UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Intelligent Robotics

Group 25

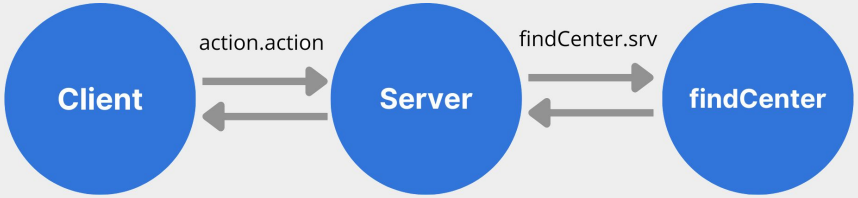
David Petrovic

Federico Luisetto

Umberto Salviati

Assignment 1: main parts

- Robot navigation
- Table detection
- Table center



Robot navigation

We used the action implemented in move_base package that allows Tiago to move to the desired position by using the navigation stack

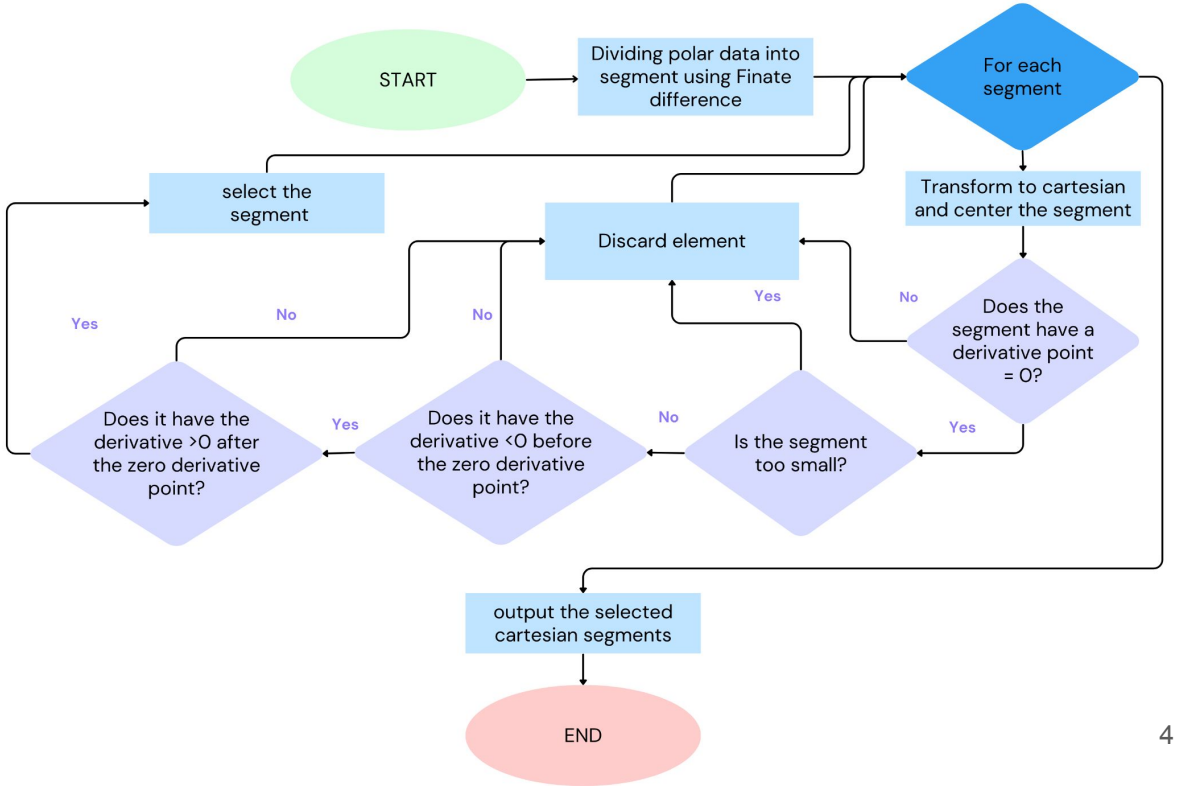
We created an action client that sends to the action server the pose (goal) we want the robot achieves

The goal is defined with respect to the reference frame named “map”

Table detection - flow chart

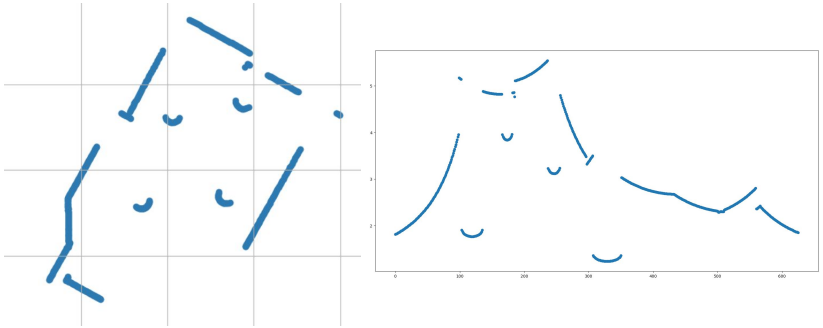
- Derivative on polar data to find segments
- Derivative on cartesian data to find concavity

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

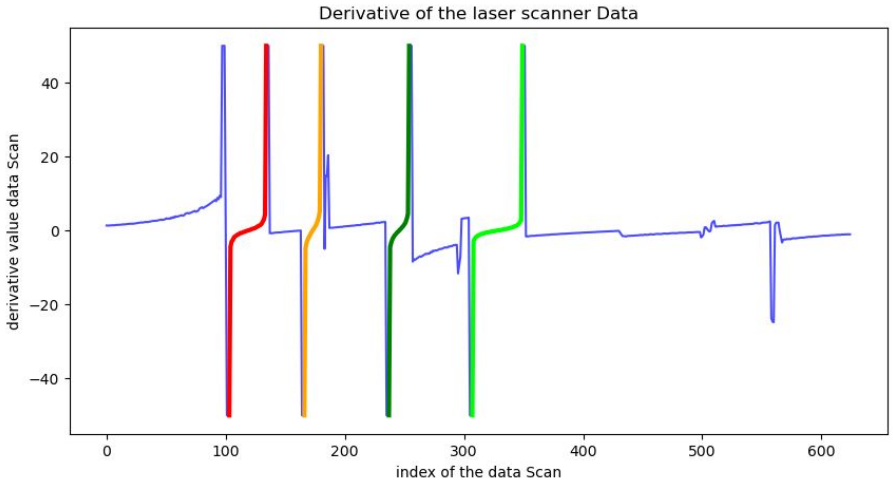


Why segmentation on polar data

- Segmenting the laserScan data then distinguish walls against tables
- Derivative on polar data because it has no overlap

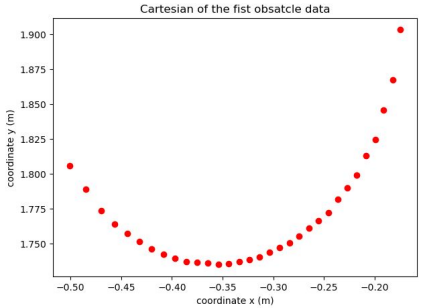


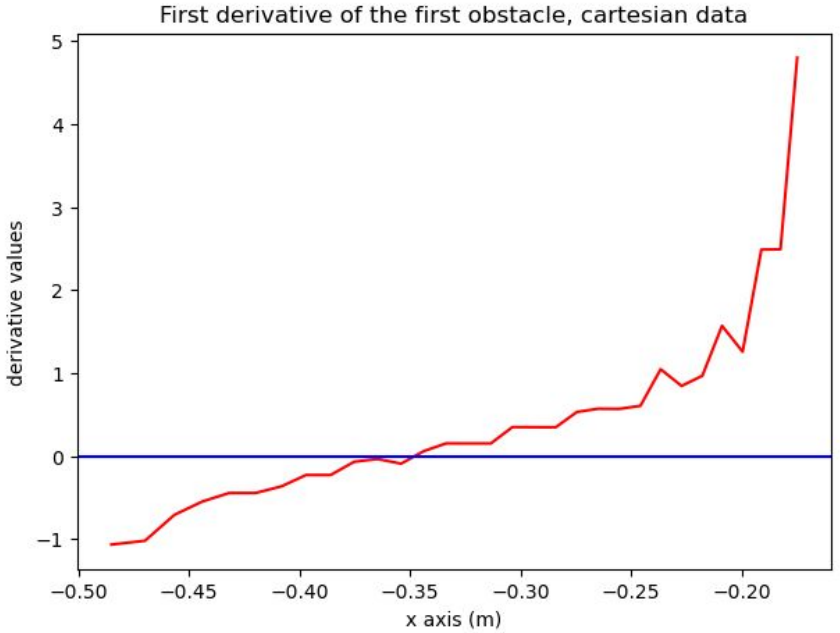
cartesian data vs polar data:
no overlap on polar data



Detecting concave shapes

- Derivative on smoothing data
- Getting only data with low number of consecutive =0 derivative
- Check if the derivative is always increasing





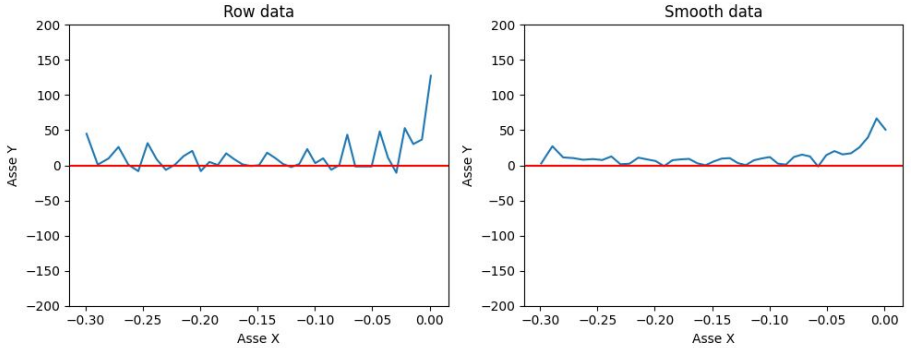
Pros and Cons

Pro

- Sensitivity:
  - Accurately detects small changes for precise circle detection.
- Quick to Compute:
  - Efficiently processes data.
- Easy Math Involved:
  - Uses simple mathematical concepts for easy implementation.
- General Purpose:
  - Flexible and adaptable due to no hardcoded values.
- Distance Independent:
  - Detects circles regardless of their distance from the scanner.
- Space Independent:
  - Works in any spatial orientation.

Cons

- Noise Sensitivity:
  - May lead to false detections if data has a lot of noise.
- Complex Shapes:
  - May struggle with complex or irregular shapes that aren't perfectly circular.



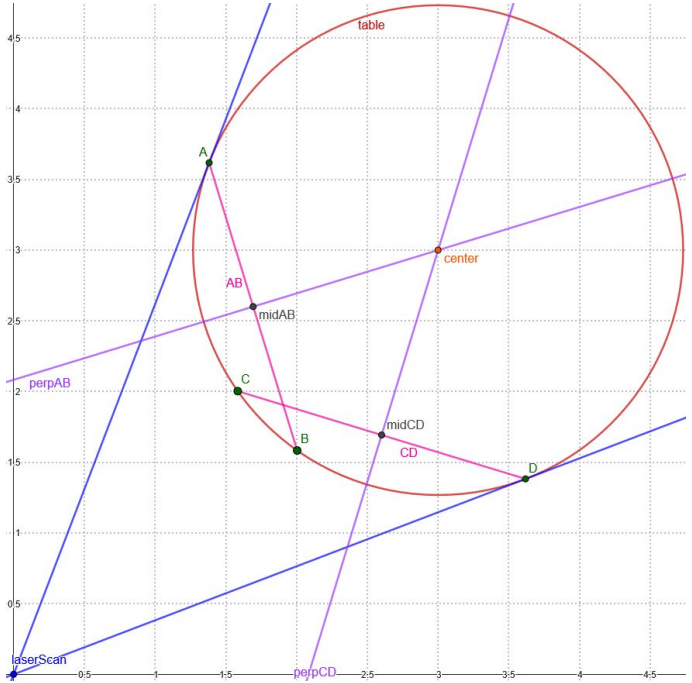
second derivative plot

Center of tables

Algorithm used:

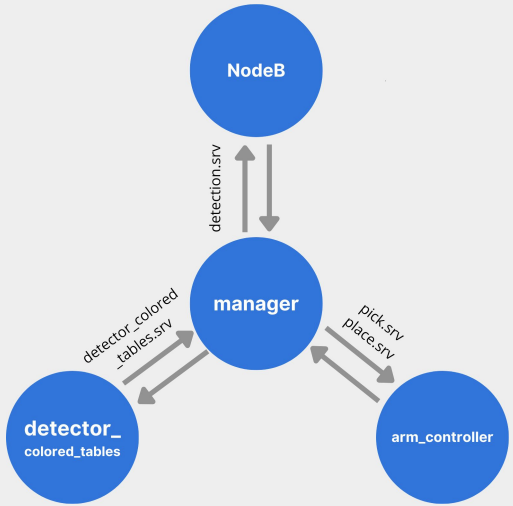
1. from laserScan, define A, B, C, D.
2. segments AB and CD.
3. middle points and perpendicular straight lines.
4. intersection of perpAB and perpCD defines center.

→ *sensitive to noise data*



Assignment 2: main parts

- robot navigation
- tables detection
- objects detections
- pick-and-place phase



Cylindrical tables detection

This task is realized by combining these two phases:

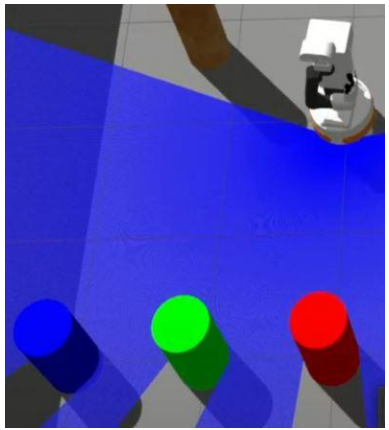
- in the first one, we get the position of the center of the tables by using the implementation of assignment 1 for the cylindrical tables detection task

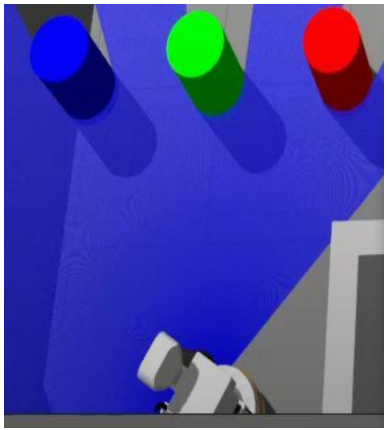
- by working with the camera equipped on Tiago, we implemented an image filtering approach to understand the order in which the three colored tables appear in front of the robot

Cylindrical tables detection with laserScan

We compute the average between two scans to have more accurate positions of the centers of the tables

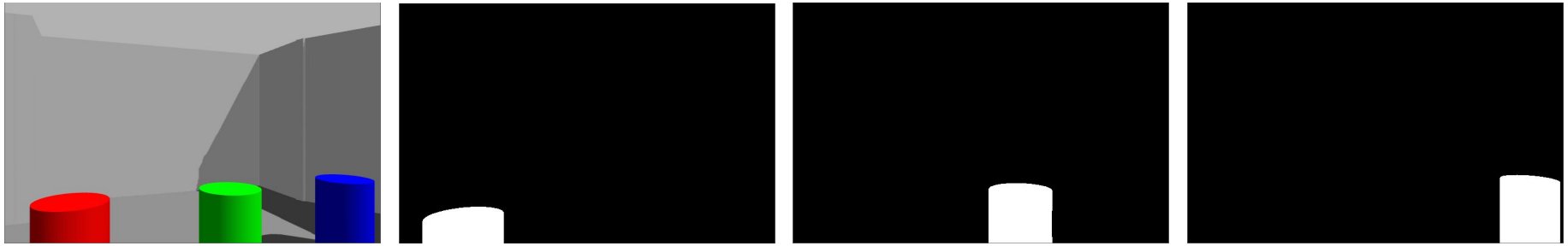
→ *increasing robustness.*





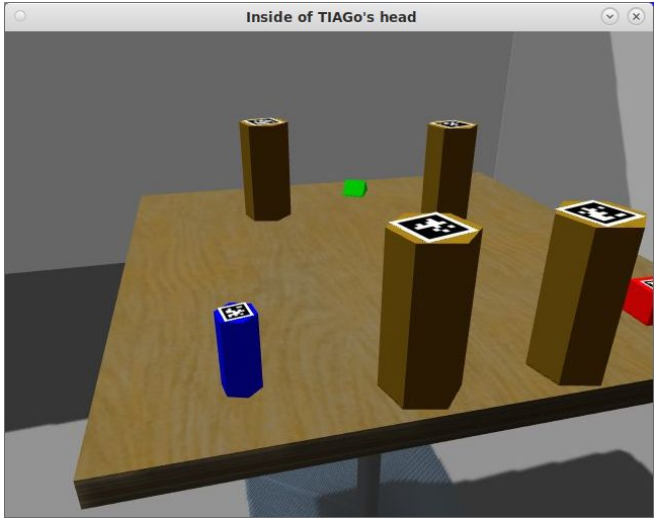
Cylindrical tables detection with color filtering

1. Filter the original image by color
2. Calculate the per-element bitwise logical operation for extracting the masks
3. Compute the mean point over the white pixels for each image and compare them



Objects detection with aprilTag

1. Move the robot to a fixed position near the table
2. Lower Tiago's head to allows the camera to point to the objects on the table
3. Exploit the topic *tag_detections* to get the IDs and poses of the objects
4. Transform the obtained pose from the camera frame to base_footprint



Motion planning - Pick phase

Arm routine for pick phase:

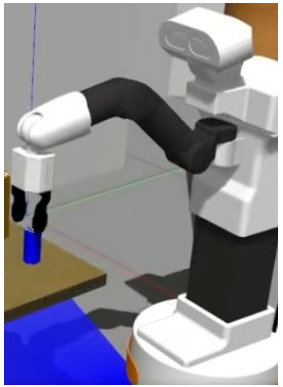
1. Open gripper

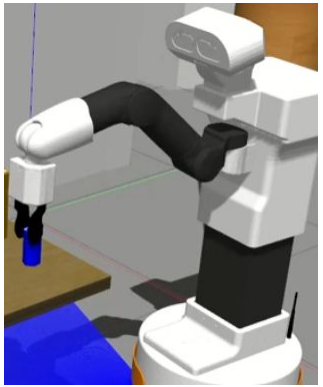
2. Arm in pre-grasp pose
   → pose from goal, offset in *z* axis
   a. Remove object to pick from collision objects

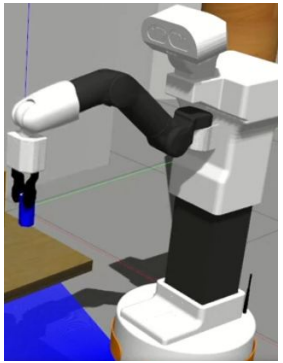
3. Arm in goal pose
   a. Close gripper
   b. Attach arm link and object link

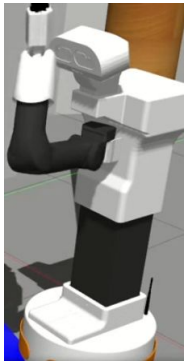
4. Arm in post-grasp pose
   → pose from pre-grasp, offset in *z* axis

5. Arm in safe pose
   → improvement: joint space









Motion planning - Place phase

Arm routine for place phase:

1. Arm in place pose
   → static movement because table always at certain distance
   a. Open gripper → before detaching otherwise “sticky” obj
   b. Detach arm link and object link

2. Arm in safe pose

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Conclusion

- We managed to implement everything and create a general purpose algorithm to detect colored tables and to pick-and-place objects (excluding waypoints)
- On the other hand, our solution is sensitive to noisy data, specifically during the tables detection and centers calculation tasks

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Intelligent Robotics

The end

Group 25

David Petrovic

Federico Luisetto

Umberto Salviati