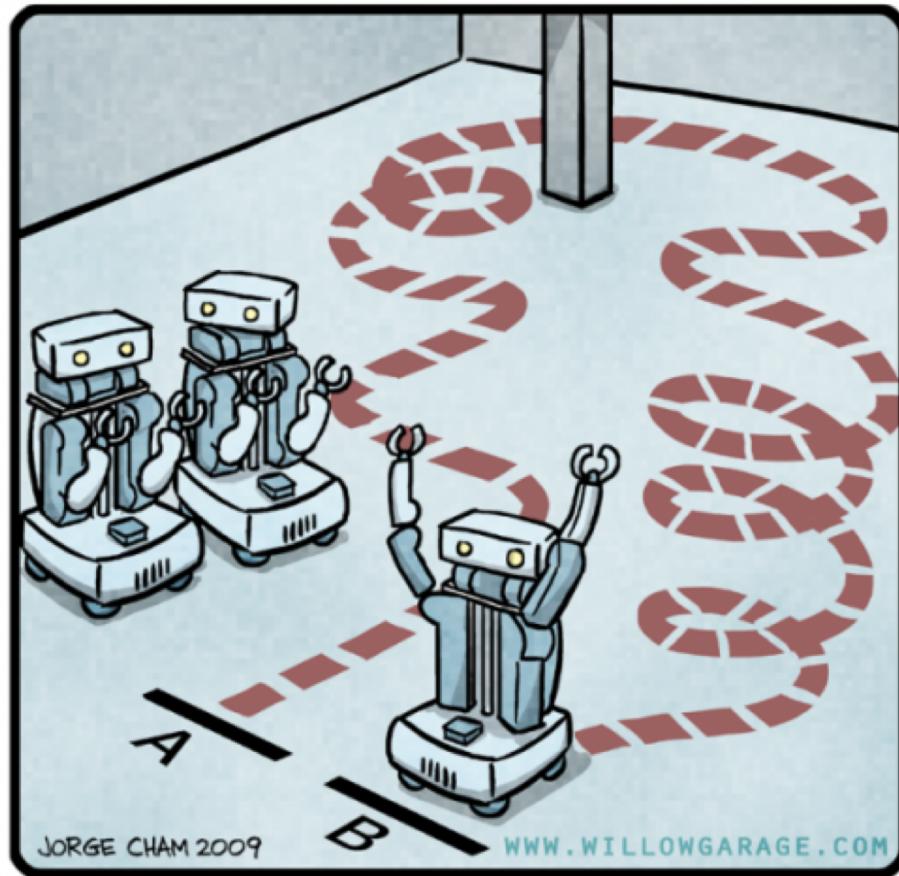


ROS-Navigation-Stack

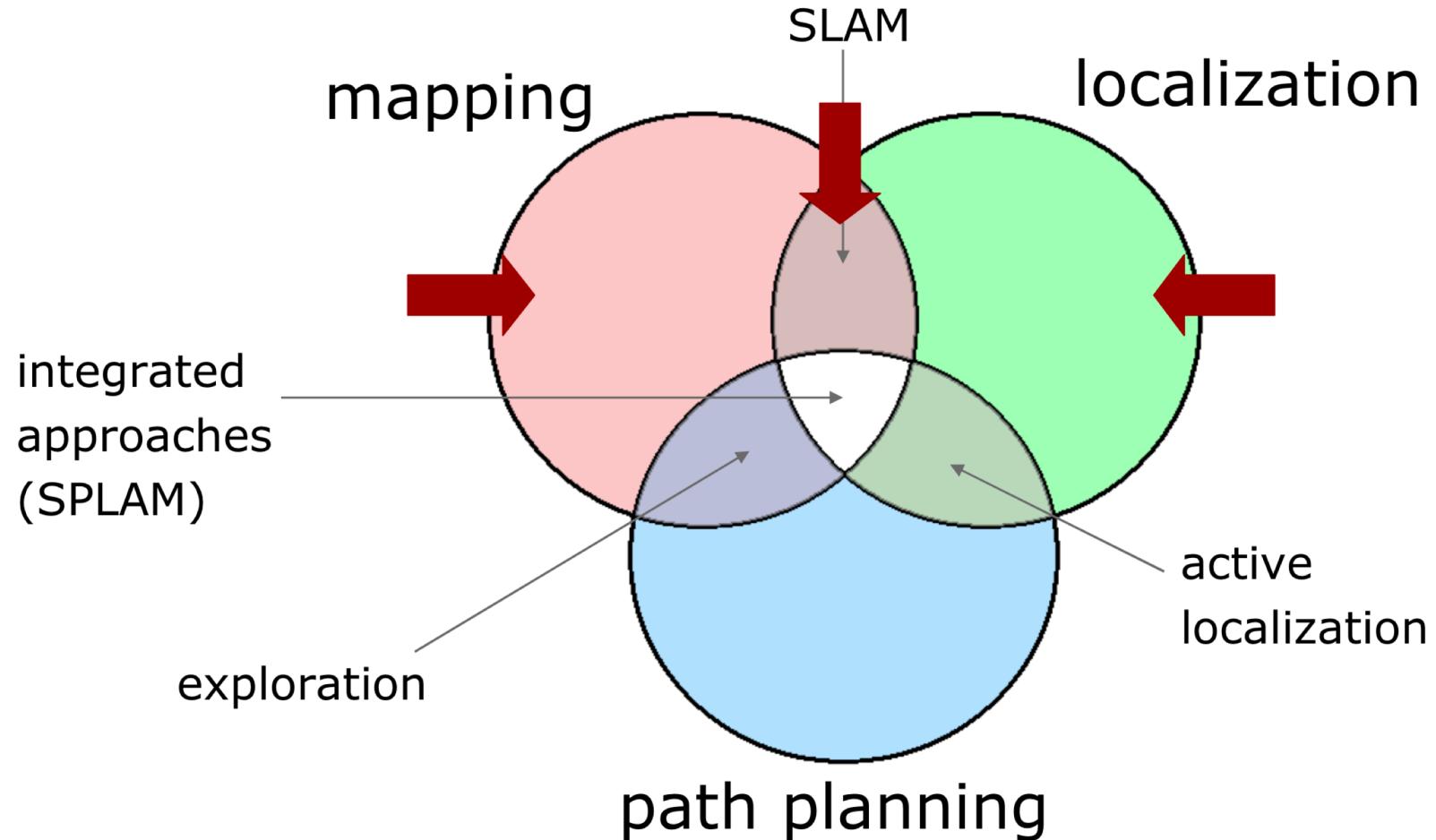


UNA Universität
Augsburg
University

R.O.B.O.T. Comics

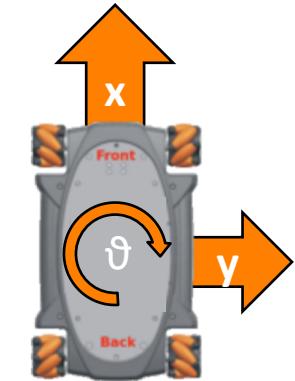


"HIS PATH-PLANNING MAY BE
SUB-OPTIMAL, BUT IT'S GOT FLAIR."

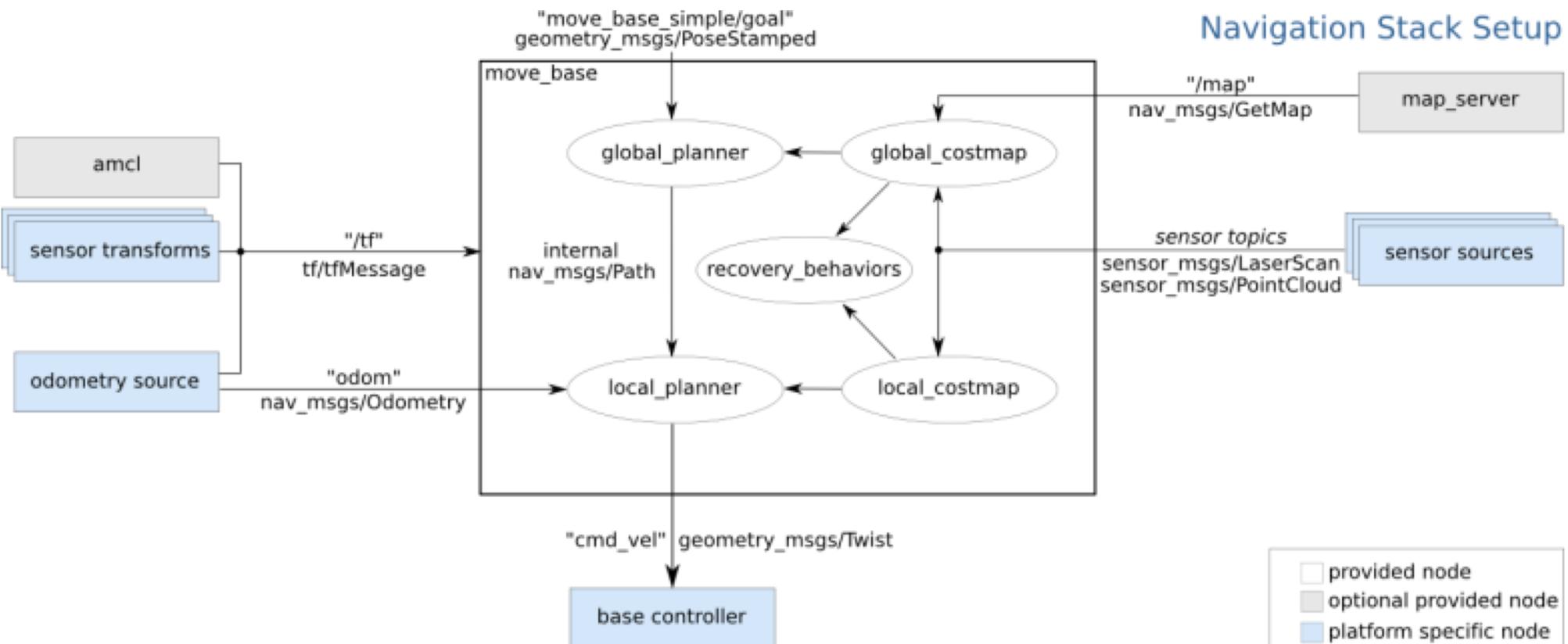


- **Ziel: Bewege einen Roboter sicher von einer Position zu einer anderen**
(ohne Unfall und ohne verloren zu gehen)
- **Input: Odometrie + Sensorwerte**
- **Output: Geschwindigkeitsbefehle** für die mobile Plattform
- Vorbedingungen:
 - Roboter muss mit ROS betrieben werden
 - TF Transformations-Baum muss vorhanden sein
 - Sensoren die ihre Werte per Topics bereitstellen
 - Dynamikmodell und Umrisse des Roboters müssen bekannt sein

- Einschränkung auf **differential- und holonomische Antriebe**
 - Roboter kann auf Geschwindigkeitskommandos reagieren
Geschwindigkeit **x**, Geschwindigkeit **y**,
Drehgeschwindigkeit ϑ
- Mindestens **ein planarer Laser** muss am Roboter angebracht sein
 - Wird für den Aufbau der Karte und zur Lokalisierung verwendet
- Algorithmen funktionieren am Besten mit **quadratischen oder runden Roboterplattformen**



move_base

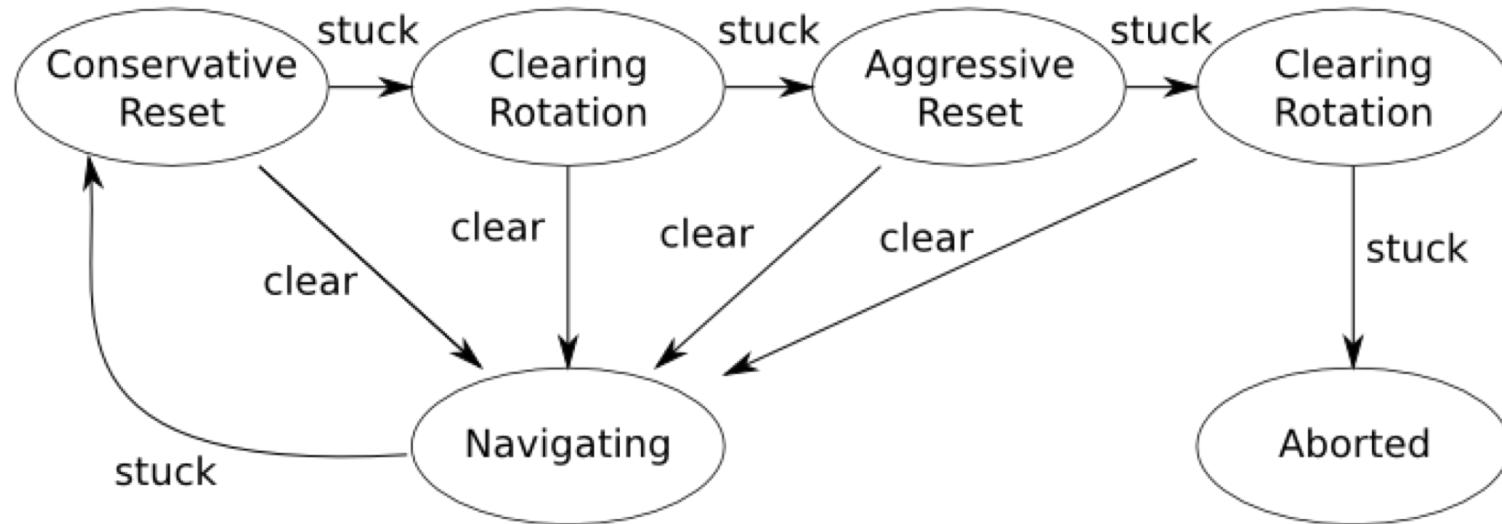


Navigation Stack Hauptkomponenten

Packet	Beschreibung
map_server	Bereitstellung von Karten als ROS Service
gmapping	Packet für laserbasiertes SLAM
amcl	Probabilistisches Lokalisierungssystem (mit Partikelfilter)
global_planner	Schneller globaler Planer für die Navigation
local_planner	Lokaler Planer für dynamische Hindernisse und Trajektorienplanung
move_base	Vereinigung des lokalen und globalen Planers um einen Navigationstask auszuführen
recovery_behaviors	Vorgabe wie sich der Roboter verhält wenn er steckenbleibt (z. B. durch dynamische Veränderungen der Umgebung)

Recovery Behaviour

move_base Default Recovery Behaviors

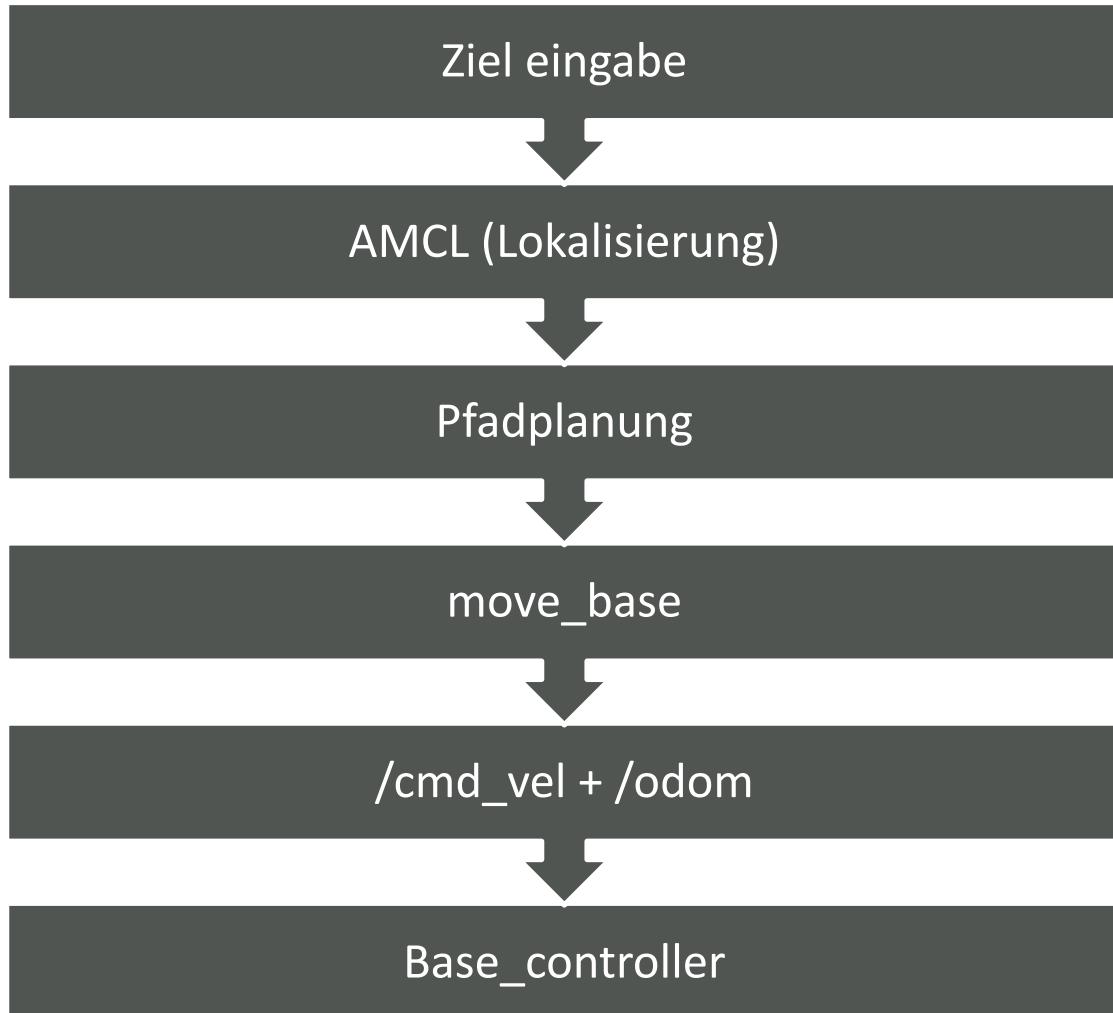


Conservative Reset: Alle Hindernisse, die nicht im Bereich der Nutzervorgabe sind, werden von der Karte gelöscht.

Clearing Rotation: Roboter dreht sich auf der Stelle

Aggressive Reset: Löschen aller Hindernisse, die nicht im Bereich des Roboters das rotieren beeinflussen

Navigation Ablauf



- Der Roboter muss folgende TF-Relationen bereitstellen:
→\map →\odom →\base_footprint →\base_link →\base_link_laser

Frame	Beschreibung
\map	Koordinatensystem welches in der Karte fixiert ist
\odom	Selbst errechnetes Koordinatensystem durch Odometrie
\base_footprint	Basis des Roboters auf dem Boden
\base_link	Basis des Roboters im Rotationszentrum
\base_link_laser	Basis des Lasersensors

Navigationstypen

- **Globale Navigation**

- wird verwendet, um Pfade für ein Ziel in der Karte oder für eine entfernte Entfernung zu erstellen
- Pfad wird berechnet vor der Roboter den nächsten Zielpunkt anfährt
- Planer arbeitet auf einer Costmap um einen optimalen Weg zum Zielpunkt zu finden (Dijkstra Algorithmus)
- Generiert eine Reihe von Wegpunkten für den lokalen Planer

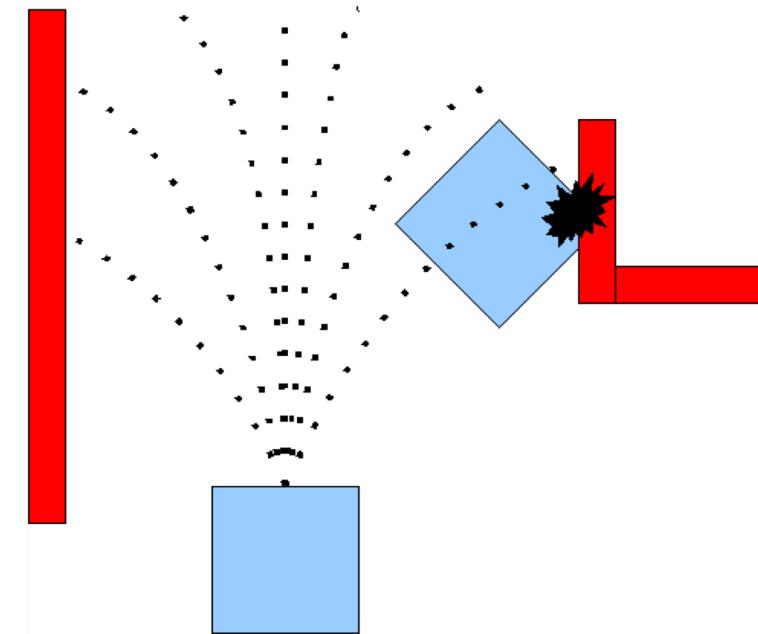
- **Lokale Navigation**

- wird verwendet, um Pfade im Umfeld des Roboters zu erstellen und Hindernisse zu vermeiden
- Berechnet die Geschwindigkeitsparameter um den nächsten Wegpunkt zu erreichen (Berücksichtigung der lokalen und globalen Costmap)
- Neuberechnung möglich um Hindernisse zu umfahren

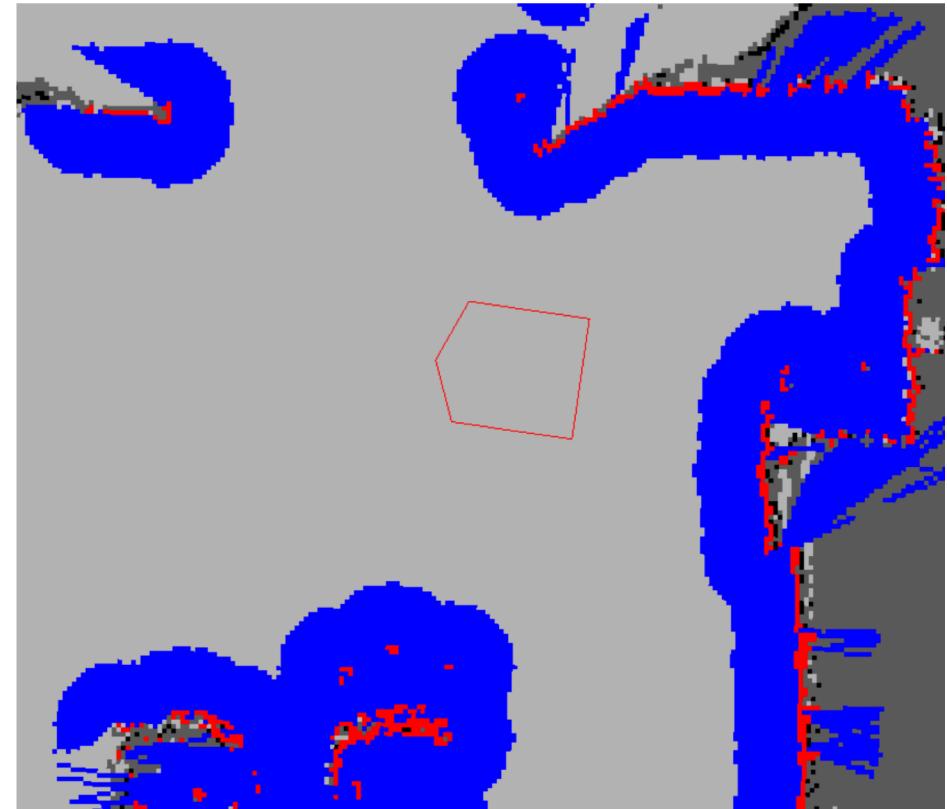
- **Algorithmus: Trajectory Rollout and Dynamic Window**

Trajectory Rollout Algorithmus

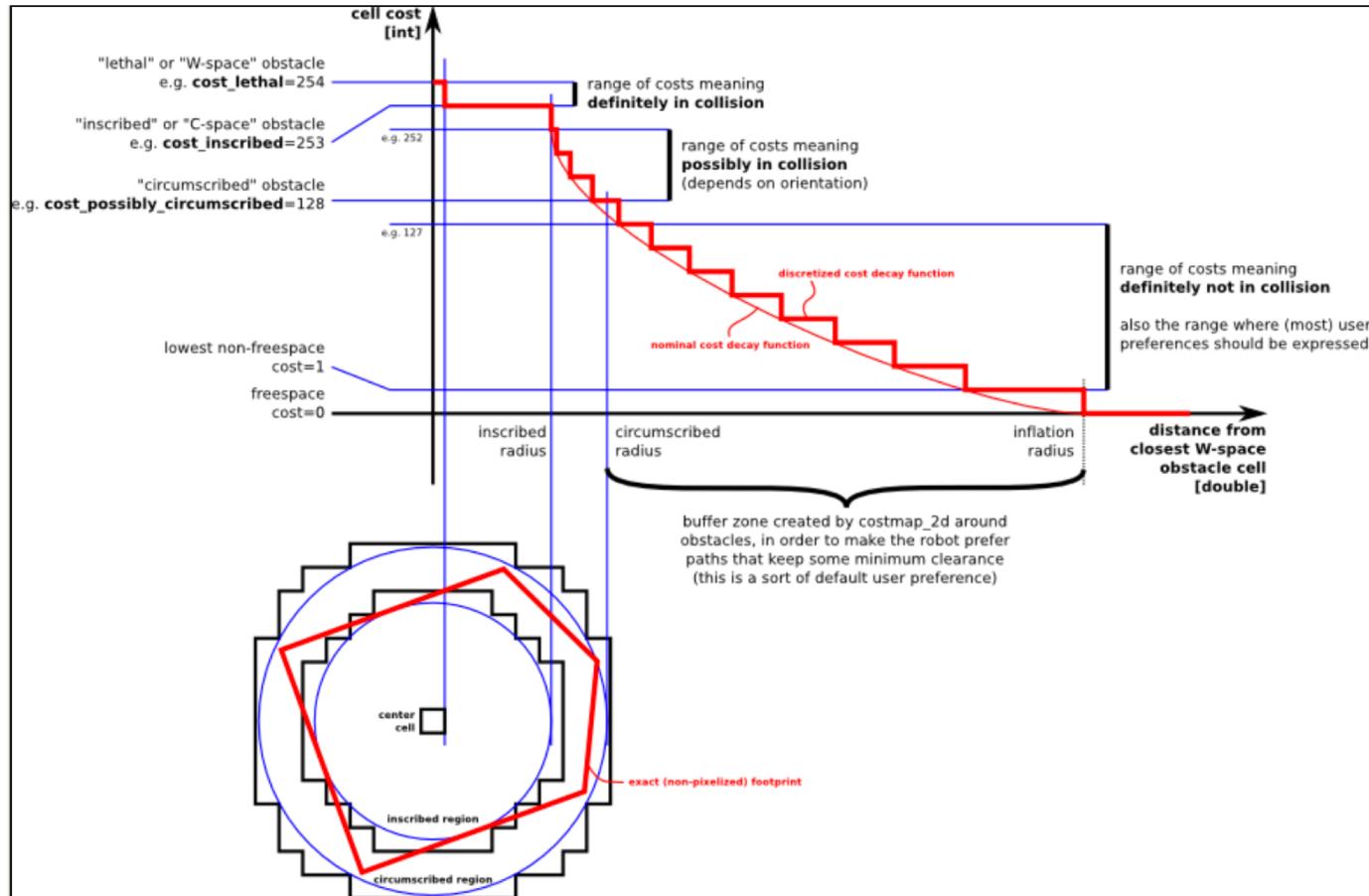
1. Erstelle mehrere Geschwindigkeitsparameter-Sets für den gesamten Arbeitsraum
2. Simuliere die Ausführung für alle Geschwindigkeitsparameter-Sets
3. Evaluiere die Simulationen (Nähe zum Ziel, Nähe zu Hindernissen, Nähe zum globalen Pfad, Geschwindigkeit, ...)
4. Lösche Pfade die zu einer Kollision führen
5. Wähle den am Besten bewerteten Pfad aus
6. Führe diesen Pfad aus



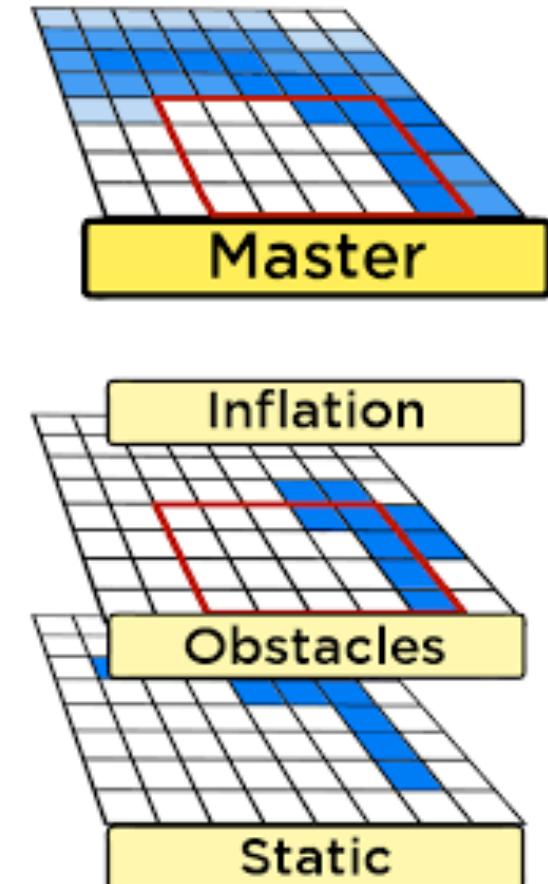
- Datenstruktur die für einen Roboter sichere Orte darstellt
- Es basiert aus Belegungsrasterkarten mit nutzerdefinierten Inflationsradius
- Es gibt zwei Arten von Costmaps
 - Globale Costmaps für globale Navigation
 - Lokale Costmaps für lokale Navigation
- Jedes Raster hat einen Integer-Wert zwischen 0 (Frei) und 255 (Unbekannt) für die Darstellung der Belegung



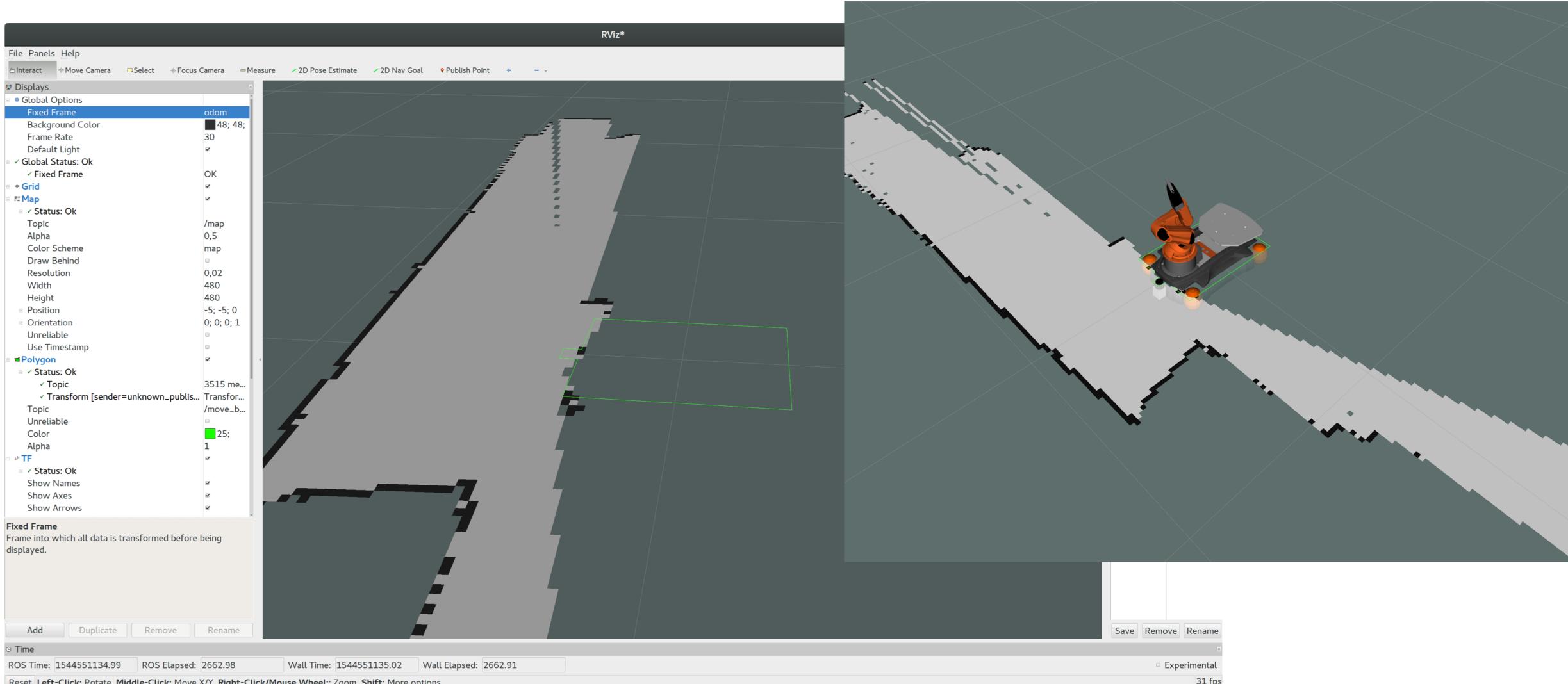
- Prozess zur Verbreitung von Kostenwerten aus besetzten Zellen



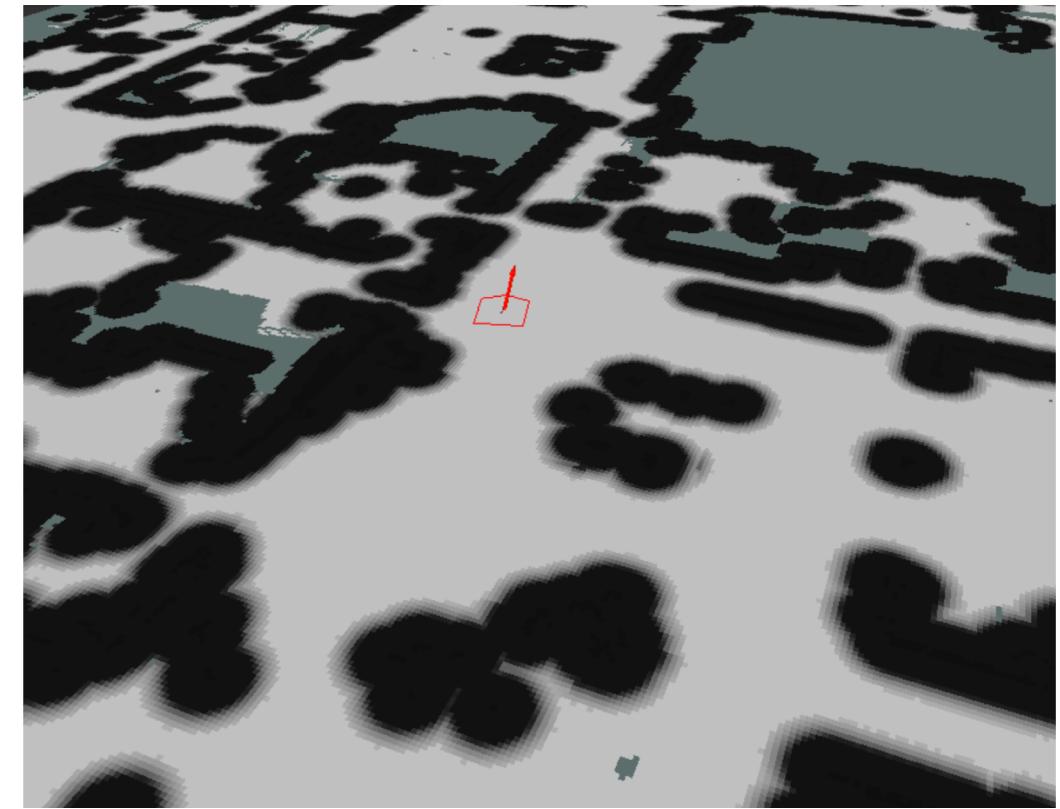
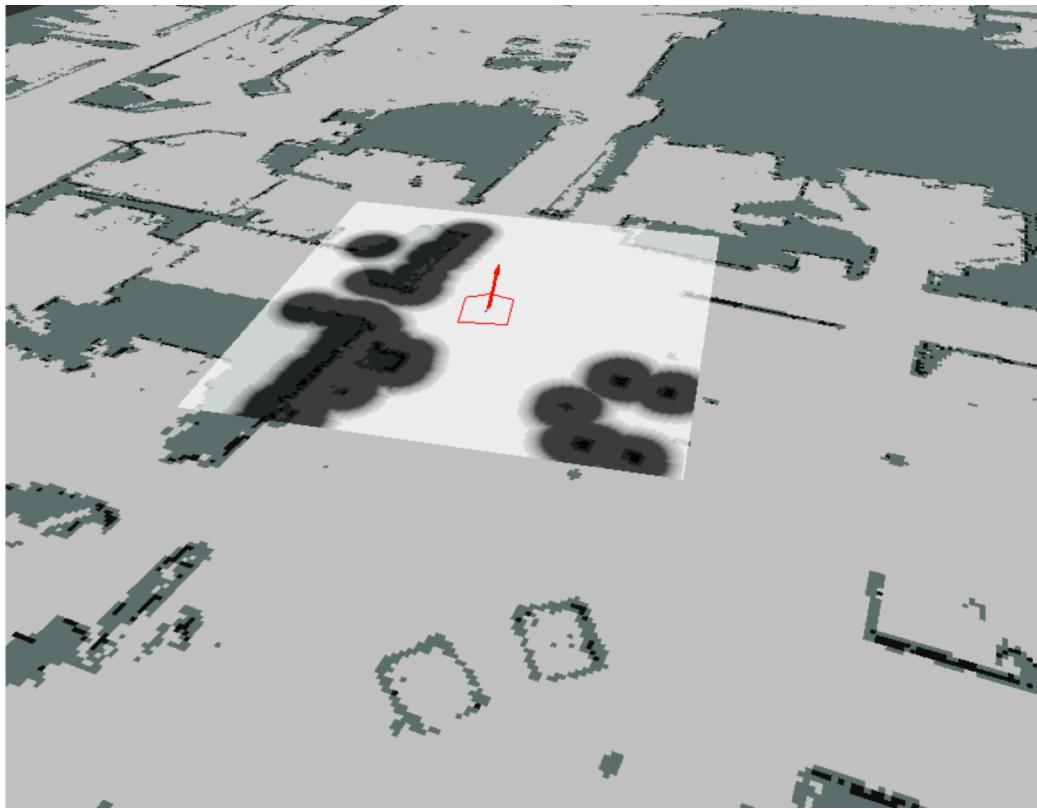
- Die Costmap wird in fixen Intervallen aktualisiert
(Parameter: update_frequency)
- Updateverfahren:
 - Sensordaten werden aufgenommen
 - Belegungsstruktur wird anhand der Sensordaten angepasst
 - Costmap wird mit der Belegungsstruktur aktualisiert
 - Inflation wird für jede Zelle ausgeführt in der ein LETHAL_OBSTACLE (Tödliches Objekt) liegt
 - Anpassung der umliegenden Zellen



RVIZ Navigation

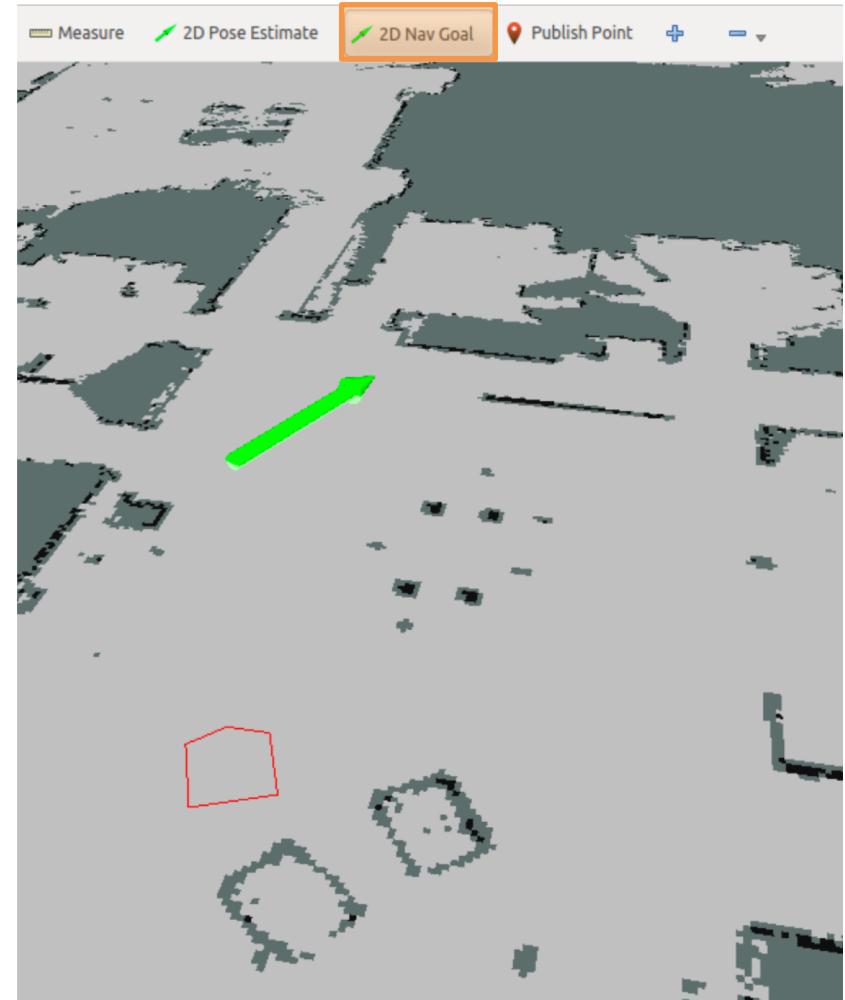


Costmaps in RVIZ

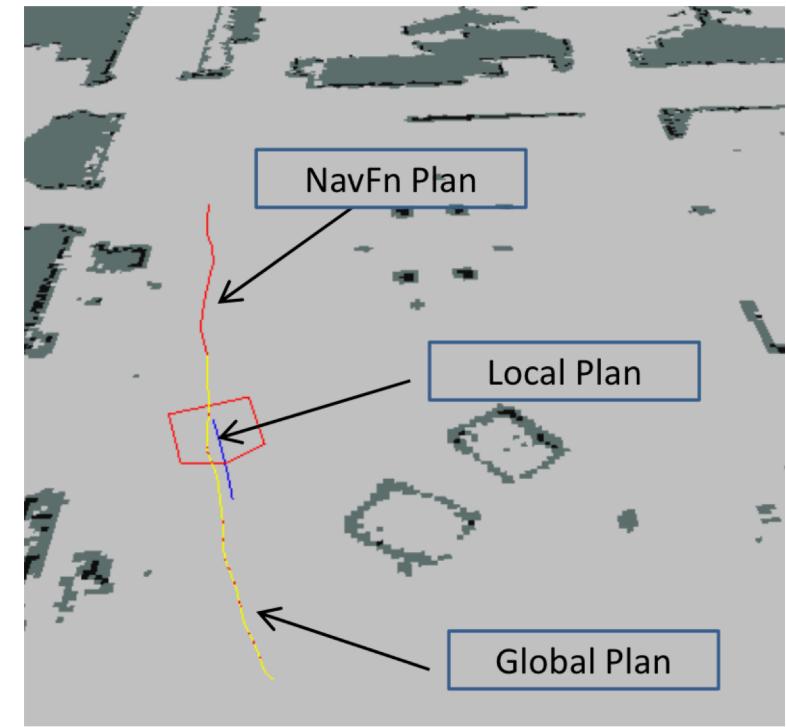


2D Nav Goal

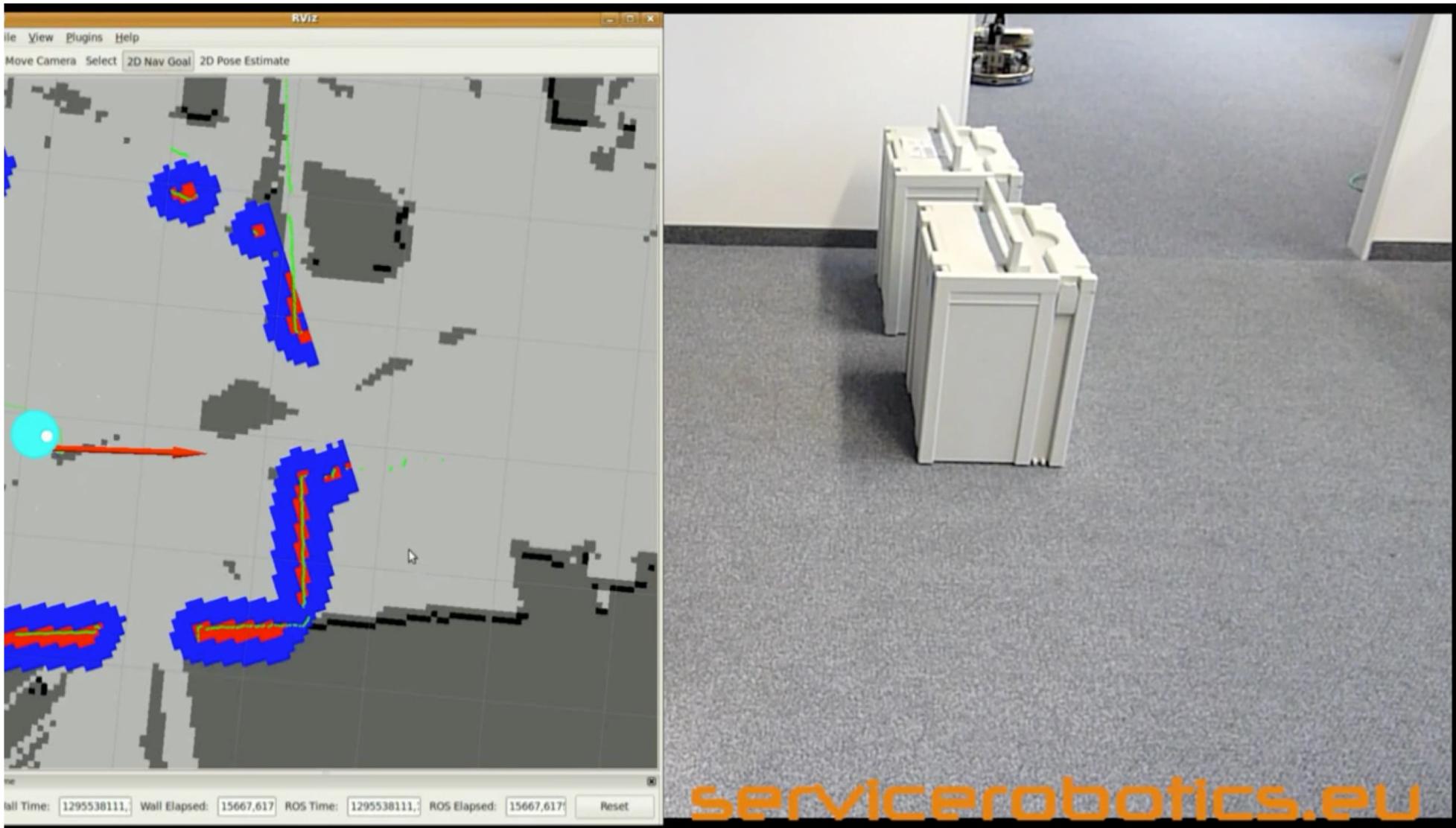
- Der 2D Nav Goal Button ermöglicht das Setzen von Zielkoordinaten für die Navigation
- Der Nutzer kann die x- und y-Koordinaten bestimmen und anschließend die Orientierung mithilfe des Pfeils bestimmen
- WICHTIG!
 - Fixed Frame muss auf /map gesetzt werden



- NavFn Plan
 - Gesamter Plan des globalen Planers
 - Topic: /move_base_node/NavfnROS/plan
- Globaler Plan
 - Ausschnitt des globalen Plans welchen der Lokale Planer gerade verfolgt
 - Topic:
/move_base_node/TrajectoryPlannerRos/global_plan
- Lokaler Plan
 - Momentaner Plan
 - /move_base_node/TrajectoryPlannerRos/local_plan

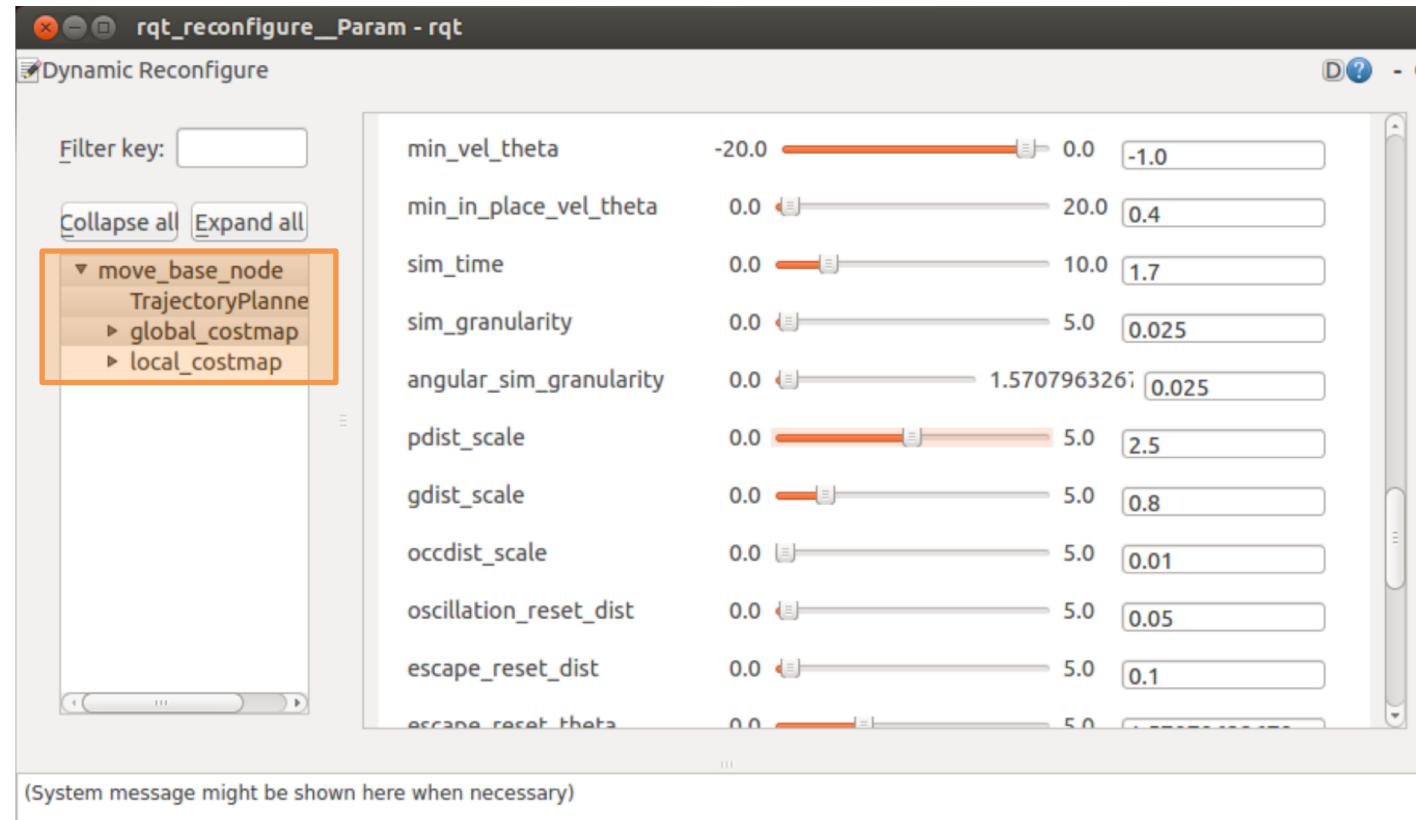


RVIZ Navigation



rqt_reconfigure

- Tool zum Anpassen der Konfigurationen
 - `$ rosrun rqt_reconfigure rqt_reconfigure`



Youbot ROS Navigation Stack

- Starten des Navigation Stacks für den Youbot:
 - `roslaunch youbot_navigation_common bringup_navigation.launch`
 - `roslaunch youbot_navigation_global 2d slam.launch`

