MRSD Project Course

Team I – AIce

Autonomous Zamboni Convoy

Progress Review 4

Team

Rathin Shah

Nick Carcione

Yilin Cai

Jiayi Qiu

Kelvin Shen

Author

Yilin Cai

April 07, 2022

Contents

1 Individual Progress

My responsibility for the Autonomous Zamboni Convoy project mainly focus on setting up and maintenance of the simulation, as well as integration and testing on RC car. My work also includes defining the whole software architecture, integrating different algorithm modules in both simulation and real RC car.

1.1 Localization integration in RC car

In follower autonomy simulation part, we have already developed and tested the software subsystems like state estimation, perception and control in simulation. In this progress view, I adapted all the algorithms developed for simulation to the real RC car hardware platform.

The localization of the RC hardware platform is important because both the motion planning and the controller will depend on it. So we first integrate the localization algorithm developed in simulation to the RC car. The encoder in wheel motor gives the rmp of the car wheels. Since all the wheels on the RC car are driven by only one motor, we think them share the same rpm based on which we calculate the vehicle velocity. The rpm of left and right wheel will not show any difference, so the encoder data can only give the distance the car has travelled. IMU data is then used to give the orientation of the car, which will then calculate the x, y, yaw. For IMU itself, we use the inbuilt imu module of the Realsense, which only publishes the angular velocity and linear acceleration. So I use the *imu_complementary_filter* package to convert the angular velocity and linear acceleration to actual orientation, as well as a filtering process. In general, the filter can fuse angular velocities, accelerations, and (optionally) magnetic readings from a generic IMU device into a quaternion to represent the orientation of the device with respect to the global frame. So the filtered IMU output gives the orientation of the RC car, which will then give the pose of the RC car.

Fig. 1 gives the testing process of the localization of RC car. I firstly teleoperated the RC car with keyboard by sending velocity and steering command. Then simultaneously, I visualize the actual path of the RC car in Rviz when the car is moving. We also measure the actual moving distance of the RC car and compared it with the odometry output. The error in x axis is less than 0.05m and the error in y axis is less than 0.2m.

1.2 Autonomous software integration in RC car

Except for the localization, I also integrate the other autonomy software parts into the real RC platform, especially, into the Navidia Jestson. Fig. 2 shows the pipeline of the software and hardware integration.

The follower localization algorithm provides the current pose and velocity of the RC car. The perception of the leader's Aruco marker provides the relative pose, which can be used to recover the leader's path combined with the follower's localization. The leader velocity estimation provides the velocity of the leader Zamboni which will be used to calculate the target velocity in controller. We solved the problems of Realsense communication with the Jetson, and run all the algorithms developed for simulation in real RC car. And it turns out such algorithm flow works smoothly in real car.

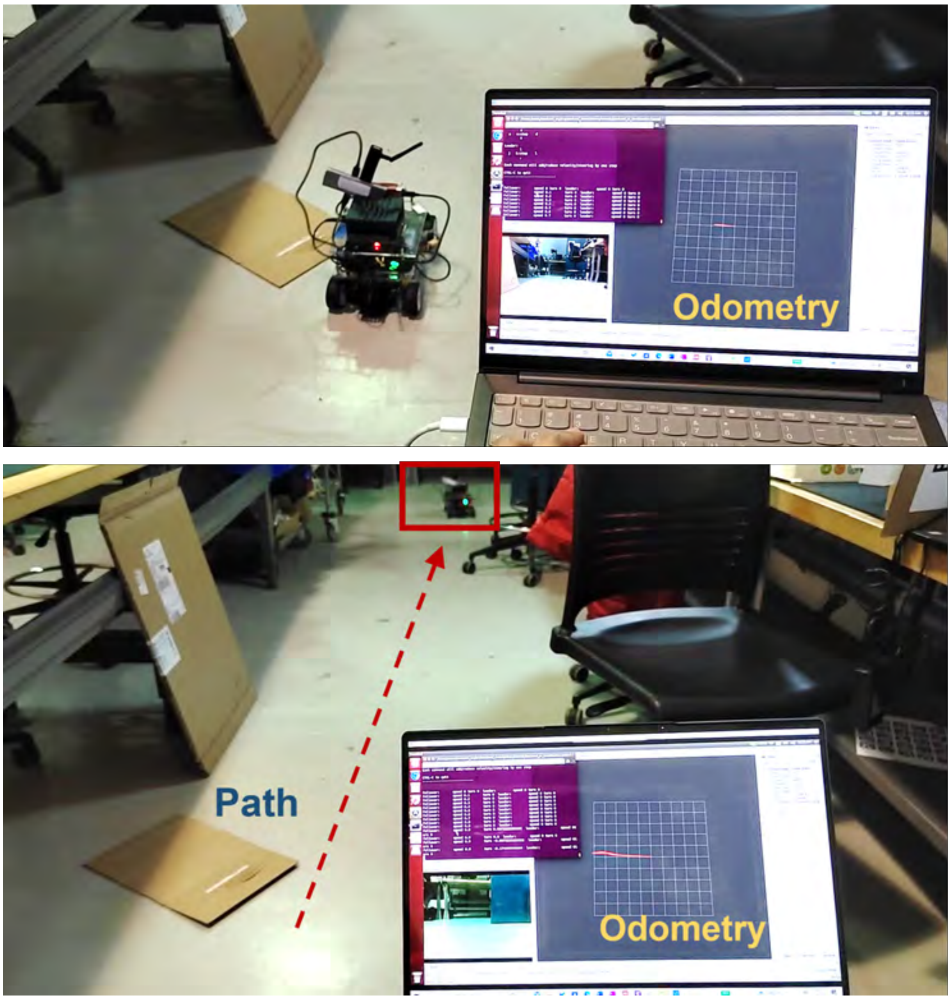

Figure 1: Remote teleportation and localization through ROS.

2 Challenges

The first major challenge lies in the mechanical parts of the RC car hardware. Due to the mechanical clearance, the steering angle always has some variance given a command. In addition, since we put the Navidia Jetson, camera, two battery on the car, it becomes very heavy. There was situation that the car did not move on the ground even I gave it a velocity around 0.5m/s. Then when I lift the car up, the wheels are actually rating, but the car was too heavy to move.

The second challenge is the connection instability of the Realsense D435i. If the Realsense cannot give the stable image and imu input, the entire perception and localization algorithm would not work. The situation is that we always need to unplug and plugin the camera many times before it can be connected successfully. We finally figured out a way to get it run better by plugin again its cable after launching the node to start it.

The third challenge when running the algorithms on RC car platform is the parameter tuning. We set the controller and EKF parameters based on the actual Zamboni vehicle in simulation. However, the RC car has different dimensions and different dynamic properties. And the testing environment is also different because we are using Husky as a leader. So the parameters in the algorithms need further tuning.

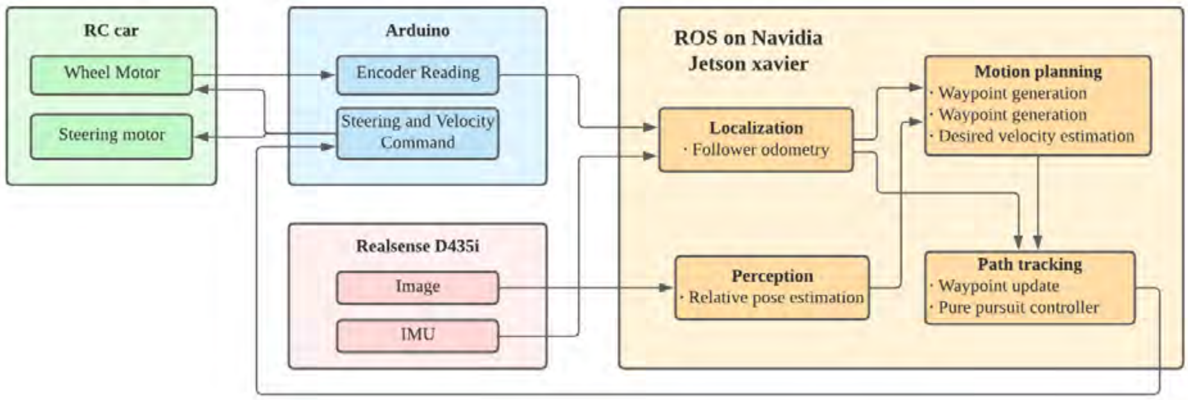

Figure 2: Autonomous convoy of the Zamboni.

3 Teamwork

Kelvin Shen is in charge of perception and recognition. His progress includes:

- Completed two camera setup with three marker board on left, rear, and right side of the follower to avoid losing track of the markers
- Got Husky running
- Integrated pose estimation with waypoint generation with Jiayi
- Soldered PCB with Rathin

Rathin Shah is in charge of controlling the real RC car. His contribution includes:

- Integration of Jetson Xavier and Arduino with Encoder on RC Car
- Validation of localization on RC Car
- Final PCB development with soldering

Nick Carcione is in charge of the RC car hardware. His contribution includes:

- Added PID speed control to the RC car using encoder measurements
- RC car integration and tire change/suspension fixing
- Helped run localization tests and debug ROS-Arduino communication

Yilin Cai is in charge of the simulation and planning. His contribution includes:

- Integrate odometry of RC car using encoder and imu for localization, and tested localization accuracy with Rathin and Nick
- Integrate remote teleoperation and odometry visualization on RC car
- Integrate perception, motion planning and control code on RC car
- Solve realsense communication problem with Kelvin

Jiayi Qiu is in charge of the leader's velocity estimation and wayponits setup. Her contribution includes:

- Integrated two-camera perception subsystem with waypoint generation with Kelvin
- Optimized waypoint generation to maintain the longitudinal offset
- Improved the initialization process for leader follower convoy

4 Plans

The next step includes further testing, including the perception accuracy based on Realsense, Jetson and actual marker board. We will test the detection accuracy in different environment, with different moving speed of the marker. The leader velocity estimation algorithm also need further testing. Since we have the Husky robot which can provide the actual running speed and be used to compare with our estimation results. In addition, the parameter tuning also need further work based on the actual RC car's condition.