



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](#)

공학석사 학위논문

사족보행 로봇을 위한
중력 기반 레이더 오도메트리

2023 년 8 월

서울대학교 대학원

기계공학부

정 상 우

사족보행 로봇을 위한 중력 기반 레이더 오도메트리

지도 교수 김 아 영

이 논문을 공학석사 학위논문으로 제출함
2023 년 4 월

서울대학교 대학원
기계공학부
정 상 우

정상우의 공학석사 학위논문을 인준함
2023 년 6 월

위 원 장 _____ 이 경 수 _____ (인)

부위원장 _____ 김 아 영 _____ (인)

위 원 _____ 조 규 진 _____ (인)

초 록

자율주행 및 로봇틱스 분야에서 로봇의 위치 측정과 주변 지도 생성을 동시에 진행하는 SLAM은 빼놓을 수 없는 연구 주제이다. 부정확한 센서 오도메트리는 SLAM 결과로 생성된 3차원 지도 또는 경로 계획 결과의 정확도에 영향을 미칠 수 있기에, 센서 오도메트리의 정확성은 SLAM 분야에서 매우 중요하다. 오도메트리의 정확도를 저하하는 주요한 문제 중 하나로는 부정확한 최적화 또는 잘못된 특징점 정합 등으로 인하여 발생하는 z축 에러 문제가 있다. IMU 센서를 활용하여 추정할 수 있는 중력 벡터를 활용할 경우, 기존의 카메라 및 라이다 SLAM에서 활용해온 4-Dof 최적화를 사용할 수 있게 된다. 이를 활용하여, 본 연구는 중력 추정 기반 4-Dof 최적화를 적용한 단일 칩 레이더 오도메트리 알고리즘을 제안한다. 제안하는 알고리즘은 Double RANSAC 테크닉을 활용하여 노이즈 및 동적 물체를 제거하였으며, 추가적으로 z축 방향으로 부정확한 포인트 또한 제거하였다. 또한 IMU 센서의 초기 가속도 정보를 기반으로 로봇의 Initial Orientation을 보정한다. 마지막으로 4-Dof 최적화에 기반한 위치 추정을 진행, 기존의 6-Dof 최적화를 기반으로 하는 칩 레이더 오도메트리 알고리즘에 비하여 z축 오차에 강인한 센서 오도메트리 정보를 계산한다. 본 알고리즘의 성능은 z축 진동이 심한 보행 로봇 플랫폼인 Boston Dynamics Spot 4족 보행 로봇에서 테스트하였으며, 실험 결과 기존 레이더 오도메트리 알고리즘과 비교하여 z축 방향 정확도가 향상된 것을 확인할 수 있다.

주요어 : 슬램, 레이더 오도메트리, 중력 추정, 최적화, 4족보행 로봇
학 번 : 2021-27880

목 차

| | |
|--|----|
| 제 1 장 서론..... | 1 |
| 제 1 절 연구의 배경..... | 1 |
| 제 2 절 연구의 내용..... | 3 |
| 제 2 장 배경 지식..... | 5 |
| 제 1 절 이론적 배경..... | 5 |
| 제 1 항 FMCW 레이더..... | 5 |
| 제 2 항 RANSAC..... | 6 |
| 제 2 절 선행 연구 조사..... | 7 |
| 제 1 항 칩 레이더 오도메트리..... | 7 |
| 제 2 항 중력 기반 4-Dof 최적화..... | 8 |
| 제 3 장 Double RANSAC 및 중력 기반 Orientation..... | 9 |
| 제 1 절 XY 평면 RANSAC..... | 9 |
| 제 2 절 XZ 평면 RANSAC..... | 10 |
| 제 3 절 중력 기반 Initial Orientation..... | 12 |
| 제 4 장 4-Dof 최적화 레이더 Factor..... | 14 |
| 제 1 절 레이더 속도 Factor..... | 14 |
| 제 2 절 Residual Errors and Jacobians..... | 16 |
| 제 5 장 실험 결과 및 분석..... | 17 |
| 제 1 절 RPM SPOT 센서 시스템..... | 17 |
| 제 2 절 실험 환경..... | 19 |
| 제 3 절 실험 결과 및 분석..... | 22 |
| 제 6 장 결론..... | 26 |
| 제 1 절 연구 내용 요약..... | 26 |
| 제 2 절 향후 연구 계획..... | 26 |
| 참고문헌..... | 27 |
| Abstract..... | 29 |
| Acknowledgements in Korean..... | 31 |

표 목차

| | |
|--------------|----|
| [표 5-1]..... | 22 |
| [표 5-2]..... | 22 |

그림 목차

| | |
|---------------|----|
| [그림 1] | 1 |
| [그림 2] | 3 |
| [그림 3] | 5 |
| [그림 4] | 6 |
| [그림 5] | 8 |
| [그림 6] | 9 |
| [그림 7] | 11 |
| [그림 8] | 13 |
| [그림 9] | 17 |
| [그림 10] | 18 |
| [그림 11] | 19 |
| [그림 12] | 20 |
| [그림 13] | 21 |
| [그림 14] | 22 |
| [그림 15] | 23 |
| [그림 16] | 23 |

제 1 장 서 론

제 1 절 연구의 배경

최근 자율주행 및 로봇틱스 기술이 화두가 되고 있는 만큼, 로봇 주변의 지도와 로봇의 위치를 동시에 추정하는 SLAM(Simultaneous Localization and Mapping) 기술의 중요성이 대두되고 있다. UGV(Unmanned Ground Vehicle)와 UAV(Unmanned Aerial Vehicle) 플랫폼을 위한 SLAM 알고리즘으로는 RGB 카메라를 활용한 Vision based SLAM, Range sensor의 일종인 LiDAR(Light Detection And Ranging) 또는 Radar(RADio Detection And Ranging)를 활용한 LiDAR based SLAM, Radar based SLAM 등이 활발히 연구되고 있으며, 수중 로봇의 경우 SONAR(Sound Navigation And Ranging)^① 센서를 활용한 SLAM 기술이 연구되고 있다.



[그림 1] Representative Range sensors for SLAM. First sensor is a spinning LiDAR from Velodyne. Second sensor is an imaging radar from Navtech. Last sensor is a chip radar from TI. Every sensor provides distance information around the sensor.

LiDAR(라이다) 센서는 레이저 빛을 발사하고, 주변 물체에 반사되어 돌아오는 시간차를 활용하여 주변을 탐지하는 센서이며, [그림 1]에서 그 예시 사진을 확인할 수 있다. 3차원 점군 맵으로 센서 데이터를 제공하며 각 포인트의 정밀도가 높다는 장점이 있다. 그러나

^① 초음파를 활용하여 수중에서 물체의 거리를 파악하는 센서

레이저 빛을 활용하는 만큼, 비, 안개 등의 날씨와 황사, 연기 등의 돌발적인 기상상황에 취약하다는 단점이 있다. 대표적인 라이다 센서를 활용한 SLAM 알고리즘으로는 Lego-loam[1], LIO-SAM[2], Fast-lio2[3], MA-LIO[4] 등이 있다.

Radar(레이더) 센서는 라디오 파를 활용하여 물체의 위치, 속도, 방향을 파악하는 센서이다. FMCW(Frequency-Modulated Continuous Wave) 칩 레이더의 경우 3차원 점군과 함께 각 포인트의 radial velocity^② 값 또한 제공한다. 라디오 파의 파장이 레이저보다 길다는 특성을 갖기 때문에 LiDAR 센서에 비하여 레이더 센서의 점군은 비, 안개, 황사, 연기 등의 기상 상황에 대한 영향에서 자유롭다는 장점을 갖는다. 비슷한 이유에서, 카메라 라이다 센서와 다르게 자동차의 범퍼 안쪽에 설치하여 사용할 경우 이물질로 인한 오염에 강인하다는 장점이 있다. 하지만 레이저 센서에서 취득한 점군은 라이다 센서 점군과 비교하여 포인트의 개수가 극히 적으며, 각 포인트의 위치 정보 정밀도(특히 z축 방향)와 해상도가 떨어진다는 단점이 공존한다. 타 센서의 경우 딥 러닝 및 머신 러닝 기술을 활용하여 측정된 객체의 종류, 크기 등의 디테일한 정보를 추정하는 연구가 활발히 진행되는 데에 반하여, 레이더 센서의 경우 물체의 유무만을 판정할 수 있는 수준이다.

레이더 센서는 크게 이미징 레이더(Imaging radar)와 칩 레이더(Chip radar)로 분류할 수 있으며, 각각 [그림 1]에서 그 예시 사진을 확인할 수 있다. 이미징 레이더는 센서 주변 정보를 2차원 데이터로 제공하며, 회전하는 구조를 가진 만큼 센서에서 멀어질수록 제공하는 데이터의 해상도가 떨어지는 희소성(Sparsity)를 갖는다. 대표적인 이미징 레이더를 활용한 SLAM 알고리즘으로는 [5], Pharaoh[6], RadarLoc[7] 등이 있다. 칩 레이더의 경우 radial velocity 정보를 포함한 4차원 점군 데이터를 제공하지만, 데이터의 해상도가 떨어지고 노이즈가 심하며, 특히 각 포인트의 z축 정밀도가 떨어진다는 단점이 있다. 이에 따라 칩레이더 센서를 활용하여 로봇의 3차원 위치(6 Dof^③ pose)를 추정할 경우 [8], [9]에서는 z축 정보를 배제한 칩 레이더 센서(2차원에 대하여 작동하는) 2개를 수직으로 부착하여 해당 문제점을 해결해왔다. 해당 방법의 경우 수학적으로 간단하게 3차원 ego-velocity^④를 측정할 수 있다는 장점이 있지만, 각 센서의 z축

② 센서와 물체의 일직선 방향으로의 속도

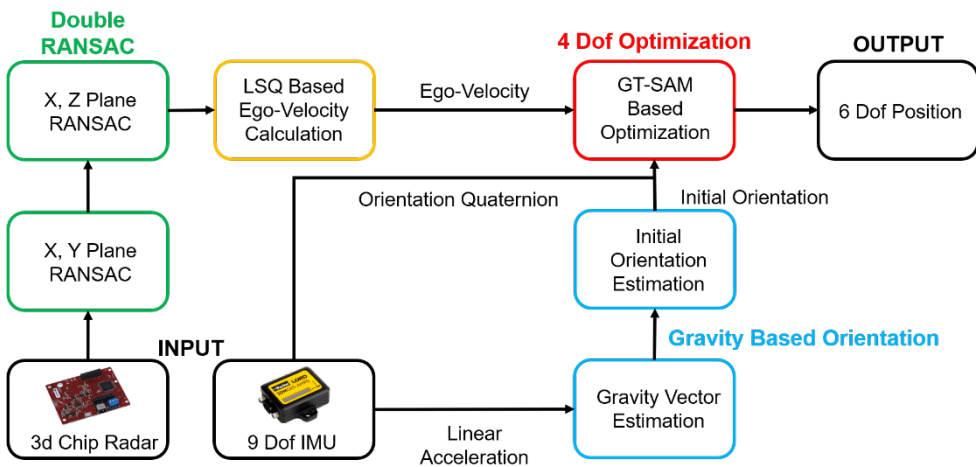
③ Degree of freedom (자유도)

④ 로봇 좌표계 기준의 로봇 속도

정보가 버려지고, 두 센서의 작동 타이밍을 맞추는 sensor synchronization 과정을 필수로 요구한다는 단점이 있다. 본 연구에서는 칩 레이더의 z축 정보를 정제하여 사용함으로써 단일 칩 레이더 오도메트리 알고리즘을 제시한다.

카메라 및 라이다 센서를 사용하는 Vins-mono[10], [11] 등의 기존의 SLAM 알고리즘의 경우 IMU(Inertial Measurement Unit) 센서를 활용하여 중력벡터를 추정, 위치 최적화(pose optimization) 단계에서 6 Dof(x, y, z, roll, pitch, yaw) 모두를 최적화하지 않고 roll과 pitch를 제외한 나머지 4 Dof만을 최적화함으로써 로봇 pose의 z축 정밀도를 높일 수 있다는 알고리즘을 사용해왔다. 이는 residual을 최소화하려는 최적화 과정에서 중력을 기반으로 계산되어 이미 신뢰할 수 있는 roll과 pitch를 고정하고 x, y, z, yaw 만을 최적화하는 아이디어로써, 본 연구에서는 해당 아이디어를 레이더 오도메트리의 최적화 단계에 도입하는 연구를 진행하였다.

제 2 절 연구의 내용



[그림 2] Pipeline of full system. Green boxes refer double RANSAC algorithm, blue boxes refer gravity based initial orientation estimation, while red box refers 4-Dof optimization.

본 연구의 내용은 크게 세 파트로 나눌 수 있다. 첫 번째 파트는 단일 칩 레이더 데이터에서 로봇의 Ego-Velocity를 계산하는 과정이다. 먼저, 칩 레이더에서 취득한 데이터에 Double RANSAC 기술을

적용한다. XY 평면에 대한 RANSAC을 먼저 적용하여 동적 물체(Dynamic Object)^⑤를 제거한다. 이후 XZ 평면에 대한 RANSAC을 적용하여 부정확한 Z축 정보를 갖는 포인트들을 제거한다. 이후 정확한 Z축 정보를 갖는 정적 물체(Static Object)^⑥ 데이터에 Least Square Optimization을 적용하여 로봇의 Ego-Velocity를 계산한다.

두 번째 파트는 IMU 센서 정보를 기반으로 중력 벡터를 추정하여 로봇의 Initial Orientation 방향을 추정하는 과정이다. 해당 과정을 통하여 Odometry 정보를 계산할 때 사용되는 Initial Orientation을 보정한다.

마지막 파트는 중력 방향에 기반하여 Roll, Pitch를 제외한 나머지 4 Dof에 대한 Optimization을 진행하는 과정으로, 기존의 6Dof Optimization 방법과 비교하여 z축 오차가 적은 odometry를 계산한다.

각 파트에 대한 보다 정확한 설명은 제 3 장 Double RANSAC 및 중력 기반 Orientation, 제 4 장 4-Dof 최적화 레이더 Factor에서 후술하였으며, 본 연구의 핵심 내용을 정리하면 다음과 같다.

- 칩 레이더 센서의 부정확한 z축 정보를 정제하여 사용하는 단일 칩 레이더 오도메트리 알고리즘을 제안하였다.
- 중력 방향을 기반으로 Roll과 Pitch를 제외한 4-Dof 최적화 기법을 레이더 오도메트리에 적용하여 Z축 정밀도를 향상시켰다.
- Z축 진동이 심한 사족보행 로봇 플랫폼을 기반으로 실험을 진행하였으며, 기존 레이더 오도메트리 알고리즘에 비하여 향상된 Z축 정밀도를 확인하였다.

^⑤ 글로벌 좌표계(Global Frame)을 기준으로 속도를 갖는 물체

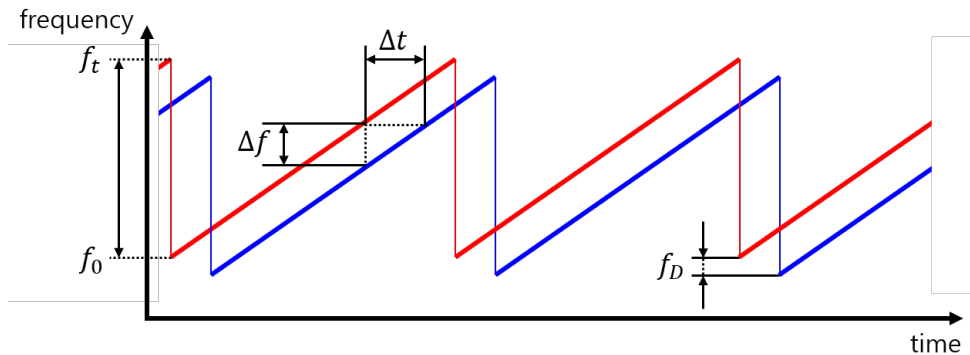
^⑥ 글로벌 좌표계(Global Frame)을 기준으로 고정되어 있는 물체

제 2 장 배경 지식

제 1 절 이론적 배경

제 1 항 FMCW 레이더

FMCW 레이더는 거리와 속도를 동시에 측정할 수 있는 레이더 센서로, 전송하는 신호(transmitted signal)의 주파수를 연속적으로 바꾸고(frequency-modulated continuous) 측정하는 신호(received signal)의 주파수를 같이 측정하는 기능을 가지고 있다. [그림 3]의 빨간 선과 파란 선에서 각 신호의 예시(Sawtooth shape)를 확인할 수 있다.



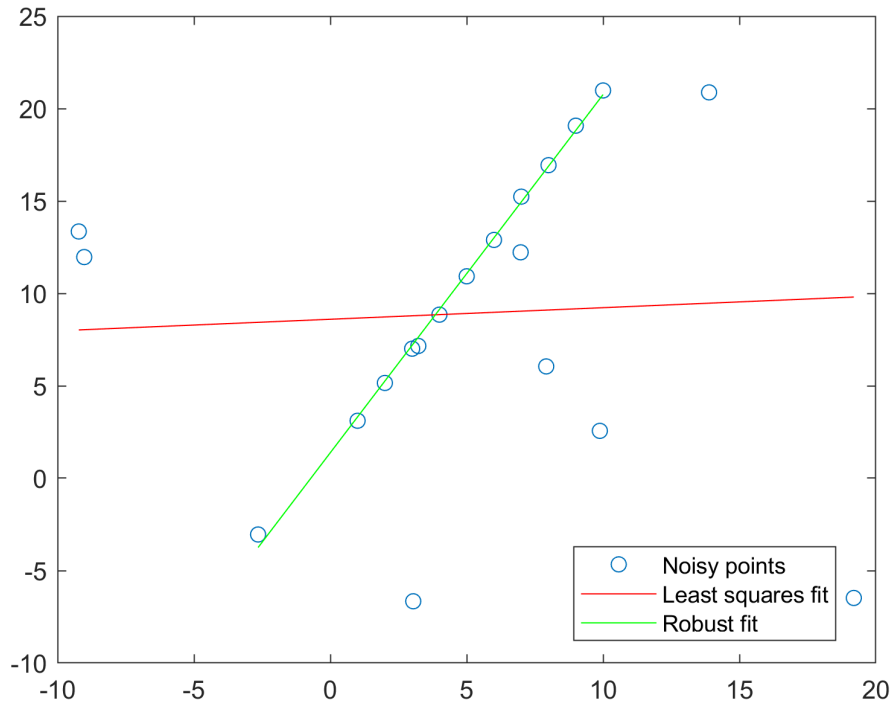
[그림 3] Shape of FMCW Signal (Sawtooth). Red line is the frequency of the transmitted signal and Blue line is the frequency of the received signal.

FMCW 레이더 시스템은 방출 신호와 측정 신호 사이의 주파수 차이([그림 3]의 Δf)를 사용하여 센서와 물체 사이의 거리를 측정한다. 또한, 센서와 물체 사이의 도플러^⑦ 주파수([그림 3]의 f_D)를 측정하여 해당 물체의 속도(radial velocity)를 계산한다. 따라서, 라이다 센서와 같은 타 range 센서에 비하여 활용 방안이 다양하다는 장점이 있다.

^⑦ Doppler shift: 움직이는 물체에서 파동을 방출하거나 측정할 경우 주파수 편이가 일어나는 현상

제 2 항 RANSAC

RANSAC(RANdOm SAmpLe Consensus)[12] 방법은 가장 많은 inlier를 갖는 모델을 선택하는 방법으로, 데이터셋에서 노이즈를 제거하고 해당 데이터를 잘 표현할 수 있는 모델을 예측하는 알고리즘이다.



[그림 4] Example of RANSAC. Result of Line fitting is much robust on RANSAC compared with the Least square fitting on noisy points including many outliers.

RANSAC 방법은 전체 데이터에서 n 개의 샘플을 선택한 뒤, 해당 샘플을 사용하여 모델(Line, Plane, 등)을 예측한다. 이후 예측한 모델에 기반하여 전체 데이터셋 중 모델과 일치하는 (오차가 임계값 이하인) 데이터의 개수를 센 뒤, 그 개수가 최대 값일 경우 모델을 새롭게 저장한다. 위의 과정을 반복하여 최대의 inlier를 갖는 모델을 계산하는 알고리즘이며, outlier에 강인하다는 장점을 가진다. [그림 4]의 Line fitting 과정에서, 빨간색의 least square fitting 결과 비하여 초록색의 RANSAC fitting 결과 그래프는 노이즈 outlier에 강인하게 line 모델을 예측한 것을 확인할 수 있다.

제 2 절 선행 연구 조사

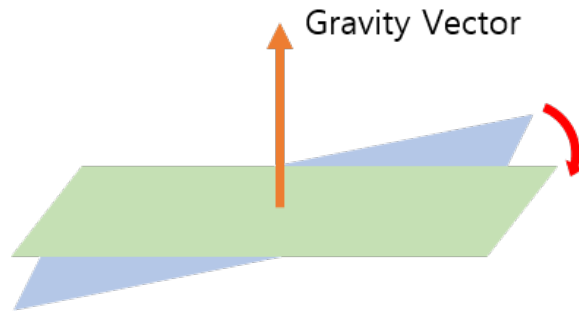
제 1 항 칩 레이더 오도메트리

칩 레이더 센서는 제공하는 점군의 3차원 위치와 더불어 radial velocity 또한 제공한다는 특징이 있다. 속도 정보를 활용함으로써, 취득한 데이터로부터 동적 물체에 해당하는 점군 제거 및 ego-motion을 간단하게 계산할 수 있다. [13]은 RANSAC 기법을 활용하여 칩 레이더 데이터에서 동적 물체 및 노이즈를 제거한 뒤, 정적 물체와 그 속도 정보에 기반하여 로봇의 2D Ego-Motion을 예측하는 방법을 제안하였다. [13]을 확장하여, [8]은 2개 이상의 칩 레이더 센서 데이터를 활용하여 로봇의 Ego-Motion을 예측하였다. [13]과 동일하게 RANSAC 기법으로 정적 물체를 추출한 뒤, Least Square 최적화를 기반으로 Ego-Motion을 계산하였다. [9]는 두 개의 칩 레이더 센서를 수직으로 (정면을 바라보는 센서 하나, 바닥을 바라보는 센서 하나) 부착한 뒤 [8]의 방법으로 로봇의 Ego-Motion을 계산, 이후 Factor Graph에 ego-velocity를 도입하여 3차원 pose를 최적화하는 알고리즘을 제안하였다. 본 연구는 칩 레이더에 기반한 오도메트리 알고리즘 중 sota(state-of-the-art)에 해당하는 [9]의 알고리즘에 기반하되, 하나의 칩 레이더를 활용하여 3차원 pose를 계산하기 위해 XY 평면과 XZ 평면에 각각 RANSAC을 진행했다는 차이점이 있다.

RANSAC 기반의 방법을 활용하지 않고 칩 레이더를 활용한 오도메트리 알고리즘을 연구한 결과 또한 존재한다. Milli-RIO[14]는 칩 레이더 센서에서 얻은 점군 데이터에 NDT(Normal Distributions Transform)[15] 방법을 적용, 로봇의 움직임을 계산하였다. [16]은 앞선 K개의 레이더 데이터를 쌓은 뒤 타겟을 추출, 타겟의 도플러 속도를 활용한 cost function에 기반하여 ego-velocity를 계산하였다. 하지만 해당 과정에서 모든 데이터가 정적 물체임을 가정하였으며 이후 최적화 과정에서 Linear velocity와 Angular velocity를 loosely coupled 상태로 최적화를 진행했다는 한계를 보였다. milliEgo[17]의 경우, 점군 형태로 제공되는 칩 레이더의 데이터를 깊이 정보를 포함한 2D 이미지 형태로 변환한 뒤, IMU 데이터와 같이 deep attention network 모델에 입력하여 로봇의 위치를 계산하였다.

제 2 항 중력 기반 4-Dof 최적화

IMU 센서를 통하여 가속도 정보를 얻을 경우, 중력 벡터의 방향을 계산할 수 있다. 해당 벡터 정보는 3차원 글로벌 좌표계 중, Roll과 Pitch를 IMU를 활용하여 정확히 측정할 수 있게 해주며, 이와 동시에 로봇의 위치 추정 문제를 6 Dof 문제에서 4 Dof (Roll과 Pitch를 제외한) 문제로 바꾼다[18].



[그림 5] Due to the known gravity vector, roll and pitch angles are observed absolutely.

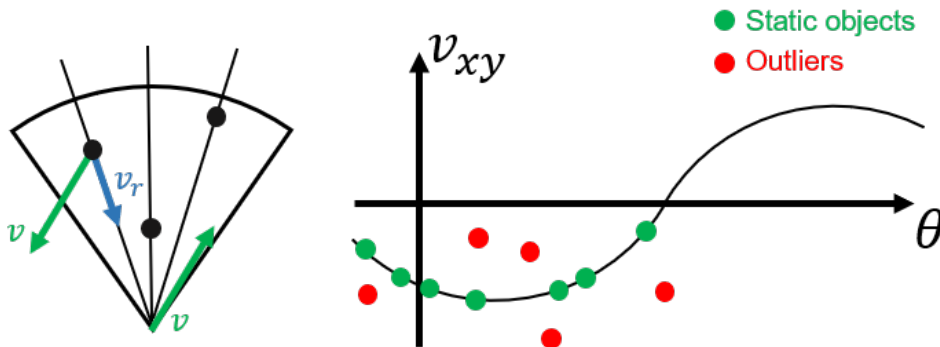
카메라 센서를 활용한 SLAM 중 대표적으로 해당 알고리즘을 차용한 경우로는 Vins-mono[10]가 있으며, x , y , z , yaw 값만을 최적화하여 로봇의 오도메트리를 계산하여, 부정확한 roll과 pitch의 최적화로 인해 발생하고 축적되는 z 축 오차를 제거했다. [11]은 위의 아이디어를 라이다 슬램의 ICP(Iterative Closest Point)[19]에 적용하여, roll과 pitch를 제외한 4 Dof에 대한 ICP를 진행하는 4-Dof ICP를 제안했으며 기존 라이다 SLAM 알고리즘에 비하여 실내외 모두에서 z 축방향으로 적은 오차를 갖는 점군 지도를 생성했다.

본 연구에서는 해당 4-Dof 최적화 아이디어를 [9]의 알고리즘에 적용, 단일 칩 레이더 사용 및 4족 보행 로봇 플랫폼으로 인하여 발생하는 z 축 오차 문제를 해결하였다.

제 3 장 Double RANSAC 및 중력 기반 Orientation

제 1 절 XY 평면 RANSAC

칩 레이더에서 얻을 수 있는 3차원 점군 데이터는 radial velocity를 제공하지만 z축 정밀도가 떨어지고, 노이즈가 심하다는 단점이 있다. 본 연구에서는 3차원 정보를 모두 활용함과 동시에 효과적으로 동적 물체를 제거, 그리고 z축 오차가 심한 포인트를 제거하기 위하여 두 번의 RANSAC 과정을 거쳤다.



[그림 6] Left Picture: Green arrow v denotes the velocity vector of the static object about the sensor(which is the negative of the ego-velocity vector of the sensor). Blue arrow v_r denotes the radial velocity of the static object. Right Picture: Green points denote the radial velocity of static objects which follow the sinusoidal curve. Red points denote the dynamic objects and noise(outliers) that does not follow the sinusoidal curve.

첫 번째 RANSAC 과정은 [8, 9, 13]의 RANSAC을 활용한 동적 물체 및 노이즈 제거 과정과 동일한 방법으로 진행한다. 먼저 3차원 포인트의 위치 및 radial velocity 벡터를 XY 평면에 정사영^⑧을 시킴으로써 위치 정보는 (x,y) , 속도 벡터는 v_{xy} 만을 남긴다. 이후, 각 포인트의 위치 정보를 (x,y) 의 직교좌표에서 (r,θ) 의 극좌표로 변환한다.

^⑧ 포인트 및 벡터의 끝에서 평면에 내린 수선의 발이 그리는 도형

만약 특정 포인트가 정적 물체일 경우, 칩 레이더 센서에서 측정된 2차원 radial velocity(v_{xy})는 로봇의 해당 포인트 방향(θ 방향) 속도와 동일해야 한다. 따라서, n 개의 모든 포인트가 정적 물체이고, v_x 와 v_y 가 각각 로봇의 x 축 방향 속도, y 축 방향 속도일 때 아래의 식이 성립하게 된다.

$$\begin{bmatrix} v_{xy,1} \\ \vdots \\ v_{xy,n} \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ \vdots & \vdots \\ \cos\theta_n & \sin\theta_n \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (3.1)$$

[8]에서 언급된 바와 같이, 각 포인트의 θ 와 v_{xy} 의 그래프를 그릴 경우, [그림 6]의 우측 그래프와 같이 정적 물체를 나타내는 포인트들만이 sinusoidal curve를 따르게 된다. 이에 따라 sinusoidal curve를 fitting하는 모델을 찾는 RANSAC 기법을 활용할 경우 로봇의 2D 속도(v_x, v_y) 및 정적 물체 점군들을 추출할 수 있다.

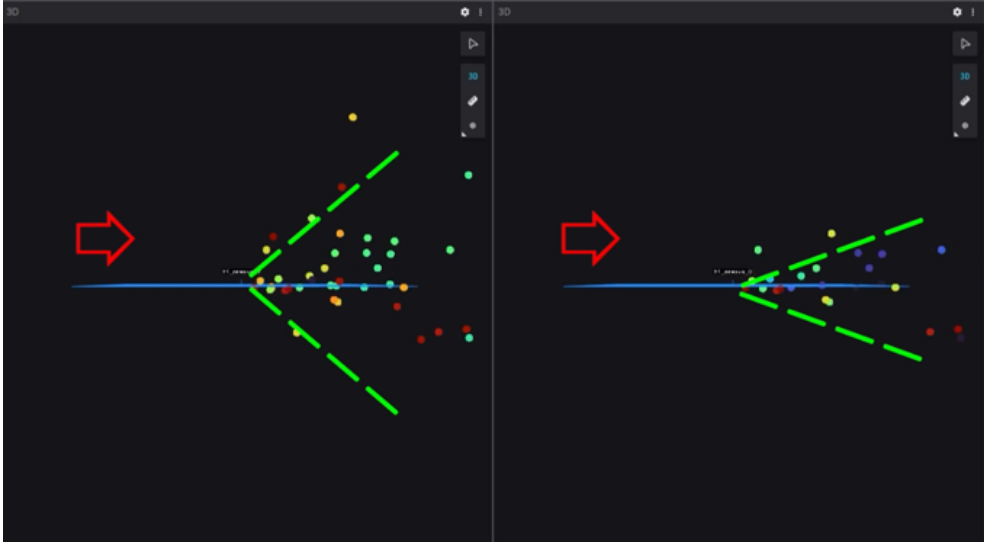
제 2 절 XZ 평면 RANSAC

앞선 XY 평면 RANSAC을 통하여 얻은 정적 물체 포인트들의 경우, 노이즈 및 동적 물체가 제거된 점군 데이터이다. 이 때, 칩 레이더 센서의 특성상 정적 물체를 나타내는 포인트들 임에도 불구하고, 각 포인트의 z 축 위치 데이터 정밀도가 떨어지게 된다. 따라서, XZ 평면에 대하여 RANSAC을 한 번 더 진행하여, 부정확한 z 축 정보를 포함하고 있는 점군을 다시 제거해야 한다.

먼저, 앞선 XY 평면 RANSAC에서 정적 물체의 포인트로 판별된 점군의 3차원 포인트들을 XZ 평면에 정사영 시킨다. 이후, 앞선 XY 평면 RANSAC과 비슷하게 각 포인트의 위치 정보를 (x, z) 의 직교좌표에서 (r, θ) 의 극좌표로 변환한다. 마찬가지로, z 축 방향 정보의 정밀도가 높은 포인트들의 경우, θ 와 2차원 radial velocity(v_{xz})의 그래프가 sinusoidal curve를 따라야 한다. 따라서 (3.1)과 비슷한 형태를 가지는 아래의 식이 성립하게 된다. 정적 물체를 표현하는 포인트의 개수를 n 개로 정의했으며, (3.1)과 흡사하게 로봇의 x 축 방향 속도는 v_x , z 축 방향 속도는 v_z 로 표기했다.

$$\begin{bmatrix} v_{xz,1} \\ \vdots \\ v_{xz,m} \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ \vdots & \vdots \\ \cos\theta_m & \sin\theta_m \end{bmatrix} \begin{bmatrix} v_x \\ v_z \end{bmatrix} \quad (3.2)$$

또한, 앞선 XY 평면 RANSAC을 통하여 얻을 수 있는 로봇의 속도 정보 중 x축 방향에 해당하는 v_x 의 경우, 노이즈와 동적 물체가 제거된 정적 물체들을 기준으로 하는 x, y 축 속도 정보만으로 계산된 결과이므로 신뢰도가 높다. 따라서, XZ 평면을 기준으로 RANSAC을 진행할 때 앞서 계산한 v_x 값을 고정한 상태로 진행할 경우 z축 정밀도가 높은 정적 물체 포인트를 검출할 수 있다.



[그림 7] Result after Double RANSAC. Left picture includes the side view of full points from the chip Radar sensor. Right picture includes the side view of inlier points after the Double RANSAC process. As the data is acquired indoors, the points above ceiling can be treated as outliers including z axis error. In the right picture, every point is including reasonable z axis value.

최종적으로 검출된 정적 물체에 해당하며 z축 정밀도가 높은 포인트들을 기반으로 이하의 수식을 세울 수 있으며, 이를 활용하여 로봇의 3차원 ego-velocity를 계산할 수 있다. 이때, 최종 inlier 포인트의 개수를 n 개로 정의하며, r_k 는 k 번째 포인트와 센서의 거리($\sqrt{(x_k)^2 + (y_k)^2 + (z_k)^2}$)를 의미한다.

$$\begin{bmatrix} v_{r,1} \\ \vdots \\ v_{r,n} \end{bmatrix} = \begin{bmatrix} \frac{x_1}{r_1} & \frac{y_1}{r_1} & \frac{z_1}{r_1} \\ \vdots & \vdots & \vdots \\ \frac{x_n}{r_n} & \frac{y_n}{r_n} & \frac{z_n}{r_n} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (3.3)$$

위의 식을 기반으로 least square optimization을 수행하여 로봇의 3차원 ego-velocity를 계산할 수 있다.

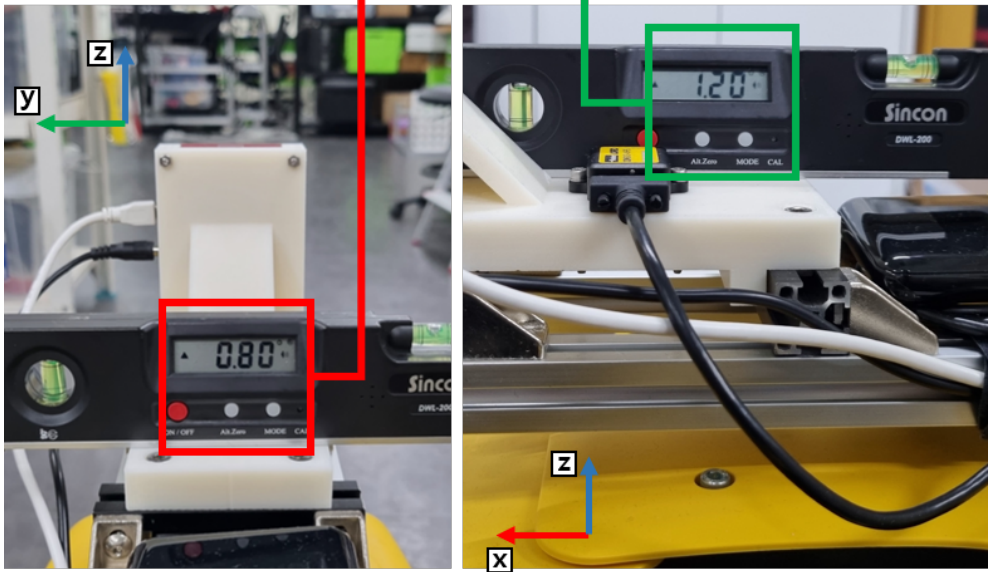
제 3 절 중력 기반 Initial Orientation

후술할 제 4 장 4-Dof 최적화 레이더 Factor의 기본 가정 조건이 “IMU 센서의 가속도 벡터를 기반으로 계산한 중력 벡터를 활용하여 Roll, Pitch의 값을 측정할 수 있으며, 이 둘을 제외한 나머지 4 Dof에 대한 최적화를 진행하여 오도메트리의 z축 오차를 줄일 수 있다” 이므로, 로봇의 초기 orientation과 중력 벡터의 차이를 보정해주는 과정(로봇의 초기 orientation을 중력 벡터 방향으로 보정하는 과정)은 필수적이다. 또한, IMU 센서는 로봇의 relative orientation(초기 orientation에 대한)만을 제공하므로, 시스템의 공차 및 주변 환경의 비틀어짐을 고려할 때, 해당 오차를 보정하기 위하여 로봇의 초기 orientation을 중력 가속도 기반으로 계산하는 과정은 필수적이다.

본 연구에서는 데이터 취득 과정에서 로봇이 움직이기 이전부터 센서 데이터 로깅을 미리 시작하여, 멈춰 있는 로봇의 IMU 센서 데이터를 1초간 취득하였다. 해당 IMU의 가속도 데이터(움직임이 없으므로 중력 가속도 벡터만을 포함) 중, 9.75~9.85 m/s²^⑨ 사이의 가속도 벡터의 평균을 계산, 중력 가속도를 구하였다. 계산된 중력 가속도 벡터와 z축 방향으로 수직하게 작용하는 실제 중력 가속도 벡터 사이의 orientation 차이를 계산하여 로봇의 초기 orientation을 보정하였다(글로벌 좌표의 z축이 중력 가속도 벡터의 방향과 동일하게 보정하였다). [그림 8]을 통하여 로봇의 공차 및 주변 환경 등으로 인하여 발생한 초기 orientation의 틀어진 방향 및 값을 추정된 결과물 기울기계로 측정한 값과 비교하여 확인할 수 있다.

^⑨ 중력 가속도의 ±0.05 m/s²로 threshold를 설정.

```
init_cnt: 100
init done
acc_avg: -0.221676  0.101251  9.80406
gravity_orien: [
    0.999744, 0.000233437, 0.0226036;
    0, 0.999947, -0.0103269;
    -0.0226048, 0.0103242, 0.999691
]
x, y degs: 0.591697, 1.29527
```



[그림 8] Upper picture: Result of gravity vector estimation based on the static IMU data. Tilted orientation about x axis and y axis (Roll and Pitch) are depicted. Lower left picture: Tilt meter on the x axis. Lower right picture: Tilt meter on the y axis (highly related with the z axis error on odometry result).

제 4 장 4-Dof 최적화 레이더 Factor

제 1 절 레이더 속도 Factor

앞선 내용과 같이, [10, 11] 등, 기존의 카메라 및 라이다 SLAM 분야에서는 중력 가속도 벡터의 방향은 일정하다는 점을 활용, Roll과 Pitch를 제외한 나머지 4 Dof(x, y, z, yaw)에 대한 최적화만을 진행함으로써 SLAM의 결과 오도메트리 및 지도의 z축 정밀도를 향상시켰다. 본 연구에서는 해당 방법론을 [9]에 포함된 레이더 factor에 적용함으로써 보다 높은 z축 정밀도의 위치 최적화를 진행하였다. 구현상의 차이점으로는, orientation 기준 yaw 만을 최적화하므로, IMU 센서가 측정한 orientation을 상수로 고정해두고, yaw 방향의 최적화를 위한 회전 행렬을 새롭게 정의하였다는 점이다.

i^{th} frame의 orientation matrix을 \mathbf{R}_i , i^{th} frame의 IMU orientation matrix 데이터를 $\mathbf{R}_{i,l}$, 그리고 i^{th} frame 에서 yaw를 보정하기 위한 orientation matrix를 $\mathbf{R}_{i,opt}$ 라고 정의할 경우, 이하의 식이 성립하게 된다.

$$\mathbf{R}_i = \mathbf{R}_{i,opt} \mathbf{R}_{i,l} \quad (4.1)$$

i^{th} frame 의 yaw를 보정하는 변수를 γ_i 로 정의할 경우, $\mathbf{R}_{i,opt}$ matrix와 해당 matrix의 Jacobian은 다음과 같이 정의된다. 3차원 기준 z축 방향으로 γ_i 만큼 회전시키는 회전 행렬과 동일한 형태를 가지게 된다.

$$\mathbf{R}_{i,opt} = R(\gamma_i) = \begin{bmatrix} \cos(\gamma_i) & -\sin(\gamma_i) & 0 \\ \sin(\gamma_i) & \cos(\gamma_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$\frac{\partial \mathbf{R}_{i,opt}}{\partial \gamma_i} = \frac{\partial R(\gamma_i)}{\partial \gamma_i} = \begin{bmatrix} -\sin(\gamma_i) & -\cos(\gamma_i) & 0 \\ \cos(\gamma_i) & -\sin(\gamma_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \dot{\mathbf{R}}_{i,opt} \quad (4.3)$$

[9]의 레이더 factor를 구성하는 residual과 Jacobian 식은 각각 후술할 (4.4~7)과 같다.

$$\mathbf{r}_{\Delta \mathbf{p}_{ij}} = \mathbf{R}_i^T (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t) - \left[\Delta \tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_j^r) + \frac{\partial \Delta \tilde{\mathbf{p}}_{ij}}{\partial \mathbf{b}^r} \delta \mathbf{b}_j^r \right] \quad (4.4)$$

$$\mathbf{r}_{\mathbf{v}_j} = \mathbf{v}_j - \left[\tilde{\mathbf{v}}_j(\bar{\mathbf{b}}_j^r) + \frac{\partial \tilde{\mathbf{v}}_j}{\partial \mathbf{b}^r} \delta \mathbf{b}_j^r \right] \quad (4.5)$$

$$\|\mathbf{r}_{\Delta \mathbf{b}_{ij}^r}\|^2 = \|\mathbf{b}_j^r - \mathbf{b}_i^r\|^2 \quad (4.6)$$

$$\begin{aligned} & \begin{pmatrix} \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \phi_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{p}_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{v}_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{b}_i^r} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \phi_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{p}_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{v}_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{b}_j^r} \\ \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \phi_i} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{p}_i} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{v}_i} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{b}_i^r} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \phi_j} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{p}_j} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{v}_j} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{b}_j^r} \\ \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \phi_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{p}_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{v}_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{b}_i^r} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \phi_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{p}_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{v}_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{b}_j^r} \end{pmatrix} & (4.7) \\ & = \begin{pmatrix} \left(\mathbf{R}_i^T (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_j \Delta t) \right)^\wedge & -\mathbf{I}_{3 \times 3} & -\mathbf{R}_i^T \Delta t & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{R}_i^T \mathbf{R}_j & \mathbf{0}_{3 \times 3} & \frac{1}{2} \mathbf{R}_i^T \mathbf{R}_j \Delta t \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{R}_j \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{pmatrix} \end{aligned}$$

(4.4~6)은 [9]에 포함된 레이더 factor의 residual 값이며, (4.7)은 해당 factor를 활용한 factor graph를 최적화하기 위하여 사용되는 Jacobian이다. (4.4~6)은 각각 i^{th} frame 과 j^{th} frame 사이의 pose difference에 대한 residual, i^{th} frame 에서의 속도에 대한 residual, 그리고 i^{th} frame 과 j^{th} frame 사이의 bias 변화에 대한 residual이다. 각 벡터와 행렬, bias의 정의 및 변수 표기는 [9]의 정의와 동일하다.

제 2 절 Residual Errors and Jacobians

(4.4~7)으로 구성된 [9]의 factor는 x, y, z, roll, pitch, yaw 모두를 한 번에 optimize하는 6-Dof optimize를 위하여 디자인되었다. 본 연구에서는 (4.1~3)의 수식에 기반하여 4-Dof optimize(x, y, z, yaw)를 위한 새로운 residual 및 Jacobian 식을 아래의 수식과 같이 정의하였다.

$$\mathbf{r}_{\Delta \mathbf{p}_{ij}} = \mathbf{R}_{i,l}^T \mathbf{R}^T(\gamma) (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t) - \left[\Delta \tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_j^r) + \frac{\partial \Delta \tilde{\mathbf{p}}_{ij}}{\partial \mathbf{b}^r} \delta \mathbf{b}_j^r \right] \quad (4.8)$$

$$\mathbf{r}_{\mathbf{v}_j} = \mathbf{v}_j - \left[\tilde{\mathbf{v}}_j(\bar{\mathbf{b}}_j^r) + \frac{\partial \tilde{\mathbf{v}}_j}{\partial \mathbf{b}^r} \delta \mathbf{b}_j^r \right] \quad (4.9)$$

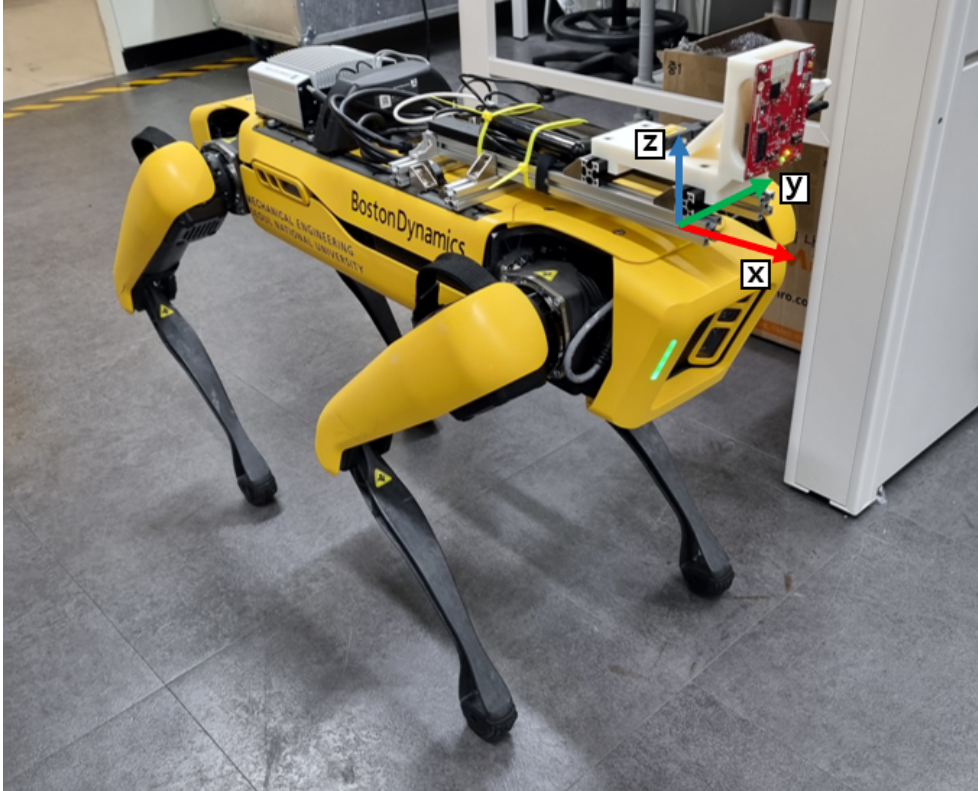
$$\left\| \mathbf{r}_{\Delta \mathbf{b}_{ij}^r} \right\|^2 = \left\| \mathbf{b}_j^r - \mathbf{b}_i^r \right\|^2 \quad (4.10)$$

$$\begin{aligned} & \begin{pmatrix} \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \phi_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{p}_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{v}_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{b}_i^r} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \phi_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{p}_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{v}_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{p}_{ij}}}{\partial \mathbf{b}_j^r} \\ \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \phi_i} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{p}_i} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{v}_i} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{b}_i^r} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \phi_j} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{p}_j} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{v}_j} & \frac{\partial \mathbf{r}_{\mathbf{v}_{ij}}}{\partial \mathbf{b}_j^r} \\ \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \phi_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{p}_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{v}_i} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{b}_i^r} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \phi_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{p}_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{v}_j} & \frac{\partial \mathbf{r}_{\Delta \mathbf{b}_{ij}^r}}{\partial \mathbf{b}_j^r} \end{pmatrix} \\ & = \begin{pmatrix} \left(\mathbf{R}_i^T \mathbf{R}_{i,opt}^T (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_j \Delta t) \right)^\wedge & -\mathbf{I}_{3 \times 3} & -\mathbf{R}_i^T \mathbf{R}_{i,opt}^T \Delta t & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{R}_i^T \mathbf{R}_{i,opt}^T \mathbf{R}_{j,opt} \mathbf{R}_j & \mathbf{0}_{3 \times 3} & \frac{1}{2} \mathbf{R}_i^T \mathbf{R}_{i,opt}^T \mathbf{R}_{j,opt} \mathbf{R}_j \Delta t \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \mathbf{R}_j \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \mathbf{I}_{3 \times 3} \end{pmatrix} \end{aligned} \quad (4.11)$$

[9]의 residual 및 Jacobian에 해당하는 6 Dof 기반의 (4.4~7) 수식과 본 연구의 residual 및 Jacobian인 (4.8~11)을 각각 비교할 경우, residual의 경우 Pose에 해당하는 4.4와 4.8만이 바뀌었으며, 해당 residual에 해당하는 Jacobian 행렬의 항들이 같이 바뀐 것을 확인할 수 있다. yaw 방향의 optimize를 위한 γ 변수와 IMU로부터 얻은 로봇의 orientation 값인 $\mathbf{R}_{i,l}$ 행렬이 기존의 \mathbf{R}_i 변수를 대체하고 있다는 점 또한 확인할 수 있다.

제 5 장 실험 결과 및 분석

제 1 절 RPM SPOT 센서 시스템

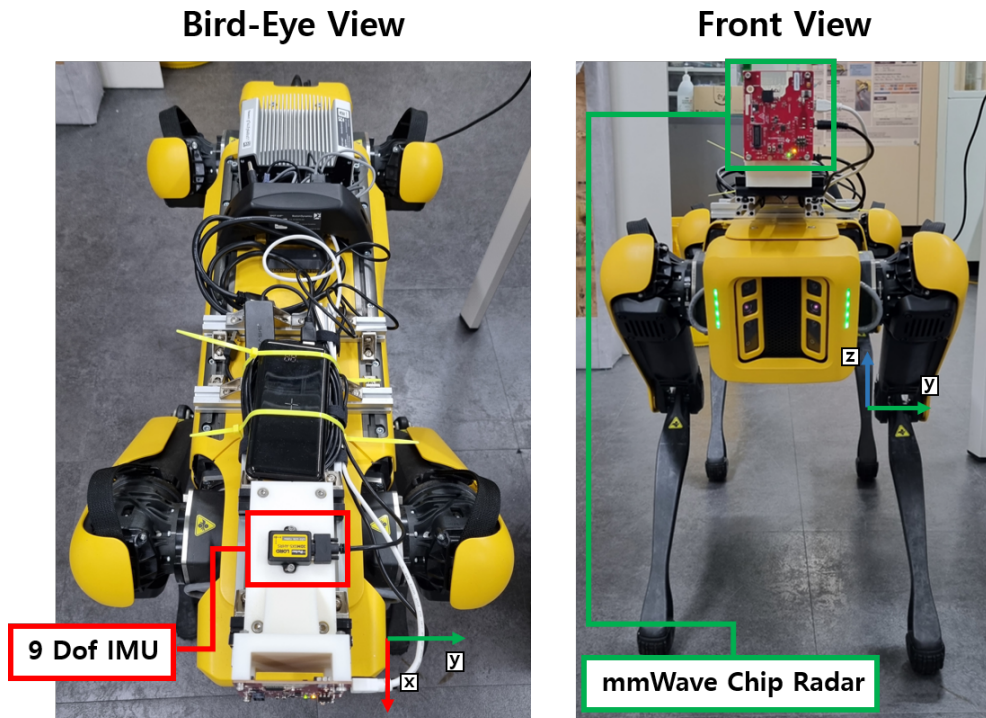


[그림 9] Overview of RPM SPOT Sensor System

본 연구의 알고리즘을 평가하기 위한 실험은 Boston Dynamics 사의 SPOT 4족보행 로봇을 플랫폼으로 활용하는 RPM SPOT 센서 시스템이 활용되었다. [그림 9]에서 전체적인 센서 시스템의 사진 및 x, y, z 축 방향을 확인할 수 있으며, IMU 및 칩 레이더 센서의 축 방향은 동일하다. SPOT 4족보행 로봇의 경우, 기존의 Wheeled UGV에 비하여 계단과 같은 환경에 제약되지 않고 움직일 수 있지만, 에너지 효율이 떨어지고 각 다리가 땅과 접촉할 때 생기는 contact impact로 인하여 z축 진동이 심하다는 단점이 있다.

칩 레이더 센서의 경우 로봇의 상단에 전면을 바라보도록 설치되어 있으며, TI mmwave의 AWR1843-Boost 제품을 사용하였다. 12개의 안테나로 신호 송수신을 진행하며, 20hz의 속도로 작동하는 제품이다.

또한, 0~14m의 탐지 범위를 가지며, 수평방향 $\pm 60^\circ$, 수직방향 $\pm 15^\circ$ 의 fov(Field Of View)를 가진다. 칩 레이더 센서의 자세한 사진은 [그림 10]의 우측 사진에서 확인할 수 있다. IMU 센서는 Microstrain의 3dm-gx5-25 제품을 사용하였다. 해당 제품은 9축 정보 (x, y, z축 방향의 linear acceleration, roll, pitch, yaw 방향의 angular velocity, 지자기장을 기반으로 하는 3차원 orientation)를 제공하며, 100 hz로 작동하는 상태로 실험을 진행했다. IMU 센서의 보다 자세한 사진은 [그림 10]의 좌측 사진에서 확인할 수 있다.

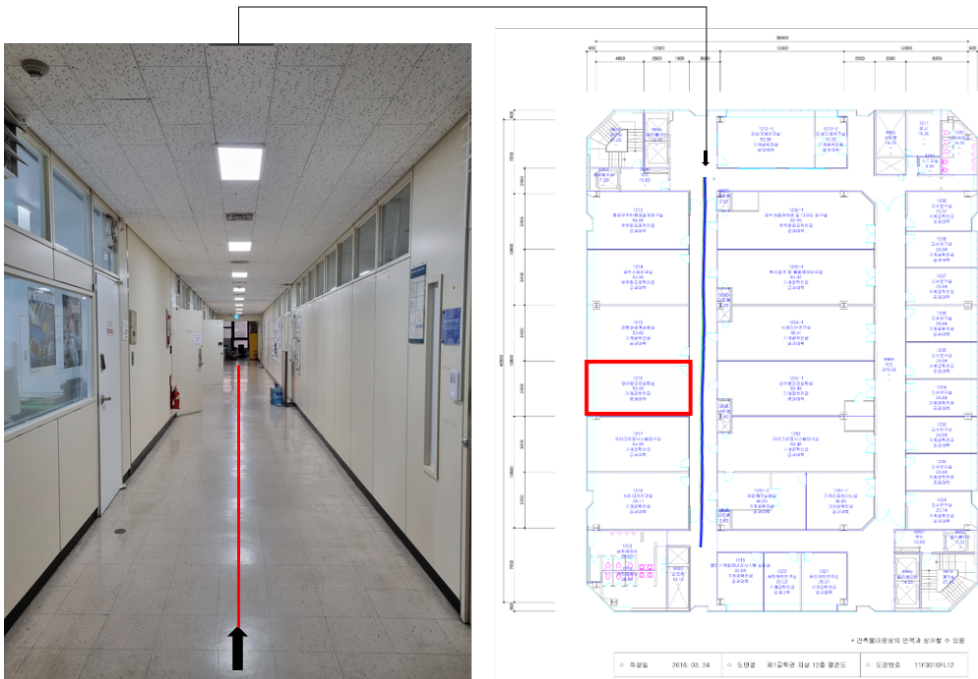


[그림 10] Bird-Eye View and Front View of RPM SPOT Sensor System

제 2 절 실험 환경

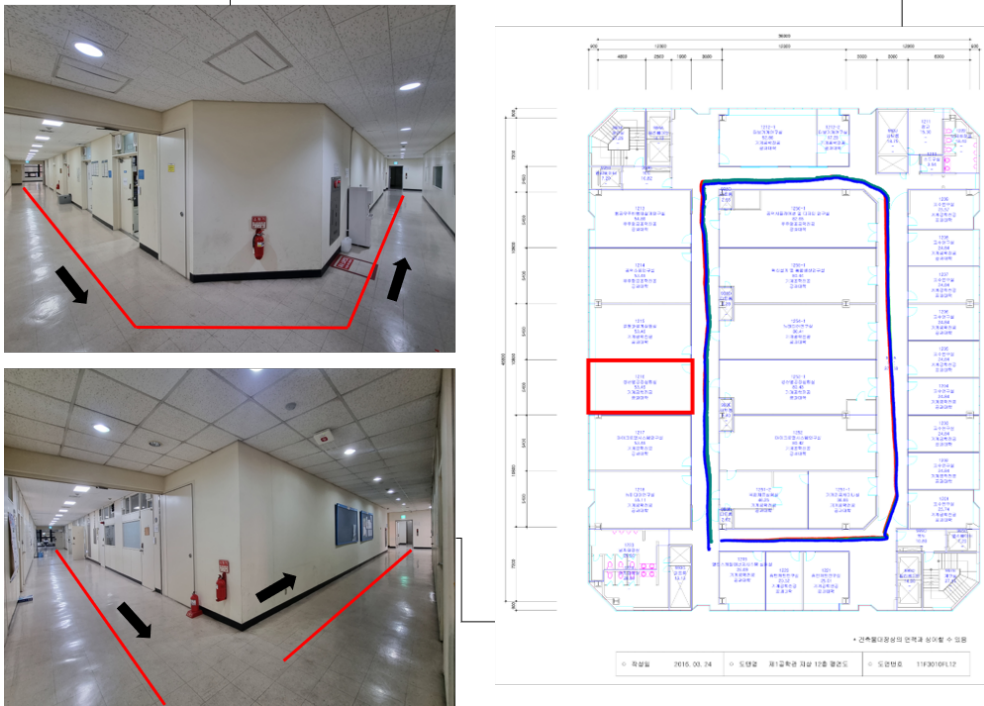
실험은 서울대학교 301동 12층 복도 및 1~2층에서 진행되었으며, 실험을 실내에서 진행한 만큼 오도메트리 참값(Ground Truth)을 구할 수 있는 GPS 등의 센서를 사용하지 못하므로, 같은 층의 z축 pos는 같아야 한다는 가정 하에 각 오도메트리의 z축 오차를 확인하여 평가하였다. 실험 비교군으로는 [9]의 알고리즘, 본 연구의 알고리즘에서 4-Dof Optimize만을 [9]와 같은 6-Dof Optimize로 변경한 알고리즘이 있다. 각각 후술할 실험의 결과 그래프에서 빨간색과 초록색으로 표기하였으며, 본 연구의 알고리즘을 활용하여 계산한 오도메트리 결과는 파란색 그래프로 표기하였다.

총 세 가지 경로에 대하여 진행했다. 첫 번째 실험은 12층 복도를 약 38m 단순 직진하는 실험이다. 해당 실험의 경우, 단순 직진 시 z축 정밀도를 확인하기 위한 실험이다. [그림 11]에서 실험 장소의 환경을 확인할 수 있다.



[그림 11] Cite of first experiment. Corridor of 12th floor of 301 SNU. Left picture is the picture of the corridor and Right picture is the Bird-Eye view of 12th floor of 301 SNU.

두 번째 실험은 서울대학교 301동 12층 복도를 한 바퀴 돌아 원위치로 돌아오는 경로로 구성되었다. 단순 직진만 포함된 경로가 아닌 yaw 방향 회전이 추가된 도합 100m 내외의 경로에서의 z축 오차를 확인하기 위한 실험으로, [그림 12]에서 실험 환경을 확인할 수 있다.



[그림 12] Cite of second experiment. Corridor of 12th floor 301 SNU. Left two pictures includes the environment from two corners of the corridor. Red line is the robust odometry of robot and black arrows are the directions of the robot at the positions.

마지막 실험은 서울대학교 301동 1층에서 출발하여 2층으로 올라간 뒤, 한 바퀴 돌아서 처음 올라왔던 2층의 원위치로 돌아오는 경로로 진행하였다. 경로 내에 roll, pitch, yaw의 변화가 모두 있을 경우의 z축 오차를 측정하기 위한 실험이며, 실험 시작 위치와 마무리 위치가 서로 다르므로, 2층에서 회전을 시작한 z축 높이와 마무리 지점의 z축 높이를 비교하여 오차를 판별했다. 해당 실험의 실험 환경은 [그림 13]에서 확인할 수 있다.



[그림 13] Cite of third experiment. 1st and 2nd floor of 301 SNU including the center stair. Upper two pictures depict the 1st floor and stair that connects the 1st and 2nd floors. Lower two pictures show the 2nd floor area while red lines are the robust odometry of robot and black arrows are the directions of the robot at the positions.

제 3 절 실험 결과 및 분석

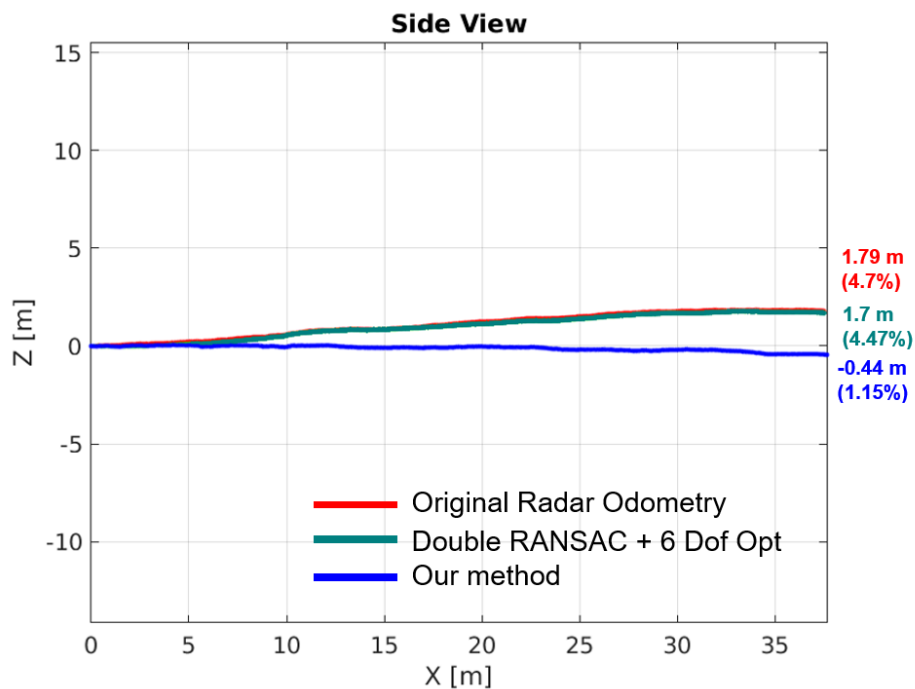
각 실험의 결과는 side view 그래프로 [그림 14,15,16]에 표기하였으며, z축 오차의 정량적인 평가는 [표 5-1,2]에 표기하였다.

| | Original [9] | D-RANSAC + 6 Dof opt | Ours |
|----------|--------------|----------------------|---------|
| Straight | 1.79 m | 1.70 m | -0.44 m |
| Square | 3.14 m | 2.23 m | -0.14 m |
| Stair | 1.73 m | 1.5 m | 0.57 m |

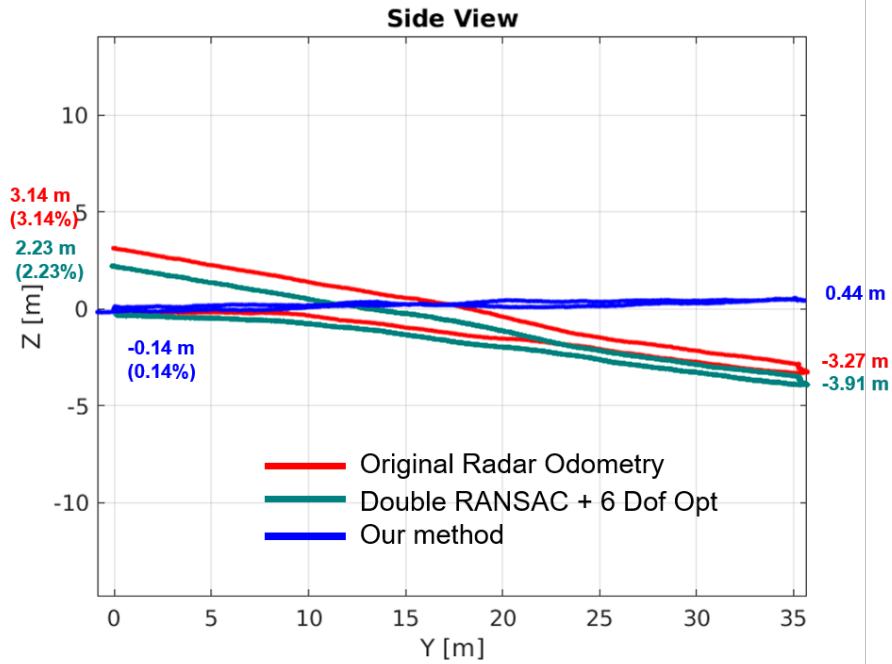
[표 5-1] z axis error of each algorithm about each experiment.

| | Original [9] | D-RANSAC + 6 Dof opt | Ours |
|----------|--------------|----------------------|--------|
| Straight | 4.7 % | 4.47 % | 1.15 % |
| Square | 3.14 % | 2.23 % | 0.14 % |

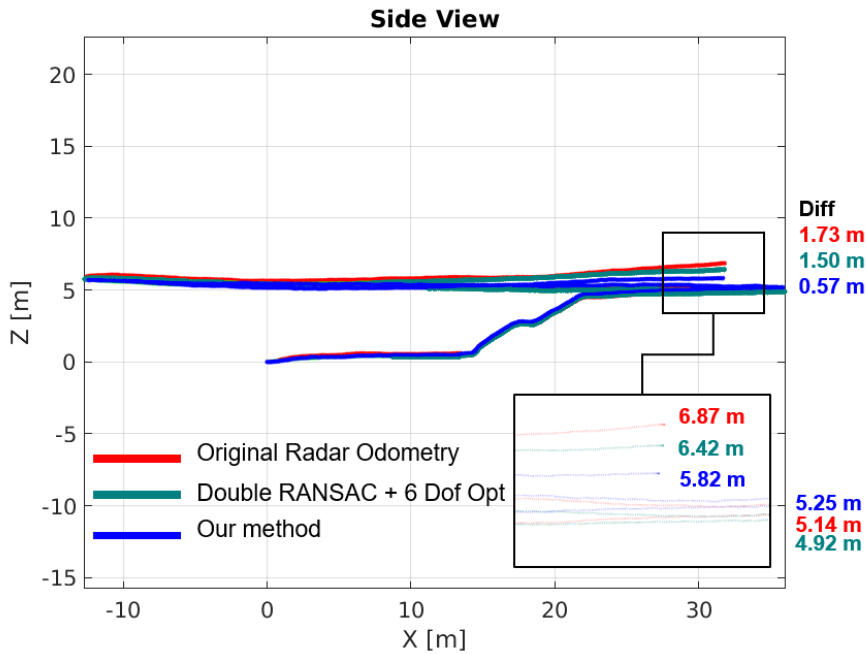
[표 5-2] z axis error of each algorithm about each experiment in percent (z error per travel distance)



[그림 14] First experiment at 12th floor. Robot go straight about 38 meters.



[그림 15] Second experiment at 12th floor. Robot follows the rectangle shape with edge length of 35m and 15m.



[그림 16] Third experiment at 1st ~2nd floor. Robot go straight about 15m, go up stairs about 6 meters, and follows square shape with total length about 80 meters.

[그림 14]의 첫 번째 실험 결과를 통하여, 타 알고리즘에 비하여 본 연구의 알고리즘에서 z 축 오차가 상대적으로 적게 발생함을 확인할 수 있다. [그림 11]의 우측 그림에 포함된 Bird-Eye View 기준 알고리즘별 오도메트리를 확인할 경우 xy 축 오도메트리 결과는 동일하게 나온 것을 확인할 수 있다.

[그림 15]의 두 번째 실험 결과에서 중력 기반 Orientation이 로봇의 initial orientation을 잘 보정하고 있으며, yaw 방향으로 회전하는 과정에서 발생하는 roll과 pitch의 오차로 인하여 z 축 오차가 크게 발생한 타 알고리즘에 비하여 본 연구의 알고리즘은 출발 지점의 반대편 코너 지점에서 첫 번째 실험의 오차와 비슷한 정도의 z 축 오차가 발생했고, 출발 지점으로 돌아온 뒤 z 축 오차가 유의미하게 감소했음을 확인할 수 있다. 또한, [그림 12]의 우측 그림에서 타 알고리즘과 같은 xy 축 오도메트리 결과를 계산했음을 확인할 수 있다. 따라서 해당 실험은 본 연구의 알고리즘이 z 축 오차를 줄이는 과정에서 xy 축 오도메트리 오차의 증가가 발생하지 않았다는 점을 시사한다.

[그림 16]의 마지막 실험 결과에서, 계단을 올라 2층으로 올라가는 시점까지는 타 알고리즘과 z 축 높이에서 큰 차이가 없음을 확인할 수 있다. 하지만 2층 복도를 한 바퀴 주행하여 2층의 원위치로 돌아온 시점에서, 타 알고리즘에 비하여 z 축 오차가 적게 발생했음을 확인할 수 있다. 해당 실험을 통하여 roll, pitch의 변화가 발생하는 실험 환경에서도 z 축 오차가 타 알고리즘에 비하여 적게 발생했음을 확인할 수 있다.

추가적으로, 모든 실험 환경이 서울대학교 301동 건물 내에서 이루어졌으며, 실험 기간이 학기 중이었던 점을 감안할 경우, 실험 현장을 돌아다니는 학생이 꾸준히 존재했으며, 이는 SLAM의 관점에서 볼 때 동적 물체로 취급할 수 있다. 따라서, 본 연구에서 단일 칩 레이더의 z 축 정밀도 한계를 극복하기 위한 방안으로 제시한 Double-RANSAC 알고리즘이 동적 물체와 노이즈, 그리고 z 축 정보가 부정확한 포인트들을 안정적으로 제거했음을 확인할 수 있다.

[표 5-1]은 각 경로별로 발생한 z 축 오차를 정리한 내용으로, 모든 실험에서 본 연구의 알고리즘이 타 알고리즘과 비교하여 1m 이상 오차를 줄였음을 확인할 수 있다. [표 5-2]는 z 축에서 발생한 오차를 전체 경로의 길이에 대한 비율인 퍼센트로 표현한 것으로, 경로의 시작 지점과 마무리 지점의 z 축 높이가 다른(서로 다른 층에 존재하는) 세 번째 실험(stair)의 경우, 타 실험과 달리 비교에서 사용할 경로의

길이도 모호하여 결과의 비교에서 제외하였다. [표 5-1]과 마찬가지로, [표 5-2]의 결과 또한 본 연구의 알고리즘이 비교군 알고리즘에 비하여 최소 2% 이상 z 축 오차가 개선된 것을 확인할 수 있다.

제 6 장 결 론

제 1 절 연구 내용 요약

본 연구는 z 축 정보가 부정확한 단일 칩 레이더 데이터에서 동적 물체와 노이즈, 그리고 z 축 오차가 심한 점군을 제거할 수 있는 Double-RANSAC 알고리즘을 제안했다. 또한, 초기 IMU 가속도 정보를 활용하여 중력 가속도 벡터를 측정, 로봇의 Initial Orientation을 추정하였으며, 마지막으로 기존에는 카메라 및 라이다 SLAM에서 사용되던, 중력 가속도 벡터를 기반으로 하는 4 Dof 최적화 방법론을 레이더 오도메트리의 최적화 과정에 적용함으로써 z 축 오차를 줄였다. 연구에서 제안한 알고리즘의 성능을 확인하기 위하여 z 축 진동이 심한 Boston Dynamics SPOT 4족 보행 로봇에 센서 시스템을 탑재하여 직진, 회전, 계단이 포함된 여러 실내 경로에서 실험을 진행하였다. 실험 결과 기존의 칩 레이더 오도메트리 알고리즘과 비교하여 1m 이상 z 축 오차가 감소했음을 확인하였다.

제 2 절 향후 연구 계획

본 연구를 통하여 기존 칩 레이더 오도메트리 알고리즘에 비하여 z 축 오차 발생 정도를 줄였으며, 단일 칩 레이더 센서에 기반한 6 Dof 위치 정보 최적화를 수행하였다. 향후 연구에서 SPOT 4족 보행 로봇을 단순히 UGV 플랫폼으로 사용하는데 그치지 않고 Joint Encoder^⑩ 및 Contact Sensor^⑪ 데이터를 적극적으로 활용, Forward Kinematics를 기반한 Leg Odometry Factor와 본 연구의 4-Dof chip radar Factor를 융합하여, 보다 강건한 4족 보행 로봇의 Odometry를 계산할 수 있는 알고리즘을 연구할 것이다. 이외에도 제안하는 Odometry 알고리즘과 라이다 센서를 활용하여, 기존의 single LiDAR SLAM에서 해결하기 어려운 degenerate^⑫ 환경에서의 mapping 문제를 해결할 것이다.

^⑩ 로봇 관절의 각도, 각속도, 토크 값 등의 정보를 제공하는 센서

^⑪ 로봇 발의 Ground Contact 상태 정보를 제공하는 센서

^⑫ 카메라 및 라이다 센서 등에서 특징점을 추출하기 어려운 환경

참고 문헌

- [1] Shan, Tixiao, and Brendan Englot. "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain." *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.
- [2] Shan, Tixiao, et al. "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping." *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020.
- [3] Xu, Wei, and Fu Zhang. "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter." *IEEE Robotics and Automation Letters* 6.2 (2021): 3317–3324.
- [4] Jung, Minwoo, Sangwoo Jung, and Ayoung Kim. "Asynchronous Multiple LiDAR-Inertial Odometry using Point-wise Inter-LiDAR Uncertainty Propagation." *IEEE Robotics and Automation Letters* (2023).
- [5] Checchin, Paul, et al. "Radar scan matching slam using the fourier-mellin transform." *Field and Service Robotics: Results of the 7th International Conference*. Springer Berlin Heidelberg, 2010.
- [6] Park, Yeong Sang, Young-Sik Shin, and Ayoung Kim. "Pharao: Direct radar odometry using phase correlation." *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
- [7] Wang, Wei, et al. "Radarloc: Learning to relocalize in fmcw radar." *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.
- [8] Kellner, Dominik, et al. "Instantaneous ego-motion estimation using multiple Doppler radars." *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014.
- [9] Park, Yeong Sang, et al. "3D ego-motion estimation using low-cost mmWave radars via radar velocity factor for pose-graph SLAM." *IEEE Robotics and Automation Letters* 6.4 (2021): 7691–7698.
- [10] Qin, Tong, Peiliang Li, and Shaojie Shen. "Vins-mono: A robust

and versatile monocular visual-inertial state estimator." *IEEE Transactions on Robotics* 34.4 (2018): 1004–1020.

[11] Kubelka, Vladimír, Maxime Vaidis, and François Pomerleau. "Gravity-constrained point cloud registration." *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[12] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." *Communications of the ACM* 24.6 (1981): 381–395.

[13] Kellner, Dominik, et al. "Instantaneous ego-motion estimation using Doppler radar." *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013.

[14] Almalioglu, Yasin, et al. "Milli-RIO: Ego-motion estimation with low-cost millimetre-wave radar." *IEEE Sensors Journal* 21.3 (2020): 3314–3323.

[15] Magnusson, Martin, Achim Lilienthal, and Tom Duckett. "Scan registration for autonomous mining vehicles using 3D-NDT." *Journal of Field Robotics* 24.10 (2007): 803–827.

[16] Kramer, Andrew, et al. "Radar-inertial ego-velocity estimation for visually degraded environments." *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[17] Lu, Chris Xiaoxuan, et al. "milliEgo: single-chip mmWave radar aided egomotion estimation via deep sensor fusion." *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 2020.

[18] Zhang, Zichao, Guillermo Gallego, and Davide Scaramuzza. "On the comparison of gauge freedom handling in optimization-based visual-inertial state estimation." *IEEE Robotics and Automation Letters* 3.3 (2018): 2710–2717.

[19] Besl, Paul J., and Neil D. McKay. "Method for registration of 3-D shapes." *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. Spie, 1992.

Abstract

Radar Odometry for Quadrupedal Robot using Gravity

Sangwoo Jung

Mechanical Engineering

The Graduate School

Seoul National University

Sensor odometry is critical in the field of SLAM as the inaccurate odometry may affect the quality of result 3D map or path planning. One of the important issues of odometry accuracy is z axis floating occurred from incorrect optimization or feature matchings. The use of gravity vector that can be estimated from IMU sensors can provide opportunity of exploiting 4-Dof optimization, the technique that has been widely utilized in the Camera and LiDAR SLAM. Inspired from the idea, we present a novel single chip radar odometry with 4-Dof optimization based on gravity estimation. Our method proposes double RANSAC technique to remove noise and dynamic objects first, and z axis inaccurate points second. Exploiting the IMU acceleration data, gravity vector is estimated and based on this, initial orientation of the robot is calibrated. Lastly, 4-Dof optimization on x, y, z, and yaw is done to calculate the sensor odometry that is more robust on z axis error. We tested our algorithm with Boston Dynamics Spot quadrupedal robot as a legged UGV which occurs high level of z axis vibration. Our algorithm demonstrates a significant improvement in z axis accuracy compared to the base method.

Keywords : SLAM, Radar Odometry, Gravity Estimation, Optimization,
Quadrupedal Robot

Student Number : 2021-27880

Acknowledgements in Korean

졸업할 때마다 느끼는 점이지만, 시간은 참 빠른 것 같습니다. 랩 인턴으로 들어와서 Husky를 분해할 때가 엇그제 같은데, 벌써 석사 졸업이 코앞으로 다가왔네요. 석사 과정에서, 많은 분께 도움을 받았고, 짧게나마 감사의 인사를 드리고자 합니다.

가장 먼저 김아영 교수님께 감사드립니다. 아직 부족하지만, 석사 과정을 거치며 발전된 시야와 통찰을 가질 수 있게 된 데에는 수평적이고 미래 지향적인 연구실의 분위기가 중요한 역할을 했다고 생각합니다. 이러한 학풍과 연구환경을 위한 교수님의 노력에 정말 감사드리며, 박사과정을 거치며 더욱더 발전된 모습을 보여드리겠습니다.

다음으로는 IRAP 선배님들과 RPM 분들께 감사드립니다. 언제나 넘치는 열정과 지치지 않는 체력으로 연구, 출장, 여행을 같이 다닌 연구실 동기 민우 님, 졸업 축하하고 앞으로 함께할 박사과정 기간에도 잘 부탁드립니다. 팔방미인이라는 단어가 잘 어울리는 혜수 형, 연구를 할 때나 과제를 할 때나 형이 있어서 언제나 든든합니다. 성실함의 아이콘인 하드웨어 장인 승상이 형, 어떤 일이라도 최선을 다하는 모습을 본받겠습니다. 재미있지만 우습지 않은, 언제나 든든한 우리의 랩장 정윤이 형, 항상 유머 속에 숨겨진 따뜻한 격려로 북돋아 줘서 감사합니다.

연구에 있어서 누구보다 진심인 딥러닝 전문가 동규 형, 언제나 모르는 점을 잘 알려주셔서 감사하고, 앞으로도 많이 여쭙겠습니다. 팀으로 움직일 때 좋은 관리자 역할을 하는 현제 님, 언제 어디서나 잘 해낼 거라고 믿습니다. 물리적으로나 연구적으로나 깊은 내공을 가진 명환이 형, 항상 좋은 조언 해 주셔서 감사합니다.

n개 국어가 가능한 코딩 차력소 마스터 동재 님, 언제나 날카로운 질문과 직관을 보여주는 재호 님, 연구 이야기를 할 때 막힌 혈을 뚫어주는 우성 님, 수학적으로 아무리 어려운 문제도 척척 풀어내는 채현 님, 그리고 이젠 우리 곁을 떠나 넓은 세상으로 나아간, 과거 4족 보행 로봇을 처음 공부할 때 같이 맨땅에 헤딩했던 원석 님께 감사의 말씀을 드립니다.

다음으로는 졸업하신 뒤에도 꾸준히 좋은 조언을 아낌없이 들려주시는 기섭이 형, 학회에서 언제나 좋은 말씀을 해 주신 만형 주완 형님, 레이더 분야에서 기틀을 잡아 주신 영상 형님, 해외로 뻗어나가는 좋은 선례를 만들어 주신 성호 형님, 언제나 감사드리며, 항상 행복하셨으면 합니다. 이외에도 지금의 연구실이 있기까지 노력해주신 모든 선배님께 진심으로 감사드립니다. 우리 연구실의 이름에 부끄럽지 않은 연구자가 될 수 있도록 최선을 다해 정진하겠습니다.

연구실 외적으로도 감사한 분이 많습니다. 대학교 신입생 시절부터 말 그대로 동고동락하며 지내온 7반 기숙사 룸메 친구들, 벌써 8년째 웃고 싸우며 이리 뛰고 저리 구른 바오밥 6기 친구들, 여러모로 힘들 때 좋은 버팀목이 되어준 답도 없는 노답 특방 친구들, 마지막으로 언제나 편하게 연락해서 깊은 이야기를 할 수 있는 좋은 친구 승일이에게 고맙다는 말을 전하고 싶습니다. 다들 지금까지 부족한 제 성격을 받아줘서 고맙고, 앞으로도 잘 부탁드립니다.

세상에서 가장 사랑하는, 지금의 제가 있게 해 주신 부모님께 감사드립니다. 집은 휴식의 공간이라며 항상 따뜻하게 맞아 주시는 엄마, 언제나 저를 믿고 지지해주시는 든든한 아빠, 표현은 잘하지 못했지만 언제나 감사드립니다. 부끄럽지 않은 아들이 될 테니 항상 건강하게 지켜봐 주세요. 마지막으로 배울 점이 정말 많은 자랑스러운 동생 현우야, 우리 각자의 분야에서 최선을 다해 발전해 나가자. 언제나 사랑한다.

지금까지 25년간 제 성장에 도움을 주신 모든 분께 감사드립니다.