



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

2017년 2월

석사학위 논문

영상기반 항법시스템 개발

조선대학교 대학원

항공우주공학과

김 종 명

영상기반 항법시스템 개발

Vision Based Inertial Navigation System

2017년 2월 24일

조선대학교 대학원

항공우주공학과

김 종 명

영상기반 항법시스템 개발

지도교수 이 현 재

이 논문을 공학석사학위 신청 논문으로 제출함

2016년 10월

조선대학교 대학원

항공우주공학과

김 종 명

김종명의 석사학위논문을 인준함

위원장 조선대학교 교수 이 상 기 (인)

위 원 조선대학교 교수 이 현 재 (인)

위 원 조선대학교 교수 김 태 규 (인)

2016년 11월

조선대학교 대학원

목 차

ABSTRACT

I . 서론	1
II. 항법 시스템	4
1. 항법시스템 소개	4
1) 좌표계의 정의	4
2) 좌표계의 변환	5
3) 운동방정식	7
III. 비선형 필터	12
1. Unscented Filtering	12
1) Unscented Kalman Filter(UKF)	12
2) UKF for Additive Noise Case	14
3) Uncorrelated Unscented Filtering(UUF)	16
IV. 영상기반 항법시스템	20
1. GPS/INS 항법시스템	20
1) 약결합방식(Loosely Coupled System)	20
2) 강결합방식(Tightly Coupled System)	22
2. 랜드마크 기반 영상 항법	25
3. 영상 기반 항법알고리즘	26
1) Prediction	27
2) Measurement Update	30

3. Measurement Covariance Update Method	33
4. 시뮬레이션	33
1) UUF기반 영상항법시스템 성능 검증 및 비교 시뮬레이션	33
2) Measurement Covariance Update Method 검증 시뮬레이션	42
V. 결론	47
【참고문헌】	48

ABSTRACT

Vision Based Inertial Navigation System

Kim Jong Myeong

Advisor : Prof. Lee Hyun Jae, Ph.D.

Department of Aerospace Engineering,

Graduate School of Chosun University

In general, GPS/INS integrated navigation systems is used to estimate exact position and velocity. But, For preparation situation that GPS get jammings and interruptions, In this paper use vision sensor.

This paper proposes a new two methods for INS/Vision navigation system to improve performance. First, Unless secure a clear front view and illumination, availability of vision sensor lower for navigation system. For solve this problem, vision sensors is installed at 0, 120 and -120 degree to overcome situation without forward view and illumination. With this, observability is higher than existing system depended on only front view. Second, Unless measure more than three landmarks, error of landmark based vision navigation system increase gradually. Second, Unless measure more than three landmarks, error of landmark based vision navigation system increase gradually. For solve this problem, in such a measured landmark less than three, augment measurement covariance for reduce weighting of vision sensor. Finally, the performance of the suggested approaches are verified by numerical simulations.

항법시스템의 중요한 연구 분야 중 하나가 필터 기술이다. 가장 일반적인 통합 항법시스템인 GPS/INS의 필터 기술은 강도에 따른 필터 설계^[7], 초기 오차가 큰 상황을 고려한 모델링 기술, 새로운 필터 개발 및 개조 등이 여러 대학, 연구소, 산업체에서 연구하고 있다.^[8-9] 이때 INS를 활용한 항법방정식은 비선형 함수이므로 확장칼만필터(Extended Kalman Filter)를 주로 사용하고 있다. 그러나 확장칼만필터의 경우 선형화를 기반으로 알고리즘이 구성되어 있기 때문에 초기 오차가 크거나, 측정값에 큰 오차가 발생하면 필터가 발산하는 문제가 생긴다. 특히 이런 문제는 저가형 센서를 사용하는 경우 더욱 빈번하게 발생한다. 이러한 문제를 해결하기 위한 여러 방법이 있고 본 연구에서는 UKF(Unscented Kalman Filter)를 기반으로 이러한 문제점을 해결하고자 한다.

UKF는 추정 값을 전파(Propagation)할 때 통계적 특성을 고려한 시그마 포인트(Sigma Point)를 선형화 없이 그대로 사용해 전파한다. 전파 후 구해진 추정 값을 사용해 오차 공분산을 계산하기 때문에 기존의 확장칼만필터에서 발생하는 선형화 오차가 발생하지 않는다. 이러한 효과로 인해 확장칼만필터를 대신해 많은 분야에서 사용하고 있다. 그러나 UKF는 기존의 확장칼만필터와 비교하면 계산량이 많다. 특히, 실시간성을 반드시 확보해야하는 항법시스템에서 UKF를 사용하기가 쉽지 않다.^[10]

이러한 문제점을 해결하기 위해 UKF의 특성을 최대한 유지하면서 계산 시간을 줄인 방법인 UUF를 사용하여 시스템을 구성하였다. UUF를 활용하면 기존의 알고리즘보다 적은 시그마 포인트를 필요로 한다. 따라서 기존의 UKF보다 계산량이 줄어들게 된다. 물론, 이렇게 계산량을 줄여도 기존에 계산량을 줄이기 위해 제시되어있는 방법에 비하면 계산량이 많다. 그러나 UUF는 기존의 UKF보다는 계산량은 줄이고 계산량을 줄인 UKF보다 오차는 작은 알고리즘으로 기존의 UKF의 특성을 최대한 활용한 알고리즘이다.^[11]

본 논문에서는 INS에 영상센서를 더하고 앞서 언급한 UUF를 사용하여 UUF기반

의 INS/Vision 항법시스템을 설계하고자 한다. 성능을 검증하기 위해 랜드마크의 개수는 같고 단일 영상센서를 사용하는 시스템과의 수치시물레이션을 통한 비교를 진행했다. 또한, 앞서 언급한 랜드마크의 개수가 2개 이하가 측정되는 경우 오차 공분산의 값을 증가시켜 영상센서의 가중치에 변화를 주는 방식을 검증하는 시물레이션을 진행했다.

본 논문의 구성은 다음과 같다. 우선, 2장에서 항법시스템의 기초인 좌표계와 운동방정식에 관해 논하고 3장에서는 본 연구에서 사용한 3가지 종류의 Unscented 필터에 대해 논한다. 4장에서는 기존의 GPS/INS 통합항법시스템, UUF 기반의 항법시스템, Measurement Covariance Update Method 그리고 이를 검증하기 위한 시물레이션으로 구성했다. 그리고 마지막으로 5장에서 결론을 마무리한다.

II. 항법 시스템

1. 항법시스템의 기초

1) 좌표계의 정의

먼저 좌표계에 대한 정의가 Figure.1에 나타나 있으며 항법시스템에서 사용하는 대표적인 좌표계는 아래의 (1)부터 (4)까지와 같다. (5)는 다른 좌표계로의 변환에 대해 설명하였다.^[1]

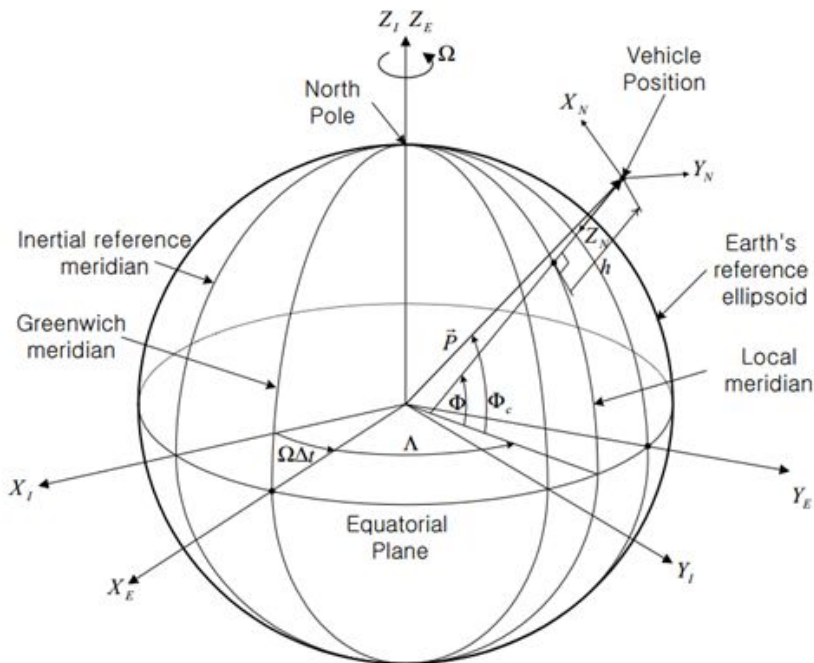


Fig. 1 좌표계 시스템

(1) 관성좌표계(Earth Centered Inertial Frame, ECI)

관성좌표계(X_I, Y_I, Z_I)는 기본적으로 지구 중심을 영점으로 하며 Vernal equinox를 X 축으로 정의한다. 또한, 북극, 즉 지구의 회전축을 Z 축 그리고 이축들과 직교하는 축을 Y 축으로 정의하는 좌표계이다. 여기서 X, Y 축으로 이뤄진

평면이 적도면이 된다.

(2) 지구중심좌표계(Earth Centered Earth Fixed Frame, ECEF)

지구중심좌표계(X_E, Y_E, Z_E)의 Z 축은 지구의 회전축으로 정의되고, X 축은 Greenwich Meridian을 기준으로 한다. ECI좌표계의 Z 축이 그린이치 천문대를 지나 적도면과 만나는 지점을 X 축으로 정의한다. 그리고 이들 2개의 축들과 직교하며 X 축을 기준으로 90도 만큼 회전한 방향을 Y 축으로 정의하는 시스템이다. 이 축은 지구에 고정돼 있어 지구중심좌표계라 한다.

(3) 항법좌표계(Navigation Frame, Geographic Frame)

항법좌표계(X_N, Y_N, Z_N)는 좌표 원점이 관성항법장치의 중심에 있다. 일반적으로 관측자 중심이라 할 수 있어서 지역좌표계(Local Frame)이라고도 불린다. X - Y 축과 이뤄진 면이 수평면으로 정의될 수 있으며, Z 축은 여기에서 직교하는 방향으로 나타낸다. 일반적으로, X 축은 북쪽을 가리키며, Y 축은 동쪽, 그리고 Z 축은 지구의 아래쪽 방향을 향한다.

(4) 동체좌표계(Body Frame)

동체좌표계는 실제 관찰자 또는 비행기의 중심에 고정되어 있는 좌표계이다. 일반적으로 동체좌표계의 X 축은 비행체의 진행방향(Roll)을 가리키고, Y 축은 날개방향(Pitch) 그리고 Z 축은 이들 둘과 직교하는 아랫방향(Yaw)을 가리킨다.

2) 좌표계 변환(Coordinate Transformation)

좌표계의 변환을 위해 좌표변환행렬인 방향코사인행렬(Direction Cosine Matrix, DCM)을 정의해야 한다. 지구중심좌표계에서 항법좌표계로의 변환은 다음과 같다.^[1,3,13]

$$C_E^N = \begin{bmatrix} -\sin\Phi\cos\Lambda & -\sin\Phi\sin\Lambda & \cos\Phi \\ -\sin\Lambda & \cos\Lambda & 0 \\ -\cos\Phi\cos\Lambda & -\cos\Phi\sin\Lambda & -\sin\Phi \end{bmatrix} \quad (1)$$

또한, 동체좌표계에서 항법좌표계로의 DCM은 오일러각 3-2-1회전방식으로 다

음과 같이 표현할 수 있다.

$$C_B^N = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\cos\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (2)$$

ECEF 좌표계상에서 위치(P^e)는 항법좌표계에서의 위치 위도(Latitude), 경도(Longitude), 고도(Height)로부터 얻어진다.

$$\begin{aligned} x_e &= (R_\lambda + h) \cos\Phi \cos\Lambda \\ y_e &= (R_\lambda + h) \cos\Phi \sin\Lambda \\ z_e &= [(1 - \epsilon^2) R_\lambda + h] \sin\Phi \end{aligned} \quad (3)$$

여기에서 $R_\lambda = R_e / (1 - \epsilon^2 \sin^2\Phi)^{1/2}$ 을 나타낸다. 역으로 위도, 경도, 고도는 ECEF 좌표계상의 3차원 위치 벡터로부터 다음과 같이 얻을 수 있다.

$$\begin{aligned} \tan\Lambda &= \left(\frac{y_e}{x_e} \right) \\ \tan\Phi &= \left(\frac{(R_\lambda + h) \sin\Phi}{\sqrt{x_e^2 + y_e^2}} \right) = \frac{z_e}{\sqrt{x_e^2 + y_e^2}} \left(1 + \frac{\epsilon^2 R_\lambda \sin\Phi}{z_e} \right) \\ h &= \frac{\sqrt{x_e^2 + y_e^2}}{\cos\Phi} - R_\lambda \end{aligned} \quad (4)$$

다음의 Table. 1에 나온 내용인 WGS-84에 관련된 기본적인 파라미터를 나타내었다. 여기에서 경도는 하나의 닫힌해로 나타낼 수 있지만, 위도는 닫힌 해 형태로 구성되어 있지 않다. 따라서 수치적인 연산을 통해서 획득하여야 한다.

1. 고도가 $h = 0$ 인 경우 $Z_e|_{h=0} = R_\lambda (1 - \epsilon^2) \sin\Phi$ 이다. 이 식을 사용하면 초기값은 다음과 같이 설정할 수 있다.

$$\tan\Phi_1 = \frac{z_e}{(1 - \epsilon^2) \sqrt{x_e^2 + y_e^2}}$$

2. $R_{\lambda_N} = R_e / (1 - \epsilon^2 \sin^2\Phi)^{1/2}$ 을 적용하면

$$\tan \Phi_{n+1} = \frac{z_e}{\sqrt{x_e^2 + y_e^2}} \left(1 + \frac{\partial^2 R_n \sin \Phi_n}{z_2} \right)$$

3. $|\Phi_{n+1} - \Phi_n| < \epsilon_\phi$ 을 만족할 때 까지 (2)를 지속적으로 수행한다. 여기에서 ϵ_ϕ 는 구하고자 하는 오차값에 해당한다.

Semimajor axis	$a(= R_e)$	6378137.000 m
Semiminor axis	b	6356752.3142 m
Flattening	f	1/298.257223563
Major eccentricity	ϵ	0.0818191908426
Axis ratio	b/a	0.996647189335
Second degree zonal Harmonic coeff. of the geopotential	$\bar{C}_{2,0}$	$-0.48416685 \times 10^{-3}$
Fourth degree zonal Harmonic coeff. of the geopotential	$\bar{C}_{4,0}$	0.53699×10^{-6}
Angular velocity of the earth	Ω	$7.292115 \times 10^{-5} (rad/sec)$
The earth' s gravitational const.	μ	$7.292115 \times 10^{-5} (rad/sec)$
Mass of earth	M	$3986005 \times 10^{-8} (m^3/sec^2)$
Theoretical gravity at the equator	γ_e	$9.7803267714 (m/sec^2)$
Theoretical gravity at the poles	γ_p	$9.8321863635 (m/sec^2)$
Mean value of theoretical gravity	$\bar{\gamma}$	$9.797644661 (m/sec^2)$

Table. 1 WGS-84 Parameter

3) 운동 방정식

(1) Gravitational Force

중력모델은 다음과 같은 르장드르(Legendre) 다항식으로 묘사가 가능하다. 일반적으로 2차항까지 고려하며 다음과 같이 나타낼 수 있다.^[1,13]

$$\begin{aligned} g_x &= -\frac{\mu}{R^2} \left[1 - \frac{3}{2} J_2 \left(\frac{R_e}{R} \right)^2 \left(5 \frac{z_e^2}{R^2} - 1 \right) \right] \frac{x_e}{R} \\ g_y &= -\frac{\mu}{R^2} \left[1 - \frac{3}{2} J_2 \left(\frac{R_e}{R} \right)^2 \left(5 \frac{z_e^2}{R^2} - 1 \right) \right] \frac{y_e}{R} \\ g_z &= -\frac{\mu}{R^2} \left[1 - \frac{3}{2} J_2 \left(\frac{R_e}{R} \right)^2 \left(5 \frac{z_e^2}{R^2} - 1 \right) \right] \frac{z_e}{R} \end{aligned} \quad (5)$$

여기서 R 은 지구중심에서 비행체까지의 거리를 나타내며 아래의 (6)과 같이 표현할 수 있다.

$$R = [x_e^2 + y_e^2 + z_e^2]^{1/2} \quad (6)$$

J_2 는 지구 포텐셜 함수의 2차 Jonal harmonics 성분은 다음과 같다.

$$J_2 = -\sqrt{5} \bar{C}_{20} = 1.082629989051944 \times 10^{-3} \quad (7)$$

(2) 병진운동(Translational Motion)

병진 운동 비행체의 속도 방정식은 다음과 같이 주어진다.

$$\left. \frac{d\mathbf{P}}{dt} \right|_I = \left. \frac{d\mathbf{P}}{dt} \right|_E + \boldsymbol{\omega}_e \times \mathbf{P} \quad (8)$$

속도 성분은 다음과 같이 주어지며

$$\mathbf{V} = \left. \frac{d\mathbf{P}}{dt} \right|_E \quad (9)$$

속도 방정식인 (8)은 다음과 같이 나타낼 수 있다.

$$\left. \frac{d\mathbf{P}}{dt} \right|_I = \mathbf{V} + \boldsymbol{\omega}_e \times \mathbf{P} \quad (10)$$

위 식을 미분함에 따라 가속도 운동방정식을 아래와 같이 얻을 수 있다.

$$\begin{aligned}\frac{d^2 \mathbf{P}}{dt^2} \Big|_I &= \frac{d\mathbf{V}}{dt} \Big|_E + \boldsymbol{\omega}_{ie} \times \mathbf{V} + \frac{d\boldsymbol{\omega}_{ie}}{dt} \Big|_E \times \mathbf{P} + \boldsymbol{\omega}_{ie} \times \frac{d\mathbf{P}}{dt} \Big|_E + \boldsymbol{\omega}_{ie} \times \boldsymbol{\omega}_{ie} \times \mathbf{P} \\ &= \frac{d\mathbf{V}}{dt} \Big|_E + 2\boldsymbol{\omega}_{ie} \times \mathbf{V} + \boldsymbol{\omega}_{ie} \times \boldsymbol{\omega}_{ie} \times \mathbf{P}\end{aligned}\quad (11)$$

다음과 같은 관계식을 이용하면

$$\frac{d^2 \mathbf{P}}{dt^2} \Big|_I = \mathbf{F} + \mathbf{G} \quad (12)$$

가속도 운동방정식은 다음과 같이 나타낼 수 있다.

$$\frac{d\mathbf{V}}{dt} \Big|_E = \mathbf{F} - 2\boldsymbol{\omega}_{ie} \times \mathbf{V} - \boldsymbol{\omega}_{ie} \times \boldsymbol{\omega}_{ie} \times \mathbf{P} + \mathbf{G} \quad (13)$$

위의 식을 ECEF좌표계로 나타내면

$$\begin{aligned}\dot{\mathbf{P}} &= \mathbf{V}^e \\ \dot{\mathbf{V}}^e &= \mathbf{C}_B^E \mathbf{F}^b - 2\boldsymbol{\Omega} \mathbf{V}^e - \boldsymbol{\Omega}^2 \mathbf{P} + \mathbf{G}\end{aligned}\quad (14)$$

여기에서 각속도에 대한 교대행렬(skew symmetric matrix)이 다음과 같이 정의된다.

$$\boldsymbol{\Omega} = [\boldsymbol{\omega}_{ie}^e]^\times \quad (15)$$

그리고 $\mathbf{F}^b$ 는 동체좌표계에서 측정되는 가속도계 출력 값을 의미하며, 코리올리와 원심력에 의한 방정식은 다음과 같이 나타낼 수 있다.

$$2\boldsymbol{\Omega} \mathbf{V}^e + \boldsymbol{\Omega}^2 \mathbf{P} = \begin{bmatrix} -(2v_y + \Omega x_e)\Omega \\ (2v_x + \Omega y_e)\Omega \\ 0 \end{bmatrix}_E \quad (16)$$

여기에서 지구회전 각속도는 다음과 같이 정의된다.

$$\boldsymbol{\omega}_{ie}^e = \begin{bmatrix} 0 \\ 0 \\ \Omega \end{bmatrix}_E \quad (17)$$

(3) 각운동 방정식

각속도를 측정하는 자이로는 관성좌표계를 기준으로 비행체의 각속도를 측정한다. ω_{ib} 를 관성좌표계상에서 비행체의 각속도라고 정의하면 다음과 같은 관계식을 갖는다.^[1,13]

$$\omega_{eb} = \omega_{tb} - \omega_{te} \quad (18)$$

즉, ECEF에서의 순수한 비행체의 각속도는 관성좌표계상에서의 동체좌표계의 각속도에서 지구의 자전속도를 고려하여 획득 할 수 있다. 식 (18)을 동체좌표계에서 다시 정리하면 다음과 같다.

$$\omega_{eb}^b = \omega_{tb}^b - C_E^B \omega_{te}^e \quad (19)$$

여기에서 ω_{ib}^b 는 비행체의 동체축에 고정된 자이로의 출력으로서 동체좌표계에서 표현된 식이다. 따라서 다음과 같이 정의할 수 있다.

$$\omega_{eb}^b = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_B \quad \omega_{tb}^b = \begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix}_B \quad (20)$$

다음으로 자세를 표현하는 방법 중 하나인 쿼터니언을 사용하여 동체좌표계에서 ECEF좌표계로의 방향코사인행렬은 다음과 같이 나타낼 수 있다.

$$C_B^E = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_3q_0) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (21)$$

여기서 쿼터니언은 아래의 식 (22)를 사용하여 구할 수 있다.

$$\dot{q}^e = \frac{1}{2} \Omega_{eb}^b q^e \quad (22)$$

여기에서 q^e 와 Ω_{eb}^b 는 아래와 같다.

$$\mathbf{q}^e = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad \Omega_{eb}^e = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \quad (23)$$

그리고 쿼터니언은 다음과 같은 구속조건을 만족해야만 한다.

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (24)$$

쿼터니언은 오일러각을 활용한 계산이 가지고 있는 문제점인 특이점이 없어 자세를 계산하기가 용이하다. 하지만 일반적으로 우리가 자세를 표현하는데 있어서 쿼터니언보단 오일러각이 더 편리하다. 따라서 계산과정이 아닌 초기 값 및 결과 값과 같이 보여지는 것에 대해서는 오일러각으로 표현한다.

쿼터니언과 오일러각은 간단한 계산 과정을 통해 서로 변환할 수 있다. 우선, 쿼터니언에서 오일러각으로의 변환은 아래와 같이 이뤄진다.

$$\begin{aligned} \tan \phi &= C_{32} / C_{33} \\ \sin \theta &= -C_{31} \\ \tan \psi &= C_{21} / C_{11} \end{aligned} \quad (25)$$

여기서, C_{ij} 는 식 (2)에서 구한 DCM의 i 번째 행, j 번째 열의 요소이다. 또한, DCM은 다음과 같은 관계식을 가진다.

$$\mathbf{C}_B^N = \mathbf{C}_E^N \mathbf{C}_B^E \quad (26)$$

이번에는 오일러에서 쿼터니언으로의 변환은 식 (21)에서의 DCM을 활용하여 다음과 같이 구할 수 있다.

$$\begin{aligned} q_0 &= \frac{1}{2} \sqrt{1 + C_{11} + C_{22} + C_{33}} \\ q_1 &= \frac{C_{32} - C_{23}}{4q_0} \\ q_2 &= \frac{C_{13} - C_{31}}{4q_0} \\ q_3 &= \frac{C_{21} - C_{12}}{4q_0} \end{aligned} \quad (27)$$

III. 비선형 필터

이번 장에서는 비선형필터인 Unscented 칼만 필터에 대해 소개한다. 칼만 필터는 본래 선형을 위한 이론이다. 이를 비선형에 적용하기 위해 주어진 비선형 모델을 선형화하여 사용하는 확장 칼만 필터(Extended Kalman Filter)를 주로 이용한다. 하지만 선형화는 확률적 특성을 왜곡시켜 불안정한 요소를 발생시킨다. 이러한 단점을 극복하기 위해 Unscented 변환을 이용한 Unscented 칼만필터를 사용한다. Unscented 칼만필터는 실제 비선형 시스템을 그대로 사용하여 전파되고 오차 공분산으로부터 선택된 최소 시그마 포인트를 이용하기 때문에 기존의 확장 칼만필터보다 정확하고 안정성이 뛰어나다.^[12]

1. Unscented Filtering

1) Unscented 칼만필터

우선 다음과 같은 이산 비선형 시스템을 고려하자.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, k, \mathbf{w}_k) \quad (28)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) \quad (29)$$

x_k 는 $n \times 1$ 의 상태변수이고 y_k 는 $m \times 1$ 의 관측벡터이다. 프로세스 노이즈인 w_k 와 관측오차 노이즈인 v_k 는 White Gaussian Noise이다. 또한, 공분산행렬 P_k , Q_k , R_k 를 활용해서 $2(n+m+p)+1$ 개의 시그마 포인트를 구할 수 있다.

$$\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k \quad (30)$$

$$\mathbf{P}_a = \begin{bmatrix} P_k & 0 & 0 \\ 0 & Q_k & 0 \\ 0 & 0 & R_k \end{bmatrix} \quad (31)$$

$$\boldsymbol{\chi}_k(i) = \hat{\mathbf{x}}_k + \left(\sqrt{(\mathbf{I} + \lambda) \mathbf{P}_a} \right)_i \quad \text{for } i=1, \dots, L \quad (32)$$

$$\boldsymbol{\chi}_k(i) = \hat{\mathbf{x}}_k - \left(\sqrt{(\mathbf{I} + \lambda) \mathbf{P}_a} \right)_i \quad \text{for } i=L+1, \dots, 2L \quad (33)$$

이렇게 얻은 시그마 포인트들은 다음과 같은 비중(W_i)을 가진다.

$$W_0 = \frac{\lambda}{(L+\lambda)} \quad (34)$$

$$W_i = \frac{1}{\{2(L+\lambda)\}} \quad i=1, \dots, 2L \quad (35)$$

여기서 $L = n + m + p$ 이고 λ 는 스케일 파라미터이다. $\sqrt{P_a}$ 는 Cholesky 분해법을 사용하여 계산한다. 또한, 위의 (30) ~ (33)을 정리하면 다음과 같다.

$$\chi_k = [\hat{x}_k \quad \hat{x}_k + \sqrt{(L+\lambda)(P_a)} \quad \hat{x}_k - \gamma\sqrt{(L+\lambda)(P_a)}] \quad (36)$$

주어진 시그마 포인트를 활용하여 아래와 같이 전파(Propagation)을 할 수 있다.

$$\begin{aligned} \chi_{k+1}(i) &= f(\chi_k(i), k, w_k) \\ \bar{x}_{k+1}(i) &= \sum_{i=0}^{2n} W_i \chi_{k+1}(i) \\ P_{k+1} &= \sum_{i=0}^{2L} W_i \{ \chi_{k+1}(i) - \bar{x}_{k+1}(i) \} \{ \chi_{k+1}(i) - \bar{x}_{k+1}(i) \}^T \\ Y_{k+1}(i) &= h(\chi_{k+1}(i), k+1, v_k) \\ \hat{y}_{k+1} &= \sum_{i=0}^{2L} W_i Y_{k+1}(i) \end{aligned} \quad (37)$$

또한, 주어진 측정값을 활용하여 측정치 업데이트를 진행하여 구하고자 하는 상태방정식의 추정 값($\hat{x}_{k+1}^+$)과 공분산(P_{k+1}^+)을 구할 수 있다.^[1, 13, 14]

$$\begin{aligned} P_{k+1}^{yy} &= \sum_{i=0}^{2L} W_i \{ Y_{k+1}(i) - \hat{y}_{k+1} \} \{ Y_{k+1}(i) - \hat{y}_{k+1} \} \\ P_{k+1}^{sy} &= \sum_{i=0}^{2L} W_i \{ \chi_{k+1}(i) - \bar{x}_{k+1} \} \{ Y_{k+1}(i) - \hat{y}_{k+1} \}^T \\ K_{k+1} &= P_{k+1}^{sy} (P_{k+1}^{yy})^{-1} \\ \hat{x}_{k+1}^+ &= \hat{x}_{k+1} + K_{k+1} (y_{k+1} - \hat{y}_{k+1}) \\ P_{k+1}^+ &= P_{k+1} - K_{k+1} P_{k+1}^{sy} K_{k+1}^T \end{aligned} \quad (38)$$

Table 2는 UKF를 요약한 과정이다.

Model	$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{k}, \mathbf{w}_k), \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t))$ $\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{k}, \mathbf{v}_k), \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$
Initialize	$\hat{\mathbf{x}}_0 = E\{\hat{\mathbf{x}}_0\}$ $\mathbf{P}_0 = \mathbf{P}_k$ Calculate Sigma points : $\mathbf{P}_a = \begin{bmatrix} P_k & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{Q}_k & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{R}_k \end{bmatrix}$ $\boldsymbol{\chi}_k(i) = \hat{\mathbf{x}}_k + (\gamma \sqrt{\mathbf{P}_a})_i \quad \text{for } i=1, \dots, L$ $\boldsymbol{\chi}_k(i) = \hat{\mathbf{x}}_k - (\gamma \sqrt{\mathbf{P}_a})_i \quad \text{for } i=L+1, \dots, 2L$
Propagation	$\boldsymbol{\chi}_{k+1}(i) = \mathbf{f}(\boldsymbol{\chi}_k(i), \mathbf{k})$ $\bar{\mathbf{x}}_{k+1}(i) = \sum_{i=0}^{2L} \mathbf{W}_i \boldsymbol{\chi}_{k+1}(i)$ $\mathbf{P}_{k+1} = \sum_{i=0}^{2L} \mathbf{W}_i \{ \boldsymbol{\chi}_{k+1}(i) - \bar{\mathbf{x}}_{k+1}(i) \} \{ \boldsymbol{\chi}_{k+1}(i) - \bar{\mathbf{x}}_{k+1}(i) \}^T$
Measurement Update	$\mathbf{Y}_{k+1}(i) = \mathbf{h}(\boldsymbol{\chi}_{k+1}(i), \mathbf{k}+1)$ $\hat{\mathbf{y}}_{k+1}^- = \sum_{i=0}^{2n} \mathbf{W}_i \mathbf{Y}_{k+1}(i)$ $\mathbf{P}_{k+1}^{yy} = \sum_{i=0}^{2L} \mathbf{W}_i \{ \mathbf{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-(i) \} \{ \mathbf{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-(i) \}^T$ $\mathbf{P}_{k+1}^{sy} = \sum_{i=0}^{2L} \mathbf{W}_i \{ \boldsymbol{\chi}_{k+1}(i) - \bar{\mathbf{x}}_{k+1}(i) \} \{ \mathbf{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-(i) \}^T$ $\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{sy} (\mathbf{P}_{k+1}^{yy})^{-1}$ $\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)$ $\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1} \mathbf{P}_{k+1}^{yy} \mathbf{K}_{k+1}^T$

Table. 2 Unscented Kalman Filtering

2) UKF for Additive Noise case

앞서 설명한 1)은 노이즈가 시스템 모델에 들어가 있는 형태인 경우에 대한 설

명이었다. 이번에는 노이즈가 모델에 더해진 형태로 다음과 같이 표현할 수 있다.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, k) + \mathbf{w}_k \quad (39)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, k) + \mathbf{v}_k \quad (40)$$

주어진 모델을 활용하여 시그마 포인트를 구하면 다음과 같다.

$$\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k \quad (41)$$

$$\boldsymbol{\chi}_k(i) = \hat{\mathbf{x}}_k + \left(\sqrt{(n+\lambda)(\mathbf{P}_k)} \right)_i \quad \text{for } i=1, \dots, n \quad (42)$$

$$\boldsymbol{\chi}_k(i) = \hat{\mathbf{x}}_k - \left(\sqrt{(n+\lambda)(\mathbf{P}_k)} \right)_i \quad \text{for } i=n+1, \dots, 2n \quad (43)$$

n 은 상태변수의 크기이고 λ 는 스케일 파라미터이다. 또한, 시그마 포인트들은 다음과 같은 비중(W_i)을 가진다.

$$W_0 = \frac{\lambda}{(n+\lambda)} \quad (44)$$

$$W_i = \frac{1}{\{2(n+\lambda)\}} \quad i=n+1, \dots, 2n \quad (45)$$

전과 과정과 측정치 업데이트 과정은 i 의 값만 제외하고 모두 동일한 과정이므로 아래 Table 3에 요약하는 것으로 마무리한다.^[1, 12-14]

Model	$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + \mathbf{G}\mathbf{w}_k, \mathbf{w}(t) \sim N(\mathbf{0}, \mathbf{Q}(t))$ $\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, k) + \mathbf{v}_k, \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$
Initialize	$\hat{\mathbf{x}}_0 = E\{\hat{\mathbf{x}}_0\}$ $\mathbf{P}_0 = \mathbf{P}_k$ Calculate Sigma points : $\boldsymbol{\chi}_k(i) = \hat{\mathbf{x}}_k + \left(\gamma \sqrt{\mathbf{P}_k} \right)_i \quad \text{for } i=1, \dots, L$ $\boldsymbol{\chi}_k(i) = \hat{\mathbf{x}}_k - \left(\gamma \sqrt{\mathbf{P}_k} \right)_i \quad \text{for } i=L+1, \dots, 2L$

Propagation	$\mathbf{x}_{k+1}(i) = \mathbf{f}(\mathbf{x}_k(i), k)$ $\bar{\mathbf{x}}_{k+1}(i) = \sum_{i=0}^{2n} W_i \mathbf{x}_{k+1}(i)$ $\mathbf{P}_{k+1}^- = \sum_{i=0}^{2n} W_i \{ \mathbf{x}_{k+1}(i) - \bar{\mathbf{x}}_{k+1}(i) \} \{ \mathbf{x}_{k+1}(i) - \bar{\mathbf{x}}_{k+1}(i) \}^T + \bar{\mathbf{Q}}_k$
Measurement Update	$\mathbf{Y}_{k+1}(i) = \mathbf{h}(\mathbf{x}_{k+1}(i), k+1)$ $\hat{\mathbf{y}}_{k+1}^- = \sum_{i=0}^{2n} W_i \mathbf{Y}_{k+1}(i)$ $\mathbf{P}_{k+1}^{yy} = \sum_{i=0}^{2n} W_i \{ \mathbf{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-(i) \} \{ \mathbf{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-(i) \}^T$ $\mathbf{P}_{k+1}^{vv} = \mathbf{P}_{k+1}^{yy} + \mathbf{R}_{k+1}$ $\mathbf{P}_{k+1}^{sy} = \sum_{i=0}^{2n} W_i \{ \mathbf{x}_{k+1}(i) - \bar{\mathbf{x}}_{k+1}(i) \} \{ \mathbf{Y}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^-(i) \}^T$ $\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{sy} (\mathbf{P}_{k+1}^{vv})^{-1}$ $\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)$ $\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1} \mathbf{P}_{k+1}^{vv} \mathbf{K}_{k+1}^T$

Table. 3 Unscented Kalman Filtering for Additive Noise case

3) Uncorrelated Unscented Filtering(UUF)

UUF는 식 (28), (29)과 같이 비선형 함수 안에 노이즈가 포함된 경우에 대해 다룬다. 우선, 상태 방정식과 측정 방정식의 확률변수 사이에 상관관계가 없다고 가정한다. 이를 활용하면 (31)을 분해시킬 수 있다.

$$\mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_k & \mathbf{0}_{n \times m} \\ \mathbf{0}_{m \times n} & \mathbf{Q}_k \end{bmatrix} \quad (46)$$

$$\mathbf{M}_k^a = \begin{bmatrix} \mathbf{P}_k & \mathbf{0}_{n \times p} \\ \mathbf{0}_{p \times n} & \mathbf{R}_k \end{bmatrix} \quad (47)$$

여기에서, $\mathbf{0}_{n \times m}$ 은 크기가 $n \times m$ 인 영행렬이고, $\mathbf{Q}_k$ 는 프로세스 오차 공분산, $\mathbf{R}_k$ 는 측정값 오차 공분산이다. n, m, p 은 오차 공분산의 크기이다.

(46), (47)과 같이 분리시키면 식 (31)을 사용할 때 보다 전체 시그마 포인트의 개수가 감소한다. 기존의 UKF는 $2(n+m+p)+1$ 개의 시그마 포인트를 필요로

하지만 UUF를 활용하면 기존의 알고리즘보다 적은 시그마 포인트를 필요로 한다. 물론, 이렇게 계산량을 줄여도 기존에 계산량을 줄이기 위해 제시되어있는 방법에 비하면 계산량이 많다. 그러나 UUF는 기존의 UKF보다는 계산량은 줄이고 계산량을 줄인 UKF보다 오차는 작은 알고리즘으로 기존의 UKF의 특성을 최대한 활용한 알고리즘이다.

UUF는 (46)과 (47)를 활용해 시그마 포인트를 두 번에 걸쳐 계산한다는 것을 제외하고 전반적으로 기존의 UKF와 유사하다. 우선, 시그마 포인트를 계산하고 주어진 모델에 대입해 나온 값을 활용해 $\hat{x}_k^-$ (predicted mean) 과 오차 공분산을 계산한다.

식 (46)을 사용하여 시그마 포인트를 계산하면

$$\chi_k^c(0) = [\hat{\mathbf{x}}_k \quad \mathbf{0}_{1 \times n}]^T \quad (48)$$

$$\chi_k^c(i) = [\hat{\mathbf{x}}_k \quad \mathbf{0}_{1 \times n}]^T + \left(\gamma_p \sqrt{\mathbf{P}_k^c} \right)_i \quad \text{for } i=1, \dots, (n+m) \quad (49)$$

$$\chi_k^c(i) = [\hat{\mathbf{x}}_k \quad \mathbf{0}_{1 \times n}]^T - \left(\gamma_p \sqrt{\mathbf{P}_k^c} \right)_i \quad \text{for } i=(n+m), \dots, 2(n+m) \quad (50)$$

여기서 $\gamma_p = \sqrt{n+m+\lambda}$ 이고 λ 는 비중상수로 사용자가 설정하는 값이다. 구해진 시그마 포인트를 식 (28)에 대입하여 아래와 같이 계산하고

$$\chi_{k+1,i}^c = f(\chi_k^c, k) \quad (51)$$

(51)을 활용해 $\hat{x}_{k+1}^-$ (predicted mean) 과 오차 공분산을 아래와 같이 구한다.

$$\hat{\mathbf{x}}_{k+1}^- = \sum_{i=(n+m)}^{(n+m)} W_i^c \chi_{k+1,i}^c \quad (52)$$

$$\mathbf{P}_{k+1}^c = \sum_{i=(n+m)}^{(n+m)} W_i^c (\chi_{k+1,i}^c - \hat{\mathbf{x}}_{k+1}^-)(\chi_{k+1,i}^c - \hat{\mathbf{x}}_{k+1}^-)^T \quad (53)$$

여기서 W_i^c 은 비중(weighting) 아래와 같이 정의한다.

$$W_0^c = \frac{\kappa}{(n+m+\kappa)} \quad (54)$$

$$W_i^c = W_{-i}^c = \frac{1}{2(n+m+\kappa)} \quad (55)$$

κ 는 비중상수로 사용자가 설정한다.

다음으로 식 (47)를 사용해 시그마 포인트를 다시 설정한다.

$$\chi_k^m(0) = [\hat{\mathbf{x}}_{k+1}^- \quad \mathbf{0}_{n \times m}]^T \quad (56)$$

$$\chi_k^m(i) = [\hat{\mathbf{x}}_{k+1}^- \quad \mathbf{0}_{n \times m}]^T + \left(\gamma_m \sqrt{\mathbf{M}_k^r} \right)_i \quad \text{for } i=1, \dots, (n+p) \quad (57)$$

$$\chi_k^m(i) = [\hat{\mathbf{x}}_{k+1}^- \quad \mathbf{0}_{n \times m}]^T - \left(\gamma_m \sqrt{\mathbf{M}_k^r} \right)_i \quad \text{for } i=(n+p), \dots, 2(n+p) \quad (58)$$

여기서 $\gamma_m = \sqrt{n+p+\lambda}$ 이고 λ 는 비중상수로 사용자가 설정하는 값이다. 구해진 시그마 포인트를 식 (29)에 대입하여 아래와 같이 계산하고

$$\chi_{k+1,i}^m = h(\chi_k^m) \quad (59)$$

(59)을 활용해 $\hat{\mathbf{y}}_{k+1}^-$ (predicted measurement mean)과 P_{k+1}^{yy} (measurement covariance)와 P_{k+1}^{xy} (cross-correlation matrix)를 구한다.

$$\hat{\mathbf{y}}_{k+1}^- = \sum_{i=(n+p)}^{(n+p)} W_i^m \chi_{k+1,i}^m \quad (60)$$

$$P_{k+1}^{yy} = \sum_{i=(n+p)}^{(n+p)} W_i^m (\chi_{k+1,i}^m - \hat{\mathbf{y}}_{k+1}^-)(\chi_{k+1,i}^m - \hat{\mathbf{y}}_{k+1}^-)^T \quad (61)$$

$$P_{k+1}^{xy} = \sum_{i=(n+p)}^{(n+p)} W_i^m (\chi_{k+1,i}^c - \hat{\mathbf{x}}_{k+1}^-)(\chi_{k+1,i}^m - \hat{\mathbf{y}}_{k+1}^-)^T \quad (62)$$

이를 활용해서 식 (38)과 같이 칼만게인, 추정값 및 공분산을 구한다. 아래의 Table 4는 UUF를 정리한 과정이다.^[11]

Model	$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{k}, \mathbf{w}_k), \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t))$ $\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{k}, \mathbf{v}_k), \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$
Initialize	$\hat{\mathbf{x}}_0 = E\{\hat{\mathbf{x}}_0\}$ $\mathbf{P}_0 = \mathbf{P}_k$ Calculate Sigma points : $\mathbf{P}_k^i = \begin{bmatrix} \mathbf{P}_k & \mathbf{0}_{n \times m} \\ \mathbf{0}_{m \times n} & \mathbf{Q}_k \end{bmatrix}$

	$\chi_k(0) = \begin{bmatrix} \hat{\mathbf{x}}_k \\ \hat{\mathbf{n}}_k \end{bmatrix}$ $\chi_k(i) = \begin{bmatrix} \hat{\mathbf{x}}_k \\ \hat{\mathbf{n}}_k \end{bmatrix} + \left(\gamma_p \sqrt{\mathbf{P}_k^p} \right)_i \quad \text{for } i=1, \dots, (n+m)$ $\chi_k(i) = \begin{bmatrix} \hat{\mathbf{x}}_k \\ \hat{\mathbf{n}}_k \end{bmatrix} - \left(\gamma_p \sqrt{\mathbf{P}_k^p} \right)_i \quad \text{for } i=(n+m), \dots, 2(n+m)$
Propagation	$\chi_{k+1,i} = f(\chi_k, k)$ $\hat{\mathbf{x}}_{k+1}^- = \sum_{i=(n+m)}^{(n+m)} W_i^c \chi_{k+1,i}^c$ $\mathbf{P}_{k+1}^- = \sum_{i=(n+m)}^{(n+m)} W_i^c (\chi_{k+1,i}^c - \hat{\mathbf{x}}_{k+1}^-)(\chi_{k+1,i}^c - \hat{\mathbf{x}}_{k+1}^-)^T$
Measurement Update	Calculate Sigma points : $\mathbf{M}_k^a = \begin{bmatrix} \mathbf{P}_{k+1}^- & \mathbf{0}_{n \times p} \\ \mathbf{0}_{p \times n} & \mathbf{R}_k \end{bmatrix}$ $\chi_k^m(0) = \hat{\mathbf{x}}_{k+1}^-$ $\chi_k^m(i) = \hat{\mathbf{x}}_{k+1}^- + \left(\gamma_m \sqrt{\mathbf{M}_k^a} \right)_i \quad \text{for } i=1, \dots, (n+p)$ $\chi_k^m(i) = \hat{\mathbf{x}}_{k+1}^- - \left(\gamma_m \sqrt{\mathbf{M}_k^a} \right)_i \quad \text{for } i=(n+p), \dots, 2(n+p)$ $\chi_{k+1,i}^m = h(\chi_k^m)$ $\hat{\mathbf{y}}_{k+1}^- = \sum_{i=(n+p)}^{(n+p)} W_i^m \chi_{k+1,i}^m$ $\mathbf{P}_{k+1}^{sy} = \sum_{i=(n+p)}^{(n+p)} W_i^m (\chi_{k+1,i}^m - \hat{\mathbf{y}}_{k+1}^-)(\chi_{k+1,i}^m - \hat{\mathbf{y}}_{k+1}^-)^T$ $\mathbf{P}_{k+1}^y = \sum_{i=(n+p)}^{(n+p)} W_i^c (\chi_{k+1,i}^c - \hat{\mathbf{x}}_{k+1}^-)(\chi_{k+1,i}^m - \hat{\mathbf{y}}_{k+1}^-)^T$ $\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{sy} (\mathbf{P}_{k+1}^{sy})^{-1}$ $\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)$ $\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1} \mathbf{P}_{k+1}^{sy} \mathbf{K}_{k+1}^T$

Table. 4 Uncorrelated Unscented Filtering

IV. 영상기반 항법시스템

1. GPS/INS 항법 시스템

이번 절에서는 GPS/INS 통항항법 시스템을 소개한다. 일반적으로 약결합 방식(Loosely Coupled System)과 강결합 방식(Tightly Coupled System)이 존재한다. 이들을 각각 살펴보기로 한다.

우선, 두 방식을 설명하기 위해 앞서 INS의 수학적 모델에 관해 간략하게 살펴보고자 한다. INS는 자이로와 가속도계로 구성된 항법시스템으로 두 센서의 수학적 모델은 [13]에 제시되어 있다. 우선, 자이로의 수학적 모델은 다음과 같다.

$$\tilde{\omega}_{ib}^b = (I + K_g) \omega_{ib}^b + \eta_{gv} + b_g \quad (63)$$

$$\dot{b}_g = \eta_{gu}, \quad \dot{s}_g = \eta_{gs} \quad (64)$$

여기에서 K_g 는 자이로 스케일 팩터의 대각행렬, b_g 는 자이로 바이어스, 그리고 η_{gu}, η_{gv} 는 영평균 가우스 잡음(Zero mean Gaussian noise)이다. 또한, 가속도계의 수학적 모델은 다음과 같다.

$$\tilde{f}_b = (I + K_a) f_b + \eta_{av} + b_a \quad (65)$$

$$\dot{b}_a = \eta_{au}, \quad \dot{s}_a = \eta_{as} \quad (66)$$

여기에서 K_a 는 가속도계 스케일 팩터의 대각행렬, b_a 는 가속도계 바이어스, 그리고 η_{au}, η_{av} 는 영평균 가우스 잡음(Zero mean Gaussian noise)이다.

1) 약결합 방식(Loosely Coupled System)

일반적으로 약결합 방식은 GPS에서 위치와 속도를 직접적으로 활용하여 INS의 오차를 보정한다. 여기에서 GPS 안테나에서 측정되는 위치와 속도는 다음과 같이 주어진다.

$$\begin{bmatrix} \tilde{P}_G^e \\ \tilde{V}_G^e \end{bmatrix} = \begin{bmatrix} P_G^e \\ V_G^e \end{bmatrix} + \begin{bmatrix} \eta_p \\ \eta_v \end{bmatrix} \quad (67)$$

여기에서 η_p 와 η_v 는 수신기의 잡음을 의미한다. 이를 활용하여 선형화된 측정

방정식을 다음과 같이 나타낼 수 있다.

$$\mathbf{z}_l = \begin{bmatrix} \tilde{\mathbf{P}}_G^e - \hat{\mathbf{P}}_G^e \\ \tilde{\mathbf{V}}_G^e - \hat{\mathbf{V}}_G^e \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{p} \\ \delta \mathbf{P}_G^e \\ \delta \mathbf{V}_G^e \\ \delta \mathbf{b}_g \\ \delta \mathbf{b}_a \\ \delta \mathbf{s}_g \\ \delta \mathbf{s}_a \end{bmatrix} + \mathbf{v}_l \quad (68)$$

여기에서 $\hat{\mathbf{P}}^e$ 과 $\hat{\mathbf{V}}^e$ 는 INS에서 추정한 위치와 속도이다. 일반적으로 GPS/INS 통합항법시스템에서는 위치정보만 활용하여 구성하는 경우도 있다. 이 경우 측정 방정식은 다음과 같다.

$$\mathbf{z}_l = [\tilde{\mathbf{P}}_G^e - \hat{\mathbf{P}}_G^e] = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{p} \\ \delta \mathbf{P}^e \\ \delta \mathbf{V}^e \\ \delta \mathbf{b}_g \\ \delta \mathbf{b}_a \\ \delta \mathbf{s}_g \\ \delta \mathbf{s}_a \end{bmatrix} + \mathbf{v}_l \quad (69)$$

GPS의 위치는 레버암(Lever arm)에 의해 오프셋 된다. 이를 반영하기 위해 다음과 같은 관계식을 갖는다.

$$\begin{aligned} \mathbf{P}_G^e &= \mathbf{P}^e + \mathbf{C}_B^E \mathbf{l}_G^b \\ \mathbf{V}_G^e &= \mathbf{V}^e + [\boldsymbol{\omega}_{eb}^e]^\times \mathbf{C}_B^E \mathbf{l}_G^b \\ &= \mathbf{V}^e + [\mathbf{C}_B^E \boldsymbol{\omega}_{eb}^b - \boldsymbol{\omega}_{ie}^e]^\times \mathbf{C}_B^E \mathbf{l}_G^b \end{aligned} \quad (70)$$

여기에서 $\mathbf{l}_G^b$ 는 레버암 벡터를 나타내고 상수라고 가정하자. 따라서 예측된 GPS의 위치와 속도는 다음과 같이 나타낼 수 있다.

$$\bar{\mathbf{P}}_G^e = \bar{\mathbf{P}}^e + \mathbf{C}_B^{\bar{E}} \mathbf{l}_G^b \quad (71)$$

$$\bar{\mathbf{V}}_G^e = \bar{\mathbf{V}}^e + [\mathbf{C}_B^{\bar{E}} \bar{\boldsymbol{\omega}}_{eb}^b - \bar{\boldsymbol{\omega}}_{ie}^e]^\times \mathbf{C}_B^{\bar{E}} \mathbf{l}_G^b \quad (72)$$

앞서 유도된 방정식을 활용하면 다음과 같은 형태로 나타낼 수 있다.

이를 칼만필터에 적용하기 위한 행렬형태로 나타내면 다음과 같이 나타낼 수

$$\delta \mathbf{P}_G^e = \delta \mathbf{P}^e - 2 \left[\bar{\mathbf{C}}_B^e \bar{\mathbf{l}}_G^p \right]^\times \delta \mathbf{f}^e \quad (73)$$

$$\begin{aligned} \delta \mathbf{V}_G^e = & \delta \mathbf{V}^e + 2 \left\{ \left[\left[\bar{\mathbf{C}}_B^e \bar{\mathbf{l}}_G^p \right]^\times \bar{\mathbf{C}}_B^e \bar{\boldsymbol{\omega}}_{lb}^b \right]^\times + \boldsymbol{\Omega} \left[\bar{\mathbf{C}}_B^e \bar{\mathbf{l}}_G^p \right]^\times \right\} \delta \mathbf{f}^e \\ & - \left[\bar{\mathbf{C}}_B^e \bar{\mathbf{l}}_G^p \right]^\times \bar{\mathbf{C}}_B^e \delta \mathbf{b}_g^b \end{aligned} \quad (74)$$

있다.

$$\mathbf{z}_l = \begin{bmatrix} \tilde{\mathbf{P}}_G^e - \hat{\mathbf{P}}_G^e \\ \tilde{\mathbf{V}}_G^e - \hat{\mathbf{V}}_G^e \end{bmatrix} = \begin{bmatrix} H_{lpq} & I & 0 & 0 & 0 & 0 & 0 \\ H_{lq} & 0 & I & H_{lg} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{p} \\ \delta \mathbf{P}_G^e \\ \delta \mathbf{V}_G^e \\ \delta \mathbf{b}_g^b \\ \delta \mathbf{b}_a^b \\ \delta \mathbf{s}_g^b \\ \delta \mathbf{s}_a^b \end{bmatrix} + \mathbf{v}_l \quad (75)$$

여기에서 $\mathbf{v}_l$ 은 측정 잡음을 나타내며, H_{lpq} , H_{lq} , H_{lg} 는 아래와 같다.^[1]

$$\begin{aligned} H_{lpq} &= -2 \left[\bar{\mathbf{C}}_B^e \bar{\mathbf{l}}_G^p \right]^\times \\ H_{lq} &= 2 \left\{ \left[\left[\bar{\mathbf{C}}_B^e \bar{\mathbf{l}}_G^p \right]^\times \bar{\mathbf{C}}_B^e \bar{\boldsymbol{\omega}}_{lb}^b \right]^\times + \boldsymbol{\Omega} \left[\bar{\mathbf{C}}_B^e \bar{\mathbf{l}}_G^p \right]^\times \right\} \\ H_{lg} &= - \left[\bar{\mathbf{C}}_B^e \bar{\mathbf{l}}_G^p \right]^\times \bar{\mathbf{C}}_B^e \end{aligned} \quad (76)$$

2) 강결합 방식(Tightly Coupled System)

실질적으로 강결합에 의한 항법시스템은 INS와 GPS 수신기를 하드웨어 수준에서 다룰 수 있다. 약결합에 의한 방법은 일반적으로 IMU와 GPS수신기를 따로 구매하여 각각의 출력에 대한 보정을 수행하게 된다. 따라서 강결합 알고리즘을 적용하기 위해서는 그에 따른 하드웨어 작업이 필요하다.

일반적으로 GPS 수신기와 위성간의 의사거리는 다음과 같이 나타 낼 수 있다.

$$R_G = \left\| \mathbf{P}_G^e - \mathbf{P}_s^e \right\| \quad (77)$$

여기에서 $\mathbf{P}_G^e, \mathbf{P}_s^e$ 는 GPS와 위성의 위치를 나타낸다. 일반적으로 GPS의 위치는 알려져 있으므로, GPS에 의한 의사거리는 다음과 같이 나타낼 수 있다.

$$\rho = R_G(\mathbf{P}_G^e) + c\tau + \varepsilon_p \quad (78)$$

여기에서 c 는 빛의 속도 τ 는 GPS의 클락 바이어스 ε_p 는 수신기의 잡음을 의미한다.

다음으로 추정된 의사거리를 다음과 같이 나타 낼 수 있다.

$$\bar{\rho} = R_G(\bar{\mathbf{P}}_G) + c\bar{\tau} \quad (79)$$

여기에서 $\bar{\tau}$ 는 추정된 바이어스이다. 따라서 선형화된 측정방정식은 다음과 같다.

$$\rho - \bar{\rho} = \mathbf{H}_{pp} \delta \mathbf{P}_G^e + c \delta \tau + \varepsilon_p \quad (80)$$

여기에서 $\delta \tau = \tau - \bar{\tau}$ 이며 $\mathbf{H}_{pp}$ 는 아래와 같다.

$$\mathbf{H}_{pp} = \frac{\partial R_G}{\partial \mathbf{P}_G^e} = \frac{(\bar{\mathbf{P}}_G - \mathbf{P}_S^e)^T}{\|\bar{\mathbf{P}}_G - \mathbf{P}_S^e\|} \quad (81)$$

다음으로 GPS에 의해 획득된 의사속도에 대해 고려해보자.

$$\dot{R}_G = \frac{(\mathbf{P}_G^e - \mathbf{P}_S^e)^T (\mathbf{V}_G^e - \mathbf{V}_S^e)}{\|\mathbf{P}_G^e - \mathbf{P}_S^e\|} \quad (82)$$

또한, 일반적으로 의사속도는 다음과 같은 관계식을 갖는다.

$$\dot{R}_G = \dot{R}_G(\mathbf{P}^e, \mathbf{V}^e) \quad (83)$$

따라서 측정된 의사속도는 다음과 같이 나타낼 수 있다.

$$\dot{\rho} = \dot{R}_G(\mathbf{P}_G^e, \mathbf{V}_G^e) + c\dot{\tau} + \varepsilon_v \quad (84)$$

그리고 추정된 의사속도는 다음과 같다.

$$\dot{\bar{\rho}} = \dot{R}_G(\bar{\mathbf{P}}_G, \bar{\mathbf{V}}_G) + c\dot{\bar{\tau}} \quad (85)$$

따라서 선형화된 측정 속도 방정식은 아래와 같이 표현 할 수 있다.

$$\dot{\rho} - \dot{\bar{\rho}} = \mathbf{H}_{vp} \delta \mathbf{P}_G^e + \mathbf{H}_{vv} \delta \mathbf{V}_G^e + c \delta \dot{\tau} + \varepsilon_v \quad (86)$$

$$\mathbf{H}_{vp} = \frac{\partial \dot{R}_G}{\partial \mathbf{P}_G^e} = \frac{(\bar{\mathbf{V}}_G^e - \mathbf{V}_S^e)^T}{\|\bar{\mathbf{P}}_G - \mathbf{P}_S^e\|} (\mathbf{I} - \mathbf{H}_{pp}^T \mathbf{H}_{pp}) \quad (87)$$

$$\mathbf{H}_v = -\frac{\partial \dot{\mathbf{R}}_G}{\partial \mathbf{V}_G^e} = \mathbf{H}_{pp} \quad (88)$$

여기에서 $\mathbf{H}_v$ 는 매우 작은 값으로 일반적으로 무시할 수 있다. 또한 τ 과 $\dot{\tau}$ 의 추정치도 필요한데, 이들은 일반적으로 다음과 같은 2차항으로 모델링하여 추정한다.

$$\frac{d}{dt} \begin{bmatrix} c\delta\tau \\ c\delta\dot{\tau} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} c\delta\tau \\ c\delta\dot{\tau} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \varepsilon_\tau \quad (89)$$

이들을 위의 식에 대입하여 정리하면 측정값 방정식을 다음과 같이 나타낼 수 있다.^[1]

$$\mathbf{z} = \begin{bmatrix} \rho - \bar{\rho} \\ \dot{\rho} - \dot{\bar{\rho}} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{pp} & 0 & \mathbf{H}_{pq} & 0 & 0 & 1 & 0 \\ \mathbf{H}_{vp} & \mathbf{H}_v & \mathbf{H}_{vq} & 0 & \mathbf{H}_{vg} & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta \mathbf{P}^e \\ \delta \mathbf{V}^e \\ \delta \mathbf{q}^e \\ \delta \mathbf{b}_a^b \\ \delta \mathbf{b}_g^b \\ c\delta\tau \\ c\delta\dot{\tau} \end{bmatrix} + \begin{bmatrix} \varepsilon_p \\ \varepsilon_v \end{bmatrix} \quad (90)$$

$$\mathbf{H}_{pp} = \frac{(\bar{\mathbf{P}}_G - \mathbf{P}_S^e)^T}{\|\bar{\mathbf{P}}_G - \mathbf{P}_S^e\|} \quad (91)$$

$$\mathbf{H}_{pq} = -2\mathbf{H}_{pp} \left[\bar{\mathbf{C}}_B^E \mathbf{l}_G^b \right]^\times \quad (92)$$

$$\mathbf{H}_{vp} = \frac{(\bar{\mathbf{V}}_G^e - \mathbf{V}_S^e)^T}{\|\bar{\mathbf{P}}_G - \mathbf{P}_S^e\|} (\mathbf{I} - \mathbf{H}_{pp}^T \mathbf{H}_{pp}) \quad (93)$$

$$\mathbf{H}_{vq} = 2\mathbf{H}_{pp} \left(\left[\left[\bar{\mathbf{C}}_B^E \mathbf{l}_G^b \right]^\times \bar{\mathbf{C}}_B^E \bar{\boldsymbol{\omega}}_b^b \right]^\times + \boldsymbol{\Omega} \left[\bar{\mathbf{C}}_B^E \mathbf{l}_G^b \right]^\times \right) - 2\mathbf{H}_{vp} \left[\bar{\mathbf{C}}_B^E \mathbf{l}_G^b \right]^\times \quad (94)$$

$$\mathbf{H}_{vg} = -\mathbf{H}_{pp} \left[\bar{\mathbf{C}}_B^E \mathbf{l}_G^b \right]^\times \bar{\mathbf{C}}_B^E \quad (95)$$

2. 랜드마크 기반 영상 항법

랜드마크 기반 항법은 영상센서의 영상을 활용하여 항체의 위치와 자세를 계산한다. Figure 2는 카메라 초점면에 투영된 랜드마크를 기반 영상항법을 보여준다. 즉, 영상 항법시스템의 구성은 항체, 카메라, 랜드마크이다. 카메라가 랜드마크를 촬영하면 이를 활용하여 각각의 랜드마크에 대한 시선벡터를 구한다. 이렇게 구해진 시선벡터와 랜드마크를 활용해 항체의 위치 및 자세를 구한다.^[2-3]

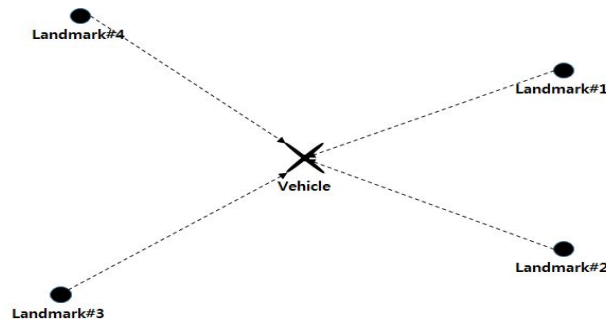


Fig. 2 Vision Navigation System

Figure 3은 카메라 초점면에 투영된 랜드마크의 관계를 나타낸 그림이다. 카메라 좌표계의 x 축은 카메라의 광축, y 축은 x 축을 바라본 상태에서 오른쪽 방향, z 축은 오른손 법칙에 따라 정의한다.^[2-3]

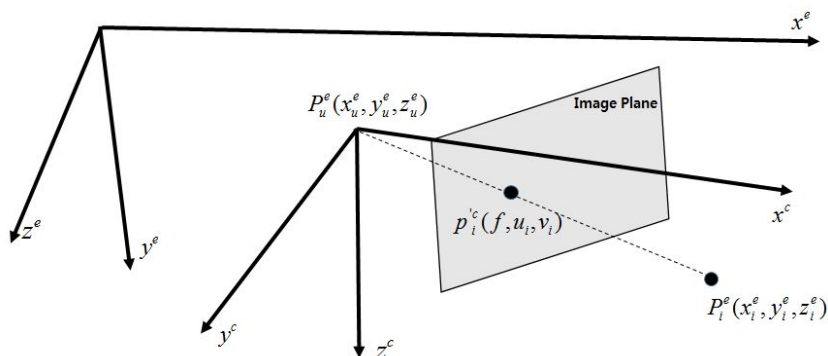


Fig. 3 Sight vector measurement in navigation system

랜드마크 측정값과 항체의 위치 관계식은 랜드마크의 위치와 영상센서에서 측정된 랜드마크 시선벡터의 관계로 얻을 수 있다. 영상센서는 항체에 설치되어 카

메라좌표계에서 작동하므로 각 좌표간의 좌표변환행렬이 필요하다. 영상센서에서 측정한 랜드마크의 시선벡터와 랜드마크의 위치를 이용한 항법방정식은 아래와 같다.

$$\mathbf{P}_i^e - \mathbf{P}_u^e = k_i C_B^E C_C^B \mathbf{p}_i^c \quad (96)$$

여기서, P_i^e 는 i 번째 랜드마크의 ECEF좌표계에서의 위치, P_u^e 는 ECEF좌표계에서의 항체의 위치, p_i^c 는 카메라좌표계에서 i 번째 랜드마크의 시선벡터이다. C_B^E 와 C_C^B 는 좌표변환행렬이고 k_i 는 측정된 이미지에서 랜드마크의 거리와 실제 항체와 랜드마크의 거리의 비로 다음과 같이 표현할 수 있다.

$$k_i = \frac{R_i}{r_i} = \frac{\|P_i^e - P_u^e\|}{\|p_i^c\|} \quad (97)$$

3. 영상 기반 항법알고리즘

본 논문에서는 UUF를 활용한 ECEF 좌표계 기반의 항법시스템을 설계하고자 한다. 사용된 센서는 자이로, 가속도계, GPS, 3개의 영상센서이다. 자이로와 가속도계는 프로세스 업데이트에 사용되며 GPS와 영상센서는 측정값으로서 사용한다. 우선, 추정하고자 하는 상태변수는 자세, 위치, 속도, 자이로와 가속도계의 바이어스와 스케일 팩터이고 아래와 같이 표현할 수 있다.

$$\hat{\mathbf{x}}_k^+ = \begin{bmatrix} \hat{\mathbf{p}}_k^+ \\ \hat{\mathbf{P}}_{k}^{e+} \\ \hat{\mathbf{V}}_{k}^{e+} \\ \hat{\boldsymbol{\beta}}_{gk}^+ \\ \hat{\boldsymbol{\beta}}_{ck}^+ \\ \hat{\mathbf{s}}_{gk}^+ \\ \hat{\mathbf{s}}_{ck}^+ \end{bmatrix} \quad (98)$$

또한, 추정치 오차 공분산(P_k), 프로세스 오차 공분산(Q_k), 측정값 오차 공분산(R_k)은 다음과 같이 정의한다.

$$P_k = \begin{bmatrix} \sigma_{\mathbf{p}_k}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_{\mathbf{p}_k'}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{v}_k'}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{p}_{gk}}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{p}_{sk}}^2 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{s}_{gk}}^2 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{s}_{sk}}^2 \end{bmatrix} \quad (99)$$

$$Q_k = \begin{bmatrix} \sigma_{\mathbf{w}_b}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_{\mathbf{b}_g}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{s}_g}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{t}_b}^2 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{b}_a}^2 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\mathbf{s}_a}^2 \end{bmatrix} \quad (100)$$

$$R_k = \begin{bmatrix} \sigma_{\mathbf{t}_{S,1}}^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{\mathbf{t}_{S,i}}^2 \end{bmatrix} \quad (101)$$

이때, 식 (101)에서 영상센서에 관련된 부분($\sigma_{\mathbf{t}_{S,i}}^2$)에서 i 는 랜드마크의 개수이다. 즉 R_k 는 랜드마크의 개수에 대비해 증가한다.

1) Prediction

예측을 위해 시그마 포인트를 정의해야한다. 식 (48) ~ (50)을 활용해 시그마 포인트를 정의하면 아래와 같다.

여기에서, $\hat{\eta}$ 는 INS의 오차 추정값으로 최적의 값은 0이고, $\hat{\mathbf{q}} = [\hat{\mathbf{p}} \ \hat{\mathbf{q}}_4]^T = [\hat{\mathbf{q}}_1 \ \hat{\mathbf{q}}_2 \ \hat{\mathbf{q}}_3 \ \hat{\mathbf{q}}_4]^T$ 이다. 즉, $\hat{\mathbf{p}}_k^+$ 는 쿼터니언의 자세 벡터이고, $\otimes$ 는 쿼터니언 곱으로 아래와 같이 계산한다. 또한 아래 식 (105)에서 $[\delta \hat{\mathbf{p}} \times]$ 는 교대행렬(skew symmetric matrix)로 아래와 같이 표현한다.

$$\chi_k^c(0) = \begin{bmatrix} \hat{\mathbf{x}}_k^+ \\ \hat{\mathbf{u}}_k^+ \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{p}}_k^+ \\ \hat{\mathbf{P}}_k^{e+} \\ \hat{\mathbf{V}}_k^{e+} \\ \hat{\beta}_{gk}^+ \\ \hat{\beta}_{ak}^+ \\ \hat{s}_{gk}^+ \\ \hat{s}_{ak}^+ \\ \hat{\eta}_g^+ \\ \hat{\eta}_{\beta_{gk}}^+ \\ \hat{\eta}_{s_{gk}}^+ \\ \hat{\eta}_l^+ \\ \hat{\eta}_{\beta_{ak}}^+ \\ \hat{\eta}_{s_{ak}}^+ \end{bmatrix} \quad (102)$$

$$\chi_k^c(i) = \begin{bmatrix} \hat{\mathbf{x}}_k^+ \\ \hat{\mathbf{u}}_k^+ \end{bmatrix} + \left(\gamma_p \sqrt{\mathbf{P}_k^a} \right)_i = \begin{bmatrix} \delta \hat{\mathbf{q}}_{k,i}^+ \otimes \hat{\mathbf{q}}_k^+ \\ \hat{\mathbf{P}}_k^{e+} + \delta \hat{\mathbf{P}}_{k,i}^{e+} \\ \hat{\mathbf{V}}_k^{e+} + \delta \hat{\mathbf{V}}_{k,i}^{e+} \\ \hat{\beta}_{gk}^+ + \delta \hat{\beta}_{gk,i}^+ \\ \hat{\beta}_{ak}^+ + \delta \hat{\beta}_{ak,i}^+ \\ \hat{s}_{gk}^+ + \delta \hat{s}_{gk,i}^+ \\ \hat{s}_{ak}^+ + \delta \hat{s}_{ak,i}^+ \\ \hat{\eta}_g^+ + \delta \hat{\eta}_{g,i}^+ \\ \hat{\eta}_{\beta_{gk}}^+ + \delta \hat{\eta}_{\beta_{gk},i}^+ \\ \hat{\eta}_{s_{gk}}^+ + \delta \hat{\eta}_{s_{gk},i}^+ \\ \hat{\eta}_l^+ + \delta \hat{\eta}_{l,i}^+ \\ \hat{\eta}_{\beta_{ak}}^+ + \delta \hat{\eta}_{\beta_{ak},i}^+ \\ \hat{\eta}_{s_{ak}}^+ + \delta \hat{\eta}_{s_{ak},i}^+ \end{bmatrix}, \quad \text{for } i=1, \dots, (n+m) \quad (103)$$

$$\delta \hat{\mathbf{q}}_k^+ \otimes \hat{\mathbf{q}}_k^+ = \begin{bmatrix} \delta \hat{q}_4 I_{3 \times 3} + [\delta \hat{\mathbf{p}} \times] \\ -\delta \hat{\mathbf{p}}^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}} \\ \hat{q}_4 \end{bmatrix} \quad (105)$$

$$\chi_k^c(i) = \begin{bmatrix} \hat{\mathbf{x}}_k^+ \\ \hat{\mathbf{n}}_k^+ \end{bmatrix} - \left(\gamma_p \sqrt{\mathbf{P}_k^r} \right)_i = \begin{bmatrix} \left(\delta \hat{\mathbf{q}}_{k,i}^+ \right)^{-1} \otimes \hat{\mathbf{q}}_k^+ \\ \hat{\mathbf{P}}_k^{\mathbf{e}+} - \delta \hat{\mathbf{P}}_{k,i}^{\mathbf{e}+} \\ \hat{\mathbf{V}}_k^{\mathbf{e}+} - \delta \hat{\mathbf{V}}_{k,i}^{\mathbf{e}+} \\ \hat{\boldsymbol{\beta}}_{gk}^+ - \delta \hat{\boldsymbol{\beta}}_{gk,i}^+ \\ \hat{\boldsymbol{\beta}}_{ak}^+ - \delta \hat{\boldsymbol{\beta}}_{ak,i}^+ \\ \hat{\mathbf{s}}_{gk}^+ - \delta \hat{\mathbf{s}}_{gk,i}^+ \\ \hat{\mathbf{s}}_{ak}^+ - \delta \hat{\mathbf{s}}_{ak,i}^+ \\ \hat{\eta}_{gk}^+ - \delta \hat{\eta}_{gk,i}^+ \\ \hat{\eta}_{\beta gk}^+ - \delta \hat{\eta}_{\beta gk,i}^+ \\ \hat{\eta}_{s_{gk}}^+ - \delta \hat{\eta}_{s_{gk,i}}^+ \\ \hat{\eta}_{l_i}^+ - \delta \hat{\eta}_{l_i,i}^+ \\ \hat{\eta}_{\beta_{ak}}^+ - \delta \hat{\eta}_{\beta_{ak,i}}^+ \\ \hat{\eta}_{s_{ak}}^+ - \delta \hat{\eta}_{s_{ak,i}}^+ \end{bmatrix}, \quad \text{for } i = (n+m)+1, \dots, 2(n+m) \quad (104)$$

$$[\delta \hat{\mathbf{p}} \times] = \begin{bmatrix} 0 & -\delta \hat{\mathbf{p}}_3 & \delta \hat{\mathbf{p}}_2 \\ \delta \hat{\mathbf{p}}_3 & 0 & -\delta \hat{\mathbf{p}}_1 \\ -\delta \hat{\mathbf{p}}_2 & \delta \hat{\mathbf{p}}_1 & 0 \end{bmatrix} \quad (106)$$

주어진 시그마 포인트를 활용해서 가속도와 각속도의 추정 값을 아래와 같이 구한다.

$$\chi_{ki}^{\omega} = \left(I - [\chi_{ki}^{S_{\mathbf{b}k}}] \right)^T (\tilde{\omega} - \chi_{ki}^{\beta_{\mathbf{b}k}} - \chi_{ki}^{\eta_k}) \quad (107)$$

$$\chi_{ki}^f = \left(I - [\chi_{ki}^{S_{\mathbf{b}k}}] \right)^T (\tilde{\omega} - \chi_{ki}^{\beta_{\mathbf{b}k}} - \chi_{ki}^{\eta_k}) \quad (108)$$

이렇게 구해진 시그마 포인트를 주어진 항법시스템 운동방정식에 대입하여 전파(Propagation)를 진행한다.

$$\dot{\mathbf{q}}^e = \frac{1}{2} \Omega_{\mathbf{e}b}^p (\chi_{k,i}^{\omega}) \chi_{k,i}^f \quad (109)$$

$$\dot{\mathbf{P}}^e = \chi_{k,i}^e \quad (110)$$

$$\dot{\mathbf{V}}^e = \mathbf{C}_B^E \chi_{k,i}^f - 2\Omega_{\mathbf{e}b}^p \chi_{k,i}^e - \Omega_{\mathbf{e}b}^p \chi_{k,i}^f + \mathbf{G}^e \quad (109)$$

$$\chi_{k+1,i}^{b_g} = \chi_{k,i}^{b_g} + \chi_{k,i}^{\eta_{b_g}} \Delta t \quad (110)$$

$$\chi_{k+1,i}^{b_a} = \chi_{k,i}^{b_a} + \chi_{k,i}^{\eta_{b_a}} \Delta t \quad (111)$$

$$\chi_{k+1,i}^{s_g} = \chi_{k,i}^{s_g} + \chi_{k,i}^{\eta_{s_g}} \Delta t \quad (112)$$

$$\chi_{k+1,i}^{s_a} = \chi_{k,i}^{s_a} + \chi_{k,i}^{\eta_{s_a}} \Delta t \quad (113)$$

우선 (107) ~ (109)은 적분하여 $\chi_{k+1,i}^{f^e}$, $\chi_{k+1,i}^{p^e}$, $\chi_{k+1,i}^{v^e}$ 을 구하고 이렇게 업데이트된 시그마 포인트를 활용해 $\hat{x}_{k+1}^-$ (predicted mean)과 공분산을 다음과 같이 구한다.

$$\hat{x}_{k+1}^- = \sum_{i=-(n+m)}^{(n+m)} W_i^{\mathcal{F}} \begin{bmatrix} \chi_{k+1,i}^{f^e} \\ \chi_{k+1,i}^{p^e} \\ \chi_{k+1,i}^{v^e} \\ \chi_{k+1,i}^{b_g} \\ \chi_{k+1,i}^{b_a} \\ \chi_{k+1,i}^{s_g} \\ \chi_{k+1,i}^{s_a} \end{bmatrix} \quad (114)$$

$$P_{k+1} = \sum_{i=-(n+m)}^{(n+m)} W_i^{\mathcal{F}} \begin{bmatrix} \chi_{k+1,i}^d \otimes (\hat{\mathbf{q}}_{k+1}^-)^{-1} & \chi_{k+1,i}^d \otimes (\hat{\mathbf{q}}_{k+1}^-)^{-1} \\ \chi_{k+1,i}^{p^e} - \hat{\mathbf{p}}_{k+1}^- & \chi_{k+1,i}^{p^e} - \hat{\mathbf{p}}_{k+1}^- \\ \chi_{k+1,i}^{v^e} - \hat{\mathbf{v}}_{k+1}^- & \chi_{k+1,i}^{v^e} - \hat{\mathbf{v}}_{k+1}^- \\ \chi_{k+1,i}^{b_g} - \hat{\mathbf{b}}_g^- & \chi_{k+1,i}^{b_g} - \hat{\mathbf{b}}_g^- \\ \chi_{k+1,i}^{b_a} - \hat{\mathbf{b}}_a^- & \chi_{k+1,i}^{b_a} - \hat{\mathbf{b}}_a^- \\ \chi_{k+1,i}^{s_g} - \hat{\mathbf{s}}_g^- & \chi_{k+1,i}^{s_g} - \hat{\mathbf{s}}_g^- \\ \chi_{k+1,i}^{s_a} - \hat{\mathbf{s}}_a^- & \chi_{k+1,i}^{s_a} - \hat{\mathbf{s}}_a^- \end{bmatrix}^T \quad (115)$$

(114)에서 쿼터니언은 더한 후 (24)와 같이 정규화(Normalization)을 진행한 후 (115)를 업데이트 한다. 또한 $W_i^{\mathcal{F}}$ 는 식 (54) ~ (55)에 정의되어 있다.

2) Measurement update

관측 갱신(Measurement update)를 진행하기에 앞서 예측(Prediction)과 마찬가지로 식(56)을 사용해 시그마 포인트를 정의해야 한다.

$$\chi_{k+1}^m(0) = \begin{bmatrix} \hat{\mathbf{x}}_k^+ \\ \hat{\mathbf{n}}_k^+ \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{p}}_k^- \\ \hat{\mathbf{P}}_k^- \\ \hat{\mathbf{V}}_k^- \\ \hat{\boldsymbol{\beta}}_{gk}^- \\ \hat{\boldsymbol{\beta}}_{ak}^- \\ \hat{\mathbf{s}}_{gk}^- \\ \hat{\mathbf{s}}_{ak}^- \\ \hat{\boldsymbol{\eta}}_{I_G^e}^- \\ \hat{\boldsymbol{\eta}}_{I_G^e}^- \\ \hat{\boldsymbol{\eta}}_{lS}^- \end{bmatrix} \quad (116)$$

$$\chi_k^n(i) = \begin{bmatrix} \hat{\mathbf{x}}_{k+1}^- \\ \hat{\mathbf{n}}_k^- \end{bmatrix} + \left(\gamma_m \sqrt{\mathbf{M}_k^e} \right)_i = \begin{bmatrix} (\delta \hat{\mathbf{q}}_{k,i}^-) \otimes \hat{\mathbf{q}}_k^- \\ \hat{\mathbf{P}}_k^- + \delta \hat{\mathbf{P}}_{k,i}^- \\ \hat{\mathbf{V}}_k^- + \delta \hat{\mathbf{V}}_{k,i}^- \\ \hat{\boldsymbol{\beta}}_{gk}^- + \delta \hat{\boldsymbol{\beta}}_{gk,i}^- \\ \hat{\boldsymbol{\beta}}_{ak}^- + \delta \hat{\boldsymbol{\beta}}_{ak,i}^- \\ \hat{\mathbf{s}}_{gk}^- + \delta \hat{\mathbf{s}}_{gk,i}^- \\ \hat{\mathbf{s}}_{ak}^- + \delta \hat{\mathbf{s}}_{ak,i}^- \\ \hat{\boldsymbol{\eta}}_{I_G^e}^- + \delta \hat{\boldsymbol{\eta}}_{I_G^e}^- \\ \hat{\boldsymbol{\eta}}_{I_G^e}^- + \delta \hat{\boldsymbol{\eta}}_{I_G^e}^- \\ \hat{\boldsymbol{\eta}}_{lS}^- + \delta \hat{\boldsymbol{\eta}}_{lS}^- \end{bmatrix}, \quad \text{for } i=1, \dots, (n+p) \quad (117)$$

$$\chi_k^m(i) = \begin{bmatrix} \hat{\mathbf{x}}_{k+1}^- \\ \hat{\mathbf{n}}_k^- \end{bmatrix} - \left(\gamma_m \sqrt{\mathbf{M}_k^e} \right)_i = \begin{bmatrix} (\delta \hat{\mathbf{q}}_{k,i}^-)^{-1} \otimes \hat{\mathbf{q}}_k^- \\ \hat{\mathbf{P}}_k^- - \delta \hat{\mathbf{P}}_{k,i}^- \\ \hat{\mathbf{V}}_k^- - \delta \hat{\mathbf{V}}_{k,i}^- \\ \hat{\boldsymbol{\beta}}_{gk}^- - \delta \hat{\boldsymbol{\beta}}_{gk,i}^- \\ \hat{\boldsymbol{\beta}}_{ak}^- - \delta \hat{\boldsymbol{\beta}}_{ak,i}^- \\ \hat{\mathbf{s}}_{gk}^- - \delta \hat{\mathbf{s}}_{gk,i}^- \\ \hat{\mathbf{s}}_{ak}^- - \delta \hat{\mathbf{s}}_{ak,i}^- \\ \hat{\boldsymbol{\eta}}_{I_G^e}^- - \delta \hat{\boldsymbol{\eta}}_{I_G^e}^- \\ \hat{\boldsymbol{\eta}}_{I_G^e}^- - \delta \hat{\boldsymbol{\eta}}_{I_G^e}^- \\ \hat{\boldsymbol{\eta}}_{lS}^- - \delta \hat{\boldsymbol{\eta}}_{lS}^- \end{bmatrix}, \quad \text{for } i=(n+p)+1, \dots, 2(n+p) \quad (118)$$

여기에서 $\hat{\boldsymbol{\eta}}$ 는 영상센서의 오차 추정값으로 최적 추정값은 0이다.

이렇게 구해진 시그마 포인트를 사용해 측정값 방정식에 대입한다.

$$\chi_{k+1,i}^m = h_S \left(\chi_{k,i}^{m,P^e}, \chi_{k,i}^{m,n,h_G^e} \right) \quad (119)$$

여기에서, h_S 는 식 (97)을 활용해서 정의한다. 즉, 주어진 시그마 포인트와 랜드마크를 사용해 시선벡터 추정치를 구한다. 또한, 식 (60) ~ (62)와 같이 $\hat{y}_{k+1}^-$ (predicted measurement mean), P_{k+1}^{yy} (measurement covariance), P_{k+1}^{xy} (cross-correlation matrix)를 구한다.

$$\hat{y}_{k+1}^- = \sum_{i=(n+p)}^{(n+p)} W_i^m \chi_{k+1,i}^m \quad (120)$$

$$P_{k+1}^{yy} = \sum_{i=(n+p)}^{(n+p)} W_i^m (\chi_{k+1,i}^m - \hat{y}_{k+1}^-)(\chi_{k+1,i}^m - \hat{y}_{k+1}^-)^T \quad (121)$$

$$P_{k+1}^{xy} = \sum_{i=(n+p)}^{(n+p)} W_i^m \begin{bmatrix} \chi_{k+1,i}^q \otimes (\hat{q}_{k+1}^-)^{-1} \\ \chi_{k+1,i}^{P^e} - \hat{P}_{k+1}^{e-} \\ \chi_{k+1,i}^{V^e} - \hat{V}_{k+1}^{e-} \\ \chi_{k+1,i}^{b_g} - \hat{b}_g^- \\ \chi_{k+1,i}^{b_a} - \hat{b}_a^- \\ \chi_{k+1,i}^{s_g} - \hat{s}_g^- \\ \chi_{k+1,i}^{s_a} - \hat{s}_a^- \end{bmatrix} [\chi_{k+1,i}^m - \hat{y}_{k+1}^-]^T \quad (122)$$

마지막으로 (123) ~ (126)을 사용해 아래와 같이 칼만게인, 공분산, 측정값을 업데이트 한다.

$$K_{k+1} = P_{k+1}^{xy} (P_{k+1}^{yy})^{-1} \quad (123)$$

$$dx = K_{k+1} (y_{k+1} - \hat{y}_{k+1}^-) \quad (124)$$

$$\hat{x}_{k+1}^+ = \hat{x}_{k+1}^- + dx \quad (125)$$

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} P_{k+1}^{xy} K_{k+1}^T \quad (126)$$

단, 식 (125)에서 쿼터니언은 식 (124)에서 자세 관련한 부분을 쿼터니언 곱을 활용해 업데이트한다. 우선 $d\mathbf{q}$ 를 다음과 같이 정의한다.

$$d\mathbf{q} = [d\mathbf{x}^p \left[d\mathbf{x}^p \sqrt{1 - (d\mathbf{x}^p)^T d\mathbf{x}^p} \right]] \quad (127)$$

이를 사용해 아래와 같이 업데이트를 한다.

$$\hat{\mathbf{q}}_{k+1}^+ = d\mathbf{q} \otimes (\hat{\mathbf{q}}_{k+1}^-) \quad (128)$$

3. Measurement Covariance Update Method

영상센서를 사용하여 항법시스템을 구축할 때 영상센서에서 측정되는 랜드마크의 개수가 3개 이상이어야 정확한 항법솔루션을 제공할 수 있다. 하지만 영상센서는 안개나 전방에 큰 건물과 같이 시야가 확보될 수 없는 상황에서는 3개 이하의 랜드마크가 측정될 가능성이 항상 존재한다. 랜드마크가 3개 이하로 측정될 경우 센서의 정확도가 떨어지고 이로 인해 시스템의 오차가 증가한다. 이러한 문제점을 해결하기 위해 본 연구에서는 랜드마크의 개수가 3개 이하 일 경우 P_{k+1}^y (Measurement Covariance)를 아래와 같이 증가시켜 필터 내에서 비전센서의 가중치에 변화를 주는 방식을 사용했다.

$$P_{k+1}^y = P_{k+1}^y + \alpha \quad (129)$$

이때, α 는 설계자가 설정할 수 있는 상수이다. 이렇게 α 를 더하여 P_{k+1}^y 를 증가시키면 칼만게인이 감소하고 이로 인해 필터에서 현재 영상센서 출력값에 대한 가중치가 줄어든다.

4. 시뮬레이션

1) UUF기반 영상항법시스템 성능 검증 및 비교 시뮬레이션

본 논문에서 제안된 필터 성능 및 방법에 대한 검증을 위해 시뮬레이션을 수행하였다. 시뮬레이션에서는 비행체의 자세, 위치, 속도, 자이로와 가속도계의 바이어스 스케일 팩터를 추정한다. 모든 센서의 관측정보는 0.1초 간격(10Hz)으로 관측된다고 가정한다. 또한, 영상센서는 9개의 랜드마크가 고정적으로 측정된다

고 가정한다. 전체 시뮬레이션 시간은 5분이며 시뮬레이션에서 사용된 센서들의 성능은 아래 Table 5과 같다.

센서	Value	UNIT
가속도계		
Max. bias	10	μg
. scale factor	200	ppm
Random walk	11	$\mu g/h^{1/2}$
Rate random walk	180	$\mu g/h^{3/2}$
자이로스코프		
Max. bias	0.3	deg/h
. scale factor	200	ppm
Angle random walk	0.03	$deg/h^{1/2}$
Rate random walk	20	$deg/h^{3/2}$
GPS Receiver		
Position Error	5	m
Velocity Error	0.05	m/s
영상센서		
FOV(Field of View)	40	deg
가시거리	500	m
초점거리	25	mm
가우시안 잡음	100	μm

Table. 5 Specification of Sensors

우선, 본 논문에서 제시한 UUF기반의 다중 영상센서 기반의 항법알고리즘에 대한 검증을 위해 자세, 위치, 속도의 수렴 성능을 검증한 결과 자세, 위치, 속도 모두 3sigma에 수렴하는 것을 확인하였다. Figure 4~6는 수렴성능에 대한 그래프이다.

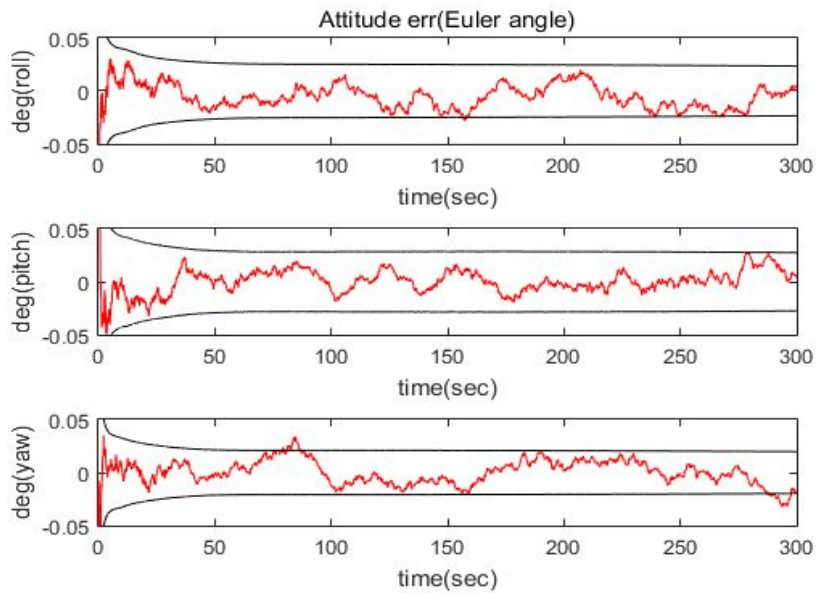


Fig. 4 Attitude error of GPS/INS/Multi-Vision integrated navigation system

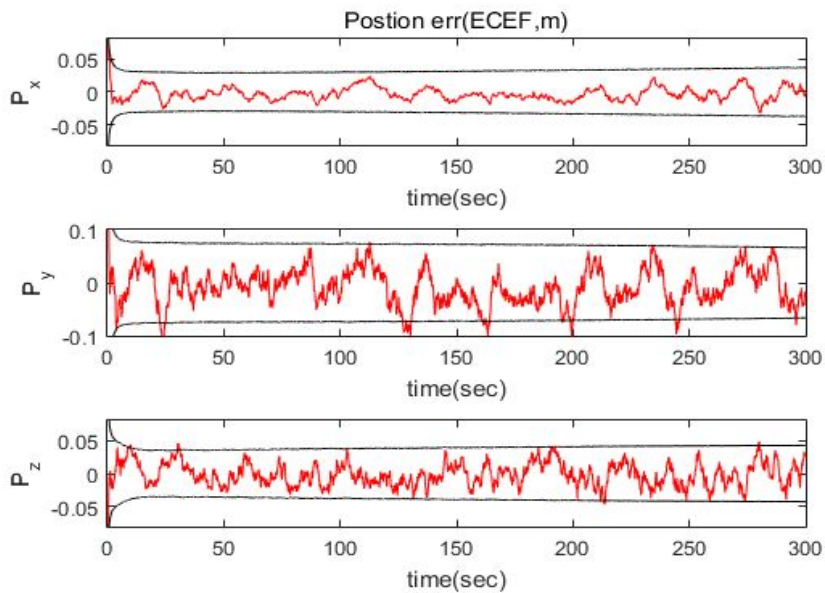


Fig. 5 Position error of GPS/INS/Multi-Vision integrated navigation system

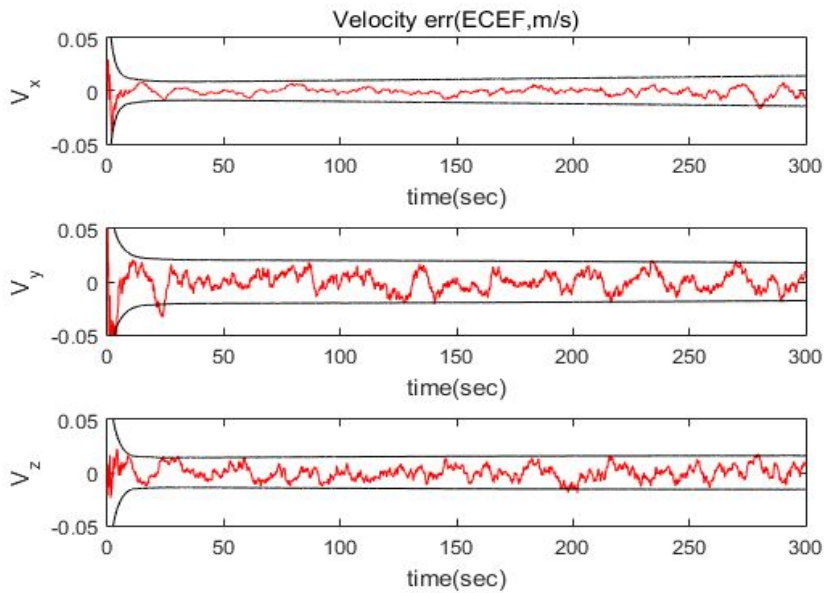


Fig. 6 Velocity error of GPS/INS/Multi-Vision integrated navigation system

다음으로 진행할 시뮬레이션은 본 논문에서 제시하는 알고리즘의 성능을 비교 및 분석하기 위한 시뮬레이션으로서 UUF를 활용하여 기존의 항법시스템에서 가장 일반적으로 사용하는 AUKE(UKF for Additive noise Case)보다 오차를 줄이고 UKF보다 계산량이 적은 UUF를 활용한 항법시스템 구축이 주된 목적이다. 이를 위해 UKF, AUKE, UUF 각각의 오차와 계산시간을 비교하는 시뮬레이션을 진행했다. 또한, FOV가 40도인 영상센서 한 개를 사용하고 랜드마크가 9개가 측정되는 경우와 아래 Figure 3과 같이 FOV가 40도인 영상..센서 세 개를 120도 간격으로 설치하고 각각의 센서에서 3개의 랜드마크가 측정되는 경우에 대한 비교를 진행하였다.

시뮬레이션 결과 대부분 주어진 센서 성능에 해당하는 수렴성능을 보여주었다. 대표적으로 자세, 위치, 속도에 대한 수렴성능은 아래 Figure 5 ~ 10과 같다. 모든 그래프는 대부분 상태변수의 오차가 3sigma에 수렴하는 양상을 보인다. 이때, 검정색 점선이 3sigma이고 빨간색 실선은 오차 값이다. 또한, 모든 오차와 3sigma를 아래와 같이 크기(norm)를 활용해 비교했다.

$$\delta x_{norm} = \sqrt{\delta x_x^2 + \delta x_y^2 + \delta x_z^2} \quad (129)$$

아래의 Figure 7~9은 1개의 영상센서를 활용한 경우에 대한 결과이고 Figure 10~12은 다중 영상센서를 활용한 결과이다.

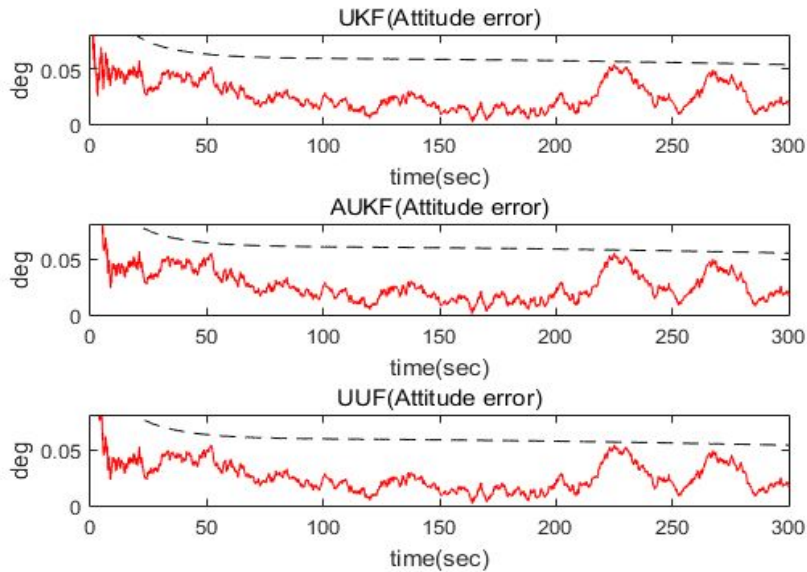


Fig. 7 Attitude Error of UKF, UKF for additive & UUF
(Single Vision Sensor)

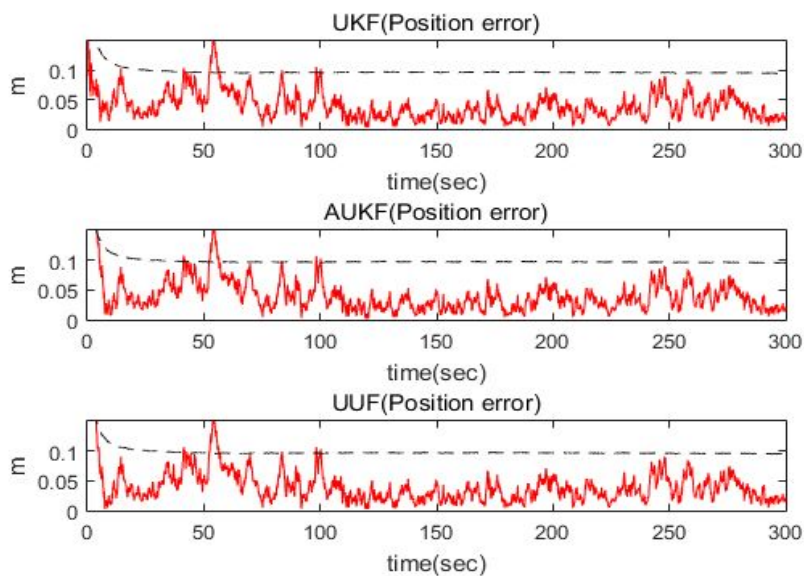


Fig. 8 Position Error of UKF, UKF for additive & UUF
(Single Vision Sensor)

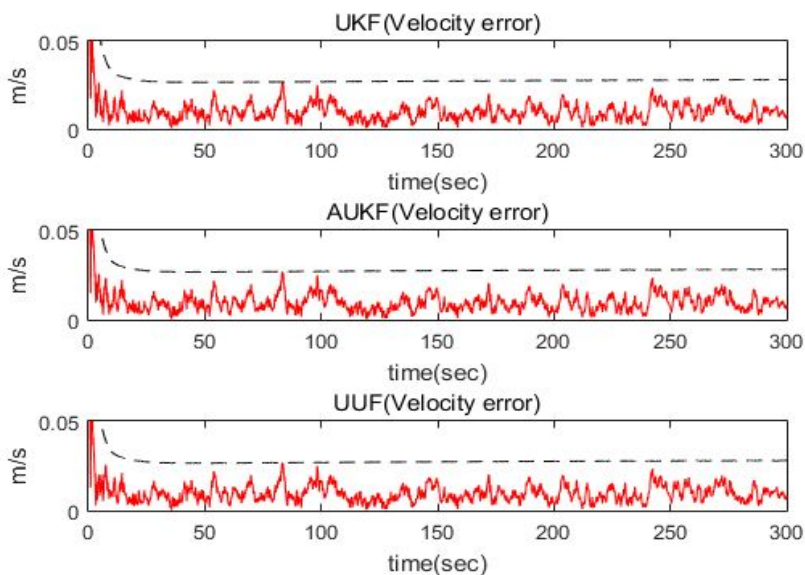


Fig. 9 Velocity Error of UKF, UKF for additive & UUF
(Single Vision Sensor)

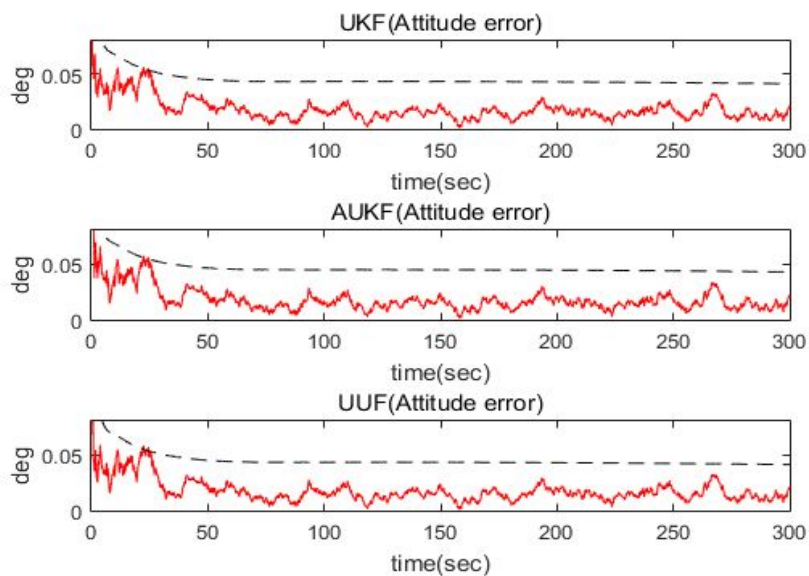


Fig. 10 Attitude Error of UKF, UKF for additive & UUF
(Multi Vision Sensors)

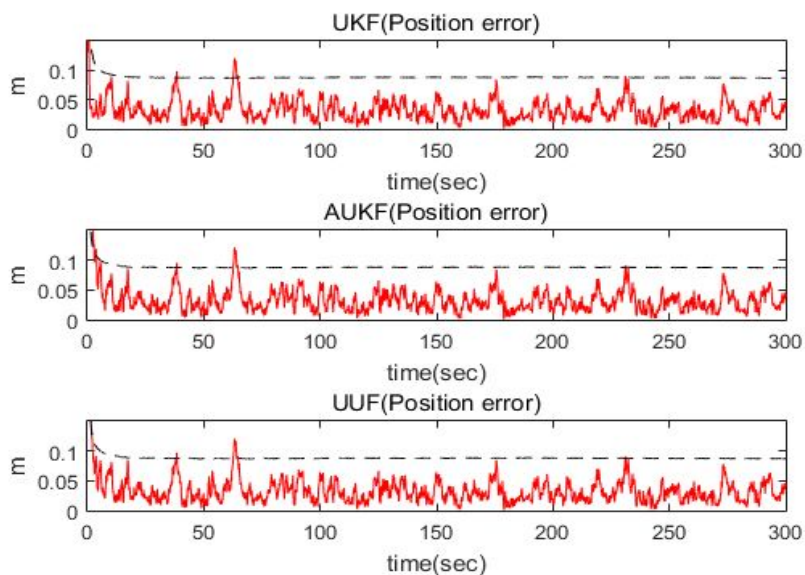


Fig. 11 Position Error of UKF, UKF for additive & UUF
(Multi Vision Sensors)

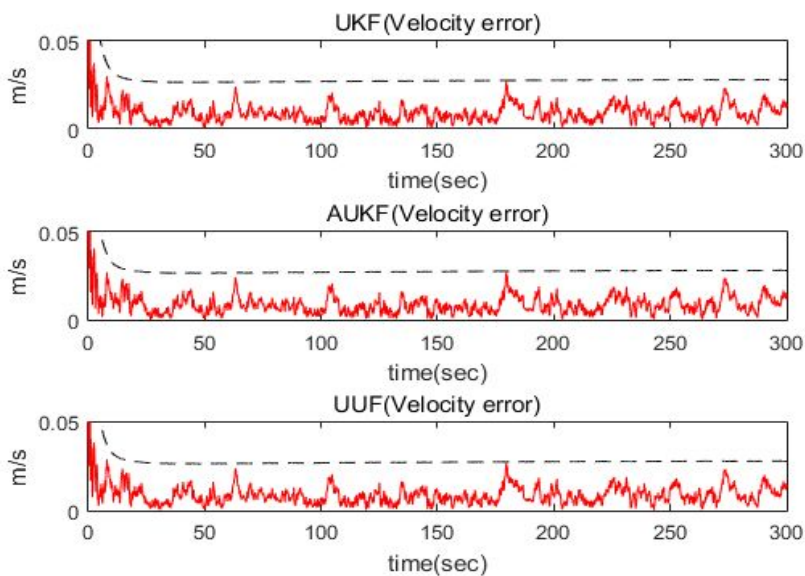


Fig. 12 Velocity Error of UKF, UKF for additive & UUF
(Multi Vision Sensors)

본 연구에서는 앞서 언급한 바와 같이 3가지 UF(Unscented Filtering)과 단일 영상센서와 다중 영상센서를 비교 및 분석하는 시뮬레이션을 진행하였다. 우선, UF의 비교는 실제 그래프만으로는 확인하기 어렵고 영상센서를 이용한 비교의 경우 두 경우모두 총 랜드마크의 개수는 9개로 동일하지만 다중 영상센서를 활용한 경우가 3sigma와 오차 모두 작음을 확인 할 수 있다. 다음으로, UF에 따른 오차 비교는 그래프만으로는 차이를 확인하기 어렵다. 따라서 각각의 오차를 아래와 Table 6과 같이 평균 제곱근 편차(Root Mean Square Error, RMSE)를 구해 정리하여 비교를 진행하였다.

Case 1. Single Vision Sensor(FOV : 40°)			
	Attitude RMSE (deg)	Position RMSE (m)	Velocity RMSE (m/s)
UKF	2.6896×10^{-2}	4.0240×10^{-2}	9.8605×10^{-3}
AUKF	2.9378×10^{-2}	4.5199×10^{-2}	9.9401×10^{-3}
UUF	2.8157×10^{-2}	4.4268×10^{-2}	9.8804×10^{-3}
Case 2. Multi Vision Sensor(FOV : 40°, 3EA)			
	Attitude RMSE	Position RMSE	Velocity RMSE
UKF	1.8117×10^{-2}	3.2833×10^{-2}	8.9114×10^{-3}
AUKF	1.8279×10^{-2}	3.7095×10^{-2}	9.0060×10^{-3}
UUF	1.7796×10^{-2}	3.6062×10^{-2}	8.9681×10^{-3}

Table. 6 RMSE(Root Mean Square Error)

우선 3가지의 UF를 비교해보면 자세, 위치, 속도가 Case 1과 2모두 UKF가 가장 적은 오차를 보였고 그 다음으로 UUF 그리고 AUKF가 가장 RMSE가 컸다. 다음으로 Case 1과 2를 비교하면 Case 2가 RMSE가 작음을 확인 할 수 있다.

마지막으로, 각각의 알고리즘의 계산량에 대한 분석을 진행하였다. 계산량 측정은 시뮬레이션을 진행했던 Matlab의 기능 중 실행 시간 측정을 활용했다. 하지만 Matlab에서 제공하는 시간은 컴퓨터 성능에 따라 전체 시간은 많은 차이를 보인다. 하지만 각각 알고리즘에 대한 비율은 유사하므로 시간이 아닌 비율로 작성하였다. 시뮬레이션의 결과는 아래 Table 7과 같다.

UKF	49.43%
UUF	35.53%
AUKF	15.04%

Table. 7 Comparison Calculation time

이번 시뮬레이션의 결과 분석은 다음과 같다. 우선, Case 1의 결과를 살펴보면 자세는 3sigma가 0.0540(UKF), 0.0552(AUKF), 0.0552(UUF)로 수렴하는 양상을 보인다. 오차의 경우 그래프를 통해서도 비교가 힘들어 평균 제곱근 편차를 구하여 비교해본 결과 UKF, UUF, AUKF 순으로 오차가 작았다. 다음으로 위치는 3sigma가 0.0951(UKF), 0.0955(AUKF), 0.0955(UUF)로 수렴하고 RMSE는 자세와 같은 결과이다. 마지막으로 속도의 3sigma는 0.0279(UKF), 0.0281(AUKF), 0.0281(UUF)이고 RMSE는 동일하다. 다음으로, Case 2에서 자세를 살펴보면 3sigma가 0.0412(UKF), 0.0428(AUKF), 0.0428(UUF)로 수렴하는 양상을 보인다. RMSE는 UKF, UUF, AUKF 순으로 오차가 작았다. 다음으로 위치는 3sigma가 0.0868(UKF), 0.0870(AUKF), 0.0870(UUF)로 수렴하고 RMSE는 자세와 같은 결과이다. 마지막으로 속도의 3sigma는 0.0278(UKF), 0.0280(AUKF), 0.0280(UUF)이고 RMSE는 동일하다. 마지막으로 계산시간은 컴퓨터의 사양에 따라 차이가 심하기 때문에 비율로 나타냈다. 결과는 Table 7에서와 같이 AUKF(15.04%), UUF(35.53%), UKF(49.43%)의 결과가 나왔다. 정리하면 UKF의 경우 계산시간이 가장 많지만 3sigma와 오차가 다른 두 방법보다 작음을 확인하였다. AUKF는 계산시간은 가장 작지만 오차가 다른 두 알고리즘 보다 크다. 마지막으로 UUF의 계산시간은 AUKF보다는 길고 UKF는 작다. 또한 오차역시 같은 결과이다.

2) Measurement Covariance Update Method 검증 시뮬레이션

본 논문에서 제안한 알고리즘의 성능을 검증하기 위해 시뮬레이션을 수행하였다. 시뮬레이션에서는 비행체의 자세, 위치, 속도, 자이로와 가속도계의 바이어스 스케일 팩터를 추정한다. 모든 센서의 관측정보는 0.1초 간격(10Hz)으로 관측된다고 가정한다. 2가지의 경우에 대한 비교를 진행했다. 랜드마크가 3개 이하인 경우에 하나는 식 (129)를 적용하고 다른 하나는 랜드마크의 개수와 상관없이 업데이트를 진행했다. 우선, 시뮬레이션을 위해 사용된 센서 성능은 다음과 같다.

센서	Value	UNIT
자이로스코프		
Random walk	0.035	deg/s
Bias	10	deg/h
Scale Factor	100	ppm
가속도계		
Random walk	80	μg
Bias	60	μg
Scale Factor	100	ppm
영상센서		
FOV(Field of View)	40	deg
가시거리	500	m
초점거리	25	mm
가우시안 잡음	200	μm

Table. 8 Specification of Sensors

또한, 초기 오차는 자세가 20°, 위치가 10m, 속도가 1m/s이다. 모든 오차는 각 축에 대한 오차이다. 비교 및 검증 시뮬레이션은 앞선 Fig 7~12와 같이 Norm을 활용하여 그래프를 표현하고 RMSE를 비교한다. 아래의 Fig 13~16가 시뮬레이션 결과이다.

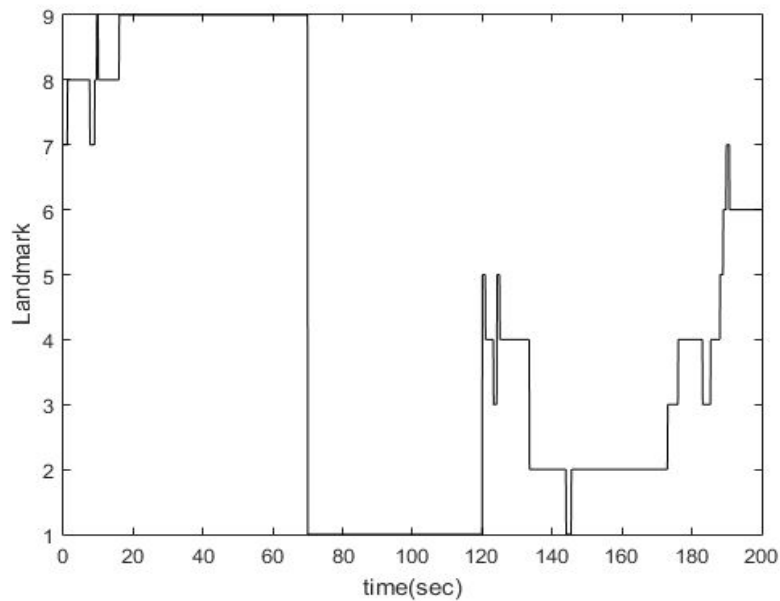


Fig. 13 Number of Landmark

Figure 13은 영상센서에서 측정되는 랜드마크의 개수이다. 이는 시뮬레이션의 시나리오로서 중간에 랜드마크가 1~2개가 측정되는 부분이 오차가 증가하는 구간이다.

Figure 14~16을 살펴보면 자세, 위치, 속도에 대한 그래프이다. 빨간색이 일반적인 알고리즘을 그대로 사용한 결과이고 검정색 그래프가 본 논문에서 제안한 방식을 적용한 그래프이다.

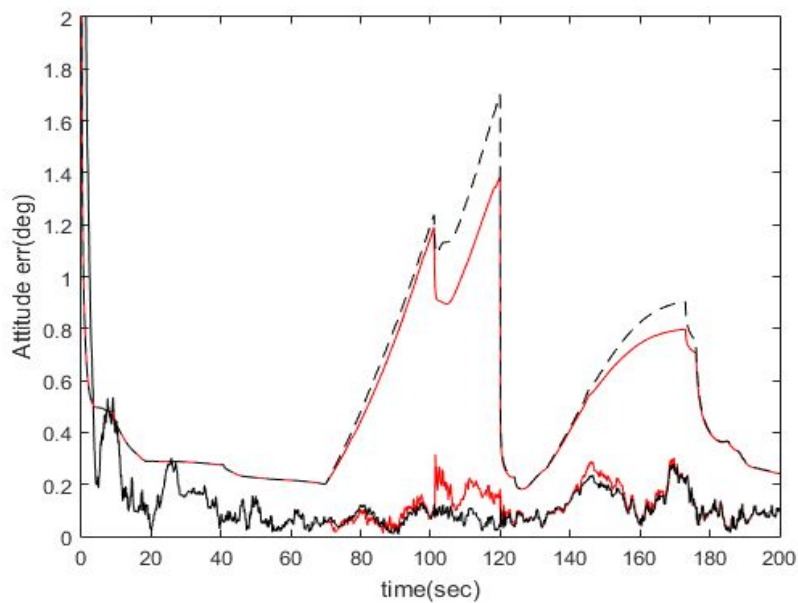


Fig. 14 Attitude error of Measurement Covariance Update Method

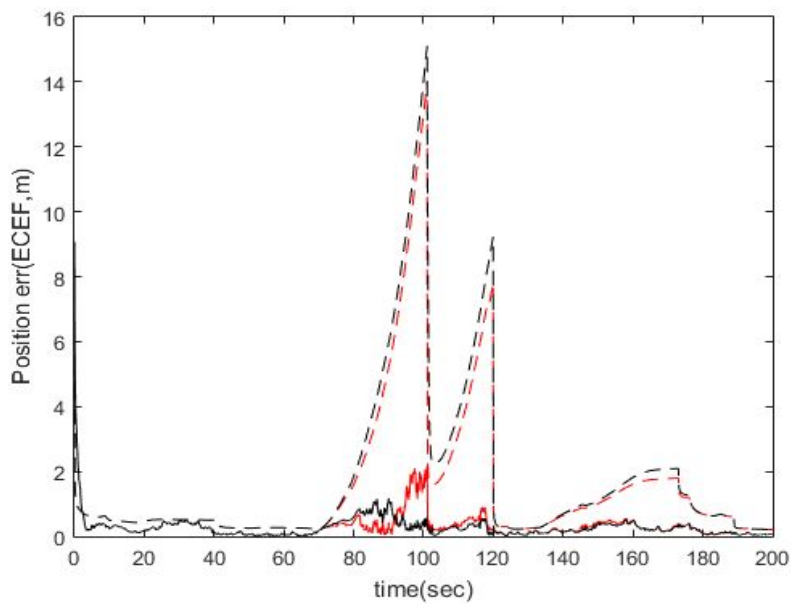


Fig. 15 Position error of Measurement Covariance Update Method

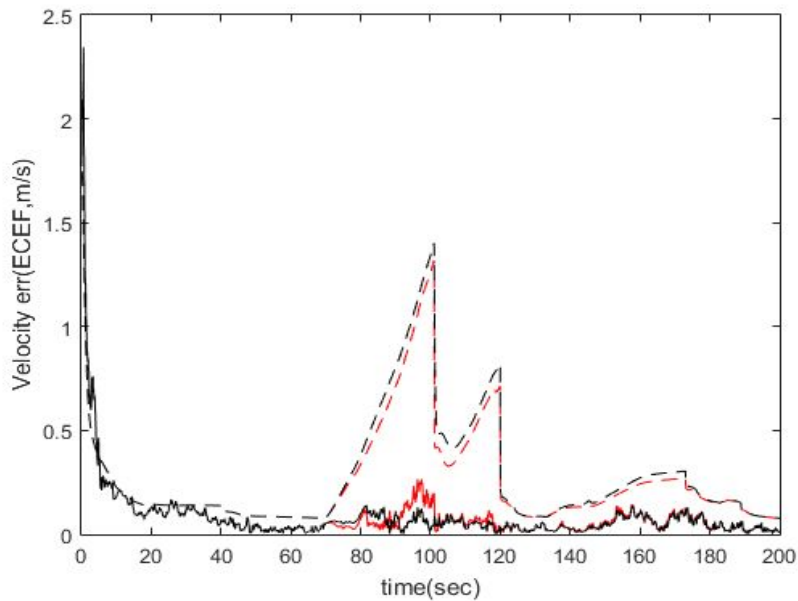


Fig. 16 Velocity error of Measurement Covariance Update Method

그래프를 보면 3sigma는 MCUM이 더 크다. 이는 Measurement Update과정에서 랜드마크가 적은 경우 측정값 공분산의 값을 키웠기 때문이다. 하지만 오차는 기존의 알고리즘보다 더 작음을 확인 할 수 있다. 더 확실한 결과 비교를 위해 RMSE를 구해 비교했다.

	Attitude RMSE	Position RMSE	Velocity RMSE
Original	0.1675	0.3107	0.0906
proposed	0.1538	0.2759	0.0859

Table. 9 RMSE(Root Mean Square Error)

Table 9에서 확인 할 수 있듯이 본 연구에서 제안한 알고리즘을 적용한 결과 랜드마크가 충분하지 않은 상황에서도 자세가 9%, 위치가 11%, 속도가 5% 정도의 성능 증가를 보였다.

V. 결론

본 논문에서는 전방 시야에 의존하는 기존의 비전 항법시스템의 문제점을 개선하고자 다수의 영상센서를 활용한 영상 항법시스템을 구축하고 랜드마크의 개수가 3개 이하일 경우 측정값 오차의 증가가 항법시스템에 미치는 영향을 최소화시키기 위한 알고리즘을 제안하고 이를 검증하였다.

먼저, UUF를 기반으로 GPS/INS/다중영상센서를 사용한 항법시스템을 설계하고 이를 수치 시뮬레이션을 통해 검증했다. 성능검증을 위해 UUF뿐만 아니라 UKF와 AUKF(UKF for Additive system)을 기반으로 항법시스템을 설계하여 비교를 진행했다. 또한, 다중 영상센서를 활용한 항법시스템에 미치는 영향을 검증을 위해 다음과 같은 두 가지 경우에 대한 비교를 진행했다. 첫 번째로, FOV가 40도이고 랜드마크가 9개가 측정되는 경우로 총 랜드마크는 9개가 측정되는 영상센서를 모델링하고(Case 1), 두 번째는 FOV가 40도이고 3개의 랜드마크가 측정되는 영상센서 3개를 활용한 경우로(Case 2) 총 랜드마크의 개수는 9개로 두 경우 모두 총 랜드마크 개수는 동일이다.

또한, UKF의 본래 특성을 유지하고 오차를 감소시킨 UUF를 영상 항법시스템에 적용하였고 UKF보다 계산량은 적고 AUKF보다 오차는 작음을 확인하였다. 또한, 다중 영상센서를 활용한 항법알고리즘을 제시하고 이를 단일 영상센서 항법 알고리즘과 비교하여 측정되는 총 랜드마크의 개수가 같더라도 오차가 작아짐을 확인했다. 이로써, UUF 기반의 다중 영상센서를 활용한 항법시스템을 구현할 수 있을 것으로 판단된다.

다음으로, 랜드마크의 개수가 3개 이하일 경우 측정값 오차의 증가가 항법시스템에 미치는 영향을 최소화시키기 위한 알고리즘인 식 (129)을 검증하기 위해 UUF기반의 INS/Vision 항법시스템을 활용했다. 검증을 위해 일정 구간에서 랜드마크의 개수를 3개 이하로 설정했다. 그 결과 본 연구에서 제안한 알고리즘을 적용한 필터의 성능이 우수함을 확인하였다.

【참고문헌】

- [1] Groves, Paul D. Principles of GNSS, inertial, and multisensor integrated navigation systems. Artech house, 2013.
- [2] Kim, Youngsun, and Dong-Hwan Hwang. "INS/Vision integrated navigation system considering error characteristics of landmark-based vision navigation." Journal of Institute of Control, Robotics and Systems 19.2 (2013): 95-101.
- [3] George, Michael, and Salah Sukkarieh. "Camera aided inertial navigation in poor GPS environments." 2007 IEEE Aerospace Conference. IEEE, 2007.
- [4] Chatterji, G. B., P. K. Menon, and B. Sridhar. "GPS/machine vision navigation system for aircraft." IEEE Transactions on aerospace and electronic systems 33.3 (1997): 1012-1025.
- [5] Bryson, Mitch, and Salah Sukkarieh. "Building a Robust Implementation of Bearing only Inertial SLAM for a UAV." Journal of Field Robotics 24.1 2 (2007): 113-143.
- [6] Borenstein, Johann, H. R. Everett, and Liqiang Feng. "Where am I? Sensors and methods for mobile robot positioning." University of Michigan 119.120 (1996): 27.
- [7] J. Wendel and G. F. Trommer, "Tightly coupled GPS/INS integration for missile applications," Aerospace Science and Technology, vol. 8, no. 7, pp. 627-634, Oct. 2004.
- [8] H. S. Hong, J. G. Lee, and C. G. Park, "Performance improvement of in-flight alignment for autonomous vehicle under large initial heading error," IEE Proceedings - Radar, Sonar, and Navigation, vol. 151, no. 1, pp. 57-62, Feb. 2004.
- [9] S. Y. Cho and B. D. Kim, "Adaptive IIR/FIR fusion filter and its application to INS/GPS integrated system," Automatica, vol. 44, no. 8, pp. 2040-2047, Aug. 2008.

- [10] W. Y. Kang, K. J. Kim, and C. G. Park, "'Performance analysis of in-flight alignment using UKF,'" Journal of Institute of Control, Robotics and Systems, vol. 12, no. 11, pp. 1124-1129, Nov. 2006.
- [11] Leeghim, Henzeh, Yoonhyuk Choi, and Belgacem A. Jaroux. "Uncorrelated unscented filtering for spacecraft attitude determination." Acta Astronautica 67.1 (2010): 135-144.
- [12] Crassidis, John L., and F. Landis Markley. "Unscented filtering for spacecraft attitude estimation." Journal of guidance, control, and dynamics 26.4 (2003): 536-542.
- [13] Crassidis, John L., and John L. Junkins. Optimal estimation of dynamic systems. CRC press, 2011.
- [14] Simon, Dan. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons, 2006.