[19] W.-L. Wang, G. Jin, Y. Yan, and M. Wu, "Image feature extraction with the optical Haar wavelet," *Opt. Eng.,* vol. 34, pp. 1238–1242, 1995.
[20] Y. S. Ho and A. Gersho, "Classified transform coding of images using vector quantization," in *IEEE Int. Conf. ASSP,* 1989, pp. 1890–1893.
[21] A. D. Calway and R. Wilson, "Curve extraction in images using a multiresolution framework," *CVGIP: Image Understanding,* vol. 59, pp. 359–366, 1994.
[22] H. Wang and S.-F. Chang, "A highly efficient system for automatic face region detection in MPEG video," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 7, pp. 615–628, 1997.
[23] K. C. Kluge, C. Kreucher, and S. Lakshmanan, "Tracking lane and pavement edges using deformable templates," in *Proc. SPIE Intell. Veh. Highway Syst. Conf.,* 1998.
[24] J. Kosecka, R. Blasi, C. J. Taylor, and J. Malik, "Vision-based lateral control of vehicles," in *Proc. IEEE Conf. Intell. Transportation Syst.,* 1997.
[25] E. Dickmanns and B. D. Mysliwetz, "Recursive 3-D road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 14, pp. 199–213, 1992.

On the Use of Linear Camera-Object Interaction Models in Visual Servoing

Benedetto Allotta and Carlo Colombo

Abstract—**We investigate the exploitation of linear models of camera-object interaction for an efficient modeling and control of image-based visual servoing systems. The approach includes a method for coping with those representation ambiguities typical of linear interaction models which may affect both planning and control. The implementation of an eye-in-hand servoing system based on affine camera models and using image contours as relevant visual features is described and discussed. The system, including an image planner, a two-dimensional/three-dimensional (2-D/3-D) controller, and a visual analysis module, allows an intuitive specification and execution of relative positioning tasks w.r.t. still or moving rigid objects. Results of real-time experiments with a robotic platform featuring a PUMA manipulator provide a further insight into characteristics and performance of affine visual servoing systems.**

Index Terms—**Affine interaction models, eye-in-hand systems, visual servoing.**

I. Introduction

The use of visual sensors in the exteroceptive feedback loop of a robot system, referred to as *visual servoing*, appears to be a natural approach to developing flexible positioning strategies, with applications including robot grasping, manipulation, and navigation (conveyor belt management, part-placement, assembly, etc.) [1], [2].

Several approaches to visual servoing were experimented in the recent past. Three-dimensional (3-D)-based approaches rely on closing the loop in the Cartesian space [3], [4]. These are less robust than image-based approaches [5], [6], where error signal measurement and loop closure are performed directly at the image level, thus bypassing any inaccuracies in camera calibration and kinematic models [7].

Notwithstanding the improvements of the last few years, much work remains to be done on both the control and computer vision aspects of visual servoing. Design issues with a key impact on the overall characteristics and performance of a visual servoing system are the models of visual interaction and the type of image features used for object tracking. The tracking strategies proposed so far have been based mainly on realistic models of camera projection (e.g., perspective) but rather simple primitives such as points or lines [8]–[10]. Yet, on the one hand tracking such primitives can be infeasible and/or lead to unreliable results for some robotic contexts, and on the other hand simpler camera models are accurate enough to carry out visual analysis in a number of real contexts [11], [12].

In this paper, we investigate using linear approximations in the modeling of an image-based visual servoing system with the purpose of devising control strategies. In Section II, we show that embedding an affine camera-object interaction model into the system model, the linear mapping between appearance evolution and 3-D motion assumes a form which is particularly suitable for real-time servoing, since it is independent of the number and type of features being tracked over time. We also expound a method to cope with the intrinsic limitations of linear models while keeping the advantages of the proposed framework.

The implementation of an affine visual servoing system featuring a manipulator-mounted camera and using active, affine-deformable contours as image features is described and motivated in Section III. The system includes a planner which permits unambiguous and safe task completion, an image-based servo controller, and an estimation module for on-line system parameter identification. Experimental results obtained with a setup featuring a PUMA manipulator are discussed in Section IV, providing an insight into system performance in terms of robustness and application perspectives.

II. Theoretical Framework

A. Preliminaries and Notation

Let us constrain the positioning problem to the geometric interaction between a camera and a single rigid object, in motion one w.r.t. the other. If $\boldsymbol{P}$ is a generic point of the object, its coordinates are expressed in a generic reference frame $\{w\} = \{\boldsymbol{P}_w,\, {}^wX,\, {}^wY,\, {}^wZ\}$ as ${}^w\boldsymbol{P} \in \mathbb{R}^3$. In the following, we use a camera-centered frame $\{c\}$ fixed to the camera and with ${}^cZ$ parallel to the optical axis, and a frame $\{o\}$ fixed to the object. Relative motion of camera and object is described by means of the relative twist screw $\Delta\boldsymbol{V} = \boldsymbol{V}_c - \boldsymbol{V}_o$, where $\boldsymbol{V}_c = [\boldsymbol{T}_c^T \quad \boldsymbol{\Omega}_c^T]^T$ and $\boldsymbol{V}_o = [\boldsymbol{T}_o^T \quad \boldsymbol{\Omega}_o^T]^T$ are, respectively, the camera and object twist screws, $\boldsymbol{T}$ and $\boldsymbol{\Omega}$ indicating translational and angular velocities.

Let $\boldsymbol{p} = [x \quad y]^T \in \mathbb{R}^2$ be the image projection of point $\boldsymbol{P}$. The generic camera projection model is expressed by $\boldsymbol{p} = \pi({}^o\boldsymbol{P}, \boldsymbol{\gamma})$, where $\boldsymbol{\gamma} \in \mathbb{R}^g$ is a vector of camera parameters. The image velocity of $\boldsymbol{p}$ can be expressed as

$$\dot{\boldsymbol{p}} = B(\boldsymbol{p}, \zeta, \boldsymbol{\gamma})\; {}^c\Delta\boldsymbol{V} \tag{1}$$

where matrix B is a function of the image point, its associated depth $\zeta\colon \mathbb{R}^2 \times \mathbb{R}^g \to \mathbb{R}$ s.t. $\zeta(\boldsymbol{p}, \boldsymbol{\gamma}) = {}^cZ$, and camera parameters. By choosing the $2N$-vector $\boldsymbol{p} = [\boldsymbol{p}_1^T \quad \boldsymbol{p}_2^T \quad \cdots \quad \boldsymbol{p}_N^T]^T$ of image point coordinates as the visual state, and the relative twist ${}^c\Delta\boldsymbol{V}$ as the input vector, the dynamics of image appearance is driftless, time-varying

Manuscript received May 26, 1997; revised October 26, 1998. This paper was recommended for publication by Associate Editor S. Hutchinson and Editor V. Lumelsky upon evaluation of the reviewers' comments.

B. Allotta is with PERCRO Lab, PERCeptual RObotics Laboratory, Scuola Superiore Sant'Anna, Pisa I-56127, Italy (e-mail: ben@sssup.it).
C. Colombo is with the Dipartimento di Sistemi e Informatica, Firenze I-50139, Italy (e-mail: columbus@dsi.unifi.it).

Publisher Item Identifier S 1042-296X(99)01211-2.

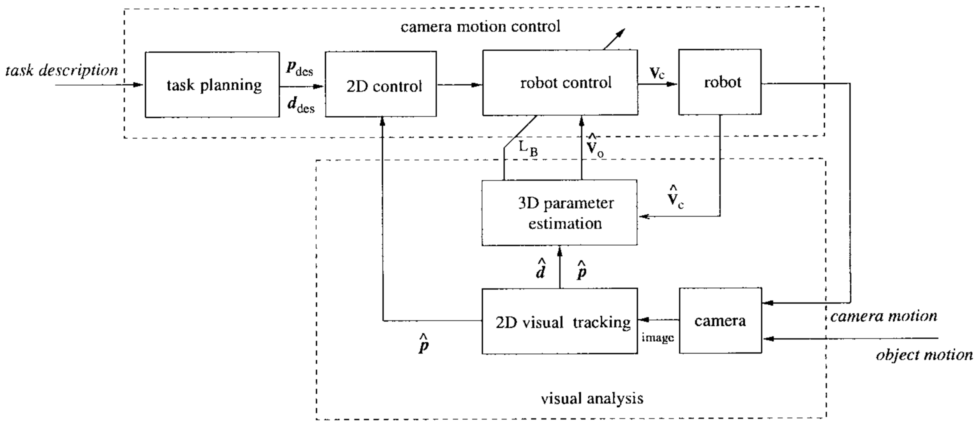


Fig. 1. Block diagram of the image-based visual servoing scheme.

and input-affine

$$\dot{\boldsymbol{p}} = \mathrm{B}(\boldsymbol{p}, t)\ {}^c\Delta\boldsymbol{V} \tag{2}$$

where $\mathrm{B}(\boldsymbol{p}, t) = [B^T(\boldsymbol{p}_1, t) \quad \cdots \quad B^T(\boldsymbol{p}_N, t)]^T$ is a $2N \times 6$ matrix, whose explicit dependence on t encodes the dependence on ζ and $\boldsymbol{\gamma}$.

Image-based control requires that image points $\boldsymbol{p}_i$ (or any other kind of features used) be tracked, and other parameters related to object pose and shape be estimated. Specifically, the structure of B depends on camera model π and object depth ζ.

B. Visual Servoing with Affine Interaction Models

Linear approximations of camera-object interaction rely on the assumption that object depth be a linear function of image coordinates. Simple linear approximations of perspective projection are the *weak perspective* and *para-perspective* camera models [13]. Embedding in (1) the first order Taylor expansion of depth $\zeta(\boldsymbol{p}, \boldsymbol{\gamma}) \approx \zeta_b + [\partial\zeta/\partial\boldsymbol{p}]|_{\boldsymbol{p}_b, \gamma} \cdot (\boldsymbol{p} - \boldsymbol{p}_b)$ yields an image velocity $\dot{\boldsymbol{p}}$ which is linear in $\boldsymbol{p}$

$$\dot{\boldsymbol{p}} = \dot{\boldsymbol{p}}_b + M_b\,(\boldsymbol{p} - \boldsymbol{p}_b) \tag{3}$$

being $\boldsymbol{p}_b$ the imaged object centroid or any other well trackable point, and M_b a 2×2 time-varying matrix independent of $\boldsymbol{p}$. As a consequence of (3), system dynamics of (2) can be synthetically expressed as

$$\boldsymbol{d} = \mathrm{L}_b\ {}^c\Delta\boldsymbol{V} \tag{4}$$

where $\boldsymbol{d} \in \mathbb{R}^6$, and L_b is a time-varying 6×6 matrix which is independent of system state, and expressing a one-to-one correspondence between two-dimensional (2-D) appearance evolution and 3-D relative motion of camera and object.

Once the control action $\boldsymbol{d} = g(\boldsymbol{p}_{des}, \boldsymbol{p})$ has been synthesized, an appropriate generalized inverse of L_b is used to generate relative motion. Since the size of $\boldsymbol{d}$ is constant and small regardless of state dimension, control complexity is decoupled from image tracking complexity (which still depends, of course, on the number N of tracked points).

Possible choices for $\boldsymbol{d}$ are linear combinations of the components of $\dot{\boldsymbol{p}}_b = [u_b \quad v_b]^T$ and of the entries of the image velocity Jacobian

$$M_b = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix}$$

using

$$\boldsymbol{d} = [u_b \quad v_b \quad u_x + v_y \quad v_x - u_y \quad u_x - v_y \quad v_x + u_y]^T \tag{5}$$

and the paraperspective camera model, the structure of L_b is shown in (6) at the bottom of the page, where λ is the focal length, p and q are the components (camera coordinates) of the depth gradient evaluated at $({}^cX_b, {}^cY_b)$, and $c = \zeta(\boldsymbol{0}, \gamma)$ is s.t.

$$c/\zeta_b = (\lambda - p\,x_b - q\,y_b)/\lambda. \tag{7}$$

III. System Implementation

Fig. 1 shows the basic blocks of a control scheme embodying the concepts of the proposed framework in a eye-in-hand robotic system. The basic components of the camera motion control block are a task planner, an image-level controller, and a robot controller producing 3-D motion commands for the robot manipulator based on a 2-D control signal. Visual analysis is devoted to track the object image appearance and to estimate on-line 3-D parameters relevant for control.

$$L_b = \begin{bmatrix} -\dfrac{\lambda}{\zeta_b} & 0 & \dfrac{x_b}{\zeta_b} & 0 & -\left(\dfrac{\lambda^2 + x_b^2}{\lambda}\right) & y_b \\ 0 & -\dfrac{\lambda}{\zeta_b} & \dfrac{y_b}{\zeta_b} & \left(\dfrac{\lambda^2 + y_b^2}{\lambda}\right) & 0 & -x_b \\ \dfrac{p}{c} & \dfrac{q}{c} & \dfrac{2}{c} - 3\dfrac{p}{c}\dfrac{x_b}{\lambda} - 3\dfrac{q}{c}\dfrac{y_b}{\lambda} & -3\dfrac{y_b}{\lambda} & 3\dfrac{x_b}{\lambda} & 0 \\ -\dfrac{q}{c} & \dfrac{p}{c} & -\dfrac{q}{c}\dfrac{x_b}{\lambda} + \dfrac{p}{c}\dfrac{y_b}{\lambda} & \dfrac{x_b}{\lambda} & \dfrac{y_b}{\lambda} & -2 \\ \dfrac{p}{c} & -\dfrac{q}{c} & \dfrac{p}{c}\dfrac{x_b}{\lambda} - \dfrac{q}{c}\dfrac{y_b}{\lambda} & \dfrac{y_b}{\lambda} & \dfrac{x_b}{\lambda} & 0 \\ \dfrac{q}{c} & \dfrac{p}{c} & \dfrac{q}{c}\dfrac{x_b}{\lambda} + \dfrac{p}{c}\dfrac{y_b}{\lambda} & -\dfrac{x_b}{\lambda} & \dfrac{y_b}{\lambda} & 0 \end{bmatrix} \tag{6}$$

A. Visual Analysis and Pose Disambiguation

We consider as image state $\boldsymbol{p}$ a set of N spline control points $\boldsymbol{p}_i$, which are used by the 2-D visual tracking module to represent and track object appearance by means of active contours [14]. A quadratic B-spline representation is used for active contours

$$\boldsymbol{p}(s) = \sum_{i=1}^{N} f_i(s)\,\boldsymbol{p}_i \tag{8}$$

where $\boldsymbol{p}(s),\ s \in [0,\ 1]$ is the generic contour point, and the f_i's are the spline blending functions; such a representation allows to compactly encode object image shape—small values of N for a fixed shape complexity—and optimize image processing. Point $\boldsymbol{p}_b$ is taken here as the contour centroid. Apart from providing at any time a robust estimate of image state based on a Kalman filter, the tracking module also produces an on-line estimate of the state evolution $\boldsymbol{d}$ defined in (5): this is accomplished through a temporal analysis of affine contour deformations [15].

Control relies on the matrix L_b defined in (6), whose entries are initialized and estimated on-line by assuming a weak perspective camera model. As shown in [16], such a model depends on a few camera parameters, namely λ (intrinsic, and assumed to be known), and p, q, c, and ζ_b (extrinsic). At startup, pose and distance extrinsic parameters are estimated as expounded in [16] via a least squares match of the current image appearance against a reference object appearance. In particular, pose computation takes place in two steps. First, a solution affected by the well-known reversal ambiguity (with any linear camera model, so with weak perspective, the same visual appearance is shared by pairs of object poses) is obtained. Second, the disambiguation strategy proposed in [13] is used by choosing as solution, of the two ambiguous poses found, the one which better fits raw image data into a nonlinear, full perspective camera model. (Notice that alternative methods exist for computing the pose of planar objects directly under full perspective—see, e.g., [17] and references.) At run-time, extrinsic parameters are obtained as in [15] by coupling together an on-line estimator and a first-order predictor. Specifically, the estimator relies on explicitly embedding the weak perspective constraint $x_b = y_b = 0$ in (6). This is combined with (4) and (5), to obtain the vector $\boldsymbol{z}(p,\,q,\,c) = [p/c \quad q/c \quad 1/c]^T$ as the solution of the linear system

$$\begin{bmatrix} \Delta T_x & \Delta T_y & 2\Delta T_z \\ \Delta T_x & -\Delta T_y & 0 \\ \Delta T_y & \Delta T_x & 0 \end{bmatrix} \boldsymbol{z} = \begin{bmatrix} u_x + v_y \\ u_x - v_y \\ v_x + u_y \end{bmatrix} \tag{9}$$

from which p, q, and c are soon obtained, and used in (7) to evaluate the remaining extrinsic parameter ζ_b.

As shown in [9], the accuracy of camera parameters estimates affects the speed of convergence; yet, as it will be clear in Section IV, even a rough estimate of camera parameters is sufficient in our system.

B. Control Strategy

In image-based servoing, a visual task can be described in terms of a desired evolution $\dot{\boldsymbol{p}}_{des}(t)$ of object appearance. According to our formulation, the task is represented in a synthetic way by $\boldsymbol{d}_{des}(t)$. The 2-D controller computes a 6-D error $\boldsymbol{\epsilon}(\boldsymbol{p}_{des},\,\boldsymbol{p})$ between the desired and estimated image states evaluated in a least squares sense as shown in [15]. Such an error is used by the 3-D controller to generate a robot velocity twist reference command. The manipulator Jacobian is assumed to be known with sufficient accuracy to use the Cartesian motion vector in the place of the joint motion vector for control: besides, the closure of a feedback loop at the image level ensures that even inaccuracies in the knowledge of the robot kinematics can be largely compensated by the control system. The presence of singularities in the kinematic structure holding the camera also has to be considered and taken into account. For instance, if the camera is mounted on the end-effector of a robot manipulator, in some configurations the singularities of the manipulator Jacobian may cause the impossibility of producing the desired camera motion.

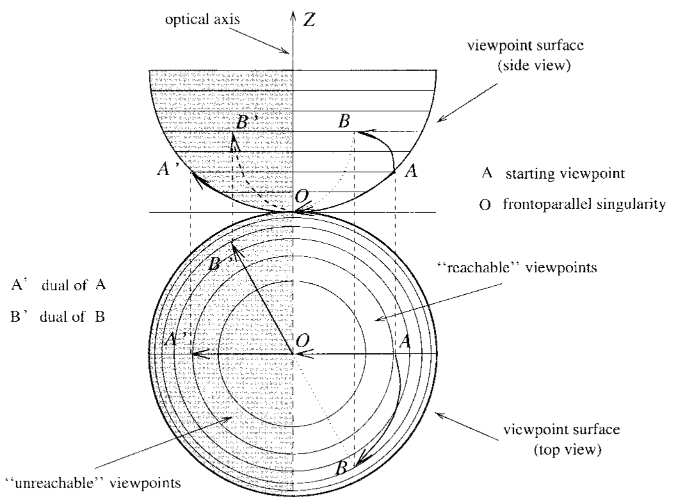


Fig. 2. Viewpoint surface, pose ambiguity, and frontoparallel singularity.

The following control law, based on the interaction model (4) and using both feedforward (planning, $\boldsymbol{d}_{des}$) and feedback (visual analysis, $\boldsymbol{\epsilon}$) information is adopted

$${}^{c}\Delta\boldsymbol{V} = \mathrm{L}_b^{\dagger}\,(\boldsymbol{d}_{des} + \mathrm{K}\boldsymbol{\epsilon}) \tag{10}$$

where K is a 6 × 6 diagonal matrix of feedback gains. In order to cope with moving objects, an estimate $\hat{\boldsymbol{V}}_o$ of object velocity is also required, which is obtained as

$$\hat{\boldsymbol{V}}_o = \hat{\boldsymbol{V}}_c - \mathrm{L}_b^{\dagger}\,\hat{\boldsymbol{d}} \tag{11}$$

$\hat{\boldsymbol{V}}_c$ being derived from manipulator proprioceptive sensor information and $\hat{\boldsymbol{d}}$ from visual analysis. The required velocity setpoint for the camera motion is finally computed as $\boldsymbol{V}_c = {}^{c}\Delta\boldsymbol{V} + \hat{\boldsymbol{V}}_o$.

C. Planning and Task Disambiguation

The planner is in charge for generating a smooth trajectory for the desired visual appearance of the object toward the goal one. Our planner module generates a trajectory for each of the control points, which is polynomial (degree $h \geq 1$) in time and linear in the image space. Smooth trajectories are obtained, e.g., with cubic ($h = 3$) and quintic ($h = 5$) polynomials [18].

The pose ambiguity determined by the use of linear approximations to camera-object interaction has to be explicitly considered and solved also at the planner level. Let us introduce the *viewpoint surface* as the semisphere whose points (corresponding one-to-one to the innerwise unit normals) represent all possible relative orientations of the object plane—i.e., the plane which fits best the object's visible surface—w.r.t. the camera (Fig. 2). Any continuous change of orientation corresponds to a curvilinear path on this surface. The polynomial planner automatically selects as goal viewpoint, of the two dual solutions which reflect the pose ambiguity for any given and 2-D goal appearance, the one which is closest to the initial 3-D viewpoint moving along a geodesic path of the viewpoint surface. For instance, let $\boldsymbol{A}$ be the initial viewpoint, and $\boldsymbol{A}'$ be its dual, placed at its opposite side. It is evident that $\boldsymbol{A}'$ cannot be reached from $\boldsymbol{A}$ via only one polynomial planning step (the initial and goal 2-D appearances do actually coincide) and that the relative orientation will not change.

In order to reach $\boldsymbol{A}'$, we augment the planning strategy, by splitting the path in two parts, namely $\boldsymbol{AO}$ and $\boldsymbol{OA}'$, where $\boldsymbol{O}$ represents a frontoparallel view of the object. Hence, in the general case, given $\boldsymbol{A}$ and a goal visual appearance corresponding to the two dual views $\boldsymbol{B}$ and $\boldsymbol{B}'$, augmented planning proceeds as follows:

1) determine the final viewpoint, and establish whether it is "reachable" $(\boldsymbol{B})$ or "unreachable" $(\boldsymbol{B}')$ via a single polynomial planning step;
2) in the first case, plan $\boldsymbol{AB}$ and execute;
3) in the second case, split planning and execution into the two steps $\boldsymbol{AO}$ and $\boldsymbol{OB}'$.

Visualizing smooth pose changes as trajectories on the viewpoint surface also provides an insight into the problem of kinematic singularities and how to avoid them. It is easy to show that with our formulation the frontoparallel pose $\boldsymbol{O}$ corresponds to the only algorithmic singularity, in that the determinant of L_b vanishes identically iff $p = q = 0$. In order to avoid arbitrarily large desired camera velocities during a two-step planning strategy, a "forbidden" region $\mathcal{F}$ is defined on the viewpoint surface around $\boldsymbol{O}$, inside which $\sqrt{p^2+q^2}$ is small enough to assume that L_b be singular. When the system enters $\mathcal{F}$, the loop is opened thus allowing to cross $\boldsymbol{O}$ without falling into singularity. As a case study, consider positioning the camera w.r.t. the object from an initial pose $\boldsymbol{B}$: $p = q = 0.01$ to its dual pose $\boldsymbol{B}'$: $p = q = -0.01$ as the imaged centroid remains at the image origin ($\boldsymbol{p}_b^{des} = \mathbf{0}$). Let the relative distance and focal length be respectively $\zeta_b = c = 400$ mm and $\lambda = 8$ mm, and the border of the forbidden region be the parallel on the interaction surface with radius $0.0001\sqrt{2}$. Then the loop is opened as the pose $\boldsymbol{B}_{\mathcal{F}}$: $p = q = 0.0001$ is reached, and closed again at $\boldsymbol{B}'_{\mathcal{F}}$: $p = q = -0.0001$. In either case, the velocity twist (in mm/s and rad/s) equals ${}^{c}\Delta\boldsymbol{V} = [2000\ \ -2000\ \ -0.0\ \ -5.0\ \ -5.0\ \ 0.0005]^T$, so that the relative velocity twist is maintained without discontinuity. During the loop opening the system continues to estimate the 3-D parameters and in particular p and q, in order to detect the breakthrough condition $\sqrt{p^2+q^2} > 0.0001\sqrt{2}$ allowing to reactivate the loop closure.

IV. Experiments

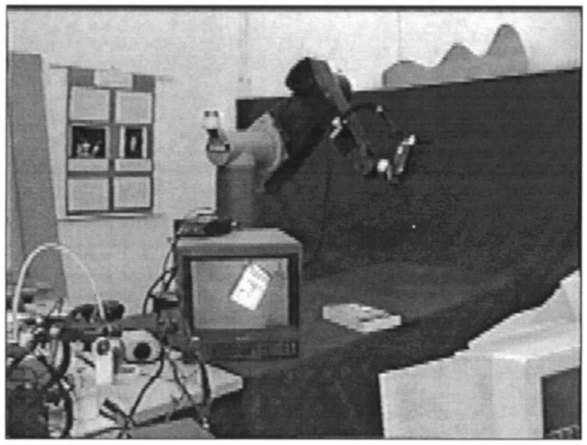
In this Section, results are presented for two sets of experiments involving a camera being positioned w.r.t. still planar and non planar objects between the same initial and goal configurations of the robot arm (see Fig. 3).

A. System Setup and Operation

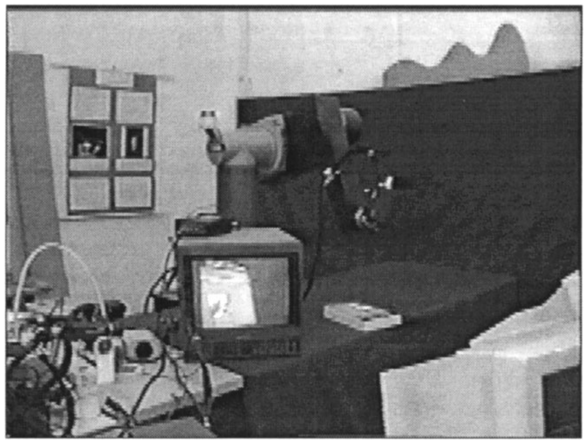
1) Hardware and Software: The hardware setup for eye-in-hand experiments consists of a PUMA 560 manipulator equipped with a wrist-mounted Sony camera and commanded through a MARK III controller, and a PC 486/66 MHz equipped with an Imaging Technology frame grabber (Fig. 4). The MARK III controller runs VAL II programs and communicates with the PC via the ALTER real-time protocol using an RS-232 serial interface. Due to the computational burden of tracking algorithms, a multirate real-time control was implemented. New velocity setpoints are generated by the PC with a sampling rate $\tau_2 = K\tau_1$, where $\tau_1 = 28$ ms is the sampling rate of the ALTER protocol, and K is an integer which depends on the dimension of the state vector N. The visual computation time using $N = 10$ control points for contour tracking is less than 100 ms; hence, K is set to 4. A suitably fast communication process maintains the handshaking with the MARK III controller by communicating the most recent velocity twist setpoint generated by the high level—and slower—process via a mailbox.

2) Parameter Setting and Initialization: Camera optics datasheets provide a raw value for focal length and pixel dimensions; the remaining intrinsic parameters of the camera are ignored. Infinite impulse response digital filters are used for smoothing sensory data and enhance the quality of all visual measurements (visual state, object motion). Filter and feedback gains are tuned experimentally. The task planning module uses cubic trajectories, which provide a good tradeoff between smoothness and contour inertia.

Operation with the system starts by bringing the robot in its goal configuration, and initializing—by means of a computer mouse—an active contour in the image, which automatically locks on the goal image appearance. The locked contour position (centroid coordinates) and shape are recorded, and the robot is moved to the initial configuration while the active contour tracker keeps itself locked to the time-changing object appearance. The matrix L_b of (6) is initialized "by hand" based on a visual evaluation of distances and angles, thus providing a coarse estimate of p, q, and c with an



(a)



(b)

Fig. 3. (a) Initial and (b) goal configurations for the first set of experiments. The monitor displays the scene as seen by the wrist-mounted camera.

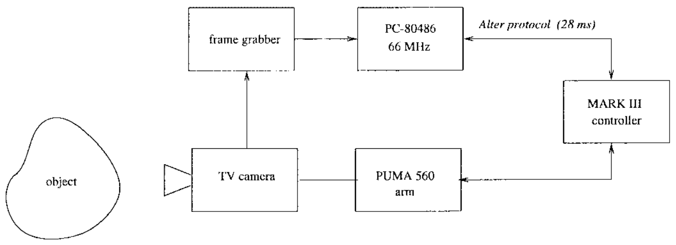


Fig. 4. System setup and communications.

TABLE I
GOAL AND REACHED CONFIGURATIONS (ENCODER MEASUREMENTS) OF THE ROBOT END-EFFECTOR AFTER 110 s FOR EXPERIMENTS 1, 2, AND 3. THE QUANTITIES "o," "a," AND "t" ARE THE ORIENTATION, ALTITUDE, AND TOOL ANGLES (DEGREES), RESPECTIVELY, USED BY VAL II TO REPRESENT THE END-EFFECTOR ORIENTATION. END-EFFECTOR TRANSLATIONS "x," "y," AND "z" ARE IN MILLIMETERS

	x	y	z	o	a	t
goal	5.0	755.7	-263.7	98.6	57.8	-153.0
regulation (1)	33.7	768.7	-244.3	96.2	63.4	-156.7
servoing (2)	26.7	759.1	-248.4	99.0	61.9	-153.6
planning (3)	18.9	758.6	-253.5	99.0	60.5	-154.2

error up to 50% w.r.t. the ground truth values. Besides, in the experiments presented, it was deliberately chosen not to update L_b at run-time, so as to provide difficult operating conditions for the system—the speed of convergence is significantly reduced without on-line parameter estimation, as expected and as confirmed by experimental evidence—and prove the robustness of the control scheme. It is worth noting that uncertainty in the estimate of L_b and finite resolution of robot velocity commands may affect steady state accuracy.

B. Discussion of Experiments

1) Experiments 1–3—Control Modes: The system can be run in diverse control modes, according to the planning information it uses. In the general case, planned trajectories are specified both in terms of desired state ($\boldsymbol{p}_{des}$) and of differential evolution ($\boldsymbol{d}_{des}$).

In the following experiments, for a fixed task—positioning the camera w.r.t. a book upon a table—the system is run in three different modes. This allows us to gain an insight into system behavior and performance.

In experiment 1, the system is run in *output regulation mode*: no planning is provided ($\boldsymbol{p}_{des}(t) = \boldsymbol{p}^{goal}\ \forall t$ and $\boldsymbol{d}_{des} = \boldsymbol{0}$), thus forcing the system to rely only on feedback. This experiment is useful to assess the stability of control, and tune up the feedback gains so as to obtain a slightly underdamped behavior. Since the feedback error depends on the mismatch between the current and goal states, which can be large, high feedback gains should be avoided in this mode, as they could cause system instability due to the presence of various sources of inaccuracy. The final desired 3-D configuration is reached with a considerable error, also due to the fact that, although the robot is commanded in velocity, the resolution of the velocity command is limited to 16 b, so that any velocity command under threshold is truncated to zero and does not produce any motion of the manipulator. Specifically, in this experiment the magnitude of the manipulator position error evaluated from the data of Table I is 37.0 mm.

The *servoing mode* (characterized by $\boldsymbol{d}_{des} = \boldsymbol{0}$ and a not constant $\boldsymbol{p}_{des}(t)$) is used in experiment 2 to assess the tracking performance of the control scheme as the system is forced to compensate by feedback the tracking error $\boldsymbol{\epsilon}(\boldsymbol{p}_{des}(t), \hat{\boldsymbol{p}}(t))$. The presence of trajectory planning keeps the error small; hence a stable behavior is obtained even by using higher gains than before. This avoids the presence of large errors in the final 3-D configuration which, as a typical performance, is reached with an error of within a few degrees (orientation) and millimeters (translation). In this experiment, after 110 s, the magnitude of the position error is 26.7 mm.

Experiment 3 is executed in *planning mode* (i.e., the normal mode), featuring the simultaneous use of trajectory planning, feedforward, and feedback control (Fig. 5). Results in terms of errors during execution are expected to be smaller than in the previous case, due to the presence of the feedforward control based on the desired trajectory $\boldsymbol{d}_{des}$. Indeed, after 110 s, the magnitude of the position error is 17.5 mm, with a significant improvement w.r.t. both the servoing and ouput regulation modes.

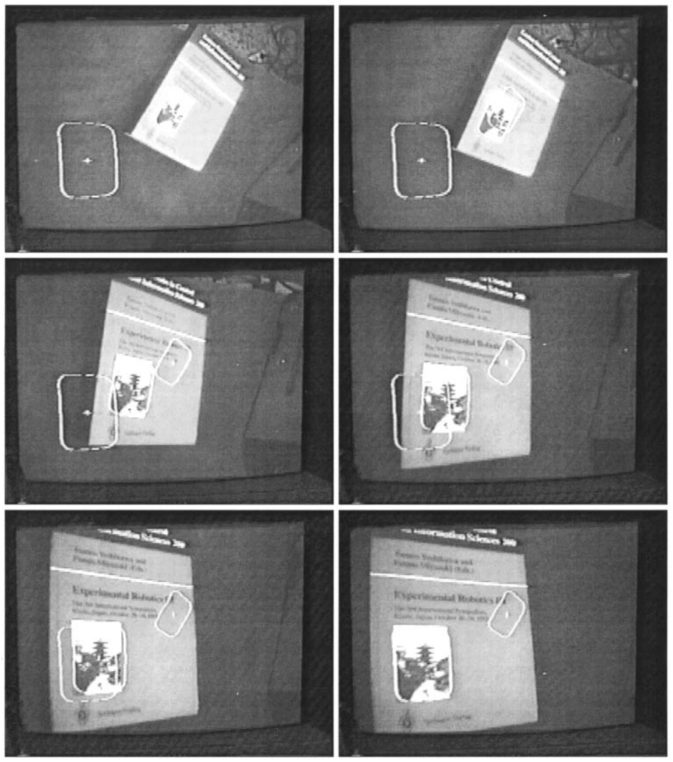
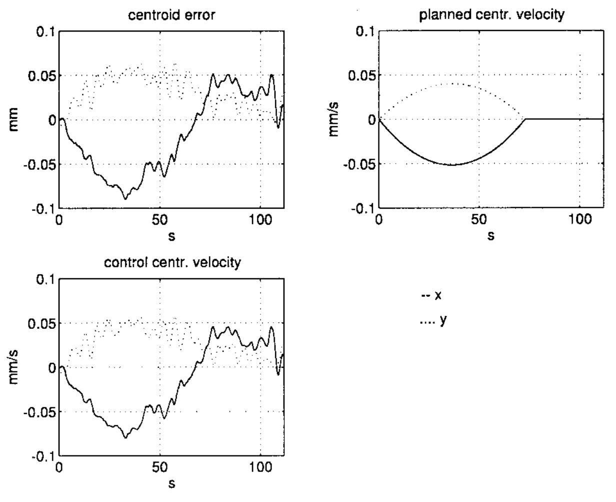


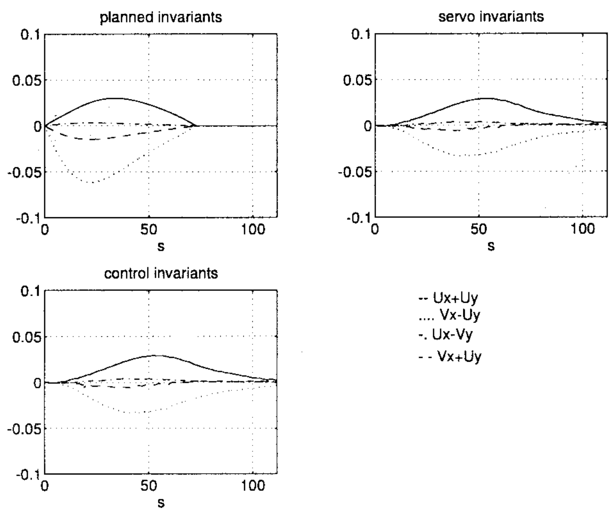
Fig. 5. Positioning in planning mode: image plane. The initial, current, and goal contours are shown in each frame of the sequence.

Figs. 6 and 7 allow a further performance comparison between the servoing and planning modes. The data reported in the figures are the first two (centroid) components and the last four (shape) components of the three vectors $\boldsymbol{d}$, $\boldsymbol{d}_{des}$, and $\boldsymbol{\epsilon}(\boldsymbol{p}_{des}, \boldsymbol{p})$ appearing in (10), and the translational and angular camera velocities. The image centroid error is in both cases always below one tenth of millimeter, i.e., less than 1% of the linear dimension of the CCD imaging area. The desired object appearance is reached with a negligible error within about 75 s in planning mode, while about 110 s are required in servoing mode: this demonstrates the effectiveness of adding feedforward to the control, as the job of the feedback is significantly alleviated and the tracking lag reduced. Concerning velocities, cubic-based planning, and effective tracking in both modes contribute to obtain relative velocity and acceleration profiles with a negligible chattering. Yet, notice that the feedforward term (which is dropped out after about 75 s in experiment 3) makes planning mode more sensitive than servoing mode to the accuracy of camera extrinsic parameters. In summary, despite the absence of run-time 3-D parameters updating, both the planning and servoing modes exhibit a satisfactory performance (see again Table I). A tradeoff is also demonstrated for the two operating modes, between speed of convergence and sensitivity to extrinsic parameters estimation.

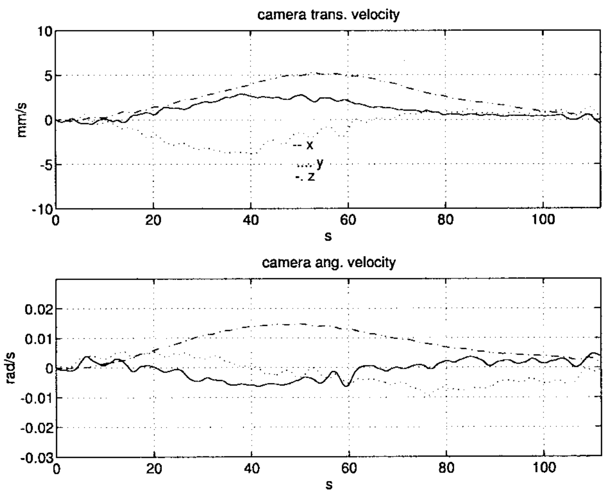
2) Experiments 4–6—Planar versus Nonplanar Objects: A second set of experiments is devoted to illustrate the dependence of system performance on the object planarity condition. To this aim, the system



(a)

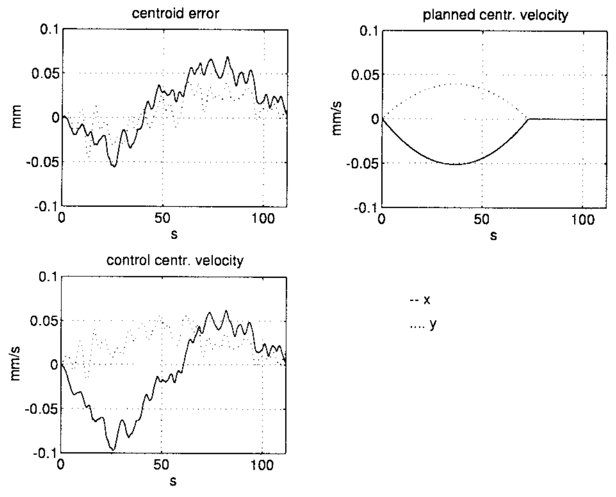


(b)

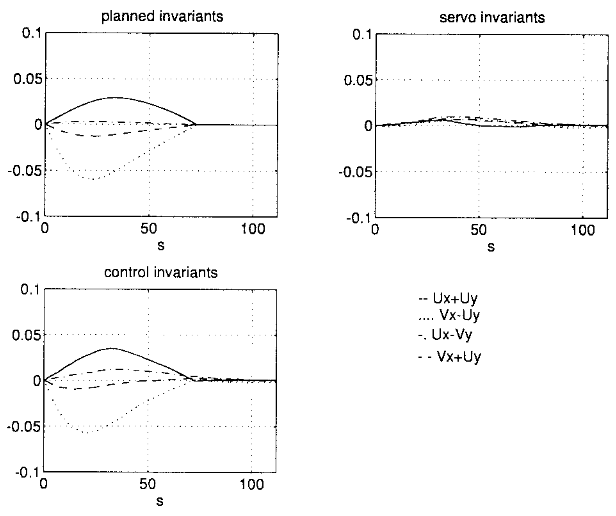


(c)

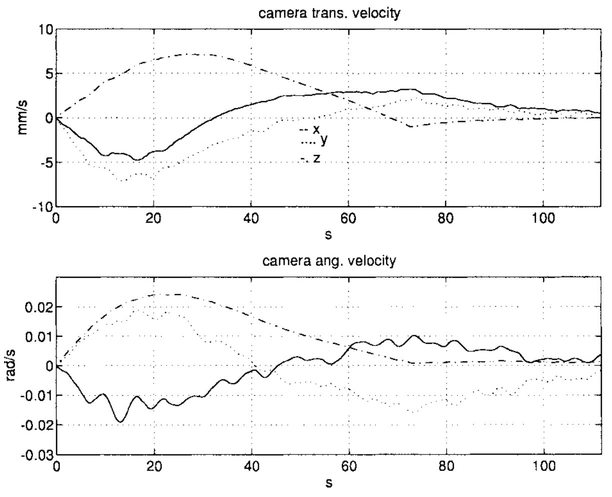
Fig. 6. Servoing mode: (a) contour centroid error, (b) contour shape error, and (c) camera velocities.



(a)



(b)



(c)

Fig. 7. Planning mode: (a) contour centroid error, (b) contour shape error, and (c) camera velocities.

is run to achieve—in planning mode—the same positioning task in the presence of a planar object (a computer diskette) and a nonplanar object (the shoe part of Fig. 8).

Fig. 9(a) shows the plots of 2-D control quantities and 3-D velocities obtained with the planar object, and row 2 of Table II gives the final positioning mismatch for experiment 4. In this experiment,

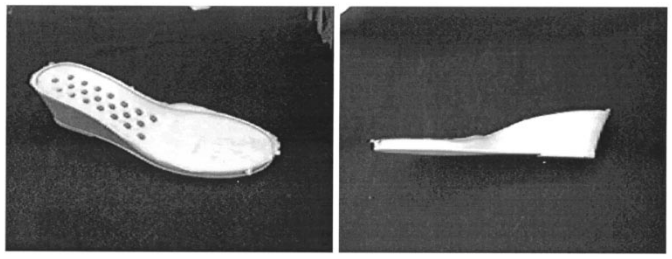


Fig. 8. Two views of the nonplanar object used in experiment 5.

TABLE II
GOAL AND REACHED CONFIGURATIONS (ENCODER MEASUREMENTS) OF THE ROBOT END-EFFECTOR AFTER 144 s FOR EXPERIMENTS 4, 5, AND 6. ROW 2 IS OBTAINED WITH THE PLANAR OBJECT AND A SET GAINS LARGE ENOUGH TO LEAD THE SYSTEM TO INSTABILITY WITH THE NONPLANAR OBJECT. ROWS 3 AND 4 ARE OBTAINED WITH THE NONPLANAR AND PLANAR OBJECTS, RESPECTIVELY, USING SMALLER GAINS

	x	y	z	o	a	t
goal	-108.8	733.9	-66.4	90.1	56.9	179.9
planar (4)	-125.8	746.6	-64.8	87.9	55.4	177.8
non planar (5)	-175.7	770.4	-102.7	83.3	49.2	179.1
planar (6)	-148.3	753.4	-63.4	86.9	53.4	177.2

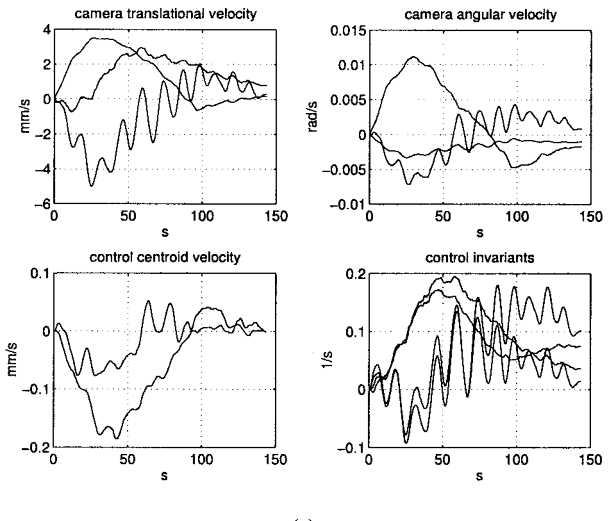
control gains were tuned so as to obtain a quite underdamped behavior. The same, critical gains provided an unstable behavior when attempting to replicate experiment 4 with the nonplanar object. After a new set of (smaller) stable gains was found, a worse system performance was observed (experiment 5) with the nonplanar object—see Fig. 9(b) and row 3 in Table II. The set of gains used in experiment 5 was used again with the planar object (experiment 6), with a resulting intermediate performance—Fig. 9(c) and row 4 in Table II.

The results of experiments 4–6 indicate that the affine servoing approach can be successfully applied to nonplanar objects, provided that control gains be suitably lowered w.r.t. the planar object case. The modeling error caused by the presence of nonplanar objects also produces a slight performance degradation for any given set of stable gains tuned for the planar case.

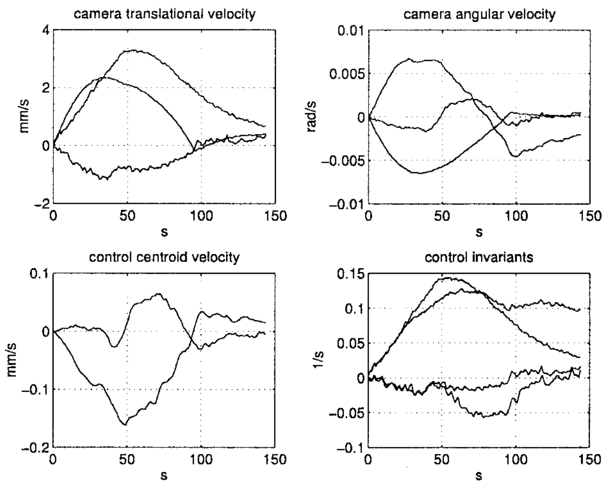
V. CONCLUSION

The use of linear approximations to simplify both state description and control interaction in image-based visual servoing was investigated. A way to cope with the intrinsic limitations of linear interaction models was expounded. An eye-in-hand implementation featuring a PUMA arm and embodying the linearized framework was also described and discussed. The system includes modules for visual analysis, planning, and control; objects are represented and tracked by means of active contours. Experimental results demonstrating the behavior of the system at work were shown. The method allows positioning w.r.t, planar and nonplanar objects as well, although in the latter case at the expense of a diminished accuracy and stability of the system.

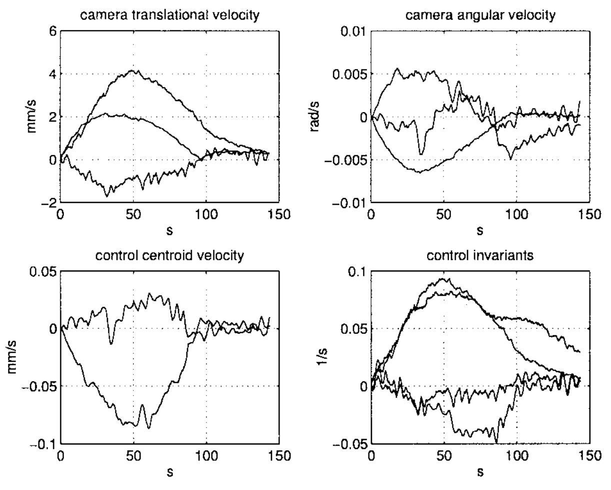
The main contribution of the work is a formulation that makes it easy and intuitive the problem of specifying, planning, and controlling an eye-in-hand robotic system that must perform positioning tasks. Future study will concern the possibility of using the same approach in mobile robotics, where additional problems arise due to the presence of constraints in the motion of a camera mounted on a nonholonomic mobile base. Another direction of investigation is the improvement of the approach to resolve the disambiguation problem in a more efficient way.



(a)



(b)



(c)

Fig. 9. Camera velocities and control signals for experiments (a) 4 (planar object, critical gains), (b) 5 (nonplanar object, small gains), and (c) 6 (planar object, small gains).

ACKNOWLEDGMENT

The authors gratefully acknowledge F. Conticelli for his help during the experiments and the anonymous referees for their comments and suggestions.

References

[1] P. I. Corke, "Visual control of robot manipulators—A review," in *Visual Servoing,* K. Hashimoto, Ed. Singapore: World Scientific, 1993, pp. 1–31.

[2] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.,* vol. 12, pp. 651–670, Oct. 1996.

[3] G. C. Buttazzo, B. Allotta, and F. P. Fanizza, "Mousebuster: A robot for real-time catching," *IEEE Contr. Syst. Mag.,* vol. 14, no. 1, pp. 49–56, 1994.

[4] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Trajectory filtering and prediction for automated tracking and grasping of a moving object," in *Proc. 1992 IEEE Int. Conf. Robot. Automat. ICRA'92,* Nice, France, 1992, pp. 1850–1856.

[5] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robot. Automat.,* vol. RA-3, pp. 404–417, Oct. 1987.

[6] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision," *IEEE Trans. Robot. Automat.,* vol. 9, pp. 14–35, Feb. 1993.

[7] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.,* vol. 8, pp. 313–326, June 1992.

[8] A. J. Koivo and N. Houshangi, "Real-time vision feedback for servoing robotic manipulator with self-tuning controller," *IEEE Trans. Syst., Man, Cybern.,* vol. 21, pp. 134–142, Jan./Feb. 1991.

[9] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *Proc. 3rd Int. Symp. Experimental Robot. ISER'93,* Kyoto, Japan, 1993, pp. 187–193.

[10] N. P. Papanikolopoulos and P. K. Khosla, "Adaptive robotic visual tracking: Theory and experiments," *IEEE Trans. Automat. Contr.,* vol. 38, no. 3, pp. 429–445, 1993.

[11] R. Cipolla and N. J. Hollinghurst, "Visually guided grasping in unstructured environments," *Robot. Auton. Syst.,* vol. 19, no. 3/4, pp. 337–346, 1997.

[12] R. Cipolla and A. Blake, "Image divergence and deformation from closed curves," *Int. J. Robot. Res.,* vol. 16, no. 1, pp. 77–96, 1997.

[13] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, "Object pose: The link between weak perspective, paraperspective, and full perspective," *Int. J. Comput. Vision,* vol. 22, no. 2, pp. 173–189, 1997.

[14] A. Blake, R. Curwen, and A. Zisserman, "A framework for spatiotemporal control in the tracking of visual contours," *Int. J. Comput. Vision,* vol. 11, no. 2, pp. 127–145, 1993.

[15] C. Colombo, B. Allotta, and P. Dario, "Affine visual servoing: A framework for relative positioning with a robot," in *Proc. 1995 IEEE Int. Conf. Robot. Automat. ICRA'95,* Nagoya, Japan, 1995, pp. 464–471.

[16] C. Colombo and B. Allotta, "Image based robot task planning and control using a compact visual representation," in *IEEE Trans. Syst., Man, Cybern.,* vol. 29, no. 1, pp. 92–100, 1999.

[17] R. M. Haralick, C. N. Lee, K. Ottenberg, and M. Nolle, "Review and analysis of solutions of the 3-point perspective pose estimation problem," *Int. J. Comput. Vision,* vol. 13, no. 3, pp. 331–356, 1994.

[18] M. Brady, "Trajectory planning," in *Robot Motion: Planning and Control,* M. Brady *et al.,* Eds. Cambridge, MA: MIT Press, 1982, pp. 221–243.

On the Trajectory Planning of a Planar Elastic Manipulator Under Gravity

Pritam Kumar Sarkar, Motoji Yamamoto, and Akira Mohri

***Abstract*—Feedforward control design of flexible robot can take advantage of planning algorithm. In this paper two such algorithms have been presented—one is for static case and the other is for dynamic case. Both the algorithms are iterative in nature and are computationally based on the minimization of position or tracking error defined at the end-effector level. The algorithms are simple but computationally expensive. The performance is tested on a planar two-link arm through simulation and the results show the capability of the methods to compensate error due to deflection of the flexible links at the time of static condition of positioning and to track a given trajectory effectively at the time of dynamic condition of tracking under end kinematic constraints.**

***Index Terms*—Elastic manipulator, tracking error minimization, trajectory planning, under gravity.**

I. Introduction

A key issue in the feedforward compensation is the computation of actuator torques that are required for an arm to track a specified trajectory. In case of rigid link the computation is straightforward whereas in case of flexible link it is complex because of the link deformations during motion. One of the early works has been reported by Bayo *et al*. [2]. The work has been based on solving the inverse dynamic equations in the frequency domain for a given end point acceleration profile. The authors have also pointed out the noncausal behavior of the flexible arm. However, the technique results in a large time penalty at the start as well as at the end of motion thus increasing the travelling time considerably. Another important point is that since the method is based on nominal joint motion the tip accuracy may not be very high. In fact, as the flexibility of the arm increases the accuracy at the tip decreases. Asada and Ma [1] have proposed a technique that considers assumed mode model for a general n-link case and solves the inverse dynamic problem by using a special moving coordinate systems, called virtual rigid link coordinates.

On the other hand, when a distributed parameter system is forced at one point and its response is measured at another point, the system is said to be noncollocated. Cannon and Schmitz [3] have utilized a noncollocated controller to actively control both the rigid-body angle and the vibration of the flexible system. To avoid nonminimum phase dynamics, other researchers [6], [9] have considered the tracking point that is shifted from the end point to a point where the internal stability is preserved. Therefore, in these results, even if the desired tracking is realized at the tracking point, there is always a possibility that the error is caused at the end point. Acceleration feedback has been used by Kotnik *et al*. [5] and the work has been compared with the more conventional end point position feedback. Eisler *et al*. [4] have presented a more complicated recursive quadratic programming, coupled with a homotopy method, to generate approximate minimum time and minimum tracking-error tip trajectories for two-link flexible manipulator movements in

Manuscript received June 15, 1998; revised January 5, 1999. This paper was recommended for publication by Associate Editor J. Wen and Editor A. De Luca upon evaluation of the reviewers' comments.

The authors are with the Department of Intelligent Machinery and Systems, Faculty of Engineering, Kyushu University, Fukuoka 812, Japan.

Publisher Item Identifier S 1042-296X(99)03389-3.