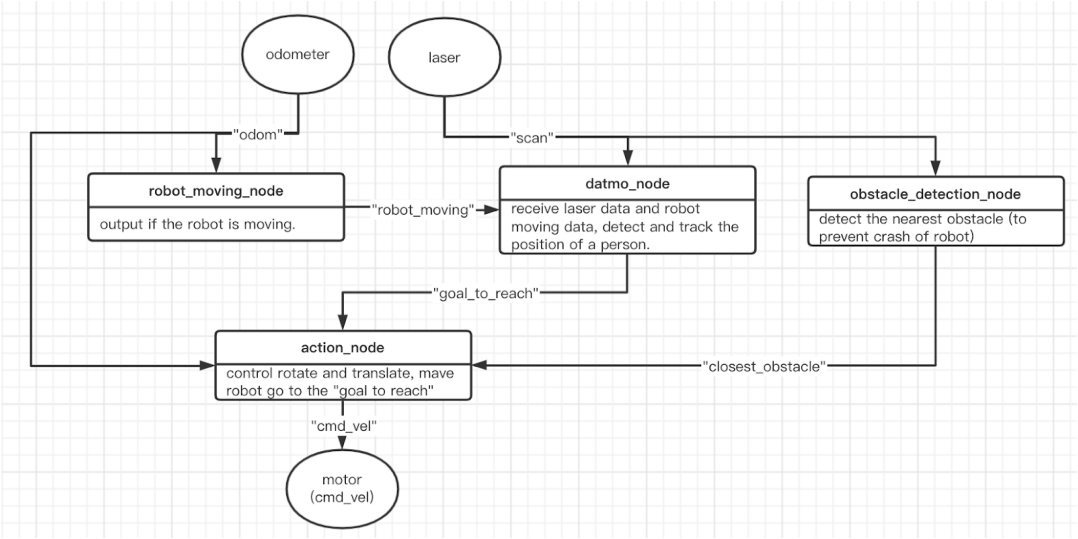
Report of Follow Me

1. ROS Nodes



2. Interesting parts and Difficulties

a. Difference of “datmo_node” and “detection_node”

During testing process, we started “detection_node” instead of “datmo_node” for a long time, and we observe that robot cannot follow a person. It goes to the desk, the wall, other persons. Then hits them. After observing RVIZ, we find that really too many objects are recognized as person. Even the wall behind a person (because we use absolute value of difference to decide if a laser ray is dynamic). So the tracking algorithm is important here. Only with this, the robot can focus on some exact target and ignore other noise.

b. Combine rotation and translation

The variable “coef_rotation” and “coef_translation” smartly combine move and translate smoothly, without sudden stop during moving. At first we haven’t fully understand these coefs, we let **real translation speed = translation speed max * coef**. The robot is hard to brake. Actually **real translation speed = current speed(translation_to_do) * coef**. With this calculation, the robot can stop now.

We didn’t implemented real PID control for rotation and translation. Because we didn’t have enough time to adjust the arguments Kp, Ki, Kd. We use “translation_to_do” as “current speed”, the robot already works well.

3. Tests

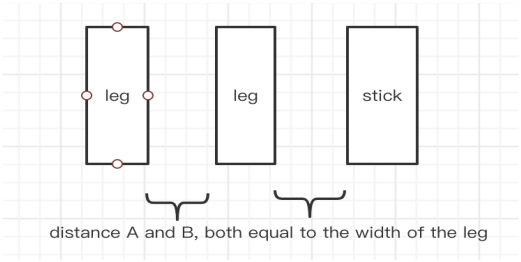
a. Test1

i. The goal of the test:

Evaluate the “detect_motion” function in datmo_node.
This function generates a “dynamic” array using the absolute value of the difference between background scan and current scan. The array will influence the detection of a moving person. The test will vary the input of “detect_motion”, and see the final influence on the moving of the robot.

ii. Test scenario

Take a stick together with two legs, make the two gaps both the same size as a leg. And observe if the robot treat the gaps (the wall behind) as a person.



iii. Results and observations

We tested about 5 times. Most of the time the robot works normally. It tracks the middle of two legs, or the middle of a let and the stick. If one leg or one stick is removed (lifted), the robot will continue tracking the middle of other two objects.
Only one time the robot seems didn’t perfectly brake. It hitted shoes.

Rosbag address:
/home/student-3/Documents/rosbag_file_test/test1.bag

iv. Analysis

We expected that the robot sometimes recognizes the background as person (because of the absolute value), track it, and hit real person’s shoes. But in our test, this rarely happens(for the only one time we still not sure if it is really in this case). The reason can be that, although the background is sometimes recognized as person, the tracking algorithm is so powerful that it makes the robot focus on the real person, who occurs from the beginning and much more frequent.

v. Improvement

No. Good enough.

b. Test2

i. The goal of the test:

Evaluate “frequency” variable in “track_a_moving_person” function in datmo. Normally, if a person disappears, the variable starts decreasing. Before it reaches 0, the robot will keep trying to track the same person (in the “uncertainty” region). After it reaches 0, the robot will re-detect another person to track. So we can make two situations in which person disappears for different time, corresponds to “frequency” reaches 0 or not.

ii. Test scenario

Let two persons in the view of the robot, then

1. The tracked person jumps at origin position.
2. The tracked person jumps to the back of robot.

Then observe after how long time robot starts tracking the other person.

iii. Results and observations

We tried about 2 times for each type of jump. In type 1 (jump at origin position), the robot continue tracking the same person. In type 2(jump to the back of robot), the robot start tracking another person.

Rosbag: /home/student-3/Documents/rosbag_file_test/test2.bag

iv. Analysis

Perfectly performs as we assumed. The “frequency” works well.

v. Improvement: (NO)

c. Test3

i. The goal of the test:

Evaluate the “combine_rotation_and_translation” function in action_node. When rotation_to_do is larger than rotation speed max, the “coef_rotation” variable will keep to be 1. After some rotation, rotation_to_do is smaller. “Coef_rotation” gets smaller and “coef_translation” gets larger. Normally, the robot should first rotate then gradually switches to translate mode.

ii. Test scenario

When the robot is in front of the tracked person, the person quickly moves to the back-left direction for (1, 2) meter. (move left more than move back)

iii. Results and observations

We tried about 2 times. The robot works as assumed (rotate then translate)

Rosbag: /home/student-3/Documents/rosbag_file_test/test3.bag

iv. Analysis

The function “combine_rotation_and_translation” works as assumed.

v. Improvement

Implement real PID in rotation and translation.

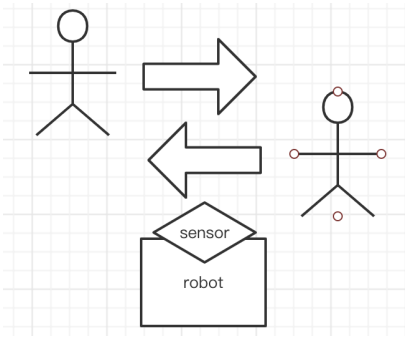
d. Test4

i. The goal of the test:

When we were running the robot in the crowded classroom, the robot always lose its tracking target. Because the tracking algorithm is based on clustering, we try to make difficulties to clustering, to get a clear situation where the tracked person will be lost.

ii. Test scenario

One person goes from the left-front to the right-front of the robot, the other goes from the right-front to the left-front. They cross at the front of the robot. Observe whom the robot will follow.



iii. Results and observations

It depends on whom is previous tracked by the robot.
If the nearer person is tracked previously (we tried 3 times for this condition), the cross of two persons will not lead the lost and switch of target. The robot will follow the same person.
If the farer person is tracked previously (4-5 times for this condition), there is about 50% probability the robot changes target to another person (who is nearer) when two persons cross each other.

Rosbag: /home/student-3/Documents/rosbag_file_test/test4.bag

iv. Analysis

In “track a moving person” function, the person in “uncertainty” with the minimum distance to the previous tracked person will be tracked. So when the tracked person is in front, he is always in the robot view and always the nearest one to be tracked. If he is in back, he will be probably hidden by another person (depends on if another person is really detected when the original tracked one is hidden). Then the minimum distance person will be this another person. By this way, the tracked person is switched.

v. Improvement

Reduce the max size of “uncertainty” circle could be a way.

e. Test5

i. The goal of the test:

When we were running the robot, it sometimes only can follow a moving person, but sometimes it can follows static person. We want to test to make sure the relationship between dynamic person and tracking.

ii. Test scenario

Let the tracked person move, then stop, then cross his two legs. While a second person keeps moving in the view of robot.

iii. Results and observations

When the tracked person is not moving, robot will continue track him (although the other person is moving).
When the tracked person cross his legs, the robot woll lost him and start tracking another person. (Tested about 2 times, hasn’t counted exactly)

Rosbag: /home/student-3/Documents/rosbag_file_test/test5.bag

iv. Analysis

We look at the detection logic carefully again. We found that in “detect_a_moving_person”, it only accepts a dynamic person. But in “track_a_moving_person” we donot have this limit. So the person only need to be dynamic when he starts to be noticed by the robot. If he is successfully tracked, his moving or not will not matter any longer.

v. Improvement

No. This test is just an observation.

Localization Report

i. The goal of the test:

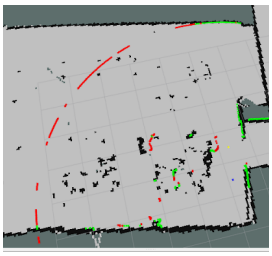
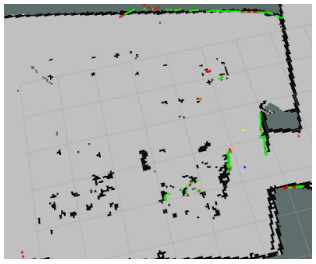
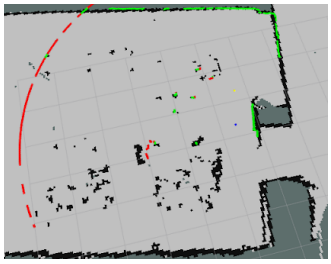
The goal of the test is to determine and implement the localization of Robair. Robair with the aid of the local scanner tries to determine its position by comparing its initial position based on the map and the predicted position. Then Robair will go through many translations to the predicted position with respect to the model in rviz.
The motion of Robair to the predicted position is compared with the best scores of what is the observed position. The scores are obtained as a comparison between the observed values of the laser scan and the map.

ii. Test scenario

Load the map in ros by opening the folder of the map, rosrun the map map_server map_server “name_of_the_map”.yaml. Display the map and localization process in rviz. In rviz, use the “2D pose estimate” and choose a position in the map to ensure the orientation of the robot. Get the initial position of the robot on the map in rviz. Note the predicted position. Compute the match between the current laser data and the map.

iii. Results and observations

During our test(without the robot) with rosbag, we saw in the map on rviz the blue and yellow color indicating Robair orientation and laser hit respectively. For every laser hit the mobile robot moves an estimated distance of 1 meter. As you will see in the screenshots provided, the initial position is noted, and the predicted position is also estimated and as soon as the localization commences, we can observe that Robair will move to the yellow dot indicated on the map with a best possible score.

iv. Analysis

The estimated distance in our case for the first motion is 1.002524 meters after which the robot stops moving to relocalize. After each laser scan hit, it will check if it matches the position of the map with its current position with a score. The indicator of the accuracy of the score is marked by the green borders on the map on rviz.