

기동특성에 따른 ARS 자세 성능향상 기법

The Improvement Method of ARS Attitude depending on Dynamic Conditions

박 찬 주* **이 상 정****
Park, Chan-Ju Lee, Sang-Jeong

ABSTRACT

The ARS(Attitude Reference System) calculates an attitude of a vehicle using inertial angular rate sensors and acceleration sensors. The attitude error of ARS increases due to the integration of angular rate sensor output. To reduce the attitude error an acceleration of sensor is used similar to leveling method of INS(Inertial Navigation System). When an acceleration of vehicle is increased, it is difficult to calculate the attitude error using acceleration sensor output. In this paper the estimation method of acceleration due to the attitude error only is proposed. Two methods of the attitude calculation depending on vehicle dynamics and the integration method of these two methods are proposed. To verify its performance the monte carlo simulation is performed and shows that it bounds attitude error of ARS to reasonable level.

주요기술용어(주제어) : ARS(자세계산장치), Kalman Filter(칼만필터)

1. 머리말

ARS(Attitude Reference System)는 각속도를 이용하여 자세를 계산하는 장치이다. 일반적으로 제어용으로 사용되는 저급의 MEMS(Micro Electronic Mechanical System) 자이로 센서를 사용하여 자이로 센서 출력인 각속도를 적분하여 자세를 구한다. 그러나 계속되는 각속도센서 오차의 적분에 의해 자세오차는 증가하게 된다. 이러한 자세오차를 줄이기 위해서 각속도계를 이용하여 각속도계 출력을 적분하지

않고 직접적으로 자세를 구하여 자세오차를 줄이는 방법이 연구되어왔다^[1~3]. 정지상태에서 각속도계는 각축에서 측정되는 중력 성분을 측정하여 항체의 자세를 계산할 수 있으며 계산된 자세의 정밀도는 각속도계의 바이어스에 비례한다. 자세를 구하기 위해서는 각속도를 적분해야 하며 적분과정에서 각속도센서 센서오차도 적분되어 자세오차가 시간이 지남에 따라 증가하지만 각속도계로 구하는 자세는 각속도계 출력을 적분하지 않고 자세를 구하여 각속도계 센서오차에 의한 자세오차는 증가하지 않으므로 시간이 지남에 따라 상대적으로 정확한 자세값이 된다. 각속도를 이용한 자세 보정 방법으로는 쿼터니언을 사용한 자세계산 알고리즘을 향상시킨 방법과 각속도로 계산된 자세와 각속도계로 구한 자세의 차를 측정치로 사용하는 필터를 구성하여 보다 정확한 자세를 추정하는

† 2008년 10월 27일 접수~2008년 12월 12일 게재승인

* 국방과학연구소(ADD)

** 충남대학교(CNU)

주저자 이메일 : nayana@add.re.kr

방법이 있다^[2]. 위의 방법은 항체의 가속에 의해 발생하는 가속도는 없고 가속도계에서 측정되는 가속도는 중력에 의한 가속도 성분만이 측정되는 것으로 가정하였다. 그러므로 항체가 한 방향으로 실제 가속이 주어질 때는 중력 외에 가속이 발생하여 가속도계에 중력성분과 항체 가속성분이 합쳐진 값이 측정되어 정확한 자세를 계산할 수 없게되어 자세추정 오차가 증가하게 된다. 이러한 경우에 칼만필터의 측정잡음 공분산값을 측정되는 가속도의 합에 따라 다른 값을 적용하는 연구가 진행되어 중력외의 가속이 생겼을 경우에도 자세계산을 할 수 있었다^[3]. 그러나 급격하게 입력되는 큰 가속이 있는 경우는 여전히 자세보정 성능에는 한계가 있으며 가속도계 센서오차에 의해 발생하는 가속도 오차 성분은 그대로 남아 있어 자세보정 오차로 작용한다. 본 논문에서 제시된 자세계산 알고리즘은 자세오차에 의해 발생하는 가속도를 플랫폼 각속도와 유사한 피드백 입력으로 주어 자세오차를 줄이는 정지상태 GINS(Gimbal Inertial Navigation System) 수평정렬 알고리즘과 유사하지만 정지상태가 아닌 항체가 기동하는 경우에 대해서 정확한 자세계산을 하는 기법을 제시한다. 제시된 자세계산 알고리즘은 두 개의 개별 자세계산 알고리즘과 이를 통합하는 통합 자세계산 알고리즘으로 구성된다. 제시된 저기동 자세계산 알고리즘은 항체의 기동이 저기동이고 작은 경우 적합한 자세계산 알고리즘이고, 고기동 자세계산 알고리즘은 칼만필터를 이용하여 자세오차로 유발된 가속도만을 추정하여 사용하며 항체의 가속도가 계속 증가하거나 급격히 커지는 고기동인 경우에 적합한 알고리즘이다. 통합 항법알고리즘에서는 항체의 기동특성에 따라 두 알고리즘의 자세계산값을 이용하여 항체의 정확한 자세계산을 하도록 하였다. 제시된 자세계산 알고리즘은 저급 MEMS IMU(Inertial Measurement Unit)를 사용한 ARS를 항공기의 비행 궤적에 적용하였고 시뮬레이션을 통하여 성능 분석을 하였다.

2. 해석적 플랫폼 제어입력 사용 자세계산 알고리즘

GINS 항법장치의 초기 수평정렬은 김벌을 회전시

켜 항법좌표계와 일치하게 만드는 과정이다. 수평정렬은 각축의 가속도계로 중력성분을 측정하여 지구 중심 방향으로 장착된 가속도계에서만 중력이 측정되도록 김벌을 회전시킨다. 이를 위하여 자세오차에 의해 수평축 가속도계에서 측정되는 가속도가 0이 되도록 각각의 김벌에 플랫폼 각속도를 주어 회전시킨다. 이 때 입력으로 주어지는 플랫폼 각속도는 수평축 가속도계에서 측정되는 가속도를 이용하여 계산되어진다. 이와 같은 GINS의 수평정렬과 유사하게 SDINS(Strapdown Inertial Navigation System)에서는 플랫폼 대신 가상의 해석적 플랫폼이 동체좌표계에서 항법좌표계로 센서에서 측정된 값을 변환시켜준다. 해석적 플랫폼은 쿼터니언, 오일러각, 회전벡터, 방향 코사인 행렬로 나타낼 수 있으며 본 논문에서는 계산량과 계산오차가 적은 쿼터니언을 사용한다^[4].

가. 정지상태 자세계산 알고리즘

자세계산 알고리즘은 자이로에서 출력된 각속도를 이용하며 해석적 플랫폼의 자세를 쿼터니언으로 구한다. 자세계산 알고리즘에서 정의되는 자세는 항법좌표계와 동체좌표계간의 회전각으로 자세오차가 없는 경우는 해석적 플랫폼과 항법좌표계가 일치한다. 자세계산을 위한 쿼터니언 계산은 두번에 나누어져 계산된다^[4]. 첫 번째 쿼터니언 계산은 자이로 출력을 이용한 관성좌표계에서 동체좌표계의 회전량으로 해석적 플랫폼에서 내부적으로 계산되는 값이다. 앞에서 구해진 첫 번째 쿼터니언은 관성좌표계와 동체좌표계간의 자세 쿼터니언으로 두 번째 쿼터니언 계산은 첫 번째 계산된 쿼터니언에서 관성좌표계에서의 항법좌표계 회전량을 보정해 주어 항법좌표계와 동체좌표계간 자세 쿼터니언을 최종적으로 구한다.

나. 수평정렬 알고리즘

정지상태 수평정렬에서 자세오차에 의해 발생한 가속도가 가속도계에서 측정되며 이를 플랫폼 각속도에 추가된 제어입력으로 되먹임하여 자세오차를 줄일 수 있다^[4]. 저급 MEMS 관성센서를 사용할 경우 수평정렬시 되먹임되는 플랫폼 각속도는 다음 식으로 나타낼 수 있다.

$$\begin{aligned} \omega^p &= \omega^c \\ \omega^p &= [\omega_n^p \ \omega_e^p \ \omega_d^p]^T \\ \omega^c &= [\omega_n^c \ \omega_e^c \ \omega_d^c]^T \end{aligned} \quad (1)$$

위 식에서 ω^p 는 해석적 플랫폼의 가속도 입력이고 ω^c 는 해석적 플랫폼 가속도에 추가된 제어입력이다. 수평축 플랫폼 가속도에 추가된 가속도는 다음과 같이 계산되어 진다^[4].

$$\omega_n^c = k_2 \delta v_e^n \quad (2)$$

$$\omega_e^c = -k_2 \delta v_n^n \quad (3)$$

$$\delta v^n = [\delta v_n^n \ \delta v_e^n \ \delta v_d^n]^T$$

추가된 플랫폼 가속도는 k_2 에 비례하는 속도오차 입력 으로 속도오차 δv^n 는 식 (4)와 (5)를 이용하여 내부적으로 계산한다^[4].

$$\delta \dot{v}_n^n = f_n^n - k_1 \delta v_n \quad (4)$$

$$\delta \dot{v}_e^n = f_e^n - k_1 \delta v_e \quad (5)$$

$$f_n^n = g \psi_e^n \quad (6)$$

$$f_e^n = -g \psi_n^n \quad (7)$$

$$f^n = [f_n^n \ f_e^n \ f_d^n]^T$$

위 식에서 f^n 은 자세오차로 유발된 가속도로 수평 정렬시 가속도계에서 측정된 가속도를 항법좌표계로 변환한 가속도이다. 위 식을 자세오차 방정식에 대입 하고 정지상태 수평정렬을 고려하면 식 (8)이 된다.

$$\begin{aligned} \dot{\psi}^n &= -(\Omega_{ie}^n + \Omega_{en}^n) \psi^n + \delta \omega^n + \omega^p \\ &= -(\Omega_{ie}^n + \Omega_{en}^n) \psi^n + \delta \omega^n + \omega^c \\ &\approx \omega^c + \delta \omega^n \end{aligned} \quad (8)$$

위 식에서 $\delta \omega^n$ 은 자이로 센서오차이고 $\Omega_{ie}^n + \Omega_{en}^n$ 은 관성좌표계에서의 항체좌표계 회전량이다. 자세오차 방정식과 속도오차 방정식을 라플라스 변환하여 각 축의 자세오차를 구하면 다음과 같이 된다.

$$(S^2 + k_1 S + k_2 g) \psi_n^n(S) = \alpha(S) \quad (9)$$

$$(S^2 + k_1 S + k_2 g) \psi_e^n(S) = \beta(S) \quad (10)$$

위 식을 시간영역으로 변환하면 다음과 같다.

$$\ddot{\psi}_n^n + k_1 \dot{\psi}_n^n + k_2 g \psi_n^n = \alpha(t) \quad (11)$$

$$\ddot{\psi}_e^n + k_1 \dot{\psi}_e^n + k_2 g \psi_e^n = \beta(t) \quad (12)$$

식 (11)과 (12)에서 $\alpha(t)$ 와 $\beta(t)$ 는 입력되는 오차항으로 센서오차와 관련된 값이다. 플랫폼 가속도 제어 입력으로 가속도를 사용한 수평정렬 알고리즘은 식 (11)과 (12)와 같이 감쇄항을 가지는 2차 시스템과 같은 형태가 되어 자세오차 수렴 성능과 정상상태 자세오차 성능을 향상됨을 알 수 있다.

3. 개별 자세계산 알고리즘

개별 자세계산 알고리즘은 수평정렬 알고리즘에서 항체가 기동하는 것을 고려하여 자세계산을 하도록 하였다. 개별 자세계산 알고리즘은 항체의 기동 특성에 따라 저기동과 고기동으로 구분하였고 직선운동을 하고 급격한 가감속이 없는 경우를 저기동으로 가정하고 회전운동을 하거나 급격한 가감속이 있는 경우를 고기동으로 가정하였다. 또한 항체의 기동특성에 따라 다른 댐핑계수를 사용하였다. 구성된 개별 알고리즘은 항공기 항법장치에 적용하는 것을 가정하여 항공기 비행 운동을 고려하였다.

가. 저기동 자세계산 알고리즘

항체가 기동하게 되면 항체의 가속도가 발생하게 되며 수평정렬시 사용되었던 자세계산 알고리즘을 항체가 운동하는 경우로 확장하여 사용하기 위해서는 가속도계에서 측정되는 식 (6)과 (7)은 항체의 가속도가 더해진 형태가 되어야 하며 그 외의 수식은 수평정렬과 동일하다.

$$f_n^n = g\psi_e^n + a_n \quad (13)$$

$$f_e^n = -g\psi_n^n + a_e \quad (14)$$

식 (13)과 (14)를 식 (4)와 (5)에 대입하여 다시 시간영역에서의 자세오차를 구해보면 다음과 같다.

$$\ddot{\psi}_n^n + k_1\dot{\psi}_n^n + k_2g\psi_n^n = k_2a_n + \alpha(t) \quad (15)$$

$$\ddot{\psi}_e^n + k_1\dot{\psi}_e^n + k_2g\psi_e^n = k_2a_e + \beta(t) \quad (16)$$

식 (15)에서 외부 오차 요소인 오른쪽 항은 센서오차인 $\alpha(t)$ 와 가속도에 의해서 발생한다. 센서오차는 운동에 무관하게 발생하지만 가속도에 의한 외란은 항체의 운동에 따라 변화되는 항이다. 일반적으로 항체가 직선으로 운동시 주로 등속운동을 하며 발생하는 가속도는 잡음과 유사한 0을 기준으로 시간에 따라 오실레이션하는 형태로 나타난다. 그러므로 자세오차를 가장 작게 만드는 감쇄계수를 결정해 주어야 하며 저기동일때는 가속도계를 이용한 자세계산 값이 신뢰성을 가지므로 큰 k_1 과 k_2 를 사용한다. 저기동 자세계산 알고리즘은 항체의 가속도가 센서오차에 비해 작은 경우는 성능이 우수하지만 항체의 가속도가 센서오차보다 커지는 경우는 자세오차가 유발된다. 이러한 경우는 항체가 회전을 하는 경우나 급격한 가속이 발생하는 경우와 항체가 회전하여 원심력이 발생하여 항체의 가속이 커지는 경우이다. 위의 두 경우 항체의 실제 가속에 의해 발생하는 자세오차를 제거하기 위해서는 자세오차에 의해 발생하는 가속도만을 가속도계 측정값에서 추정하는 방법이 필요하다.

나. 고기동 자세계산 알고리즘

앞에서 제안된 식 (15)와 (16)과 같은 감쇄가 있는 2차 시스템에서는 항체의 실제 가속도가 커지면 그에 비례하여 자세오차는 증가하게 된다. 이는 기동하는 항체에서 가속도를 플랫폼 각속도 제어입력으로 사용하여 발생하는 현상이다. 그러므로 항체가 회전하거나 급격한 가속이 있는 경우는 자세오차로 유발된 가속도를 추정하여 이 값을 플랫폼 각속도로 사용하여 식 (15)와 (16)의 항체 가속도를 보상해야 항체의 가속도에 의해 발생하는 자세오차를 줄일 수 있다. 자세오차로 유발되는 가속도는 급격하게 변하지 않으므로 칼만필터를 통해 이 값을 추정할 수 있다. 설계된 칼만필터는 자세오차로 유발되는 가속도를 상태변수로 정하고 이 값이 상수로 시간에 따라 변하지 않는 것으로 모델링 하였다. 측정값은 가속도계에서 측정되는 가속도로 이 측정값은 자세오차로 유발된 가속도로 구성되고 항체의 가속도와 센서오차항을 모두 측정잡음으로 간주하였다. 구해진 칼만필터는 3차로 상태 방정식 및 측정 방정식은 다음과 같다.

$$x = [f^\psi] \quad (17)$$

$$x(k) = w(k) \quad (18)$$

$$z(k) = x(k) + v(k) \quad (19)$$

$f^\psi(k)$: 자세오차로 유발된 가속도

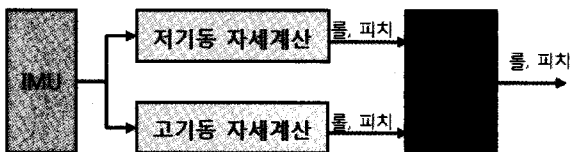
$w(k)$: 자세오차로 유발된 가속도의 백색잡음

$v(k)$: 가속도계 측정값의 백색잡음

식 (17)~(19)의 자세오차로 유발된 가속도 추정 칼만필터는 모든 비행 구간에서 사용되어지며 충분히 큰 측정잡음 공분산 값을 주어 측정되는 가속도가 커져 측정잡음이 커져도 상태변수 추정성능이 저하되지 않게 하였다. 고기동에 사용되는 k_1 과 k_2 는 저기동 자세계산 알고리즘에 사용된 값보다 작은값을 사용하여 고기동 일때 식 (15)와 (16)의 오른쪽항인 항체 가속도에 의해 발생하는 자세오차가 작아지도록 하였다.

4. 통합 자세계산 알고리즘

앞 절에서 제시한 개별 자세계산 알고리즘은 각각 항체가 저기동일때와 고기동일때 자세오차가 작아지도록 하였다. 그러므로 저기동 자세계산 알고리즘은 항체가 저기동일때 자세오차가 가장 작고 항체가 고기동인 경우는 자세오차가 커진다. 또한 고기동 자세계산 알고리즘은 항체가 고기동일때 저기동 자세오차 알고리즘에 자세오차가 작아지며 항체가 저기동인 경우는 저기동 자세계산 알고리즘에 비해 자세오차가 크다. 위 개별 알고리즘은 항체의 기동에 따라 자세오차 성능이 다르며 이에 따라 개별 알고리즘 각각에 가중치를 주어 사용하면 최적의 자세오차 성능을 낼 수 있다. 그 방법으로는 고기동 자세계산 알고리즘은 어떠한 운동에서도 일정 정확도의 자세를 계산할 수 있고, 저기동 자세계산 알고리즘은 가속도 크기가 작고 저기동인 경우 정확한 자세를 계산할 수 있으므로 저기동 자세계산 알고리즘에서 계산한 자세를 기준으로 고기동 자세계산 알고리즘을 보정하여 사용하도록 한다. 보정방법은 고기동 자세계산 알고리즘의 자세계산 값의 자세오차를 칼만필터로 추정하여 보정하는 방법이다. 칼만필터는 고기동 자세계산 알고리즘 자세계산 값의 오차를 상태변수로 사용하고 고기동과 저기동 자세계산 알고리즘의 자세계산 값의 차를 측정값으로 사용한다. 최종 피치각과 롤각 계산은 저기동과 고기동 자세계산 알고리즘의 자세출력값 차의 변화율이 기준값 이하일때만 저기동과 고기동 자세계산 알고리즘의 자세출력 차를 측정치로 사용하였고, 그렇지 않은 경우는 측정치를 0으로 두어 측정치 갱신을 하지 않았다. 전체 통합 자세계산 알고리즘의 구성도는 그림 1과 같다.



[그림 1] 통합 자세계산 알고리즘 전체 구성도

가. 롤각 계산

항체 저기동과 고기동 롤각 차의 변화율이 작은 경우는 저기동 자세계산 알고리즘이 고기동 알고리즘보다 정확하고 그 외에는 고기동 알고리즘이 더 정확하다. 그러므로 롤각 변화율이 기준값 보다 작은 경우는 아래의 칼만필터를 사용하여 고기동 자세계산 알고리즘에서 계산된 롤각의 오차를 추정하여 정확한 롤각을 구하고 그렇지 않은 경우는 고기동 자세계산 알고리즘의 롤각이 정확하다고 판단하여 측정치 갱신을 하지 않는다. 최종적으로 사용된 칼만필터와 계산된 통합 롤각은 식 (26)과 같다^[5].

$$x = [\delta\phi^{high}] \quad (20)$$

$$\dot{x}(k) = w(k) \quad (21)$$

$$K(k) = \frac{P(k|k-1)}{P(k|k-1) + R} \quad (22)$$

$$z(k) = \begin{cases} \phi^{high} - \phi^{low}, & \text{if } \left| \frac{\phi^{high} - \phi^{low}}{\Delta T} \right| \leq 0.02 \\ 0 & \text{if } \left| \frac{\phi^{high} - \phi^{low}}{\Delta T} \right| < 0.02 \end{cases} \quad (23)$$

$$\delta\phi^{high}(k) = \delta\phi^{high}(k-1) + K(k)(z(k) - \delta\phi^{high}(k-1)) \quad (24)$$

$$P(k) = (1 - K(k))P(k|k-1) \quad (25)$$

$$\phi^{high}(k) = \phi^{high}(k-1) - \delta\phi^{high}(k) \quad (26)$$

ϕ^{low} : 저기동 자세계산 알고리즘의 피치각

ϕ^{high} : 고기동 자세계산 알고리즘의 피치각

$\delta\phi^{high}$: 고기동 자세계산 알고리즘의 피치각 오차

식 (23)에서 저기동과 고기동을 판단하는 기준값은 0.02°/sec을 사용하였다.

나. 피치각 계산

통합 피치각 계산은 통합 롤각 계산 방법과 같이 칼만필터를 사용하여 고기동 자세계산 알고리즘의 피치각 출력의 오차를 추정하여 정확한 피치각을 구한다. 롤각과 유사하게 직선운동을 하는 저기동인 경우 고기동 자세계산 알고리즘보다 저기동 자세계산 알고리즘이 항체의 가속도가 작아 정확하므로 이 경우 두 피치각의 차를 측정치로 사용하는 칼만필터로 고기동 자세계산 알고리즘의 피치각 오차를 구한다. 최종적으로 구하는 통합 피치각은 다음과 같다.

$$x = [\delta\theta^{high}] \tag{27}$$

$$\dot{x}(k) = w(k) \tag{28}$$

$$K(k) = \frac{P(k|k-1)}{P(k|k-1) + R} \tag{29}$$

$$z(k) = \begin{cases} \theta^{high} - \theta^{low} , if \left| \frac{\theta^{high} - \theta^{low}}{\Delta T} \right| \leq 0.03 \\ 0 \quad if \left| \frac{\theta^{high} - \theta^{low}}{\Delta T} \right| < 0.03 \end{cases} \tag{30}$$

$$\delta\theta^{high}(k) = \delta\theta^{high}(k-1) + K(k)(z(k) - \delta\theta^{high}(k-1)) \tag{31}$$

$$P(k) = (1 - K(k))P(k|k-1) \tag{32}$$

$$\theta^{high}(k) = \theta^{high}(k-1) - \delta\theta^{high}(k) \tag{33}$$

θ^{low} : 저기동 자세계산 알고리즘의 피치각
 θ^{high} : 고기동 자세계산 알고리즘의 피치각
 $\delta\theta^{high}$: 고기동 자세계산 알고리즘의 피치각 오차

피치각 기동을 구분하는 식 (31)에서 저기동과 고기동을 구분하는 기준값은 0.03°/sec을 사용하였다.

5. 시험 및 시험결과

본 논문에서는 제시된 자세계산 알고리즘의 성능을 분석하기 위해 몬테칼로 시뮬레이션을 100회 수행하였다. 시뮬레이션에 사용된 저급 MEMS IMU 성능은 표 1과 같다. 사용된 IMU 데이터는 50Hz로 샘플링되어 계산되어지며 고기동 및 통합 자세계산 알고리즘에서 사용된 칼만필터도 샘플링 주파수와 같이 계산되어진다. 비행하기전 수평정렬 과정에서 입력되는 각속도 값이 없다고 가정하여 10초 동안 평균한 자이로 출력값을 자이로 동작 바이어스로 생각하여 자이로 출력값에서 빼주어 제거하였다. 시뮬레이션에서 사용된 궤적은 항체가 사각형 형상의 궤적을 따라 비행하는 것으로 가정하였다. 사용된 비행궤적은 그림 2와 같다. 시뮬레이션에 사용된 개별 알고리즘의 댐핑계수는 표 2와 같다. 고기동 및 통합 자세계산 알고리즘에서 사용된 칼만필터의 공정잡음 공분산값은 10⁻⁵을 측정잡음 공분산값은 10을 사용하였다.

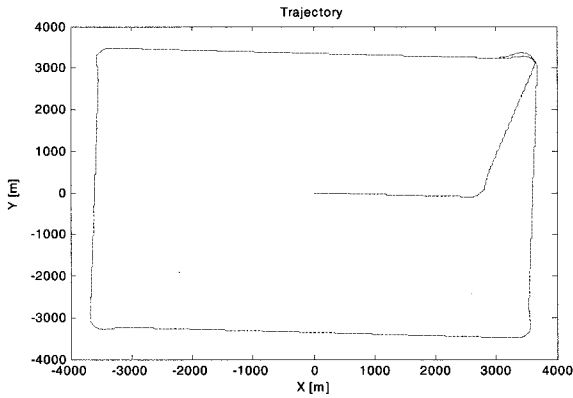
[표 1] 시뮬레이션에 사용된 IMU 성능

센서 종류	오차 종류	크기	단위
자이로	바이어스 반복도	0.5	deg/sec
	환산계수 안정도	1000	ppm
	백색잡음	0.2	deg/sec
가속도 센서	바이어스 반복도	10	mg
	환산계수 안정도	1000	ppm
	백색잡음	3	mg

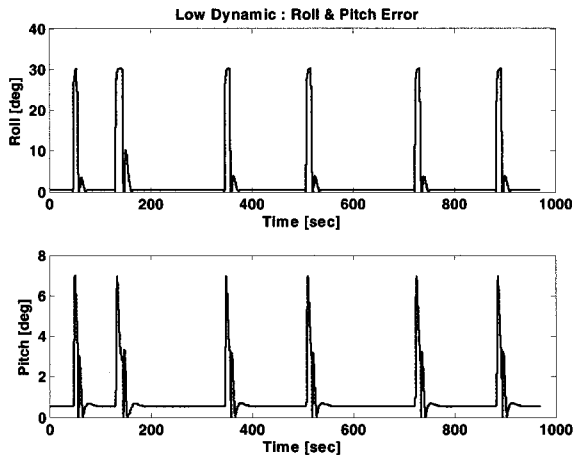
저기동 및 고기동 자세계산 알고리즘에 적용된 댐핑계수는 표 2와 같다.

[표 2] 시뮬레이션에 사용된 댐핑계수

알고리즘	k ₁	k ₂
저기동 자세계산	4	1
고기동 자세계산	0.02	10 ⁻⁵

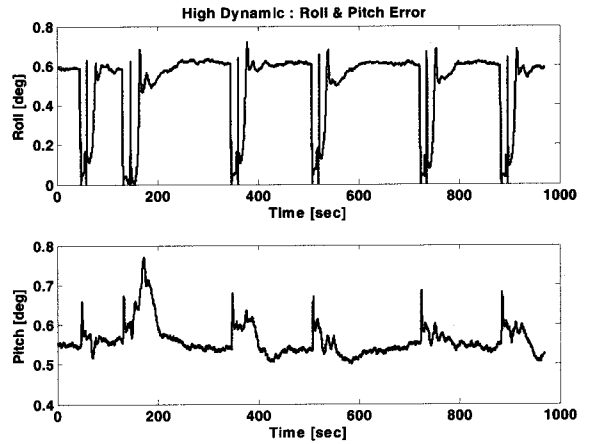


[그림 2] 시뮬레이션에 사용된 비행궤적

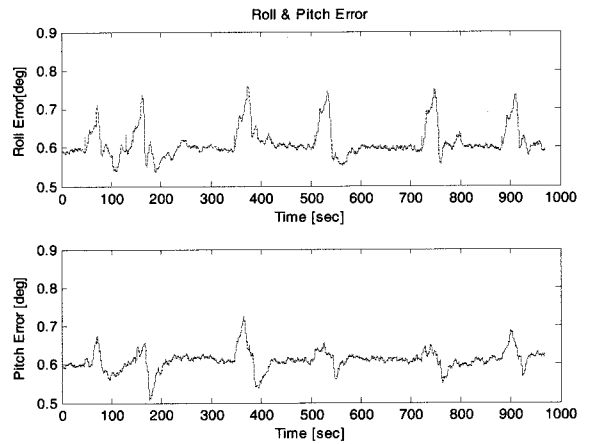


[그림 3] 저기동 자세계산 알고리즘 자세오차

비행궤적에서 90° 회전하는 구간에서 항체는 뱅크턴을 하므로 원심력과 중력 가속도 성분에 의해 가속이 증가하게 된다. 그림 3은 저기동 자세계산 알고리즘의 결과로 회전시 자세 계산 오차가 증가하게 되며, 직진운동하는 동안은 가속도계 바이어스 오차와 비례하는 자세오차만 생기는 것을 알 수 있다. 고기동 자세계산 알고리즘의 결과로 그림 4에서는 회전하는 경우 자세오차에 의해 발생한 가속도를 추정하여 자세오차를 보정해 주므로 저기동 자세계산 알고리즘에 비해 정확한 자세를 계산할 수 있음을 알 수 있다. 저기동 및 고기동의 통합 자세계산 알고리즘의 자세오차는 그림 5와 같으며 가장 큰 자세오차는 롤과 피치 모두 0.8° 이다. 이는 표 1에서와 같이 가속도



[그림 4] 고기동 자세계산 알고리즘 자세오차



[그림 5] 통합 자세계산 알고리즘 자세오차

계 바이어스 및 환산계수에 의해 유발되는 자세오차가 약 0.6° 이므로 기동특성에 의해 발생하는 자세오차는 약 0.2° 이하임을 알 수 있다.

6. 맺음말

본 논문에서는 가속도를 이용하여 ARS의 자세성능을 향상시키는 기법을 제시하였고 몬테칼로 시뮬레이션을 통하여 성능을 분석하였다. 제시된 기법은 자세오차에 의해 발생한 가속도를 자세계산 쿼터니언에 피드백해 줌으로 자세오차를 감소시키는 기법이고, 항

체의 운동특성에 따라 저기동과 고기동에서의 자세계산 알고리즘과 이를 통합하는 통합 자세계산 알고리즘으로 구성되었다. 저기동 자세계산 알고리즘은 항체의 기동이 직선운동을 하고 급격한 가속이 없는 경우로 이에 적합한 큰 댄핑계수를 선정하여 저기동에서 자세 성능이 우수하도록 구성 하였다. 고기동 자세계산 알고리즘은 자세오차로 유발된 가속도를 칼만필터를 통해 추정하여 사용하였고 항체의 가속도가 증가하거나 가속도가 급격이 커진 경우 저기동 자세계산 알고리즘에 비해 자세오차 성능이 향상되었음을 알 수 있었다. 통합 자세계산 알고리즘에서는 항체의 기동에 따라 두 자세계산 알고리즘의 자세결과를 조합하여 기동 특성을 고려한 항체의 자세계산을 하도록 하였다. 제시된 자세계산 알고리즘은 직선운동을 하는 경우 가속도계 센서오차에 의한 자세오차가 발생하였고, 회전하는 경우 자세오차에 의해 발생하는 가속도 추정성능과 비례하여 자세오차가 발생하였다. 가속도계 오차에 의한 자세오차를 제외하면 제시된 자세계산 알고리즘은 동적인 기동에서 0.2° 이하의 자세오차 성능을 보여 전체 비행 구간에서 자이로 센서 오차에 의해 자세오차가 증가하지 않고 자세오차를 일정값으로 제한하는 성능을 보였다.

참 고 문 헌

- [1] E. R. Bachmann, I. Duman, U. Y. Usta, R. B. McGhee, X. Yun and M. J. Zyda, "Orientation Tracking for Humans and Robots using Inertial Sensors", International Symposium on Computational Intelligence in Robotics and Automation, pp. 187~194, Monterey, CA, 1999.
- [2] Xiaoping Yun, Conrado Aparicio, Eric R. Bachmann and Robert B. McGhee, "Implementation and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking", Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, April, 2005
- [3] 강철우, 유명민, 김현욱, 박찬국, "변형된 오일러 각 기반의 칼만필터를 이용한 자세 추정 성능 향상", 제2회 항법기술 워크샵, 2008.
- [4] Oleg Salychev, Applied Inertial Navigation : Problems and Solutions, BMSTU Press, 2004.
- [5] Peter S. Maybeck, Stochastic Models, Estimation and Control Volume 1, Academic Press, 1979.