Trajectory Optimization using Model Scale Cars

Diogo André Fernandes da Silva
diogo.f.silva@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

May 2017

Abstract

This work proposes two Optimal Control formulations for trajectory optimization: Minimum time path tracking and minimum Time trajectory optimization. Firstly, the vehicle dynamics is introduced and is reformulated in a way that eases the optimization formulation. Secondly, the track is parametrized and a change of variables is performed to adapt the optimal control problem. To transform the problem in a large scale optimization problems direct transcription methods were employed. At last, the problem is solved exploiting nonlinear optimization methods, such as interior-point(IP), sequential quadratic programming(SQP) and Second Order Cone Programming (SOCP).
Keywords: Trajectory Optimization, Optimal Control, Motion Planning, Second Order Cone Programming applications

I. Introduction

Trajectory optimization is a specificity of the more general and commonly used term motion planning. In [1] LaValle presented motion planning as an important collision course of robotics, artificial intelligence and control theory. In robotics, motion planning is related with problems such as how to move an object from one division into another without hitting anything. The field has been maturing and nowadays is common to include uncertainties and multiple bodies. In regards of artificial intelligence, planning was typically related with the sequence of operations necessary for the transition from an initial state into a desired one. Mostly since the 1990's, there has been a growing interest within the control field to find the open-loop trajectories for non-linear systems. Motion planning is a key subject that is inherent to a great number of robotics applications, from medical applications [2, 3, 4], industrial applications [5], autonomous urban-navigation [6, 7, 8, 9] and military logistics [10]. In the 1980's and beggining of 1990's, robotics manipulators were too slow to justify their use economically. Their productivity was limited by speed, which was limited by torque actuators. Nonetheless, increasing torques was not seen as the best solution, due to the increase in cost, power consumption and inertia. The solution to justify the use of manipulators was to perform minimum time motion planning, i.e. minimize the time needed to perform a given task, subject to actuator's constraints [11]. The scientific community that study and concern motion planning and trajectory optimization, including minimum time and minimum energy problems, has three widely accepted approaches to solve this family of problems. The first method was proposed by Brobow, Dubowsky and Gibson in [11]. It is a practical method that surpassed the limited computational of that time by defining a path by a small number of control points and then using a novel algorithm, at that time, to compute the minimum time velocity profile. Next the control points position is varied in order to get the minimum time trajectory, thus it can be reckoned as an indirect method. The solution for a special family of paths was proven to be bang-bang. This method was mainly used for manipulators, but was also successfully applied to an unmanned ground vehicle (UGV) over an uneven terrain [12]. Later, the algorithm was improved to deal with manipulator dynamic singularities [13], which expanded its applicability, since it presented a method to solve non bang-bang problems. Other similar approaches were presented such as [14], which have modelled the torque constraints as quadratic instead of a mere function of velocity and position . Shiller's later work, once again exploits this approach to introduce more efficient algorithms, for offline and online motion planning, using the branch and bound method with more careful constraints classification for a car over uneven terrain and with multiple obstacles [15]. The solution provided in [16] was also innovative since it is assumed to be one order of magnitude faster than [11, 14]. Such results were achieved by an exhaustive exploration of the switching points timing, i.e., the timing determines when the robot should accelerate or decelerate, which is the most demanding task in terms of computational time. Another strategy to solve this problem is to apply Bellman's dynamic programming principal [17, 14], but due to its high computational effort has not been widely used.

The second great method to solve this problem is by direct transcription [18, 19, 20], which is mostly, within the field of trajectory optimization, a strategy to solve Optimal Control problems. This method transforms a continuous problem into a dicrete one, reducing therefore the complexity of some problems and increasing its computational efficiency. To use direct transcription one has to formulate the problem as convex, and once it is done each variable is discretized, facilitating the solution of the problem. This approach has been successfully applied not only to a manipulator to minimize time over a fixed path [21, 22, 23] but also for minimizing time in a track subject to its boundaries and to the car dynamics [24, 25, 26, 27].

II. Vehicle Dynamics

In this section is presented the vehicle dynamics model chosen to further be used, which is the bicycle model herewith a linear simplified tyre model. The tyre model adopted assumes that the lateral, F_y, and longitudinal, F_x forces vary linearly and proportionally with slip angle, α, and slip ratio, SR. The bicycle model is a simple model, in the way that it condenses in two wheels, one for each axle, the behaviour of four wheels. Therefore, both front wheels cornering stiffness are taken as a single quantity, C_{α_f}, and the same is done for the real axle, C_{α_r}. As for the longitudinal force, the same assumption was done, consequently, the contribution of the longitudinal force for the yaw moment is neglected. This approximation was done to avoid modelling an electric or mechanical differential. Moreover, Ackermann geometry concerns [28] are also neglected, hence each wheel steering angle is equal. The bicycle model is a well known model for control, since it is easily described as a state-space model, accounting for tyre and rigid body dynamics, which results in more reliable control actions, when compared with kinematic models. Load transfer could be added to this model, however that was not the choice, once again to simplify the problem. Nonetheless, such assumption results in a conservative solution, since the bicycle model underestimates the cornering and accelerating capabilities of the vehicle [29]. It occurs, due to the fact that load transfer increase the vertical load on the outer wheels and, consequently, its cornering stiffness. Moreover, the outer wheel, typically, has a higher slip angle which multiplies with the greater cornering stiffness, resulting in greater lateral force produced by the axle. A schematic of this model is represented in figure 1.

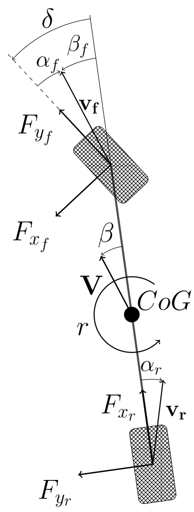

Figure 1: Representation of the longitudinal model including slip angles, $\alpha_{f,r}$, vehicle sideslip, β and the loads on each wheel, F_{x_f},F_{y_f},F_{x_r} and F_{y_r}.

with δ as the steering angle, α_f and α_r as the front and rear slip angles, respectively. Additionally, , each wheel velocity vector defined as $\mathbf{v_{f,r}} = (v_{x_{f,r}}, v_{y_{f,r}})$, and $\beta_{f,r} = \frac{v_{y_{f,r}}}{v_{x_{f,r}}}$. Furthermore, the vehicle velocity vector as $\mathbf{V} = (v_x, v_y)$, the body sideslip as $\beta = \frac{v_y}{v_x}$, and the yaw rate quantity as r. Moreover, each axle lateral force is defined as $F_{y_{f,r}} = C_{\alpha_{f,r}}\alpha_{f,r}$, arising from the tyre model linearization [30], defining the cornering stiffness as $C_\alpha = \left|\lim_{\alpha\to 0} \frac{\partial F_y}{\alpha}\right|$. At last, the longitudinal wheel load, represented as $F_{x_{f,r}} = C_{\mathrm{SR}_{f,r}}\mathrm{SR}_{f,r}$, with SR as the slip ratio and the *longitudinal slip coefficient* defined as $C_{\mathrm{SR}} = \left|\lim_{\mathrm{SR}\to 0} \frac{\partial F_x}{\mathrm{SR}}\right|$.

Considering small steering angles, $\sin(\delta) \approx 0$ and $\cos(\delta) \approx 1$ and taking the longitudinal, lateral and yaw moment equilibrium, one arrives at

$$\begin{aligned} m\dot{v}_x &= mrv_y + C_{SR_f}SR_f + C_{SR_r}SR_r - Kv_x^2 \\ m\dot{v}_y &= -mrv_x + C_{\alpha_f}\alpha_f + C_{\alpha_r}\alpha_r \\ I_z\dot{r} &= a_1C_{\alpha_f}\alpha_f - a_2C_{\alpha_r}\alpha_r \end{aligned} \tag{1}$$

with K reprsenting a longitudinal resistance coefficient and a_1 and a_2 as the front and rear axles distance to the CoG, respectively. I_z is the moment of inertia taken about the z-axis. The slip angles α_f and α_r, may be defined as

$$\alpha_f = \delta - \frac{v_y + a_1r}{v_x} \tag{2a}$$

$$\alpha_r = \frac{v_y - a_2r}{v_x} \tag{2b}$$

Notwithstanding, the biclycle model of equation 1 can be derived by means of the Euler-Lagrange approach, using $q = (x, y, \psi)$ as generalized coordinates, which constitutes an advantage for the optimization to be presented in the following sections1. Thus, from the Euler-Lagrange equations

$$\frac{d}{dt}\frac{\partial \mathcal{L}^T}{\partial \dot{q}} - \frac{\partial \mathcal{L}^T}{\partial} = \mathbf{u} \tag{3}$$

with $\mathbf{u}$ as the generalized forces and $\mathcal{L}(q, \dot{q}) = T(q, \dot{q}) - V(q)$, in which T and V represent the kinetic the potential energies, respectively. Exploiting such set of equations results in

$$\mathcal{M}\ddot{q} + \mathcal{C}(\dot{q}) + \mathcal{F}(\dot{q}) = \mathbf{u} \tag{4}$$

where $\mathcal{M}$ is the mass matrix, $\mathbb{R}^{3x3}$, $\mathcal{C}$ is the matrix accounting for centripetal and resistive forces, i.e. drag, $\mathbb{R}^{3x1}$ and $\mathcal{F}$ is a matrix with the tyre friction related coefficients, $\mathbb{R}^{3x1}$.

$$\begin{aligned}
\mathcal{M} &= \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} \\
\mathcal{C}(\dot{q}) &= \begin{bmatrix} -m\dot{y}\dot{\psi} - K\dot{x}^2 \\ m\dot{x}\dot{\psi} \\ 0 \end{bmatrix} \\
\mathcal{F}(\dot{q}) &= \begin{bmatrix} 0 \\ C_{\alpha_f}(\frac{\dot{y}}{\dot{x}} + a_1\frac{\dot{\psi}}{\dot{x}}) + C_{\alpha_r}(\frac{\dot{y}}{\dot{x}} - a_2\frac{\dot{\psi}}{\dot{x}}) \\ a_1 C_{\alpha_f}(\frac{\dot{y}}{\dot{x}} + a_1\frac{\dot{\psi}}{\dot{x}}) - a_2 C_{\alpha_r}(\frac{\dot{y}}{\dot{x}} - a_2\frac{\dot{\psi}}{\dot{x}}) \end{bmatrix} \\
\mathbf{u} &= \begin{bmatrix} 4.6875m & 0 \\ 0 & C_{\alpha_f} \\ 0 & a_1 C_{\alpha_f} \end{bmatrix} \begin{bmatrix} u_p \\ \delta \end{bmatrix}
\end{aligned} \tag{5}$$

where u_p is the pedal position, $-1 \leq u_p(t) \leq 1$.

III. Problem Formulation

For sake of simplicity, it is assumed that the vehicle moves along the track centerline, from the initial point s_0 until the final point s_f. To allow the car deviate from this path, a vector $\vec{n}(s)$, orthogonal to the body $x - axis$, is defined at each point s as $\mathbf{n(s)} = [\sin(\psi_0(s)), -\cos(\psi_0(s))]$. Furthermore, the car transversal car position may vary from $-b_{max}$ until b_{max} along the direction $\vec{n}(s)$, fig. 2. Thus, the transversal car position is defined as $b(s) \in [-b_{max}(s), -b_{max}(s)]$. Moreover, defining $r_0(s) = (x_0(s), y_0(s))$ one arrives at

$$\begin{aligned}
&r(s) = r_0(s) + b(s)\mathbf{n(s)} \Leftrightarrow \\
&\Leftrightarrow \begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} x_0(s) \\ y_0(s) \end{bmatrix} + b(s)\begin{bmatrix} \sin(\psi_0(s)) \\ -\cos(\psi_0(s)) \end{bmatrix}
\end{aligned} \tag{6}$$

One is assuming a normalized path vector, $s(0) = 0 \leq s(t) \leq s(T) = 1$, with T as the final time. Applying the chain rule to change time dependency to

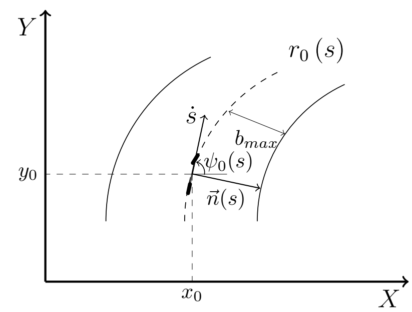

Figure 2: Representation of the track parametrization adopted

spatial dependency, one obtains the velocities and accelerations in equation 7.

$$\begin{aligned}
v_x = \dot{x} = x'\dot{s} \quad a_x = \ddot{x} = x'\ddot{s} + x''\dot{s}^2 \\
v_y = \dot{y} = y'\dot{s} \quad a_y = \ddot{y} = y'\ddot{s} + y''\dot{s}^2
\end{aligned} \tag{7}$$

Computing now the first and second derivative of x and y, in equation 7, with respect to s

$$\begin{aligned}
x'(s) &= x_0'(s) + b'(s)\sin(\psi_0(s)) + b(s)\psi_0'\cos(\psi_0(s)) \\
y'(s) &= y_0'(s) - b'(s)\cos(\psi_0(s)) + b(s)\psi_0'\sin(\psi_0(s)) \\
x''(s) &= x_0''(s) + b''(s)\sin(\psi_0(s)) \\
&\quad + 2b'(s)\psi_0'(s)\cos(\psi_0(s)) \\
&\quad + b\psi_0''(s)\cos(\psi_0(s)) - b\psi_0'^2(s)\sin(\psi_0(s)) \\
y''(s) &= y_0''(s) - b''(s)\cos(\psi_0(s)) \\
&\quad + 2b'(s)\psi_0'(s)\sin(\psi_0(s)) + b\psi_0''(s)\sin(\psi_0(s)) \\
&\quad + b\psi_0'^2(s)\cos(\psi_0(s))
\end{aligned} \tag{8}$$

To increase readability of the afore equations, one adds the following notation for the vector $\vec{n}(s)$ and its first and second derivatives taken for s

$$\begin{aligned}
n_x(s) &= \sin(\psi_0(s)) \\
n_y(s) &= -\cos(\psi_0(s)) \\
n_x'(s) &= \psi_0'(s)\cos(\psi_0(s)) \\
n_y'(s) &= \psi_0'(s)\sin(\psi_0(s)) \\
n_x''(s) &= \psi_0''(s)\sin(\psi_0(s)) - \psi_0'^2(s)\sin(\psi_0(s)) \\
n_y''(s) &= \psi_0''(s)\sin(\psi_0(s)) + \psi_0'^2(s)\cos(\psi_0(s))
\end{aligned}$$

Combining equations 8 and 9 one arrives at

$$\begin{aligned}
x'(s) &= x_0'(s) + b'(s)n_x(s) + b(s)n_x'(s) \\
y'(s) &= y_0'(s) + b'(s)n_y(s) + b(s)n_y'(s) \\
x''(s) &= x_0''(s) + b''(s)n_x(s) + 2b'(s)n_x'(s) \\
&\quad + b(s)n_x''(s) \\
y''(s) &= y_0''(s) + b''(s)n_y(s) + 2b'(s)n_y'(s) \\
&\quad + b(s)n_y''(s)
\end{aligned} \tag{9}$$

Substituting equations 9 on equations 7, one obtains

$$\begin{aligned}
v_x(s) &= \left[x_0'(s) + b'(s)n_x(s) + b(s)n_x'(s)\right]\dot{s}\\
v_y(s) &= \left[y_0'(s) + b'(s)n_y(s) + b(s)n_y'(s)\right]\dot{s}\\
a_x(s) &= \left[x_0'(s) + b'(s)n_x(s) + b(s)n_x'(s)\right]\ddot{s}\\
&+ \left[x_0''(s) + b''(s)n_x(s) + 2b'(s)n_x'(s) + b(s)n_x''(s)\right]\dot{s}^2\\
a_y(s) &= \left[x_0'(s) + b'(s)n_y(s) + b(s)n_y'(s)\right]\ddot{s}\\
&+ \left[y_0''(s) + b''(s)n_y(s) + 2b'(s)n_y'(s) + b(s)n_y''(s)\right]\dot{s}^2
\end{aligned} \tag{10}$$

The only quantity yet to define is the yaw, ψ, which would be computed as

$$\begin{aligned}
\psi(s) &= \tan^{-1}\left(\frac{y_0(s) + b(s)n_y(s)}{x_0(s) + b(s)n_x(s)}\right) \approx \psi_0(s)\\
\psi'(s) &= \psi_0'(s)\\
\psi''(s) &= \psi_0''(s)
\end{aligned} \tag{11}$$

which represents a linearization, which remains valid when the track width, b_{max} is small. Such approximation is quite useful, since it avoids highly non-linear terms, potentially harm for implementation.

Once again resorting to the chain rule one can rewrite the time dependent generalized coordinates, q as a function of the path variable s

$$\dot{q} = \frac{dq}{ds}\frac{ds}{dt} = q'\dot{s} \tag{12}$$

$$\ddot{q} = \frac{dq}{ds}\frac{d^2s}{dt^2} + \frac{d}{ds}\left(\frac{dq}{ds}\right) = q'\ddot{s} + q''\dot{s}^2 \tag{13}$$

At this point, one is capable of rewriting equations 5 in

$$\begin{aligned}
\mathcal{M}\ddot{q}(s) &= \mathbf{m_1}(s, b(s), b'(s))\ddot{s}\\
&+ \mathbf{m_2}(s, b(s), b'(s), b''(s))\dot{s}^2\\
\mathcal{C}(\dot{q}) &= \mathbf{c}(s, b(s), b')\\
\mathcal{F}(\dot{q}) &= \mathbf{f}(s, b(s), b')
\end{aligned} \tag{14}$$

which for sake of space will not be shown here.

The modified vehicle governing dynamics can be condensed in

$$\begin{aligned}
&\mathbf{R}(s)\left[\mathbf{m_1}(s, b(s), b'(s))\ddot{s} + \mathbf{m_2}(s, b(s), b'(s), b''(s))\dot{s}^2\right.\\
&\left.+\mathbf{f}(s, b(s), b'(s))\right] + \mathbf{c}(s, b(s), b'(s)) = \mathbf{u}(s)
\end{aligned} \tag{15}$$

where $\mathbf{R}(s)$ is defined as

$$\begin{bmatrix} \cos(\psi_0(s)) & -\sin(\psi_0(s)) & 0\\ \sin(\psi_0(s)) & \cos(\psi_0(s)) & 0\\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

and represents the rotation from the fixed reference axis coordinates to body coordinates. The component $\mathbf{c}(s, b(s), b'(s))$ is excluded from this operation, since it is already defined in body coordinates.

Next, as in [26], one must perform a change of the integration variable of the minimum time integral from time, t, to space, s

$$T = \int_0^T 1dt = \int_{s(0)}^{s(T)} \frac{ds}{\dot{s}} = \int_0^1 \frac{ds}{\dot{s}} \tag{17}$$

Second, two new optimization variables are introduced

$$\begin{aligned}
a(s) &= \ddot{s}\\
f(s) &= \dot{s}^2
\end{aligned} \tag{18}$$

To ensure the relation between $a(s)$ and $f(s)$ an additional constraint is added

$$f'(s) = 2a(s) \tag{19}$$

which comes from

$$\dot{f}(s) = f'\dot{s} \tag{20}$$

and

$$\dot{f}(s) = \frac{d(\dot{s}^2)}{dt} = 2\ddot{s}\dot{s} = 2a(s)\dot{s} \tag{21}$$

Resorting to equation 15 and defining $c(s) = b'(s)$ and $d(s) = b''(s)$, one can formulate the minimum time trajectory problem, following the afore mentioned notation, as an optimal control problem:

$$\min_{a(\cdot),f(\cdot),b(\cdot),c(\cdot),d(\cdot),\mathbf{u}(\cdot)} \int_0^1 \frac{1}{\sqrt{f(s)}} \tag{22}$$

$$+ \frac{\gamma_1}{\sqrt{f(s)}}\left(\sum_{i=1}^{n}\frac{u_i^2(s)}{\bar{u}_i^2}\right) + \gamma_2\left(\sum_{i=1}^{n}\frac{|u_i'(s)|}{|\bar{u}_i|}\right) ds \tag{23}$$

$$\begin{aligned}
&\text{subject to } \mathbf{u}(s) = \mathbf{R}(s)\left[\mathbf{m_1}(s, b(s), b'(s))a(s)\right.\\
&\left.+ \mathbf{m_2}(s, b(s), b'(s), b''(s))f(s) + \mathbf{f}(s, b(s), b'(s))\right]\\
&+ \mathbf{c}(s, b(s), b'(s))
\end{aligned} \tag{24}$$

$$f(0) = \dot{s}_0^2 \tag{25}$$

$$f(1) = \dot{s}_T^2 \tag{26}$$

$$b(0) = b_0 \tag{27}$$

$$b(1) = b_T \tag{28}$$

$$f'(s) = 2a(s) \tag{29}$$

$$b'(s) = c(s) \tag{30}$$

$$c'(s) = d(s) \tag{31}$$

$$f(s) \geq 0 \tag{32}$$

$$-b_{max} \leq b(s) \leq b_{max} \tag{33}$$

$$\underline{\alpha}(s) \leq g(\dot{q}(s), \delta(s)) \leq \bar{\alpha}(s) \tag{34}$$

$$\underline{\mathbf{u}}(s(t)) \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(s(t)) \tag{35}$$

$$\text{for } s \in [0, 1] \tag{36}$$

in which b_0 and b_T represent, respectively, the vehicle starting and finishing transversal deviation from the reference trajectory. $\dot{s}_0^2$ and $\dot{s}_T^2$ represent the initial and final velocities of the vehicle, respectively. The problem defined by equations 23-36 can be regarded as an optimal control problem in differential algebraic form (DAE), with pseudo time s, three differential states b, c and f, an algebraic state variable $\mathbf{u}$ and two input variables a and d.

The two last terms of equation 23 represent additional weightings to the objective function. The first term is defined as the thermal energy which is computed as

$$\int_0^1 u_i^2(s)dt = \int_0^1 \frac{u_i^2(s)}{\dot{s}}ds = \int_0^1 \frac{u_i^2(s)}{\sqrt{f(s)}}ds \quad (37)$$

The second term is the integral of the absolute value of the inputs rate of change, which is useful to avoid jitter, and is defined as

$$\int_0^T |\dot{u}_i(s)|dt = \int_0^1 \frac{|u_i'(s)\dot{s}|}{\dot{s}}ds = \int_0^1 |u_i'(s)|ds \quad (38)$$

Equation 34 is an example of a constraint that can be added, in this case is limiting the slip angle between a lower bound, $\underline{\alpha}(s)$, and an upper bound, $\bar{\alpha}(s)$. Many other constraints and weights of the objective can be added [21].

To guarantee the convexity of an optimization problem one must have a convex objective function, linear dynamic system and convex constraints. In equations 23-36, the objective function is convex, because a power function is convex [31], but the dynamic system, equation 24, is not linear due to the terms $b(s)$, $c(s)$ and $d(s)$. Although the other system constraints are convex, the fact that the system dynamics is nonlinear defines this problem as non-convex. It is easy to prove that if b tends to zero the problem is one of path tracking, which has been proved to be convex [21, 26].

IV. Numerical Solution

To solve the generalized optimal control problem there are three well recognized methods. Dynamic programming methods [17, 14, 32], which are by nature a decision-based method instead of a gradient based, hence falling in a group of methods that require a very careful implementation due to the sequential way this method solve the problem. Furthermore, dynamic programming is $\mathcal{O}(n)$, requiring, therefore, considerable computational time. The second method falls under the category of indirect methods, which are characterized by the several sub-steps necessary to solve the problem [12, 11]. At last, most recently, methods such as the direct method, motivated by the work developed by [18, 19] and others, allowed the solution of the optimal control using quadratic and/or convex optimization methods. This set of solutions is also called direct transcription. Reformulating the problem obtained from the direct transcrption one can get to a convex optimal control problem in the form of a Second Order Cone Programming, thus enabling the use of efficient mature solvers, such as SeDuMi [33] or CVX [34], as suggested by [21]. In this section, firstly, one demonstrates the direct transcription method applied to the general optimal control problem, transforming it in a large scale optimization problem. Secondly, for the special case of path tracking, a Second Order Cone Programming formulation is employed.

Direct Transcription

For the direct transcription adopted in this work, first the path, $s \in [0, 1]$, is discretized, resulting in $K + 1$ points, as in equation 39.

$$s^0 = 0 \le s^k \le s^K \text{for } k = 0, ..., k \quad (39)$$

Next the acceleration, $a(\cdot)$, is parametrized as a piecewise constant function, fig. 11 in Appendix, the squared velocity, $f(\cdot)$, is consequently a piecewise constant function, fig. 12 in Appendix , and the inputs, $u_i(\cdot)$ result to be piecewise nonlinear functions. Analogously, the optimal control in DAE form input $d(\cdot)$ is parametrized as a piecewise constant function, fig. 8 in Appendix, the first spatial derivative of $b(s)$, $c(\cdot)$, is piecewise linear, fig. 9 in Appendix, and thereupon the path transversal position $b(\cdot)$ is piecewise quadratic, fig. 10 in Appendix. Therefore, to d is evaluated at the middle points, $d^k = d(s^{k+\frac{1}{2}})$, c is evaluated at the end points and in the middle function, its values is $c^{k+\frac{1}{2}} = \frac{c^k + c^{k+1}}{2}$ and, finally b also has its transcribed values at the end points but can be computed at middle points by $b^{k+\frac{1}{2}} = b^k + \frac{c^k \Delta s_k}{2} + \frac{d^k \Delta s_k^2}{8}$.

One can easily infer that due to the discretization the relations between d, c and b are

$$\begin{aligned} b^{k+1} - b^k &= \Delta s_k c^k + \frac{1}{2}\Delta s_k^2 d^k \\ c^{k+1} - c^k &= \Delta s_k d^k \end{aligned} \quad (40)$$

So, following such transcription, it is useful to define f^k on the grid points s^k and one can define the continuous velocity function, due to the assumption of piecewise linear, as

$$f(s) = f^k + \frac{f^{k+1} - f^k}{s^{k+1} - s^k}(s - s^k) \quad \text{for } s \in [s^k, s^{k+1}] \quad (41)$$

From here on, it is assumed that $f(s^k) = f^k$ and, by ease of implementation, a^k and u_i^k are evaluated at the middle points, particularly at $s^{k+\frac{1}{2}}$, as can be inspected in fig. 11 to 13 in Appendix. Hence, $a^k = a(s^{k+\frac{1}{2}})$ and $u_i^k = u_i(s^{k+\frac{1}{2}})$.

It is important to mention that the evaluation of $\mathbf{R}(s)$, $x_0(s)$,$y_0(s)$, $\psi_0(s)$ and equations 9 is done at middle points and its computation is performed in pre-processing, which relieves optimization computational cost.

To arrive at the optimization problem in its final form, for sake of space, one only presented the final equations, more details can be found in [21, 26]. Hence, the optimization problem after the direct transcription can be regarded as

$$\min_{a^k,f^k,b^k,c^k,d^k,\mathbf{u}^k} \sum_{k=0}^{K-1} \frac{2\Delta s^k \left(1+\gamma_1 \sum_{i=1}^{n} \frac{(u_i^k)^2}{\bar{u}_i^2}\right)}{\sqrt{f^k}+\sqrt{f^{k+1}}}$$

$$+\gamma_2 \sum_{k=0}^{K-1} \left(\sum_{i=1}^{n} \frac{|\Delta u_i^k|}{|\bar{u}_i|}\right) \quad (42)$$

$$\text{subject to } \mathbf{u}^k = \mathbf{R}^k \Big[\mathbf{m_1}(s^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}})a^k$$

$$+\mathbf{m_2}(s^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}}, d^k) f^{k+\frac{1}{2}}+$$

$$+\mathbf{f}^k(s^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}})\Big] + \mathbf{c}(s^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}}) \quad (43)$$

$$f^0 = \dot{s}_0^2 \quad (44)$$

$$f^K = \dot{s}_T^2 \quad (45)$$

$$b^0 = b_0 \quad (46)$$

$$b^K = b_T \quad (47)$$

$$(f^{k+1} - f^k) = 2a^k \Delta^k \quad (48)$$

$$b^{k+1} - b^k = \Delta s_k c^k + \frac{1}{2}\Delta s_k^2 d^k \quad (49)$$

$$c^{k+1} - c^k = \Delta s_k d^k \quad (50)$$

$$f^k \geq 0 \quad (51)$$

$$-b_{max} \leq b^k \leq b_{max} \quad (52)$$

$$\underline{\alpha}^k \leq g(q_0^k, \mathbf{n}^k, \mathbf{n'}^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}}, \delta^k) \leq \bar{\alpha}^k \quad (53)$$

$$\underline{\mathbf{u}}^k \leq \mathbf{u}^k \leq \bar{\mathbf{u}}^k \quad (54)$$

$$\text{for } s \in [0,1] \quad (55)$$

The nonlinear optimization problem of 42-55 , formulated in this form, appears to be simple and well formulated. The issue is that the problem is nonlinear and non-convex, due to its nonlinear constraints, and, therefore, much more difficult to solve. An important statement is that the nonlinearity comes from the variables b^k, c^k and d^k and it is easy to prove that if c^k and d^k are zero, than the problem falls in the convex problem of path tracking. Thus, if one considers b^k, c^k and d^k to be small, which, actually, is a reasonable assumption on a race track, since b_{max} is usually small and the racing line is supposed to be smooth, the constraints 43 may be considered slightly near linear, enabling the use, without much error, of previously mentioned SCP and SQP methods to linearize the constraints.

Second Order Cone Programming

The second order cone program is a type of optimization in which a linear function is minimized over the intersection of an affine set and the product of second-order (quadratic) cones [35]. Furthermore, SOCP are nonlinear convex problems that include not only linear but also quadratic programs as particular cases. One of the main advantages of SOCP are the several primal-dual interior-point methods that have been developed over the last two decades, which allow the solution of difficult problems in a very fast and efficient fashion. A wide band of applications of optimal control problems have been successfully solved using SOCP, power electronics [36], switching problems [37], smart buildings climate control [38] and, finally, optimal path tracking for manipulator robots [21].

A second order cone programming problem is of the form

$$\min h^T x \quad (56)$$

$$\text{subject to } \parallel A_i x + b_i \parallel \leq c_i^T x + d_i, \quad (57)$$

$$\text{with } i = 1, ..., N \quad (58)$$

with $x \in \mathbb{R}^n$ as the optimization variable, and the problem parameters are $h \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n_i - 1}$, $b_i \in \mathbb{R}^{n_i - 1}$, $c_i \in \mathbb{R}^n$ and $d_i \in \mathbb{R}$. The operator $\parallel (\cdot) \parallel$ is defined as the standard Euclidean norm, $\parallel x \parallel = (x^T x)^{\frac{1}{2}}$.

The second order cone constraint of dimension n_i is the inequation

$$\parallel A_i x + b_i \parallel \leq c_i^T x + d_i \quad (59)$$

The standard cone of dimension k, which is also called Lorentz cone, is defined as

$$\mathcal{H}_k = \left\{ \begin{bmatrix} x \\ w \end{bmatrix} | x \in \mathbb{R}^{k-1}, w \in \mathbb{R}, \parallel x \parallel \leq w \right\} \quad (60)$$

The set of points defining a second order cone constraint are given by

$$\parallel A_i x + b_i \parallel \leq c_i^T x + d_i \Leftrightarrow \begin{bmatrix} A_i \\ c_i^T \end{bmatrix} x + \begin{bmatrix} c_i \\ d_i \end{bmatrix} \in \mathcal{H}_{n_i} \quad (61)$$

which is convex, thus SOCP is a convex programming problem since the objective function and the constraints both define convex sets. Several suitable SOCP objective functions and constraints can be found in [35].

To what concerns the scope of this work, one has to reformulate the objective function and constraints of problem 42-55, and to that end a new set of optimization variables must be introduced. An issue when adding new optimization variables is that frequently the physical meaning is lost, but, as mentioned before, this is done in regard of substantial increase of computational efficiency.

To rewrite equation the previous objective function 42 as linear, one has to introduce slack variables d^k with $k = 0, ..., K-1$ and $\mathbf{e}^k \in \Re^n$ for $k = 1, ..., K-1$, in a way that equation 42 may be rewritten as

$$\sum_{k=0}^{K-1} 2\Delta s^k d^k + \gamma_2 \sum_{k=1}^{max} \mathbf{1^T e}^k \quad (62)$$

with $\mathbf{1} \in \Re^n$ as dimension n vector with all entries equal to 1. It is now necessary to augment the problem 42-55 with the inequalities contraints [21]

$$\frac{1+\gamma_1 \sum_{n}^{i=1} \frac{\left(u_i^k\right)^2}{\bar{u}_i^2}}{\sqrt{f^{k+1}}+\sqrt{f^k}} \leq d_k, \quad \text{for } k=0,...,K-1 \tag{63}$$

and

$$-\mathbf{e}^k \leq \begin{bmatrix} \frac{\Delta u_1^k}{|\bar{u}_1|} \\ \vdots \\ \frac{\Delta u_n^k}{|\bar{u}_n|} \end{bmatrix} \leq \mathbf{e}^k, \quad \text{for } k=0,...,K-1 \tag{64}$$

Additionally, to obtain a SOCP, constraints 63 must be replaced by two equivalent constraints by introducing variables c^k for $k=0,...,K$ as

$$\begin{aligned} \frac{1+\gamma_1 \sum_{n}^{i=1} \frac{\left(u_i^k\right)^2}{\bar{u}_i^2}}{\sqrt{c^{k+1}}+\sqrt{c^k}} &\leq d_k \quad &&, \text{for } k=0,...,K-1 \\ c^k &\leq \sqrt{f^k} \quad &&, \text{for } k=0,...,K \end{aligned} \tag{65}$$

Now, conditions 65 may be rewritten as two second order cone constraints

$$\begin{aligned} &\frac{1+\gamma_1 \sum_{n}^{i=1} \frac{\left(u_i^k\right)^2}{\bar{u}_i^2}}{\sqrt{c^{k+1}}+\sqrt{c^k}} \leq d_k \Leftrightarrow \\ \Leftrightarrow &\left\| \begin{matrix} 2 \\ 2\sqrt{\gamma_1}\frac{u_1^k}{\bar{u}_1} \\ \vdots \\ 2\sqrt{\gamma_1}\frac{u_n^k}{\bar{u}_n} \\ c^{k+1}+c^k-d^k \end{matrix} \right\| \leq c^{k+1}+c^k+d^k \\ &\text{for } k=0,...,K-1 \end{aligned} \tag{66}$$

and

$$c^k \leq \sqrt{b^k} \Leftrightarrow \left\| \begin{matrix} 2c^k \\ b^k-1 \end{matrix} \right\| \leq b^k+1, \quad \text{for } k=0,...,K \tag{67}$$

After defining the new linear objective function and the second order cone constraints one can finally arrive at the minimum time problem formulation in the classic SOCP form 56-58

$$\min_{a^k,f^k,\mathbf{u}^k,c^k,d^k,e^k} \sum_{k=0}^{K-1} 2\Delta s^k d^k + \gamma_2 \sum_{k=1}^{K-1} \mathbf{1}^T \mathbf{e}^k \tag{68}$$

$$\text{subject to } \mathbf{u}^k = \mathbf{m}(s^{k+\frac{1}{2}})a^k + \mathbf{c}(s^{k+\frac{1}{2}})f(s) + \mathbf{f}(s^{k+\frac{1}{2}}) \tag{69}$$

$$f(0) = \dot{s}_0^2 \tag{70}$$

$$f(K) = \dot{s}_T^2 \tag{71}$$

$$(f^{k+1} - f^k) = 2a^k\Delta^k \tag{72}$$

$$\underline{\alpha}(s^{k+\frac{1}{2}}) \leq g(\dot{q}(s^{k+\frac{1}{2}}), \delta^k) \leq \bar{\alpha}(s^{k+\frac{1}{2}}) \tag{73}$$

$$\underline{\mathbf{u}}(s^{k+\frac{1}{2}}) \leq \mathbf{u}^k \leq \bar{\mathbf{u}}(s^{k+\frac{1}{2}}) \tag{74}$$

$$\left\| \begin{matrix} 2 \\ 2\sqrt{\gamma_1}\frac{u_1^k}{\bar{u}_1} \\ \vdots \\ 2\sqrt{\gamma_1}\frac{u_n^k}{\bar{u}_n} \\ c^{k+1}+c^k-d^k \end{matrix} \right\| \leq c^{k+1}+c^k+d^k, \text{ for } k=0,...,K-1 \tag{75}$$

$$\left\| \begin{matrix} 2c^k \\ b^k-1 \end{matrix} \right\| \leq b^k+1 \text{ and } f^k \geq 0, \text{ for } k=0,...,K \tag{76}$$

$$-\mathbf{e}^k \leq \begin{bmatrix} \frac{\Delta u_1^k}{|\bar{u}_1|} \\ \vdots \\ \frac{\Delta u_n^k}{|\bar{u}_n|} \end{bmatrix} \leq \mathbf{e}^k, \text{ for } k=1,...,K-1 \tag{77}$$

$$\text{for } s \in [0,1] \tag{78}$$

The formulation as a SOCP may not be the simplest, due to the introduction of auxiliary variables a^k,c^k, d^k and e^k, yet it allows the use of much efficient solvers, for further details see [21]. Using this formulation, after obtaining a solution, one uses the variable b^k to compute the minimum time over the fixed path

$$t(s) = \int_0^1 \frac{1}{\sqrt{b(s)}} ds \tag{79}$$

V. Numerical Results

In this section the results of the afore mentioned methods are presented. First, the results of the optimal control trajectory optimization are shown and next, for the same, track, the results of the optimal control SOCP formulation path tracking are presented. The car parameters used were $m = 0.68Kg$, $C_{\alpha_f} = 8$ N/rad, $C_{\alpha_r} = 8$ N/rad, $a_1 = 0.08025$m, $a_2 = 0.084975$m, $I_z = 0.01$Kgm^2, $\delta_{max} = 22$ degrees and $K = 0.25$. Regarding the optimization parameters, 75 points equally spaced points were used to discretize the path, γ_2 was set to 1e^{-2} and a slip angle limitation, $|\alpha| \leq 5$, was introduced. In fig. 3 one can find the obtained path, in which the tendency to increase the curvature radius is obvious, since an higher radius leads to greater longitudinal velocity, represented in fig. 4. As for the longitudinal velocity, a more careful optimization parameters choice, namely γ_2, would eliminate the oscillations, which are clear in the inputs fig. 5. It is difficult to say that these results are global optimal, due to the non-convexity of the problem. These results were obtained using the Sequential Quadratic Programming(SQP) solver of the MATLAB function *fmincon*, which linearizes the problem at each iteration to overcome the nonlinearity. Even though, this issue might have not been surpassed.

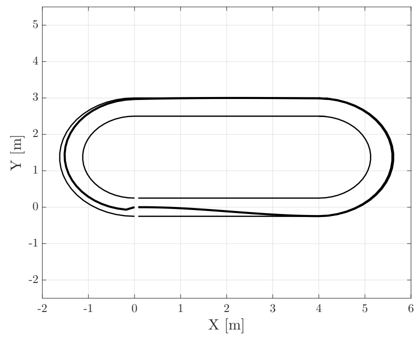

Figure 3: Optimal control minimum time trajectory

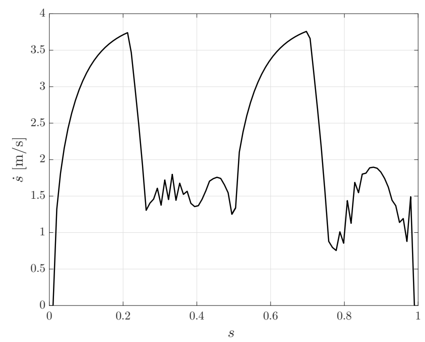

Figure 4: Optimal control minimum time trajectory longitudinal velocity in m/s

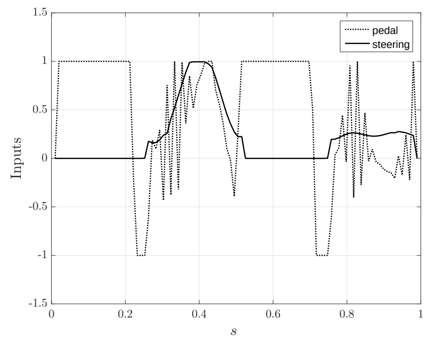

Figure 5: Optimal control minimum time trajectory longitudinal normalized inputs

For the special case of the optimal control path tracking, for the fig. 3 track's middle line, the longitudinal velocity is in fig. 6 and the inputs in fig. 7. This algorithm delivered a very smooth longitudinal velocity profile likewise the inputs. The algorithm processing time is reduced, about 3.5 seconds, using 200 points for the path discretization, using an Intel i5 2410m processor in a machine with 4Gb of RAM.

Both formulations proved to be effective and, regarding the SOCP formulation, very efficient. Nevertheless, it still lacks a more mature tyre model, which justifies the steering angle of 1 in fig. 5 and 7, since the current model assumes the lateral force varies linearly with the slip angle, which increases as the steering angle increases.

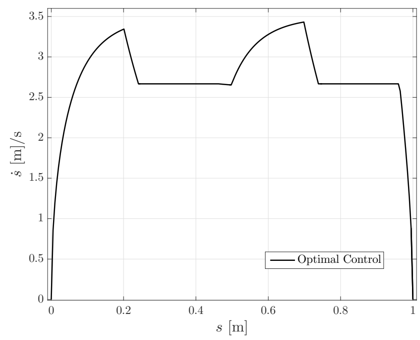

Figure 6: Optimal control minimum time longitudinal velocity in m/s

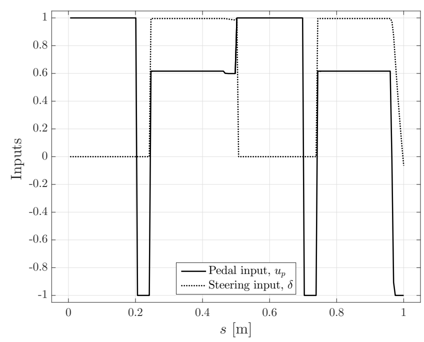

Figure 7: Pedal input and steering inputs, u_p and δ.

VI. Conclusions

In this work the minimum lap time is formulated as an optimal control problem. Vehicle dynamics were taken in account to obtain the minimum time problem. Next it was reformulated as a large scale optimization method using a direct transcription method. The results obtained could be improved by adapting the solvers used to be able to solve nonlinear optimization problems.

VII. Appendix

Figures 8-13 represent the parametrization done on the direct transcription applied in the minimum time optimal control trajectory optimization and minimum time optimal control path tracking formulations.

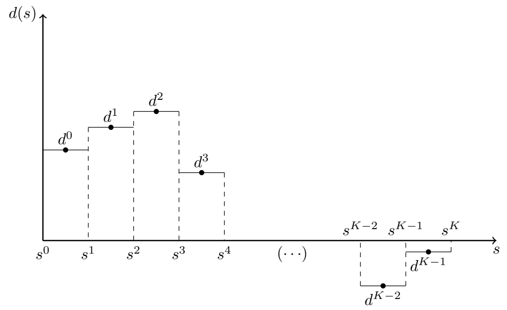

Figure 8: Second spatial derivative of $b(s)$, $d(s)$, represented as a piecewise constant function for $s \in [0, 1]$

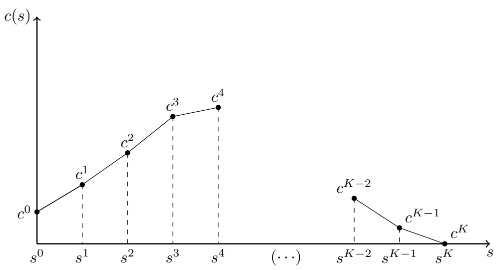

Figure 9: First spatial derivative of $b(s)$, $c(s)$, represented as a piecewise linear function for $s \in [0, 1]$

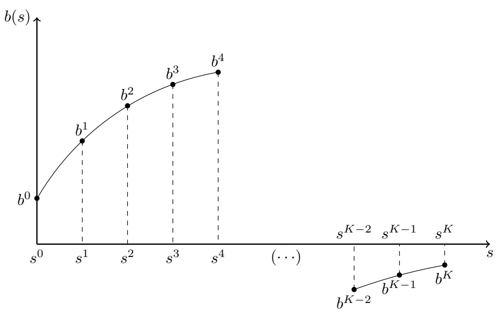

Figure 10: Car transversal position in the track, $b(s)$, represented as piecewise quadratic functions for $s \in [0, 1]$

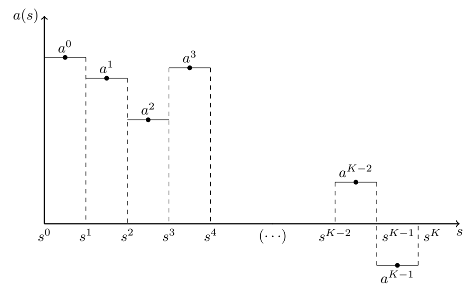

Figure 11: Acceleration, $a(s)$, represented as a piecewise constant function for $s \in [0, 1]$

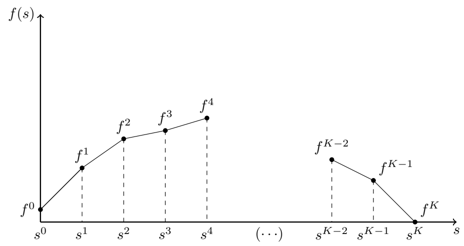

Figure 12: Velocity, $f(s)$, represented as a piecewise linear function for $s \in [0, 1]$

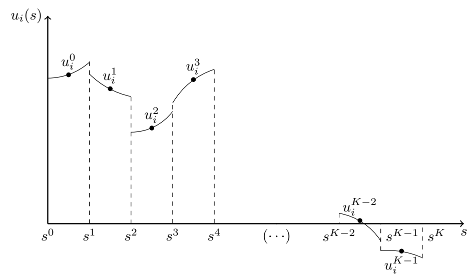

Figure 13: Inputs, $u_i(s)$, represented as piecewise nonlinear functions for $s \in [0, 1]$

References

[1] S. M. LaValle, *Planning algorithms.* Cambridge university press, 2006.

[2] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, "Constrained cartesian motion control for teleoperated surgical robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, 1996.

[3] P. Kazanzides, J. Zuhars, B. Mittelstadt, and R. H. Taylor, "Force sensing and control for a surgical robot," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on.* IEEE, 1992, pp. 612–617.

[4] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. controlling robot

motion with color image segmentation," *IEEE Engineering in Medicine and Biology Magazine*, vol. 16, no. 1, pp. 40–45, 1997.

[5] J.-C. Latombe, *Robot motion planning.* Springer Science & Business Media, 2012, vol. 124.

[6] C. Chen, M. Rickert, and A. Knoll, "Kinodynamic motion planning with space-time exploration guided heuristic search for car-like robots in dynamic environments," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on.* IEEE, 2015, pp. 2666–2671.

[7] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *International Conference on Intelligent Robotics and Applications.* Springer, 2012, pp. 588–597.

[8] T. Gu, J. Snider, J. M. Dolan, and J.-w. Lee, "Focused trajectory planning for autonomous on-road driving," in *Intelligent Vehicles Symposium (IV), 2013 IEEE.* IEEE, 2013, pp. 547–552.

[9] J. Larson, M. Bruch, R. Halterman, J. Rogers, and R. Webster, "Advances in autonomous obstacle avoidance for unmanned surface vehicles," DTIC Document, Tech. Rep., 2007.

[10] S. Teller, M. R. Walter, M. Antone, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, J. Glass, J. P. How, A. S. Huang *et al.*, "A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on.* IEEE, 2010, pp. 526–533.

[11] J. E. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The international journal of robotics research*, vol. 4, no. 3, pp. 3–17, 1985.

[12] Z. Shiller and Y.-R. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 241–249, 1991.

[13] Z. Shiller, "Time-energy optimal control of articulated systems with geometric path constraints," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on.* IEEE, 1994, pp. 2680–2685.

[14] K. Shin and N. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Transactions on Automatic Control*, vol. 31, no. 6, pp. 491–500, 1986.

[15] Z. Shiller, "Off-line and on-line trajectory planning," in *Motion and Operation Planning of Robotic Systems.* Springer, 2015, pp. 29–62.

[16] J.-J. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 118–124, 1989.

[17] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal on Robotics and Automation*, vol. 3, no. 2, pp. 115–123, 1987.

[18] J. T. Betts and W. P. Huffman, "Path-constrained trajectory optimization using sparse sequential quadratic programming," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 1, pp. 59–68, 1993.

[19] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[20] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[21] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[22] D. Costantinescu and E. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *Journal of robotic systems*, vol. 17, no. 5, pp. 233–249, 2000.

[23] T. Ardeshiri, M. Norrlöf, J. Löfberg, and A. Hansson, "Convex optimization approach for time-optimal path tracking of robots with speed dependent constraints," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 14 648–14 653, 2011.

[24] G. Perantoni and D. J. Limebeer, "Optimal control for a formula one car with variable parameters," *Vehicle System Dynamics*, vol. 52, no. 5, pp. 653–678, 2014.

[25] R. Lot and F. Biral, "A curvilinear abscissa approach for the lap time optimization of racing vehicles," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7559–7565, 2014.

[26] D. Q. Tran and M. Diehl, "An application of sequential convex programming to time optimal trajectory planning for a car motion," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on.* IEEE, 2009, pp. 4366–4371.

[27] D. Casanova, "On minimum time vehicle manoeuvring: The theoretical optimal lap," 2000.

[28] W. C. Mitchell, A. Staniforth, and I. Scott, "Analysis of ackermann steering geometry," SAE Technical Paper, Tech. Rep., 2006.

[29] A. Rucco, G. Notarstefano, and J. Hauser, "Computing minimum lap-time trajectories for a single-track car with load transfer," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on.* IEEE, 2012, pp. 6321–6326.

[30] R. Jazar, *Vehicle dynamics: theory and application.* Springer, 2008.

[31] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[32] M. Leu and S. Singh, "Optimal trajectory generation for robotic manipulators using dynamic programming," *ASME JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL*, vol. 109, pp. 88–96, 1987.

[33] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[34] M. Grant, S. Boyd, and Y. Ye, "CVX: MATLAB software for disciplined convex programming," 2008.

[35] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear algebra and its applications*, vol. 284, no. 1-3, pp. 193–228, 1998.

[36] M. Farivar, R. Neal, C. Clarke, and S. Low, "Optimal inverter VAR control in distribution systems with high pv penetration," in *Power and Energy Society General Meeting, 2012 IEEE.* IEEE, 2012, pp. 1–7.

[37] S. C. Bengea and R. A. DeCarlo, "Optimal control of switching systems," *automatica*, vol. 41, no. 1, pp. 11–27, 2005.

[38] F. Oldewurtel, A. Parisio, C. N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth, "Energy efficient building climate control using stochastic model predictive control and weather predictions," in *American control conference (ACC), 2010.* IEEE, 2010, pp. 5100–5105.