Navigation Illumination

Shedding Light on the ROS Navstack

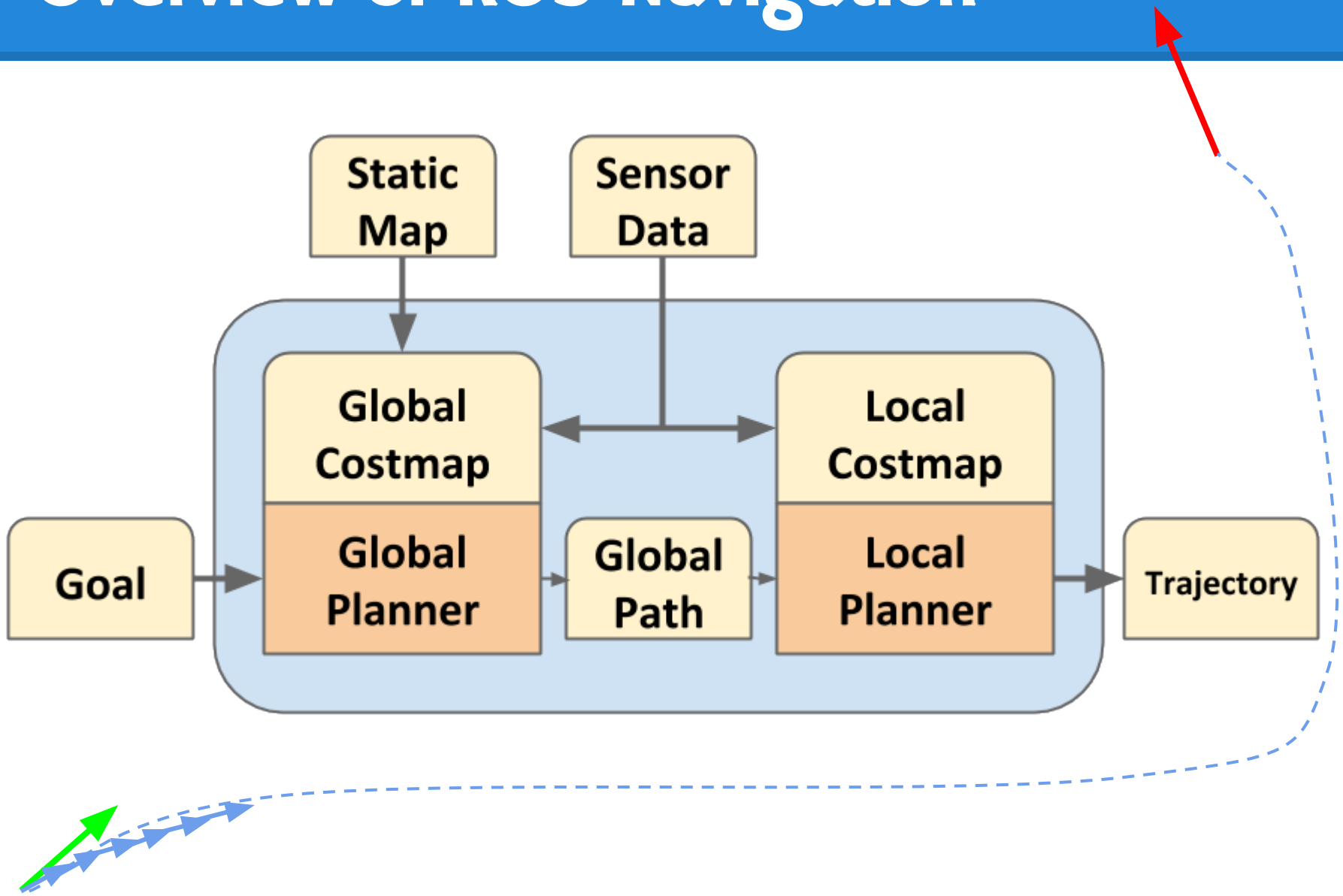
David V. Lu!!

Robots Using ROS

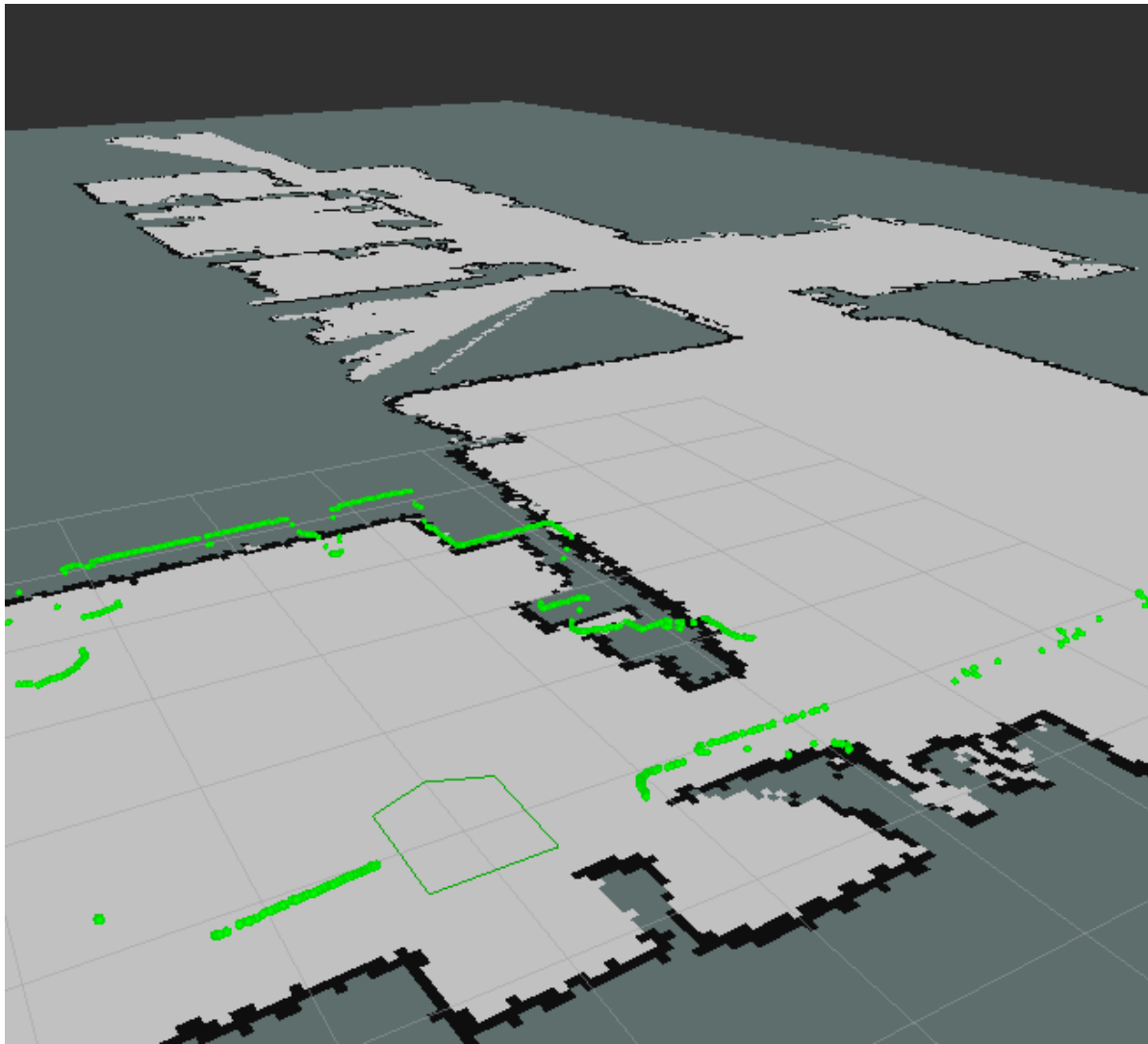
Cornerstone of ROS Open Source Platform

Dozens of Supported Hardwares

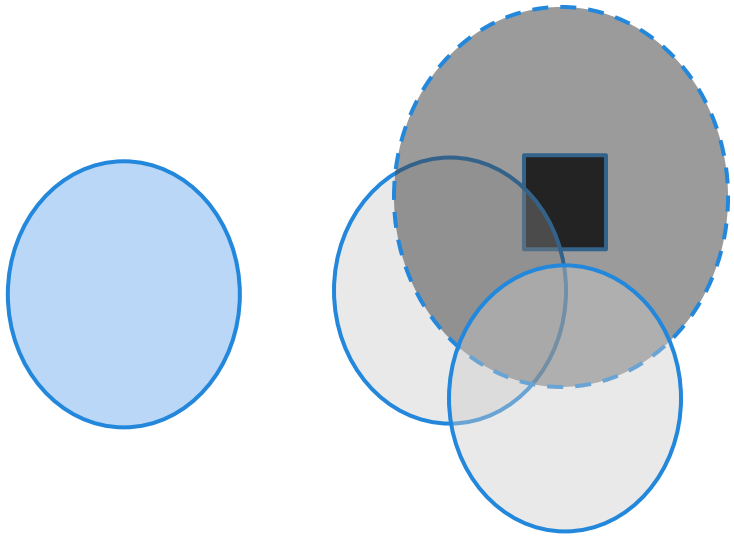
Overview of ROS Navigation

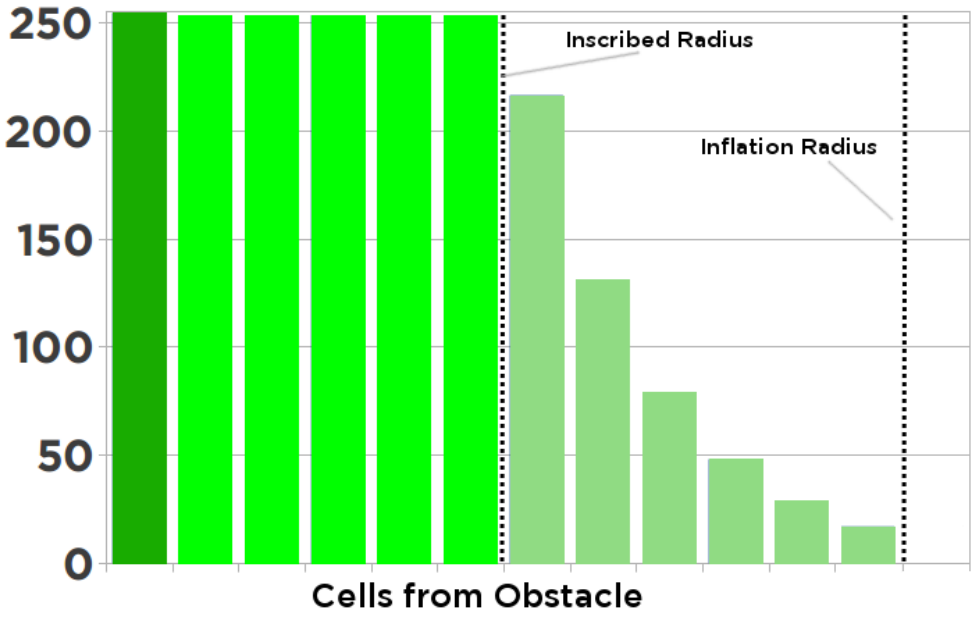


Costmap Data Sources

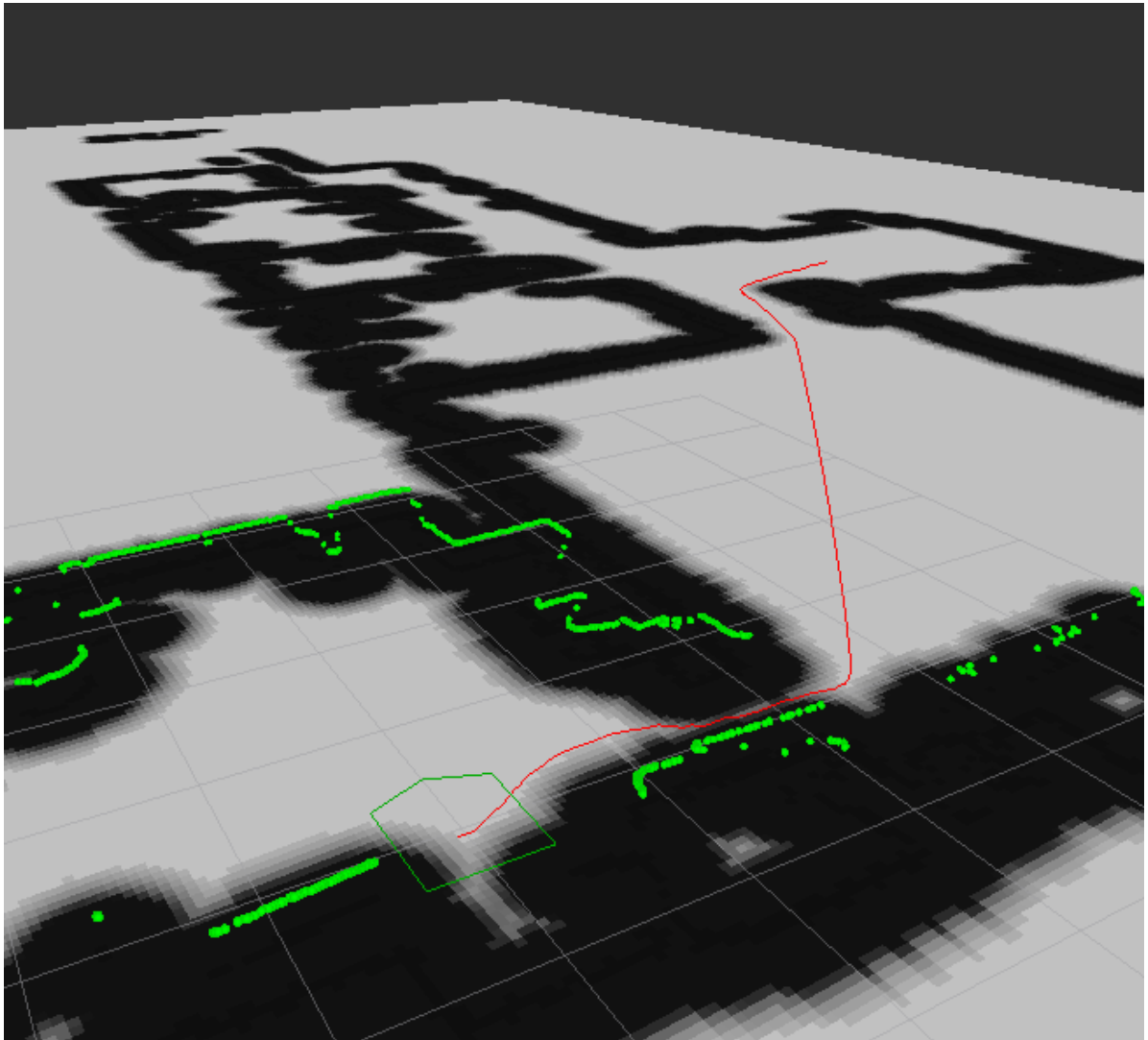


Obstacle Inflation



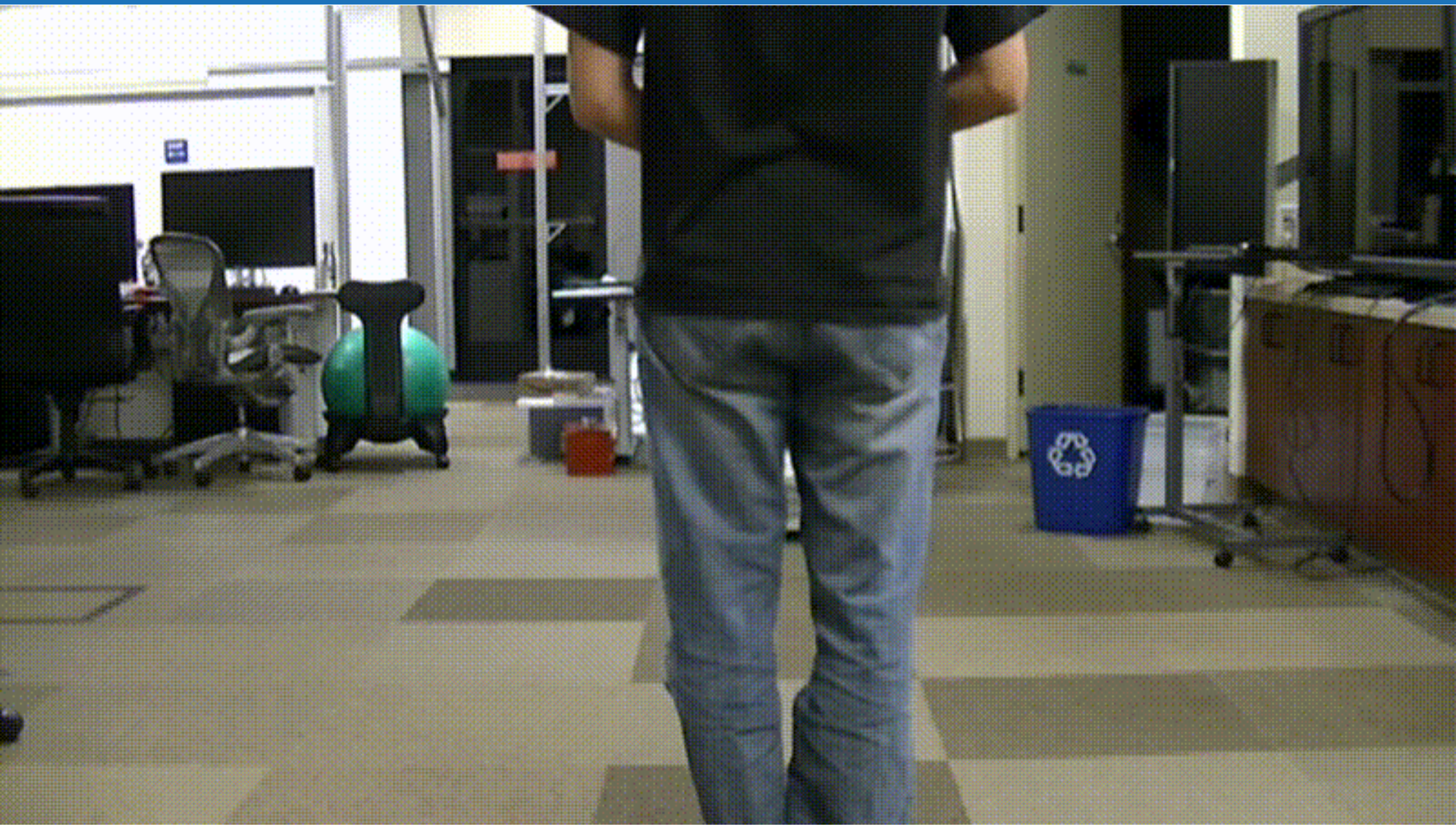


Global Costmap and Plan



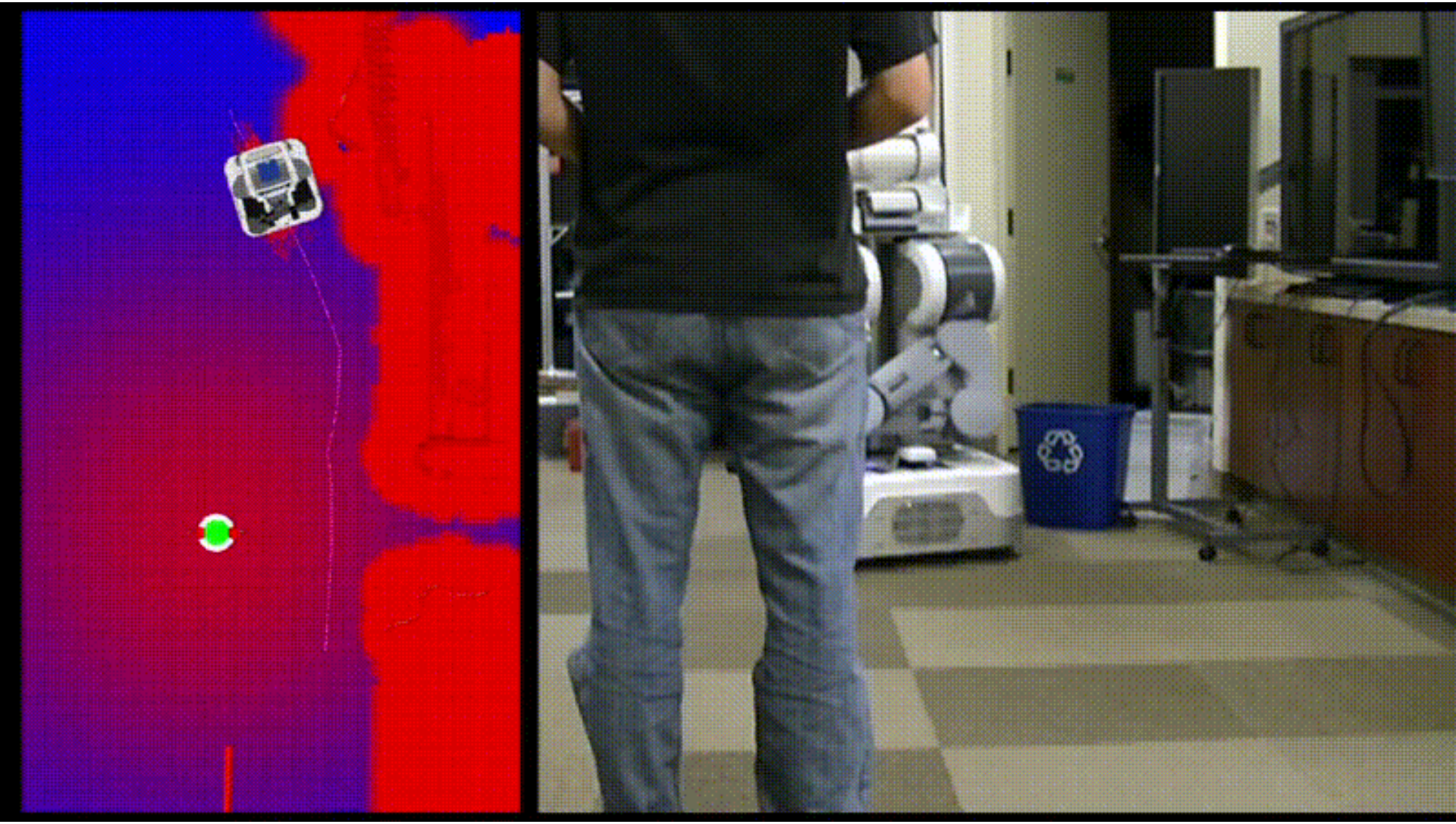

ROS
PR2
Willow Garage

Standard Navigation

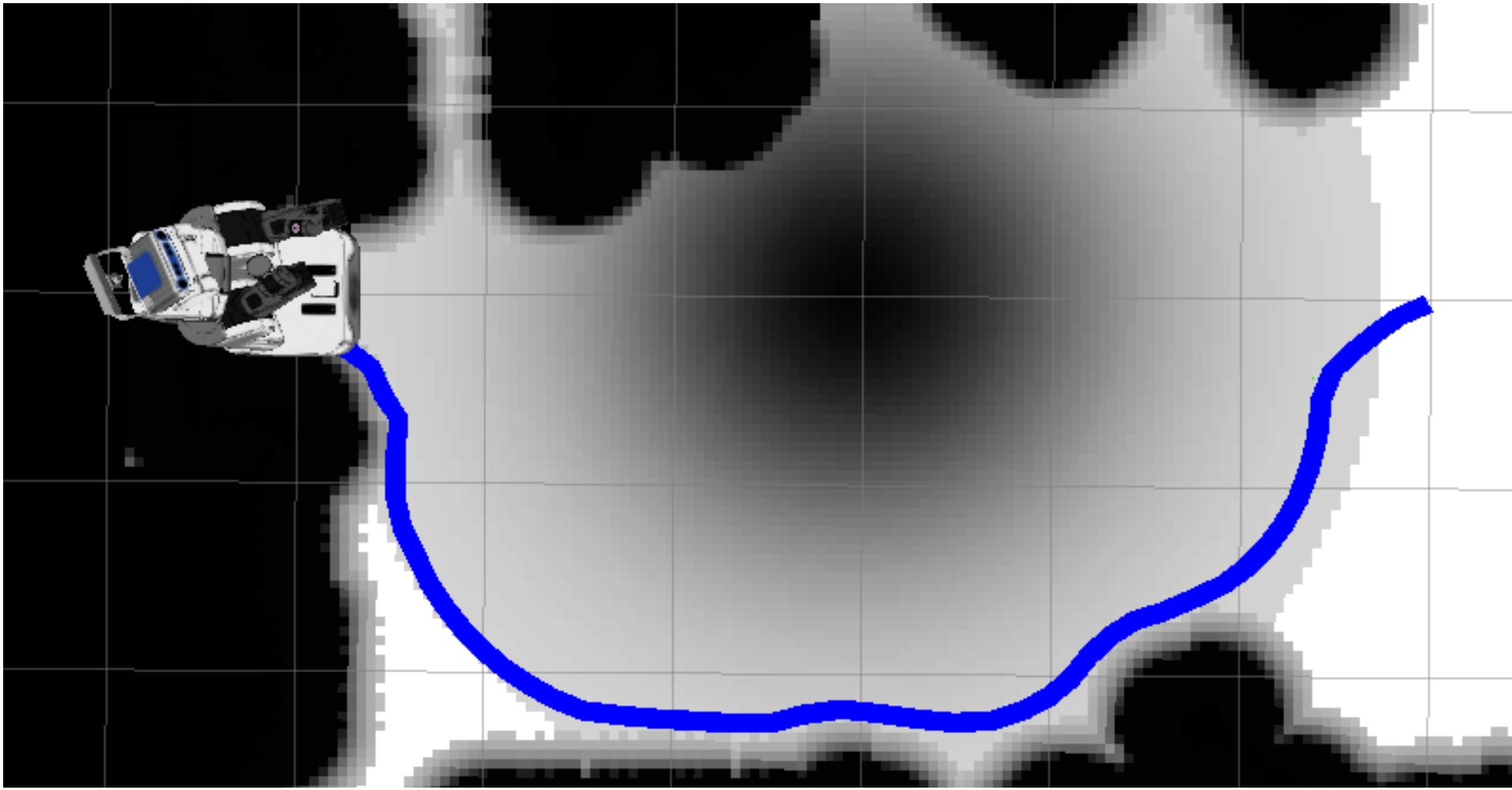



ROS
PR2
Willow Garage

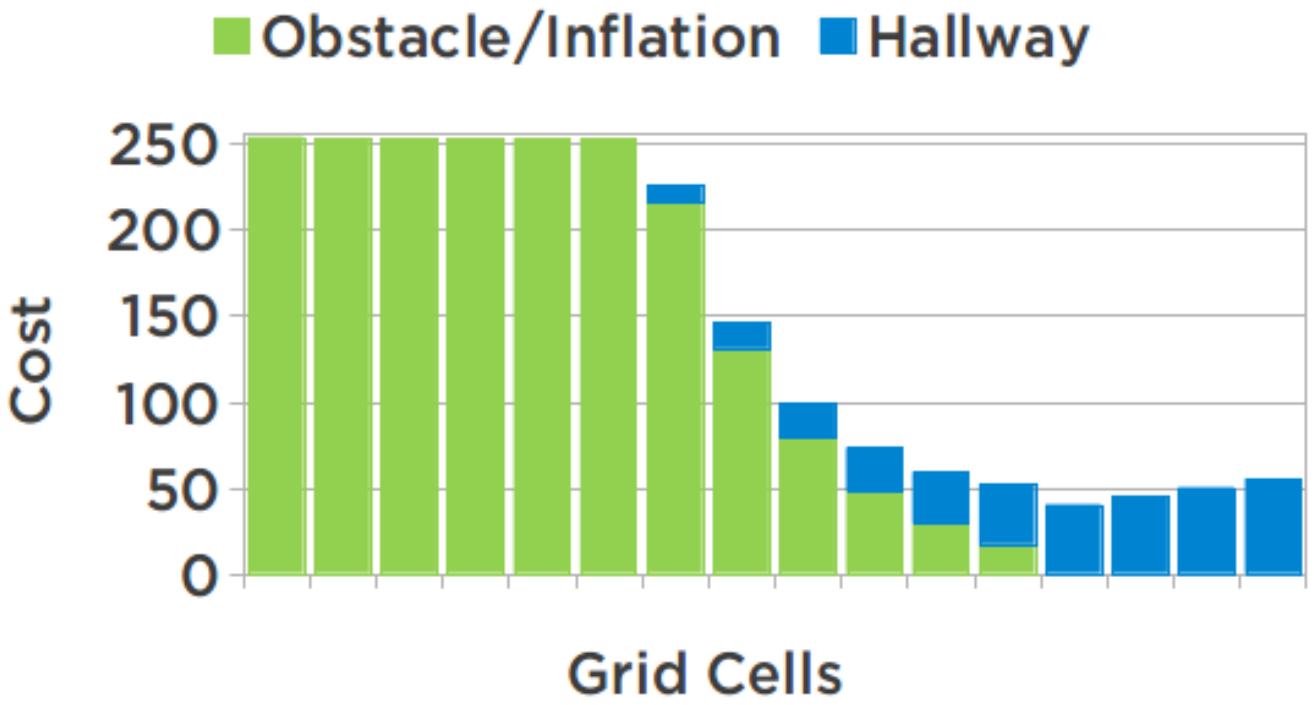
Social Navigation



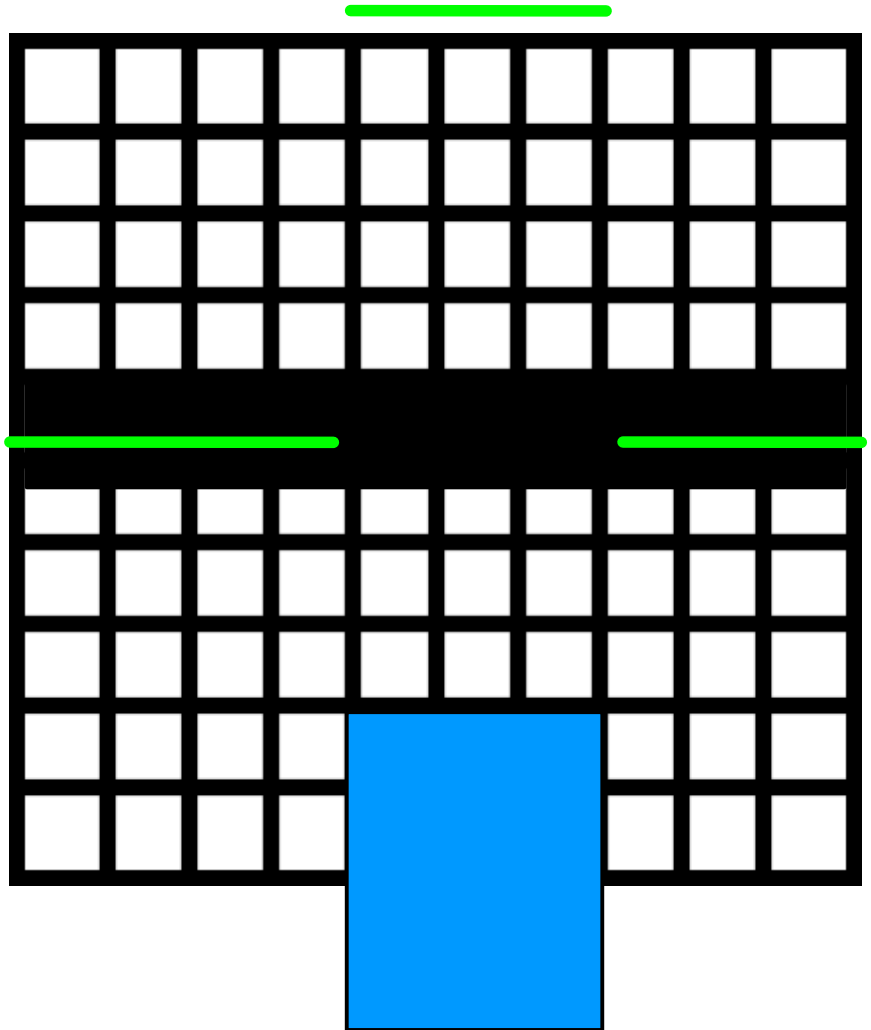
Monolithic Costmap



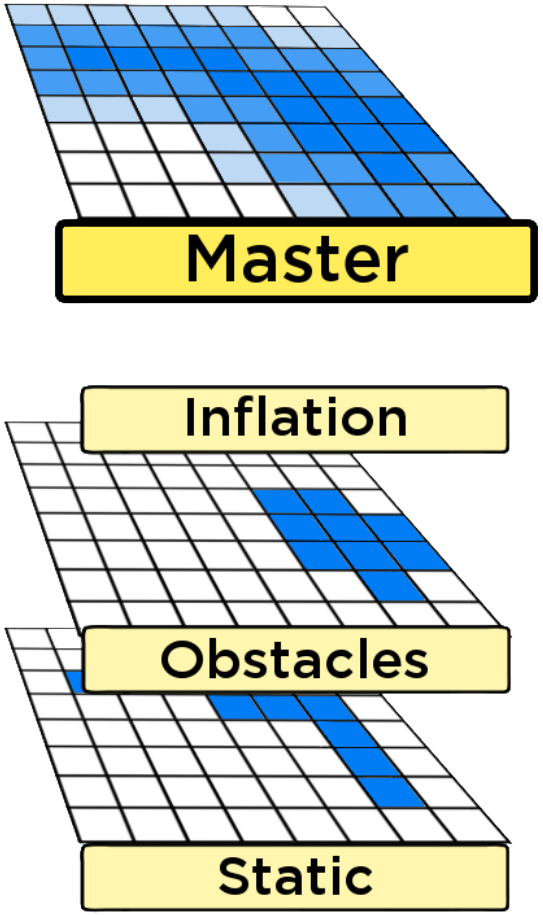
Costmap Overlap



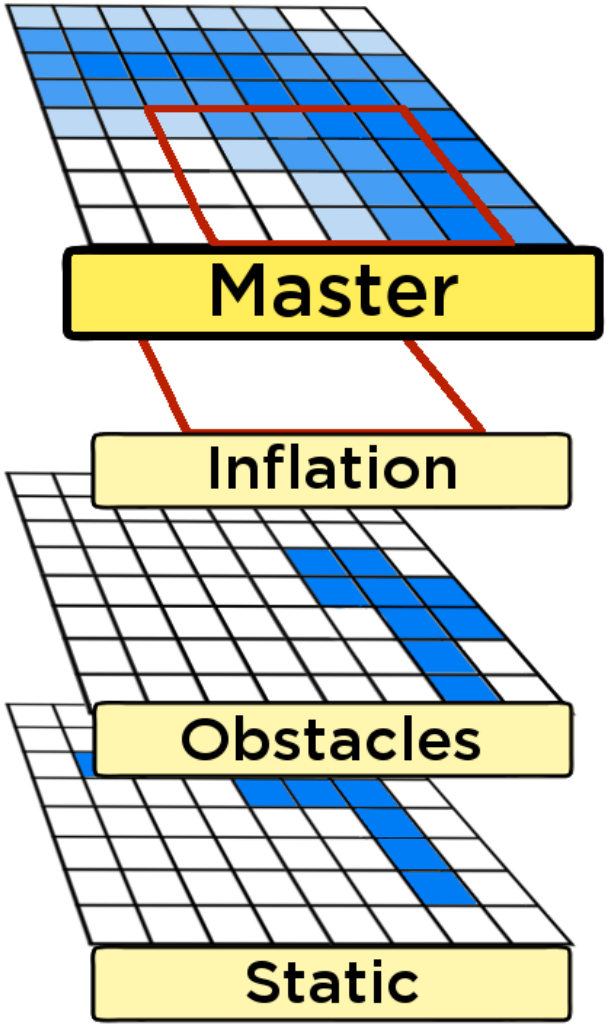
Limited Update Information



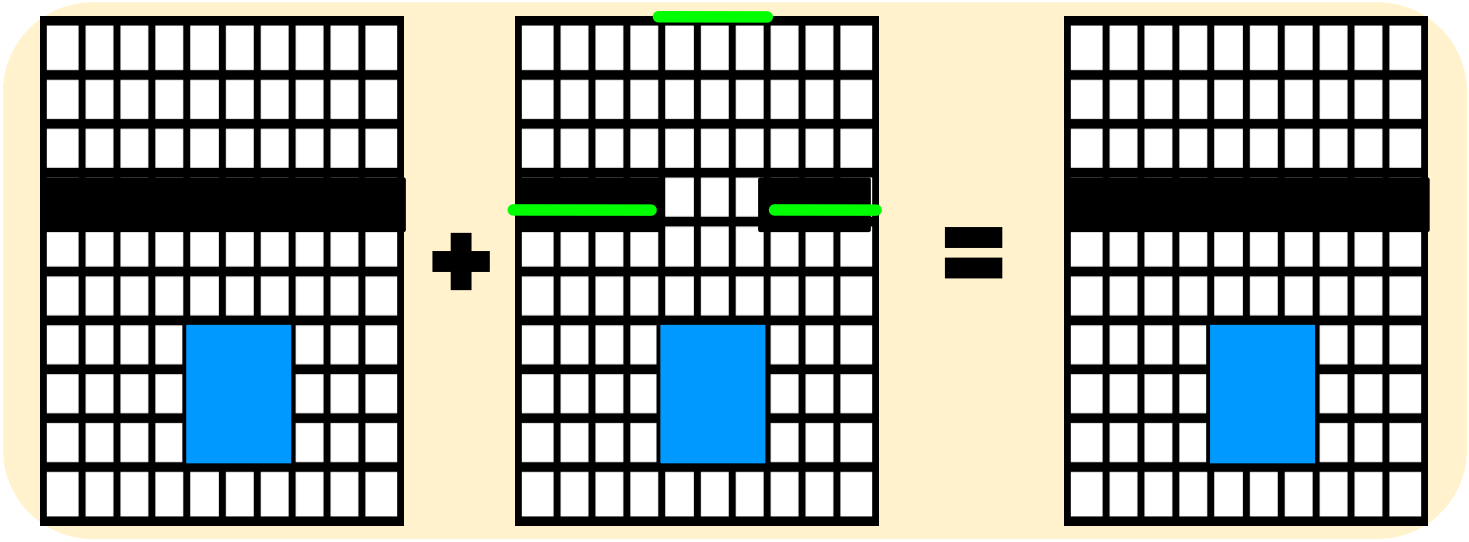
Layered Costmaps

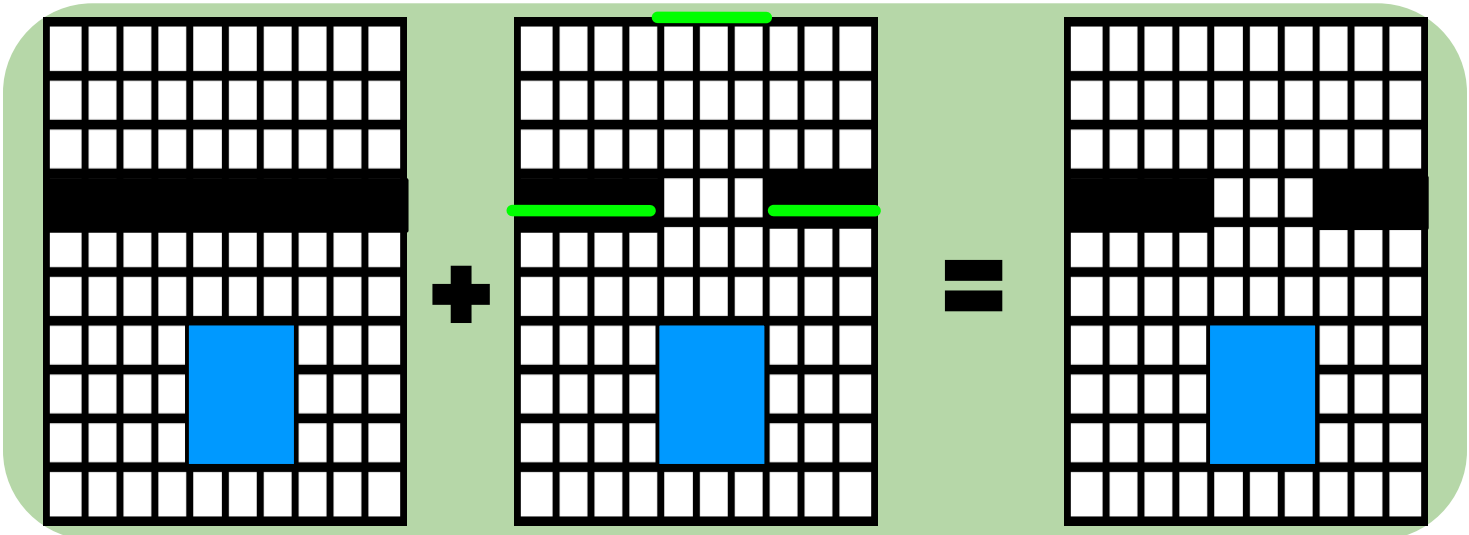


Two Pass Update Process



Layer Combination





Costmap Layers

Obstacles Layer

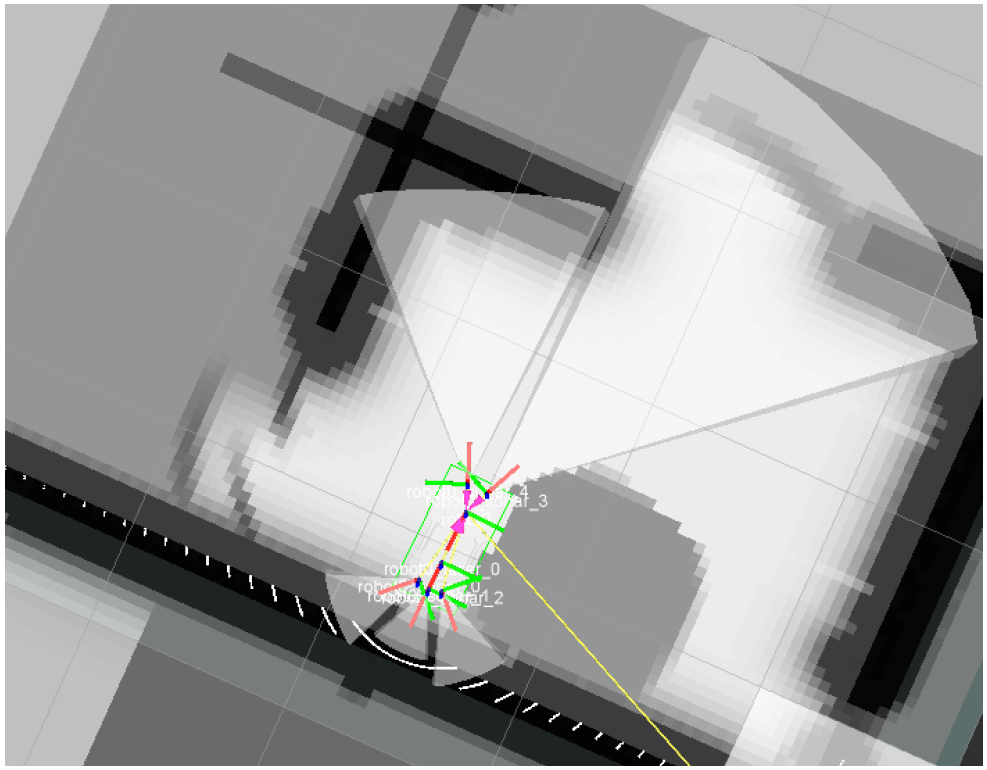
Inflation Layer

Static Layer

Range Sensor Layer

Proxemic Layer

Claustrophobic Layer



Implementing a Layer

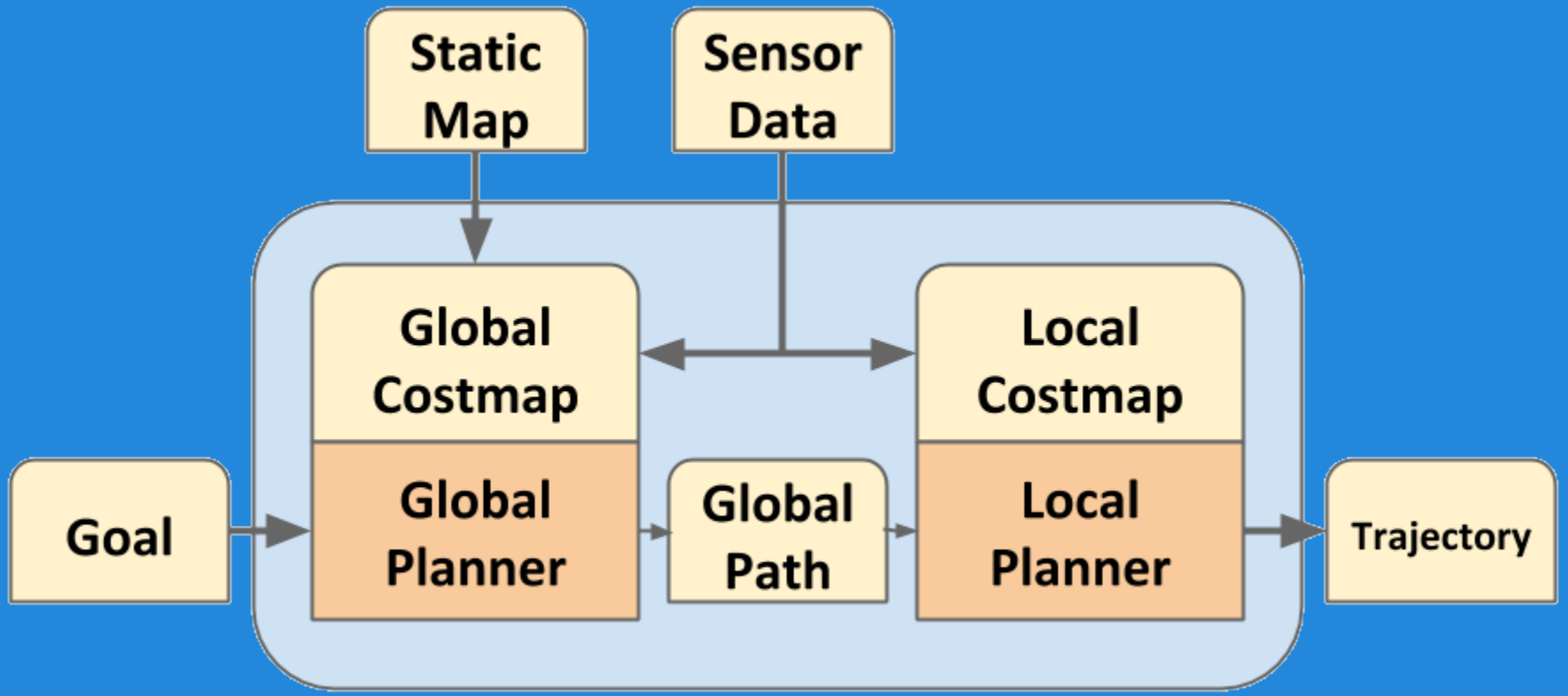
```
class Layer
{
public:
 void initialize( LayeredCostmap* parent, std::string name,
                  tf::TransformListener *tf );

 virtual void updateBounds(
   double robot_x, double robot_y, double robot_yaw,
   double* min_x, double* min_y, double* max_x, double* max_y) {}

 virtual void updateCosts(Costmap2D& master_grid,
       int min_i, int min_j, int max_i, int max_j) {}
```

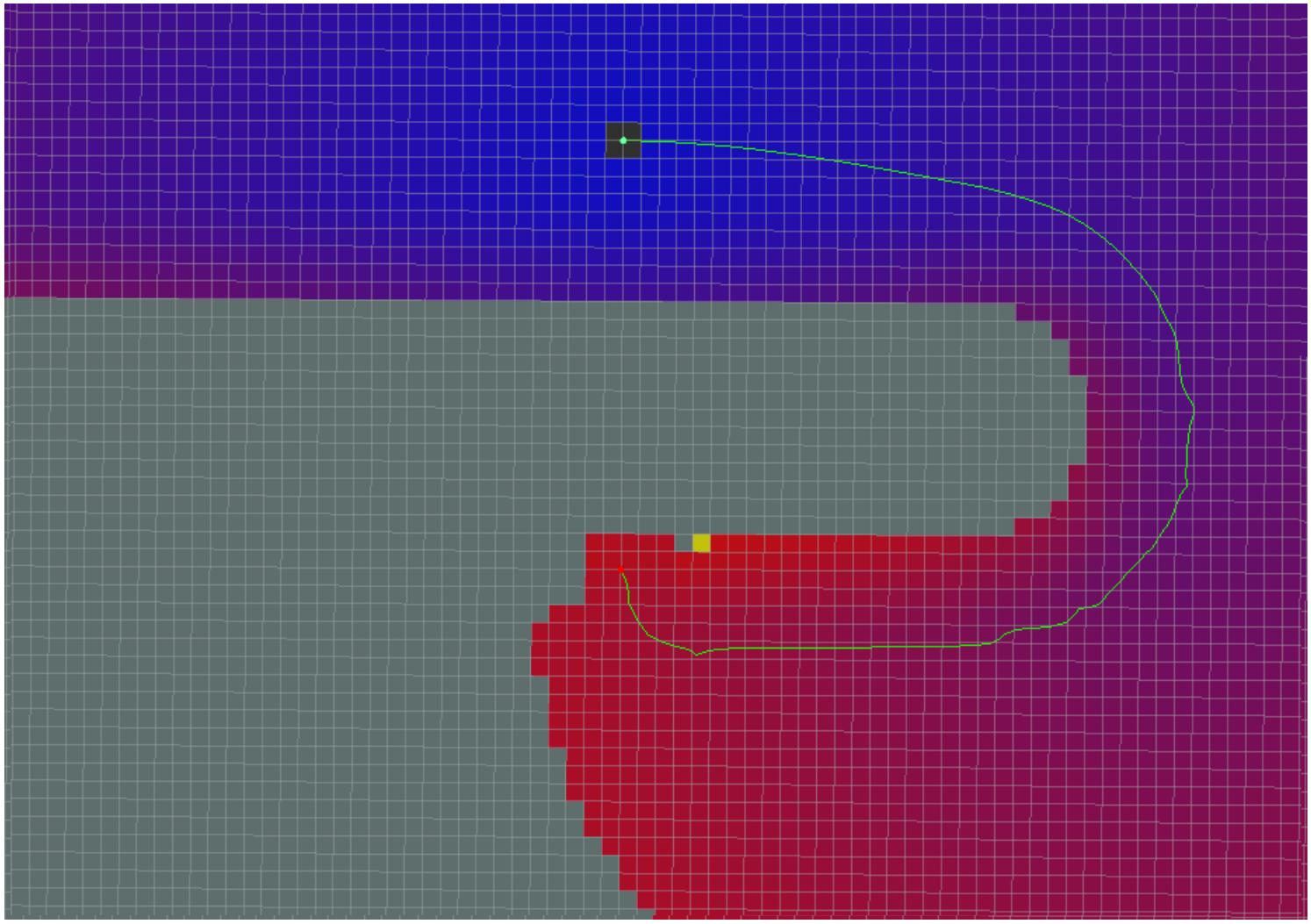
Layered Costmaps

github.com/ros-planning/navigation
github.com/wg-perception/people
github.com/DLu/navigation_layers

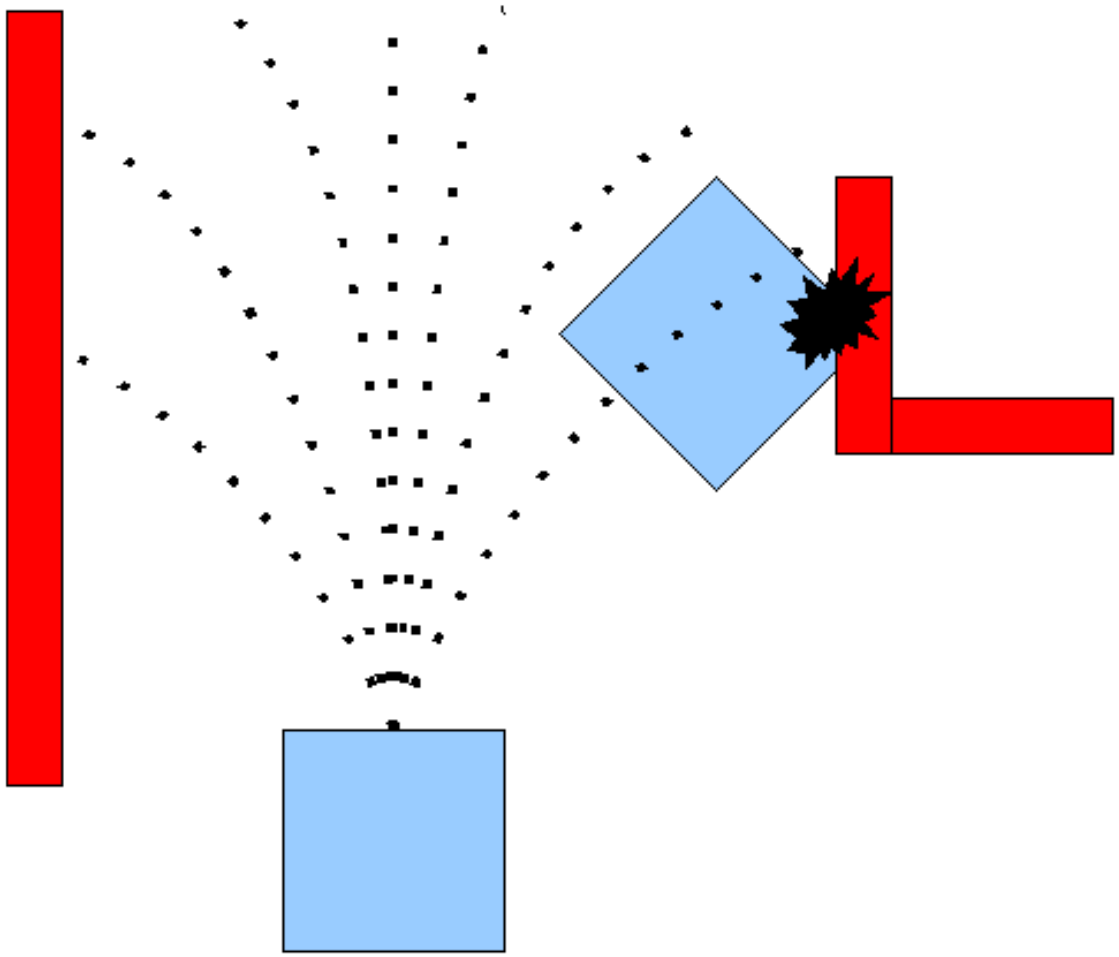


Overview of ROS NavStack

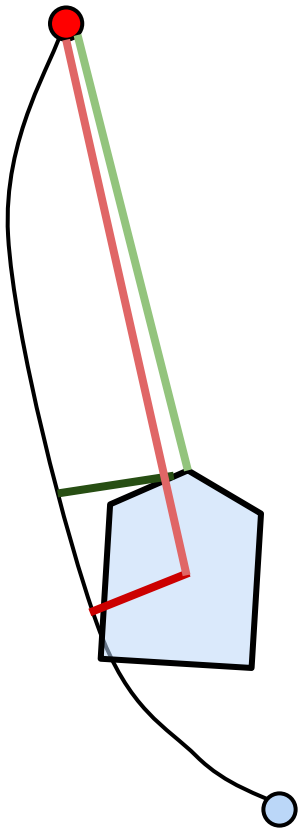
global_planner



BaseLocalPlanner vs. DWALocalPlanner



Scoring Trajectories



Weighted Sum =

oscillation_cost

+ costmap_cost

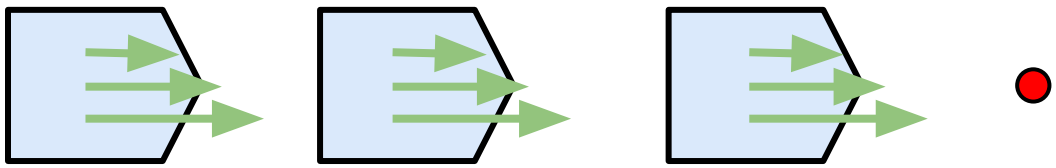
+ goal_distance_cost

+ path_distance_cost

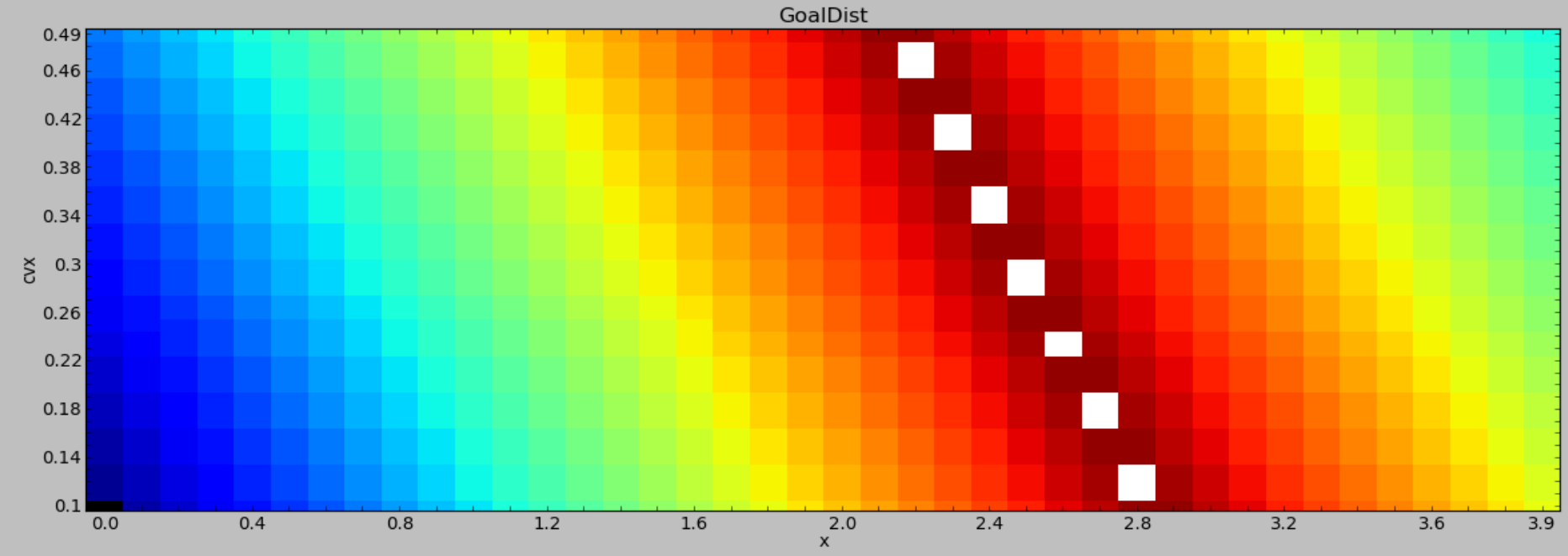
+ goal_alignment_cost

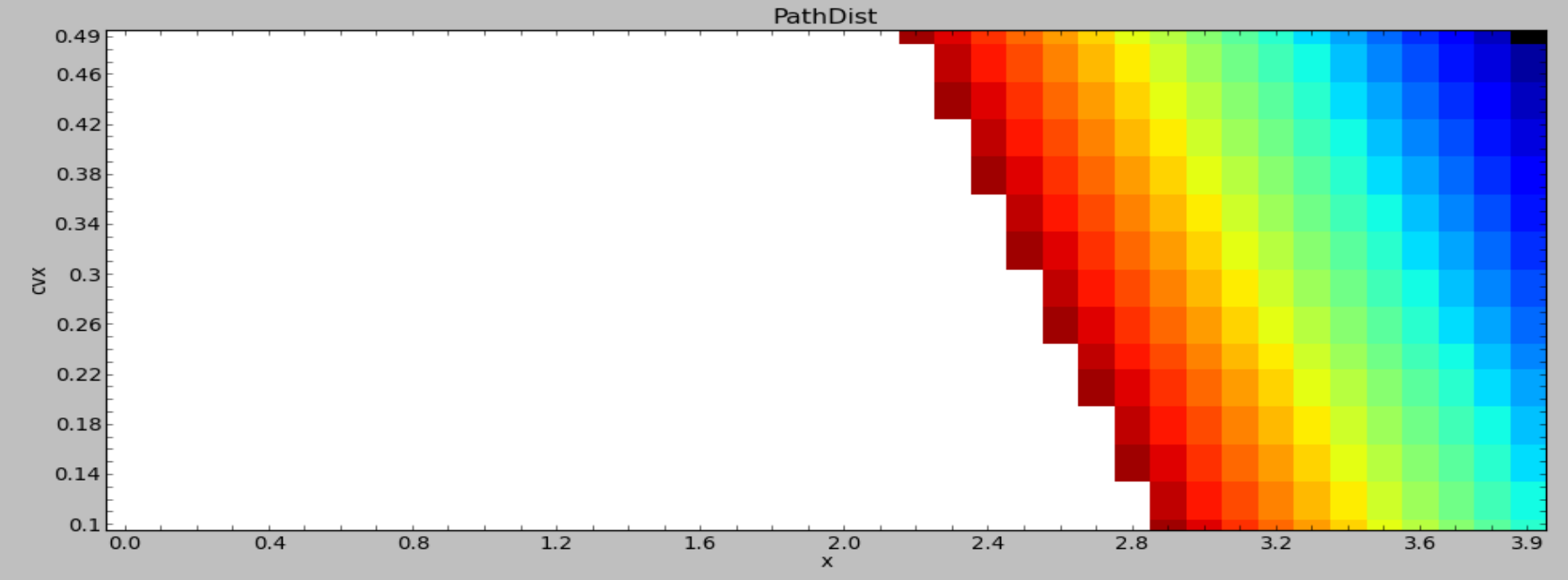
+ path_alignment_cost

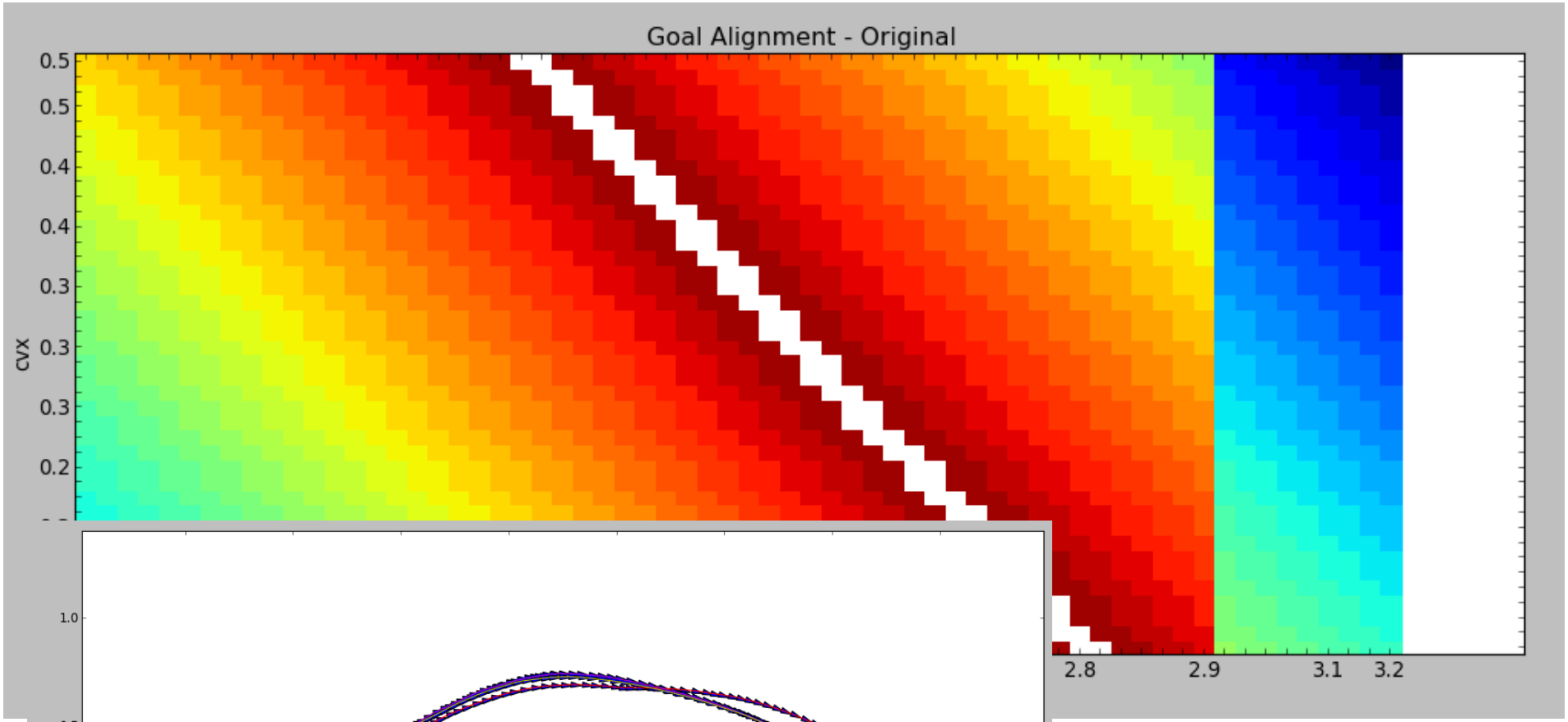
Scoring Different Trajectories

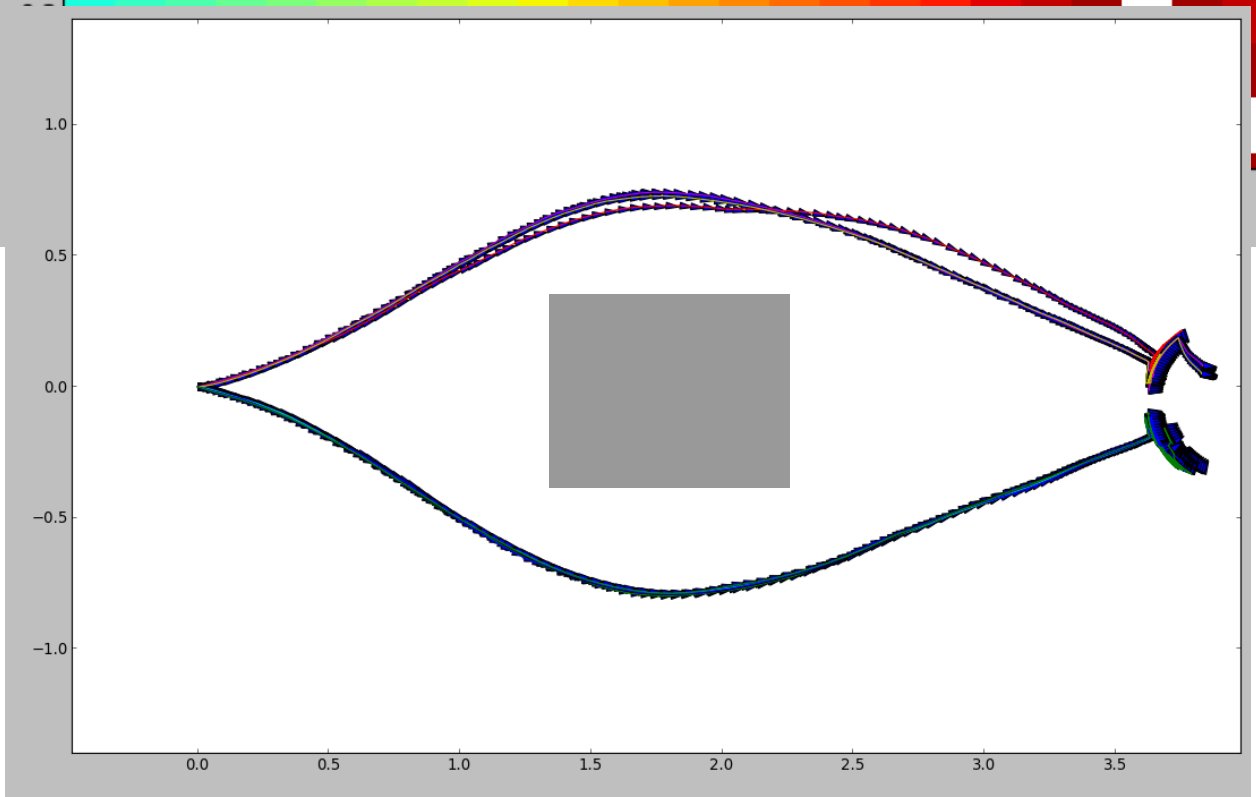


Vary x position
and x velocity


GoalDist
cvx
0.49
0.46
0.42
0.38
0.34
0.3
0.26
0.22
0.18
0.14
0.1
0.0
0.4
0.8
1.2
1.6
2.0
2.4
2.8
3.2
3.6
3.9
x


PathDist
cvx
0.49
0.46
0.42
0.38
0.34
0.3
0.26
0.22
0.18
0.14
0.1
0.0
0.4
0.8
1.2
1.6
2.0
2.4
2.8
3.2
3.6
3.9
x

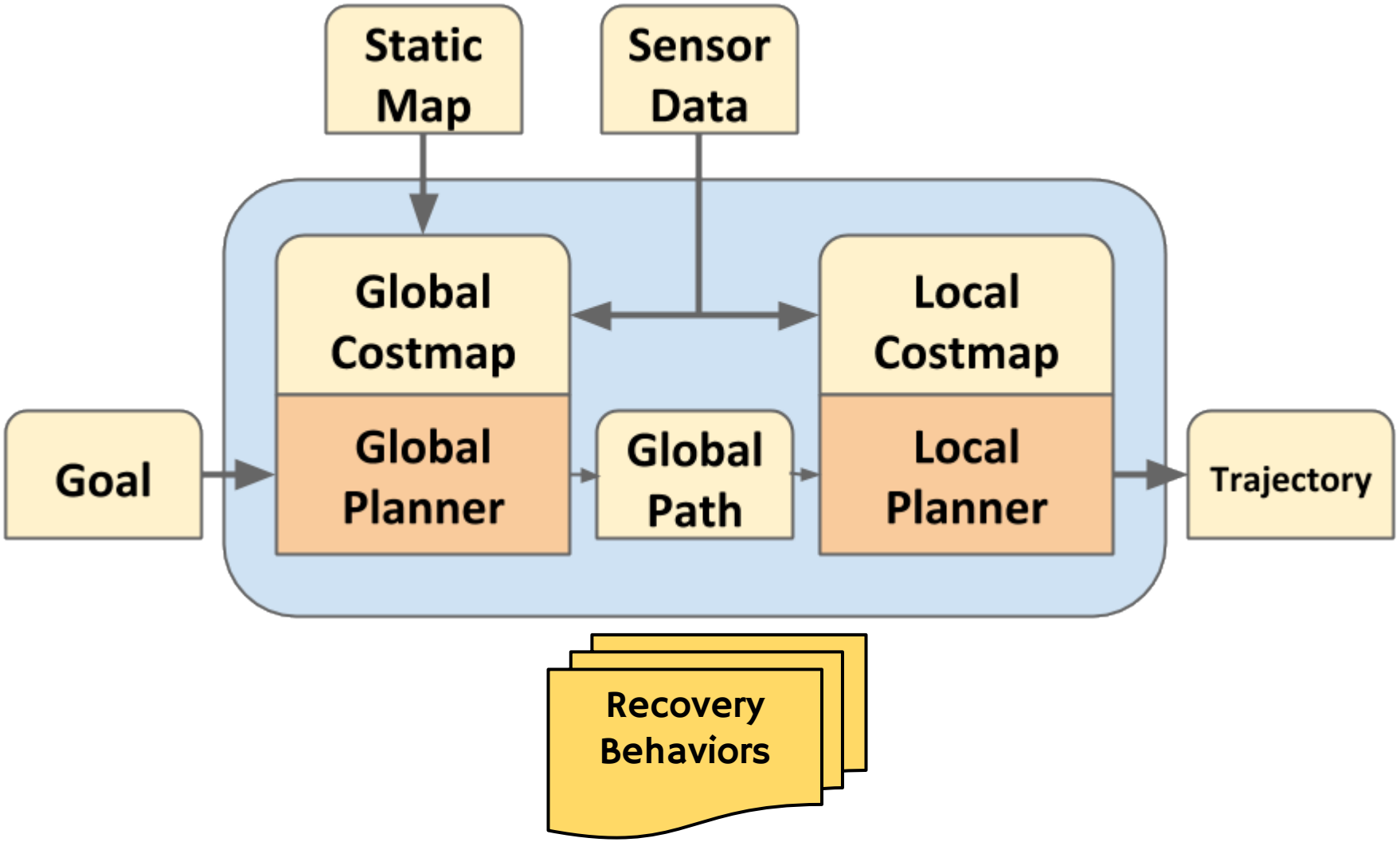

Goal Alignment - Original
cvx
0.5
0.5
0.4
0.4
0.3
0.3
0.2
2.8
2.9
3.1
3.2


1.0
0.5
0.0
−0.5
−1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

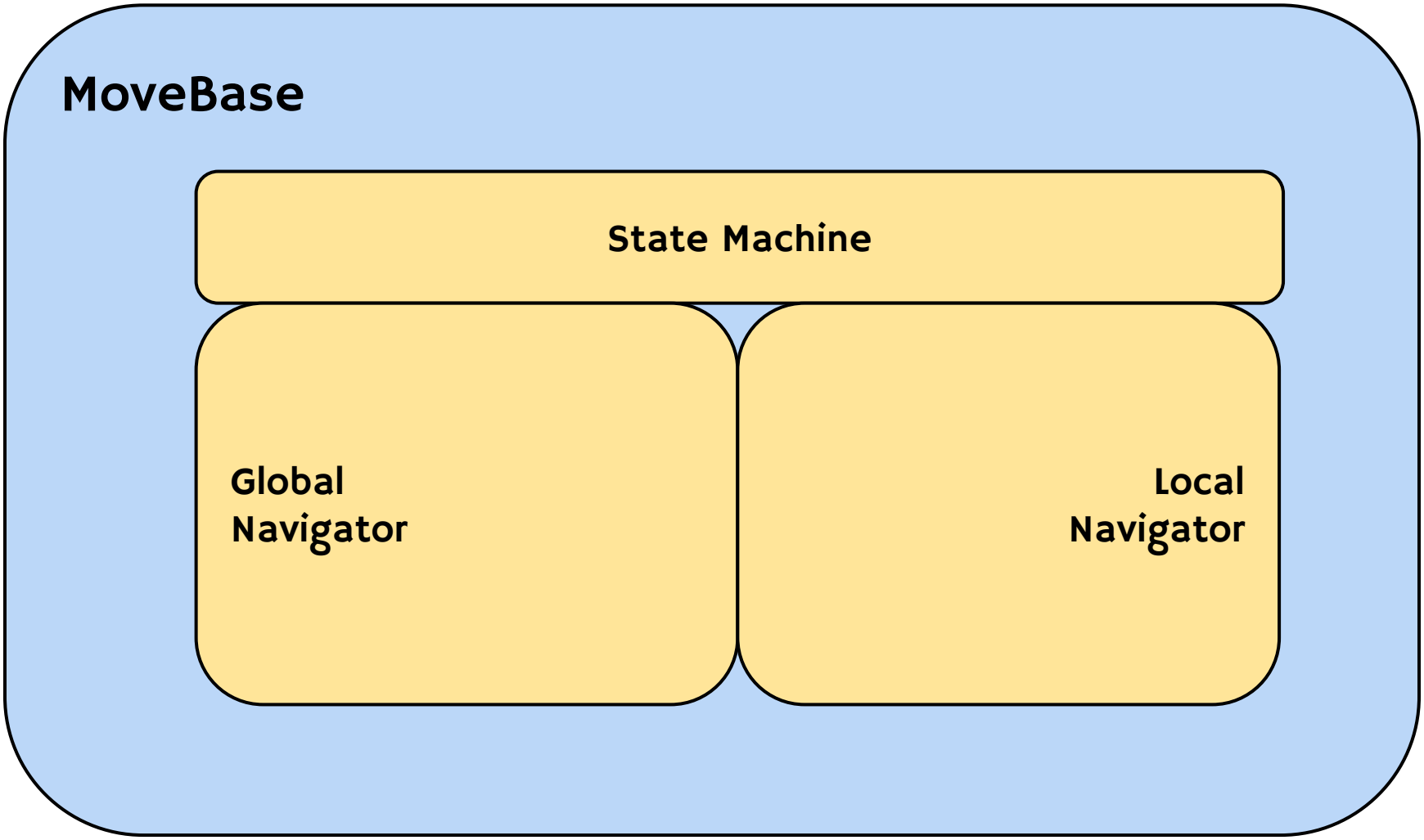
Implementing a Cost Function

```
class TrajectoryCostFunction {
 ...
 virtual double scoreTrajectory(Trajectory &traj);
 ...
}
```

MoveBase


Static Map
Sensor Data
Global Costmap
Local Costmap
Goal
Global Planner
Global Path
Local Planner
Trajectory
Recovery Behaviors

MoveBase The Next Generation



Move Base Social State Machine

DWA Plugin Planner and MoveBase2

https://github.com/DLu/navigation
/tree/groovy_plugin_planner

https://github.com/DLu/navigation
/tree/groovy_mbsplit

ROS Navigation

https://github.com/ros-planning/navigation