Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge

A. Broggi *, P. Medici, P. Zani, A. Coati, M. Panciroli
VisLab – Dipartimento di Ingegneria dell’Informazione, Università Degli Studi di Parma, Via G.P. Usberti, 181, I-43124 Parma, Italy

1. Introduction

The VisLab Intercontinental Autonomous Challenge (VIAC)\(^1\) has been a unique chance to tackle the problem of designing a local path planner capable of dealing with multiple sources of global planning data, such as

- digital maps;
- waypoints provided on a radio link by a leader vehicle;
- waypoints generated by a leader follower applications.

While also exploiting information on the immediate vehicle surroundings, including

- obstacles detected by LIDARs;
- obstacles detected by the stereo vision systems;
- lane markings;
- ground roughness and ditches;
- position and trajectory of the preceding vehicle, if any.

Local path planning still represents one of the major areas of research in robots motion planning, and is usually carried out by using one of the following approaches:

- Sliding Mode Path Following (SMPF), where the steering control law is based on the error between the current position/orientation and a given feasible trajectory;
- Traversability-Anchored Dynamic Path Following (TADPF), where starting from the current position all feasible trajectories are evaluated and the safest one is determined.

Both strategies can nevertheless be combined in real-time in order to boost their strengths, as suggested by Maček, Philippson, and Siegwart (2009).

During the 2005 DARPA Grand Challenge the TerraMax\(^2\) vehicle was able to negotiate 220 miles of offroad driving with no human intervention using a variant of RTT\(^3\) (Braid, Broggi, & Schmiedel, 2006), a common approach to select the most promising trajectories: up to 2000 different trajectories were evaluated for each planning cycle to determine the best path. TerraMax (Chen et al., 2008) in...
the DARPA Urban Challenge implemented different state-of-the-art trajectory generators that were triggered by a high level behavior supervisor, depending on both the tasks that had to be executed and environmental conditions.

Stanley, the Stanford Racing Team’s entry in the DARPA Grand Challenge (Hoffmann, Tomlin, Montemerlo, & Thrun, 2007), presented a new approach in tracking the generated trajectory based on a lateral error estimation and considering the front wheels orientation. The trajectory was required to be smooth and match feasible curvatures (Thrun et al., 2006).

During the DARPA Urban Challenge Junior (Dolgov, Thrun, Montemerlo, & Diebel, 2010) implemented a mid-term path planner which generated trajectories on a low resolution grid using a hybrid A* algorithm, considering both obstacles and vehicle constraints. As a second step, this optimal trajectory was smoothed using a Conjugate Gradient technique on a higher resolution Voronoi field. An important issue discussed in that paper is the trade-off between keeping obstacles at a safe distance and the cost of extending the route. Finally the smooth trajectory that was generated was followed using a SMPF technique.

Ben Franklin Racing Team’s (Bohren et al., 2008) used a similar formula based on lateral error to control the steering angle.

Caltech’s Alice (Linderoth, Soltesz, & Murray, 2008) implemented the generation of optimal trajectories using a non-linear optimization of the cost function. The initial rough trajectories were optimized at a later stage and the generated path was followed by a lateral control error tracking. If the error exceeded a given threshold, the vehicle invoked a replanning phase.

CMU’s Boss (Urmson et al., 2008, 2009) used a model-predictive trajectory generator, the same proposed by Howard, Green, Kelly, and Ferguson (2008), to produce dynamically feasible actions between the initial and the desired vehicle states using numerical linearization and inversion of the forward vehicle dynamic model.

Talos (Kuwata et al., 2009) implemented a variant of RTT: the feasibility of the proposed trajectories was evaluated on a drivability grid, where each cell stored a drivable/non-drivable flag and the cost to drive over it.

In more recent times VisLab designed and built BRAiVE (see Fig. 1), a mobile laboratory for the development and validation of control and machine vision technologies. The sensors suite comprises 10 cameras, 5 laser scanners, 1 radar, 1 GPS+IMU and 1 E-Stop (emergency stop) system.

Redundancy is intentional and enables the use of different subsets of sensors when developing Advanced Driver Assistance Systems (ADAS) applications (e.g. pedestrian detection, obstacle detection, vehicle following, collision warning, traffic sign recognition, parking slot detection, backup maneuver, collision detection, etc.).

A shown in Fig. 2, a dSpace MicroAutoBox unit acts as a gateway between actuators and BRAiVE autonomous driving system. This computer translates the CAN messages coming from the Control PCs into another stream, suitable for actuators, and stops the car when it receives the emergency signal or when it detects a fault condition.

All these robots show a clear division between the path planner and the low level steering control, having to perform global tasks which require a medium-term planning. In fact they demonstrate that the use of a global planner allows to execute complex maneuvers such as parking and U-turns.

To complete the VIAC expedition, the vehicles had to be able to run relying on a local planner only, that in the future could be extended to a global path planner. Methods to evaluate trajectories based on potential field are common in the literature, allowing a modular development of sensors. For this reason the trajectory is generated from the evaluation of a subset of possible acceptable vehicle trajectories on a cost map. An optimal trajectory is provided at a high rate in order to cope with the unstable nature of vehicle dynamics and with rapid environmental changes.

The remainder of this paper is organized as follows. In Section 2 the electric vehicles involved in expedition are presented and in Section 3 the equations underlying the control system are explained. Since proper knowledge of vehicle dynamics is essential, a calibration phase is also described in this paper. The control system is described in Section 4 and finally, the results of the tests and the whole experiment, based on a large amount of kilometers of autonomous driving, are presented in Section 5.

2. Vehicle setup

Thanks to the cooperation with Piaggio, the electric vehicles selected for VIAC are Piaggio Porter Electric Power vans (Fig. 3).

To control vehicle speed and steering, different devices have been installed:

- A servo motor (Fig. 4), provided by TopCon, is directly connected to the steering wheel column and controlled in position, speed, and torque through CAN-BUS messages. The set-point is internally followed using a PID controller.
- Speed is adjusted by changing the duty cycle of the PWM signal that controls the engine through a custom board (Fig. 5) connected to the CAN-BUS.
- The brake system is operated by an electric linear actuator (Fig. 6) which acts directly on the brake pedal and it is controlled via a CANopen interface.

As a safety measure, all actuators were designed to be easily overridden by a human driver.

The control system takes advantage of the information provided by a TopCon AGI3 unit, an inertial and geolocation device, installed on the vehicle roof. The GPS can use the Egnos/WAAS correction in
order to achieve sub-meter accuracy, while gyroscopes show a noise level less than 0.03 rad/s at working temperature. Both systems provide their information at a 20 Hz rate.

The vehicles are equipped also with four LIDARs and seven digital cameras, that provide frontal, lateral, and rear sensing.

The processing units are composed by three off-the-shelf multi-core computers, two of which are dedicated to sensing and one entirely devoted to planning and control. Communication happens on a GigaBit ethernet local network.

3. Kinematic model

Due to low vehicle speeds, a kinematic model has been chosen over the dynamic one. This section shows how it was extended to handle also a higher speed steady-state curve condition.

Since vehicles are rigid bodies, they can be modelled as oriented points moving on a plane. In a kinematic model this point is located on the rear axle. This configuration is defined by position \((x, y)\) and orientation \(\theta\) in an absolute reference frame. The motion model (shown in Fig. 8) is described by the kinematic bicycle differential equations:

\[
\begin{align*}
\dot{x} &= \cos \theta \\
\dot{y} &= \sin \theta \\
\dot{\theta} &= \kappa \\
\kappa &= \sigma
\end{align*}
\]

where \(\kappa\) is the curvature and \(\sigma\) the sharpness of the followed path. To get more compact formulas, \(f\) indicates the derivative with respect to the curvilinear abscissa. This notation can be easily converted into the time derivative

\[
\frac{d}{dt} x = v \frac{dx}{ds} = \nu \kappa
\]

when the longitudinal velocity \(v\) of the car is known.

Typically steering control is faster than the variation of longitudinal velocity, and therefore the speed can be considered as...
which produces a side slip angle. To balance the effect of the centrifugal force cannot be neglected. To balance the inertial properties of the vehicle are not involved. A detailed explanation of vehicle dynamics can be found in Rajamani, 2006.

To have realistic motion simulations, the maximum steering angle and the steering wheel turning speed are subject to the following constraints:

\[ |\delta| < \delta_{\text{max}}, \quad \left| \sigma \right| < \sigma_{\text{max}} \]  

where \( \delta_{\text{max}} \) and \( \sigma_{\text{max}} \) are experimentally measured.

The relation between the steering angle and the curvature path is given by the Ackermann kinematic model:

\[ \kappa = \frac{\tan \delta}{L} \]  

where \( \delta \) is the front wheel angle (i.e. the angle between the wheel and the car longitudinal axis) and \( L \) is the distance, in meters, between the front and rear axes.

It is interesting to underline that, in the Ackermann steering geometry, the two wheels have slightly different steering angles:

\[ \tan \delta_0 = \frac{L}{R + \frac{q}{2}} \quad \tan \delta_1 = \frac{L}{R - \frac{q}{2}} \]  

For this reason the relationship between the steering wheel angle \( \phi \) and the actual steering angle \( \delta \) could also be non-linear.

At high speeds the instant motion direction is slightly different from the vehicle longitudinal axis. However it is possible to define the steady-state behavior when the vehicle is in a non-time-varying condition, for example when it negotiates a constant radius curve at a constant forward speed. In the analysis of steady-state handling performance, the inertial properties of the vehicle are not involved. A detailed explanation of vehicle dynamics can be found in Gillespie (1992) and Wong (2001).

When a vehicle is negotiating a turn at moderate or high speeds, the effect of the centrifugal force cannot be neglected. To balance this force, the tires must develop an appropriate cornering force, which produces a side slip angle.

Forces \( F_{yr} \) and \( F_{yr} \) acting on tires can be determined from the dynamic equilibrium of the vehicle in the lateral direction:

\[ F_{yr} = m \frac{l_y v_y^2}{2 R} \quad F_{yr} = m \frac{l_y v_y^2}{2 R} \]  

where \( m \) is the total vehicle weight, \( v_y \) the vehicle longitudinal speed, \( l_y \) and \( l_r \) the distances between the center of gravity and the front and rear axles respectively.

As shown in Fig. 9, in this condition the vehicle instantaneous rotation center is no longer fixed onto the rear axle, but depends on the slip angles. The relationship among slip angles \( \alpha_f \) and \( \alpha_r \), steering angle \( \delta \), and turning radius \( R \) is given by

\[ \alpha_f + \delta - \alpha_r \simeq \frac{L}{R} \]  

For small slip angles the relationship between \( \alpha_f, \alpha_r \), and the lateral force are approximately linear:

\[ \alpha_f \simeq \frac{F_{yr}}{2C_{yr}} \quad \alpha_r \simeq \frac{F_{yr}}{2C_{yr}} \]  

where \( C_{yr} \) and \( C_{yr} \) are the cornering stiffnesses of the front and rear tires respectively. Substituting Eqs. (8) and (6) in (7) it is possible to deduce the relationship between vehicle forward speed and curvature:

\[ \kappa \simeq \frac{\delta}{L + K_y v_y^2} \]  

where \( K_y \) is usually referred to as the understeer coefficient. When \( K_y > 0 \) the vehicle is said to be understeering and when \( K_y < 0 \) it is said to be oversteering. In general, at different speeds \( K_y \) can have different values.

It is clear that the relationship between the steering wheel angle \( \phi \) and the performed curvature \( \kappa \) is strictly not-linear and depends also on the forward vehicle speed \( v_y \). To further generalize this relationship, Eq. (9) could be expressed as a Taylor power series like

\[ \kappa = \frac{p_1 \phi + p_2 \phi^3 + \cdots}{L + q_1 v_y + q_2 v_y^2 + \cdots} \]  

To provide a reliable control, it is important to recover the correct relationship between the steering wheel angle, the input controlled by the system, and the actual vehicle curvature.

### 3.1. Model parameters recovery

A convenient way of determining the model parameters is to observe the vehicle reactions, measuring the radius of curvature produced by a particular steering wheel angle, under the assumption that the vehicle performs a constant radius trajectory when both the steering wheel angle and the forward speed are kept constant.

Various techniques can be used to measure the instantaneous vehicle trajectory curvature (Rajamani, 2006), depending on the available sensors. Using a gyroscope and a speed sensor the instantaneous curvature is
4. Vehicle Control Design

In order to provide reliable steering, throttle, and brake control, the pathplanner considers the different inputs obtained by camera and laser processing, GPS/INS, and the high level control interfaces, as shown in Fig. 11. The kinematic model is determined according to these information, and provides the speed $r$ and the curvature $k$ that are used as the set points for the low-level control sub-systems.

4.1. Steering control

Perception data processing, data transmission, and commands actuation introduce a delay between data acquisition and planning referred to $t_0$ and $t_1$ in the following, respectively. This means that the coordinates of the perceived objects do not refer to the current time $t_0$ but are relative to time $t_0$. To cope with this problem, the points whose coordinates have been computed at time $t_0$ are retranslated, according to vehicle dynamics and tracking information, in order to predict their position at time $t_1$.

Since GPS waypoints information, unlike perception data, are natively represented in world coordinates instead of vehicle coordinates, no transformation needs to be performed on them.

The choice of working in vehicle coordinates has been dictated by the need of having world perception when GPS data is not available. Moreover vehicle coordinates are more accurate than GPS ones for small movements, although they are subject to drift.

Using the information provided by sensors, a cost map is generated in order to determine the traversable areas. A cost map allows to handle different inputs in a unified way and, at the same time, to implement complex behaviors (Gerdes & Rossetter, 2001). Fig. 12 shows the primitives used to generate the map according to the type of input: waypoints, lanes, obstacles, and ditches (negative obstacles on the road side). Each input has a different weight according to its importance: obstacles and ditches are dangerous, so their weight is greater than that of lanes or waypoints. Determining the actual value of those weights is a classical problem of autonomous navigation, and is a trade off between collision mitigation and obstacle avoidance. An additional constraint on primitives is to force the vehicle to remain in the trajectory neighborhood: if the vehicle moves too far from it, then the control system will put the vehicle in stop mode, and a manual intervention is needed to recover from this condition. Despite the reduced accuracy caused by input information quantization, cost maps provide a scalable and computationally efficient solution.

Given the vehicle braking performance (Fig. 13) a perception depth of 30 m is enough to handle a 50 km/h speeds. An occupancy

\[ k = \frac{r}{v_o} \]  \hspace{1cm} (11)

where $r$ is the yaw-rate provided by the on-board gyroscope. Using differential wheels speeds it can also be computed as

\[ k = \frac{2}{B} \frac{v_o - v_i}{v_o + v_i} \]  \hspace{1cm} (12)

where $v_o$ and $v_i$ are the wheels speeds on the same axis. Finally using an accelerometer and a speed sensor $k$ is given by

\[ k = \frac{a_y}{v_o^2} \]  \hspace{1cm} (13)

where $a_y$ is the lateral acceleration.

Eq. (10) can be easily rearranged using the measured curvature $k$ in order to provide a linear equation with unknown $(p_1, \ldots, p_m, q_1, \ldots, q_m)$ and known $(\phi, k, v_o)$ terms to a linear solver in order to estimate the model parameters.

Fig. 10 shows a comparison between the curvature reported by a gyroscope with a speed sensor and the one predicted using the calibrated steering wheel. The average difference between the two curves is 0.003 rad/m.

A model parameters recovery technique for the dynamic vehicle model can be found in Macek, Thoma, Glatzel, and Siegwart (2007) and Lenain, Thuilot, Cariou, and Martinet (2010).

3.2. Steer offset estimation

The electrical power steering wheel lacks an encoder that stores the zero position before switching off. For this reason driving cannot start before the execution of a bootstrap phase during which the vehicle tries to compensate for the yaw-rate bias and the steering wheel offset by minimizing the following error

\[ \phi - \phi_0 \approx k \frac{L}{p_1} \]  \hspace{1cm} (14)

where $k$ is the instantaneous curvature measured by sensors as described in Section 3.1.

Once the bootstrap phase is complete, Eq. (14) is used to estimate the discrepancy between the predicted and observed models in real-time. Using this information, local dynamic changes introduced by lateral wind or road bank can be corrected.

Fig. 10. Curvature. The red line represents the curvature measured using inertial information in Eq. (11), and the green line represents curvature predicted by steering wheel angle using Eq. (10). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 11. Flowchart of the planning and control layer.
A grid of 280 × 280 cells representing a 28 × 28 m area is used. Each cell stores a floating point number encoding the road conditions of the area surrounding the vehicle: low values represent safe regions, whereas high values are used to locate dangerous ones. A 10 cm cell size has proven suitable to fit our needs for a reliable and accurate navigation.

### 4.2. Trajectory planner

When the curvature is constant the vehicle follows circular trajectories, but while the steering wheel is turning, a non-negligible curvature variation is introduced and the vehicle’s motion can be approximated with a clothoid path.

To perform steering control, all feasible trajectories are evaluated using cost maps in order to select the best one to follow. The implemented planner reschedules the computation of the best trajectory at a given constant frequency (scheduling time has been chosen to be 100 ms in order to receive new fresh data from all sensors).

In the planning phase, all possible curvatures $j_1$ are considered, where $j_0$ is the currently performed curvature and $j_l$ is the maximum curvature variation that the vehicle is able to handle at the current moment. The value of $j_0$ is computed from the steering wheel angle and the dynamic offset as described in Eq. (14).

For each evaluated path the maximum curvature depends both on vehicle limits, as the maximum steering angle $\kappa_{\text{max}}$, and comfort constraints, as the lateral acceleration $|\kappa| < a_{\text{max}}/v^2$, defined to prevent fishtailing.

For each curvature $k_i$, the corresponding trajectory is estimated in order to evaluate the best path that defines, at time $t_1$, the new steering set-point. Fig. 14 shows the curvature variation with respect to time:

![Curvature Variation](image-url)
from $t_1$ to $t_{1a}$ the vehicle does not react to the new set-point and it keeps the curvilinear trend $k_0$;

- from $t_{1a}$ to $t_{1b}$ the curvature changes linearly from $k_0$ to $k_1$ following a clothoid;

- between $t_{1b}$ and $t_2$ the vehicle follows the curvilinear trend $k_1$.

In the same way at time $t_2$ the vehicle will execute a new replanning. For each evaluated $k_1$, new values of $k_2$ in a neighborhood of $k_1$ are considered and the trajectory for the next 8 m is estimated. In such a way the lookahead distance is partially speed-dependent and ranges from 8 to 14 m.

**Fig. 14.** Curvature variation function of time. Two steps are performed to generate an acceptable trajectory.

**Fig. 15.** Evaluation of vehicle trajectories: green, yellow, and red curves represent evaluated trajectories; the blue trajectory represents the best one. In the potential map dark areas represent points at low cost, while the light areas are high cost points generated by obstacles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
Those two trajectories have proven to be enough to plan in real-time over a 30 m long area.

Examples of trajectories generated during a curve are shown in Fig. 15 together with the lowest cost path associated with the chosen route.

For each path, the integral of the cost map is computed and used to find the set-point $k_1$ associated to the trajectory with the minimum integral.

Vehicle size must be considered as well when evaluating each trajectory cost. As shown in Fig. 16, for performance reasons instead of the whole area $\Omega$, only six distinctive points are considered in order to estimate, for each simulation step, the traversability cost on the area covered by the vehicle.

Two different steps are used to sample $k_1$ and $k_2$ respectively inside the ranges described above. A fixed sampling technique has the effect of evaluating more trajectories at low speeds, when in fact there is need of precision, than when moving at high speed, when there is more need for reactivity. The sampling step for $k_1$ has been set to 0.0015 rad/m, and it is further increased at the second simulation step to 0.01 rad/m. Overall 917 configurations are computed on average, with peaks of about 16,000.

An exhaustive analysis of all trajectories is then performed: to reduce the amount of computation caused by the huge number of trajectories, the sampling method described in Howard and Kelly (2007) can be applied, as it provides a fast numerical technique to optimize the choice of trajectories focusing only on the interesting ones.

For any evaluated path a weight function is used to increase or decrease its traversability cost in order to improve the navigation comfort. Moreover to avoid unnecessary direction changes, if the...
new curvature $\kappa_j$ is similar (95–105%) to the previous one, the previous trajectory is kept. Curvatures $\kappa_1 \approx 0$ are also weighed positively and $|\kappa_1| \approx \kappa_{\text{max}}$ is weighted negatively.

The set-point is converted into a steering wheel angle $\phi$ inverting the vehicle dynamic model represented by Eq. (10), and finally transmitted to the steering control.

Since the trajectory is constantly monitored, the planner adjusts the controller signal so that the vehicle can achieve the best path even in the presence of noise, moving obstacles and track errors.

### 4.3. Longitudinal control

The instantaneous target speed and acceleration are computed according to different contributions:

- the speed setpoint determined by the high level control;
- road roughness, evaluated considering the INS information;
- lateral acceleration: to ensure stability while avoiding vehicle skidding, the speed bound given by the lateral acceleration can be computed as:

$$v^2 < \frac{a_{\text{max}}}{|\kappa|}$$

(15)

- a target speed $v_k$ that allows keeping a safety distance $d$ to the front vehicle in leader–follower mode. This is a function modelled as

$$v_k = f(\Delta v, v_{k-1}, d_{k-1})$$

(16)

where $\Delta v$ is the speed difference between the follower and the leader, estimated with a Kalman filter, and $d_{k-1}$ is the current distance from the front vehicle estimated by sensors.

Once the speed setpoint has been established, a low-level controller generates the corresponding accelerator and brake pedal signals. The design of this component has been challenging, given the very complex vehicle behavior, which is greatly influenced by the load, current speed and acceleration, battery status and ground slope and roughness. As it has already been discussed in Thrun et al. (2006), Kuwata et al. (2009), and Braid et al. (2006) the non-linearities are handled through the use of PID controllers, which have proven to perform adequately, especially with reduced vehicle speed.

The speed tracking error $e = v_{\text{target}} - v_{\text{cur}}$ is used to switch between throttle and brake pedal control using the hysteresis cycle displayed in Fig. 17.

Braking is performed using a linear actuator operated through a CANopen interface and connected to the pedal, as shown in Fig. 6. When the vehicle is in braking mode, the rod position is deter-

**Fig. 21.** Plot of lateral crosstrack error (a) and relative histogram (b) during the experiment of 21 min of autonomous operations in urban and rural environments with vehicle in autonomous leader vehicle following.

**Fig. 22.** Histograms of lateral crosstrack errors in autonomous leader vehicle following mode generated by the analysis of 8244 km collected during VIAC.
mined using a PID controller with parameters that depend on \( v_{\text{curr}} \). To avoid windup, the integral term is only computed over a time window of 2 s.

The electric engine throttle value and direction are set using the custom board shown in Fig. 5, which is connected to the control PC through a CAN interface.

When determining the engine throttle value, a PID controller is used to set the amount by which the engine rotation speed should be changed. This ensures that once a stable speed is reached the setpoint remains unchanged. Moreover, a deadband of ±0.278 m/s has been set around the target speed value to further reduce the number of transitions performed.

Fig. 18 shows an example of the vehicle speed control results; during the first 40 s the setpoint is oscillating too fast to be reached, while in the following 50 s the imposed speed is compatible with the vehicle dynamics, and it is followed closely.

### 5. Conclusions

This paper presented the control system developed for VIAC and used in other autonomous driving experiments. The most important feature of the controller consists in limiting the number of evaluated trajectories, generating only candidates that the underlying control system can handle. The use of cost maps allows fusion of information provided by several sensors in a very simple and effective way providing a fast assessment of the trajectories reliability. Therefore the planner is capable of generating and following a trajectory in real-time.

During the development several tests were performed, involving a number of kilometers of autonomous driving in different weather and environmental conditions. The experimental results presented in this paper were obtained by testing the vehicles inside Parma University Campus (Fig. 19), a 2 km long loop repeated several times in autonomous driving, and in the surroundings of Parma both in urban and rural scenarios.

In Fig. 20a and b the results of Waypoint following autonomous tests (six laps performed inside the campus) are presented. The mean crosstrack error is 0.13 m, and its standard deviation is 0.15 m. The average speed on this test was 26 km/h, with a maximum of 46 km/h.

Autonomous driving performance in leader-follower mode in urban and rural areas are shown in Fig. 21a and b. The mean crosstrack error is 0.17 m, and its standard deviation is 0.18 m. The average speed on this test was 28 km/h, with a maximum of 50 km/h.

A significant increase in the lateral crosstrack error is indeed caused by the presence of obstacles along the road. The potential field generated by obstacles influences the trajectory performed by the vehicle, moving it away from the waypoint trajectory.

The controller has been validated experimentally on the vehicles during the VIAC experiment and demonstrated good capabilities to support the cross-Asia autonomous trip. Since no maps were available to cover the whole trip, the global pathplanner was never used and vehicles drove autonomously in lane-keeping, waypoints following, or leader-follower mode.

The collected data refers to the 61 effective days of autonomous driving, driving which a total amount of 214 h were traveled, divided into 191 different runs. Each run could be terminated either by technical reasons, for example due to exhausted batteries, or by logistical requirements, such as customs or need for rest. The maximum distance traveled in autonomous mode in a single run was 96.7 km.

A rough statistical pathplanner error estimation using the route performed by the leader vehicle as ground-truth was also estimated. The histogram with the distribution of the differences between the paths driven by the leader and the follower is shown in Fig. 22. This histogram reports the error on the whole 8244 km covered in autonomous mode at an average speed of 38.4 km/h and a maximum speed of 70.9 km/h. As previously mentioned the difference between the two paths is determined by the fact that the follower vehicle uses the leader trajectory as a hint only, and merges it with the sensor data to generate optimal trajectories.

VisLab has gone (autonomously) a long way towards safe, comfortable, and energy saving intelligent vehicles control, but lots of research still needs to be performed to completely achieve this goal, and the data recorded during VIAC will be of paramount importance to improve our ideas further.

### Acknowledgments

The work described in this paper has been developed in the framework of the Open intelligent systems for Future Autonomous Vehicles (OFAV) Project funded by the European Research Council (ERC) within an Advanced Investigators Grant.

### References


Alberto Broggi received the Dr. Ing. (Master) degree in Electronic Engineering and the Ph.D. degree in Information Technology both from the Università di Parma, Italy, in 1990 and 1994, respectively. He is now Full Professor at the Università di Parma. He is also the Director of the Artificial Vision and Intelligent Systems Lab, VisLab, and author of more than 150 publications on international scientific journals, book chapters, refereed conference proceedings. He served as Editor-in-Chief of the IEEE
Paolo Medici, born in Reggio Emilia, Italy, holds a Ph.D. in Information Technology and a MSc in Electronic Engineering, both from the Università di Parma, Italy. After graduating in 2004 he began working as researcher in the Artificial Vision and Intelligent System Laboratory (Vislab) with the Dipartimento di Ingegneria dell’Informazione, Università di Parma. His researches embrace many fields of computer vision, sensors calibration techniques, ensemble learning and vehicle control for the development of advanced driver assistance systems and future autonomous vehicles.

Paolo Zani received the MSc degree in Computer Engineering from the Università di Parma, Parma, Italy, in 2005. In 2009, he received the Ph.D. degree in Information Technology from the Università di Parma. His research activity is focused on efficient computer vision algorithms for intelligent and autonomous vehicles. He participated to the 2005 DARPA and 2007 DARPA Challenges, and to the VisLab Intercontinental Autonomous Challenge as X-by-wire and lane detection systems developer.

Alessandro Coati obtained his BSc and MSc degrees in Information Engineering from Università degli Studi di Parma, Italy, in 2005 and 2009, respectively. From 2009, he is working as a researcher at the VisLab, the artificial vision laboratory of Parma University, involved in several projects about AGV and obstacle detection for industrial environment. Since 2011, he is a Ph.D. student in “Information Technologies” from the Department of Information Engineering of the Università di Parma.

Matteo Pancirolli was born in Castelnovo ne’ Monti, Reggio Emilia, Italy, on December 29, 1984. He received – “summa cum laude” – the MSc degree in Information Engineering, with a thesis on “Progettazione e sviluppo di un sistema di percezione tramite fusione sensoriale per ACC” (“Design and development of a perception system based on sensor fusion for ACC”) from the Università di Parma, Parma, Italy, in July, 2008. Since January 2009, he is a Ph.D. student in “Information Technologies” from the Department of Information Engineering (DII) of the Università di Parma. In 2010 he has been in charge of sensor calibration task during VIAC project. His research activity field is on autonomous vehicle and advanced driver assistance systems and it is focused on sensor calibration and 3d vision applications.