The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

On-line reference trajectory generation for manually convoying a platoon of automatic urban vehicles

P. Avanzini, B. Thuilot, T. Dallej, P. Martinet and J.P. Derutin
LASMEA - 24 avenue des Landais - 63177 Aubière - FRANCE
Pierre.AVANZINI@lasmea.univ-bpclermont.fr

Abstract— **Various "Urban Transportation Systems" are currently in developing, in order to put forward solutions to congestion and pollution in dense areas. Autonomous electric vehicles in free-access can be seen as an attractive approach, in view of the large flexibility that can be expected. One instrumental functionality linked to this solution is platoon motion: several autonomous vehicles accurately follow the trajectory of a manually driven first vehicle, with pre-specified inter-distances. A global decentralized platoon control strategy, supported by inter-vehicle communications and relying on nonlinear control techniques is here proposed. Each vehicle is controlled with respect to the same smooth reference trajectory, inferred on-line from the motion of the first vehicle via B-spline optimization. Experimental results, carried out with four urban vehicles, demonstrate the capabilities of the proposed approach.**

Index Terms— **mobile robots, nonlinear control, platooning, automatic guided vehicles, trajectory generation, path following.**

I. INTRODUCTION

Congestion of vehicle traffic in dense areas, with correlated pollution and waste of time, is currently a serious concern. As a consequence, new alternative transport systems are in developing. Autonomous electric vehicles in free-access appear as a promising approach, especially when the public demand is properly structured, e.g. commutations within inner-cities or large industrial estates. The large flexibility that can be obtained with such a transport system (commutation at any time and along any route) is definitely its main attractive feature and should meet user expectations.

One functionality of special interest which can enhance this transport system is automated platooning, i.e. several autonomous vehicles following the trajectory of a first one, with pre-specified inter-distances. Such a functionality, on the one hand allows to easily adapt the transport offer to the actual need (via platoon length), and on the other hand eases maintenance operations, since only one person (driving the first vehicle) can then move several vehicles at a time (e.g. to bring vehicles back to some station). Moreover, since cooperative navigation can ensure more coherent motions, an increase in traffic as well as an enhancement in safety can be expected. Platooning is therefore considered in this paper.

Different approaches can be proposed. They can be classified into three categories, according to the information used for vehicle control: local, semi-global and global strategies.

The most standard approaches rely on ***local strategies***, i.e. each vehicle is controlled exclusively from data relative to the immediate front vehicle. They are also named *leader-follower approaches*. For instance, visual tracking of the preceding vehicle has been proposed in [1] and generic control laws have been designed in [2] and [3].

Some approaches based on a structural analogy constitute ***semi-global strategies***: neighboring vehicles (and not only the preceding one) are taken into account in a formation criterion, evaluated from direct measurements. For instance, a *virtual structure approach*, characterized by a serial chain of spring-damper, is proposed in [4] and a control law is then derived from the combined front and rear virtual forces.

These two strategies present however some drawbacks, the most concerning one being error accumulation: the servoing errors, induced by sensor noises and/or actuator delays, are inevitably growing from the first vehicle to the last one, leading to unacceptable oscillations. Such problems can be overcome by considering ***global strategies***, i.e. each vehicle is now controlled from the data received from all vehicles, collected from appropriate communication channels. Most of the *virtual structure approaches* belong to this category. In [5], a mechanical analogy is used to design feedback controllers to achieve straight line motion. A single virtual rigid structure is also considered in [6], relying on graph theory. Nevertheless, since these techniques aim at imposing some pre-specified geometric pattern, the vehicles cannot all reproduce the trajectory of the first one: in a curve, if vehicles are kept according to a straight line pattern, they all describe a distinct trajectory. In contrast, in previous work [7], a trajectory-based global control strategy has been proposed: relying on nonlinear control techniques, vehicle lateral and longitudinal control have been exactly decoupled. Lateral guidance of each vehicle with respect to the same reference path can then be achieved independently from longitudinal control, designed to maintain a pre-specified curvilinear vehicle inter-distance.

In this paper, this work is enhanced in order that the platoon reference path can be defined on-line, from the trajectory of a manually driven first vehicle. The objective is to manually convoy a platoon of automatic vehicles, following accurately the trajectory chosen by the driver of the first one. Such a manual guidance mode is very attractive, for public transportation as well as for maintenance operations, since vehicles can then instantaneously be driven along any route, without requiring a previously recorded reference trajectory.

On-line trajectory generation and its application to vehicle platooning is therefore specifically addressed in this paper. Trajectory generation has been extensively discussed

in robotics community (see [8], [9], [10] for recent surveys). With respect to platooning applications, high-order continuous trajectories (at least C^2) are required, and moreover analytic representations are mandatory (in order that the point on the reference trajectory the closest to the vehicle controlled point can be accurately evaluated). In view of these requirements, the approaches derived from the *Computer Aided Design* (CAD) community appear quite attractive: in [11], [12], [13] parametric functions like Bezier and B-Spline curves are used to provide smooth curves according to boundary conditions. Constraints on curvature are added in [14]. Trajectory modification in dynamic environment and on-line navigation have also been addressed in [15], [16], [17]. With respect to these techniques, the specificities of the considered application (the trajectory has to be generated on-line and must be as close as possible to the actual first vehicle trajectory) have led to propose an extension procedure based on B-Spline curves.

The paper is organized as follows: the global decentralized control strategy for vehicle platooning is first sketched in Section II. The trajectory generation process is then described in Section III. Finally, in Section IV, simulations investigate the sensitivity of this process to some parameter values and experiments carried out with four electric vehicles demonstrate the capabilities of the proposed approach.

II. Global decentralized control strategy

A. Modeling assumptions

Urban vehicles involved in platooning applications are supposed to move at quite low speed (less than 5m.s^{-1}) on asphalted roads. Therefore dynamic effects can be neglected and a kinematic model can satisfactorily describe their behavior, as corroborated by extensive tests performed with our experimental vehicles shown in Fig. 1.

Fig. 1. Experimental vehicles: two Cycab leading two RobuCab

In this paper, the kinematic tricycle model is considered: the two actual front wheels are replaced by a unique virtual wheel located at the mid-distance between the actual wheels. The notation is illustrated in Fig. 2.

- Γ is the common reference path for any vehicle (to be inferred from the trajectory of the first one), defined in an absolute frame $[A, X_A, Y_A]$.
- O_i is the center of the i^{th} vehicle rear axle.
- M_i is the closest point on Γ to O_i.
- s_i is the arc-length coordinate of M_i along Γ.
- $c(s_i)$ is the curvature of path Γ at M_i, and $\theta_\Gamma(s_i)$ is the orientation of the tangent to Γ at M_i w.r.t. $[A, X_A, Y_A]$.

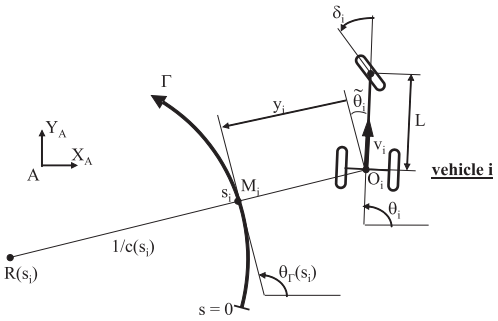

Fig. 2. Tricycle model description

- θ_i is the heading of i^{th} vehicle w.r.t. $[A, X_A, Y_A]$.
- $\tilde{\theta}_i = \theta_i - \theta_\Gamma(s_i)$ is the angular deviation of the i^{th} vehicle w.r.t. Γ.
- y_i is the lateral deviation of the i^{th} vehicle w.r.t. Γ.
- δ_i is the i^{th} vehicle front wheel steering angle.
- L is the vehicle wheelbase.
- v_i is the i^{th} vehicle linear velocity at point O_i.
- l is the number of vehicles in the platoon, i.e. $i \leq l$.

B. Vehicle state space model

The configuration of the i^{th} vehicle can be described without ambiguity by the state vector $(s_i,\, y_i,\, \tilde{\theta}_i)$. The current values of these variables can be inferred on-line by comparing vehicle absolute localization to the reference path. It can then be shown (see [18]) that tricycle state space model is:

$$\begin{cases} \dot{s}_i = v_i \, \frac{\cos\tilde{\theta}_i}{1 - y_i \, c(s_i)} \\ \dot{y}_i = v_i \, \sin\tilde{\theta}_i \\ \dot{\tilde{\theta}}_i = v_i \left(\frac{\tan\delta_i}{L} - \frac{c(s_i) \, \cos\tilde{\theta}_i}{1 - y_i \, c(s_i)} \right) \end{cases} \tag{1}$$

Platooning objectives can then be described as ensuring the convergence of y_i and $\tilde{\theta}_i$ to zero, by means of δ_i, and maintaining the gap between two successive vehicles to a fixed value d^*, by means of v_i. It is considered that $y_i \neq \frac{1}{c(s_i)}$ (i.e. vehicles are never on the reference path curvature center). In practical situations, if the l vehicles are well initialized, this singularity is never encountered.

C. Control law design

In previous work [7], it has been shown that exact linearization techniques offer a relevant framework to address platoon control: equations (1), as most of kinematic models of mobile robots, can be converted in an exact way into a so-called chained form, see [18]. Such a conversion is attractive, since the structure of chained form equations allows to address independently lateral and longitudinal control.

Steering control laws δ_i can first be designed to achieve the lateral guidance of each vehicle within the platoon. In these control laws, v_i just appears as a free parameter. Since conversion of equations (1) into chained form is exact, all nonlinearities are explicitly taken into account. High tracking performances (accurate to within $\pm$5cm when relying on an RTK GPS sensor) can then be ensured, whatever initial errors or reference path curvature are. Details can be found in [19].

Control variables v_i can then be designed to achieve longitudinal control. In nominal situation, the objective for the i^{th} vehicle is to regulate $e_i^1 = s_1 - s_i - (i-1)\,d^*$, i.e. the arc-length longitudinal error w.r.t. the leader. This control objective is attractive, since the location s_1 of the leader represents a common index for all the vehicles into the platoon, so that error accumulation and inherent oscillations can be avoided. In addition, since it is an arc-length error, this control objective remains consistent whatever the reference path curvature is (in contrast with euclidian inter-distances). Nevertheless, for obvious safety reasons, the location of the preceding vehicle cannot be ignored. Therefore, in previous work [7], the longitudinal control law has been designed to control a composite error: a smooth commutation function gives the predominance either to the global error e_i^1 or to the local one $e_i^{i-1} = s_{i-1} - s_i - d^*$ according to some security distance. Once more, exact linearization techniques have been used, so that nonlinearities in equations (1) are still explicitly accounted, ensuring high accurate regulation. More details, as well as experiment results carried out with Cycab and RobuCab vehicles (see Fig. 1), relying on RTK GPS sensors for vehicle localization and WiFi technology for inter-vehicle communications, can be found in [7].

III. TRAJECTORY GENERATION

The error variables to be regulated in lateral and longitudinal control are either a gap w.r.t. the reference trajectory (i.e. y_i and $\tilde{\theta}_i$) or a measurement w.r.t. this trajectory (i.e. s_i). As a result, the reference trajectory representation must exhibit the following features:

- it should accept an analytic expression, in order to enable an accurate computation of the error variables,
- it should be at least C^2, since the reference path curvature $c(s_i)$ is needed in control laws.

In order to meet these requirements, it is here proposed to rely on B-Spline curves. The raw data from which these curves have to be inferred is the set, denoted Ω, of successive absolute localizations of the first vehicle. The difficulty lies in extending the reference trajectory without modifying what had been previously built, in order for any variable (s_i, y_i, $\tilde{\theta}_i$, etc.) to keep consistent values although the first vehicle is moving. Approximation of raw data by B-Spline is first recalled, and then the extension process is described.

A. B-Spline optimization

B-Spline curves consist in the concatenation of 2-dim. polynomial curves $Q^i(t) = (\, Q_x^i(t)\,,\, Q_y^i(t)\,)$, with $1 \le i \le n$ and $t \in [0,1]$. Each polynomial $Q^i(t)$ is a linear combination of basis polynomials $\{b_j(t)\}_{(0\le j\le d)}$ whose degree d and coefficients are selected such that continuity constraints at the connection between $Q^i(t)$ and $Q^{i+1}(t)$ are satisfied. The i^{th} B-Spline curve $Q^i(t)$ can then be expressed as:

$$Q^i(t) = \begin{pmatrix} b_0(t) & \dots & b_d(t) \end{pmatrix} \begin{pmatrix} P_x^i & P_y^i \\ \vdots & \vdots \\ P_x^{i+d} & P_y^{i+d} \end{pmatrix} \quad (2)$$

The coefficients $(\, P_x^k\,,\, P_y^k\,)_{(i\le k\le i+d)}$ of the linear combination are called control points. They shape the B-Spline curve $Q^i(t)$, as shown in Fig. 3.

Approximating raw data Ω consists then in finding optimal values for $(\, P_x^k\,,\, P_y^k\,)$, such that B-Spline $\{Q^i(t)\}_{(1\le i\le n)}$ fits at best with the raw trajectory Ω. More precisely, let Ω^i be the subset of Ω associated with the B-Spline curve $Q^i(t)$ and $(\, \Omega_x^{i,j}\,,\, \Omega_y^{i,j}\,)_{(j\in J_i)}$ be the J_i raw coordinates within Ω^i. Then, optimal control points $(\, P_x^k\,,\, P_y^k\,)$ are obtained by minimizing criterion (3) via standard least square method.

$$\sum_{i=1}^{n} \sum_{j\in J_i} \left[(\Omega_x^{i,j} - Q_x^i(t_j))^2 + (\Omega_y^{i,j} - Q_y^i(t_j))^2 \right] \quad (3)$$

B. Local optimization criterion

When the first vehicle is moving, its successive absolute localizations are constantly filling Ω. Let Ω_T be the raw data available at instant T. Ideally, each time Ω_T is upgraded, criterion (3) should be minimized, supplying new control point coordinates and possibly demanding for the addition of new control points. However, in order to limit the modifications on the reference trajectory previously built, only the most recent polynomials $Q^i(t)$ are here updated. Two additional advantages are a bounded computing time (since optimization duration increases with the number of optimized control points), and a guarantee that inter-vehicle communication band-width is not saturated (since a bounded number of control points has then to be transmitted).

More precisely, let n_{ac} *(as active curves)* be the number of polynomials $Q^i(t)$ entering into the optimization process. The local criterion (deduced from (3)) is:

$$\sum_{i=n-n_{ac}+1}^{n} \sum_{j\in J_i} \left[(\Omega_{T,x}^{i,j} - Q_x^i(t_j))^2 + (\Omega_{T,y}^{i,j} - Q_y^i(t_j))^2 \right] \quad (4)$$

Finally, let n_{ap} *(as active points)* be the number of control points $(\, P_x^k\,,\, P_y^k\,)$ whose coordinates are to be modified when minimizing (4). It can be observed in (2) that each polynomial $Q^i(t)$ is defined from $d+1$ control points :

- $Q^1(t)$ is defined from $(P_x^1,\, P_y^1), \dots, (P_x^{d+1},\, P_y^{d+1})$,
- $Q^{n-n_{ac}+1}(t)$ is defined from: $(P_x^{n-n_{ac}+1},\, P_y^{n-n_{ac}+1}), \dots, (P_x^{n-n_{ac}+d+1},\, P_y^{n-n_{ac}+d+1})$,
- $Q^n(t)$ is defined from: $(P_x^n,\, P_y^n), \dots, (P_x^{n+d},\, P_y^{n+d})$.

Therefore, in order to freely shape n_{ac} polynomials $Q^i(t)$ when minimizing (4), then it must be chosen $n_{ap} = n_{ac} + d$ [1]. The drawback of such a choice is that $n_{ac}+d$ polynomials ($Q^{n-n_{ac}-d+1}(t)$ to $Q^n(t)$) are then altered, when the minimization of only n_{ac} is explicitly addressed in (4). The fitting performances on ($Q^{n-n_{ac}-d+1}(t)$ to $Q^{n-n_{ac}}(t)$) can be damaged, and consequently the overall fitting performances.

In contrast, if it is chosen $n_{ap} = n_{ac}$, then criterion (4) cannot be driven to its minimum[2]. Therefore, the local fitting performances (and thus the overall one) can also be damaged.

The best compromise on (d, n_{ac}, n_{ap}) values is discussed, via extensive numerical simulations, in Section IV.

[1] $(P_x^{n-n_{ac}+1},\, P_y^{n-n_{ac}+1})$ to $(P_x^{n+d},\, P_y^{n+d})$ should be free

[2] since $(P_x^{n-n_{ac}+1},\, P_y^{n-n_{ac}+1})$ to $(P_x^{n-n_{ac}+d},\, P_y^{n-n_{ac}+d})$ can no longer be adjusted

C. Implementation details

When a new localization data is available, two possibilities have to be considered. They are illustrated in Fig. 3, when $d = 3$, $n_{ac} = 3$ and $n_{ap} = 4$.

- either the arc-length along the last polynomial $Q^n(t)$ is lower than a threshold $c_1 > 0$. Then, the new raw data is just incorporated into the last subset Ω_T^n, see top-figure 3, and the minimization of criterion (4) supplies optimized control point coordinates.
- or the arc-length along $Q^n(t)$ exceeds threshold c_1. Then, subset Ω_T^n is split in such a way that the arc-length of the new last subset Ω_T^{n+1} is $c_1 - c_2$, with $0 < c_2 < c_1$. A new control point is added, when the oldest active one is fixed, see bottom-figure 3. According to this procedure, the arc-length of any subset, except the last one, is c_2. Finally, optimized control points coordinates are obtained by minimizing criterion (4).

In order to reduce computing time, new raw localizations are taken into account only if their distance from the last data entered into Ω_T is superior to some minimum threshold. Finally, in order to be able to compute the arc-length coordinate s_1 of the first vehicle between two raw acquisitions, its motion is predicted on some short horizon (relying on model (1)), and predicted locations are incorporated into Ω_T before minimizing criterion (4), see Fig. 3.

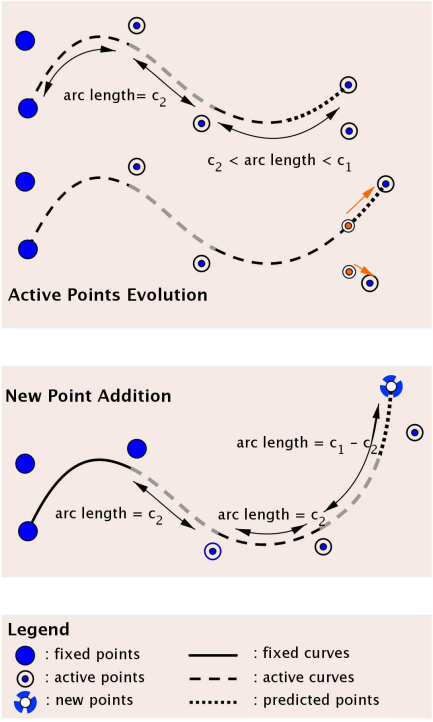

Fig. 3. B-Spline extension process

IV. SIMULATIONS AND EXPERIMENTAL RESULTS

This section is divided into two parts: first optimization results are presented to investigate the sensitivity of the on-line reference trajectory generation to its main parameters, and eventually propose their most suitable values. Then, full-scale experiments carried out with four vehicles are reported to demonstrate the performances of the proposed approach.

A. Simulation results

In order for the simulations to be representative of actual conditions, an experimental vehicle has first been manually driven and the raw localization data supplied by its RTK GPS receiver have been recorded (the vehicle and the GPS sensor are both described in Section IV-B). The recorded trajectory is shown in Fig. 4.

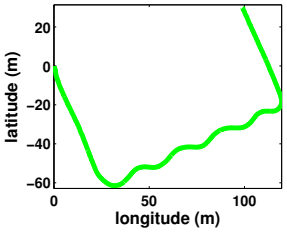

Fig. 4. Trajectory for simulations

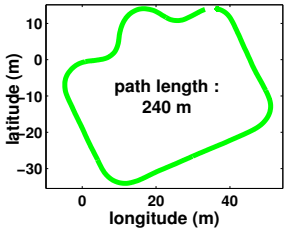

Fig. 5. Trajectory for experiments

First, a posteriori optimizations (i.e. minimization of criterion (3)) have been performed, in order to investigate the influence of parameters d and c_2. Clearly, parameter c_2 has to depend on both the vehicle velocity and the acquisition rate. Since raw localization data have here been recorded at a 10Hz sampling frequency, with a limited measurement noise (the accuracy of the RTK GPS is 2cm), when the vehicle was driven at a velocity of 1m.s^{-1}, $c_2 \in [1.5\,,\,5]$ appear as consistent values (this means 15 to 50 raw data within each subset Ω^i). Next, since the rate of change in the reference path curvature has to be bounded (in view of the dynamic constraints on actual vehicles), it is not useful to investigate large values for parameter d: tests have therefore been limited to $d \in \{3, 4, 5\}$. The average and the maximum euclidian distances between the raw data (i.e. the actual vehicle locations) and their projection on the proposed reference trajectory (i.e. the B-Spline) are supplied in Fig. 6. As it could have been intuitively expected, whatever the value of d is, the best approximations are obtained with the smallest value of c_2. Moreover, they are very satisfactory, since the absolute average error is less than 5mm.

d	c_2 (m)	maximum error (m)	average error (m)
3	**1.5**	**0.0231**	**0.0047**
3	3	0.0372	0.0077
3	5	0.1078	0.0224
4	**1.5**	**0.0265**	**0.0044**
4	3	0.0439	0.0099
4	5	0.0832	0.0222
5	**1.5**	**0.0205**	**0.0046**
5	3	0.0362	0.0077
5	5	0.0855	0.0216

Fig. 6. Influence of d and c_2 on a posteriori optimization

c_2 (m)	n_{ap}	d	n_{ac}	maximum error (m)	average error (m)	average absolute variations in distance (m)	direction (rad)	curvature (m^{-1})
1.5	3	3	3	0.0621	0.0088	0.0025	0.0039	0.0103
"	4	3	4	0.0559	0.0087	0.0056	0.0102	0.0234
"	4	3	3	0.0533	0.0107	0.0071	0.0127	0.0316
"	4	4	4	0.0722	0.0130	0.0052	0.0096	0.0232
"	**5**	**3**	**5**	**0.0466**	**0.0060**	**0.0028**	**0.0057**	**0.0145**
"	5	4	4	0.0557	0.0088	0.0044	0.0082	0.0193
"	5	4	5	0.0357	0.0077	0.0034	0.0067	0.0166
"	5	5	3	0.0676	0.0121	0.0053	0.0089	0.0208
"	6	3	6	0.0453	0.0069	0.0041	0.0081	0.0183
"	**6**	**4**	**6**	**0.0383**	**0.0075**	**0.0038**	**0.0073**	**0.0165**
"	6	5	6	0.0483	0.0088	0.0043	0.0083	0.0191
"	**8**	**5**	**8**	**0.0449**	**0.0079**	**0.0042**	**0.0080**	**0.0174**
1	5	3	5	0.2041	0.0110	0.0056	0.0191	0.6591

Fig. 7. Influence of n_{ac} and n_{ap} on on-line optimization

Next, on-line optimizations (i.e. minimization of criterion (4)) have been performed with the most satisfactory choices for (d, c_2) (shown in bold in Fig. 6), in order to investigate the influence of parameters n_{ac} and n_{ap}. The most significant results are reported in Fig. 7.

Maximum and average error values show clearly that, for a given d, the higher n_{ac} value is, the more the on-line optimization is accurate, as it could have been expected. However, beyond some threshold (depending on the value of d), increasing n_{ac} value does no longer improve significantly the quality of the approximation. It can also be observed that, for any given d, the most satisfactory results are obtained when $n_{ap} = n_{ac}$. This means that modifying polynomials $Q^i(t)$ which are not explicitly considered in local criterion (4) (cases $n_{ap} > n_{ac}$) is more harmful, with respect to the optimization accuracy, than minimizing criterion (4) with a non-maximum degree of freedom (case $n_{ap} = n_{ac}$).

According to the on-line optimization process, the last n_{ac} polynomials $Q^i(t)$ are always varying. Consequently, the reference path for the first follower vehicles cannot be perfectly constant. However, the average absolute variations in distance, in direction and in curvature between the reference paths supplied at two successive iterations are quite small, as shown in Fig. 7. And once more, the best results are obtained when $n_{ap} = n_{ac}$.

Finally, if the lowest value for n_{ap} is sought (in order to reduce the computing time), then the parameter set (d=3, n_{ac}=n_{ap}=5 and c_2=1.5m) appears to be the most appropriate and has therefore been used in the experiments reported in the next section. A shorter value $c_2 = 1$m had also been investigated. However, as can be seen in Fig. 7, the results are poor, especially w.r.t. variations in curvature. For the current application context, $c_2 = 1.5$m appears as a lower bound.

B. Experiments

Several experiments have been carried out in Clermont-Ferrand on "PAVIN Site", an open platform devoted to urban transportation system evaluation. The video attachment presents some sequences recorded during the experiments.

1) Experimental set-up: The experimental vehicles are shown in Fig. 1. They are electric vehicles, powered by lead-acid batteries providing 2 hours autonomy. Two *(resp. four)* passengers can travel aboard the Cycab *(resp. the RobuCab)*. Their small dimensions (length 1.90m, width 1.20m) and their maximum speed (5m.s^{-1}) are appropriate to urban environments. On-board RTK GPS receivers provide absolute localization measurements accurate to within 2cm at a 10Hz sampling frequency. Inter-vehicle communication is ensured via WiFi technology. Platoon control laws are implemented in C++ language on Pentium-based computers using RTAI-Linux OS.

2) Experimental results: The experiments have been carried out with four vehicles. The leading one was manually driven along the 240m-long path shown in Fig. 5, at a constant velocity of 1m.s^{-1}. The accuracy of the on-line reference trajectory generation is shown in Fig. 8: the average and maximum errors between the raw localization data of the first vehicle and their projection on the B-Spline are resp. 0.76cm and 5.49cm. These figures are quite similar to what was expected from simulation trials, see Fig. 7.

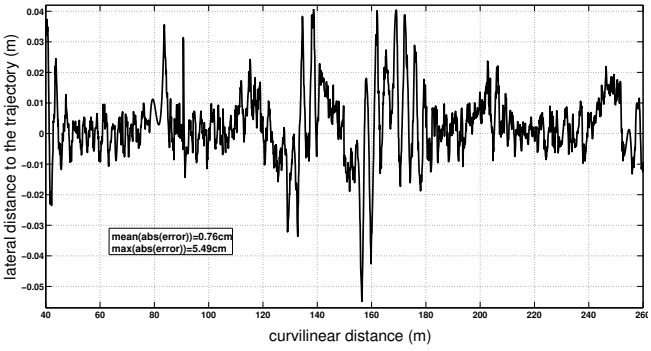

Fig. 8. Error in the approximation of the first vehicle trajectory

The lateral deviations of each vehicle w.r.t. the raw positions of the leader are shown in Fig. 9. It can be observed that the behavior of vehicle 2 is slightly different from those of vehicles 3 and 4. This small difference is due to the

fact that the reference trajectory is still slightly varying for vehicle 2 (since it is close to the leader), when it has become constant for vehicles 3 and 4. However, the lateral guidance of vehicle 2 is roughly as accurate as for vehicles 3 and 4, and as satisfactory as in previous work, when all vehicles were guided w.r.t. a pre-specified reference trajectory, see [19].

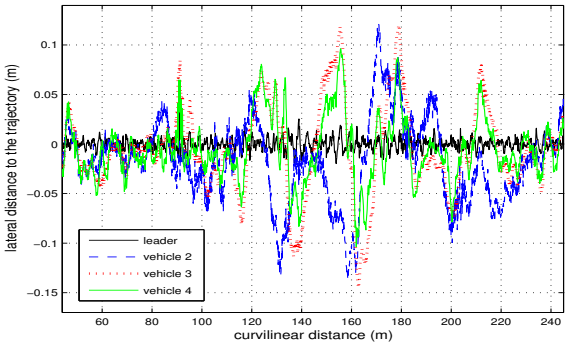

Fig. 9. Vehicle lateral deviations

Finally, the accuracy of the longitudinal control laws is investigated in Fig. 10. Once the platoon is in nominal mode (i.e. all vehicles have reached their desired inter-distances), the behavior is identical to what was observed in previous work, when vehicles were guided w.r.t. a pre-specified reference trajectory (see [7]), namely a 10cm accuracy: the on-line reference trajectory generation does not disturb the longitudinal performances. Behaviors when the platoon is initialized or when the first vehicle stops abruptly are also identical to what is reported in [7].

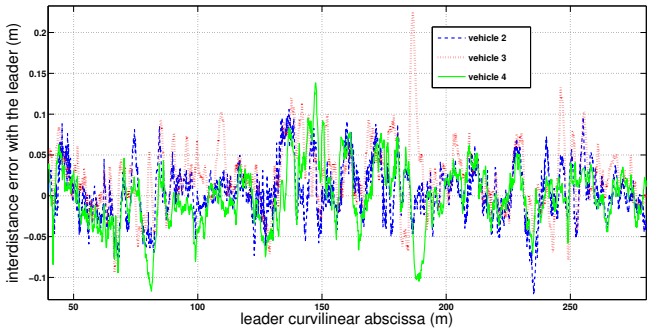

Fig. 10. Vehicle inter-distance errors

V. CONCLUSION AND FUTURE WORK

In this paper, vehicle platooning with respect to a manually driven leader, i.e. each autonomous vehicle must accurately follow the trajectory chosen by the driver of the first one with pre-specified inter-vehicle distances, has been addressed. A global decentralized control strategy, taking advantage of inter-vehicle communications, has been proposed in order to avoid error accumulation inherent to local control approaches. An on-line reference trajectory generation, relying on B-Spline curves, has been developed to accurately represent the first vehicle actual trajectory. Since only a fixed number of control points has to be adjusted at each step, computing time is bounded and communication channel is ensured not to be saturated. Then, nonlinear control techniques have been considered to take explicitly into account the nonlinearities in vehicle models, in order to enable high accurate guidance. Finally, full scale experiments, carried out with four vehicles, have demonstrated the efficiency of the proposed approach.

Current developments aim at refining the on-line reference path generation: enhanced criteria, with adaptive subset length (instead of a constant length c_2), are in developing in order to reduce the number of control points with nevertheless an identical or higher accuracy of the reference path approximation.

REFERENCES

[1] S. Benhimane and E. Malis. Vision-based control for car platooning using homography decomposition. In *IEEE International Conference on Robotics and Automation*, pages 2173–2178, 2005.

[2] D. Wang and M. Pham. Unified control design for autonomous car-like vehicle tracking maneuvers. In *Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications*, 2006.

[3] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 2008.

[4] S.Y. Yi and K.T. Chong. Impedance control for a vehicle platoon system. *Mechatronics*, 2004.

[5] R. Caicedo, J. Valasek, and J.L. Junkins. Preliminary results of one-dimensional vehicle formation control using structural analogy. In *American Control Conference*, 2003.

[6] W. Dong and Y. Guo. *Formation Control of Nonholonomic Mobile Robots Using Graph Theoretical Methods*, chapter Cooperative Systems, pages 369–386. 2007.

[7] J. Bom, B. Thuilot, F. Marmoiton, and P. Martinet. Nonlinear control for urban vehicles platooning, relying upon a unique kinematic GPS. In *$22^{n.d}$ Intern. Conf. on Robotics and Automation (ICRA'05)*, Barcelona (Spain), April 2005.

[8] H.H. Gonzales-Banos, D. Hsu, and J.C. Latombe. Motion planning: Recent developments. In *Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications*, 2006.

[9] S.R. Lindemann and S.M. LaValle. Current issues in sampling-based motion planning. In *Robotics Research Intern. Symposium*, 2005.

[10] J.T. Schwartz and M. Sharir. A survey of motion planning and related geometric algorithms. *Artificial Intelligence*, 1988.

[11] K. Komoriya and K. Tanie. Trajectory design and control of a wheel-type mobile robot using B-Spline curve. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 1989.

[12] Vazquez G.B., Sossa A. H., and Diaz deLeon S.J. Auto guided vehicle control using expanded time B-Splines. In *Int. Conf. on Systems, Man, and Cybernetics*, 1994.

[13] H. Eren, Fung C.C., and J. Evans. Implementation of the spline method for mobile robot path control. In *Instrumentation and Measurement Technology Conference*, 1999.

[14] M. Khatib, H. Jaouni, R. Chatila, and J.P. Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *International Conference on Robotics and Automation*, 1997.

[15] J.H. Hwang, R.C. Arkin, and Kwoon D.S. Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control. In *Intelligent Robotics and Systems*, 2003.

[16] A. Piazzi, C.G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi. Quintic G^2-splines for the iterative steering of vision-based autonomous vehicles. In *IEEE Transactions on Intelligent Transportation Systems*, 2002.

[17] Z. Qu, J. Wang, and C.E. Plaisted. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, 20(6):978–993, 2004.

[18] C. Samson. Control of chained systems: application to path following and time-varying point stabilization of mobile robots. *IEEE Trans. on Automatic Control*, 40(1):64–77, January 1995.

[19] B. Thuilot, J. Bom, F. Marmoiton, and P. Martinet. Accurate automatic guidance of an urban electric vehicle relying on a kinematic GPS sensor. In *5^{th} IFAC Symposium on Intelligent Autonomous Vehicles (IAV'04)*, Lisboa (Portugal), July 2004.