Development of model predictive motion planning and control for autonomous vehicles

Guido Tosolin, Senior Manager, Chassis Development
Jaume Cartró, Project Engineer
Vishwas Sharma, Project Engineer

1 Abstract

Full autonomous vehicles for the general public are getting closer every year. Among all the challenges to overcome, one of them is the acceptance of this technology which translates to make the passengers enjoy being driven. To achieve this objective, automated vehicles will have to focus on performance attributes such as comfort, stability or efficiency and vehicles dynamics development will take care of it.

At Idiada, research about this topic is being carried out and strategies about motion planning and control (path follower) will be proposed in this paper. The development is based on optimal control method like Model Predictive Control (MPC).

Among all the possibilities to face the problem, MPC was chosen for several reasons. MPC allows setting constraints in our control inputs like maximum steering wheel angle or vehicle states like accelerations. However, the main reason to use MPC is its way of planning in advance control actions that behaves very similar to how a human driver would do. This feature is key in our understanding to make autonomous vehicles be accepted by all passengers.

Also, our contribution yields in finding the correct vehicle dynamics metrics to design and adjust all the cost functions.

Finally, thanks to our new acquisition, the DIM 250 VI-Grade Simulator which is able to reproduce up to 2.5G accelerations, all our development will be evaluated in a fast and secure testing environment.

2 Automated driving overview

Figure 1 provides an overview of how automated driving can be structured:

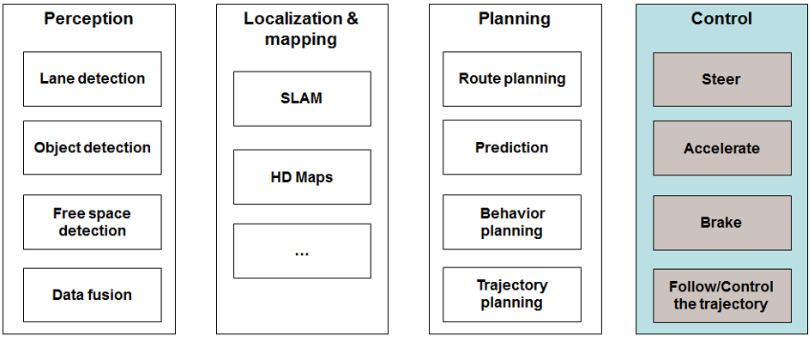

Figure 1: Automated driving blocks

The first layer of perception will get all data available from the sensors to capture the environment around the vehicle. Then a second layer will locate the vehicle and build the surroundings with data of the first layer. Then planning algorithms will find a trajectory to follow and finally the control of the vehicle is managed to get the desired vehicle motion.

3 Development

In this paper a high-level controller will be presented. The outputs of this module are the steering wheel angle and pedals reference positions to be followed by other low-level controllers and as a result, the controller is able to follow the path described by planning accurately.

On the planning side, this block can be divided into several layers very related to the frequency of operation as described in [1]. The first layer is the route planning which ideally runs once per trip. Then a local motion planner decides which maneuver to do considering all obstacles, dynamic and static. This last layer normally considers a horizon based on what the sensors can see which can be of hundreds of meters and with very combinatorial solutions which makes it very hard to use with a realistic dynamic model. Then the most common strategies such as A* search algorithms [1], or RRT algorithms [2], use a very simplified vehicle dynamic model or directly a kinematic model.

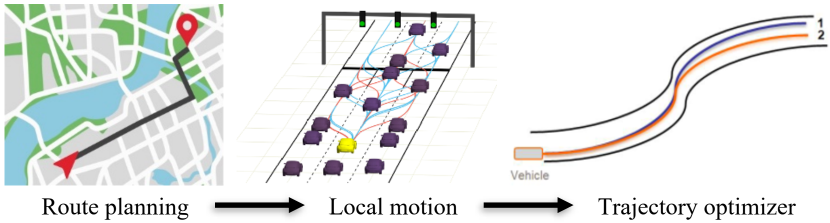

Figure 2: Planning layers workflow

That is why this research work proposes to add a new layer that optimizes the final trajectory using a more detailed dynamic model. In the end, this is going to be the trajectory that passengers would feel and to keep them in good shape and conscious.

Then the contribution of this paper lies on the dynamic behavior that can be referred as a dynamic controller which includes the dynamic planner, that adapts the trajectory references for the controller, and the controller that follows the optimized path.

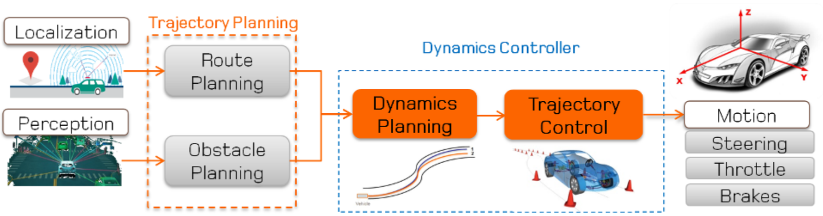

Figure 3: Dynamics controller coupling with the rest of layers

Then, two different approaches for trajectory optimization are discussed in the paper. On one hand a planner which works off-line and uses a nonlinear model to iterate till final solution and, on the other hand, one that works online with linear model constraints.

3.1 MPC Controller

The controller receives the position of the discretized trajectory to follow and predict the control inputs. The number of prediction points chosen is in the range of 15 to 30 and are separated depending on the sample time and speed at each point. The sample time is fixed and around tens of milliseconds so in the end, the total distance predicted will vary depending on the speed reference profile.

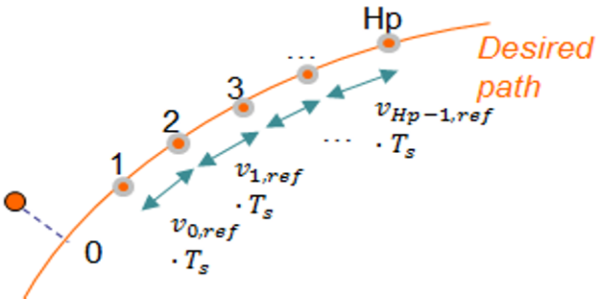

Figure 4: Controller references from planning

The controller uses a 3DOF bicycle model (Figure 5), which is very efficient computationally and at the same time represents all the essential dynamics of the vehicle. For lateral tire forces, the linear region is considered which will be suitable for most scenarios. Only in some emergency maneuvers, the vehicle could drive at its limits and then there are some solutions to consider this effect that are effective (explained later on).

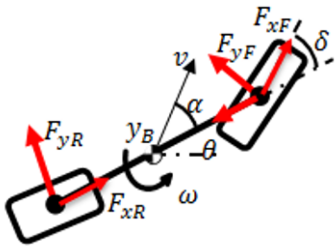

Figure 5: 3DOF bicycle model

Before showing the MPC problem, say that quadratic programming solvers are used to find the optimal solution. This means that the cost function is restricted to be quadratic and the restrictions must be linear. By using these kinds of solvers instead of working with the nonlinear vehicle model, a linear vehicle model is predicted for each prediction and thus the space of solution that the solver works with becomes convex. Then the solver converges to a unique solution very close to the one that a nonlinear solver would found but in a much faster rate and enables us to work in real time. This approximation is correct when working in high frequencies of sampling like the controller is doing.

The cost function used in the controller looks like (Equation 1):

$$\boldsymbol{min}\ (\boldsymbol{\omega_e \cdot Je + \omega_u \cdot Ju + \omega_c \cdot Jc}) \qquad \text{(Equation 1)}$$

– Je: error in trajectory tracking (lateral and angular deviation)
– Ju: control effort → smooth actions (steering wheel rate and longitudinal force rate)
– Jc: stability/comfort parameters (lateral/longitudinal accelerations/jerks)
– ω_i : associated weight

Notice that MPC works as a filter so to really follow correctly the reference the controller emphasizes error tracking. For what concerns restrictions, they are provided by the bicycle model, one for each prediction, and control actions operational limits.

So far, these are the typical constraints and cost function. However, more features have been added to have a more stable and comfortable controller.

For example, not only accelerations are minimized but also a limit acceleration ellipse is set, Figure 6, to always try to be inside of this behavior and allowing to be out of it (for example in an emergency maneuver) but penalizing more in the cost function. It must be remembered though, that constraints must be linear so the implementation is done by multiple straights.

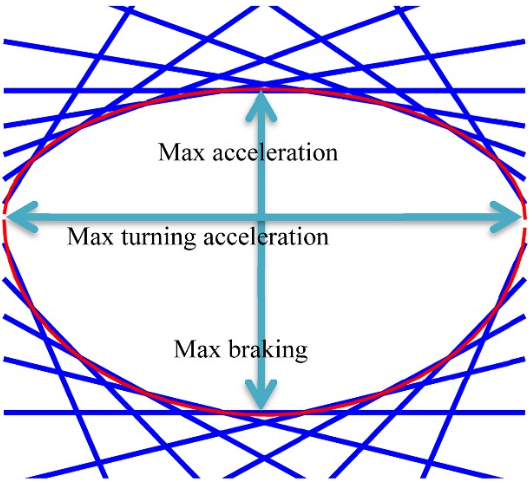

Figure 6: Acceleration ellipse limit

Also, to face the problem of tire limits commented before, the following strategy is adopted.

For understeering behavior, looking at Figure 7, for front tire lateral force the limit is reached for different vertical loads at a specific front sideslip angle. Then a new constraint is added (Equation 2), with a new variable, ε_1, which is added in the cost function with a high weight. Then this variable will be zero with no penalizing effect in the cost function unless the front sideslip angle is overpassed and then solver will try to minimize front sideslip.

$$-Sideslip_{front\ limit} - \varepsilon_1 \leq Sideslip_{front} \leq\ Sideslip_{front\ limit} + \varepsilon_1$$

$$\varepsilon_1 \geq 0 \qquad \text{(Equation 2)}$$

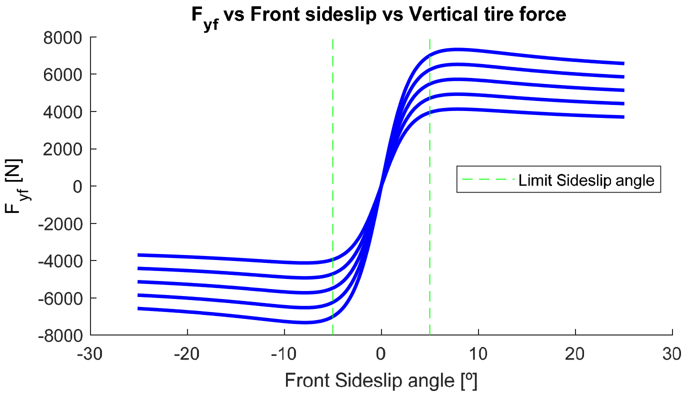

Figure 7: Pacejka tire model for lateral force

Finally, for oversteering behavior, the rear sideslip angle is penalized and thus adding a stabilizing effect in the whole driving experience.

To implement the controller Simulink was used, scheme in Figure 8, as it is a good testing environment to integrate simulation software like CarMaker or CarRealTime to have a realistic vehicle behavior. Then inputs are: in yellow the position and velocity references plus current vehicles states. Outputs are vehicle control inputs.

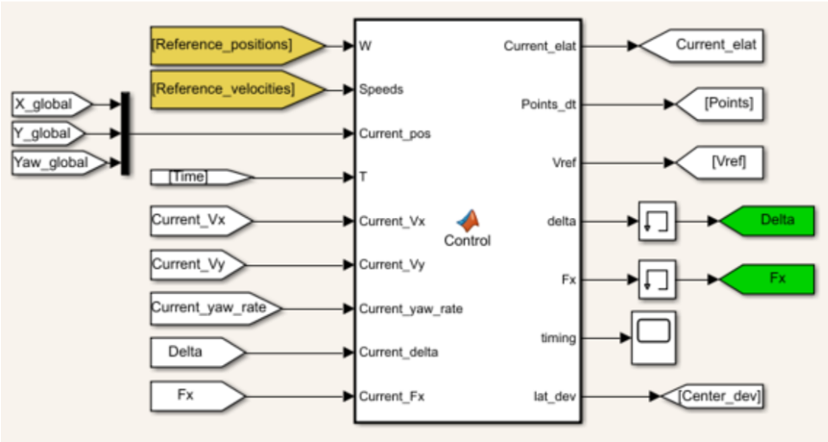

Figure 8: Simulink implementation

3.2 MPC Off-line Planner optimizer

This first approach for planning solves an optimization problem like the controller explained in section 3.1 but with some changes. As an input, it gets a trajectory and optimizes it making more feasible for driving taking into account dynamic metrics.

The first change is that the horizon of prediction considered is bigger than the controller prediction. The horizon selected can vary depending on the purpose and can cover as much distance as desired taking into account that the solution time will increase.

Then the cost function no longer penalizes so much the deviation errors but it focuses on other metrics:

$$\min \ (\omega\_c \cdot Jcomf + \omega\_s \cdot Jstab + \omega\_t \cdot Jtime) \qquad \text{(Equation 3)}$$

- Jcomf: comfort parameters (vertical acceleration, lateral and longitudinal jerk)
- Jstab: stability parameters (sideslip and sideslip rate)
- Jtime: time minimization
- ω_i : associated weight

In this case, the planner uses the same 3DOF of the controller for the handling dynamics plus a quarter car suspension model for the vertical dynamics. The planner no longer linearizes the model for each prediction and uses the nonlinear model directly.

The workflow of the planner is detailed as follows. The input trajectory is discretized in points which the solver is able to move in the normal direction of the trajectory, always respecting boundary limits. Then, in one iteration, it moves all points and runs a simulation where the vehicle follows the trajectory (a controller is also incorporated in the algorithm in order to track the optimized trajectory and provide accurate states). When the iteration is finished using the metrics commented before it defines the trajectory points for the next iteration and also the speed to follow.

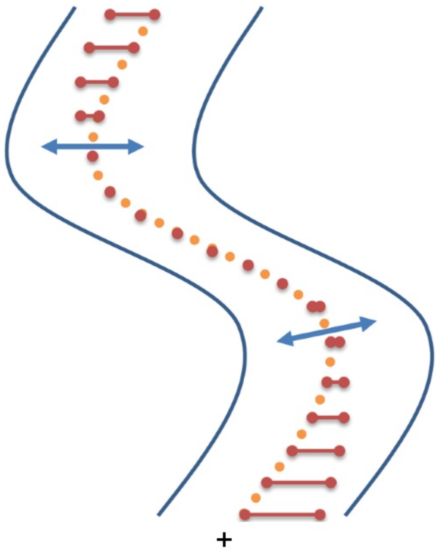

Figure 9: Orange: input trajectory; Red: optimized trajectory

The planner optimizer is following a sectional planning approach to decrease the solving time. However, due to its nonlinear nature is not able to run in real time. Nevertheless, it serves a relevant purpose, because it can be used to validate online optimizers that assume more simplifications.

3.3 MPC On-line Planner optimizer

The planner of section 3.2 provides good results with the only problem of real-time feasibility. That is why Idiada started developing a planner optimizer capable of working in real time.

In collaboration with the 'Institut de Robòtica Industrial (IRI)' of Polytechnic University of Catalonia (UPC), an online planner is being developed. The idea of this planner is very similar to the one of the controller: use MPC structure giving linear vehicle model for each prediction. Again, as the objective is to change the trajectory, the cost function will not focus on tracking errors but on comfort and efficiency metrics. The most difficult aspect is that, unlike the controller model, that considers a relatively short horizon and for which it is easy to get future states, the planner optimizer considers a longer horizon and the future states are changing at each sample time since the trajectory optimized is being updated continuously. The best way to deal with this problem is to use the predicted states in the previous trajectory found by MPC solver to get future vehicle models. This implementation is possible due to the rate of sampling used to update the trajectory which is about hundreds of milliseconds.

The way of computing the linear vehicle model was changed [3]. Typically for linearizing the vehicle model and in order to get a State Space model, Taylor derivative methods are used. On the other side, now a novel technique is used which no longer uses derivate but instead gets a Linear Parameter Varying (LPV) model that translates to reduce the calculations and help with the real-time feasibility.

4 Results

The following section contains the optimization results of the controller and offline planner.

4.1 MPC Controller

For evaluation of the controller, two scenarios are presented. In the first scenario, the trajectory selected is the one in Figure 10. The scenario created emulates an urban scenario where the speed limit is at 40 km/h and in Figure 11 it can be seen how the vehicle is able to follow the reference trajectory.

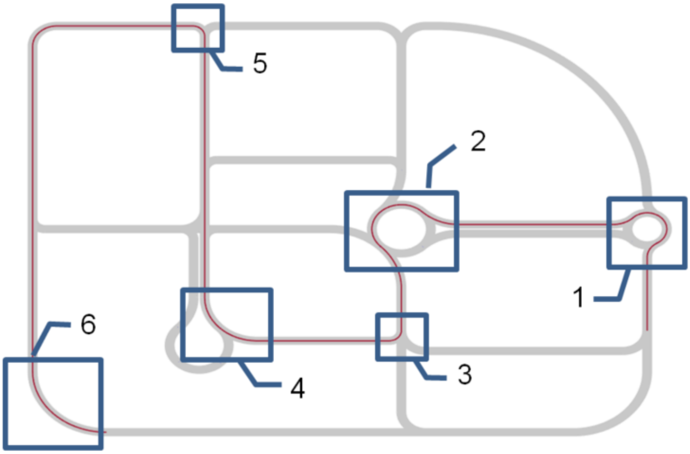

Figure 10: Urban track

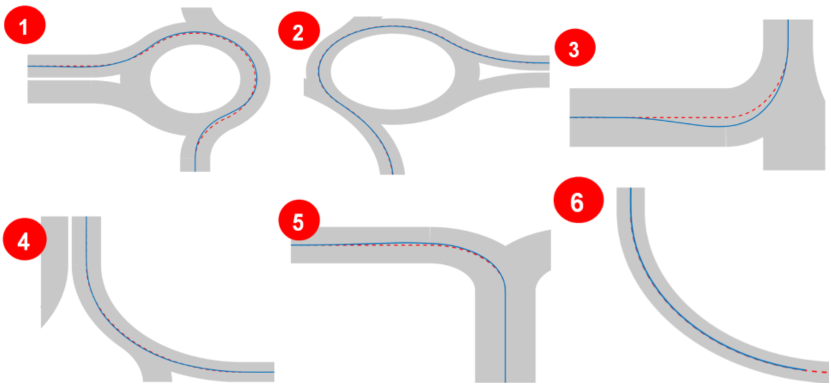

Figure 11: Urban track, 6 trajectory curves detail (red: reference; blue: real trajectory)

In Figure 12 some metrics are displayed like speed reference or real, lateral deviation, acceleration or wheel toe angle. It shall be remarked the fact that longitudinal and lateral tracking is done simultaneously as both dynamics are dependent on each other. That is an important feature because it allows avoiding if for instance, our planner tells the vehicle to do a curve at a speed which is not safe, the MPC solver will reduce the speed in order to maintain the safety parameters like maximum lateral acceleration.

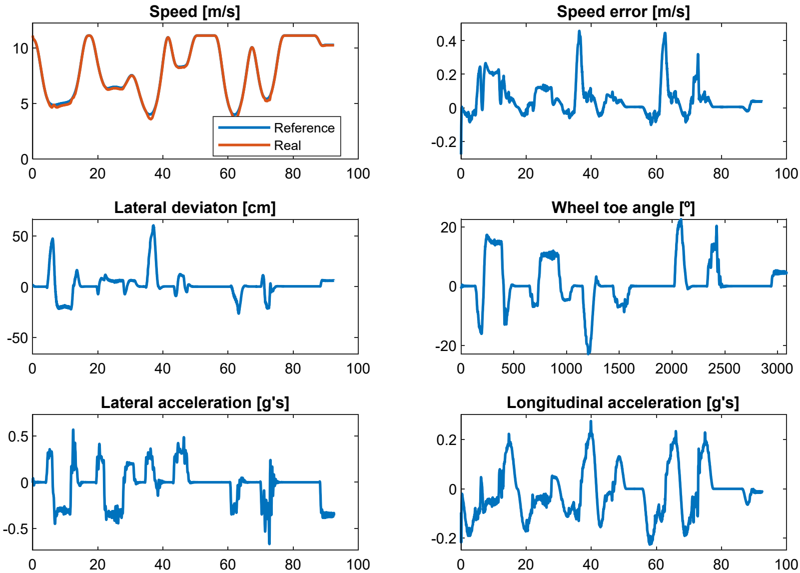

Figure 12: Analytical results at urban scenario

The second scenario is one of Idiada's testing tracks. The idea of this test was to adjust the controller for very demanding situations. Also, the aim of the test was to quickly evaluate the cost function weights without a trajectory planner module. For this, it was decided to use a driving simulator (Figure 13) to generate the trajectories., An experienced driver executed track laps at different levels of intensity and telemetry was transmitted in real time to give references to the controller. An extreme race car was used for this experiment.

Figure 13: VI-Grade Compact Simulator

The results in Table 1 show that the controller is able to follow the trajectory even at higher levels of lateral accelerations. Lateral deviation increases but in an acceptable range considering that that vehicle was driven at the limit reaching drifting behavior in some curves.

Table 1: Results from following a real driver at different intensity

	Reference time	Lap time	Average lateral deviation	Max. lateral ac-celeration
Relax lap	62.9 sec	62.52 sec	13 cm	1.41 g
High intensity	48.3 sec	48.81 sec	36.31 cm	2.22 g
Limit lap	44.6 sec	45.3 sec	59.39 cm	2.75

4.2 MPC Off-line Planner optimizer

For showing the effectiveness of the planner three different scenarios are chosen to be analyzed: a double lane change maneuver, again Idiada's racing track 'Dry Handling' and an S track.

Double lane change is chosen as it is a demanding maneuver which is very useful to evaluate the lateral dynamics behavior. Starting at a speed of 80 km/h, the trajectory in blue in Figure 14, based on splines considering geometric kinematics, is given as an input to the planner optimizer. In orange, the optimized trajectory based on the dynamic model is obtained. It can be seen in Figure 15 that the jerk and sideslip are considerably reduced as compared to the simulation with the not optimized trajectory, hence producing a comfortable trajectory for the passengers.

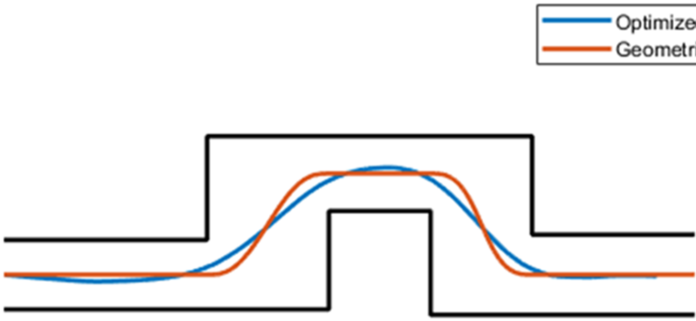

Figure 14: Trajectory comparison between input and optimized

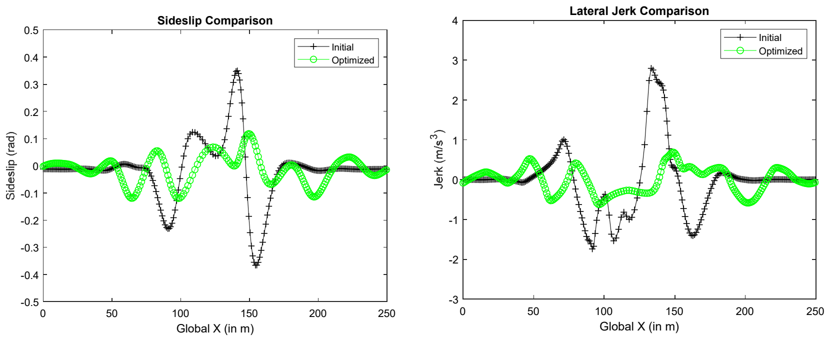

Figure 15: Double lane change trajectory optimization metrics

On the second analysis, Dry Handling track is used for the optimisation. With this kind of scenario, the planner not only has to optimize the next curve, but it also has to take into account the following ones.

As an input centre line was given and the result is in Figure 17.

Figure 16: Dry Handling testing track in Idiada

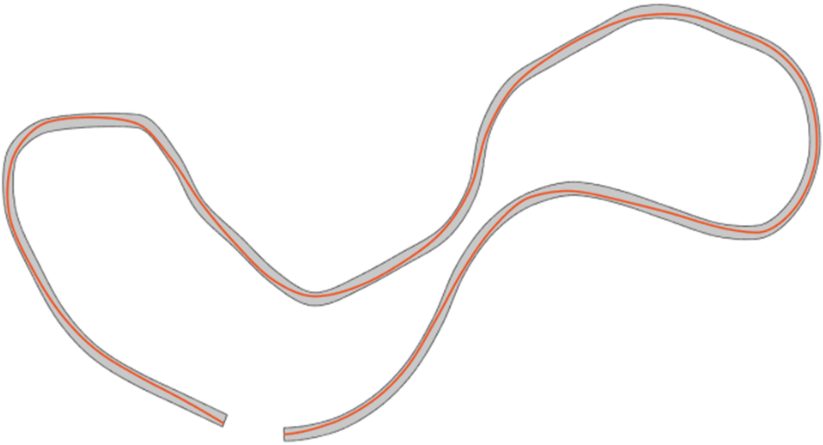

Figure 17: Dry handling optimized

Finally, the third scenario was used to demonstrate the possibility to have different optimised trajectories, depending on the adjustment of the cost function of the dynamic planner. This feature shall be used to define different driving behaviours that shall be tuned in order to fit the driving scenarios or the preference of the people inside the vehicle.

In this case, an S track is optimized for two different sets of weights in the cost function of the planner: one where sideslip is minimized and the other one where lateral jerk is the metric of focus. In Figure 18 differences can be observed.

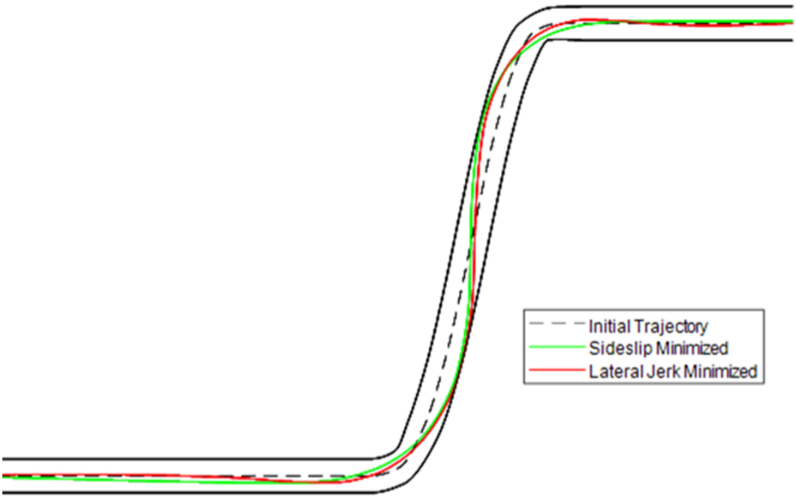

Figure 18: S track optimization

5 Future developments

The future work will focus on the following points:

1. Real time feasibility and integration of the on-line planner based on LPV.
2. Validation and testing of the dynamic controller with subjective feedback.

For the second point, the plan is to use a dynamic simulator. Idiada has recently acquired a DiM250 dynamic simulator from VI-Grade, which will be ready to use in June of 2019.

The simulator will be used in order to refine the cost functions in terms of the metrics that are considered and in ters of the balance between different weight factors. In fact, the simulator allows to receive a direct subjective feedback about how the vehicle motion is perceived by the people in the vehicle.

The big difference with tests done so far will be the capability of evaluating the strategies with one or two passengers inside the car that will be able to do objective and subjective analysis. Compared to tests that can be done in a real car, the dynamic simulator is a safe environment and also very adjustable and easy to change from one situation or scenario to another one. In this line of research, Idiada has integrated the AV Scaner Studio software, to facilitate the task of generating scenarios.

Figure 20: Driving simulator DIM 250

6 Summary and conclusions

In this paper route optimization and control strategies based on Model Predictive techniques is presented for autonomous vehicles. Both strategies have been already tested in a simulation environment and will soon be tested in a dynamic simulator. MPC has proven to be suitable for autonomous driving applications as behaves very similar to how human drivers act.

The authors believe that vehicle motion and will be an important aspect to consider for achieving the best driving experience for the passengers of the future. For this, autonomous driving controllers shall incorporate vehicle dynamics aspects in order to achieve a high level of acceptance in the transition from human to self-driving driving vehicles.

7 References

[1] 'Search-based optimal motion planning for automated driving', Zlatan Ajanovic1, Bakir Lacevic, Barys Shyrokau, Michael Stolz and Martin Horn, August 2018

[2] 'Real-time Motion Planning with Applications to Autonomous Urban Driving', Yoshiaki Kuwata, Gaston Fiore, Emilio Frazzoli, September 2009

[3] 'Autonomous racing using Linear Parameter Varying -Model Predictive Control (LPV-MPC)', Eugenio Alcalá, Vicenç Puig, Joseba Quevedo, Ugo Rosolia