International Journal of Intelligent Control and Systems
Vol. 3, No. 4 (1999) 409–441
© World Scientific Publishing Company

THE ARGO AUTONOMOUS VEHICLE'S VISION AND CONTROL SYSTEMS*

ALBERTO BROGGI
Dip. di Informatica e Sistemistica, Università di Pavia, Via Ferrata 1 Pavia, I-27100 Pavia, Italy

MASSIMO BERTOZZI, ALESSANDRA FASCIOLI
CORRADO GUARINO LO BIANCO, AURELIO PIAZZI
Dip. di Ingegneria dell'Informazione, Università di Parma, Parco Area delle Scienze 181A Parma, I-43100 Parma, Italy

Received (November 1999)
Revised (May 2000)

This paper presents the current status of the ARGO Project, whose main target is the development of an active safety system and an automatic pilot for a standard road vehicle. First the ARGO project is briefly described along with its main objectives, then the prototype vehicle and its functionalities are presented. The perception of the environment is performed through the processing of images acquired from the vehicle; details about the detection of lane markings, generic obstacles and leading vehicles are given. The paper describes the current implementation of the control system, based on a gain scheduled controller, which allows the vehicle to follow the road and/or other vehicles, while future control strategies (flatness approach) are presented with simulations results. The paper ends with a description of the *MilleMiglia in Automatico* tour, a journey through Italy performed in automatic driving, together with some concluding remarks.

Keywords: Vision-based autonomous vehicle, Image processing, Obstacle detection, Lane detection, Vehicle detection, Visual servoing, Automatic steering, Flatness approach, Trajectory planning.

1. Introduction

Automatic Vehicle Driving is a generic term used to address a technique aimed at automating –entirely or in part– one or more driving tasks. The automation of these tasks carries a large number of benefits, such as: a higher exploitation of the road network, lower fuel and energy consumption, and –of course– improved safety conditions compared to the current scenario.

The tasks that automatically driven vehicles are able to perform include the possibility to follow the road and keep within the right lane, maintaining a safe

*This research has been partially supported by the Italian MURST and CNR under the frame of Progetto Madess 2 and under the contract n. 99.00619.CT12.

distance between vehicles, regulating the vehicle's speed according to traffic conditions and road characteristics, moving across lanes in order to overtake vehicles and avoid obstacles, helping to find the correct and shortest route to a destination, and the movement and parking within urban environments.

Two main cooperative solutions are possible to achieve automatic driving functionalities: they require to act on infrastructures or vehicles. Both scenarios have their own pros and cons, depending on the specific application[1]; the research documented in this paper is focused exclusively on vehicles.

The following section presents the ARGO project and the prototype vehicle developed within this framework. The vehicle has visual and control capabilities: section 3 describes how the ARGO prototype vehicle can perceive the surrounding environment and the basics of lane and vehicle detection functionalities, while section 4 describes the current control subsystem and a new method based on a flatness approach. Section 5 gives an overview of the extensive test, called *MilleMiglia in Automatico*, that took place in 1998 when the vehicle drove itself for more than 2000 km in automatic mode on Italian highways, and section 6 presents some concluding remarks and possible future research directions.

2. The ARGO Project

The main target of the ARGO Project[1] is the development of an active safety system which can also act as an automatic pilot for a standard road vehicle.

In order to achieve an autonomous driving system which fits into the existing road network with no need for specific infrastructures, the capability of perceiving the environment is essential for the intelligent vehicle. Although very efficient in some fields of application, active sensors –besides polluting the environment– feature some specific problems in automotive applications due to inter-vehicle interference amongst the same type of sensors, and due to the wide variation in reflection ratios caused by many different reasons, such as obstacles' shape or material. Moreover, the maximum signal level must comply with safety rules and must be lower than a safety threshold. For this reason in the implementation of the ARGO vehicle only the use of passive sensors, such as cameras, has been considered.

A second design choice was to keep the system costs low. These costs include both production costs (which must be minimized to allow a widespread use of these devices) and operative costs, which must not exceed a certain threshold in order not to interfere with the vehicle performance. Therefore low cost devices have been preferred, both for the image acquisition and the processing: the prototype installed on ARGO is based on cheap cameras and a commercial PC.

2.1. *The ARGO Vehicle Prototype*

ARGO, shown in figure 1, is an experimental autonomous vehicle equipped with vision systems and an automatic steering capability.

It is able to determine its position with respect to the lane, to compute the road

Fig. 1. The ARGO prototype vehicle.

geometry, to detect generic obstacles on the path, and to localize a leading vehicle. The images acquired by a stereo rig placed inside the windscreen are analyzed in real-time by a computing system located into the boot. The results of the processing are used to drive an actuator mounted onto the steering wheel and other driving assistance devices.

The system was initially conceived as a safety enhancement unit: in particular it is able to supervise the driver behavior and issue both optic and acoustic warnings or even take control of the vehicle when dangerous situations are detected. Further developments have extended the system functionalities to automatic driving.

2.1.1. *The Data Acquisition System*

Only passive sensors (two cameras and a speedometer) are used on ARGO to sense the surrounding environment. In addition, a button-based control panel has been installed enabling the driver to modify a few driving parameters, select the system functionality, issue commands, and interact with the system.

The stereoscopic vision system installed on ARGO consists of two low cost synchronized cameras able to acquire pairs of grey level images simultaneously. The cameras lie inside the vehicle at the top corners of the windscreen, so that the longitudinal distance between the two cameras is maximum.

The vehicle is also equipped with a speedometer to detect its velocity. A Hall effect-based device has been chosen due to its simplicity and its reliability and has been interfaced to the computing system via a digital I/O board with event counting capabilities. The resolution of the measuring system is about 9 cm/s.

2.1.2. *The Processing System*

The architectural solution currently installed on the ARGO vehicle is based on a standard 450 MHz Pentium II processor. Thanks to recent advances in computer technologies, commercial systems offer nowadays sufficient computational power for this application. All the processing needed for the driving task (image feature extraction and vehicle trajectory control) is performed in real-time: 50 single field images are processed every second.

2.1.3. *The Output System*

Several output devices have been installed on ARGO (see figure 2).

Acoustical (stereo loudspeakers) and optical (led-based control panel) devices are used to issue warnings to the driver in case dangerous conditions are detected, while a color monitor is mainly used as a debugging tool.

A single mechanical device has been integrated on ARGO to provide autonomous steering capabilities. It is composed of an electric stepping motor coupled to the steering column by means of a belt. During automatic driving the output provided by the vision system is used to turn the steering wheel so to maintain the vehicle inside the lane or to follow the leading vehicle.

2.2. *Automatic Driving Functionalities*

Thanks to the control panel the driver can select the level of system intervention. The following three driving modes are integrated.

Manual Driving: the system simply monitors and logs the driver's activity.

Supervised Driving: in case of dangerous situations the system warns the driver with acoustic and optical signals.

Automatic Driving: the system fully controls of the vehicle's trajectory.

In the automatic driving operative mode two different functionalities can be selected: *Road Following* or *Platooning*.

The *Road Following* task, namely the automatic movement of the vehicle along the lane, is based on *Lane Detection* which includes the localization of the road, the determination of the relative position between vehicle and road, and the analysis of the vehicle's heading direction.

Conversely, the *Platooning* functionality, namely the automatic following of the preceding vehicle requires the localization and tracking of the target vehicle *(Vehicle Detection)*, which relies on the recognition of specific vehicle's characteristics.

3. Visual Perception of the Environment

In this section the vision algorithms implemented on the ARGO vehicle are described. Initially the extraction of the road geometry from monocular images is presented. Then, the algorithm for the recognition and localization of the preceding vehicle is discussed.

3.1. *Lane Detection*

The Lane Detection functionality is based on the removal of the perspective effect

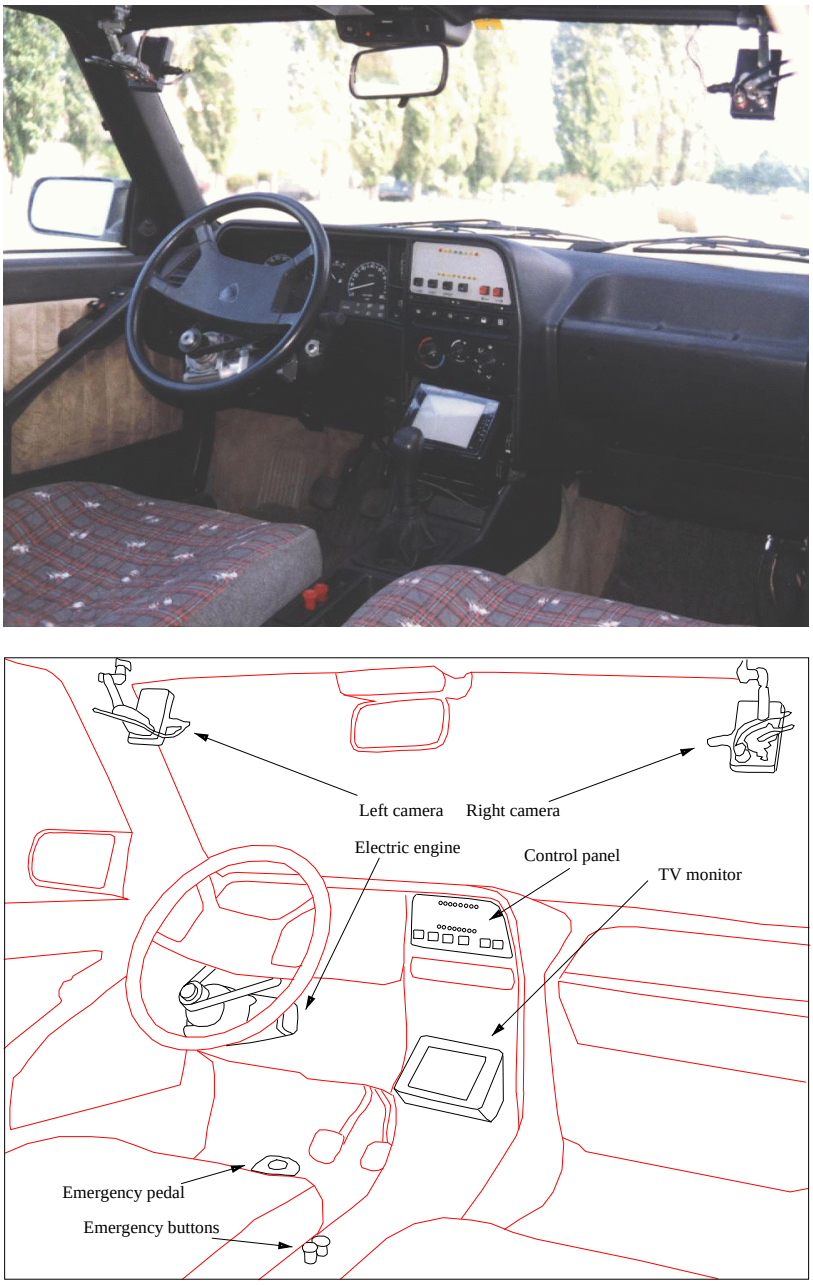


Fig. 2. Internal view of the ARGO vehicle.

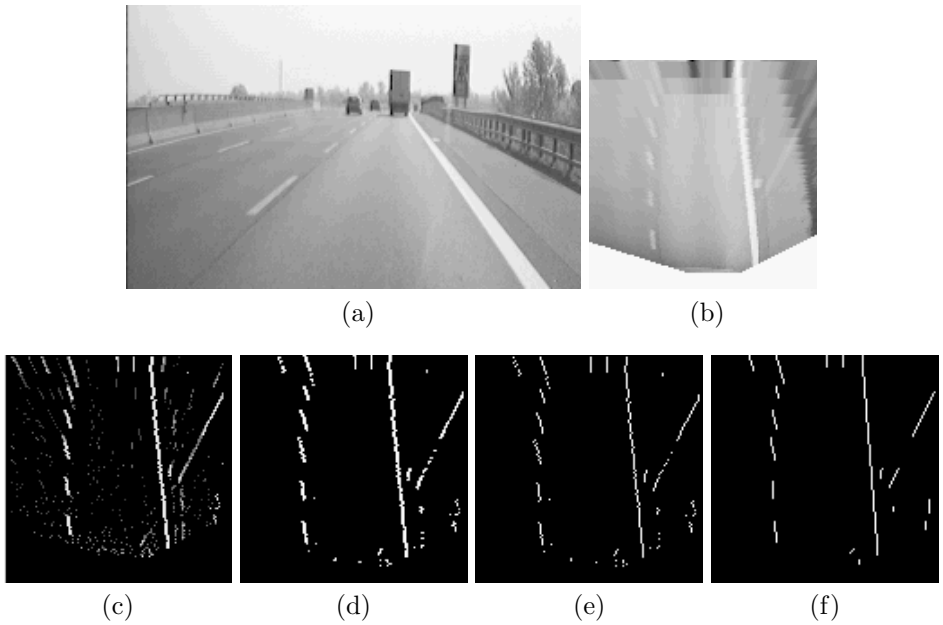


Fig. 3. The sequence of images produced by the low-level Lane Detection phase: (*a*) original; (*b*) remapped; (*c*) filtered; (*d*) binarized; (*e*) chains; (*f*) polylines.

obtained through the Inverse Perspective Mapping (IPM).[1] The IPM allows to remove the perspective effect from incoming images remapping each pixel toward a different position. It exploits a knowledge about the acquisition parameters (camera orientation, position, optics,...) and the assumption of a flat road in front of the vehicle. The result is a new 2-dimensional array of pixels (the *remapped image*) that represents a bird's eye view of the road region in front of the vehicle. Figures 3.a and 3.b show an image acquired by the ARGO vision system and the corresponding remapped image respectively.

The following steps are divided in low level and high level processing.

3.1.1. *Low Level Processing for Lane Detection*

In the remapped image a road marking resembles a semi-vertical line of constant width brighter than the dark road background. Therefore, road marking's pixels feature a higher brightness value than their horizontal left and right neighbors at a given distance. Consequently, the first phase of road markings detection is based on the determination of horizontal black-white-black transitions, while the following process is aimed at extracting information and reconstructing road geometry.

Feature Extraction

The brightness value of a generic pixel belonging to the remapped image is compared to the two horizontal left and right neighbors at a given distance. A new image,

whose values encode the presence of a road marking, is computed assigning:

1. zero to the pixels whose one or both of the two neighbors have a higher brightness value, or

2. the absolute difference between the pixel's brightness and their neighbors' ones to the pixels whose brightness is higher than the ones of the two neighbors.

The resulting filtered image is shown in figure 3.c.

Due to different light conditions (e.g. in presence of shadows), pixels representing road markings may have different brightness, yet maintaining their superiority relationship with their horizontal neighbors. Therefore, since a simple threshold seldom gives a satisfactory binarization, the image is enhanced exploiting its vertical correlation. The result is presented in figure 3.d.

Road Geometry Reconstruction

The binary image is scanned row by row in order to build chains of 8-connected non-zero pixels (see figure 3.e).

Subsequently, each chain is approximated with a *polyline* composed by one or few segments, by means of an iterative process. Initially, a single segment that joins the two extrema of the chain is considered. The horizontal distance between segment's mid point and the chain is used to determine the quality of the approximation. In case it is larger than a threshold, two segments sharing an extremum are considered for the approximation of the chain. Their common extremum is the intersection between the chain and the horizontal line that passes through the segment's mid point. The process is iterated until a satisfactory approximation has been reached. At the end of the processing all chains are approximated by polylines (see figure 3.f).

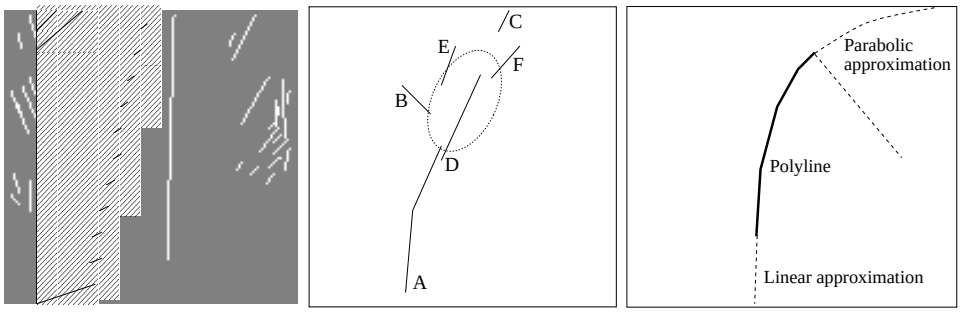
3.1.2. *High Level Processing for Lane Detection*

After the first low level stage, in which the main features are localized, and after the second stage, in which the main features are extracted, the new data structure (a list of polylines) is now processed in order to semantically group homologous features and to produce a high level description of the scene.

This process in divided into: filtering of noisy features and selection of the feature that most likely belong to the line marking; joining of different segments in order to fill gaps caused by occlusions, dashed lines, or even worn lines; selection of the best representative and reconstruction of the information that may have been lost, on the basis of continuity constraints; then the result is kept for reference in the next frames and displayed onto the original image.

Feature filtering and selection

Each polyline is compared against the result of the previous frame, since continuity constraints provide a strong and robust selection procedure. The distance between the previous result and each extremum of the considered polyline is computed: if all



(*a*) (*b*) (*c*)

Fig. 4. High level processing for Lane Detection, first steps: (*a*) selection of polylines almost matching the previous left result, (*b*) joining of similar polylines, (*c*) continuation of short polylines.

the polyline extrema lay within a stripe centered onto the previous result then the polyline is marked as useful for the following process. This stripe is shaped so that it is small at the bottom of the image (namely close to the vehicle, therefore short movements are allowed) and larger at the top of the image (far from the vehicle, where also curves that appear quickly must be tracked). This process is repeated for both left and right lane markings.

Figure 4.a shows the previous result with a heavy solid line and the search space with a gridded pattern; it refers to the left lane marking.

Polylines joining

Once the polylines have been selected, all the possibilities are checked for their joining. In order to be joined, two polylines must have similar direction; must be not too far away; their projection on the vertical axis must not overlap; the higher polyline in the image must have its starting point within an elliptical portion of the image; in case the gap is large also the direction of the connecting segment is checked for uniform behavior. Figure 4.b shows that polyline A cannot be connected to: B due to high difference of orientation; C due to high distance (does not lay within the ellipse); D due to the overlapping of their vertical projections; E since their connecting segment would have a strongly mismatching orientation. It can only be connected to F.

Selection of the best representative

All the new polylines, formed by concatenations of the original ones, are then evaluated. Starting from a maximum score, each of the following rules provides a penalty. First each polyline is segmented; in case the polyline does not cover the whole image, a penalty is given. Then, the polyline length is computed and a proportional penalty is given to short ones, as well as to polylines with extremely varying angular coefficients. Finally, the polyline with the highest score is selected as the best representative of the line marking.

Reconstruction of lost information

The polyline that has been selected at the previous step may not be long enough to cover the whole image; therefore a further step is necessary to extend the polyline. In order to take into account road curves, a parabolic model has been selected to be used in the prolongation of the polyline in the area far from the vehicle. In the nearby area, a linear approximation suffices. Figure 4.c shows the parabolic and linear prolongation.

Model fitting

The two reconstructed polylines (one representing the left and one the right lane markings) are now matched against a model that encodes some more knowledge about the absolute and relative positions of both lane markings on a standard road. A model of a pair of parallel lines at a given distance (the lane width) and in a specific position is initialized at the beginning of the process; a specific learning phase allows to adapt the model to errors in camera calibration (lines may be non perfectly parallel). Furthermore, this model can be slowly changed during the processing to adapt to new road conditions (lane width and lane position), thanks to a learning process running in the background.

The model is kept for reference: the two resulting polylines are fitted to this model and the final result is obtained as follows. First the two polylines are checked for non-parallel behavior; a small deviation is allowed since it may derive from vehicle movements or deviations from the flat road assumption, that cause the calibration to be temporarily incorrect (diverging of converging lane markings).

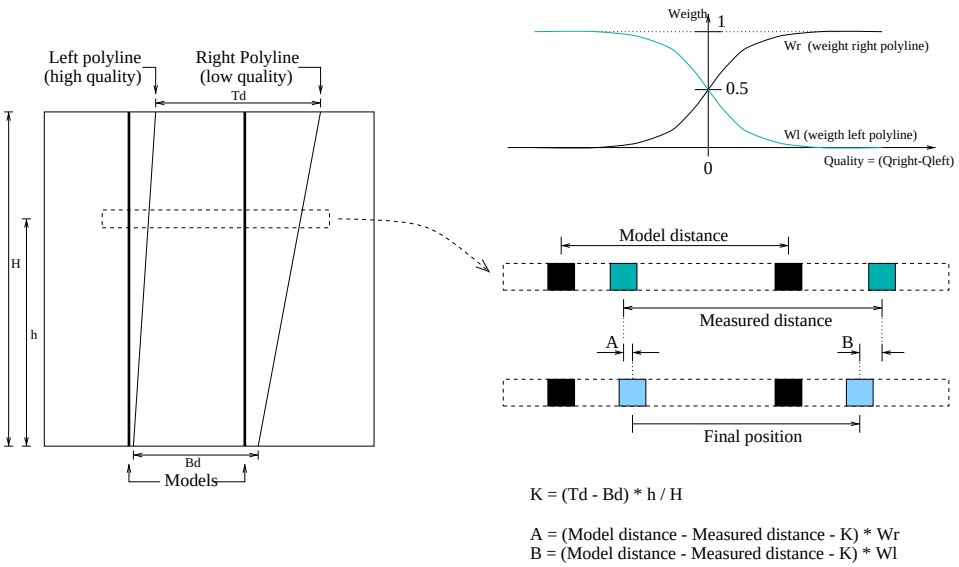


Fig. 5. Generation of the final result.

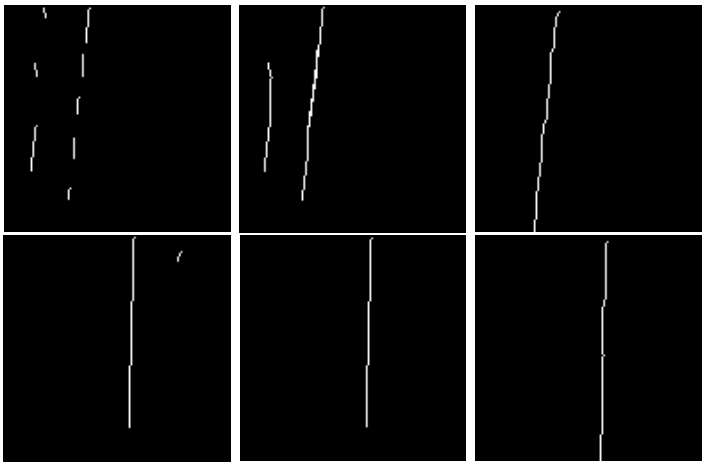
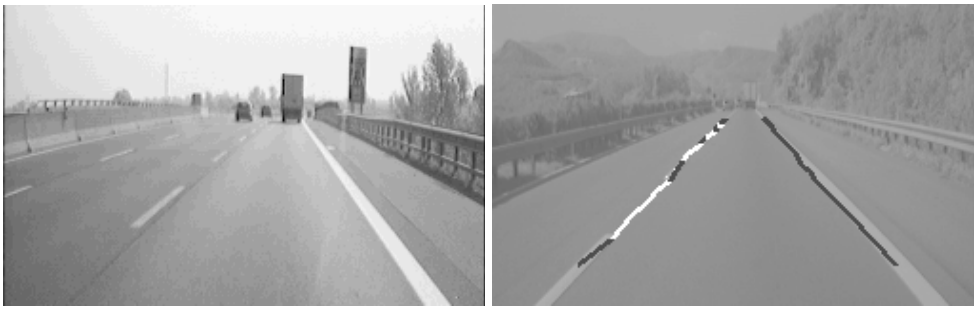


Fig. 6. Filtered polylines, joined polylines, and model fitting for the left (upper row) and right (bottom row) lane markings.

Then the quality of the two polylines, as computed in the previous steps, is matched: the final result will be attracted toward the polyline with the highest quality with a higher strength. In this way, polylines with equal or similar quality will equally contribute to the final result; on the other hand, in case one polyline has been heavily reconstructed, or is far from the original model, or is even missing, the other polyline will be used to generate the final result. The weights for the left and right polylines are computed as shown in figure 5. Then, each horizontal line of the two polylines is used to compute the final results, as shown in figure 5.

Finally figure 6 presents the resulting images referring to the example presented in figure 3. It shows the results of the selection, the joining, and the matching phase phases for the left (upper row) and for the right (bottom row) lane markings.

Figure 7 presents the final result of the process.



(a) (b)

Fig. 7. The result of lane detection: (*a*) the acquired image, (*b*) the result of lane detection shown onto a saturated copy of the original image; black markers represent effective lane markings, while white markings represent interpolations between them.

Fig. 8. Lane Detection: the result is shown with black markings superimposed onto a brighter version of the original image.

3.1.3. *Results of Lane Detection*

Figure 8 presents a few results of lane detection in different conditions ranging from ideal situations to road works, patches of non-painted roads, the entry and exit from a tunnel. Both highway and extraurban scenes are provided for comparison; the systems proves to be robust with respect to different illumination situations, missing road signs, and overtaking vehicles which occlude the visibility of the left lane marking. In case two lines are present, the system selects the continuous one.

Concerning the quantitative performance, the algorithm requires an overall average time of 4 ms for the processing of a frame.

3.2. *Vehicle Detection*

The platooning functionality is based on the detection of the distance, speed, and

heading of the preceding vehicle, which is localized and tracked using a single monocular image sequence.

The vehicle detection algorithm is based on the following considerations: a vehicle is generally symmetric, characterized by a rectangular bounding box which satisfies specific aspect ratio constraints, and placed in a particular region of the image. These features are used to identify vehicles in the image in the following way: first an area of interest is identified on the basis of road position and perspective constraints. This area is searched for possible vertical symmetries; not only gray level symmetries are considered, but vertical and horizontal edges symmetries as well, in order to increase the detection robustness. Once the symmetry position and width has been detected, a new search begins, which is aimed at the detection of the two bottom corners of a rectangular bounding box. Finally, the top horizontal limit of the vehicle is searched for, and the preceding vehicle localized.

The tracking phase is performed through the maximization of the correlation between the portion of the image contained into the bounding box of the previous frame (partially stretched and reduced to take into account small size variations due to the increment or reduction of the relative distance) and the new frame.

Symmetry detection

In order to search for symmetrical features, the analysis of gray level images is not sufficient. Figure 9 shows that strong reflections cause irregularities in vehicle symmetry, while uniform areas and background patterns present highly correlated symmetries. In order to cope with these problems, also symmetries in other domains are computed.

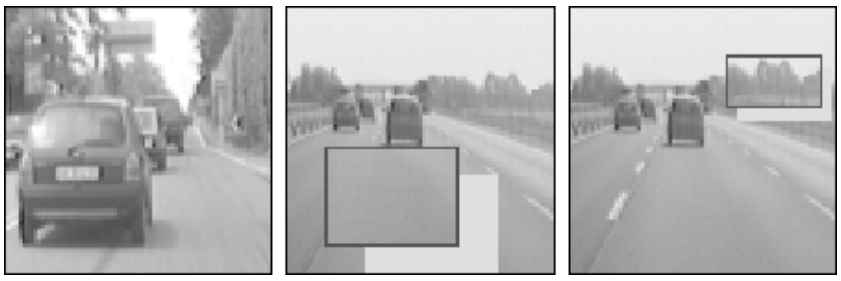


Fig. 9. Typical road scenes: in the leftmost image a strong sun reflection reduces the vehicle gray level symmetry; in the center image a uniform area can be regarded as a highly symmetrical region; the rightmost image shows background symmetrical patterns.

To get rid of reflections and uniform areas, vertical and horizontal edges are extracted and thresholded, and symmetries are computed into these domains as well. Figure 10 shows that although a strong reflection is present on the left side of the vehicle, edges are anyway visible and can be used to extract symmetries; moreover, in uniform areas no edges are extracted and therefore no symmetries are detected. Figure 11 shows two examples in which gray level symmetries alone can be successful for vehicle detection, while figure 12 shows the result of edge symmetry.

Fig. 10. Edges enforce the detection of real symmetries: strong reflections have lower effects while uniform areas are discarded since they do not present edges.

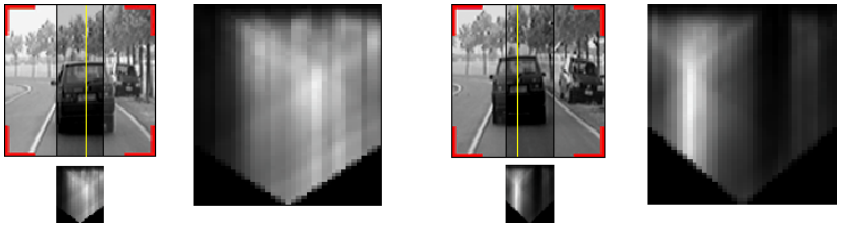


Fig. 11. Grey level symmetries: the rightmost image for each case shows a symmetry map encoding high symmetries with bright points.

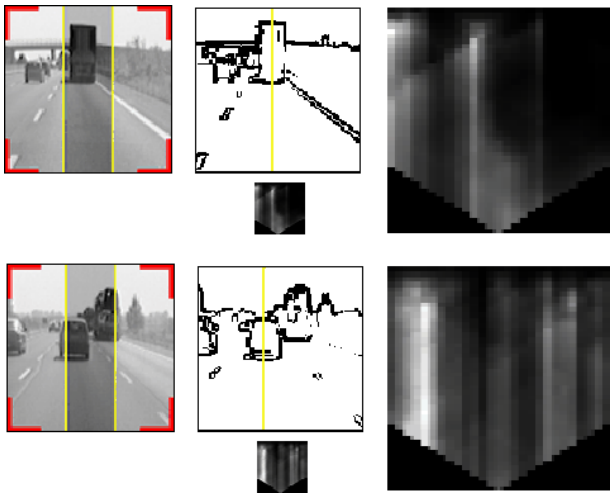


Fig. 12. Edge symmetries: the symmetries are computed on the binary images obtained after thresholding the gradient image.

For each image, the search area is shown in dark gray and the resulting vertical axis is superimposed. For each image its symmetry map is also depicted both in its original size and –on the right– zoomed for better viewing. Bright points encode the presence of high symmetries. The 2D symmetry maps are computed by varying the axis' horizontal position within the grey area (shown in the original image) and the symmetry horizontal size. The lower triangular shape is due to the limitation in scanning large horizontal windows for peripheral vertical axes.

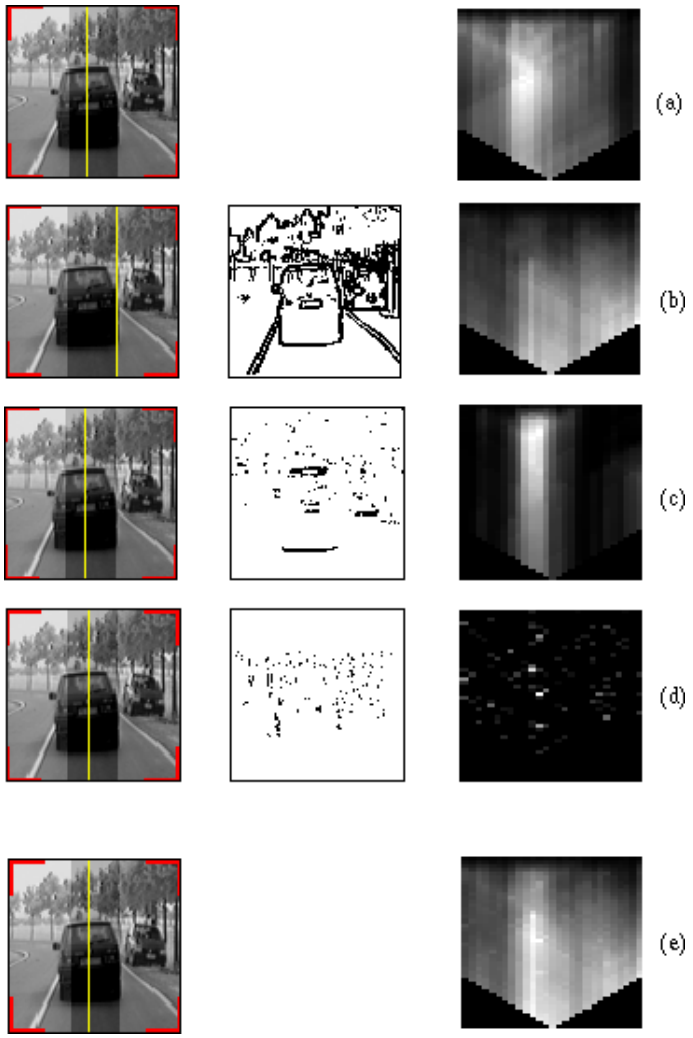


Fig. 13. Computing the resulting symmetry: (*a*) grey-level symmetry; (*b*) edge symmetry; (*c*) horizontal edges symmetry; (*d*) vertical edges symmetry; (*e*) total symmetry. For each row the resulting symmetry axis is superimposed onto the leftmost original image.

Similarly, the analysis of symmetries of horizontal and vertical edges produces other symmetry maps, which –with specific coefficients detected experimentally– can be combined with the previous ones to form a single symmetry map. Figure 13 shows all symmetry maps and the final one, that allows to detect the vehicle.

Bounding box detection

After the localization of the symmetry, the symmetrical region is checked for the presence of two corners representing the bottom of the bounding box around the vehicle. Perspective constraints as well as size constraints are used as search criteria.

Figure 14 presents the results of the lower corners detection.

Fig. 14. Detection of the lower part of the bounding box: (a) original image with symmetry axis; (b) edges; (c) localization of the two lower corners.

This process is followed by the detection of the top part of the bounding box, which is looked for in a specific region whose location is again determined by perspective and size constraints.

Backtracking

Sometimes it may happen that in correspondence to the symmetry maximum no correct bounding boxes exist. Therefore, a backtracking approach is used: the symmetry map is again scanned for the next local maximum and a new search for a bounding box is performed. Figure 15 shows a situation in which the first symmetry maximum, generated by a building, does not lead to a correct bounding box; on the other hand, the second maximum leads to the correct detection of the vehicle.

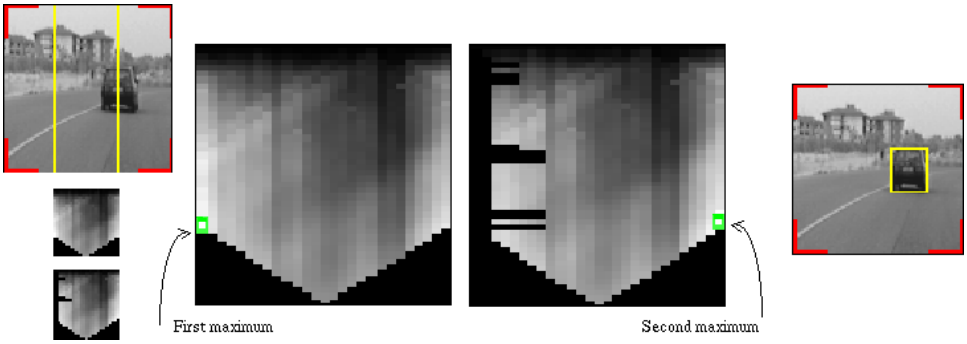


Fig. 15. A case in which the total background symmetry is higher than the vehicle symmetry. Original image; first symmetry map; second symmetry map after the backtracking process has removed the peak near the maximum; final bounding box detection.

3.2.1. *Results of Vehicle Detection*

Figure 16 shows some qualitative results of vehicle detection in different situations: the preceding vehicle is correctly detected at different distances, even when several other vehicles are present on the road.

The quantitative performance has also been assessed: the algorithm runs in 23 ms when no vehicle has been detected in the previous frame and therefore a new search is started. However, the time required for the processing decreases to 9 ms when the target vehicle is being tracked.

Fig. 16. Vehicle Detection: the images show the search area and the detected vehicle with black markings superimposed onto a brighter version of the original image.

4. Vehicle Control

In recent years, the problem of automatic steering of an autonomous vehicle has gained considerable attention from both the theoretical[2,3] and experimental side.[4,5]

Roughly speaking this problem is centered on finding a satisfactory law for the command of the steering wheel. Many works have been reported in the literature[6,7,8,9,10,11,12,13,14] and various steering control designs were proposed for systems in which the sensing is performed with nonvisual devices (for example, guiding wire, microwave radars, etc.).

On the other hand, a visual servoing paradigm was proposed by Epiau *et al.*[15] by considering a simple omnidirectional mobile robot. Neural networks were adopted and subsequently developed in the RALPH project.[16,17] A comparative survey on various vision-based control strategies for autonomous vehicles can be found in the paper of Taylor *et al.*[5]

Subsection 4.1 presents the gain scheduled proportional controller currently implemented on the ARGO vehicle. By using a feedback supervisor this control law can be adopted to perform both path following and platooning. A simple proportional control law was previously examined by Özgüner *et al.*[18] for the path following functionality solely. Subsection 4.2 exposes a differential flatness approach that will

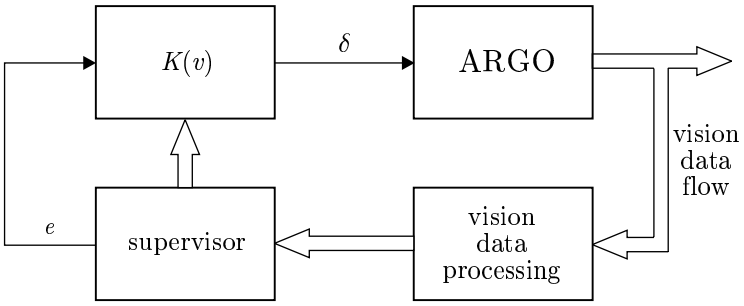


Fig. 17. Control scheme with the gain scheduled proportional controller.

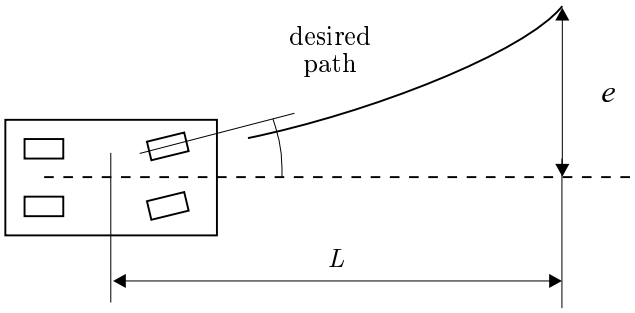


Fig. 18. Offset from the desired path, estimated by the vision system.

be the base of the future ARGO's control system. The main result (Proposition 1) characterizes the path of a vehicle lateral dynamics as G^2-path, i.e. a path with second order geometric continuity. Subsection 4.3 presents the vehicle's trajectory planning with the quintic G^2-splines and a new recursive trajectory control scheme for path following. The new overall control approach can be regarded as a generalization of the control strategy described by Tsugawa *et al.*[19] Simulation results regarding this new approach are reported in the last subsection.

4.1. *Gain Scheduled Proportional Controller*

The controller currently adopted for the ARGO vehicle was initially designed and optimized for a road following task. Minor changes have been introduced to implement also the platooning functionality.

The basic control scheme is visible in figure 17. The command steering angle δ is obtained with a variable gain proportional controller. The vision based system reconstructs the road environment and the supervisor uses the results to select the most appropriate gain for the proportional controller and estimate the error signal. Initially, the offset e existing between the vehicle heading and the desired path is computed at the look-ahead distance L (see figure 18). The estimated signal e is inherently noisy so that it cannot be directly supplied to the proportional

Table 1. Parameters for the evaluation of the look-ahead distance.

v_{min}	=	8.33 ms^{-1} (30 km/h)
v_{max}	=	22.22 ms^{-1} (80 km/h)
t_l	=	1.5 s
L_{min}	=	12.5 m
L_{max}	=	33.33 m

controller. To reduce the disturbances, e is preliminary filtered with a moving average filter. The look-ahead distance is variable and depends on the vehicle speed; more precisely, L is obtained according to the following expression

$$L(v) := \begin{cases} L_{min} & \text{if } v < v_{min} \\ v\, t_l & \text{if } v_{min} \leq v \leq v_{max} \\ L_{max} & \text{if } v > v_{max} \end{cases} \tag{4.1}$$

where $L_{min} = v_{min} t_l$ and $L_{max} = v_{max} t_l$ indicate the minimum and the maximum look-ahead distance respectively, t_l is the look-ahead time, v is the vehicle speed. The choice of L influences the behaviour of the controller. It has been demonstrated[5] that, as v increases, the damping factor of the closed loop system gets worse and can be improved, under certain limits, by increasing the look-ahead distance. For the ARGO vehicle, the supervisor uses the parameters reported in tab. 1.

To further improve the performances of the closed loop system a gain scheduling technique has been adopted for the proportional controller. Specifically K inversely depends on the velocity v according to:

$$K(v) := \begin{cases} K_{max} & \text{if } v < v^* \\ K_A / v & \text{if } v^* \leq v\ . \end{cases} \tag{4.2}$$

If the velocity becomes smaller then v^*, the proportional gain is upper bounded by K_{max} (for ARGO $v^* = 2.777$ $\text{ms}^{-1} = 10$ km/h). $K(v)$ is continuous because K_A must satisfy the equation $K_{max} = K_A / v^*$. The parameter K_A (and consequently K_{max}) has been set by means of a series of experiments on the ARGO vehicle.

The controller sampling time is imposed by the vision system (it is given by the refresh rate of the cameras) and is equal to 0.02 s (50 Hz). The average computation time that comprises both vision and control algorithm processings is equal to 0.004 s.

The control strategy adopted for platooning takes advantage of the previously defined control scheme (see figure 17). The main and crucial difference with respect to the path-following functionality is on the supervisor estimation of the offset error e. When the platooning functionality is activated, the target point is centered on the preceding vehicle so that the target look-ahead distance L' is nor constant neither the most appropriate for the current velocity (see figure 19). Obviously, using this look-ahead distance L' and the corresponding target offset error e' could degrade

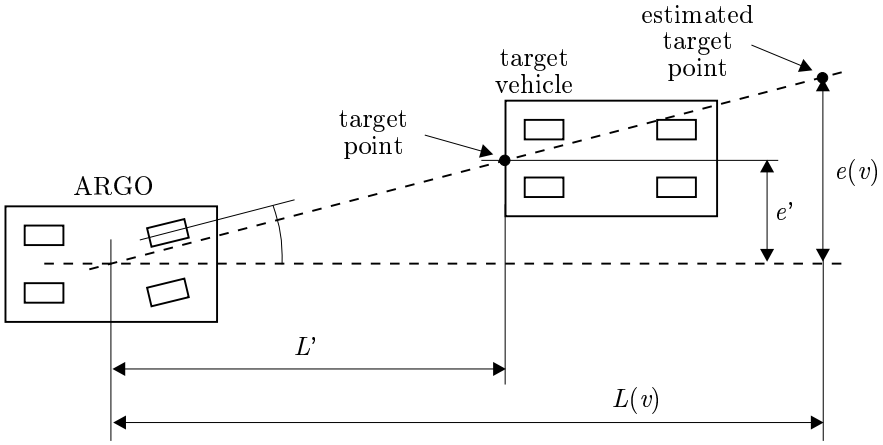


Fig. 19. Evaluation of the proper error signal for a platooning application.

the performance of the platooning functionality. The efficiency of the platooning control algorithm is recovered by scaling the tracking error e' measured at L' to an estimated offset error $e(v)$ given through a "virtual" target point placed at the appropriate look-ahead distance $L(v)$ (cf. (4.1)):

$$e(v) := \frac{L(v)}{L'}\, e'. \tag{4.3}$$

This approach has revealed to be effective for highway driving tasks.

4.2. *The Flatness Approach*

The previous gain scheduling proportional controller has been designed with a visual servoing approach regardless of a quantitative model of the vehicle lateral motion dynamics. A possible improvement in the design of the vehicle's automatic steering can start by considering the following simplified nonholonomic model of the lateral motion dynamics:

$$\begin{cases} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \frac{v}{l}\tan\delta \end{cases} \tag{4.4}$$

The state variables x, y, and θ are the planar coordinates of the rear axle midpoint and the vehicle's heading angle respectively (see figure 20). The vehicle's velocity is v, the inter-axle distance is l and δ is the front wheel steering angle.

Flatness is a differential property of a broad class of dynamic systems.[20,21] Roughly stating, a system is flat when it is possible to determine, at any given time t, system's state and input from the sole knowledge of the output and its derivatives (till a finite order) at the same instant t. As a consequence, open-loop control problems relative to flat systems can be addressed by posing an output trajectory planning and then computing the corresponding input via a dynamic

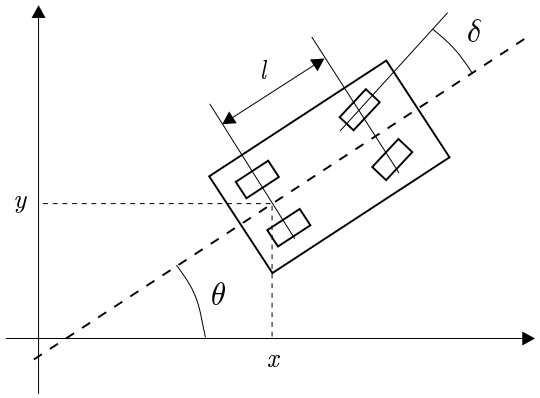


Fig. 20. Vehicle's variables of the model (4.4).

inversion that exploits the flatness property. Subsequently, the resulting system's state and output trajectories are stabilized against disturbances and modeling errors by designing a suitable feedback controller.[22]

Focusing on trailer systems, that are nonholonomic, it is possible to use flatness for deriving the motion planning.[23,24] Pursuing a similar aim regarding the nonholonomic system (4.4), we exploit its flatness by studying the differential properties of the vehicle's cartesian trajectory. Before presenting the proposition summarizing the main findings, we introduce the following terminology and definitions.

A curve on the $\{x, y\}$-plane can be described by a parameterization $\mathbf{p}(u) = [x(u)\ y(u)]^T$ with real parameter $u \in [u_0, u_1]$. The associated "path" is the image of $[u_0, u_1]$ under the vectorial function $\mathbf{p}(u)$. We say that the curve $\mathbf{p}(u)$ is regular if there exists $\dot{\mathbf{p}}(u)$ over $[u_0, u_1]$ and $\dot{\mathbf{p}}(u) \neq 0\ \forall u \in [u_0, u_1]$. A curve $\mathbf{p}(u)$ has first order geometric continuity, i.e. $\mathbf{p}(u) \in G^1$, if $\mathbf{p}(u)$ is regular and its unit tangent vector is a continuous function over $[u_0, u_1]$. In turn, a curve $\mathbf{p}(u)$ has second order geometric continuity, i.e. $\mathbf{p}(u) \in G^2$, if $\mathbf{p}(u) \in G^1$ and its curvature vector is continuous over $[u_0, u_1]$.[25] By natural extension we say that a $\{x, y\}$-path, i.e. a set of points in the $\{x, y\}$-plane, belongs to $G^i,\ i \in \{1, 2\}$ if there exists a parametric curve $\mathbf{p}(u) \in G^i$ such that its image is the $\{x, y\}$-path. The Euclidean norm of a vector $\mathbf{p}$ is denoted with $\|\mathbf{p}\|$.

Proposition 1. A path on plane $\{x, y\}$ is generated by vehicle model (4.4) via a continuous control input $\delta(t)$ if and only if the $\{x, y\}$-path is a G^2-path.

Proof.

Necessity

Consider a $\{x, y\}$-path generated by model (4.4) with a continuous $\delta(t)$. A natural parameterization over this path is given by the explicit solution $\mathbf{p}(t) = [x(t)\ y(t)]^T$ of the model (4.4) when t belongs, say, to the interval $[t_0, t_1]$. The unit tangent

vector can be expressed as

$$\frac{\dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|} = \frac{[\dot{x}(t)\ \dot{y}(t)]^T}{\sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}} = \frac{v[\cos\theta(t)\ \ \sin\theta(t)]^T}{\sqrt{v^2\cos^2\theta(t) + v^2\sin^2\theta(t)}} = [\cos\theta(t)\ \ \sin\theta(t)]^T. \quad (4.5)$$

Considering that $\delta(t)$ is continuous we have, from model (4.4) that $\theta(t)$ is continuous and then, by (4.5), $\dot{\mathbf{p}}/\|\dot{\mathbf{p}}\|$ is continuous too. With a similar reasoning we observe that $\ddot{\mathbf{p}}$ is a continuous function. Indeed, from model (4.4) we infer that

$$\begin{cases} \ddot{x} = -v\ \dot{\theta}\sin\theta = -\frac{v^2}{l}\tan\delta(t)\sin\theta(t) \\ \ddot{y} = v\ \dot{\theta}\cos\theta = \frac{v^2}{l}\tan\delta(t)\cos\theta(t) \end{cases} \quad (4.6)$$

Therefore, the curvature vector, that can be expressed as ($\times$ denotes the vector cross product)

$$\frac{(\dot{\mathbf{p}} \times \ddot{\mathbf{p}}) \times \dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|^4}, \quad (4.7)$$

is a continuous function too. In conclusion this proves that $\mathbf{p}(t)$ is a G^2-curve.

Sufficiency

Let be given, on the $\{x, y\}$-plane, a G^2-curve $\mathbf{p}(u) = [x(u)\ y(u)]^T$ with parameter $u \in [u_0, u_1]$. We want to show that there exist initial conditions and a continuous control $\delta(t)$ such that the $\{x, y\}$-path generated by the vehicle's model (4.4) exactly matches the path of $\mathbf{p}(u)$. First, introduce the arc length function

$$s(u) := \int_{u_0}^{u} \sqrt{\dot{x}(v)^2 + \dot{y}(v)^2}dv \quad (4.8)$$

and denote by $s^{-1} : [0, s(u_1)] \to [u_0, u_1]$ its inverse function. Associated with every point of $\mathbf{p}(u)$ we consider the orthonormal frame $\{\mathbf{e}_1(u), \mathbf{e}_2(u)\}$ that is oriented in the same way of axes $\{x, y\}$ and where $\mathbf{e}_1(u)$ coincides with the unit tangent vector of the curve $\mathbf{p}(u)$. As known from Frenet formulae, the curvature vector is $\kappa(u)\mathbf{e}_2(u)$ where $\kappa(u)$ is the scalar curvature with well defined sign. From (4.7) it can be deduced that

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{(\dot{x}(u)^2 + \dot{y}(u)^2)^{3/2}}. \quad (4.9)$$

Considering that $\mathbf{p}(u)$ is a G^2-curve, the function $\kappa(u)$ is continuous. For model (4.4) consider, at time t_0, the initial conditions $x(u_0)$, $y(u_0)$, $\arg(\mathbf{e}_1(u_0))$ and the continuous input function

$$\delta(t) = \arctan[l\kappa(s^{-1}(v(t - t_0)))]. \quad (4.10)$$

In the following it is proved that the explicit solution to model (4.4) is given by the time functions

$$x(s^{-1}(v(t - t_0))), \quad (4.11)$$

$$y(s^{-1}(v(t - t_0))), \quad (4.12)$$

$$\arg[\mathbf{e}_1(s^{-1}(v(t - t_0)))]. \quad (4.13)$$

From standard derivation rules we obtain

$$\begin{aligned}\frac{d}{dt}[x(s^{-1}(v(t-t_0)))] &= v\,\frac{\dot{x}(u)}{\|\dot{\mathbf{p}}(u)\|}\bigg|_{u=s^{-1}(v(t-t_0))} = \\ &= v\cos\left[\arg[\|\dot{\mathbf{p}}(u)\|^{-1}\,\dot{\mathbf{p}}(u)]\right]\Big|_{u=s^{-1}(v(t-t_0))} \end{aligned} \tag{4.14}$$

and analogously

$$\begin{aligned}\frac{d}{dt}[y(s^{-1}(v(t-t_0)))] &= v\,\frac{\dot{y}(u)}{\|\dot{\mathbf{p}}(u)\|}\bigg|_{u=s^{-1}(v(t-t_0))} = \\ &= v\sin\left[\arg[\|\dot{\mathbf{p}}(u)\|^{-1}\,\dot{\mathbf{p}}(u)]\right]\Big|_{u=s^{-1}(v(t-t_0))}. \end{aligned} \tag{4.15}$$

Considering that $\mathbf{e}_1(u) \equiv \|\dot{\mathbf{p}}(u)\|^{-1}\,\dot{\mathbf{p}}(u)$, identities (4.14) and (4.15) verify the first two equations of model (4.4). Define as $\theta(t)$ the time function appearing in (4.13). Hence, we have

$$\theta(t) = \begin{cases} \arctan\left[\dot{y}(u)/\dot{x}(u)\right]\Big|_{u=s^{-1}(v(t-t_0))} & \text{if } \dot{x}(u) \geq 0 \\ \pi + \arctan\left[\dot{y}(u)/\dot{x}(u)\right]\Big|_{u=s^{-1}(v(t-t_0))} & \text{if } \dot{x}(u) < 0 \end{cases} \tag{4.16}$$

and, by derivation, obtain

$$\begin{aligned}\dot{\theta}(t) &= \frac{\frac{d}{dt}[\dot{y}(u)]\dot{x}(u) - \dot{y}(u)\frac{d}{dt}[\dot{x}(u)]}{\dot{x}(u)^2 + \dot{y}(u)^2}\bigg|_{u=s^{-1}(v(t-t_0))} = \\ &= v\,\frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{\left(\dot{x}(u)^2 + \dot{y}(u)^2\right)^{3/2}}\bigg|_{u=s^{-1}(v(t-t_0))}. \end{aligned} \tag{4.17}$$

Note that, in the definition (4.16) of $\theta(t)$ we have assumed

$$\arctan\left[\dot{y}(u)/\dot{x}(u)\right]\Big|_{\dot{x}(u)=0} := \begin{cases} +\pi/2 & \text{if } \dot{y}(u) > 0 \\ -\pi/2 & \text{if } \dot{y}(u) < 0 \end{cases} \tag{4.18}$$

By virtue of (4.9) it is then proved that

$$\dot{\theta}(t) = v\,\kappa(s^{-1}(v(t-t_0))). \tag{4.19}$$

On the other hand relation (4.10) implies

$$\frac{v}{l}\tan\delta(t) = v\,\kappa(s^{-1}(v(t-t_0))) \tag{4.20}$$

so that (4.19) and (4.20) verify the third equation of model (4.4).

From (4.11) and (4.12) we finally note that the $\{x, y\}$-path generated by input (4.10) exactly matches the path of $\mathbf{p}(u)$. □

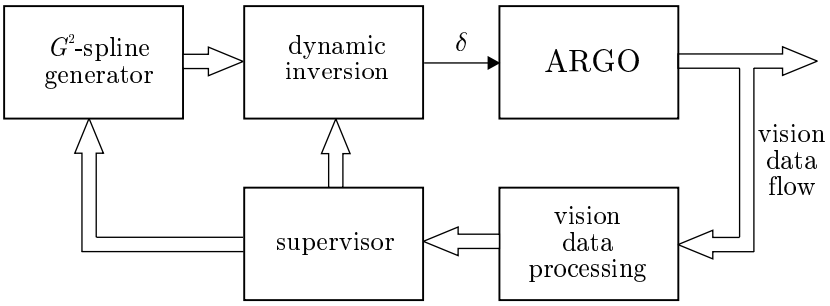


Fig. 21. The new recursive trajectory control scheme.

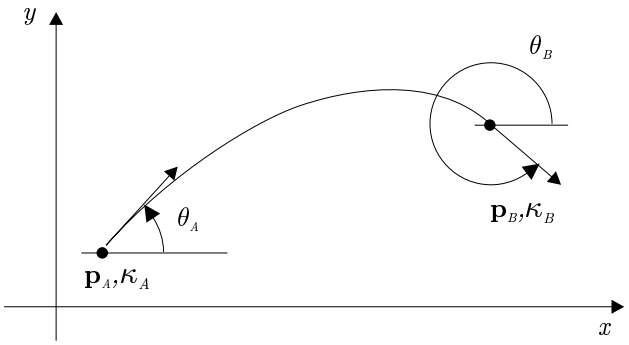


Fig. 22. The G^2-interpolating problem on the $\{x, y\}$-plane.

4.3. *Quintic G^2-splines and Recursive Trajectory Control*

The sufficiency proof of Proposition 1 gives the explicit dynamic inversion formulae (4.8)–(4.10) for the open-loop steering control of vehicle (4.4) provided that the desired path to follow is assigned as a given parametric curve. Therefore, consider the following interpolating path problem. Let be given on plane $\{x, y\}$ two distinct points $\mathbf{p}_A = [x_A \ y_A]^T$ and $\mathbf{p}_B = [x_B \ y_B]^T$ with assigned unit tangent vectors defined by θ_A and θ_B and scalar curvatures κ_A and κ_B (see figure 22). The signs of κ_A and κ_B are given according to the Frenet formulae; cf. the introduction to (4.9) in the proof of Proposition 1. The data $\mathbf{p}_A$, θ_A, and κ_A represent the vehicle's current status at a given time t_0, i.e. the coordinates x_A and y_A of the rear axle midpoint, the heading angle, and the curvature κ_A given by

$$\kappa_A = (1/l) \tan \delta(t_0) \tag{4.21}$$

where $\delta(t_0)$ is the current steering angle. The data $\mathbf{p}_B$, θ_B, and κ_B are the desired future status of the vehicle.

The parametric curve to consider is a quintic polynomial vector function $\mathbf{p}(u) = [x(u) \ y(u)]^T$, $u \in [0, 1]$ where

$$x(u) \quad := \quad x_0 + x_1 u + x_2 u^2 + x_3 u^3 + x_4 u^4 + x_5 u^5 \tag{4.22}$$

$$y(u) \quad := \quad y_0 + y_1 u + y_2 u^2 + y_3 u^3 + y_4 u^4 + y_5 u^5. \tag{4.23}$$

with interpolating conditions (cf. (4.5) and (4.9)):

$$\mathbf{p}(0) = \mathbf{p}_A\,, \qquad \mathbf{p}(1) = \mathbf{p}_B\,, \tag{4.24}$$

$$\frac{\dot{\mathbf{p}}(0)}{\|\dot{\mathbf{p}}(0)\|} = \begin{bmatrix} \cos\theta_A \\ \sin\theta_A \end{bmatrix}, \quad \frac{\dot{\mathbf{p}}(1)}{\|\dot{\mathbf{p}}(1)\|} = \begin{bmatrix} \cos\theta_B \\ \sin\theta_B \end{bmatrix}, \tag{4.25}$$

$$\kappa(0) = \kappa_A\,, \qquad \kappa(1) = \kappa_B\,. \tag{4.26}$$

Focusing on polynomial parametric curves and working with arbitrary interpolating data, it is necessary to use, at least, quintic polynomials as in (4.22), (4.23), in order to satisfy conditions (4.24)–(4.26).[26] Nevertheless, conditions (4.24)–(4.26) leave four degrees of freedom that are exploited by the following closed-form solution:

$$x_0 = x_A \tag{4.27}$$

$$x_1 = \eta_1 \cos\theta_A \tag{4.28}$$

$$x_2 = \frac{1}{2}\left(\eta_3 \cos\theta_A - \eta_1^2 \kappa_A \sin\theta_A\right) \tag{4.29}$$

$$\begin{aligned} x_3 = {} & 10(x_B - x_A) - (6\eta_1 + \frac{3}{2}\eta_3)\cos\theta_A - (4\eta_2 - \frac{1}{2}\eta_4)\cos\theta_B \\ & + \frac{3}{2}\eta_1^2 \kappa_A \sin\theta_A - \frac{1}{2}\eta_2^2 \kappa_B \sin\theta_B \end{aligned} \tag{4.30}$$

$$\begin{aligned} x_4 = {} & -15(x_B - x_A) + (8\eta_1 + \frac{3}{2}\eta_3)\cos\theta_A + (7\eta_2 - \eta_4)\cos\theta_B \\ & - \frac{3}{2}\eta_1^2 \kappa_A \sin\theta_A + \eta_2^2 \kappa_B \sin\theta_B \end{aligned} \tag{4.31}$$

$$\begin{aligned} x_5 = {} & 6(x_B - x_A) - (3\eta_1 + \frac{1}{2}\eta_3)\cos\theta_A - (3\eta_2 - \frac{1}{2}\eta_4)\cos\theta_B \\ & + \frac{1}{2}\eta_1^2 \kappa_A \sin\theta_A - \frac{1}{2}\eta_2^2 \kappa_B \sin\theta_B \end{aligned} \tag{4.32}$$

$$y_0 = y_A \tag{4.33}$$

$$y_1 = \eta_1 \sin\theta_A \tag{4.34}$$

$$y_2 = \frac{1}{2}\left(\eta_3 \sin\theta_A + \eta_1^2 \kappa_A \cos\theta_A\right) \tag{4.35}$$

$$\begin{aligned} y_3 = {} & 10(y_B - y_A) - (6\eta_1 + \frac{3}{2}\eta_3)\sin\theta_A - (4\eta_2 - \frac{1}{2}\eta_4)\sin\theta_B \\ & - \frac{3}{2}\eta_1^2 \kappa_A \cos\theta_A + \frac{1}{2}\eta_2^2 \kappa_B \cos\theta_B \end{aligned} \tag{4.36}$$

$$\begin{aligned} y_4 = {} & -15(y_B - y_A) + (8\eta_1 + \frac{3}{2}\eta_3)\sin\theta_A + (7\eta_2 - \eta_4)\sin\theta_B \\ & + \frac{3}{2}\eta_1^2 \kappa_A \cos\theta_A - \eta_2^2 \kappa_B \cos\theta_B \end{aligned} \tag{4.37}$$

$$\begin{aligned} y_5 = {} & 6(y_B - y_A) - (3\eta_1 + \frac{1}{2}\eta_3)\sin\theta_A - (3\eta_2 - \frac{1}{2}\eta_4)\sin\theta_B \\ & - \frac{1}{2}\eta_1^2 \kappa_A \cos\theta_A + \frac{1}{2}\eta_2^2 \kappa_B \cos\theta_B \end{aligned} \tag{4.38}$$

The real parameters η_i, $i = 1, \ldots, 4$, appearing in the above relations, can be packed together to form the four-dimensional parameter vector $\boldsymbol{\eta} := [\eta_1 \eta_2 \eta_3 \eta_4]^T$ so that the resulting parametric curve be concisely denoted as $\mathbf{p}(u; \boldsymbol{\eta})$.

Proposition 2 (Guarino Lo Bianco and Piazzi[26]). Given any interpolating data $\mathbf{p}_A$, θ_A, κ_A and $\mathbf{p}_B$, θ_B, κ_B, the parametric curve $\mathbf{p}(u; \boldsymbol{\eta})$ satisfies conditions (4.24)–(4.26) for all $\eta_1, \eta_2 \in \mathrm{IR}^+$ and all $\eta_3, \eta_4 \in \mathrm{IR}$. Conversely, given any quintic polynomial curve $\mathbf{p}(u)$ with $\dot{\mathbf{p}}(0) \neq 0$, $\dot{\mathbf{p}}(1) \neq 0$ satisfying (4.24)–(4.26) there exists parameters $\eta_1, \eta_2 \in \mathrm{IR}^+$ and $\eta_3, \eta_4 \in \mathrm{IR}$ such that it can be expressed as $\mathbf{p}(u; \boldsymbol{\eta})$.

Note that $\eta_1 = \dot{\mathbf{p}}(0; \boldsymbol{\eta})$, $\eta_2 = \dot{\mathbf{p}}(1; \boldsymbol{\eta})$ are "velocity" parameters whereas η_3, η_4 can be denoted as "twist" parameters of the curve. These parameters can be chosen in order to appropriately shape the trajectory, for example minimizing the curvature in a lane changing maneuver or minimizing the variations of curvature following a road arc with constant curvature.

The recursive use of the parametric curve $\mathbf{p}(u; \boldsymbol{\eta})$ permits to exactly interpolate any given sequence of cartesian points with arbitrarily assigned unit tangent and curvature vectors. In such a way it results an overall G^2-curve, i.e. a curve with second order geometric continuity. In the following we refer to $\mathbf{p}(u; \boldsymbol{\eta})$ as the quintic G^2-spline.

Figures 21 and 23 help describing the new overall feedforward/feedback control of the ARGO vehicle on the path following functionality. A new quintic G^2-spline is planned at the chosen trajectory updating rate. The vision data system with the IPM can give, on a planar $\{x, y\}$-coordinate system, the road scene with known car's position and path to follow. Hence $\mathbf{p}_A$ and θ_A are known and the current curvature κ_A of the vehicle path can be computed with relation (4.21).

The interpolating point $\mathbf{p}_B$ is determined on the path to follow at the interpolating distance I_D from vehicle's rear axle midpoint. The vision data processing should also provide the tangent angle θ_B with respect to axis x and the path curvature κ_B at $\mathbf{p}_B$. The supervisor (see figure 21), collect the interpolating data, assign the shaping vector parameter $\boldsymbol{\eta}$, and pass all the data to the G^2-spline generator. From the knowledge of $\mathbf{p}(u; \boldsymbol{\eta})$ a dynamic inversion based on model (4.4) is performed via formulae (4.8)–(4.10) for obtaining the steering angle function

$$\delta(t) = \arctan[l\kappa(u; \boldsymbol{\eta})]|_{u = s^{-1}(v(t - t_0))} \,. \tag{4.39}$$

In (4.39), $\kappa(u; \boldsymbol{\eta})$ denotes the curvature of the G^2-spline $\mathbf{p}(u; \boldsymbol{\eta})$ and v is the vehicle velocity that is considered constant during the trajectory updating time, i.e. the time slot where the steering action (4.39) is based on the updated G^2-spline.

The feedback action is issued by the supervisor by planning a new G^2-spline, so that the steering control (4.39) is updated, when the vehicle has covered a relatively small fraction of the current G^2-spline. This mechanism is recursively applied to the new G^2-spline resulting in an overall continuous steering control that makes the vehicle converging on the desired path.

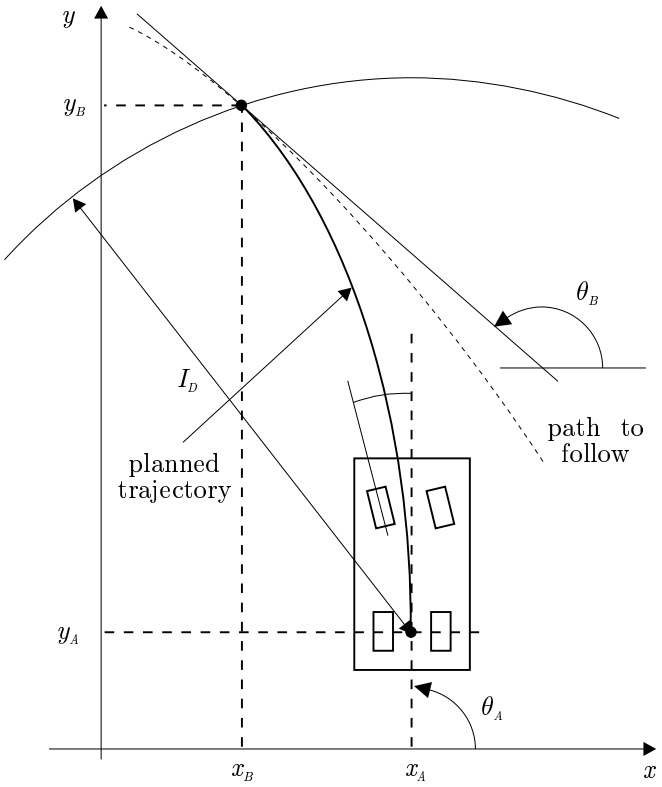


Fig. 23. Vision-based planning of the G^2-spline.

4.4. *Simulation Results*

The new flatness-based recursive trajectory control scheme, shown in figure 21, has been simulated. To obtain realistic results, the Wong car lateral model has been adopted.[27] It takes into account the geometric and dynamic characteristics of the vehicle as well as the tires characteristics. A summary of the model parameters is reported in tab. 2. In that table m indicates the vehicle mass, J the inertia with respect to the center of gravity, l_f and l_r indicate the distance of the center gravity with respect to the front and the rear wheels respectively, C_f and C_r are the cornering stiffness of the front and the rear wheels and μ is the coefficient of road adhesion ($\mu = 1$ for dry roads). It has been supposed that the frame rate of the visual system is the same of the proportional controller currently used in the ARGO vehicle: 50 Hz. The delay time from the image acquisition to the control actuation has been considered in the simulations. A reasonable value of 0.008 s, deduced from the gain scheduled proportional controller used by real ARGO vehicle, has been adopted. The supervisory controller renews the spline coefficients with a rate that is a fraction of the frame rate. The reason for this choice is intrinsic to the selected control strategy. In fact, to correctly approach the desired

Table 2. Parameters for the simulated car.

m	=	1300 kg
J	=	2900 kg m^2
l_f	=	1.15 m
l_r	=	1.52 m
$C_f = C_r$	=	45000 N rad^{-1}
μ	=	1

road path, a proper percentage of the spline path must be covered before it is updated. Practically, we have a spline updating rate equal to $50/\nu$ with ν being a positive integer. The parameter ν is selectable and is chosen to obtain the best performance from the controlled system. Normally, the spline updating rate have to be increased proportionally with the car speed. Also the interpolating distance I_D, i.e. the ahead distance used to plan the spline, is proportional to the vehicle's velocity in a way similar to (4.1). Vector $\boldsymbol{\eta}$ has been chosen to guarantee the generation of optimal trajectories in the cases of lane change, straight roads or curves with small curvatures (the curvature range considered practically covers all the standard highways curvatures). "Optimal" means that the error between the generated spline and the road path is minimum and, at the same time, this result is obtained with minimum variations of the steering command δ.[26] Roads with very large curvatures requires different values for $\boldsymbol{\eta}$. A suboptimal behaviour must be expected for sharp curves because in this simulations $\boldsymbol{\eta}$ is supposed to be constant ($\boldsymbol{\eta} = [25\ 25\ -45\ 45]^T$) .

The simulation results reported in the following refer to the road shown in figure 24. It is composed by a sequence of line segments and circular arcs joined to obtain an overall G^1-path (cf. subsection 4.2). The road shape, that is discontinuous at segment/arc joints, complicates the control task because the controller always plans trajectories with continuous curvatures. Road curvatures are chosen inside the interval $[0, 0.005]$ m^{-1}. The maximum curvature (corresponding to 200 m of radius) is bigger than standard highway curvatures: the purpose is to test the control strategy in critical situations, i.e. when the chosen value of $\boldsymbol{\eta}$ is not the most appropriate.

Two different car speeds have been used for the simulations. In the first run a constant velocity of 10 m/s (36 km/h) has been chosen, while in the second run the velocity has been increased to 30 m/s (108 km/h). In the first simulation ν was fixed equal to 30 (which corresponds to a spline update time of 0.6 s) while in the second one it was decreased to 10 (spline update time equal to 0.2 s). Owing to the Wong's model used to describe the car, the actual car paths are different for the two cases, but in both situations the maximum tracking error is smaller than 20 cm and it occurs when the curvature radius of the road becomes equal to 200 meters. On the straight stretches the error is very close to zero.

At the beginning, if the vehicle is not on the correct path, it converges toward

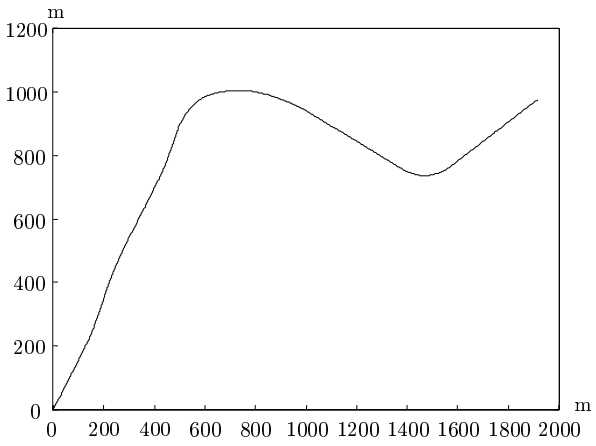


Fig. 24. The test road.

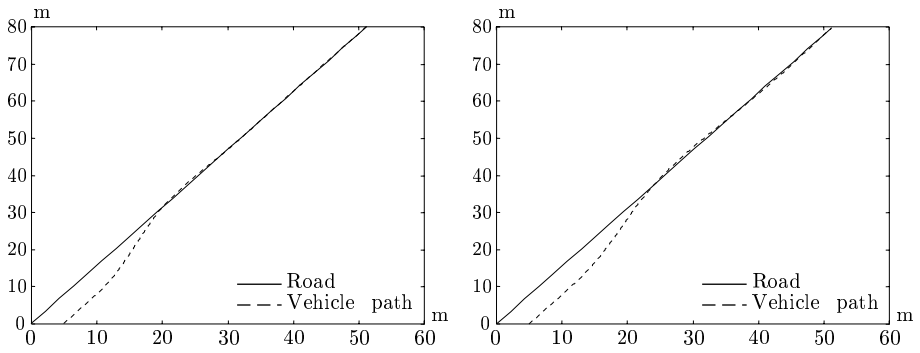


Fig. 25. Low speed (left) and high speed (right) road approaches.

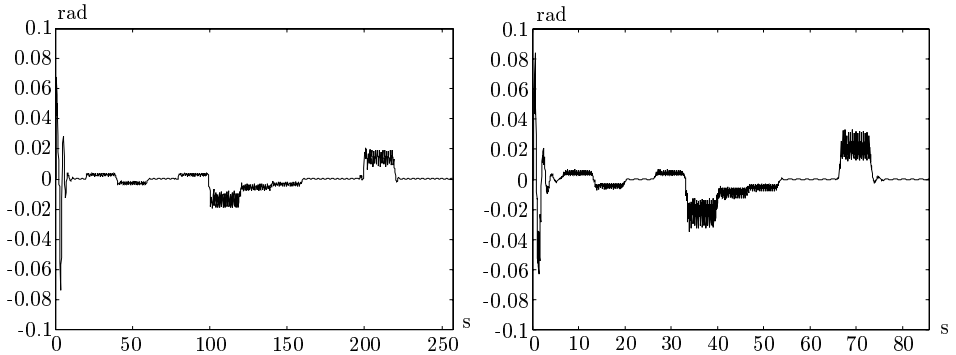


Fig. 26. Command steering angle δ for a car speed equal to 10 m/s (36 km/h) (left) and 30 m/s (108 km/h) (right).

the desired path. Figure 25 shows how the road is approached in the two cases: the behavior is similar even if, as expected, at low speeds the car has to cover a smaller distance to approach the road.

Figure 26 shows the steering angle δ for the whole road path. The largest

steering angle is detected at the beginning of the simulation when the vehicle has to approach the road, but even in that case it is smaller than 0.1 rad (close to 5^o). As expected, the worst situation for the control algorithm is detected when the vehicle approaches curves with the biggest curvature (0.005 m^{-1}). Nevertheless, in the worst case (i.e at low speed) the steering command δ has a peak-to-peak amplitude equal to 0.02 rad (1^o). In future works, the performances will be further improved by considering a variable $\boldsymbol{\eta}$, chosen depending on the road curvature.

5. Results

The following paragraphs present a 2000 km test on the road. For each system component a detailed analysis of the main problems encountered during the test is presented and the overall system performance is discussed.

5.1. *The* MilleMiglia in Automatico *Tour*

In order to extensively test the Road Following functionality under different traffic situations, road environments, and weather conditions, a 2000 km journey was carried out in June 1998 along the Italian highway network. The Italian road network is particularly suited for such an extensive test since it is characterized by quickly varying road scenarios including different weather conditions and a generally large amount of traffic. During the journey, besides the usual tasks of data acquisition and processing for automatic driving, the system logged the most significant data, such as speed, position of the steering wheel, lane changes, user interventions and commands, and dumped the whole status of the system (images included) in correspondence to situations in which the system had difficulties in reliably detecting the road lane.

After the end of the tour, the collected logs have been processed in order to compute the overall system performance, such as the percentage of automatic driving, and to analyze unexpected situations. At the end of the tour, the system logs contained more than 1200 Mbyte of raw data.

5.2. *Performance Analysis*

5.2.1. *The Vision System*

The low-cost cameras installed on the ARGO vehicle demonstrated to be the weakest component of the whole system. Although they have a high sensitivity even in low light conditions (e.g. the twilight), a quick change in the illumination of the scene causes a degradation in the image quality. In particular, in correspondence to a tunnel exit images are completely saturated for about 100÷200 ms therefore nullifying the processing. The use of cameras featuring automatic gain control and higher dynamics is now under evaluation.

5.2.2. *The Processing System*

The processing system installed on ARGO during the tour was a a commercial PC with a 200 MHz Pentium processor and 32 Mbyte of memory. It was able to process up to 25 pairs of stereo frames per second and provide the control signals for autonomous steering every 40 ms (e.g. one refinement on the steering wheel position for every meter at 100 km/h), proving to be powerful enough for the automatic driving of the vehicle.

5.2.3. *The Visual Processing*

The IPM approach used for Lane Detection, proved to be effective for the whole trip. Even if on Italian highways the flat road assumption is not always valid, anyway, the approximation of the road surface with a planar surface was acceptable. The wrong calibration, in fact, generates a lateral offset in the vehicle trajectory. Nevertheless, since the highway lanes' width is sufficiently large, this offset has never caused serious problems. Anyway, an enhancement to the IPM transform is currently under development.[28]

5.2.4. *The Control System*

The control system tested during the tour was based on the gain scheduled proportional controller previously discussed. With this kind of control, for speeds reaching around $90 \div 95$ km/h there was no noticeable difference in comparison to a human driver, while for higher speeds (up to 123 km/h) the vehicle tended to oscillate inside the lane.

Regarding the mechanical part, an electric stepping motor allows the rotation of the steering wheel with a high resolution and a reduced power consumption.

5.2.5. *Environmental Conditions*

During the tour, the system's behavior was evaluated in various environmental conditions. The route was chosen in order to include areas with different morphological characteristics: from flat areas to sloping territories of the Appennines region and heavy traffic zones, inevitably encountering stretches of highway with road works, absent or worn horizontal road signs, and diversions, and various weather conditions. The different environmental conditions demonstrated the robustness of the image processing algorithms.

5.2.6. *Statistical analysis of the tour*

The analysis of the data collected during the tour[29] allowed the computation of a number of statistics regarding system performance (see table 3). The automatic driving percentage and the maximum distance automatically driven show high values despite the presence of many tunnels and of several stretches of road with absent or worn lane markings or even no lane markings at all.

Table 3. Statistical data regarding the system performance during the tour.

Stage	Departure		Arrival	km	Average speed [km/h]	Maximum speed [km/h]	Percentage of automatic driving	Maximum distance in automatic [km]
1	Parma	→	Turin	245	86.6	109	93.3	23.4
2	Turin	→	Pavia	175	80.2	95	85.1	42.2
3	Pavia	→	Ferrara	340	89.8	115	86.4	54.3
4	Ferrara	→	Ancona	260	89.8	111	91.1	15.1
5	Ancona	→	Rome	365	88.4	108	91.1	20.8
6	Rome	→	Florence	280	87.5	110	95.4	30.6
7	Florence	→	Parma	195	89.0	123	95.1	25.9

It is important to note that some stages included passing through toll stations and transiting in by-passes with heavy traffic and frequent queues during which the system had to be switched off.

6. Conclusions and Future Work

The experience gained during these years of work within the ARGO project clearly highlighted some of the main problems of automatic driving, whilst the extensive use of the ARGO prototype helped to find their most promising solutions.

The main aims established at the beginning of the project were achieved, namely the development of a prototype vehicle and its use as a test-bed for both the hardware and software aspects of the project.

The critical analysis of the results of the *MilleMiglia in Automatico* tour, as well as the experience gained during the last few years, will be helpful for the future research within the ARGO project, namely the development of a new vehicle which will include the automatic control of both the steering wheel and the speed. New and more robust algorithms for Vehicle Detection are currently under development as well as a new module for Pedestrian Detection.

Also the control subsystem will derive benefit from the new strategies highlighted in this work: the expected advantages of the new flatness-based recursive trajectory control over the previous proportional control are basically: (i) superior road following with smooth cruising and (ii) highly-flexible functionality. In particular flexibility can be simply obtained by modifying the supervisor strategy in order to perform, for example, lane changing, lane inserting, platooning and even car parking maneuvers.

References

1. A. Broggi, M. Bertozzi, A. Fascioli, and G. Conte, *Automatic Vehicle Guidance: the Experience of the ARGO Vehicle*, World Scientific, Apr. 1999. ISBN 981-02-3720-0.

2. R. Frezza, S. Soatto, and G. Picci, "Visual path following by recursive spline updating," in *Proceedings of the 36th Conference on Decision and Control*, pp. 1130–1134, (San Diego, California USA), Dec. 1997.
3. Y. Ma, J. Košecká, and S. Sastry, "Vision guided navigation for a nonholomonic mobile robot," *IEEE Transaction on Robotics and Automation* 15(3), pp. 521–536, 1999.
4. C. Hatipoğlu, K. Redmill, and Ü. Özgüner, "Steering and lane change: a working system," in *IEEE Conference on Intelligent Transportation Systems, ITSC'97*, (Boston, MA), Nov. 1997.
5. C. J. Taylor, J. Košecká, R. Blasi, and J. Malik, "A comparative study of vision-based lateral control strategies for autonomous highway driving," *The International Journal of Robotic Research* 18(5), pp. 442–453, 1999.
6. H. Peng and M. Tomizuka, "Lateral control of front-wheel steering rubber-tire vehicles," Tech. Rep. UCB-ITS-PRR-90-5, Institute of transportation studies, Berkeley, CA, 1990.
7. R. E. Parson and W. B. Zhang, "Program on advanced technology for the highway lateral guidance/control," in *Proc. of the 1st Int'l Conf. Applications of advanced Technology in Transpotation Engineering*, pp. 275–280, (San Diego, CA), 1989.
8. W. Zhang, R. E. Parsons, and T. West, "An intelligent roadway reference system for vehicle lateral guidance/control," in *Proc. of the 1990 American Control Conference, ACC90*, pp. 281–286, (San Diego, CA), 1990.
9. N. Matsumoto and M. Tomizuka, "Vehicle lateral velocity and yaw rate control with two independent control inputs," in *Proc. of the 1990 American Control Conference, ACC90*, pp. 1868–1875, (San Diego, CA), 1990.
10. T. Hessburg and M. Tomizuka, "Fuzzy logic control for lateral vehicle guidance," *IEEE Control System Magazine* 14, pp. 55–63, Aug. 1994.
11. J. Ackermann and W. Sienel, "Robust control for automated steering," in *Proc. of the 1990 American Control Conference, ACC90*, pp. 795–800, (San Diego, CA), 1990.
12. J. Ackermann, "Robust car steering by yaw rate control," in *Proc. of the 1990 Conference on Decision and Control, CDC90*, pp. 2033–2034, (Honolulu), 1990.
13. J. Ackermann, *Robust Control: systems with uncertain phisical parameters*, Communications and Control Engineering, Springer-Verlag, London, Great Britain, 1993.
14. R. H. Byrne, C. T. Abdallah, and P. Dorato, "Experimental results in robust lateral control of highway vehicles," *IEEE Control Systems* 18(2), pp. 70–76, 1998.
15. B. Epiau, F. Chaumette, and P. Rives, "A new approach to visual servoing," *IEEE Trans. on Robotics and Automation* 8(3), pp. 313–326, 1992.
16. D. Pomerleau, "Progress in neural network-based vision for autonomous robot driving," in *Proc. of the Int. Vehicles '92 Symposium*, pp. 391–396, (Detroit), 1992.
17. D. Pomerleau, "RALPH: Rapidly adapting lateral position haldler," in *Proc. of the Int. Vehicles '95 Symposium*, pp. 54–59, 1995.
18. Ü. Özgüner, K. Ünyelioglu, and C. Hatipoğlu, "An analytical study of vehicle steering control," in *Proc. of the IEEE Conf. on Control Applications*, pp. 125–130, (Washington, DC), 1995.
19. S. Tsugawa, H. Mori, and S. Kato, "A lateral control algorithm for vision-based vehicles with a moving target in the field of view," in *1998 International Conference on Intelligent Vehicles*, vol. 1, pp. 41–45, (Stuttgart, Germany), Oct. 1999.
20. M. Fliess, J. Lévin, P. Martin, and P. Rouchon, "Sur les systèmes non linéaires différentiellement plots," *C. R. Acad. Sci. Paris* I–315, pp. 619–624, 1992.
21. M. Fliess, J. Lévin, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: introduction theory and examples," *Int. J. of Control* 61(6), pp. 1327–1361, 1995.
22. J. Lévine, "Are there new industrial perspectives in the control of mechanical sys-

tems?," in *Advances in Control: highlights of ECC'99*, P. M. Frank, ed., pp. 197–226, Springer-Verlag, (London, Great Britain), 1999.
23. P. Rouchon, M. Fliess, J. Lévin, and P. Martin, "Flatness, motion planning and trailer systems," in *Proceedings of the 32nd IEEE Conference on Decision and Control, CDC'93*, pp. 2700–2705, 1993.
24. P. Rouchon, M. Fliess, J. Lévin, and P. Martin, "Flatness and motion planning: the car with n trailers," in *Proceedings of the European Control Conference, ECC'93*, pp. 1518–1522, (Groninger, Netherlands), 1993.
25. B. A. Barsky and J. C. Beatty, "Local control of bias and tension in beta-spline," *Computer Graphics* 17(3), pp. 193–218, 1983.
26. C. Guarino Lo Bianco and A. Piazzi, "Quintic G^2-splines for trajectory planning of automated vehicles," Tech. Rep. TSC03/99, Dip. Ingegneria Informazione, University of Parma, Parma (Italy), Oct. 1999.
27. J. Wong, *Theory of ground vehicles*, John Wiley & Sons Inc., New York, 2nd ed., 1993.
28. M. Bertozzi, A. Broggi, and A. Fascioli, "An extension to the Inverse Perspective Mapping to handle non-flat roads," in *Procs. IEEE Intelligent Vehicles Symposium'98*, pp. 305–310, (Stuttgart, Germany), Oct. 1998.
29. A. Broggi, M. Bertozzi, and A. Fascioli, "The 2000 km Test of the ARGO Vision-Based Autonomous Vehicle," *IEEE Intelligent Systems* 14, pp. 55–64, Jan.–Feb. 1999.