Global Path Planning for Competitive Robotic Cars*

Tizar Rizano¹, Daniele Fontanelli¹, Luigi Palopoli¹, Lucia Pallottino² and Paolo Salaris²

Abstract—In this paper, we consider the optimal motion planning problem for an autonomous race car. A competitive autonomous car must acquire environmental and opponent information to compute, in real time, the minimum time collision free path and the low level control to track the chosen path. To cope with these requirements, we first solve the problem for a car running in isolation considering the optimal sequence of manoeuvres to approach bends and straight stretches of track. We then propose a discrete abstraction to derive a problem of graph optimisation that has a very efficient, albeit suboptimal, solution. In this context, an overtake manoeuvre against a slower car will be obtained excluding from the path the arcs that could potentially generate a collision. Finally, a control algorithm is used to ensure that the car always remains close to the planned path.

I. INTRODUCTION

The problem addressed in this paper can be summarised in the following terms: given an autonomous car–like vehicle that runs on a known track sharing it with other vehicles, plan a trajectory that allows the vehicle to complete a given number of laps on the track in minimum time while avoiding collisions with other vehicles. Path planning in a competition track is very different from path planning in an urban or extra urban road [1]. First, the environment is strongly structured and well known upfront. Second, the track is accessible only to a small and predetermined number of cars. Third, all the cars in the track have very similar spatial footprint and dynamic characteristics. This reduces the amount of real–time data that the planner requires. On the other hand, the high speed of the car requires the information acquisition, the optimal planning with collision avoidance and the control phases to be executed within tight real–time constraints.

a) Related Work: Autonomous driving for cars is a very popular theme amongst a multidisciplinary research community [2], [3]. In particular, trajectory optimisation for race car simulators has received a constant attention, with production of research papers and of patents, see e.g. [4], [5], [6].

In the robotic community, the path of shortest length has been investigated by Dubins[7], by Reeds&Sheep [8] and by Alexander et al. [9] for car–like vehicles, and by other authors for differential drive robots with limited field of view sensors [10], [11]. The minimum wheel rotation for differential drive robots has been investigated by Lavalle et al. [12]. The minimum time trajectories have been investigated for differential drive [13], omnidirectional [14] and mobile robots with trailers using bounded controls [15] and robots with independently driven wheels [16].

The approach we advocate in this paper builds on the extremal manoeuvres, i.e. that verify the Pontryagin Minimum Principle necessary conditions for optimality [17], to steer the car from one configuration to another in minimum time. Geometric constraints imposed by the boundaries of the track must be taken into account, as in the work of Velenis et al. [18], who analyse different strategies for local path planning (making a bend) based on the dynamic behaviour of the suspension and of the tyre. Among the possible choices existing in literature for real car dynamic models, e.g. [19], [20], we adopt the kinematic model described in [21].

Finally, a very important inspiration for this work is the technique usually referred to as “discrete abstraction” [22], [23], [24], [25], [26], [27], whereby a system with dense state is translated into a discrete system (essentially a state machine) to simplify planning and verification of properties.

b) Overview on the approach: The approach of this paper is pictorially represented in Figure 1, where squares denote information, ovals denote steps and squares with rounded corners denote results of the steps. Part of the steps are carried out offline and part are carried out on-line. The backbone of our approach is an abstraction that allows us to reformulate the path planning problem in a discrete graph–based setting. Such abstraction is constructed starting from way lines that are distributed on the track. By quantising the
possible position and velocities of the car when it intersects the way line, we partition the state space into a finite number of cells, which are associated to nodes in a graph. Two nodes are connected if they are located on adjacent way lines and if there exist a feasible sequence of manoeuvres that steer the vehicle from the initial to the final cell. The sequence of manoeuvres that solves this local path planning problem is discussed in a different paper [28]. Each arc has a cost given by the time required by the sequence of manoeuvres that executes the transition between the two cells. If the car runs in isolation, a suboptimal motion plan is constructed by solving on-line the shortest path problem on the graph. The control law guarantees the correct execution of the planned path: if the plan includes a transition from cell A to cell B, it ensures that if the initial state is within cell A then its final state will be within cell B. As a consequence, the control law determines the minimum resolution of the quantisation.

The presence of multiple vehicles is dealt with by constructing a graph for each vehicle on the track (the graph depends on the kinematic parameters of the vehicle). We assume each vehicle to be neither cooperative (i.e., does not facilitate the overtake) nor competitive (i.e., do not hinder the overtake on purpose): it simply keeps moving along its optimal path. When a slower vehicle is in a leading position, assume each vehicle to be neither cooperative (i.e., does not facilitate the overtake) nor competitive (i.e., do not hinder the overtake on purpose): it simply keeps moving along its optimal path. When a slower vehicle is in a leading position, it overtake the faster vehicle (i.e., if the faster vehicle is too slow to overtake) nor competitive (i.e., do not hinder the overtake on purpose): it simply keeps moving along its optimal path. When a slower vehicle is in a leading position, it overtake the faster vehicle (i.e., if the faster vehicle is too slow to overtake).

The motion plan algorithm is modified to make sure that the shortest path selected on the graph does not violate such temporal constraints.

The paper is organised as follows. In Section II we introduce the most important concepts the paper revolves around and propose a formal statement for the problem. In Section III we discuss our graph based discrete abstraction of the problem. In Section IV, we discuss the construction of arc between the nodes summarising the results of the local path planning problem [28]. In Section V, we support our technique by a large set of simulation results. Finally, in Section VI, we offer our conclusions and announce future directions of work.

II. PROBLEM DEFINITION

In this section we introduce the race car model and the geometric model of the track. Then, we formalise the global motion planning problem addressed in the next sections.

A. Car Model

A realistic model of the car, comprising the kinematics and the most important dynamic effects has been proposed by Hoffman et al. [21] and used in our work. Albeit this dynamic model comprises tires slip, steering servo motor effects and all the major dynamic interactions involved, it is not explicitly conceived for racing cars that usually suffer of relevant drafting effects or loading differences during accelerations or braking manoeuvres. Nevertheless, the presence of the aerodynamical drag forces, that have been additionally considered in the local motion planning analysis [28], makes the model sufficiently realistic for our purposes. As shown by Hoffman et al., the motion planning problem can be approached restricting to the kinematics of the vehicle as far as an appropriate controller is applied to the dynamics. For this reason, we will simply focus here on the kinematics.

Let \( \{W\} = \{O_w, X_w, Y_w, Z_w\} \) be a right-handed fixed reference frame (see Figure 2). The configuration of the vehicle is described by \( \xi(t) = (x(t), y(t), \theta(t), \varphi(t)) \), where \( p(t) = (x(t), y(t)) \) is the position in \( \{W\} \) of the rear–wheel axis midpoint, \( \theta(t) \) is the orientation of the vehicle with respect to the \( X_w \) axis and \( \varphi(t) \) is the steering angle of the front wheel with respect to the vehicle, see Figure 2. Let \( v(t) \) be the linear velocity of rear–wheel axis midpoint. Choosing \( \varphi \) and the traction acceleration \( a \) as control inputs, the car–like model adopted in this work is

\[
\dot{q} = \begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta} \\
\dot{\varphi} \\
\dot{v}
\end{bmatrix} = \begin{bmatrix}
v \cos \theta \\
v \sin \theta \\
tan \varphi \frac{L}{T} \\
\tan \varphi \frac{R}{L} \\
\frac{v^2}{a - c_a v^i}
\end{bmatrix},
\]

where the term

\[
F_{aero} = m c_a v^i
\]

accounts for the aerodynamic force, \( m \) is the total mass of the vehicle, \( c_a = m c_a \) is the aerodynamic drag coefficient and \( i = 1 \) for air laminar motion, while \( i = 2 \) for turbulent motion. The parameter \( L \) is the distance between the wheel axes. The control inputs are constrained in the sets \( \varphi \in [-\varphi, \varphi] \) and \( \theta \in [\alpha, \pi] \) (with \( \alpha < 0 < \pi \), i.e. maximum braking and acceleration applicable actions), respectively. Since we are not interested in parking or docking manoeuvres we assume the velocity \( v \in [v, \pi] \) with \( v > 0 \). For practical reasons, the maximum velocity is assumed to be less than \( \pi \). For the laminar regime, and can be identified numerically for the turbulent regime.

For bending paths, the constraint imposed by the maximum lateral acceleration \( a_i \) of the vehicle is additionally considered. In fact, given a planned curve to travel, we have

\[
v^2 \leq a_i R
\]
where \( R = \frac{L}{\cos\theta} \). Notice that both \( a_i \) and \( b \) are function of the aerodynamic load as well as the tires grip (which depends on the ground characteristics, e.g. dry or wet asphalt, off road, etc.), and provides a constraint depending on the state variable \( v \) and the control variable \( \varphi \).

### B. Track Description

The track contains straight sectors and turn sectors. A **straight sector** contains a straight line and is modelled as two parallel lines \((B_r \text{ and } B_l)\) at distance \( W \), while the **turn sector** contains a left or a right bend consisting of a sequence of two (possibly of infinitesimal length) parallel lines \( L_{in} \) (the approach to the bend), two concentric arcs of circle (the bend) and other two (possibly of infinitesimal length) parallel lines \( L_{out} \) (the exit of the bend).

Referring to Figure 3, the bend boundaries \((B_r \text{ and } B_l)\) are characterised by the centres \( C_b \) and two radii \( R_b \). In particular, \( R_b^o \) refers to the inner boundary, while \( R_b^o \) to the outer where \( R_b^o - R_b^o = W \). The angle \( \gamma \) is the characteristic curve angle, i.e. the angle between the two straight line segments of each boundary. The angles \( \alpha \) and \( \pi - \beta \) are the orientations of \( L_{in} \) and \( L_{out} \) w.r.t. the orthogonal of the \( \gamma \) bisector, respectively. The entire track is hence a sequence of sectors \( S_i \) where each sector can be of straight or turn type. Furthermore, a sector \( S_j \) is characterised by two segments (way-lines) of width \( W \) orthogonal to the sector borders. The way-line \( s_{il} \) is the segment at the beginning of sector \( S_i \) and is considered as the starting line of \( S_i \) and the final way-line of \( S_{i-1} \) while the orthogonal segment \( a_{il} \) at the end of sector \( S_i \) is considered as starting line of \( S_{i+1} \) and final line of \( S_i \), i.e. \( s_{i+1} = a_{il} \), see Figure 3. Since in each turn sector the track bend is preceded and followed by a straight stretch of track we may assume that the car will always cross the starting and final lines \( s_{il} \) and \( a_{il} \) perpendicularly, i.e. \( \theta(t_i) = \theta(t_{i+1}) = 0 \). Finally, considering a circuit of \( n_s \) sectors, we assume \( S_{n_s+1} = S_1 \). In other words, the track re-starts over at the end of each lap.

### C. Reference frames

Let \( \{T\} = \{O_T, X_T, Y_T, Z_T\} \) be the *track reference frame* with \( O_T \) a point on \( B_r \), \( X_T \) tangent to \( B_r \) and \( Y_T \) pointing towards the left boundary \( B_l \) (see Figure 3). This frame is a Frenet frame attached to the right boundary of the track. Consider the *manoeuvre initial reference frame* \( \{I_m\} \) as \( \{T\} \) placed on the first straight line of \( B_r \) and the *manoeuvre final reference frame* \( \{E_m\} \) as \( \{T\} \) placed on the second straight line of \( B_r \). Similarly, \( \{I\} \) and \( \{E\} \) represent the *track initial reference frame* and the *track final reference frame*, respectively. Notice that, in general, \( \{I\} \equiv \{E\} \).

A point \( T_{pr} \) on \( X_T \) has a corresponding point on \( B_l \) given by \( T_{pr} = T_{pr} + [0, W]^T \). In particular, any point inside the track at the same distance of \( T_{pr} \) from the curve can be expressed as \( T_{p} = T_{pr} + k[0, W]^T \), where \( 0 \leq k \leq 1 \). We will denote by \( \mathcal{P} \) the set of all points lying inside the track, for which there exist a Frenet frame in which their \( X \) coordinate is 0 and their \( Y \) coordinate is smaller than \( W \): \( \mathcal{P} = \{v|3(T) \text{ s.t. } T v = k[0, W]^T, 0 \leq k \leq 1\} \).

### D. The optimal control problem

The goal of this paper is to find the sequence of manoeuvres that allow the car to complete a generic number of laps in minimum time. Hence, the cost function to be minimised over the track is

\[
\int_0^T L(q, a, \varphi) \, dt,
\]

where \( L(q, a, \varphi) = 1 \).

Given a configuration \( \xi = [x, y, \theta, \varphi], p_r = [x, y] \) denotes the position of the midpoint of the rear axle, while the midpoint of the front axle is \( p_f = (x + L \cos \theta, y + L \sin \theta) \). Let \( \Sigma \) be the sequence of sectors to be traversed. In other words, \( \Sigma \) is the region of configurations such that the vehicle is inside the track. This happens if both \( p_f \) and \( p_r \) are inside the track: \( \Sigma = \{\xi|p_f, p_r \in \mathcal{P}\} \). Moreover, at time \( t_i \) the configuration is supposed to lie on \( s_{il} = [\xi|p_f = k[0, W]^T, 0 \leq k \leq 1, p \in \mathcal{P}] \), i.e. the starting region of sector \( S_i \). If the circuit has \( n_s \) sectors and the number of laps is \( n_l \), \( \Sigma \) comprises \( n_l \) ordered sequences of \( n_s \) sectors. We can now state the:

**Problem 1: Track Optimal Problem**

\[
\min_{a(t), \varphi(t)} \sum_{i=0}^{n_l n_s - 1} \int_{t_i}^{t_{i+1}} dt, \text{ subject to }
\]

- \( q(t) \) solution of (1),
- \( \xi(t) \in \Sigma, \xi(t_i) \in s_{il}, \forall i = 1, \ldots, n_l n_s \),
- \( v(t) \in [\underline{v}, \bar{v}] \),
- \( v^2(t) \tan \varphi \leq a_L \),
- \( a(t) \in [\underline{a}, \bar{a}] \),
- \( \varphi(t) \in [\underline{\varphi}, \bar{\varphi}] \).

This problem refers to a single car and it requires that the kinematic model, and the different geometric and dynamic constraints are respected. In particular, the constraints for
\(\xi(t)\) refer to the initial and final configurations. More precisely, at time \(t_1 = 0\) the initial configuration is supposed to lie on \(sl_1 = \{\xi[p_f] = k[0, W]^T, 0 \leq k \leq 1, \ p \in \mathcal{P}\}\), with \(sl_1 = sl_{n_s}\) while \(sl_{2n_s} = \{\xi[p_f] = k[0, W]^T, 0 \leq k \leq 1, \ p \in \mathcal{P}\}\). An additional requirement (addressed in the final part of the paper) is that no collision happen with the other vehicles, assuming that they also adopt a time optimal strategy.

III. Graph-based Circuit Representation

A solution to Problem 1 can be found using nonlinear optimal control theory or MPC-like tools. However, a complete characterisation of the solution as well as the high computational cost required for such solutions (in particular, if the algorithm has to be executed on an autonomous robotic vehicle endowed with limited computing resources) leads to a more manageable solution. In this paper, we decide to represent the track using a discrete abstraction of its possible configurations. More precisely, a possible representation of the track can be given in terms of a graph.

Consider a discretisation of dimension \(d_w\) of the width \(W\) of the track. The \(d_w\) points laying on the orthogonal lines \(sl_i\), the starting points of \(S_i\) and final points of \(S_{i-1}\), while the points laying on the orthogonal lines \(al_i = sl_{i+1}\) are considered as starting points of \(S_{i+1}\) and ending points of \(S_i\). Considering a circuit of \(n_s\) sectors, the total number of nodes in the graph is \((n_s + 1)d_w\), where we assume \(S_{n_s+1} = S_1\). In other words, to close the circuit, the nodes of \(S_1\) are considered twice: one as initial nodes of \(S_1\) and one as final nodes of \(S_{n_s}\). The arcs of the graph represent the sectors and hence any node on \(sl_i\) is connected through an arc to any node on \(sl_{i+1}\) or \(sl_{i-1}\). Therefore, the graph consists of \(n_sd_w^2\) arcs. A cost \(c_{i,j}\) equal to the minimum time required to go from a node \(i\) to \(j\) is associated to arc \((i, j)\). Notice that the combination of minimum time sectors does not result in a minimum time lap. Nevertheless, it is necessary to reach \(al_i\) from \(sl_i\) along \(S_i\) in the minimum time and with the maximum velocity in \(al_i\).

Therefore, for a sector \(S_i\), the initial and final configurations assumed by the vehicle are constrained on two lines, defined by \(sl_i = \{\xi[p_f] = k[0, W]^T, 0 \leq k \leq 1, \ p \in \mathcal{P}\}\) and \(al_i = \{\xi[p_f] = k[0, W]^T, 0 \leq k \leq 1, \ p \in \mathcal{P}\}\), respectively. Hence, in place of Problem 1 the following set of optimal control problems can be defined

**Problem 2: Optimal Control Problem**

\[
\min_{a(t), \varphi(t)} t_{i+1} - t_i, \text{ subject to } \\
q(t) \text{ solution of (1), } \\
\xi(t) \in \Sigma, \xi(t_i) \in sl_i, \xi(t_{i+1}) \in al_i \\
v^s(t_i) = v_i^1, v(t_{i+1}) = v_i^F \\
v(t) \in [\underline{v}, \overline{v}] \\
v^2(t) \tan \varphi \leq a_i L \\
a(t) \in [\underline{a}, \overline{a}] \\
\varphi(t) \in [-\overline{\varphi}, \overline{\varphi}].
\]

It is worth noting that, with respect to Problem 1, the optimal solution in each sector must be obtained given the speeds \(v_i^1\) and \(v_i^F\) at the beginning and at the end of the sector \(S_i\). Based on the principle of optimality, those constraints are introduced to obtain, as shown below, the concatenation of optimal solutions of Problem 2 for each sector that is the optimal solution of Problem 1.

To solve this problem, a discretisation of dimension \(d_v\) of the speed space \(v\) is also provided, so that each point on \(sl_i\) can be crossed at \(d_v\) different speed values. Hence, \(d_wd_v\) nodes are associated to any initial segment \(sl_i, \forall i\). With \(n_s\) sectors, the total number of nodes in the graph is now \((n_s + 1)d_wd_v\), while the number of arcs turns to \(n_s d_w^2 d_v^2\). More formally, a node \(k\) is represented by a triplet \(k = (S_i, p_k, v_k)\) where \(S_i\) is the sector, \(p_k\) is one of the \(d_w\) position of the point represented by \(k\) on \(sl_k\), and \(v_k\) is one of the \(d_v\) speeds pertaining to a point \(p_k\). Given nodes \(i\) and \(j\) the arc \((i, j)\) belongs to the graph if and only if \(S_j = S_{i+1}\). The cost \(c_{i,j}\) associated to arc \((i, j)\) is equal to the minimum time required to go from a node \(i\) to \(j\) with the corresponding speeds. Hence, the cost \(c_{i,j}\) equal to the solution of the minimum time Problem 2 with \(v(0) = v_i, v(T) = v_j\) from \(p_i\) to \(p_j\).

In order to apply standard shortest paths algorithms such as Dijkstra [29] it is necessary to introduce two nodes. An initial node \(I\) and a final node \(F\) and all arcs \((I, i)\) and \((j, F)\) where \(S_i = S_1\) and \(S_j = S_{n_s+1}\). The associated costs are null, i.e. \(c_{I,i} = c_{j,F} = 0\). With the introduction of such nodes algorithm such as Dijkstra provide shortest path from any node of the graph to \(F\). In particular, from any point on \(S_1\) to \(S_{n_s+1}\). This way, a suboptimal solution of Problem 1 is obtained.

Determining the minimum path from the nodes associated to the circuit starting line to the same set of nodes considered as nodes of the arrival line, the minimum time lap can be determined with the associated sequence of manoeuvres described in previous section. It is worth noting that with this approach the best trajectory for the qualifying lap is determined. If we are interested in 2 or more minimum time laps the graph must be extended duplicating the nodes and the arcs. Indeed, a graph associated to two laps on the circuit must be taken into account considering \(S_i\) of the first lap different from \(S_{n_s+1}\) of the second lap. A graph with \(2n_sd_wd_v\) nodes and \(2(n_s + 1)d_w^2 d_v^2\) arcs is hence considered. The same construction of nodes \(I\) and \(F\) with associated arcs and costs can be followed and the optimal trajectory for a generic lap of the race can be found.

The discretisation of the width \(W\) and of the speed \(v\) will obviously provide a suboptimal solution. However, a finer quantisation provides a better solution but with the drawback of having a huge graph and hence a higher computational costs.

A. Avoiding obstacles

The graph abstraction can easily be applied to account for the presence of other (slower) cars in front of the vehicle that are neither cooperative nor competitive. This is done working with two graphs (one for each vehicle). The first step is to create a relation between the arcs of the two graphs. A pair of arcs belongs to the relation if it is possible to have a collision when the arcs are taken with a wrong timing. Suppose that
vehicle $A$ follows and vehicle $B$ leads, and assume that
the couples of arcs $(a_A, b_A)$ and $(a_B, b_B)$ (belonging to
the graph of $A$ and $B$ respectively) potentially lead to a
collision. Using simple kinematic considerations (which we
do not detail for the sake of brevity) it is possible to find a
minimum interarrival time $τ(a_A, b_A)→(a_B, b_B)$ such that if $A$
enters the arc $(a_A, b_A)$ after a time $τ(a_A, b_A)→(a_B, b_B)$ elapsed
since $B$ entered $(a_B, b_B)$, the collision is avoided.

The algorithm for path planning can be modified as follows.
At the beginning $A$ detects the position of $B$ and assumes that
it will use its graph for a minimum time path planning. As a consequence it knows the position of $B$ in its
graph for any time in the future. As a first step $A$ finds the
shortest path using its graph and annotates it with the time
each node is reached. If the obtained path contains an arc
$(a_A, b_A)$ that is in relation with another arc $(a_B, b_B)$ that $B$
uses, the algorithm checks if the minimum interarrival time is
respected. If not the arc $(a_A, b_A)$ is removed from the
graph and the Djikstra algorithm is repeated on the updated
graph. These steps are repeated until a “clear” path is found.
The algorithm also checks when $A$ becomes the leader. From
that point on the arcs that are possibly removed during the
algorithm execution are reinserted and $A$ can use its entire
d graph.

IV. TOWARDS AN OPTIMAL SOLUTION

Instrumental to the construction of the graph is the solution of
the local planning problem: how to steer the car from a
configuration $ξ(t₁)$ where each configuration is defined by
position and velocity. This problem has been addressed in a
different paper [28]. We report here the essential results for
the sake of completeness.

The solution of the local planning problem requires: 1) iden-
tification of an alphabet of elementary maneuvers (extrem-
tals) arising from the application of Pontryagin Minimum
Principle (see e.g. [17] and [30] for the constraints on the
state and control variables); 2) identification of the optimal
concatenation of extremals; 3) computation of the optimal
free parameters.

Computation of the extremals

The application of the Pontryagin Minimum Principle
to the kinematic–model disregarding the constraints on the
configurations $ξ$ that impose that the vehicle is on the track
leads to the following set of extremals:

1) Straight line $S$, travelled with any velocity profile,
compatible with the maximum and minimum acceler-
ations, i.e. $v$ and $a$, respectively;
2) Circular curve $C_v$ travelled with constant maximum
velocity $v_x$; the radius is fixed to the maximum value
$v$ that is compatible with the constraint on the lateral
acceleration: $v = \sqrt{v_x^2} / a_l$, where $a_l$ is the maximum
lateral acceleration;
3) Circular curve $C_\perp$ travelled at (possibly time–varying)
velocity $v_x \leq v_x \leq \sqrt{\frac{a_l L}{|\tan(\perp)|}} = v_x^\perp$; the radius is
fixed to the minimum possible value allowed by the
vehicle: $\perp = \frac{L}{|\tan(\perp)|};$
4) Variable radius curves $V_{\tau}$ and $V_{\underline{\tau}}$ executed with
maximum or with minimum acceleration respectively,
which always verify the relation $v_x^\perp |\tan(\perp)| = a_1 L$.

Optimal sequences of extremals

The computation of the optimal sequence of extremals is
very difficult to be addressed geometrically due to the nature
of the variable radius curves $V_{\tau}$ and $V_{\underline{\tau}}$. In the paper [28]
we identified a heuristic solution derived from geometric
considerations on simplified cases, that are:

- For straight sector:
  \[ ST_{\underline{\tau}}C_{\perp}C_{\tau}S; \]
- For a turn sector:
  \[ ST_{\underline{\tau}}C_{\perp}S[C_{\tau}, |C_{\tau}|]SC_{\tau}T_{\tau}S. \]

In this case we make use of the $T$ manoeuvres, i.e.,

\[ T_{\underline{\tau}} = C_{\tau}V_{\underline{\tau}}; \]
\[ T_{\tau} = V_{\tau}C_{\tau}. \]

In addition, we can have either a curve with maximum or
with minimum radius $|C_{\tau}/|C_{\tau}|$ of the sector to account for
the geometry of the track. Notice that each extremal can be
of zero length.

Additionally, each curved maneuver has a superscript
equal to $- +$ depending on the fact that the curve turns in
the clockwise or counterclockwise direction (thus decreasing
or increasing the value of the angle $θ$ according to the right–
hand rule).

Computation of the parameters

Each extremal is characterised by a set of parameters.
For instance a straight line $S$ is associated with its length,
initial velocity and final velocity. When the extremals are
concatenated together some of the choices become bound
by the previous extremal in the sequence. For instance, if a
straight line is followed by a bend, the initial velocity of the
bend will have to be equal to the final velocity of the straight
line. Similar constraints holds for the trajectory tangents
values in the concatenation point. Additional constraint are
obviously imposed by the initial and the final configuration.
Nevertheless some of the parameters remain open and can be
considered as decision variables in an optimisation procedure
aiming for the minimum time solution. Efficient solution
strategies for this challenging problem are still under inves-
tigation. In the simulation section below we report solutions
obtained with a combination of simulated annealing and
gradient descent method.

The optimal solution of this problem is the weight of the arc
connecting the two nodes of the graph.

V. SIMULATIONS

The simulations are performed with two cars running on
a track for 5 laps. Both cars have equal size (3.5m in
length and 1.8m wide) and same maximum and minimum
accelerations $a = 34.5m/s^2$ and $a = -20m/s^2$ respectively.
A minimum curvature radius of $15m$ is imposed on both cars
to account for their minimum turning radius.
The optimal trajectory of a vehicle is affected by the choice of parameter values (maximum velocity, maximum acceleration, length of the track, etc). Therefore, the graph differs for different set of parameter values. Figure 4.a shows the result of two independent simulations for the two cars (with their different value for the viscous friction).

Figure 4.b shows the result of a similar simulation with a reduced (half) value of $a_l$. Due to the reduced maximum lateral acceleration, both cars cannot find a trajectory where they can reach the maximum velocity along the curve.

Table I offers simulation results for different choices of quantisation parameters. Table I is related to the track shown in Figure 4 where the length of rectilinear paths are 160m and 80m while the curvature radii are 80m.

As shown in Table I, the time to complete a lap of Car 1 does not change by increasing the dimension $d_v$ of the velocity space. This is due to the fact that both cars can reach their maximum velocity and hold it until the end of the lap. However, increasing the dimension $d_w$ enables Car 1 to reduce the lap time by having a trajectory with shorter length. Conversely, the time to complete a lap of Car 2 on a dry road condition is unaffected by changing $d_w$ or $d_v$. This is due to the fact that there is only one trajectory that enables Car 1 to reach the highest feasible velocity.

### B. Obstacle Avoidance

A more interesting example is the one where we have more than one car on the track. Similarly to the previous example a graph of feasible trajectories is built for each car with the parameters given in the beginning of this section. Two optimal trajectories are produced for each graph. We did this for a track inspired to Interlagos Formula 1 track, which was divided in 14 sectors. Figure 5.(A) shows the two sets of optimal trajectories. The blue line represents the trajectory of car 1 and the red line represents the trajectory of car 2 before applying the collision avoidance algorithm described in Section III-A, while the green line represents the new optimal trajectory of car 2 with collision avoidance. The inset shows the configurations of the two cars at time $t_i$ before and after the path planning. A more detailed view is depicted in Figure 5.(B) where we took a snapshot of the cars configurations at three different times.

To tackle this problem, we have applied the path planning algorithm on the two graphs and removed edges from the graph of the car at the rear if the two cars collide along the trajectories represented by the edges. The algorithm produces two new optimal paths where the two cars do not collide as shown in Figure 5.(B).

### VI. Conclusion

In this work, we have proposed a motion planning technique that applies to robot race cars. The kinematic model adopted is taken from [21]. The use of specific dynamic models conceived for racing cars [31] is reserved for our future work. By using a discrete abstraction, we are able to generate a motion plan that optimises the completion time of the race in a short time, both when the car runs in isolation and when it has to overtake a slower car.

The leading car is not cooperative but is not competitive as well (it does not try to obstruct the overtake manoeuvre). An obvious future direction is to implement game theoretic strategies that allows the car to overtake even if the leader opposes. Another important area for future work will be on the implementation of this idea on a robotic vehicle.

From the control theoretical point of view, the authors are interested in the study of the apparent ideal line that the vehicles seem to converge to during the race.

### References


Fig. 5. (A) Trajectories of two cars, on the Interlagos-like circuit, with different viscous friction $b$ parameters before graph pruning (red) and after pruning (green), (B) Closeup of the trajectories of two cars, on the Interlagos circuit, with different viscous friction $b$ parameters before graph pruning (red) and after pruning (green).


