Time optimal control of ground vehicles

György Max∗, Béla Lantos†

∗Budapest University of Technology and Economics, Department of Control Engineering and Information Technology
H-1117 Budapest, Magyar tudósok krt. 2, Hungary Email: max@iit.bme.hu

†Széchenyi István University, Research Centre of Vehicle Industry,
H-9026 Győr, Egyetem tér 1, Hungary Email: lantos@iit.bme.hu

Abstract—The paper deals with the time optimal control of automatically driven cars modeled with gear shift as discrete control input beside the continuous ones in a test path between corridors. The car is required to avoid a static obstacle or performing double lane change. The problem can be formulated as a mixed-integer optimal control problem (MIOCP). The resulting MIOCP is solved by reformulating it to static mixed-integer nonlinear program (MINLP) using time discretization and direct multiple shooting method. Non-commercial open software packages are applied that substantially use the gradients of the objective function and the Jacobians of the constraints exploiting sparsity. A novel algorithm and implementation is presented for computing the derivatives of the complex state trajectory joining equations. This algorithm was given in the form of matrix differential equations whose structure allowed to compute their solution using RK4 in matrix form. The elaborated method can be applied both for combustion engine and electric driven cars. It can form the basis to generate an offline database of a central general collision avoidance system (CAS) for varying path parameters on a grid which can support real time applications.

Index Terms—Time optimal control, Vehicle model, Direct multiple shooting, Mixed integer nonlinear programming, Sparse gradient and Jacobian

I. INTRODUCTION

Pursuit of optimal behavior in technical and other (biological, economical etc.) type of systems is a common goal of human research. In this area dynamic and static optimization problems are of different complexity. Especially, the difficulty increases if some of the optimization variables are integer or binary valued.

Interest in MIOCPs (mixed-integer optimal control problems) has been increased over the past few years. The first applications of MIOCPs appeared in the field of chemical processes and transportation systems. In this paper we examine the time optimal control of combustion engine driven ground vehicles with gear shifts as discrete control input beside the other continuous control variables. Similar problems were discussed in the literature for cars described by ordinary (ODE) or differential-algebraic (DAE) equations in [18], [11], [6] considering fixed or moving time interval or based on moving horizon predictive approaches [12], [5].

There are several techniques to solve MIOCP problems: dynamic programming approach by solving the Hamilton-Jacobi-Bellmann equation, indirect methods (known as first optimize, then discretize) and direct methods (first discretize, then optimize). The latter group of techniques can be applied to large-scale optimal control problems.

In direct methods the continuous time infinite dimensional MIOCP problem is first discretized and reformulated to a finite-dimensional MINLP problem (mixed-integer nonlinear program).

A professional software package to solve the original continuous time problem may be MUSCOD-II [7], however it is not an open software. Hence, the novel professional and open optimization system OPTI [10] was chosen, which has also an interface to AMPL modeling language to formulate optimum problems at high level and has better performance than MATLAB Optimization Toolbox.

The structure of the paper is as follows. In Section II the dynamic model of the car is introduced, furthermore the test path and the dynamic optimal control problem are formulated. In Section III the multiple shooting method is presented and the reformulation of the dynamic MIOCP problem to the static MINLP problem by using time, state and control discretization are described. The novel algorithm of computing the derivatives of the complex trajectory joining constraints for the Jacobians are presented in Section III. Because the problem of time optimal control of cars is a large-scale one, hence sparsity of the gradient and Jacobians is taken into consideration. Numerical results of the time optimal control are given in Section IV. The paper is concluded in Section V.

II. CONCEPT OF OPTIMAL CONTROL PROBLEM

A. Dynamic car model

Consider a car moving in a horizontal plane driven by the rear wheels and steered by the front wheels. It is assumed that the right (r) and left (l) side of the vehicle is symmetrical, thus the two halves can be merged to a single-track model, see Figure 1. Furthermore, the rolling and pitching of the car is neglected. In the sequel a combustion engine driven car is assumed but the results can be extended to electrical cars as well.

The state variables of the car model are $p_x, p_y, v, \beta, \psi, \omega_z, \delta_w$, i.e. the x-y position of the center of gravity (CoG), the magnitude of velocity, the side-slip angle, the orientation, the yaw rate and the steering angle of the front wheel respectively.

The control variables that represent the driver’s input to the vehicle are denoted with $\omega_s, F_b, \phi, \mu$, which are respectively, the angular velocity of the steering wheel, the total braking
force, the accelerator pedal’s position and the gear shift mapped to the gear transmission ratio.

The single-track dynamics of the car can be given in the form of $\dot{x} = f(t, x, u)$ by the following system of ordinary differential equations (ODE) similar in [16] and [15]:

$$\dot{p}_x = v \cos(\psi + \beta)$$  \hspace{1cm} (1a)  
$$\dot{p}_y = v \sin(\psi + \beta)$$  \hspace{1cm} (1b)  
$$\dot{v} = \frac{1}{m}[(F^\mu_{\text{tr}} - F^\mu_{\text{ax}}) \cos(\beta) + F_{\text{tr}} \cos(\delta_w - \beta) + (F_t - F_{\text{ax}}) \sin(\beta) - F_{\text{tr}} \sin(\delta_w - \beta)]$$  \hspace{1cm} (1c)  
$$\dot{\delta}_w = \omega \delta$$  \hspace{1cm} (1d)  
$$\dot{\beta} = \frac{1}{m_0}[(F_{\text{ax}} - F^\mu_{\text{tr}}) \sin(\beta) + F_{\text{tr}} \sin(\delta_w - \beta) + (F_t - F_{\text{ax}}) \cos(\beta) + F_{\text{tr}} \cos(\delta_w - \beta)] - \omega_z$$  \hspace{1cm} (1e)  
$$\dot{\omega}_z = \frac{1}{I_{zz}}[F_{\text{tr}} l_f \cos(\delta_w) - F_{\text{tr}} l_R - F_{\text{ax}} v_{SP} + F_{\text{tr}} l_f \sin(\delta_w)]$$  \hspace{1cm} (1f)  

with state and control variable vectors of

$$x = (p_x, p_y, v, \beta, \psi, \omega_z, \delta_w)^T, \quad u = (\omega, F_b, \phi, \mu)^T$$

External forces $F_1$, $F_t$, $F_{\text{ax}}$ are the transversal, longitudinal and aerodynamic forces acting on the car respectively. The second subscripts $R$, $F$ and $x$, $y$ indicate the rear, front wheel and the direction of the aerodynamic force respectively.

The transversal (lateral) forces are described by Pacejka’s magic formula, see in [3] and [13]. The front and rear slip angles and transversal forces are given as follows:

$$\alpha_f = \delta_w - \arctan\left(\frac{l_f \dot{\psi} + v \sin \beta}{v \cos \beta}\right)$$  \hspace{1cm} (2)  
$$\alpha_e = \arctan\left(\frac{l_e \dot{\psi} - v \sin \beta}{v \cos \beta}\right)$$  \hspace{1cm} (3)  
$$F_{\text{tr}, \text{fr}} = D_f \sin(C_{\text{fr}} \arctan(B_{\text{fr}} \alpha_{\text{fr}} - E_{\text{fr}}(B_{\text{fr}} \alpha_{\text{fr}} - \arctan(B_{\text{fr}} \alpha_{\text{fr}}))))$$  \hspace{1cm} (4)  

The total braking force is distributed between the front and rear wheel in fixed (2/3 and 1/3) proportion

$$F_{\text{tr}, \text{fr}} = \frac{2}{3} F_b, \quad F_{\text{tr}, \text{rr}} = \frac{1}{3} F_b$$  \hspace{1cm} (5)  

The rolling resistance of the front and rear wheel is computed from the friction as a velocity dependent function and the static load distribution, see [12], as follows:

$$f_r(v) = 9 \cdot 10^{-3} + 7.2 \cdot 10^{-5} v + 5.038848 \cdot 10^{-10} v^4$$  \hspace{1cm} (6)  

$$F_{\text{tr}} = f_r(v) \frac{m l_{\text{fr}} g}{l_p + l_{\text{fr}}} \quad F_{\text{tr}, \text{rr}} = f_r(v) \frac{m l_{\text{fr}} g}{l_p + l_{\text{fr}}}$$  \hspace{1cm} (7)  

The longitudinal tire force at the front wheel consists of the braking force $F_{\text{tr}, \text{fr}}$ and the rolling resistance

$$F_{\text{fr}} = -F_{\text{tr}, \text{fr}}$$  \hspace{1cm} (8)  

Due to rear wheel drive, the longitudinal force at the rear wheel is extended by the total transmitted engine torque $M_w$ as the function of the effective engine torque $M^\mu_e$, the accelerator pedal’s position $\phi$ and the selected gear shift $\mu$ given by the following equations

$$F^\mu_{\text{tr}} = M_w(\phi, \mu) - F_{\text{tr}, \text{rr}}$$  \hspace{1cm} (9a)  
$$M_w(\phi, \mu) = \frac{i^\mu_g i_t}{R} M^\mu_e(\phi, \mu)$$  \hspace{1cm} (9b)  

where $i^\mu_g$ is the gearbox transmission ratio according to the selected gear, $i_t$ is the constant transmission ratio of the axle drive and $R$ is the radius of the wheel. The effective engine torque is transmitted twice and defined in [6] as follows:

$$\omega^\mu_e = \frac{i^\mu_g i_t v}{R} \quad \text{engine’s rotary frequency}$$  \hspace{1cm} (10a)  
$$f_1(\phi) = 1 - \exp(-3\phi)$$  \hspace{1cm} (10b)  
$$f_2(\omega^\mu_e) = -37.8 + 1.54 \omega^\mu_e - 0.0019(\omega^\mu_e)^2$$  \hspace{1cm} (10c)  
$$f_3(\omega^\mu_e) = -34.9 - 0.04775 \omega^\mu_e$$  \hspace{1cm} (10d)  
$$M^\mu_e(\phi, \mu) = f_1(\phi)f_2(\omega^\mu_e) + f_3(\omega^\mu_e)(1 - f_1(\phi))$$  \hspace{1cm} (10e)  

It is assumed that only longitudinal drag force acts on the vehicle (i.e. no side wind). Thus, the aerodynamic force due to air resistance can be determined by $F_{\text{ax}}$ as follows:

$$F_{\text{ax}} = \frac{1}{2} c_{w,p} A v^2, \quad F_{\text{ay}} = 0$$  \hspace{1cm} (11)  

The numerical parameters of the dynamical model are from [11] and contained in Table 1.

**B. Test path**

A test path between corridors is considered as a double-lane change maneuver. The driver is required to overtake a static obstacle by changing and returning to the car’s initial lane. The corridor is defined by cubic spline interpolating polynomials and the lower $P_l(x)$ and upper $P_u(x)$ path boundaries are shown in Figure 2(a) and 2(d). Details on similar test path can be found in [17]. For safety reasons, the car is restricted to move in the region of $B = 0.75$ m half-car width vertically from the path boundaries.
C. Dynamic optimal control problem

The aim is to drive the car as fast as possible through the test path by maintaining a smooth comfort level (i.e. minimal steering effort). A natural objective function to this problem is to test path by maintaining a smooth comfort level (i.e. minimal steering effort). A natural objective function to this problem is to minimize the time $t_f$ needed to complete the path and regulate the driver’s input $ω_b$. Thus, the resulting Mixed-Integer Optimal Control Problem (MIOCP) can be read as:

$$\begin{align*}
\min_{x(\cdot), u(\cdot)} & \quad t_f + \int_0^{t_f} \omega_b^2(t)dt \\
\text{s.t.} & \quad \dot{x}(t) = f(t, x(t), u(t)) \\
& \quad p_\phi(t) \in [P_l(p_x(t)) + B, P_u(p_x(t)) - B] \\
& \quad \omega_b(t) \in [-0.5, 0.5] \\
& \quad F_B(t) \in [0, 1.5 \cdot 10^4] \\
& \quad \phi(t) \in [0, 1] \\
& \quad \mu(t) \in \{1, 2, 3, 4\} \\
& \quad x(t_0) = (-30,\text{free},10,0,0,0,0) \\
& \quad p_x(t_f) = 140, \quad \psi(t_f) = 0
\end{align*}$$

(12a)

(12b)

(12c)

(12d)

(12e)

(12f)

(12g)

(12h)

(12i)

where $f$ is the ODE system described in (1). Dimensions are in SI units. The path boundary conditions are formulated in (12c), the constraints of the continuous control inputs are given in (12d–12f). The gear shift $μ$ in (12g) is considered to be an integer valued control. Initial and final values are defined by Eqs. (12h) and (12i). Notice, that the initial vertical position of the car can be chosen freely.

III. IMPLEMENTATION STRATEGIES

In this section the control design possibilities based on existing software systems are discussed and the novel design strategy is presented. To solve the time-optimal control problem introduced in Section II-C, the application of OPTI Toolbox [10] and the non-commercial, open source solvers IPOPT and BONMIN [2] was preferred. For further details on these software systems see [9], [1] and [14]. OPTI can solve static optimization problems with continuous $(x_i \in R^1)$, integer $(x_i \in Z)$ and binary $(x_i \in \{0, 1\})$ components.

The standard form of mixed-integer nonlinear program (MINLP) that can be solved by using OPTI is of type:

$$\begin{align*}
\min_x & \quad f(x) \\
\text{s.t.} & \quad Ax \leq b \\
& \quad A_eq x = b_eq \\
& \quad c(x) \leq d \\
& \quad c_eq(x) = d_eq \\
& \quad l_b \leq x \leq u \\
& \quad x_i \in Z \\
& \quad x_j \in \{0, 1\}
\end{align*}$$

(13a)

(13b)

(13c)

(13d)

(13e)

(13f)

(13g)

(13h)

In order to solve the dynamic optimization problem (12), it has to be transformed into the form of (13) using control and state discretization, see [5].

A. Discretization for varying final time

The final time varies, thus discretization for normalized interval $[0, 1]$ has to be performed in order to make the controls and states independent from $t_f$. The time transformation is:

$$t(\tau) = t_0 + h \tau, \quad h = 1/m$$

(14)

where $m$ is the number of grid points and $\tau_0 = 0 < \tau_1 < \cdots < \tau_{m-1} < \tau_m = 1$ is the equidistantly discretized interval with step size $h$. It follows that the time grid will be $t_i = h t_f$ for $i = 0, 1, \ldots, m$.

B. Direct multiple shooting

The time-optimal control problem of the vehicle motion is solved by using the direct multiple shooting method. The purpose of this technique is to transform the MIOCP problem (12) into a finite dimensional (mixed-integer) nonlinear optimization problem by discretization of the state and control functions on the time grid, first described in [4] and [8]. Since then it has become a popular part of professional software systems for optimum control design.

The origin of the strategy lies in the observation that for unstable or weakly damped systems the ODE solvers may have important errors which can further increase during the numerical optimization. Similarly, the initial and final values of the trajectory are often only partially defined, thus it is an extensive problem finding feasible initial solutions for starting the numerical optimization.

Hence, perturbations were introduced in the initial state at the rand of the time intervals in the grid to obtain by force, i.e.
by appropriately chosen additional equality constraints, that the entire state trajectory becomes a continuous solution in the progress of the numerical optimization.

In the sequel, the control inputs are considered to be piecewise constant functions and the time grid is assumed to be the same for both multiple shooting and control.

Consider the grid $G_m$ with subintervals $[t_i, t_{i+1}]$, $i = 0, \ldots, m - 1$. On each interval $[t_i, t_{i+1}]$ of the grid the solution $x_i(t)$ of the initial value problem with modified initial value $\sigma_i$ has to be found which satisfies

$$\dot{x}_i(t) = f(t, x_i(t), w_i), \quad \forall t \in [t_i, t_{i+1}]$$

(15)

where $w_i = (q_i^T, v_i^T)^T$ is the control signal in the actual time interval having the continuous part $q_i$ and integer part $v_i$.

This means that an initial value is shot out and the solution is determined that belongs to it in the interval. Denote with $x_i(t, \sigma_i, w)$ the solution of the initial value problem (IVP). Notice that for the integration a further subdivision of the interval $[t_i, t_{i+1}]$ has been introduced which was chosen to $n = 20$ in the application independently of the number of grid points $m$.

As a consequence, separate solutions can be obtained for each interval that are not necessarily connected continuously at the end of the intervals. Hence, an additional equality constraint has to be introduced for each interval guaranteeing continuity of the entire solution:

$$s_i = \sigma_{i+1} - x_i(t_{i+1}, \sigma_i, w_i) = 0, \quad i = 0, \ldots, m - 1$$

(16)

As a consequence, beside $w_i$, further discrete variables $\sigma = (\sigma_0, \ldots, \sigma_m)^T \in \mathbb{R}^{n_{x, (m+1)}}$ have to be optimized.

One speaks about a single shooting method if the shooting is applied only for one time instant, typically for the initial time and the initial value. Sometimes this approach is satisfactory, especially if good initial feasible approximations are available for the optimal solution.

Let $z = (t_f, s_0, \ldots, s_m, w_0, \ldots, w_{m-1})$, then the problem is of type

$$\min F(z) \quad \text{s.t.} \quad G(z) \leq 0, \quad H(z) = 0$$

(17)

C. Calculations of gradients and Jacobians

The applicability of OPTI needs the gradient $F'$ of the objective function and the Jacobians $G'$ and $H'$ of the constraints.

Let us here consider only the condition of trajectory joining as the most complex problem of computing the derivatives of the intermediate state in trajectory joining constraints. The derivative of nonlinear equality constraint (16) by $s_{i-1}$ is simple, however the derivative of $x(t_i)$ is much more complicated as will be illustrated. Notice that the derivatives are needed to the constraint Jacobians.

The question is how to determine the derivatives of $x(t, s, q)$ by the initial condition $s$, i.e. the shooting, and the parameter $q$, i.e. the actual control, assumed to be constant in the time interval.

$$S(t) := \frac{dx(t)}{ds}$$

\[ \frac{dS}{dt} = \frac{d}{dt} \frac{dx(t)}{ds} = \frac{d}{ds} \frac{dx(t)}{dt} = \frac{d}{ds} f(t, x(t), q) \]

$$\dot{S}(t) = f_x(t, x(t), q) \frac{dx(t)}{ds}$$

(18)

$$Q(t) := \frac{dx(t)}{dq}$$

\[ \frac{dQ}{dt} = \frac{d}{dt} \frac{dx(t)}{dq} = \frac{d}{dq} \frac{dx(t)}{dt} = \frac{d}{dq} f(t, x(t), q) \]

$$\dot{Q}(t) = f_x(t, x(t), q) \frac{dx(t)}{dq} + f_q(t, x(t), q)$$

(19)

The results can be collected in the following matrix differential equation:

$$\begin{bmatrix} \dot{S}(t) \\ \dot{Q}(t) \end{bmatrix} = \begin{bmatrix} f_x(t, x(t), q) & f_q(t, x(t), q) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} S(t) \\ Q(t) \end{bmatrix} \quad W(0) = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

(20)

Notice that $x(t_i)$ and $W(t_i)$ should be integrated numerically between $t_{i-1}$ and $t_i$. Because of the special structure of the matrix ODE, the Runge-Kutta method RK4 in matrix form can be applied.

The computation of the gradients and Jacobians are time consuming due to the large number of optimization variables, hence all of them are implemented in sparse forms.

D. Stored form of optimization variables

In the sequel MATLAB convention will be used in notation. The chosen storing concept is based on column-wise storing convention of matrices in MATLAB and easy and time optimal conversion between matrices and vectors.

Regarding the functions needed in the optimization process, the useful form for storing optimization variables in MATLAB would be a single scalar $\xi$ and a rectangular matrix $X$ structured as follows:

$$X = \begin{bmatrix} s_{0, 1} & s_{1, 1} & \cdots & s_{m-1, 1} & s_{m, 1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{0, 7} & s_{1, 7} & \cdots & s_{m-1, 7} & s_{m, 7} \\ q_{0, 1} & q_{1, 1} & \cdots & q_{m-1, 1} & \text{NaN} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ q_{0, 3} & q_{1, 3} & \cdots & q_{m-1, 3} & \text{NaN} \\ v_0 & v_1 & \cdots & v_{m-1} & \text{NaN} \end{bmatrix}$$

(21)

Hence, in each function participating in the optimization and using $x_i$, a matrix $X = \text{zeros}(11, m+1)$ and an augmented vector $xia = \text{zeros}(11*(m+1), 1)$ are defined. If OPTI
passes the optimization vector $x_i$ to the function then the conversion process is

$$t_f = x_i(1);$$
$$x_{ia}(:) = [x_i(2:end); NaN*ones(4,1)];$$
$$X(:) = x_{ia};$$

The backward conversion may be

$$x_{ia}(:) = X; x_i = [t_f; x_i(1:end-4)];$$

Notice that integer $v \in \mathbb{Z}$ is assumed here. If instead binary variables $w \in \{0, 1\}^4$ are used to encode the four values of $v$ then the above code can easily be modified.

IV. MAIN RESULTS

In this section the numerical results of the time optimal control problem are presented. The algorithm described in Section III was implemented in MATLAB with the software package OPTI for both integer and continuous NLP problems. The former was solved by using BONMIN with branch and bound and convex relaxation technique, the latter by using IPOPT with interior point filtered line search method. Cubic interpolating polynomial was applied for the relaxed gear shift $\mu$ in the derivative Jacobian structure between two succeeding integer values.

All computations were performed on a Windows 7 x64 based PC equipped with Intel Core i5 3.30 GHz processor and 8 GB of RAM. The optimization subproblems were solved to an integer and continuous tolerance level of $10^{-4}$ and $10^{-3}$ respectively. Numerical results were computed with grid points $m = 20$ and 40. Table II gives an overview of the number of variables and constraints.

The optimal paths, state trajectories and control inputs of the double lane changing maneuver are shown in Figure 2. As it can be seen, the solution of the discretized motion of the car satisfies all path, state and the multi-shooting trajectory joint constraints. The vehicle accelerates in the entire time interval as it was expected for time optimal solution. The optimal gear shifts (integer problem) increases in time of 0.67, 2.69, 6.38 sec and 0.84, 2.69, 6.38 sec for $m = 20$ and 40 respectively. In all cases, the optimization resulted $\phi = 1$ full acceleration and low 5–10 N braking force, within numerical precision, similarly the side-slip angle remained between the prescribed constraints (not shown in Figure 2).

The path and state trajectories of the MINLP and NLP problems are almost identical, although the control signals are slightly different.

In Table III the numerical results of the integer and continuous problems is summarized. Notice that the MINLP resulted lower final time than for NLP in case of $m = 40$ grid points respectively.

---

**TABLE II**

<table>
<thead>
<tr>
<th>$m$</th>
<th>$N_{var}$</th>
<th>$N_{term}$</th>
<th>$N_{eq}$</th>
<th>$N_{ineq}$</th>
<th>$N_{bound}$</th>
<th>$N_{sum}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>11m + 8</td>
<td>8</td>
<td>7$m$ + 2</td>
<td>2$m$ + 2</td>
<td>22$m$ + 16</td>
<td>31$m$ + 26</td>
</tr>
<tr>
<td>40</td>
<td>228</td>
<td>8</td>
<td>140</td>
<td>42</td>
<td>456</td>
<td>646</td>
</tr>
</tbody>
</table>

---

**Fig. 2.** Solution of the mixed-integer (circle) and nonlinear problem (cross) with grid points $m = 20$ and 40
which seems to be a contradiction. Similar behavior was also detected in [6].

The reason may be found in the local optimum character of deterministic optimization and slightly different shooting values. Notice however, that the use of multiple shooting tries to bring the solution near to the global optimum.

Multiple shooting is partly related to stochastic optimization bringing continuous modifications into the state trajectory and control, but without the need of a large number of individuals of the population.

The number of test path parameters is $5 + 3 = 8$, hence a grid of parameters can be chosen and for all grid points (i.e. path) the time optimal control trajectory can be determined offline and stored in a database. For a given situation of lane change or collision avoidance in the presence of static obstacle the nearest solution in the database can be selected, perhaps interpolation can be involved and the control trajectory can be performed in real-time.

If the state trajectory is also stored in the database then it can be applied in the case of additional moving obstacle to check in real-time whether the obstacle avoidance can be performed or emergency braking is necessary.

V. CONCLUSION

In this paper, we investigated the extensively researched mixed-integer time optimal control problem for ground vehicles. The MIOCP was solved by using time discretization and multiple shooting method. The approach was implemented with the aid of non-commercial software packages that substantially use the gradients of the objective function and the Jacobians of the constraints. For the latter, we presented a novel algorithm for computing the derivatives of the complex trajectory joining equations. This algorithm was given in the form of matrix differential equations whose structure allowed to compute their solution using RK4 in matrix form. Because of the large number of variables the gradients and Jacobians were performed in sparse form. Our solution showed competitive results to the ones obtained by closed source and state of the art commercial solvers.

Future research will concentrate on the extension of the presented algorithm to moving horizon predictive control approaches as well as arbitrary path constraints and different car models.

ACKNOWLEDGEMENT

The work was supported by the European Union and Hungary in the project TÁMOP-4.2.2.A-11/1/KONV-2012-0012: "Basic research for the development of hybrid and electric vehicles".