2015 European Control Conference (ECC)
July 15-17, 2015. Linz, Austria

Clothoid-Based Model Predictive Control for Autonomous Driving

Pedro F. Lima[1], Marco Trincavelli[2], Jonas Mårtensson[1], and Bo Wahlberg[1]

Abstract— **This paper presents a novel linear time-varying model predictive controller (LTV-MPC) using a sparse clothoid-based path description: a LTV-MPCC. Clothoids are used world-wide in road design since they allow smooth driving associated with low jerk values. The formulation of the MPC controller is based on the fact that the path of a vehicle traveling at low speeds defines a segment of clothoids if the steering angle is chosen to vary piecewise linearly. Therefore, we can compute the vehicle motion as clothoid parameters and translate them to vehicle inputs. We present simulation results that demonstrate the ability of the controller to produce a very comfortable and smooth driving while maintaining a tracking accuracy comparable to that of a regular LTV-MPC. While the regular MPC controllers use path descriptions where waypoints are close to each other, our LTV-MPCC has the ability of using paths described by very sparse waypoints. In this case, each pair of waypoints describes a clothoid segment and the cost function minimization is performed in a more efficient way allowing larger prediction distances to be used. This paper also presents a novel algorithm that addresses the problem of path sparsification using a reduced number of clothoid segments. The path sparsification enables a path description using few waypoints with almost no loss of detail. The detail of the reconstruction is an adjustable parameter of the algorithm. The higher the required detail, the more clothoid segments are used.**

I. INTRODUCTION

The idea of autonomous vehicles has been present for more than 50 years [1]. Yet, only in the last few decades has technology made it possible and a big progress has been made in order to help the driver in decision making. Some examples are the advanced driver assistance systems (ADAS) such as anti-lock braking system (ABS), electronic stability control (ESC) and adaptive cruise control (ACC) that help the driver by intervening in critical situations. Also, the need for increased efficiency in road utilization, fuel consumption and goods transportation [2] makes autonomous vehicles a very attractive solution.

While today the transition from driver control to autonomous vehicles is performed mostly in urban environments, autonomous vehicles with a much more specialized application in closed environments such as gravel pits, mining areas, construction sites, and loading terminals are expected to emerge. In urban environments, a substantial investment and expansion in infrastructure and law adaption are required for autonomous driving, whereas these are more easily adjusted in close environments. The automation of vehicles in these closed environments is a huge opportunity since it is a very cost efficient option and it eliminates repetitive jobs that can lead to inattention and accidents, resulting in an efficiency and production increase.

One of the main challenges of autonomous vehicles is the actual control of the vehicle. Model predictive control (MPC) is a very powerful approach that can address the autonomous vehicle guidance problem [3]–[5] due to its ability to predict the vehicle behavior for a given set of inputs. Moreover, it handles nonlinear time-varying models and constraints in a quite methodical manner. In MPC, a chosen cost function is minimized and the optimal sequence of inputs is computed in order to better follow a given trajectory under given constraints on states and inputs. An input subset of the optimal sequence is applied to the vehicle and the process is then repeated. In [3], a control architecture based on a linear time-varying MPC (LTV-MPC) is presented to address the lane keeping and obstacle avoidance problems for a passenger car driving on low curvature roads. In [4], a nonlinear MPC controller (NMPC) is presented in order to stabilize a vehicle along a desired path while fulfilling its physical constraints. Since a nonlinear vehicle model is considered the resulting MPC controller has a enormous computational burden which makes it difficult to be implemented in real-time. In [5], a hierarchical framework based on MPC for autonomous vehicles is presented. At the high-level MPC, a desired trajectory is computed online using simplified models, which is then given to a low-level MPC that computes the vehicle inputs using a detailed nonlinear vehicle model.

As in [4], MPC controllers limit experimental validation due to the utilization of complex models for predicting the vehicle behavior, which makes the optimization complex and slow. Also, adding constraints, such as imposing certain comfort measures, increases the problem complexity and the computational burden.

In this paper, we formulate an MPC controller that provides a smooth driving to the autonomous vehicle by taking advantage of a special type of curves: clothoids. We propose a novel clothoid-based LTV-MPC controller (LTV-MPCC). Clothoids are used for road design since their curvature vary linearly with the path arc-length. This prevents sudden changes in lateral acceleration and therefore yields low values of lateral jerk. With our approach, we are able to compute smooth driving controls while predicting the vehicle behavior

[1]School of Electrical Engineering, ACCESS Linnaeus Centre, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. pfrdal@kth.se, jonas1@kth.se, bo@kth.se

[2]Research and Development, Scania CV AB, 151 87 Södertälje, Sweden, marco.trincavelli@scania.com

The research presented in this paper was funded by iQMatic which is a project led by Scania CV AB together with Saab, Autoliv, Linköping University and KTH The Royal Institute of Technology. iQMatic has been financed by Vinnova (FFI) and by Scania CV AB.

along a long prediction distance. We assume that the motion of a vehicle traveling at low speeds (such that the lateral vehicle dynamics are negligible, i.e. until speeds around 15 $\mathrm{m\,s^{-1}}$) defines a segment of clothoids if the steering angle is chosen to vary piecewise linearly. To describe the vehicle behavior between each pair of waypoints, it is enough to evaluate the clothoid parameters at those waypoints. Furthermore, we can relate the desired curvature with the desired vehicle steering angle, making it possible for lateral control of the vehicle. One of the main advantages of this approach is that the search space of the optimization problem is small. This is due to that we only need to compute the input of the vehicle in each clothoid starting point, allowing us to have a long prediction distance (with a small prediction horizon) with no significant increase of computational time and still maintain a good accuracy when compared with a regular LTV-MPC.

In this paper, we also deal with the reference generation for such controller by fitting a number of clothoids to recorded data. The problem of clothoid fitting and sketching has been subject of research in recent years [6], [7]. In [6], an algorithm for fitting a clothoid sequence with G^2 continuity to a curve is proposed where a clothoid rational approximation is used. In [7], a nonlinear algorithm for clothoid fitting between two given points is proposed using the Newton-Raphson method in a fast and robust fashion. We therefore propose a novel path sparsification algorithm by addressing the problem of clothoid fitting using l_0-norm minimization techniques. This approach has found itself to be very useful in terms of path compression and description since it is able to represent a path with few points with almost no loss of detail. The waypoints contain the clothoid parameters allowing an efficient and fast curve reconstruction and utilization.

The rest of this paper is organized as follows: in Section II, we address the problem of clothoid-based path following using an MPC framework by developing a LTV-MPCC for the lateral control of a vehicle; in Section III, we introduce the problem of path sparsification and detail how it can be solved using l_0-norm minimization techniques; in Section IV, we demonstrate the performance of the proposed controller through simulations using a validated vehicle model and real path references; finally, in Section V, we provide some concluding remarks and outline future work.

II. CLOTHOID-BASED MPC

In this section, we formulate a clothoid-based MPC as LTV-MPCC by linearizing the vehicle model for low-speeds around the reference clothoid-based trajectory.

A. Problem formulation

The movement of a car-like nonholonomic vehicle at low speeds, i.e. when the lateral dynamics have little influence, can approximately be described by its time-domain kinematic equations [8]

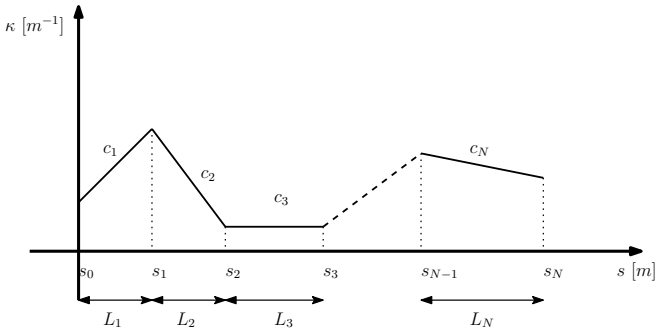

Fig. 1. Piecewise linear curvature function. Each i-th clothoid segment is described by a pair of kink points, by its curvature slope c_i, and arc length L_i.

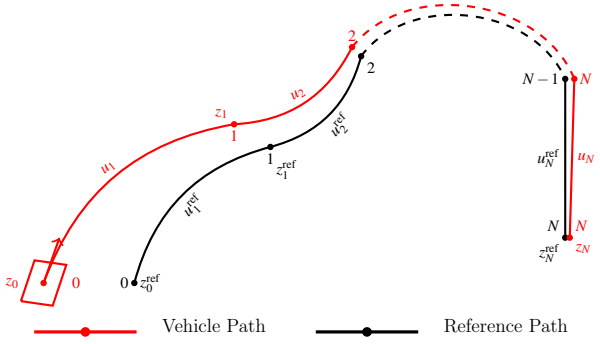

Fig. 2. The path is represented by $N-1$ clothoids which are described by N kink points. The clothoid-based MPC controller results in a sequence of inputs that will make the vehicle follow the reference path.

$$\frac{dx(t)}{dt} = v(t)\cos(\theta(t)) \tag{1a}$$
$$\frac{dy(t)}{dt} = v(t)\sin(\theta(t)) \tag{1b}$$
$$\frac{d\theta(t)}{dt} = \frac{v(t)}{D}\tan(\delta(t)), \tag{1c}$$

where x and y are the coordinates of the vehicle in the global coordinate system, θ is the yaw angle, D is the distance between the front and rear axle, v is the longitudinal velocity in the vehicle coordinate system, and δ is the steering angle of the front wheels. We can translate these equations to space-domain by making $v(t)\cdot \mathrm{dt} = \mathrm{ds}$ and assuming that $v(t) \neq 0$ and $v(t)$ is a continuous function as

$$\frac{dx(s)}{ds} = \cos(\theta(s)) \tag{2a}$$
$$\frac{dy(s)}{ds} = \sin(\theta(s)) \tag{2b}$$
$$\frac{d\theta(s)}{ds} = \frac{1}{D}\tan(\delta(s)). \tag{2c}$$

Let $\frac{d\theta(s)}{ds} = \kappa(s)$ where $\kappa(s)$ is the vehicle curvature along the path. If we limit $\kappa(s)$ to be a linearly varying function, i.e. $\kappa(s) = cs + \kappa(0)$ then (2) describes a clothoid. Let our path be represented by $N-1$ clothoid

segments which are described by N waypoints (or kink points) such that the curvature is a piecewise linear function as shown in Fig. 1. The vehicle state at each kink-point is defined as $\mathbf{z}(s_i) = \mathbf{z}_i = [x_i, y_i, \theta_i, \kappa_i]^\intercal$ and input is defined as $\mathbf{u}(s_i) = \mathbf{u}_i = [c_i, L_i]^\intercal$ where c_i and L_i are the i-th clothoid curvature slope (constant by definition) and arc-length respectively. Thus, using (2) and assuming $\kappa(s)$ to be a linearly varying function, the vehicle state evolution at each kink point is computed using the discrete function $f(\mathbf{z}_i, \mathbf{u_i})$ which is defined as

$$x_{i+1} = x_i + \int_{s_i}^{s_{i+1}} \cos(\theta_i + \kappa_i(s-s_i) + c_{i+1}\frac{(s-s_i)^2}{2})\ \mathrm{ds} \tag{3a}$$

$$y_{i+1} = y_i + \int_{s_i}^{s_{i+1}} \sin(\theta_i + \kappa_i(s-s_i) + c_{i+1}\frac{(s-s_i)^2}{2})\ \mathrm{ds} \tag{3b}$$

$$\theta_{i+1} = \theta_i + \kappa_i L_{i+1} + c_{i+1}\frac{{L_{i+1}}^2}{2} \tag{3c}$$

$$\kappa_{i+1} = \kappa_i + c_{i+1}L_{i+1}, \tag{3d}$$

where $i = 1, .., N$.

We want to use only the kink points information to control a car-like nonholonomic vehicle to follow the path as idealized in Fig. 2. We can formulate this problem as a standard MPC problem

$$\min_{\tilde{\mathbf{u}}_1,\ldots,\tilde{\mathbf{u}}_H} \quad \sum_{i=1}^{H} \tilde{\mathbf{z}}_i^\intercal \mathbf{Q}\tilde{\mathbf{z}}_i + \tilde{\mathbf{u}}_{i-1}^\intercal \mathbf{R}\tilde{\mathbf{u}}_{i-1} \tag{4a}$$

$$\text{s.t.} \quad \mathbf{z}_{i+1} = f(\mathbf{z}_i, \mathbf{u_i}),\ \mathbf{z}_0 = \mathbf{z}(s_0) \tag{4b}$$

$$\tilde{\mathbf{u}}_\mathbf{i} \in \mathbb{U}, \tag{4c}$$

where $H \geq 1$ is the MPC prediction horizon, $\tilde{\mathbf{z}}_i = \mathbf{z}_i - \mathbf{z}_i^{\text{ref}}$, $\tilde{\mathbf{u}}_i = \mathbf{u}_i - \mathbf{u}_i^{\text{ref}}$, $\mathbf{Q} \succ 0$ and $\mathbf{R} \succ 0$ are the state penalization matrix and the input penalization matrix, respectively and they are chosen to be diagonal. Intuitively, we are trying to find the set of clothoids that will drive the vehicle onto the desired path while minimizing the state error (position, orientation and curvature) at the kink points and the input error (curvature slope and arc-length).

B. Linearization

We want to be able to explicitly compute the position of the kink points of the MPC output path without the need of computing the integrals in (3a) and (3b). These integrals are known as the Fresnel integrals which do not have an explicit expression [9]. A solution is to linearize the MPC output path around the reference path. First, we rewrite (3) by substituting the integral by its corresponding arc length

$$x_{i+1} = x_i + L_{i+1}\cos(\theta_i + \kappa_i L_{i+1} + c_{i+1}\frac{{L_{i+1}}^2}{2}) \tag{5a}$$

$$y_{i+1} = y_i + L_{i+1}\sin(\theta_i + \kappa_i L_{i+1} + c_{i+1}\frac{{L_{i+1}}^2}{2}) \tag{5b}$$

$$\theta_{i+1} = \theta_i + \kappa_i L_{i+1} + c_{i+1}\frac{{L_{i+1}}^2}{2} \tag{5c}$$

$$\kappa_{i+1} = \kappa_i + c_{i+1}L_{i+1}, \tag{5d}$$

and then the linearization takes the form

$$\tilde{\mathbf{z}}_{i+1} = \mathbf{A}_i\tilde{\mathbf{z}}_i + \mathbf{B}_i\tilde{\mathbf{u}}_i, \tag{6}$$

where

$$\mathbf{A}_i = \left.\frac{\partial f(\mathbf{z_i}, \mathbf{u_i})}{\partial \mathbf{z_i}}\right|_{\substack{\mathbf{z_i}=\mathbf{z_i}^{\text{ref}},\\ \mathbf{u_i}=\mathbf{u_i}^{\text{ref}}}} =$$

$$\begin{bmatrix} 1 & 0 & -L_{i+1}^{\text{ref}}\sin\theta_{i+1}^{\text{ref}} & -{L_{i+1}^{\text{ref}}}^2\sin\theta_{i+1}^{\text{ref}} \\ 0 & 1 & L_{i+1}^{\text{ref}}\cos\theta_{i+1}^{\text{ref}} & -{L_{i+1}^{\text{ref}}}^2\cos\theta_{i+1}^{\text{ref}} \\ 0 & 0 & 1 & L_{i+1}^{\text{ref}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

and

$$\mathbf{B}_i = \left.\frac{\partial f(\mathbf{z_i}, \mathbf{u_i})}{\partial \mathbf{u_i}}\right|_{\substack{\mathbf{z_i}=\mathbf{z_i}^{\text{ref}},\\ \mathbf{u_i}=\mathbf{u_i}^{\text{ref}}}} =$$

$$\begin{bmatrix} -\frac{{L_{i+1}^{\text{ref}}}^3}{2}\sin\theta_{i+1}^{\text{ref}} & \cos\theta_{i+1}^{\text{ref}} - L_{i+1}^{\text{ref}}\kappa_{i+1}^{\text{ref}}\sin\theta_{i+1}^{\text{ref}} \\ \frac{{L_{i+1}^{\text{ref}}}^3}{2}\cos\theta_{i+1}^{\text{ref}} & \sin\theta_{i+1}^{\text{ref}} + L_{i+1}^{\text{ref}}\kappa_{i+1}^{\text{ref}}\cos\theta_{i+1}^{\text{ref}} \\ \frac{{L_{i+1}^{\text{ref}}}^2}{2} & \kappa_{i+1}^{\text{ref}} \\ L_{i+1}^{\text{ref}} & c_{i+1}^{\text{ref}} \end{bmatrix}. \tag{8}$$

Now, it is possible to reformulate the MPC problem (4) in the usual quadratic programming (QP) form. We introduce the following vectors

$$\tilde{\mathbf{Z}}_{i+1} = \begin{bmatrix} \tilde{\mathbf{z}}_{i+1|i} \\ \tilde{\mathbf{z}}_{i+2|i} \\ \vdots \\ \tilde{\mathbf{z}}_{i+H|i} \end{bmatrix} \text{ and } \quad \tilde{\mathbf{U}}_i = \begin{bmatrix} \tilde{\mathbf{u}}_{i|i} \\ \tilde{\mathbf{u}}_{i+1|i} \\ \vdots \\ \tilde{\mathbf{u}}_{i+H-1|i} \end{bmatrix}, \tag{9}$$

where the notation $\tilde{\mathbf{z}}_{j|i}$ indicates the estimated state error at instant j predicted at instant i, and $\tilde{\mathbf{u}}_{j|i}$ indicates the optimal input error at instant j predicted at instant i.

Thus, it is possible to write the MPC cost function as

$$\Phi_i = \tilde{\mathbf{Z}}_{i+1}^\intercal \bar{\mathbf{Q}}\tilde{\mathbf{Z}}_{i+1} + \tilde{\mathbf{U}}_i^\intercal \bar{\mathbf{R}}\tilde{\mathbf{U}}_i, \tag{10}$$

with $\bar{\mathbf{Q}} = \text{diag}(\mathbf{Q}, \ldots, \mathbf{Q})$ and $\bar{\mathbf{R}} = \text{diag}(\mathbf{R}, \ldots, \mathbf{R})$.

Furthermore, (6) can be rewritten as a function of the initial state error $\tilde{\mathbf{z}}_{i|i}$

$$\tilde{\mathbf{Z}}_{i+1} = \tilde{\mathbf{A}}_i\tilde{\mathbf{z}}_{i|i} + \tilde{\mathbf{B}}_i\tilde{\mathbf{U}}_i, \tag{11}$$

where

$$\tilde{\mathbf{A}}_i = \begin{bmatrix} \mathbf{A}_i \\ \mathbf{A}_{i+1}\mathbf{A}_i \\ \vdots \\ \mathbf{A}_{H-1}\ldots\mathbf{A}_{i+1}\mathbf{A}_i \\ \mathbf{A}_H\mathbf{A}_{H-1}\ldots\mathbf{A}_{i+1}\mathbf{A}_i \end{bmatrix} \tag{12}$$

and

$$\tilde{\mathbf{B}}_i = \begin{bmatrix} \mathbf{B}_i & 0 & \ldots & 0 \\ \mathbf{A}_{i+1}\mathbf{B}_i & \mathbf{B}_{i+1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{H-1}\ldots\mathbf{A}_{i+1}\mathbf{B}_i & \mathbf{A}_{H-1}\ldots\mathbf{A}_{i+2}\mathbf{B}_{i+1} & \ldots & 0 \\ \mathbf{A}_H\mathbf{A}_{H-1}\ldots\mathbf{A}_{i+1}\mathbf{B}_i & \mathbf{A}_H\mathbf{A}_{H-1}\ldots\mathbf{A}_{i+2}\mathbf{B}_{i+1} & \ldots & \mathbf{B}_{H-1} \end{bmatrix}. \tag{13}$$

Manipulating algebraically (10), we can rewrite the objective function into a standard QP form

$$\Phi_i = \frac{1}{2}\tilde{\mathbf{U}}_i^\intercal \mathbf{H}_i \tilde{\mathbf{U}}_i + \mathbf{f}_i^\intercal \tilde{\mathbf{U}}_i + \mathbf{d}_i, \tag{14}$$

with

$$\mathbf{H}_i = 2(\tilde{\mathbf{B}}_i^\intercal \bar{\mathbf{Q}} \tilde{\mathbf{B}}_i + \bar{\mathbf{R}}) \tag{15a}$$

$$\mathbf{f}_i = 2\tilde{\mathbf{B}}_i^\intercal \bar{\mathbf{Q}} \tilde{\mathbf{A}}_i \tilde{\mathbf{z}}_{i|i} \tag{15b}$$

$$\mathbf{d}_i = \tilde{\mathbf{z}}_{i|i}^\intercal \tilde{\mathbf{A}}_i^\intercal \bar{\mathbf{Q}} \tilde{\mathbf{A}}_i \tilde{\mathbf{z}}_{i|i}. \tag{15c}$$

C. LTV-MPCC formulation

Finally, the MPC problem (4) can be formulated as a LTV-MPCC

$$\min_{\tilde{\mathbf{U}}_\mathbf{i}} \quad \frac{1}{2}\tilde{\mathbf{U}}_i^\intercal \mathbf{H}_i \tilde{\mathbf{U}}_i + \mathbf{f}_i^\intercal \tilde{\mathbf{U}}_i \tag{16a}$$

$$\text{s.t.} \quad \tilde{\mathbf{U}}_\mathbf{i} \in \mathbb{U}. \tag{16b}$$

The constraint $\tilde{\mathbf{U}}_\mathbf{i} \in \mathbb{U}$ expresses the existence of upper and lower bounds on the input values. In this problem we consider that $L_{\min} \leq L_i \leq L_{\max}$ and $c_{\min} \leq c_i \leq c_{\max}$. These constraints can be written in matrix form

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{-I} \end{bmatrix} \tilde{\mathbf{U}}_\mathbf{i} = \begin{bmatrix} \mathbf{u}_{\max} - \mathbf{u}_i^{\text{ref}} \\ -\mathbf{u}_{\min} + \mathbf{u}_i^{\text{ref}} \end{bmatrix}, \tag{17}$$

where $\mathbf{u}_{\min} = [L_{\min}, c_{\min}]^\intercal$ and $\mathbf{u}_{\max} = [L_{\max}, c_{\max}]^\intercal$.

Assuming that the vehicle's state can be measured each T_s seconds we can recompute the MPC solution and apply the first optimal input. Since the MPC formulation (16) relies on a linearization around the reference trajectory, we need to find a suitable point in the reference path to relinearize the vehicle path at each step. Assuming that the vehicle starts and will always maintain itself in the vicinity of the reference path, we can compute the new reference point on the reference path by the method exemplified graphically in Fig. 3. The distance that the vehicle has traveled along the reference path since the last sample is approximated by

$$l = d\cos(\alpha), \tag{18}$$

where α and d are the angle and the Euclidean distance, respectively, between the last reference point on the path used $(x,y)^{\text{ref}}_{\text{prev}}$ and the vehicle position (x,y). The new point $(x,y)^{\text{ref}}_{\text{new}}$ is found by traveling a distance l along the reference path starting in $(x,y)^{\text{ref}}_{\text{prev}}$. The first clothoid segment of the vehicle path is linearized around the clothoid defined between $(x,y)^{\text{ref}}_{\text{new}}$ and the closest clothoid kink point ahead.

III. PATH SPARSIFICATION

In order to use a clothoid-based MPC, we have to generate and feed the controller with a clothoid-based reference path. A path is usually described using a dense concentration of linearly interpolated points. We developed a strategy that allows us to sparsify the path description with almost no loss of detail as depicted in Fig. 4.

The problem of sparsification consists in describing a set of raw points $(x_{\text{raw}}, y_{\text{raw}}, \theta_{\text{raw}})$ using a set of clothoids. The path described by these set of clothoids must not deviate more than a certain small tolerance ε from the original path. The clothoids can be uniquely described using their kink points $(x_{\text{clot}}, y_{\text{clot}}, \theta_{\text{clot}}, \kappa_{\text{clot}}, L_{\text{clot}})$, i.e. the beginning and the end of a clothoid. The raw points contain the position and orientation of the vehicle while the kink points extend this description to include the curvature at each point κ_{clot} and the distance until the next kink point L_{clot}. Furthermore, we intend to describe the original path with as few clothoid segments as possible. The problem of filtering the curvature function with the minimum number of linear functions involves the l_0-norm. Such problems are NP-hard and therefore very difficult to solve due to their computational burden. Thus, the problem we propose to solve is the following

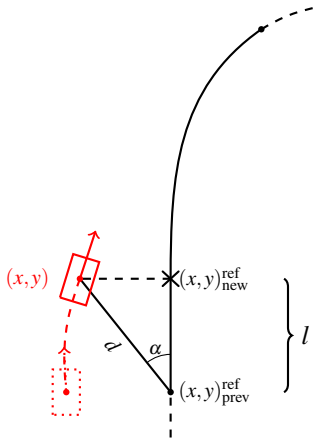

Fig. 3. It is assumed that the vehicle is always in the vicinity of the reference path. Thus, the distance traveled along the path can be computed using a simple trigonometric expression.

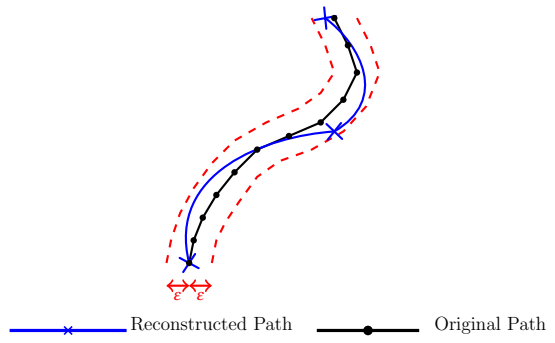

Fig. 4. The reference path is a sequence of GPS waypoints very close to each other and linearly interpolated between each other. The reconstructed path is a sequence of waypoints that handle the description of a clothoid pairwise. This path has a maximum deviation ε from the original one.

$$\min_{\kappa} \quad ||\mathbf{D}\kappa||_0 \tag{19a}$$

$$\text{s. t.} \quad |\mathbf{x}(\kappa) - \mathbf{x}_{\text{raw}}| \leq \varepsilon \tag{19b}$$

$$|\mathbf{y}(\kappa) - \mathbf{y}_{\text{raw}}| \leq \varepsilon, \tag{19c}$$

where κ is the piecewise linear curvature function to be estimated, ε is the maximum allowed error of the reconstructed curve, $\mathbf{D}$ is the matrix operator that calculates second order differences of a vector

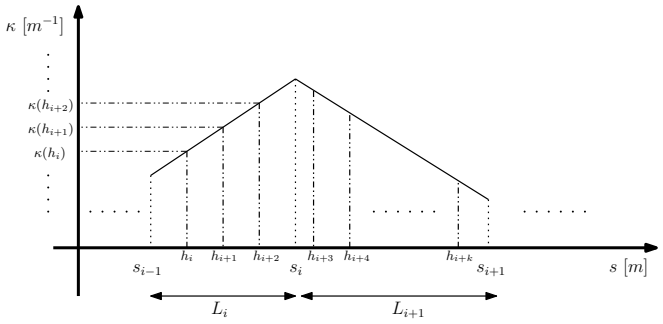

Fig. 5. Exemplification of how the curvature function $\kappa(s)$ is discretized each h_i steps.

$$\mathbf{D} = \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \end{bmatrix},$$

and $\mathbf{x}(\boldsymbol{\kappa})$ and $\mathbf{y}(\boldsymbol{\kappa})$ are the reconstructed path coordinates using (3a) and (3b) defining $\mathbf{x}(\boldsymbol{\kappa}) = [x_1, x_2, \ldots, x_{N_h}]^\intercal$ and $\mathbf{y}(\boldsymbol{\kappa}) = [y_1, y_2, \ldots, y_{N_h}]^\intercal$ where N_h is the number of discretization points. Note that the coordinates of the clothoid-based path $(\mathbf{x}, \mathbf{y})$ are only dependent on the clothoid curvature $\boldsymbol{\kappa}$ and that the inequalities (19b) and (19c) are performed elementwise. The discretization of (3a) and (3b) is based on the fact that integrals can be approximated by Riemann sums. Thus,

$$x(\boldsymbol{\kappa}) = \sum_{i=0}^{N_h} \cos\left(\sum_{j=0}^{i} \boldsymbol{\kappa}(h_j)\Delta h_j\right)\Delta h_i \tag{20a}$$

$$y(\boldsymbol{\kappa}) = \sum_{i=0}^{N_h} \sin\left(\sum_{j=0}^{i} \boldsymbol{\kappa}(h_j)\Delta h_j\right)\Delta h_i, \tag{20b}$$

where $\Delta h_i = h_{i+1} - h_i$ and h_i are the curvature function abscissa values as exemplified in Fig. 5. The equivalent vectorial version is

$$\mathbf{x}(\boldsymbol{\kappa}) = \mathbf{S}\cos(\mathbf{S}\boldsymbol{\kappa}) \tag{21a}$$

$$\mathbf{y}(\boldsymbol{\kappa}) = \mathbf{S}\sin(\mathbf{S}\boldsymbol{\kappa}), \tag{21b}$$

where $\mathbf{S}$ is a lower triangular matrix that performs the cumulative sum operation along a vector

$$\mathbf{S} = \begin{bmatrix} \Delta h_0 & & & & \\ \Delta h_0 & \Delta h_1 & & & \\ \Delta h_0 & \Delta h_1 & \Delta h_2 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \Delta h_0 & \Delta h_1 & \Delta h_2 & \ldots & \Delta h_{N_h-1} \end{bmatrix}.$$

For a piecewise linear function, the smoothness term, i.e. $||\mathbf{D}\boldsymbol{\kappa}||_0$, is exactly zero everywhere except in a few points, the so called kink points. This strategy induces sparsity in the solution meaning that we will end up with a function that is piecewise linear or, in other words, linear everywhere except at the kink points. These type of problems are usually relaxed using the l_1-norm which normally results in convex relaxations of the original problem. As proposed in [10], a method to solve these problems is applying an algorithm from the family of Majorization-Minimization (MM) algorithms, where the minimization of a generic function (l_0-norm) is performed by iteratively minimizing a convex majorizer function (l_1-norm)[1]. To accomplish this, we have to rewrite the problem (19) as

$$\begin{aligned} \min_{\boldsymbol{\kappa}} \quad & ||\mathbf{W} \odot \mathbf{D}\boldsymbol{\kappa}||_1 \\ \text{s.t.} \quad & |\mathbf{x}(\boldsymbol{\kappa}) - \mathbf{x}_{\text{raw}}| \leq \varepsilon \\ & |\mathbf{y}(\boldsymbol{\kappa}) - \mathbf{y}_{\text{raw}}| \leq \varepsilon, \end{aligned} \tag{22}$$

where $\mathbf{W}$ is a vector that contains weights that are initially set to 1 and $\odot$ is the Hadamard product. $\mathbf{W}$ is updated in each iteration as

$$\mathbf{W}_{j+1} = \frac{1}{|\mathbf{W}_j \odot \mathbf{D}\boldsymbol{\kappa}|}. \tag{23}$$

At each iteration, the weighting vector is normalized so that the weights sum up to the number of samples. Intuitively, the weights are updated such that the entries of the smoothness term close to zero receive a higher weight, while the others obtain a smaller weight resulting in a reweighed l_1-norm that is more similar to the l_0-norm.

However, problem (22) is not in convex form due to the fact that the reconstructed path $(\mathbf{x}(\boldsymbol{\kappa}), \mathbf{y}(\boldsymbol{\kappa}))$ is computed using (21) which are clearly nonlinear equations. A linear approximation is done using a first-order Taylor approximation of both cosine and sine around an estimated curvature $\hat{\boldsymbol{\kappa}}$. To be able to apply such approximation we have to estimate the original path curvature and discretize its function in Δh_i different parts for $i \in 0, ..., N_h$, where Δh_i is the distance between the i-th and $(i+1)$-th raw points. The linearization takes the form

$$\mathbf{x}(\boldsymbol{\kappa}) = \mathbf{S}\left(\cos(\mathbf{S}\hat{\boldsymbol{\kappa}}) - \sin(\mathbf{S}\hat{\boldsymbol{\kappa}})\mathbf{S}(\boldsymbol{\kappa} - \hat{\boldsymbol{\kappa}})\right) \tag{24a}$$

$$\mathbf{y}(\boldsymbol{\kappa}) = \mathbf{S}\left(\sin(\mathbf{S}\hat{\boldsymbol{\kappa}}) + \cos(\mathbf{S}\hat{\boldsymbol{\kappa}})\mathbf{S}(\boldsymbol{\kappa} - \hat{\boldsymbol{\kappa}})\right), \tag{24b}$$

where estimated curvature $\hat{\kappa}_k$ is found by approximating a circle passing by three consecutive points P_{k-1}, P_k and P_{k+1} of the raw data. Then the inverse of the circle radius is the curvature approximation. Thus, as described in [11], the discrete curvature approximation is given by

$$\hat{\boldsymbol{\kappa}} = \frac{4A_\Delta}{P_\Delta}, \tag{25}$$

where A_Δ and P_Δ are the area and perimeter of the triangle formed by the considered three points, respectively.

Finally, we can formulate the original problem (19) as a convex problem

[1]As discussed in [10], there are several properties of this algorithm that need more research and understanding. In this paper, the algorithm is analysed from a practical point of view and we do not claim any specific property such as convergence or optimality.

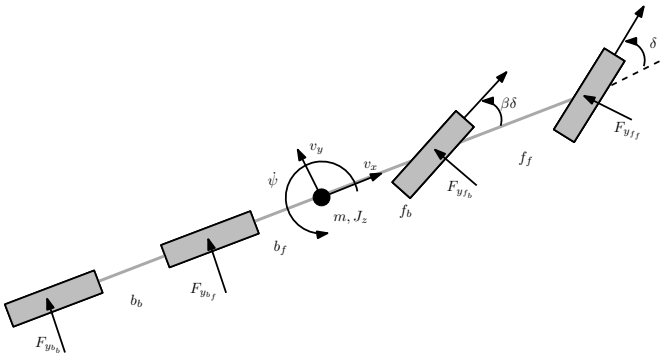

Fig. 6. 4-axles bicycle model illustration.

$$\min_{\boldsymbol{\kappa}} \quad ||\mathbf{W} \odot \mathbf{D}\boldsymbol{\kappa}||_1 \tag{26a}$$

$$\text{s.t.} \quad |\mathbf{S}(\cos(\mathbf{S}\hat{\boldsymbol{\kappa}}) - \sin(\mathbf{S}\hat{\boldsymbol{\kappa}})\mathbf{S}(\boldsymbol{\kappa} - \hat{\boldsymbol{\kappa}})) - \mathbf{x}_{\text{raw}}| \leq \varepsilon \tag{26b}$$

$$|\mathbf{S}(\sin(\mathbf{S}\hat{\boldsymbol{\kappa}}) + \cos(\mathbf{S}\hat{\boldsymbol{\kappa}})\mathbf{S}(\boldsymbol{\kappa} - \hat{\boldsymbol{\kappa}})) - \mathbf{y}_{\text{raw}}| \leq \varepsilon. \tag{26c}$$

IV. SIMULATION RESULTS

In this section, we present the main results obtained with the LTV-MPCC in simulation. The simulation is based on real data for generating the reference paths and on a realistic vehicle model. We describe how the data collection is performed and the properties of its sparsification, as well as the 4-axles bicycle model. In the end, a comparison between a regular LTV-MPC for path following using a regular path description and the developed LTV-MPCC using the sparse path description is performed.

A. 4-axles bicycle model

The model developed is based on a Scania 480G truck called *Astator*. With this model, we intend to describe the lateral dynamics of the truck. With this purpose, we developed a 4-axles nonlinear bicycle model since *Astator* has 4 axles as well: 2 steering axles in the front and 2 traction axles in the back.

The notation used in the 4-axles bicycle model is shown in Fig. 6. The vehicle lateral dynamics are defined by the following set of differential equations describing the lateral force and the momentum

$$m(\dot{v}_y + \dot{\psi} v_x) = F_{y_{b_b}} + F_{y_{b_f}} + F_{y_{f_b}} \cos\delta_b + F_{y_{f_f}} \cos\delta_f \tag{27a}$$

$$J_z \ddot{\psi} = f_f F_{y_{f_f}} \cos\delta_f + f_b F_{y_{f_b}} \cos\delta_b - b_f F_{y_{b_f}} - b_b F_{y_{b_b}}, \tag{27b}$$

where v_x and v_y denote the longitudinal and the lateral speed of the vehicle, and $\dot{\psi}$ denotes the yaw rate. The constants m and J_z denote the vehicle mass and moment of inertia about the yaw axis, respectively, and $f_{b/f}$ and $b_{b/f}$ represent the distances from the center of gravity to the front and rear axles respectively. Finally, $\delta_{f/b}$ are the steering angles of the front axles. In Fig. 6, $\delta = \delta_f$ and $\delta_b = \beta\delta_f$, where β is the wheel angle ratio between the two front axles. Since the model focuses on the lateral dynamics, v_x has been treated as an input. In this model, the side forces are assumed to be linear functions of the slip angles

$$F_{y_{f_f}} = -C_{f_f}\alpha_{f_f} \tag{28a}$$

$$F_{y_{f_b}} = -C_{f_b}\alpha_{f_b} \tag{28b}$$

$$F_{y_{b_f}} = -C_{b_f}\alpha_{b_f} \tag{28c}$$

$$F_{y_{b_b}} = -C_{b_b}\alpha_{b_b}, \tag{28d}$$

where the constants $C_{b_{f/b}}$ and $C_{f_{f/b}}$ are the cornering stiffnesses of the rear and the front axles respectively. The axles cornering stiffnesses have been computed by linearizing the Pacejka's formula [12] for no longitudinal slip, no camber, small slip angles and using the steady-state vertical loads on each axle obtained through ADAMS simulation. The slip angles are defined as

$$\alpha_{f_f} = \arctan\left(\frac{v_y + \dot{\psi} f_f}{|v_x|}\right) - \delta_f \tag{29a}$$

$$\alpha_{f_b} = \arctan\left(\frac{v_y + \dot{\psi} f_b}{|v_x|}\right) - \delta_b \tag{29b}$$

$$\alpha_{b_f} = \arctan\left(\frac{v_y - \dot{\psi} b_f}{|v_x|}\right) \tag{29c}$$

$$\alpha_{b_b} = \arctan\left(\frac{v_y - \dot{\psi} b_b}{|v_x|}\right). \tag{29d}$$

Also, we derived a formula that allows us to use the curvature information (computed in problem (16)) and relate it with a desired steering angle (input of the vehicle model (27)) assuming steady-state conditions and that $v_x >> v_y$. Equation (30) uses the curvature information to compute the steering angle which makes it dependent on the square of the vehicle velocity and on the stiffnesses of both rear and front wheels

$$\delta_f = \frac{(C_{f_f} + C_{f_b}) \arctan\left((D + K_{us} v_x^2)\kappa\right)}{C_{f_f} + C_{f_b}\beta}, \tag{30}$$

where K_{us} is the understeering gradient of the vehicle which, by definition, contains the information about the vehicle mass and its center of mass, and D is the wheelbase of the truck. In simulation, this relation holds. However, in a real experiment, one could expect the need of feedback control of the steering angle using the vehicle real curvature.

The model parameters (27) were validated with real measurements. Several maneuvers were recorded such as slaloms and U-turns at constant speed. To perform the validation, the model was subjected to the same steering input and the same longitudinal speed as the recorded measurements.

B. Simulation setup and results

Position measurements are recorded using an RTK-GPS installed in the truck. The data is from Scania's test track located in Södertälje, Sweden, south of Stockholm. The recorded data is post-processed offline using the path sparsification algorithm described in Section III with a small ε. Finally, the LTV-MPCC is used on a simulated vehicle using the 4-axles bicycle model.

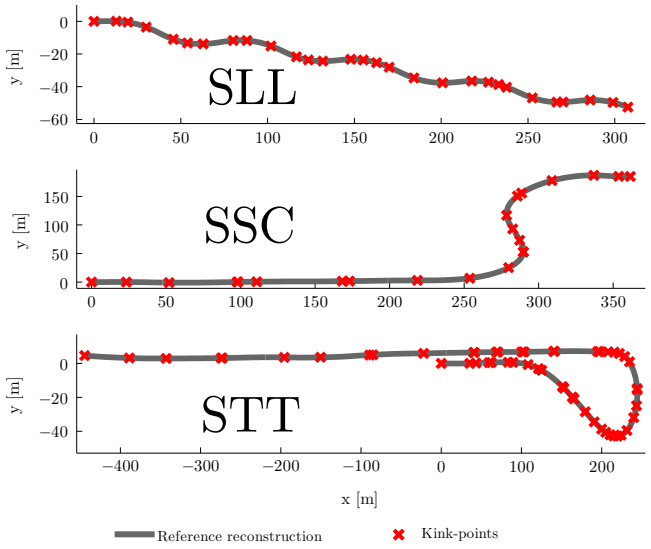

Fig. 7. The three different data sets: SLL, SSC and STT.

TABLE I

REFERENCE PATH INFORMATION FOR $\varepsilon = 0.01$

Path	SLL	SSC	STT
Number of points of the original path	2926	13171	6666
Number of points of the sparse path	29	22	52
Average distance between points in the sparse path (m)	11	24	19

1) Reference path acquisition: To show our results we use 3 different reference paths: a slalom (SLL), a S-shape curve (SSC) and a portion of the Scania's test track where we have both a high speed straight and sharp curves (STT). These data sets are depicted in Fig. 7. Information such as the number of points of the original and the sparsified paths, and the average distance between points is summarized in Table I. The compression ratio is about 1% but the compressed path has almost no loss of detail since the maximum deviation allowed is 1 cm.

2) LTV-MPCC simulation: In Fig. 8 and 9 the performance of the LTV-MPCC of a single run on the reference path SSC is shown. The penalization matrices are chosen as $\mathbf{Q} = \text{diag}(10^1, 10^1, 10^3, 10^3)$ and $\mathbf{R} = \text{diag}(10^1, 10^2)$, the prediction horizon is $H = 10$ which provides an average prediction[2] of 180 m, the vehicle travels at a constant velocity of 10 $\text{m}\,\text{s}^{-1}$ and the MPC solution is recomputed each $T_s = 0.01$ s. The optimization problem (16) is solved using *cvxgen* [13]. The vehicle never deviates more that 35 cm from the reference path meaning that the tracking accuracy is good. The first three entries of the state penalization matrix $\mathbf{Q}$ penalize position and orientation error respectively while $\mathbf{R}$ penalizes the deviations from the reference inputs. The last entrance of matrix $\mathbf{Q}$ act as an entry that promotes lateral jerk

[2]Note that if there are 10 kink points ahead of the vehicle the average prediction distance should be 240 m. However, since the vehicle is traveling along the path, the actual prediction distance is reduced as the distance to next kink point is reduced. So, the vehicle average prediction distance is slightly smaller that the average distance between kink points.

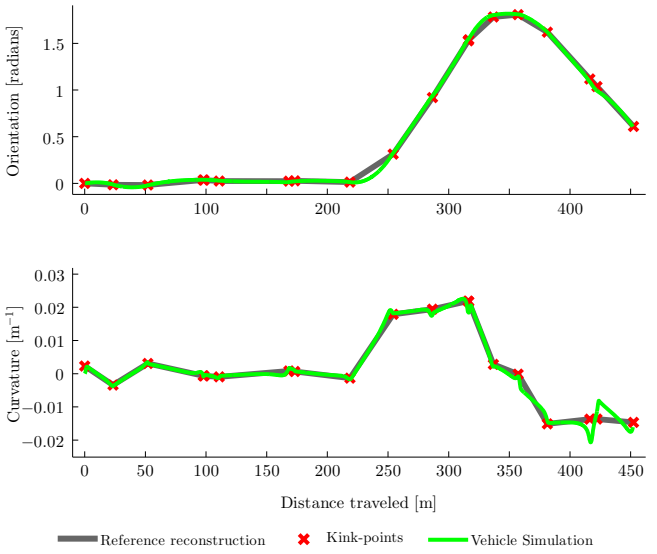

Fig. 8. Simulation results using SSC as reference path. The figure on top shows the reference and the vehicle curvature and the figure below shows the reference and the vehicle orientation.

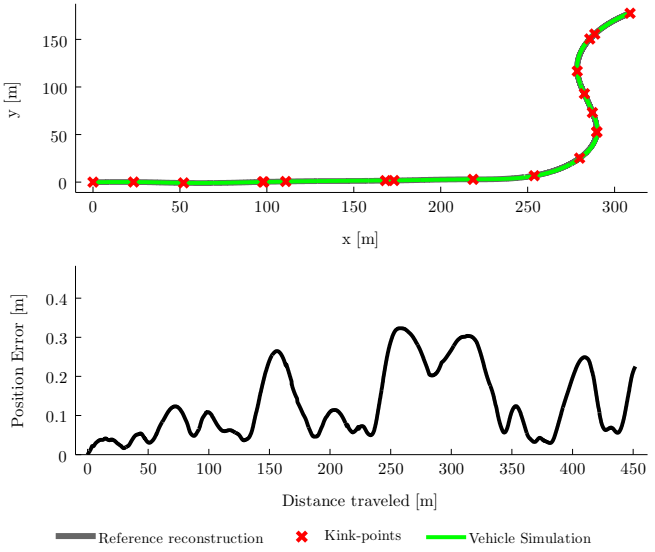

Fig. 9. Simulation results using SSC as reference path. The figure on top shows the reference and the vehicle path and the figure below shows the position error of the vehicle.

minimization, since it contributes to a bigger penalization caused by deviations from the reference curvature. The lateral jerk is the derivative with respect to time of the lateral acceleration. The passenger comfort is mainly affected by high jerk values, where a maximum of 0.9 $\text{m}\,\text{s}^{-3}$ is usually recommended when constructing highways [14]. However, we have a natural trade-off between tracking accuracy and smooth driving. The smoother we impose our controller to be, the less accurate it will be as well.

3) Comparison between LTV-MPCC and LTV-MPC: The LTV-MPC implemented for comparative purposes is based in [15]. This MPC uses a path description where the points are discretized along the original path. These points are separated

TABLE II
CONTROLLER COMPARISON

Controller	LTV-MPCC			LTV-MPC		
Reference	SSL	SSC	STT	SSL	SSC	STT
J_{MA} ($\mathrm{m\,s^{-3}}$)	0.364	0.105	0.171	1.270	1.405	1.920
Pos_{RMSE} (m)	0.222	0.158	0.208	0.281	0.173	0.173
Pos_{MAXE} (m)	0.392	0.323	0.765	0.509	0.302	0.557
T_{opt} (ms)	1.70	1.83	1.77	4.25	4.10	3.75
H_{dist} (m)	90	180	122	20	20	20

2 m apart from each other. The penalization matrices[3] are chosen as $\mathbf{Q} = \mathrm{diag}(10^1, 10^1, 10^3)$ and $\mathbf{R} = 10^1$, the prediction horizon is $H = 20$ which provides a prediction of 20 m the vehicle travels at a constant velocity of $10\ \mathrm{m\,s^{-1}}$ and the MPC solution is recomputed each $T_s = 0.01$ s. The idea is to compare the performance of the LTV-MPC using such description with the LTV-MPCC using a sparse description where the points are significantly more spaced. Here, the performance is measured using the mean absolute (MA) lateral jerk (J_{MA}), the average time to solve the optimization problem (T_{opt}), the average prediction distance (H_{dist}), and the position root mean squared error (RMSE) (Pos_{RMSE}) and its maximum error (Pos_{MAXE}).

Table II summarizes the main results achieved with the LTV-MPCC and LTV-MPC for the same set of reference paths using the same reference speed. The parameters of both the LTV-MPC and LTV-MPCC are chosen as before. Analyzing the results, it is possible to verify that LTV-MPCC achieves a much smoother and more comfortable driving while maintaining roughly the same path following accuracy. The biggest deviation happens at STT where both controllers have a maximum error of more than 0.5 m. This happens in a very sharp curve of Scania's test track where the speed has to be reduced, whereas in this experiment we are just assuming constant speed. Furthermore, the computational burden of the LTV-MPCC is one fourth of the LTV-MPC since the search space is now reduced due to the clothoid-based problem formulation, only allowing for linearly varying curvature with the clothoid arc-length. Also, the prediction distance of the LTV-MPCC is considerable larger than of the LTV-MPC, where the prediction horizon has to be chosen such that the prediction distance is shorter to ensure that the computation finishes within the sampling period.

V. CONCLUSIONS

In this paper, we present a novel LTV-MPCC controller that addresses the problem of clothoid-based path following by a nonholonomic vehicle. This controller formulation is based on the fact that the path of a vehicle traveling at low speeds defines a segment of clothoids if the steering angle is chosen to vary piecewise linearly. Using these assumptions, we are able to compute the vehicle inputs based on the clothoid parameters and to use a large prediction distance with very low computational burden. To feed the controller with a clothoid-based reference path, we presented a novel approach to the path sparsification problem where a reduced number of clothoids is used to describe the reference path. This approach relies on a reweighed l_1-norm approximation of the l_0-norm. The clothoid-based path has few points, where each point has embedded the clothoid segment properties, resulting in a computationally inexpensive path description. The number of clothoids used to describe the path is dependent on the parameter that defines the maximum deviation allowed between the original and the clothoid-based path. The smaller the maximum allowed deviation, the greater the number of clothoids. In the end, we provide simulation results based on a 4-axles bicycle model and on real reference paths. The performance of the controller opens the possibility for future experimental validation of such controller. As future work, we can consider the computational efficiency of the proposed MPC for possible fast re-planning in case of eventual path blocking. Also, the longitudinal dynamics of the vehicle must be included in the LTV-MPCC formulation such that it can handle large curvature turns by, for example, adjusting the vehicle reference speed.

[3]The implemented LTV-MPC has just one input, so the matrix $\mathbf{R}$ is in this case a scalar.

REFERENCES

[1] U. S. Dpt. of Transportation - Federal Highway Administration, Ed., *Public Roads*, vol. 71, no. 1, July/August 2007. [Online]. Available: www.fhwa.dot.gov/publications/publicroads/07july/07.cfm

[2] European Road Transport Research Advisory Council, "Multi-annual implementation plan for Horizon 2020," March 2013.

[3] V. Turri, A. Carvalho, E. Tseng, K. H. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," in *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013)*, Hague, Netherlands, Oct. 2013, pp. 378–383.

[4] F. Borrelli, P. Falcone, and T. Keviczky, "MPC-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3(2), pp. 265–291, 2005.

[5] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *Proceedings of the American Control Conference*, Seattle, Washington, USA, June 2008, pp. 3719–3724.

[6] J. McCrae and K. Singh, "Sketch-based interfaces and modeling (sbim): Sketching piecewise clothoid curves," *Computers and Graphics*, vol. 33, no. 4, pp. 452–461, Aug. 2009.

[7] E. Bertolazzi and M. Frego. (2012, September) Fast and accurate clothoid fitting. University of Trento, Italy. [Online]. Available: arxiv.org/abs/1209.0910

[8] A. D. Luca, G. Oriolo, and C. Samson, *Feedback control of a nonholonomic car-like robot*. Springer Berlin Heidelberg, 1998, vol. 229, ch. 4, pp. 171–253.

[9] M. A. Heald, "Rational approximations for the fresnel integrals," *Mathematics of Computation*, vol. 44, pp. 459–461, Apr. 1985.

[10] E. J. Candes, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14(5), pp. 877–905, 2008.

[11] A. G. Belyaev, "A note on invariant three-point curvature approximations," The University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan, Tech. Rep., 1999.

[12] H. Pacejka, *Tyre and vehicle dynamics*, 2nd ed. Elsevier, 2005.

[13] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, pp. 1–27, 2012.

[14] A. S. Kilinç and T. Baybura, "Determination of minimum horizontal curve radius used in the design of transportation structures, depending on the limit value of comfort criterion lateral jerk," in *TS06G - Engineering Surveying, Machine Control and Guidance*, Rome, Italy, May 2012.

[15] F. Kühne, J. da Silva Jr., and W. F. Lages, "Mobile robot trajectory tracking using model predictive control," in *IEEE II Latin-American Robotics Symposium*, September 2005.