

Global Path-Planning for Constrained and Optimal Visual Servoing

Graziano Chesi, *Senior Member, IEEE*, and Y. S. Hung, *Senior Member, IEEE*

Abstract—Visual servoing consists of steering a robot from an initial to a desired location by exploiting the information provided by visual sensors. This paper deals with the problem of realizing visual servoing for robot manipulators taking into account constraints such as visibility, workspace (that is obstacle avoidance), and joint constraints, while minimizing a cost function such as spanned image area, trajectory length, and curvature. To solve this problem, a new path-planning scheme is proposed. First, a robust object reconstruction is computed from visual measurements which allows one to obtain feasible image trajectories. Second, the rotation path is parameterized through an extension of the Euler parameters that yields an equivalent expression of the rotation matrix as a quadratic function of unconstrained variables, hence, largely simplifying standard parameterizations which involve transcendental functions. Then, polynomials of arbitrary degree are used to complete the parametrization and formulate the desired constraints and costs as a general optimization problem. The optimal trajectory is followed by tracking the image trajectory with an IBVS controller combined with repulsive potential fields in order to fulfill the constraints in real conditions.

Index Terms—Path-planning, robot manipulator, trajectory constraints, trajectory costs, visual servoing.

I. INTRODUCTION

VISUAL servoing via the “teaching-by-showing” approach has received increasing attention in recent years. It consists of steering an eye-in-hand robot from a current location to a desired location via a feedback control law based on the image projections of some reference features in the current and desired view. See for example [1]–[3] for detailed classifications of visual servoing approaches. Several methods to deal with this task have been proposed. Some examples are the position-based visual servoing (PBVS) where the feedback error is the camera pose (see for example [4], [5]), and the image-based visual servoing (IBVS) where the feedback error is the image error (see for example [6], [7]). In the 2 1/2 D visual servoing [2], the feedback error contains both the camera pose and the image error. Other methods exploit partitioning techniques [8]–[10], navigation functions [11], invariance with respect to the intrinsic parameters [12], image moments [13], and generation of circular-like trajectories [14].

Manuscript received August 9, 2006; revised December 7, 2006. This paper was recommended for publication by Associate Editor P. Rives and Editor L. Parker upon evaluation of the reviewers’ comments.

The authors are with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong (e-mail: chesi@eee.hku.hk; yshung@eee.hku.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2007.903817

When controlling a robot, convergence and robustness are not the only important issues. In fact, workspace limits should be taken into account to ensure that the trajectory of the robot does not collide with obstacles present in the scene. Joint limits should be considered because the robot may not be able to reach a certain location due to its particular structure. Besides issues concerning the physical constraints, there are also issues concerning the performance. In fact, when more than one feasible robot trajectory exists, one may be interested in finding the shortest trajectory, or the smoothest trajectory, or the trajectory which maximizes the visibility margin.

This paper proposes a visual servoing design approach for robot manipulators which allows one to take into account all these issues. In particular, a new path-planning technique in the image space is introduced, which allows one to consider constraints on visibility, workspace and joints, together with the objective of minimizing trajectory costs such as length, curvature, and spanned image area. This is achieved by parameterizing in the six-dimensional rigid motion space all the trajectories connecting the initial to the desired location which, although unknown, can be computed up to a scale factor through an object reconstruction from image measurements and, if available, also the CAD model of the observed object. In order to deal with calibration errors and image noise, a robust object reconstruction is proposed which allows one to obtain image trajectories satisfying the boundary conditions. In order to obtain functions that can be efficiently handled in optimization tools, the rotation path is parameterized through an extension of the Euler parameters which yields an equivalent expression of the rotation as a quadratic function of unconstrained variables. Polynomials of arbitrary degree are then used to complete the parametrization. The so obtained optimization can be solved through any tool such as gradient search and simplex, by simply evaluating the roots of one-variable polynomials at each step. Once the image trajectory is computed, the camera is steered to the desired location by using an IBVS controller such as that proposed in [15]. Finally, repulsive potential fields are combined with the IBVS tracking control law in order to fulfill the constraints when the image trajectory cannot be exactly followed due to calibration errors and image noise. It is worthwhile to note that other path-planning methods have been proposed in [15]–[18] which solve related problems by exploiting, in a discretized framework, optimal control formulations, screw motions, and geodesic paths modulated by repulsive potential fields respectively. The approach in this paper proposes a new strategy which consists of parameterizing all the possible trajectories satisfying a number of constraints while minimizing a cost. This is achieved by using a new approach based on parameter-de-

pendent object reconstruction, extended Euler parameters, and a combination of repulsive potential fields with the IBVS tracking control law.

The paper is organized as follows. Section II introduces the problem formulation. Section III presents the proposed path-planning scheme. Section IV illustrates some constraints and costs that can be considered, and describes the strategy proposed to track the found trajectory. Section V presents some simulation results in ideal and in real conditions. Lastly, Section VI ends the paper with some final remarks.

II. PRELIMINARIES

Let us introduce the following notation. Let \mathbb{R} denote the real number set, $SO(3)$ the set of rotation matrices in $\mathbb{R}^{3 \times 3}$, and $SE(3)$ the Cartesian product $SO(3) \times \mathbb{R}^3$. We denote with v_i the i th component of the vector \mathbf{v} , \mathbf{I}_n the $n \times n$ identity matrix, $\mathbf{0}_n$ the $n \times 1$ null vector, $\mathbf{1}_n$ the $n \times 1$ vector with all elements equal to 1, \mathbf{e}_i the i th column of \mathbf{I}_3 , and $[\mathbf{v}]_\times$ the skew-symmetric matrix of $\mathbf{v} \in \mathbb{R}^3$. Also, given the camera frames $\mathcal{F}_1, \mathcal{F}_2$ with pose $\{\mathbf{O}_1, \mathbf{c}_1\}, \{\mathbf{O}_2, \mathbf{c}_2\} \in SE(3)$ with respect to the absolute frame, we define the pose of \mathcal{F}_1 with respect to \mathcal{F}_2 as

$$\begin{cases} \mathcal{P}(\mathcal{F}_1, \mathcal{F}_2) = \{\mathbf{O}_{12}, \mathbf{c}_{12}\} \\ \mathbf{O}_{12} = \mathbf{O}_2^T \mathbf{O}_1 \\ \mathbf{c}_{12} = \mathbf{O}_2^T (\mathbf{c}_1 - \mathbf{c}_2). \end{cases} \quad (1)$$

Moreover:

- $\mathbf{A} \in \mathbb{R}^{3 \times 3}$: upper triangular intrinsic parameters matrix;
- $\mathcal{F}^\circ, \mathcal{F}^*$: initial and desired camera frames with pose;
- $\{\mathbf{O}^\circ, \mathbf{c}^\circ\}, \{\mathbf{I}_3, \mathbf{0}_3\} \in SE(3)$ with respect to the absolute frame (taken to be \mathcal{F}^*);
- $\boldsymbol{\rho}_i \in \mathbb{R}^3$: i th point in the three-dimensional space expressed with respect to the absolute frame;
- $\mathbf{p}_i^\circ = [x_i^\circ, y_i^\circ, 1]'$, $\mathbf{p}_i^* = [x_i^*, y_i^*, 1]'$ $\in \mathbb{R}^3$: projections in pixel coordinates of the i th point on \mathcal{F}° and \mathcal{F}^* according to

$$\begin{cases} \alpha_i^\circ \mathbf{p}_i^\circ = \mathbf{A} \mathbf{O}^\circ (\boldsymbol{\rho}_i - \mathbf{c}^\circ) \\ \alpha_i^* \mathbf{p}_i^* = \mathbf{A} \boldsymbol{\rho}_i \end{cases} \quad (2)$$

where $\alpha_i^\circ, \alpha_i^* \in \mathbb{R}$ are the point depths.

Let us suppose that a set $\mathcal{S} = \{(\mathbf{p}_i^\circ, \mathbf{p}_i^*), i = 1, \dots, n\}$ of n object point correspondences is available. The problem consists of steering the camera from the initial to the desired location:

- 1) satisfying visibility, workspace, and joint constraints;
- 2) optimizing a certain trajectory cost.

III. PATH-PLANNING

The strategy proposed in this paper consists of generating image trajectories of the object points and then tracking them by using IBVS controllers. We indicate the image trajectory of the i th point as $\mathbf{p}_i(w)$ where $w \in [0, 1]$ is the trajectory abscise, with $w = 0$ indicating the initial location and $w = 1$ the desired location. The vector $\mathbf{p}_i(w)$ must satisfy the boundary conditions

$$\begin{cases} \mathbf{p}_i(0) = \mathbf{p}_i^\circ \\ \mathbf{p}_i(1) = \mathbf{p}_i^* \end{cases} \quad (3)$$

The above conditions are not the only constraints that $\mathbf{p}_i(w)$ must satisfy. In fact, the set of $\mathbf{p}_i(w)$, $i = 1, \dots, n$, must be

such that there exists a parameter-dependent camera frame from which the observed object points match the $\mathbf{p}_i(w)$ for all $w \in [0, 1]$. In order to cope with this problem as well as allow one to deal with constraints and trajectory costs defined outside the image space, we introduce a new parametrization as described in Sections III-A–D.

A. Trajectory Parametrization

Let us define the camera pose of \mathcal{F}^* with respect to \mathcal{F}° as

$$\mathcal{P}(\mathcal{F}^*, \mathcal{F}^\circ) = \{\mathbf{R}, \mathbf{t}\}. \quad (4)$$

Given \mathcal{S} , \mathbf{A} , and the CAD model of the object (that is the set of physical points $\boldsymbol{\rho}_i$), one can calculate the camera pose $\{\mathbf{R}, \mathbf{t}\}$ by solving (2)–(4) through linear least-squares techniques.

If the CAD model of the object is not available, \mathbf{t} can be computed only up to a scale factor which stands for the unknown distance between the initial and desired frame origins. Indeed, the normalized camera pose $\{\mathbf{R}, \mathbf{t}_{\text{norm}}\}$ with $\mathbf{t}_{\text{norm}} = \mathbf{t}/\|\mathbf{t}\|$ can be computed through the essential matrix algorithm or the homography matrix algorithm relative to a virtual plane in the case of non coplanar features supposing $n \geq 8$. If the features are known to be coplanar, the camera pose can be computed through the homography matrix algorithm supposing $n \geq 4$. See [19] and [20] for details. For pure rotation motion, i.e., $\mathbf{t} = \mathbf{0}_3$, the normalized translation is defined as $\mathbf{t}_{\text{norm}} = \mathbf{0}_3$.

We will henceforth assume that the camera pose has been estimated as $\{\mathbf{R}, \mathbf{d}\}$, \mathbf{d} being either the physical translation \mathbf{t} or the normalized translation \mathbf{t}_{norm} . Let $\mathcal{F}_d(w)$ be the camera frame along the reconstructed camera trajectory, and define the pose of $\mathcal{F}_d(w)$ with respect to \mathcal{F}° as

$$\mathcal{P}(\mathcal{F}_d(w), \mathcal{F}^\circ) = \{\mathbf{R}(w), \mathbf{d}(w)\}. \quad (5)$$

At the extreme points of the trajectory, this pose must satisfy

$$\begin{cases} \{\mathbf{R}(0), \mathbf{d}(0)\} = \{\mathbf{I}_3, \mathbf{0}_3\} \\ \{\mathbf{R}(1), \mathbf{d}(1)\} = \{\mathbf{R}, \mathbf{d}\}. \end{cases} \quad (6)$$

In ideal conditions, that is in the absence of calibration errors and image noise, the trajectory of the i th object point can be expressed in terms of $\mathbf{R}(w)$ and $\mathbf{d}(w)$ as

$$\bar{\alpha}_i(w) \mathbf{p}_i(w) = \mathbf{A} \mathbf{R}(w)^T (\mathbf{u}_i - \mathbf{d}(w)) \quad (7)$$

where $\bar{\alpha}_i(w)$ is the parameterized point depth and \mathbf{u}_i is the i th physical point $\boldsymbol{\rho}_i$ in the camera frame \mathcal{F}° , referred to as *object reconstruction*. This object reconstruction can be computed by solving the system

$$\begin{cases} \bar{\alpha}_i(0) \mathbf{p}_i^\circ = \mathbf{A} \mathbf{u}_i \\ \bar{\alpha}_i(1) \mathbf{p}_i^* = \mathbf{A} \mathbf{R}^T (\mathbf{u}_i - \mathbf{d}) \end{cases} \quad (8)$$

which amounts to calculating an SVD by eliminating the point depths $\bar{\alpha}_i(0), \bar{\alpha}_i(1)$. Hence, the sought trajectory can be expressed as the solution of the constrained minimization problem shown in (9) at the bottom of the next page, where $g_k(w)$, $k = 1, \dots, m_g$, indicate the constraints that the camera must satisfy along the trajectory, and h is the cost function to be minimized by the trajectory.

In order to establish if $g_k(w)$ is positive for all $w \in [0, 1]$, one must evaluate, at each step of the optimization procedure, each function $g_k(w)$ at $w = 0$, at $w = 1$, and at the points in $[0, 1]$ where its derivative vanishes in order to find all the global minima. However, this turns out to be difficult and computationally intensive with standard parameterizations because $\mathbf{R}(w)$, and consequently $g_k(w)$, contains transcendental terms (for example exponential coordinates, x - y - z angles, etc.).

In order to cope with this problem, we will derive in the sequel a special parametrization of the trajectories for which the functions $g_k(w)$ are polynomial in w . In fact, the roots of polynomials can be easily found. Polynomial trajectories have been used in [21] for the problem of following a straight line.

B. Robust Object Reconstruction

In the presence of calibration errors and image noise, the system (8) may admit no solution for \mathbf{u}_i . Clearly, a least-squares solution can still be computed through SVD, but this solution cannot allow the image trajectory $\mathbf{p}_i(w)$ provided by (7) to satisfy the image boundary conditions (3), even if the CAD model of the object is available (if this model is available one can obtain more accurate estimates of \mathbf{R} , \mathbf{t} as explained in Section III-A, and then use these estimates in the following procedure).

Therefore, we look for a *robust object reconstruction* which minimizes the effect of uncertainties on our robot control. In particular, we introduce a parameter-dependent object reconstruction $\mathbf{u}_i(w)$ satisfying

$$\begin{cases} \tilde{\alpha}_i(0)\mathbf{p}_i^\circ = \mathbf{A}\mathbf{u}_i(0) \\ \tilde{\alpha}_i(1)\mathbf{p}_i^* = \mathbf{A}\mathbf{R}^T(\mathbf{u}_i(1) - \mathbf{d}) \end{cases} \quad (10)$$

for some point depths $\tilde{\alpha}_i(0), \tilde{\alpha}_i(1)$. Among all possible solutions $\mathbf{u}_i(w)$ for the above system, we select the linear solution because it is the simplest and because it is the closest to the ideal constant \mathbf{u}_i in (8). Hence, let us write $\mathbf{u}_i(w)$ as

$$\mathbf{u}_i(w) = (1-w)\mathbf{a}_i + \mathbf{b}_i. \quad (11)$$

Fig. 1 illustrates the idea of robust object reconstruction for path-planning.

Since we are interested in the solution closest to the constant one, we aim to find the $\mathbf{u}_i(w)$ with the smallest \mathbf{a}_i which satisfies (10), that is

$$\min_{\mathbf{a}_i, \mathbf{b}_i} \|\mathbf{a}_i\| \text{ s.t. (10)-(11)}. \quad (12)$$

In order to solve (12), let us observe that, by eliminating the point depths $\tilde{\alpha}_i(0), \tilde{\alpha}_i(1)$, (10) can be rewritten as

$$\begin{cases} \mathbf{F}_i(\mathbf{a}_i + \mathbf{b}_i) = \mathbf{0}_2 \\ \mathbf{F}_i^*(\mathbf{b}_i - \mathbf{d}) = \mathbf{0}_2 \end{cases} \quad (13)$$

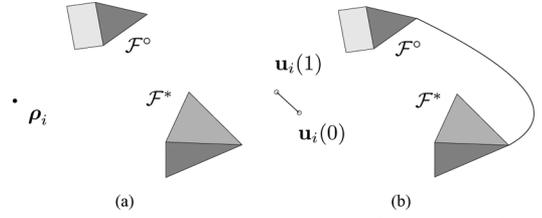


Fig. 1. (a) Observed point ρ_i and camera frames \mathcal{F}° and \mathcal{F}^* . (b) Robust object reconstruction $\mathbf{u}_i(w)$ for the point ρ_i as function of the trajectory abscise in the presence of calibration errors and image noise.

where

$$\begin{cases} \mathbf{F}_i = \begin{bmatrix} \mathbf{e}_1^T - \bar{x}_i^\circ \mathbf{e}_3^T \\ \mathbf{e}_2^T - \bar{y}_i^\circ \mathbf{e}_3^T \end{bmatrix} \\ \mathbf{F}_i^* = \begin{bmatrix} \mathbf{r}_1^T - \bar{x}_i^* \mathbf{r}_3^T \\ \mathbf{r}_2^T - \bar{y}_i^* \mathbf{r}_3^T \end{bmatrix} \\ [\bar{x}_i^\circ, \bar{y}_i^\circ, 1]^T = \mathbf{A}^{-1}\mathbf{p}_i^\circ \\ [\bar{x}_i^*, \bar{y}_i^*, 1]^T = \mathbf{A}^{-1}\mathbf{p}_i^* \end{cases} \quad (14)$$

and $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are the columns of \mathbf{R} . The solution of (12) can be found through the Lagrange's multipliers and is given by

$$\begin{cases} \mathbf{a}_i = -\mathbf{F}_i^T (\mathbf{F}_i \mathbf{F}_i^T)^{-1} \mathbf{F}_i (\gamma_i \boldsymbol{\delta}_i + \mathbf{d}) \\ \mathbf{b}_i = \gamma_i \boldsymbol{\delta}_i + \mathbf{d} \end{cases} \quad (15)$$

where

$$\begin{cases} \gamma_i = -\frac{\boldsymbol{\delta}_i^T \mathbf{F}_i^T (\mathbf{F}_i \mathbf{F}_i^T)^{-1} \mathbf{F}_i \mathbf{d}}{\boldsymbol{\delta}_i^T \mathbf{F}_i^T (\mathbf{F}_i \mathbf{F}_i^T)^{-1} \mathbf{F}_i \boldsymbol{\delta}_i} \\ \boldsymbol{\delta}_i = \bar{x}_i^* \mathbf{r}_1 + \bar{y}_i^* \mathbf{r}_2 + \mathbf{r}_3. \end{cases} \quad (16)$$

The robust object reconstruction is finally given by

$$\mathbf{u}_i(w) = \left(\mathbf{I}_3 - (1-w)\mathbf{F}_i^T (\mathbf{F}_i \mathbf{F}_i^T)^{-1} \mathbf{F}_i \right) (\gamma_i \boldsymbol{\delta}_i + \mathbf{d}). \quad (17)$$

We observe that the above expression is well-posed because the matrix $\mathbf{F}_i \mathbf{F}_i^T$ is always nonsingular, being

$$\det(\mathbf{F}_i \mathbf{F}_i^T) = 1 + \bar{x}_i^{\circ 2} + \bar{y}_i^{\circ 2}. \quad (18)$$

Lastly, the image projection $\mathbf{p}_i(w)$ along the trajectory is given by the solution of

$$\tilde{\alpha}_i(w)\mathbf{p}_i(w) = \mathbf{A}\mathbf{R}(w)^T(\mathbf{u}_i(w) - \mathbf{d}(w)). \quad (19)$$

C. Rotation Parameter

The proposed approach aims to derive a simple parametrization for the rotation to enable the optimization to be formulated as easily as possible. In particular, we look for rational parameterizations of the rotation since they can be more easily handled than others such as irrational or trigonometrical.

We start by considering the representation of rotation matrices through the Euler parameters (see for example [22]). According to this representation, any rotation matrix can be rep-

$$\begin{aligned} & \min_{\mathbf{R}(w):\mathbb{R} \rightarrow SO(3), \mathbf{d}(w):\mathbb{R} \rightarrow \mathbb{R}^3} h(\mathbf{R}(w), \mathbf{d}(w)) \\ & \text{s.t.} \quad \begin{cases} \text{pose boundary conditions (6)} \\ \text{image boundary conditions (3)} \\ g_k(w, \mathbf{R}(w), \mathbf{d}(w)) > 0 \quad \forall w \in [0, 1] \quad \forall k = 1, \dots, m_g \end{cases} \end{aligned} \quad (9)$$

resented as shown in (20) at the bottom of the page for some $\phi \in \mathbb{R}^4$ satisfying $\|\phi\| = 1$. Also, $\Lambda(\phi)$ is a rotation matrix for any ϕ satisfying $\|\phi\| = 1$. The Euler parameters corresponding to \mathbf{R} can be found as $\phi = \xi(\mathbf{R})$ where

$$\xi(\mathbf{R}) = \left[\sin \frac{\theta}{2} \mathbf{u}^T, \cos \frac{\theta}{2} \right]^T \quad (21)$$

where $\theta \in [0, \pi]$ and $\mathbf{u} \in \mathbb{R}^3$, $\|\mathbf{u}\| = 1$, are respectively the rotation angle and axis in the exponential coordinates of \mathbf{R} , i.e., $\mathbf{R} = e^{[\theta \mathbf{u}]_{\times}}$.

Hence, one could parameterize the rotation through (20) and the vector ϕ . However, ϕ is not free due to the constraint $\|\phi\| = 1$. Clearly, one could then parameterize ϕ in order to ensure this constraint, but this would require the introduction of irrational or trigonometrical functions.

Therefore, in order to get rid of the constraint $\|\phi\| = 1$ and obtain a rational parametrization of the rotation, we introduce the following extended Euler parametrization:

$$\Phi(\phi) = \frac{1}{\|\phi\|^2} \Lambda(\phi). \quad (22)$$

This parametrization is complete, in fact for any rotation matrix \mathbf{R} there exists ϕ such that $\mathbf{R} = \Phi(\phi)$. In particular

$$\mathbf{R} = \Phi(\gamma \xi(\mathbf{R})) \quad \forall \gamma \neq 0. \quad (23)$$

Moreover, $\Phi(\phi)$ is a rotation matrix for all $\phi \neq \mathbf{0}_4$ (through limit operations one can see that $\Phi(\phi)$ is a rotation matrix also for $\phi = \mathbf{0}_4$ but this limit is not unique).

We observe that we have gotten rid of the constraint $\|\phi\| = 1$ at the expense of the denominator in (22). As it will become clear in Section IV, this denominator does not affect the optimization problem.

D. Polynomial Parametrization

Let us parameterize the rotation of the camera frame $\mathcal{F}_d(w)$ as

$$\mathbf{R}(w) = \Phi(\phi(w)) \quad (24)$$

where $\phi(w)$ is a parameterized extended Euler parameter. The camera pose along the trajectory is, hence, described by $\phi(w)$ and $\mathbf{d}(w)$. Let us express these vectors as polynomials according to

$$\begin{cases} \phi(w) = \tilde{\mathbf{M}} [w^{\delta_M}, w^{\delta_M-1}, \dots, w, 1]^T \\ \mathbf{d}(w) = \tilde{\mathbf{N}} [w^{\delta_N}, w^{\delta_N-1}, \dots, w, 1]^T \end{cases} \quad (25)$$

where $\tilde{\mathbf{M}} \in \mathbb{R}^{4 \times \delta_M+1}$ and $\tilde{\mathbf{N}} \in \mathbb{R}^{3 \times \delta_N+1}$ are coefficient matrices. In order to satisfy the rotation boundary conditions in (6), we impose

$$\begin{cases} \phi(0) = [0, 0, 0, 1]^T \\ \phi(1) = \xi(\mathbf{R}) \end{cases} \quad (26)$$

(clearly, one can equivalently impose the same quantity scaled by the same factor γ with $\gamma \neq 0$). Then, taking into account (25), one has that the pose boundary conditions (6) are satisfied if and only if

$$\begin{cases} \tilde{\mathbf{M}} [\mathbf{0}_{\delta_M}^T, 1]^T = [\mathbf{0}_3^T, 1]^T, & \tilde{\mathbf{N}} [\mathbf{0}_{\delta_N}^T, 1]^T = \mathbf{0}_3 \\ \tilde{\mathbf{M}} \mathbf{1}_{\delta_M+1} = \xi(\mathbf{R}), & \tilde{\mathbf{N}} \mathbf{1}_{\delta_N+1} = \mathbf{d} \end{cases} \quad (27)$$

which imply that the matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{N}}$ can be parameterized as

$$\begin{cases} \tilde{\mathbf{M}} = [\xi(\mathbf{R}) - \mathbf{M} \mathbf{1}_{\delta_M-1} - [\mathbf{0}_3^T, 1]^T, \mathbf{M}, [\mathbf{0}_3^T, 1]^T] \\ \tilde{\mathbf{N}} = [\mathbf{d} - \mathbf{N} \mathbf{1}_{\delta_N-1}, \mathbf{N}, \mathbf{0}_3] \end{cases} \quad (28)$$

where $\mathbf{M} \in \mathbb{R}^{4 \times \delta_M-1}$ and $\mathbf{N} \in \mathbb{R}^{3 \times \delta_N-1}$ are free matricial parameters. Therefore, the camera pose along the trajectory is parameterized by the matrices \mathbf{M} and \mathbf{N} , and the optimization problem (9) can be rewritten as

$$\begin{aligned} \min_{\mathbf{M} \in \mathbb{R}^{4 \times \delta_M-1}, \mathbf{N} \in \mathbb{R}^{3 \times \delta_N-1}} & h(\mathbf{M}, \mathbf{N}) \\ \text{s.t.} & g_k(w, \mathbf{M}, \mathbf{N}) > 0 \\ & \forall w \in [0, 1] \\ & \forall k = 1, \dots, n_g. \end{aligned} \quad (29)$$

Let us observe that the boundary conditions (6) and (3) are implicitly satisfied in (29). In fact, the pose boundary conditions (6) are ensured by (25)–(28), while the image boundary conditions (3) are ensured by (17)–(19).

Remark 1: Problem (9) may not admit any solution, for example if the desired pose is out of the workspace. This is detected in problem (29) by a minimum cost equal to infinity. Let us also observe that, if there exists a solution to (9) with continuous derivatives along the trajectory, then there exist matrices \mathbf{M}, \mathbf{N} approximating arbitrarily well this solution from standard arguments of functions representation via polynomial series.

Remark 2: The solution of (29) may not be unique, that is the same optimal cost may be achieved by different trajectories. For example, this is the case when, minimizing the trajectory length, there exist different feasible trajectories with the same projection on the translational space. Let us observe that one does not need to distinguish among these solutions, that is one can simply adopt the solution returned by the solver, since all these solutions satisfy the imposed constraints and achieve the same cost.

$$\Lambda(\phi) = \begin{bmatrix} \phi_1^2 - \phi_2^2 - \phi_3^2 + \phi_4^2 & 2(\phi_1\phi_2 - \phi_3\phi_4) & 2(\phi_1\phi_3 + \phi_2\phi_4) \\ 2(\phi_1\phi_2 + \phi_3\phi_4) & -\phi_1^2 + \phi_2^2 - \phi_3^2 + \phi_4^2 & 2(\phi_2\phi_3 - \phi_1\phi_4) \\ 2(\phi_1\phi_3 - \phi_2\phi_4) & 2(\phi_2\phi_3 + \phi_1\phi_4) & -\phi_1^2 - \phi_2^2 + \phi_3^2 + \phi_4^2 \end{bmatrix} \quad (20)$$

IV. CONSTRAINTS, COSTS, AND TRACKING

A. Visibility Constraint

Let us write the image projection $\mathbf{p}_i(w)$ as $\mathbf{p}_i(w) = [x_i(w), y_i(w), 1]$. In order to guarantee that the image projections along the trajectory remain in the field of view of the camera, we have to introduce the visibility constraint

$$\left. \begin{array}{l} x_{\min} < x_i(w) < x_{\max} \\ y_{\min} < y_i(w) < y_{\max} \end{array} \right\} \forall w \in [0, 1] \forall i = 1, \dots, n \quad (30)$$

where $x_{\min}, x_{\max}, y_{\min}, y_{\max} \in \mathbb{R}$ are the screen limits. This can be done as follows.

By eliminating the point depth $\tilde{\alpha}_i(w)$ in (19) one obtains

$$\begin{aligned} x_i(w) &= \frac{\mathbf{e}_1^T \mathbf{A} \mathbf{R}(w)^T (\mathbf{u}_i(w) - \mathbf{d}(w))}{\mathbf{e}_3^T \mathbf{R}(w)^T (\mathbf{u}_i(w) - \mathbf{d}(w))} \\ y_i(w) &= \frac{\mathbf{e}_2^T \mathbf{A} \mathbf{R}(w)^T (\mathbf{u}_i(w) - \mathbf{d}(w))}{\mathbf{e}_3^T \mathbf{R}(w)^T (\mathbf{u}_i(w) - \mathbf{d}(w))} \end{aligned} \quad (31)$$

Taking into account the parametrization of $\mathbf{R}(w)$ in (22), one has for $x_i(w)$

$$\begin{aligned} x_i(w) &= \frac{\mathbf{e}_1^T \mathbf{A} \frac{1}{\|\boldsymbol{\phi}(w)\|^2} \boldsymbol{\Lambda}(\boldsymbol{\phi}(w))^T (\mathbf{u}_i(w) - \mathbf{d}(w))}{\mathbf{e}_3^T \frac{1}{\|\boldsymbol{\phi}(w)\|^2} \boldsymbol{\Lambda}(\boldsymbol{\phi}(w))^T (\mathbf{u}_i(w) - \mathbf{d}(w))} \\ &= \frac{\mathbf{e}_1^T \mathbf{A} \boldsymbol{\Lambda}(\boldsymbol{\phi}(w))^T (\mathbf{u}_i(w) - \mathbf{d}(w))}{\mathbf{e}_3^T \boldsymbol{\Lambda}(\boldsymbol{\phi}(w))^T (\mathbf{u}_i(w) - \mathbf{d}(w))}. \end{aligned} \quad (32)$$

As we can see, the denominator in (22) does not affect the image projections, which means that in the optimization problem the rotation is equivalently parameterized by a simple quadratic expression, namely $\boldsymbol{\Lambda}(\boldsymbol{\phi}(w))$, which depends quadratically on the parameter $\boldsymbol{\phi}(w)$ and, according to (25)–(28), depends quadratically on the parameter \mathbf{M} . Let us observe that $\boldsymbol{\phi}(w)$ is unconstrained and, hence, \mathbf{M} is as well. This allows one to simplify (29) because it does not require the introduction of additional constraints on \mathbf{M} .

Let us introduce the polynomials

$$a_{i,j}(w) = \mathbf{e}_j^T \mathbf{A} \boldsymbol{\Lambda}(\boldsymbol{\phi}(w))^T (\mathbf{u}_i(w) - \mathbf{d}(w)), \quad i = 1, \dots, n, \quad j = 1, 2, 3. \quad (33)$$

It follows that

$$\left\{ \begin{array}{l} x_i(w) = \frac{a_{i,1}(w)}{a_{i,3}(w)} \\ y_i(w) = \frac{a_{i,2}(w)}{a_{i,3}(w)}. \end{array} \right. \quad (34)$$

Hence, the constraint (30) can be rewritten as

$$\left. \begin{array}{l} a_{i,1}(w) - x_{\min} a_{i,3}(w) > 0 \\ -a_{i,1}(w) + x_{\max} a_{i,3}(w) > 0 \\ a_{i,2}(w) - y_{\min} a_{i,3}(w) > 0 \\ -a_{i,2}(w) + y_{\max} a_{i,3}(w) > 0 \\ a_{i,3}(w) > 0 \end{array} \right\} \forall w \in [0, 1] \quad \forall i = 1, \dots, n. \quad (35)$$

Let us observe that the inequality $a_{i,3}(w) > 0$ has been included in order to ensure that the object remains in front of the camera

for the whole trajectory. This allows us to get rid of the denominator and obtain only the polynomial inequalities in (35). Each of these inequalities represents one inequality in (29).

B. Workspace Constraints

Due to obstacles present in the scene, the robot is not allowed to reach certain positions of the three-dimensional space. This means that the camera center can assume values in a subset of \mathbb{R}^3 only. In our path-planning, the camera center is represented by the translation vector $\mathbf{d}(w)$ as pose with respect to the initial frame \mathcal{F}^0 . Therefore, workspace constraints correspond to constraints on this vector.

Now, depending on the information available for the robot control problem we are considering, two kinds of workspace constraints can be taken into account.

1) *Absolute Workspace Constraints*: If the CAD model of the object is available, one can calculate the nonnormalized translation between \mathcal{F}^* and \mathcal{F}^0 as explained in Section III-A. This means that one can constrain the optimization problem (29) so that the camera center of the generated trajectories belongs to an absolute set referred to either the initial or the desired camera frame. Since the desired camera frame usually represents a reference location for the robot, we consider for instance the case of this absolute set referred to \mathcal{F}^* . Let \mathcal{F} be the current camera frame and let us define

$$\mathcal{P}(\mathcal{F}, \mathcal{F}^*) = \{\mathbf{O}, \mathbf{c}\}. \quad (36)$$

The absolute set can be expressed as

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{R}^3 : f_j(\mathbf{c}) > 0, \quad j = 1, \dots, n_f\} \quad (37)$$

where $f_j : \mathbb{R}^3 \rightarrow \mathbb{R}$ are polynomials. Then, since

$$\mathbf{c} = \mathbf{R}^T(\mathbf{d}(w) - \mathbf{d}) \quad (38)$$

one can constrain the trajectories in (29) so that the camera center belongs to \mathcal{C} by imposing

$$f_j(\mathbf{R}^T(\mathbf{d}(w) - \mathbf{d})) > 0, \quad j = 1, \dots, n_f. \quad (39)$$

2) *Scaled Workspace Constraints*: If the CAD model of the object is not available, one can calculate only the normalized translation \mathbf{t}_{norm} between \mathcal{F}^* and \mathcal{F}^0 . This means that only scaled constraints on the camera center can be imposed in the optimization problem (29). In particular, these constraints have the form (39) but in this case they define the set

$$\mathcal{C}_{\text{norm}} = \{\mathbf{c} \in \mathbb{R}^3 : f_j(\|\mathbf{t}\|\mathbf{c}) > 0, \quad j = 1, \dots, n_f\}. \quad (40)$$

As we can see, this set depends on the initial camera pose through the term $\|\mathbf{t}\|$. Although less general than the absolute workspace constraints, the scaled workspace constraints can, however, model typical situations in which the robot works.

Finally, let us observe that:

1) one can consider also workspace constraints on the camera orientation by defining

$$\tilde{f}_j(\boldsymbol{\phi}(w)) > 0, \quad j = 1, \dots, n_{\tilde{f}} \quad (41)$$

where $\tilde{f}_j : \mathbb{R}^4 \rightarrow \mathbb{R}$ are polynomials;

2) workspace constraints on the links of the manipulator cannot be directly considered through inequalities such as (39), however, one can introduce more conservative workspace constraints on the camera center in order to keep the links sufficiently far away from the forbidden region.

C. Joint Constraints

Workspace constraints are not the only constraints which limit the set of positions that the robot can reach. In fact, this set is limited also by the physical limits of the joints of the robot. In principle, one can impose joint constraints through the workspace constraints defined in the previous section, but this may result in a complex system of inequalities $c_j(\cdot) > 0$ required to define the set \mathcal{C} . A more direct way consists of exploiting the kinematic relations of the robot if these are available. In particular, let $\mathbf{q} \in \mathbb{R}^{n_q}$ be the vector of robot joints, and consider the problem of imposing the condition that \mathbf{q} belongs to the set

$$\mathcal{Q} = \{\mathbf{q} \in \mathbb{R}^{n_q} : q_{i,\min} < q_i < q_{i,\max}\}. \quad (42)$$

Let $\mathcal{L} : \mathbb{R}^{n_q} \rightarrow SE(3)$ be the kinematic function describing the pose of the camera frame \mathcal{F} as

$$\mathcal{L}(\mathbf{q}) = \{\mathbf{O}, \mathbf{c}\}. \quad (43)$$

Suppose that the nonnormalized translation between \mathcal{F}^* and \mathcal{F}^0 can be calculated (as explained in Section III-A this requires additional information such as the CAD model of the object) and, hence, $\mathbf{d} = \mathbf{t}$. Then, one can impose $\mathbf{q} \in \mathcal{Q}$ on the trajectories of problem (29) by imposing that

$$\mathcal{L}^{-1}(\{\mathbf{R}^T \mathbf{R}(w), \mathbf{R}^T(\mathbf{d}(w) - \mathbf{d})\}) \in \mathcal{Q} \quad \forall w \in [0, 1]. \quad (44)$$

However, one cannot include this constraint in (29) if the inverse kinematic function is nonpolynomial.

This problem can be dealt with by introducing polynomial functions $l_{i,\min}, l_{i,\max} : SE(3) \rightarrow \mathbb{R}^{n_q}$ satisfying

$$l_{i,\min}(\{\mathbf{O}, \mathbf{c}\}) \leq \mathcal{L}_i^{-1}\{\mathbf{O}, \mathbf{c}\} \leq l_{i,\max}(\{\mathbf{O}, \mathbf{c}\}) \quad \forall \{\mathbf{O}, \mathbf{c}\} \in \mathcal{S} \quad \forall i = 1, \dots, n_q \quad (45)$$

where \mathcal{S} is a bounded subset of $SE(3)$ denoting the region of interest for the robot. One can now impose $\mathbf{q} \in \mathcal{Q}$ by requiring that

$$\begin{aligned} & \left. \begin{aligned} l_{i,\min}(\{\mathbf{R}^T \mathbf{R}(w), \mathbf{R}^T(\mathbf{d}(w) - \mathbf{d})\}) &> q_{i,\min} \\ l_{i,\max}(\{\mathbf{R}^T \mathbf{R}(w), \mathbf{R}^T(\mathbf{d}(w) - \mathbf{d})\}) &< q_{i,\max} \end{aligned} \right\} \\ & \quad \forall w \in [0, 1]. \quad (46) \end{aligned}$$

We observe that the proposed strategy can also deal with the case of uncertain inverse kinematic function. In fact, suppose that $\mathcal{L}(\mathbf{q})$ is not exactly known and, hence, so is $\mathcal{L}^{-1}(\cdot)$. By introducing the functions $l_{i,\min}$ and $l_{i,\max}$ satisfying the constraint (45) for all admissible values of the uncertainties, one can ensure that the planned trajectories belong to \mathcal{Q} for all considered uncertainties.

In some cases depending on the structure of the robot, it can be complicated to compute the functions $l_{i,\min}$ and $l_{i,\max}$. In these cases one can compute the trajectory in the joint space at each iteration of the solver for (29) and establish if this trajectory remains in \mathcal{Q} as in [15].

D. Costs

One can consider several cost functions in (29) depending on the specific problem. Some of the most useful are the following.

1) *Spanned Image Area*: Here we consider the problem of minimizing the area spanned by the image trajectory. This can be done by defining the cost function $h(\mathbf{M}, \mathbf{N}) = \Delta$ where

$$\Delta = (\sigma_{x,\max} - \sigma_{x,\min})(\sigma_{y,\max} - \sigma_{y,\min}) \quad (47)$$

where $\sigma_{x,\min}, \sigma_{x,\max}, \sigma_{y,\min}, \sigma_{y,\max}$ denote the coordinates of the boundary box of the image trajectory according to

$$\begin{cases} \sigma_{x,\min} = \min\{x_i(w), w \in [0, 1], i = 1, \dots, n\} \\ \sigma_{x,\max} = \max\{x_i(w), w \in [0, 1], i = 1, \dots, n\} \\ \sigma_{y,\min} = \min\{y_i(w), w \in [0, 1], i = 1, \dots, n\} \\ \sigma_{y,\max} = \max\{y_i(w), w \in [0, 1], i = 1, \dots, n\}. \end{cases} \quad (48)$$

In order to compute these coordinates, let us consider first $\sigma_{x,\min}$. Since $x_i(w)$ in (34) is a rational function whose denominator is positive for all $w \in [0, 1]$, one can compute $\sigma_{x,\min}$ by evaluating each $x_i(w)$ at the extremes point of the trajectory and at the points in $[0, 1]$ where its derivative vanishes, as shown in (49) at the bottom of the page. Therefore, the computation of $\sigma_{x,\min}$ requires just the computation of the roots of a one-variable polynomial. The other coordinates $\sigma_{x,\max}, \sigma_{y,\min}, \sigma_{y,\max}$ are analogously calculated.

2) *Trajectory Length*: Another useful cost is the length of the camera trajectory in the three-dimensional space, which can be imposed as

$$h(\mathbf{M}, \mathbf{N}) = \int_0^1 \left\| \frac{\partial \mathbf{d}(w)}{\partial w} \right\| dw. \quad (50)$$

The integral can be computed through finite difference approximations in order to speed up the calculation.

$$\begin{cases} \sigma_{x,\min} = \min\{x_{i,\min}, i = 1, \dots, n\} \\ x_{i,\min} = \min \left\{ \frac{a_{i,1}(0)}{a_{i,3}(0)}, \frac{a_{i,1}(1)}{a_{i,3}(1)}, \bar{x}_{i,\min} \right\} \\ \bar{x}_{i,\min} = \min \left\{ \frac{a_{i,1}(w)}{a_{i,3}(w)}, w : \frac{\partial a_{i,1}(w)}{\partial w} a_{i,3}(w) - a_{i,1}(w) \frac{\partial a_{i,3}(w)}{\partial w} = 0 \right\} \end{cases} \quad (49)$$

3) *Curvature*: Also, one can minimize the curvature of the trajectory given by

$$\kappa = \left\| \left\| \frac{\partial \mathbf{d}(w)}{\partial w} \right\|^2 \frac{\partial^2 \mathbf{d}(w)}{\partial w^2} - \left(\frac{\partial^2 \mathbf{d}(w)^T}{\partial w^2} \frac{\partial \mathbf{d}(w)}{\partial w} \right) \frac{\partial \mathbf{d}(w)}{\partial w} \right\|. \quad (51)$$

This can be done by defining

$$h(\mathbf{M}, \mathbf{N}) = \max_{w \in [0,1]} \kappa. \quad (52)$$

This cost function can be simply computed by finding the maxima of κ in the interval $[0, 1]$, operation that amounts to evaluating κ at $w = 0$, at $w = 1$, and at the points where its derivative vanishes as is done for Δ in (47).

E. Tracking the Computed Image Trajectory

Once the problem (29) has been solved and the optimal values of \mathbf{M} and \mathbf{N} are available, one can compute the planned image trajectory as

$$\mathbf{s}(w) = [x_1(w), y_1(w), \dots, x_n(w), y_n(w)]^T \quad (53)$$

where $x_i(w)$ and $y_i(w)$ are given by (34) with $\phi(w)$ and $\mathbf{d}(w)$ defined as in (25)–(28) for the optimal values of \mathbf{M} and \mathbf{N} . In order to track this trajectory, one can use an IBVS controller similar to that proposed in [15]. Specifically, let $\mathbf{s} = [x_1, y_1, \dots, x_n, y_n]^T$ be the vector containing the image projections in the current camera frame \mathcal{F} . Define the error function

$$\mathbf{e} = \mathbf{E}(\mathbf{s} - \mathbf{s}(w)) \quad (54)$$

where $\mathbf{E} \in \mathbb{R}^{6 \times 2n}$ is a full rank matrix. We have that $\dot{\mathbf{e}} = \mathbf{E}(\dot{\mathbf{s}} - \dot{\mathbf{s}}(w))$ where

$$\begin{aligned} \dot{\mathbf{s}} &= \frac{\partial \mathbf{s}}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \mathbf{q}} \dot{\mathbf{q}} \\ &= \mathbf{LJ} \dot{\mathbf{q}} \end{aligned} \quad (55)$$

in which $\mathbf{Y} \in \mathbb{R}^6$ is the camera pose vector, $\mathbf{L} \in \mathbb{R}^{2n \times 6}$ is the interaction matrix between \mathbf{s} and the camera pose (see for example [15]), and $\mathbf{J} \in \mathbb{R}^{6 \times n_q}$ is the robot Jacobian linking the camera velocity to the joint velocities. Define the following control law in the joint space:

$$\dot{\mathbf{q}} = \lambda \mathbf{e} + \mathbf{J}^+ \mathbf{L}^+ \dot{\mathbf{s}}(w) \quad (56)$$

where $\lambda \in \mathbb{R}$ is a control gain, and \mathbf{J}^+ and \mathbf{L}^+ are the pseudoinverse of \mathbf{J} and \mathbf{L} . It follows that

$$\dot{\mathbf{e}} = \lambda \mathbf{E} \mathbf{L} \mathbf{J} \mathbf{e} + \mathbf{E} (\mathbf{L} \mathbf{J} \mathbf{J}^+ \mathbf{L}^+ - \mathbf{I}_{2n}) \dot{\mathbf{s}}(w). \quad (57)$$

Let us select $\mathbf{E} = \bar{\mathbf{E}} \mathbf{L}^+$ and consider the case of a 6 degree-of-freedom (DOF) robot, hence, implying that $\mathbf{J} \mathbf{J}^+ = \mathbf{I}_6$. It follows that

$$\begin{aligned} \dot{\mathbf{e}} &= \lambda \bar{\mathbf{E}} \mathbf{L}^+ \mathbf{L} \mathbf{J} \mathbf{e} + \bar{\mathbf{E}} \mathbf{L}^+ (\mathbf{L} \mathbf{J} \mathbf{J}^+ \mathbf{L}^+ - \mathbf{I}_{2n}) \dot{\mathbf{s}}(w) \\ &= \lambda \bar{\mathbf{E}} \mathbf{J} \mathbf{e}. \end{aligned} \quad (58)$$

Hence, the tracking system is asymptotically stable provided that $\lambda \bar{\mathbf{E}} \mathbf{J}$ is a Hurwitz matrix, that is a matrix whose eigenvalues have negative real part. For example, this can be achieved by selecting $\lambda < 0$ and $\bar{\mathbf{E}} = \mathbf{J}^+$.

Let us observe that, in the presence of image noise and calibration errors, the real image trajectory will differ from the planned one, and the constraints may not be satisfied. In order to cope with this problem, one can consider more conservative constraints (that is, larger safety margins) in the planning phase in order to facilitate their fulfillment by the real camera trajectory. Moreover, one can add repulsive components to the control law (56) based on potential field methods. In particular, for the visibility constraint one can proceed in a way similar to [9] and add the repulsive component- ρ to the z -component of the translational velocity, with

$$\rho = \begin{cases} \left(\frac{1}{\nu} - \frac{1}{\mu_\nu} \right) \frac{1}{\nu^2}, & \text{if } \nu < \mu_\nu \\ 0, & \text{otherwise} \end{cases} \quad (59)$$

where μ_ν is the thickness of the potential field and

$$\nu = \min_{i=1, \dots, n} \nu_i \quad (60)$$

$$\nu_i = \min \{x_i, y_i, x_{\max} - x_i, y_{\max} - y_i\}. \quad (61)$$

For the joint constraint we can proceed in a way similar to [15] and define the potential field

$$\begin{aligned} v_q &= \sum_{i=1}^{n_q} (1 - v_{q,i})^{-1} \\ v_{q,i} &= \begin{cases} \mu_q^{-2} (q_i - q_{i,\max} + \mu_q)^2, & \text{if } q_i > q_{i,\max} - \mu_q \\ \mu_q^{-2} (q_i - q_{i,\min} - \mu_q)^2, & \text{if } q_i < q_{i,\min} + \mu_q \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (62)$$

where μ_q is analogous to μ_ν , and similarly we can define v_w for the workspace constraint. Then, the control law (56) becomes

$$\dot{\mathbf{q}} = \lambda \mathbf{e} + \mathbf{J}^+ (\mathbf{L}^+ \dot{\mathbf{s}}(w) - \lambda_\nu \rho \mathbf{Y}_z) - \left(\lambda_w \frac{\partial v_w}{\partial \mathbf{q}} + \lambda_q \frac{\partial v_q}{\partial \mathbf{q}} \right) \quad (63)$$

where λ_ν , λ_w , λ_q are positive scalar gains and $\mathbf{Y}_z \in \mathbb{R}^6$ is the column of \mathbf{I}_6 identifying the position of the z -component of the translation in the camera pose vector \mathbf{Y} .

V. EXAMPLES

In this section, we illustrate the proposed approach by simulation examples. We present two types of simulations:

- simulations in ideal conditions;
- simulations in real conditions, that is in the presence of image noise and uncertainties on the intrinsic and extrinsic parameters.

Moreover, we also simulate a 6 DOF robot in order to derive the control signal in the joint space. In particular, we consider the industrial Cartesian robot XRS-0047RH shown in Fig. 2(b) and schematically represented in Fig. 2(a) with a 3 DOF wrist

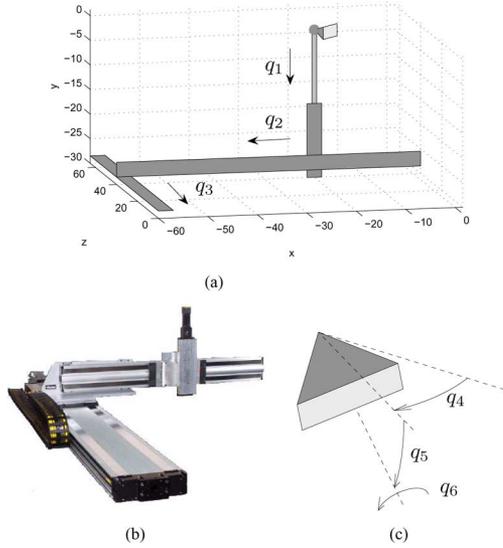


Fig. 2. (b) Industrial Cartesian robot XRS-0047RH and (a) its schematic representation with the camera in the desired frame \mathcal{F}^* coinciding with the absolute frame. (c) Camera orientation.

mounting the camera. The work area of the XRS-0047RH is $65 \times 60 \times 15$ cm. The robot is located with respect to the absolute frame \mathcal{F}^* so that the camera pose in (36) is described by the kinematic functions

$$\begin{cases} \mathbf{c} = [-q_2, -q_1, q_3]^T \\ \mathbf{O} = e^{[q_4 \mathbf{e}_2]_{\times}} e^{[q_5 \mathbf{e}_1]_{\times}} e^{[q_6 \mathbf{e}_3]_{\times}} \end{cases} \quad (64)$$

where q_1, \dots, q_6 are the translation and orientation joint variables shown in Fig. 2(a)–(c) constrained by

$$q_1 \in [0, 15] \text{ cm}, \quad q_2 \in [0, 60] \text{ cm}, \quad q_3 \in [0, 65] \text{ cm}. \quad (65)$$

Then, we adopt the PULNiX CCD camera considered in [23] with 6 mm lens and screen size 640×480 pixels, whose intrinsic parameters are calculated as

$$\mathbf{A} = \begin{bmatrix} 833 & 0.14 & 304 \\ 0 & 833 & 207 \\ 0 & 0 & 1 \end{bmatrix}. \quad (66)$$

The real conditions are characterized as follows.

- (RC1) Image noise: each image projection is randomly shifted, with uniform distribution, in a square with side equal to 1 pixel centered on the point itself.
- (RC2–RC3) Calibration errors: the matrices \mathbf{J} and \mathbf{A} are supposed coarsely estimated as

$$\hat{\mathbf{J}} = \begin{bmatrix} \mathbf{R}_E & [\mathbf{t}_E]_{\times} \mathbf{R}_E \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_E \end{bmatrix} \mathbf{J}, \quad \hat{\mathbf{A}} = \begin{bmatrix} 790 & 0.1 & 294 \\ 0 & 782 & 215 \\ 0 & 0 & 1 \end{bmatrix} \quad (67)$$

where $\mathbf{t}_E \in \mathbb{R}^3$ and $\mathbf{R}_E \in SO(3)$ account for errors in the coordinate transformation and are selected as

$$\mathbf{t}_E = [1, -0.5, 0.3]^T \text{ cm}, \quad \mathbf{R}_E = e^{[\pi[0.02, 0.03, 0.03]^T]_{\times}}. \quad (68)$$

- (RC4) Nonperfect IBVS control: the matrix \mathbf{L} in (56) is not exactly known as it depends on the point depths in

the current camera frame \mathcal{F} . In particular, we select to use the point depths in the camera frame $\mathcal{F}_d(w(t))$. We note that this choice may not be the best choice and that better results could be obtained by calculating the point depths by estimating at each iteration the camera pose $\mathcal{P}(\mathcal{F}, \mathcal{F}^*)$.

The problem (29) is solved for simple polynomials $\phi(w)$ and $\mathbf{d}(w)$ in (25) of degree 2 by using the function “fminsearch” of Matlab and the criteria described in Sections III–IV for evaluating the fulfillment of the constraints and the value of costs. The initial camera pose $\{\mathbf{R}, \mathbf{t}\}$ is estimated through the simple essential matrix algorithm. Once the image trajectory is computed, the IBVS control law (56) is applied by selecting $\lambda = 0.1 \text{ s}^{-1}$ and the trajectory abscise w equal to $w = 1 - e^{-t/\tau}$ where $\tau = 10 \text{ s}$. The visibility constraint is imposed in the planning phase with a margin of 50 pixels, and the same margin is used to define the potential field extension for the visibility constraint. Joint and workspace constraints are considered in Examples 2 and 3, respectively.

A. Example 1

Consider the situation depicted in Fig. 3(a) where a set of balls represents the object observed by the camera in the initial and desired frames \mathcal{F}^o and \mathcal{F}^* . Fig. 3(b) shows the centers of the balls in both initial and desired camera views. It is supposed that the CAD model of the object is not available.

Consider the problem to steer the camera from \mathcal{F}^o to \mathcal{F}^* minimizing the trajectory length. Fig. 3(c)–(d) shows the results obtained by planning the image trajectory through (29) and then tracking this trajectory with (56) in ideal conditions. Fig. 3(e)–(h) shows the results obtained by performing both planning and tracking phases in the real conditions RC1–RC4. Let us observe that some image projections reach the area of the potential field for the visibility constraint in Fig. 3(g), producing the deformation of the camera center trajectory according to the backward motion generated by this potential field in (63).

Let us also observe that, although the image trajectory in real conditions is different from that in ideal conditions, the trajectory length is quite similar, being 85.9 and 85.4 cm, respectively.

B. Example 2 (Joint Constraint)

Consider the situation depicted in Fig. 4(a) where the same object of Example 1 is observed from a different initial frame \mathcal{F}^o , and the problem to steer the camera minimizing the spanned image area. Fig. 4(c)–(d) shows the results obtained without the joint constraints in (64)–(65). Along this trajectory the camera goes above the horizontal plane $y = 0$ which is not allowed since $y = c_2 = -q_1$ and $q_1 \geq 0$.

In order to take into account the bound on q_1 , one can proceed as described in Section IV-C and impose the constraint

$$q_1 = -\mathbf{e}_2^T \mathbf{R}^T (\mathbf{d}(w) - \mathbf{d}) \geq 0 \quad (69)$$

(let us observe that the CAD model of the object is not required since the inequality is proportional to the translation). Fig. 4(e)–(f) shows the results obtained in ideal conditions, whereas Fig. 4(g)–(j) shows the results in the real conditions RC1–RC4. As we can see, the constraint $q_1 \geq 0$ is satisfied along the trajectory. Also, the potential field for the visibility constraint is not required in this example.

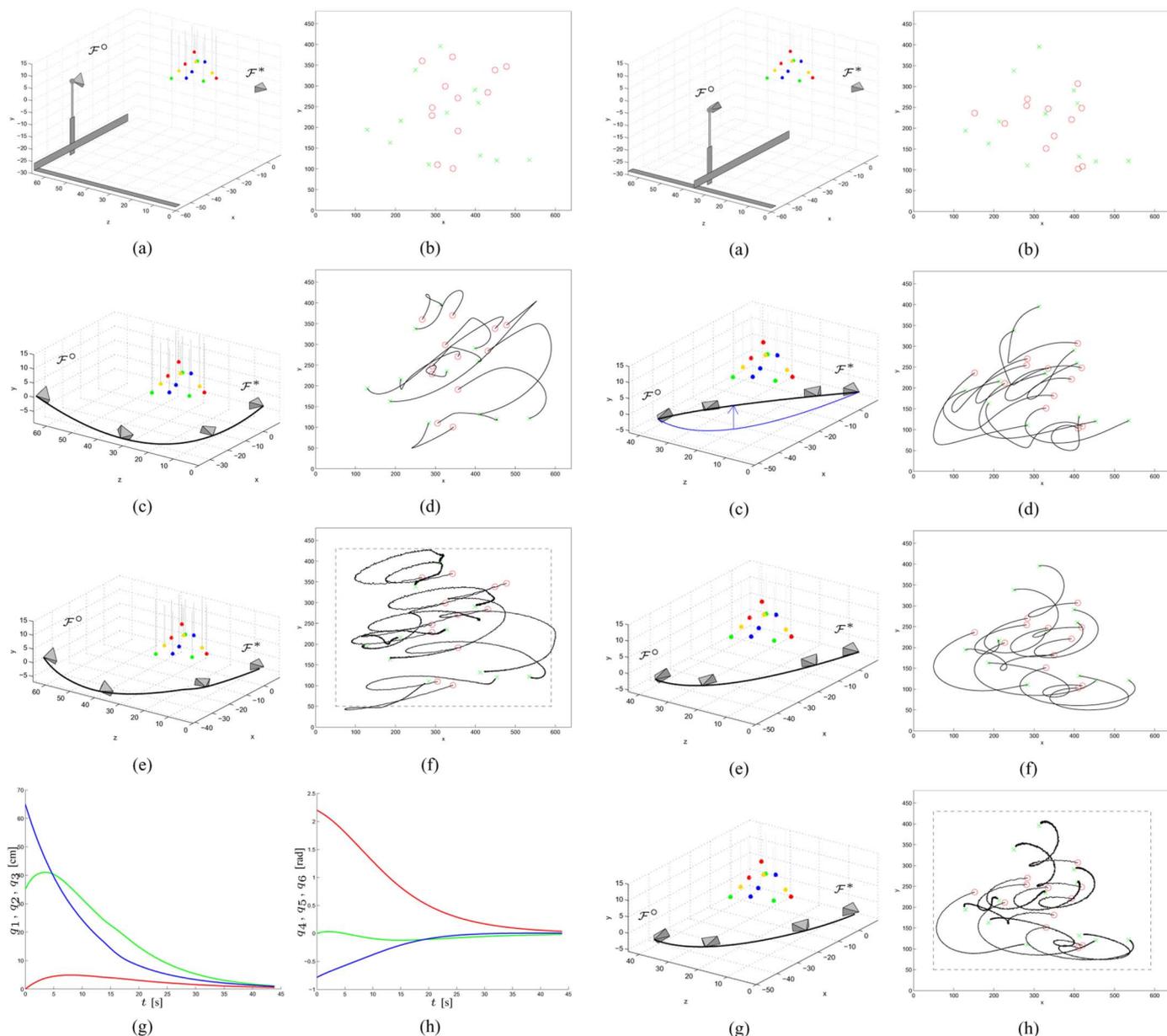


Fig. 3. Example 1. (a)-(b) Initial configuration. (c)-(d) Results in ideal conditions. (e)-(h) Results in real conditions with the joints in (g)-(h). In the camera views, the cross represents a point in the initial view, the circle represents a point in the desired view, and the dashed line the potential field area.

C. Example 3 (Obstacle Avoidance)

Consider the situation depicted in Fig. 5(a) where the large white spots of three dices are observed by the camera. Fig. 5(b) shows the centers of the spots in both initial and desired camera views. The hanging ball at $[-20, 0, 0]^T$ cm represents an obstacle which limits the robot workspace.

Consider the problem to steer the camera minimizing the curvature of the trajectory. Fig. 5(c)–(d) shows the results obtained without considering the workspace constraint imposed by the obstacle. As we can see, the trajectory of the camera intersects the obstacle.

In order to take into account this constraint, we can proceed as described in Section IV-B if the CAD model is available.

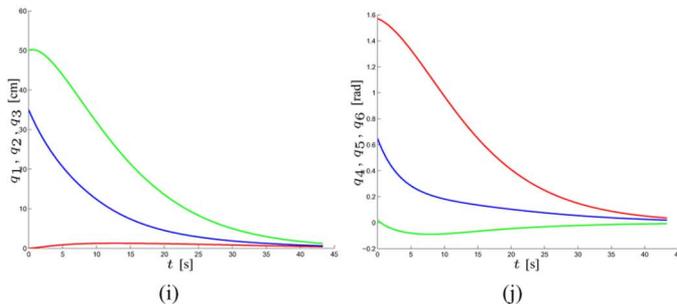


Fig. 4. Example 2. (a)-(b) Initial configuration. (c)-(d) Results for unconstrained joints: the camera goes above the plane $y = 0$ which is not allowed by (65). The arrow in (c) links the camera trajectory and its projection on the plane $y = 0$. (e)-(f) Results in ideal conditions taking into account the joint constraint. (g)-(j) Results in real conditions.

In particular, we can model the obstacle through a surrounding ellipsoid. We select a conservative ellipsoid in order to prevent the camera passing above the object (not allowed by the joint

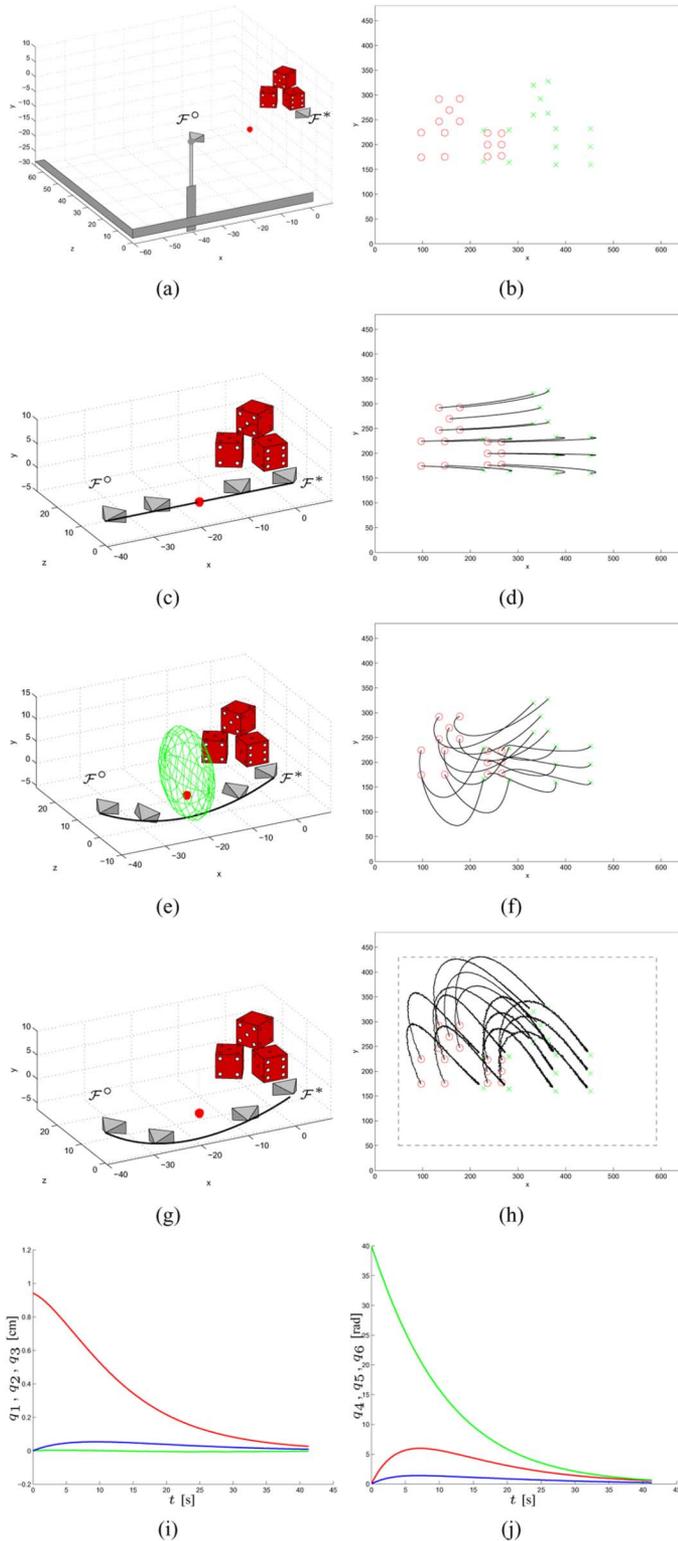


Fig. 5. Example 3. (a)–(b) Initial configuration. (c)–(d) Results for unconstrained workspace: the camera collides with the obstacle. (e)–(f) Results in ideal conditions taking into account the workspace constraint. (g)–(i) Results in real conditions.

constraints (65) or behind the obstacle (this would occlude the object). In particular, we select the allowed region as

$$\left(\frac{c_1 + 20}{4}\right)^2 + \left(\frac{c_2 - 5}{10}\right)^2 + \left(\frac{c_3}{10}\right)^2 - 1 > 0. \quad (70)$$

Fig. 5(e)–(f) shows the results obtained in ideal conditions, whereas Fig. 5(g)–(j) shows the results in the real conditions RC1–RC4 (the initial camera pose is estimated from the CAD model through a least-squares solution). As we can see, the camera avoids the obstacle.

Let us also observe that the image trajectory in real conditions is significantly different from that in ideal conditions though the trajectories of the camera center in the three-dimensional space are very similar. This is due to the existence of quite different solutions satisfying the imposed constraints (visibility and workspace constraints) and achieving similar costs (curvature of the trajectory).

VI. CONCLUSION

A new path-planning scheme has been proposed for constrained and optimal visual servoing. The contribution is to provide a global parametrization of the trajectories that the robot can follow in order to pick up, with simple computations, the one minimizing a general cost under a number of possible constraints including obstacle avoidance. In particular, all the trajectories connecting the initial to the desired location are parameterized in the six-dimensional rigid motion space. This is achieved by introducing a new object reconstruction which allows one to obtain image trajectories satisfying the boundary conditions even in the presence of calibration errors and image noise. In order to obtain functions that can be efficiently handled in optimization tools, the rotation path is parameterized through an extension of the Euler parameters which allows one to obtain an equivalent expression of the rotation as a quadratic function of unconstrained variables. Polynomials of arbitrary degree are used to complete the parametrization and formulate a general optimization where a number of constraints such as visibility, workspace, and joint constraints, and a number of costs such as spanned image area, trajectory length, and curvature, can be considered. Thanks to this parametrization, establishing the fulfillment of the constraints in the optimization procedure reduces to the simple calculation of the roots of a one-variable polynomial, hence, largely simplifying standard parameterizations which involve transcendental functions. Once the image trajectory is computed, the camera is steered to the desired location by using an IBVS controller combined with repulsive potential fields in order to fulfill the constraints in real conditions. Future work will deal with the problem to achieve robustness against all possible calibration errors.

ACKNOWLEDGMENT

The authors would like to thank the Editor, the Associate Editor, and the reviewers for their useful and constructive comments.

REFERENCES

- [1] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
- [2] E. Malis, F. Chaumette, and S. Boudet, “2 1/2 D visual servoing,” *IEEE Trans. Robot. Automat.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.
- [3] K. Hashimoto, “A review on vision-based control of robot manipulators,” *Adv. Robot.*, vol. 17, no. 10, pp. 969–991, 2003.
- [4] C. Taylor and J. Ostrowski, “Robust vision-based pose control,” in *Proc. IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, 2000, pp. 2734–2740.

- [5] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: Keeping the object in the field of vision," in *Proc. IEEE Int. Conf. Robot. Automat.*, Washington, DC, 2002, pp. 1624–1629.
- [6] A. C. Sanderson, L. E. Weiss, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Trans. Robot. Automat.*, vol. RA-3, no. 5, pp. 404–417, Oct. 1987.
- [7] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1991, pp. 2267–2272.
- [8] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proc. Int. Conf. Intell. Robots Syst.*, 1998, pp. 705–711.
- [9] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Automat.*, vol. 17, no. 4, pp. 507–515, Aug. 2001.
- [10] P. Oh and P. Allen, "Visual servoing by partitioning degrees of freedom," *IEEE Trans. Robot. Automat.*, vol. 17, no. 1, pp. 1–17, Feb. 2001.
- [11] N. Cowan, J. Weingarten, and D. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 521–533, Aug. 2002.
- [12] E. Malis, "Visual servoing invariant to changes in camera-intrinsic parameters," *IEEE Trans. Robot. Automat.*, vol. 20, no. 1, pp. 72–81, Feb. 2004.
- [13] O. Tahri and F. Chaumette, "Image moments: Generic descriptors for decoupled image-based visual servo," in *Proc. IEEE Int. Conf. Robot. Automat.*, New Orleans, LA, 2004, pp. 1185–1190.
- [14] G. Chesi and A. Vicino, "Visual servoing for large camera displacements," *IEEE Trans. Robotics*, vol. 20, no. 4, pp. 724–735, Aug. 2004.
- [15] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 534–549, Aug. 2002.
- [16] H. Zhang and J. Ostrowski, "Visual motion planning for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 18, no. 2, pp. 199–208, Apr. 2002.
- [17] J. Park and M. Chung, "Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy," *IEEE Trans. Robot. Automat.*, vol. 19, no. 2, pp. 250–258, Apr. 2003.
- [18] Y. Mezouar and F. Chaumette, "Optimal camera trajectory with image-based control," *Int. J. Robot. Res.*, vol. 22, no. 10–11, pp. 781–803, 2003.
- [19] O. Faugeras and Q.-T. Luong, *The Geometry of Multiple Images*. Cambridge, MA: MIT Press, 2001.
- [20] E. Malis and F. Chaumette, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. J. Comput. Vis.*, vol. 37, no. 1, pp. 79–97, 2000.
- [21] G. Chesi, D. Prattichizzo, and A. Vicino, "Visual servoing: Reaching the desired location following a straight line via polynomial parameterizations," in *IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, 2005, pp. 2559–2564.
- [22] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
- [23] Z. Zhang, "A flexible new technique for camera calibration," Microsoft, Redmond, WA, Tech. Rep. MSR-TR-98-71, 2002 [Online]. Available: <http://research.microsoft.com/~zhang/>



Graziano Chesi (S'00–M'01–SM'06) received the Laurea degree in information engineering from the University of Firenze, Firenze, Italy, in 1997, and the Ph.D. degree in systems engineering from the University of Bologna, Bologna, Italy, in 2001.

He was a Visiting Scientist at the Department of Engineering of the University of Cambridge, Cambridge, U.K., during 1999–2000, and he held fellowships from the Japanese Society for the Promotion of Science and the European Union at the Department of Information Physics and Computing of the University of Tokyo, Tokyo, Japan, during 2001–2004. He was a Research Associate and an Assistant Professor at the Department of Information Engineering of the University of Siena, Siena, Italy, during 2000–2006, and he joined the Department of Electrical and Electronic Engineering of the University of Hong Kong in 2006. His research interests include computer vision, nonlinear systems, robotics and robust control.

Dr. Chesi was the recipient of the Best Student Award of the Faculty of Engineering of the University of Firenze in 1997. He is an Associate Editor of *Automatica* and the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.



Y. S. Hung (M'88–SM'02) received the B.Sc. (Eng.) degree in electrical engineering and B.Sc. degree in mathematics, both from the University of Hong Kong, Hong Kong, and the M.Phil. and Ph.D. degrees from the University of Cambridge, Cambridge, U.K.

He worked as a Research Associate at the University of Cambridge and as a Lecturer at the University of Surrey before he joined the University of Hong Kong, where he is now an Associate Professor and the Head of the Department of Electrical and Electronic Engineering. His research interests include robust control, systems theory, robotics, computer vision and biomedical engineering.

Dr. Hung is a recipient of the Best Teaching Award in 1991 from the Hong Kong University Students' Union. He is a chartered engineer and a Fellow of IEE, U.K., and HKIE, Hong Kong.