

AA274A Section 5: Point to Point Navigation

Brian Dobkowski

AA274A, Stanford University

BDOBKOWS@STANFORD.EDU

1. Question 1

What topics are subscribed to by the robot:

1. **map** - this enables the robot to receive new map information and maintain an updated map of the environment
2. **map_metadata** - receives map metadata like height, width, grid resolution, origin location to store and maintain this data onboard the robot
3. **cmd_nav** - loads in a goal from the navigation commands if that goal is different from the current goal. Used to maintain a goal state x_g , y_g , and θ_g that the robot can plan towards achieving.

What topics are published by the robot:

1. **planned_path** - publishes the planned robot path for visualization. This includes the state of the robot (x, y, θ)
2. **cmd_smoothed_path** - after the planned path has undergone smoothing, (e.g. splines or another smoothing method after grid planning), the resulting trajectory is published to this topic to provide the robot state's along the trajectory for visualization
3. **cmd_smoothed_path_rejected** - this topic is published to when a new plan is created and rejected because its duration is longer than the current plan. The robot will continue to follow the current plan if a newly planned path is longer.
4. **cmd_vel** - the robot publishes its commanded velocities (both translational and angular velocities) to this topic when being teleoperated by an operator.

2. Question 2

Modes of the machine:

1. **IDLE** - robot does nothing, no replanning, no control is run on the robot
2. **ALIGN** - runs the heading controller, which tries to align the θ state of the robot to its desired value. This may be switched to if the robot gets off track and needs to be aligned to get back to its trajectory, or if we want the robot to face a certain direction before it starts moving or planning. The robot switches to mode TRACK when it is aligned.

3. **TRACK** - runs the closed-loop trajectory tracking controller we implemented in HW3 to follow a prescribed trajectory with pre-specified gains. In this mode, the robot also continually replans as it moves along its trajectory. The robot switches to mode PARK when it is near its goal.
4. **PARK** - runs the pose controller to get the robot to exactly be in the goal state. This mode is used when the robot is near the end of its trajectory and needs to scurry around for a few seconds to reach its goal state. The trajectory control will not get it exactly in the goal state. After parking is successfully complete, the robot switches to mode IDLE.

3. Question 3

Command to create a new catkin package: `catkin_create_pkg section5 std_msgs rospy message-generation`

The `section5` name is new for this package we just created. The `std_msgs` is a dependency that allows us to use the ROS standard messages in this package, and the `message-generation` is a dependency that allows us to create messages and publish/subscribe to messages. `rospy` is a dependency that tells we are using Python with ROS.

4. Question 4

Figures showing our robot moving to the 2D Nav Goal using the controllers we developed in HW.

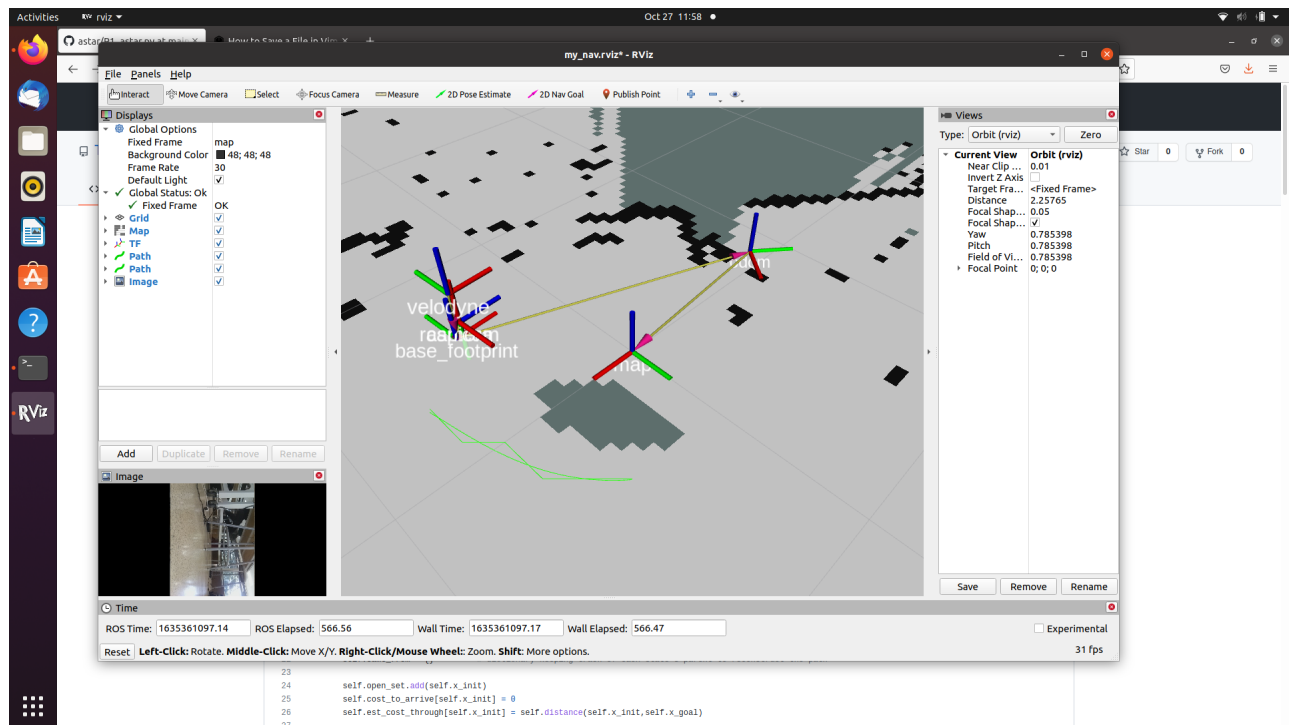


Figure 1: Rviz visualization of robot moving to goal

```
[INFO] [1635361076.521853]: Planning Succeeded
[INFO] [1635361076.552379]: Path too short to track
[INFO] [1635361076.583537]: Switching from Mode.PARK -> Mode.PARK
[INFO] [1635361080.449452]: Navigator: computing navigation plan
[INFO] [1635361080.518908]: Planning Succeeded
[INFO] [1635361080.548198]: Ready to track
[INFO] [1635361080.570042]: Switching from Mode.PARK -> Mode.TRACK
[INFO] [1635361083.092220]: replanning because far from start
[INFO] [1635361083.168923]: Navigator: computing navigation plan
[INFO] [1635361083.208791]: Planning Succeeded
[INFO] [1635361083.284963]: New plan rejected (longer duration than current plan)
[INFO] [1635361084.993679]: replanning because far from start
[INFO] [1635361085.107787]: Navigator: computing navigation plan
[INFO] [1635361085.186469]: Planning Succeeded
[INFO] [1635361085.268774]: New plan rejected (longer duration than current plan)
[INFO] [1635361086.002304]: Switching from Mode.TRACK -> Mode.PARK
[INFO] [1635361100.005002]: replanning because of new map
[INFO] [1635361100.028235]: Navigator: computing navigation plan
[INFO] [1635361100.084789]: Planning Succeeded
[INFO] [1635361100.107849]: Path too short to track
[INFO] [1635361100.129663]: Switching from Mode.PARK -> Mode.PARK
[INFO] [1635361126.581644]: replanning because of new map
[INFO] [1635361126.626202]: Navigator: computing navigation plan
[INFO] [1635361126.686105]: Planning Succeeded
[INFO] [1635361126.726600]: Not aligned with start direction
[INFO] [1635361126.757289]: Switching from Mode.PARK -> Mode.ALIGN
[INFO] [1635361129.202194]: Switching from Mode.ALIGN -> Mode.TRACK
[INFO] [1635361131.202106]: replanning because far from start
[INFO] [1635361131.239887]: Navigator: computing navigation plan
[INFO] [1635361131.272788]: Planning Succeeded
[INFO] [1635361131.296341]: New plan rejected (longer duration than current plan)
[INFO] [1635361133.402755]: replanning because far from start
[INFO] [1635361133.455299]: Navigator: computing navigation plan
[INFO] [1635361133.483740]: Planning Succeeded
[INFO] [1635361133.508779]: Path too short to track
[INFO] [1635361133.547762]: Switching from Mode.TRACK -> Mode.PARK
^C[turtlebot_navigator-1] killing on exit
shutting down processing monitor...
... shutting down processing monitor complete
done
aa274@aa274-taro:~/catkin_ws/src/asl_turtlebot/scripts/planners$
```

Figure 2: Terminal output from robot moving to goal

5. Question 5

Didn't have time for this problem and will complete in Section 6.

6. Question 6

Didn't have time for this problem and will complete in Section 6.