

이 장 규

서울대학교 공과대학

운 용 중 *

제어계측공학과

1. 서론

SINS(Strapdown Inertial Navigation System)는 안정대(platform)를 사용하지 않고 자이로와 가속도계를 집적 항체에 부착하여 항체의 초기상태와 현재 상태의 상대적인 자세, 위치, 속도등을 계산기로 추정하는 것이다 [1,2]. 이와같은 SINS에서는 계산기 및 알고리즘의 성능이 가장 중요하다.

본 논문에서는 자세추정알고리즘에 사용되는 적분 방법, update rate 및 계산기의 단어길이 가 시스템에 주는 영향에 대하여 분석하였으며, 또한 시스템의 대역폭을 결정하는데 필요한 항체운동주파수와 update rate와의 관계를 조사하였다.

1971년 Jordan은 회전벡터 추정기간동안 자이로의 출력을 오차다항식으로 가정한 후 회전벡터를 추정하여 Quaternion을 update시켰으며, 1983년 Robin B. Miller는 회전벡터 추정기간동안 자이로의 출력을 3차다항식으로 가정한 후 같은 방법으로 Quaternion을 update시켰다 [3].

또한 본 연구에서는 자이로 출력을 4차다항식으로 가정한 자세추정알고리즘을 개발하였다. 이들 알고리즘들의 성능을 비교하기 위하여 회전벡터가 시간에 따라 불규칙하게 변화하는 원추운동을 발생하여 각각의 알고리즘을 시뮬레이션하였다. 원추각, 회전벡터의 변화주파수, update rate를 각각 변화시킬때의 오차를 비교하였으며, 삼각함수로 표현된 Quaternion항들을 2차, 4차역곱수로 전개하였을때의 오차를 각각 비교분석하였다.

2. 자세추정 알고리즘

Quaternion를 이용한 항체자세(roll, pitch, yaw) 추정 알고리즘에 대하여 알아본다. 각 속도(\underline{w})와 회전벡터($\underline{\phi}$)과의 관계는 참고문헌 [6]에 유도되어 있으며, 그 결과는 아래와 같다.

$$\dot{\underline{\phi}}(t) = \underline{w}(t) + \frac{1}{2}(\underline{\phi}(t) \times \underline{w}(t)) + \frac{1}{12}(\underline{\phi}(t) \times (\underline{\phi}(t) \times \underline{w}(t))) \quad (1)$$

식(1)의 해를 구하는 방법으로는 직접 미분 방정식을 푸는 방법과 Taylor급수 전개방법이 있는데 계산시간을 단축하기 위하여 후자의 방법이 많이 사용된다. Jordan은 update 기간 동안에 각속도를 오차다항식으로 가정하여 식(1)의 해를 구하였으며, Robin B. Miller는 3차 다항식으로 가정하여 회전벡터를 구하였다 [3,4,5]. 이와같이 얻어진 회전벡터는 각각 아래와 같이 표현된다.

$$\underline{\phi}(T+h) = \theta_1 + \theta_2 + \frac{2}{3}(\theta_1 \times \theta_2) \quad (2)$$

$$\underline{\phi}(T+h) = \theta_1 + \theta_2 + \theta_3 + 0.45(\theta_1 \times \theta_3) + 0.675\theta_2 \times (\theta_3 - \theta_1) \quad (3)$$

여기서 θ 는 각 샘플링 기간동안의 각도증분이다.

비슷한 방법으로 update 기간 동안에 각속도를 식(4)와 같이 4차 다항식으로 근사화하여 식(1)의 해를 구하여 얻은 회전벡터는 식(5)와 같다 [6].

$$\underline{\phi} = \int_h^{T+h} \underline{w}(\tau) d\tau = a\tau + b\tau^2 + c\tau^3 + d\tau^4 \quad (4)$$

$$\begin{aligned} \phi(T+h) = & \theta_1 + \theta_2 + \theta_3 + \theta_4 + 0.7(\theta_1 \times \theta_2) + 0.146 \\ & (\theta_1 \times \theta_3) + 0.23(\theta_1 \times \theta_4) + 1.545(\theta_2 \times \theta_3) \\ & + 1.257(\theta_2 + \theta_4) - 0.41(\theta_3 \times \theta_4) \quad (5) \end{aligned}$$

식(4)에서 h는 update 기간 동안의 시간이며, t는 $0 < t < h$ 범위의 임의의 시간이며, a, b, c, d는 임의의 벡터이다.

Update 기간 동안의 Quaternion $\hat{q}(h)$ 는 다음과 같은 회전벡터로 표현된다 [5].

$$\hat{q}(h) = \begin{bmatrix} \cos \phi_0 / 2 \\ \phi_x / \phi_0 \sin \phi_0 / 2 \\ \phi_y / \phi_0 \sin \phi_0 / 2 \\ \phi_z / \phi_0 \sin \phi_0 / 2 \end{bmatrix} = \begin{bmatrix} C \\ \phi_x S \\ \phi_y S \\ \phi_z S \end{bmatrix} \quad (6)$$

여기서 $\phi_0 = \phi_x^2 + \phi_y^2 + \phi_z^2$ 이고, C는 $\cos \phi_0 / 2$, S는 $\frac{1}{\phi_0} \sin \phi_0 / 2$ 이다.

식(5)에 포함된 오차가 최소가 되도록 하기 위하여 Quaternion의 참값을 q(h)라 한다면, Quaternion의 오차 $\tilde{q}(h)$ 는 q(h)와 $\hat{q}^{-1}(h)$ 의 Quaternion 곱으로 표현된다 [5].

$$\begin{aligned} \tilde{q}(h) &= q(h) * \hat{q}^{-1}(h) \\ &= \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} * \begin{bmatrix} C \\ -\phi_x S \\ -\phi_y S \\ -\phi_z S \end{bmatrix} \\ &= \begin{bmatrix} P_0 C - S(-P_1 \phi_x - P_2 \phi_y - P_3 \phi_z) \\ P_1 C - S(P_0 \phi_x - P_3 \phi_y + P_2 \phi_z) \\ P_2 C - S(P_3 \phi_x + P_0 \phi_y - P_1 \phi_z) \\ P_3 C - S(-P_2 \phi_x + P_1 \phi_y + P_0 \phi_z) \end{bmatrix} \quad (7) \end{aligned}$$

식(7)에서 Quaternion 참값(q)이 식(8)과 같은 원추운동 모델로 표현된다면, 식(8)을 식(7)에 대입하여 Quaternion의 오차식을 얻은 후 그 오차식이 최소가 되는 조건으로부터 식(5)의 계수값들이 결정된다 [6]. 이들 계수값을 식(5)에 적용하여 얻은 회전벡터는 식(9)와 같다 [5].

$$\begin{aligned} q(h) &= \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = Q^{-1}(t) * Q(t+h) \\ &= \begin{bmatrix} 1 - 2 \sin^2 \phi_0 / 2 \sin^2 \frac{Wh}{2} \\ \sin^2 \phi_0 / 2 \sin Wh \\ -\sin \phi_0 \sin \frac{Wh}{2} \sin W(t + \frac{h}{2}) \\ \sin \phi_0 \sin \frac{Wh}{2} \cos W(t + \frac{h}{2}) \end{bmatrix} \quad (8) \end{aligned}$$

$$\begin{aligned} \phi(T+h) = & \theta_1 + \theta_2 + \theta_3 + \theta_4 + \frac{92}{105} \theta_1 \times (\theta_3 - \theta_2) \\ & + \frac{54}{105} (\theta_1 \times \theta_4) + \frac{306}{105} (\theta_2 \times \theta_3) \quad (9) \end{aligned}$$

3. 알고리즘 성능분석

Update 기간 동안에 3회 샘플링한 경우의 오차(ϕ_ϵ)는 $(\omega h)^5$ 차까지는 영이 되고, (ωh) 의 7차 이상 항으로 표현된다.

$$\phi_\epsilon = \frac{\phi_0^2 (\omega h)^7}{204120} + \text{high order term} \quad (10)$$

Update 기간 동안에 4회 샘플링한 경우의 오차(ϕ_ϵ)는 $(\omega h)^7$ 까지는 영이 되고 (ωh) 의 9차 이상의 항으로 표현된다.

$$\phi_\epsilon = \frac{\phi_0^2 (\omega h)^9}{725760} + \text{high order term} \quad (11)$$

Update 기간 동안 2회 샘플링하여 회전벡터를 구하여 자세를 추정할 알고리즘 오차는 아래와 같다.

$$\phi_\epsilon = \frac{\phi_0^2 (\omega h)^5}{960} + \text{high order term} \quad (12)$$

알고리즘의 오차에 대한 시뮬레이션 결과에 대하여 비교 검토한다. 원추운동에서 회전벡터의 크기는 2도(degree)로 고정하였고, 회전벡터의 주파수는 1, 10, 20, 25 Hz로 변화시켰으며, update 기간은 0.04, 0.02, 0.01, 0.005, 0.004 초(sec)로 변화시키면서, 각각에 대하여 60분 동안 update 시켰을 때 발생하는 오차를 2 point 알고리즘, 3 point 알고리즘, 4 point 알고리즘에 대하여 조사하였다.

원추운동 주파수 및 update rate 와 오차와의 관계는 그림 1 및 2와 같다.

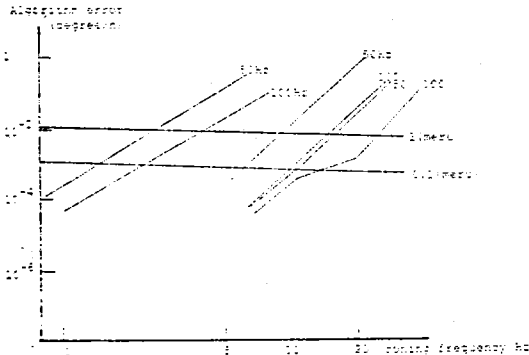


그림 1. 원추운동주 파수와 알고리즘 오차와의 관계

Update기간 동안의 2회 샘플링한 알고리즘 경우는 계산기의 단어 길이가 64비트 일때와 32비트일때의 알고리즘 오차는 비슷하다. 즉 update기간 동안에 2회 샘플링하여 자세를 추정 하고자 할때는 계산기 단어의 길이를 64비트로 하여도 오차는 줄어들지 않는다. 또한 update rate가 0.01sec, 0.005sec, 0.004sec일때의 오차는 서로 비슷하므로, 32비트 단어 길이를 갖는 계산기를 사용할 경우에는 update 기간이 0.01sec 이하이면 비트수를 더 늘려도 알고리즘의 오차는 줄어들지 않음을 알 수 있다.

Quaternions의 $\cos \frac{\phi_0}{2}$ 양과 $\sin \frac{\phi_0}{2}$ 양을 2차, 4차, 5차의 멱급수로 전개하여 알고리즘에 적용하였을 때의 오차와, 계산기의 삼각함수로 부터 직접 구한 알고리즘의 오차는 비교 같다.

표 1. Truncation 샘플 레이션 결과 ($\phi_0=2, f=5$)

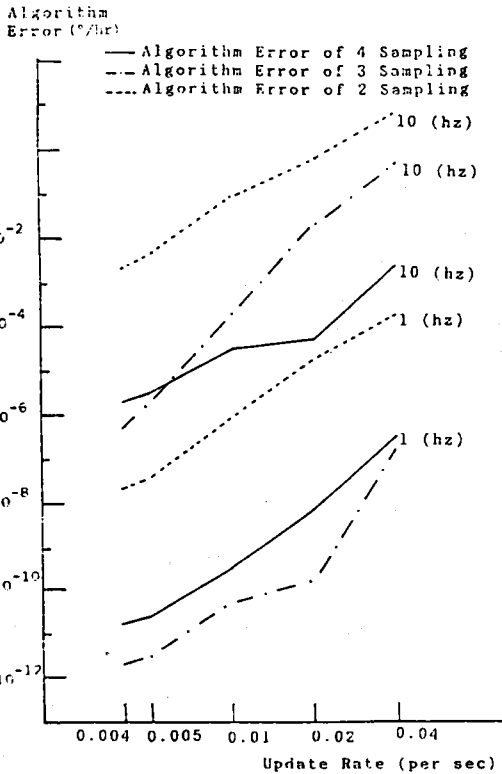


그림 2. Update rate와 알고리즘 오차와의 관계

order	Update Attitude		0.04 (sec)	0.02 (sec)	0.01 (sec)	0.005 (sec)	0.004 (sec)
	Roll	Yaw					
한수	Roll	Pitch	1.7415588 × 10 ⁻⁴	3.07410235 × 10 ⁻⁴	2.0377083 × 10 ⁻⁴	5.4395868 × 10 ⁻⁴	7.4163677 × 10 ⁻⁴
	Pitch	Yaw	1.84602333 × 10 ⁻⁴	4.3742787 × 10 ⁻⁴	8.04911693 × 10 ⁻⁴	8.79296635 × 10 ⁻⁴	2.198241589 × 10 ⁻³
	Yaw	Roll	5.1832469 × 10 ⁻⁴	8.0616291 × 10 ⁻⁴	1.0889453 × 10 ⁻³	5.758337 × 10 ⁻⁴	7.1127688 × 10 ⁻⁴
2	Roll	Pitch	1.7418214 × 10 ⁻⁴	3.0723576 × 10 ⁻⁴	2.046405 × 10 ⁻⁴	5.21649389 × 10 ⁻⁴	7.3140182 × 10 ⁻⁴
	Pitch	Yaw	1.7991244 × 10 ⁻⁴	3.907965 × 10 ⁻⁴	3.8857806 × 10 ⁻⁴	9.6866959 × 10 ⁻⁴	2.2953861 × 10 ⁻³
	Yaw	Roll	5.18363906 × 10 ⁻⁴	8.0653316 × 10 ⁻⁴	1.091502 × 10 ⁻³	5.9196387 × 10 ⁻⁴	6.4389850 × 10 ⁻⁴
4	Roll	Pitch	1.7415579 × 10 ⁻⁴	3.073670377 × 10 ⁻⁴	2.044142 × 10 ⁻⁴	7.6386676 × 10 ⁻⁴	7.22072765 × 10 ⁻⁴
	Pitch	Yaw	1.7949530 × 10 ⁻⁴	3.8332048 × 10 ⁻⁴	2.9143354 × 10 ⁻⁴	8.8429264 × 10 ⁻⁴	4.0531404 × 10 ⁻³
	Yaw	Roll	5.1832469 × 10 ⁻⁴	8.06162906 × 10 ⁻⁴	1.0889453 × 10 ⁻³	5.758337 × 10 ⁻⁴	7.112764759 × 10 ⁻⁴
5	Roll	Pitch	1.74354482 × 10 ⁻⁴	3.079674481 × 10 ⁻⁴	2.0464234 × 10 ⁻⁴	7.9386385 × 10 ⁻⁴	7.142424717 × 10 ⁻⁴
	Pitch	Yaw	1.7555401 × 10 ⁻⁴	5.40401057 × 10 ⁻⁴	8.32667268 × 10 ⁻⁴	1.010025397 × 10 ⁻³	2.3703256 × 10 ⁻³
	Yaw	Roll	5.181188895 × 10 ⁻⁴	8.042191939 × 10 ⁻⁴	1.075822467 × 10 ⁻³	4.895903099 × 10 ⁻⁴	1.064555598 × 10 ⁻³

표 1에서 알수 있는 바와 같이 Quaternion의 삼각함수항들의 절삭오차는 큰 영향을 주지 못함을 알 수 있다.

4. 결론

본 연구에서는 SINS의 자세추정 알고리즘 성능을 분석하기 위하여 컴퓨터 시뮬레이션 기법을 사용하였다. 시뮬레이션에 사용한 항체운동은 회전각이 2°(degree)인 원추운동 모델을 이용하였으며, 회전벡터 주파수와 update rate를 변화시키면서 알고리즘 오차를 조사하였다. 시뮬레이션 결과 항체운동주파수가 높은 경우에 4차 알고리즘의 성능이 다른 알고리즘의 성능보다 우수하였으며, 계산기의 단어길이는 2차 알고리즘에서는 32비트, 3,4차 알고리즘에서는 64비트로 사용하는 것이 매우 모과적이였다.

참 고 문 헌

- [1] Kenneth R. Britting Sc. D, Inertial Navigation System Analysis, Wiley-Interscience. 1971.
- [2] Chalres Broxnever, Inertial Navigation System, McGraw-Hill, 1964.
- [3] Robin B. Miller, " Strapdown Inertial Navigation System : An Algorithm for Attitude and Navigation Computation," AR-002-285. Dudy, Report 13, 1978.
- [4] Nurse, J. Prohaska, D. Riegsecker, " A New Baseline for the Inertial Navigation Strapdown Simulation, " Vol. I, The Charles Stark Draper Laboratory, Inc. R-1136, 1978.
- [5] Robin B. Miller, " A New Strapdown Attitude Algorithm, " J. Guidance, Vol. 6, No. 4, July 1983.
- [6] 은용중, " 스트랩다운 방식을 이용한 자세추정 장치의 구현 및 성능분석, " 서울대학교 공과대학 제어계측공학과 석사논문, 1986.