

9-DOF IMU Data Processing

Hanaro 전자팀 남도현, 김형찬, 김윤서

생성일: 2024.12.26 | 최종 수정일: 2025.01.01

1 소개

본 문서는 2024년 5월부터 시작된 IMU 데이터 처리 방법에 관한 논의 내용을 정리한 것이며 하나로 전자팀의 남도현을 필두로 하여 김형찬, 박선호, 김윤서 등이 참여하였다. IMU 센서에서 수집하는 각속도, 가속도, 자기장 및 GPS와 기압고도계를 토대로 하여 동체의 비행 경로를 정밀하게 추적하는 것이 본 문서 및 데이터 처리 시스템의 목적이다.

2 개요

2.1 동기

데이터 처리 시스템은 2024년 6월, Identity 3의 발사 및 회수에 처음 성공하면서 고도 그래프를 그리던 중에 시작되었다. Identity 3의 전자부에는 Arduino Nano 33 BLE Sense 보드를 사용하며 내장된 IMU 센서로 값을 측정하는데 고도는 기압고도계로부터 계산하지만 부차적으로 가속도, 각속도 및 자기장 값을 읽기 때문에 이로부터 경로를 계산할 수 있을 것이라는 생각을 하게 되었고 이후 IMU + GPS Sensor fusion, Filtering 등에 대해 찾아보게 되었다. 데이터 처리 시스템은 로켓의 발사 및 회수를 최우선적으로 여기던 하나로에서는 새로운 시도이며 그만큼 자유도가 상당히 높은 영역이다.

2.2 목적 및 요구사항

가히 이 문서에서 제일 중요한 부분이라고 칭할 수 있다. 데이터 처리 시스템의 개발 철학, 그것을 묻는 것이다. 데이터 처리 시스템의 목적은 동체 내의 여러 센서 값으로부터 동체의 비행 경로를 정밀하게 계산하는 것이다. 정밀하게 계산한다는 것은 여러 센서 값의 fusion을 통하여 각각의 오차를 보정한 경로를 산출한다는 의미이며 실제 비행 경로를 알지 못하기 때문에 기준을 정할 수 없다. 이 부분은 추후에 타당성 검증을 통하여 알아볼 수 있다.

데이터 처리는 비행 과정에서 동체 내부, 지상국 및 비행 종료 이후의 사후 처리로 나뉜다. 동체 내부에서 경로 계산의 전 과정을 실시간으로 진행하려면 높은 사양의 DSP가 필요하다. 하지만 단순 적분 경로 계산과 같은 나름 간단한 방식의 계산도 노트북의 CPU로 처리하는데 분 단위의 시간이 걸리기 때문에 동체 내부에서 직접 계산하는 것은 거의 불가능하다. 지상국 통신을 통해 받아온 데이터로 실시간 추적을 하는 것도 높은 확률로 불가능할 것이다. 이 또한 계산하는 기기(아마 데이터를 확인하는 노트북일 것)의 사양에 따라 다르겠지만 “실시간”으로 계산한다는 의미를 곧이 곧대로 이어나가긴 쉽지 않다. 그러므로 본 데이터 처리 시스템은 사후 처리를 통해 비행 경로를 알아내는 것을 주 목표로 한다. 각 센서의 오차를 보정해야 하기 때문에 GPS와 IMU 센서의 값을 모두 사용하도록 하며 알고리즘의 확정 및 간소화 과정을 통해 복잡도를 줄여 딜레이가 거의 없이 경로 계산이 가능하도록 하는 것을 그 다음 순위로 둔다. 만일 실시간 계산을 통한 경로의 구체적인 형태가 필요하다면 이는 즉각적으로 계산이 가능한 상용품을 쓰도록 하며 사후 처리는 상용품이 관여하지 못하는 Sensor fusion의 영역을 다루는 것이다.

3 데이터 처리 알고리즘

데이터 처리 알고리즘이란 IMU 센서의 원본 데이터를 토대로 경로를 추적하는 방법을 말한다. 제일 간단하게는 gps와 기압고도계를 활용하여 좌표를 직접 찍을 수 있으며 가속도, 각속도 및 자기장 값으로부터 더 나은 보정 과정을 취할 수도

있다.

3.1 GPS 기반 경로

원본 데이터에는 기압고도계의 기압 및 GPS의 위도, 경도가 있다.

기압: p (hPa)

위도: φ ($\times 10^7$ °)

경도: λ ($\times 10^7$ °)

$$z = 44307.69396 * (1.0 - (p/101325)^{0.190284}) \quad (1)$$

고도 z 는 barometric formula를 통해 m 단위로 얻을 수 있다. 하지만 이는 해수면 기준 고도이기 때문에 발사 고도 z_0 에 대하여 $z - z_0$ 가 비행 높이라고 할 수 있다. 2차원 좌표는 위도와 경도로 구한다. 이 두 숫자로 구면 상에 있는 두 지점의 위치 관계를 알 수 있는데 haversine formula로 두 지점 사이의 최단 거리; 두 지점을 지나는 대원의 원호의 길이를 구할 수 있다,

$$d = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (2)$$

R 은 지구 반지름으로 약 6378km이며 위도, 경도는 radian 단위이다. d 는 두 지점 사이의 거리이고 일반적으로는 이를 $\Delta\lambda$, $\Delta\varphi$ 의 비율에 맞추어 각 축 방향의 성분 크기를 구하면 되지만 Cartesian 좌표계에서는 투영한 값을 써야 한다는 주장이 있다.

$$x = d \Delta\lambda \cos \varphi \quad (3)$$

$$y = d \Delta\varphi \quad (4)$$

후자가 맞다고 생각하지만 코드 상에는 각도의 비율에 따라서 d 를 배분하였다. 이 방법으로 구한 (x, y, z) 가 동체의 경로이다.

3.2 단순 적분 경로

동체에는 IMU 센서가 부착되어 동체 좌표계(body frame)에서 측정한 가속도, 각속도 및 자기장을 알 수 있다. 이론적으로

3.3 필터링 알고리즘

데이터 처리 알고리즘(이하 알고리즘)은 사원수를 기반으로 한 Kalman Filtering(KF) 기법을 활용한다. KF는 센서의 오차 보정에 있어서 널리 쓰이는 방법이며 그 오차의 독립성 및 무작위성이 보장되는 경우에 주로 사용된다. 알고리즘의 구체적인 특성에 대해서는 얘기하지 않겠다. 알고리즘을 구성하는 식은 부록을 참고하기를 바란다. KF는 선형계에 대해서 작용하며 알고리즘의 관계식이 선형적이지 않기 때문에 Extended Kalman Filtering(EKF) 기법을 활용한다.

4 프로젝트 구조

프로젝트는 3단계로 나뉜다: 데이터 전처리, 경로 계산, 시각화. 경로를 산출하는 방식에 따라서 경로 계산 및 시각화 단계에서 차이가 있을 수 있으나 전반적으로는 주어진 원본 데이터(.csv)를 전처리한 후 특정 방식을 따라 경로를 계산하고 이를 시각화하는 순서를 따른다.

4.1 데이터 전처리

주어진 원본 데이터의 구성은 다음과 같다 (데이터 처리 시스템에서 활용할 값만 표시하였다).

- gpsLatitude
- gpsLongitude
- pressure
- xAxisAcc
- yAxisAcc
- zAxisAcc
- xAxisAngVal
- yAxisAngVal zAxisAngVal xAxisMagF yAxisMagF zAxisMagF

- 데이터 별로 csv 파일에 어떻게 나타나고 이를 보정할 것인지 - state 구별하는 방법 - 초기조건 설정 방법

4.2 경로 계산

- rotator 계산 과정

4.3 시각화

- rotator로 좌표계 회전 - RK4 기반 적분 - body frame 표시

5 검증

6 결과

References

A Kalman Filtering Algorithm

Prediction Phase:

- State Prediction:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$$

- Error Covariance Prediction:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

Update Phase:

- Kalman Gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

- State Update:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1})$$

- Error Covariance Update:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

Notations:

- $\hat{\mathbf{x}}_{k|k-1}$: Predicted state estimate at time k .
- $\hat{\mathbf{x}}_{k|k}$: Updated state estimate at time k .
- $\mathbf{P}_{k|k-1}$: Predicted error covariance.
- $\mathbf{P}_{k|k}$: Updated error covariance.
- \mathbf{F}_k : State transition matrix.
- \mathbf{B}_k : Control input matrix.
- \mathbf{u}_k : Control vector at time k .
- \mathbf{Q}_k : Process noise covariance.
- \mathbf{K}_k : Kalman Gain.
- \mathbf{H}_k : Observation matrix.
- \mathbf{R}_k : Measurement noise covariance.
- \mathbf{z}_k : Measurement vector at time k .
- \mathbf{I} : Identity matrix.

B Extended Kalman Filtering

Prediction Phase:

- State Prediction:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k)$$

- Error Covariance Prediction:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

where $\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k}$ is the Jacobian matrix of $f(\cdot)$ with respect to the state.

Update Phase:

- Kalman Gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

where $\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}$ is the Jacobian matrix of $h(\cdot)$ with respect to the state.

- State Update:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}))$$

- Error Covariance Update:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

Notations:

- $f(\cdot)$: Nonlinear state transition function.
- $h(\cdot)$: Nonlinear measurement function.
- $\hat{\mathbf{x}}_{k|k-1}$: Predicted state estimate at time k .
- $\hat{\mathbf{x}}_{k|k}$: Updated state estimate at time k .
- $\mathbf{P}_{k|k-1}$: Predicted error covariance.
- $\mathbf{P}_{k|k}$: Updated error covariance.
- \mathbf{F}_k : Jacobian of $f(\cdot)$ with respect to \mathbf{x} .
- \mathbf{H}_k : Jacobian of $h(\cdot)$ with respect to \mathbf{x} .
- \mathbf{Q}_k : Process noise covariance.
- \mathbf{R}_k : Measurement noise covariance.
- \mathbf{z}_k : Measurement vector at time k .
- \mathbf{K}_k : Kalman Gain.
- \mathbf{I} : Identity matrix.