A Position-Based Visual Impedance Control for Robot Manipulators

Vincenzo Lippiello, Bruno Siciliano, and Luigi Villani

Abstract— In this paper, an approach to interaction control of a robot manipulator with a partially known environment is proposed. The environment is a rigid object of known geometry but of unknown and possibly time-varying position and orientation. An algorithm for online estimation of the object pose is adopted, based on visual data provided by a camera as well as on forces and moments measured during the interaction with the environment. This algorithm is embedded into an impedance control scheme, resulting in a position-based visual impedance control. Experimental results are presented for the case of an industrial robot manipulator in contact with a planar surface.

I. INTRODUCTION

Vision and force are two complementary sensing capabilities that can be exploited in a synergic way to enhance the autonomy of a robot manipulator. In fact, thanks to the visual perception, a robot may achieve global information on the surrounding environment that can be used for task planning and obstacle avoidance. On the other hand, the perception of the force applied to the end effector allows adjusting its motion so that the local constraints imposed by the environment during the interaction are satisfied.

In recent years, several approaches where force and vision measurements are combined in the same feedback control loop have been proposed, such as hybrid visual/force control [1], shared and traded control [2], [3] or visual impedance control [4], [5].

The approach adopted in this work is based on the observation that, when the robot moves in free space, a position-based visual servoing strategy [6] can be adopted, where vision is used to estimate the relative pose of the robot with respect to the environment. On the other hand, when the robot comes into contact with the environment, a position-based impedance interaction control strategy can be adopted [7]. The impedance controller may take advantage of the estimation of the geometry of the environment computed online from all the available sensor data, i.e., visual, force and joint position measurements.

The estimation algorithm, based on the Extended Kalman Filter (EKF), is an extension of the visual pose estimation approach proposed in [8] to the case that also force and joint position measurements are used. Remarkably, the same estimation algorithm can be adopted both in free space and during the interaction, simply modifying the measurements set of the EKF. The resulting control scheme can be classified as a position-based visual impedance control.

PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, Via Claudio 21, 80125 Napoli, ITALY {lippiell,siciliano,lvillani}@unina.it
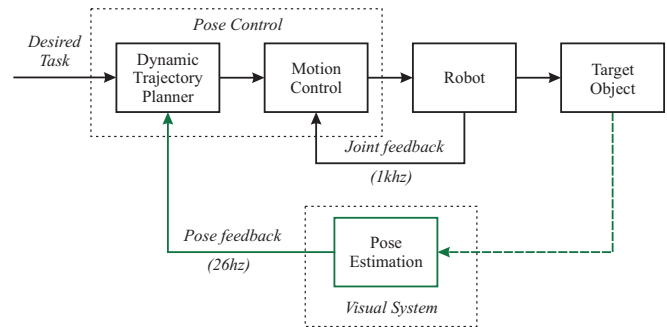
Fig. 1.   Position-based visual servoing.

This scheme was originally proposed in [9], where only preliminary simulation results are shown. In this paper, the position-based impedance control strategy is experimentally tested on a 6-DOF industrial robot equipped with a standard analog camera and a wrist force/torque sensor.

II. POSITION-BASED VISUAL IMPEDANCE

The position-based visual impedance control proposed in this paper can be derived from a typical position-based visual servoing scheme represented in Fig. 1. This scheme requires the estimation of the relative pose of the robot end-effector with respect to a target object by using the vision system; the estimated pose is then fed back to a pose controller.

Notice that pose estimation is a computationally demanding task, because it requires processing of the measurements of some geometric features extracted from the images of one or more cameras. Hence, the frequency of the pose estimation algorithm is usually lower than the frequency of the pose control loop (greater or equal to 500 Hz in standard robot manipulators, to guarantee tracking accuracy and disturbance rejection). The bottleneck is usually represented by the camera frame rate (between 25 Hz and 60 Hz for low-cost cameras used in common applications).

Pose control is performed through an inner-outer control loop. The inner loop implements motion control (independent joint control or any kind of joint space or task space control). In the outer loop, the block named Dynamic Trajectory Planner computes the trajectory for the end-effector on the basis of the current pose of the robot with respect to the object and of the desired task.

If the robot interacts with the target object, an interaction control strategy must be adopted, based on force and moment measurements achieved via a force/torque sensor mounted at the end-effector wrist. In the presence of uncertainties on the object location, one of the most effective interaction control
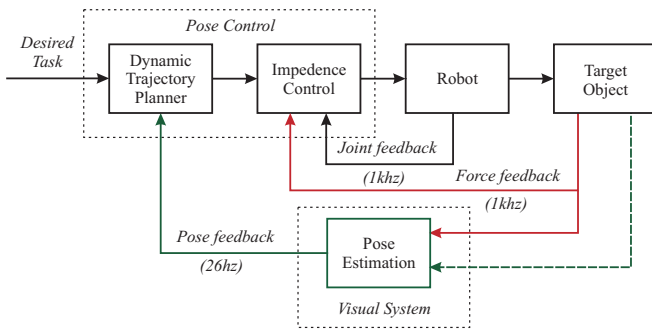
Fig. 2.   Position-based visual impedance control.

strategies is impedance control, which, in addition, is able to manage both contact and free-space motion phases [7].

When the robot is in contact with the target object, the force/torque measurements, as well as the joint position measurements, can be suitably exploited, together with the visual measurements, to improve the accuracy of the estimation of the target object pose.

The resulting control algorithm, represented in Fig. 2, can be classified as a position-based visual impedance control.

A position-based impedance control is adopted, based on the concept of compliant frame [7]. In detail, the Dynamic Trajectory Planner generates a pose trajectory for a desired end-effector frame d specified in terms of the position of the origin $\boldsymbol{p}_d$ and orientation matrix $\boldsymbol{R}_d$. Moreover, a compliant frame r is introduced, specified in terms of $\boldsymbol{p}_r$ and $\boldsymbol{R}_r$. Then, a mechanical impedance between the desired and the compliant frame is considered, so as to contain the values of the interaction force $\boldsymbol{h}$ and moment $\boldsymbol{m}$. In other words, the desired position and orientation, together with the measured contact force and moment, are input to the impedance equation which, via a suitable integration, generates the position and orientation of the compliant frame to be used as a reference for the motion controller of the robot end effector.

As far as the compliant frame is concerned, the position $\boldsymbol{p}_r$ can be computed via the translational impedance equation

$$\boldsymbol{M}_p\Delta\ddot{\boldsymbol{p}}_{dr} + \boldsymbol{D}_p\Delta\dot{\boldsymbol{p}}_{dr} + \boldsymbol{K}_p\Delta\boldsymbol{p}_{dr} = \boldsymbol{h}, \qquad (1)$$

where $\Delta\boldsymbol{p}_{dr} = \boldsymbol{p}_d - \boldsymbol{p}_r$, and $\boldsymbol{M}_p$, $\boldsymbol{D}_p$ and $\boldsymbol{K}_p$ are positive definite matrices representing the mass, damping, and stiffness characterizing the impedance.

The orientation of the reference frame $\boldsymbol{R}_r$ is computed via a geometrically consistent impedance equation similar to (1), in terms of an orientation error based on the (3×1) vector $^r\boldsymbol{\epsilon}_{dr}$, defined as the vector part of the unit quaternion that can be extracted from $^r\boldsymbol{R}_d = \boldsymbol{R}_r^T\boldsymbol{R}_d$. The corresponding mass, damping and inertia matrices are $\boldsymbol{M}_o$, $\boldsymbol{D}_o$ and $\boldsymbol{K}_o$ respectively. More details about the geometrically consistent impedance based on the unit quaternion can be found in [7].

The estimation of the object pose is crucial in order to compute the desired absolute end-effector motion from the desired task assigned in terms of the relative motion with respect to the target object. In the following, the main issues concerning pose estimation are considered in detail.

III. MODELLING

Consider a robot in contact with an object, a wrist force sensor and a camera mounted on the end-effector (eye-in-hand) or fixed in the workspace (eye-to-hand). In this section, some modelling assumptions concerning the object, the robot and the camera are presented.

A. Object

The position and orientation of a frame attached to a rigid object O_o–$x_o y_o z_o$ with respect to a base coordinate frame O–xyz can be expressed in terms of the coordinate vector of the origin $\boldsymbol{o}_o = \begin{bmatrix} x_o & y_o & z_o \end{bmatrix}^T$ and of the rotation matrix $\boldsymbol{R}_o(\boldsymbol{\varphi}_o)$, where $\boldsymbol{\varphi}_o$ is a $(p \times 1)$ vector corresponding to a suitable parametrization of the orientation. In the case that a minimal representation of the orientation is adopted, e.g., Euler angles, it is $p = 3$, while it is $p = 4$ if unit quaternions are used. Hence, the $(m\times1)$ vector $\boldsymbol{x}_o = \begin{bmatrix} \boldsymbol{o}_o^T & \boldsymbol{\varphi}_o^T \end{bmatrix}^T$ defines a representation of the object pose with respect to the base frame in terms of $m = 3 + p$ parameters.

The homogeneous coordinate vector $\tilde{\boldsymbol{p}} = \begin{bmatrix} \boldsymbol{p}^T & 1 \end{bmatrix}^T$ of a point P of the object with respect to the base frame can be computed as $\tilde{\boldsymbol{p}} = \boldsymbol{H}_o(\boldsymbol{x}_o)^o\tilde{\boldsymbol{p}}$, where $^o\tilde{\boldsymbol{p}}$ is the homogeneous coordinate vector of P with respect to the object frame and $\boldsymbol{H}_o$ is the homogeneous transformation matrix representing the pose of the object frame referred to the base frame:

$$\boldsymbol{H}_o(\boldsymbol{x}_o) = \begin{bmatrix} \boldsymbol{R}_o(\boldsymbol{\varphi}_o) & \boldsymbol{o}_o \\ \boldsymbol{0}_3^T & 1 \end{bmatrix},$$

where $\boldsymbol{0}_3$ is the (3×1) null vector.

It is assumed that the geometry of the object is known and that the interaction involves a portion of the external surface which satisfies the continuously differentiable scalar equation $\varphi(^o\boldsymbol{p}) = 0$.

The unit vector normal to the surface at the point $^o\boldsymbol{p}$ and pointing outwards can be computed as:

$$^o\boldsymbol{n}(^o\boldsymbol{p}) = \frac{(\partial\varphi(^o\boldsymbol{p})/\partial\,^o\boldsymbol{p})^T}{\|(\partial\varphi(^o\boldsymbol{p})/\partial\,^o\boldsymbol{p}\|}, \qquad (2)$$

where $^o\boldsymbol{n}$ is expressed in the object frame.

Notice that the object pose $\boldsymbol{x}_o$ is assumed to be unknown and may change during the task execution. As an example, a compliant contact can be modelled assuming that $\boldsymbol{x}_o$ changes during the interaction according to an elastic law.

A further assumption is that the contact between the robot and the object is of point type and frictionless. Therefore, when in contact, the tip point P_q of the robot instantaneously coincides with a point P of the object, so that the tip position $^o\boldsymbol{p}_q$ satisfies the constraint equation:

$$\varphi(^o\boldsymbol{p}_q) = 0. \qquad (3)$$

Moreover, the (3×1) contact force $^o\boldsymbol{h}$ is aligned to the normal unit vector $^o\boldsymbol{n}$.

B. Robot

The case of a n-joints robot manipulator is considered, with $n \geq 3$. The tip position $\boldsymbol{p}_q$ can be computed via the direct kinematics equation:

$$p_q = k(q), \qquad (4)$$

where q is the $(n \times 1)$ vector of the joint variables. Also, the velocity of the robot's tip v_{P_q} can be expressed as

$$v_{P_q} = J(q)\dot{q}$$

where $J = \partial k(q)/\partial q$ is the robot Jacobian matrix. The vector v_{P_q} can be decomposed as

$$^o v_{P_q} = {}^o\dot{p}_q + \Lambda({}^o p_q){}^o \nu_o, \qquad (5)$$

with $\Lambda(\cdot) = [I_3 \quad -S(\cdot)]$, where I_3 is the (3×3) identity matrix and $S(\cdot)$ denotes the (3×3) skew-symmetric matrix operator. In Eq. (5), ${}^o\dot{p}_q$ is the relative velocity of the tip point P_q with respect to the object frame while ${}^o\nu_o = [{}^o v_{O_o}^T \quad {}^o\omega_o^T]^T$ is the velocity screw characterizing the motion of the object frame with respect to the base frame in terms of the translational velocity of the origin v_{O_o} and of the angular velocity ω_o; all the quantities are expressed in the object frame.

When the robot is in contact to the object, the normal component of the relative velocity ${}^o\dot{p}_q$ is null, i.e.:

$$^o n^T ({}^o p_q){}^o\dot{p}_q = 0. \qquad (6)$$

C. Camera

A frame O_c–$x_c y_c z_c$ attached to the camera (either in eye-in-hand or in eye-to-hand configuration) is considered. By using the classical pin-hole model, a point P of the object with coordinates ${}^c p = \begin{bmatrix} x & y & z \end{bmatrix}^T$ with respect to the camera frame is projected onto the point of the image plane with coordinates

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{\lambda_c}{z} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (7)$$

where λ_c is the focal length of the lens of the camera.

Let H_c denote the homogeneous transformation matrix representing the pose of the camera frame referred to the base frame. For eye-to-hand cameras, the matrix H_c is constant, and can be computed through a suitable calibration procedure, while for eye-in-hand cameras this matrix depends on the camera current pose x_c and can be computed as:

$$H_c(x_c) = H_e(x_e){}^e H_c$$

where H_e is the homogeneous transformation matrix of the end effector frame e with respect to the base frame, and ${}^e H_c$ is the homogeneous transformation matrix of camera frame with respect to end effector frame. Notice that ${}^e H_c$ is constant and can be estimated through suitable calibration procedures, while H_e depends on the current end-effector pose x_e and may be computed using the robot kinematic model.

Therefore, the homogeneous coordinate vector of P with respect to the camera frame can be expressed as

$$^c\tilde{p} = {}^c H_o(x_o, x_c){}^o\tilde{p} \qquad (8)$$

where ${}^c H_o(x_o, x_c) = {}^c H^{-1}(x_c) H_o(x_o)$. Notice that x_c is constant for eye-to-hand cameras; moreover, the matrix ${}^c H_o$

does not depend on x_c and x_o separately but only on the relative pose of the object frame with respect to the camera frame.

The velocity of the camera frame with respect to the base frame can be characterized in terms of the translational velocity of the origin v_{O_c} and of angular velocity ω_c. These vectors, expressed in camera frame, define the velocity screw ${}^c\nu_c = [{}^c v_{O_c}^T \quad {}^c\omega_c^T]^T$. Analogously to (5), the absolute velocity of the origin O_o of the object frame can be computed as

$$^c v_{O_o} = {}^c\dot{o}_o + \Lambda({}^c o_o){}^c\nu_c, \qquad (9)$$

where ${}^c o_o$ is the vector of the coordinates of O_o with respect to camera frame and ${}^c\dot{o}_o$ is the relative velocity of O_o with respect to camera frame; all the quantities are expressed in camera frame. On the other hand, the absolute angular velocity ${}^c\omega_o$ of the object frame expressed in camera frame can be computed as

$$^c\omega_o = {}^c\omega_{o,c} + {}^c\omega_c \qquad (10)$$

where ${}^c\omega_{o,c}$ represents the relative angular velocity of the object frame with respect to the camera frame. The two equations (9) and (10) can be rewritten in the compact form

$$^c\nu_o = {}^c\nu_{o,c} + \Gamma({}^c o_o){}^c\nu_c \qquad (11)$$

where ${}^c\nu_o = [{}^c v_{O_o}^T \quad {}^c\omega_o^T]^T$ is the velocity screw corresponding to the absolute motion of the object frame, ${}^c\nu_{o,c} = [{}^c\dot{o}_o^T \quad {}^c\omega_{o,c}^T]^T$ is the velocity screw corresponding to the relative motion of the object frame with respect to camera frame, and the matrix $\Gamma(\cdot)$ is defined as

$$\Gamma(\cdot) = \begin{bmatrix} I_3 & -S(\cdot) \\ O_3 & I_3 \end{bmatrix},$$

where O_3 denotes the (3×3) null matrix.

The velocity screw ${}^r\nu_s$ of a frame s with respect to a frame r can be expressed in terms of the time derivative of the vector x_s representing the pose of frame s through the equation

$$^r\nu_s = {}^r L(x_s)\dot{x}_s \qquad (12)$$

where ${}^r L(\cdot)$ is a Jacobian matrix depending on the particular choice of coordinates for the orientation. The expressions of ${}^r L(\cdot)$ for different kinds of parametrization of the orientation can be found, e.g., in [10].

IV. OBJECT POSE ESTIMATION

When the robot moves in free space, the unknown object pose can be estimated online by using the data provided by the camera; when the robot is in contact to the target object, also the force measurements and the joint position measurements can be used. In this section, the equations mapping the measurements to the unknown position and orientation of the object are derived. Then, the estimation algorithm based on the EKF is presented.

A. Vision

Vision is used to measure the image features, i.e., any structural feature that can be extracted from an image, corresponding to the projection of a physical feature of the object onto the camera image plane.

An image feature can be characterized by a set of scalar parameters f_j that can be grouped in a vector $\boldsymbol{f} = [f_1 \cdots f_k]^T$, where k is the dimension of the image feature parameter space. The mapping from the position and orientation of the object to the corresponding image feature vector can be computed using the projective geometry of the camera and can be written in the form

$$\boldsymbol{f} = \boldsymbol{g}_f({}^c\boldsymbol{H}_o(\boldsymbol{x}_o, \boldsymbol{x}_c)), \tag{13}$$

where only the dependence from the relative pose of the object frame with respect to camera frame has been explicitly evidenced.

For the estimation of the object pose, the computation of the Jacobian matrix

$$\boldsymbol{J}_f = \frac{\partial \boldsymbol{g}_f}{\partial \boldsymbol{x}_o}.$$

is required. To this purpose, the time derivative of (13) can be computed in the form

$$\dot{\boldsymbol{f}} = \frac{\partial \boldsymbol{g}_f}{\partial \boldsymbol{x}_o}\dot{\boldsymbol{x}}_o + \frac{\partial \boldsymbol{g}_f}{\partial \boldsymbol{x}_c}\dot{\boldsymbol{x}}_c, \tag{14}$$

where the second term in the right hand side is null for eye-to-hand cameras. On the other hand, the time derivative of (13) can be expressed also in the form

$$\dot{\boldsymbol{f}} = \boldsymbol{J}_{o,c}{}^c\boldsymbol{\nu}_{o,c} \tag{15}$$

where the matrix $\boldsymbol{J}_{o,c}$ is the Jacobian mapping the relative velocity screw of the object frame with respect to the camera frame into the variation of the image feature parameters. The expression of $\boldsymbol{J}_{o,c}$ depends on the choice of the image features; examples of computation can be found in [10].

Taking into account the velocity composition (11), Eq. (15) can be rewritten in the form

$$\dot{\boldsymbol{f}} = \boldsymbol{J}_{o,c}{}^c\boldsymbol{\nu}_o - \boldsymbol{J}_c{}^c\boldsymbol{\nu}_c \tag{16}$$

where $\boldsymbol{J}_c = \boldsymbol{J}_{o,c}\boldsymbol{\Gamma}({}^c\boldsymbol{o}_o)$ is the Jacobian corresponding to the contribution of the absolute velocity screw of the camera frame, known in the literature as interaction matrix [11]. Considering Eq. (12), the comparison of (16) with (14) yields

$$\boldsymbol{J}_f = \boldsymbol{J}_{o,c}{}^c\boldsymbol{L}(\boldsymbol{x}_o). \tag{17}$$

B. Force and joint measurements

In the case of frictionless point contact, the measure of the force $\boldsymbol{h}$ at the robot tip during the interaction can be used to compute the unit vector normal to the object surface at the contact point ${}^o\boldsymbol{p}_q$, i.e.,

$$\boldsymbol{n}_h = \frac{\boldsymbol{h}}{\|\boldsymbol{h}\|}. \tag{18}$$

On the other hand, vector $\boldsymbol{n}_h$ can be expressed as a function of the object pose $\boldsymbol{x}_o$ and of the robot position $\boldsymbol{p}_q$ in the form

$$\boldsymbol{n}_h = \boldsymbol{R}_o{}^o\boldsymbol{n}({}^o\boldsymbol{p}_q) = \boldsymbol{g}_h(\boldsymbol{x}_o, \boldsymbol{p}_q), \tag{19}$$

being ${}^o\boldsymbol{p}_q = \boldsymbol{R}_o^T(\boldsymbol{p}_q - \boldsymbol{o}_o)$.

For the estimation of the object pose, the computation of the Jacobian matrix

$$\boldsymbol{J}_h = \frac{\partial \boldsymbol{g}_h}{\partial \boldsymbol{x}_o}.$$

is required. To this purpose, the time derivative of (19) can be expressed as

$$\dot{\boldsymbol{n}}_h = \frac{\partial \boldsymbol{g}_h}{\partial \boldsymbol{x}_o}\dot{\boldsymbol{x}}_o + \frac{\partial \boldsymbol{g}_h}{\partial \boldsymbol{p}_q}\dot{\boldsymbol{p}}_q. \tag{20}$$

On the other hand, the time derivative of (19) can be computed also in the form

$$\dot{\boldsymbol{n}}_h = \dot{\boldsymbol{R}}_o{}^o\boldsymbol{n}({}^o\boldsymbol{p}_q) + \boldsymbol{R}_o{}^o\boldsymbol{N}({}^o\boldsymbol{p}_q){}^o\dot{\boldsymbol{p}}_q, \tag{21}$$

where ${}^o\boldsymbol{N}({}^o\boldsymbol{p}_q) = \partial{}^o\boldsymbol{n}/\partial{}^o\boldsymbol{p}_q$ depends on the surface curvature and ${}^o\dot{\boldsymbol{p}}_q$ can be computed from (5). Hence, comparing (20) with (21) and taking into account (12) and the equality $\dot{\boldsymbol{R}}_o{}^o\boldsymbol{n}({}^o\boldsymbol{p}_q) = -\boldsymbol{S}(\boldsymbol{n}_h)\boldsymbol{\omega}_o$, the following expression can be found:

$$\boldsymbol{J}_h = -[\boldsymbol{N} \quad \boldsymbol{S}(\boldsymbol{n}_h) - \boldsymbol{N}\boldsymbol{S}(\boldsymbol{p}_q - \boldsymbol{o}_o)]\boldsymbol{L}(\boldsymbol{x}_o), \tag{22}$$

where $\boldsymbol{N} = \boldsymbol{R}_o{}^o\boldsymbol{N}({}^o\boldsymbol{p}_q)\boldsymbol{R}_o^T$.

The measurement of the joint position vector $\boldsymbol{q}$ can be used to evaluate the position of the point P of the object when in contact to the robot's tip point P_q, using the direct kinematics equation (4). In particular, it is significant computing the scalar

$$\delta_{hq} = \boldsymbol{n}_h^T\boldsymbol{p}_q = g_{hq}(\boldsymbol{x}_o, \boldsymbol{p}_q), \tag{23}$$

using also the force measurements via Eq. (18).

For the estimation of the object pose, the computation of the Jacobian matrix

$$\boldsymbol{J}_{hq} = \frac{\partial g_{hq}}{\partial \boldsymbol{x}_o}.$$

is required. As in the previous subsection, the time derivative of δ_{hq} can be expressed as

$$\dot{\delta}_{hq} = \frac{\partial g_{hq}}{\partial \boldsymbol{x}_o}\dot{\boldsymbol{x}}_o + \frac{\partial g_{hq}}{\partial \boldsymbol{p}_q}\dot{\boldsymbol{p}}_q. \tag{24}$$

On the other hand, the time derivative of δ_{hq} can be computed also as

$$\dot{\delta}_{hq} = \dot{\boldsymbol{n}}_h^T\boldsymbol{p}_q + \boldsymbol{n}_h^T\boldsymbol{R}_o({}^o\dot{\boldsymbol{p}}_q + \boldsymbol{\Lambda}({}^o\boldsymbol{p}_q){}^o\boldsymbol{\nu}_o)$$

where the expression of the absolute velocity of the point P_q in (5) has been used. Using the identity (6), the above equation can be rewritten as

$$\dot{\delta}_{hq} = \boldsymbol{p}_q^T\dot{\boldsymbol{n}}_h + \boldsymbol{n}_h^T\boldsymbol{\Lambda}(\boldsymbol{p}_q - \boldsymbol{o}_o)\boldsymbol{\nu}_o. \tag{25}$$

Hence, comparing (24) with (25) and taking into account (21), (22) and (12), the following expression can be found

$$\boldsymbol{J}_{hq} = \boldsymbol{p}_q^T\boldsymbol{J}_h + \boldsymbol{n}_h^T\boldsymbol{\Lambda}(\boldsymbol{p}_q - \boldsymbol{o}_o)\boldsymbol{L}(\boldsymbol{x}_o). \tag{26}$$

C. Extended Kalman Filter

The pose vector $\boldsymbol{x}_o$ of the object with respect to the base frame can be estimated using an Extended Kalman Filter.

To this purpose, a discrete-time state space dynamic model has to be considered, describing the object motion. The state vector of the dynamic model is chosen as $\boldsymbol{w} = \begin{bmatrix} \boldsymbol{x}_o^{\mathrm{T}} & \dot{\boldsymbol{x}}_o^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. For simplicity, the object velocity is assumed to be constant over one sample period T_s. This approximation is reasonable in the hypothesis that T_s is sufficiently small. The corresponding dynamic modeling error can be considered as an input disturbance $\boldsymbol{\gamma}$ described by zero mean Gaussian noise with covariance $\boldsymbol{Q}$. The discrete-time dynamic model can be written as

$$\boldsymbol{w}_k = \boldsymbol{A}\boldsymbol{w}_{k-1} + \boldsymbol{\gamma}_k, \qquad (27)$$

where $\boldsymbol{A}$ is the $(2m \times 2m)$ block matrix

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I}_m & T_s\boldsymbol{I}_m \\ \boldsymbol{O}_m & \boldsymbol{I}_m \end{bmatrix}.$$

The output of the Kalman filter, in the case that all the available data can be used, is the vector of the measurements at time kT_s

$$\boldsymbol{\zeta}_k = \begin{bmatrix} \boldsymbol{\zeta}_{f,k}^T & \boldsymbol{\zeta}_{h,k}^T & \zeta_{hq,k}{}^T \end{bmatrix}^T,$$

where $\boldsymbol{\zeta}_{f,k} = \boldsymbol{f}_k + \boldsymbol{\mu}_{f,k}$, $\boldsymbol{\zeta}_{h,k} = \boldsymbol{h}_k + \boldsymbol{\mu}_{h,k}$, and $\zeta_{hq,k} = \delta_k + \mu_{hq,k}$, being $\boldsymbol{\mu}$ the measurement noise. The measurement noise is assumed to be zero mean Gaussian noise with covariance $\boldsymbol{\Pi}$.

Taking into account the Eqs. (13), (19), and (23), the output model of the Kalman filter can be written in the form:

$$\boldsymbol{\zeta}_k = \boldsymbol{g}(\boldsymbol{w}_k) + \boldsymbol{\mu}_k,$$

where $\begin{bmatrix} \boldsymbol{\mu}_{f,k}^T & \boldsymbol{\mu}_{h,k}^T & \mu_{hq,k}^T \end{bmatrix}^T$ and

$$\boldsymbol{g}(\boldsymbol{w}_k) = \begin{bmatrix} \boldsymbol{g}_f^T(\boldsymbol{w}_k) & \boldsymbol{g}_h^T(\boldsymbol{w}_k) & g_{hq}^T(\boldsymbol{w}_k) \end{bmatrix}^T \qquad (28)$$

where only the explicit dependence on the state vector $\boldsymbol{w}_k$ has been evidenced.

Since the output model is nonlinear in the system state, the EKF must be adopted, which requires the computation of the Jacobian matrix of the output equation

$$\boldsymbol{C}_k = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\hat{\boldsymbol{w}}_{k,k-1}} = \begin{bmatrix} \dfrac{\partial \boldsymbol{g}(\boldsymbol{w})}{\partial \boldsymbol{x}_o} & \boldsymbol{O} \end{bmatrix}_{\boldsymbol{w}=\hat{\boldsymbol{w}}_{k,k-1}},$$

where $\boldsymbol{O}$ is a null matrix of proper dimensions corresponding to the partial derivative of $\boldsymbol{g}$ with respect to the velocity variables, which is null because function $\boldsymbol{g}$ does not depend on the velocity.

The Jacobian matrix $\partial \boldsymbol{g}(\boldsymbol{w})/\partial \boldsymbol{x}_o$, in view of (17), (22), and (26) has the expression

$$\frac{\partial \boldsymbol{g}(\boldsymbol{w})}{\partial \boldsymbol{x}_o} = \begin{bmatrix} \boldsymbol{J}_f^T & \boldsymbol{J}_h^T & \boldsymbol{J}_{hq}^T \end{bmatrix}^T.$$

The equations of the recursive form of the EKF are standard and are omitted here.

Notice that the proposed algorithm can be used to estimate online the pose of an object in the workspace; hence it allows the computation of the constraint Eq. (3) with respect to the base frame in the form $\varphi(\boldsymbol{R}_o^T(\boldsymbol{p}_q - \boldsymbol{o}_o)) = 0$. This information can be suitably exploited to implement any kind of interaction control strategy.

Fig. 3. Experimental setup.

V. Experiments

The experimental setup (Fig. 3) is composed by a 6-DOF industrial robot Comau SMART-3 S with an open control architecture based on RTAI-Linux operating system. A six-axis force/torque sensor ATI FT30-100 with force range of ± 130 N and torque range of ± 10 N·m is mounted at the arm's wrist, providing readings of six components of generalized force at 1 ms. A visual system composed by a PC equipped with a Matrox GENESIS board and a Sony 8500CE B/W camera is available. The camera is fixed and calibrated with respect to the base frame of the robot.

The environment is a planar wooden horizontal surface, described by the equation ${}^o\boldsymbol{n}^T {}^o\boldsymbol{p} = 0$, assuming that the origin O_o of the object frame is a point of the plane and the axis z_o is aligned to the normal ${}^o\boldsymbol{n}$. The plane has an estimated stiffness (along ${}^o\boldsymbol{n}$) of about 46000 N/m. The object features are 8 landmark points lying on the plane at the corners of two rectangles of 10×20 cm size (as in Fig. 3).

The end-effector tool is a rigid stick of 15 cm length ending with a small sphere, so that the hypothesis of point contact is reasonable.

The impedance parameters are chosen as: $\boldsymbol{M}_p = 40\boldsymbol{I}_3$, $\boldsymbol{D}_p = 26.3\boldsymbol{I}_3$ and $\boldsymbol{K}_p = 1020\boldsymbol{I}_3$, $\boldsymbol{M}_o = 15\boldsymbol{I}_3$, $\boldsymbol{D}_o = 17.4\boldsymbol{I}_3$ and $\boldsymbol{K}_o = 3\boldsymbol{I}_3$; a 1 ms sampling time has been selected for the impedance and pose control. Notice that the stiffness of the object is much higher that the positional stiffness of the impedance, so that the environment can be considered rigid.

The desired task is planned in the object frame and consists in a straight-line motion of the end-effector along the z_o-axis keeping a fixed orientation with the stick normal to the $x_o y_o$-plane. The final position is ${}^o\boldsymbol{p}_f = \begin{bmatrix} 0 & 0 & 0.05 \end{bmatrix}^T$ m, which is chosen to have a normal force of about 50 N at the equilibrium, with the selected value of the impedance positional stiffness. A trapezoidal velocity profile time-law is adopted, with a cruise velocity of 0.01 m/s. The absolute trajectory is computed from the desired relative trajectory using the current object pose estimation. The final position of the end-effector is held for 2 s; after, a vertical motion in the opposite direction is commanded.

In the EKF, the non null elements of the matrix $\boldsymbol{\Pi}$ have been set equal to 2.5 pixel^2 for $\boldsymbol{f}$, $5 \cdot 10^{-4}$ for $\boldsymbol{n}_h$ and 10^{-6} m^2 for δ_{hq}. The state noise covariance matrix has been selected so as to give a rough measure of the errors due to the

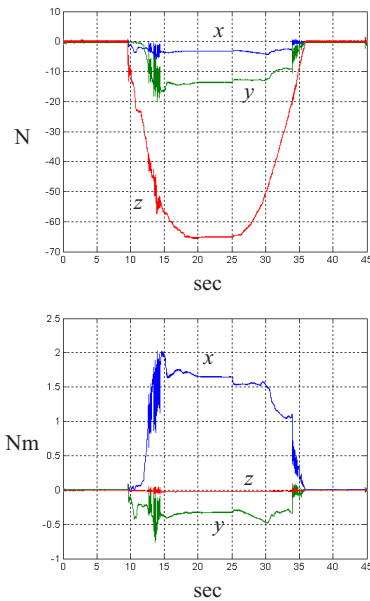Fig. 4. Measured force (top) and moment (bottom) in the 1st experiment.

Fig. 5. Measured force (top) and moment (bottom) in the 2nd experiment.

simplification introduced on the model (constant velocity), by considering only velocity disturbance, i.e.

$$\boldsymbol{Q} = \mathrm{diag}\{0, 0, 0, 0, 0, 0, 0, 10, 10, 10, 1, 1, 1\} \cdot 10^{-9}.$$

Notice that the unit quaternion has been used for the orientation in the EKF, to avoid any occurrence of representation singularities. Moreover a 38 ms sampling time has been used for the estimation algorithm, corresponding to the typical camera frame rate of 26 Hz.

Two different experiments are presented, to show the effectiveness of the use of force and joint position measurements, besides visual measurements.

In the first experiment only the visual measurements are used. The time history of the measured force and moments in the sensor frame are reported in Fig. 4. Notice that the force is null during the motion in free space and becomes different from zero after the contact. The impedance control keeps the force limited during the transient while, at steady state, the force component along the z axis reaches a value of about -65 N, which is different from the desired value of -50 N. This is caused by the presence of estimation errors on the position of the plane due to calibration errors of the camera and to the use of a monocular vision system. Moreover, the moment measured by the sensor is different from zero due to the misalignment between the estimated and the real normal direction of the plane.

The same task is repeated using also the contact force and the joint position measurements for object pose estimation; the results are reported in Fig. 5. It can be observed that the benefit of using additional measurements in the EKF produces a force in the vertical direction which is very close to the desired value, due to the improved estimation of the position of the plane; moreover, the contact moment is also reduced because of the better estimation of the normal direction of the plane.
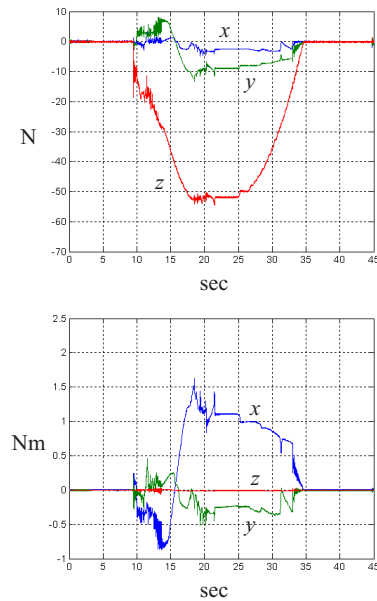
VI. CONCLUSION

A 6-DOF position-based visual impedance control scheme was proposed in this paper. The environment is a rigid object of known geometry but of unknown and possibly time varying pose. A pose estimation algorithm is adopted, based on visual, force and joint positions data. The proposed approach can be cast into any kind of interaction control strategy. Moreover, it can be applied also to the case of multiple cameras in hybrid eye-in-hand/eye-to-hand configuration.

REFERENCES

[1] K. Hosoda, K. Igarashi, and M. Asada, "Adaptive hybrid control for visual and force servoing in an unknown environment," *Robotics & Automation Magazine*, vol. 5, no. 4, pp. 39–43, 1998.
[2] B. J. Nelson, J. D. Morrow, and P. K. Khosla, "Improved force control through visual servoing," in *1995 American Control Conference*, pp. 380–386.
[3] J. Baeten and J. D. Schutter, *Integrated Visual Servoing and Force Control. The Task Frame Approach.* Heidelberg, D: Springer, 2004.
[4] G. Morel, E. Malis, and S. Boudet, "Impedance based combination of visual and force control," in *1998 IEEE Int. Conf. on Robotics and Automation*, pp. 1743–1748.
[5] T. Olsson, R. Johansson, and A. Robertsson, "Flexible force-vision control for surface following using multiple cameras," in *2004 IEEE/RSJ Int. Conf. on Intelligent Robots and System*, pp. 798–803.
[6] W.J. Wilson, C.C.W. Hulls, and G.S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 684–696, 1996.
[7] B. Siciliano and L. Villani, *Robot Force Control.* Boston, MA: Kluwer Academic Publishers, 1999.
[8] V. Lippiello and L. Villani, "Managing redundant visual measurements for accurate pose tracking," *Robotica*, vol. 21, pp. 511–519, 2003.
[9] V. Lippiello, B. Siciliano, L. Villani, "Robot interaction control using force and vision," *2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1470–1475.
[10] V. Lippiello, B. Siciliano, and L. Villani, "3D pose estimation for robotic applications based on a multi-camera hybrid visual system," in *2006 IEEE Int. Conf. on Robotics and Automation*, 2732–2737.
[11] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 313–326, 1992.