Robotics and Autonomous Systems 136 (2021) 103706

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Accurate autonomous navigation strategy dedicated to the storage of buses in a bus center

Eric Lucet *, Alain Micaelli, François-Xavier Russotto

CEA, LIST, Interactive Robotics Laboratory, Gif-sur-Yvette, F-91191, France

ARTICLE INFO

Article history:
Available online 28 November 2020

Keywords:
Heavy car-like mobile robot
Linear constrained predictive regulator
Curvilinear abscissa dependent sliding estimator
Bus center autonomous storage

ABSTRACT

This paper deals with an innovative autonomous bus navigation and parking system in a bus depot, in order to optimize their movements in a confined area. The kinematic model of the vehicle is defined. Considering its dimensions and weight as well as the centimetric accuracy required, a predictive controller is designed, based on its model linearized around the changing path curvature value, to perform accurate curved paths tracking with a limited tracking error guaranteed by the consideration of a constraint. This controller and additional sliding observers are designed according to the distance traveled, allowing maneuvers to be performed at any forward or backward speed with constant accuracy. In addition, these observers are not affected by path tracking errors. The implementation on an industrial vehicle, operated under realistic conditions, demonstrates the performance and robustness of this navigation system.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Motivation and prior work

The development of autonomous driving technologies on public transport systems is now underway, including the modernization of existing service buses [1] to enable automated driving, and also the design of new bus architectures [2]. In this context, the European EBSF2 project [3] aims to develop a new generation of urban buses using new technologies and infrastructures, and to test them in operational scenarios on several European bus networks.

In particular, one of the main problems encountered concerns the storage of buses in closed centers. These multi-storey centers have narrow passageways and limited parking spaces. This leads to bus collisions with the infrastructure, or even to a collision between buses. Therefore, the speed in the bus center is generally limited to 8 km h^{-1}.

In order to limit these accidents and save drivers time when parking and retrieving their buses, the project proposes to set up an Automatic Bus Guidance System (ABGS). This system is intended for the automated guidance of buses in a known semi-open private environment (storage center). The map of the environment and the trajectory to be followed are assumed to be known *a priori*. This data is made directly available by the bus depot operator. For other bus use cases, the trajectory can be reconstructed online, for example via lane detection based on the data of a 2D laser scanner with trajectory planning based on this detection [4]. The principle of use is as follows. A bus driver who starts his working day arrives at the entrance to the bus station and, via a dedicated interface, selects the bus with which he wishes to work. The bus parked in the bus center automatically leaves its location and follows a predefined path to the bus center exit where it stops at a dedicated and ready-to-use location. Similarly, at the end of the day, the driver who ends his work day leaves the bus at a designated location at the entrance to the bus center. Then, the bus must automatically follow a predefined route in its own area (no pedestrians allowed) to the assigned parking space. This type of functionality has already been investigated, particularly for lane keeping and precision docking of an articulated bus using a linear steering rate controller [5].

For the implementation of such a system, the mechanical architecture of the bus is kept intact. Localization sensors (odometry of the front steering angle and wheel speed, stereoscopic cameras for absolute positioning with respect to the surrounding infrastructure) and safety sensors (lasers for obstacle detection) are installed on the bus. An embedded computer is connected to these sensors and to the bus actuators (steering and propulsion). On the basis of instructions given via the man–machine communication interface, the ABGS must ensure autonomous bus navigation.

Autonomous storage of buses in a bus center requires the control of heavy and large vehicles in confined spaces, for an accurate tracking of paths that may have sharp curves. In particular, a promising solution for the nonadmissible trajectory

* Corresponding author.
E-mail addresses: eric.lucet@cea.fr (E. Lucet), alain.micaelli@cea.fr (A. Micaelli), francois-xavier.russotto@cea.fr (F.-X. Russotto).

https://doi.org/10.1016/j.robot.2020.103706
0921-8890/© 2020 Elsevier B.V. All rights reserved.

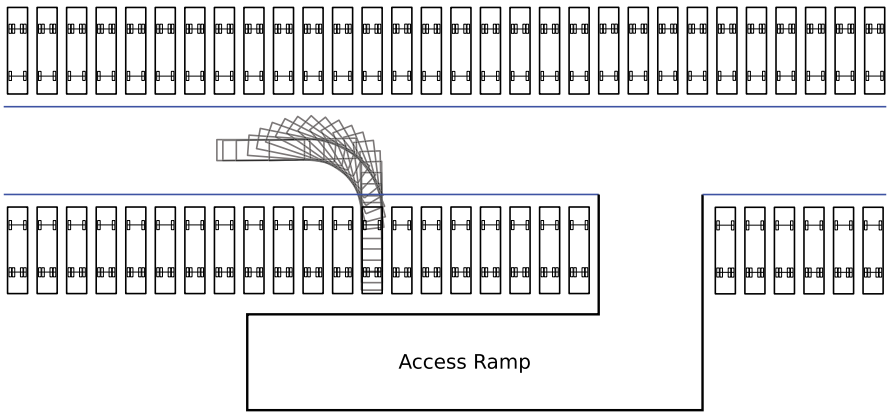


Fig. 1. Bus center configuration.

tracking problem is the transverse function approach that has been proposed in [6] and exploited in [7] with an application to car-like vehicles, the feasibility of which remains to be proven in practice. Then, the control of these heavy vehicles is a problem, due to the high inertia and non-linear behavior of the actuators. Generally, the pneumatic braking system of a heavy vehicle is not linear and is subject to high uncertainties. For example, an adaptive controller for the precise stopping of a bus has been demonstrated during a docking maneuver in [8]. Also, a model of a truck hydraulic power steering system was computed and used to design a yaw rate controller in [9]. More specifically for the automation of the parking process, a fuzzy controller was introduced for the rear parking movement of trucks in the presence of obstacles in [10]. The method requires eight distance sensors integrated into the vehicle and is validated in simulation. Quantitative information is provided to estimate the accuracy of this method. A divergence-based control law is proposed for docking a robot close to an upright surface in [11]. Based on optical flow, this method requires good weather conditions for the camera to detect the upright surface at a reasonable distance. Another vision-based docking strategy, which achieves centimetric accuracy, is the use of image-plane measurements [12]. Similarly, other preliminary results for autonomous backward parking were obtained from a multi-sensor-based interaction model [13]. Here, a model-based predictive controller (MPC) is designed. Based on sensors data, this controller constantly anticipates the respect of a maximum position error constraint. This kind of controller can be used to manage the vehicle's braking to avoid collisions while ensuring lateral stability [14], for example. Also, constrained MPC has already been used in mobile robotics, with several model choices being possible. For example, application cases are available in [15] or in [16]. In [17], authors consider a discrete linear vehicle lateral and yaw dynamic model with a constant longitudinal velocity, their objective being to maintain a collision-free path on the road. Our case study is different, in that the speed is not constant and can even change sign, and the model considered must include a non-zero curve path. However, the lateral positioning constraints remain similar. In addition, the choice of the model used is crucial with respect to the intended application. In practice, the parameters of a dynamic model-based predictive control are often difficult to set, as this setting fluctuates greatly with the environment, in particular the state of the ground and the tire which wears down over time, as well as the variable inclination of the ground in a bus depot. Under these conditions, such methods as those proposed by [18] or by [19], for example, do not continuously guarantee the expected accuracy, which is why a kinematic model with slip consideration is preferred. Then, if a non-linear model is theoretically more accurate, the difficulty in implementing an MPC is to find a solver that guarantees to find a solution in a limited time for online use. However, if several nonlinear programming problem solvers exist ([20,21], etc.), none of them can always guarantee such a result. Other stochastic solvers could be envisaged for these cases of nonlinear MPC [22], but their robustness to changes in the operating environment is not guaranteed, and they do not comply with the bus operator's specifications for implementing a deterministic solution meeting machine standards. Subsequently, the aim of this work is to demonstrate that, with some adjustments, the use of a well-defined linear model leads to accurate results, even in real test situations.

Then, an observer described in [23] is computed independently of the controller by optimizing a quadratic criterion, a similar strategy having already been proposed in [24] for example in the automotive field, but using a particle filter. The aim is to periodically and accurately update the model parameters. In case of lack or failure of global positioning or to improve dead-reckoning performance, other recent works are proposing a solution for estimating the position, orientation and speed [25] or the wheel slip angle [26], from proprioceptive measurements only.

1.2. Approach

In this work, control of a heavy mobile robot with front steering axle and rear driven wheels, equipped with an absolute position measurement system, is considered. The objective is to follow a predefined trajectory in the bus center with sufficient lateral accuracy, including control and localization errors, to ensure safe navigation with other buses and infrastructure at a speed of 2.2 m s^{-1}, the maximum authorized speed in the bus center (2.2 m s^{-1} = 8 km h^{-1}). This trajectory is divided into tracks of variable curvature, some of which have to be followed in forward and others in reverse. The bus center configuration with buses is presented in Fig. 1. Buses always park in reverse.

Parking and exit processes are proposed to be automated with a dedicated control approach. Indeed, a linearized bus model is being considered to design a predictive control that can take into account control inputs and state constraints in relation to the environment. In particular, this approach must ensure a minor lateral deviation of the bus from its reference trajectory over a certain prediction horizon throughout the navigation path. It also allows positive and negative speed movements, without any discontinuity problems during transitions between positive and negative speeds via zero speed. block diagram in Fig. 2 describes the control architecture.

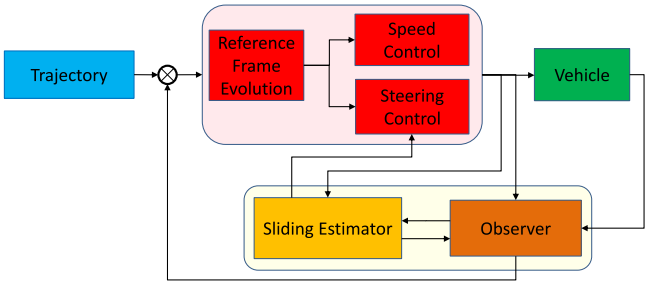


Fig. 2. Control block diagram.

This architecture consists of two blocks:

- the first block above is dedicated to the control itself, the steering controller being built on the basis of a 2D kinematic horizontal plane model of the vehicle;
- the second block below is an online observer, the sliding estimation being dedicated to the correction of sliding values of the control model.

This kind of architecture, based on a kinematic model updated online, makes it easier to handle low speed movements without penalty for higher speed movements via a sliding correction. Vehicle state and sliding are observed on the basis of this kinematic model.

1.3. Layout

In this document, models for path generation, observation and control are designed for an automated bus-type vehicle in Section 2. The model chosen is horizontal plane-roll and pitch are not taken into account. Then, the linear predictive control algorithm designed in Section 3 respects, when necessary, the position constraints during navigation. The vehicle state observation presented in Section 4 is based on an extended Kalman filter; it is complemented by an estimate of longitudinal sliding and sideslip angles. The path is defined in Section 5 as a series of tracks parametrized by their curvilinear abscissa and curvature. These algorithms are validated by kinematic and dynamic simulations, and results are presented in Section 6. On-site tests with a real system are presented in Section 7, with the various parameters involved in control and observation adjusted. Finally, conclusions and perspectives are presented in Section 8.

1.4. Discussion about choices

The kinematic model and the controller are linearized, which simplifies the controller's expression and allows less demand for calculations for its use in realistic situations. It also makes it easier to implement a predictive controller, in order to prevent a possible collision or to determine a control solution, when it exists, in a constrained environment (lateral position limit constraints for example).

Kinematic model equations are written and linearized around the track curvature value, which varies from one track to another, for tracking purposes. The predicted state equations of the discretized model are written by considering a variation in the path curvature at each sampling step, so that the control can handle the case of a variable curvature over the prediction horizon.

In addition, a polynomial defines a fictive curvature as a function of the actual path curvature, in order to feed the control with a more progressive variation curvature allowing a smoothing of the path and the control. In this way, paths are followed with good accuracy on different types of track geometries, including inter-track transitions, as long as the variation in the curvature value is kept within a reasonable range with respect to the response time of the bus steering actuator.

Finally, an Extended Kalman Filter observer is designed. And a longitudinal slip and sideslip angles estimation makes it possible to compensate for wheel-soil sliding phenomena with a dynamic that depends on the refresh period of the position measurement, but also to identify parameters that are difficult to measure with precision, such as the effective diameter of the wheel (necessary for longitudinal odometry) and steering angle offsets.

Thus, the advantage of this global method lies in the resulting accuracy for any forward or reverse navigation speed under constraints of confined areas.

2. Modeling

In this section, the different kinematic models used for control and observation are computed with a rigid body motion formalism using the tools of linear algebra and screw theory [27]. A control algorithm based on a linearized model is defined.

2.1. Kinematic model

The kinematic model defined for the control synthesis is an extension of a previous model (see [28]) adapted to Fig. 3. Each of the two front and rear wheels of the bus are considered equivalent to two virtual wheels located halfway between the real wheels. The bus is then simplified into a conventional bicycle model in top view. Given the Ackerman-type mechanical front steering system, the correspondence between the angle of the virtual wheel in the middle of the front axle, which corresponds to the steering setpoint, and the angle of each of the two real front wheels can be obtained using the classical Ackerman relations. In addition, to ensure that the desired front steering setpoint value is applied, an observer presented in Section 4.2 will be used to estimate any possible steering offset. The frame $(\mathbf{f})$ is fixed to the bus front wheel center with its longitudinal axis in the wheel steering direction. The same applies to frame $(\mathbf{r})$. The controlled frame is $(\mathbf{r})$. The controller ensures the convergence of this frame on the reference path. Reference front $\left(\mathbf{c}_f\right)$ and rear $(\mathbf{c}_r)$ frames calculation is described more in Section 5.1. Notations are the following. ${}^k\mathbf{T}_{i/j}$ is the twist at the origin of frame (i) with respect to frame (j), expressed in frame (k). For simplicity, (k) and (j) are not specified if $k = i$ or if $j = 0$, with (0) the world frame. ${}^j\mathbf{Ad}_i$ is the adjoint matrix, for transporting frame (i) into frame (j). Notations are the same for ${}^j\mathbf{R}_i$, ${}^j\mathbf{t}_i$ and scalars, not specifying (j) when $j = 0$.

2.1.1. Equations of motion

Referring to Fig. 3, it is assumed that the sliding angles $\delta_{f,r}$ are small, so that they can be assimilated to a variation of $\alpha_{f,r}$. Kinematic equations are defined in the two frames $(\mathbf{f})$ and $(\mathbf{r})$ by $\mathbf{T}_f$ and $\mathbf{T}_r$ twists, and $\mathbf{T}_{f/m}$ and $\mathbf{T}_{r/m}$ the twists in relation to the vehicle body frame $(\mathbf{m})$.

Then, the following classical car-like system equations are obtained by combination of these twists:

$$\begin{cases} \dot{\theta}_m = \frac{\sin\beta_f}{l_f - l_r} v_f = \frac{\tan\beta_f}{l_f - l_r} v_r \\ v_r = v_f \cos\beta_f \end{cases} \tag{1}$$

2.1.2. Trajectory following equations

The projection of both frames $(\mathbf{f})$ and $(\mathbf{r})$ on the trajectory tracks in the corresponding frame, respectively $\left(\mathbf{c}_f\right)$ and $(\mathbf{c}_r)$, is considered. With respect to these frames, evolution of $(\mathbf{f})$ and $(\mathbf{r})$ is given by:

$${}^f\mathbf{T}_{f/c_f} = \mathbf{T}_f - {}^f\mathbf{Ad}_{c_f}\mathbf{T}_{c_f}, \quad {}^r\mathbf{T}_{r/c_r} = \mathbf{T}_r - {}^r\mathbf{Ad}_{c_r}\mathbf{T}_{c_r} \tag{2}$$

with:

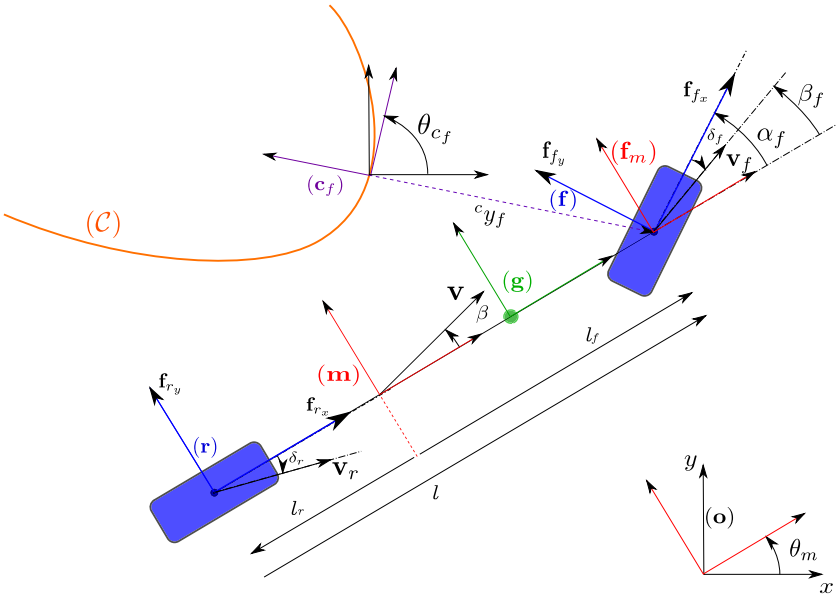


Fig. 3. Planar horizontal model.

- $^{f,r}\mathbf{Ad}_{c_{f,r}} = \begin{bmatrix} 1 & 0_{1\times 2} \\ -\mathbf{J}^{f,r}\mathbf{t}_{c_{f,r}} & {}^{f,r}\mathbf{R}_{c_{f,r}} \end{bmatrix}$;
- $^{f,r}\mathbf{t}_{c_{f,r}} = -^{f,r}\mathbf{R}_{c_{f,r}} \begin{bmatrix} 0 \\ {}^{c_{f,r}}y_{f,r} \end{bmatrix}$; $\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$;
- $^{f,r}\mathbf{R}_{c_{f,r}} = \begin{bmatrix} \cos\left(\theta_m - \theta_{c_{f,r}} + \beta_{f,r}\right) & \sin\left(\theta_m - \theta_{c_{f,r}} + \beta_{f,r}\right) \\ -\sin\left(\theta_m - \theta_{c_{f,r}} + \beta_{f,r}\right) & \cos\left(\theta_m - \theta_{c_{f,r}} + \beta_{f,r}\right) \end{bmatrix}$;
- $\mathbf{T}_{c_{f,r}} = \begin{bmatrix} c_{f,r}\dot{s}_{f,r} \\ \dot{s}_{f,r} \\ 0 \end{bmatrix}$, with $s_{f,r}$ and $c_{f,r}$ the curvilinear abscissa and curvatures associated with the trajectory in $(\mathbf{c}_f)$ and $(\mathbf{c}_r)$ frames respectively.

Especially for the front axle:
$^{f}\mathbf{t}_{c_f} = {}^{f}\mathbf{R}_{c_f} \begin{bmatrix} 0 \\ -{}^{c_f}y_f \end{bmatrix} = \begin{bmatrix} {}^{c_f}y_f \sin(\theta_{cf} - \theta_f) \\ -{}^{c_f}y_f \cos(\theta_{cf} - \theta_f) \end{bmatrix}$
and: $-\mathbf{J}^{f}\mathbf{t}_{c_f} = \begin{bmatrix} -{}^{c_f}y_f \cos(\theta_{cf} - \theta_f) \\ -{}^{c_f}y_f \sin(\theta_{cf} - \theta_f) \end{bmatrix}$, with $\theta_f = \theta_m + \beta_f$.

Thus: $^{f}\mathbf{Ad}_{c_f} =$
$\begin{bmatrix} 1 & 0 & 0 \\ -{}^{c_f}y_f \cos(\theta_{cf} - \theta_f) & \cos(\theta_{cf} - \theta_f) & -\sin(\theta_{cf} - \theta_f) \\ -{}^{c_f}y_f \sin(\theta_{cf} - \theta_f) & \sin(\theta_{cf} - \theta_f) & \cos(\theta_{cf} - \theta_f) \end{bmatrix}$; and:
$^{f}\mathbf{T}_{f/c_f} = \begin{bmatrix} \dot{\theta}_m - c_f\dot{s}_f + \dot{\beta}_f \\ v_f + y_f \cos\left(\theta_{cf} - \theta_f\right) c_f\dot{s}_f - \cos(\theta_{cf} - \theta_f)\dot{s}_f \\ 0 + y_f \sin\left(\theta_{cf} - \theta_f\right) c_f\dot{s}_f - \sin(\theta_{cf} - \theta_f)\dot{s}_f \end{bmatrix}$. Considering $^{c_f}\mathbf{T}_{f/c_f} = {}^{c_f}\mathbf{Ad}_f\,{}^{f}\mathbf{T}_{f/c_f}$ we have:
$^{f}\mathbf{R}_{c_f}^{t}\mathbf{J}^{f}\mathbf{t}_{c_f} = \begin{bmatrix} -\cos\left(\theta_{cf} - \theta_f\right) & -\sin\left(\theta_{cf} - \theta_f\right) \\ \sin\left(\theta_{cf} - \theta_f\right) & -\cos\left(\theta_{cf} - \theta_f\right) \end{bmatrix} \begin{bmatrix} {}^{c_f}y_f \cos\left(\theta_{cf} - \theta_f\right) \\ {}^{c_f}y_f \sin\left(\theta_{cf} - \theta_f\right) \end{bmatrix} = \begin{bmatrix} -{}^{c_f}y_f \\ 0 \end{bmatrix}$;
and then: $^{c_f}\mathbf{Ad}_f = {}^{f}\mathbf{Ad}_{c_f}^{-1} =$
$\begin{bmatrix} 1 & 0 & 0 \\ -{}^{c_f}y_f & \cos\left(\theta_{cf} - \theta_f\right) & \sin\left(\theta_{cf} - \theta_f\right) \\ 0 & -\sin\left(\theta_{cf} - \theta_f\right) & \cos\left(\theta_{cf} - \theta_f\right) \end{bmatrix}$.

In frame $(\mathbf{c}_f)$:
$^{c_f}\dot{\theta}_f = \dot{\theta}_m - c_f\dot{s}_f + \dot{\beta}_f$;
$^{c_f}\dot{x}_f = v_f \cos\left(\theta_{c_f} - \theta_f\right) - \dot{s}_f - {}^{c_f}y_f\left(\dot{\theta}_m + \dot{\beta}_f - 2c_f\dot{s}_f\right)$.
Thus: $^{c_f}\dot{x}_f =$
$v_f \cos\left(\theta_{c_f} - \theta_f\right) - \dot{s}_f\left(1 - c_f\,{}^{c_f}y_f\right) - {}^{c_f}y_f\,{}^{c_f}\dot{\theta}_f$;
and: $^{c_f}\dot{y}_f = -v_f \sin\left(\theta_{c_f} - \theta_f\right)$.

By writing the relative velocity $^{f}\mathbf{T}_{f/c_f}$ in frame $(\mathbf{c}_f)$ without transport, meaning by multiplying the velocity of translation by matrix $^{f}\mathbf{R}_{c_f}^{t}$, we have:

$$\begin{cases} {}^{c_f}\dot{\theta}_f &= \frac{\sin\beta_f}{l} v_f + \dot{\beta}_f - c_f\dot{s}_f \\ 0 &= v_f \cos\left(\theta_f - \theta_{c_f}\right) - \dot{s}_f(1 - c_f\,{}^{c_f}y_f) \\ {}^{c_f}\dot{y}_f &= v_f \sin\left(\theta_f - \theta_{c_f}\right) \end{cases} \tag{3}$$

Similarly $^{r}\mathbf{T}_{r/c_r}$ is expressed in frame $(\mathbf{c}_r)$:

$$\begin{cases} {}^{c_r}\dot{\theta}_r &= \frac{\tan\beta_f}{l} v_r - c_r\dot{s}_r \\ 0 &= v_r \cos\left(\theta_r - \theta_{c_r}\right) - \dot{s}_r(1 - c_r\,{}^{c_r}y_r) \\ {}^{c_r}\dot{y}_r &= v_r \sin\left(\theta_r - \theta_{c_r}\right) \end{cases} \tag{4}$$

2.2. Linearized model

The navigation path considered is a circle of constant curvature c_r. Above models are then linearized around the steady state:

- $\theta_r = \theta_{c_r}$;
- $^{c_r}y_r = 0$;
- $\tan\beta_f = lc_r = \tan\beta_c$.

β_c is the constant front steering angle of the bus to track the circular path.

Decomposition law of the tangent function into its Taylor series at point β_c is considered: $\tan(x) \approx$
$\tan\beta_c + (1 + \tan^2\beta_c)(x - \beta_c) + \tan\beta_c(1 + \tan^2\beta_c)(x - \beta_c)^2$
And, by writing:

- $\theta = \theta_r - \theta_{c_r}$;
- $v = v_r$;
- $y = {}^{c_r}y_r$.

The linearization of Eq. (4) gives:
$\dot{s}_r = \frac{v\cos\theta}{1 - c_r y} \approx \frac{v}{1 - c_r y}$;
$\dot{y} = v\sin\theta \approx v\theta = (1 - c_r y)\dot{s}_r\theta$;
$\dot{\theta} = \left[lc_r + (1 + l^2c_r^2)(\beta_f - \beta_c)\right]\frac{v}{l} - c_r\dot{s}_r$.

Then, by considering the curvilinear derivative $y' = \frac{dy}{ds_r} = \frac{dy}{dt}\frac{dt}{ds_r} = \frac{\dot{y}}{\dot{s}_r}$ and keeping only the linear terms, we have:

$$\begin{cases} \theta' &= \frac{(1+l^2c_r^2)}{l}(\beta_f - \beta_c) - c_r^2 y \\ y' &= \theta \\ y'' &= \theta' \\ y''' &= \frac{(1+l^2c_r^2)}{l}\beta_f' - c_r^2 y' \end{cases} \tag{5}$$

with $(\cdot)'$ expressing a derivative with respect to the curvilinear abscissa. Finally, considering the state vectors $\mathbf{y} = \begin{bmatrix} y \\ y' \\ y'' \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} y \\ \theta \\ \beta_f - \beta_c \end{bmatrix}$ leads to the following relations:

$$\begin{cases} \mathbf{y} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_r^2 & 0 & \frac{1+l^2c_r^2}{l} \end{bmatrix}\mathbf{x} \\ &= \mathbf{P}\mathbf{x}; \\ \mathbf{y}' &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -c_r^2 & 0 \end{bmatrix}\mathbf{y} + \begin{bmatrix} 0 \\ 0 \\ \frac{1+l^2c_r^2}{l} \end{bmatrix}\beta_f' \\ &= \mathbf{A}\mathbf{P}^{-1}\mathbf{y} + \mathbf{B}u. \end{cases} \tag{6}$$

2.3. Observation models

The state observation is based on Eqs. (1), (3) and (4), and on the pose $\mathbf{X}_m = \begin{bmatrix} \theta_m \\ \mathbf{x}_m \end{bmatrix}$.

Twists and positions of frames (**f**) and (**r**) are used for sliding observation. Unlike previous work (e.g. [29,30] or [31]), observation does not depend on the path. The major advantage is that the observer is not disturbed in case of a significant path tracking error.

3. MPC for steering control

In order to take severe accessibility constraints into account when navigating in bus centers, a predictive controller is implemented. This controller is based on a discretized model obtained from Eqs. (6). It optimizes a criterion that depends on predicted states and future control inputs over a spatial window of length nS, S being the discretization step and n the number of selected steps, guaranteeing, if possible, certain constraints on the state (an acceptable position error) and control (steering speed limits). This controller regulates the bus steering, while an independent speed controller is defined further in Section 5.2.

3.1. Linearized model discretization

Given the linearized model (6), state matrices of the continuous system are $\mathbf{A}_c = \mathbf{A}\mathbf{P}^{-1}$ and $\mathbf{B}_c = \mathbf{B}$. The solution of the differential equation $\mathbf{y}' = \mathbf{A}_c\mathbf{y} + \mathbf{B}_c u$ is given by $\mathbf{y}(s) = \mathbf{y}_0 e^{\mathbf{A}_c s} + \int_0^s e^{\mathbf{A}_c(s-\tau)}\mathbf{B}_c u(\tau)d\tau$.

$\mathbf{A}_c$ and $\mathbf{B}_c$ are assumed to be constant with respect to the curvilinear abscissa s. This hypothesis is maintained over a small distance S, meaning that $\mathbf{A}_{c_k}$ and $\mathbf{B}_{c_k}$ are functions of a curvature c_{r_k} which is constant over a step S.

Performing a discretization of one step of length S along the curvilinear abscissa, state matrices of the discrete system $\mathbf{y}_{k+1} = \mathbf{A}_{d_k}\mathbf{y}_k + \mathbf{B}_{d_k}u_k$, $k \in \mathbb{N}$, are given by:

$$\begin{cases} \mathbf{A}_{d_k} &= e^{\mathbf{A}_{c_k}S} \\ &= \mathbf{I}_3 + S\mathbf{A}_{c_k} + \cdots + \frac{S^i}{i!}\mathbf{A}_{c_k}^i + \cdots \\ \mathbf{B}_{d_k} &= e^{\mathbf{A}_{c_k}S}\int_0^S e^{-\mathbf{A}_{c_k}\tau}d\tau\ \mathbf{B}_{c_k} = \frac{\mathbf{B}_{c_k}}{\mathbf{A}_{c_k}}(\mathbf{A}_{d_k} - \mathbf{I}_3) \end{cases} \tag{7}$$

u_k control input is assumed to be constant over a sampling step S. For the consideration of forward and backward displacements, these calculations have to be made for $S > 0$ and $S < 0$.

3.2. Predicted states

Predicted states are computed from the knowledge of the current state $\mathbf{y}_0$ and by using matrices $\mathbf{A}_{d_k}$ and $\mathbf{B}_{d_k}$, previously computed. A recursive calculation on n steps gives:

$$\begin{cases} \mathbf{y}_1 &= \mathbf{A}_{d_0}\mathbf{y}_0 + \mathbf{B}_{d_0}u_0 \\ \mathbf{y}_2 &= \mathbf{A}_{d_1}\mathbf{A}_{d_0}\mathbf{y}_0 + \mathbf{A}_{d_1}\mathbf{B}_{d_0}u_0 + \mathbf{B}_{d_1}u_1 \\ \mathbf{y}_3 &= \mathbf{A}_{d_2}\mathbf{A}_{d_1}\mathbf{A}_{d_0}\mathbf{y}_0 + \mathbf{A}_{d_2}\mathbf{A}_{d_1}\mathbf{B}_{d_0}u_0 + \mathbf{A}_{d_2}\mathbf{B}_{d_1}u_1 \\ &\quad + \mathbf{B}_{d_2}u_2 \\ &etc. \end{cases}$$

In matrix form:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{d_0} \\ \mathbf{A}_{d_1}\mathbf{A}_{d_0} \\ \vdots \\ \mathbf{A}_{d_{n-1}}\cdots\mathbf{A}_{d_0} \end{bmatrix}\mathbf{y}_0 + \begin{bmatrix} \mathbf{B}_{d_0} & \mathbf{0}_{3\times1} & \cdots & \mathbf{0}_{3\times1} \\ \mathbf{A}_{d_1}\mathbf{B}_{d_0} & \mathbf{B}_{d_1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0}_{3\times1} \\ \mathbf{A}_{d_{n-1}}\cdots\mathbf{A}_{d_1}\mathbf{B}_{d_0} & \mathbf{A}_{d_{n-1}}\cdots\mathbf{A}_{d_2}\mathbf{B}_{d_1} & \cdots & \mathbf{B}_{d_{n-1}} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{bmatrix} \tag{8}$$

and more synthetically:

$$\mathbf{Y} = \mathcal{A}\mathbf{y}_0 + \mathcal{B}\mathbf{U} \tag{9}$$

3.3. Criterion

The criterion to be optimized is a quadratic function of the predicted state $\mathbf{Y}$ and the future control input $\mathbf{U}$. It is expressed as follows:

$$crit = \frac{1}{2}\mathbf{Y}^t\mathcal{Q}\mathbf{Y} + \frac{1}{2}\mathbf{U}^t\mathcal{R}\mathbf{U} \tag{10}$$

$\mathcal{Q}$ and $\mathcal{R}$ are weighting matrices of sizes $3n \times 3n$ and $n \times n$ respectively for the state and the control. $\mathcal{Q}$ has a block diagonal structure of elements $\gamma_Q^k\mathbf{Q}_{3\times3}$ and $\mathcal{R}$ has a block diagonal structure of elements $\gamma_R^k\mathbf{R}_{1\times1}$. $(\gamma_Q; \gamma_R) \in]0; 1[^2$ are forgetting factors, $k \in [1; n]$ is the index of a block in the diagonal, and $\mathbf{Q}_{3\times3}$ and $\mathbf{R}_{1\times1}$ are positive definite chosen matrices.

Expression of the criterion as a function of $\mathbf{U}$, is given by:

$$crit = \frac{1}{2}\mathbf{U}^t\left(\mathcal{B}^t\mathcal{Q}\mathcal{B} + \mathcal{R}\right)\mathbf{U} + \mathbf{y}_0^t\mathcal{A}^t\mathcal{Q}\mathcal{B}\mathbf{U} \tag{11}$$

where the parameter depending on $\mathbf{y}_0$ is expressed separately.

3.4. Constraints

The expression of state constraints for given vehicle dimensions is defined hereafter by Eq. (12). By simple approximation, the vehicle is modeled as a rectangle with front and rear ends at a distance of D_f and D_r respectively from the front steering axle along the longitudinal axis. Specifying that these ends remain within a tolerance δ_{gap} around the path can be written as follows:

$$\begin{cases} |y - D_r\sin(\theta_m - \theta_c)| < \delta_{gap} \\ |y + D_f\sin(\theta_m - \theta_c)| < \delta_{gap} \end{cases}$$

Thus, by linearizing and writing $\theta = \theta_m - \theta_c$, it results:

$$-\begin{bmatrix} \delta_{gap} \\ \delta_{gap} \end{bmatrix} \leq \begin{bmatrix} 1 & -D_r & 0 \\ 1 & D_f & 0 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} \delta_{gap} \\ \delta_{gap} \end{bmatrix} \tag{12}$$

With $\mathbf{D} = \begin{bmatrix} 1 & -D_r & 0 \\ 1 & D_f & 0 \\ -1 & D_r & 0 \\ -1 & -D_f & 0 \end{bmatrix} \mathbf{P}^{-1}$, and $\mathbf{d} = \delta_{gap}\mathbf{1}_{4\times 1}$, this constraint can be rewritten as:

$$\mathbf{D}\mathbf{y} + \mathbf{d} \geq \mathbf{0}_{4\times 1} \tag{13}$$

Extended to all the n predicted states and written as a function of future controls, it becomes:

$$\mathfrak{D}\mathcal{B}\mathbf{U} + \mathfrak{d}_{gap} + \mathfrak{D}\mathcal{A}\mathbf{y}_0 \geq \mathbf{0}_{4n\times 1} \tag{14}$$

with:

- $\mathfrak{D}$, a block diagonal matrix consisting of n blocks $\mathbf{D}$;
- $\mathfrak{d}_{gap} = \delta_{gap}\mathbf{1}_{4n\times 1}$.

3.5. Control synthesis

Control input vector $\mathbf{U}$ results as the solution to the following quadratic problem:

$$\begin{cases} \min\limits_{U} \dfrac{1}{2}\mathbf{U}^t \left(\mathcal{B}^t\mathcal{Q}\mathcal{B} + \mathcal{R}\right)\mathbf{U} + \mathbf{y}_0^t\mathcal{A}^t\mathcal{Q}\mathcal{B}\mathbf{U} \\ \quad \mathfrak{D}\mathcal{B}\mathbf{U} + \mathfrak{d}_{gap} + \mathfrak{D}\mathcal{A}\mathbf{y}_0 \geq \mathbf{0}_{4n\times 1} \end{cases} \tag{15}$$

Extracting $\beta_f' = u_0$, the first element of solution $\mathbf{U}$, gives the control law: $\dot{\beta}_f = v_f u_0$.

3.6. Management of curvature variations at inter-track transitions

When linearizing the bus kinematic model around a track curvature c_r, a nominal β_f forward steering angle configured for the corresponding curvature is assumed *a priori*. So, a direction angle variation can be observed in the prediction algorithm between two tracks. A too important variation will not be feasible in a real situation.

One solution to this problem is to artificially modify the value of the curvature c_r over the n prediction horizon, so as to obtain a gradual variation at the change of track.

Since the bus is currently running on a c_{r_0} curvature track and has to reach a $c_{r_{n-1}}$ curvature track at nS horizon, the c_{r_i}, $1 \leq i \leq n-2$ curvature is redefined at all intermediate positions by the following polynomial:

$$c_{r_i} = \frac{c_{r_{n-1}} - c_{r_0}}{n-1} i + c_{r_0}$$

for a linear progression between c_{r_0} and $c_{r_{n-1}}$.

Other choices are possible, including more complex higher degree polynomials. Depending on the case, the ratio between the value of the prediction horizon nS and the size of tracks is to be considered. Indeed, in the case where the prediction horizon nS is small in front of the track size, this method ensuring a redefinition of each of the intermediate curvatures c_{r_i} will be the most relevant, as it is similar to a virtual subdivision of these curvatures. On the other hand, if the prediction horizon nS becomes significant with regard to the size of tracks, which may be the case, for example, by applying the proposed method at the end of the following paragraph 5.1.1, a virtual variation of the curvature values is not always necessary. Otherwise, it must be ensured that any new virtual curvature value does not distort the trajectory by always remaining between the actual value and the next different real value.

4. Observer

The observer is designed in three stages. The first stage consists in observing the position, orientation and speed. The second stage consists in observing a pseudo longitudinal slippage and, rather than observing a real slippage, an equivalent wheel diameter is observed. The last stage consists in observing the lateral slippage.

For these stages, measured proprioceptive state variables are taken into account, such as:

- ω_f and ω_r, the average rotation speeds of the front and rear wheel axles;
- α_f, the front steering angle;
- $\dot{\alpha}_f$, the front steering velocity.

For measurements of exteroceptive variables, it is assumed that the following variables are available:

- ${}^o\mathbf{x}_{m_{mes}}$, the measured position of frame ($\mathbf{m}$) with respect to the reference frame ($\mathbf{o}$);
- ${}^o\theta_{m_{mes}}$, the measured vehicle orientation with respect to the reference frame ($\mathbf{o}$).

Thereafter, and to simplify writing, exponent o referring to the reference frame is removed.

4.1. State observation

For this observer, the estimated sideslip angle $\hat{\delta}_f$ is taken into account (see paragraph 4.2.2):

$$\hat{\beta}_f = \alpha_f + \hat{\delta}_f \tag{16}$$

as well as the estimated mean wheel diameter $\hat{d}_{wheel}$ (see paragraph 4.2.1).

4.1.1. Odometry

The first stage in observing the state is to compute an estimate of front and rear speeds v_f and v_r consistent with the kinematic model, using Eq. (1). Considering $v_{f_{mes}} = \frac{\hat{d}_{wheel}}{2}\omega_{f_{mes}}$ and $v_{r_{mes}} = \frac{\hat{d}_{wheel}}{2}\omega_{r_{mes}}$, a solution to this problem is given by:

$$\begin{cases} \hat{v}_f = v_{f_{mes}} + \dfrac{q_f^{-1}\hat{\phi}_v}{q_f^{-1}\hat{\phi}_v^2 + q_r^{-1}}\Delta \\ \hat{v}_r = v_{r_{mes}} + \dfrac{q_r^{-1}}{q_f^{-1}\hat{\phi}_v^2 + q_r^{-1}}\Delta \end{cases} \tag{17}$$

where:

- $\Delta = v_{r_{mes}} - \hat{\phi}_v v_{f_{mes}}$;
- $\hat{\phi}_v = \cos\hat{\beta}_f$;
- q_f^{-1} et q_r^{-1} are filtering parameters, but may also be considered as $v_{f_{mes}}$ and $v_{r_{mes}}$ error variances.

These measured and estimated variables are used to compute the following expression:

$$\mathbf{T}_{m_{odo}} = {}^m\mathbf{Ad}_{f_{odo}}\left(\mathbf{T}_{f_{odo}} - \mathbf{T}_{f/m_{odo}}\right) \tag{18}$$

with the subscript $(\cdot)_{odo}$ meaning *odometry*.

4.1.2. State estimate

Based on generic vectors $\mathbf{X} = \begin{bmatrix} \theta \\ \mathbf{x} \end{bmatrix}$, $\mathbf{T} = \begin{bmatrix} \omega \\ \mathbf{v} \end{bmatrix}$ and the corresponding matrices $[\mathbf{X}] = \begin{bmatrix} \mathbf{R}_\theta & \mathbf{x} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$, and $[\mathbf{T}] =$

$\begin{bmatrix} \Omega & \mathbf{v} \\ \mathbf{0}_{1\times 2} & 0 \end{bmatrix}$, with $\Omega = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}$, the state equation is given by:

$$[\mathbf{X}_m]_{k+1} = [\mathbf{X}_m]_k \, \mathrm{e}^{dt\left[\mathbf{T}_{m_{odo_k}}\right]} \tag{19}$$

This is a nonlinear equation. Variation of the estimation of $[\mathbf{X}_m]_{k+1}$, expressed as a twist function of the variation of $[\mathbf{X}_m]_k$ and linearized to the first order, can be written as follows:

$$\hat{\mathbf{T}}_{m_{k+1}} = {}^{k+1}\mathbf{A}d_k \hat{\mathbf{T}}_{m_k} + dt\mathbf{T}_{m_{odo_k}} \tag{20}$$

where:

- ${}^{k+1}\mathbf{A}d_k = \mathbf{A}d_{\mathrm{e}^{-dt\left[\mathbf{T}_{m_{odo_k}}\right]}}$ is the adjoint matrix related to $\mathrm{e}^{-dt\left[\mathbf{T}_{m_{odo_k}}\right]}$;
- $\hat{\mathbf{T}}_{m_k}$ and $\hat{\mathbf{T}}_{m_{k+1}}$ are expressed in frames $\mathbf{X}_{m_k}$ and $\mathbf{X}_{m_{k+1}}$ respectively.

The variance relative to the state equation (19) is function of the variance of the odometry twist $\mathbf{T}_{m_{odo}}$ which, for a short enough sampling period dt, can be expressed by:

$$\mathfrak{Q}_k = dt^2 \mathfrak{q}_{odo_k} \tag{21}$$

where $\mathfrak{q}_{odo_k}$ is the variance of $\mathbf{T}_{m_{odo_k}}$. And this variance expression is given by:

$$\mathfrak{P}_{k+1/k} = {}^{k+1}\mathbf{A}d_k \mathfrak{P}_{k/k} {}^{k+1}\mathbf{A}d_k^t + \mathfrak{Q}_k \tag{22}$$

If the $\mathbf{X}_{mes}$ measurement is available synchronously with $\mathbf{X}_{m_k}$, then an $\hat{\mathbf{X}}_{m_k}$ estimator of $\mathbf{X}_{m_k}$ can be written as:

$$\left[\hat{\mathbf{X}}_{m_k}\right] = \left[\mathbf{X}_{m_k}\right] \mathrm{e}^{\left[\mathbf{L}_X \mathbf{T}_{m_{mes_k}}\right]} \tag{23}$$

where $\mathbf{L}_X$ is an observation gain matrix and $\mathbf{T}_{m_{mes_k}}$, is the error twist expressed in the $\mathbf{X}_{m_k}$ frame. The estimator minimizing the variance is given by:

$$\mathbf{L}_X = \mathfrak{P}_{k/k-1}\left(\mathfrak{P}_{k/k-1} + \mathfrak{R}\right)^{-1} \tag{24}$$

where:

- $\mathfrak{P}_{k/k-1}$ is the $\mathbf{X}_{m_k}$ state variance matrix before estimation;
- $\mathfrak{R}$ is the $\mathbf{X}_{m_{mes}}$ measurement variance matrix.

The variance of the resulting state $\hat{\mathbf{X}}_{m_k}$ is then given by:

$$\mathfrak{P}_k = \mathfrak{P}_{k/k-1} - L_X \mathfrak{P}_{k/k-1} \tag{25}$$

The error and variance are propagated to the following states using Eq. (20) until the measurement is available. If the $\mathbf{X}_{mes}$ measurement is available before $\mathbf{X}_{m_k}$, for the next state $\mathbf{X}_{m_{k+1}}$ the formulation is:

$$\begin{cases} \left[\hat{\mathbf{X}}_{m_{k+1}}\right] &= \left[\mathbf{X}_{m_{k+1}}\right] \mathrm{e}^{\left[{}^{k+1}\mathbf{A}\mathbf{d}_k \mathbf{L}_X \mathbf{T}_{m_{mes_k}}\right]} \\ \mathfrak{P}_{k+1/k} &= {}^{k+1}\mathbf{A}\mathbf{d}_k \mathfrak{P}_k {}^{k+1}\mathbf{A}\mathbf{d}_k^t + \mathfrak{Q}_k \end{cases} \tag{26}$$

These equations correspond to prediction state equations that are modified according to (23) when a new measurement is available.

4.2. Longitudinal slippage and sideslip angle estimates

Thereafter, estimated speeds and positions of (**f**) and (**r**) frames, given by the following relationships, are used:

$$\begin{cases} \hat{\mathbf{T}}_{f,r} &= {}^{m}\widehat{\mathbf{A}d}_{f,r}^{-1} \hat{\mathbf{T}}_m + \hat{\mathbf{T}}_{f,r/m} \\ \hat{\mathbf{X}}_{f,r} &= \hat{\mathbf{X}}_m + \begin{bmatrix} \hat{\beta}_{f,r} \\ l_{f,r} \begin{bmatrix} \cos\hat{\theta}_m \\ \sin\hat{\theta}_m \end{bmatrix} \end{bmatrix} \end{cases} \tag{27}$$

For estimates of longitudinal slippage and sideslip angle, two successive estimates of $\mathbf{X}_{f,r}$ state are considered, $\hat{\mathbf{X}}_{f,r_k}$ and $\hat{\mathbf{X}}_{f,r_{k+n}}$, synchronous with measurements (n number of sampling steps between two measurements). If the odometry is correct, the variation between these two estimates is given by:

$$\left[\hat{\mathbf{X}}_{f,r_{k+n}}\right] = \left[\hat{\mathbf{X}}_{f,r_k}\right] \prod_{1\le j\le n} \mathrm{e}^{dt\left[\hat{\mathbf{T}}_j\right]} \tag{28}$$

where $\hat{\mathbf{T}}_j$ is the odometry twist calculated at step $k+j$. Based on a variation of a λ parameter to be identified, the variation of $\hat{\mathbf{X}}_{f,r_{k+n}}$, which is also the state correction given by the Kalman estimator and expressed in the (**f**) and (**r**) frames by the second equation of (27), can be written as follows:

$$\mathrm{vec}\left(\left[\hat{\mathbf{X}}_{f,r_{k+n}}\right]^{-1} \frac{\partial\left[\hat{\mathbf{X}}_{f,r_{k+n}}\right]}{\partial\lambda}\right) = dt \sum_{1\le j\le n}^{k+j} \mathbf{A}\mathbf{d}_{k+n}^{-1} \frac{\partial\hat{\mathbf{T}}_j}{\partial\lambda} \tag{29}$$

Considering this equation (29), a solution to minimize a quadratic criterion with respect to the λ parameter is then given by:

$$\begin{cases} \dot{\lambda} &= l_\lambda \sum_{1\le j\le n} \frac{\partial\hat{\mathbf{T}}_j}{\partial\lambda}^t {}^{k+j}\mathbf{A}\mathbf{d}_{k+n}^{-t} \hat{\mathbf{T}}_{f,r_{mes_{k+n}}} \\ \hat{\lambda}_+ &= \hat{\lambda}_- + dt\dot{\lambda} \end{cases} \tag{30}$$

where l_λ is a gain and the twist $\hat{\mathbf{T}}_{f,r_{mes_{k+n}}}$ corresponds to the difference between the measured and theoretical postures at the time $k+n$. For more details, a full description of this parameter estimator is available in [23].

In the following paragraphs, estimator (30) is applied to longitudinal slippage and sideslip angles.

4.2.1. Wheel diameter estimate

For estimating the longitudinal slippage or the equivalent wheel diameter, $\mathbf{X}_f$ is taken into account. To simplify equations, $\hat{\beta}_f$ is assumed to be invariant during a sampling step. Thus, the partial derivative of the odometry twist $\hat{\mathbf{T}}_{f_j}$ with respect to $\hat{d}_{wheel}$ is given by:

$$\frac{\partial\hat{\mathbf{T}}_j}{\partial\hat{d}_{wheel}} = \begin{bmatrix} \hat{\omega}_{fj} \frac{\sin\hat{\beta}_{f_j}}{l} \\ \hat{\omega}_{f_j} \\ 0 \end{bmatrix} \tag{31}$$

where $l_{d_{wheel}}$ is the gain l_λ of estimator (30), for the estimation of d_{wheel}.

4.2.2. Front sideslip angle estimate

For estimating the front sideslip angle, $\mathbf{X}_f$ is considered. In addition, $\hat{\beta}_f$ is supposed to be invariant during a sampling step. Then, the partial derivative of the odometry twist $\hat{\mathbf{T}}_{f_j}$ with respect to $\hat{\delta}_f$ is given by:

$$\frac{\partial\hat{\mathbf{T}}_{f_j}}{\partial\hat{\delta}_f} = \begin{bmatrix} \hat{v}_{f_j} \frac{\cos\hat{\beta}_{f_j}}{l} \\ 0 \\ \hat{v}_{f_j} \end{bmatrix} \tag{32}$$

where l_{δ_f} is the gain l_λ of estimator (30), for the estimation of δ_f.

5. Trajectory following

This section is dedicated to the reference trajectory frame definition, and to the control of vehicle movements.

5.1. Calculation of the reference frame for the tracking

Algorithms for tracking the $(\mathbf{c}_f)$ frame by the front axle and for tracking the $(\mathbf{c}_r)$ frame by the rear axle are developed in this section.

5.1.1. Definition of the trajectory

The trajectory is supposed to be defined by $\mathbf{x}(u) : \mathbf{R} \mapsto \mathbf{R}^2$ parametrized with u parameter. $\mathbf{x}(\cdot)$ is a piecewise C^3 function. This expression allows a complete definition of all the desired characteristics in terms of tangent, normal, curvature and curvature variation:

$$\begin{cases} \mathbf{t}_x &= \frac{\mathbf{x}'_u}{\|\mathbf{x}'_u\|} \\ \mathbf{n}_x &= \mathbf{t}_{\perp x} \\ c_x &= \frac{\det(\mathbf{x}'_u, \mathbf{x}''_u)}{\|\mathbf{x}'_u\|^3} \\ g_x &= \frac{\det\left(\mathbf{x}'_u, \mathbf{x}'''_u \|\mathbf{x}'_u\|^2 - 3(\mathbf{x}'^t_u \mathbf{x}''_u)\mathbf{x}''_u\right)}{\|\mathbf{x}'_u\|^6} \end{cases} \tag{33}$$

where $(\cdot)'_u$, $(\cdot)''_u$, and $(\cdot)'''_u$ are defining respectively a first, second and third order derivative along u, and $\det(\mathbf{a}, \mathbf{b}) = \mathbf{a}^t_\perp \mathbf{b}$. If u is the curvilinear abscissa s, thus $\|\mathbf{x}'_u\| = 1$ and Eqs. (33) can be simplified as follows:

$$\begin{cases} \mathbf{t}_x &= \mathbf{x}'_s \\ \mathbf{n}_x &= \mathbf{t}_{\perp x} \\ c_x &= \det\left(\mathbf{x}'_s, \mathbf{x}''_s\right) \\ g_x &= \det\left(\mathbf{x}'_s, \mathbf{x}'''_s\right) \end{cases} \tag{34}$$

Usually, the path is defined as a succession of segments and arcs of circles, the bus position on this path varying according to a curvilinear abscissa s. A possible issue is that the implemented control law will tend to prioritize keeping a zero error along one track, including at the end of that track, even if there is a large variation in value of curvature at the next track. This results in errors when changing the track, due in particular to the limitations of actuators. In practice to limit this phenomenon, depending on the size of the tracks and the variation in their curvature, two successive tracks with a significant variation in curvature are subdivided into several smaller tracks in order to have a minimum variation in curvature.

5.1.2. Evolution of the trajectory u parameter

The axes of the trajectory frame are given by $\mathbf{t}_x$ and $\mathbf{n}_x$, and the value of parameter u, corresponding to the reference point to be tracked so that points $\mathbf{c}$ and $\mathbf{x}$ remain as close as possible to each other, has to be determined. For this, several solutions are possible:

- A time law defined by a planner can give the evolution of u. The variable y can then be evaluated as a projection of the vector $\mathbf{xf}$ on the vector $\mathbf{n}_x$; this implies that the vehicle motion control is efficient enough to track the current point.
- The value of u can be established by projecting point $\mathbf{f}$ onto path $(\mathcal{C})$.

In the second case, the following algorithm provides a projection method that avoids a new search at each sampling period. It is a control law whose objective is as follows:

$$\mathbf{x}'^t_u (\mathbf{x} - \mathbf{f}) = 0 \tag{35}$$

Variation of u can then be defined as:

$$\dot{u} = \mu \left[\mathbf{x}'^t_u \mathbf{v}_f - k_u \mathbf{x}'^t_u (\mathbf{x} - \mathbf{f})\right] \tag{36}$$

where $\mu > 0$ is a normalization factor and k_u is a gain regulating the convergence dynamics.

5.2. Velocity and position control along the path

One possibility suitable for vehicle control is to define the path as tracks delimited by pairs $(\mathbf{x}(u_{cib}), v_{cib})$. Then, the vehicle aims at navigating along each track at a desired velocity v_{des} up to point $\mathbf{x}(u_{cib})$, where the waypoint velocity is v_{cib}, these three setpoints being defined *a priori* as part of the initial definition of the path tracks. If $v_{cib} = 0$, the vehicle aims at reaching point $\mathbf{x}(u_{cib})$, while respecting kinematic constraints, such as maximum speed and acceleration (v_{max}, γ_{max}).

The proposed *generation of motion* includes two nested control loops for speed and position.

5.2.1. Position loop

It is assumed that the variation of the curvilinear abscissa δs can be calculated from $\mathbf{x}(u)$ to $\mathbf{x}(u_{cib})$. The velocity v of the frame $(\mathbf{f})$ must not exceed the following velocities in modulus:

- v_{max}, the maximum speed allowed;
- $|v_{des}|$, the desired velocity when the vehicle is not close to $\mathbf{x}(u_{cib})$;
- $\sqrt{2\gamma_{max}|\delta s| + v^2_{cib}}$, the convergence velocity towards the target velocity, without exceeding the acceleration γ_{max} until the point $\mathbf{x}(u_{cib})$.

Considering $v_{sup} = \inf\left(v_{max}, |v_{des}|, \sqrt{2\gamma_{sup}|\delta s| + v^2_{cib}}\right)$, the following reference control velocity is proposed:

$$v_{com} = -v_{sup} \frac{\delta s}{\sqrt{\delta s^2 + \frac{v^2_{sup}}{k^2_s}}} \tag{37}$$

where k_s is a setting parameter and γ_{sup} is an acceleration slightly lower ($\sim$ 10 %) than γ_{max} in order to take into account wheel-ground sliding phenomena.

If $|\delta s|$ is high, then $v_{com} \approx -v_{sup}\text{sign}(\delta s)$; if $|\delta s|$ is small, then $v_{com} \approx -k_s \delta s$. If $v_{cib} \neq 0$, it can be considered that $v_{com} = -v_{sup}\text{sign}(\delta s)$, because a position control is not necessary.

5.2.2. Velocity loop

As soon as the reference v_{com} is known, and considering $\delta v = v - v_{com}$, the reference acceleration can be formulated as follows:

$$\gamma_{com} = -\gamma_{max} \frac{\delta v}{\sqrt{\delta v^2 + \frac{\gamma^2_{max}}{k^2_v}}} \tag{38}$$

where k_v is a setting parameter.

This reference acceleration is finally used to update the speed control output sent to the lower level of the bus, according to the following expression: $v_{out} = v_{out} + \gamma_{com} dt$, with dt the sampling period.

6. Simulation results

In order to get closer to real conditions, a response time of 0.15 s of the steering actuator is added.

6.1. Path tracking without and with sideslip angles correction

The performance of the observer described in Section 4.2 with respect to sliding compensation, including steering offset clearance, is analyzed here. Simulation parameters are defined in Table 1.

The tracking of a **U**-shaped path is illustrated in Fig. 4. The front steering angle of the bus has an uncorrected initial offset. In

Table 1
Path tracking parameters.

Velocity reference	2.0 m/s
Max velocity	2.5 m/s
Max acceleration	0.35 m/s^2
Offset steering angle	0.1 rad
Max steering angle	0.6 rad
Max steering velocity	0.45 rad/s
Front steer offset	0.01 rad
Initial orientation	−0.2 rad
Initial lateral gap	3.0 m
Middle path lateral gap	2.0 m

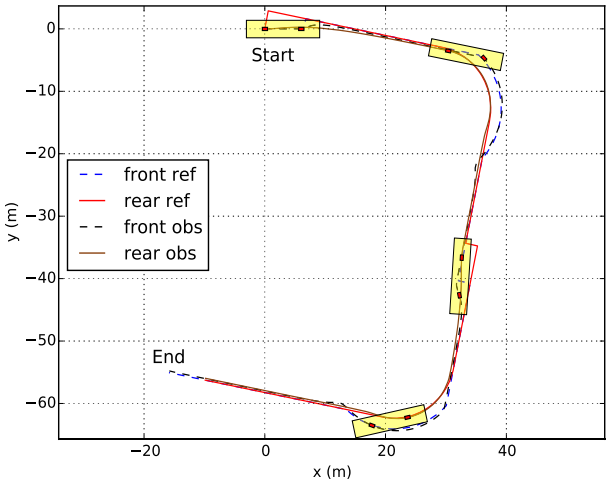


Fig. 4. U trajectory without sliding compensation.

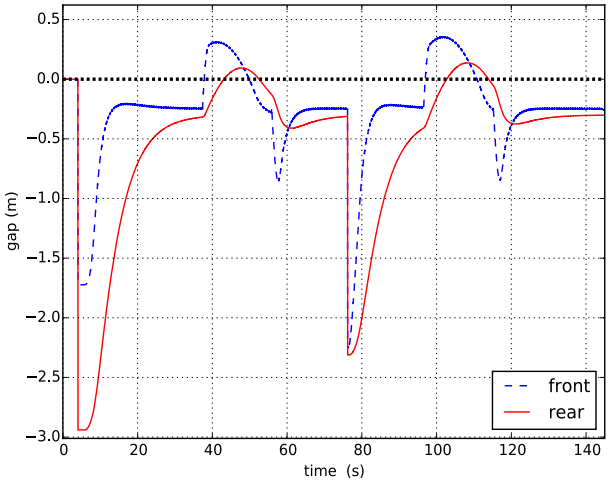


Fig. 5. Tracking error without sliding compensation.

order to facilitate the comparison of results, steering limitations and offsets are set with the same values as on the actual platform.

The displayed path Fig. 4 is generally followed, although Fig. 5 shows a nominal tracking error of about 0.3 m for the front and rear axles. With the sideslip angles correction (see Figs. 6 and 7), offsets are cleared. Lateral gaps are properly corrected, with in particular a solid brown curve that converges better towards the solid red one in Fig. 6.

6.2. Lateral error limits

Lateral errors, at the first turn when following a U-shaped path using the predictive controller, are displayed in Figs. 8 and 9.

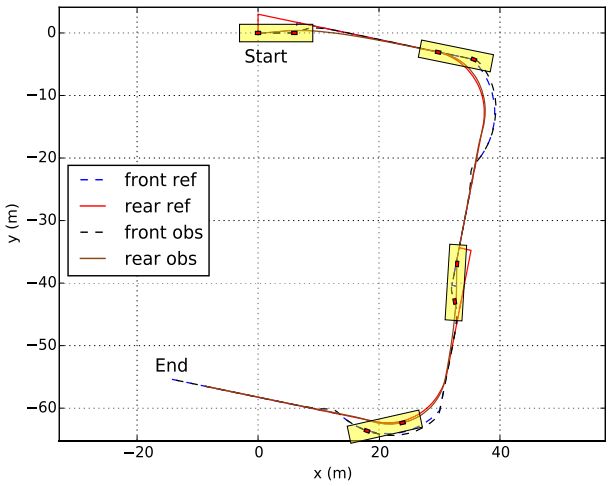


Fig. 6. U trajectory with sliding compensation.

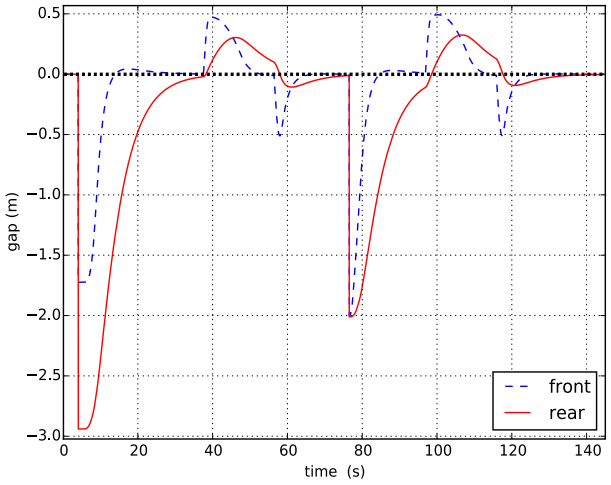


Fig. 7. Tracking error with sliding compensation.

A $\pm$ 0.10 m lateral error constraint has to be met. The vehicle starts moving close to this lateral error limit. Without taking this constraint into account, the vehicle moves beyond, while it does not exceed the limit when handled by the controller.

6.3. Comments

Designed algorithms and kinematic simulations did not take into account several characteristics of the real system dynamics:

- Stresses in the wheel ground interaction can lead to actuation limitations (steering axle rotation impossible when the vehicle is stationary or at very low speed), or kinematic model errors (instantaneous center of rotation offset relative to the theoretical model). Tests with the real vehicle allow these effects to be quantified and corrected if necessary.
- The controller takes into account the maximum longitudinal acceleration and maximum steering axle angular speed characteristics. Preliminary tests on the vehicle assess whether other characteristics such as response time or delays are significant and how they may affect algorithms and settings.
- Observation quality depends on the measurement accuracy and delay. Such criteria are assessed and taken into account by the observation.

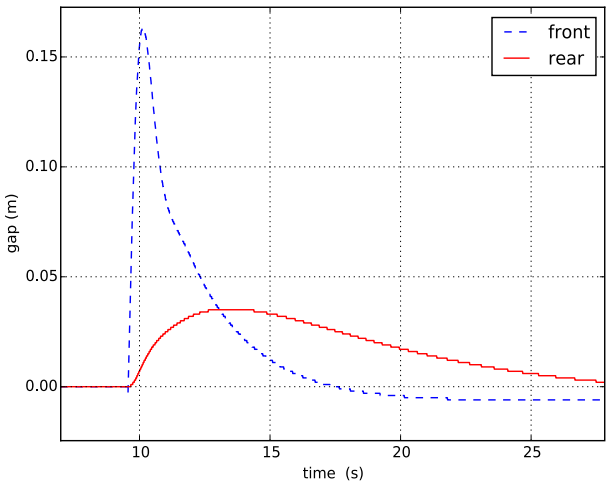


Fig. 8. Lateral limits exceeded.

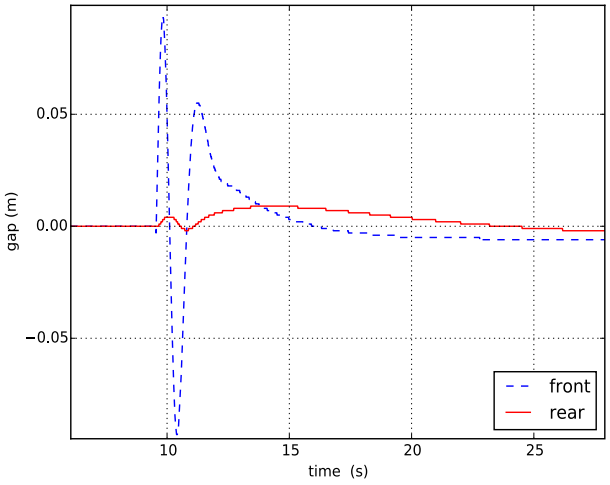


Fig. 9. Respect of lateral limits.

Table 2
Vehicle parameters.

Total mass	$M = 20000$ kg
Length	$L = 12.00$ m
Width	$W = 2.75$ m
Track	$tr = 2.5$ m
Spread	$l_f + l_r = 6.12$ m
Wheel diameter	$d_w = 0.939$ m

7. Experimental results

7.1. An automated bus

The equipped vehicle is shown hereafter in the bus center in Fig. 10. It is a bus dedicated to the daily transport of people. It has rear driven wheels and an Ackermann front steering system. Specially equipped for autonomous driving, this Automotive Intelligent Vehicle (AIV) has the properties indicated in Table 2.

Two stereoscopic camera pairs are used, one for the localization function and one for white line detection and parking space number recognition. The modular system used for locating the bus in the depot is a stereoscopic head composed of two IDS 3240 global shutter type cameras, allowing a wide angle of view necessary for visual SLAM algorithm. The distance between the two cameras is 30 cm. The bus is also equipped with five laser rangefinders for obstacle detection: two fronts and one on each other side of the vehicle.

The on-board controller consists of two controllers interfaced via Ethernet network. One of the controllers is dedicated to performing the localization function with the cameras connected to it; it is an MSI GT73 laptop PC with an Intel core i7 7820HK processor, 16 GB RAM and an SSD disk. The other controller provides the safety functions (obstacle detection), navigation control and interface with the remote bus fleet supervisor via wifi communication, as well as to a low-level control board on the bus via a CAN bus clocked at 10 ms. It is a MicroAutoBox dSPACE controller. The low-level card is connected to the proprioceptive wheel and steering odometry sensors and to the actuators, in parallel with the manual controls (steering wheel and pedals) for drive-by-wire. In particular, it provides steering management (proportional error loop) and speed control (acceleration/braking), in addition to safety functions.

Computation of the bus localization as well as the detection of white lines and the recognition of seat number is a component of Intempora's RTMaps software. This RTMaps localization diagram is executed on the controller dedicated to the localization function. Localization information by the cameras is delivered every 60 ms on average with a maximum delay of 250 ms in case of failure of the camera image analysis. Other components of observation, control, obstacle detection, and interface with the low-level bus control and with the fleet supervisor are implemented as a second RTMaps diagram running on the navigation controller, in which a component dedicated to the observation and control module is installed. Odometry measurements are received through the CAN bus every 10 ms. They include angular increments from the rear wheels sensor and the steering angle of front axle. The position and orientation of the vehicle are then obtained by fusion of those measurements. The control setpoints sent every 10 ms are the velocity v_f of the front axle and the steering angle α_f of the front axle.

7.2. Localization with cameras

A first trajectory in the bus center allows the creation of a 3D cartography of the environment. This mapping is made with the data coming from the two localization cameras used as input of the SLAM type algorithm and whose output is a mapping of the environment including the reconstructed 3D points associated with the camera poses to so-called "key images". Then, online, the inputs are the previously created environment mapping and the data from those cameras. Inputs are processed by a multi-camera SLAM type algorithm with relocation (correspondence between the current view and the previously created cartography) in order to achieve real-time localization at any time. Accuracy between the previously created mapping of the environment and the real-time localization is a few centimeters. During tests, the localization precision is evolving with the position of the bus in the scene according to visual landmarks detected, with degradation in situations of occlusion.

7.3. Steering axles calibration

For the control of the front steering angle, the actuation is electric, controlled by a dedicated low-level controller. The control is performed by means of a loop proportional to the steering angle error, without derivative or integral action. The steering angle is limited to 45° (0.78 rad) and the steering velocity is limited to 30 °s^{-1} (0.52 rad s^{-1}).

Steering sensor calibration process is performed using the sideslip angle observer defined in Section 4.2.2 while the vehicle is moving for 30 s with zero steering angle setpoints at low

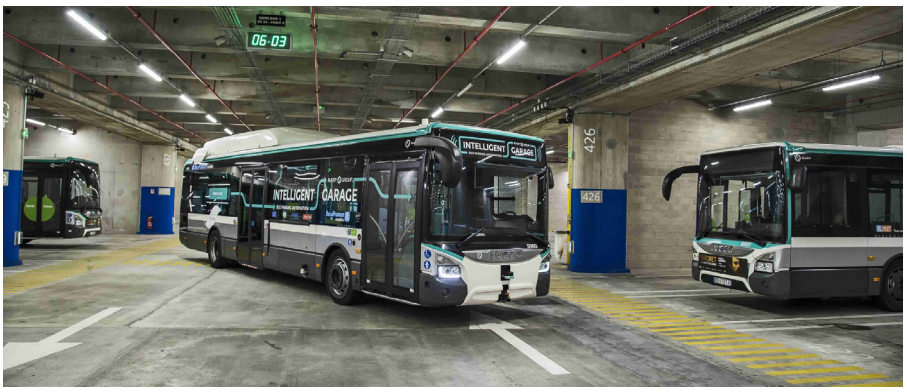


Fig. 10. Bus operating in the bus center.[1]

velocity ($v_f = 0.3$ m s^{-1}) along a white straight reference line for optimal localization, the precision and convergence time of the measurement depending on it.

7.4. Controller tuning

The spatial window S is the discretization step, here chosen at 10 cm for an accurate model. Decreasing it increases the accuracy but also the calculations. Similarly, $n = 20$ is the number of steps chosen to define the prediction horizon $nS = 2$ m. The prediction horizon is chosen greater than the system settling time (2.2 m s^{-1} max speed $\times 0.15$ s steering response time $= 0.33$ m), and large enough to anticipate the trajectory, but not too large to remain within the validity domain of the model. Predictive control parameters, previously defined in Section 3, are set as follows: $Q(0,0) = 20.0$, $Q(1,1) = 122.4$, $Q(2,2) = 224.7$, other coefficients of Q are zero, and $R = 1.0$. Parameters of the diagonal of Q that define the tracking efficiency refer to the lateral error and its two successive derivatives with respect to the curvilinear abscissa. Parameter of R which defines the tracking smoothness refers to the variation in front axle steering relative to the curvilinear abscissa. k_s and k_v are proportional gains for other independent position and speed control loops (see Section 5.2). They are set to $k_s = 0.4$ and $k_v = 50.0$ respectively to allow efficient path tracking at reference velocity and position, with smooth movements. Parameters defined in Table 1 remain the same. All parameters were first tuned on a simulator with regard to the relative importance of performance in different loops, and then adjusted on the real vehicle. With this configuration, over a sampling period of 10 ms the computation time demand of the control algorithm is between 2.5 ms and 4.5 ms with some peaks below 6 ms.

7.5. Experiments in the bus center

All experiments are performed in realistic conditions on asphalt. The bus used for the tests navigates in the bus center in order to carry out a complete parking and exit test, from the entrance of the bus depot to its exit. Thus, the test focus on tracking a path describing a loop with four turns until the parking space, then a parking maneuver, a departure maneuver, and then again four turns up to the exit.

This route taken by the bus is shown in Fig. 11. Fig. 12 zooms in more particularly on parking and departure maneuvers, and Fig. 13 zooms in on position shifting by the camera at time = 528 s. The bus localization at the center of its rear axle is plotted in dotted cyan. This measure is provided by the stereoscopic

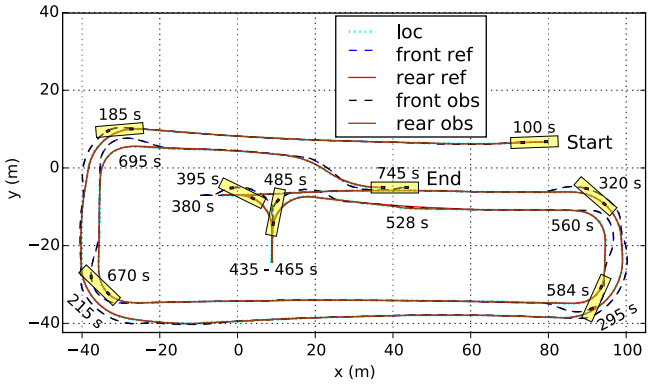


Fig. 11. Path tracking in the bus depot.

cameras installed at the rear of the bus in order to locate itself on the bus center. The accuracy of the camera's position measurement depends on the number of visual landmarks detected. The reference path to be followed at the center of the front axle and at the center of the rear axle are plotted as blue dashed line and solid red line respectively. The estimated bus position at the center of the front axle and at the center of the rear axle are plotted as black dashed line and brown solid line respectively. This estimate is provided by an extended Kalman filter considering the odometry of the wheels and front steering, as well as the position provided by the stereoscopic cameras.

From the starting position at the entrance to the depot until the first turn, the bus runs on flat ground. From the entrance of the first turn at the $x = -20$ m abscissa to the exit of the second turn at $x = -20$ m the bus descends a first floor. He then briefly drives on flat ground until $x = 0$ m, then again he goes down a second floor until he reaches the third turn at $x = 90$ m. Then he drives on flat ground at the end of the third turn, in the fourth turn, in a straight line, and during the parking maneuver. After its departure maneuver, the bus goes up both floors in the same way until it reaches the exit of the bus depot. Tracking errors are plotted in Fig. 14 as a blue dashed line at the center of the front axle and as a solid red line at the center of the rear axle.

A maximum tracking error of almost -40 cm is observed at 528 s for the rear axle. This error is due to a shift in the localization. During the bus departure maneuver, the accuracy of the camera's position measurement is less accurate due to an insufficient number of landmarks, as it sees the wall against which the bus was parked or the parking space next to it. Then, this accuracy becomes optimal at 528 s, when the bus has finished leaving its parking space and more visual markers become visible to the

[1] © RATP - Denis SUTTON - 06/03/2018 - 14887D30.

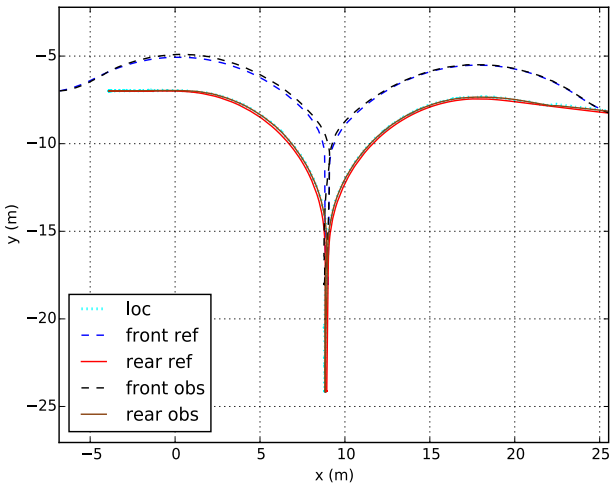


Fig. 12. Zoom on parking and departure maneuvers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

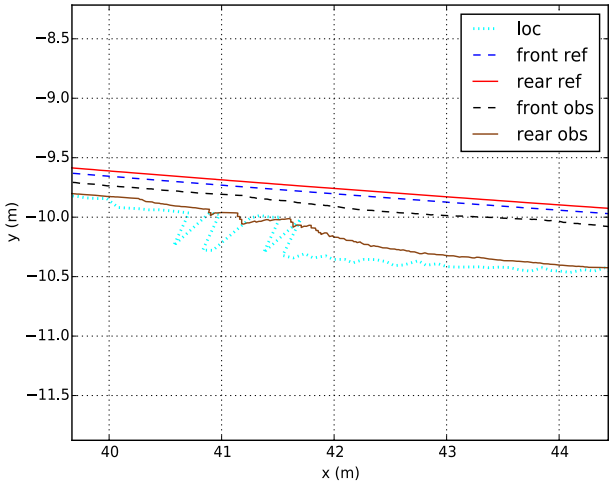


Fig. 13. Zoom at time $t = 528$ s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

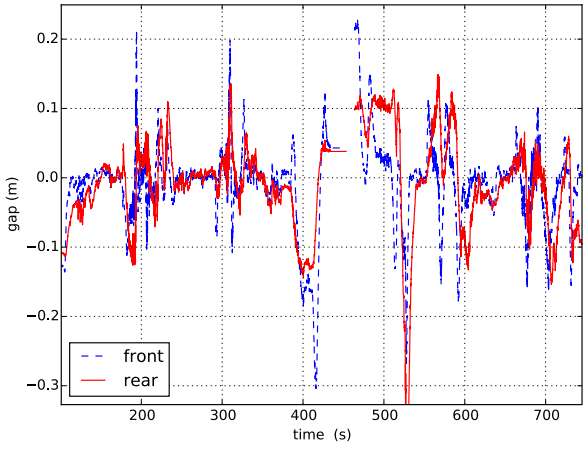


Fig. 14. Front and rear tracking errors.

camera. Therefore, the correction of the actual bus position that is plotted in cyan dotted line in Fig. 13 results in a sudden shift in the bus localization and thus a high instantaneous tracking

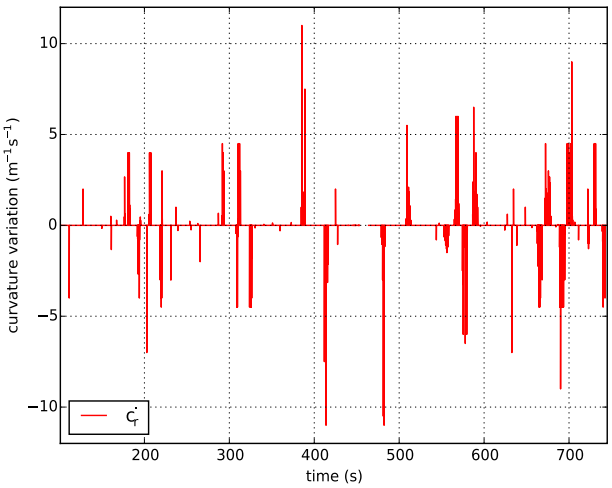


Fig. 15. Rear curvature variation.

error. A +23 cm error of the front axle is also observed at the beginning of the departure maneuver at 465 s due to a difficulty in initializing the localization. Otherwise, shifts in the localization are less than 10 cm everywhere else on the trajectory. And the forthcoming addition of a Sysnav Blue Force inertial unit should make it possible to further improve the robustness (in case of lack of texture, very fast movement or camera occlusion) and accuracy (by merging visual and inertial data) aspects during navigation. Other maximum tracking errors are cumulative localization and control errors, the latter occurring mainly at the entrance or exit of curves when the variation in curvature is highest, for example +21.5 cm for the front axle at the end of the first turn at 194 s or +19.8 cm for the front axle at the end of the third turn at 310 s. Indeed, in the event of such variations, an establishment distance is required to converge on the next part of the path, resulting in an overrun in the variation area. The variation in the curvature of the reference trajectory at the center of the front axle is plotted as a solid red line in Fig. 15. The most important curvature variation values were recorded during the parking and departure maneuvers, the zoom of which is shown in Fig. 12. During the parking maneuver (380 s − 435 s), a front axle tracking error of 6.5 cm is recorded at the turn entry (385 s) at a peak of the curvature variation at +11.00 m^{-1} s^{-1}, and front axle tracking errors of −30.4 cm then 12.1 cm are recorded at the turn exit (412 s) at a peak of the curvature variation at −11.00 m^{-1} s^{-1}, these errors close to the parking position being also due to the localization because of a lack of visibility. During the departure maneuver (465 s−514 s), a front axle tracking error of 15.2 cm is recorded at the entrance to the turn (481 s) during a peak of the curvature variation at −11.00 m^{-1} s^{-1}, and a front axle tracking error of −13.0 cm is recorded at the end of the curve (510 s) at a peak of the curvature variation at +5.50 m^{-1} s^{-1}.

The front longitudinal velocity is reported in Fig. 16, with red dashed line for the control setpoint and blue plain line for wheel odometer measurements, showing oscillations until 0.36 m s^{-1} of amplitude due to an inaccurate odometry wheel sensor used for the low level velocity control. This would need to be improved for better setpoint tracking, especially on slopes, even if this has no significant impact on the accuracy of the trajectory tracking. A negative speed is observed during the parking maneuver. Front steering velocity control input is reported in Fig. 17 in red plain line. Front steering angle is plotted Fig. 18, in red dashed line for the control and in blue plain line for the odometry measure. The control and measure plots are very close. Low oscillations at the end of each turning motion allow a better handling of

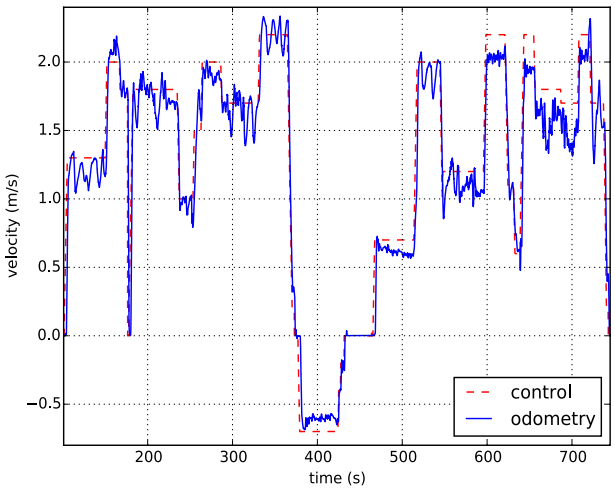


Fig. 16. Front longitudinal velocity.

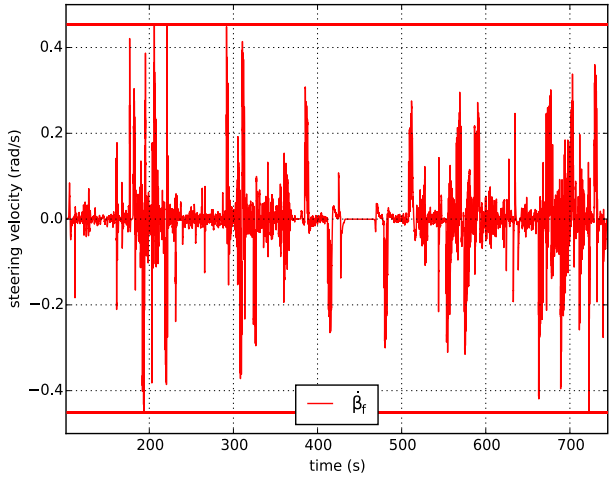


Fig. 17. Front control steering velocity.

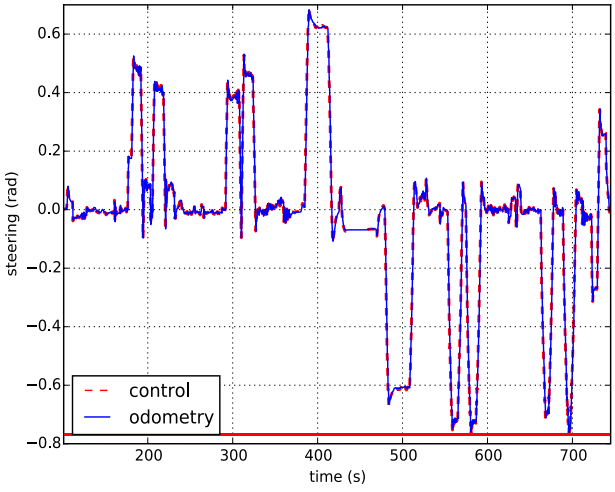


Fig. 18. Front steering angle.

the curvature variation. Steering angle and velocity remain below the maximum setting values of respectively 0.77 rad and 0.45 rad s^{-1}, reported in plain red lines in those figures.

Thus, tracking errors remain between plus or minus 20 cm during the testing, 10 cm because of the localization accuracy and 10 cm because of the controller's accuracy, except for three errors up to 40 cm mainly due to punctual location errors and also to high variations in curvature. Regardless of these performances, a very positive result is their repeatability. This test was repeated several times with always similar results, with a repeatability of less than 4 cm along the entire trajectory. Thus, since the system's performance is known with a trajectory correctly defined *a priori*, public demonstrations were conducted safely in the presence of other moving vehicles (officialvideo).

8. Conclusions and future work

An innovative autonomous guidance system has been implemented for controlling a bus on its trajectory during navigation, including parking and departure maneuvers, in a closed bus center. Design and development of this system was carried out on a real industrial bus.

In order to achieve this objective, predictive control laws have been designed to effectively ensure bounded path-following errors in any situation, thus providing a guarantee of non-collision with the bus center infrastructure. Since these algorithms are linearized around a given curvature value, they remain sensitive to a significant variation in curvature between two path tracks, which must therefore be defined accordingly. In addition, the sliding was treated independently of path tracking errors to maintain accuracy and stability throughout the course. And the software design was carried out in conformity with a safety analysis of the target vehicle's overall system.

The guidance system has been successfully integrated into the industrial vehicle. Several tests were carried out under realistic operational conditions. With a few exceptions close to parking and departure maneuvers, due to poor localization and a strong variation in curvature that limits the performance of the control, tracking errors recorded during the tests remained below 20 cm, including 10 cm due to the control itself, these performances being highly repeatable.

Future work towards an industrial version of the autonomous guidance system will now focus on porting the software to a more cost-effective embedded hardware architecture, designing auto-calibration strategies for correcting sensor biases to maximize system availability in daily use (preliminary solutions presented in [32] and [33]), and designing self-diagnostic strategies for safe and reliable failure detection of system components.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research and development work was carried out in the scope of the European Bus System of the Future 2 (EBSF2) project. This project has received funding from the *European Union's Horizon 2020 research and innovation program* under grant agreement N° 63300, and from the French RATP Group.

References

[1] K.C. Lin, C.D. Shih, J. Li, From rail to railless: Retrofitting servicing buses for safe autonomous public transportation, in: 2019 IEEE International Conference on Embedded Software and Systems, ICESS, 2019, pp. 1–8, http://dx.doi.org/10.1109/ICESS.2019.8782530.

[2] M. Ginn, F. Amuna, J.E. Colmenares, A. Stewart, B. Diong, Y. Wang, J. Yang, Conceptual design and prototyping of a Slim Semi-autonomous Bus Rapid Transit Vehicle, in: SoutheastCon 2017, 2017, pp. 1–6, http://dx.doi.org/10.1109/SECON.2017.7925292.
[3] EBSF 2, Paris | ebsf2.eu, 2018, URL http://ebsf2.eu/demonstration-sites/paris.
[4] S. Zhiwei, H. Weiwei, W. Ning, W. Xiaojun, W.C.Y. Anthony, V.B. Saputra, B.C.H. Quan, C.J. Simon, Z. Qun, Y. Susu, H.B. Siew, Map free lane following based on low-cost laser scanner for near future autonomous service vehicle, in: 2015 IEEE Intelligent Vehicles Symposium (IV), 2015, pp. 706–711, http://dx.doi.org/10.1109/IVS.2015.7225767.
[5] H. Tan, J. Huang, Design of a high-performance automatic steering controller for bus revenue service based on how drivers steer, IEEE Trans. Robot. 30 (5) (2014) 1137–1147, http://dx.doi.org/10.1109/TRO.2014.2331092.
[6] P. Morin, C. Samson, Control of nonholonomic mobile robots based on the transverse function approach, IEEE Trans. Robot. 25 (5) (2009) 1058–1073, http://dx.doi.org/10.1109/TRO.2009.2014123.
[7] D. Pazderski, K. Kozłowski, Motion control of a car-like vehicle with front driving wheels using an approximate decoupling based on the transverse function approach, in: 2017 11th International Workshop on Robot Motion and Control, RoMoCo, 2017, pp. 154–159, http://dx.doi.org/10.1109/RoMoCo.2017.8003907.
[8] F. Bu, H.S. Tan, Pneumatic brake control for precision stopping of heavy-duty vehicles, IEEE Trans. Control Syst. Technol. (2007).
[9] B. Gao, K. Sanada, K. Furihata, A study on yow rate control of hydraulic-power-steering heavy duty vehicles, in: SICE, 2007 Annual Conference, 2007, pp. 2866–2870.
[10] M. Sharafi, A. Zare, A. Haji Vosough, Intelligent parking method for trucks in presence of fixed and moving obstacles, in: 2010 International Conference on Information Networking and Automation, Vol. 1, ICINA, pp. V1–280–V1–284.
[11] C. McCarthy, N. Barnes, R. Mahony, A robust docking strategy for a mobile robot using flow field divergence, IEEE Trans. Robot. 24 (4) (2008) 832–842.
[12] D. Amarasinghe, G. Mann, R. Gosine, Vision-based hybrid control strategy for autonomous docking of a mobile robot, in: Proceedings of IEEE Conference on Control Applications, CCA 2005, Toronto, Canada, 2005, pp. 1600–1605.
[13] D. Pérez-Morales, O. Kermorgant, S. Domí nguez Quijada, P. Martinet, Multi-sensor-based predictive control for autonomous backward perpendicular and diagonal parking, in: 10th Work. Planning, Percept. Navig. Intell. Veh. PPNIV'18 IEEE/RSJ IROS, Madrid, Spain, 2018, pp. 173–180.
[14] S. Cheng, L. Li, H. Guo, Z. Chen, P. Song, Longitudinal collision avoidance and lateral stability adaptive control system based on MPC of autonomous vehicles, IEEE Trans. Intell. Transp. Syst. (2019) 1–10, http://dx.doi.org/10.1109/TITS.2019.2918176.
[15] F. Yakub, Y. Mori, Model predictive control for car vehicle dynamics system – comparative study, in: Third International Conference on Information Science and Technology, Yangzhou, Jiangsu, China, 2013, pp. 172–177.
[16] S.S. Oyelere, The application of model predictive control (MPC) to fast systems such as autonomous ground vehicles (AGV), IOSR J. Comput. Eng. 16 (3) (2014) 27–37.
[17] J. Ji, A. Khajepour, W.W. Melek, Y. Huang, Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints, IEEE Trans. Veh. Technol. 66 (2) (2017) 952–964, http://dx.doi.org/10.1109/TVT.2016.2555853.
[18] M. Krid, F. Benamar, R. Lenain, A new explicit dynamic path tracking controller using Generalized Predictive Control, Int. J. Control Autom. Syst. (2016) 1–10.
[19] Z. Zuo, M. Yang, Y. Wang, A novel dynamic model-based predictive control of unmanned ground vehicles, in: 13th World Congress on Intelligent Control and Automation, Changsha, China, 2018, pp. 461–466.
[20] S. G. Johnson, The NLopt nonlinear-optimization package, 2016, URL https://nlopt.readthedocs.io/en/latest/.
[21] S. Agarwal, K. Mierle, et al., Ceres solver, 2010, URL http://ceres-solver.org.
[22] B.W. Wah, Y.-X. Chen, Constrained genetic algorithms and their applications in nonlinear constrained optimization, in: 12th IEEE Internationals Conference on Tools with Artificial Intelligence, ICTAI, IEEE, Vancouver, BC, Canada, 2000, pp. 286–293, http://dx.doi.org/10.1109/TAI.2000.889884.
[23] E. Lucet, A. Micaelli, A kinematic parameter estimator applied to a bi-directional vehicle, IFAC, Int. Fed. Autom. Control (2020) URL https://www.rayseven.com//r7/runtime/vdi/ifac2020/files/finalpapers/2280/attachments/1857.pdf?1593808155.
[24] K. Berntorp, Particle filter for combined wheel-slip and vehicle-motion estimation, in: 2015 American Control Conference, ACC, 2015, pp. 5414–5419, http://dx.doi.org/10.1109/ACC.2015.7172186.
[25] M. Brossard, A. Barrau, S. Bonnabel, AI-IMU dead-reckoning, IEEE Trans. Intell. Veh. (2020) http://dx.doi.org/10.1109/TIV.2020.2980758.
[26] R. Chaichaowarat, W. Wannasuphoprasit, Wheel slip angle estimation of a planar mobile platform, in: 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics, ICA-SYMP, 2019, pp. 163–166, http://dx.doi.org/10.1109/ICA-SYMP.2019.8646198.
[27] R.M. Murray, Z. Li, S.S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, Boca Raton, 1994.
[28] A. Micaelli, C. Samson, Trajectory Tracking for Unicycle-Type and Two-Steering-Wheels Mobile Robots, (RR-2097) INRIA, 1993.
[29] C. Cariou, R. Lenain, B. Thuilot, M. Berducat, Automatic guidance of a four-wheel-steering mobile robot for accurate field operations, J. Field Robotics 26 (6–7) (2009) 504–518.
[30] R. Lenain, B. Thuilot, C. Cariou, P. Martinet, Adaptive and predictive non linear control for sliding vehicle guidance: application to trajectory tracking of farm vehicles relying on a single RTK GPS, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, IROS, pp. 455 – 460.
[31] R. Lenain, B. Thuilot, C. Cariou, P. Martinet, Sideslip angles observer for vehicle guidance in sliding conditions: application to agricultural path tracking tasks, in: Robotics and Automation (ICRA 2006). Proceedings 2006 IEEE International Conference on, 2006, pp. 3183–3188.
[32] A. Hill, E. Lucet, R. Lenain, A new neural network feature importance method: Application to mobile robots controllers gain tuning, in: ICINCO 2020, 17th Int. Conf. on Informatics in Control, Automation and Robotics, 2020, pp. 188–194.
[33] A. Hill, J. Laneurit, R. Lenain, E. Lucet, Online gain setting method for path tracking using CMA-ES: Application to off-road mobile robot control, in: IROS 2020, International Conference on Intelligent Robots and Systems, Las Vegas, USA, 2020, pp. 7697–7702.



Eric Lucet is researcher in the field of mobile robotics modelization and control. He received the Engineer, the Master and the Ph.D. degrees in mechanics, automatic control and signal processing in 2006 and 2010, respectively, from the engineer's schools ESTP and ENSAM, and from the University UPMC, Paris 6, France. He joined the Robosoft service robots Company in 2006, first as Ph.D. student, then as project manager, being in charge of national and European research projects and scientific coordination. Since 2013, he is now continuing to actively participating to the research for the promotion of the mobile robotics area in the Robotics Unit of the French Atomic Energy Commission (CEA).



Alain Micaelli received the Engineer and the Ph.D. degrees in automatic control and signal processing in 1979 and 1982, respectively, from the Ecole Nationale Supérieure des Télécommunications, Paris, France, and from the University of Paris-Sud, France. He joined the Robotics Unit of the French Atomic Energy Commission (CEA) in 1982 and has been involved in several national and international projects dealing with teleoperation and mobile robotics. He is now Research Director in the field of automatic control. His research interests include the control of manipulators, telemanipulators, mobile robots, Virtual Reality and more specifically Virtual Manikin.



François Xavier Russotto is a research engineer and project leader. He has worked at Thales Optronics SA, as a development engineer working on control and on-board electronics for aircraft embedded systems, and at PSA Peugeot Citroën, as a project manager on development of innovative by-wire man–machine interfaces for tourism vehicles, before joining CEA LIST in 2002. He managed a research activity focused on Supervisory Control for Telerobotics Systems, leading to development of a SCADA software currently in industrial development phase for integrating operational robotics equipment of AREVA NC La Hague. He is now the head of the robotic control and supervision laboratory of CEA LIST.