


2. 리 군 $SO(3)$ 과 리 대수 $\mathfrak{so}(3)$

개념	정의	실전적 의미	
Lie Group $SO(3)$	회전 행렬 R 들의 집합	“상태”가 놓이는 비선형 공간	
Lie Algebra $\mathfrak{so}(3)$	반대칭 행렬 공간 $\{\omega^\wedge \mid \omega \in \mathbb{R}^3\}$	“속도(각속도)”가 놓이는 선형 접공간	
Hat(\wedge) 연산	$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \mapsto \omega^\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$	벡터 \rightarrow 대수	
Vee(\vee) 연산	$\cdot^\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$	대수 \rightarrow 벡터	

왜 “회전(rotation)”이 비선형적인가?

- **구조적 이유**: 회전은 3차원 실수 공간 \mathbb{R}^3 안의 단순한 벡터 더하기가 아니라, **특수 직교군** $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det \mathbf{R} = 1\}$ 위의 군(group) 연산으로 정의됩니다.
- **곡률(curvature)**: $SO(3)$ 는 3차원의 리(Lie) 군이자 매니폴드(manifold)로, 국소적으로는 \mathbb{R}^3 와 동형이지만 전체적으로는 곡률이 있는 구면 위에 놓인 공간입니다. 따라서 단순 덧셈으로는 전역 연산을 표현할 수 없습니다.
- **구성(composition)**: 두 회전 $\mathbf{R}_1, \mathbf{R}_2$ 의 합성은 행렬 곱 $\mathbf{R}_1 \mathbf{R}_2$ 로 정의되며, 이는 **비가환(non-commutative)**입니다. 즉 $\mathbf{R}_1 \mathbf{R}_2 \neq \mathbf{R}_2 \mathbf{R}_1$.

췌...

회전은 우리가 일상적으로 다루는 숫자나 벡터처럼 덧셈, 뺄셈, 평균 계산이 직접적으로 성립하지 않는 **고유한 수학적 구조**를 가지고 있다는 의미입니다. 이러한 특성 때문에 회전을 다루는 것은 복잡하며, 이를 '비선형 구조

예를 들어보자.

우리가 3차원 공간에서의 위치(Position)를 생각해보자.

- A 지점 (1, 0, 0)에서 B 지점 (0, 1, 0)으로 이동하는 것은 벡터 (-1, 1, 0) 만큼의 변화.
- 이 변화량들을 더하거나 빼는 것은 매우 직관적이고 간단합니다. $(1, 0, 0) + (-1, 1, 0) = (0, 1, 0)$ 처럼. 이것이 바로 **선형 공간(Linear Space)**의 특징이다.

하지만 3차원 회전은 다르다.

1. 회전의 순서가 중요하다 (교환 법칙이 성립하지 않음)

2. 책을 한 권 준비합니다.

3. 경우 A:

- 먼저, 책을 앞쪽으로 90도 기울입니다 (Pitch).
- 그 다음, 오른쪽으로 90도 돌립니다 (Yaw).

4. 경우 B:

- 먼저, 책을 오른쪽으로 90도 돌립니다 (Yaw).
- 그 다음, 앞쪽으로 90도 기울입니다 (Pitch).

경우의 최종 책의 방향이 완전히 다른 것을 확인할 수 있습니다. 만약 회전이 선형적이라면 회전 A + 회전 B 와 회전 B + 회전 A 의 결과는 같아야 하지만, 실제로는 그렇지 않다.

2. 회전의 '평균'을 구하기 어렵다

두 개의 회전 상태, Rotation1 과 Rotation2 가 있다고 가정해봅시다. 이 두 회전의 '중간'에 해당하는 회전은 어떻게 계산할 수 있을까?

- **오일러 각 (Euler Angles: Roll, Pitch, Yaw):** 가장 직관적인 표현법. 하지만 특정 각도(예: Pitch 가 $\pm 90^\circ$)에서 Roll과 Yaw가 하나의 축에 묶어버리는 **짐벌 락(Gimbal Lock)** 현상이 발생. 이 지점에서는 회전 자유도 하나를 잃어버리게 되어 올바른 계산이 불가능. 이는 오일러 각 표현법이 특정 지점에서 무너지는, 즉 비선형적이고 불안정하다는 것을 보여주는 대표적인 예.
- **회전 행렬 (Rotation Matrix):** 3x3 행렬로 회전을 표현합니다. 짐벌 락 문제는 없지만, 다음과 같은 제약 조건이 따릅니다.
 - $R^T R = I$ (직교 행렬이어야 함)
 - $\det(R) = 1$ (반사를 포함하지 않는 순수 회전이어야 함)

두 회전 행렬 R1과 R2를 단순히 더하거나 $(R1+R2)/2$ 와 같이 평균을 내면, 그 결과는 더 이상 위 제약 조건을 만족하는 유효한 회전 행렬이 아니다. 최적화 과정에서 이 제약 조건을 계속 유지시켜주는 것은 매우 계산 비용이 비쌉니다 (모든 계산 시에 추가적인 저 계산들을 확인해줘야 하기 때문에.)

위 내용들로 인해서,

IMU 센서에서 짧은 시간 간격(Δt)으로 측정되는 수많은 각속도(Angular Velocity)를 단순히 더해서 누적 회전을 계산하면 오차가 빠르게 쌓이고, 비선형성으로 인해 최적화가 매우 복잡

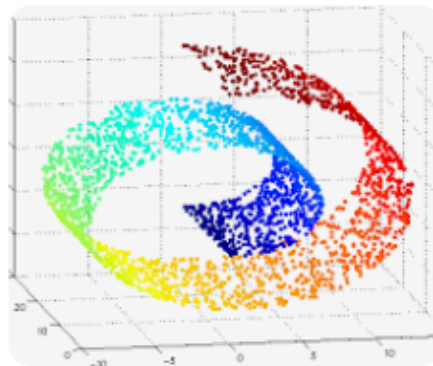
IMU Preintegration on Manifold for Efficient Visual Inertial Maximum a Posteriori Estimation 논문에서는 이 '회전의 비선형 구조' 문제를 해결하기 위해 **매니폴드(Manifold)**와 **리 그룹(Lie Group)** 이론을 사용.

3차원 공간의 모든 가능한 회전들의 집합은 '평평한' 벡터 공간이 아니라, 지구 표면처럼 '**휘어진**' 공간을 형성합니다. 이러한 휘어진 공간을 수학적으로 **매니폴드(Manifold)**라고 부른다.

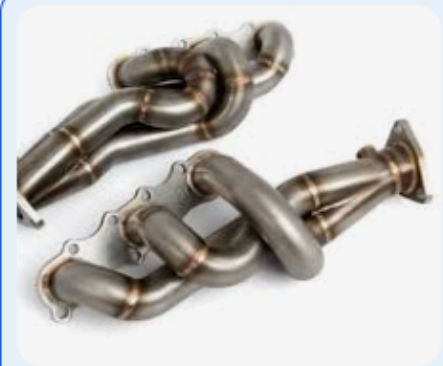
****Many(많은)+fold(끼다)의 합성어**



네이버블로그 - NAVER
매니폴드(Manifold)의 ...



Hack your life - 티스토리
데이터분석] 매니폴드(manifold)...



중국 개스킷, 축매 변환기, 고무 제...
배기 매니폴드란 무엇인가? - 지식



기계치와 함께하는 슬기로운 기계...
배기 매니폴드란? - 정의, 작동...



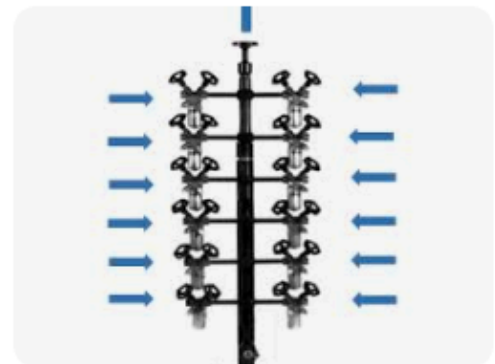
네이버블로그 - NAVER
매니폴드(Manifold)의 정의와 ...



중국 PEX 파이프, PEX 파...
라디에이터 매니폴드란? -...



한국미스미
매니폴드 기술정보 및 설계사례 | MISUMI | 미스미 종합 W...



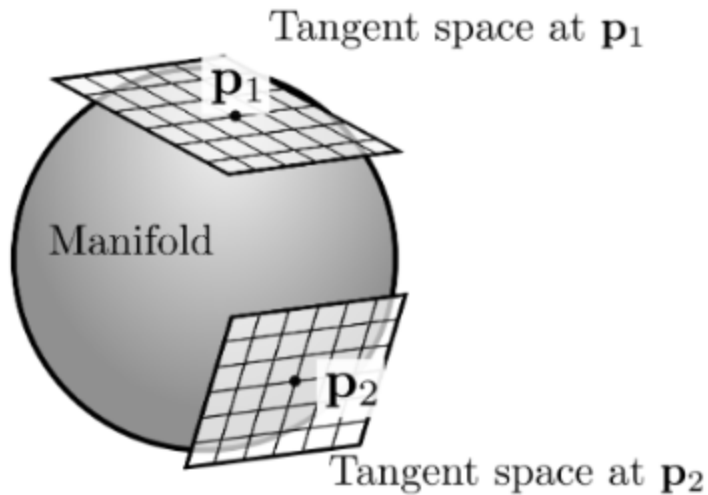
네이버블로그 - NAVER
매니폴드(Manifold)의 정의와 구분! :...

SO(3) 매니폴드: 3차원 회전 행렬들이 모여있는 '공간' 그 자체입니다. 이 공간은 비선형적이고 휘어져 있다.

직접 덧셈이나 평균을 계산하는 것은 복잡하고 비효율적

"휘어진 공간(SO(3))에서 직접 계산하지 말고, 특정 지점에서 그 공간에 접하는 평평한 공간 (Tangent Space)을 빌려서 계산한 뒤, 다시 원래의 휘어진 공간으로 되돌리자!"

이 '접하는 평평한 공간'이 바로 **리 대수(Lie Algebra)**이며, $SO(3)$ 에 대응하는 리 대수는 $\mathfrak{so}(3)$ 라고 표기



$\mathfrak{so}(3)$ 리 대수:

- $SO(3)$ 매니폴드의 '단위 행렬(Identity, 회전 없음)' 지점에서의 **접선 공간(Tangent Space)**.
- 이 공간은 우리가 잘 아는 **3차원 벡터 공간**입니다. 즉, **선형적(Linear)**!
- $\mathfrak{so}(3)$ 에 있는 벡터는 물리적으로 **각속도(Angular Velocity)** 또는 **아주 작은 회전 변화량**을 의미.
- 이 공간 안에서는 벡터의 덧셈, 뺄셈, 스칼라 곱셈이 자유롭게 성립.

위 두 공간을 사상하는 방법이 바로 지수 사상과 로그 사상.

- **지수 사상 (Exponential Map): $\mathfrak{so}(3) \rightarrow SO(3)$**
 - 리 대수 $\mathfrak{so}(3)$ 에 있는 선형적인 각속도 벡터 $\omega \in \mathbb{R}^3$ 를 가져다가, 이를 리 그룹 $SO(3)$ 에 있는 실제 회전 행렬 R 로 변환해주는 역할.
 - 물리적 의미: "각속도 ω 로 Δt 시간만큼 회전하면 어떤 최종 회전 상태가 되는가?"를 계산.
 - 수식: $R = \exp(\omega \Delta t)$
- **로그 사상 (Logarithmic Map): $SO(3) \rightarrow \mathfrak{so}(3)$**
 - 지수 사상의 반대 과정.
 - 리 그룹 $SO(3)$ 에 있는 회전 행렬 R 을 가져다가, 이를 만들어내는 리 대수 $\mathfrak{so}(3)$ 의 벡터 $\omega \Delta t$ 로 변환.
 - 물리적 의미: "두 회전 상태 R_1 과 R_2 의 차이(상대 회전)는 어떤 회전 벡터로 표현되는가?"를 계산할 때 사용.

1. **IMU 측정:** IMU는 짧은 시간 간격마다 각속도(ω)와 가속도(a)를 측정합니다. 이 각속도는 리 대수 $so(3)$ 공간의 벡터
2. **전통적인 방식의 문제점:**
 - k 시점의 회전 R_k 가 주어지면, $k+1$ 시점의 회전은 $R_{k+1}=R_k \times \exp(\omega_k \Delta t)$ 로 계산됩니다.
 - 이 계산은 **전역 좌표계(Global Frame)**에 대한 회전 R_k 에 의존합니다.
 - 최적화 과정에서 i 번째 키프레임의 자세(회전)가 약간 수정되면, 그 이후의 모든 IMU 측정값들을 처음부터 다시 적분해야 합니다. 이는 엄청난 계산 낭비

논문의 해결책: 매니폴드 상에서의 사전 통합

- 두 키프레임 i 와 j 사이의 IMU 측정값들을 미리 하나의 **상대적인(relative)** 움직임으로 통합(요약)해 둡니다.
- 이때, 각속도들을 단순히 더하는 것이 아니라 **매니폴드의 구조를 존중하며** 적분합니다.
- 적분은 **키프레임 i 의 몸체 좌표계(Body Frame)**를 기준으로 수행됩니다. 즉, i 시점의 전역 자세 R_i 와는 무관하게 상대적인 회전 변화량 ΔR_{ij} 만을 계산합니다.
- **핵심:** 각속도 ω_k 는 리 대수 $so(3)$ 상의 벡터이므로, 이 **선형 공간**에서 적분과 관련된 계산을 수행합니다. 그리고 그 결과를 **지수 사상(Exponential Map)**을 통해 $SO(3)$ 매니폴드 위의 유효한 회전 변화량 ΔR_{ij} 로 변환합니다.

즉, 비선형적인 회전의 누적을 직접 계산하는 대신, 선형적인 리 대수 공간에서 변화량을 계산한 뒤, 그 결과를 비선형적인 매니폴드로 옮기는 방식을 통해 정확성과 계산 효율을 모두 잡은 것

최적화의 이점:

- 최적화 과정에서 키프레임 i 의 자세 R_i 가 업데이트되어도, i 와 j 사이의 IMU 측정값을 재적분할 필요가 없습니다.
- 미리 계산해둔 상대 회전 ΔR_{ij} 를 새로운 R_i 에 곱해주기만 하면 되므로($R_j=R_i \times \Delta R_{ij}$), 계산량이 획기적으로 줄어듭니다.
- 바이어스(Bias)가 업데이트될 때도, 이 바이어스가 사전 통합 값에 미치는 영향을 **자코비안(Jacobian)**을 통해 선형적으로 근사하여 빠르게 보정합니다. 이 과정 역시 매니폴드 위에서 정의된 연산을 따릅니다.

-> 약간의 요약

imu에서 받아내는 데이터들이 body frame 기준이었는데 이전까지는 이것을 gps 좌표계나 lidar 좌표계로 옮겨서 표현된 R로 누적을 시켰는데 지금은 imu 센서 데이터(각속도)를 사전적으로 계속 누적시키고 이것을 Rotation 매트릭스로 표현한다는 말

구분	전통적 적분	사전 통합 (Preintegration)
관점	전역(World) 관점	지역(Body) 관점
누적 대상	매 순간의 전역 자세 R_W	두 키프레임 간의 상대 변환 ΔR
의존성	이전 스텝의 전역 자세에 강하게 의존	시작 키프레임의 전역 자세와 독립적
최적화	시작점 수정 시 전체 재계산 (비효율적)	미리 계산된 값 재사용 (효율적)

World Frame 관점

$$\text{수식: } \dot{R}_W(t) = R_W(t) \omega^B(t)^\wedge$$

의미: "로봇의 절대 방향($R_W(t)$)의 시간당 변화율은, 현재 로봇의 절대 방향($R_W(t)$)에 로봇 자신의 관점에서 측정한 각속도($\omega^B(t)$)를 적용한 것과 같다.

Body Frame 관점

$$\frac{d}{dt} \Delta R_{it} = \Delta R_{it} \omega^B(t)^\wedge$$

의미: "시점 i 를 기준으로 한 상대 방향(ΔR_{it})의 시간당 변화율은, 현재의 상대 방향(ΔR_{it})에 로봇 자신의 관점에서 측정한 각속도($\omega^B(t)$)를 적용한 것과 같다."

전통적인 방식: "GPS 신호에 맹목적으로 의존하는 map 앱"

이 방식은 GPS가 알려준 절대 좌표(World Frame)를 기준으로 모든 IMU 데이터를 즉시 해석하려고 한다.

1. (T=0초) GPS 신호 수신:

- GPS가 "현재 위치는 광화문 광장 좌표 (A)이고, 북쪽을 보고 있음"이라고 알려줍니다.
- 앱은 이 정보를 바탕으로 지도에 파란 점을 찍습니다. 이 위치와 방향이 현재의 '절대 자세' $R_{W,0}$ 가 됩니다.

2. (T=0.005초) IMU 데이터 수신:

- IMU가 "폰이 오른쪽으로 0.1도 기울었음"이라는 신호를 보냅니다. (body frame 기준)
- 앱은 즉시 이 정보를 '절대 자세'에 반영합니다.
- 새로운 절대 자세 = (T=0초의 절대 자세) $R_{W,0} \times$ (오른쪽 0.1도 회전) .

3. (T=0.010초) IMU 데이터 수신:

- IMU가 "폰이 또 오른쪽으로 0.1도 기울었음"이라고 신호를 보냅니다.
- 앱은 또다시 방금 계산한 '새로운 절대 자세'에 이 회전을 누적하여 더 새로운 절대 자세를 계산합니다.

4. (T=1초) 문제 발생 - GPS 보정 쇼크:

- 1초 동안 200번의 IMU 계산을 마친 후, 새로운 GPS 신호가 도착합니다.
- GPS: "어, 미안. 사실 지금 위치는 200번의 계산 결과보다 동쪽으로 1m 더 간 (B) 지점이야."
- **결과:** 지도 위의 파란 점이 갑자기 **순간이동**하듯 1m 옆으로 튕 하고 튕니다.

5. 최적화 비용의 증가:

- 시스템이 이 오차를 바로잡기 위해 생각합니다. "아, 1초 전 출발점(RW,0)에 대한 내 믿음이 약간 틀렸구나."
- 이 출발점을 약간 수정하면, 그동안 누적해 온 **200개의 IMU 계산 전체를 처음부터 다시 수행****

BCH formula 유도 -part 1 기본편

<https://steemit.com/krmath/@beoped/bch-formula-part-1>

아래 수식은 다음에...

$$\begin{aligned}\text{Exp}(\omega + \Delta\omega) &= \text{Exp}(\mathbf{J}_l \Delta\omega) \text{Exp}(\omega) \\ \text{Exp}(\Delta\omega) \text{Exp}(\omega) &= \text{Exp}(\omega + \mathbf{J}_l^{-1} \Delta\omega)\end{aligned}$$

$$-\mathbf{J}_l = \mathbf{J}_l(\omega) \in \mathbb{R}^{3 \times 3}$$

$$-\mathbf{J}_l(\omega) = \mathbf{I} + \left(\frac{1 - \cos(\|\omega\|)}{\|\omega\|^2} \right) \omega^\wedge + \left(\frac{\|\omega\| - \sin(\|\omega\|)}{\|\omega\|^3} \right) (\omega^\wedge)^2 : \text{SO}(3) \text{군의 왼쪽 자코비안(left jacobian)}$$

$$-\mathbf{J}_l(\omega)^{-1} = \mathbf{I} - \frac{1}{2} \omega^\wedge + \left(\frac{1}{\|\omega\|^2} - \frac{1 + \cos(\|\omega\|)}{2 \|\omega\| \sin(\|\omega\|)} \right) (\omega^\wedge)^2 : \text{SO}(3) \text{군의 왼쪽 자코비안의 역행렬}$$

All expressions simplify if we assume the magnetic field to point in the z -direction, in which case the frequency matrix becomes

$$\Omega_L = \begin{pmatrix} 0 & \omega_L & 0 \\ -\omega_L & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (2.823)$$

so that

$$\cos \frac{\Omega_L t}{2} = \begin{pmatrix} \cos \omega_L t/2 & 0 & 0 \\ 0 & \cos \omega_L t/2 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.824)$$

and

$$\frac{\sinh \Omega_L t/2}{\Omega_L t/2} = \begin{pmatrix} 0 & \sin \omega_L t/2 & 0 \\ -\sin \omega_L t/2 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.825)$$

whose determinant is

$$\det \left(\frac{\sinh \Omega_L t/2}{\Omega_L t/2} \right) = (\sinh \omega_L t/2)^2 \quad (2.826)$$