Predictive Control for Constrained Image-Based Visual Servoing

Guillaume Allibert, Estelle Courtial,
and François Chaumette

Abstract—This paper deals with the image-based visual servoing (IBVS), subject to constraints. Robot workspace limitations, visibility constraints, and actuators limitations are addressed. These constraints are formulated into state, output, and input constraints, respectively. Based on the predictive-control strategy, the IBVS task is written into a nonlinear optimization problem in the image plane, where the constraints can be easily and explicitly taken into account. Second, the contribution of the image prediction and influence of the prediction horizon are pointed out. The image prediction is obtained due to a model. The latter can be a local model based on the interaction matrix or a nonlinear global model based on 3-D data. Its choice is discussed with respect to the constraints to be handled. Finally, simulations that were obtained with a 6-degree-of-freedom (DOF) free-flying camera highlight the potential advantages of the proposed approach with respect to the image prediction and the constraint handling.

Index Terms—Predictive control, visual servoing.

I. INTRODUCTION

Visual servoing has been investigated heavily over the past decades. Its task consists of determining the control input that was applied to a robotic system so that a set of visual features, which were designed from image measurements, reaches a desired static reference or follows a desired dynamic reference. In an image-based visual servoing (IBVS), the feedback information is defined in the image plane. Mathematically, the visual servoing comes down to zero to regulate an error $e(t)$, which is expressed in the image, between the current features $s(t)$ and the reference features s^* [6]. The relationship between the camera velocity $\tau(t)$ and the time variation of the visual features $\dot{s}(t)$ is given by the interaction matrix, which is noted as L_s. An exponential decay of the error $e(t)$ is usually specified, thus leading to the classical feedback control law as follows:

$$\tau(t) = -\lambda \widehat{L}_s^+ e(t) \qquad \text{with} \qquad \lambda > 0 \qquad (1)$$

where $\widehat{L}_s^+$ is the pseudoinverse of an approximation of L_s.

The classical IBVS is very easy to implement, but its drawbacks are its possible unsatisfactory behavior [5] and the difficulty of constraint handling. The unsuitable behavior occurs for large displacements or rotations to achieve when the initial and desired locations are distant. It produces a motion that may not be allowed by the robot workspace, the joint limitations, or the visibility constraints.

In this paper, the tricky problem of constraint handling in IBVS is addressed. The 2-D constraint, which is also called the visibility constraint, has to guarantee that the image measurements stay into the camera field of view. Indeed, if the visibility of the target is no longer ensured, then the control algorithm is stopped. Three-dimensional constraints, such as workspace limits, have to make sure that the robot achieves admissible motions in its workspace all along the task.

Among the numerous works that have investigated this critical issue, three points of view can be distinguished. The first one consists of designing adequate visual features. Mahony *et al.* [17], for instance, have shown that the system behavior explicitly depends on the kind of features. Consequently, lines, spheres, and moments may be used and combined to obtain good decoupling and linearizing properties. This implicit solution may not guarantee the constraint satisfaction.

Another way to deal with the constraint handling is to combine path planning and trajectory tracking [7], [8], [16], [18], [23]. When it is successful, this solution allows to ensure both an optimal trajectory of the camera in the Cartesian space and the visibility of the features. Path planning via linear-matrix-inequality (LMI) optimization has recently been proposed in [8] to fulfill 2-D and 3-D constraints. Once the LMI optimization has found a 3-D path-planning solution, i.e., no solution may exist for the considered problem and the solution is also not unique, an image-based controller is used to track the corresponding camera trajectory.

In the third approach, the effort is done on the control-law design. The visual features considered are generally basic, namely, point-like features. Advanced control laws, such as optimal control [15], [21], LMI [10], [11], and predictive control, have been reported in the literature. The existing works in predictive control generally deal with 3-D visual servoing. In [19], the stabilization of a robot arm is treated with a predictive approach in 3-D and without constraint handling. The predictive controller is used in [13] and [14] for motion compensation in target tracking, in addition to a classical IBVS controller. In [4], a linear DMC controller is used to cope up with the process delay in vision-based control. In [22], the predictive controller is used from ultrasound images for a medical application. In [1] and [2], the IBVS objective is formulated into a nonlinear optimization problem that was expressed in the image plane. When applied to a mobile robot or a robot arm with a catadioptric camera, the predictive strategy shows its capabilities. The interest of the image prediction has been presented in [3] in the case of a local model (LM) but without constraint handling.

The contributions of this paper concern the constraint handling and the image prediction. First, even if it is not uncommon to handle constraints on the manipulated variables, it is rarely the case on state variables or output variables. Workspace limitations can be formulated into state variables, and the visibility constraints are formulated into output constraints. Due to the optimization formulation of the IBVS task, all constraints can be easily and explicitly taken into account in the control-law design. Second, the image prediction is obtained due to a model. The latter can be a nonlinear global model (GM) that combines the robot model and the camera one or an LM that relies on the interaction matrix. The choice of the model is addressed and discussed in the sequel of the paper. As we will see, the image prediction can provide better camera trajectories. The influence of the prediction horizon is also pointed out.

The paper is organized as follows. In Section II, the principle of the proposed strategy is presented; the control structure, the considered constraints, and the mathematical formulation are presented. Section III is devoted to the statement of the image-prediction model. In Section IV, simulations on a 6-degree-of-freedom (DOF) free-flying camera illustrate the capabilities of the proposed approach in terms of tracking efficiency and constraint handling. Simulation results of
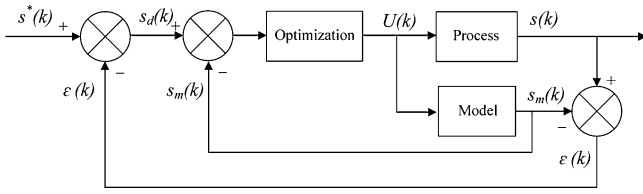
Fig. 1.   Control structure.

visual predictive control (VPC) with LM or GM are compared. The robustness with respect to modeling errors and measurement noise is tested. Finally, conclusions are given in the last section.

II. VISUAL PREDICTIVE CONTROL FOR CONSTRAINED IMAGE-BASED VISUAL SERVOING

The extension of the predictive-control strategy to visual servoing is named as VPC. As for all predictive strategies, a reference trajectory, a model of the dynamic process, a cost function, and a solving optimization method are necessary. We first introduce the control structure, and then state the mathematical formulation of VPC.

A. Control Structure

The control structure considered is based on the well-known internal model-control (IMC) structure [20] (see Fig. 1). The process block contains the robotic system and the camera. The input U is the robotic control variable, and the output s is the current value of the visual features. The reference s^* is expressed in the image plane, and it can be static or dynamic. The error signal ε represents all modeling errors and disturbances between the current features and the values that were predicted from the model of the system, where k is the current time

$$\varepsilon(k) = s(k) - s_m(k). \qquad (2)$$

The optimization algorithm minimizes the difference between the desired trajectory s_d and the predicted model output s_m. Indeed, according to Fig. (1), we can write

$$s_d(k) = s^*(k) - \varepsilon(k) = s^*(k) - (s(k) - s_m(k)) \qquad (3)$$

from which we deduce

$$s_d(k) - s_m(k) = s^*(k) - s(k). \qquad (4)$$

Consequently, to track the reference features s^* by the process output s is equivalent to the tracking of the desired features s_d by the model output s_m.

The reference s^* is supposed to be known. A model predicts the behavior of the visual features s_m over a finite-prediction horizon N_p. The choice of the model will be discussed in Section III. The difference between the desired features and the predicted model features is used to define the cost function J to be minimized with respect to a control sequence $\tilde{U}$ over N_p. Only, the first component $U(k)$ of the optimal-control sequence is really applied to the process. At the next sampling time, due to disturbances and model mismatches, the measurements are updated, the prediction horizon moves one step forward, and the procedure starts again.

B. Constraint Handling

Model-predictive-control strategies are widespread in industry due to their capability of constraint handling. Indeed, constraints can be explicitly taken into account in the control-law design. For an IBVS

task, three kinds of constraints can be distinguished, which are as follows.

1) Three-dimensional constraints can be, typically, mechanical constraints, such as workspace limits or joint saturations. These constraints can be taken into account if the geometric model of the robot is available, since the camera pose, which is described by $p(k)$, depends on the joint measurements $q(k)$

$$p_{\min} \le p(k) \le p_{\max}$$
$$q_{\min} \le q(k) \le q_{\max}. \qquad (5)$$

2) Two-dimensional constraints are also named as visibility constraints to ensure that the visual measurements stay in the image plane or to represent forbidden areas in the image. The latter can be very useful in dealing with obstacle avoidance or image occlusion

$$s_{\min} \le s_m(k) \le s_{\max}. \qquad (6)$$

3) Control constraints, such as actuator limitations in amplitude or velocity

$$U_{\min} \le U(k) \le U_{\max}. \qquad (7)$$

The constraints (5)–(7) can be written as nonlinear functions [9]

$$\begin{cases} C(U) \le 0 \\ C_{\text{eq}}(U) = 0. \end{cases} \qquad (8)$$

and define a constraint domain $\mathbf{K}$.

C. Mathematical Formulation

The mathematical formulation of a constrained VPC strategy can be written in discrete time as follows:

$$\min_{\tilde{U} \in \mathbf{K}} J(U) \qquad (9)$$

with

$$J(U) = \sum_{j=k+1}^{k+N_p} [s_d(j) - s_m(j)]^T Q(j)[s_d(j) - s_m(j)] \qquad (10)$$

and

$$\tilde{U} = \{U(k), U(k+1), \dots, U(k+N_c), \dots, U(k+N_p-1)\} \qquad (11)$$

subject to

$$s_d(j) = s^*(j) - \varepsilon(j) \qquad (12)$$

$$\begin{cases} x(j) = f(x(j-1), U(j-1)) \\ s_m(j) = h(x(j)). \end{cases} \qquad (13)$$

The variables $x \in \mathbb{R}^n$, $U \in \mathbb{R}^m$, and $s_m \in \mathbb{R}^p$ are the state, the input, and the output of the model, respectively. In Section III, we will see that the state can be differently chosen with respect to the prediction model used and the constraints to be handled. The first nonlinear equation of (13) describes the dynamics of the system, where $x(j)$ represents the predicted state at time j $\forall j \in [k+1; k+N_p]$. The weighted matrix $Q(j)$ is a symmetric definite-positive matrix.

To implement a visual predictive controller, the following steps have to be checked.

1) At the current time k, the process output $s(k)$ is measured.
2) The signal error is calculated as follows: $\varepsilon(k) = s(k) - s_m(k)$.

3) This error is assumed to be constant over the prediction horizon

$$\varepsilon(j) = \varepsilon(k) = s(k) - s_m(k) \qquad \forall j \in [k+1; k+N_p].$$
(14)

4) The desired trajectory $s_d(j)$ is computed $\forall j \in [k+1; k+N_p]$, according to (12).

5) The model output $s_m(k)$, needed for the prediction, is initialized with the measured process output $s(k)$ that guarantees the feedback of the IMC structure. Moreover, note that a second feedback is ensured by the error signal $\varepsilon(j)$ to deal with modeling errors and disturbances.

6) Finally, a constrained-optimization algorithm determines the optimal-control sequence (11). From $U(k+N_c+1)$ to $U(k+N_p-1)$, the control input is constant and is equal to $U(k+N_c)$, where N_c is the control horizon.

The VPC setting parameters are the prediction horizon (N_p), the control horizon (N_c), and the weighted matrix ($Q(j)$), which are as follows.

1) The prediction horizon is chosen in order to satisfy a compromise between stability (long horizon) and numerical feasibility, in term of computational time requirement (short horizon). If N_p tends to infinity, the control problem becomes an optimal control that was known to ensure closed-loop stability [12]. An increase in N_p also results in less-aggressive control inputs.

2) The control horizon allows to attend the objective in N_c steps, but it increases the computational effort. The control horizon is often equal to 1, which corresponds to a unique constant control over N_p.

3) The weighted matrix $Q(j)$ is either the identity matrix $I_{p \times p}$ $\forall j$, which is constant over the prediction horizon, or a time-varying matrix [3]. In this last case, the matrix weights the error at each sampling instant increasingly over the prediction horizon and, therefore, stresses the error at the end of N_p that corresponds to the final objective.

Numerous constrained-optimization routines are available in software libraries to solve this kind of problem: projected gradient methods, penalty methods, etc. In our case, a sequential quadratic program (SQP) is used (more precisely, the function *fmincon* from MATLAB optimization toolbox). A common algorithm has been deliberately chosen to show that the success of the VPC is only due to the optimization formulation in the image plane with the image prediction, and it does not depend on the optimization algorithm.

D. Particular Case of the Classical Image-Based Visual Servoing

At time k, the aim of the classical IBVS is to regulate the image error $e(k+1) = s^*(k+1) - s_m(k+1)$ to zero at time $k+1$. There is only a prediction of one step forward, which is equivalent to a prediction horizon N_p, and a control horizon N_c is equal to 1. The predicted model output $s_m(k+1)$ is computed due to an LM based on the interaction matrix (see Section III-A) that was initialized to $s_m(k) = s(k)$. The image error $\varepsilon(k)$ is consequently null. According to the control structure, $s^*(k+1) = s_d(k+1) - \varepsilon(k+1)$. Since $\varepsilon(k+1)$ is unknown, it is kept constant and equal to $\varepsilon(k)$. The standard formulation of the classical IBVS is then equivalent to the minimization of the following cost function, where the weighted matrix $Q(j)$ is the identity matrix

$$\min_{U(k)} [s_d(k+1) - s_m(k+1)]^T [s_d(k+1) - s_m(k+1)].$$
(15)

In the sequel, this particular case is noted as optimal IBVS. The classical IBVS defined by (1) and the optimal IBVS with an LM give similar

results. The only difference is the control input behavior that does not have an exponential decay in the last case. The optimization formulation has the advantage of being able to deal with constraints.

III. MODELS OF IMAGE PREDICTION

All advanced control laws are based on the knowledge of a model. In VPC approach, the model plays a crucial role, since it is used to predict the evolution of the visual features with respect to the camera velocity over a finite-prediction horizon. Its choice will impact on the tracking accuracy and on the fulfillment of the constraints. Two kinds of model can be considered to perform the image prediction: an LM based on the interaction matrix and a nonlinear GM. The model equations (13) are now detailed for both cases.

In the following sections, we consider a 6-DOF free-flying perspective camera that observes fixed-point features. A 3-D point, with coordinates $P = (X, Y, Z)$ in the camera frame, is projected in the image plane as a 2-D point with coordinates $s_m = (u, v)$. The sampling period is T_e, and the control input is the camera velocity, which is noted as $\tau = (T_x, T_y, T_z, W_x, W_y, W_z)$.

A. Local Model

The LM is based on the interaction matrix. For a point-like feature s_m expressed in normalized coordinates such that $u = X/Z$ and $v = Y/Z$, the interaction matrix, which is related to s_m, is given by [6]

$$L_s = \begin{bmatrix} -\dfrac{1}{Z} & 0 & \dfrac{u}{Z} & uv & -(1+u^2) & v \\ 0 & -\dfrac{1}{Z} & \dfrac{v}{Z} & 1+v^2 & -uv & -u \end{bmatrix}.$$
(16)

The dynamic equation that describes the visual variation with respect to the camera velocity

$$\dot{s}_m(t) = L_s(t)\tau(t)$$
(17)

is approximated with the Newton–Euler method and becomes

$$s_m(k+1) = s_m(k) + \widehat{L}_s(k)T_e\tau(k).$$
(18)

Concerning the computation of the interaction matrix $\widehat{L}_s$, two cases can be distinguished with respect to the depth parameter Z, which are as follows.

1) The depth is replaced by its value given at the reference position Z^*. Consequently, the interaction matrix varies only through the current measure of the visual features: $\widehat{L}_s(t) = \widehat{L}_{(s(t), Z^*)}$. The LM based on this interaction matrix is noted as LM_p.

2) The depth is updated at each sampling time by using 3-D reconstruction methods: $\widehat{L}_s(t) = \widehat{L}_{(s(t), Z(t))}$. The LM is noted as LM_c in this case.

The models LM_c and LM_p can, respectively, be used if the 3-D data are available or not. The LM is given by the set of equations that describe the process dynamics and the visual output. By considering the visual feature s_m as the state x, we obtain

$$\begin{cases} x(k+1) = x(k) + \widehat{L}_s(k)T_e\tau(k) = f(x(k), \tau(k)) \\ s_m(k) = x(k) = h(x(k)). \end{cases}$$
(19)

In this case, 2-D constraints (6) can be taken into account, since the model state and output are the visual features. The 3-D constraints (5) can also be considered if the robot joints are measured and if the robot geometric model is known.

B. Nonlinear Global Model

The GM is based on the knowledge of the 3-D target coordinates in the camera frame. These normalized coordinates, noted as $(X, Y, Z, 1)$ for one point, are taken as the system state x. The dynamic equation is given by

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_{k+1} = \begin{pmatrix} R(\tau) & T(\tau) \\ 0_{1 \times 3} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_k \tag{20}$$

where R and T are the matrix of rotation and the vector of translations, respectively, which depend on the control input τ. Equation (20) can be summarized by

$$x(k+1) = f(x(k), \tau(k)). \tag{21}$$

The GM output is the visual features s_m that were expressed in the image plane. In the case of a perspective camera, the output equation for one point-like feature in normalized coordinates can be written as follows:

$$s_m(k) = \begin{pmatrix} u(k) \\ v(k) \end{pmatrix} = \begin{pmatrix} X(k)/Z(k) \\ Y(k)/Z(k) \end{pmatrix} = h(x(k)). \tag{22}$$

The nonlinear GM is then completely defined by (21) and (22). For the state initialization at time k, two cases can be distinguished.

1) The depth $Z(k)$ is approximated by Z^*. Consequently, the initial state is taken, which is equal to

$$\begin{cases} X(k) = u(k)\, Z^* \\ Y(k) = v(k)\, Z^* \\ Z(k) = Z^*. \end{cases} \tag{23}$$

2) The depth $Z(k)$ is estimated by $\hat{Z}(k)$ at each sampling time. Consequently, the initial state is taken, which is equal to

$$\begin{cases} X(k) = u(k)\, \hat{Z}(k) \\ Y(k) = v(k)\, \hat{Z}(k) \\ Z(k) = \hat{Z}(k). \end{cases} \tag{24}$$

In the same spirit of the LM, the GM is noted as GM_p (23) or GM_c (24), respectively, if the depth is approximated or estimated. These two GMs can be used, according to the 3-D data available. The nonlinear GM is appropriate to deal with 2-D and/or 3-D constraints. Even if the depth is coarsely approximated by Z^*, the nonlinear GM has a large validity domain, and thus, it can be used for large displacements, as shown in Section IV.

IV. SIMULATIONS

For all the presented simulations, the sampling period T_e is equal to 40 ms, which corresponds to an usual camera (25 frames per second). The control task consists of positioning a perspective free-flying camera with respect to a target, which is composed of four points. These four points form a square of 20 cm in length in the Cartesian space. The reference image is obtained when the target pose, expressed in the camera frame (R_C), is equal to $P_{T/C} = (0, 0, 0.5, 0, 0, 0)$. The first three components are the translation that are expressed in meters, and the last three components are the roll, pitch, and yaw angles that are expressed in radians. The coordinates of the four points in
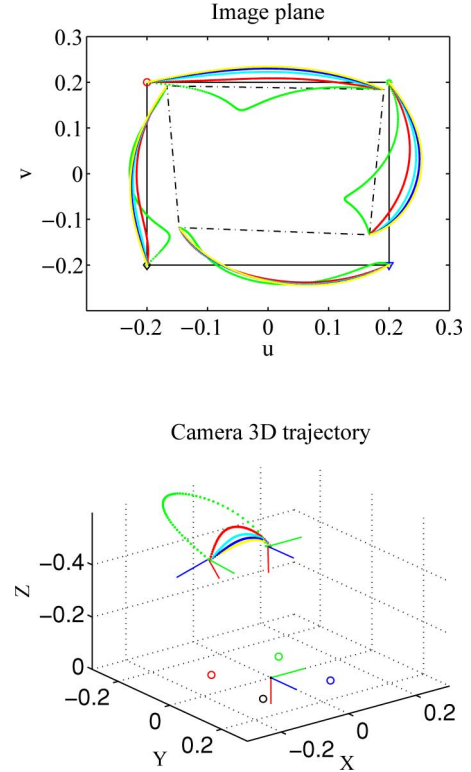

Fig. 2. $\mathrm{VPC}_{\mathrm{LM}_p}$ with (green) $N_p = 1$, (red) $N_p = 3$, (blue) $N_p = 5$, (black) $N_p = 7$, and (yellow) $N_p = 9$.

the reference image are $s^* = (u_{d1}, v_{d1}, u_{d2}, v_{d2}, u_{d3}, v_{d3}, u_{d4}, v_{d4}) = (-0.2, 0.2, 0.2, 0.2, 0.2, -0.2, -0.2, -0.2)$.

In all the presented simulations, the control input is limited to the following bounds: 25 cm/s for the translation speed and 0.25 rad/s for the rotation speed. The weighted matrix $Q(j)$ is taken to the identity matrix $I_{p \times p}$.
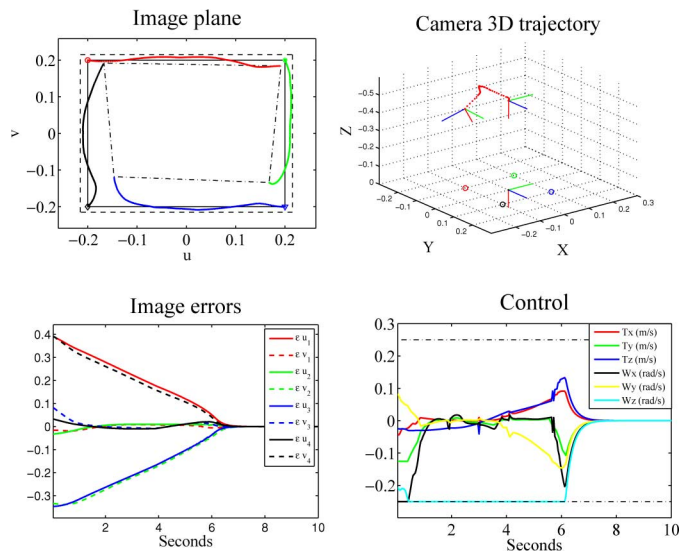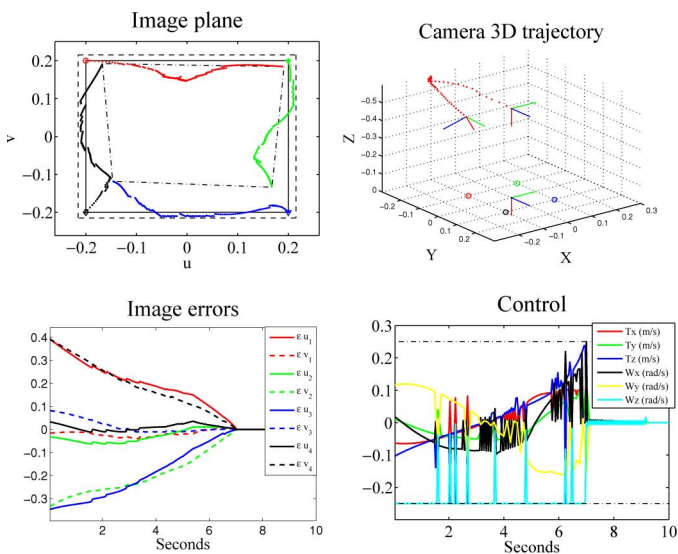
The simulation results illustrate
1) the influence of the image prediction;
2) the robustness with respect to modeling errors, measurement noises, and control disturbances;
3) the constraint handling.

The classical IBVS, the optimal IBVS, and the VPC are first compared without 3-D knowledge (based on LM_p or GM_p with Z^*), and then with 3-D knowledge (based on LM_c or GM_c with $Z(t)$).

The initial target pose, expressed in the camera frame, is given by $P_{T/C} = (0.01, 0.01, 0.6, -0.4, 0.1, -1.6)$. In this particular configuration, the classical approach [see (1) with $\widehat{L}_s(t) = \widehat{L}_{(s(t), Z^*)}$] leads to a failure because the movement produced makes the visual features to leave the image. Before the constraint handling (visibility and mechanical) is shown, the influence of the prediction horizon is studied (see Fig. 2). When the prediction horizon increases, we can underline that the image trajectories become more continuous, and the camera motion becomes straightforward. Due to lack of place, only the results obtained with the LM with Z^* (LM_p) are shown.

Remark: Since the cost function J is equivalent to the standard criterion of the classical IBVS, the result obtained for the classical IBVS with Z^* and the optimal $\mathrm{IBVS}_{\mathrm{LM}_p}$ are identical. The only difference is the control behavior that ensures an exponential decay of the error after the saturation in the first case (classical IBVS) and, consequently, increasing a little bit the time to convergence.
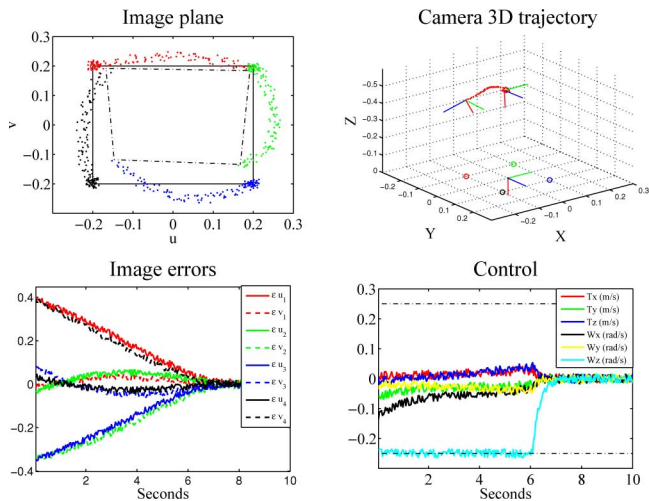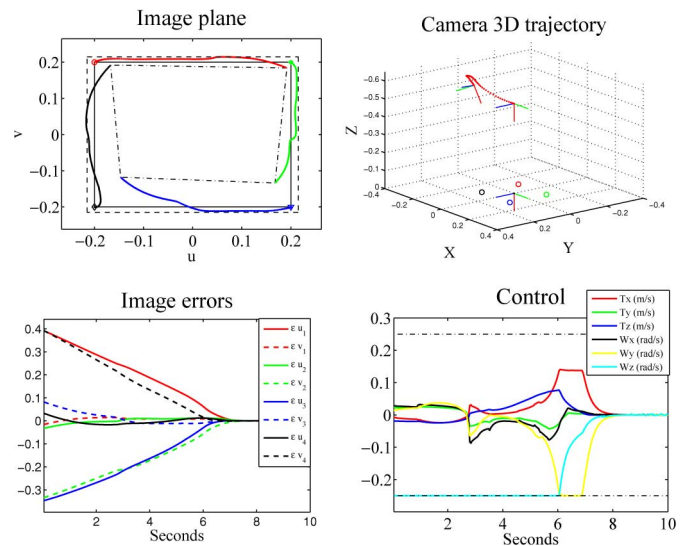
Fig. 3. Robustness of VPC_{LM_p} with $N_p = 10$.

Fig. 4. Optimal $\text{IBVS}_{\text{LM}_p}$ and visibility constraints.

Fig. 5. VPC_{LM_p} with $N_p = 10$ and visibility constraints.

Fig. 6. VPC_{GM_p} with $N_p = 10$ and visibility constraints.

To test the robustness of the proposed approach, a white noise that is uniformly distributed with amplitude of 0.02 is then added to the process output to simulate image-processing errors. Moreover, to simulate that the robot is not perfectly actuated, the control inputs are artificially affected by a white noise with the same characteristics, as previously discussed. Finally, modeling errors of 15% on the camera intrinsic parameters are considered. Because of the IMC structure, the control objective is still satisfied (see Fig. 3), even if all variables (input control, image measurements, and camera pose) are affected by noises.

To illustrate the capability of the visibility-constraint handling, the previous image trajectories are now constrained to stay in a window, which is defined by the following inequalities:

$$\begin{bmatrix} u_{\min =} - 0.21 \\ v_{\min =} - 0.21 \end{bmatrix} \leq s_m(j) \leq \begin{bmatrix} u_{\max} = 0.21 \\ v_{\max} = 0.21 \end{bmatrix}. \quad (25)$$

The optimal IBVS is able to fulfill the visibility constraints, but discontinuities in the image plane as well as on the control inputs can be noted (see Fig. 4). This is due to a corrective behavior of the controller, i.e., prediction at the next step only, instead of a preventive

behavior obtained with the VPC, i.e., prediction over N_p steps (see Fig. 5). As stated earlier, when a prediction horizon is considered, the camera motion becomes straightforward, and the discontinuities in the image plane disappear.

Concerning the GM, i.e., GM_p with constant depth Z^*, all the visibility constraints are not perfectly satisfied because the model that is used to predict the image evolution does not describe the process faithfully (see Fig. 6). Indeed, the approximation of $Z(t)$ by Z^* induces a coarse estimation of the visual features. Although the model satisfies the constraints, the process does not verify the constraints due to the modeling errors with Z^*. However, this problem can be easily overcome by taking a margin of error in the definition of visibility constraints (25).

If the correct depth Z is now considered at each sampling time, the results obtained with LM and GM are very similar in the unconstrained case or in the constrained case. Of course, for each case, 2-D and 3-D trajectories are different. Only the case with the GM_c and visibility constraints is presented (see Fig. 7). The control task and the visibility constraints are simultaneously satisfied.
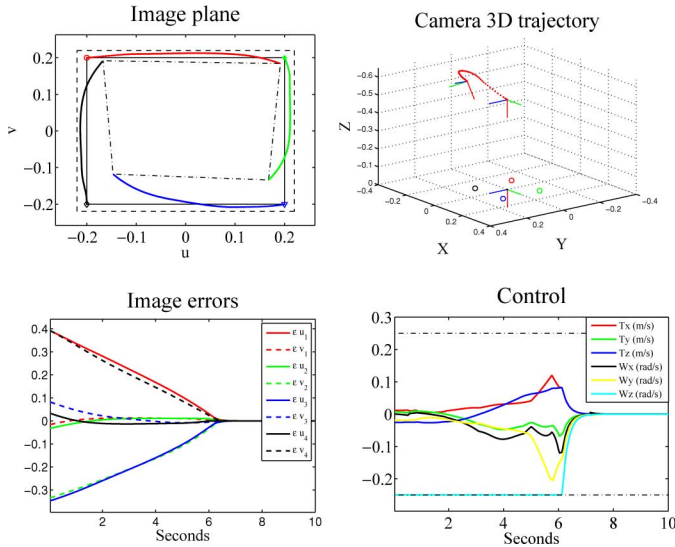
Fig. 7. VPC$_{\mathrm{GM}_c}$ with $N_p = 10$ and visibility constraints.

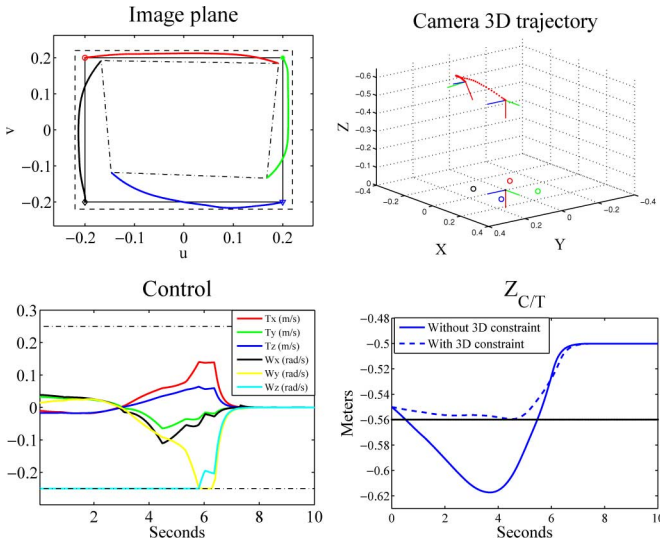Fig. 8. VPC$_{\mathrm{GM}_c}$ with $N_p = 10$ and visibility and mechanical constraints.

Finally, to illustrate the 3-D constraints, the camera workspace is limited along the Z-axis to $[-0.5; -0.6]$. As can be seen in Fig. 8, the VPC$_{\mathrm{GM}_c}$ controller converges under visibility and mechanical constraints (the same result is obtained with VPC$_{\mathrm{LM}_c}$). If no admissible trajectory that ensures both visibility and 3-D constraints exists, the VPC stops at the position that minimizes the constrained cost function.

Concerning the computational time, for the worst case presented in this section, i.e., presence of mechanical and visibility constraints with a prediction horizon equal to 10, the required computational time to solve the optimization problem is about 156 ms per sampling period. In the unconstrained case, if the prediction horizon is equal to 1 (respectively, 5), the computational time is 47 ms (respectively, 78 ms). This illustrates the computational effort required, according to the prediction horizon.

The over-all computational time can be greatly reduced by optimization of the code, as well as by using a more-efficient minimization algorithm. We can use, for example, an interior penalty method that guarantees the constraint satisfaction at any time. Consequently, the

computed control is always an admissible control so that the algorithm can be stopped once the time allowed is over. It will undoubtedly allow an implementation at video rate.

V. CONCLUSION

In this paper, IBVS has been expressed as a nonlinear optimization problem in the image plane over a prediction horizon. We have shown that VPC provides an efficient solution to both the weak points of the classical IBVS.

1) Because of the optimization formulation, the constraints, such as visibility constraints and 3-D limitations, can be easily handled.
2) Because of the image prediction, convergence and satisfactory behaviors can be obtained for displacements that lead to failures in the classical approach.

The prediction of the visual feature evolution has been ensured by a model. Different kinds of models have been proposed. It can be an LM based on the interaction matrix or a nonlinear GM. Its choice depends on the availability of 3-D data. The LM is easy to implement. The GM has a large validity domain that provides convergence for large displacements.

Simulations highlight the efficiency of VPC, and underline that the improvements with respect to the classical IBVS are only due to the image prediction and are not due to the optimization algorithm. The robustness of the proposed approach has also been validated in the presence of modeling errors and noise.

The VPC approach suffers from the same drawback of predictive-control approach that is the proof of the stability. Because of the constrained nonlinear optimization, the solution is numerical, and then, it is theoretically difficult to prove the stability [12]. However, let us recall that only the local asymptotic stability can be demonstrated for the classical IBVS. This local stability is thus also ensured for the optimal IBVS that we have considered.

Finally, the VPC strategy is very flexible, and it can be used whatever the robotic system, i.e., mobile robot [2] or robot arm, and the camera, i.e., perspective or catadioptric [1]. The 2-D constraint handling can also be very useful for obstacle avoidance.

REFERENCES

[1] G. Allibert, E. Courtial, and Y. Touré, "Visual predictive control for manipulators with catadioptric camera," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, May 2008, pp. 510–515.

[2] G. Allibert, E. Courtial, and Y. Touré, "Real-time visual predictive controller for image-based trajectory tracking of mobile robot," presented at the 17th IFAC World Congr., Seoul, Korea, Jul. 2008.

[3] G. Allibert and E. Courtial, "What can prediction bring to imabe-based visual servoing?," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, Oct. 2009, pp. 5210–5215.

[4] J. P. Barreto, J. Batista, and H. Aratijo, "Model predictive control to improve visual control of motion: Applications in active tracking of moving targets," presented at the Int. Conf. Pattern Recognit., Barcelona, Spain, Sep. 2000.

[5] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control* (Lecture Notes in Control and Information Sciences, vol. 237), D. Kriegman, G. Hager, A. S. Morse, Eds.. New York: Springer-Verlag, 1998, pp. 66–78.

[6] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[7] G. Chesi and A. Vicino, "Visual servoing for large camera displacements," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 724–735, Aug. 2004.

[8] G. Chesi, "Visual servoing path planning via homogeneous forms and LMI optimizations," *IEEE Trans. Robot. Autom.*, vol. 25, no. 2, pp. 281–291, Apr. 2009.

[9] E. Chong and S. H. Zak, *An Introduction to Optimization*, 2nd ed. New York: Wiley, 2001, ISBN-10: 0471391263.

[10] S. Durola, P. Danès, D. Coutinho, and M. Courdesses, "Rational systems and matrix inequalities to the multicriteria analysis of visual servos," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp.1504–1509.

[11] P. Danès, D. Bellot, "Towards an LMI approach to multicriteria visual servoing in robotics," *Eur. J. Control*, vol. 12, no. 1, pp. 86–110, 2006.

[12] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," presented at the Benelux Meeting Syst. Control, Veldhoven, Pays Bas, The Netherlands, 2002.

[13] J. Gangloff and M. De Mathelin, "Visual servoing of a 6 dof manipulator for unknown 3-D profile following," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 511–520, Aug. 2002.

[14] R. Ginhoux, J. Gangloff, M. De Mathelin, M. Soler, and L. Sanchez, "Active filtering of physiological motion in robotized surgery using predictive control," *IEEE Trans. Robot. Autom.*, vol. 21, no. 1, pp. 67–79, Feb. 2005.

[15] K. Hashimoto and H. Kimura, "LQ optimal and nonlinear approaches to visual servoing," in *Visual Servoing*(World Scientific Series in Robotics and Intelligent Systems), K. Hashimoto, Ed, vol. 7. Singapore: World Scientific, 1993, pp. 165–198.

[16] M. Kazemi, K. Gupta, and M. Mehrandezh, "Global path planning for robust visual servoing in complex environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp. 326–332.

[17] R. Mahony, P. Corke, and F. Chaumette, "Choice of image features for depth-axis control in image-based visual servo control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, Oct. 2002, pp. 390–395.

[18] Y. Mezouar and F. Chaumette, "Optimal camera trajectory with image-based control," *Int. J. Robot. Res.*, vol. 22, no. 10, pp. 781–804, 2003.

[19] T. Murao, T. Yamada, and M. Fujita, "Predictive visual feedback control with eye-in-hand system via stabilizing receding horizon approach," in *Proc. 45th IEEE CDC*, San Diego, CA, Dec. 2006, pp.1758–1763.

[20] M. Morari and E. Zafiriou, *Robust Control*. Paris, France: Dunod, 1983.

[21] N. Papanikolopoulos, P. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: A combination of vision and control," *IEEE Trans. Robot. Autom.*, vol. 9, no. 1, pp. 14–35, Feb. 1993.

[22] M. Sauvée, P. Poignet, E. Dombre, and E. Courtial, "Image based visual servoing through nonlinear model predictive control," in *Proc. 45th IEEE CDC*, San Diego, CA, Dec. 2006, pp.1776–1781.

[23] F. Schramm and G. Morel, "Ensuring visibility in calibration-free path planning for image-based visual servoing," *IEEE Trans. Robot. Autom.*, vol. 22, no. 4, pp. 848–854, Aug. 2006.

A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing

Farrokh Janabi-Sharifi and Mohammed Marey

Abstract—The problem of estimating position and orientation (pose) of an object in real time constitutes an important issue for vision-based control of robots. Many vision-based pose-estimation schemes in robot control rely on an extended Kalman filter (EKF) that requires tuning of filter parameters. To obtain satisfactory results, EKF-based techniques rely on "known" noise statistics, initial object pose, and sufficiently high sampling rates for good approximation of measurement-function linearization. Deviations from such assumptions usually lead to degraded pose estimation during visual servoing. In this paper, a new algorithm, namely iterative adaptive EKF (IAEKF), is proposed by integrating mechanisms for noise adaptation and iterative-measurement linearization. The experimental results are provided to demonstrate the superiority of IAEKF in dealing with erroneous *a priori* statistics, poor pose initialization, variations in the sampling rate, and trajectory dynamics.

Index Terms—Adaptation, Kalman filter (KF), control, pose estimation, robotic manipulator, visual servoing.

I. INTRODUCTION

In computer vision, the problem of *pose estimation* is to determine the position and orientation (pose) of a camera with respect to an object's coordinate frame using the image information. The problem is also known as extrinsic camera-calibration problem with its solution playing a crucial rule in the success of many computer-vision applications, such as object recognition [1], intelligent surveillance [2], and robotic visual servoing (RVS) [3]. Estimation of the camera displacement (CD) between the current and desired pose for RVS [4], [5] is also relevant to this problem. However, the focus of this study will be on pose estimation for RVS where the relative pose between a camera and an object is used for real-time control of a robot motion [3].

In RVS, the control error can be calculated in the image space, Cartesian space, or both (hybrid) spaces [3], [6], [7]. While partial estimation of the pose vector (e.g., depth) is required for image-based and hybrid visual-servoing schemes [8], [9], an important class of visual-servoing methods, namely the position-based visual-servoing (PBVS) scheme, requires full pose estimation to calculate Cartesian error of the relative pose between the endpoint and the object [10]. Two major difficulties with pose estimation for RVS are related to the requirements for efficiency and robustness of pose estimation [11].

The solutions to pose-estimation problem usually focus on using sets of 2-D–3-D correspondences between geometric features and their projections on the image plane. Although high-level geometric features, such as lines and conics, have been proposed, point features are typically used for pose estimation due to their ease of availability