Motion Estimation

Creating a Motion Estimation module for the EPFL Racing Team racecar

DS Master
Optional Semester Project
Alexandre Dherse
22/01/2021

Professor:
Alexandre Alahi

Contents

1 Introduction

This project is done as part of the effort from the EPFL Racing Team to make their racecar driverless, in order to participate to the Formula Student competitions in driverless events. While the Racing Team will not participate at the 2021 contests, it aims to obtain the best possible results at the 2022 edition. It will participate in all the driverless events[1]:

- Static events, judging business plans, manufacturing and engineering design
- Dynamics event, judging the performance of the driverless racecar.
    - Skidpad event: a figure eight pattern, judging the cornering ability of the car. The dimensions and cone placement of this event are known in advance.
    - Acceleration event: a straight line with a length of 75m, judging the acceleration ability.
    - Autocross and Efficiency events: The events that caused the most trouble for previous year teams. In these events, the track layout is **not** known in advance. The autonomous vehicle should do a recognition lap in which it maps the track layout before trying to complete laps as fast as possible.

In order to succeed in those events, the development of the car's software is separated into four different modules: Perception, Motion Estimation/Mapping, Path Planning and Control. The Motion Estimation challenge is to compute a reliable and accurate estimate of the vehicle's state - its position relative to the beginning of the track, its speed and acceleration, its heading and yaw rate.

This information will be used by two other sub-parts of the architecture:

- The Simultaneous Localization and Mapping (SLAM) algorithm. Using data about the position of the cones coming from the Perception team and the state of the vehicle, this algorithm will construct the map of the track layout in the autocross events. Inaccurate estimation of the vehicle's state may lead to an inaccurate map and the inability to be competitive in the later laps.
- The Control module, in order to estimate the acceleration, braking and steering angle of the racecar necessary to navigate through the course. Inaccurate data from the estimation module may lead to actions producing unexpected results. For example, an underestimated velocity could produce understeer when turning, leading to a missed corner.

Figure 1.1 presents those modules and the interactions between them at the end of this first semester.

[1] Formula student rules, accessible at https://www.formulastudent.de/fileadmin/user_upload/all/2020/rules/FS-Rules_2020_V1.0.pdf

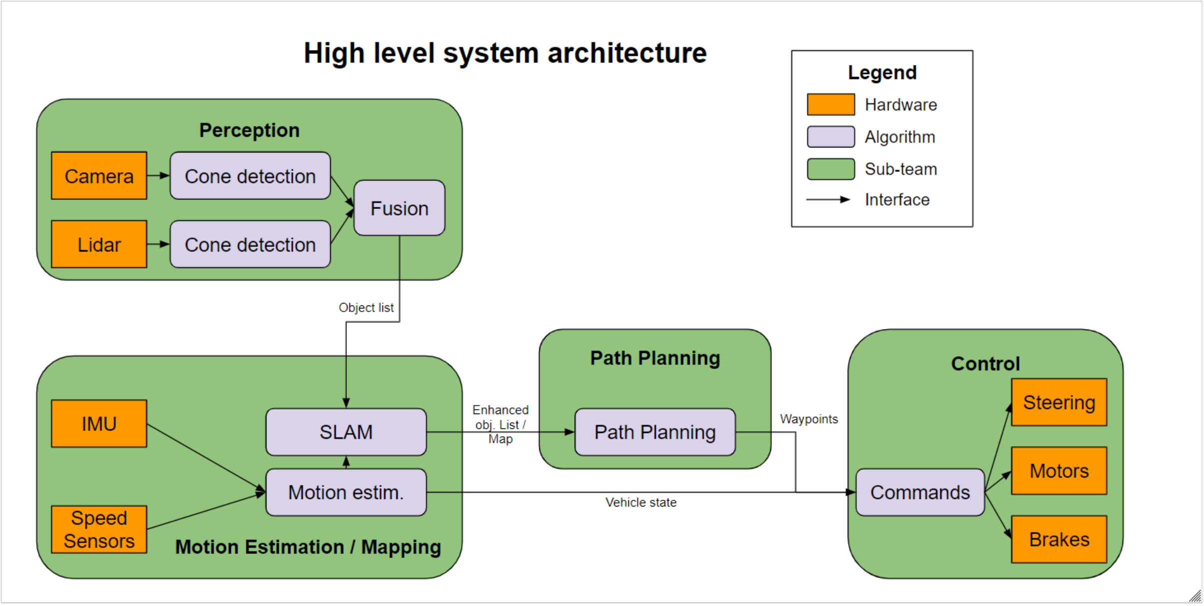

Figure 1.1: Racing team driverless architecture.

2 Motion Estimation

For this first semester, my focus was to build a system that could work in situations where the car is not driven at the limits, but could be modified in order to account for it. As we will see during the next sections, spirited driving leads to more complexity in how the car behaves, leading to more sophisticated algorithms. Since my commitment to the EPFL Racing Team spans over an academic year, the next semester will serve as my Master's Semester Project and will be dedicated to handling this scenario. This semester's work will be used for the first lap of the autocross event, since the mapping of the track is done at slow speeds.

2.1 Hardware

Since the EPFL RT track car is in development since 2017, multiple sensors and hardware pieces are already installed on the car, and can be used to estimate the state of the car. They were used during previous years for telemetry purposes, and can now be repurposed for allowing driverless operations.

2.1.1 Inertial Navigation System

The most useful sensor for our purpose is an Inertial Navigation System (INS), giving us information about every variable in the state of the car, and crucial for estimating position and heading. It is composed of an Inertial Measurement Unit (IMU), itself composed of gyroscopes, accelerometers and magnetometers, as well as a Global Navigation Satellite System (GNSS) receiver and an RTK receiver.

The particular INS we're going to use is the Ellipse-N from SBG systems, suited for automotive applications. It provides us heading with an accuracy of 0.2°, and position with an accuracy of 1.2m, using GPS single point.[2] When using RTK GPS, the position accuracy greatly

[2] Ellipse series leaflet, accessible at https://www.sbg-systems.com/wp-content/uploads/Ellipse_Series_Leaflet.pdf

improves to 1cm. Unfortunately, this technology requires the installation and use of a base station, which is not possible during the FS events. The rules state: "(D)GPS may be used, but there will be no space to securely build up base stations on the competition site". We will thus only use RTK GPS to obtain the ground truth during our tests, and use a standard GPS to estimate our position during competitions.

Finally, it should be noted that the accelerometers are susceptible to drift, leading to inaccurate measures over time. This will be taken into account next semester when dealing with multiple laps, and the loop closure of the course will help us re-center the IMU.

2.1.2 Wheel Speed Sensors

We also have at disposition Wheel Speed Sensors on the front wheels, more specifically Texens RS-M10WS. They measure the frequency at which the wheels turn, and are more precise than the IMU velocity measures.

However, these sensors cannot deal with tire slip. Tire slip occurs when driving at the limits of the racecar's grip. A poorly dosed braking might induce tire lock-up, and cornering at great speeds will induce under or oversteer. Without a proper method of detecting tire slip, they are thus only useful during the first lap of the Autocross events.

2.1.3 Processing unit

Lastly, the car is equipped with an embedded controller, the sbRIO-9627. It contains a configurable FPGA capable of reading and sending CAN messages, as well as receiving analog signals. This data is accessible by a CPU running NI Linux Real-Time. This processing unit was retrieved from the EPFLoop project, and kept as "its computational power is able to perform image processing", and "will allow, in further year, to develop DV car"[3]. All the computations to obtain the vehicle's state will thus be done on this sbRIO, there is no need for additional processing hardware.

2.2 Software

2.2.1 Kalman Filter

Our method should be able to fuse data from those different sensors to produce an accurate estimate of the racecar's state, even in case of sensor failure. It needs to be able to deal with measurements coming from sensors with different frequencies, and allow us to add other inputs later in the year. For example, visual odometry will likely be added next semester to improve results, and the cameras run at a frequency of about 30Hz, while the IMU output at 200Hz. The Extended Kalman Filter is thus a great fit. It is a popular method of combining GPS and Inertial data [1] [2] and [3] indicates that "according to [multiple research papers], EKF is the most appropriate technique to be adopted for inertial and visual fusion".

The Kalman Filter is a 2-step algorithm: a time update, where the filter estimates the state ahead using a predetermined model, and a measurement update, where the filter corrects that estimate using data from the sensors. The measurement update can be run with a fraction of the measurement sensors, this algorithm can thus run asynchronously.

[3] Simon Maksay and Aurélien Kinet's report on Data Acquisition and Motion Control for EPFL Racing Team, 2020

We need to estimate the following state:

$$\vec{x}(t) = \left(x \;\; y \;\; \theta \;\; v \; a \; \dot{\theta}\right)^T \tag{1}$$

with the following equations:

$$\vec{x}(t+T) = f\left(\vec{x}(t)\right) + \vec{w}(t) \tag{2}$$
$$\vec{z}(t) = h\left(\vec{x}(t)\right) + \vec{v}(t) \tag{3}$$

f(●) is the estimation function, $\vec{z}(t)$ is the observation vector and h(●) is the observation function. $\vec{w}(t)$ and $\vec{v}(t)$ are the process and observation noise vectors. For following equations, Q_t and R_t are process and observation noise covariance matrix, P_t is the posterior covariance of the estimate, K_t is the Kalman gain and J_f(●) is the jacobian of f. I is the identity matrix.

The time update equations are the following:

$$\vec{x}(t+T) = f\left(\vec{x}(t)\right) \tag{4}$$

$$P_{t+T} = J_f\left(\vec{x}(t)\right) * P_t * J_f^T\left(\vec{x}(t)\right) + Q_t \tag{5}$$

And the measurement update equations the following:

$$\vec{x}(t) = \vec{x}(t) + K_t(\vec{z}(t) - h(\vec{x}(t)) \tag{6}$$

$$K_t = P(t) * J_h^T\left(\vec{x}(t)\right) * (J_h\left(\vec{x}(t)\right) * P(t) * J_h^T\left(\vec{x}(t)\right) + R_t)^{-1} \tag{7}$$

$$P_t = \left(I - K_t * J_h\left(\vec{x}(t)\right)\right) * P_t \tag{8}$$

2.2.2 Estimation Model

The motion model used to estimate the state of the car in the next iteration is important. If a sensor updates at a lower frequency than others, or malfunctions, the filter will output this estimate. In the case of the INS, the GPS data is less frequent than the IMU data, so the model needs to predict accurately the position (x,y) at every iteration even though most of the measurement updates will not contain GPS data. According to [4], the best tracking performance is obtained by using Constant Turn Rate and Acceleration (CTRA) or Constant Curvature and Acceleration (CCA). The CCA model requires a much higher computation cost so the CTRA model will be used in our project. The equations are as follows:

$$f\left(\overrightarrow{x(t)}\right) = \overrightarrow{x(t)} + \left(\Delta x(T) \;\; \Delta x(T) \;\; \dot{\theta}T \;\; aT \;\; 0 \;\; 0\right)^T \tag{9}$$

with:

$$\Delta x(T) = \frac{1}{\omega^2}[(v\omega + a\omega T)sin(\theta + \omega T) + a * cos(\theta + \omega T) - v\omega * sin\theta - a * cos\theta] \tag{10}$$

and:

$$\Delta y(T) = \frac{1}{\omega^2}[(-v\omega - a\omega T)cos(\theta + \omega T) + a * sin(\theta + \omega T) + v\omega * cos\theta - a * sin\theta] \tag{11}$$

It should be noted that this is a kinematic model, as the dynamics of the racecar are not taken into account. While top teams as AMZ have a dynamic estimation model, taking into account steering dynamics for instance, other improvements to the filter presented in section 4 can lead to bigger improvements. A dynamic model will thus only be implemented next semester if time permits.

2.2.3 Sensor failure detection

Lastly, the extended Kalman filter is not resistant to outliers. If a sensor outputs erroneous data for a few iterations, the predicted state will also be erroneous. A way of dealing with sensor errors is necessary to ensure a coherent output from the filter. For now, a simple rejection threshold is set dynamically, according to the current state of the car, and impossible measurements are rejected. As noted by [5], this method is not ideal, and our filter could benefit from a more sophisticated approach. I was however not able to get any information from the students in charge of the sensors regarding frequency or examples or such failures. Producing a real-life dataset of measures from the INS and other sensors on the car would greatly help us determining if a more elaborate method could improve detection.

3 Results

Since no dataset was created with the EPFL RT racecar and its hardware yet, I had to find an appropriate dataset. I was unfortunately not able to find a dataset with the combination of sensors currently or soon-to-be on the car (see section 4). I used a dataset containing IMU and GPS data, from a loop of about 2 kilometers in city driving, as well as plotting tools[4]. I mistakenly believed this dataset contained an accurate localization of the vehicle in addition to these sensors, but it only contained inaccurate satellite-based GPS data, which invalidated the results I had obtained. I have requested access to the Oxford Robocar Dataset[5] RTK ground truth to conduct meaningful experiments as soon as possible.

Even though no relevant results can be extracted from these tests, we can still see in figures 3.1 and 3.2 that the filter followed the course correctly and made accurate estimations for the course and velocity.

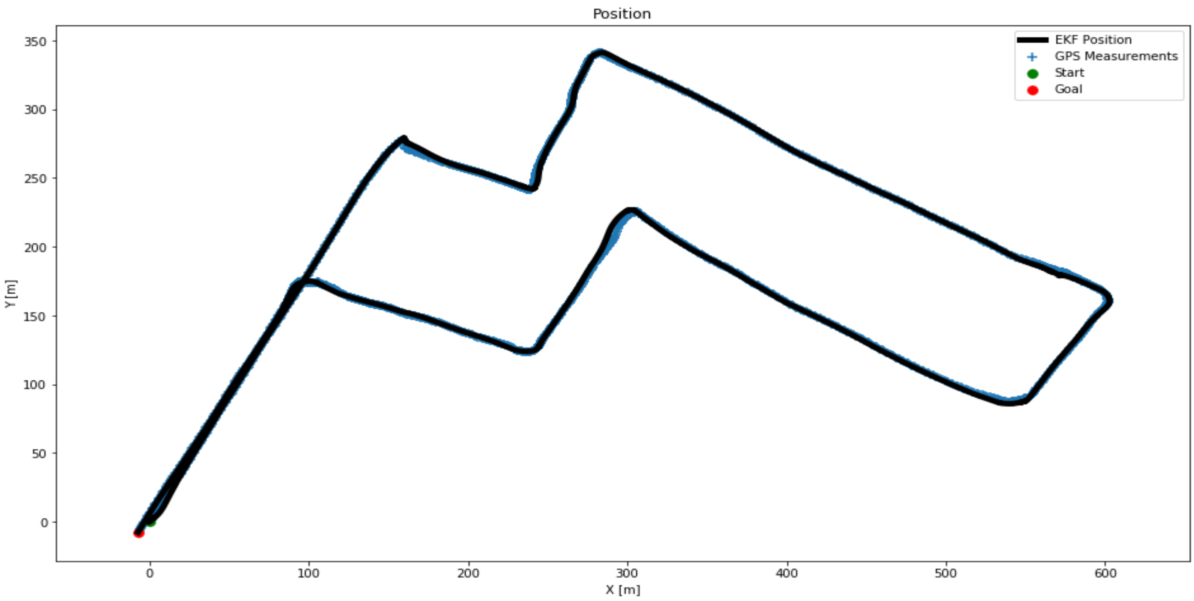

Figure 3.1*: Estimation of the car's position*

[4] https://github.com/balzer82/Kalman
[5] https://robotcar-dataset.robots.ox.ac.uk/

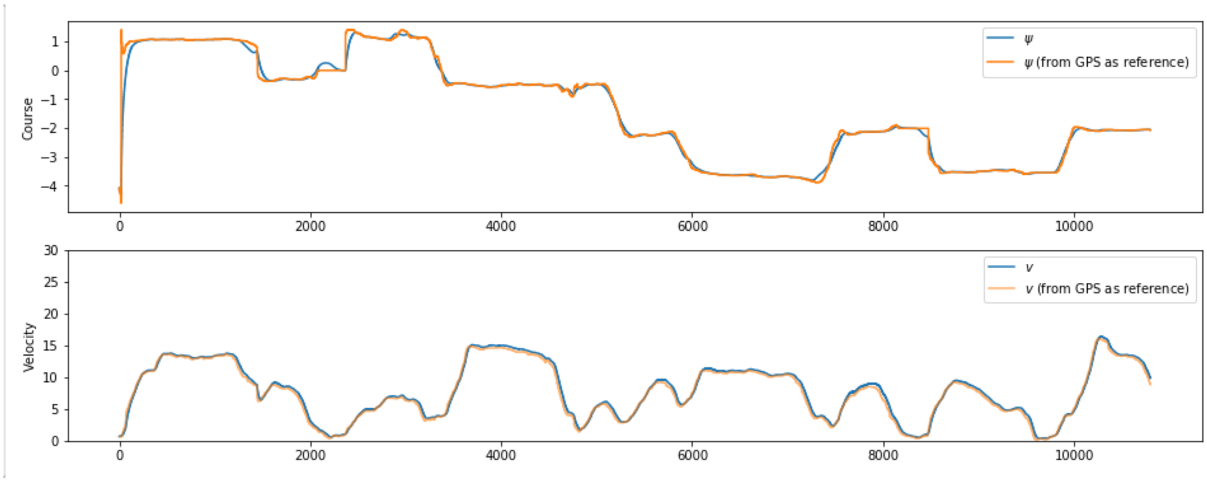

Figure 3.2: *Estimation of the car's heading and velocity*

Thanks to the help and time of Carlos Conejo, I was also able to try the filter on the VITA lab robots. Unfortunately, this did not help me test the performance of my filter, as the robots are only using one INS and using a Kalman Filter. There was no data fusion to perform, and the data was already processed. This still confirmed that the filter worked great in real-time, and was outputting a path very similar to the actual path of the robot.

4 Research for next semester

The priority for next semester is to properly test the method proposed this semester. On top of the Oxford dataset, the simulation of the car and its sensors should be operational soon, as we started merging the codes of the driverless students. If sanitary restrictions permit, we may also be able to create our own real-life dataset.

In order to achieve good results at Formula Student events, the filter should also be able to work when driving at the limits of the car, or when the surface of the track is wet, inducing tire slip. It should also locate the car within the track more precisely, as the position relative to the width of the track is going to be crucial for path planning and control. Two main improvement are thus in order:

- Slip detection. While the lateral slip can be detected [1] or computed using the sensors on our racecar [6] [7], having an accurate slip angle detector would improve accuracy and reduce software complexity. An optical sensor using LED illumination mounted on the car could serve this purpose, and would allow us to get a more precise velocity measurement, even in case of tire slip.

  After checking the FS rules regarding the mounting and utilization of such a sensor (DV 4) and confirming nothing prevents it, my research led me to one specific model: the Kistler Correvit SFII. It is suited for racing applications, only weighs about 200g, can measure slip angle and velocity accurately from 0.3 to 250km/h[6]. Kistler is also a brand interested by investing in student research as they are sponsors of a few FS teams, such as the TUfast Racing Team (TU of Munich) or AMZ (ETH Zurich). I reached out to them to initiate a partnership and they responded positively,

[6] Correvit SFII datasheet, accessible at https://www.kistler.com/files/document/000-812e.pdf?callee=frontend

proposing a Correvit SF II in exchange for visibility. While the deal is not finalized yet, the conversation will continue between the end of January/February.

- Using data from other modules for added precision. First is the map of the racetrack generated by the SLAM module. Under normal circumstances, the car should not leave the mapped racetrack, and we can thus project our unconstrained estimation onto the feasible space of positions, between the cones, as presented in [5] [8] [9]. We can then combine this information with data from the perception module, giving us information about the relative cone positions. As stated before, [3] indicates that the EFK is a great method of combining IMU and perception data. We can thus adjust our position on the width of the racetrack to obtain more accuracy. According to our contact from AMZ, this is what gave them the greatest improvement over the simpler designs. Visual odometry could also help us remove the dependency over GPS utilization.

5 Conclusion

The Extended Kalman filter was not easy to apprehend at first, and very tedious to debug as there is often no clear indication of where the error comes from. Despite the lack of testing of my implementation, the Extended Kalman Filter has been a proven method of combining IMU, GPS and velocity data in past research, and allows us to improve it next semester by fusing data from additional sensors. The opportunity to work with state-of-the-art hardware such as Kistler or SBG-Systems sensors to build a concrete project is very exciting and I’m eager to continue improving my motion estimation module.

Bibliography

[1] C. Jonsson, «Velocity estimation in land vehicle applications,» Master's Thesis in Systems Engineering, 2016.

[2] S. Rezaei et R. Sengupta, «Kalman Filter-Based Integration of DGPS and Vehicle Sensors for Localization,» *IEEE Transactions on Control Systems Technology,* 2007.

[3] M. B. Alatise et G. P. Hancke, «Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter,» *Sensors,* 2017.

[4] S. Robin, R. Eric et W. Gerd, «Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking,» *IEEE,* 2008.

[5] Z. Yueming, «GPS/IMU Integrated System for Land Vehicle Navigation based on MEMS,» *Licentiate thesis in Geodesy ,* 2011.

[6] V. T. Minh, «VEHICLE STEERING DYNAMIC CALCULATION AND SIMULATION,» *DAAAM International 23rd Symp,* 2012.

[7] T. L. Anh, «Application of Extended Kalman Filtering for Estimating Immeasurable Vehicle State Variables,» *Journal of Advanced Engineering and Computation,* 2017.

[8] S. D., «Kalman filtering with state constraints: a survey of linear and nonlinear algorithms,» *IET Control Theory and Applications,* 2009.

[9] D. Simon et T. L. Chia, «Kalman filtering with state equality constraints,» *IEEE Transactions on Aerospace and Electronic Systems,* 2002.