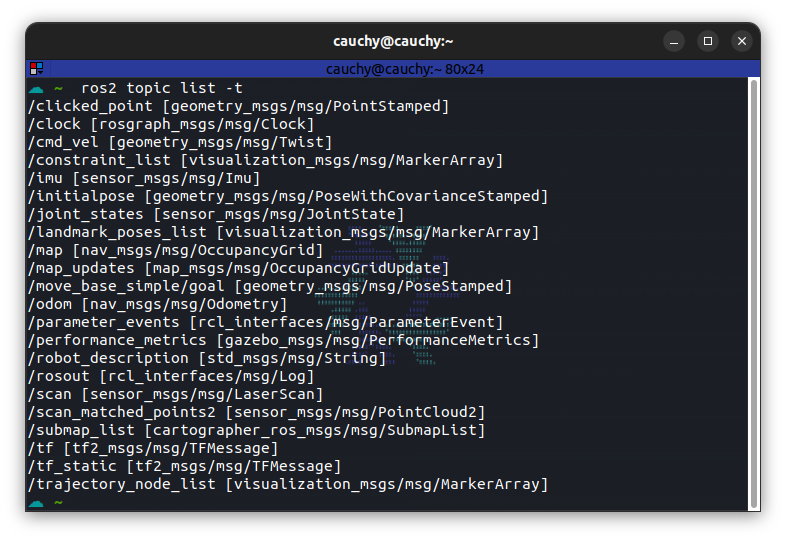
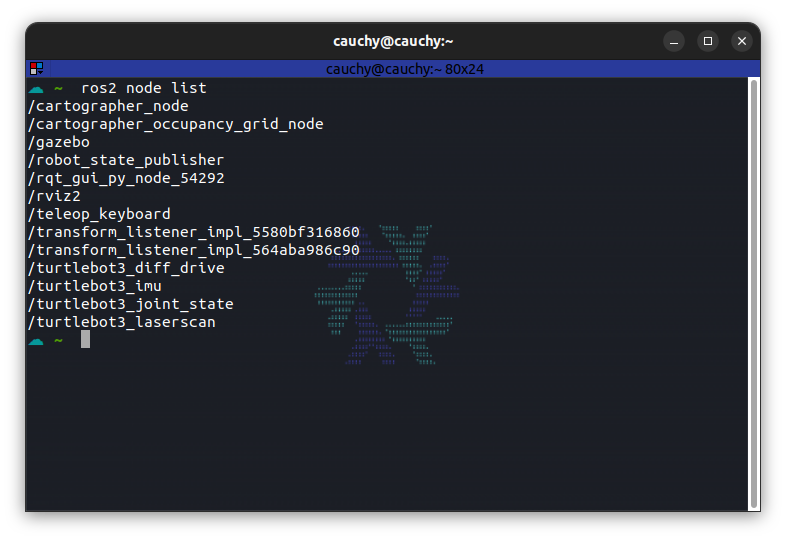
**\* SLAM**

– Topic list



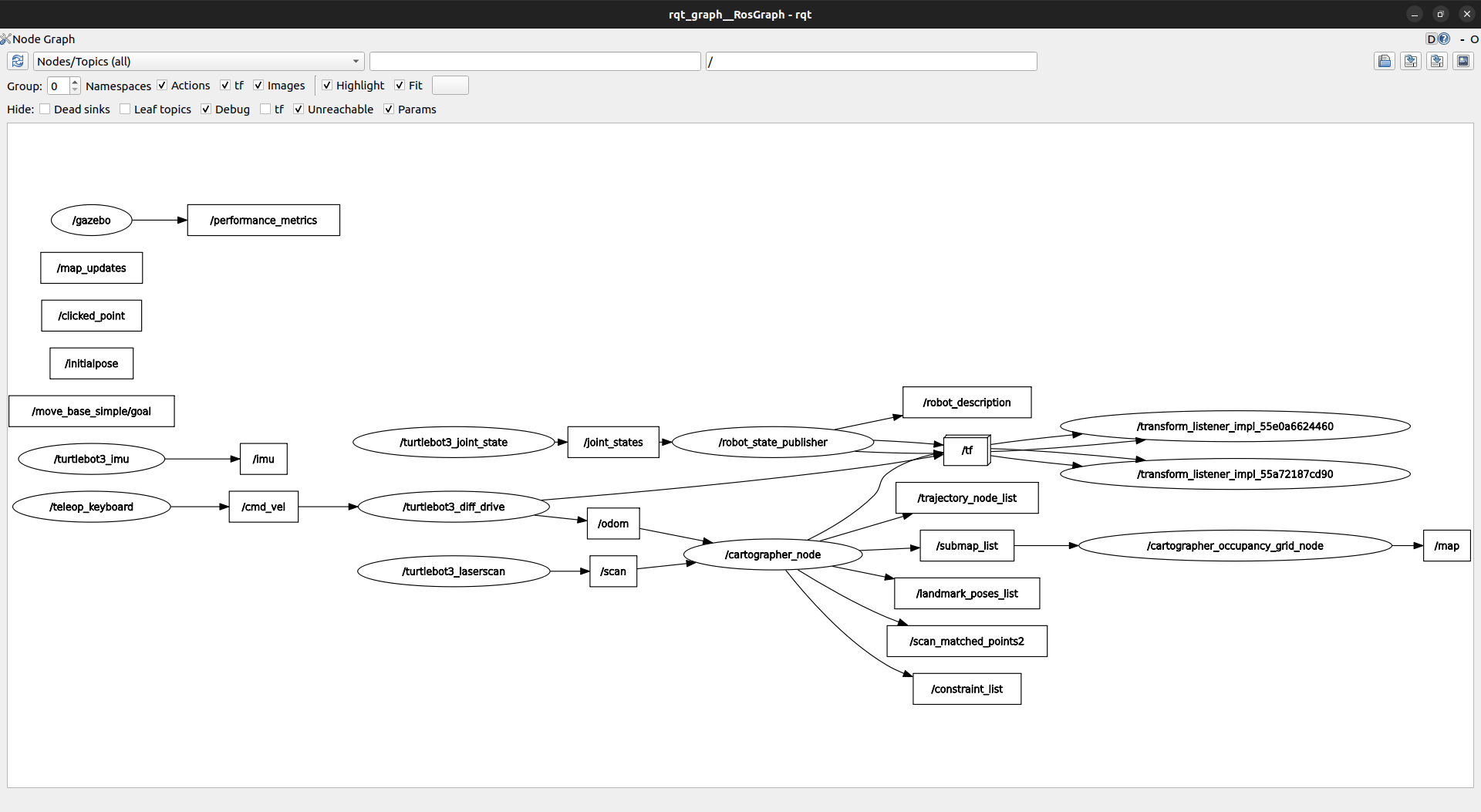
– Node list



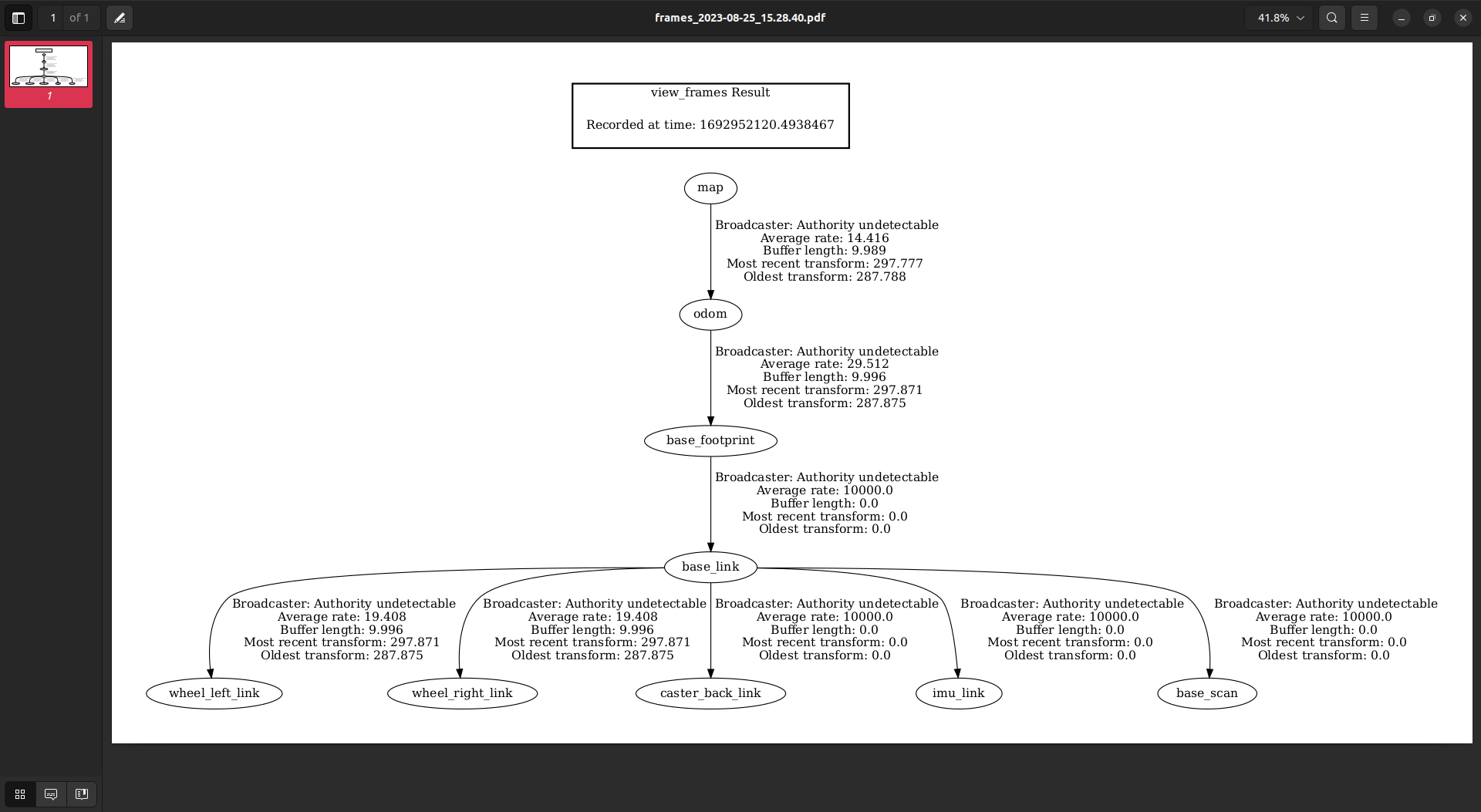
- Để xem Freq của từng topic:

| ros2 topic hz /topic |
| --- |

– rqt\_graph

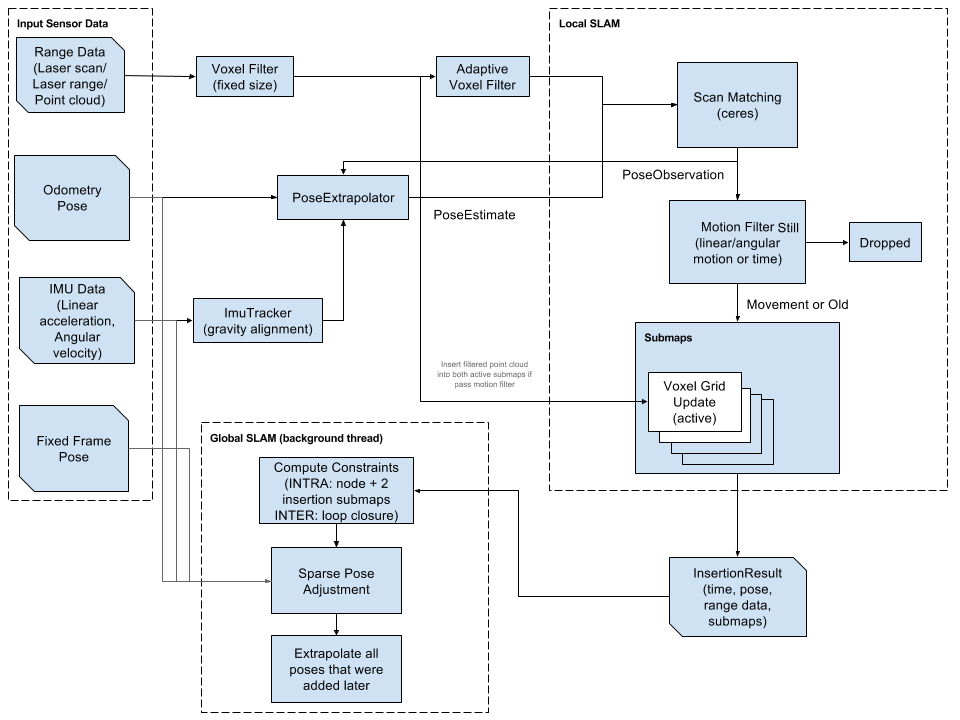


– tf2 tree



=> broadcaster ???

\*\* **Cartographer**



**– Input:**

* Range Data: dữ liệu từ cảm biến LiDAR.
* Odometry Pose: dữ liệu về sự di chuyển của robot và vj trí của robot theo thời gian (dữ liệu từ encoder...).
* IMU Data: dữ liệu từ IMU (gia tốc, tốc độ góc,..).
* Fixed Frame Pose: thông tin về vị trí và hướng của robot trong hệ tọa độ của khung cố định (vị trí ban đầu của robot...).

- Cartographer áp dụng một bộ lọc thông dải (bandpass filter) chỉ giữ lại các giá trị khoảng cách nằm trong một khoảng giữa một giá trị min và max cụ thể. Các giá trị min và max (mét) này nên được lựa chọn dựa trên các thông số kỹ thuật của robot và cảm biến của nó.

| TRAJECTORY\_BUILDER\_nD.min\_range  TRAJECTORY\_BUILDER\_nD.max\_range |
| --- |

- Cảm biến Lidar tạo ra một lượng lớn dữ liệu điểm, tạo ra dữ liệu rất dày đặc và khó quản lý → sử dụng voxel filter, chia dữ liệu điểm thành các khối (cubes) có kích thước cố định trong không gian 3D. Mỗi khối là một phần của không gian và chứa tất cả các điểm từ dữ liệu gốc nằm trong khối đó.

- Kích thước của các khối có thể được điều chỉnh để kiểm soát mức độ chi tiết của dữ liệu. Khi chọn kích thước khối lớn hơn, loại bỏ nhiều điểm hơn, và dữ liệu sẽ trở nên ít chi tiết hơn, nhưng xử lý sẽ nhanh hơn. Ngược lại, kích thước khối nhỏ hơn giữ lại nhiều chi tiết hơn, nhưng có thể làm tăng tải tính toán.

| TRAJECTORY\_BUILDER\_nD.voxel\_filter\_size |
| --- |

- Sau khi sử dụng voxel filter (fixed-size), Cartographer cũng áp dụng một adaptive voxel filter, xác định kích thước voxel tối ưu để đạt được một số lượng điểm mục tiêu.

| TRAJECTORY\_BUILDER\_nD.\*adaptive\_voxel\_filter.max\_length  TRAJECTORY\_BUILDER\_nD.\*adaptive\_voxel\_filter.min\_num\_points |
| --- |

- ImuTracker (gravity alignment) được sử dụng để xác định và căn chỉnh hướng trọng lực bằng dữ liệu từ IMU, giúp robot biết được hướng mặt đất trong không gian, điều này quan trọng để xác định vị trí tương đối của robot và xây dựng bản đồ.

| TRAJECTORY\_BUILDER\_2D.use\_imu\_data  TRAJECTORY\_BUILDER\_nD.imu\_gravity\_time\_constant |
| --- |

**– Local SLAM**

- Khi một scan đã được tạo ra và lọc từ nhiều dữ liệu khoảng cách, nó sẵn sàng để được sử dụng trong thuật toán SLAM cục bộ (Local SLAM). Local SLAM là quá trình xây dựng bản đồ và định vị trong thời gian thực trong một phạm vi cục bộ. Mục tiêu của Local SLAM là cập nhật bản đồ và vị trí của robot khi robot di chuyển trong một phạm vi nhỏ.

- Khi Local SLAM muốn chèn một scan mới vào bản đồ cục bộ hiện tại, nó thực hiện quá trình scan matching. Scan matching là quá trình tìm kiếm sự tương đồng giữa scan mới và scan đã lưu trữ trong submap hiện tại. Điều này giúp xác định vị trí tương đối của scan mới trong submap và cập nhật vị trí của robot.

- Submap là một phần của bản đồ tổng thể mà robot xây dựng khi di chuyển trong môi trường. Nó là một phần bản đồ cục bộ, thường là một vùng nhỏ của môi trường mà robot đã quét và xây dựng bản đồ trong khoảng thời gian cụ thể. Submap có thể hiểu đơn giản là một phần của bản đồ toàn cục của môi trường, chứa thông tin về các vật thể, cấu trúc, và đặc điểm trong phạm vi của nó.

- Local SLAM sử dụng Pose Extrapolator, là một thành phần trong hệ thống Cartographer, và nhiệm vụ chính của nó là dự đoán vị trí và hướng của robot dựa trên thông tin từ các cảm biến khác để dự đoán vị trí tiếp theo nên được chèn vào submap.

- Để tránh chèn quá nhiều scan vào mỗi submap, sau khi biến đổi giữa hai scan được tìm thấy bởi scan matcher, nó đi qua một bộ lọc chuyển động (motion filter). Một scan sẽ bị loại bỏ (dropped) nếu phép chuyển động dẫn đến nó không được coi là đủ đáng kể. Một scan chỉ được chèn vào submap hiện tại nếu phép chuyển động của nó vượt qua một ngưỡng cố định về khoảng cách, góc, hoặc thời gian.

| TRAJECTORY\_BUILDER\_nD.motion\_filter.max\_time\_seconds  TRAJECTORY\_BUILDER\_nD.motion\_filter.max\_distance\_meters  TRAJECTORY\_BUILDER\_nD.motion\_filter.max\_angle\_radians |
| --- |

- Một submap được coi là hoàn chỉnh khi hệ thống Local SLAM đã nhận được một lượng dữ liệu khoảng cách cụ thể. Local SLAM thường bị trôi theo thời gian, và hệ thống Global SLAM được sử dụng để khắc phục sự trôi này. Submap phải đủ nhỏ để đảm bảo rằng sự trôi xảy ra bên trong chúng nhỏ hơn độ phân giải, từ đó chúng được coi là chính xác ở mức cục bộ. Tuy nhiên, chúng cũng cần đủ lớn để có thể được phân biệt đối với loop closure để hoạt động một cách chính xác.

-

| TRAJECTORY\_BUILDER\_nD.submaps.num\_range\_data |
| --- |

**– Global SLAM**

- Trong quá trình xây dựng bản đồ, local SLAM tạo ra các submap cục bộ. Tuy nhiên, để tạo ra một bản đồ toàn cầu liên tục và chính xác, cần có một quá trình tối ưu hóa toàn cầu chạy ẩn sau nền. Quá trình này giúp sắp xếp lại các submap để chúng hợp thành một bản đồ toàn cầu có tính nhất quán.

- Tối ưu hóa toàn cầu thường được thực hiện theo từng "batches" dữ liệu. Mỗi khi một số lượng cụ thể các "trajectory nodes" (điểm trong đường đi của robot) đã được thêm vào, quá trình tối ưu hóa sẽ được kích hoạt. Tùy thuộc vào tần suất cần chạy nó, có thể điều chỉnh kích thước của các batches này.

| POSE\_GRAPH.optimize\_every\_n\_nodes |
| --- |

- Trong Global SLAM, hệ thống sử dụng một phương pháp được gọi là "GraphSLAM." Đây là một quá trình tối ưu hóa đồ thị vị trí, trong đó các ràng buộc (constraints) được xây dựng giữa các nút (nodes) và các submap. Ràng buộc này có thể được xem như những dây thừng nhỏ kết nối tất cả các nút lại với nhau.

- Quá trình sparse pose adjustment thực hiện để điều chỉnh và cải thiện đồ thị các ràng buộc. Nó có vai trò giữ chặt các ràng buộc này lại với nhau để tạo ra một mạng (net) hoàn chỉnh. Kết quả cuối cùng của quá trình này được gọi là "đồ thị vị trí" (pose graph), nó biểu thị vị trí của robot và các submap một cách đồng nhất và nhất quán.

+ Ràng buộc không toàn cầu (còn được gọi là ràng buộc trong submap, hoặc intra submaps constraints) được tạo tự động giữa các nút mà theo sau nhau một cách gần nhau trên một đường đi. Một cách trực quan, những "sợi dây không toàn cầu" này giữ cho cấu trúc cục bộ của đường đi duy trì tính nhất quán.

+ Ràng buộc toàn cầu (còn được gọi là oop closure constraints hoặc inter submaps constraints) được tìm kiếm định kỳ giữa một submap mới và các nút trước đó mà được coi là "gần đủ" trong không gian (nằm trong một cửa sổ tìm kiếm cụ thể) và có độ tương quan mạnh (phù hợp tốt khi thực hiện scan matching). Một cách trực quan, những "ràng buộc toàn cầu" này giúp xác định vị trí của robot và cấu trúc lặp lại trong bản đồ toàn cầu.

| POSE\_GRAPH.constraint\_builder.max\_constraint\_distance  POSE\_GRAPH.fast\_correlative\_scan\_matcher.linear\_search\_window  POSE\_GRAPH.fast\_correlative\_scan\_matcher\_3d.linear\_xy\_search\_window  POSE\_GRAPH.fast\_correlative\_scan\_matcher\_3d.linear\_z\_search\_window  POSE\_GRAPH.fast\_correlative\_scan\_matcher\*.angular\_search\_window |
| --- |

- Để giới hạn số lượng ràng buộc (và tính toán), Cartographer chỉ xem xét một tập con được lấy mẫu từ tất cả các nút gần nhau để xây dựng các ràng buộc. Điều này được điều khiển bằng một hằng số tỷ lệ mẫu (sampling ratio). Việc lấy mẫu quá ít nút có thể dẫn đến việc bỏ lỡ các ràng buộc và các lần đóng vòng không hiệu quả. Lấy mẫu quá nhiều nút sẽ làm chậm quá trình SLAM toàn cầu và ngăn chặn việc đóng vòng thời gian thực.

| POSE\_GRAPH.constraint\_builder.sampling\_ratio |
| --- |

- Khi Cartographer chạy quá trình tối ưu hóa, nó sử dụng Ceres để điều chỉnh lại vị trí của các submap dựa trên nhiều dư thừa (residual). Dư thừa được tính bằng cách sử dụng các hàm chi phí có trọng số. Quá trình tối ưu hóa toàn cầu sử dụng các hàm chi phí để xem xét nhiều nguồn dữ liệu, bao gồm các ràng buộc toàn cầu (loop closure), ràng buộc không toàn cầu (matcher), đo đạc gia tốc và quay từ IMU, các ước tính vị trí thô từ Local SLAM, dữ liệu odometry hoặc một khung cố định như hệ thống GPS. Trọng số và các tùy chọn của Ceres có thể được cấu hình theo cách được mô tả trong phần Local SLAM.

| POSE\_GRAPH.constraint\_builder.loop\_closure\_translation\_weight  POSE\_GRAPH.constraint\_builder.loop\_closure\_rotation\_weight  POSE\_GRAPH.matcher\_translation\_weight  POSE\_GRAPH.matcher\_rotation\_weight  POSE\_GRAPH.optimization\_problem.\*\_weight  POSE\_GRAPH.optimization\_problem.ceres\_solver\_options |
| --- |

- Khi quỹ đạo hoàn thành, Cartographer thực hiện một tối ưu hóa toàn cầu mới với số lần lặp thường là nhiều hơn so với các tối ưu hóa toàn cầu trước đó. Điều này được thực hiện để làm cho kết quả cuối cùng của Cartographer trở nên hoàn thiện hơn và thường không cần phải thời gian thực, vì vậy việc có một số lần lặp lớn thường là lựa chọn đúng.

| POSE\_GRAPH.max\_num\_final\_iterations |
| --- |

**\*\* Nav 2**

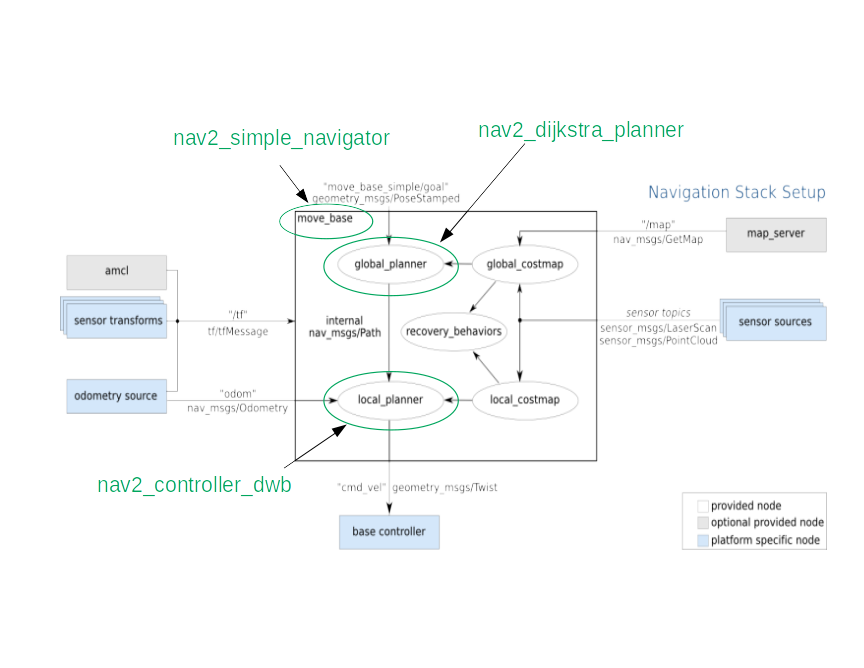
**\* Simulation**

– Topic list:

| /amcl/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /amcl\_pose [geometry\_msgs/msg/PoseWithCovarianceStamped]  /behavior\_server/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /bond [bond/msg/Status]  /bt\_navigator/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /clicked\_point [geometry\_msgs/msg/PointStamped]  /clock [rosgraph\_msgs/msg/Clock]  /cmd\_vel [geometry\_msgs/msg/Twist]  /cmd\_vel\_nav [geometry\_msgs/msg/Twist]  /controller\_server/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /cost\_cloud [sensor\_msgs/msg/PointCloud2]  /diagnostics [diagnostic\_msgs/msg/DiagnosticArray]  /downsampled\_costmap [nav\_msgs/msg/OccupancyGrid]  /downsampled\_costmap\_updates [map\_msgs/msg/OccupancyGridUpdate]  /evaluation [dwb\_msgs/msg/LocalPlanEvaluation]  /global\_costmap/clearing\_endpoints [sensor\_msgs/msg/PointCloud2]  /global\_costmap/costmap [nav\_msgs/msg/OccupancyGrid]  /global\_costmap/costmap\_raw [nav2\_msgs/msg/Costmap]  /global\_costmap/costmap\_updates [map\_msgs/msg/OccupancyGridUpdate]  /global\_costmap/footprint [geometry\_msgs/msg/Polygon]  /global\_costmap/global\_costmap/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /global\_costmap/published\_footprint [geometry\_msgs/msg/PolygonStamped]  /global\_costmap/voxel\_grid [nav2\_msgs/msg/VoxelGrid]  /global\_costmap/voxel\_marked\_cloud [sensor\_msgs/msg/PointCloud2]  /goal\_pose [geometry\_msgs/msg/PoseStamped]  /imu [sensor\_msgs/msg/Imu]  /initialpose [geometry\_msgs/msg/PoseWithCovarianceStamped]  /joint\_states [sensor\_msgs/msg/JointState]  /local\_costmap/clearing\_endpoints [sensor\_msgs/msg/PointCloud2]  /local\_costmap/costmap [nav\_msgs/msg/OccupancyGrid]  /local\_costmap/costmap\_raw [nav2\_msgs/msg/Costmap]  /local\_costmap/costmap\_updates [map\_msgs/msg/OccupancyGridUpdate]  /local\_costmap/footprint [geometry\_msgs/msg/Polygon]  /local\_costmap/local\_costmap/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /local\_costmap/published\_footprint [geometry\_msgs/msg/PolygonStamped]  /local\_costmap/voxel\_grid [nav2\_msgs/msg/VoxelGrid]  /local\_costmap/voxel\_marked\_cloud [sensor\_msgs/msg/PointCloud2]  /local\_plan [nav\_msgs/msg/Path]  /map [nav\_msgs/msg/OccupancyGrid]  /map\_server/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /map\_updates [map\_msgs/msg/OccupancyGridUpdate]  /marker [visualization\_msgs/msg/MarkerArray]  /mobile\_base/sensors/bumper\_pointcloud [sensor\_msgs/msg/PointCloud2]  /odom [nav\_msgs/msg/Odometry]  /parameter\_events [rcl\_interfaces/msg/ParameterEvent]  /particle\_cloud [nav2\_msgs/msg/ParticleCloud]  /performance\_metrics [gazebo\_msgs/msg/PerformanceMetrics]  /plan [nav\_msgs/msg/Path]  /plan\_smoothed [nav\_msgs/msg/Path]  /planner\_server/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /received\_global\_plan [nav\_msgs/msg/Path]  /robot\_description [std\_msgs/msg/String]  /rosout [rcl\_interfaces/msg/Log]  /scan [sensor\_msgs/msg/LaserScan]  /smoother\_server/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /speed\_limit [nav2\_msgs/msg/SpeedLimit]  /tf [tf2\_msgs/msg/TFMessage]  /tf\_static [tf2\_msgs/msg/TFMessage]  /transformed\_global\_plan [nav\_msgs/msg/Path]  /velocity\_smoother/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /waypoint\_follower/transition\_event [lifecycle\_msgs/msg/TransitionEvent]  /waypoints [visualization\_msgs/msg/MarkerArray] |
| --- |

– Node list:

| /amcl  /behavior\_server  /bt\_navigator  /bt\_navigator\_navigate\_through\_poses\_rclcpp\_node  /bt\_navigator\_navigate\_to\_pose\_rclcpp\_node  /controller\_server  /gazebo  /global\_costmap/global\_costmap  /launch\_ros\_6871  /lifecycle\_manager\_localization  /lifecycle\_manager\_navigation  /local\_costmap/local\_costmap  /map\_server  /nav2\_container  /planner\_server  /robot\_state\_publisher  /rviz2  /rviz\_navigation\_dialog\_action\_client  /smoother\_server  /transform\_listener\_impl\_55783cbf6ec0  /transform\_listener\_impl\_5621ee79ece0  /transform\_listener\_impl\_7f7a04007310  /transform\_listener\_impl\_7f7a08002cc0  /transform\_listener\_impl\_7f7a2c004a00  /transform\_listener\_impl\_7f7a340024a0  /turtlebot3\_diff\_drive  /turtlebot3\_imu  /turtlebot3\_joint\_state  /turtlebot3\_laserscan  /velocity\_smoother  /waypoint\_follower |
| --- |



- Các package đã đổi:

+ amcl: Ported to nav2\_amcl

+ map\_server: Ported to nav2\_map\_server

+ nav2\_planner: Replaces global\_planner

+ nav2\_controller: Replaces local\_planner

+ Navfn: Ported to nav2\_navfn\_planner

+ DWB: Replaces DWA and ported to ROS 2 under nav2\_dwb\_controller metapackage

+ nav\_core: Ported as nav2\_core with updates to interfaces

+ costmap\_2d: Ported as nav2\_costmap\_2d

- Những package mới:

+ nav2\_bt\_navigator: thay thế move\_base

+ nav2\_lifecycle\_manager: xử lý vòng đời chương trình máy chủ

+ nav2\_waypoint\_follower: có thể lấy nhiều điểm để thực hiện một nhiệm vụ phức tạp

+ nav2\_system\_tests: bộ test tích hợp cho CI và hướng dẫn cơ bản về mô phỏng

+ nav2\_rviz\_plugins: Một plugin rviz để điều khiển các máy chủ Navigation2, ra lệnh, hủy và điều hướng

+ navigation2\_behavior\_trees: các trình bao bọc cho thư viện behavior trees để gọi các máy chủ hành động ROS cùng với nhiều máy chủ khác.

