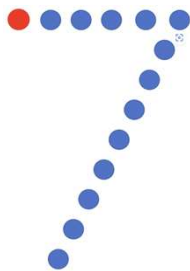




# SEVEN

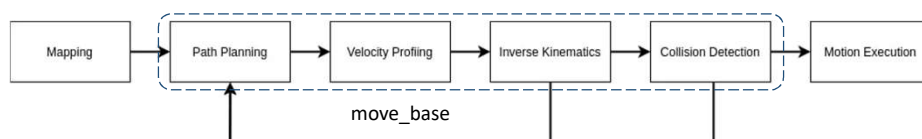
## Autonomous Exploration



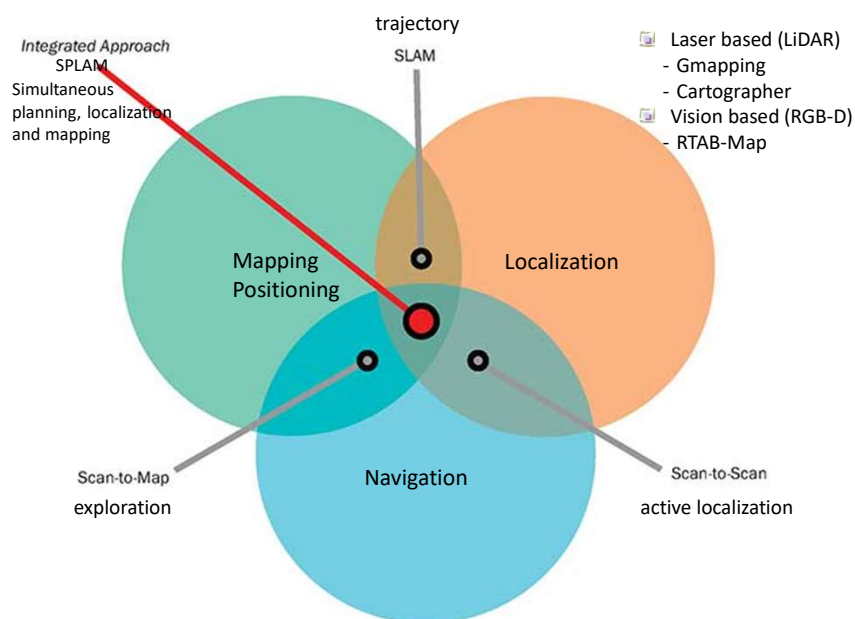
1



## SLAM (simultaneous localization and mapping)

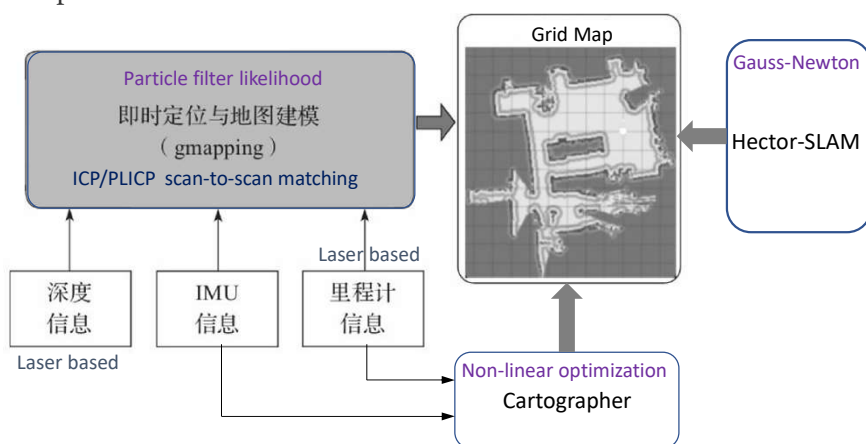


2



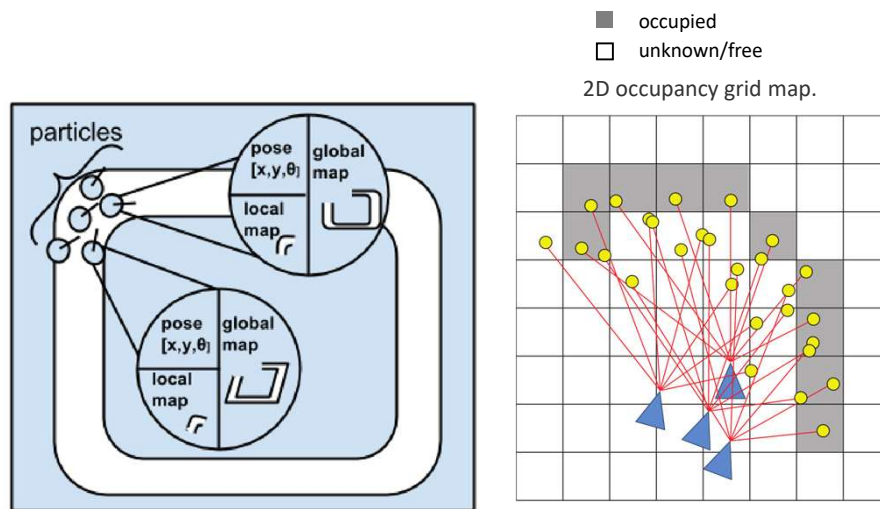
3

The purpose of the LiDAR odometry is to produce a local map by creating an estimate of the motion between two neighboring point cloud frames



## Laser-based SLAM

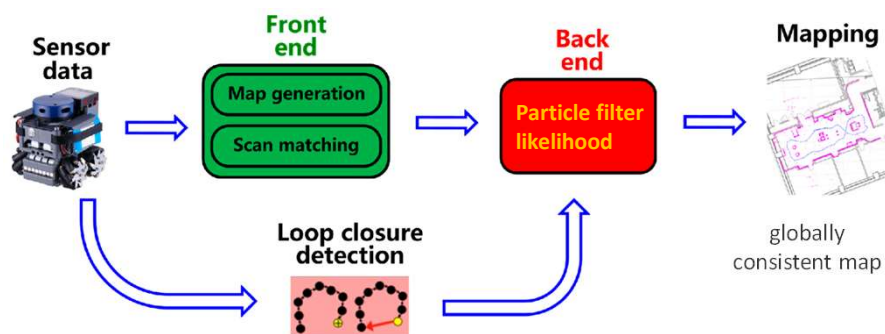
4



## Grid-based scan-to-scan matching

5

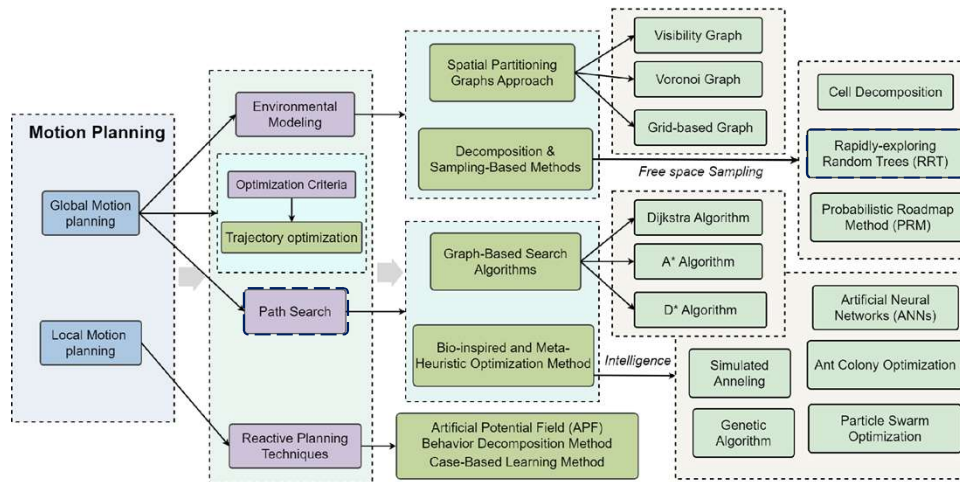
Global data association corrects cumulative mistakes by recognising if the robot has reached the location it has arrived at in the historical instant



## Combined particle filter and scan-matching SLAM

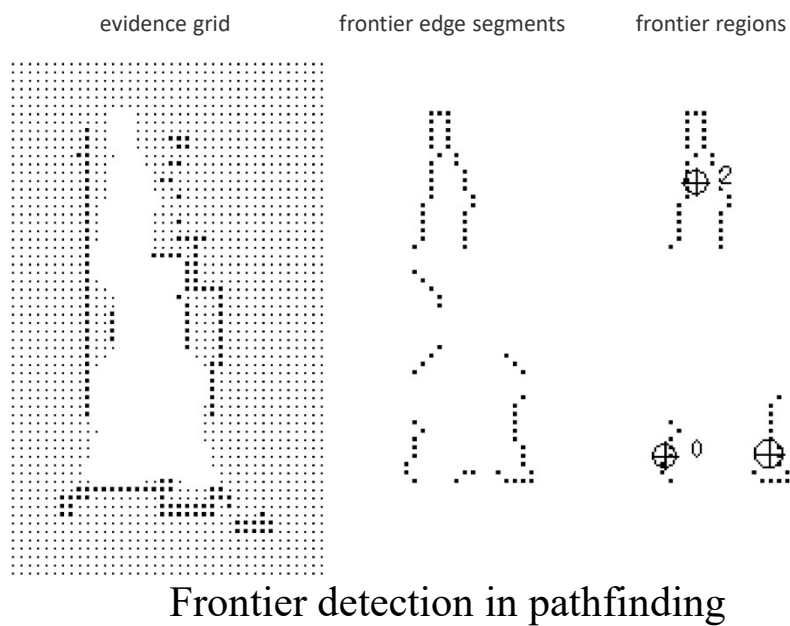
6



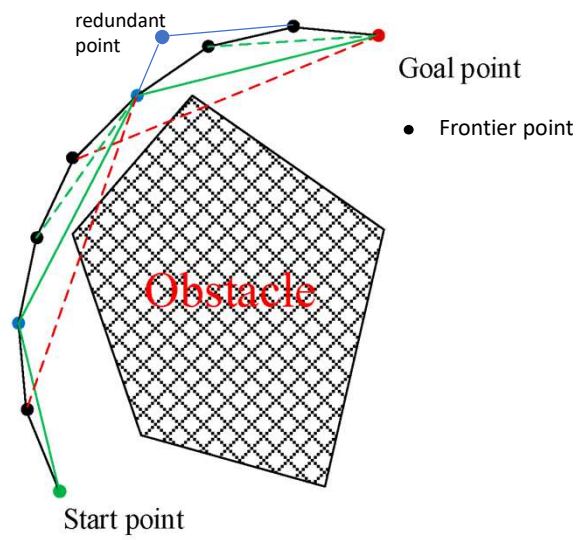


## Motion planning techniques in robotics

9

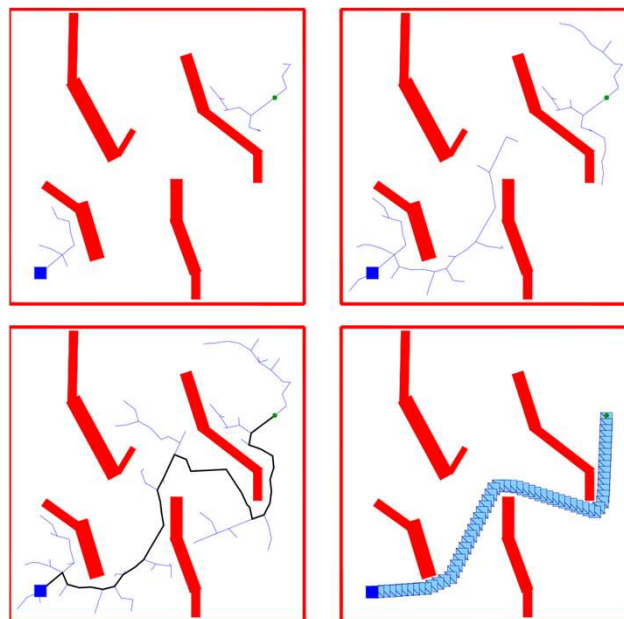


10



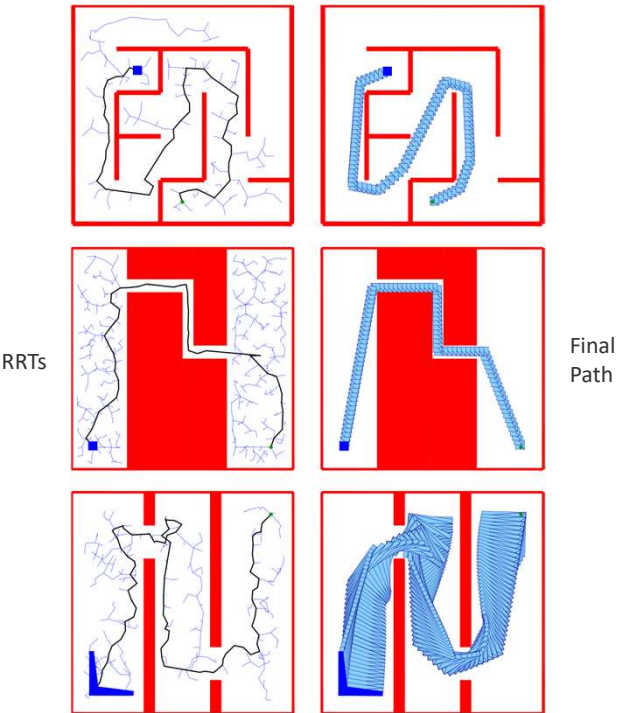
Frontier detection in RRT

11



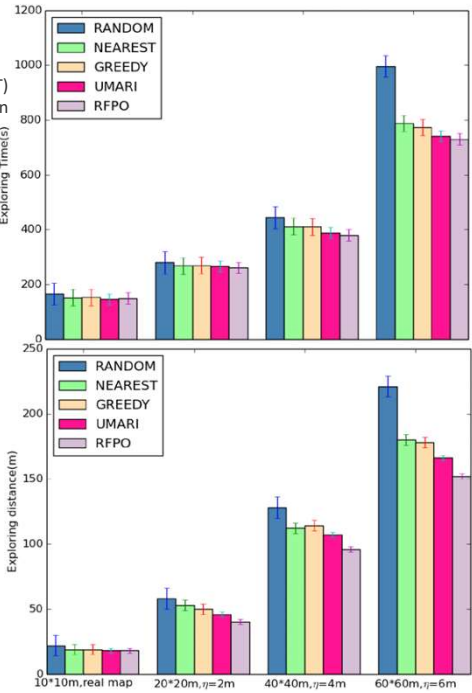
RRT pathfinding

12



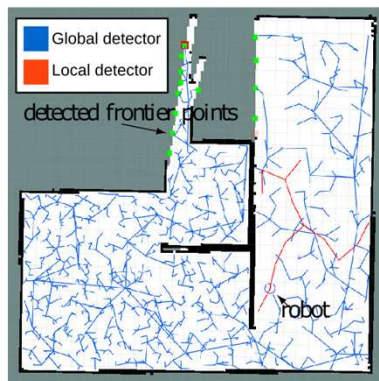
13

Umari and Mukhopadhyay (RRT)  
random frontier points' optimization



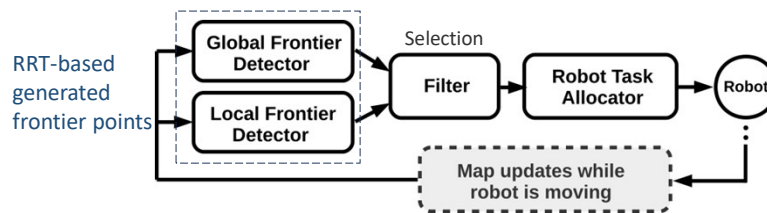
$\eta$  represents the growth rate  
of the tree

14



selection of the frontier point with the highest evaluation value (information gain and exploration cost) - the key to efficient exploration

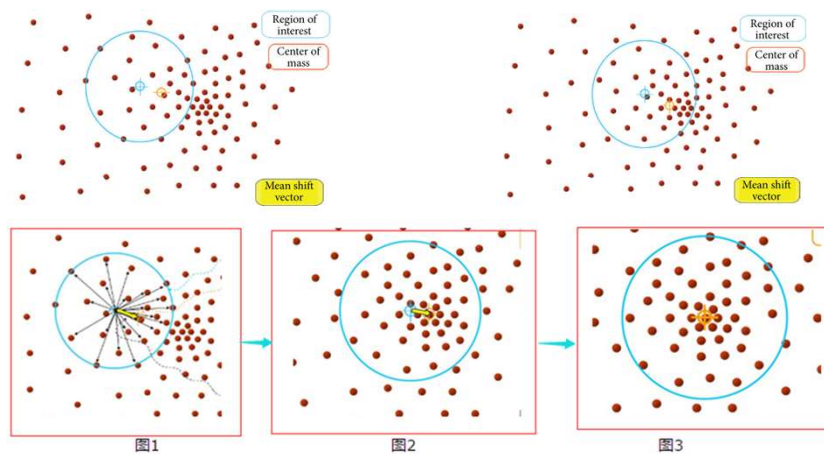
path planning to the selected clustered frontier points (frontier region) from the filter module



## RRT autonomous exploration

15

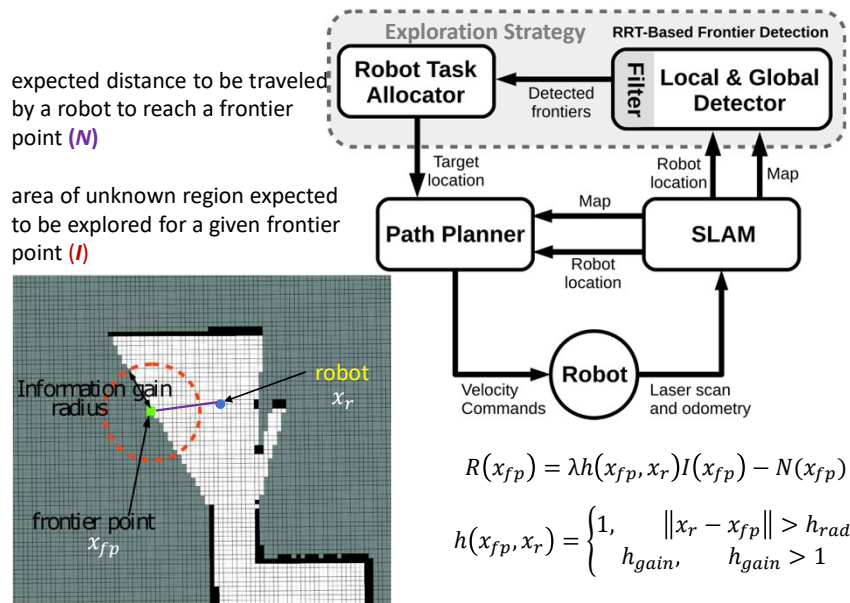
The filter module clusters the frontier points with mean shift clustering algorithm and stores them. The module also deletes invalid and old frontier points.



## Sampling mean shift algorithm

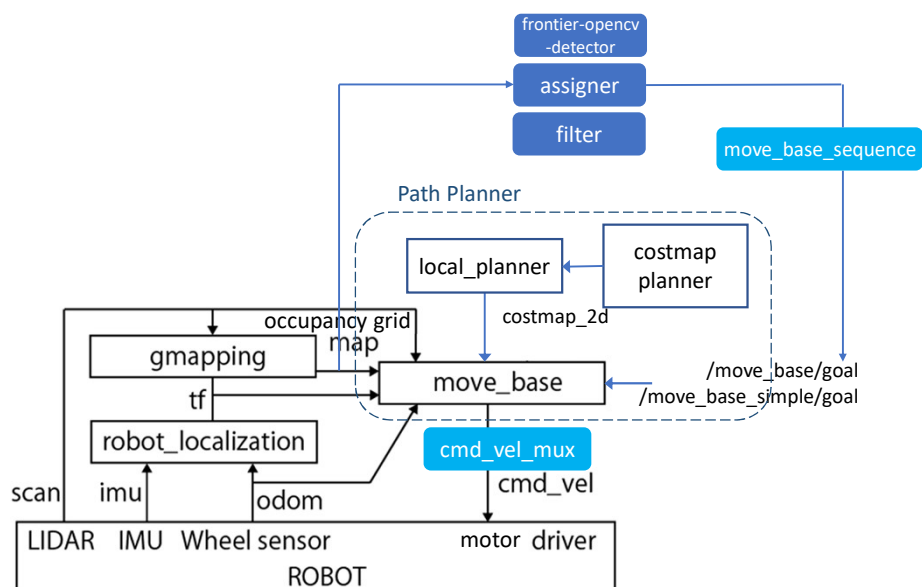
16





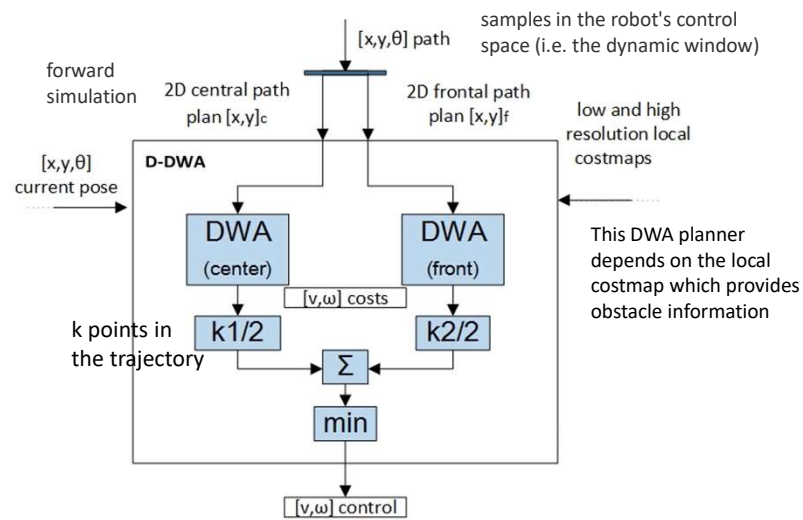
## RRT autonomous exploration

17



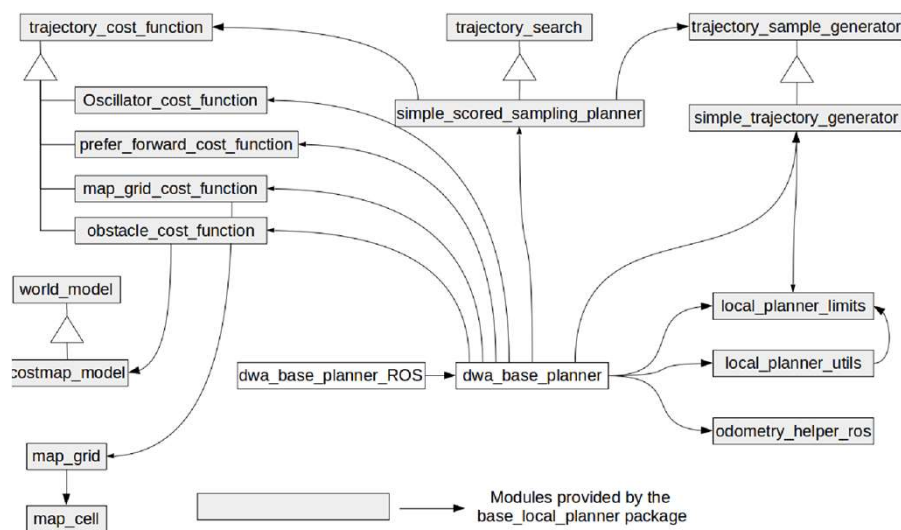
## RRT autonomous exploration

18



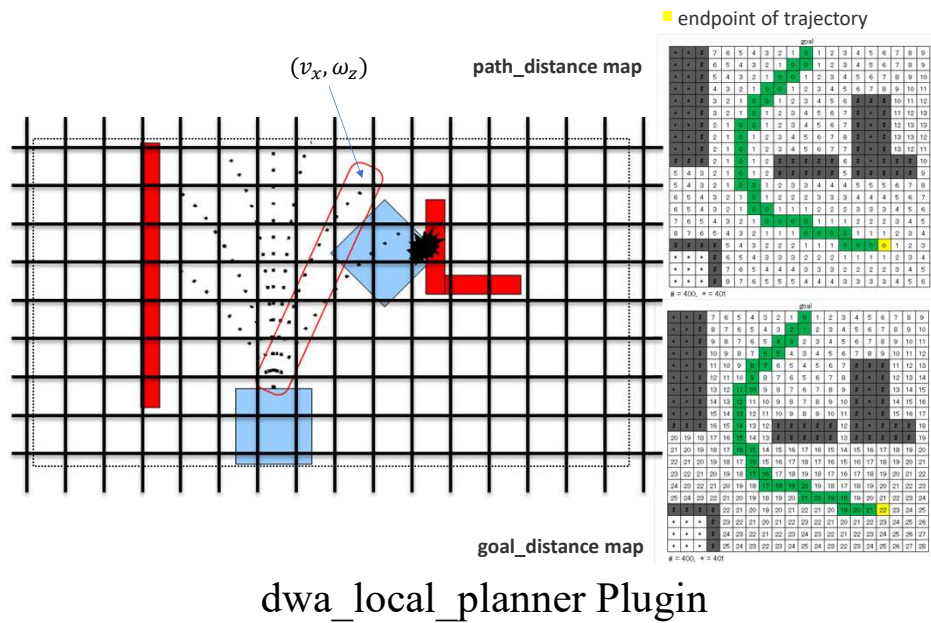
## Dynamic Window Approach local planner

19



## dwa\_local\_planner Plugin

20



21

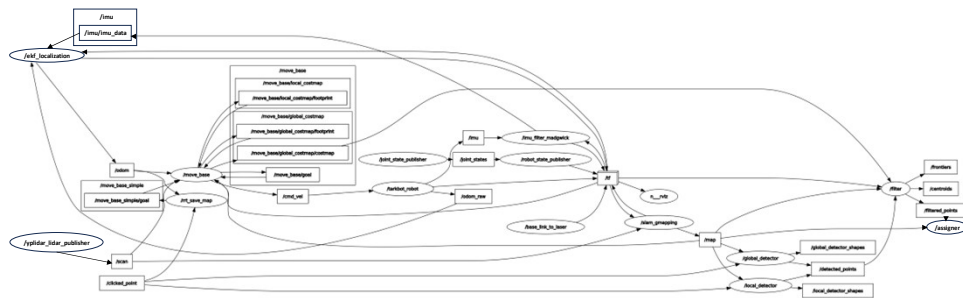
DWA maximizes an objective function that depends on

- (1) the progress to the target,
- (2) clearance from obstacles, and
- (3) forward velocity to produce the optimal velocity pair.

cost = path distance bias \* (distance(m) to path from the endpoint of the trajectory)  
 + goal distance bias \* (distance(m) to local goal from the endpoint of the trajectory)  
 + occdist scale \* (maximum obstacle cost along the trajectory in obstacle cost (0-254))

dwa\_local\_planner Plugin

22



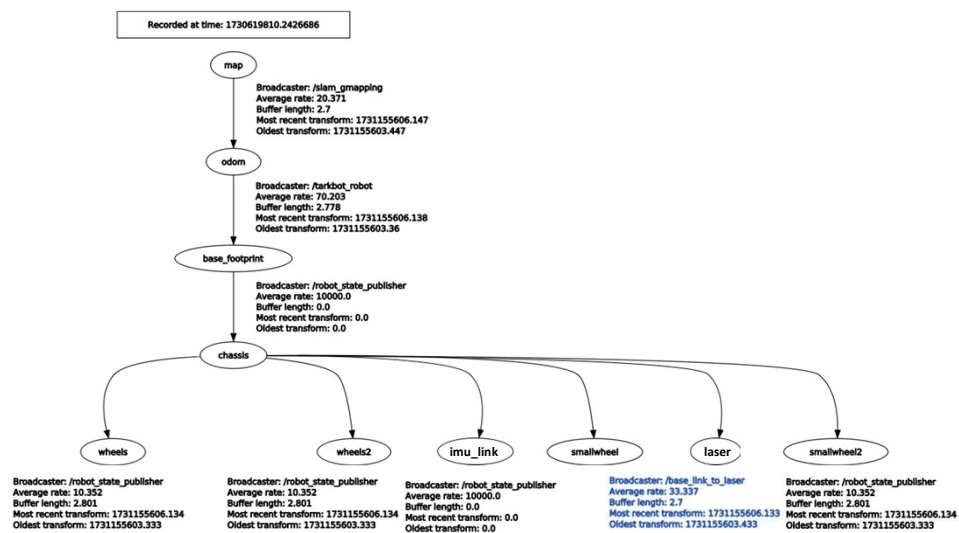
```

* /move_base/DWAPlannerROS/global_plan [nav_msgs/Path]
* /move_base/DWAPlannerROS/local_plan [nav_msgs/Path]
* /move_base/DWAPlannerROS/cost_cloud [sensor_msgs/PointCloud2]
* /move_base/DWAPlannerROS/trajectory_cloud [sensor_msgs/PointCloud2]

* /move_base/NavfnROS/plan [nav_msgs/Path]

```

23



24

