



HAL
open science

A Flat Model Predictive Controller For Trajectory Tracking In Image Based Visual Servoing

Guillaume Allibert, Estelle Courtial, Youssef Touré

► **To cite this version:**

Guillaume Allibert, Estelle Courtial, Youssef Touré. A Flat Model Predictive Controller For Trajectory Tracking In Image Based Visual Servoing. IFAC Symposium on Nonlinear Control Systems, Aug 2007, Pretoria, South Africa. pp.993-998, 10.3182/20070822-3-ZA-2920.00165 . hal-01994137

HAL Id: hal-01994137

<https://hal.science/hal-01994137>

Submitted on 25 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A FLAT MODEL PREDICTIVE CONTROLLER FOR TRAJECTORY TRACKING IN IMAGE BASED VISUAL SERVOING

Guillaume Allibert Estelle Courtial
Youssef 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: Image-Based Visual Servoing (IBVS) is a control strategy using visual information to control the motion of robotic systems. Classical IBVS can not take into account either the mechanical constraints (joint and actuator limitations) or the visibility constraints, very important in this scheme. Model Predictive Control (MPC) is well adapted to deal with these drawbacks. However, applied to fast systems (e.g. mobile robots), the computational time is a great challenge for real time applications. One way to reduce this time is to use the concept of differential flatness. In this paper, a new IBVS strategy based on a flat MPC approach is proposed. The capabilities of this approach in terms of trajectory tracking and obstacle avoidance are pointed out. Applied to mobile robot trajectory tracking, a simulation experiment shows the efficiency and the robustness of this new control scheme. The computational time required by the proposed solution is compared with the nonlinear solution and easily enables a real-time application.

Keywords: visual servoing, model predictive control, visibility constraints, flatness.

1. INTRODUCTION

Visual servoing has become an attractive strategy for the motion control of autonomous manipulators and mobile robots. 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 [Hutchinson et al. (1996)].

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. In the classical IBVS approach, an interaction matrix, also named image jacobian, converts image errors into Cartesian errors. This matrix depends on the visual feature considered (generally the pixel coordinates $(u_p v_p)$), on the intrinsic camera parameters and on the depth, i.e. the distance of the considered point w.r.t the camera frame. The control design usually requires the inversion of the interaction matrix. An exponential control law is then applied, ensuring a fast decay of the image error.

While the main interest of IBVS is its robustness to modeling errors (camera calibration for example), several drawbacks should be mentioned [Chaumette (1998)]:

- to simplify the determination of the interaction

matrix, an approximate value of the depth at the convergence is usually used. This choice involves a non optimal trajectory motion between the initial position and the desired one : the trajectory and the visibility of the features are not controlled.

- locally, due to the number of the visual features, the interaction matrix may not be invertible and the synthesis of the control law is then not possible.

- classical IBVS can not take constraints into account.

In the visual servoing concept, Model Predictive Control (MPC) [Qin and Badgwell (2003)] seems to be well-adapted to deal with these drawbacks. It is not necessary to determine the interaction matrix and its inverse. The constraints, such as actuator limitations and visibility constraints, can be easily taken into account. This alternative approach to IBVS has already been applied for point to point stabilization tasks in [Allibert et al. (2006)] and [Sauvée et al. (2006)]. In [Allibert et al. (2006)], the point stabilization of a mobile robot with a Visual Predictive Control (VPC) strategy is considered. The visual information is given by a catadioptric camera embedded on the mobile robot (eye-in-hand). A real time application highlights the efficiency and the robustness of the VPC approach. In [Sauvée et al. (2006)], the visual servoing of a manipulator in an eye-to-hand configuration is studied. The model considered is a linear one (feedback linearization), thus reducing the computational time.

Up to now, only stabilization tasks have been addressed with a predictive strategy. We propose in this paper to extend the NMPC approach to trajectory tracking in the image plane. Many papers in the literature deal with trajectory tracking of a mobile robot [Kühne et al. (2004)], [Gu and Hu (2006)]. The computational time is a real challenge for on-line application on fast systems. One way to reduce this time 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. No constraint can be handled.

In the proposed approach, we address both: the trajectory tracking directly in the image plane by the MPC strategy (with the constraint handling), and the real-time ability by using differential flatness.

The paper is organized as follows. In section II, the principle of IBVS is briefly recalled. In section III, the concept of flatness and the process modeling is presented. The proposed strategy is developed in section V. A simulation experiment, in section VI, shows the efficiency of the approach.

2. IMAGE-BASED VISUAL SERVOING

The principle of IBVS is to control the motion of robotic systems in the image plane. The task is then to minimize an image error between the desired image and the current image from the camera. The control scheme considered in this paper (figure 1), enables a reference trajectory to be tracked in the image plane by a mobile robot.

The perspective camera, in an eye-to-hand con-

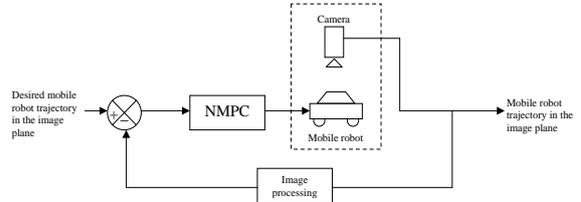


Fig. 1. Control scheme

figuration, looks at the scene. The posture of the mobile robot is linked to the posture of its gravity center (figure 2).

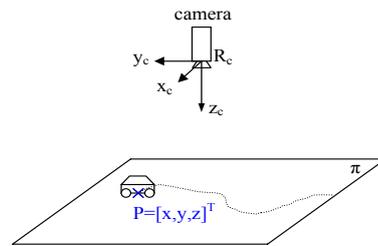


Fig. 2. Trajectory tracking

3. DIFFERENTIALLY FLAT SYSTEMS

3.1 The flatness property

The idea of differential flatness was first introduced by [Fliess et al. (1995)]. A system is said to be differentially flat if there exists a set of independent variables referred to as flat output such that every other system variable (including the input variables) is a function of the flat output and a finite number of its successive time derivatives.

A nonlinear system

$$\begin{aligned} \dot{x} &= f(x, u) & x \in \mathbb{R}^n, u \in \mathbb{R}^m \\ y &= h(x) & y \in \mathbb{R}^m \end{aligned} \quad (1)$$

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

$$\begin{aligned} x &= \psi(z, \dot{z}, \dots, z^\alpha) \\ u &= \varphi(z, \dot{z}, \dots, z^\alpha) \end{aligned} \quad (2)$$

The flat output and its derivatives provide an alternate representation of the system dynamics. One of the properties of the flat system is that the flat output is known as a function of time: all the system states and the corresponding inputs can

be obtained by equation (2).

Another consequence of the previous result is that the system (1) can be transformed into a linear system (Brunovsky normal form) by dynamic feedback and coordinate change [Nieuwstadt et al. (1995)].

3.2 Process modeling

3.2.1. Mobile robot

The nonholonomic mobile robot considered in this paper 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. The nonholonomic model is

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = w \end{cases} \quad (3)$$

or, in a compact form as

$$\dot{x} = f(x, u) \quad (4)$$

where $u = [v \ w]^T$ is the control inputs, respectively the linear and the angular velocity. The robot posture (position and orientation) is defined by the vector $x = [x \ y \ \theta]^T$ (figure 3).

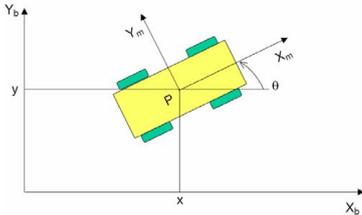


Fig. 3. Posture definition

3.2.2. Perspective camera modeling

Let us consider P , a 3D point located on the positive side of the z_c axis and its coordinates $[x \ y \ z]^T$ expressed in the camera frame Rc (figure 2). It can be easily shown that the transformation between the 3D point and this correspondent in the image plane $p = [u_p \ v_p]^T$ is obtained by :

$$\begin{cases} u_p = \alpha_u \cdot \frac{x}{z} + u_0; \\ v_p = \alpha_v \cdot \frac{y}{z} + v_0; \end{cases} \quad (5)$$

where $\alpha_u, \alpha_v, u_0, v_0$ are constants and correspond to intrinsic parameters of the camera model. Since the mobile robot moves on a plane π , the depth z , i.e. the distance along the optical axis z_c , is constant (figure 2). Under this consideration,

the previous model of the camera can be rewritten in the following linear form:

$$p = \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} = \begin{pmatrix} \lambda_u & 0 & u_0 \\ 0 & \lambda_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (6)$$

$\lambda_u = \frac{\alpha_u}{z}$ and $\lambda_v = \frac{\alpha_v}{z}$ are constants.

3.2.3. Flat global model

Combining (4) and (6), the global nonlinear model (robot + camera) is given by:

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x) \end{cases} \quad (7)$$

with

$$f(x, u) = \begin{pmatrix} v \cos \theta \\ v \sin \theta \\ w \end{pmatrix} \quad (8)$$

$$h(x) = \begin{pmatrix} \lambda_u \cdot x + u_0 \\ \lambda_v \cdot y + v_0 \end{pmatrix} \quad (9)$$

where

- $u = [v \ w]^T$ is the control input of the mobile robot;
- $x = [x \ y \ \theta]^T$ the state of the model, i.e. the robot posture;
- $y = [u_p \ v_p]^T$ is the output, i.e. the projection, in the image plane of the gravity center of the mobile robot.

It can be shown that (7) is a flat system where the flat output is given by the vector y .

Proof: Inverting (6) gives

$$\begin{cases} x = \frac{u_p - u_0}{\lambda_u} \\ y = \frac{v_p - v_0}{\lambda_v} \end{cases} \quad (10)$$

Differentiating the previous equation (10) with respect to time, and using (3), we can write:

$$\theta = \arctan\left(\frac{\dot{v}_p \lambda_u}{\dot{u}_p \lambda_v}\right) \quad (11)$$

All the system states can be expressed in terms of the flat output and its derivatives (eq. 10 and 11).

For the input vector, squaring and adding the expression of \dot{x} and \dot{y} (eq. 3) and substituting with the derivative of eq. (10) in respect to the time, we obtain:

$$v = \sqrt{\frac{\dot{u}_p^2}{\lambda_u^2} + \frac{\dot{v}_p^2}{\lambda_v^2}} \quad (12)$$

Similarly, differentiating eq. (11) with respect to time, we obtain the second control input:

$$w = \frac{\ddot{v}_p \lambda_u \dot{u}_p \lambda_v - \ddot{u}_p \lambda_v \dot{v}_p \lambda_u}{\dot{u}_p^2 \lambda_v^2 + \dot{v}_p^2 \lambda_u^2} \quad (13)$$

All the system states (eq. 10, 11) and the control inputs (eq. 12, 13) can be expressed in terms of the flat output and its derivatives.

As a consequence of the previous flatness result,

the flat system (7) can be transformed into the linear Brunovsky's canonical form by dynamic feedback and coordinate change.

Differentiating (6) twice with respect to time gives:

$$\begin{aligned} \ddot{u}_p &= \lambda_u \dot{v} \cos \theta - \lambda_u v \dot{\theta} \sin \theta = v1 \\ \ddot{v}_p &= \lambda_v \dot{v} \sin \theta + \lambda_v v \dot{\theta} \cos \theta = v2 \end{aligned} \quad (14)$$

where the vector $V = [v1 \ v2]^T$ is the new control input in the flat space. The previous relation (22) can be written in the following linear Brunovsky's canonical form:

$$\dot{Z} = AZ + BV \quad (15)$$

where

$$Z = \begin{pmatrix} u_p \\ \dot{u}_p \\ v_p \\ \dot{v}_p \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (16)$$

Our objective is to design a controller in the flat space, such that the mobile robot tracks a reference trajectory given by:

$$\text{image}_{\text{ref}} = \begin{pmatrix} u_p \text{ ref} \\ v_p \text{ ref} \end{pmatrix} \quad (17)$$

In the next section, we propose to use a MPC controller in the flat space, to perform the trajectory tracking.

4. FLATNESS-BASED MPC FOR VISUAL SERVOING

A Nonlinear Model Predictive Controller formulates a trajectory tracking problem into a nonlinear optimization problem. A cost function, defined by a quadratic form of the error between the desired trajectory and the predicted model, has to be minimized with respect to a control sequence \tilde{u} . The control law is computed in discrete-time and thus, it is necessary to discretize the model. Combined with the well-known Internal Model Control (IMC) structure (figure 5), the NMPC problem is mathematically written as follows:

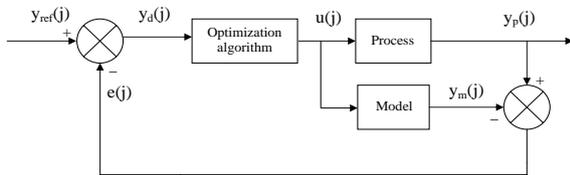


Fig. 4. IMC structure

$$\min_{\tilde{u}} J = \sum_{j=k+1}^{k+N_p} (y_d(j) - y_m(j))^T Q (y_d(j) - y_m(j)) \quad (18)$$

subject to system dynamics :

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ y_m(k) = h(x(k)) \end{cases} \quad (19)$$

Q is a positive definite symmetric matrix, N_p is the prediction horizon, $x \in X \subset \mathbb{R}^n$, $u \in U \subset \mathbb{R}^m$, $y \in Y \subset \mathbb{R}^p$, where X, U, Y are compact sets defined by the constraints respectively on the states, the inputs and the outputs.

4.1 Application to Image-Based Visual Servoing

MPC can easily address both tasks of IBVS: minimization of an image error and constraint handling.

The cost function is defined by:

$$J_{vs} = \sum_{j=k+1}^{k+N_p} \text{error}(j)^T Q \text{error}(j) \quad (20)$$

with

- $\text{error}(j) = \text{image}_d(j) - \text{image}_m(j)$
- $\text{image}_d(j) = [u_{dp}(j) \ v_{dp}(j)]^T$: desired image
- $\text{image}_m(j) = [u_p(j) \ v_p(j)]^T$: predicted image by the nonlinear global model (7).

Two kind of constraints can be taken into consideration :

- mechanical constraints such as actuator limitations in amplitude or velocity:

$$\begin{aligned} u_{\min} &\leq u_j \leq u_{\max} \\ \Delta u_{\min} &\leq u_j - u_{j-1} \leq \Delta u_{\max} \end{aligned} \quad (21)$$

- visibility constraints such as image limitations which ensure that the visual features stay in the image plane. Visibility constraints, the weak point of the classical IBVS, can easily be added to the optimization problem.

The constrained optimization problem (18) 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 considered model is then used. The predicted image is ensured by (15) instead of (7).

Once the inputs, $V = [v1 \ v2]^T$ in the flat space, have been computed by the minimization of (20), the dynamic feedback (22) is used to obtain the inputs of the original nonlinear system (7), which are applied to the real process.

$$\begin{pmatrix} \dot{v} \\ w \end{pmatrix} = \begin{pmatrix} \frac{\cos \theta}{\lambda_u} & \frac{\sin \theta}{\lambda_v} \\ -\frac{\sin \theta}{v\lambda_u} & \frac{\cos \theta}{v\lambda_v} \end{pmatrix} \begin{pmatrix} v1 \\ v2 \end{pmatrix} \quad (22)$$

5. A SIMULATION EXPERIMENT

The simulations of the control approach described in the previous section have been performed on a PC pentium IV, 3GHz under Matlab.

The predictive horizon is chosen as $N_p = 10$, the control horizon $N_c = 1$. The sampling time T_e is equal to 100ms. The optimization problem is

solved by the Matlab function *fmincon*. All the simulations have been performed in both cases :
- with the controller based on the nonlinear model described by eq. (7): NL-MPC case;
- with the controller based on the equivalent linear model described by eq. (15): Flat-MPC case.
In all cases, the control is applied to the original nonlinear model (3). For lack of space, only the simulations with the flat MPC controller are presented.

Simulation 1 : Figure 5 shows the efficient trajectory tracking of the mobile robot in the image plane from an arbitrary initial posture. The control inputs (figure 6), subject to limitations ($-0.5m.s^{-1} \leq v(t) \leq 0.5m.s^{-1}$, $-4rad.s^{-1} \leq w(t) \leq 4rad.s^{-1}$) verify the constraints.

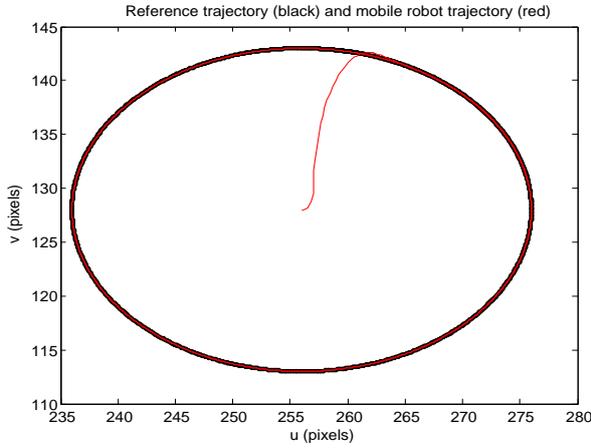


Fig. 5. Trajectory tracking

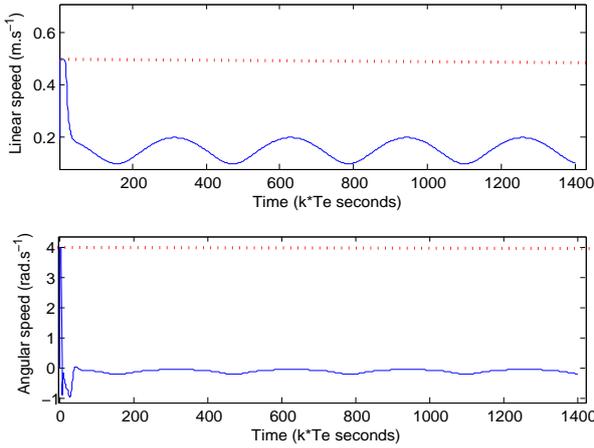


Fig. 6. Constrained control inputs

Simulation 2 : The robustness w.r.t modeling errors (20% on the intrinsic camera parameters: α_u , α_v , u_0 , v_0) and disturbances (white noise added to the output) is tested (figures 7,8). Due to the IMC structure, the tracking is still satisfactory: the pixel error is less than two pixels.

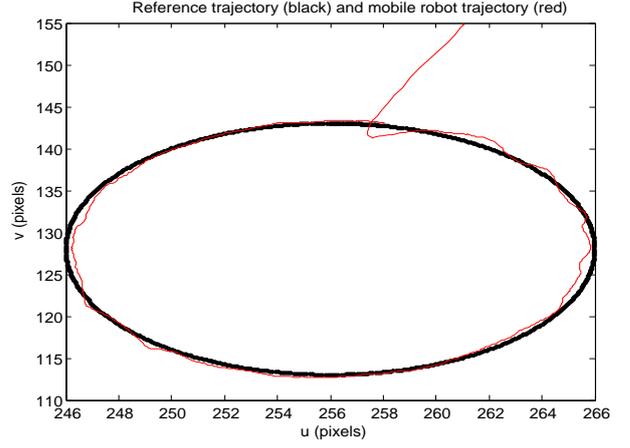


Fig. 7. trajectory tracking with disturbances and modeling errors

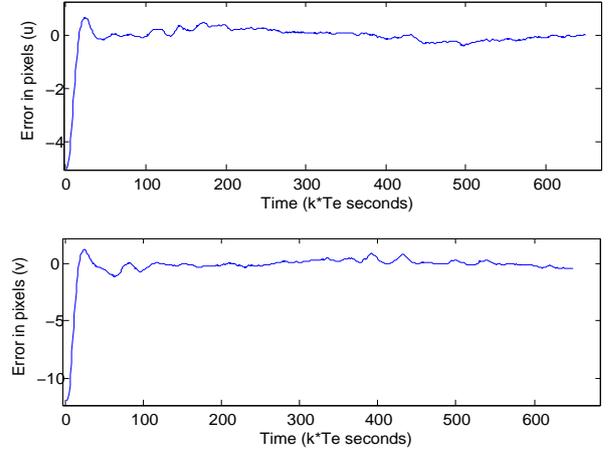


Fig. 8. Error in pixels

Simulation 3 : To illustrate the capability of handling visibility constraints (figures 9,10,11), we define a forbidden area in the image, converted in visibility constraints and describing an obstacle in the robot workspace. The mobile robot tracks for the best the reference, avoiding the obstacle. The control constraints are still satisfied.

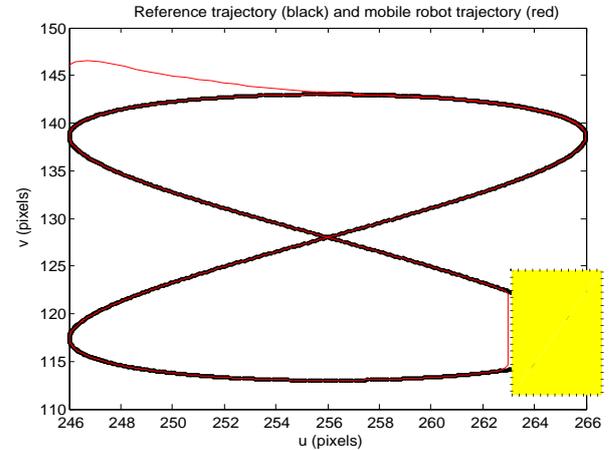


Fig. 9. Trajectory tracking in spite of an obstacle

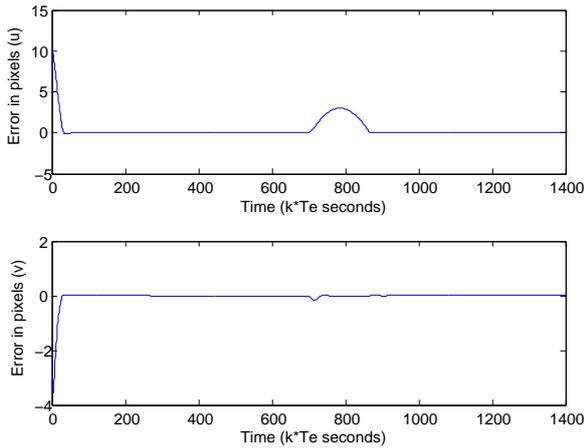


Fig. 10. Error in pixels

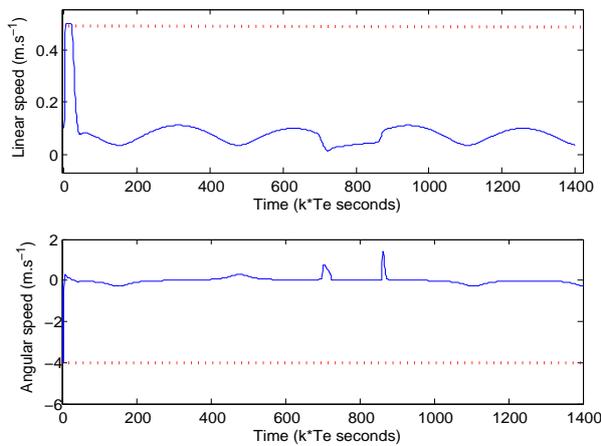


Fig. 11. Command vector

The performance comparison of both cases (NL-MPC and Flat-MPC) is summarized in table 1. For each simulation, we give:

- the average of the computational time (in milliseconds, ms) required at each iteration to solve the constrained nonlinear optimization problem;
- one criterium error (in pixels, ps) given by: $(\sqrt{(\sum error^2)})$;

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.

6. CONCLUSION

In this paper, trajectory tracking, directly in the image plane, is addressed with an MPC strategy.

The advantage of the proposed approach is to be able to handle easily different kind of constraints (actuator limitations, visibility constraints) which is impossible with a classical IBVS approach. The great challenge of computational time for real applications is taken up by the use of differential flatness. The simulation experiment then shows the efficiency of the Flat-MPC approach in terms of trajectory tracking, robustness w.r.t to modeling errors and disturbances, control input feasibility and obstacle avoidance. The computational time obtained for simulations is encouraging with a view to an on-line application. This is already under investigation on a Khepera mobile robot. The capabilities of this new approach are also very promising for collision avoidance.

REFERENCES

- G. Allibert, Courtial E., and Y. Touré. Visual predictive control. Grenoble, France, October 2006. IFAC Workshop on Nonlinear Predictive Control for Fast Systems.
- 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.
- M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of nonlinear systems. *Int. J. Control*, 61:1327–1361, 1995.
- D. Gu and H. Hu. Receding horizon tracking control of wheeled mobile robots. *IEEE Trans. on Control Systems Technology*, 14(4):743–749, 2006.
- S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.
- F. Kühne, Jr. Gomes da Silva, and W.F. Lages. Predictive control of a mobile robot using linearization. Aachen, Germany, 2004. Mechatronics and Robotics.
- 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.
- S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Eng. Prac.*, 11(7):733–764, 2003.
- 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. Sauvée, 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.