13차가 된다. 이 결과는 앞에서 제시한 알고리즘들 보다 우수한 성능을 나타내는 것으로 유도 과정이 간단하다는 것을 알 수 있다.

앞에서 제시된 알고리즘들의 오차를 정리하면 표 1과 같다.

위 표에서 전체 샘플수가 동일하면 알고리즘의 오차는 동일한 결과를 나타냄을 알 수 있다. 또한 소구간(T)이 일정한 알고리즘과 미소구간(ΔT)이 일정한 알고리즘 오차의 차수는 동일하며 계수만이 다르게 된다. 본 논문에서 새로이 제시한 알고리즘 6-0은 13차의 가장 적은 오차를 보이며, 알고리즘 5-0은 알고리즘 4-1과 동일한 차수의 오차를 나타내지만 크기가 다소 감소함을 알 수 있다.

V. 결론

본 논문에서는 원추오차를 최소로 하는 일반화된 SDINS 최적 자세계산 알고리즘을 제시함으로써, 지금까지 발표된 여러가지 알고리즘들을 포함하고 보다 많은 알고리즘들로 쉽게 확장할 수 있음을 보였다. 일반화된 자세계산 알고리즘은 지금까지의 방법과는 달리 알고리즘을 행렬식으로 간략하게 표현할 수 있으며, 확장성이 매우 우수하다. 이는 새로운 알고리즘 6-0을 쉽게 유도함으로써 입증하였다. 또한 알고리즘의 오차도 일반화된 형태로 표현함으로써 여러가지 알고리즘의 오차를 쉽게 파악할 수 있었으며 그 결과 전체 샘플수가 동일하면 소구간의 샘플 횟수에 관계없이 동일한 차수의 오차를 나타냄을 보였다. 본 논문에서 제시된 일반화된 알고리즘은 사용자가 원하는 사양에 맞도록 알고리즘을 결정할 수 있다는 면에서 일반적인 항체나 유도무기 등의 스트랩다운 관성항법시스템을 사용하는 많은 시스템의

자세계산에 유용하게 사용될 수 있을 것이다.

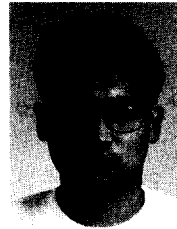
참고문헌

- [1] J. C. Wilcox, "A new algorithm for strapdown inertial navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-3, no. 5, pp. 796-802, 1967.
- [2] R. B. Miller, "A new strapdown attitude algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 6, no. 4, pp. 287-291, 1983.
- [3] J. G. Lee, et al., "Extension of strapdown attitude algorithm for high-Frequency base motion," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 4, pp. 738-743, 1990.
- [4] Y. F. Jiang and Y. P. Lin, "Improved strapdown coning algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 2, pp. 484-490, 1992.
- [5] J. E. Bortz, "A new mathematical formulation for strapdown inertial navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 7, no. 1, pp. 61-66, 1971.
- [6] J. W. Jordan, "An accurate strapdown direction cosine algorithm," *Report TN D-5384*, NASA, Washington, DC, Sept., 1969.
- [7] M. B. Ignagni, "Optimal strapdown attitude integration algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 4, pp. 738-743, 1990.
- [8] Y. F. Jiang and Y. P. Lin, "On the rotation vector differential equation," *IEEE Transactions on Aerospace and Electronic Systems*, AES-27, pp. 181-183, Jan., 1992.
- [9] 이장규, 박찬국, 전창배, "스트랩다운 관성항법장치의 자세계산 알고리즘," 국방과학연구소 연구보고서, 4, 1988.
- [10] P. G. Savage, "Strapdown system algorithms," AGARD-LS-133, 1984.
- [11] M. B. Ignagni, "Efficient class of optimal coning compensation algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 2, pp. 424-429, 1996.



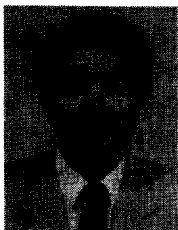
박 찬 국

1961년 9월 13일생. 1985년 서울대 공대 제어계측공학과 졸업. 1987년 동 대학원 제어계측공학과 졸업(석사). 1993년 동 대학원 제어계측공학과 졸업(공학박사). 1993년 서울대 제어계측신기술연구센터 박사후 연구원. 현재 광운대 제어계측공학과 조교수. 연구관심분야는 관성항법시스템, 칼만필터링, 대형시스템.



김 광 진

1972년 10월 29일생. 1995년 광운대학교 제어계측공학과 졸업. 현재 동 대학원 제어계측공학과 석사과정 재학중. 연구관심분야는 스트랩다운 항법알고리즘.



이 장 규

1946년 3월 28일생. 1971년 서울대 공대 전기공학과 졸업. 1974년 Univ. of Pittsburgh 전기공학과 졸업(석사). 1977년 Univ. of Pittsburgh 전기공학과 졸업(공학박사). 1977. 4 ~ 1981. 6 The Analytic Sciences Corporation 연구원(Technical Staff). 1981. 6 ~ 1982. 7 Charles Stark Draper Laboratory 연구원(Technical Staff). 1982. 9 ~ 현재 서울대학교 제어계측공학과 교수. 1994. 12 ~ 현재 서울대 자동제어특화연구센터 소장. 연구관심분야는 관성항법시스템, 계수식별, 자율주행항체(AGV), 표적 추적 시스템.