# Final Project Report on:

# LIO-SAM with the Spot - a Robot Dog

**Gitlab username for other files: mistry.dhy**

## Introduction

Autonomous mobile robots are at the forefront of innovation, transforming industries and redefining operational capabilities. Among these robots, Boston Dynamics' Spot has gained recognition for its agility, advanced autonomy, and versatility in navigating complex environments. Leveraging state-of-the-art SLAM (Simultaneous Localization and Mapping) algorithms, such as LIO-SAM, enhances Spot's ability to operate in GPS-denied areas, enabling applications in construction, inspection, and disaster response.

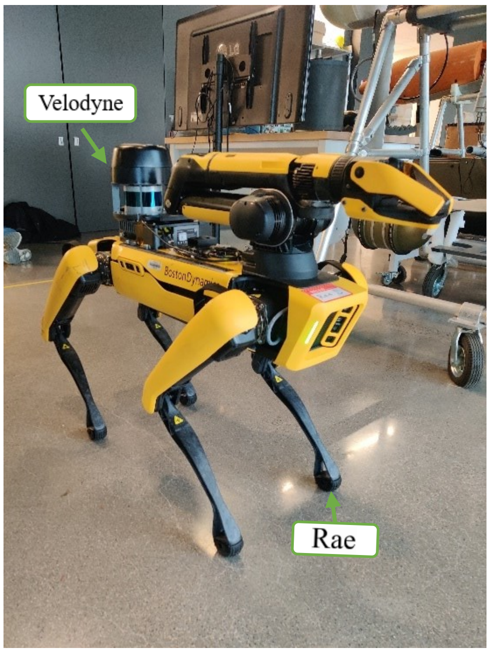


*fig-1: Spot with Velodyne lidar*

LIO-SAM is a cutting-edge algorithm that combines data from LiDAR and IMU sensors to achieve real-time, high-precision 3D mapping and localization. The integration of these technologies allows robots like Spot to generate accurate maps of their surroundings while maintaining robust localization.

Our project aims to implement LIO-SAM on Spot to achieve high-precision 3D mapping and localization in dynamic environments, focusing on overcoming practical challenges and exploring its potential in dynamic, real-world scenarios. While the project is still not at the stage that we desired it to be, significant progress has been made in understanding and applying the algorithm, resolving challenges, and collecting high-quality datasets with Spot in and around the EXP building.

### Motivation

Spot, Boston Dynamics' agile and autonomous robot, is transforming site operations by enhancing safety, enabling remote monitoring, and promoting data-driven decision-making as seen in fig-2. Its great mobility and advanced autonomy make it a versatile tool in industries such as construction, inspection, and disaster management. Integrating Spot with state-of-the-art SLAM algorithms like LIO-SAM extends its utility across environments requiring precise localization and mapping, even in areas without GPS access. This project aimed to leverage Spot's capabilities to demonstrate LIO-SAM's potential in real-world applications.

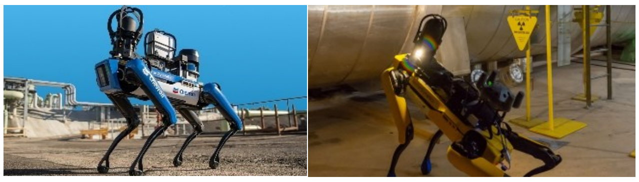


*fig-2: Spot working in industries*

## Methodology

LIO-SAM is a SLAM algorithm designed for real-time 3D mapping and localization. It tightly integrates LiDAR and IMU data, providing robust and accurate results even in dynamic and complex environments. Its key features include:

- **Tight Integration of LiDAR and IMU**: Ensures precise spatial accuracy and motion compensation.
- **Factor Graph Optimization**: Maintains global consistency and reduces drift over time.

- **Real-Time Operation**: Makes it suitable for mobile robotics applications.
- **Loop Closure Detection**: Enhances map accuracy by recognizing revisited locations.

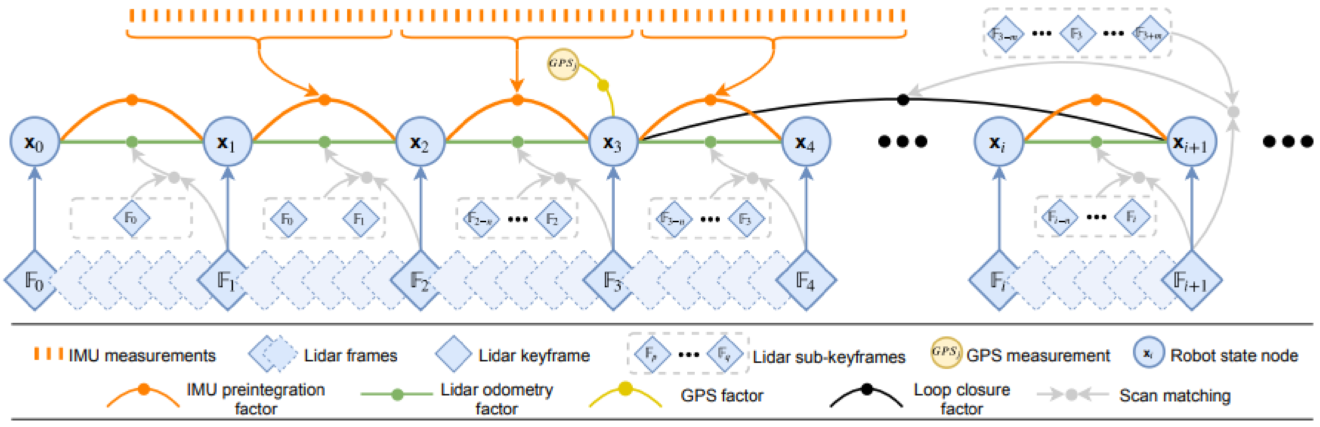


*fig-3: LIO-SAM Algorithm working*

**Algorithm Overview**

The LIO-SAM algorithm operates through a systematic pipeline:

1. **Input Sensors**:
    - **IMU**: Captures motion data such as orientation and acceleration.
    - **LiDAR**: Generates 3D point clouds of the environment.
    - **GPS (Optional)**: Provides additional global positioning constraints.

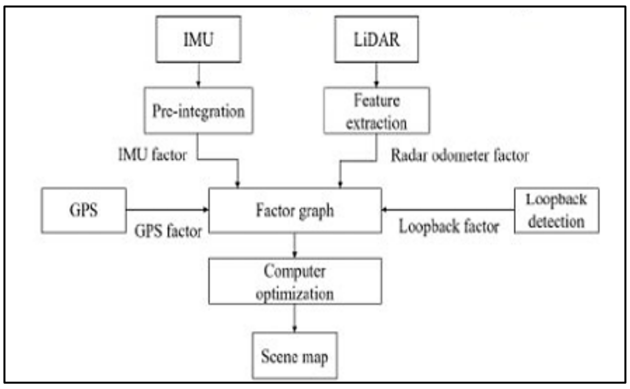


*fig-4: LIO-SAM Flow chart*

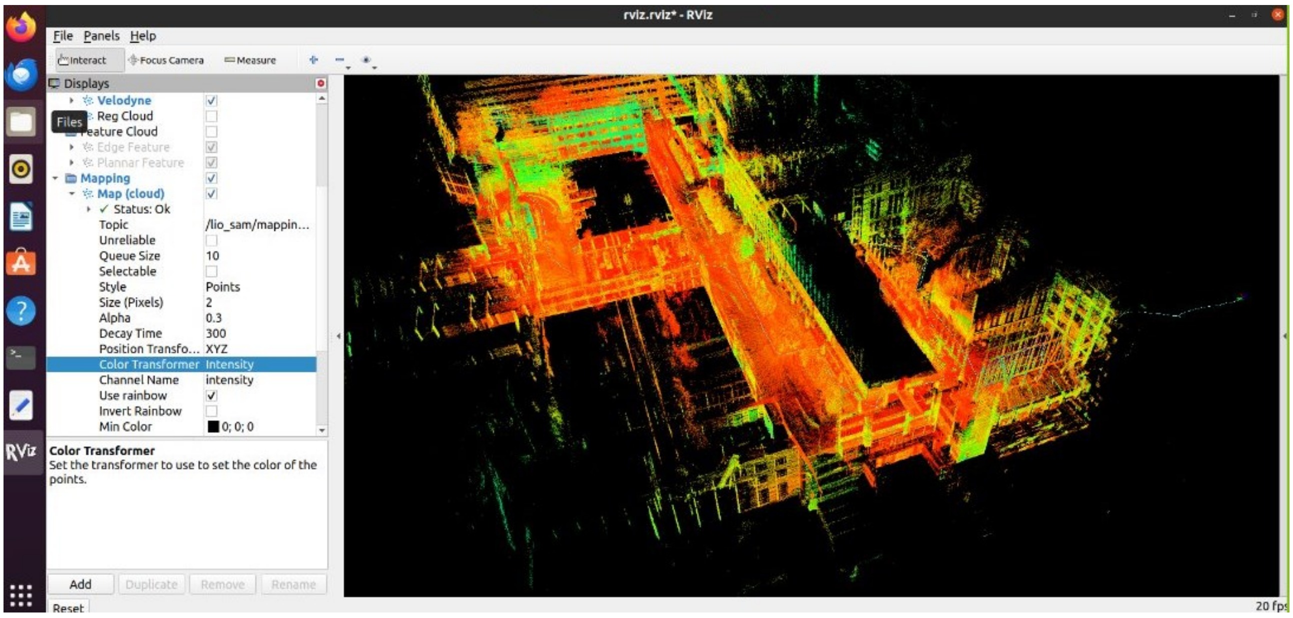
2. **Data Processing**:
    - **IMU Pre-integration**: Estimates motion trajectories between successive scans.
    - **LiDAR Feature Extraction**: Identifies significant edge and planar features for localization.
    - **Factor Graph Construction**: Combines constraints from IMU, LiDAR, GPS, and loop closure.
3. **Output**:
    - Produces a globally consistent 3D map and accurate pose estimates.

The more detailed and "interesting" math can be found in their paper here.

## Experimentation

The implementation process began with a thorough review of the foundational paper, "LIO-SAM: Tightly-Coupled Lidar Inertial Odometry via Smoothing and Mapping," to understand the theoretical underpinnings of the algorithm. We also analyzed existing implementations to identify potential challenges and limitations in deploying the algorithm on Spot.

After going through the research paper and their GitHub repository, we cloned it in the ROS1 noetic environment in Ubuntu 20.04. Initially, we thought that this was only supported by the ROS1 but, very later in the project, we found out the ROS2 compatibility too. After that we implemented the LIO-SAM algorithm on the KITTI dataset (map of the MIT campus). This step allowed us to validate the algorithm's capabilities and ensure its functionality in generating accurate maps (fig-5). During this stage, we successfully applied loop closure(fig-6), which significantly improved map consistency and point cloud visuals in Rviz. In fig-



```
Displays
  Velodyne              ✓
  Reg Cloud
  Feature Cloud
  Edge Feature          ✓
  Plannar Feature       ✓
Mapping                 ✓
  Map (cloud)           ✓
    ✓ Status: Ok
    Topic               /lio_sam/mappin...
    Unreliable
    Queue Size          10
    Selectable
    Style               Points
    Size (Pixels)       2
    Alpha               0.3
    Decay Time          300
    Position Transfo... XYZ
    Color Transformer   Intensity
    Channel Name        intensity
    Use rainbow         ✓
    Invert Rainbow
    Min Color           0; 0; 0

Color Transformer
Set the transformer to use to set the color of the points.
```

*fig-5: LIO-SAM Algorithm on MIT dataset*

6 the square box represents the points identified for the loop closer and the yellow lines connecting the blue path of the robot indicates the working of ICP algorithm.

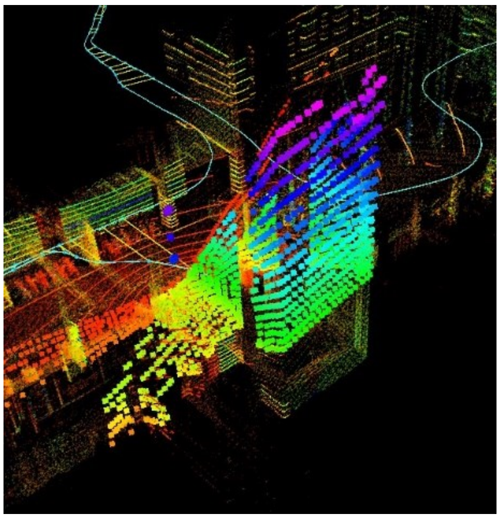


*fig-6: Loop closer and ICP working*

Once the algorithm was validated, we proceeded to data collection. Initially, we collected the data in the High Bay using Spot's onboard sensors and Velodyne lidar. We took the spot and went around the tables indoors in the High Bay and then plotted the visualized data in Rviz. In this trial data, we got clean point cloud data, with great detailing, and got spot's position in the IMU topic.

*fig-7: [Data collection with the Spot in the High Bay]*

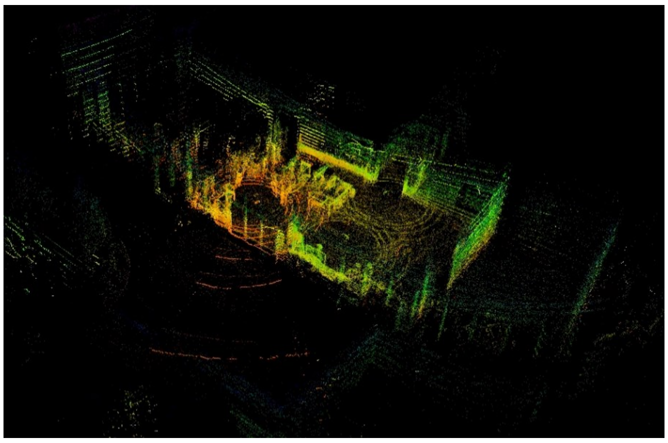


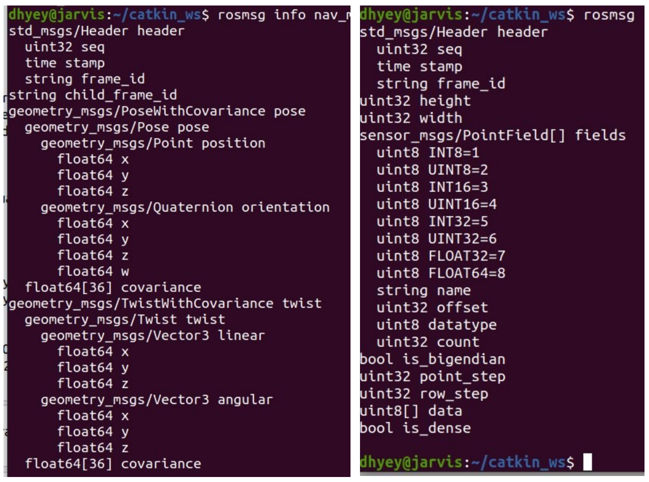
*fig-8: [High-Bay point cloud data in Rviz]*

But the fig-8 is just the point cloud representation of the data and not the LIO-SAM algorithm map. As the LIO-SAM requires the integration of the IMU and lidar data in the algorithm, the data that we got from the spot doesn't had the IMU data that was required by the LIO-SAM algorithm to work.

To solve that we had a few solutions:

- Take IMU data from Spot's SDK file
- Use the quaternions values in odometry msg, and write a custom file to integrate the new IMU values into our algorithm
- Simply add an external Vectornav IMU onto the spot

We went through all the solutions one by one; the first solution was not that straightforward so we skipped it and thought we would return to it at last.



```
dhyey@jarvis:~/catkin_ws$ rosmsg info nav_
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
string child_frame_id
geometry_msgs/PoseWithCovariance pose
  geometry_msgs/Pose pose
    geometry_msgs/Point position
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
  float64[36] covariance
geometry_msgs/TwistWithCovariance twist
  geometry_msgs/Twist twist
    geometry_msgs/Vector3 linear
      float64 x
      float64 y
      float64 z
    geometry_msgs/Vector3 angular
      float64 x
      float64 y
      float64 z
  float64[36] covariance
```

```
dhyey@jarvis:~/catkin_ws$ rosmsg
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
uint32 height
uint32 width
sensor_msgs/PointField[] fields
  uint8 INT8=1
  uint8 UINT8=2
  uint8 INT16=3
  uint8 UINT16=4
  uint8 INT32=5
  uint8 UINT32=6
  uint8 FLOAT32=7
  uint8 FLOAT64=8
  string name
  uint32 offset
  uint8 datatype
  uint32 count
bool is_bigendian
uint32 point_step
uint32 row_step
uint8[] data
bool is_dense

dhyey@jarvis:~/catkin_ws$ 
```

*fig-9: Spot's ros msg info*

Then we tried writing our own script to modify the odometry data to the algorithm's required IMU data. But in that, we had many compatibility problems and at last, it was resolved but then the lidar msg-type had a few things missing like the rings and the intensity values. The quick solution to that was, directly adding the missing values in the msg as dummy values but that didn't work either.

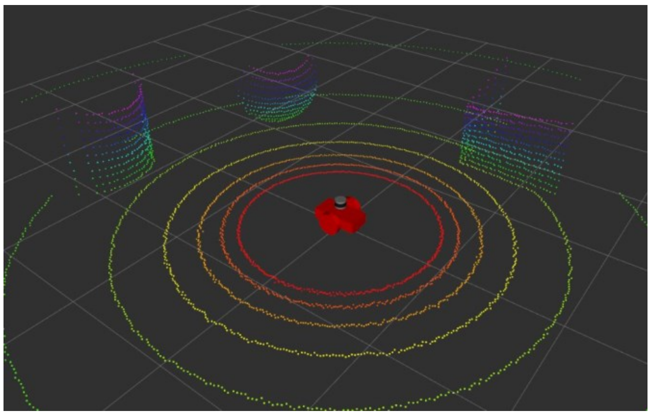


*fig-10: Rings in the Lidar data*

There were many different errors and problems that arise one after another in all of our devices. We tried this LIO-SAM in both ROS1 and ROS2 environments. We used docker, virtual environment, ros-bridge, and many other methods to resolve the error. After trying multiple times we arrived at our final solution, using an external vector nav IMU on Spot.

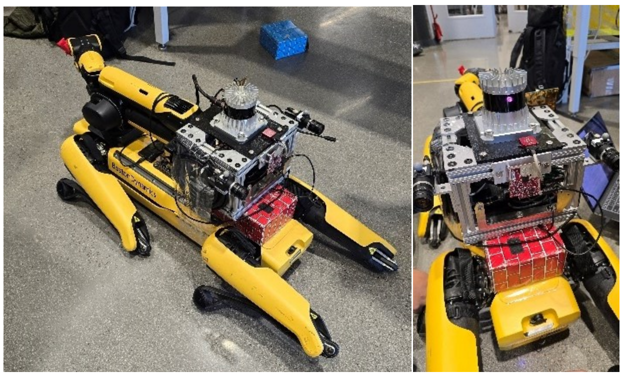


*fig-11: Spot with new rig having Lidar, vector nav IMU, cameras, and the jetson nano.*

To enhance the system's capabilities, we used a custom rig featuring a Jetson Nano, a LiDAR, and a VectorNav IMU (fig-11). This rig was designed to augment Spot's data collection potential, particularly in complex scenarios. Finally, we integrated the rig with ROS2, enabling seamless data acquisition and processing to support our mapping and localization objectives. We took the spot for a walk around the EXP building from the High Bay, with the new rig attached on it, and made a loop around the building.

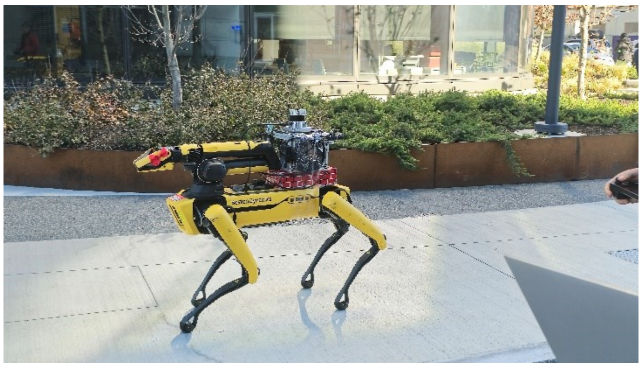


*fig-12: [Spot data recording session around EXP]()*

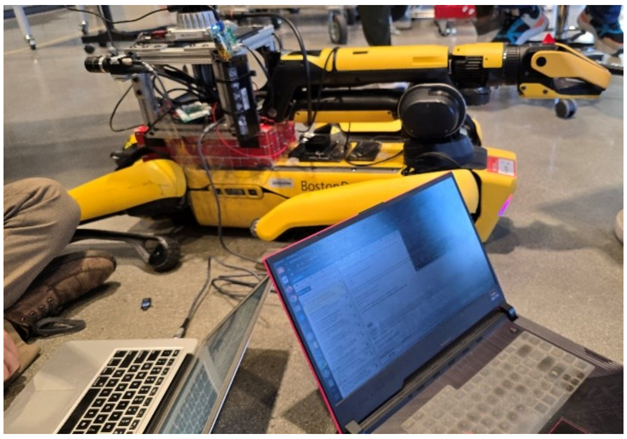
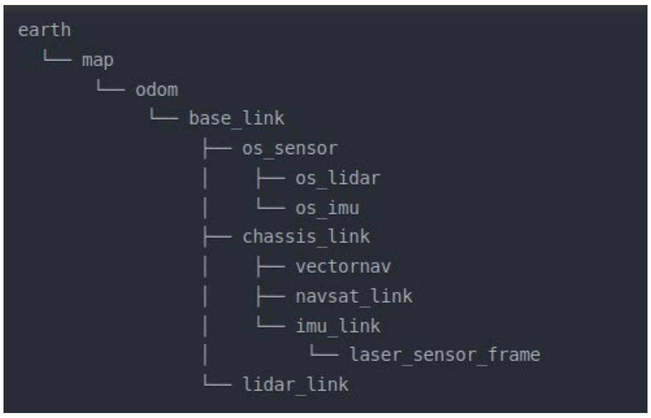


*fig-13: Working on Spot in High Bay*

The data was recorded successfully we got the IMU and Lidar topics, but for some unknown reason, the 2nd loop's camera data was not recorded so the teams using RTAB can't use the 2nd loop data. For us, it was good to go, now as the data was recorded in the Jetson Nano's internal storage, we began to transfer it to our local computer via the "scp" command in Ubuntu but mistakenly copied it again two times in the Jetson itself and during the changing of the name of the "New_Volme" disk in another laptop we almost crashed it by unmounting the internal harddisk of laptop. But, it was a good learning opportunity for us.

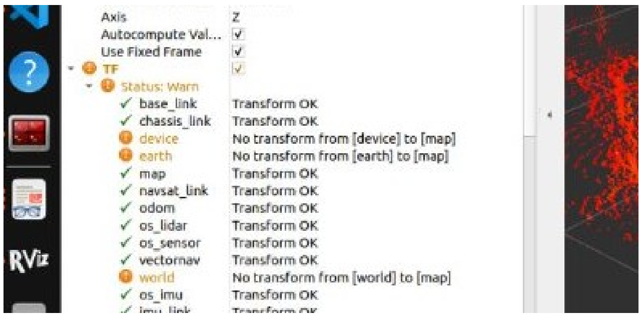
As the data that we recorded was recorded in ROS2 and that too in ".mcap" format, so initially we were trying to convert the .mcap to .db3 and then that in to ROS1 bag files, but later we got to know about the LIO-SAM'S ROS2 support and used it and also we were able to play".mcap" files directly so we didn't had to convert it.

The next error we encountered was during the implementation of LIO-SAM in our new recorded data. We started getting errors in TF when viewed in Rviz. The desired TF tree is shown in fig-14 and the TF tree that we got is shown in fig-16, in which we can see on the top right that it has world to device loop for some reason and that might be either hindering with the process and causing the errors.



```
earth
  └── map
        └── odom
              └── base_link
                    ├── os_sensor
                    │     ├── os_lidar
                    │     └── os_imu
                    ├── chassis_link
                    │     ├── vectornav
                    │     ├── navsat_link
                    │     └── imu_link
                    │           └── laser_sensor_frame
                    └── lidar_link
```

*fig-14: Desired and correct tree for the Transformation Frame*



```
Axis                 Z
Autocompute Val...   ✓
Use Fixed Frame      ✓
TF                   ✓
  Status: Warn
    base_link        Transform OK
    chassis_link     Transform OK
    device           No transform from [device] to [map]
    earth            No transform from [earth] to [map]
    map              Transform OK
    navsat_link      Transform OK
    odom             Transform OK
    os_lidar         Transform OK
    os_sensor        Transform OK
    vectornav        Transform OK
    world            No transform from [world] to [map]
    os_imu           Transform OK
    imu_link         Transform OK
```

*fig-15: Error in TF Rviz*

We were also not getting the IMU data in the package as seen in the fig-17, and due to that our LIO-SAM algorithm couldn't be implemented up to the full extent. We were getting great and dense point cloud data from the lidar but were unable to make a map from it due to the unavailability of the IMU data. The fig-18 shows the point cloud of new collected data visualized in Rviz.

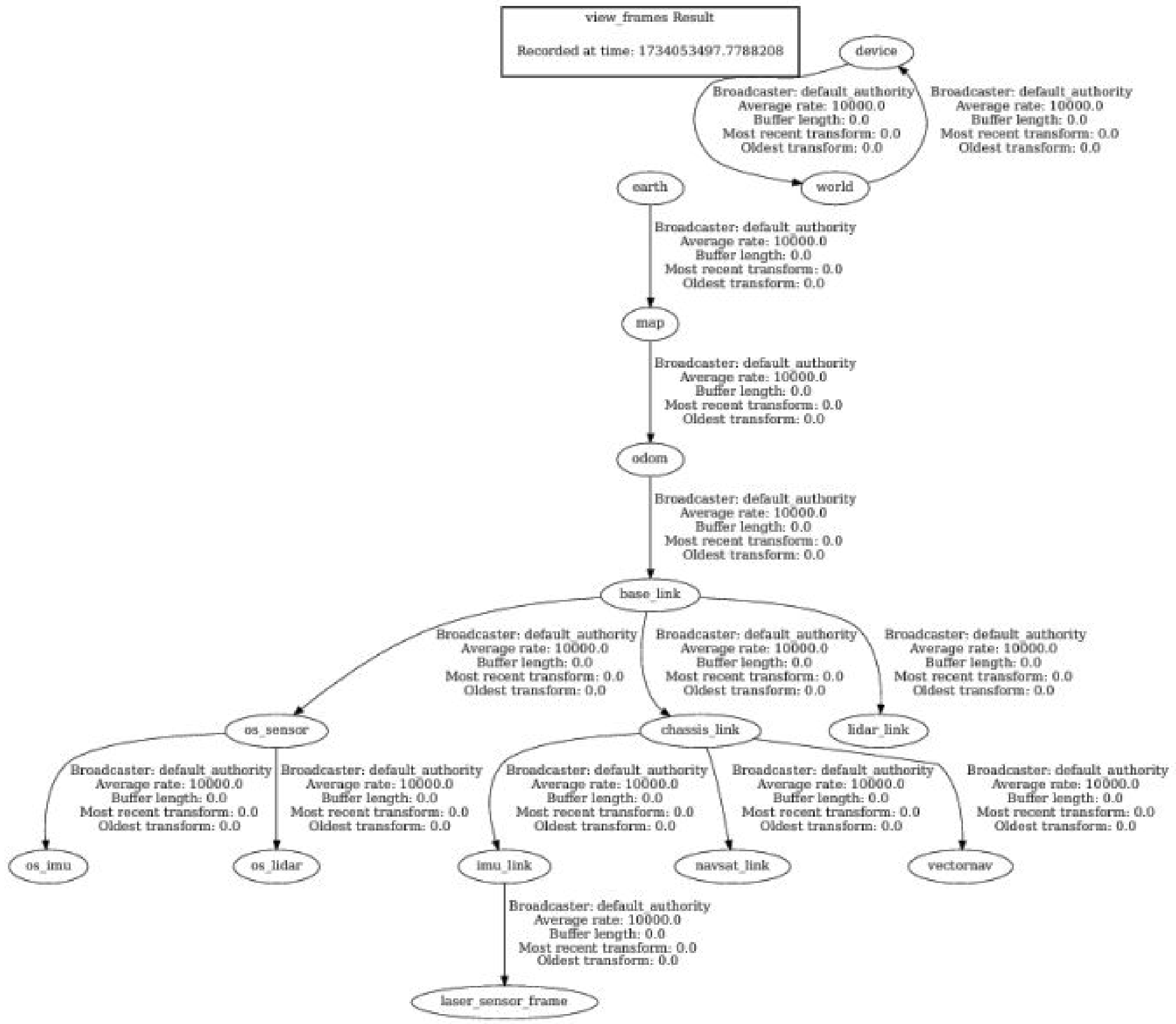
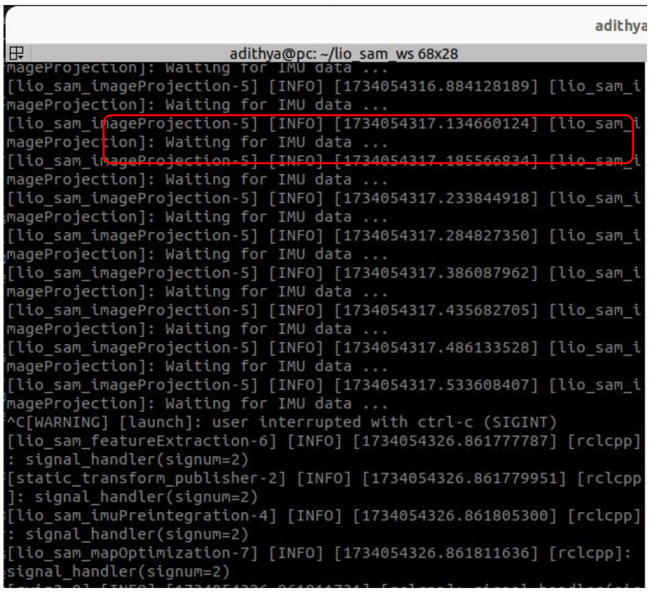


*fig-16: Our TF tree with device-world loop*



```
adithya@pc: ~/lio_sam_ws 68x28
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054316.884128189] [lio_sam_i
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054317.134660124] [lio_sam_i
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054317.185566834] [lio_sam_i
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054317.233844918] [lio_sam_i
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054317.284827350] [lio_sam_i
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054317.386087962] [lio_sam_i
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054317.435682705] [lio_sam_i
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054317.486133528] [lio_sam_i
mageProjection]: Waiting for IMU data ...
[lio_sam_imageProjection-5] [INFO] [1734054317.533608407] [lio_sam_i
mageProjection]: Waiting for IMU data ...
^C[WARNING] [launch]: user interrupted with ctrl-c (SIGINT)
[lio_sam_featureExtraction-6] [INFO] [1734054326.861777787] [rclcpp]
: signal_handler(signum=2)
[static_transform_publisher-2] [INFO] [1734054326.861779951] [rclcpp
]: signal_handler(signum=2)
[lio_sam_imuPreintegration-4] [INFO] [1734054326.861805300] [rclcpp]
: signal_handler(signum=2)
[lio_sam_mapOptimization-7] [INFO] [1734054326.861811636] [rclcpp]:
signal_handler(signum=2)
```

*fig-17: No IMU data*

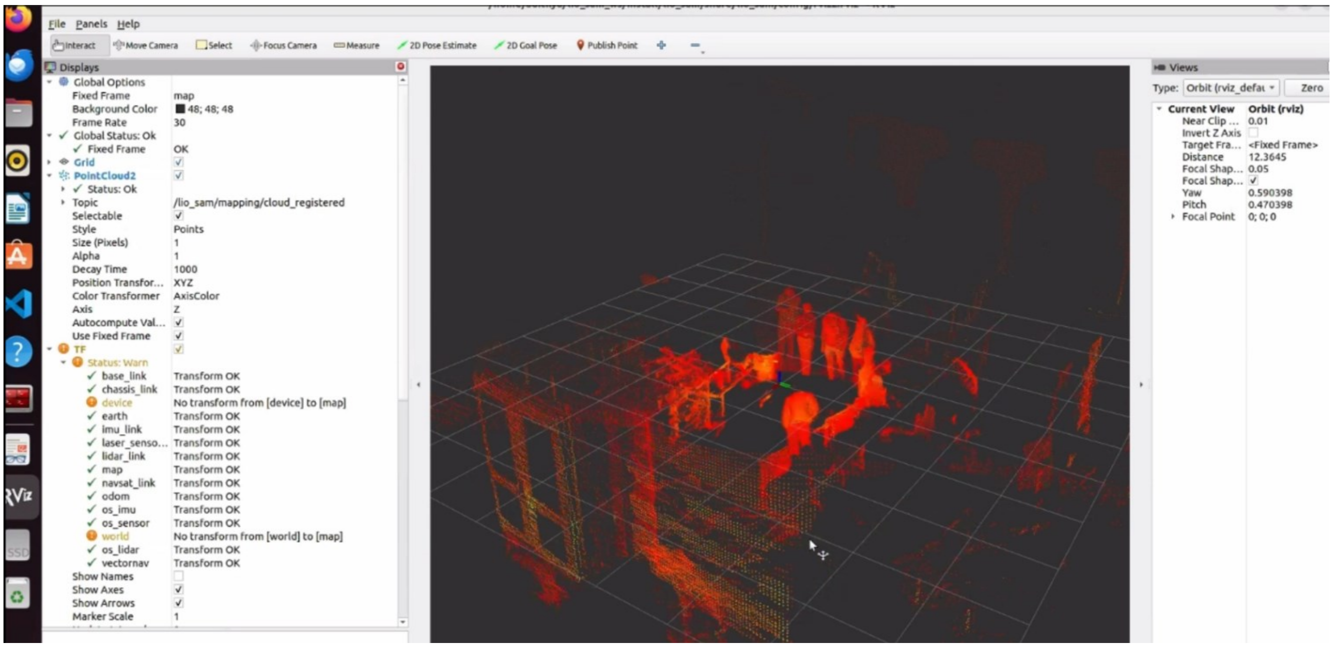
Since the LIO-SAM is been tested in Microstrain IMUs and we are using Vectornav's IMU, the LIO-SAM package is unable to receive, process, and visulize the data properly.

## Results and Discussion

### Achievements

1. **Algorithm Validation**:
   - Successfully demonstrated LIO-SAM's functionality on the KITTI dataset, achieving real-time loop closure and mapping.
   - Generated high-quality point clouds and consistent maps during initial trials.

- Even though we couldn't make the final map, because of the IMU problem, we tested this algorithm



```
Global Options
  Fixed Frame              map
  Background Color         ■ 48; 48; 48
  Frame Rate               30
✓ Global Status: Ok
  ✓ Fixed Frame            OK
Grid                       ✓
PointCloud2                ✓
  ✓ Status: Ok
  Topic                    /lio_sam/mapping/cloud_registered
  Selectable               ✓
  Style                    Points
  Size (Pixels)            1
  Alpha                    1
  Decay Time               1000
  Position Transfor...     XYZ
  Color Transformer        AxisColor
  Axis                     Z
  Autocompute Val...       ✓
  Use Fixed Frame          ✓
TF                         ✓
  Status: Warn
    ✓ base_link            Transform OK
    ✓ chassis_link         Transform OK
      device               No transform from [device] to [map]
    ✓ earth                Transform OK
    ✓ imu_link             Transform OK
    ✓ laser_senso...       Transform OK
    ✓ lidar_link           Transform OK
    ✓ map                  Transform OK
    ✓ navsat_link          Transform OK
    ✓ odom                 Transform OK
    ✓ os_imu               Transform OK
    ✓ os_sensor            Transform OK
      world                No transform from [world] to [map]
    ✓ os_lidar             Transform OK
    ✓ vectornav            Transform OK
  Show Names               ☐
  Show Axes                ✓
  Show Arrows              ✓
  Marker Scale             1
```

```
Type: Orbit (rviz_defa
Current View               Orbit (rviz)
  Near Clip ...            0.01
  Invert Z Axis            ☐
  Target Fra...            <Fixed Frame>
  Distance                 12.3645
  Focal Shap...            0.05
  Focal Shap...            ✓
  Yaw                      0.590398
  Pitch                    0.470398
  Focal Point              0; 0; 0
```

*fig-18: Precise point cloud map with new data on rig*

in the other data set and it worked great with loop closer too.

2. **Data Collection Success**:

   - Captured detailed maps of the High Bay and EXP building environments.
   - Got to learn working on Spot, integrating and reading sensors, and collecting and processing the data.
   - Produced visualizations in Rviz, highlighting the hardware and algorithm accuracy.

3. **Hardware Integration**:

   - We experienced both Spot GXP (General expansion payload) (also EAP 2) and an external rig mounted on Spot for data collection.
   - Deployed a robust system combining Spot, Jetson Nano, LiDAR, and VectorNav IMU for enhanced data acquisition. (new rig, thanks to Prof Hanu and TAs)

**Challenges Overcome**

- **Timestamp Mismatches**: Resolved

  discrepancies in custom driver timestamps, ensuring data synchronization. (but didn't use it in the end)
- **ROS2 bag files compatibility**: Learned about LIO-SAM's ROS2 compatibility and using ".mcap" bag files directly without conversions.
- **Missing IMU data**: This was the main challenge of this project, where we didn't got the IMU data in hand but found many different way of getting it.

**Challenges Faced**

**Data Compatibility**

- The SPOT_IMU messages lacked ring and intensity data, complicating LiDAR integration despite their visualization in Rviz.

- LIO-SAM's original design for Microstrain IMUs necessitated significant adaptations for VectorNav IMU data.

**Transform Frame (TF) Issues**

- Encountered looping errors in the world and device TFs, causing inaccuracies in pose estimation and mapping.

**Algorithm Adaptation**

- Required modifications to quaternion data and troubleshooting IMU integration to accommodate hardware-specific constraints.
- We also changed a few things in the LIO-SAM's algorithm script, in order for it to be compatible with our data. We changed the parameters in the script multiple times according to the IMU data availability. The edited files are uploaded in GitLab (mistry.dhy).

**Discussion**

**Insights Gained**

1. **Algorithmic Understanding**:
    - Enhanced knowledge of factor graph optimization, sensor fusion, and SLAM principles.
    - Developed skills to adapt algorithms for diverse hardware configurations.
2. **Team Collaboration**:
    - Fostered problem-solving skills through iterative debugging and collaboration, not only within the team but also across other teams too.

**Limitations**

1. **Hardware Constraints**:
    - Dependency on specific IMU sensors limited flexibility.
2. **Software Issues**:
    - Transform errors and data integration challenges hindered progress in certain environments.

**Future Directions (We would like to work on it)**

1. **Advanced Testing**:
    - Solve the IMU data problem, and implement the LIO-SAM in recorded data.
    - Evaluate LIO-SAM in multi-floor and tunnel environments to assess scalability.
2. **Algorithm Improvements**:
    - Enhance sensor dependency to support diverse configurations and extend the applicability of the LIO-SAM on other IMUs and Lidars.

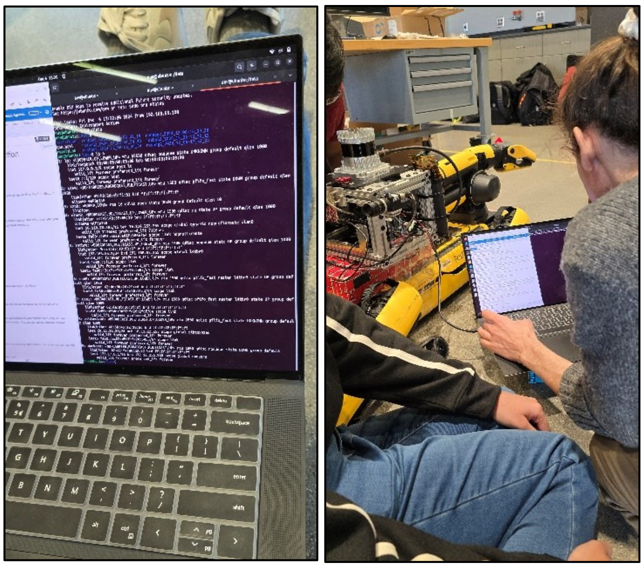


*fig-19: Learning & Recording session in High Bay*

**Conclusion**

The project highlights the potential of LIO-SAM for achieving high-precision mapping and localization in GPS-denied environments. Despite challenges in adapting the algorithm for custom hardware configurations, the progress underscores the importance of flexibility in SLAM solutions.

We thoroughly enjoyed the hands-on experience, learning opportunity, and working on Spot, so thanks to Prof. Hanu for giving us this opportunity and our TAs, Jasen and Vishnu for always being supportive and available. There are many things that we didn't explicitly explain, like all the other small errors related to Ubuntu that we encountered, how the spot SDK works, how we can extract the data from SDK, how we connect to spot and collect the data and how Parth showed us another way of doing the same and gRPC protocol. Because of the time constraints and also including that would make this more like a manual and less like a report. We think that facing all those challenges has taught us a lot, it is good that we had problems because, without challenges and errors, there is no learning.

Future work will focus on overcoming identified limitations and expanding testing, further validating the algorithm's robustness and real-world applicability of the Spot.

## References

1. T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping."
2. LIO-SAM GitHub Repository: https://github.com/TixiaoShan/LIO-SAM
3. https://ras.papercept.net/images/temp/IROS/files/0063.pdf
4. https://github.com/TixiaoShan/LIO-SAM/tree/ros2
5. https://bostondynamics.com/
6. https://dev.bostondynamics.com/docs/python/quickstart#system-requirements
7. https://github.com/bdaiinstitute/spot_ros2

**Our Drive link:** **https://drive.google.com/drive/folders/1ofWXLbhqu4m_-7foQu_30O7dn41ErS4w?usp=sharing**

**Gitlab: mistry.dhy**

**Final Project Report on: LIO-SAM with the Spot**

**by -> Group -7**

- **Dhyey Mistry**
- **Yash Wakde**
- **Adithya Rajendran**
- **Kevin Jason**