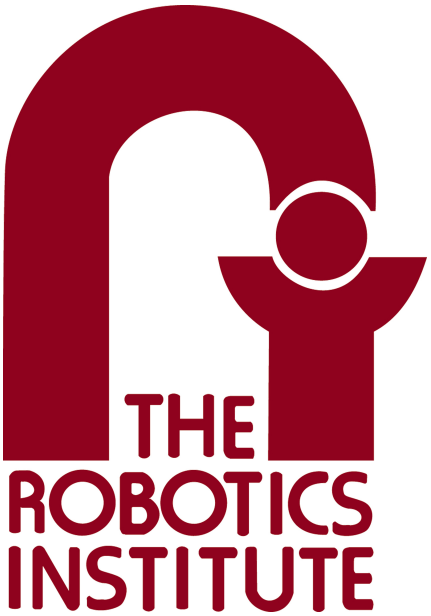

MRSD Project Course

Team I – AIce

Autonomous Zamboni Convoy

Individual Lab Report 6

Team

Rathin Shah

Nick Carcione

Yilin Cai

Jiayi Qiu

Kelvin Shen

Author

Kelvin Shen

Sep 8, 2022

Table of Contents

Individual Progress

Our project is the Autonomous Zamboni Convoy that aims to follow a manually driven Zamboni vehicle by another autonomous one with a fixed lateral offset. Due to the delay in the shipment of the Zamboni vehicle, we have a tight timeline converting it to a drive-by-wire system such that we can interface it with our completed autonomy stack. Given the unavailability of the vehicle as it will undergo a drive-by-wire design and integration by Isuzu, we have to resort to another platform, Yamaha Utility ATV, which is the closest to the Zamboni vehicle in terms of both size and controls. Therefore, we will test and integrate our autonomy stack completed last semester onto the ATV and thoroughly tested its leader-follower functionality while waiting for the Zamboni vehicle to be retrofitted and shipped back from Isuzu. My role is to continue to improve our perception subsystem by solving any remaining issues we saw left in SVD and taking advantage of the sensor suite on the ATV.

Review ATV Software Stack and Integrate our perception subsystem

The ATV comes with a slightly different autonomy stack, as shown in Fig 1. In particular, it uses LiDAR and IMU as input to its odometry while our localization uses wheel encoder and IMU. In addition, its uses model predictive path integral controller as the local trajectory planner to track a path, which only takes in waypoints without velocity, while ours is based on the pure pursuit that takes in both waypoints and velocity.

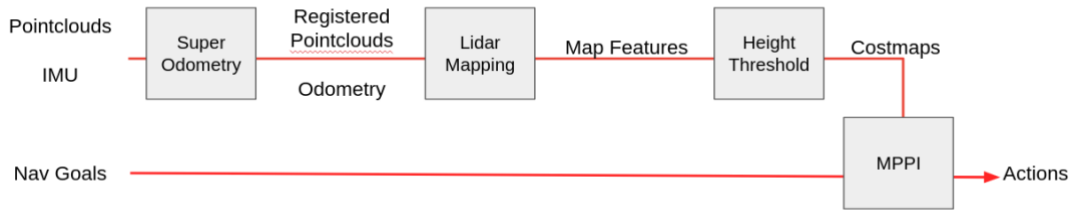

Figure 1. Baseline Autonomy Stack on the Yamaha ATV

Therefore, in order to integrate our original stack into ATV's, we have to:

(1) Replace the MPPI with our pure pursuit controller that takes in both waypoints to follow and the target velocities

(2) Modify our odometry so that it adapts to inputs of point clouds and IMU, while keeping our original wheel odometry as a backup in case where LiDAR fails (the ATV is also equipped with wheel encoders but not in use in Fig 1.)

(3) Replace the Nav Goals input into the planner with the output of our perception subsystem, which involves detecting the leader pose relative to the follower ATV, estimating leader's velocity and generating leader waypoints.

In Fig 2. We show the updated pipeline after integrating our system into the ATV software stack.

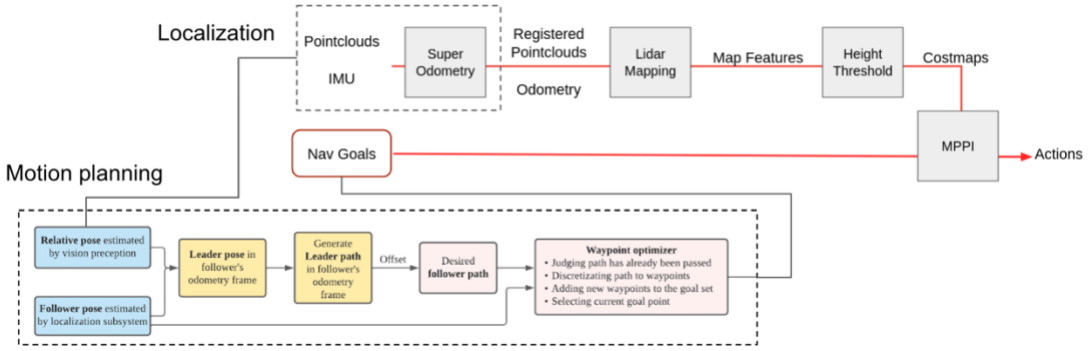

Figure 2. Updated Autonomy Stack on the Yamaha ATV

Refresh memory of SVD codebase and ROS

I also reviewed our codebase after not touching it throughout the summer. In particular, I familiarized myself with the key ROS nodes we have used as well as the new nodes on the ATV that can be used out-of-the-box by us, listed in Table 1.

Table 1. Key ROS nodes used in ATV software stack

Node	Filepath	Inputs	Outputs
Novatel Driver	(desktop-legacy branch) physics_atv_ws/src/novatel_span	None	/odom /odometry/filtered_odom /novatel/imu/data
Velodyne Driver	(baseline branch) physics_atv_ws/src/platform/sensors/velodyne_launch	None	/velodyne_1/... /velodyne_2/...
Super Odometry	(baseline branch) physics_atv_ws/src/slam/deadfast-slam	/novatel/imu/data /velodyne_1/velodyne_points	/integrated_to_init /velodyne_cloud_registered_with_features /tf
Lidar Mapping	(gridmap_rewrite branch) physics_atv_ws/src/perception/physics_atv_lidar_mapping	/velodyne_cloud_registered_with_features /tf	/local_grid_map
Lethal-height Costmapping	(gridmap_rewrite branch) physics_atv_ws/src/perception/physics_atv_lidar_mapping	/local_grid_map	/lethal_height_cost_map
Torch MPC	(baseline branch) physics_atv_ws/src/control/torch_mpc	/lethal_height_cost_map /integrated_to_init /next_waypoints/odom /steering_angle	/mppi/viz /joy_auto /controller/target_input

I also went through the workflow of our previous stack in simulation by launching the perception subsystem, detecting the leader Zamboni, estimating its pose and publishing it to the waypoint generator.

As practice, I also refactored the physics_atv_deep_stereo_vo package on the ATV stack by adding RealSense D435i to one of its supported inputs (it originally only supported inputs from the MultiSense S21 camera equipped on the ATV). This involved adding correct camera intrinsics to the configuration of the package, installing RealSense environment on the ATV laptop and resolving code conflicts due to different number of input channels between RealSense D435i and S2. As such, we would be able to transfer our previous vision sensor onto the ATV without compatibility issues and continue to use our perception subsystem without refactoring it.

Challenges

The major challenges I have encountered when reviewing the new stack are:

- The ATV documentation is primitive with little information if we can't see the ATV personally. For example, there's little explanation regarding in what kind of format the signals between each subsystem are, which makes it difficult for us to estimate how much efforts it would take for us to integrate our previous stack into the new one. In addition, the sensor suite on the ATV is unlisted in the documentation. I got the response from Wenshan that the camera used was a stereo camera called MultiSense S21 that is interfaced using the multisense_ros package. We also scheduled a visit to the facility to look into the ATV with Wenshan this week.
- The ATV software stack is huge and hence takes time to be familiar with. Therefore I have chosen one of the packages as the entry point and refactored it so that it would be compatible with inputs from our vision sensor (RealSense D435i) in addition to the S21 camera it was originally built upon.

Teamwork

- Nick updated the performance requirements according to the new leader-follower system and adapted the specifications from the RC car platform to the real Zamboni platform. He also brainstormed potential drive-by-wire steering design with Rathin.
- Rathin updated the functional requirements according to the new leader-follower system. He studied motor controller for CAN bus communication which will be used in throttler control.
- Jiayi handled the communication and collaboration with Isuzu, who genuinely offered to help us convert the Zamboni vehicle into a drive-by-wire system. She also updated potential drive-by-wire steering and braking designs to Isuzu

so that they could purchase essential components before the Zamboni is shipped.

- Yilin worked on brainstorming the integration of our system onto the ATV together with me, proposing ideas as well as risks that need to be mitigated specifically after using the ATV.

Plans

Before the next progress review, I plan to get familiar with all the packages mentioned in Table 1 so that I can get the baseline autonomy stack running on the ATV. I will also test the perception subsystem on the ATV after installing the RealSense D435i onto it. Finally, I will look into camera-LiDAR fusion methods for 3D object detection to complement cases when ArUco marker detection fails using the camera only.