論文 J. of The Korean Society for Aeronautical and Space Sciences 45(6), 463-472(2017)
DOI:https://doi.org/10.5139/JKSAS.2017.45.6.463
ISSN 1225-1348(print), 2287-6871(online)

가상의 초기위치를 이용한 SDINS 폐루프 자체 정렬 알고리즘

김태원*

SDINS Closed Loop Self-Alignment Algorithm using Pseudo Initial Position

Taewon Kim*
Agency for Defense Development

ABSTRACT

Inertial Navigation System Alignment is the process to determine direction cosine matrix which is the transformation matrix between the INS body frame and navigation frame. INS initial position value is necessary to INS attitude calculation, so that user should wait until he get such value to start the INS alignment. To remove the waiting time, we propose an alignment algorithm that immediately starts after the INS power on by using pseudo initial position input and then is completed with attitude error compensation by entering true position later. We analyse effect of INS sensor error on attitude in process of time and verify the performance and usefulness of the close-loop alignment algorithm which corrects attitude error from the change of initial position.

초 록

관성항법장치(Inertial Navigation System)는 항법 수행 전 동체 좌표계(body frame)와 항법 좌표계(navigation frame)사이의 좌표 변환 행렬(Direction Cosine Matrix: DCM)을 결정하여 초기자세를 구하는데 이 과정을 정렬(alignment)이라 한다. 정렬을 시작하기 위해서는 INS의 초기 위치 정보가 필요한데 해당 정보가 INS에 미리 입력되어 있지 않거나 당장에 초기위치를 모를 경우 이로 인해 INS에 전원이 인가된 후 정렬에 진입하기까지의 대기시간이 존재한다. 이러한 대기시간을 제거하기 위하여 본 논문에서는 INS 전원 인가 즉시 현재위치와 상이한 가상의 초기위치 값을 장입하여 스트랩다운 INS 정렬을 시작하고 추후에 정확한 위치를 INS에 입력하여 자세오차를 보상하는 정렬 알고리즘을 제시하였다. 항법 좌표계에서의 INS 센서 오차가 시간이 지남에 따라 자세오차에 미치는 영향성을 분석하여 가상의 초기위치 값 입력 시 발생하는 자세오차 만큼을 보상하는 폐루프 정렬 알고리즘의 성능을 검증하였다.

Key Words : Alignment(정렬), SDINS(스트랩다운 관성항법장치), Pseudo position(가상의 위치), Inertial Navigation System(관성항법장치)

† Received : January 26, 2017 Revised : May 24, 2017 Accepted : May 31, 2017
* Corresponding author, E-mail : vorton@add.re.kr

Ⅰ. 서 론

관성항법장치(Inertial Navigation System: INS)는 외부의 도움 없이 항체의 현재 위치, 속도 및 자세를 결정하는 장치이다. INS는 비행체의 회전각속도를 출력하는 자이로와 비행체에 작용하는 비력을 출력하는 가속도계를 장착하고 있어서 내장된 항법컴퓨터가 이를 이용하여 위치, 속도, 자세를 계산하게끔 한다. INS의 형태는 크게 김블형(Gimballed INS)과 스트랩다운형(SDINS)으로 구분할 수 있다. 현대의 INS는 기계적인 복잡성을 피하기 위하여, 가속도계와 자이로의 출력이 동체좌표계에서 이루어지는 스트랩다운 형태로 많이 이용된다.

INS가 외부 장치의 도움 없이 항체의 위치, 속도 및 자세를 획득하기 위해서는 항법좌표계와 동체좌표계 사이의 좌표 변환 행렬을 구하는 과정이 필요한데 이를 INS의 정렬이라 한다. SDINS의 정렬은 방법 및 형태에 따라 전달, 자체 정렬 및 개루프, 폐루프 정렬로 나누어진다. 개루프 정렬은 항법컴퓨터의 시뮬레이션 없이 곧바로 좌표 변환 행렬의 초기 값을 추정한다. 그러나 폐루프 정렬은 항법 알고리즘을 이용하여 항법 좌표계에서의 시뮬레이션을 통해 좌표 변환 행렬을 구한다. 초기 자세 오차 및 센서 오차에 의해 발생한 비정렬 각(misalignment angle)을 항법컴퓨터에서 제어 신호를 인가하여 0으로 수렴시키는 제어 루프를 구성한다[1-2]. SDINS 자체 정렬 성능을 향상시키기 위한 알고리즘 단계에서의 연구가 그동안 많이 진행되어 왔고[3-6] 특별히 정렬 수행시간을 단축하기 위하여 수평축 자세 오차를 이용해 수직축 오차를 즉각 추정하는 방법과 적응형 필터를 적용하는 방법 등이 연구되었다[7-8].

SDINS가 자체 정렬을 시작하기 위해서는 반드시 SDINS의 초기 위치 정보가 필요하기 때문에 Fig 1과 같이 전원인가 후 운용자가 초기위치 정보를 입력하는 대기 시간이 필요하다. 혹은 운용자가 당장에 초기 위치 정보를 모른다면 SDINS는 자체 정렬을 시작할 수 없고 대기시간은 더욱 길어진다.

위와 같은 대기시간을 없애기 위해 본 논문에서는 SDINS의 초기위치정보 없이 임의의 위치로 정렬을 시작하고 추후에 입력된 위치정보로 SDINS의 자세를 보상하는 폐루프 정렬알고리즘을 소개한다. SDINS의 가속도계 및 자이로 오차가 정렬 시작 후 시간에 지남에 따라 자세오차에

Fig. 1. Conventional SDINS Operation before Navigation mode starts

미치는 영향을 분석하여 같은 방법으로 초기 위치 오차가 유발하는 SDINS 자세오차를 계산한다. 이 방법을 통해 SDINS에 전원이 인가되는 즉시 정렬을 시작함으로서 전원인가 시점부터 항법 진입 시점까지의 시간을 단축시킬 수 있다. 더욱이 SDINS의 초기위치정보 부재 시에도 일단 정렬을 시작하고 추후에 획득한 초기위치 정보로 하여금 정렬을 완료하게 할 수 있어서 항법 진입 시점까지의 시간을 절약할 수 있다.

Ⅱ. 기존 및 새로운 정렬 방법

2.1 기존의 자체 정렬 방법

SDINS의 자체정렬은 제어 루프를 구성하는 알고리즘, 정렬 수행시간 등이 INS 제품에 따라 조금씩 다를 수 있으나 일반적으로 그 과정이 개략, 정밀 정렬로 나누어지며 10분 내외로 수행된

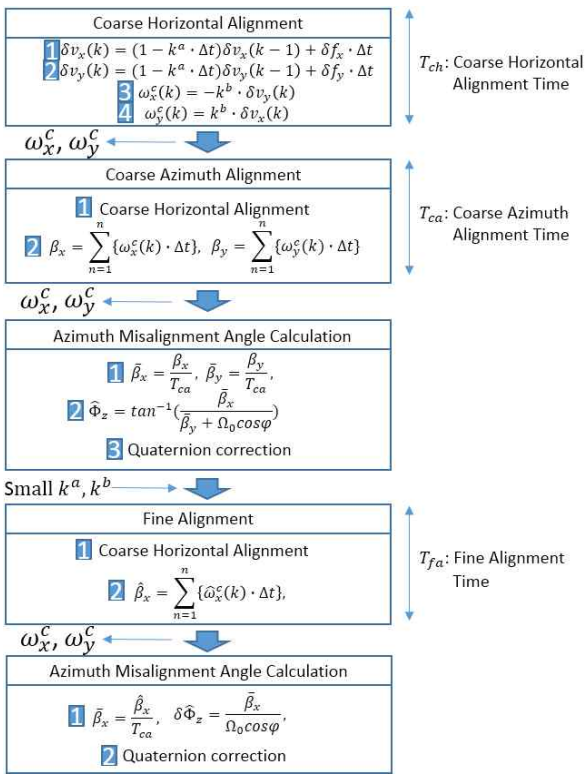

Fig. 2. Self-Alignment functional scheme

다. 예를 들어 본 논문의 연구 검증을 위해 사용된 SDINS는 정렬을 10분간 수행하며 그 과정은 개략 수평 정렬 20초, 개략 방위각 정렬 40초, 루프 제어 안정화 20초, 정밀 방위각 정렬 520초로 이루어져 있다. SDINS 자체 정렬의 자세한 과정은 Fig. 2와 같다. k는 epoch, x, y, z,는 각각 동쪽, 북쪽, 지구타원체 수직 위쪽 방향 축, $\delta v_x, \delta v_y$는 속도 오차, Δt는 1 epoch에 해당하는 시간, ω_x^c, ω_y^c는 개략 정렬 루프 제어 신호, $\hat{\omega}_x^c, \hat{\omega}_y^c$는 정밀 방위각 정렬에서의 루프 제어 신호, k^a, k^b는 루프 제어 이득, $\delta f_x, \delta f_y$는 플랫폼 수평축 가속도,$\beta_x, \beta_y, \overline{\beta}_x, \overline{\beta}_y$는 각각 루프 제어 신호의 합과 평균, Ω_0는 항법좌표계에서의 지구회전각속도, φ는 초기위치의 위도 값, $\hat{\Phi}_z$은 방위각 오차, $\delta\hat{\Phi}_z$은 정밀 방위각 정렬에서의 방위각 오차이다.

2.2 센서 오차가 정렬 오차에 미치는 영향

동체가 정지해 있는 경우에 자이로 바이어스 오차는 SDINS의 자세오차를 유발한다. SDINS의 자체정렬 중 시간에 대한 1차 함수의 시변 센서 오차에 의해 유발되는 자세오차를 수평축과 수직축 정렬 오차로 나누어 분석할 수 있다.

2.2.1 정렬 오차방정식

항법 좌표계에서의 가속도 오차는 정지 상태에서 항법좌표계에서의 중력가속도 g^N이 플랫폼 좌표계(platform frame)로 투영되어 생긴 오차와 가속도 오차가 포함된 동체좌표계에서의 가속도계 출력이 플랫폼 좌표계로 투영되어 생긴 오차의 합과 같다. 즉, 항법좌표계에서의 가속도 오차를 $\delta\dot{v}$, 동체좌표계에서의 가속도계 오차를 $\delta\alpha^B$, 동체좌표계에서 가속도계에 실제로 작용하는 가속도를 f라 하고 $C_B^P \approx C_B^N$을 가정할 때 $\delta\dot{v}$는 다음과 같이 기술된다. 단, 여기서 C_B^P은 동체좌표계로부터 플랫폼좌표계로의 좌표변환행렬, C_B^N는 동체좌표계로부터 항법좌표계로의 좌표변환행렬, $g^N = [0,\ 0,\ g]^T$이다.

$$\begin{aligned}\delta\dot{v} &= (C_N^P \cdot g^N - g^N) + C_B^P \cdot (f^B + \delta\alpha^B) - f^N \\ &\approx (C_N^P - I_{3\times3}) \cdot g^N + C_B^N \cdot (f^B + \delta\alpha^B) - f^N \\ &= (C_N^P - I_{3\times3}) \cdot g^N + C_B^N \cdot \delta\alpha^B\end{aligned} \quad (1)$$

비정렬을 플랫폼 좌표계와 항법좌표계사이의 좌표변환행렬 C_N^P로 정의할 때 비정렬 각 ϕ의 크기가 작을 경우 C_N^P는 다음과 같이 정리된다. 단, 여기서 $\phi = [\phi_x,\ \phi_y,\ \phi_z]^T$이고 $[\phi\times]$을 자세 오차 벡터의 교대행렬로 정의한다.

$$C_N^P = I - [\phi\times] = \begin{bmatrix} 1 & \phi_z & -\phi_y \\ -\phi_z & 1 & \phi_x \\ \phi_y & -\phi_x & 1 \end{bmatrix} \quad (2)$$

항법좌표계에서의 가속도계 센서 오차 $\delta\alpha^N = [\delta\alpha_E,\ \delta\alpha_N,\ \delta\alpha_U]^T$는 $\delta\alpha^N = C_B^N \cdot \delta\alpha^B$라고 하고, 위 식의 C_N^P를 (1)에 대입하여 정리하면 다음과 같다. 단, 가속도계 센서 오차의 아래첨자 E, N, U는 각각 동쪽, 북쪽, 지구타원체 수직 위쪽 방향 축을 말한다.

$$\begin{aligned}\delta\dot{v}_x &= -g\phi_y + \delta\alpha_E \\ \delta\dot{v}_y &= +g\phi_x + \delta\alpha_N \\ \delta\dot{v}_z &= \delta\alpha_U\end{aligned} \quad (3)$$

정지 상태에서 자세 오차 변화량을 기술하기 위해, 먼저 플랫폼 좌표계에 대하여 살펴보면 다음과 같다. 플랫폼 좌표계는 항법컴퓨터에 의해 계산된 항법좌표계로서, 만약 자세 오차가 존재하지 않아 플랫폼 좌표계가 항법좌표계와 일치한다면, 플랫폼 좌표계는 항법좌표계와 동일한 각속도(지구회전각속도)로 회전하게 된다. 그런데, (2)와 같이 자세오차가 존재하면, 플랫폼 좌표계와 항법좌표계에 인가되는 각속도가 상이하고 이 차이는 지속적으로 플랫폼 좌표계의 자세 오차를 유발한다. 자세 오차 ϕ가 존재할 때, (A.5)으로부터 플랫폼 좌표계의 단위시간 당 변화량 또는 자세 오차 변화율 $\dot{\phi} = [\dot{\phi}_x,\ \dot{\phi}_y,\ \dot{\phi}_z]^T$는 다음과 같이 기술된다.

$$\dot{\phi} + C_N^P \omega_{IN}^N = \omega_{IP}^P \quad (4)$$

단, 여기서 ω_{AB}^C는 C좌표계에서 측정한 A→B 좌표계로의 회전 각속도, 좌표변환행렬과 회전각속도의 아래첨자 및 위첨자로 사용되는 B, N, I, P는 각각 동체좌표계, 항법좌표계, 지구중심좌표계(earth-centered inertial frame), 플랫폼 프레임으로 정의한다.

2.2.2 수평축 정렬 알고리즘

폐루프 정렬 알고리즘은 기본적으로 항법을 수행하는 중 플랫폼 좌표계와 항법 좌표계 사이의 비정렬 각(자세 오차)을 0으로 만들기 위한 루프 제어 입력을 인가하는 방식이다. 특히, 수평축 정렬 알고리즘이란 앞서 기술한 항법좌표계 두 수평축에서의 가속도 오차와 자세 오차 변화량을 제어 대상으로 하여 모든 오차를 0으로 수렴토록 하는 제어기 설계 또는 제어기를 의미한

다. 제어기 설계를 위해 (4)의 정렬오차방정식으로부터 수평축 제어신호가 인가되는 폐루프 시스템 상태방정식을 세운다. 정지 상태를 가정하여 (4) 우변의 ω_{IP}^{P}에 대해 다음과 같이 항법좌표계에서의 지구회전각속도 Ω, 입력신호에 의한 루프 제어신호 ω^c, 항법좌표계에서의 자이로 드리프트 $\delta\beta^N$으로 성분을 나눌 수 있다.

$$\begin{aligned}\omega_{IP}^{P} &= \omega_{IN}^{N}+\omega^c+\delta\beta^N \\ &= \omega_{IE}^{N}+\omega_{EN}^{N}+\omega^c+\delta\beta^N \\ &= \omega_{IE}^{N}+\omega^c+\delta\beta^N\end{aligned} \tag{5}$$

$\delta\beta^N=[\delta\beta_E,\ \delta\beta_N,\ \delta\beta_U]^T$을 자이로 오차라 하면, (4)의 자세 오차 방정식은 다음과 같다.

$$\dot{\phi}+C_N^P\Omega=\Omega+\omega^c+\delta\beta^N \tag{6}$$

$\Omega=\begin{bmatrix}0,\ \Omega_N,\ \Omega_U\end{bmatrix}^T$, $\omega^c=\begin{bmatrix}\omega_x^c,\ \omega_y^c,\ 0\end{bmatrix}^T$일 때 (6)에 (2)의 C_N^P을 대입하여 정리하면 다음과 같다.

$$\begin{aligned}\dot{\phi}_x &= -\Omega_N\phi_z+\Omega_U\phi_y+\omega_x^c+\delta\beta_E \\ \dot{\phi}_y &= -\Omega_U\phi_x+\omega_y^c+\delta\beta_N \\ \dot{\phi}_z &= \Omega_N\phi_x+\delta\beta_U\end{aligned} \tag{7}$$

이제, 수평축 자세 오차에서 루프 제어 신호를 제외한 나머지 양을 $\delta\beta_x$와 $\delta\beta_y$로 다음과 같이 정의하자.

$$\begin{aligned}\delta\beta_x &= -\Omega_N\phi_z+\Omega_U\phi_y+\delta\beta_E \\ \delta\beta_y &= -\Omega_U\phi_x+\delta\beta_N\end{aligned} \tag{8}$$

정렬을 위한 제어기 설계를 위해 (1)의 결과에도 루프 제어 입력 신호를 적용하여 (7), (8)와 함께 정리하면, 수평축에서의 가속도 오차와 자세오차 변화량을 독립적인 두개의 2차 시스템으로 나타낼 수 있다. 이를 정렬루프에 대한 오차 방정식으로 정리하면 다음과 같다[1].

$$\begin{bmatrix}\dot{\phi}_y\\ \delta\dot{v}_x\end{bmatrix}=\begin{bmatrix}0 & +k^b\\ -g & -k^a\end{bmatrix}\begin{bmatrix}\phi_y\\ \delta v_x\end{bmatrix}+\begin{bmatrix}\delta\beta_y\\ \delta\alpha_E\end{bmatrix} \tag{9}$$

$$\begin{bmatrix}\dot{\phi}_x\\ \delta\dot{v}_y\end{bmatrix}=\begin{bmatrix}0 & -k^b\\ +g & -k^a\end{bmatrix}\begin{bmatrix}\phi_x\\ \delta v_y\end{bmatrix}+\begin{bmatrix}\delta\beta_x\\ \delta\alpha_N\end{bmatrix} \tag{10}$$

단, 여기서 자세 오차 ϕ_y와 ϕ_x의 변화량에 인가되는 k^a, k^b는 정렬루프 계수이다.

2.2.3 수평축 정렬 오차

정렬의 최종 목표는 항법좌표계에서의 자세 오차를 모두 0으로 하는 것이나, 가관측하지 않은 자세 오차나 센서 오차로 인해서, 정렬은 자세 오차가 아닌 속도 오차 변화량과 자세 오차 변화량이 0이 되는 정상 상태에 머물게 된다. 때문에 정렬에는 항상 오차가 존재하고 여기서 먼저 수평축 정렬 오차에 대해서 기술하도록 한다. 일반적으로 정렬은 개략정렬과 정밀정렬의 두 단계로 나뉘는데, 개략정렬에 의해 방위각이 어느 정도 추정된 이후 정밀정렬이 수행된다. 정밀정렬 수행 중 ϕ가 충분히 작은 값일 때 $\delta\beta_x$와 $\delta\beta_y$은 (8)와 같이 기술된다. 이를 (9), (10)의 수평축 정렬 방정식에 대입하여 정리하면 다음과 같다.

$$\begin{aligned}\begin{bmatrix}\dot{\phi}_y\\ \delta\dot{v}_x\end{bmatrix} &= \begin{bmatrix}0 & +k^b\\ -g & -k^a\end{bmatrix}\begin{bmatrix}\phi_y\\ \delta v_x\end{bmatrix}+\begin{bmatrix}-\Omega_U\phi_x+\delta\beta_N\\ \delta\alpha_E\end{bmatrix} \\ \begin{bmatrix}\dot{\phi}_x\\ \delta\dot{v}_y\end{bmatrix} &= \begin{bmatrix}0 & -k^b\\ +g & -k\end{bmatrix}\begin{bmatrix}\phi_x\\ \delta v_y\end{bmatrix}+\begin{bmatrix}-\Omega_N\phi_z+\Omega_U\phi_y+\delta\beta_E\\ \delta\alpha_N\end{bmatrix}\end{aligned} \tag{11}$$

위 두 식의 양변을 라플라스 변환하면 다음과 같다.

$$\begin{aligned}\begin{bmatrix}\phi_y(s)\\ \delta v_x(s)\end{bmatrix} &= \frac{k^b g}{s^2+k^a s+k^b g}\begin{bmatrix}\dfrac{s+k^a}{k^b g} & +\dfrac{1}{g}\\ -\dfrac{1}{k^b} & \dfrac{s}{k^b g}\end{bmatrix} \\ &\quad\times\begin{bmatrix}\phi_y(0)-\Omega_U\phi_x(s)+\delta\beta_N(s)\\ \delta v_x(0)+\delta\alpha_E(s)\end{bmatrix} \\ \begin{bmatrix}\phi_x(s)\\ \delta v_y(s)\end{bmatrix} &= \frac{k^b g}{s^2+k^a s+k^b g}\begin{bmatrix}\dfrac{s+k^a}{k^b g} & -\dfrac{1}{g}\\ +\dfrac{1}{k^b} & \dfrac{s}{k^b g}\end{bmatrix} \\ &\quad\times\begin{bmatrix}\phi_x(0)-\Omega_N\phi_z(s)+\Omega_U\phi_y(s)+\delta\beta_E(s)\\ \delta v_y(0)+\delta\alpha_N(s)\end{bmatrix}\end{aligned} \tag{12}$$

위 식을 살펴보면, 센서 오차가 없는 경우 초기값에 의한 응답은 정상상태에서 0임을 알 수 있다. 그러므로 충분한 정밀정렬 시간을 가정하면, 정렬 완료 시점에서 정렬 오차는 센서 오차에 의한 영향만을 고려하면 된다. (12)에서 수직축 자세 오차와 센서 오차는 모두 상수, ϕ_z^0는 수직축 자세오차, ϕ_x^0,ϕ_y^0는 각각 x,y축에 해당하는 수평축 자세 오차, $\delta\alpha_E^0,\delta\alpha_N^0$는 각각 동쪽, 북쪽 방향의 축 가속도계 바이어스 오차, $\delta\beta_E^0,\delta\beta_N^0$는 각각 동쪽, 북쪽 방향의 축 자이로 바이어스 오차라 가정하면 정렬 후 정상상태의 수평축 자세 오차 $\phi_{x,\infty},\phi_{y,\infty}$와 속도 오차 $\delta v_{x,\infty},\delta v_{y,\infty}$는 다음과 같다.

$$\begin{aligned}\phi_{y,\infty} &= -\frac{k^a}{k^b g}(\Omega_U\phi_x^0-\delta\beta_N^0)+\frac{1}{g}\delta\alpha_E^0 \\ \phi_{x,\infty} &= -\frac{k^a}{k^b g}(\Omega_N\phi_z^0-\Omega_U\phi_y^0-\delta\beta_E^0)-\frac{1}{g}\delta\alpha_N^0 \\ \delta v_{x,\infty} &= \frac{1}{k^b}(\Omega_U\phi_x^0-\delta\beta_N^0) \\ \delta v_{y,\infty} &= -\frac{1}{k^b}(\Omega_N\phi_z^0-\Omega_U\phi_y^0-\delta\beta_E^0)\end{aligned} \tag{13}$$

센서 오차의 시간에 대한 정확한 함수 또는 그 라플라스 변환을 알고 있다면, 정렬 시간의 함수로 정렬 후의 수평축 자세 오차를 계산 할 수 있다. 그러나 모든 가능한 경우에 대한 자세 오차를 구하기는 현실적으로 어려우므로 여기서는 상수 오차 이외에 시간에 대해 1차 함수로 증가하는 센서 오차에 대한 자세 오차를 구한다. 먼저 전달함수가 $k^b g/(s^2+ks+k^b g)$이고 입력이 시간의 1차 함수 $A\times t$인 시스템의 출력을 $y(t)$라 하면 $y(t)$의 라플라스 변환 $Y(s)$는 다음과 같다.

$$Y(s)=\frac{s^2+ks+k^b g-s^2-ks}{s^2+ks+k^b g}\mathcal{L}\,[At] \tag{14}$$

위 식을 $Y(s)-\mathcal{L}\,[At]$에 대해서 정리하면 다음과 같다.

$$Y(s)-\mathcal{L}\,[At]=-\frac{s^2+ks}{s^2+ks+k^b g}\frac{A}{s^2} \tag{15}$$

$y(t)-At=\mathcal{L}^{-1}[Y(s)-\mathcal{L}\,[At]]$의 정상상태 응답은 다음과 같다.

$$\lim_{t\to\infty}(y(t)-At)=-\lim_{s\to 0}s\frac{s^2+ks}{s^2+ks+k^b g}\frac{A}{s^2}=-\frac{k}{k^b g}A \tag{16}$$

즉 $y(t)$는 오차의 정상상태 관점에서 상수오차 $-kA/k^b g$를 가지고 입력 $A\times t$를 추종하게 되므로 $y(t)$는 시간의 함수로 다음과 같다.

$$y(t)=At-\frac{k}{k^b g}A \tag{17}$$

단, 여기서 오차가 정상상태에 도달한 후의 시간만을 고려한다. 한편, 연산자 s는 시간 영역에서 1차 미분과 동일하므로 입력이 1차 함수 At인 경우에 주파수 영역에서 연산자 s가 곱해지면 시간영역에서의 상수 입력 A와 동일하게 작용한다. 그러므로 (12)에서 정렬 중 방위각 오차 ϕ_z와 센서 오차가 각각 시간에 대한 1차 함수로 써, 그 기울기가 $\phi_z^1, \delta\beta_N^1, \delta\alpha_E^1, \delta\beta_E^1, \delta\alpha_N^1$의 상수이고 초기치가 모두 0이라면 t초간 정렬 후 정상상태의 수평축 자세 오차는 다음과 같다.

$$\begin{aligned}\phi_{y,t}&=\left(\frac{k^a}{k^b g}\delta\beta_N^1+\frac{1}{g}\delta\alpha_E^1\right)t\\&\quad+\frac{1}{k^b g}\delta\beta_N^1-\frac{(k^a)^2}{(k^b g)^2}\delta\beta_N^1-\frac{k^a}{k^b g^2}\alpha_E^1\\\phi_{x,t}&=\left(-\frac{k^a}{k^b g}(\Omega_N\phi_z^1-\delta\beta_E^1)-\frac{1}{g}\delta\alpha_N^1\right)t+\frac{k^a}{k^b g^2}\alpha_N^1\\&\quad-\frac{1}{k^b g}(\Omega_N\phi_z^1-\delta\beta_E^1)+\frac{(k^a)^2}{(k^b g)^2}(\Omega_N\phi_z^1-\delta\beta_E^1)\end{aligned} \tag{18}$$

단, 여기서 t는 오차 $\phi_{x,t}, \phi_{y,t}$가 정상상태에 도달한 시점부터의 시간이다. 즉, $\phi_{x,t=0}, \phi_{y,t=0}$은 정상상태의 오차이다.

2.2.4 방위각 정렬 알고리즘 및 오차

수평축 정렬 알고리즘의 기술에서 ω_x^c와 ω_y^c로 표현되는 루프 제어 신호는 (13), (14)에서 각각 u_2, u_1의 성분으로 사용된다. 실제의 구현에서 현재 항법좌표계에서의 속도오차에 비례하는 크기의 루프 제어 신호가 자세 계산 알고리즘에 입력된다.

$$\begin{aligned}\omega_y^c&=+k^b\delta v_x\\\omega_x^c&=-k^b\delta v_y\end{aligned} \tag{19}$$

방위각 정렬 알고리즘 수행 시 루프 제어 신호를 이용하여 방위각 계산이 수행된다. 먼저 수평축 정렬 시스템이 정상상태에 도달하여 수평축 가속도 오차가 0이 된다. 정밀한 방위각 계산을 위해 수평축 자세오차에 비해 수직축 자세오차는 여전히 큰 값을 가진다고 가정할 때 C_N^P에 대해 다시 기술하면 다음과 같다.

$$C_N^P=\begin{bmatrix}\cos\phi_z & \sin\phi_z & -\phi_y\\ -\sin\phi_z & \cos\phi_z & \phi_x\\ \phi_y\cos\phi_z+\phi_x\sin\phi_z & \phi_y\sin\phi_z-\phi_x\cos\phi_z & 1\end{bmatrix} \tag{20}$$

새롭게 정의된 C_N^P를 (6)에 대입하여 정리하면 다음과 같다.

$$\begin{aligned}\dot{\phi}_x&=-\Omega_N\sin\phi_z+\Omega_U\phi_y+\omega_x^c+\delta\beta_E\\\dot{\phi}_y&=\Omega_N-\Omega_N\cos\phi_z-\Omega_U\phi_x+\omega_y^c+\delta\beta_N\\\dot{\phi}_z&=-\Omega_N\phi_y\sin\phi_z+\Omega_N\phi_x\cos\phi_z+\delta\beta_U\end{aligned} \tag{21}$$

수평축 정렬 시스템이 정상상태에 도달한 것을 고려하여 $\dot{\phi}_x=\dot{\phi}_y=0$, $\Omega_U\phi_y, \Omega_U\phi_x \approx 0$을 가정하고 센서 오차를 무시하면, 루프 제어 신호는 다음과 같이 써질 수 있다.

$$\begin{aligned}\omega_x^c&\approx\ \Omega_N\sin\phi_z\\\omega_y^c&\approx-\Omega_N+\Omega_N\cos\phi_z\end{aligned} \tag{22}$$

(22)을 ϕ_z의 함수로 정리하면 각각 다음과 같다.

$$\begin{aligned}\Omega_N\sin\phi_z&=\omega_x^c\\\Omega_N\cos\phi_z&=\omega_y^c+\Omega_N\end{aligned} \tag{23}$$

즉, (23)를 정리하여 $\tan^{-1}$ 취하면 방위각 오차를 구할 수 있다.

$$\hat{\phi}_z=\tan^{-1}\frac{\omega_x^c}{\omega_y^c+\Omega_N} \tag{24}$$

실제 알고리즘 구현에서는 측정 잡음의 영향을 제거하기 위하여 루프 제어 신호를 평균 취한 값이 사용된다. 즉, 정렬 시간을 T라 하면 먼저 루프 제어 신호의 시간에 대한 평균은 다음과 같다.

$$\overline{\beta}_x = \frac{1}{T}\int_0^T \omega_x^c dt \quad \text{and} \quad \overline{\beta}_y = \frac{1}{T}\int_0^T \omega_y^c dt \tag{25}$$

$\overline{\beta}_x$와 $\overline{\beta}_y$로부터 (24)의 방위각 오차 추정식은 다음과 같이 다시 써질 수 있다.

$$\hat{\phi}_{z,coarse} = \tan^{-1}\frac{\overline{\beta}_x}{\overline{\beta}_y + \Omega_N} \tag{26}$$

수평축 정렬은 제어 개념을 도입하여 자세 오차가 0으로 수렴하도록 구성되어 있는 반면, 방위각 정렬은 루프 제어 신호도 출력을 이용하여 추정하는 개념으로써 수평축 정렬 방식과 차이가 있다. 즉, 수평축 정렬은 정렬 중 자세 오차를 지속적으로 0으로 수렴하도록 동작하는 반면, 방위각 정렬은 정렬 수행 동안 (26)을 통해 추정치를 계산만 하고 항법 진입 직후 방위각 오차 추정치를 업데이트 한다. 한편, 방위각 정렬은 위의 계산 방식의 개략방위각 정렬과 개략방위각 정렬 후 정밀 방위각 정렬의 두 단계로 수행된다. (26)에서 'coarse'란 개략방위각정렬을 의미한다. 개략방위각 정렬 후 방위각 오차 ϕ_z는 작은 값이어서, $\cos\phi_z \approx 1$과 $\sin\phi_z \approx \phi_z$을 만족한다고 가정하면, 식 (22)에서 $\omega_x^c \approx \Omega_N\phi_z$, $\omega_y^c \approx 0$이다. 그러므로 정밀 방위각 정렬 계산은 다음과 같이 간단하게 계산될 수 있다.

$$\hat{\phi}_{z,fine} = \frac{\beta_x^c}{\Omega_N} \tag{27}$$

단, 여기서 β_x^c는 정밀방위각 정렬 중의 루프 제어 신호 평균치이고, 평균치 계산 위한 적분은 속도 오차 또는 수평축 자세 오차의 과도 응답 상태를 제외하기 위해 정밀 정렬 시작 후 일정 시간이 흘러 안정화 된 이후부터 수행된다. 한편, (21)에서 정밀방위각 정렬 중 수평축 자세 오차와 방위각 오차가 작은 값이라면 수직축 자세 오차 식은 근사적으로 $\dot{\phi}_z \approx \delta\beta_U$이므로, 센서 오차 $\delta\beta_U$가 정렬 중 상수라고 가정하면 정렬 중 수직축 자세 오차 $\phi_z(t)$는 다음과 같다.

$$\phi_z(t) \approx \delta\beta_U^0 t + \phi_z^0 \tag{28}$$

수직축 자세 오차는 시간에 대한 1차 함수이고, $\delta\beta_E$와 $\delta\alpha_N$ 역시 시간에 대한 1차 함수라고 가정하면 (13)와 (18)로부터 정상상태에서 δv_y 역시 시간에 대한 1차 함수가 된다. 따라서 (19)의 $\omega_x^c = -k^b\delta v_y$는 다음과 같이 정리된다.

$$\begin{aligned}\omega_x^c = {} & \Omega_N\left(\delta\beta_U^0 t + \phi_z^0\right) - \delta\beta_E^1 t - \delta\beta_E^0 \\ & - \frac{k}{k^b g}\left(\Omega_N\delta\beta_U^0 - \delta\beta_E^1\right) - \frac{1}{g}\delta\alpha_N^1 - \Omega_U\phi_y^0\end{aligned} \tag{29}$$

(29)를 (25)에 대입하여 얻은 $\overline{\beta}_x$를 이용해서 (27)을 정리하면 방위각 오차 추정치 $\hat{\phi}_{z,fine}$을 얻을 수 있다. 즉, 수직축 자세오차, 방위각 추정을 위한 정밀 정렬 중 수직축 자세 오차가 시간에 대한 1차 함수로써 근사적으로 $\delta\beta_U^0 t + \phi_z^0$이고, 센서 오차 $\delta\beta_E$와 $\delta\alpha_N$도 시간에 대한 1차 함수여서 각각 $\delta\beta_E^1 t + \delta\beta_E^0$, $\delta\alpha_N^1 t + \delta\alpha_N^0$이면, T_f 시간 정밀 정렬을 했을 때에 방위각 오차 추정치 $\hat{\phi}_{z,fine}$와 추정하지 못한 방위각 오차 $\delta\psi$는 다음과 같다.

$$\begin{aligned}\hat{\phi}_{z,fine} = {} & \frac{\delta\beta_U^0 T_f}{2} + \phi_z^0 - \frac{\delta\beta_E^1 T_f}{2\Omega_N} - \frac{\delta\beta_E^0}{\Omega_N} \\ & - \frac{k^a}{k^b g}\left(\delta\beta_U^0 - \frac{\delta\beta_E^1}{\Omega_N}\right) - \frac{1}{\Omega_N g}\delta\alpha_N^1 - \phi_y^0\tan\varphi \\ \delta\psi = \hat{\phi}_{z,fine} - \phi_z = {} & -\frac{\delta\beta_U^0 T_f}{2} - \frac{\delta\beta_E^1 T_f}{2\Omega_N} - \frac{\delta\beta_E^0}{\Omega_N} \\ & - \frac{k}{k^b g}\left(\delta\beta_U^0 - \frac{\delta\beta_E^1}{\Omega_N}\right) - \frac{1}{\Omega_N g}\delta\alpha_N^1 - \phi_y^0\tan\varphi\end{aligned} \tag{30}$$

2.3 새로운 정렬방법

2.3.1 센서 오차 모델링

본 논문이 제시하는 새로운 정렬 방법은 임의의 초기위치로 SDINS 정렬을 시작한 뒤, 이후에 정확한 초기위치 값을 획득했을 때에 위도 값 차

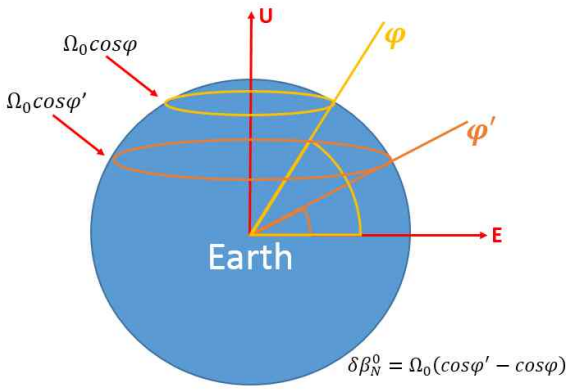

Fig. 3. The influence of Initial position error

이로 인해 발생한 자세 오차만큼을 보상해주는 것이다. 임의의 초기위치를 입력했을 때 발생하는 위도 오차가 가속도계 출력에 미치는 영향은 미미하므로 다음과 같이 나타낼 수 있다.

$$\delta\alpha_E^0, \delta\alpha_N^0, \delta\alpha_E^1, \delta\alpha_N^1 \approx 0 \tag{31}$$

그러나 Fig. 3을 보면 위도 오차의 영향이 자이로 출력에는 바이어스 형태로 나타나는 것을 확인할 수 있으므로 초기위치 위도 값을 φ, 임의의 초기위치 위도 값을 φ'라고 했을 때

$$\begin{aligned}&\delta\beta_E^0, \delta\beta_E^1, \delta\beta_N^1 = 0\\ &\delta\beta_N^0 = \Omega_0(\cos\varphi' - \cos\varphi)\\ &\delta\beta_U^0 = \Omega_0(\sin\varphi' - \sin\varphi)\end{aligned} \tag{32}$$

으로 모델링 할 수 있다. ϕ_z^1는 (28)에 따라 $\phi_z^1 = \delta\beta_U^0$로 표현할 수 있다. 위의 모델링된 센서 오차에서 알 수 있듯이 정렬 후 초기위치 오차에 의한 자세 오차는 오직 위도 변화에 의해서만 영향을 받고 경도와 고도 변화로부터는 영향을 받지 않는다.

2.3.2 새로운 정렬 알고리즘의 자세 오차 보상

최종적으로 SDINS가 t 시간동안 정렬한 후의 수평축 자세오차 ϕ_x, ϕ_y는 센서 오차를 상수로 간주한 (13)에서의 $\phi_{x,\infty}, \phi_{y,\infty}$와 시간에 대한 1차함수로 간주한 (29)에서의 $\phi_{x,t}, \phi_{y,t}$의 합으로 표현할 수 있다. 또한 t 시간동안 정렬한 후의 수직축 자세오차 ϕ_z를 (30)로부터 구하면 결과적으로 정리된 자세오차는 다음과 같다.

$$\begin{aligned}\phi_x &= \phi_{x,t} + \phi_{x,\infty}\\ &= \left(-\frac{k^a}{k^b g}(\Omega_N\phi_z^1 - \delta\beta_E^1) - \frac{1}{g}\delta\alpha_N^1\right)t + \frac{k^a}{k^b g^2}\alpha_N^1\\ &\quad - \frac{1}{k^b g}\left(\Omega_N\phi_z^1 - \delta\beta_E^1\right) + \frac{(k^a)^2}{(k^b g)^2}\left(\Omega_N\phi_z^1 - \delta\beta_E^1\right)\\ &\quad - \frac{k^a}{k^b g}(\Omega_N\phi_z^0 - \Omega_U\phi_y^0 - \delta\beta_E^0) - \frac{1}{g}\delta\alpha_N^0\end{aligned} \tag{33}$$

$$\begin{aligned}\phi_y &= \phi_{y,t} + \phi_{y,\infty}\\ &= \left(\frac{k^a}{k^b g}\delta\beta_N^1 + \frac{1}{g}\delta\alpha_E^1\right)t\\ &\quad + \frac{1}{k^b g}\delta\beta_N^1 - \frac{(k^a)^2}{(k^b g)^2}\delta\beta_N^1 - \frac{k^a}{k^b g^2}\alpha_E^1\\ &\quad - \frac{k^a}{k^b g}(\Omega_U\phi_x^0 - \delta\beta_N^0) + \frac{1}{g}\delta\alpha_E^0\end{aligned} \tag{34}$$

$$\begin{aligned}\phi_z &= \hat{\phi}_{z,fine} + \frac{\delta\beta_U^0 T_f}{2} + \frac{\delta\beta_E^1 T_f}{2\Omega_N} + \frac{\delta\beta_E^0}{\Omega_N}\\ &\quad + \frac{k^a}{k^b g}\left(\delta\beta_U^0 - \frac{\delta\beta_E^1}{\Omega_N}\right) + \frac{1}{\Omega_N g}\delta\alpha_N^1 + \phi_y^0\tan\varphi\end{aligned} \tag{35}$$

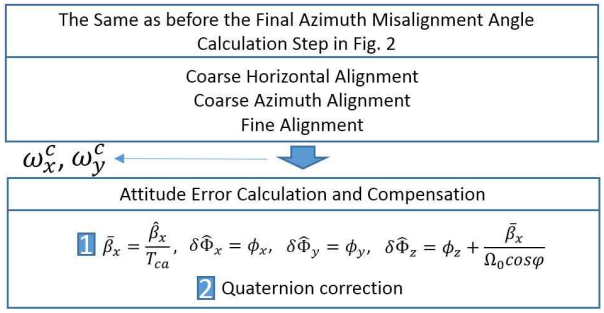

Fig. 4. New Alignment Functional Scheme

따라서 $t = T_f$ 시간동안 정밀 방위각 정렬을 수행한 후 발생된 자세오차 ϕ_x, ϕ_y, ϕ_z를 추정하고자 할 때 (31)와 (32)를 이용하고 $T_f \gg 1$임을 고려하면 (33), (34), (35)를 근사화할 수 있다. 근사화 후 추정 가능한 성분들로 ϕ_x, ϕ_y, ϕ_z를 구성하면 최종적으로 식(36)과 같다. 단, 여기서 $\delta\beta_N^0, \delta\beta_U^0$는 (32)를 따른다.

$$\begin{aligned}\phi_x &\approx -\frac{k^a}{k^b g}\Omega_N\phi_z^1 T_f\\ \phi_y &\approx \frac{k^a}{k^b g}\delta\beta_N^0\\ \phi_z &\approx \hat{\phi}_{z,fine} + \frac{\delta\beta_U^0 T_f}{2} + \frac{k^a}{k^b g}\delta\beta_U^0 + \phi_y^0\tan\varphi\end{aligned} \tag{36}$$

새로운 정렬 알고리즘이 기존의 자체정렬과 다른 점은 기존의 개략 정렬, 정밀 정렬 모두 동일하나 마지막에 자세오차를 보상하는 단계가 추가된다는 점에서 다르다. Fig. 4와 같이 정밀방위각 정렬이 끝난 후 초기위치 오차에 의해 발생된 자세오차 $\phi_x, \phi_y, \phi_z - \hat{\phi}_{z,fine}$을 기존의 수직축 자세오차 추정치 $\bar{\beta}_x/(\Omega_0\cos\varphi)$와 함께 보상한다.

2.4 새로운 정렬 알고리즘 탑재 시험 결과

본 논문에서 소개한 새로운 알고리즘을 자체 개발한 중급 SDINS에 탑재하여 Table 1의 위치에서 정렬을 수행하였고 기존의 자체 정렬 알고리즘과 성능을 비교하였다. 정렬 알고리즘은 20초간 개략 수평 정렬, 40초간 개략 방위각 정렬, 20초간 제어 안정화, 520초간 정밀 방위각 정렬의 단계를 거쳐 총 10분간 정렬을 수행한다.

새로운 알고리즘과 기존의 자체정렬 알고리즘

Table 1. True SDINS Position

	Lattitude (deg)	Longitude (deg)	Height (m)
Experimental Position	36.411078	127.328247	249.5

모두 Table 3에 소개된 가상의 초기위치로 정렬을 시작하여 매 시험마다 가상의 위도 값을 변경해 가며 자세오차를 측정한다. 단, 새로운 알고리즘의 경우 Fig. 5와 같이 정확히 정렬 시작 400초 후에 Table 1의 실제 초기위치를 장입한다.

Table 2. Reference SDINS Attitude

	Roll(deg)	Pitch(deg)	Yaw(deg)
SDINS Attitude	0.172746	0.57462	-61.5422

Table 3. Definition of Latitude Error

Pseudo Initial Position Latitude Value(deg)	Latitude Error (deg)	Pseudo Initial Position Latitude Value(deg)	Latitude Error (deg)
-83.588922	-120	6.411078	-30
-73.588922	-110	16.411078	-20
-63.588922	-100	26.411078	-10
-53.588922	-90	36.411078	0
-43.588922	-80	46.411078	10
-33.588922	-70	56.411078	20
-23.588922	-60	66.411078	30
-13.588922	-50	76.411078	40
-3.588922	-40	86.411078	50

Table 4. Attitude Error of New Algorithm due to a Wide Range of Latitude Change

Latitude Error(deg)	Roll Error (deg)	Pitch Error (deg)	Yaw Error (deg)
-120	-0.000278	0.000138	0.0210
-110	-0.000192	0.000095	0.0151
-100	-0.000111	0.000055	0.0095
-90	-0.000042	0.000021	0.0046
-80	0.000011	-0.000006	0.0008
-70	0.000047	-0.000024	-0.0018
-60	0.000065	-0.000034	-0.0032
-50	0.000067	-0.000035	-0.0036
-40	0.000056	-0.000030	-0.0031
-30	0.000041	-0.000022	-0.0023
-20	0.000023	-0.000013	-0.0012
-10	0.000008	-0.000004	-0.0004
0	0	0	0
10	0.000001	0	-0.0002
20	0.000012	-0.000005	-0.0011
30	0.000031	-0.000015	-0.0025
40	0.000056	-0.000028	-0.0042
50	0.000081	-0.000042	-0.0060

400초는 임의로 선정한 값이다. 실제 초기위치 장입 시점과 관계없이 정밀 방위각 정렬 종료 시점에 자세오차 보상을 수행하므로 400초가 아닌 다른 값을 이용해도 동일한 결과를 얻게 된다.

Table 4에서 알 수 있듯이 시험 결과 새로운 정렬 알고리즘은 가상의 초기위치 위도 값으로 설정된 -83°~86°의 구간에서 모든 자세오차 값이 수평오차 0.1 mil 이내, 방위각 오차 1 mil 이내의 성능을 보여주었다. 가상의 초기위치 입력에 대해 발생한 자세오차에 대해서 기존의 알고리즘과 새로운 알고리즘을 각각 Self Alignment와 New Alignment로 Fig. 6, 7, 8에 도시하였다.

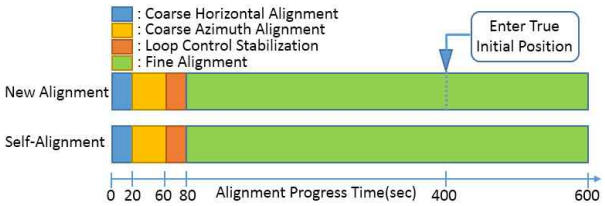

Fig. 5. Comparison between Conventional Self-Alignment Algorithm and New Algorithm

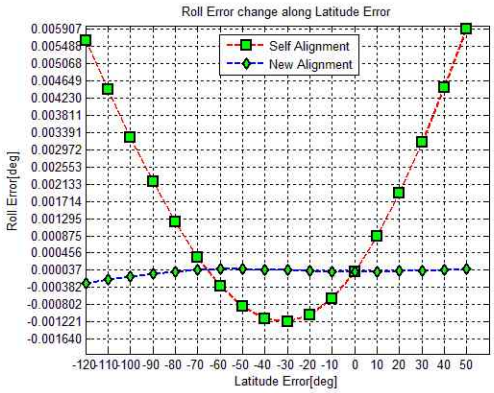

Fig. 6. Roll Error Change along Latitude Error

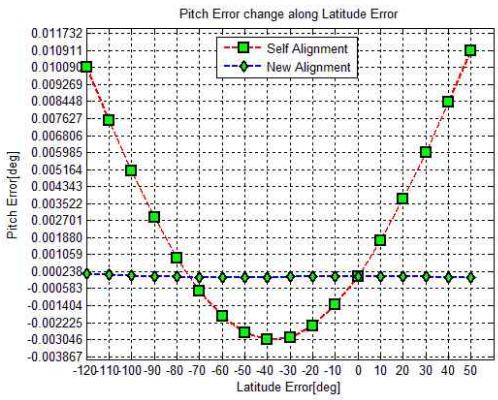

Fig. 7. Pitch Error Change along Latitude Error

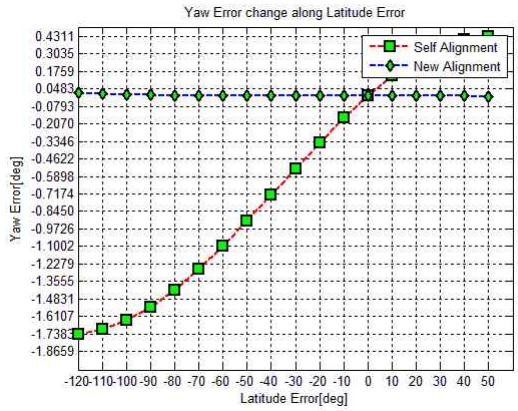

Fig. 8. Yaw Error Change along Latitude Error

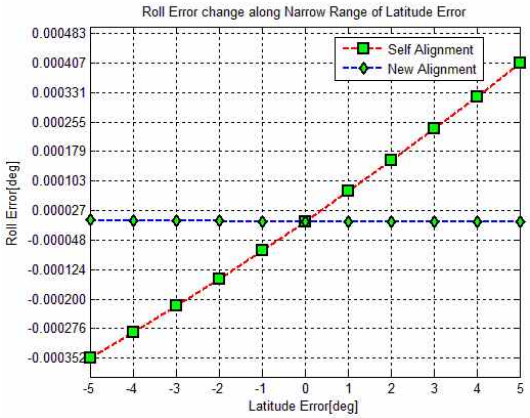

Fig. 9. Roll Error Change along Narrow Range of Latitude Error

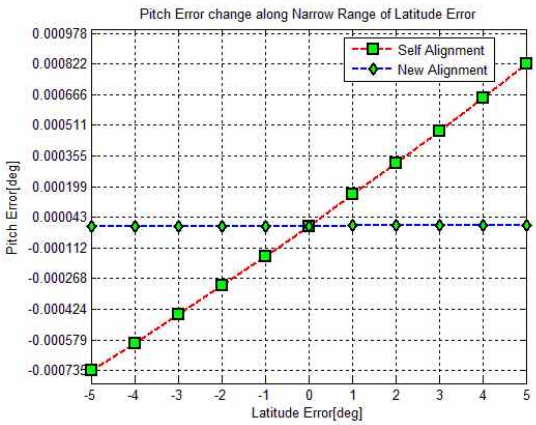

Fig. 10. Pitch Error Change along Narrow Range of Latitude Error

그래프를 보면 기존의 자체정렬은 위도 변화에 따라 자세오차가 크게 변화하지만 새로운 알고리즘은 자세오차가 매우 작게 변화하는 것을 확인할 수 있다. 자체정렬 시 초기위치 오차에 따른

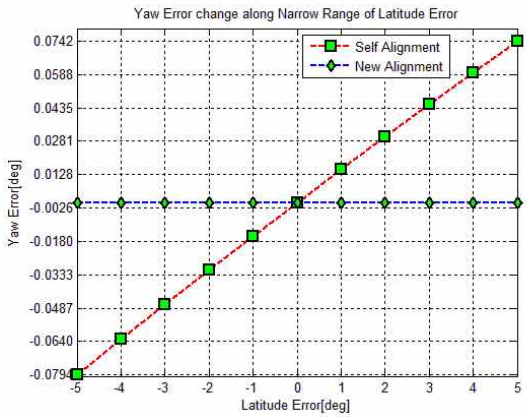

Fig. 11. Yaw Error Change along Narrow Range of Latitude Error

Table 5. Attitude Error of New Algorithm due to a Narrow Range of Latitude Change

Latitude Error(deg)	Roll Error (deg)	Pitch Error (deg)	Yaw Error (deg)
-5	0.172749	0.574619	-61.5423
-4	0.172748	0.574619	-61.5423
-3	0.172747	0.574619	-61.5423
-2	0.172747	0.57462	-61.5422
-1	0.172746	0.57462	-61.5422
0	0.172746	0.57462	-61.5422
1	0.172745	0.574621	-61.5422
2	0.172745	0.574621	-61.5422
3	0.172745	0.574621	-61.5422
4	0.172745	0.574621	-61.5422
5	0.172745	0.574621	-61.5422

자세오차가 오직 위도에 의해서만 영향을 받는다는 사실을 고려할 때 새로운 알고리즘은 지구상의 위치와 관계없이 거의 대부분의 지역에서 사용할 수 있음을 알 수 있다. SDINS 초기위치 위도 오차 ±5°이내의 미시적인 영역에서 위도 변화에 따른 정렬 후 수평축 자세오차를 관찰하면 Table 5와 같이 새로운 알고리즘은 0.002mil 수준의 미세한 자세오차도 보상할 수 있다.

III. 결 론

본 논문에 소개된 새로운 정렬 알고리즘이 실험 결과를 통해 기존의 자체정렬과 거의 동일한 성능을 보이는 것이 확인되었다. 또한 SDINS에 전원을 인가하는 즉시 정렬을 시작할 수 있으므

로 전원 인가 후 초기위치를 장입할 때까지의 시간을 절약할 수 있다. 지구상 어느 곳에 위치하든 일시적으로 초기위치를 알 수 없는 상황에서도 일단 정렬을 수행한 후 추후 획득한 초기 위치를 이용하여 곧바로 SDINS의 자세를 얻을 수 있다. 이 같은 효율성 및 효용성으로 본 논문이 소개한 알고리즘은 다양한 분야에 활용도 높게 사용될 것으로 예상된다.

References

1) Oleg S. Salychev, "Applied Inertial Navigation: Problems and Solutions", BMSTU Press, Moscow, Russia, 2004.

2) D. H. Titterton and J. L. Weston, " Strapdown inertial navigaton technology", Peter Peregrinus Ltd, London, United Kingdom, 1997.

3) K. R. Britting and T. Palsson, "Self-alignment techniques for strapdown inertial navigation systems with aircraft application", *Journal of Aircraft*, Vol. 7, 1970, pp.302～307.

4) Lee, J. G., Park, C. G., and Park, H. W., "Multiposition alignment of strapdown inertial navigation system", *IEEE Transaction on Aerospace and Electronic Systems*, Vol. 29, Oct. 1970, pp.1323～1328.

5) W. Gao, Y. Ben, and X. Zhang, "Rapid Fine Strapdown INS Alignment Method under Marine Mooring Condition", *IEEE Transaction on Aerospace and Electronic Systems*, Vol. 47, Oct. 2011, pp.2887～2896.

6) I. Y. Bar-Itzhack, N. Berman, "Control Theoretic Approach to Inertial Navigation Systems", *Journal of Guidance, Control, and Dynamics*, vol. 11. no. 3, 1988. pp. 237-245.

7) Ahn, H. S. and Won, C. H., "Fast alignment using rotation vector and adaptive Kalman filter", *IEEE Transaction on Aerospace and Electronic Systems*, Vol. 42, Jan. 2006, pp.70～83.

8) Fang, J. C. and Wan, D. J., "A fast initial method for strapdown inertial navigation system on stationary base", *IEEE Transaction on Aerospace and Electronic Systems*, Vol. 32, Oct. 1996, pp.1501～1504.

부 록

$\Delta C = C_B^N - C_B^P$으로 정의하여 (4)를 수식적으로 유도하기 위해 먼저 $\left[\omega_{PN}^P\times\right]$을 좌표변환행렬의 미분 방정식을 이용하여 기술하면 다음과 같다. 단, 여기서 $[\omega\times]$는 각속도 ω의 교대행렬이다.

$$\begin{aligned}\left[\omega_{PN}^P\times\right] &= C_N^P\left[\omega_{PN}^N\times\right]C_P^N \\ &= C_N^P\left(\left[\omega_{IN}^N\times\right]+\left[\omega_{PI}^N\times\right]\right)C_P^N \\ &= \left(C_N^P\left[\omega_{IN}^N\times\right]-\left[\omega_{IP}^P\times\right]C_N^P\right)C_P^N \\ &= C_N^P\left[\omega_{IN}^N\times\right]C_P^N-\left[\omega_{IP}^P\times\right]\end{aligned} \tag{A.1}$$

또한 비정렬 각 ϕ의 크기가 작을 때 $C_N^P = I-[\phi\times]$을 만족하므로 ΔC에 대하여 $\Delta C = C_B^N - C_N^P C_B^N = \left(I-C_N^P\right)C_B^N = [\phi\times]C_B^N$을 만족한다. $\dot{C}_B^N = C_B^N\left[\omega_{NB}^B\times\right] = C_B^N\left(\left[\omega_{NP}^B\times\right]+\left[\omega_{PB}^B\times\right]\right)$이고 $\dot{C}_B^P = C_B^P\left[\omega_{PB}^B\times\right]$임을 감안할 때 $\Delta\dot{C}$는 다음과 같이 정리할 수 있다.

$$\begin{aligned}\Delta\dot{C} &= [\dot{\phi}\times]C_B^N+[\phi\times]\dot{C}_B^N \\ &= [\dot{\phi}\times]C_B^N+[\phi\times]C_B^N\left(\left[\omega_{NP}^B\times\right]+\left[\omega_{PB}^B\times\right]\right) \\ &\approx [\dot{\phi}\times]C_B^N+[\phi\times]C_B^N\left[\omega_{PB}^B\times\right]\end{aligned} \tag{A.2}$$

(A.2)에서 ω_{NP}^B의 크기가 ω_{PB}^B에 비해 많이 작다고 가정하여 그 영향을 무시한다. $\Delta\dot{C}$을 $\dot{C}_B^N$와 $\dot{C}_B^P$의 차이로 정리하면 다음과 같다.

$$\begin{aligned}\Delta\dot{C} &= \dot{C}_B^N-\dot{C}_B^P \\ &= C_B^N\left[\omega_{NP}^B\times\right]+\left(C_B^N-C_B^P\right)\left[\omega_{PB}^B\times\right] \\ &= C_B^N\left[\omega_{NP}^B\times\right]+[\phi\times]C_B^N\left[\omega_{PB}^B\times\right]\end{aligned} \tag{A.3}$$

(A.2)과 (A.3)은 동치로 기술하고 이 식을 정리하면 다음과 같다.

$$\begin{aligned}&C_B^N\left[\omega_{NP}^B\times\right]C_N^B = \left[\omega_{NP}^N\times\right] = [\dot{\phi}\times] \\ &\Leftrightarrow \left[\omega_{PN}^P\times\right] = -\left[\omega_{NP}^N\times\right] = -[\dot{\phi}\times] \\ &\Leftrightarrow \omega_{PN}^P = -\dot{\phi}\end{aligned} \tag{A.4}$$

(A.4)의 결과를 (A.1)에 대입하여 정리하면 아래와 같은 일반적인 정렬 오차방정식을 얻을 수 있다.

$$\dot{\phi}+C_N^P\omega_{IN}^N = \omega_{IP}^P \tag{A.5}$$