Robot action planning and execution

Project background

- ROS

- Motion planning

- 3D Simulation

LUND UNIVERSITY

ROS (Robot Operating System)

- Open source

- Available on Linux (preferably Ubuntu) and Mac OS X Platforms

- Provides all the services expected from an operating system

- It is language independent, meaning that it is relatively easy to implement in any modern language

- The main goal of ROS is to support code reuse in robotics research and development

How does it work?

- Nodes are all the different processes (wheel motors, lasers, path planning, etc)

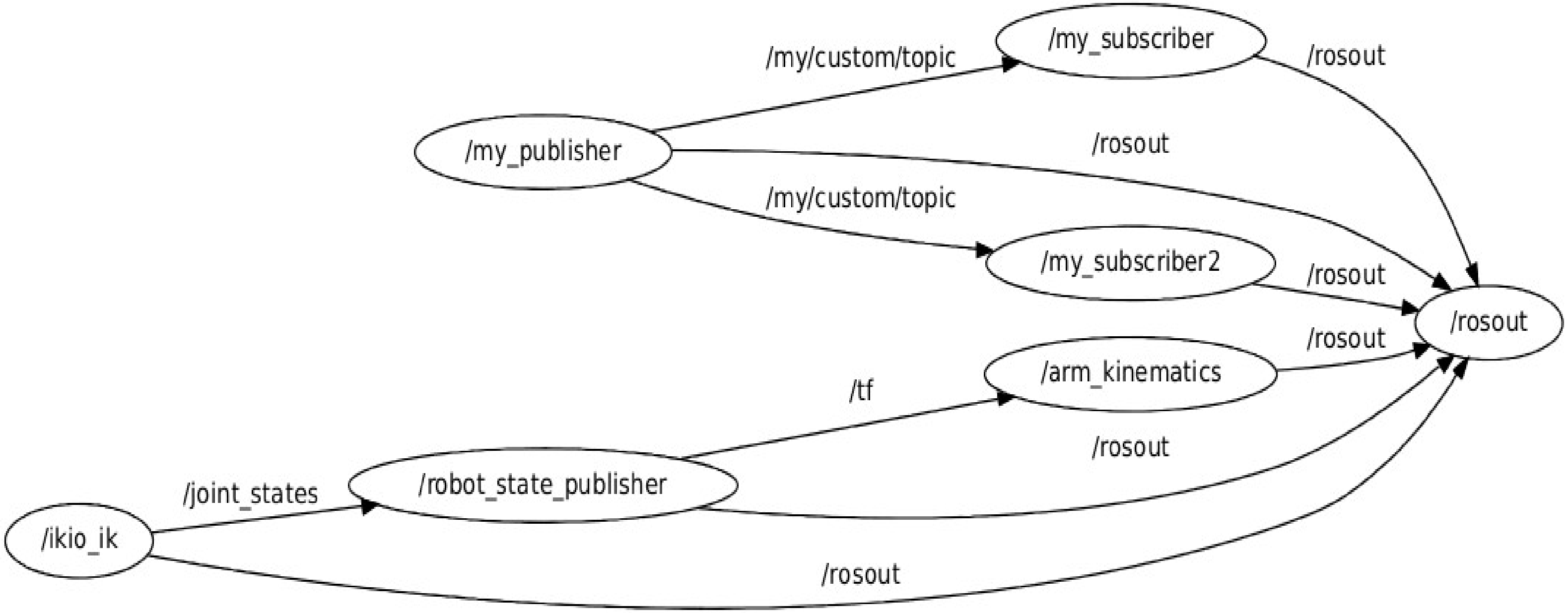
- The master server allows communication between the nodes

- The nodes communicate between each other by sending messages

- The node publishes a message to a given a topic, another node subscribes to messages from the same topic

LUND UNIVERSITY

ROS Framework

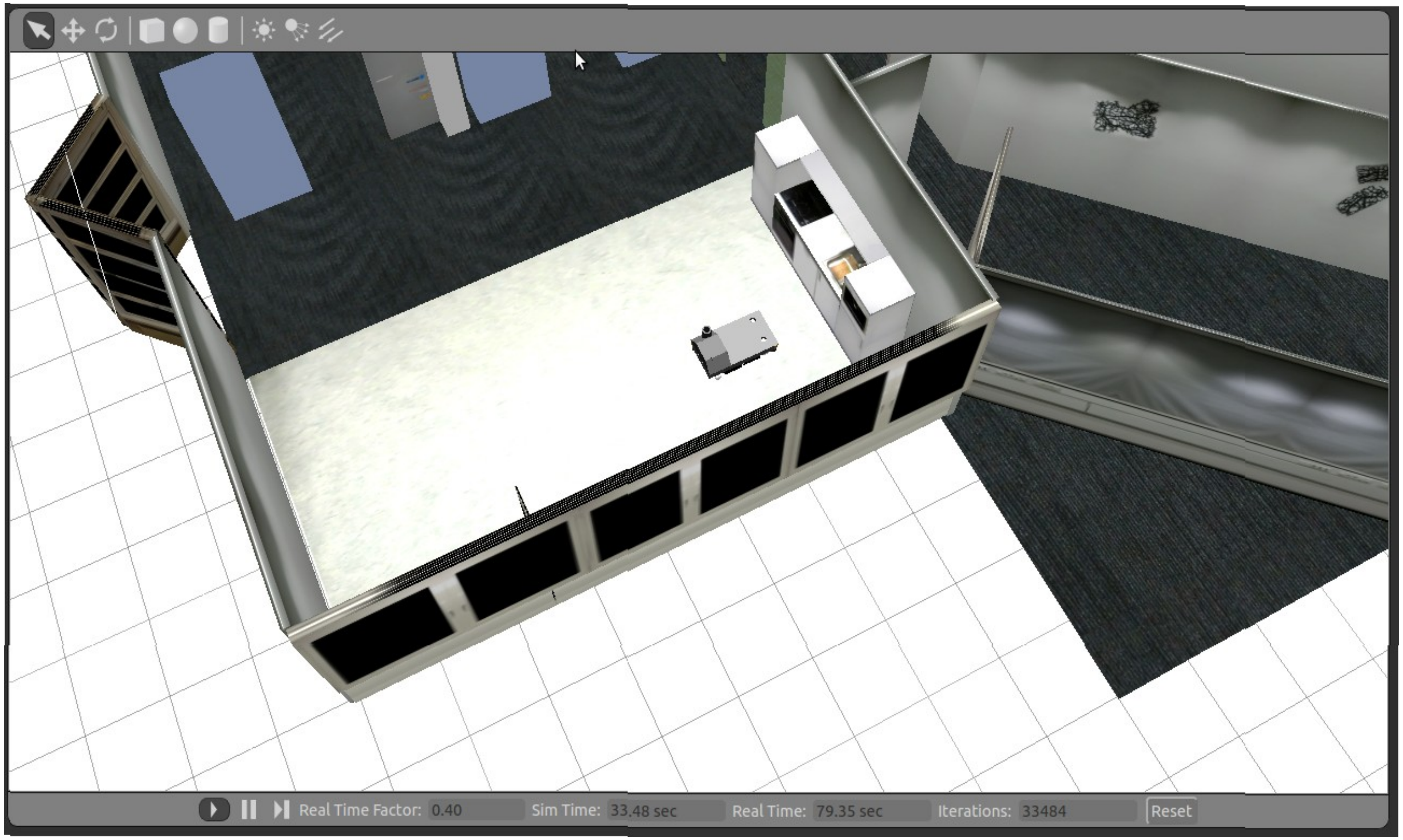


Rviz

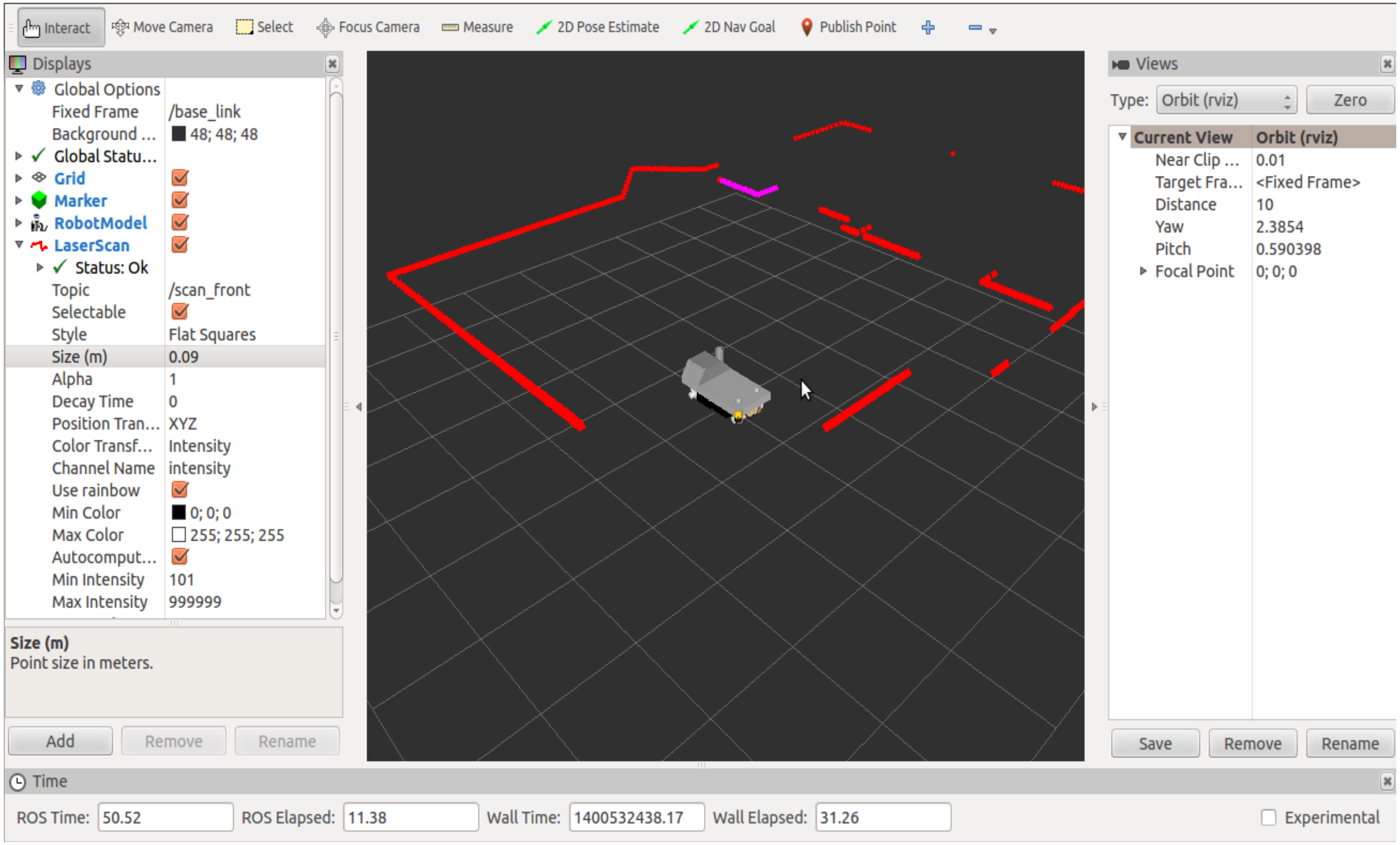
- 3D Visualization tool for ROS

- Allows to visualize messages published by nodes in ROS, eg: Laser scan, map, etc.

Simulation in Gazebo (3D Simulator)



Visualization in rviz



Motion planning

- Process of breaking down a task into different specific goals that satisfy the movement constraints

- Actions like avoiding walls, obstacles, reaching objects on tables…

- The purpose is to convert high-level human specifications tasks into low-level detailed Robot tasks.

LUND UNIVERSITY

Planning involves

- **State space:** Possible situations.
- **Time:** Sequence of decisions applied over time.
- **Action:** Manipulate the state and how it changes.
- **Initial and goal states**
- **A criterion:** Desired plan (feasibility and optimality)
- **A plan:** Specific strategy, behavior or decision maker.

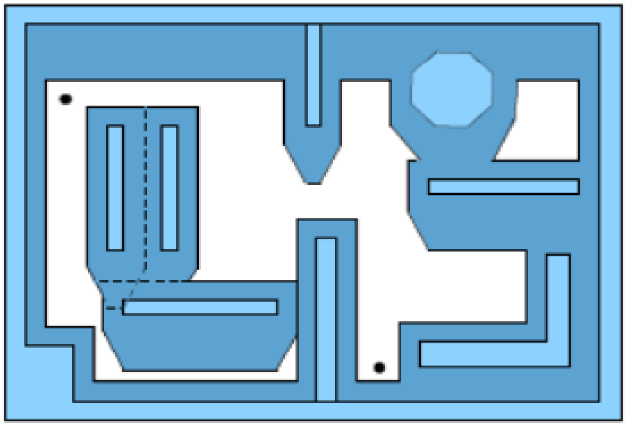
LUND UNIVERSITY

Robotic mapping

- Capability of a robot to be able to construct or use a floor map and to localize itself in it.

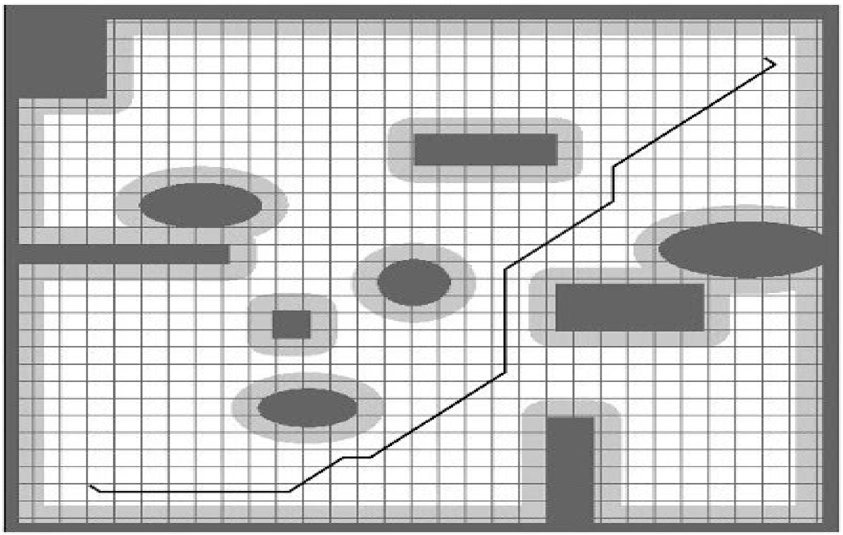
- It corresponds to the sensors of the robot an its knowledge of the environment (camera, laser scanner, GPS).

C-Space transformation.

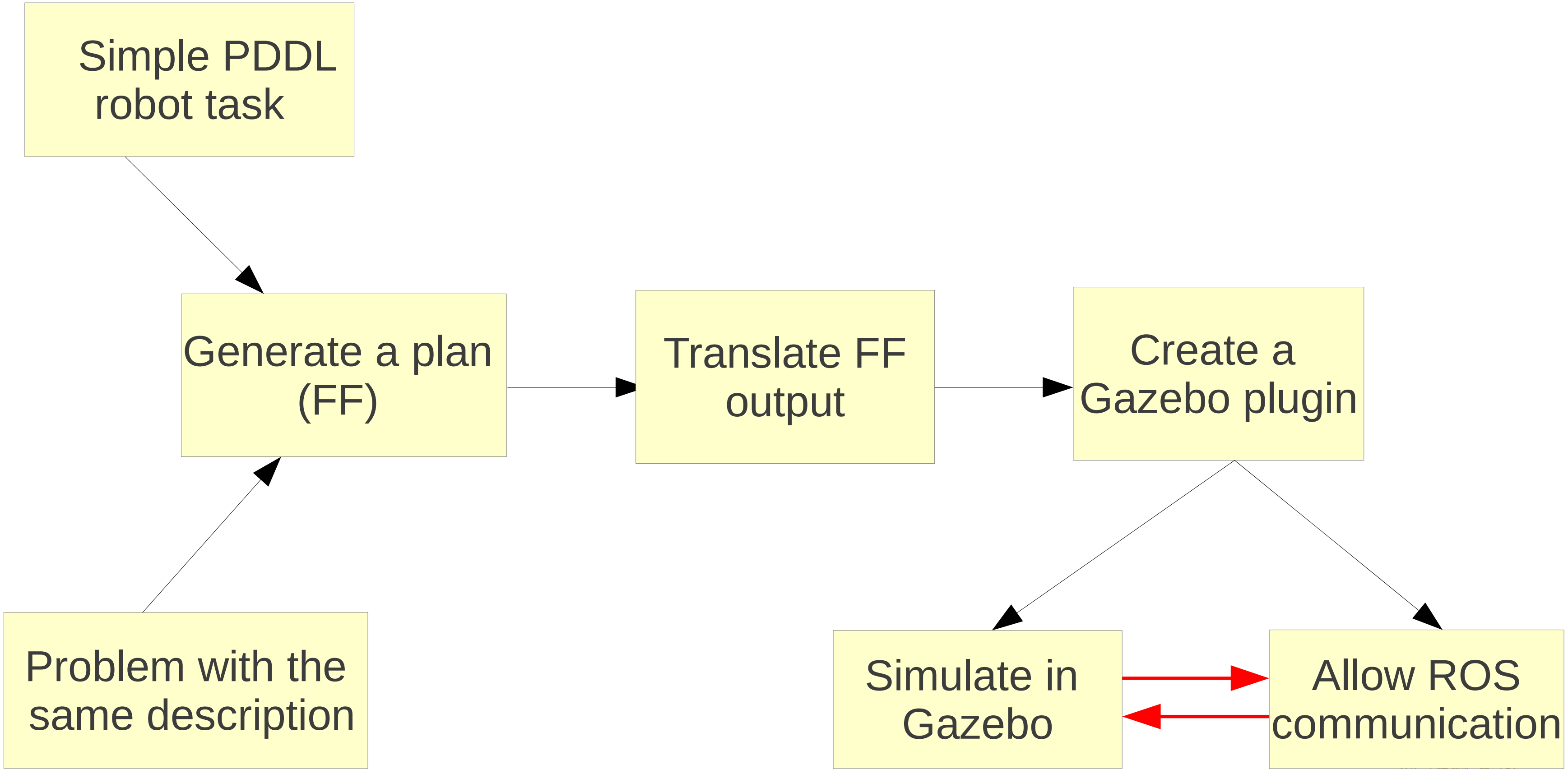


Sampling-Based Motion Planning

- Cell decomposition.
- Define a grid using coordinates.
- Path planning.



Project Goal



LUND UNIVERSITY

PDDL Description (STRIPS)

Actions

- Move
- Fetch
- Fetch from obstacle
- Lift arm
- Lower arm
- Deposit

Problem description

- 8 x 8 Grid

- 5 Obstacles randomly located in the world

- 10 Objects, 5 located on top of the obstacles and other 5 randomly located in the world

- Robot starts in a given position (usually lower left corner)

- There is a box in a random position

- The goal of the robot is to collect all the objects and deposit them in the box

Plan (FF)

```
 0: MOVE ROBOT SQ-1-1 SQ-1-2
 1: MOVE ROBOT SQ-1-2 SQ-1-3
 2: MOVE ROBOT SQ-1-3 SQ-1-4
 3: MOVE ROBOT SQ-1-4 SQ-2-4
 4: MOVE ROBOT SQ-2-4 SQ-2-5
 5: FETCH ROBOT OBJ_10 SQ-2-5 SQ-2-6
 6: MOVE ROBOT SQ-2-5 SQ-3-5
 7: MOVE ROBOT SQ-3-5 SQ-3-4
 8: DEPOSIT ROBOT OBJ_10 SQ-3-4 SQ-3-3 BOX
 9: MOVE ROBOT SQ-3-4 SQ-3-5
10: MOVE ROBOT SQ-3-5 SQ-2-5
11: LIFT_ARM ROBOT
12: FETCH_FROM_OBSTACLE ROBOT OBJ_5 SQ-2-5 SQ-1-5
13: MOVE ROBOT SQ-2-5 SQ-3-5
14: LOWER_ARM ROBOT
15: MOVE ROBOT SQ-3-5 SQ-3-4
16: DEPOSIT ROBOT OBJ_5 SQ-3-4 SQ-3-3 BOX
```

LUND UNIVERSITY

Translation

Movements: How does the robot measure just 1 square?

Arm movement: How does the robot know how much it has to lift its arm to reach the object?

Object distance: How does the robot know how far is the object from the gripper?

C++ Gazebo plugin

Once we have the translation, the next step is to create a gazebo model plugin that allows to modify the robot mobile base and joints so that it moves its arm precisely to obtain all the objects from the world

References

- Wiki.ros.org

- Minoshia.files.wordpress.com

- Robot Motion Planning Steven M. LaValle presented by Vera Bazhenova (Max Planck institut informatik)

- Jaulin, L. (2001). "Path planning using intervals and graphs“

- Planning algorithms Steven M. LaValle, Cambridge University

LUND UNIVERSITY

LUND

UNIVERSITY