BFMC 2025 : Status Report 1
Date : Monday, January 20, 2025
Prepared by : Pothole Avoidance
Reporting Period : Tuesday, December 17, 2024 - Monday, January 20, 2025

Overview of Project Status

The previous integration test in December revealed a critical dependency in our approach to autonomous navigation : our model predictive control (MPC) approach is heavily dependent on the calibration of all our sensors, motors and actuators. Further progress could not be achieved without a precise characterization of these. Another identified weakness was the delay caused by using ROS1 as the main communication framework. Many of the previous month's tasks were dedicated to finding solutions to these issues.

Current Phase

Calibration and adjustment of both the steering and speed proved to be more complex than what suggested on the competition's documentation. Having done the full hardware improvement, we attempted to adjust the steering by tuning the zero default and step values. However, the more we understood the code, the less we understood why it was written in this way. We concluded that it had to be rewritten such as to send values with our desired accuracy. Characterization of input duty cycles to output given steering angles was graphed and a curve was fit for a more precise input steering[1].

While calibrating the steering, we noticed different limits of steering than those specified in the competition documentation. This made it such that the constraints for our MPC algorithm were to be adjusted with the actual values. Extensive simulator testing was therefore done to evaluate how these new constraints impacted our model.

Even though ROS1 uses TCP as a backbone for communication, the fact that all messages have to run through the ROS master creates significant delays and loss of messages when attempting to send data to a different host. This was manifest in our previous media file upload. We therefore endeavoured to create our own client and server scripts to leverage all of TCP's advantages, reduce delays and losses.

Task Status - Overview

December was a month where a lot of work was accomplished : the winter holidays meant that the team was not occupied by studies or internship work, and more time could be spent working on the project. This was important as many tasks proved to take more time than what was initially expected. A detailed status of the tasks can be found in the attached task tracker table export[2]. The main tasks will be briefly discussed in the following.

Steering/Speed Calibration : This task was of high priority and is complete. Characterization of the input-output (steering angle - duty cycle) relationship has been done. Much better results have been achieved than those with the original computation.

Communication with car : This task was of high priority and is well advanced. Communication over ROS1 was unstable and prone to significant delays. Data sent to the dashboard was therefore

---

[1] See *true_hardware_calibration_charts.pdf*
[2] See *task_tracker_export_status_2.pdf*

implemented using TCP[3]. A C++ TCP client is created on the Jetson, while a python TCP server is created on the host computer to display the dashboard. The server parses the data upon reception and uses this stream to display what is required on the dashboard.

IMU Data - Realsense : The IMU data from the RealSense was unstable when using it to compute the yaw/heading[4]. Further research into how the yaw is computed on the BNO055 sensor revealed the importance of sensor fusion algorithms ("the secret sauce to stable and reliable values"). The absence of a magnetometer on our version of the RealSense means that no reliable yaw can be extracted. We will therefore resort to using only the BNO055 sensor.

Blocking Points and Encountered Issues

Mechanical constraints : Steering calibration revealed how the hardware constraints limit the degree to which a precise steering angle can be considered. When doing a full calibration and attempting to set the duty cycle for an angle of 0∘, we ran into the following problem : the duty cycle for 0∘ depends on whether the previous turn was a right turn or a left turn. We sourced the problem to be a result of the play in the wheels and between all the mechanical joints.

Steering angle restrictions : Our characterization of the steering lead us to identify the maximum steering angles to be between -21∘ and 21∘. A new trajectory had to be computed for these tight turns since we could no longer make the turn the same easy.

Power distribution board fault : Like many teams in the competition, we encountered a fault with our power distribution board. In one of our worksessions in mid-December, the powerboard stopped outputting 5V power. It still supplies 8V power to the motor, but all current is blocked at the buck inverter.[5]

Next Steps

The integration test revealed some noticeable flaws and weaknesses in our approach to the problem. The first of these is the sustained problem of the launch of the camera in our camera node and its inconsistent frames. While this problem was thought to have been solved in a previous task[6], it seems our evaluation of it was erroneous and this issue will be reinvestigated. The MPC algorithm outputting some sharp steering angles at the beginning of the trajectory was also troublesome as it threw off the car from the desired trajectory. To address this, the STM32 microcontroller will use the steering angle to compute a desired yaw using the kinematic bicycle model. This desired yaw will then be used as a reference angle to be matched with a PID controller. This should give an overall smoother adjustment of the steering.[7] Furthermore, while the camera feed was implemented over TCP, some dashboard functionalities still functioned over ROS, making them slow. Implementing these over TCP is planned in the coming weeks. The powerboard failure is something we would like to be protected against, so design of a custom powerboard for our needs is underway, to ensure our independence in terms of manufacturing and to better respond to our specific needs. Finally, some signs were not being recognized or acknowledged due to either the camera position or the lack of model training. Addressing these issues, a new camera mount closer to the higher admissible limit of 30 centimeters is being designed and the image recognition model will be retrained on a more extensive dataset.

---

[3] See *communication_protocol.pdf*
[4] See *imu_overview_presentation.pdf*
[5] See *powerboard_problem.pdf*
[6] See *task_tracker_export_status_2.pdf*
[7] See STM32*_steering_control.pdf*