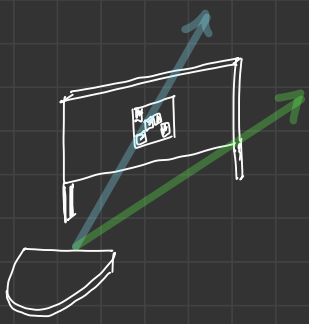
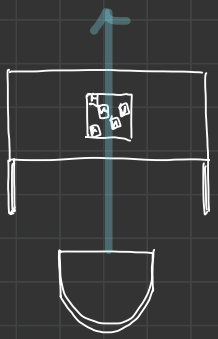
11/15





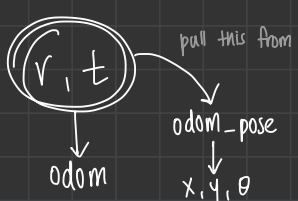
if the Neato is approaching the Apriltag at different angles, then the image of would be warped.

↳ do we have to account for this?

if so, how do we do it?

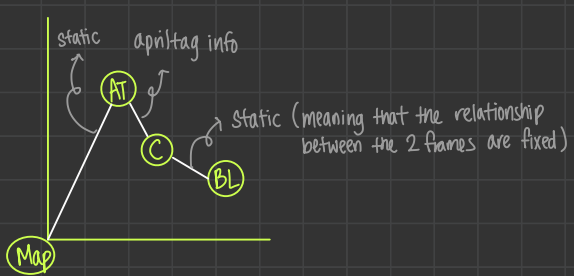
Paul's Advice

- maybe don't use Euler angles?
- extract the unit vectors
- the question is how do we map base_link → map?

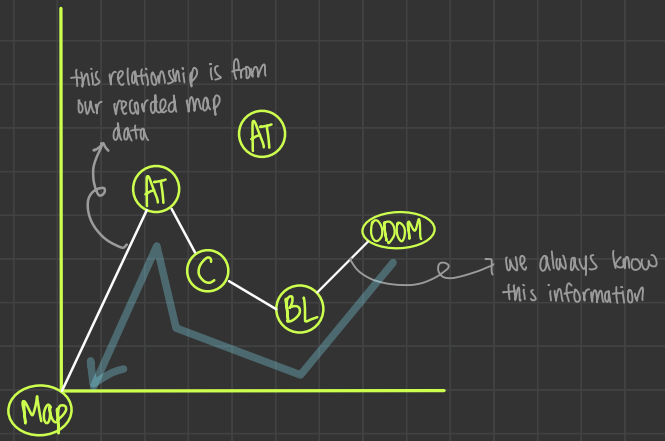


FRAMES

AT = Apriltag
C = Camera
BL = Base_link

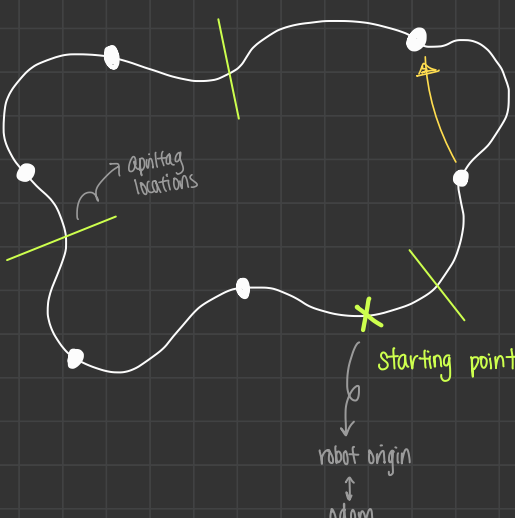


- 3x3 matrix calculation → use Numpy instead of ros2 library / PyKDL



Potential Plan

1. Record → localization X
2. Drive
   - Starting point in the map frame? origin?
   - transformation of keypoints?



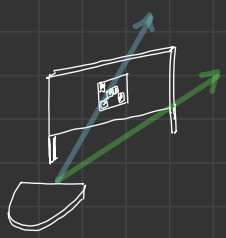
after moving a certain distance, save the pose

- At start: Base Link
- Odom → Base Link transformation

* when recording, save coords based on the "odom frame"

Question

Conversion of april_tag ⟷ Base_Link ?

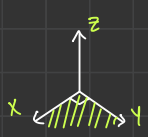


this currently returns a 2×3 matrix,

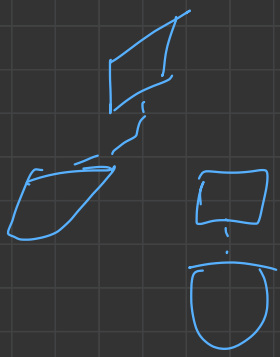
[dist_x , angle_x
dist_y , angle_y
dist_z , angle_z]

which is the ① rotation in each axis then
② {dist} movement in ①'s direction

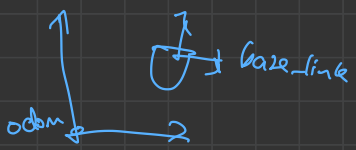
also, the april_tag and base link frames are flipped:

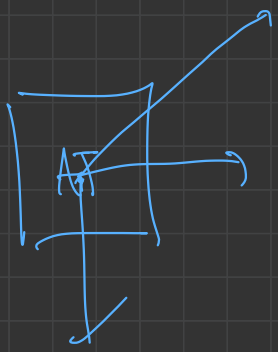


however, we are only interested in extracting (r, θ) from the above matrix, which would give us the translation and rotation from the Neato to the nearest AprilTag; converting this 3D transformation problem to a 2D transformation problem.



[map 만드는 코드.
[map기반으로 driving

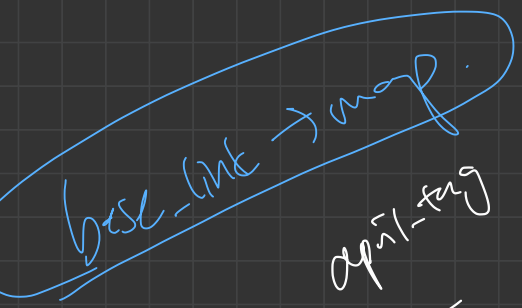


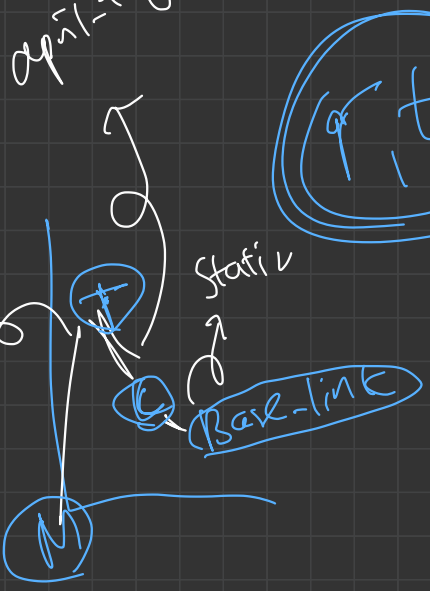


maybe do it use Euler angles

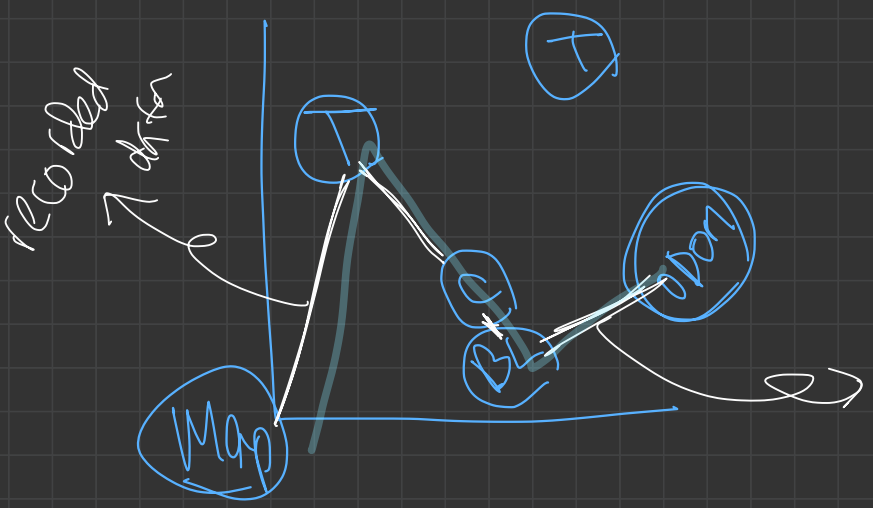
extract the unit vectors

base_link → map.





3차 → Numpy



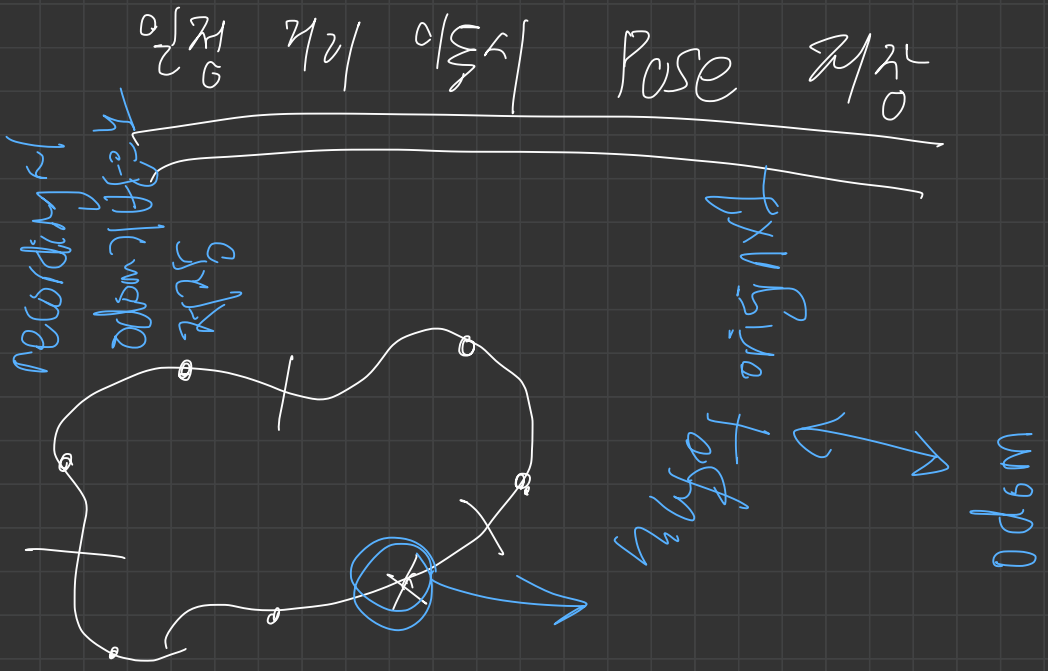
1. Record → localization X
2. Drive

시작 버튼

시작 위치 map frame

Origin

Transformation

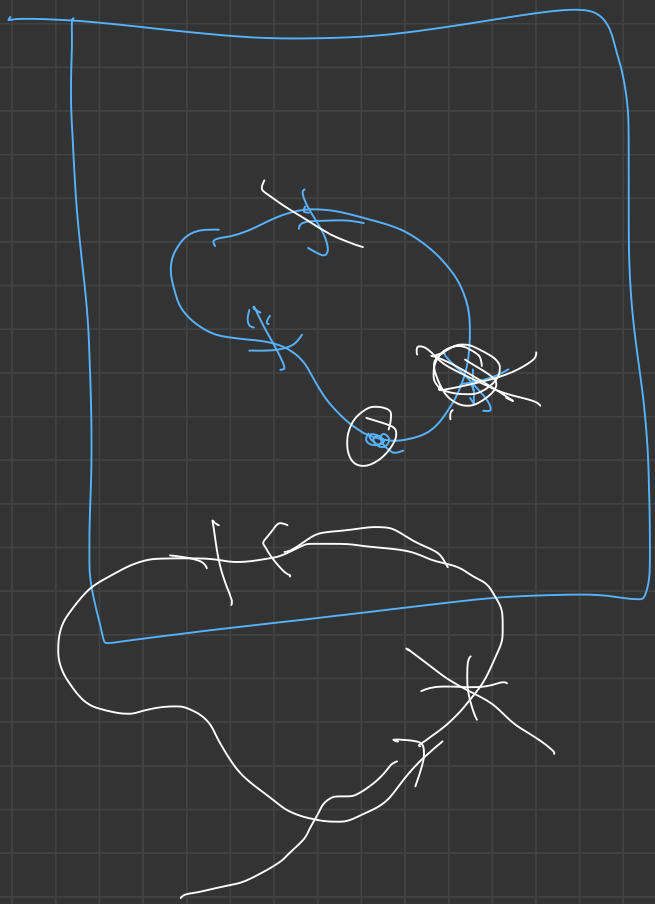


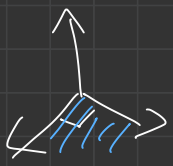
시작점 Base Link

Odom → base link

Transformation

Odom 기준 계산





apil_tag ↔ base_sink