Steering and Speed Control for Autonomous Electric Vehicles

Filipe Parrado
filipe.parrado@tecnico.ulisboa.pt
Instituto Superior Técnico, Portugal

***Abstract*—Self-driving vehicles have in recent years come to the forefront of engineering studies as a commercially viable solution. There are many objectives associated with autonomous driving, namely minimising the error between the vehicle and the reference provided, optimising either the time or the energy needed to complete the self-driving task, and most important of all, accomplishing these objectives safely. A solution that tackles all of these problems simultaneously is the formulation of the autonomous driving task as an optimisation problem with multiple, often concurrent, objectives. By doing so, it is possible to use optimal control techniques to solve the autonomous driving problem. One such optimal control technique is Model Predictive Control. In this work, a controller is proposed to solve the autonomous driving task, formulated as a path-following problem, while minimising the energy needed. This controller consists of an outer loop that provides a minimum energy speed reference profile and an inner loop that uses Model Predictive Control to compute the optimal steering and acceleration commands that lead the vehicle through the provided path. To test the controller, a robust and complex model of VIENA, an eletric car developed by IST was created. The controller is then compared against well-known path-following controllers and it is capable of producing results that are either better or on par with the aforementioned controllers. This was accomplished while producing a smooth steering signal, a relevant contribution to provide a safer driving experience.**

I. Introduction

In 2016, around 1.4 million people were injured and 25600 lost their lives in road traffic accidents in the Member States of the European Union [1]. In 2018 approximately 25100 road fatalities were reported by the Member States [2]. In 2016, a study carried by the United States of America NHTSA found that human error accounts for 94% to 96% of all car accidents [3]. By diminishing human error, Autonomous Ground Vehicles (AGVs) have the potential to dramatically reduce the number of traffic accidents that occur.

Driverless vehicles are, at their core, autonomous decision-making systems who, through the use of sensors, process acquired information about the surrounding environment and use said information along with prior knowledge (about the road network, driving rules, vehicle dynamics and sensor models) to make the optimal driving decision in any given situation. This decision system is hierarchically organised into four components [4]. At its highest level, a route is planned through the road network. Next, a behavioural layer is responsible for selecting a driving task that progresses the car in the way of its objective and abides by the rules of the road. This is followed by a motion planning module, that creates a continuous path through the environment to accomplish the local navigational task. A local feedback control system reactively corrects the errors in the execution of the planned path. Figure 1 illustrates the decision making hierarchy of an AGV. This work's focus is on the last decision making level, the local feedback control.

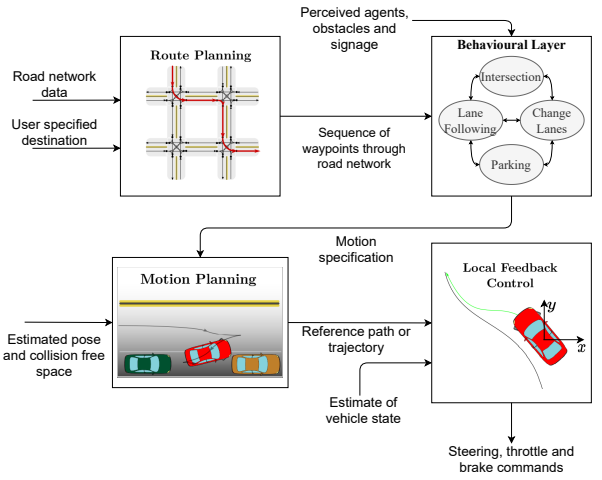


Fig. 1. The decision making hierarchy of an AGV, taken from [4] and adapted.

The objective of this work is to develop a local feedback control technique that acts on the vehicle's steering and speed and validate it in a simulation environment.

To validate the local feedback control technique, a model of an AGV must be used. The vehicle characterised in the simulation environment is VIENA[1], a car developed by IST, which is the subject of many projects, all of which culminate with the creation of a

[1]http://viena.tecnico.ulisboa.pt/pt/

fully autonomous electric vehicle. Figure 2 presents a photograph of VIENA.

Fig. 2. The VIENA car.

II. VIENA Car Model

In this section, the equations that characterise VIENA's behaviour are introduced. Subsection II-A presents the equations that govern VIENA's movement, also known as its kinematic model. Subsection II-B presents the equations that model VIENA's dynamics, from the forces and frictions that produce its acceleration, in subsection II-B1, to the model of its induction motor, and the Field Oriented Control that is used to drive the motor's rotor to its desired velocity, presented in subsection II-B3. Figure 4 shows the block diagram of VIENA's model.

A. VIENA Kinematic Model

One of the models already used to characterise VIENA is based on a simple 2D body frame dynamic model with Ackermann steering [5].

In this model, it is assumed that the vehicle is subjected to the non-holonomic constraint. Rotation of the vehicle occurs only around the z-axis, which is pointed upward. Other rotations are the consequence of suspension movement and road unevenness. The vehicle's Euler angle θ (rotation along the y-axis) at the wheel axis level is close to zero and it will be considered zero. There is neither lateral nor longitudinal slip, the tires are capable of sustaining the lateral forces generated during dynamics and they do not loose contact to the ground.

Using the assumptions above and a simplified bicycle model, the equations that govern this model are

$$\begin{cases} \dot{x} = v_r \cos(\theta)\cos(\phi) \\ \dot{y} = v_r \sin(\theta)\cos(\phi) \\ \dot{v}_r = a \\ \dot{\theta} = \frac{v_r}{L}\tan(\delta)\cos(\phi) \end{cases} \tag{1}$$

B. VIENA Dynamic Model

The model of VIENA's behaviour is composed of a model of an induction motor, controlled via Field Oriented Control (FOC), a simple gearbox, the dynamics which the vehicle is subjected to and its kinematic model. The following subsections provide further insight into the remainder of these components.

1) Vehicle Dynamics: VIENA's vehicle dynamics take as input the vehicle's wheel torque and determines the vehicle's angular wheel speed as well as its linear speed. To characterise VIENA's vehicle dynamics, one makes use of Newton's second law of motion:

$$\sum F = Ma. \tag{2}$$

The forces applied to the vehicle are the traction torque, T_t, applied in the rear wheels, the component of the force created by the vehicle's own weight in line with the vehicle's movement, F_g', the friction force applied between the vehicle's wheels and the ground, F_r, and the aerodynamic drag force applied whilst the vehicle is moving, F_a. The vehicle's total mass, M, is divided into the vehicle's mass, m, and the equivalent mass of the rotating parts, m_{eq}^{rot}. Therefore, (2) can be rewritten as

$$\frac{T_t}{r_w} + F_g + F_r + F_a = (m + m_{eq}^{rot})a \tag{3}$$

where

$$F_g' = -mg\sin(\phi) \tag{4}$$

denotes the force created by the gravitational pull, with g denoting earth's acceleration and ϕ the path's inclination,

$$F_r = -C_{rr}mg\cos(\phi) \tag{5}$$

denotes the rolling resistances force, with C_{rr} denoting the rolling resistance coefficient,

$$F_a = -\frac{1}{2}\rho C_d A_f (v_r - (-v_w))^2 \tag{6}$$

denotes the aerodynamic frag force, with ρ denoting the air density, C_d the drag coefficient, A_f the frontal area of the vehicle, v_r the vehicle's rear wheel speed and v_w the wind's speed opposite to the vehicle movement and

$$m_{eq}^{rot} = \frac{1}{r_w^2}\left(I_w + I_m g_r^2\right) \tag{7}$$

denotes the equivalent mass of the rotating parts, with I_w and I_m denoting the wheel's and the motor's rotor inertia, respectively and g_r denoting the gear ratio of the gearbox between the motor and the wheels.

Please note that the forces described by equations (4) through (6) have associated with them a minus sign because they are contrary to the vehicle's movement. The vehicle's rear wheel speed is given by

$$v_r(t) = \int_0^t a(t)dt \tag{8}$$

and the wheel's angular speed is given by

$$\omega_w(t) = \frac{v_r(t)}{r_w}. \tag{9}$$

2) Gearbox Subsystem: The adopted gearbox model simply uses the gear ratio to convert the wheel's speed into the motor's speed and the motor's torque into the wheel's torque, using the following equalities:

$$\omega_m = g_r \omega_w, \tag{10}$$

$$T_w = g_r T_m, \tag{11}$$

where ω_m and T_m denotes the motor's angular speed and torque, respectively.

3) Induction Motor and Field Oriented Control: To further characterise VIENA's behaviour, the induction motor that receives the desired reference speed and outputs the needed torque to achieve the desired speed was modelled and integrated into the overall system. The presence of the dynamics of the induction motor creates the need to control it and therefore Field Oriented Control (FOC) is used to drive the rotor speed to its desired value. The use of FOC supposes the characterisation of the induction motor behaviour in the direct-quadrature-zero, *dq0*, model. In [6] it is presented the *dq0* model as well as the FOC structure and the equations that govern it.

FOC aims to control the torque and magnetic flux of the induction machine separately, using its quadrature, i_q and direct currents, i_d, respectively. This allows for a good control of the torque while maintaining magnetic flux of the induction machine at the rated conditions. Figure 3 presents the FOC structure system.

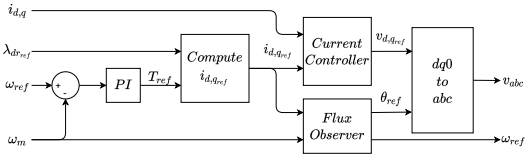


Fig. 3. Field Oriented Control Structure.

III. Proposed Controller

The controller proposed to solve the path-following problem consists of an inner MPC controller that computes the optimal acceleration and steer commands that drives the controlled variables as close as possible to their desired reference values whilst complying with the given constraints, and an outer loop speed profiler that updates the reference speed that should be tracked in case the vehicle is lagging behind the reference travelled distance. Figure 4 presents a macroscopic view of the system's architecture.

A. MPC Controller

The MPC controller, receives as input the reference states, z_{ref}, and the current vehicle state, z_{meas} and outputs the optimal acceleration and steer commands, u_{opt}, that drive the vehicle as close as possible to the desired pose and speed.

It does so by first predicting the future states by using the VIENA kinematic model. These predictions, $\bar{Z}$, are then used to linearize the VIENA kinematic model so that a Optimisation Control Problem (OCP) can be solved. This OCP outputs a sequence of vehicle control signals that are used to update the predictions made using the VIENA kinematic model. When the difference between two OCP iterative outputs is smaller than a given threshold, Δu_{min}, or the maximum number of iterations, K, has been reached, the algorithm stops and outputs the optimal control signal sequence, u_{opt}. Figure 5 presents a flowchart of the MPC algorithm.

1) Predicting VIENA's Motion: The prediction of VIENA's future pose and speed is made with the help of its kinematic model, presented in Subsection II-A, and the input signals produced by the Linear MPC's Optimisation Control Problem. This results in a series of predictions made along the prediction horizon chosen for the problem at hand.

2) Linear MPC Formulation: The optimisation control problem (OCP) solved by the MPC controller optimises the control vector, u, over a certain prediction horizon, denoted T_p. Since the implementation of an OCP implies a discretization of the problem at hand, not only the state-space model of the linearized VIENA vehicle needs to be discretized, but also the prediction horizon must be converted into a discrete number, N. Let $\Delta t \in \mathbb{R}$ denote the discretization step of the considered problem. N is therefore

$$N = \frac{T_p}{\Delta t} \in \mathbb{N}. \tag{12}$$

It is important to take into account that the OCP produces a solution for each step of the prediction horizon. Let m denote the dimension of the control vector and p denote the dimension of the state vector.

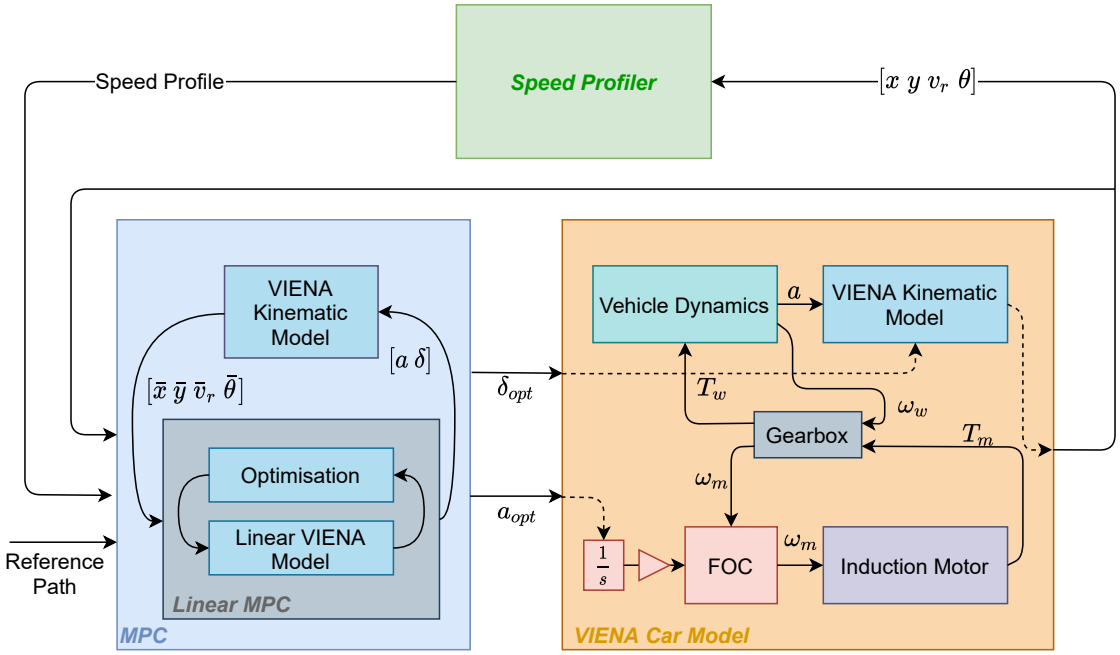


Fig. 4. Overall System Architecture.

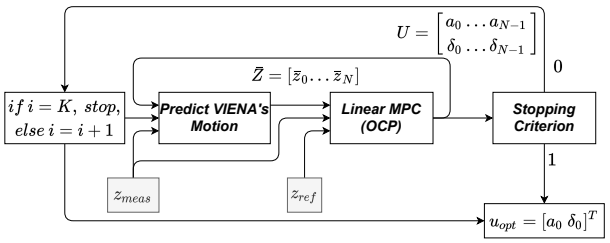


Fig. 5. MPC Algorithm.

For the considered dimensions, the OCP would output a solution in which $U \in \mathbb{R}^{m \times N-1}$ and $Z \in \mathbb{R}^{p \times N}$, where U denotes the set of control inputs and Z the states produced by subjecting the internal state-space model to the computed control signals. In this case, the state vector is given by the quantities of VIENA's kinematic model, $z = [x\ y\ v_r\ \theta]^T \in \mathbb{R}^4$, and the input vector is the acceleration and steer command computed, $u = [a\ \delta]^T \in \mathbb{R}^2$.

Let $z_k^{ref} \in \mathbb{R}^4$ denote the provided reference state at iteration k. The proposed MPC controller, which makes use of the template provided in [7] and was based on [8], with slight changes to the notation, is

$$\underset{U \in \mathbb{R}^{2 \times N-1}}{\text{minimize}} \quad (z_N^{ref} - z_N)^T W_f (z_N^{ref} - z_N) + \tag{13a}$$

$$\sum_{k=0}^{N-1} (z_k^{ref} - z_k)^T W (z_k^{ref} - z_k) \tag{13b}$$

$$+ u_k^T R u_k + \tag{13c}$$

$$(u_k - u_{k-1})^T R_d (u_k - u_{k-1}) \tag{13d}$$

subject to

$$z_{k+1} = A(\bar{z}_k) z_k + B(\bar{z}_k) u_k + C(\bar{z}_k), \tag{13e}$$

$$\delta_{min} \leq \delta_k \leq \delta_{max}, \tag{13f}$$

$$a_{min} \leq a_k \leq a_{max}, \tag{13g}$$

$$|\delta_k - \delta_{k-1}| \leq \Delta\delta_{max}, \quad k = 0 \dots N-1, \tag{13h}$$

$$v_{r_{min}} \leq v_{r_k} \leq v_{r_{max}}, \quad k = 0 \dots N, \tag{13i}$$

$$z_0 = z_{meas}. \tag{13j}$$

Equations (13a) - (13d) denote the problem's cost function, which is divided into two separate costs: the terminal cost, present in equation (13a), and the stage cost, present in equations (13b) through (13d). As it can be seen, the problem's cost function is composed by sums of weighted squared errors with $W_f \in \mathbb{R}^{4 \times 4} \succ 0$ denoting the terminal cost weight matrix, $W \in \mathbb{R}^{4 \times 4} \succ 0$ denoting the stage cost matrix, $R \in \mathbb{R}^{2 \times 2} \succ 0$ the input cost weight matrix and $R_d \in \mathbb{R}^{2 \times 2} \succ 0$ the input difference cost weight matrix. Matrix R allows the penalisation of very large inputs and matrix R_d

the penalisation of big differences between consecutive inputs, in an effort to have smoother input signals.

To find the best possible values for these matrices, a fine-tuning process was conducted. In summary, a grid search was performed to find the combination of matrices that produced the smallest possible position and speed error in the path-following problem. The matrices that produced the best results are

$$W_f = W = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1.15 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}, \quad R = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.4 \end{bmatrix}, \quad R_d = \begin{bmatrix} 0.01 & 0 \\ 0 & 1.11 \end{bmatrix}. \tag{14}$$

Moving on to the OCP's constraints, they can be broken down into two categories: the state contraints and the input constraints. Beginning with the state constraints, the internal linearized VIENA model is reflected in equation (13e), where $A(\bar{z})$, $B(\bar{z})$ and $C(\bar{z})$ are the linearized discrete matrices that model VIENA's movement. These matrices are updated using the predictions made using the VIENA kinematic model, as seen in Figure 5, and are given by

$$A(\bar{z}) = \begin{bmatrix} 1 & 0 & \cos(\bar{\theta})\cos(\phi)\Delta t & -\bar{v}_r \sin(\bar{\theta})\cos(\phi)\Delta t \\ 0 & 1 & \sin(\bar{\theta})\cos(\phi)\Delta t & \bar{v}_r \cos(\bar{\theta})\cos(\phi)\Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{15}$$

$$B(\bar{z}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \frac{\bar{v}_r \cos(\phi)\Delta t}{L} \end{bmatrix}, \tag{16}$$

$$C(\bar{z}) = \begin{bmatrix} \bar{v}_r \bar{\theta} \sin(\bar{\theta})\cos(\phi)\Delta t \\ -\bar{v}_r \bar{\theta} \cos(\bar{\theta})\cos(\phi)\Delta t \\ 0 \\ 0 \end{bmatrix}. \tag{17}$$

The remaining state constraints are the VIENA's maximum and minimum speed constraints, given by equation (13i).

The input constraints encompass the upper and lower bounds of the steer and acceleration commands and the maximum steering speed, and are given by equations (13f) through (13h).

The initial condition constraint is simply given by the measured state vector, denoted by equation (13j).

3) Stopping Criterion for the MPC: The MPC algorithm stops either when the final iteration has been reached or when the sum of the differences between two consecutive vehicle control signals does not exceed a certain threshold. When two consecutive input signals computed by the Linear MPC are very similar, there is no point in adapting the model further as the Linear MPC has reached a form of consensus on the optimal input signals.

The variation on two consecutive input signals Δu is therefore computed as

$$\Delta u = \sum_{i=1}^{N-1} |\delta_i - \delta_{i-1}| + \sum_{i=1}^{N-1} |a_i - a_{i-1}|. \tag{18}$$

Please note that in (18), $N-1$ denotes the dimension of the computed input signal matrices, as seen in III-A2.

When the variation is lower than a given threshold Δu_{min}, the MPC algorithm stops. As a mathematical expression, the stopping criterion is

$$\Delta u \leq \Delta u_{min}. \tag{19}$$

B. Optimal Speed Profiler

Since there is the need to produce a reference for the vehicle's speed, the possibility for the computation of a optimal speed profile arises. The proposed solution consists on a minimum energy speed profile optimisation that takes into account a time constraint, by providing the optimisation problem a time interval in which the track should be performed and penalising differences between said time interval and the time the vehicle took to complete the given reference path.

1) Energy Computation Method: The proposed formulation assumes constant acceleration between two points. This allows the use of the typical equations of linear motion. If the acceleration is constant then

$$v(t) = at + v_i \tag{20}$$

and

$$x(t) = \frac{1}{2}at^2 + v_i t + x_i. \tag{21}$$

Let v_i denote the speed at track point x_i and v_{i+1} denote the speed at track point x_{i+1}. Applying (20) results in

$$\Delta t = \frac{v_{i+1} - v_i}{a} \Leftrightarrow \Delta t = \frac{\Delta v}{a} \tag{22}$$

where Δt is the time interval between two spatial points, x_i and x_{i+1}. It is also possible to rewrite (21) as

$$x_{i+1} - x_i = \frac{1}{2}a\Delta t^2 + v_i\Delta t \Leftrightarrow \Delta x = \frac{1}{2}a\Delta t^2 + v_i\Delta t. \tag{23}$$

Using (22) into (23) results in

$$\Delta x = \frac{1}{2}a\frac{\Delta v^2}{a^2} + v_i\frac{\Delta v}{a} \Leftrightarrow \Delta x = \frac{1}{2}\frac{\Delta v}{a} + v_i\frac{\Delta v}{a} \Leftrightarrow$$
$$\Leftrightarrow a = \frac{1}{\Delta x}\left(\frac{\Delta v^2}{2} + v_i\Delta v\right). \tag{24}$$

Please note that, in the scope of this problem, Δx is a known value, and therefore, only Δv impacts the value of a. To compute the needed energy to complete the path, two distinct cases must be considered: the one in which $\Delta v \neq 0$ and its counterpart, i.e. $\Delta v = 0$.

Beginning with the latter, if $\Delta v = 0$, then $a = 0$ and one can compute Δt knowing that:

$$t_{i+1} = \frac{x_{i+1} - x_i}{v_i} + t_i \Leftrightarrow$$
$$\Leftrightarrow \Delta t = \frac{\Delta x}{v_i} \tag{25}$$

Since $a = 0$, the speed is constant, as is the traction power:

$$E_{T_i} = P_{T_i}\Delta t_i \Leftrightarrow E_{T_i} = F_{T_i}v_i\Delta t_i \Leftrightarrow$$
$$\Leftrightarrow E_{T_i} =$$
$$\left(mg\left(\sin(\phi) + C_{rr}\cos(\phi)\right) + \frac{1}{2}\rho C_d A_f v_i^2\right)v_i\Delta t_i. \tag{26}$$

In the case that $\Delta v \neq 0$, Δt can be obtained by solving (23) as function of Δt:

$$\Delta t_i = \frac{-b_1 + \sqrt{b_1^2 - 4a_1c_1}}{2a_1} \tag{27}$$

with

$$a_1 = \frac{1}{2}a$$
$$b_1 = v_i \tag{28}$$
$$c_1 = -\Delta x.$$

Knowing that

$$E_{Ti} = \int_0^{\Delta t_i} P_{Ti}dt, \tag{29}$$

and

$$P_{Ti} = F_{Ti}v_i \Leftrightarrow$$
$$\Leftrightarrow P_{Ti} =$$
$$\left(ma + \frac{1}{2}\rho C_d A_f v_i^2 + mg(C_{rr}\cos(\phi) + \sin(\phi))\right)v_i \tag{30}$$

and recalling (20) it is possible to compute E_{Ti}:

$$E_{Ti} = ma\Delta t_i\left(a\frac{\Delta t_i}{2} + v_i\right) + \frac{\beta}{4a}\left((a\Delta t_i + v_i)^4 - v_i^4\right)$$
$$+ \gamma\Delta t_i\left(a\frac{\Delta t_i}{2} + v_i\right) \tag{31}$$

where β and γ are respectively

$$\beta = \frac{1}{2}\rho C_d A_f$$
$$\gamma = mg(C_{rr}\cos(\phi) + \sin(\phi)). \tag{32}$$

In the case that there is no regenerative braking,

$$E_{T_i} = \max\{0, E_{T_i}\}. \tag{33}$$

2) Optimisation Problem Formulation: The current speed profile is found by solving the following optimisation problem:

$$\underset{v \in \mathbb{R}^N}{\text{minimize}} \quad \sum_{i=0}^{N} E_{T_i} + \alpha|t_{max} - \sum_{n=0}^{N-1}\Delta t_i| \tag{34a}$$
$$\text{subject to}$$
$$v_0 = v_{init}, \tag{34b}$$
$$v_N = v_{final}, \tag{34c}$$
$$|a_i| - a_{max} \leq 0, \quad i = 0, \ldots, N-1. \tag{34d}$$

In optimisation problem (34a) - (34d), E_{T_i} is computed using the solution proposed in subsubsection III-B1. Equations (34b) and (34c) provide the initial and final speeds and (34d) guarantees that the acceleration computed does not exceed the maximum acceleration available.

The optimisation problem's cost function minimises both the energy required to complete the path and the difference between the target time and the time elapsed, in which α is a weighting constant translating the relative importance of one term of the cost function in relation to the other. The reason for having this latter term in the cost function rather than in the constraints is to smooth the time constraint. Would it be a constraint, it could happen that the time needed to complete the course would exceed the time available, which would result in an infeasible solution. By having this term in the

cost function, the optimisation problem becomes more lenient, and the importance of completing the course in a given time frame can be adjusted by increasing the value of α. Please note that t_{max} is computed with the help of the equations of linear motion.

IV. Results and Experiments

The following sections seek to compare the performances of three different controllers: the designed MPC controller, the Pure Pursuit Controller, and the Front Wheel Position Based Feedback Controller, henceforth known as the Stanley controller. Since the aforementioned controllers do not contemplate the control of the vehicle's speed, a simple proportional controller was implemented to have the vehicle follow the speed profile as close as possible. The VIENA model, presented in Section II will be used as the vehicle model.

In order to measure the performance of the aforementioned controllers, a set of metrics must be chosen. To evaluate how closely the controller is capable of making the vehicle follow the reference path, the euclidean distance between the current vehicle position and the closest path point will be used. To quantify the error between the reference speed and the vehicle's speed, the absolute value of the error was chosen as a metric. To further analyse the controller's performance, a small statistical characterisation of the errors is performed, namely its maximum, minimum and average values are presented.

Since the focus of the speed profile optimisation is the minimisation of the energy spent, the energy spent during the path-following process will also serve has a performance metric. It is also important that the vehicle completes the track in the allotted time and thus the elapsed time during the path-following process in comparison with the predicted time will also serve as a performance metric. One final metric that must be accounted for is the smoothness of the steer command. It is desirable that there are not abrupt changes in it as to provide a safe and comfortable driving experience. To evaluate this, the moving standard deviation of the steer command was used.

A. Path-Following Experiment

The objective of this first experience is to provide a grasp of the overall performance of the designed MPC controller when compared against the Pure Pursuit and Stanley Controllers.

The results for the first track can be found in Figure 6 and on Tables I and II. The analysis of Figures 6(a) and 6(c) and of Table II highlights the fact that, when it comes to the path-following problem, the proposed MPC controller is at least on par with the solutions found by the Pure Pursuit and the Stanley controllers, finding a solution in which the mean position error is the smallest. It is worth noting the interesting capability of the MPC controller of anticipating the turns along the path. This allows the MPC controller to lead VIENA to make turns on the inside whereas the Pure Pursuit and Stanley controllers are only capable of making turns on the outside. Unlike the MPC controller, the Pure Pursuit and Stanley controllers only look at one point on the path and are thus incapable of making decisions based on anything but that point.

Figure 6(e) shows that the steer signal computed by the proposed MPC controller is much smoother than its counterparts, highlighting the importance of the R_d weight matrix.

Figures 6(b) and 6(d) and Table I highlight the major limitation of the proposed MPC controller, its difficulty of following the reference speed profile, a problem which can have severe implications on the energy needed to complete the track. In this matter, the proportional controller used in the Pure Pursuit and Stanley controllers experiments is capable of computing an acceleration command that leads VIENA's speed much closer to its desired values.

TABLE I
Energy and Time metrics for the Generic Path.

		Target Values	Real Values	$\epsilon_r[\%]$
MPC	E_T [kWh]	0.5058	0.5372	6.208
	t [s]	44.67	42.6	4.639
Pure Pursuit	E_T [kWh]	0.5416	0.5378	0.6908
	$t[s]$	43.67	42.80	1.998
Stanley	E_T [kWh]	0.5395	0.5363	0.5996
	t [s]	43.67	42.8	1.998

TABLE II
Error Metrics for the Generic Path

	e_{pos}^{max} [m]	e_{pos}^{min} [m]	$\bar{e}_{pos}$ [m]
MPC	1.9341	0.011	0.8401
Pure Pursuit	1.975	0.0057	0.9269
Stanley	1.774	0.011	0.9041
	e_v^{max} [m]	e_v^{min} [m/s]	$\bar{e}_v[m/s]$
MPC	1	0.0014	0.1794
Pure Pursuit	1	0.0020	0.1041
Stanley	1.001	0.0157	0.1423

B. Sharp Turn Path

The results for this path-following experiment are presented in Figure 7 and in Tables III and IV. This path has the peculiarity of having a very sharp turn, and it is an useful case study to evaluate the performance of the controllers in the presence of a fringe case such as this one.

As it can be seen in Figure 7(a) and Table IV, while Stanley and the MPC controller are capable of dealing

with the sharp turn in a satisfactory manner, the Pure Pursuit controller not only strays rather far away from the given reference path but also displays an oscillatory behaviour when trying to realign VIENA with it. These results are further reinforced in Table IV in which it is visible that proposed MPC controller is capable of providing the solution with both smallest maximum deviation from the path and the smallest average position error. It is worthy to point where the steering signal changes in a more abrupt manner is the sharp turn location, which was expected. Figure 7(f) validates this behaviour. While the Pure Pursuit controller exhibits the smoothest steering signal, it strays to far from the path to be a considered a viable option to solve problems of this nature.

In what concerns the reference speed tracking however, the proposed MPC controller has severe difficulties in keeping VIENA's speed close to the desired values, as it can be seen in Figures 7(b) and 7(d). This has a severe implication on the energy spent while trekking through this path, as it can be seen in Table III.

TABLE III
ENERGY AND TIME METRICS FOR THE SHARP TURN PATH.

		Target Values	Real Values	$\epsilon_r[\%]$
MPC	E_T [kWh]	1.488	1.708	14.80
	t [s]	60.42	62.6	3.617
Pure Pursuit	E_T [kWh]	1.692	1.690	0.1164
	t [s]	59.42	66	11.08
Stanley	E_T [kWh]	1.620	1.618	0.1149
	t [s]	59.42	63.4	6.707

TABLE IV
ERROR METRICS FOR THE SHARP TURN PATH.

	e_{pos}^{max} [m]	e_{pos}^{min} [m]	$\bar{e}_{pos}$ [m]
MPC	3.381	0.011	1.237
Pure Pursuit	11.07	0.011	1.655
Stanley	3.992	0.011	1.285
	e_v^{max} [m]	e_v^{min} [m/s]	$\bar{e}_v[m/s]$
MPC	1.423	0.0011	0.4444
Pure Pursuit	1	0.0041	0.0995
Stanley	1.162	0.0067	0.1542

V. CONCLUSION AND FUTURE WORK

In this work a Model Predictive Controller that solves the path-following problem with the least energy required was proposed. Along with the controller proposed, a model of an electric vehicle was also developed. This model characterises not only the dynamics of the vehicle's movement, but also the dynamics of its induction motor, providing an internal control technique that drives its speed to the desired reference.

As it was shown in Section IV, the proposed controller has several advantages over the well established Pure Pursuit and Front Wheel Based Feedback controllers. Not only is it the controller that consistently produces the smallest average distance to the path, it is also the one that is capable of doing so with the smoothest steering signal. Its capacity of computing simultaneously a steering and acceleration signal makes it capable of reducing VIENA's speed to keep it close to the provided reference path and its predictive capabilities makes it possible to have the vehicle perform turns on the inside, rather than on the outside.

In future works, the fine-tuning process of the weight matrices could be improved. A machine learning approach could provide better overall results. Moreover, focusing on computing a more realistic reference speed profile that takes into account the path's characteristics, such as its curvature, could greatly improve the speed tracking error.

Another aspect that suggests further study is the implementation of the proposed controller that decreases the computation time of the vehicle control signals. As it stands, the current MPC implementation takes around 0.3 seconds to compute the desired commands. As noted before the proposed implementation was done in Matlab. For a real-time implementation, a far faster implementation needs to be considered. It is possible that implementing the proposed controller in a language such as C++ will decrease the computation time significantly.

The further development of VIENA's model is also an important step to evaluate the possibility of a real-time implementation. The development of models that contemplate sensor dynamics is needed to evaluate the controller's robustness to uncertainty and noise.

REFERENCES

[1] E. R. S. Observatory. Annual accident report 2018. [Online]. Available: https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/pdf/statistics/dacota/asr2018.pdf

[2] E. Brevio and S. Meder. 2018 road safety statistics: what is behind the figures? [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/MEMO_19_1990

[3] NHTSA. Traffic safety facts. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812456

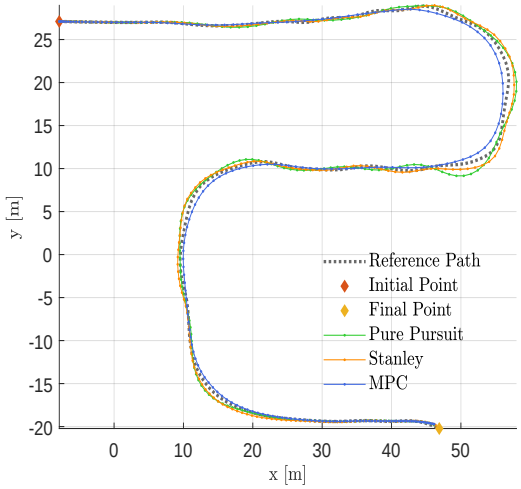
[4] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7490340

[5] B. Tibério, "An online filter study for inertial properties estimation based on low-cost sensors." 2017. [Online]. Available: https://www.semanticscholar.org/paper/An-online-filter-study-for-inertial-properties-on-Almeida/4caaf8b97c040a88928a35ea721ace33a7ce0802

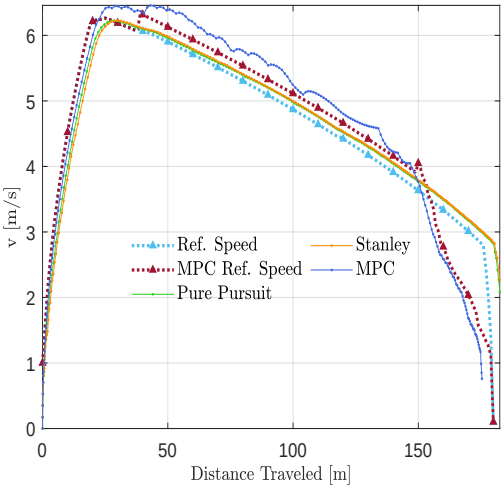
[6] J. Fernandes, F. Silva, and B. Tibério, "Electrical drives and electrical vehicles - *dq* model for the induction motor," Lecture Notes, 2020.

[7] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017. [Online]. Available: https://dl.acm.org/doi/book/10.5555/3164811

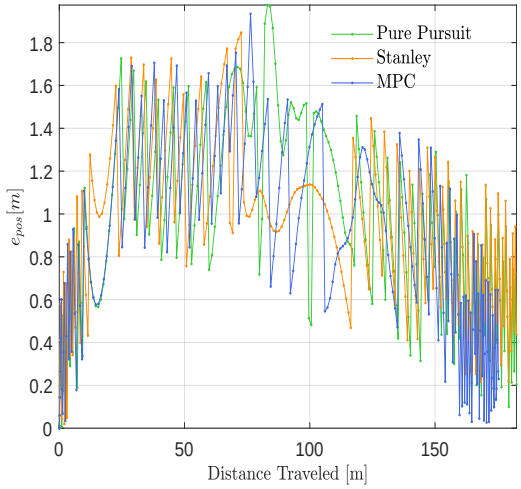
[8] A. Sakai. Path tracking. [Online]. Available: https://pythonrobotics.readthedocs.io/en/latest/modules/path_tracking.html



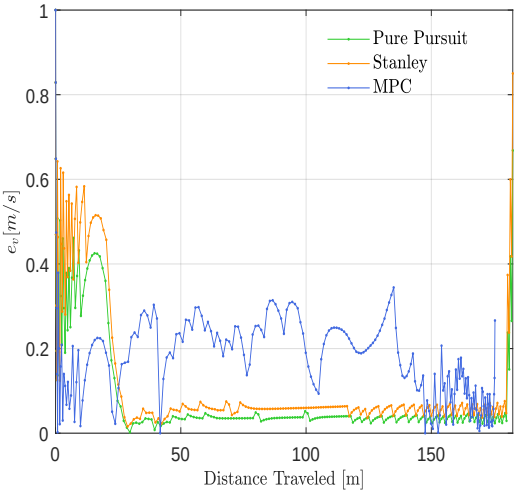
(a) Reference path and Vehicle Position



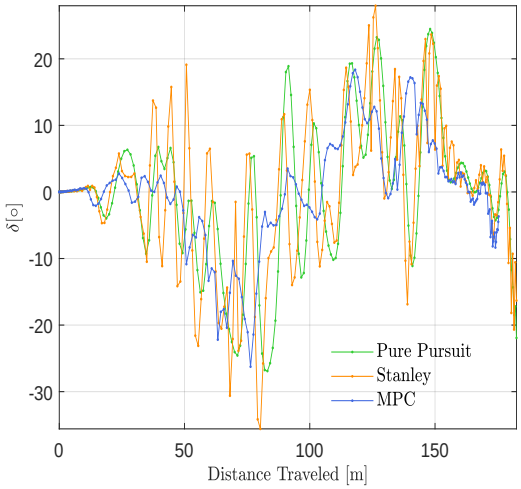
(b) Speed profile



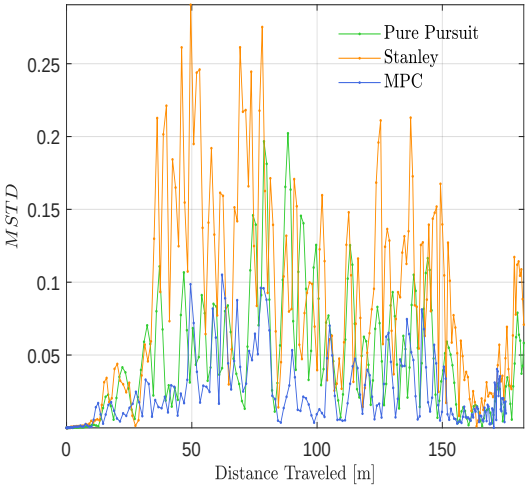
(c) Distance to Path



(d) Speed Tracking Error

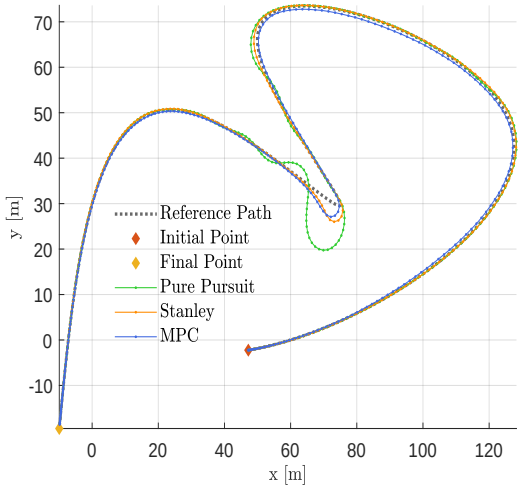


(e) Steering Command

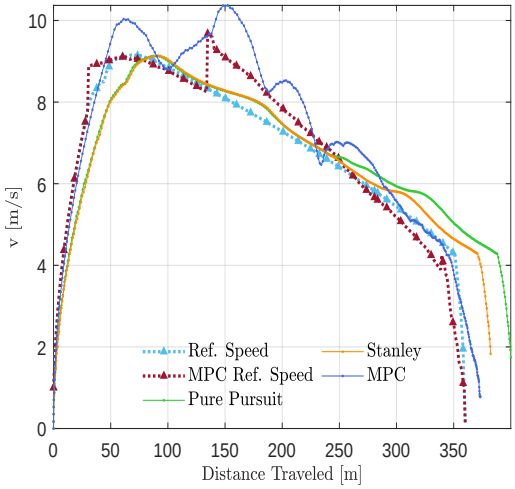


(f) Steering Moving Standard Deviation

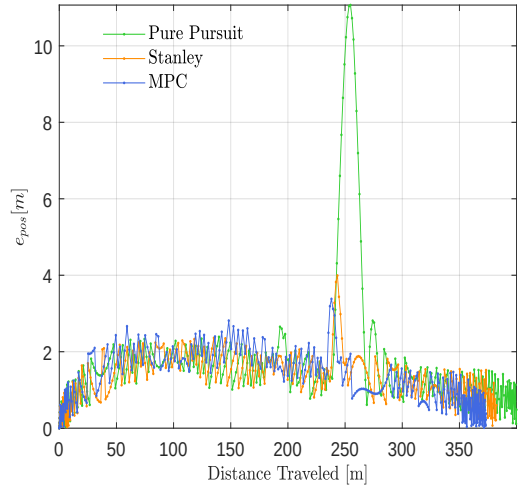
Fig. 6. Results for the Controller Fine-tuning Path Experiment.



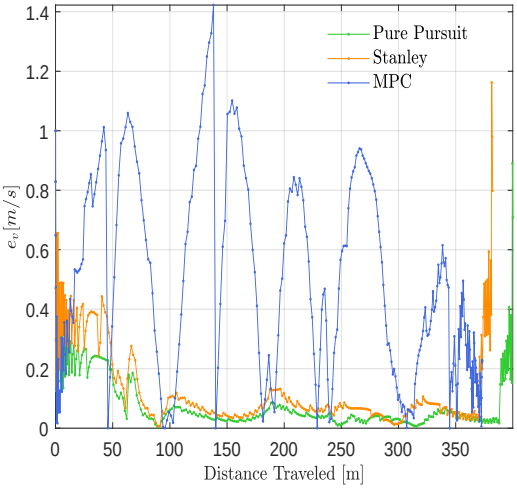
(a) Reference path and Vehicle Position



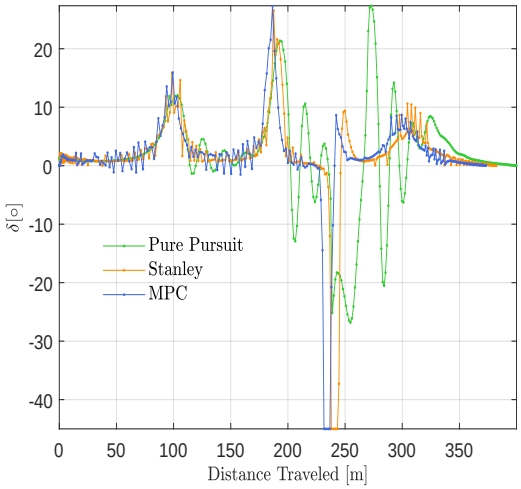
(b) Speed profile



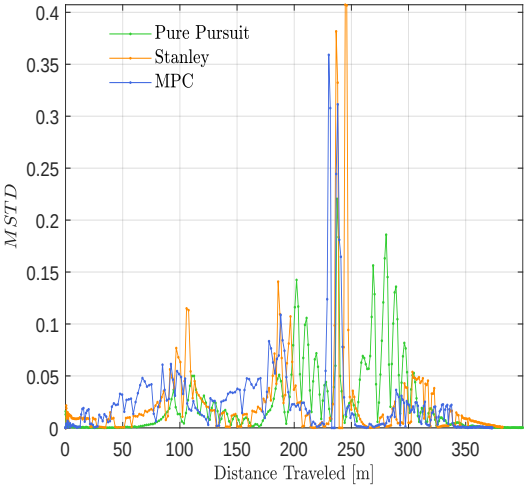
(c) Distance to Path



(d) Speed Tracking Error



(e) Steering Command



(f) Steering Moving Standard Deviation

Fig. 7. Results for the Sharp Turn Path Experiment.