Toward autonomous car driving by a humanoid robot: A sensor-based framework

Antonio Paolillo, Andrea Cherubini, François Keith, Abderrahmane Kheddar, Marilena Vendittelli

HAL Id: lirmm-01247234

https://hal-lirmm.ccsd.cnrs.fr/lirmm-01247234

Submitted on 1 Mar 2017

Toward Autonomous Car Driving by a Humanoid Robot: A Sensor-Based Framework

Antonio Paolillo[1] Andrea Cherubini[2] François Keith[2,3] Abderrahmane Kheddar[2,3] Marilena Vendittelli[1]

Abstract— To achieve the complete car driving task with a humanoid robot, it is necessary to develop a set of basic action primitives, including: walking to the vehicle, manually controlling its commands (ignition, accelerator and steering), and moving with the whole-body, for car ingress/egress. In this paper, we propose an approach for realizing the central part of the complete task, consisting in driving the car along a road. The proposed method is composed of two main parts. First, a vision-based controller uses image features of the road, to provide the reference angle for the steering wheel. Second, an admittance controller allows the humanoid to safely rotate the steering wheel with its hands and realize the desired steering command. We present results from a car driving experience, by humanoid robot HRP-4, within a video game setup.

I. INTRODUCTION

Acquiring car driving skills demands time-consuming training. In all countries, the right to drive requires a license, that is provided after months of practice, followed by tough examination test, aimed at assessing the candidate's understanding of road signals, along with his/her dexterity and reflexes. To design a robot driving in conditions similar to ours, how can such knowledge and skills be transferred?

If the vehicle cannot be customized, it is certainly convenient to design the driving robot to be anthropomorphic, since cars are designed for humans. Learning road signs and driving rules is an algorithmic problem, that can be solved relatively quickly. Instead, recognition of road signs and driving situations is harder and related to image processing and reasoning. Then, consider the robot motion control necessary for operations such as: reaching the car, entering it, sitting in a sustained and stable posture, controlling its commands (e.g., ignition, steering wheel, pedals), and finally egressing the car. All these skills can be seen as action primitives or robotic tasks to be tailored to each vehicle and robot, and, more importantly, to be properly combined to realize the complete driving task.

Tremendous efforts are made, in the robotics community, to automate vehicle guidance, with the ultimate goal of reproducing the tasks usually performed by human drivers [1], [2], [3], by relying on visual sensors [4] [5]. The success of the DARPA Urban Challenges [6] [7], and the impressive demonstrations made by Google [8] have heightened expectations that autonomous cars will soon be able to operate in environments of realistic complexity. Considering this, why bother making a robot drive a car,

if the car can make its way without a robot? This is clearly a legitimate question.

One possible answer springs from the complexity of the autonomous cars, which host a *distributed robot*, with various sensors and actuators controlling the different driving tasks. With a *centralized robot*, such embedded automation devices can be removed from the car. The skeptical reader may also wonder in which situations should a *centralized robot* be preferred to an automated car?

This question is partly answered by the recent DARPA Robotic Challenge [9], where the first among the eight robot tasks consists in driving utility vehicles. Indeed, in disaster situations, such as those exemplified by major nuclear accidents, the robot helper must operate utility vehicles normally driven by humans, not only to travel around the site, but also to transport tools, debris, or whatever needs to be moved. The robot should also be capable of operating cranks or other specific tools attached to the vehicle [10], [11]. It is unlikely to see such vehicles instrumented for automated driving. A second driving robot application demand comes from the car manufacturing industry. Typically, current crash-tests dummies are passive and unactuated, whereas in crash situations, real humans perform protective motions and stiffen their postures, operations that are programmable on humanoid robots. Therefore, robotic crash-test dummies could be more realistic in reproducing typical human behaviours. Finally, other potential applications are in the field of entertainment.

These applications highlight the interest of developing a robot driver. However, this requires the solution of an unprecedented "humanoid in the loop" control problem. In fact, one of the main contributions of this study, is that the robot must autonomously control another system. In fact, the robotic driver represents a special case study of the more general problem of operating human-tailored devices and machinery (including, e.g., levers, pedals, buttons).

In this work, we successfully tackle this issue, and demonstrate the capability of a humanoid robot to drive a car in a game setup. The next section formalizes the problem and sketches the proposed approach. The main components of the proposed control architecture are a vision-based road following controller providing the car desired steering angle described in Sect. III and an admittance control scheme used to turn the steering wheel of the desired angle illustrated in Sect. IV. Section V describes the experimental setup used to produce the results reported in Sect. V which contains also a description of the developed lane detection algorithm. Finally, Sect. VII concludes the paper and provides a short perspective on future work.

[1]Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy. E-mail: {paolillo, vendittelli}@diag.uniroma1.it

[2]Laboratory for Computer Science, Microelectronics and Robotics LIRMM - Université de Montpellier 2 CNRS, 161 Rue Ada, 34392 Montpellier, France. E-mail: {andrea.cherubini, francois.keith, kheddar}@lirmm.fr

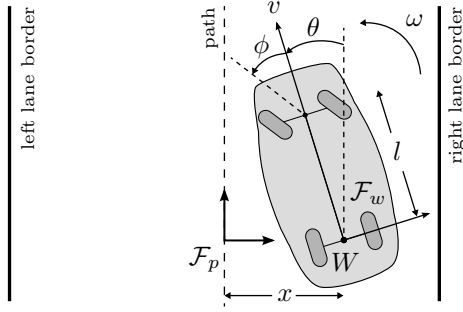[3]CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan.

Fig. 1. Top view of the car with relevant variables.

Fig. 2. Steering wheel, with rotation angle α, hand and steering frames, $\mathcal{F}_h$ and $\mathcal{F}_s$; parameters r and β are also shown here.

Fig. 3. Block diagram of the humanoid robot driving framework.

II. PROBLEM FORMULATION AND PROPOSED APPROACH

The objective of this work is to enable a humanoid robot to drive a human-tailored car along a road while keeping it at the center of a road lane using only its on-board sensors to get information about the environment.

In designing the proposed solution some simplifying assumptions have been introduced that allow capturing the conceptual structure of the problem without losing generality:

I. The car brake and clutch pedals are not considered and the driving speed is assumed to be positive and independently controlled through the throttle pedal. Hence, the steering wheel is the only vehicle control available for task execution.

II. The hands grasping configuration is constant through operation and the robot is already in its driving posture on the seat, with the hands on the steering wheel and the foot on the pedal.

III. The road is assumed to be locally flat and delimited by parallel borders, and its turns feasible with respect to the car maximum steering curvature constraint. Crossings, fire lights, or pedestrians are not negotiated, and road signs are not interpreted.

For control design purposes, we consider a kinematic model of the car. This is a reasonable approximation, in view of the simplifying assumptions of flat terrain and low accelerations. We consider a typical rear-wheel driving car model, where the front wheels can be steered, while the rear wheels are directly controlled by the engine.

Road following consists in driving the rear wheel axis center W onto the curvilinear path that is equally distant from the left and right road lane borders (see Fig. 1), while aligning the car with the tangent to this path. We consider a Frenet Frame $\mathcal{F}_p$, tangent to the path, and with origin on the normal projection of W onto it. The car configuration with respect to the path, is defined by the Cartesian abscissa of W in $\mathcal{F}_p$, x, and by the car orientation with respect to the path tangent, $\theta \in \mathbb{SO}(2)$. Then, following the path consists in zeroing both x and θ. Describing the car motion through the model of a unicycle with an upper curvature bound $c_M \in \mathbb{R}^+$, x and θ evolve according to:

$$\begin{cases} \dot{x} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \qquad \left| \frac{\omega}{v} \right| < c_M. \qquad (1)$$

In eq. (1), v and ω represent respectively the linear and angular velocity of the unicycle. The kinematic model (1) is used
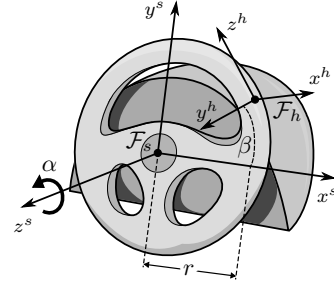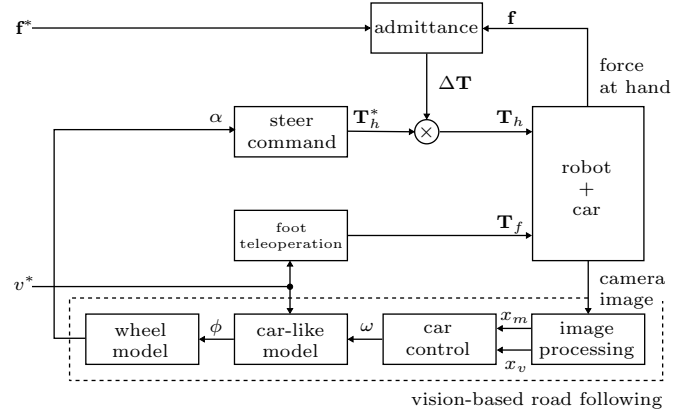
to design the vision-based controller providing the reference angular velocity for the car. The task of the controller is to stabilize system (1) at the origin $\{x, \theta\} \to \{0, 0\}$.

The front wheel orientation ϕ can be approximately related to v and ω through:

$$\phi = \arctan \left(\frac{\omega l}{v} \right), \qquad (2)$$

with l the constant distance between rear and front wheel axes[1]. Neglecting the dynamics of the steering mechanism, considered faster than that of the vehicle, the front wheel orientation ϕ is set proportional to the steering wheel angle α, controlled by the driver hands, and shown in Fig. 2. Assuming small angles $\omega l/v$ in (2), this leads to:

$$\alpha = k_\alpha \frac{\omega}{v}, \qquad (3)$$

with k_α a positive scalar characteristic of the car (and accounting also for l). An admittance controller [12] allows the robot to safely operate the steering wheel to realize the desired α.

The car velocity is controlled by manually teleoperating the robot foot so as to obtain a desired velocity v^*. As will appear clear in the following, this simplification does not cause any loss of generality of the proposed vision-based road following strategy, to be detailed in Sect. III.

The proposed control architecture can be sketched as in the block diagram in Fig. 3. The vision-based road following

---

[1]Note that the maximum curvature bound in (1) leads to a bound on the front wheel axis orientation: $|\phi| < \arctan(c_M l)$.
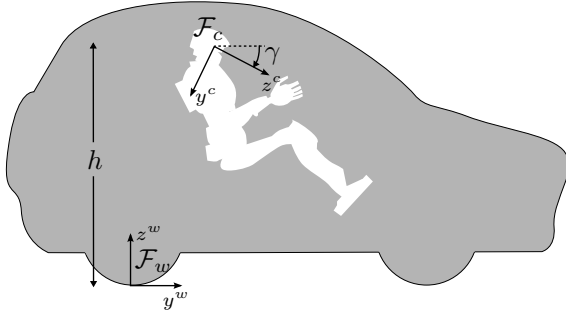
Fig. 4. Schema of a humanoid robot driving a car. The robot on-board camera frame $\mathcal{F}_c$ is at an height h from $\mathcal{F}_w$ and is tilted by an angle γ.

Fig. 5. Definition of the middle and vanishing points, respectively x_m and x_v, given the road borders.

controller provides reference values for the steering angle α according to the feedback from the robot camera. These are transformed into reference poses, described by $\mathbf{T}_h^*$, for the robot hands and further adapted, as described by $\mathbf{T}_h$, by the admittance controller to achieve safe interaction, as will be explained in Sect. IV. The diagram also shows the conceptual control scheme for the accelerator pedal operation: given the desired velocity v^*, a foot reference pose described by $\mathbf{T}_f$ is generated by the teleoperation module embedded in the robot controller. This architecture allows easy integration of an automatic control module for the driving velocity, possibly based on vision, in future developments.

III. VISION-BASED ROAD FOLLOWING

This section describes the vision-based controller that, based on the elaboration of the images from the robot on-board camera, generates motion commands for the robot hands to drive the car along the road by zeroing both x and θ.

The controller is based on the algorithm introduced in [13] for unicycle corridor following, and recently applied to the navigation of humanoids in maze-like environments [14]. In view of the assumption III in the previous section, the same algorithm can be applied here. In the following, we first recall the derivation of the features model and of the control law presented in [13], [15], then we illustrate the adaptation needed to deal with the considered problem.

With reference to Fig. 4, the Cartesian coordinates of the road feature points in $\mathcal{F}_p$ are related to their projection on the camera image plane through the projection matrix $\mathbf{P}$:

$$\lambda \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \mathbf{P} \begin{pmatrix} x_p^g \\ y_p^g \\ 1 \end{pmatrix}, \qquad (4)$$

where, considering that the road points are on the ground, we have set the z coordinate to zero.

The projection matrix $\mathbf{P}$ is given by:

$$\mathbf{P} = \begin{pmatrix} f_x c_\theta & -f_x s_\theta & -f_x x c_\theta \\ -f_y s_\gamma s_\theta & -f_y s_\gamma c_\theta & f_y(s_\gamma x s_\theta + h c_\gamma) \\ c_\gamma s_\theta & c_\gamma c_\theta & h s_\gamma - x c_\gamma s_\theta \end{pmatrix}. \quad (5)$$

In (4) and (5):

- $(x_i, y_i)^T$ is the projection on the image plane (in pixels) of the ground point $(x_p^g, y_p^g)^T$ (in meters);
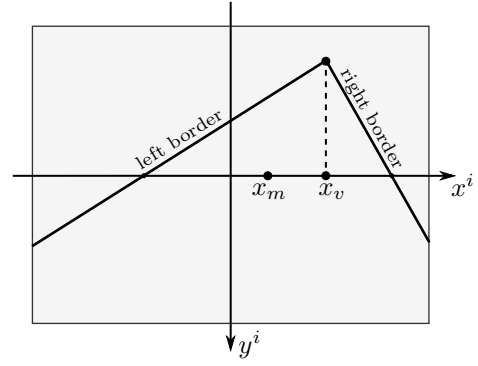- $\lambda > 0$ is the $(x_p^g, y_p^g)^T$ depth in the camera frame $\mathcal{F}_c$;

- $f_{x,y}$, are the camera focal lengths along x and y, expressed in pixels;
- c_* and s_* denote $\cos(*)$ and $\sin(*)$ respectively;
- x and θ define the car error with respect to the path (see Fig. 1);
- $\gamma \in (0, \pi/2)$ is the tilt angle of the camera (positive downwards);
- $h > 0$ is the camera height from the ground.

The images of the two road borders intersect as a consequence of the perspective transformation. Two features can be defined in the image plane: (i) the *vanishing point*, i.e., the intersection of the borders images and (ii) the *middle point*, i.e., the midpoint of the segment between the intersection points of the same borders images with the horizontal axis of the image plane (see Fig. 5).

The abscissa of the vanishing and middle point, respectively denoted by x_v and x_m, can be expressed as [13]:

$$\begin{cases} x_v = k_1 \tan \theta \\ x_m = k_2 \dfrac{x}{c_\theta} + k_3 \tan \theta, \end{cases} \qquad (6)$$

with $k_1 = -f_x/c_\gamma$, $k_2 = -f_x s_\gamma/h$ and $k_3 = -f_x c_\gamma$.

Note that, the values of the above constants have been determined considering the following operational conditions: the principal point coincides with the image center, there is no image distortion, the ordinate y of the camera optical center in $\mathcal{F}_w$ is negligible, and the camera pose in $\mathcal{F}_w$ is constant throughout operation.

Stabilization of x_m dynamics by feedback provides the following vehicle angular velocity control [13]:

$$\omega = \frac{k_1}{k_1 k_3 + x_m x_v} \left(-\frac{k_2}{k_1} v x_v - k_p x_m \right), \qquad (7)$$

with k_p a positive scalar gain. This controller guarantees asymptotic convergence of both middle and vanishing points to zero, under the conditions that $v > 0$, and that k_2 and k_3 have the same sign, which is always true if $\gamma \in (0, \pi/2)$. According to (6), asymptotic convergence of x_v and x_m implies convergence of x and θ to zero achieving the desired path following task.

To realize the desired angular velocity ω in (7), the steering wheel must be turned according to (3):

$$\alpha = \frac{k_\alpha k_1}{k_1 k_3 + x_m x_v} \left(-\frac{k_2}{k_1} x_v - k_p \frac{x_m}{v} \right). \qquad (8)$$

In (8), x_m and x_v can be estimated through an image processing algorithm like the one described in Sect. VI, while the value of the velocity v can be either estimated through vision (for instance, by looking at the car dashboard, or via optical flow) or set equal to the nominal value commanded through the pedal inclinations. In either case, the vision-based controller needs some degree of robustness to inaccuracies in the estimation of v, an issue that will be discussed in the experimental Section.

The reference steering angle α is converted to reference poses of the hands grasping the wheel through the rigid transformation $\mathbf{T}_h^* = \mathbf{T}_s(\alpha)\mathbf{T}_h^s(r, \beta)$, where $\mathbf{T}_h^*$ and $\mathbf{T}_s$ are the transformation matrices expressing respectively the poses of frames $\mathcal{F}_h$ and $\mathcal{F}_s$ in Fig. 2 with respect to $\mathcal{F}_w$ in Fig. 4; $\mathbf{T}_h^s$ expresses the pose of $\mathcal{F}_h$ with respect to $\mathcal{F}_s$ and depends on the steering wheel radius r, and on the angle β parameterzing the position of the hands on the wheel.

IV. ADMITTANCE INTERACTION CONTROL

The vision-based road following algorithm provides continuously the hand reference pose for turning the steering wheel. For a safe interaction between the robot grippers and the steering wheel, it is obvious to think of an admittance or impedance controller, rather than solely a force or position controller. We choose an admittance scheme for the robot arms which consists of the following equations:

$$\mathbf{f} - \mathbf{f}^* = \mathbf{M}\Delta\ddot{\mathbf{x}} + \mathbf{B}\Delta\dot{\mathbf{x}} + \mathbf{K}\Delta\mathbf{x}, \tag{9}$$

where $\mathbf{f}$ and $\mathbf{f}^*$ are respectively the sensed and desired generalized interaction forces in $\mathcal{F}_h$; $\mathbf{M}$, $\mathbf{B}$ and $\mathbf{K} \in \mathbb{R}^{6\times6}$ are respectively the mass, damping and stiffness diagonal matrices. As a consequence of the force applied on $\mathcal{F}_h$, and on the base of the values of the admittance matrices, (9) generates variations of pose $\Delta\mathbf{x}$, velocity $\Delta\dot{\mathbf{x}}$ and acceleration $\Delta\ddot{\mathbf{x}}$ of $\mathcal{F}_h$ with respect to $\mathcal{F}_s$. Thus, the solution of (9) leads to the vector $\Delta\mathbf{x}$ that can be used to compute the transformation matrix $\Delta\mathbf{T}$, and to build up the new desired pose for the robot hands: $\mathbf{T}_h = \mathbf{T}_h^*\Delta\mathbf{T}$.

V. EXPERIMENTAL SETUP

As experimental platform, we used the full-sized humanoid robot HRP-4 by Kawada Industries. HRP-4 is 1.5 m tall, weighs 39 kg and has 34 degrees of freedom. Among the other sensors, HRP-4 is equipped with two ATI Mini45 force/torque sensors mounted at each wrist, and an Asus Xtion RGB-D sensor placed on its head, used in this work as a monocular camera.

The force sensors provide measures of the forces and torques acting on the robot hands at 200 Hz, while the Xtion provides a video stream with a resolution of 640×480 at 30 Hz. A standard calibration process has been used to obtain the values of camera focal lengths: $f_x = f_y \simeq 530$ pixels.

To reproduce realistic driving conditions, we used the video game *City Car Driving*^{TM}. Thus, the real car has been replaced by pedals and a steering wheel *Thrustmaster T500*^{TM} set, used as remote controllers to send velocity and steering commands to the simulated vehicle embedded in the game. As a response, the view from the car is displayed on a monitor, and observed by the robot, in order to close the visual feedback loop through (8).
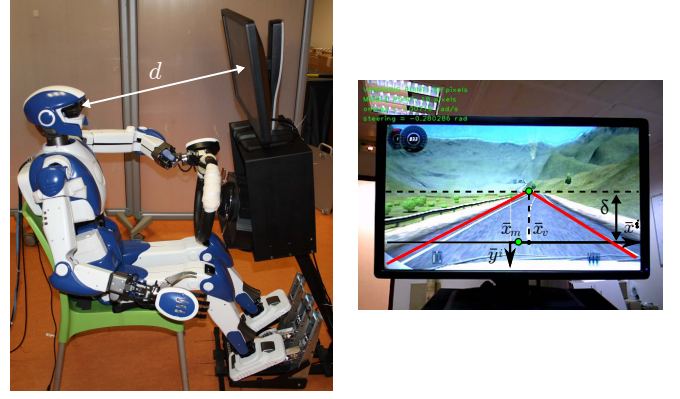

Fig. 6. Driving experimental setup.

The robot camera points to the monitor, where the simulated road is shown. Hence, a second projection must be modeled, from the video game virtual camera, to compute $\mathbf{P}$ in (4). The optical axis of this camera is constantly parallel to the ground, a feature that cannot be modified by the gamer. Without loss of generality, we have decided to align the robot camera optical axis with the virtual camera, as shown in Fig. 6, i.e., to set $\gamma = 0$. This alignment place the horizon in the center of the image plane and is easily doable since the robot neck is fully actuated.

The game camera projection then simply introduces on the image coordinates the scale factors $S_x = f_x^g r_x/d$ and $S_y = f_y^g r_y/d$, with $f_{x,y}^g$ the game camera focal lengths (in pixels), r_x (r_y) the horizontal (vertical) size of the monitor pixels in meters, and $d > 0$ the distance (in meters) between the robot camera optical center and the monitor plane. For the monitor used in the experiments, it is $r_x = r_y \simeq 0.00027$ m/pixels and it has been placed at distance $d = 0.55$ m from the robot camera.

In this setup the projection model (4) becomes:

$$\mathbf{P} = \begin{pmatrix} S_x f_x c_\theta & -S_x f_x s_\theta & -S_x f_x x c_\theta \\ 0 & 0 & S_y f_y h \\ s_\theta & c_\theta & -x s_\theta \end{pmatrix},$$

and the vanishing and middle points coincide:

$$x_v = x_m = -S_x f_x \tan\theta. \tag{10}$$

This is a crucial issue for the controller, since only the orientation θ of the vehicle can be regulated, while feedback control of the lateral offset x is impossible.

The reason for (10) is that, since the game camera optical axis is parallel to the ground, the horizon (including the vanishing point) collapses on the image abscissa axis. Obviously, modifying the robot camera tilt angle with respect to the monitor will not solve this problem. Instead, lowering the image plane abscissa by a negative offset δ (in pixels), so that in the new frame $\overline{y} = y + \delta$, yields a proper expression for the vanishing and middle points (see Fig. 6):

$$\begin{cases} \overline{x}_v = \overline{k}_1 \tan\theta \\ \overline{x}_m = \overline{k}_2 \dfrac{x}{c_\theta} + \overline{k}_3 \tan\theta, \end{cases}$$

with $\overline{k}_1 = \overline{k}_3 = -S_x f_x$, $\overline{k}_2 = \frac{S_x f_x \delta}{S_y f_y h}$.
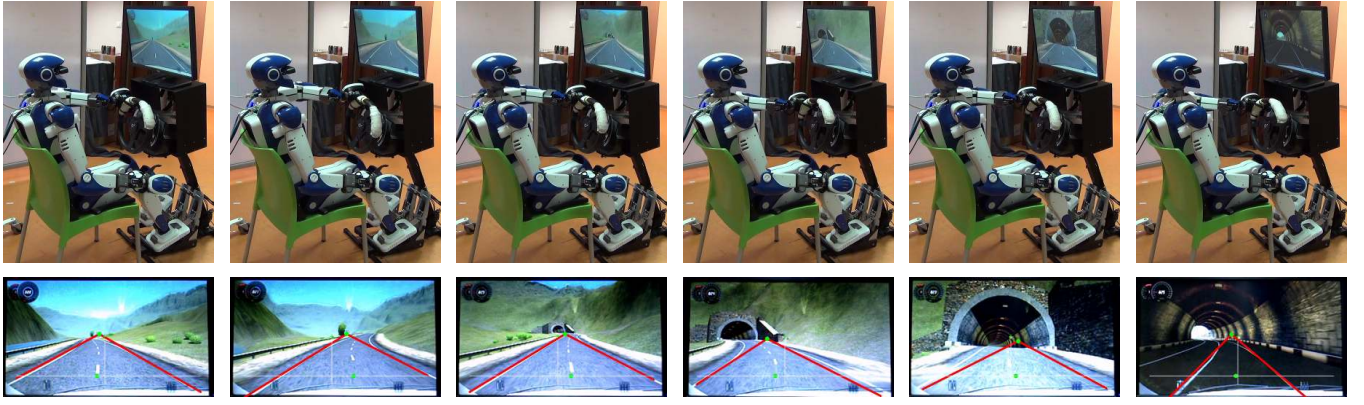
Fig. 7. Snapshots from a driving experiment with HRP-4 in a simulated environment (top) and the corresponding image processing frames (below).

Note that the above expressions of the features are formally equal to the nominal case (6). Then, since $\bar{k}_2$ and $\bar{k}_3$ have the same sign ($\delta < 0$), the control law (8) with $\bar{k}_{1,2,3}$ in place of $k_{1,2,3}$, will guarantee path following.

VI. EXPERIMENTAL RESULTS

This section presents the results obtained with the above described experimental setup. The control parameters were set to: $\delta = -100$ pixels, $v^* = 25$ km/h, $k_\alpha = 1.5$ m and $k_p = 5.0$ rad/s. The focal length and the height of the camera embedded in the video game with respect to the ground could not be directly estimated. For this reason, their value has been set heuristically ($f_x^g = f_y^g = 500$ pixels and $h = 0.5$ m), and verified to be reasonable during the experimental tests. However, from a practical point of view, this is not a relevant issue, since once the framework will be ported on a real car it will be possible to estimate the robot camera height for example by localization of the humanoid, like in [16].

A calibration phase at the beginning of each experiment ensured the alignment between the focal axes of the robot camera and the simulated car view, while an initial procedure was used to arrange the robot on the seat, with the hands on the wheel and the foot on the pedal to comply with the assumption of known robot posture within the car.

The lane borders extractor algorithm is based on OpenCV [17], a toolbox library for vision-based applications. In particular, first a Gaussian filter is applied to smooth the image, then the Canny edge detector and the Hough transform are used to extract all line segments (except for the vertical ones) visible on the lower half of the image plane. Each detected line segment i ($i = 1, \dots, n$) is characterized by its length l_i, and by its equation $y_i = m_i x_i + q_i$. These segments are used to compute the $x-$coordinate of the vanishing point according to:

$$x_v = \frac{\sum_{i=1}^{n} \sum_{j>i}^{n} x_{i,j} w_{i,j}}{\sum_{i=1}^{n} \sum_{j>i}^{n} w_{i,j}}$$

where $x_{i,j}$ is the intersection of lines i and j, and $w_{i,j}$ is a weight associated with each intersection, and defined as:

$$w_{i,j} = t_1(l_i + l_j) + t_2|m_i - m_j| + t_3|q_i - q_j|.$$

The $w_{i,j}$ are high if the two lines are similar; the t_k are three constants set heuristically to $t_1 = 1$, $t_2 = 10$ and $t_3 = 10$.

Once the vanishing point has been detected, the lane borders have to be found, among the subset of lines passing near the vanishing point. These are grouped in two sets based on their slopes, and then the groups are merged according to the average of their slope, to obtain the right and left lane borders. Finally, the middle point is obtained as midpoint of the intersection of the two borders with the horizontal axis at height δ. The reference steering angle α computed with (8) is cleaned from the measurement noise, via a low-pass filter.

Note that, since the focus of this work was not to advance the state of the art on lane detection but rather to propose a control architecture for humanoid car driving, in our implementation we choose to develop a simple image processing algorithm. More performing results can be obtained using approaches like, e.g., [18], [19], [20], [21].

To obtain a good tracking of α while realizing a safe interaction of the robot with the wheel, the admittance system (9) has been designed to have: (i) fast and stiff behaviour along the $z-$axis of the hand frame $\mathcal{F}_h$, and (ii) slow and compliant motion along the x and $y-$axes. This corresponds to the following diagonal elements of the mass, damping and stiffness matrices (only force components):

$$m_x = m_y = 2000 \text{ kg}, \ m_z = 10 \text{ kg},$$

$$b_x = b_y = 1600 \text{ kg/s}, \ b_z = 240 \text{ kg/s},$$

$$k_x = k_y = 20 \text{ kg/s}^2, \ k_z = 1000 \text{ kg/s}^2.$$

To improve grasp stability, the desired force component along the $y-$axis of $\mathcal{F}_h$ has been set to a non-null value:

$$f_x^* = f_z^* = 0 \text{ N}, f_y^* = 5 \text{ N}.$$

The car velocity v^* has been set to 25 km/h through teleoperation of the robot foot.

Finally, the reference pose for left hand[2] $\mathbf{T}_h$ is sent to the kinematic controller embedded in the stack of tasks framework [22]. A specific task is also defined for the gaze to keep the camera fixed with respect to the monitor (i.e., to guarantee constant h and γ and the alignment with the focal of the simulated camera.).

Snapshots from one of the run experiments are shown in Fig. 7, and in the video accompanying the paper. The robot

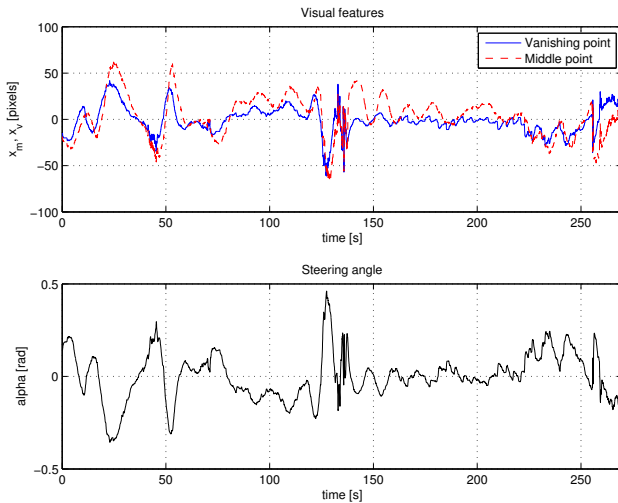[2]Using one or both hands is conceptually equivalente in the hypothesis of constant grasp configuration.

Fig. 8. Visual features (top) and reference steering angle (below) evolution.

successfully drives the car for about 300 s along a curved road, with varying slope and lighting conditions (compare, e.g., the first and fifth snapshots in Fig. 7).

Figure 8 shows the evolution of x_m and x_v, and the corresponding reference steering angle α. Peaks in the values of x_m and x_v correspond to road turns. We are confident that accounting for the path curvature in the controller will further reduce its value. Both x_m and x_v are close to zero when the road is straight (e.g., between 160 to 220 s), and overall varying between -50 and 50 pixels, i.e., within a vertical strip that is less than 10% of the image width.

Finally, it is worth noticing that the driving velocity v^* used in the controller (8) might not correspond to the real velocity v of the car due to the presence of slopes in the road, friction, and delay and errors in human teleoperation. In this work an extensive experimental campaign has provided evidence of the robustness of controller with respect to inaccuracies in velocity estimation. We are currently studying the stability of the system under this and other non-persistent perturbations due, e.g., to non-parallel road borders.

VII. Conclusions

In this paper we have proposed a control architecture for car driving by a humanoid robot. The approach consists in extracting road visual features used to determine a reference steering angle to keep the car at the center of a road lane. The steering wheel is compliantly operated by the robot hands, while, in this preliminary work, the acceleration pedal is controlled by teleoperating the robot foot. To our knowledge, this closed-loop approach is novel, since all of the DARPA Robotic Challenge participants, including WRECS, the winners of the driving task [23], have remotely operated the robot.

The results also open encouraging perspectives for real outdoor experiments. We have started testing road feature extraction on images recorded from a real human-driven car, and the results are shown in the accompanying video. Future work include the integration in the control framework of a vision-based velocity estimation module, accelerator and brake control, variable hand grasping configurations during operation.

REFERENCES

[1] U. Nunes, C. Laugier, and M. M. Trivedi, "Guest editorial introducing perception, planning, and navigation for intelligent vehicles," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 375–379, 2009.
[2] Y. Zhang, W. Lin, and Y.-K. Chin, "Driving skill characterization: A feasibility study," in *IEEE Int. Conf. on Robotics and Automation, 2008*, 2008, pp. 2571–2576.
[3] M. Hentschel and B. Wagner, "Autonomous robot navigation based on open street map geodata," in *13th Int. IEEE Conf. on Intelligent Transportation Systems*, 2010, pp. 1645–1650.
[4] P. Newman, *et al.*, "Navigating, recognizing and describing urban spaces with vision and lasers," *The Int. Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1406–1433, 2009.
[5] A. Broggi, L. Bombini, S. Cattani, P. Cerri, and R. I. Fedriga, "Sensing requirements for a 13,000 km intercontinental autonomous drive," in *2010 IEEE Intelligent Vehicles Symposium (IV)*, 2010, pp. 500–505.
[6] K. L. M. Buehler and S. S. (Editors), "Special issue on the 2007 darpa urban challenge, part I-III," *Journal of Field Robotics*, 2008.
[7] S. Thrun, *et al.*, "Stanley: The robot that won the Darpa Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.
[8] http://en.wikipedia.org/wiki/Google_driverless_car.
[9] www.fbo.gov/spg/oda/darpa/cmo/darpa-baa-12-39/listing.html.
[10] H. Hasunuma, K. Nakashima, M. Kobayashi, F. Mifune, Y. Yanagihara, T. Ueno, K. Ohya, and K. Yokoi, "A tele-operated humanoid robot drives a backhoe," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 2998–3004.
[11] K. Yokoi, K. Nakashima, M. Kobayashi, H. Mihune, H. Hasunuma, Y. Yanagihara, T. Ueno, T. Gokyuu, and K. Endou, "A tele-operated humanoid operator," *Int. Journal of Robotics Research*, vol. 5-6, pp. 593–602, 2006.
[12] N. Hogan, "Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications," *Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, 1985.
[13] J. Toibero, C. Soria, F. Roberti, R. Carelli, and P. Fiorini, "Switching visual servoing approach for stable corridor navigation," in *IEEE Int. Conf. on Advanced Robotics*, 2009, pp. 1–6.
[14] A. Faragasso, G. Oriolo, A. Paolillo, and M. Vendittelli, "Vision-based corridor navigation for humanoid robots," in *IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 152–158.
[15] R. F. Vassallo, H. J. Schneebeli, and J. Santos-Victor, "Visual servoing and appearance for navigation," *Robotics and Autonomous Systems*, vol. 31, no. 1–2, pp. 87–97, 2000.
[16] G. Oriolo, A. Paolillo, L. Rosa, and M. Vendittelli, "Vision-based odometric localization for humanoids using a kinematic EKF," in *2012 12th IEEE-RAS Int. Conf. on Humanoid Robots*, 2012, pp. 153–158.
[17] G. Bradski, *Dr. Dobb's Journal of Software Tools*, 2000.
[18] W. Liu, H. Zhang, B. Duan, H. Yuan, and H. Zhao, "Vision-based real-time lane marking detection and tracking," in *11th Int. IEEE Conf. on Intelligent Transportation Systems, 2008*, 2008, pp. 49–54.
[19] K. H. Lim, K. P. Seng, A. C. L. Ngo, and L.-M. Ang, "Real-time implementation of vision-based lane detection and tracking," in *Int. Conf. on Intelligent Human-Machine Systems and Cybernetics, 2009*, vol. 2, 2009, pp. 364–367.
[20] M. Meuter, S. Muller-Schneiders, A. Mika, S. Hold, C. Nunny, and A. Kummert, "A novel approach to lane detection and tracking," in *12th Int. IEEE Conf. on Intelligent Transportation Systems, 2009*, 2009, pp. 1–6.
[21] M. Nieto, A. Cortés, O. Otaegui, J. Arróspide, and L. Salgado, "Real-time lane tracking using rao-blackwellized particle filter," *Journal of Real-Time Image Processing*, pp. 1–13, 2012.
[22] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks," in *Int. Conf. on Advanced Robotics*. IEEE, 2009, pp. 1–6.
[23] http://spectrum.ieee.org/automaton/robotics/humanoids/darpa-robotics-challenge-trials-results.