Multi-Stage Model Predictive Control for Trajectory Tracking problem on a real autonomous car

Vítor Alveirinho
vitor.alveirinho@tecnico.ulisboa.pt
Instituto Superior Técnico

***Abstract*—This paper presents a Model Predictive Control (MPC) approach to the trajectory tracking problem at real-time to an autonomous car. The MPC is a technique based on solving an optimal control problem. Known as Multi-Stage MPC (MSMPC), the technique presented here is based on a tree-like expansion in order to avoid solving the optimal control problem. This expansion is made with a functional as reference, as well as in the "common" MPC, and the plant used is the non-linear vehicle's dynamics. This strategy is also capable of dealing with time varying uncertainties, leading to a robust controller's implementation.**

I. Introduction

A system's dynamics can be described as,

$$x_{k+1} = f(x_k, u_k), \tag{1}$$

where each step k, is given by the system sampling time and the state evolution is a non-linear plant described by $f(x_k, u_k)$, in this scenario, with respect to the vehicle's dynamics.

The MPC technique applies the optimal control signal u^* into the system, given by the value's function optimization, as the one in equation (2). The MPC strategy takes into account the prediction, over a certain horizon N_p, and minimizes the value's function cost taken into that horizon. From that sequence of optimal control signals sequence, the first one is applied to the plant and the state is updated. The MSMPC approach as a similar structure although, in order to avoid solving the optimal control problem, a discretized variation of the control signal is applied to each predicted state, i.e. for each state a new state is expanded as much times as the control signal is discretized, and to that new state this expansion is also made, until the prediction horizon is reached. When, to all the possible states, the prediction horizon is reached, the functional cost is evaluated and the state that has a minimum cost over the horizon is evaluated in order to apply the first control signal that gave origin to the sequence of optimal states. When u^* is applied to the system, the state of the vehicle is then updated. In the simulation environment, as one will see, there is no uncertainties to the system, but when performing real-time tests on the vehicle, disturbances due to measurement errors, sensors noise or bad plant parameters estimation will led to state estimation errors. To solve this situation, one robust approach of the MSMPC, that tries to model uncertainty over time will be analyzed.

$$J(x_k, u_k) = \sum_{k=1}^{N} (\hat{x}_k^T Q \hat{x}_k) + \sum_{k=0}^{N-1} (u_k^T R u_k) + F(x_N) \tag{2}$$

II. Vehicle's Dynamics

Before presenting the MPC statements and the approach taken, the controller's plant needs to be established. Two main models are used to represent the vehicle, the Ackerman model [1] and the bicycle model [2][3][4]. Due to this study goal, the task of trajectory tracking of a Fiat Seicento Elettra on controlled environments, the model chosen was the bicycle model. In this section this model will be presented and the it will be analyzed the actuating forces on the vehicle during acceleration, braking and turning. Also, a simple tire model will be presented, due to the fact that the vehicle control proposed will be tested in a controlled environment, at small velocities and low steering rate.

A. Vehicle Kinematics

The simplest way to characterize a vehicle movement is describing its kinematics, as equations 3 present.

$$\dot{x} = V cos(\psi + \beta) \tag{3a}$$

$$\dot{y} = V sin(\psi + \beta) \tag{3b}$$

$$\dot{\psi} = \frac{V}{l_r} sin(\beta) \tag{3c}$$

$$\beta = \arctan(\frac{l_r}{L} tan(\delta)) \tag{3d}$$

The vehicle pose is described by (x, y, ψ), where x and y are the vehicle position and ψ the vehicle orientation. V is the vehicle's velocity and β represents the velocity angle with respect to the vehicle and l_r and L are vehicle parameters, in particular the distance from the center of mass to the rear axle and the distance between axles.

B. Bicycle Model

Since the vehicle's kinematics are not enough to describe the vehicle movement, its need to proceed for the dynamics description.

To perform this task, as figure 1 represents, the bicycle model was used. This model was chosen due to the representation simplicity and also due to the fact that is possible to represent the actuating forces without loosing relevant information.

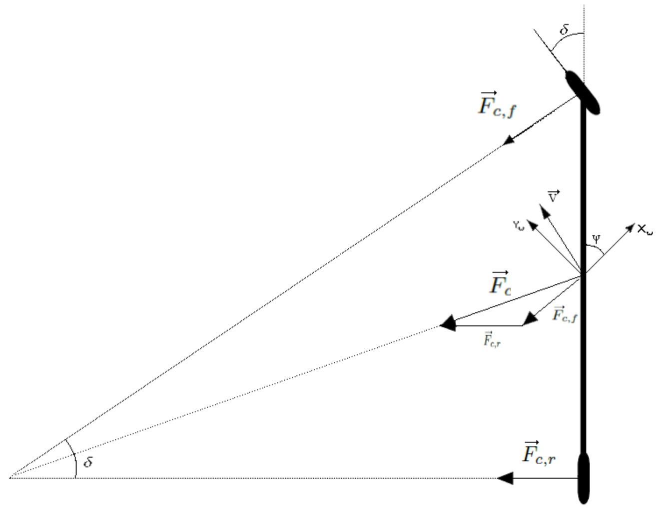

Figure 1: Bicycle model

C. Tire Model

Trying to define the real tire's behavior can be a complicated task. Since it is not the purpose of this study, in this work it will be used a simple model that will include the lateral force caused by its deformation on the road and also its motion. Tire importance is related to the fact that the only component in contact with ground are the tires and, as one can see in [5], in stressful situations, such as aquaplaning, the tire can reach performance limits, losing adherence properties that could cause the loss of vehicle control, which would not be desired. Other models, more complicated ones, such as [6], can be applied to the vehicle, although there is no necessity of using such models due to the purpose of this study.

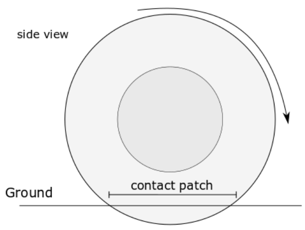

Figure 2: Tire Lateral view

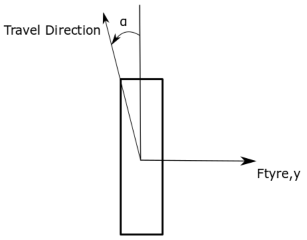

Figure 3: Tire front view

As figure 2 represents, when the tire is in contact with the ground, exists tire deformation, causing a lateral force on it, given by the following expression:

$$F_{tire,y} = -C_\alpha \alpha, \tag{4}$$

where C_α represents the cornering stiffness, that is the relation between $F_{tire,y}$ and the tire slip angle, α, when α is small. The value of cornering stiffness depends of the vehicle, the tire and the environment conditions. Due the difficulty on estimation such parameter, a similar value as the one presented in [3] is used.

D. Actuating Forces

To start vehicle's motion, motor needs to produce torque, and that must be transferred to wheels. When in motion, if $\delta \neq 0$ i.e if the vehicle is turning, several forces actuate over the vehicle due to angular acceleration, expressed in equation (5).

$$\dot{\omega} = \frac{T}{I_z}, \tag{5}$$

where,

$$\omega = \dot{\theta} = \frac{v}{L} tan(\delta). \tag{6}$$

The vehicle acceleration is related with the total forces actuating on the vehicle, as in equation (7), which will be now examined.

$$a = \frac{F_{tot}}{M}. \tag{7}$$

1) Centripetal Force: The presence of centripetal force is felt on the vehicle during turning. This force, orthogonal from where it is applied on the body, pushes the vehicle to the curvature center. It is defined, on the vehicle center of mass as in equation (8)

$$F_c = \frac{-Mv^2 tan^2(\delta)}{L} \begin{bmatrix} \frac{1}{2} \\ cot(\delta) \end{bmatrix}. \tag{8}$$

The centripetal force can be decomposed into two components, front and rear, applied on vehicle's wheels, as showed in equations (9) and (10) respectively, as figure 1 shows.

$$F_{c,f} = \frac{-Mv^2 tan^2(\delta)}{L} \begin{bmatrix} \frac{1}{2} \\ \frac{cot(\delta)}{2} \end{bmatrix}. \tag{9}$$

$$F_{c,r} = \frac{-Mv^2 tan^2(\delta)}{L} \begin{bmatrix} 0 \\ \frac{cot(\delta)}{2} \end{bmatrix}. \tag{10}$$

2) Traction Force: The engine torque produced is applied directly on the rear vehicle wheels, causing the vehicle motion. Due to dissipative factors, on both and rear wheels there is a traction force that contradicts the vehicle's movement.

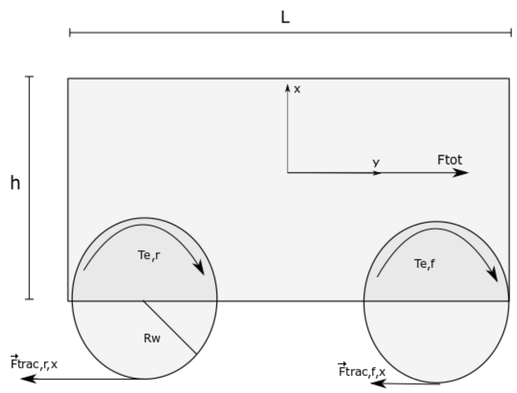

Figure 4: Lateral Vehicle view and traction forces

During braking, a torque is generated on the wheels, contradicting the car movement. This torque T_{brake} is distributed as the following,

$$T_{brake} = 2\frac{T_{e,f} + T_{e,r}}{1 + cos(\delta)}. \tag{11}$$

F

The total torque acting on front and rear wheels is given by equations (12) and (13), where it is applied the torque generated on engine, or brakes, and the dissipative effect coming from ground contact.

$$T_{mov,f} = T_{e,f} - F_{trac,f,x}R_w, \tag{12}$$

$$T_{mov,r} = T_{e,r} - F_{trac,r,x}R_w. \tag{13}$$

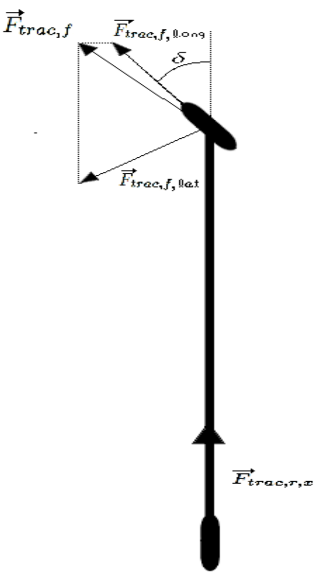

Figure 5: Traction forces

3) "Rotational" Force: When the car is in movement, exists a force that is responsible for the angular movement of the vehicle, that can be decomposed in front and rear components, and those can be decomposed in two components each, longitudinal and lateral, which are given by:

$$F_{tor,f,y} = -\frac{\dot{w}I_B}{L} \tag{14}$$

$$F_{tor,f,x} = F_{tor,f,y}tg(\delta) \tag{15}$$

$$F_{tor,r,y} = \frac{I_B - I_z}{I_B}F_{tor,f,y} \tag{16}$$

4) Total Force: After describing the forces that are produced by the vehicle and the ones that actuate on the vehicle, the front and rear produced force are given by equations (17) and (18) respectively.

$$F_{prod,f} = \begin{bmatrix} F_{prod,f,x} \\ F_{prod,f,y} \end{bmatrix} = F_{c,f} + F_{tor,f} + \begin{bmatrix} cos(\delta) & -sin(\delta) \\ sin(\delta) & cos(\delta) \end{bmatrix} \begin{bmatrix} F_{trac,f,x} \\ F_{tire,y} \end{bmatrix}. \tag{17}$$

$$F_{prod,r} = \begin{bmatrix} F_{prod,r,x} \\ F_{prod,r,y} \end{bmatrix} = \begin{bmatrix} F_{trac,r,x} \\ 0 \end{bmatrix} + F_{c,r} + F_{tor,r} + \begin{bmatrix} 0 \\ F_{tire,y} \end{bmatrix}. \tag{18}$$

The total produced force that actuates longitudinal to vehicle, causing motion is given by equation (19), while the one that actuates laterally to the vehicle is given by expression (20).

$$F_{prod} = F_{prod,r,x} + F_{prod,f,x}, \tag{19}$$

$$F_{lat} = F_{prod,r,y} + F_{prod,f,y}. \tag{20}$$

5) Dissipative Force: During motion, dissipative effects contradict the vehicle's movement. Those forces are caused by ground friction, given by expression (22), by the drag caused due to the vehicle form, with the form as in equation (23) or due to inclination on the road, as expressed in (24). The dissipative forces are the summation of these three components.

$$F_{diss} = F_{at} + F_{grav} + F_{drag}, \tag{21}$$

$$F_{at} = K_{atr,lin}V + K_{atr,quad}V^2, \tag{22}$$

$$F_{drag} = \frac{\rho_{air}ThV^2}{2}, \tag{23}$$

$$F_{grav} = Mgsin(\phi_{road}). \tag{24}$$

6) Total Force: To conclude, the total force acting on the vehicle is given by expression (25). Is easy to conclude that the total force produced is given by all the forces produced on the vehicle and the ones that contradict the movement.

$$F_{total} = F_{prod} - F_{diss} \tag{25}$$

In order to avoid phenomena like sliding and slipping the maximum force that can be exerted need to be below a certain threshold, given by expression (26), where $g = 9.8m.s^{-2}$ and μ is the friction coefficient that changes due to factors such as dry or wet pavement, tires conditions and also from road surface characteristics (asphalt, gravel ,etc...).

$$F_{max} = \frac{gM}{2}\mu_s \tag{26}$$

III. Multi-Stage Non-linear MPC

Model Predictive Control, also known as Receding Horizon Control (RHC), is a control technique where the current control action taken at present step, given by a finite horizon open-loop optimal control problem and next action, is chosen due to the minimization of a value function, expressed as in equation (27), using as initial state the current state of the plant [7].

$$\begin{aligned} J(x_i, u_i) = &\textstyle\sum_{i=1}^{N}((\hat{x}_i - x_{ref})^T Q(\hat{x}_i - x_{ref}) \\ &+ \textstyle\sum_{i=1}^{N}((u_i - u_{i-1})^T R(u_i - u_{i-1})) \\ &+ F(x_{N_p}) \end{aligned} \tag{27}$$

In order to implement a Model Predictive Controller, a strategy of implementation was defined. Known as Multi-Stage MPC, this approach is based on a tree-like structure where, beginning on the current state of the plant, each control signal is iterated, generating a new state and attributing a cost given the value function. To each one of the new states generated, this technique is again applied until the prediction horizon is reached. When all the states at the prediction horizon are computed, the best one is chosen and the first control signal of that sequence is sent to the vehicle.

Briefly speaking, since more details will be given further, the vehicle control input signal consists in two variables, the motor torque and the steering angle. To each one of those control signals a discretization is made. For example, if there exists three possible values for the motor torque and six for the steering angle, all possible combinations between those two variables are made. In this scenario, there are eighteen combinations and, from the initial state, eighteen new ones are generated. To each new state, this action will be repeated until the horizon is reached. In this section, a robust approach will be presented and analyzed. Further, implementation details and simulations will be presented.

A. Robust Multi-stage Non-linear MPC

Due to several reasons, an on-line system, where data is obtained from sensors, is always subject to uncertainty. To deal with this issue, the proposed technique to solve this situation is a tree-like structure controller's implementation.

This robust implementation is the conjugation of two tree-like structures. The basic idea behind this is to create tree branches, depending on different uncertainties' types and values. With this, the model is able to "decide" what is the uncertainty that is more adequate to the current state and which is the state evolution that models the trajectory tracking better.

At each time-step, acquisition of new data relative to the system's state takes place, each decision taken by the controller can guide the state to a more accurate one, creating in that way a closed-loop non-linear robust MPC. Each uncertainty value taken at each branching of the tree structure of the MPC is previously defined. This means that it cannot be updated, i.e it will not be considered time varying uncertainty. However, since at each time-step new measurements are taken, there is system update avoiding in that way the accumulation of state's uncertainty.

A system can be perturbed due to several factors, like wind disturbance, sensor noise, odometry errors, parameter estimation errors and unmodeled effects. Phenomena like those will add uncertainty to the system's state, so there is a need to ensure that such additions do not lead to state constraints' violations [3].

$$x_{i+1} = f(x_i, u_i) + w(x_i, u_i). \tag{28}$$

A system that is subject to perturbations can be described like the one in equation (28), where $w(x_i, u_i)$ represents an additive uncertainty where it is assumed that $w(x_i, u_i) \in \mathscr{W}$, where $\mathscr{W}$ its a compact set that contains the origin.

B. Robust horizon

As described, the MPC is based on optimal control over a certain prediction horizon. In the implemented strategy a new horizon arises for the uncertainty terms [8]. The idea behind Multi-Stage Robust MPC is to perform a tree-like structure, as the one in the Multi-Stage MPC, but now contemplating certain values of uncertainty. For this, there is a need to quantify uncertainty, which may not be very accurate, discretize it, and assume that it will be either an additive or a multiplying uncertainty. With this, the tree-structure will increase its dimension, which could be problematic when expanding each node, since more computational time and will be taken in order compute the control signal with this new feature.

Figure 6 represents a tree-structure that contemplates this additional feature.

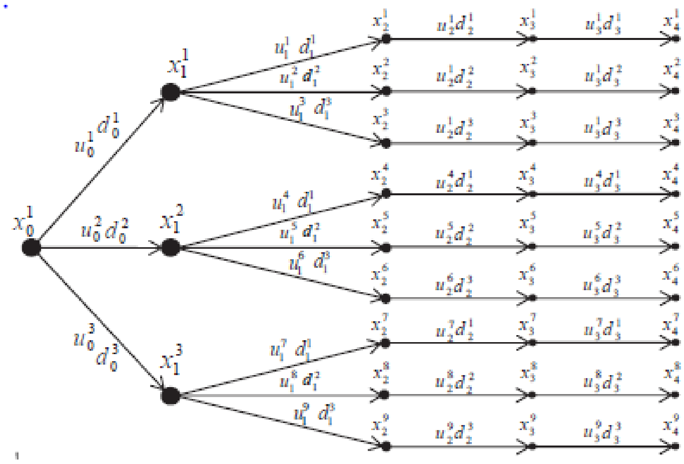

Figure 6: MPC implementation structure with uncertainties [8]

It is also possible to expand each state without considering new control signals and only considering uncertainty addition. This will allow predict more accurate states, in which the controller takes into consideration possible time-varying uncertainty under a certain horizon (Robust Horizon).

The design of vehicle control systems requires sufficiently accurate system models in order to achieve a desired level of performance [3].

One major drawback of this approach to deal with uncertainty is that complexity increases with the power of N_p, the prediction horizon. If, for each stage the number of stages increases, with uncertainty insertion, the algorithm will become heavy and difficult to compute in a reasonable time. One solution proposed in [8] is to, starting on a certain stage, only expand the state itself by one type of uncertainty, as figure 6 shows, which the next state could also follow this approach. However, since we are dealing with a real-time autonomous vehicle, this approach is, yet, not plausible.

C. MPC structure

One needs to define the system's state representation and which are the control inputs, in order to perform the trajectory tracking problem implementation. Equations (29) and (30) represent the vehicles' state and the input signal, respectively.

$$S = \begin{bmatrix} X \\ Y \\ \psi \end{bmatrix}, \tag{29}$$

$$u = \begin{bmatrix} T_{e,r} \\ \delta \end{bmatrix}. \tag{30}$$

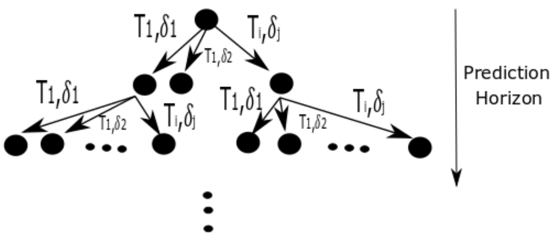

Figure 7: MPC implementation structure

The control signals, motor torque $T_{e,r}$ and steering angle δ, were iterated over the previous values to generate new possible states. In each one of these states, the cost associated with the value function and a new set of states was computed. The number of expanded states is determined by the prediction horizon. As heuristic, we decided that if a state cost exceeds a certain threshold, then there is not necessary on proceeding with this expansion. The Multi-stage MPC pseudo code is here shown to ease the comprehension on the implemented algorithm.

Algorithm 1 MPC algorithm

```
function MPC(x)
    for T_{e,r} in SomeTorqueRange do
        for δ in SomeDeltaRange do
            x_new = CreateNewState(x,u = (T_{e,r}, δ))
            MPC(x_new)
        end for
    end for
    if Horizon reached then
        if Is Best? then
            best = x
        end if
    end if
    Return to step 1 with x = best
end function
```

This approach, although an optimization problem would not be explicitly solved, has an increase complexity since that the state expansion is done for all the possible combinations of admissible control signals. This situation can be problematic for big prediction horizons, but as it will be further shown, it is completely feasible for this scenario. To evaluate this implementation, a set of tests was created. To understand the impact of the cost function parameters and control horizon would take on the trajectory tracking for different trajectories.

IV. Simulations

In order to understand the MPC trajectory tracking solution, one needs to start with simpler trajectories such as, for example, straight lines. When those simpler tests are completed successfully then one can start with more complex ones. For each situation, all the parameters should be extensively studied in order to avoid unexpected vehicle behavior. Afterwards, a conclusion about value function components, control horizon and other parameters such as sampling time, will be taken.

In this work, it is necessary to understand the value function used is represented in equation (27), where $Q^{3\times3}$ and $R^{2\times2}$ are two diagonal matrices. The second term of the value function was implemented as an artifact in order to create a stoppage criteria for the vehicle.

The trajectory chosen to perform the simulation of MPC using the vehicle's dynamics as the controller's plant was an eight-like trajectory. Its dimensions take into account a possible trajectory to test the vehicle on IST car park. Among several tests, seven were chosen as the most relevant ones, where the cost function parameters are represented in following table.

Value Function Parameters					
Test	$Q_{1,1}$	$Q_{2,2}$	$Q_{3,3}$	$R_{1,1}$	$R_{2,2}$
1	1	1	1	0.1	0.1
2	1	1	1.5	0.1	0.1
3	1	1	1.5	0.1	1
4	1.1	1.1	1.6	0.1	0.1
5	0.5	0.5	1.5	0.1	0.1
6	0.5	0.5	2	0.1	0.1
7	0.5	0.5	1.5	0.25	0.1

Table I: Cost function parameters

Figure 8 shows the trajectory obtained from tests presented above. Among those, the 2^{nd} and 5^{th} were considered the best ones, and can be analyzed separately on figure 9.

Those comparisons were made not only regarding the trajectory performance, but also the vehicle's velocity and orientation variation, as both figures 10 and 11 show, respectively.

After a careful analysis, the following was concluded: Parameters $Q_{1,1}$ and $Q_{2,2}$, in which $Q_{1,1} = Q_{2,2}$, represent the weight of a strict trajectory following. $Q_{3,3}$ is the weight given to the orientation reference following. Matrix R represents the difficulty on changing input parameters. If $R_{1,1}$ is a large value, it will create resistance on changing the vehicles' velocity. The same type of behavior for $R_{2,2}$, instead that this one characterizes the difficulty on changing the steering angle.

Test 6 was the one with biggest deviations from the defined trajectory. This happened due to a high $Q_{3,3}$ value, when comparing to $Q_{1,1}$, meaning that the car would follow the path contemplating more the orientation than the position. This caused that deviations from trajectory were not so heavy penalized, causing such oscillations. In

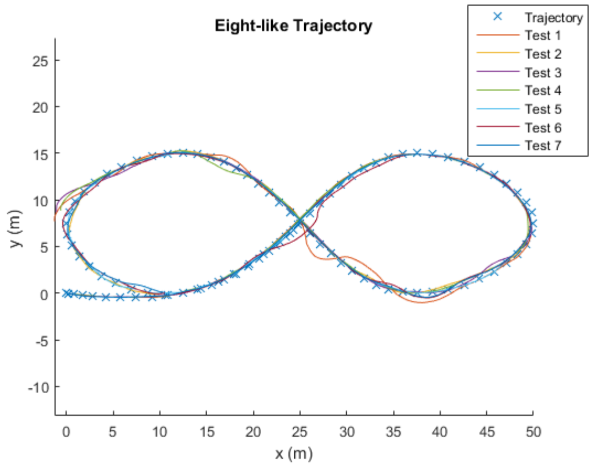

Figure 8: Path in a eight-like trajectory

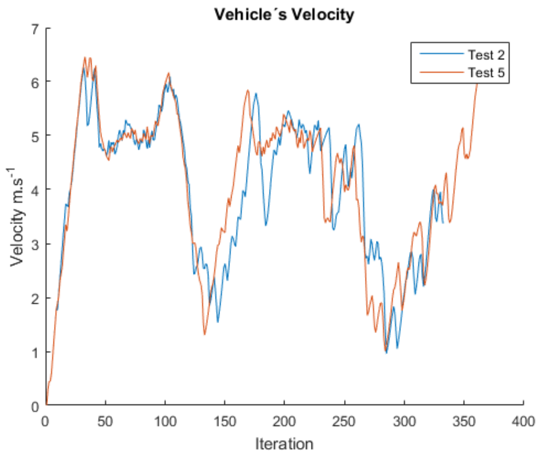

Figure 10: Velocity for test 2 and 5 of an eight-like trajectory

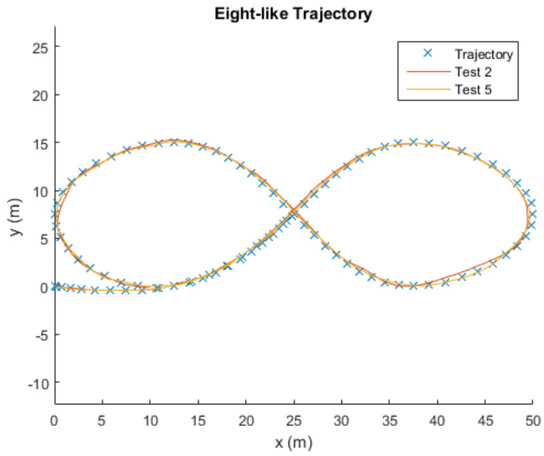

Figure 9: Best tracking results for an eight-like trajectory

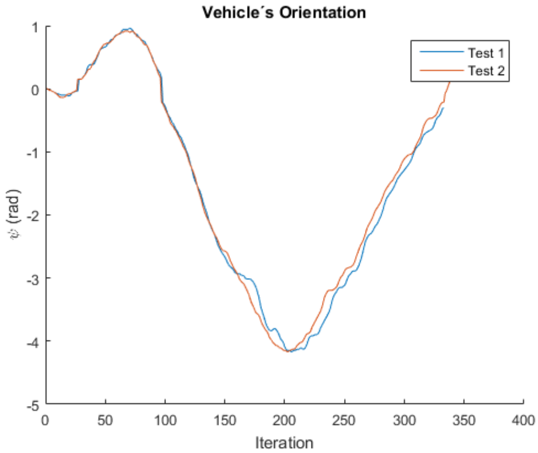

Figure 11: Orientation for test 2 and 5 of an eight-like trajectory

relation to tests one, three, four and seven, which do not complete the trajectory proposed, several factors can be pointed. When relation between $Q_{1,1}$ and $Q_{3,3}$ is equal or similar, the vehicle did not complete the task. This comes with the fact that, when computing the state cost, both parameters have similar relations and would led to change the "way" that the vehicle follows the trajectory, causing it to fail. Also, when $R2,2$ is bigger, it causes difficulty in changing the steering angle, which during turning can be problematic if one needs sudden change, as one can see in test three. Finally, on test seven, one can notice that this test is similar to test five, which is a successful one, but a change on $R_{1,1}$ caused failure. This comes with the fact that this parameters weights the velocity's variation. If the controller has difficulties on accelerate (or decelerate) the vehicle, specially during turning, where the vehicle needs lower velocity in order to avoid sliding and perform correctly the curve, it will exit the track due to a higher velocity than the one desired. This last case led us to a problematic situation. If $R_{1,1}$ is to high, than the vehicle will have difficult in reducing velocity for turning, while if its to low, then a control signal for velocity, as figure 10 presents, gives an oscillatory behavior. This behavior is not desired, as one could imagine a vehicle constantly changing its acceleration, causing an unpleasant travel.

In this test, where two full rotations were made, passing the critical situation border transition on the vehicle's orientation, since it is defined between $[0, 2\pi[$, the results obtained were considered satisfactory.

A. Prediction horizon and computational time

As shown previously, the effects that parameters Q and R provokes are now known. Other parameter that affect the value function minimization is the prediction horizon, N.

As one may think, bigger the number of steps, better the prediction. This may be true, however there is a large increasing in the complexity when prediction horizon grows.

Figure 12 represents the execution time for one, two and three iterations. In order to have a better understanding about the figure above, one should observe the following table:

N	Average	Median	STD
1	0.0173	0.0170	0.0016
2	0.1429	0.1464	0.0171
3	1.3469	1.3620	0.2089
4	9.7705	-	-

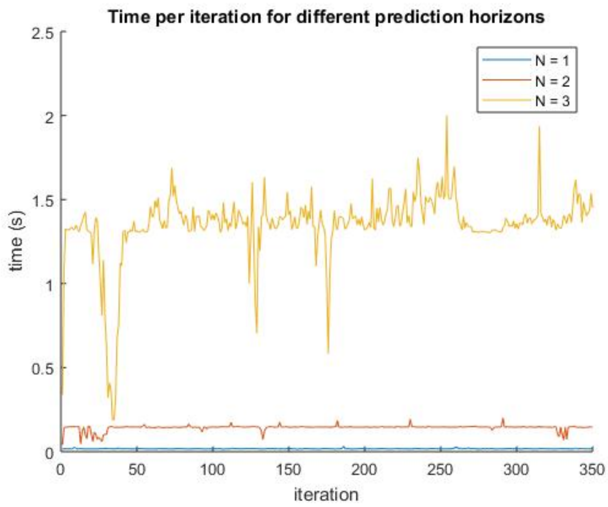

Figure 12: MPC execution time for different N's

Table II: Statistical features for different prediction horizons

As predictable, increasing the prediction horizon would increase the execution time, once that for each n states, n^2 new ones would be generated. For $N=1$ the execution time is almost insignificant but, only one prediction is made and its that same control signal that will be used as input, loosing the controller's prediction component. For the other three horizons analyzed, $N=4$ was automatically excluded due to the fact that having a controller that takes about ten seconds to produce a control signal for real-time trajectory tracking is completely unpractical. So now there are left two other horizons that could be considered.

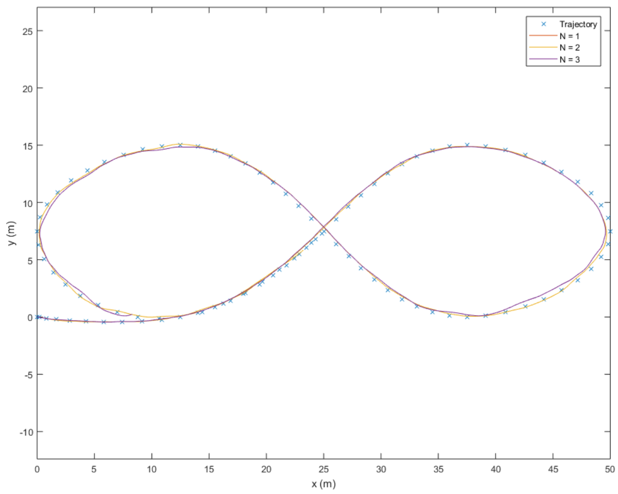

Figure 13: Trajectory for different prediction horizons

Figure 13 demonstrates the results for the eight-like trajectory when changing the prediction horizon. Although the task was only completed for $N=2$, due to the only fact that the value function was "calibrated" for that horizon, can be seen that for both scenarios, the result can be obtained. So the choice can now be taken in favor of the one with the lowest execution time, given a system with more control samples and if tested in real-time, more controllable one.

B. Sampling time

While performing the simulations on subsection IV-A, it can be verified that the average execution time for $N=2$ is slightly above 0.1 seconds. So, if executing the MPC live, it would generate a problem, since the execution time is bigger than the chosen sampling time.

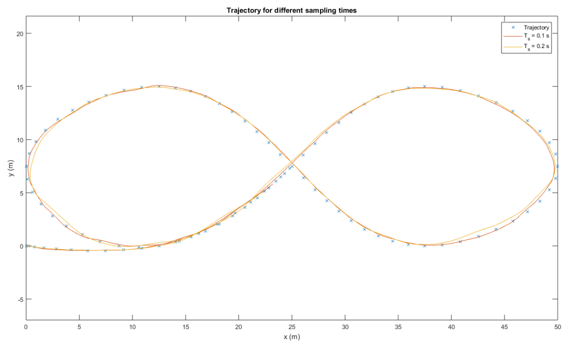

Figure 14: Trajectory for different sampling times

Analyzing figure 14, although the value function was only calibrated for one situation, it can be seen that for smaller sampling times, the controller reacts faster, having also a better following to the defined trajectory. So, the lower the sampling time, the better, only limited by the execution time. For this situation the solution for having a sampling time that is lower than the MPC execution time is to create a signal that, at every sample, interrupts the MPC execution and extracts the best solution already computed. This solution does not guarantee optimal solutions, but sub-optimal ones, which given the results is sufficient for this problem. Several MPC implementations as [9], also purpose sub-optimal solutions reducing the problem .

C. Trajectory tracking at constant velocity

One problem that can be seen analyzing figure 10, is the large velocity variation. To solve this situation it was proposed that, instead of having control over the vehicle's velocity and steering angle, only have control over the steering angle. The first advantage of this situation is that, instead of having a state's combination that have different velocity and steering angle values, only have states that correspond to the ones generated only by the steering angle variation. With this, several different advantages' exist. The first is to maintain the same number of control signals for the steering angle, reducing that way the number of states that would correspond to the velocity's variation. With it the algorithm would be lighter and faster. The second one is to maintain the number of states of expansion, having more possible steering angles, creating in that way, smoother trajectories. The third one is a combination of both, when it is possible to reduce the number of states and at the same time increase the number of values possible for the steering angle. For last, it is also possible reducing the number of states (increasing or not the number of possible values for the steering angle) and having a bigger prediction horizon without increasing the execution time of the MPC.

To test this possibility, while executing the eight-like trajectory, it was increased the prediction horizon from two to three and doubled the amount of possible steering angle values. The result, as figure 15 presents, is representative of the trajectory made at six different constant velocities.

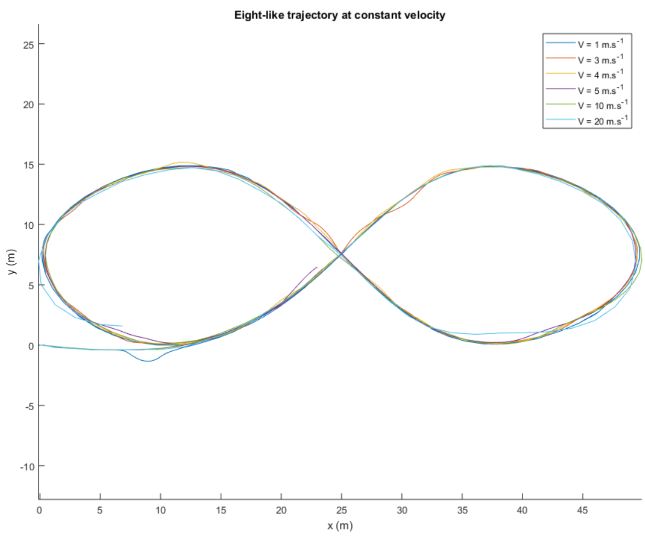

Figure 15: Eight-like trajectory at constant velocity

For small velocities, there is an almost perfect approach to the proposed trajectory, and even a smoother orientation's variation due to the fact that the increment of the steering angle being lower. For higher velocities, such as $10m.s^{-1}$ and $20m.s^{-1}$, there is a significant deviation from the trajectory, which is completely reasonable if we imagine a person performing a trajectory like this at about $72km.h^{-1}$. To verify the increase of the turning smoothness, one can compare the vehicle's orientation variation at constant velocity, represented in figure 16 to the previous test, represented in 11.

V. Conclusions

The main goal of this work was to create a controller able to perform real-time trajectory tracking for an autonomous car. This problem arose from the possibility to create an autonomous car, in specific, to transform a "normal" Fiat Seicento Elettra, starting from scratch. Presented in this thesis, on section II, were described the vehicle's dynamics. This had the purpose to serve as the controller's plant in order to simulate the vehicle behavior, for different trajectories.

Further, some mathematical background knowledge over the Multi-Stage MPC was presented in order to give the reader a deeper perspective about the controller. After, implementation details were explored, in order to validate

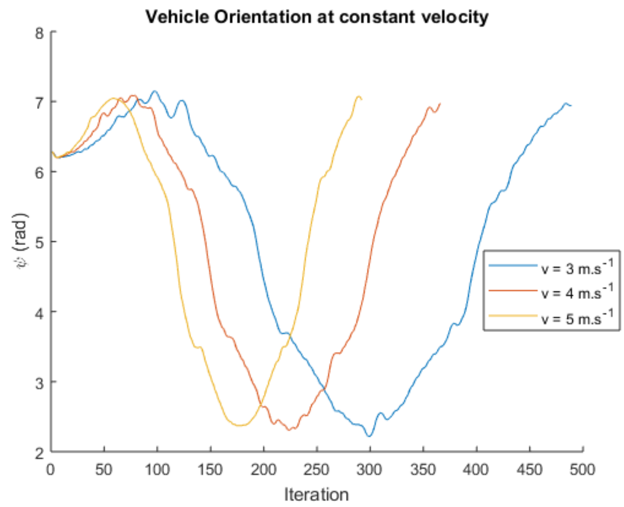

Figure 16: Orientation variation at constant velocity.

the controller's approach with the vehicle's dynamics as the MPC's plant. To validate this implementation, an eight-like trajectory was simulated and evaluated. For this complex trajectory, several tests concerning value function parameters, sampling time and prediction horizon were made. It was possible to verify that lower the sampling time, and higher the control horizon were, better were the results. But there was trade-off, computational time. Knowing that a trajectory tracking system must react fast to the perform is task, there is a necessity of having small sampling time in order to be capable of processing new (and possibly unexpected) information, not only trajectory information but also, measurements information. So a choice between execution time and the prediction horizon value must be made, concerning this details. As it is possible to verify on that same section, for several situations the prediction horizon was two samples and, in situations like constant velocity, three samples. On these scenarios it is possible to have a sampling time of 0.1 seconds which, when compared to a human response to an unexpected situation is at least ten times faster [10]. When compared with to continuous driving, if we imagine the situation of driving at $15m.s^{-1}$ ($54km.h^{-1}$), where at a sampling time the car travels $1.5m$, it can be really significant not having any control signal over the vehicle.

Regarding all, we considered that the work performed could be able to answer the expectations of the proposed project, although not optimally.

References

[1] P. Simionescu and D. Beale, "Optimum synthesis of the four-bar function generator in its symmetric embodiment: the ackermann steering linkage," *Mechanism and Machine Theory*, vol. 37, no. 12, pp. 1487 – 1504, 2002.

[2] J. Ackermann, "Robust car steering by yaw rate control," in *29th IEEE Conference on Decision and Control*, pp. 2033–2034 vol.4, Dec 1990.

[3] J. Ryu, *STATE AND PARAMETER ESTIMATION FOR VEHICLE DYNAMICS CONTROL USING GPS*. PhD thesis, stanford university, dec 2004.

[4] J. Ackermann and W. Sienel, "Robust control for automatic steering," in *1990 American Control Conference*, pp. 795–800, May 1990.

[5] W. Sienel, "Estimation of the tire cornering stiffness and its application to active car steering," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 5, pp. 4744–4749 vol.5, Dec 1997.

[6] J. Lacombe, "Tire model for simulations of vehicle motion on high and low friction road surfaces," in *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, vol. 1, pp. 1025–1034 vol.1, 2000.

[7] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.

[8] S. Lucia, *Robust Multi-Stage Non-Linear Model Predictive Control*. PhD thesis, Technischen Universität Dortmund, 2015.

[9] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Transactions on Automatic Control*, vol. 44, pp. 648–654, Mar 1999.

[10] D. V. McGehee, E. N. Mazzae, and G. S. Baldwin, "Driver reaction time in crash avoidance research: Validation of a driving simulator study on a test track," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 44, no. 20, pp. 3–320–3–323, 2000.