

Real-time visual predictive controller for image-based trajectory tracking of a mobile robot

Guillaume Allibert, Estelle Courtial, Youssoufi Touré

*Laboratoire Vision et Robotique -LVR- UPRES EA 2078
Université d'Orléans - Polytech'Orléans,
8 rue Léonard de Vinci, 45 072 Orléans Cedex 2*

Abstract: This paper deals with the design of a real-time controller for trajectory tracking in the image plane. The Image-Based Visual Servoing (IBVS) task is addressed by a visual predictive approach. The trajectory tracking is formulated into a nonlinear optimization problem in the image plane. The unavoidable constraints in experiments are easily taken into account in the design of the predictive control law. The global model, combining the mobile robot and camera model, is used to predict the behavior of the process. The flatness property of this global model is proved in the general case, that is whatever the camera posture. The flat model permits to reduce the computational time by a factor 2. Experiments are performed on a non holonomic mobile robot with a deported perspective camera. Experimental results show the efficiency and the robustness of the real-time control approach. Visibility constraints are added to point out the capability of the control to avoid obstacles.

1. INTRODUCTION

The trajectory tracking of nonholonomic mobile robots has motivated considerable research over the past twenty years. Visual servoing, that is the use of visual information from a camera to control the motion of robotic systems, has been increasingly used. Several control schemes can be distinguished : image-based (2D), position-based (3D) and hybrid visual servoing Chaumette and Hutchinson [2006], Chaumette and Hutchinson [2007]. Since the trajectory tracking considered is in the image plane, the control architecture used will be the Image-Based Visual Servoing scheme. The IBVS principle is to minimize an image error between the desired image and the current image from the camera. While the main interest of IBVS is its robustness to modeling errors like camera calibration and measurement errors, a weak point of the classical 2D approach concerns the constraint handling. One of the critical constraints is the visibility constraint Chaumette [1998]. If some targets get out the camera field of view during the convergence, the value of the current feature can no longer be computed which leads to interruption of the control algorithm. Furthermore, mobile robots are always subject to mechanical constraints such as actuator limitations and nonholonomic constraints. The classical approach can not take into account these constraints. Model Predictive Control (MPC) strategy is thus well-adapted to address both the minimization of the image error and the constraint handling. This alternative approach, denoted Visual Predictive Control (VPC), has been applied to different control tasks in Allibert et al. [2006], Sauvée et al. [2006]. In Allibert et al. [2006], the point stabilization of a mobile robot with a VPC strategy is considered. The visual information is given by a catadioptric camera embedded in the mobile robot (eye-in-hand). In this case, the computational time is of no great interest. In

Sauvée et al. [2006], the visual servoing of a manipulator in an eye-to-hand configuration is studied. A linearizing and decoupling technique, based on the inverse dynamic model, is used. The considered model is then a linear one, thus reducing the computational time. The computational time is a real challenge for on-line applications to fast systems, for example in robotics. One way to reduce this time without loss of information about the process dynamics is to use the concept of differential flatness Fliess et al. [1995]. In Rao et al. [2003], the authors used differential flatness to perform a proportional-derivative controller in the image plane but no constraint can be handled.

In the proposed approach, we combine visual servoing task with predictive control strategy and flatness property, in order to satisfy online control requirements (constraint satisfaction and reduced computational time). The real-time controller is then applied to the mobile robot, Khepera II (from K-Team corporation). Experiment results highlight the tracking efficiency, the robustness of the approach and the capability of obstacle avoidance.

The paper is organized as follows. In section II, the principle of IBVS is briefly recalled. In section III, a visual predictive controller for trajectory tracking is proposed. In section IV, the process considered is proved to be flat and a real-time flat controller is synthesized. The last section is devoted to experiments on the Khepera mobile robot.

2. IMAGE-BASED VISUAL SERVOING

2.1 Objective

The purpose of a visual servoing principle is to control the movement of a robotic system from a current pose to a desired pose. Visual control law design depends on several parameters: the camera configuration (eye-to-hand, eye-in-hand or stereovision), the kind of camera (perspective

or catadioptric) and the control scheme. The fundamental classification of visual servoing distinguishes different approaches depending on the design of the control scheme: image-based control (2D), position-based control (3D) and a hybrid approach (2D/dt, 2D $\frac{1}{2}$). Further details about visual servoing can be found in Chaumette and Hutchinson [2006], Chaumette and Hutchinson [2007]. The principle of Image-Based Visual Servoing (IBVS) is to minimize an image error between the desired image and the current image from the camera. The control scheme considered is described in the figure below (Fig. 1). It enables a reference trajectory to be tracked in the image plane by a mobile robot.

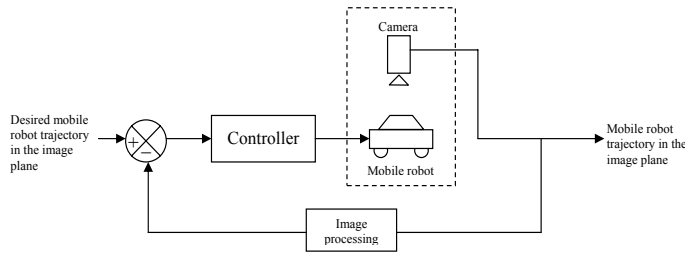


Fig. 1. IBVS scheme

In the classical IBVS approach, a local relation, based on the interaction matrix, converts image errors into Cartesian errors. This interaction matrix depends on the visual feature considered such as $(u_p \ v_p)$ for a point, on the intrinsic camera parameters and on the depth, that is the distance following the optical axis of the considered point w.r.t the camera frame. In spite of good results in general cases, several drawbacks can be underlined:

- For the computation of the interaction matrix, an approximate value of the depth at the final desired position is generally used. This choice involves a non optimal trajectory motion between the initial and desired position, and thus, the trajectory and the visibility of the features are not controlled Chaumette [1998].
- Singularities of the interaction matrix can also appear due to the number of visual features and their configurations. In this case, the synthesis of the control law is not possible Chaumette [1998].
- Constraints can not be explicitly taken into account.

To overcome these difficulties, a controller, based on the MPC strategy has been proposed Allibert et al. [2007]. Since Model-Based Control is considered, a model of the process (robot + camera) is first necessary. This global model describes the behavior of the robotic system combined with a deported camera.

2.2 Process modeling

Mobile robot

The nonholonomic mobile robot considered is a unicycle moving on a plane. It is assumed that the vehicle moves without slipping, i.e. there is a pure rolling contact between the wheels and the ground. Under these usual assumptions, the kinematic model of the mobile robot is:

$$\begin{cases} \dot{x}_r = v \cos \theta_r \\ \dot{y}_r = v \sin \theta_r \\ \dot{\theta}_r = w \end{cases} \quad (1)$$

where v and w are respectively the linear and angular velocities. The robot posture, that is this position and this orientation, linked to the middle of the axle of the driving wheels (Fig. 2), is defined by the vector $x_r = [x_r \ y_r \ \theta_r]^T$ in the robot frame (R_r).

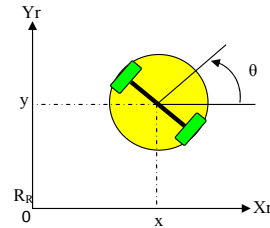


Fig. 2. Posture definition

Perspective camera modeling

Consider P , a 3D point and its coordinates $x_c = [x_c \ y_c \ z_c]^T$ expressed in the camera frame R_c (Fig. 3). It can be easily shown that the perspective transformation between the 3D point and the corresponding 2D point in the image plane $p = [u_p \ v_p]^T$ is obtained by:

$$\begin{pmatrix} u_p \\ v_p \end{pmatrix} = \begin{pmatrix} \alpha_u \frac{x_c}{z_c} + u_0 \\ \alpha_v \frac{y_c}{z_c} + v_0 \end{pmatrix} = h(x_c) \quad (2)$$

where α_u , α_v , u_0 , v_0 are constants and correspond to intrinsic parameters of the camera model.

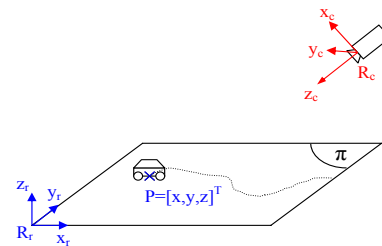


Fig. 3. Trajectory tracking

Global model

Let us assume that the transformation between the robot frame (R_r) and the camera frame (R_c) is known. The robot posture expressed in the camera frame is then given by:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_r \\ y_r \\ z_r \\ 1 \end{pmatrix} = g(x_r) \quad (3)$$

The global model, that is the robot plus the camera, is

obtained by combining (1) with (2) and (3) :

$$\begin{cases} \dot{x}_r = f(x_r, u) \\ y_m = h \circ g(x_r) \end{cases} \quad (4)$$

where

$$f(x_r, u) = \begin{pmatrix} v \cos \theta_r \\ v \sin \theta_r \\ w \end{pmatrix} \quad (5)$$

- $u = [v \ w]^T$ is the control input of the mobile robot;
- $x_r = [x_r \ y_r \ \theta_r]^T$ the state vector, that is the robot posture expressed in the robot frame;
- $y_m = [u_p \ v_p]^T$ is the output vector, that is the projection, in the image plane of the the middle of the axle of the driving wheels.

3. IMAGE-BASED TRAJECTORY TRACKING

The control objective is the tracking of a trajectory defined in the image plane under constraints. The extension of MPC to visual servoing is denoted Visual Predictive Control (VPC). The VPC principle is to determine a sequence of Nc future controls, that is robot commands, which minimizes, over a finite prediction horizon Np , the difference between the reference features and the measured process features.

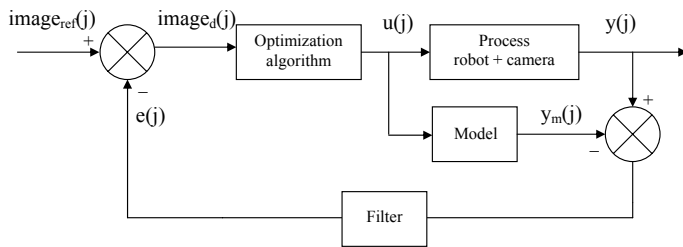


Fig. 4. Internal Model Control Structure

Due to the IMC structure used, the reference trajectory tracking by the process is equivalent to tracking a desired trajectory by the model. The control law is computed in discrete-time and the global model is then discretized. The mathematical formulation of the image-based trajectory tracking is written as:

$$\min_{\tilde{u}} J_{vs} = \sum_{j=k+1}^{k+Np} error(j)^T Q error(j) \quad (6)$$

where

- $error(j) = image_d(j) - y_m(j)$.
- $image_d(j) = [u_{dp}(j) \ v_{dp}(j)]^T$: the desired image or desired features.
- $y_m(j) = [u_p(j) \ v_p(j)]^T$: the image (or features) predicted by the nonlinear global model.
- $\tilde{u} = \{u_k, u_{k+1}, \dots, u_{k+Nc}, \dots, u_{k+Np-1}\}$ is the optimal control sequence. From $u(k + Nc + 1)$ to $u(k + Np - 1)$, the control input is constant and equal to $u(k + Nc)$.
- Q is a symmetric definitive positive matrix.
- Np : the prediction horizon; Nc : the control horizon.

Remark: The cost function (6) can be modified by adding some penalty terms :

- *a terminal constraint on the state to ensure a global asymptotic stability;*

- *a quadratic penalty term ($u^T R u$) on the control to guarantee the smoothness of the control input.*

The advantage of the proposed approach is to be able to easily handle different kinds of constraints such as actuator limitations and visibility constraints, which is impossible with a classical IBVS approach.

Two kinds of constraints can be taken into consideration:
 - mechanical constraints such as actuator limitations in amplitude or velocity:

$$\begin{aligned} u_{\min} &\leq u_j \leq u_{\max} \\ \Delta u_{\min} &\leq u_j - u_{j-1} \leq \Delta u_{\max} \end{aligned} \quad (7)$$

- visibility constraints such as image limitations which ensure that the visual features stay in the image plane. It can also represent a forbidden area in the image describing an obstacle in the robot work space for example.

$$\begin{bmatrix} u_p \\ v_p \end{bmatrix}_{\min} \leq \begin{bmatrix} u_p(j) \\ v_p(j) \end{bmatrix} \leq \begin{bmatrix} u_p \\ v_p \end{bmatrix}_{\max} \quad (8)$$

The constrained optimization problem (6) subject to (7) and (8) has to be solved at each sampling period which is very time consuming. In order to reduce the computational time, the flatness property of the nonlinear global model is used.

4. REAL-TIME REQUIREMENT

A great challenge of MPC approaches is the computational time for real applications. One way to reduce it is to use the differential flatness property.

4.1 Flatness property

The idea of differential flatness was first introduced by Fliess et al. [1995]. A nonlinear system

$$\begin{cases} \dot{x} = f(x, u) & x \in \mathbb{R}^n, u \in \mathbb{R}^m \\ y = h(x) & y \in \mathbb{R}^m \end{cases} \quad (9)$$

is differentially flat if we can find, $z \in \mathbb{R}^m$, called flat output, of the form:

$$\begin{aligned} x &= \psi(z, \dot{z}, \dots, z^r) \\ u &= \varphi(z, \dot{z}, \dots, z^r) \end{aligned} \quad (10)$$

The flat output and its derivatives provide an alternate representation of the system dynamics.

It can be shown that the global model (4) is a flat system where the flat output is given by the vector y_m .

Remark: only a sketch of proof is given because of the lack of space. The symbolic calculus is computed with Maple software.

Since the mobile robot moves on the plane π (Fig. 3), the coordinate z_r of the mobile robot is constant and can be considered equal to zero. Under this consideration, the transformation between the camera frame (R_c) and the robot frame (R_r) can be written as:

$$\begin{cases} x_c = R_{11}x_r + R_{12}y_r + T_x; \\ y_c = R_{21}x_r + R_{22}y_r + T_y; \\ z_c = R_{31}x_r + R_{32}y_r + T_z; \end{cases} \quad (11)$$

and

$$\begin{cases} u_p = \alpha_u \frac{R_{11}x_r + R_{12}y_r + T_x}{R_{31}x_r + R_{32}y_r + T_z} + u_0 \\ v_p = \alpha_v \frac{R_{21}x_r + R_{22}y_r + T_y}{R_{31}x_r + R_{32}y_r + T_z} + v_0 \end{cases} \quad (12)$$

From (12), we can write :

$$\begin{cases} x_r = \psi_1(u_p, v_p, C_1) \\ y_r = \psi_2(u_p, v_p, C_2) \end{cases} \quad (13)$$

where the constants C_i depend on several parameters:

$$C_i = \varphi_i(u_0, v_0, \alpha_u, \alpha_v, R_{11}, R_{12}, R_{21}, R_{22}, R_{31}, R_{32}, T_x, T_y, T_z) \quad (14)$$

Differentiating the previous equation (13) with respect to time and using (1), we can write:

$$\theta_r = \text{atan2}\left(\frac{\dot{y}_r}{\dot{x}_r}\right) = \psi_3(\dot{u}_p, \dot{v}_p, u_p, v_p, C_3) \quad (15)$$

All the system states can be expressed in terms of the flat output and its derivatives (eq. 13 and 15).

For the input vector, squaring and adding the expression of \dot{x}_r and \dot{y}_r (eq. 1) and substituting with the derivative of eq. (13) in respect to the time, we obtain:

$$v = \sqrt{\dot{x}_r^2 + \dot{y}_r^2} = \psi_4(\dot{u}_p, \dot{v}_p, u_p, v_p, C_4) \quad (16)$$

Similarly, differentiating eq. (15) with respect to time, we obtain the second control input:

$$w = \frac{\dot{x}_r\ddot{y}_r - \dot{y}_r\ddot{x}_r}{\dot{x}_r^2 + \dot{y}_r^2} = \psi_5(\ddot{u}_p, \ddot{v}_p, \dot{u}_p, \dot{v}_p, u_p, v_p, C_5) \quad (17)$$

All the system states (eq. 13, 15) and the control inputs (eq. 16, 17) can be expressed in terms of the flat output and its derivatives.

4.2 Prediction model

A consequence of the latter result is that the nonlinear model is equivalent to a linear Brunovsky canonical form model with dynamic feedback and coordinate change Nieuwstadt et al. [1995].

Differentiating the output vector y_m (eq. 4) twice with respect to time gives:

$$\begin{cases} \ddot{u}_p = \psi_6(\dot{v}, w, C_6) = v1 \\ \ddot{v}_p = \psi_7(\dot{v}, w, C_7) = v2 \end{cases} \quad (18)$$

where the vector $V = [v1 \ v2]^T$ is the new control input in the flat space. The previous relation (18) can be written in the following linear Brunovsky canonical form:

$$\dot{Z} = AZ + BV \quad (19)$$

where

$$Z = \begin{pmatrix} u_p \\ \dot{u}_p \\ v_p \\ \dot{v}_p \end{pmatrix} A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (20)$$

Instead of using the nonlinear model for the prediction over Np which is time-consuming, the linear Brunovsky model ensures the prediction very quickly.

The optimization problem is solved with regard to $V = [v1 \ v2]^T$ the control inputs in the flat space. The constraints on $u = [v \ w]^T$, the original input of the nonlinear system (4), become very nonlinear. On the one hand, the prediction is simplified by using (19) and on the other hand, the constrained optimization problem become more complex since the constraints are strongly nonlinear. Once the inputs $V = [v1 \ v2]^T$ in the flat space have been computed, the original inputs of the mobile robot $u = [v \ w]^T$ are obtained by inverting (18). The diffeomorphism always exists Fließ et al. [1995].

Different simulations such as different initial postures, modeling errors and tracking of different trajectories have been performed to compare the computational time required in both cases :

- NL-MPC case with the controller based on the nonlinear model described by eq. (4);
- Flat-MPC case with the controller based on the equivalent linear model described by eq. (19).

In all cases, the control is applied to the original nonlinear model (1). The results are summarized in table 1. The average of the computational time in milliseconds (ms), required at each iteration to solve the constrained nonlinear optimization problem and a criterium error (in pixels $\sqrt{(\sum error^2)}$, ps) are given.

Comparison	NL-MPC	Flat-MPC
Simulation 1	49.5ps / 27ms	45.7ps / 14ms
Simulation 2	27.9ps / 23ms	33.3ps / 11ms
Simulation 3	46.5ps / 28ms	44.5ps / 17ms

We can see that the accuracy of the trajectory tracking is equivalent for both cases (NL-MPC and Flat-MPC). However, in spite of highly nonlinear control constraints in the inputs (v_1, v_2), the proposed approach, Flat-MPC, is twice as fast as NL-MPC. These encouraging results, obtained in Allibert et al. [2007], motivate a real time application.

In order to reduce the computational time even more, a constraint transformation method has been tested on the control inputs of the mobile robot. The $u_{min} \leq u \leq u_{max}$ kind of constraints is well-adapted to this method. The transformation considered is given by:

$$\phi = \frac{1}{2}(u_{max} + u_{min}) + \frac{1}{2}(u_{max} - u_{min}) \tanh u \quad (21)$$

The constrained optimization problem becomes an unconstrained optimization problem. Comparisons have been performed in the both cases:

- the constrained optimization problem was solved with the *fmincon* function of the software Matlab;
- the unconstrained optimization problem was solved with the *fminunc* function;

Due to the high nonlinearity of the constraints, the gain of computational time is unfortunately very low. For these reasons, in the following experiments, the optimization problem is solved under the nonlinear constraints with a flat VPC strategy.

5. EXPERIMENTS

The proposed flat visual predictive controller is implemented on a Khepera robot (Fig. 5), product of K-team corporation. This mobile robot has two control inputs, v and w , expressed in pulses per second (1 pulse = 8mm/s). Limitations on actuators are respectively 50 cm/s and 4 rad/s for v and w .

The communication Robot-Pc is ensured by a RS232 communication via the standard serial port. The intrinsic parameters of the perspective camera are roughly identified. The software used is Matlab 7.0 on a Pentium IV, 3 GHz. The predictive horizon is chosen as $N_p = 10$, the control horizon $N_c = 1$. The sampling time T_e is equal to 100ms.

Remarks:

- The choice of $N_p=10$ is a compromise between the tracking efficiency and the computational time requirement.

- Concerning the choice of $N_c = 1$, simulations with $N_c > 1$ lead to a very long computational time for a tracking performance equivalent to $N_c = 1$.

- The symmetric definitive positive matrix Q is equal to the identity matrix. This choice allows to give the same weight to each output, u_p and v_p .

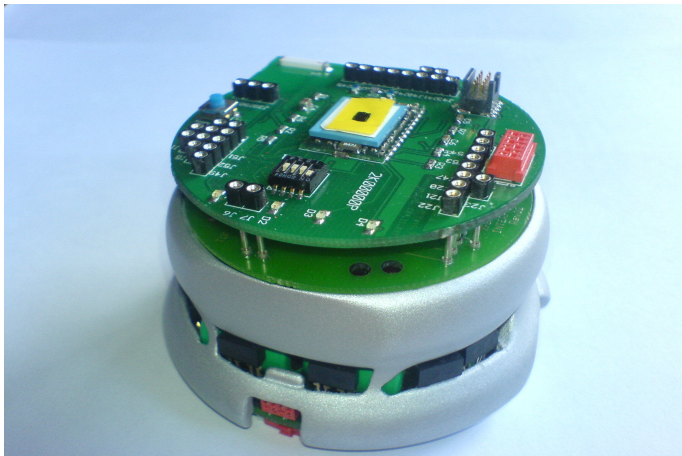


Fig. 5. Khepera robot

Experiment 1: Figure 6 shows the efficient trajectory tracking of the mobile robot in the image plane from an arbitrary initial posture. The robot trajectory is given in blue and the reference in red. The tracking is very satisfactory : errors in the image are lower than five pixels (Figure 7). Furthermore, as shown in Figure 8, the control inputs always satisfy the constraints.

Experiment 2: The robustness w.r.t image processing and modeling errors (20% on the intrinsic camera parameters ($\alpha_u, \alpha_v, u_0, v_0$)) are tested in Figure 9 for another reference trajectory. Due to the IMC structure, the tracking of the reference (in red) by the mobile robot (in blue) is still satisfactory. We can remark that the model trajectory (in green) does not track the reference which is the standard behavior of IMC.

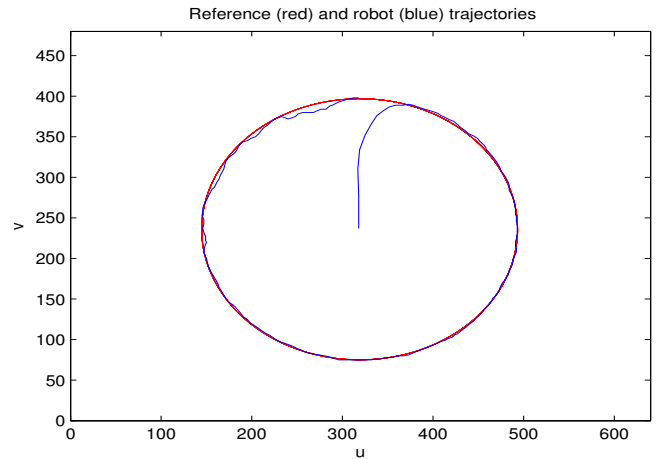


Fig. 6. Trajectory tracking

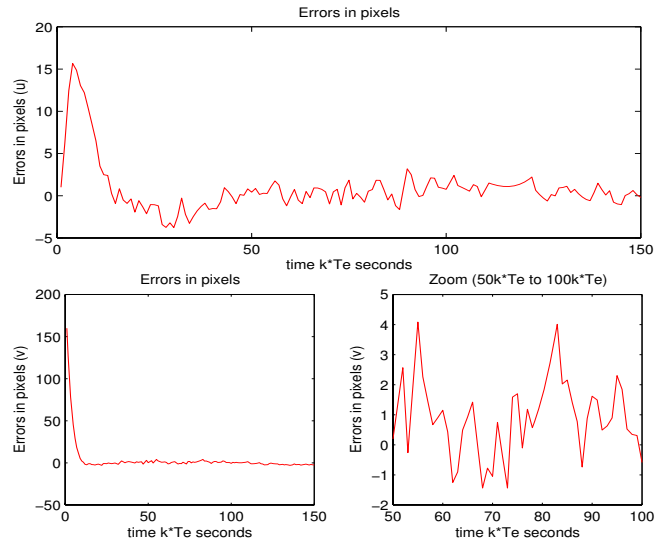


Fig. 7. Errors in pixels in the image plane

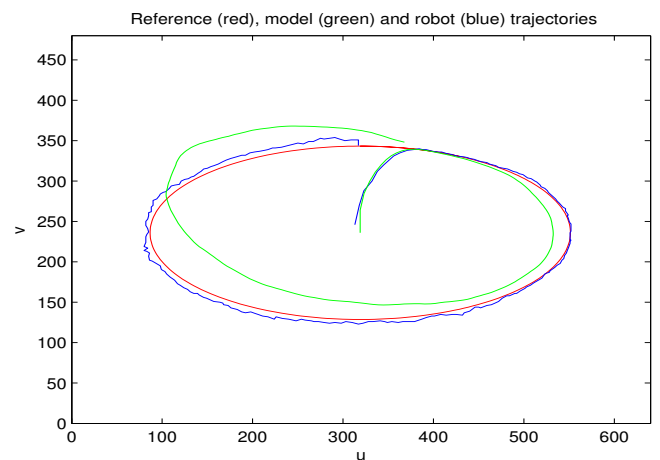


Fig. 8. Trajectory tracking with modeling errors

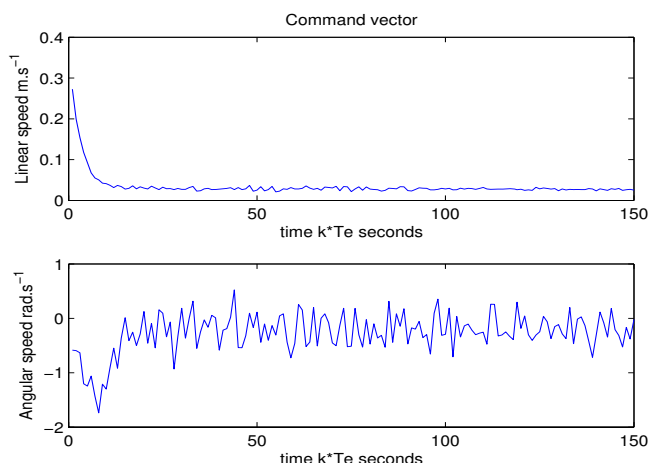


Fig. 9. Control inputs

Experiment 3: To illustrate the capability of handling visibility constraints, we define a forbidden area in the image, converted into visibility constraints and describing an obstacle in the robot workspace. The reference trajectory to be tracked is hidden by this obstacle. As shown in Figure 10, the mobile robot remains as close as possible to the reference while respecting visibility constraints. Furthermore, the control inputs always satisfy the mechanical constraints (Fig. 11).

All Video clips of these experiments can be requested by e-mail to guillaume.allibert@univ-orleans.fr.

6. CONCLUSION

Experiments on the Khepera mobile robot highlight both the efficiency and the real-time applicability of the proposed approach. The flat visual predictive controller can deal with the online requirement and the trajectory tracking under constraints in the image plane. The great challenge of computational time for real applications is taken up by the use of differential flatness. The mechanical and visibility constraints are easily taken into account in the optimization problem. The capabilities of this controller for obstacle avoidance are also pointed out. This latter result is of great interest for mobile robot navigation in hostile environments.

REFERENCES

G. Allibert, Courtial E., and Y. Touré. Visual predictive control. Grenoble, France, October 2006. IFAC Workshop on Nonlinear Predictive Control for Fast Systems.
 G. Allibert, Courtial E., and Y. Touré. A flat model predictive controller for trajectory tracking in image based visual servoing. Pretoria, South Africa, August 2007. 7th IFAC Symposium on Nonlinear Control Systems, NOLCOS'07.
 F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and A.S. Morse, editors, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag, 1998.
 F. Chaumette and S. Hutchinson. Visual servo control, part i: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, December 2006.

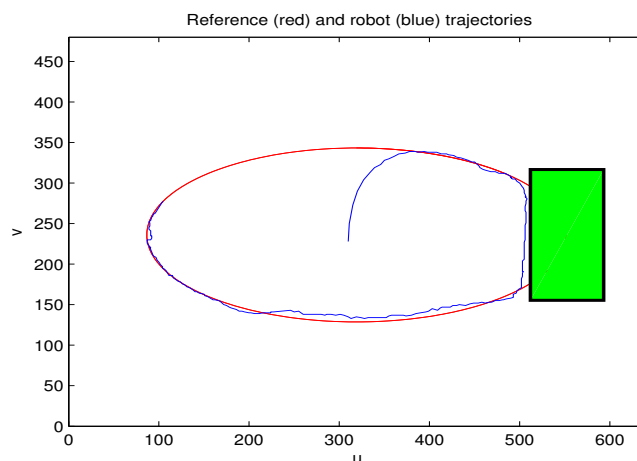


Fig. 10. Trajectory tracking with visibility constraints

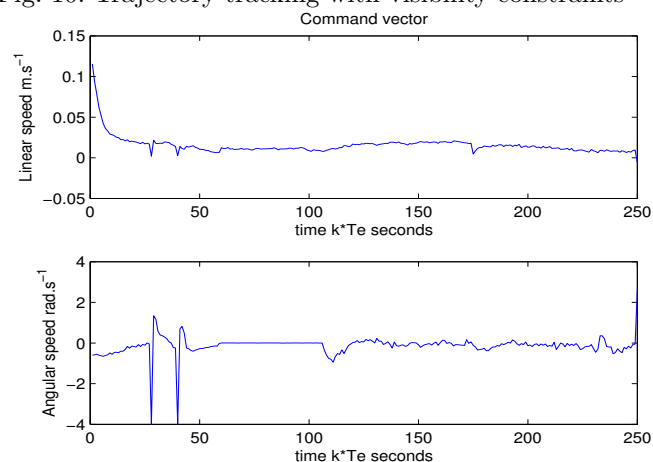


Fig. 11. Control inputs

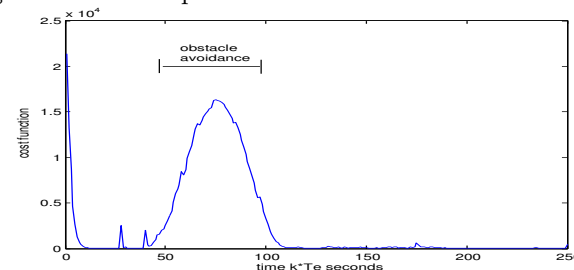


Fig. 12. Cost function

F. Chaumette and S. Hutchinson. Visual servo control, part ii: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118, March 2007.
 M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of nonlinear systems. *Int. J. Control*, 61: 1327–1361, 1995.
 M. Van Nieuwstadt, M. Rathinam, and R.M. Murray. Differential flatness and absolute equivalence of nonlinear control systems. Technical Report 94-006, CIT/CDS, 1995.
 R. Rao, V. Kumar, and C. Taylor. Visual servoing of ugv from a uav using differential flatness. IEEE Workshop on Intelligent Robots and Systems, 2003.
 M. Sauvé, P. Poignet, E. Dombre, and E. Courtial. Robust image based visual servoing through nonlinear model predictive control. San Diego, USA, December 2006. 45th IEEE CDC.