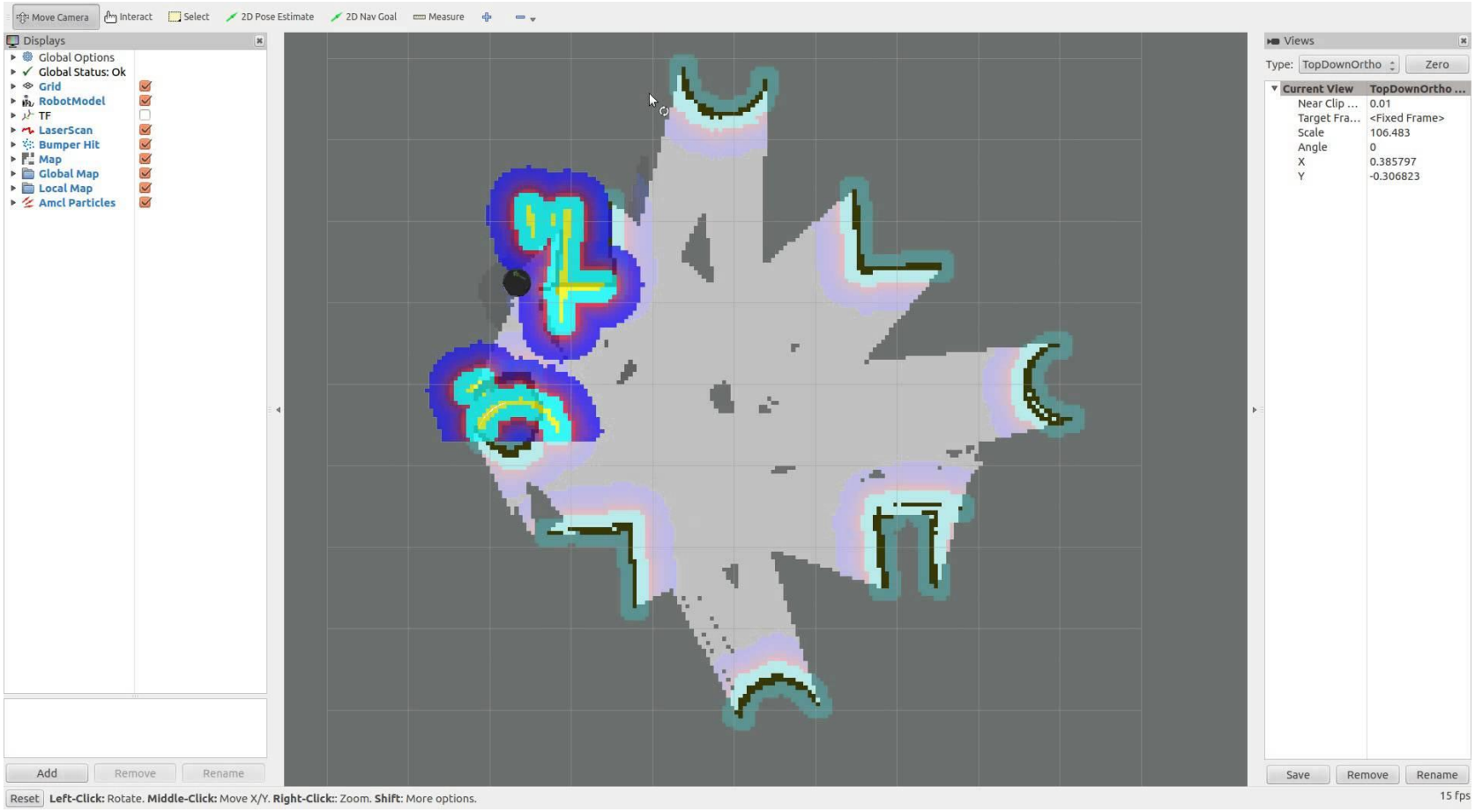
INTELLIGENT ROBOTS

LAB 1: INTRODUCTION TO ROS NAVIGATION

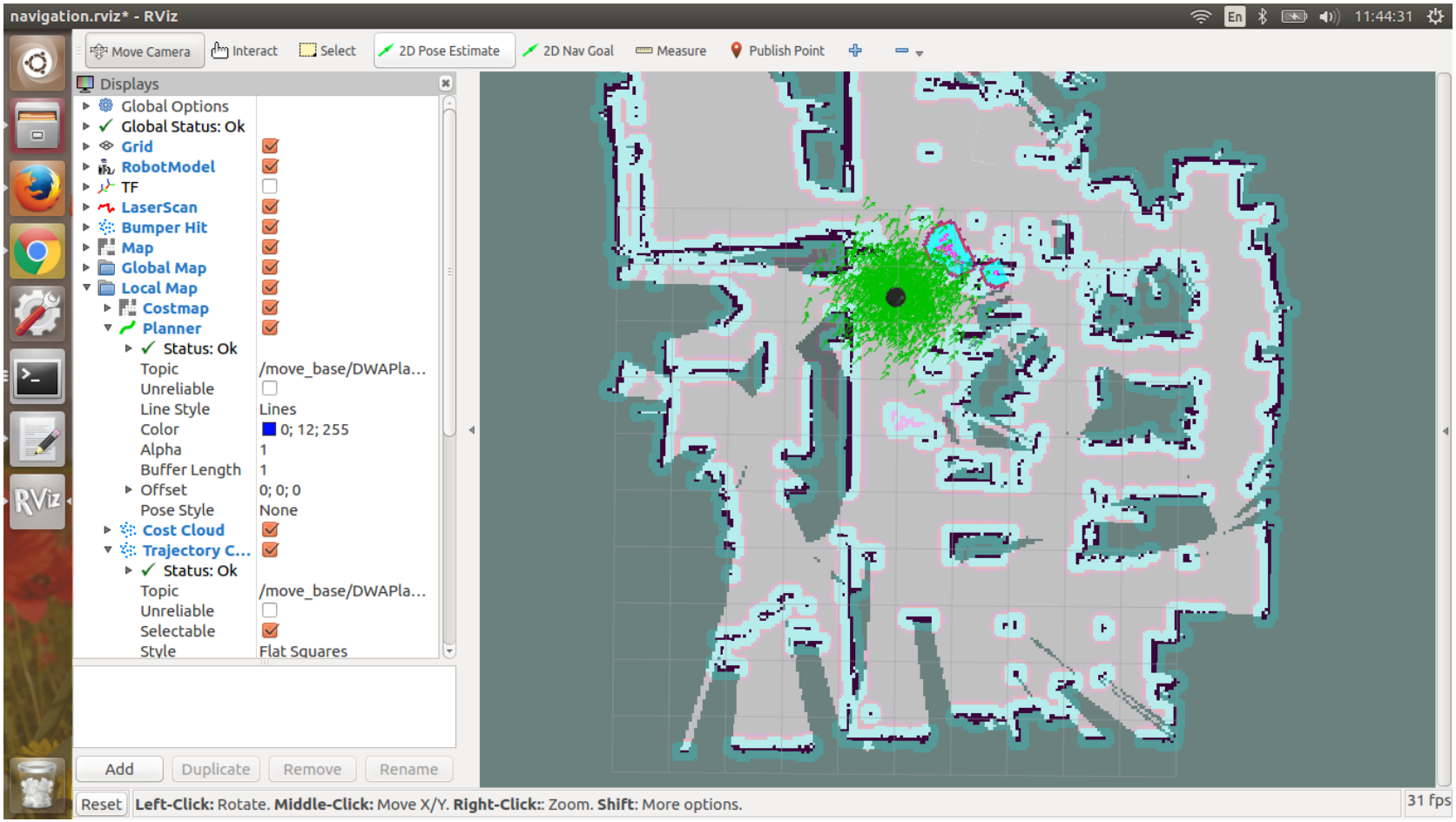
Robot navigation

- Mapping(movement&sensor readings)
- Localization (position&orientation)
- Path Planning (global&local)
- Obstacle Avoidance (Not in map)

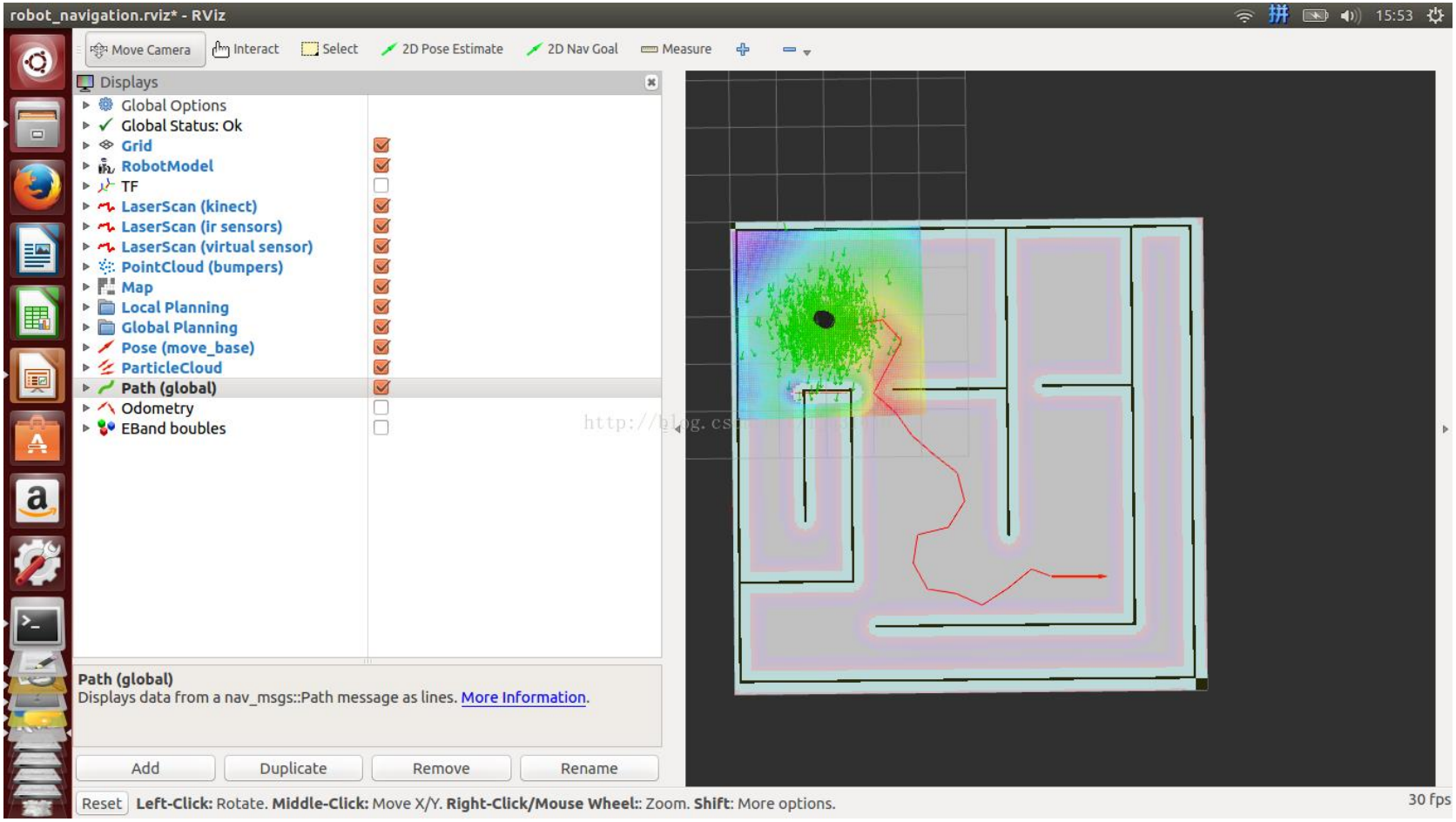
Mapping


Move Camera
Interact
Select
2D Pose Estimate
2D Nav Goal
Measure
Displays
Global Options
Global Status: Ok
Grid
RobotModel
TF
LaserScan
Bumper Hit
Map
Global Map
Local Map
Amcl Particles
Add
Remove
Rename
Views
Type: TopDownOrtho
Zero
Current View
TopDownOrtho ...
Near Clip ...
0.01
Target Fra...
<Fixed Frame>
Scale
106.483
Angle
0
X
0.385797
Y
-0.306823
Save
Remove
Rename
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
15 fps

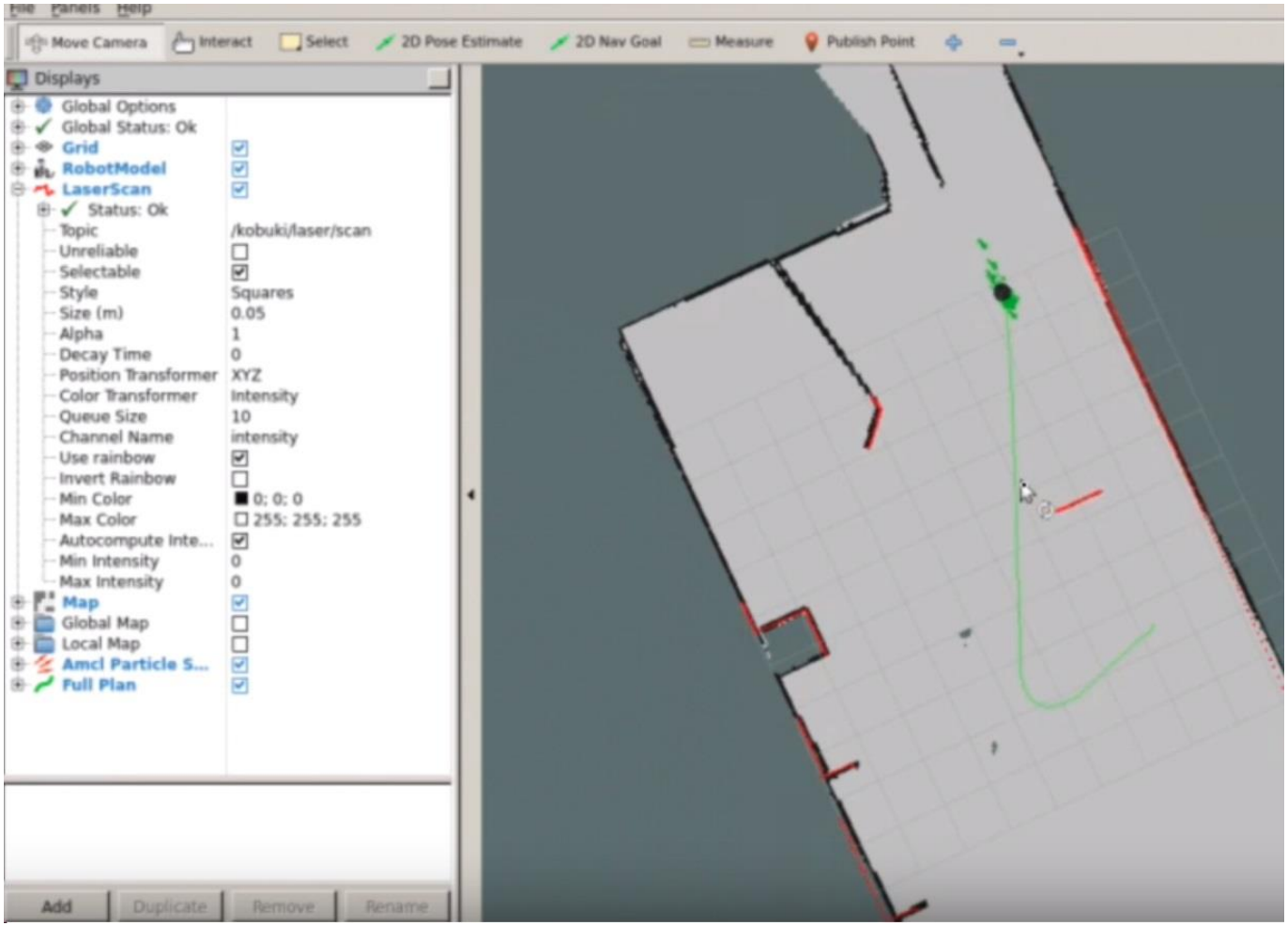
Localization



Path Planning



Obstacle Avoidance



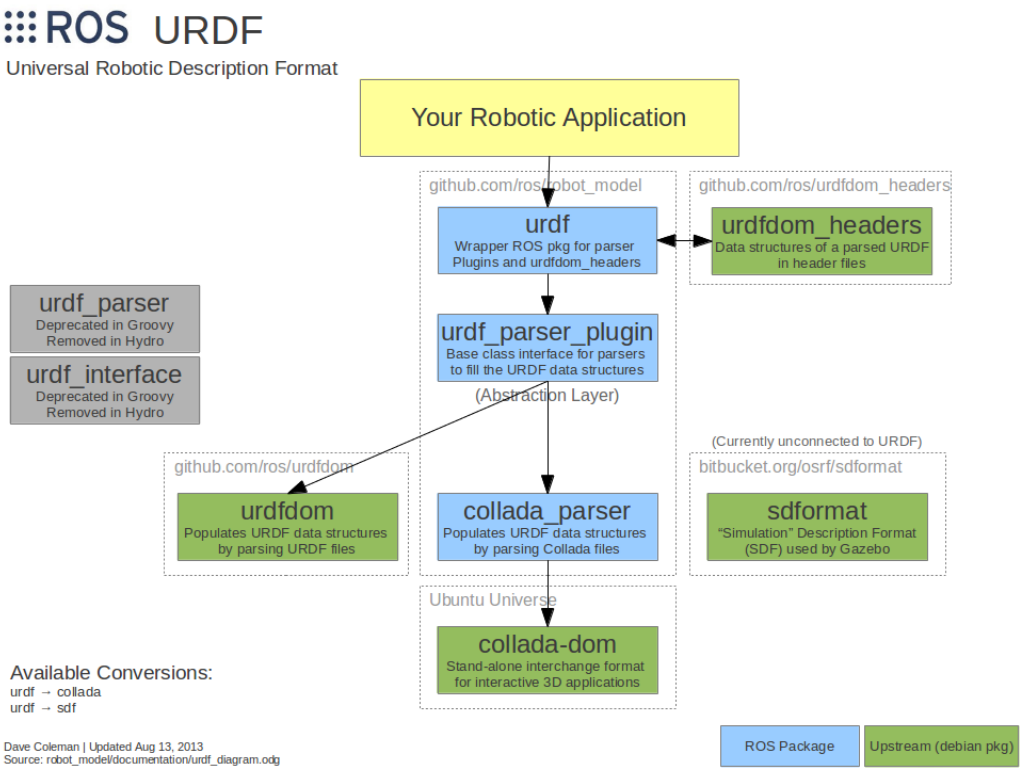
Robot Configuration

Robot Configuration Is How You Robot Is Defined:

- Dimensions
- Location Of Sensors
- Holonomic Or Non-holonomic
- Differential Wheeled Or Not

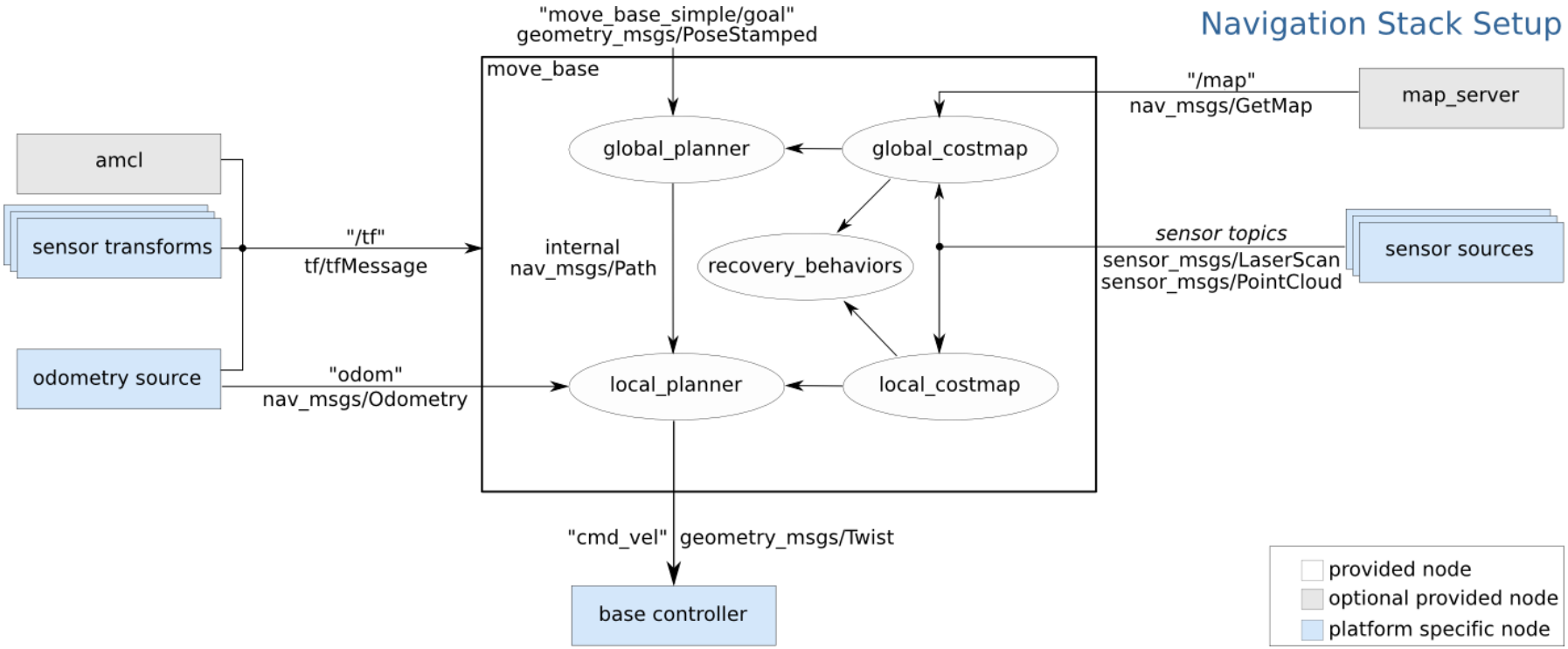
URDF (Universial Robotic Description Format)

XML specifications for robot models, sensors, scenes, etc.



ROS Navigation Stacks Setup

ROS navigation stack is a set of ROS nodes and algorithms which are used to autonomously move a robot from one point to another, avoiding all obstacles the robot might find in its way.



Costmap

				25	24	25				25	24	23	22	21	22
			25	24	23	24	25		25	24	23	22	21	20	21
		25	24	23	22	23	24	25	24	23	22	21	20	19	20
	25	24	23	22	21	22	23	24	23	22	21	20	19	18	19
					20	21	22		22	21	20	19	18	17	18
16	15	16	17	18	19	20	21		23	22				16	17
15	14	15	16	17	18	19	20		24	23	24			15	16
14	13	14	15	16	17	18	19			24	G			14	15
13	12	13	14	15	16	17	18							13	14
12	11	12	13	14	15	16	17							12	13
11	10													11	12
10	9			6	5	4	3	4	5	6	7	8	9	10	11
9	8		6	5	4	3	2	3	4	5	6	7	8	9	10
8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9
7	6	5	4	3	2	1	S	1	2	3	4	5	6	7	8
8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9

ROS Navigation Stacks Setup

- **Odometry source:** Odometry data of a robot gives the robot position with respect to its starting positiona. Main odometry sources are wheel encoders, IMU, and 2D/3D cameras(visual odometry). The odom value should publish to the navigation stack, which has a message type of nav_msgs/odometry. The odom message can hold the position and velocity of the robot.

- **Sensor source:** sensors are used for two tasks in navigation: one for localizing the robot in the map(using for example the laser) or to detect obstacles in the path of the robot(using the laser, sonar or points clouds).

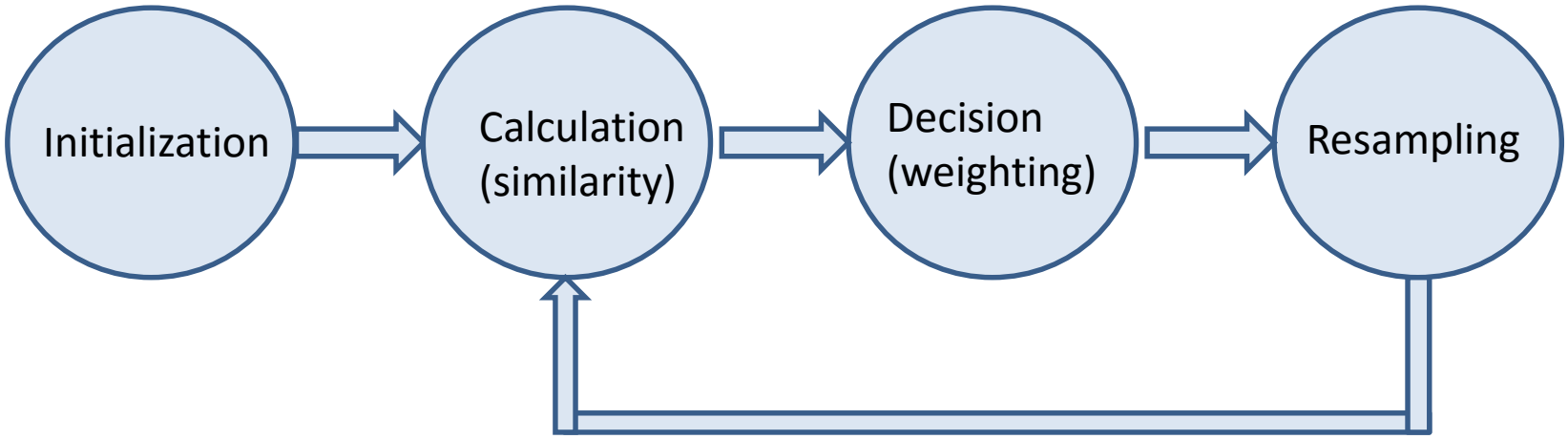
ROS Navigation Stacks Setup

- **Sensor transformations/tf**: the data captured by the different robot sensors must be referenced to a common frame of reference(usually the base_link) in order to be able to compare data coming from different sensors. The robot should publish the relationship between the main robot coordinate frame and the different sensors' frame using ROS tf.

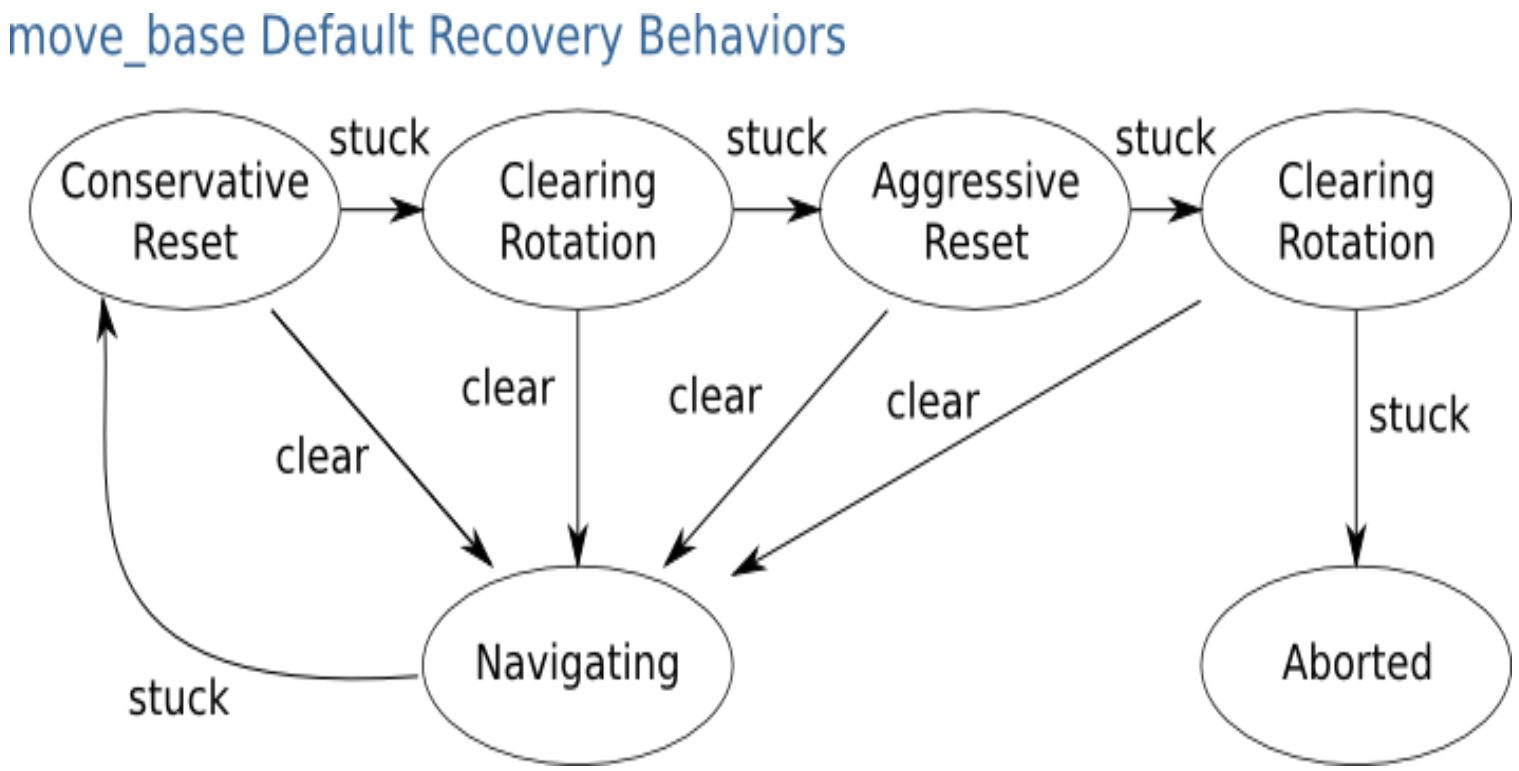
- **Base-controller** : the main function of the base controller is to convert the output of the navigation stack, which is a Twist(geometry_msgs/Twist） message, and convert it into corresponding motor velocities of the robot.

AMCL(adaptive (or KLD-sampling) Monte Carlo localization)

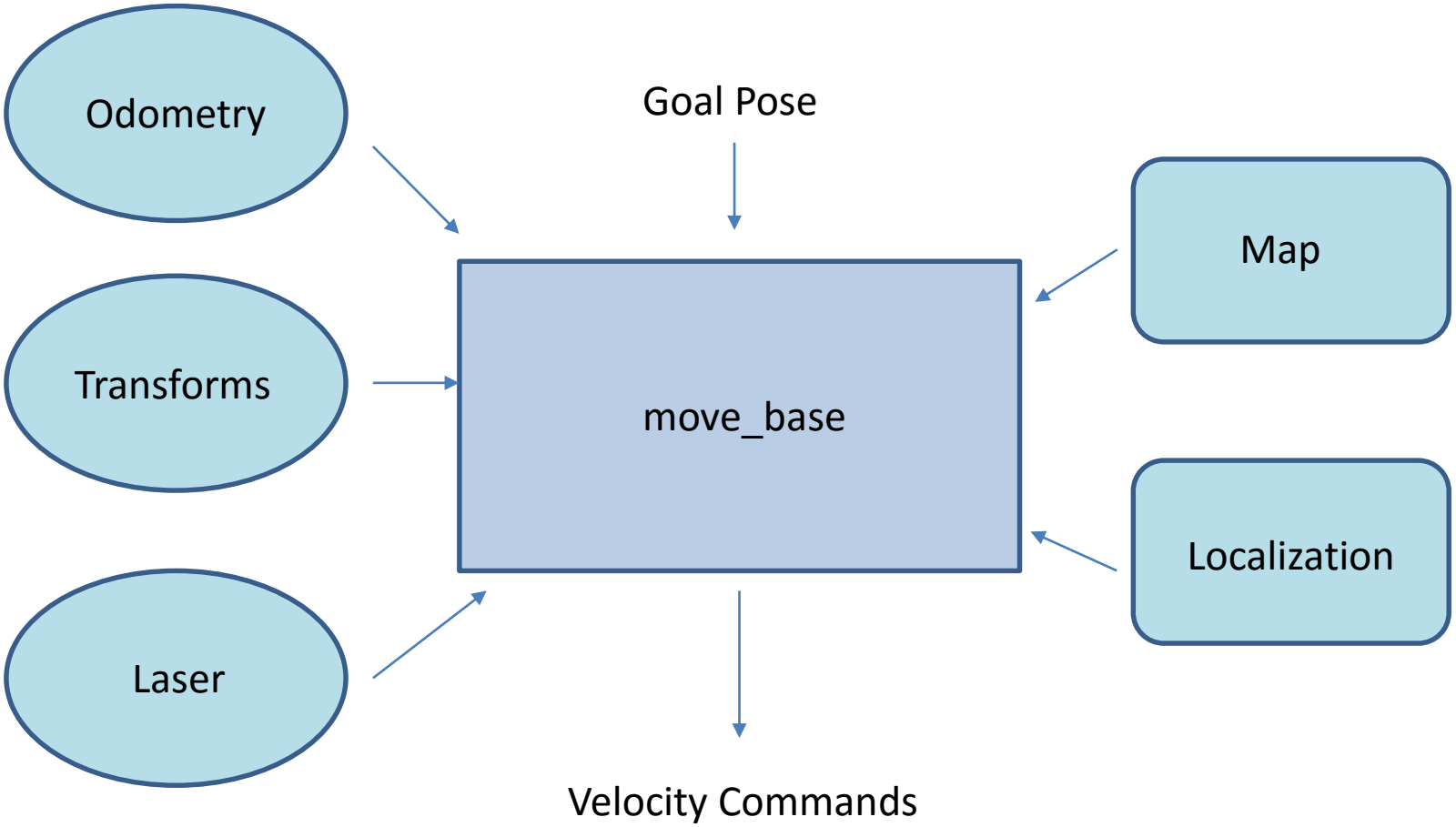
- AMCL is a probabilistic localization system for a robot moving in 2D. It implements the adaptive (or KLD-sampling) Monte Carlo localization approach (as described by Dieter Fox), which uses a particle filter to track the pose of a robot against a known map.



Recovery_behaviours



ROS Navigation with move_base node



Tasks

- Understand the basic theories of ROS Navigation
- Finish the beginner level tutorials of ROS
- Try to run Demos of ROS Navigation Stacks in Rviz and Gazebo