DIIS - I3A
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 2009-V7

Parking with the Essential Matrix without Short Baseline Degeneracies [1]

G. López-Nicolás, C. Sagüés and J.J. Guerrero

If you want to cite this report, please use the following reference instead:
Parking with the Essential Matrix without Short Baseline Degeneracies, G. López-Nicolás, C. Sagüés and J.J. Guerrero
IEEE International Conference on Robotics and Automation, pages 1098-1103, Kobe - Japan, 2009.

[1]This work was supported by projects DPI2006-07928 and IST-1-045062-URUS-STP.

Parking with the Essential Matrix without Short Baseline Degeneracies

G. López-Nicolás, C. Sagüés and J. J. Guerrero

Abstract

This paper addresses the problem of visual control of a mobile robot. The system consists of a calibrated camera fixed onboard a robot with nonholonomic motion constraints. The parking task is defined by a reference image taken at the target location. The proposed control law is based on the essential matrix, but unlike traditional methods, it is not used to compute pose parameters. Instead, the control law is defined directly in terms of individual entries of the essential matrix by means of the input-output linearization of the system. Here we solve the problem of degeneracies due to short baseline by taking advantage of the planar motion constraint of the robot. Thus, a virtual target is defined providing a stable estimation of the essential matrix without degeneracies despite short baseline.

I. Introduction

We consider a mobile robot with a fixed calibrated camera mounted onboard, and a goal defined by a reference image taken at the target location. The problem is to design a visual control that autonomously drives the robot to the goal by using visual information. Visual control, or visual servoing, is an extensive field of research in which computer vision is used in the design of motion controllers, an overview is given in [1].

Some traditional visual control approaches are based on the epipolar geometry [2]–[5], but this model degenerates with short baseline (i.e., small translation). So, as the robot approaches to the target, the epipolar geometry becomes unstable. This problem has been solved using auxiliary procedures at the last stage of the motion [6]–[8]. However, it would be better to avoid the need of switching the model in favor of robustness and simplicity. A good alternative is the homography-based approach [9]–[14]. However, if no plane is detected, the homography-based control fails. This problem can be solved through virtual planes [15], but in general, estimations based on virtual planes with wide baseline are not robust to mismatches, noise or occlusions.

The algebraic representation of the epipolar geometry is the fundamental matrix. In case of using normalized image coordinates the essential matrix is obtained, being removed the effect of the known camera calibration matrix. We propose a new control scheme based on the essential matrix in which, instead of estimating pose parameters, the controller is designed directly in terms of the entries of the essential matrix. This reduces the amount of required computation and enhances the robustness of the system. The control law proposed is defined by means of the input-output linearization of the different entries of the essential matrix, transforming the nonlinear problem in a tracking task where the motion strategy is defined in terms of the essential matrix elements.

As said, a well known problem related with the epipolar geometry is that this model becomes undefined when there is no translation between the images (current and target). This results in a poor and unstable estimation of the fundamental (essential) matrix with short baseline. Instead of switching to another control law we propose a novel approach avoiding short baseline degeneracies. For this purpose we define a virtual target taking advantage of the planar motion constraint. This virtual target is generated from the visual information of the current and target images and it is defined out of the motion plane and vertically with respect to the target. This procedure relies on the epipolar transfer properties. The essential matrix computed across the current image and virtual target is always well defined despite short baseline taking into account the planar motion constraint. By using the particular form of the essential matrix with respect to the virtual target in the control law design we manage to control the robot to the target location. This approach needs the current and target images and the camera calibration for the essential matrix computation. No other additional information is required and it does not need to compute the robot location, depth or 3D information of the scene.

II. Control Scheme

An overview of the control scheme is shown in Fig. 1. Features are extracted and matched between the image taken at the current location and the virtual target image. The essential matrix $\mathbf{E}(t)$ is estimated from the set of matches and the intrinsic camera calibration. The control error is then computed with respect to the desired evolution of the essential matrix entries $\mathbf{E}^d(t)$. Then, the control law gives the velocities that allow to track the desired trajectories of the essential matrix entries reaching the target. Next Section introduces the essential matrix within our framework.

This work was supported by projects DPI2006-07928 and IST-1-045062-URUS-STP.
DIIS - I3A, Universidad de Zaragoza, Spain.
{gonlopez, csagues, jguerrer}@unizar.es

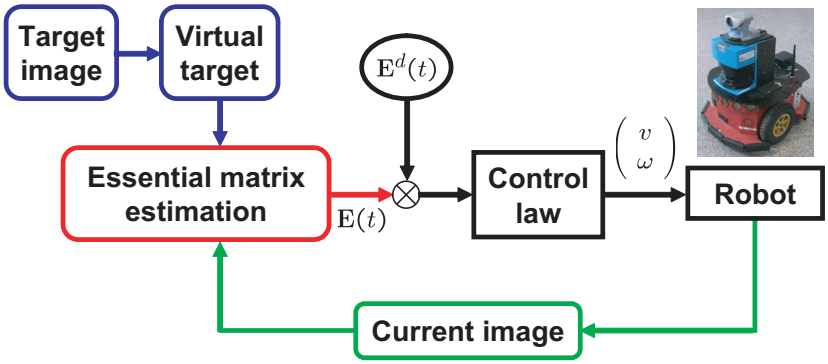
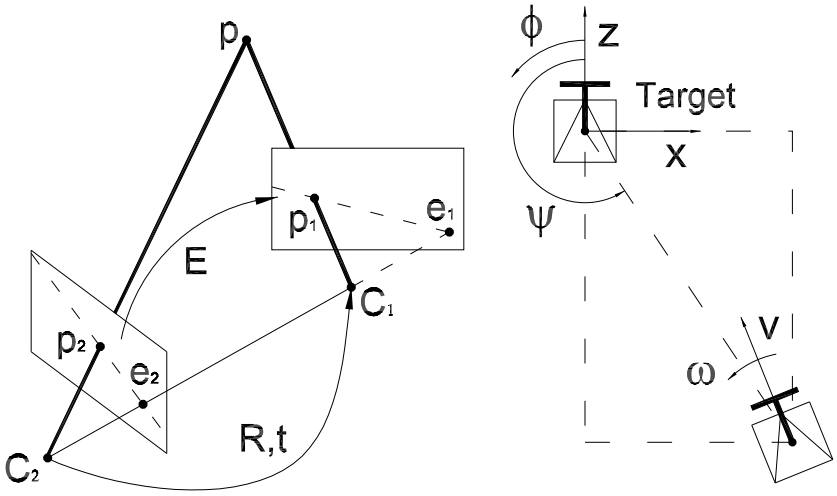


Fig. 1. Overview of the control loop.



(a) (b)

Fig. 2. (a) Essential matrix relating two views. The camera optical centers are $\mathbf{C}_1$ and $\mathbf{C}_2$. A 3D point $\mathbf{P}$ is projected in the images as ($\mathbf{p}_1$, $\mathbf{p}_2$). The epipoles are $\mathbf{e}_1$ and $\mathbf{e}_2$. (b) Coordinate system fixed in the target location.

A. The Essential Matrix

Consider the geometry of the camera to be modelled by perspective projection; and let us suppose two images obtained with the same camera. The essential matrix across two views (see Fig. 2(a)) is defined as

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}\,, \tag{1}$$

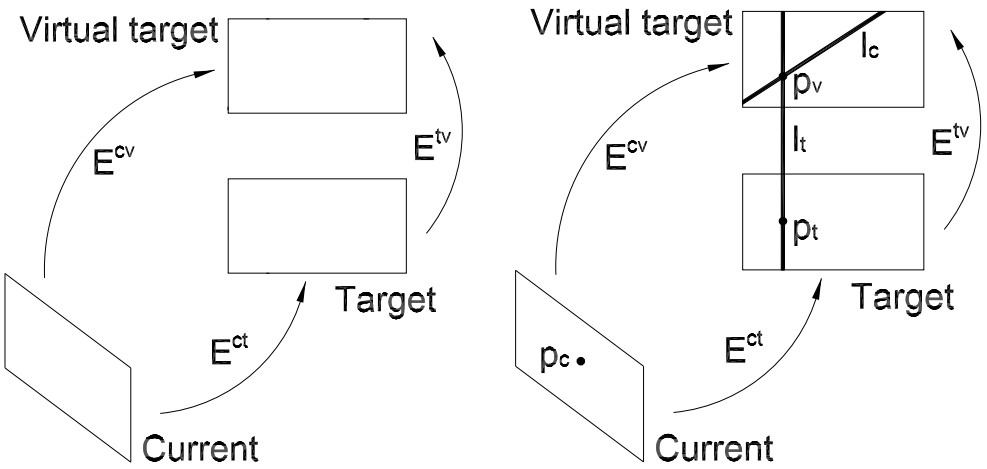
being $\mathbf{R}$ the rotation and $\mathbf{t}$ the translation between the cameras. The essential matrix can be related with the fundamental matrix with $\mathbf{E} = \mathbf{K}^T\mathbf{F}\mathbf{K}$, being $\mathbf{K}$ the internal camera calibration [16].

We assign the coordinate system to the target location as shown in Fig. 2(b), where the camera optic axis is assigned to be the z-axis of the robot frame. Under this convention the configuration of the robot system is given by $\mathbf{x} = (x, z, \phi)^T$, where x and z are the robot position in the plane, and ϕ is the orientation of the robot. We also define $\tan\psi = -x/z$. The robot has two scalar velocity inputs, linear velocity v and angular velocity ω. The framework considered in our approach consists of the target image taken at the desired location (i.e. at $\mathbf{x} = (0, 0, 0)^T$, with $y = 0$), the current image (at $\mathbf{x} = (x, z, \phi)^T$, with $y = 0$) and the virtual target image (at $\mathbf{x} = (0, 0, 0)^T$, with $y = Y = cte \neq 0$). These images are denoted as c (current), t (target) and v (virtual target). This framework is shown in Fig. 3(a), where the essential matrices across the images are denoted as $\mathbf{E}^{ct}$, $\mathbf{E}^{tv}$ and $\mathbf{E}^{cv}$. The essential matrix $\mathbf{E}^{cv}$ relating the current and virtual images can be parameterized using

$$\mathbf{R}^{cv} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix}, \; \mathbf{t}^{cv} = -\begin{pmatrix} x \\ Y \\ z \end{pmatrix}$$

with (1) as

$$\mathbf{E}_n^{cv} = \begin{bmatrix} -Y\sin\phi & z & -Y\cos\phi \\ x\sin\phi - z\cos\phi & 0 & x\cos\phi + z\sin\phi \\ Y\cos\phi & -x & -Y\sin\phi \end{bmatrix}. \tag{2}$$



(a) (b)

Fig. 3. (a) Essential matrices across the views. The virtual target is defined above the target. (b) A 3D point is projected into the three views as $(\mathbf{p}_c, \mathbf{p}_t, \mathbf{p}_v)$. The value of $\mathbf{p}_v$ is computed by means of the epipolar transfer.

The essential matrix is computed up to scale and subindex n denotes it is not normalized. Given that in our approach there is no decomposition of the matrix or estimation of depth parameters, we need to fix a scale. We can normalize $\mathbf{E}_n^{cv}$ by entry E_{13}^{cv} given that $Y \neq 0$ and considering typical field of view constraints ($|\phi| < \pi/2$). The normalization $\mathbf{E}_n^{cv}/E_{13}^{cv}$ yields

$$\mathbf{E}^{cv} = \begin{bmatrix} \tan\phi & \frac{-z}{Y\cos\phi} & 1 \\ \frac{-x\tan\phi+z}{Y} & 0 & \frac{-x-z\tan\phi}{Y} \\ -1 & \frac{x}{Y\cos\phi} & \tan\phi \end{bmatrix}. \tag{3}$$

The goal of our controller is to lead the values of the $\mathbf{E}^{cv}$ entries to their desired values in such a way that the target location is reached. The target location is determined by $(x, z, \phi) = (0, 0, 0)$ while $y = Y$ holds constant with respect to the virtual target. Using this values in (3) we obtain the desired form of the essential matrix as

$$\mathbf{E}^{tv} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}. \tag{4}$$

The advantage of using this virtual target is that $\mathbf{E}^{cv}$, unlike $\mathbf{E}^{ct}$, is well defined with short baseline and therefore, the problem of short baseline degeneracies is avoided. But first, we need to define the virtual target from the information available: the current and target images.

B. Definition of the Virtual Target

The virtual target is defined at the beginning of the navigation (i.e. only at $t = 0$). The idea is to generate the virtual target using the epipolar transfer across the three images [16]. For this purpose we need to compute the three essential matrices (Fig. 3(a)). $\mathbf{E}^{ct}(t = 0)$ is estimated from the correspondences between the current and target images and $\mathbf{E}^{tv}$ is defined as (4). The procedure for computing $\mathbf{E}^{cv}(t = 0)$ is explained next.

The matrix $\mathbf{E}^{ct}$ can be parameterized up to a unknown scale factor μ from (1) with $y = 0$ and normalized with

$$\mu\, r(0) = \sqrt{(\mu E_{12}^{ct})^2 + (\mu E_{32}^{ct})^2} = \mu\sqrt{x^2 + z^2} \tag{5}$$

giving

$$\mathbf{E}^{ct}(0) = \begin{bmatrix} 0 & \frac{z}{r} & 0 \\ \frac{x\sin\phi - z\cos\phi}{r} & 0 & \frac{x\cos\phi + z\sin\phi}{r} \\ 0 & \frac{-x}{r} & 0 \end{bmatrix}, \tag{6}$$

from which the next expression can be derived:

$$\tan\phi(0) = \frac{E_{21}^{ct}E_{32}^{ct} - E_{12}^{ct}E_{23}^{ct}}{E_{12}^{ct}E_{21}^{ct} + E_{23}^{ct}E_{32}^{ct}}. \tag{7}$$

Note that $r(t=0)$ is never zero except if the initial position is in the target location. On the other hand, we can write (2) to be compared with (6) as

$$\mathbf{E}^{cv}(0) = \begin{bmatrix} \frac{-Y\sin\phi}{r} & \frac{z}{r} & \frac{-Y\cos\phi}{r} \\ \frac{x\sin\phi - z\cos\phi}{r} & 0 & \frac{x\cos\phi + z\sin\phi}{r} \\ \frac{Y\cos\phi}{r} & \frac{-x}{r} & \frac{-Y\sin\phi}{r} \end{bmatrix} . \tag{8}$$

Entries E^{cv}_{12}, E^{cv}_{21}, E^{cv}_{22}, E^{cv}_{23} and E^{cv}_{32} are known as they are the same as the corresponding entries of (6). Entries E^{cv}_{11}, E^{cv}_{13}, E^{cv}_{31} and E^{cv}_{33} depend on ϕ, which is known from (7), and Y/r. We do not know r (because of the unknown scale μ) and we do not care about Y except it is not equal to zero. Then, we can assign Y/r an arbitrary value (not equal to zero) determining $\mathbf{E}^{cv}$. In practice, a good option is to select Y in the same order of magnitude as the initial r, for example $Y/r = 1$.

We know the three essential matrices $\mathbf{E}^{ct}$, $\mathbf{E}^{tv}$ and $\mathbf{E}^{cv}$ relating the three views and we can apply the epipolar transfer to generate the virtual target (Fig. 3(b)). Let points $\mathbf{p}_c$ and $\mathbf{p}_t$ in the current and target views be a correspondence. Our objective is to find the corresponding point in the virtual image. The required point $\mathbf{p}_v$ matches point $\mathbf{p}_c$ in the current image and $\mathbf{p}_t$ in the target image. Therefore, it must lie on the epipolar lines corresponding to $\mathbf{p}_c$ and $\mathbf{p}_t$. These epipolar lines $\mathbf{l}_c$ and $\mathbf{l}_t$ can be computed since the matrices $\mathbf{E}^{cv}$ and $\mathbf{E}^{tv}$ are known:

$$\mathbf{p}_c^T \mathbf{E}^{cv} \mathbf{p}_v = 0 \quad \Rightarrow \quad \mathbf{l}_c^T = \mathbf{p}_c^T \mathbf{E}^{cv} . \tag{9}$$

$$\mathbf{p}_t^T \mathbf{E}^{tv} \mathbf{p}_v = 0 \quad \Rightarrow \quad \mathbf{l}_t^T = \mathbf{p}_t^T \mathbf{E}^{tv} . \tag{10}$$

The required point $\mathbf{p}_v$ is the intersection of the epipolar lines

$$\mathbf{p}_v = \mathbf{l}_c \times \mathbf{l}_t = \left((\mathbf{E}^{cv})^T \mathbf{p}_c\right) \times \left((\mathbf{E}^{tv})^T \mathbf{p}_t\right) . \tag{11}$$

This procedure is repeated for all the correspondences between the current and target image at the beginning of the navigation. Then, we obtain the virtual target image defined with a set of points. This virtual target allows to compute $\mathbf{E}^{cv}(t)$ along the navigation to be used in the control law. The essential matrix $\mathbf{E}^{cv}(t)$ computed from the correspondences is then normalized for (3). Note that control performance depends on the quality of the estimated virtual target at $t = 0$.

Hereafter we denote $\mathbf{E} = \mathbf{E}^{cv}$ for ease of the notation.

C. Control Law

In this section we present the control law designed by means of the input-output linearization of the essential matrix $\mathbf{E}$ (i.e. $\mathbf{E}^{cv}$). The system to be controlled is a nonholonomic robot whose model is expressed in a general way as

$$\left\{ \begin{array}{rcl} \dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} & = & \mathbf{E}(\mathbf{x}) \end{array} \right. \tag{12}$$

where $\mathbf{x} = (x, z, \phi)^T$ denotes the state vector, $\mathbf{u}$ the system input vector (v, ω), and $\mathbf{y}$ the output vector consisting of the essential matrix entries. The kinematics of a differential drive vehicle are expressed as a function of the robot state and input velocities by

$$\begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega . \tag{13}$$

The essential matrix entries relate the state parameters non-linearly. A linearization is carried out by differentiating the essential matrix entries until we can solve for the control outputs. The derivatives of $\mathbf{E}$ with respect to time give

$$\dot{E}_{11} = \dot{E}_{33} = \frac{d}{dt}(\tan\phi) = \frac{\omega}{\cos^2\phi} \tag{14}$$

$$\dot{E}_{12} = \frac{d}{dt}\left(\frac{-z}{Y\cos\phi}\right) = -\frac{v}{Y} - E_{12}E_{33}\omega \tag{15}$$

$$\dot{E}_{21} = \frac{d}{dt}\left(\frac{z - x\tan\phi}{Y}\right) = \frac{v}{Y\cos\phi} - \frac{E_{32}\omega}{\cos\phi} \tag{16}$$

$$\dot{E}_{23} = \frac{d}{dt}\left(\frac{-z\tan\phi - x}{Y}\right) = \frac{E_{12}\omega}{\cos\phi} \tag{17}$$

$$\dot{E}_{32} = \frac{d}{dt}\left(\frac{x}{Y\cos\phi}\right) = \frac{-E_{33}v}{Y} + E_{32}E_{33}\omega \tag{18}$$

Entries $\dot{E}_{12}$ and $\dot{E}_{33}$ have been selected for the control law design. The decision is taken since this is the only combination of possible pair of entries that avoids singularities in the control matrix. In Section III it is shown that the system is controllable

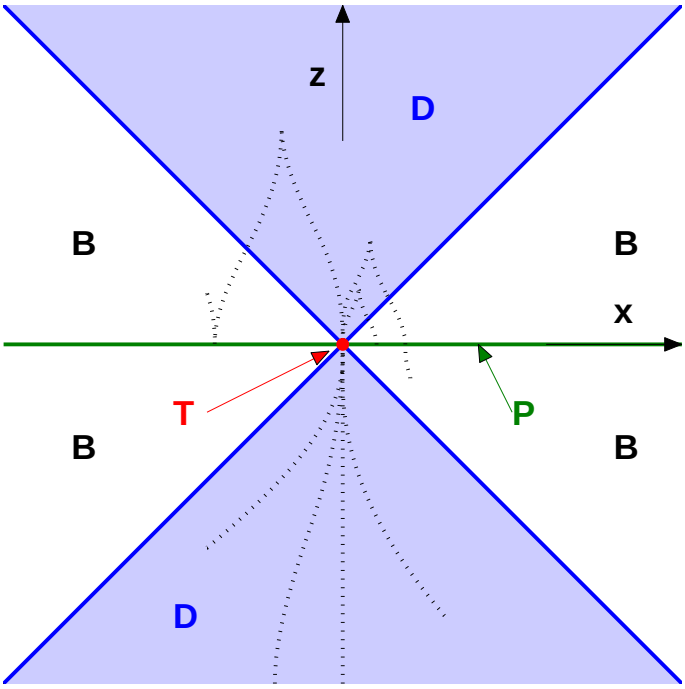


Fig. 4. Zones for defining the motion strategy: T (Target location), D (Direct motion), P (Parking manoeuvres) and B (Before parking manoeuvres). The resultant robot motion of several examples are shown in dotted line.

with these two entries. Now we have a linear relation between the velocities and the differentiated entries of the essential matrix:

$$\begin{pmatrix} \dot{E}_{12} \\ \dot{E}_{33} \end{pmatrix} = \mathbf{L} \begin{pmatrix} v \\ \omega \end{pmatrix} \text{ with } \mathbf{L} = \begin{bmatrix} \frac{-1}{Y} & -E_{12}E_{33} \\ 0 & \frac{1}{\cos^2\phi} \end{bmatrix}. \tag{19}$$

Solving for the robot velocities we have

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \mathbf{L}^{-1} \begin{pmatrix} \nu_{12} \\ \nu_{33} \end{pmatrix} , \tag{20}$$

where

$$\mathbf{L}^{-1} = \begin{bmatrix} -Y & -E_{12}E_{33}Y\cos^2\phi \\ 0 & \cos^2\phi \end{bmatrix} . \tag{21}$$

The new inputs of the control $(\nu_{12}, \nu_{33})^T$ are defined as a function of the current values of the essential matrix entries $(E_{12}, E_{33})^T$ and their desired values $(E^d_{12}, E^d_{33})^T$ which are the trajectories to track:

$$\begin{pmatrix} \nu_{12} \\ \nu_{33} \end{pmatrix} = \begin{pmatrix} \dot{E}^d_{12} \\ \dot{E}^d_{33} \end{pmatrix} - \begin{bmatrix} k_{12} & 0 \\ 0 & k_{33} \end{bmatrix} \begin{pmatrix} E_{12} - E^d_{12} \\ E_{33} - E^d_{33} \end{pmatrix} , \tag{22}$$

being $k_{12} > 0$ and $k_{33} > 0$ the control gains. The tracking error from the new control inputs results in an exponentially stable error dynamics [17].

We need to guarantee $\mathbf{L}$ to be invertible. The computation of its determinant yields

$$det(\mathbf{L}) = \frac{-1}{Y\cos^2\phi} \neq 0 , \tag{23}$$

which is always guaranteed.

D. Motion Strategy

In this section we propose the desired trajectories of the essential matrix entries (E^d_{12}, E^d_{33}) to be tracked resulting in the desired robot motion. For this purpose we first split the workspace in different zones depending on the motion strategy required to reach the target location. This partition, which is inspired in [18], is shown in Fig. 4. Zone T refers to the target location, zone P refers to locations with $z = 0$ (which is equivalent to $E_{12} = 0$). If $|z| > |x|$ (which is equivalent to $|E_{12}| > |E_{32}|$) the location is inside zone D otherwise inside zone B.

The motion strategy consists of a sequence of steps required to reach the target depending on the initial location of the robot. This is summarized in Table I, where the particular sequence of steps required to reach the target from every initial

TABLE I

SEQUENCE OF STEPS REQUIRED TO REACH THE TARGET LOCATION DEPENDING ON THE INITIAL ZONE WHERE THE ROBOT IS.

						T T_0
				D T_0	$\longrightarrow \mid \longrightarrow$ (T_1)	T T_2
		P T_0	$\longrightarrow \mid \longrightarrow$ (T_1)	D T_2	$\longrightarrow \mid \longrightarrow$ T_3	T T_4
B T_0	$\longrightarrow$	P T_1	$\longrightarrow \mid \longrightarrow$ T_2	D T_3	$\longrightarrow \mid \longrightarrow$ T_4	T T_5

TABLE II

DESIRED EVOLUTION OF THE ESSENTIAL MATRIX ENTRIES FOR EACH STEP REQUIRED TO REACH THE TARGET LOCATION.

	$E^d_{12}(t)$ and $E^d_{33}(t)$
T	$E^d_{12}(T_i \leq t < \infty) = 0$ $E^d_{33}(T_i \leq t < \infty) = 0$
D → T	(24): $E^d_{12}(T_i \leq t < T_{i+2})$, $E^d_{12}(T_{i+2}) = 0$ $\lvert\phi\rvert > \lvert\psi\rvert \Rightarrow$(26): $E^d_{33}(T_i \leq t < T_{i+2})$ $\lvert\phi\rvert \leq \lvert\psi\rvert \Rightarrow$ (25): $E^d_{33}(T_i \leq t < T_{i+1})$, $E^d_{33}(T_{i+1}) = \tan(k\psi(T_i))$; (26): $E^d_{33}(T_{i+1} \leq t < T_{i+2})$
P → D	(24): $E^d_{12}(T_i \leq t < T_{i+2})$, $E^d_{12}(T_{i+2}) = -kE_{23}(T_i)$ $\lvert\phi\rvert > \lvert\psi\rvert \Rightarrow$(25): $E^d_{33}(T_i \leq t < T_{i+2})$, $E^d_{33}(T_{i+2}) = 0$ $\lvert\phi\rvert \leq \lvert\psi\rvert \Rightarrow$ (25): $E^d_{33}(T_i \leq t < T_{i+1})$, $E^d_{33}(T_{i+1}) = -sign(E_{32}(T_i))\tan(k\psi(T_i))$; (25): $E^d_{33}(T_{i+1} \leq t < T_{i+2})$, $E^d_{33}(T_{i+2}) = 0$
B → P	(24): $E^d_{12}(T_i \leq t < T_{i+1})$, $E^d_{12}(T_{i+1}) = 0$ (25): $E^d_{33}(T_i \leq t < T_{i+1})$, $E^d_{33}(T_{i+1}) = 0$

location is shown. For example (see both Fig. 4 and Table I), if it is in zone D, the robot can perform a direct motion toward the target ($T_0 - T_2$). Depending on the initial orientation an intermediate step can be required ($T_0 - T_1 - T_2$). If the motion starts in zone P, the robot is required to move away until it reaches zone D, moving next from D to T. If the robot is inside zone B it will be led to zone P before following next steps. This is the high level motion strategy and we now define the particular expressions for the essential matrix entries to fulfill it.

Three different primitive functions are defined as follows

$$E^d_{12}(T_a \leq t < T_b) = \frac{E_{12}(T_a) + E^d_{12}(T_b)}{2} + \\ + \frac{E_{12}(T_a) - E^d_{12}(T_b)}{2} \cos\left(\frac{\pi t}{T_b - T_a}\right) , \tag{24}$$

$$E^d_{33}(T_a \leq t < T_b) = \frac{E_{33}(T_a) + E^d_{33}(T_b)}{2} + \\ + \frac{E_{33}(T_a) - E^d_{33}(T_b)}{2} \cos\left(\frac{\pi t}{T_b - T_a}\right) , \tag{25}$$

$$E^d_{33}(T_a < t \leq T_b) = E_{33}(T_a)\frac{\psi(t)}{\psi(T_a)} , \tag{26}$$

where $T_a = T_i$ or T_{i+1} and $T_b = T_{i+1}$ or T_{i+2}. We have that (24) and (25) are defined by means of sinusoids, and (26) ensures that the system evolves in such a way that lateral and orientation errors are corrected simultaneously. Basically, we can relate E_{12} with the motion along z-axis and E_{33} with the robot orientation. Using these primitive functions we can define the desired evolution of the essential matrix entries for each step of the motion strategy required to reach the target location.

The set of desired functions are given in Table II. For example, in the case of moving from zone D to T we have that the desired evolution of E^d_{12} is given by (24) defined from T_i to T_{i+2} (The value of i is derived from Table I), in which the value of $E^d_{12}(T_{i+2})$ in (24) is 0 (The value of $E^d_{12}(T_i)$ is the initial value of E^d_{12} in the current step). Similar reasoning is followed with E^d_{33}, but first checking $|\phi(T_i)| > |\psi(T_i)|$ in order to take into account the initial orientation of the robot to select the proper desired function. In several of the desired functions appears the parameter k, which is used to define the desired values of the essential matrix elements at the end of the corresponding step. This parameter has been tuned experimentally to $k = 1.7$ and it reflects the motion constrains of the platform (the maximum steering angle) and the field of view constraints. Although we have not obtained an analytical expression yet, experimental evaluation shows that higher value of k implies higher rotations during the parking manoeuvres and lower k results in longer translations along z-axis.

The primitive functions (24), (25) and (26) require to know ϕ and ψ which can be computed as

$$\phi = \arctan(E_{33}) \ , \quad \psi = \arctan\left(E_{32}/E_{12}\right) \ . \tag{27}$$

III. Controllability

In this section we study the controllability of the nonholonomic mobile platform with the selected entries of the essential matrix in the control scheme. Given that the nonlinear system (12) is driftless, its controllability can be tested with the Lie Algebra rank condition (LARC) [19]. Then, the system is small-time controllable if and only if the rank of the vector space spanned by the family of vector fields available to the system along with all their brackets is of full rank at everywhere. From (13), (21) and (3) the kinematics equations are expressed as

$$\begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{bmatrix} Y\sin\phi & -z\sin^2\phi \\ -Y\cos\phi & z\sin\phi\cos\phi \\ 0 & \cos^2\phi \end{bmatrix} \begin{pmatrix} \dot{E}_{12} \\ \dot{E}_{33} \end{pmatrix} \ . \tag{28}$$

The Lie Algebra rank condition to test is

$$rank\left[\mathbf{f}_1 \quad \mathbf{f}_2 \quad [\mathbf{f}_1,\ \mathbf{f}_2] \quad ...\ \right] = 3 \ , \tag{29}$$

where

$$\{\mathbf{f}_1, \mathbf{f}_2\} = \left\{ \begin{pmatrix} Y\sin\phi \\ -Y\cos\phi \\ 0 \end{pmatrix}, \begin{pmatrix} -z\sin^2\phi \\ z\sin\phi\cos\phi \\ \cos^2\phi \end{pmatrix} \right\} \ . \tag{30}$$

The Lie bracket $[\mathbf{f}_1,\ \mathbf{f}_2]$ is defined as

$$[\mathbf{f}_1, \mathbf{f}_2] = \begin{pmatrix} Y\cos\phi(\cos^2\phi - \sin^2\phi) \\ 2Y\sin\phi\cos^2\phi \\ 0 \end{pmatrix} \ . \tag{31}$$

In order to check the rank of (29) we compute its determinant

$$|\mathbf{f}_1 \quad \mathbf{f}_2 \quad [\mathbf{f}_1,\ \mathbf{f}_2]| = -Y^2\cos^4\phi \ . \tag{32}$$

It can be seen that the span of the vector field is full rank except at $|\phi| = \pi/2$. This situation cannot happen in practice because of the typical field of view constraints of standard cameras. Therefore, the system is small-controllable with the control law defined. In case a omnidirectional visual sensor is used, like a panoramic camera, the situation of $|\phi| = \pi/2$ is feasible but can be easily detected, and a simple procedure for leaving that situation could be added.

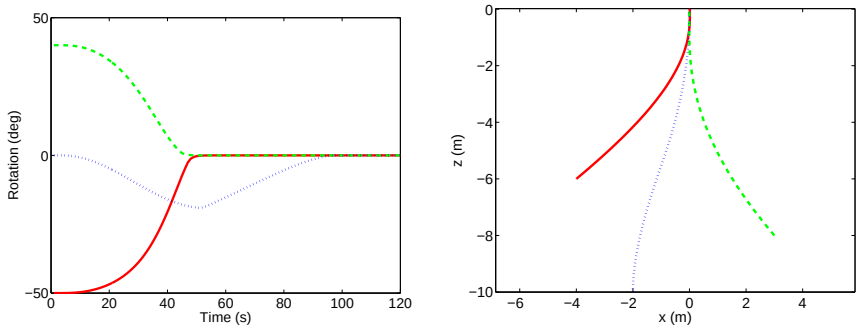
IV. Experimental Validation

In this section we present different simulations showing the performance of the control scheme proposed. The simulated data consist of a set of points projected into the camera image plane. The scene consists of a cloud of 3D points randomly distributed. In each iteration of the control loop the essential matrix is estimated from point correspondences in order to obtain the robot velocities given by the control law to reach the target location.

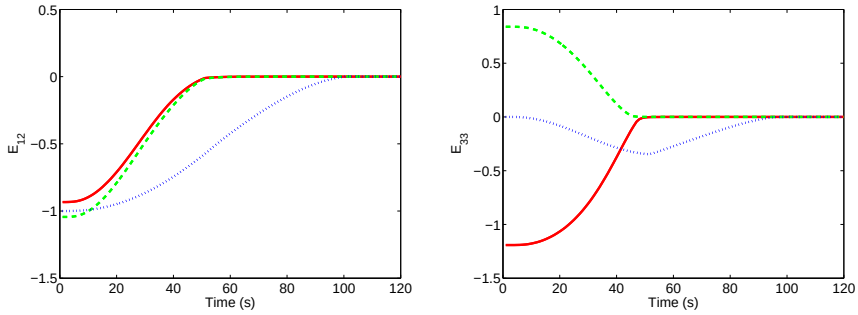
Simulations from different initial locations in zone D are shown in Fig. 5. The target location is $(0, 0, 0^\circ)$ and the initial locations are $(-4, -6, -50^\circ)$, $(-2, -10, 0^\circ)$ and $(3, -8, 40^\circ)$. The values of T_i have been selected as $\Delta T = T_{i+1} - T_i = 50\,s\ \forall i$. The three examples are superposed with different line style. The orientation and robot path evolution are shown as well as the evolution of the essential matrix entries used in the control. The resultant path in this cases is a direct motion toward the target. More simulations are shown in Fig. 6 starting from locations in zones B and P. In this case, the initial locations are $(-3, 0, 0^\circ)$, $(1, 0, 0^\circ)$ and $(2, -1, 20^\circ)$. Results show that the robot carried out the parking manoeuvres properly. More examples are given in the **video** attachment.

The results of Fig. 7 show the performance of the control law when there is image noise in the correspondences. A set of simulations have been carried out with increasing values of noise added to the image features with a standard deviation of σ. The final error in the robot coordinates is depicted, the lower errors are obtained in the robot orientation and the higher errors are obtained in the lateral coordinate (x) reflecting the nonholonomic nature of the system.

The estimation of the essential matrix requires to know the internal camera calibration. Thus, we test the performance of the control law associated with the uncertainty of the camera calibration parameters. In Fig. 8 the values of the focal length and the principal point coordinates have been fixed to $f = 8$ mm and $x_0 = 0$ pixels while their real value is changed. As expected, higher errors in the calibration parameters produce higher error in the final location. It can be seen that calibration errors mainly affect to the final lateral pose. Nevertheless, assuming usual uncertainties in the camera calibration parameters, final location errors are small and can be disregarded.

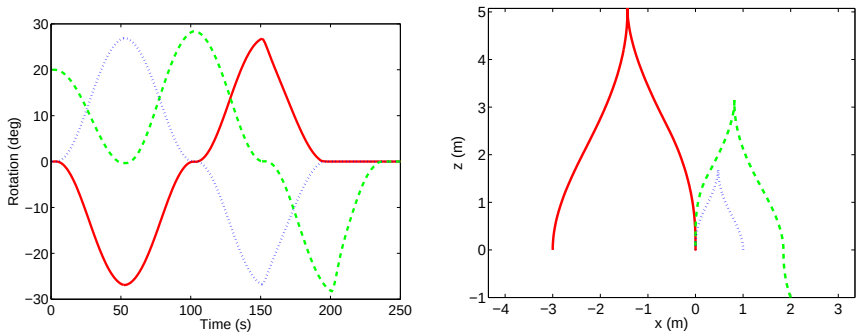


(a) Rotation ϕ (b) Paths

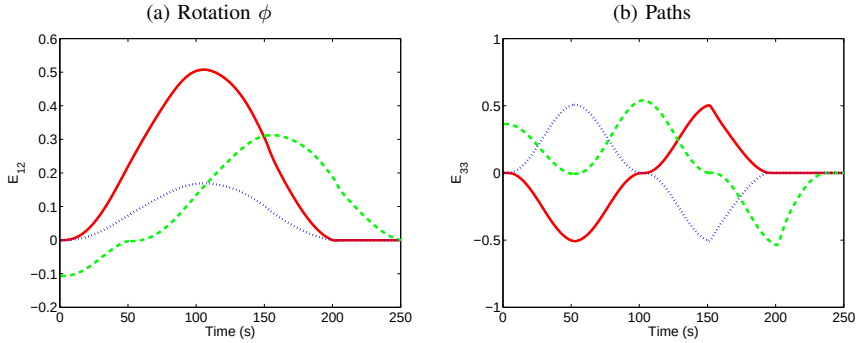


(c) Evolution of E_{12} (d) Evolution of E_{33}

Fig. 5. Simulations for three different initial positions in zone D.



(a) Rotation ϕ (b) Paths



(c) Evolution of E_{12} (d) Evolution of E_{33}

Fig. 6. Simulations for three different initial positions in zones B and P.

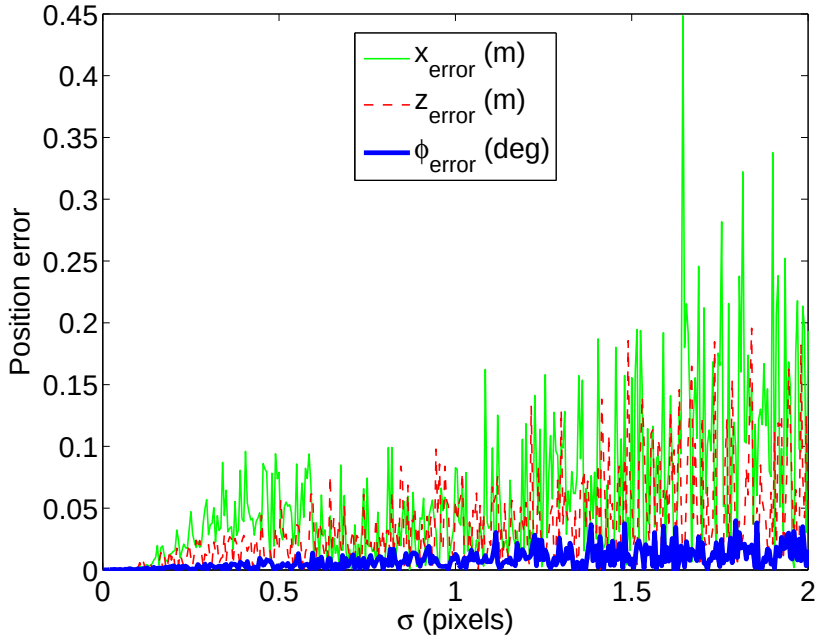


Fig. 7. Simulations with increasing image noise from $\sigma = 0$ pixels to $\sigma = 2$ pixels. The final error in $(x,\ z,\ \phi)$ is depicted.

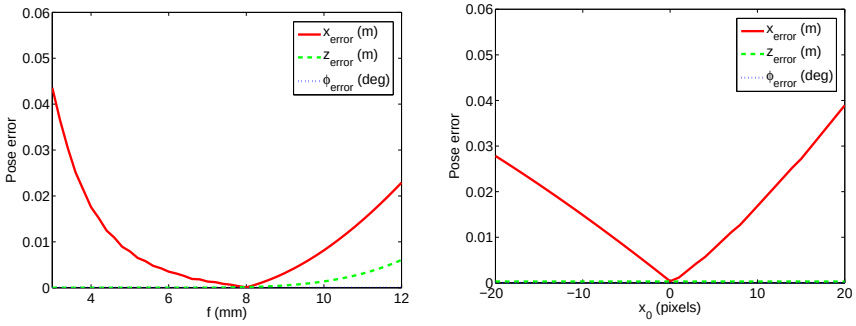


Fig. 8. Final position error in (x, z, ϕ) varying the camera calibration parameters: focal length (left) and principal point coordinates (right).

V. Conclusion

We have presented a new visual control approach for mobile robots with nonholonomic motion constraints. The contribution is the control scheme defined in terms of the essential matrix entries which does not need to compute the robot pose, depth or 3D information of the scene. The nonlinear control problem is transformed in a tracking problem where the desired trajectories of the essential matrix entries are defined. The set of desired trajectories determine the robot motion and we present the motion strategy to carry out parking maneuvers by means of sinusoids. This approach overcomes the critical issue of the essential matrix degeneracies with short baseline by means of a virtual target. The controllability analysis of the control scheme is presented and stability analysis is currently under study. The experimental evaluation consists of several simulations showing the parking manoeuvres performed to reach the target. The approach has also been tested with image noise and calibration errors.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[2] R. Basri, E. Rivlin, and I. Shimshoni, "Visual homing: Surfing on the epipoles," *International Journal of Computer Vision*, vol. 33, no. 2, pp. 117–137, 1999.

[3] P. Rives, "Visual servoing based on epipolar geometry," in *Int. Conference on Intelligent Robots and Systems*, vol. 1, 2000, pp. 602–607.

[4] G. Chesi and K. Hashimoto, "A simple technique for improving camera displacement estimation in eye-in-hand visual servoing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1239–1242, Sept. 2004.

[5] G. Chesi, G. L. Mariottini, D. Prattichizzo, and A. Vicino, "Epipole-based visual servoing for mobile robots," *Advanced Robotics*, vol. 20, no. 2, pp. 255–288, 2006.

[6] G. López-Nicolás, C. Sagüés, J. J. Guerrero, D. Kragic, and P. Jensfelt, "Nonholonomic epipolar visual servoing," in *IEEE International Conference on Robotics and Automation*, May 2006, pp. 2378–2384.

[7] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 87–100, 2007.
[8] G. López-Nicolás, C. Sagüés, J. J. Guerrero, D. Kragic, and P. Jensfelt, "Switching visual control based on epipoles for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 7, pp. 592–603, 2008.
[9] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *Transactions on Robotics and Automation*, vol. 15, no. 2, 1999.
[10] S. Benhimane, E. Malis, P. Rives, and J. R. Azinheira, "Vision-based control for car platooning using homography decomposition," in *Int. Conf. on Robotics and Automation*, April 2005, pp. 2173–2178.
[11] Y. Fang, W. E. Dixon, D. M. Dawson, and P. Chawda, "Homography-based visual servo regulation of mobile robots," *Trans. on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 5, pp. 1041–1050, 2005.
[12] J. Chen, W. Dixon, M. Dawson, and M. McIntyre, "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 406–415, April 2006.
[13] S. Benhimane and E. Malis, "Homography-based 2D visual servoing," in *Int. Conf. on Robotics and Automation*, 2006, pp. 2397–2402.
[14] G. López-Nicolás, C. Sagüés, and J. J. Guerrero, "Homography-based visual control of nonholonomic vehicles," in *IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 1703–1708.
[15] E. Malis and F. Chaumette, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.
[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
[17] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, Englewood Cliffs NJ, 1991.
[18] S. Bhattacharya, R. Murrieta-Cid, and S. Hutchinson, "Optimal paths for landmark-based navigation by differential-drive vehicles with field-of-view constraints," *Transactions on Robotics*, vol. 23, no. 1, pp. 47–59, 2007.
[19] A. Isidori, *Nonlinear Control Systems*. Springer, 1995.