

Robot Interaction Control Using Force and Vision

Vincenzo Lippiello, Bruno Siciliano, and Luigi Villani

PRISMA Lab, Dipartimento di Informatica e Sistemistica

Università di Napoli Federico II

Via Claudio 21, 80125 Napoli, ITALY

Email: {lippielli, siciliano, lvillani}@unina.it

Abstract—The problem of interaction control of a robot manipulator with a partially known environment is considered. 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 presented, based on visual data provided by a camera. The estimation accuracy is improved during the interaction by using also force and joint position measurements. The proposed method can be exploited to implement any kind of interaction control strategy. A simulation case study is considered for the case of impedance control of a robot manipulator in contact with a planar surface.

I. INTRODUCTION

The autonomy of a robotic system is strictly connected to the availability of sensing information on the external environment; among the various sensing capabilities, vision and force play a fundamental role.

In fact, thanks to the visual perception, the robotic system 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 of a robot manipulator allows adjusting the end-effector motion so that the local constraints imposed by the environment during the interaction are satisfied.

Because of such complementary nature, it should be natural to believe that vision and force could be used in an integrated and synergic way to design suitable planning and control strategies for robotic systems. In the last years, several papers on this subject has been presented. Some of them combine force of vision in the same feedback control loop, as hybrid visual/force control [1], shared and traded control [2], [3] or visual impedance control [4], [5].

The approach presented 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 used, 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 suitable interaction control strategy must be adopted [7], e.g., hybrid force/position control, parallel force/position control or impedance control. However, the interaction controller may take advantage of the estimation of the geometry of the environment computed recursively from all the available sensor data, i.e., visual, force and joint position measurements.

The estimation algorithm presented here, 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 algorithm can be adopted both in free space and during the interaction, simply modifying the measurements set of the EKF.

A 6-DOF position-based impedance control strategy [7] is adopted for the robot both in free space and during the interaction with the environment. Hence, the proposed scheme can be classified as a position-based visual impedance control.

A simulation case study on an industrial robot is presented. The results confirm the effectiveness of the proposed approach.

II. MODELLING

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

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 $\mathbf{o}_o = [x_o \ y_o \ z_o]^T$ and of the rotation matrix $\mathbf{R}_o(\boldsymbol{\varphi}_o)$, 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 $(m \times 1)$ vector $\mathbf{x}_o = [\mathbf{o}_o^T \ \boldsymbol{\varphi}_o^T]^T$ defines a representation of the object pose with respect to the base frame in terms of $m = 3 + p$ parameters.

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

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

where $\mathbf{0}_3$ 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 $\varphi(\mathbf{p}) = 0$.

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

$${}^o\mathbf{n}({}^o\mathbf{p}) = \frac{(\partial\varphi({}^o\mathbf{p})/\partial {}^o\mathbf{p})^T}{\|(\partial\varphi({}^o\mathbf{p})/\partial {}^o\mathbf{p})\|}, \quad (1)$$

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

Notice that the object pose $\mathbf{x}_o$ is assumed to be unknown and may change during the task execution. As an example, a compliant contact can be modelled assuming that $\mathbf{x}_o$ 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 ${}^o\mathbf{p}_q$ satisfies the constraint equation:

$$\varphi({}^o\mathbf{p}_q) = 0. \quad (2)$$

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

B. Robot

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

$$\mathbf{p}_q = \mathbf{k}(\mathbf{q}), \quad (3)$$

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

$$\mathbf{v}_{P_q} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$$

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

$${}^o\mathbf{v}_{P_q} = {}^o\dot{\mathbf{p}}_q + \mathbf{\Lambda}({}^o\mathbf{p}_q){}^o\boldsymbol{\nu}_o, \quad (4)$$

with $\mathbf{\Lambda}(\cdot) = [\mathbf{I}_3 \quad -\mathbf{S}(\cdot)]$, where $\mathbf{I}_3$ is the (3×3) identity matrix and $\mathbf{S}(\cdot)$ denotes the (3×3) skew-symmetric matrix operator. In Eq. (4), ${}^o\dot{\mathbf{p}}_q$ is the relative velocity of the tip point P_q with respect to the object frame while ${}^o\boldsymbol{\nu}_o = [{}^o\mathbf{v}_{O_o}^T \quad {}^o\boldsymbol{\omega}_o^T]^T$ 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 $\mathbf{v}_{O_o}$ and of the angular velocity $\boldsymbol{\omega}_o$; 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 ${}^o\dot{\mathbf{p}}_q$ is null, i.e.:

$${}^o\mathbf{n}^T({}^o\mathbf{p}_q){}^o\dot{\mathbf{p}}_q = 0. \quad (5)$$

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 ${}^c\mathbf{p} = [x \ y \ z]^T$ with respect to the camera frame is projected onto the point of the image plane with coordinates

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{\lambda_c}{z} \begin{bmatrix} x \\ y \end{bmatrix} \quad (6)$$

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

Let $\mathbf{H}_c$ denote the homogeneous transformation matrix representing the pose of the camera frame referred to the base frame. For eye-to-hand cameras, the matrix $\mathbf{H}_c$ 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 $\mathbf{x}_c$ and can be computed as:

$$\mathbf{H}_c(\mathbf{x}_c) = \mathbf{H}_e(\mathbf{x}_e){}^e\mathbf{H}_c$$

where $\mathbf{H}_e$ is the homogeneous transformation matrix of the end effector frame e with respect to the base frame, and ${}^e\mathbf{H}_c$ is the homogeneous transformation matrix of camera frame with respect to end effector frame. Notice that ${}^e\mathbf{H}_c$ is constant and can be estimated through suitable calibration procedures, while $\mathbf{H}_e$ depends on the current end-effector pose $\mathbf{x}_e$ 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

$${}^c\tilde{\mathbf{p}} = {}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c){}^o\tilde{\mathbf{p}} \quad (7)$$

where ${}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c) = {}^c\mathbf{H}^{-1}(\mathbf{x}_c)\mathbf{H}_o(\mathbf{x}_o)$. Notice that $\mathbf{x}_c$ is constant for eye-to-hand cameras; moreover, the matrix ${}^c\mathbf{H}_o$ does not depend on $\mathbf{x}_c$ and $\mathbf{x}_o$ 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 $\mathbf{v}_{O_c}$ and of angular velocity $\boldsymbol{\omega}_c$. These vectors, expressed in camera frame, define the velocity screw ${}^c\boldsymbol{\nu}_c = [{}^c\mathbf{v}_{O_c}^T \quad {}^c\boldsymbol{\omega}_c^T]^T$. Analogously to (4), the absolute velocity of the origin O_o of the object frame can be computed as

$${}^c\mathbf{v}_{O_o} = {}^c\dot{\mathbf{o}}_o + \mathbf{\Lambda}({}^c\mathbf{o}_o){}^c\boldsymbol{\nu}_c, \quad (8)$$

where ${}^c\mathbf{o}_o$ is the vector of the coordinates of O_o with respect to camera frame and ${}^c\dot{\mathbf{o}}_o$ 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 ${}^c\boldsymbol{\omega}_o$ of the object frame expressed in camera frame can be computed as

$${}^c\boldsymbol{\omega}_o = {}^c\boldsymbol{\omega}_{o,c} + {}^c\boldsymbol{\omega}_c \quad (9)$$

where ${}^c\boldsymbol{\omega}_{o,c}$ represents the relative angular velocity of the object frame with respect to the camera frame. The two equations (8) and (9) can be rewritten in the compact form

$${}^c\boldsymbol{\nu}_o = {}^c\boldsymbol{\nu}_{o,c} + \mathbf{\Gamma}({}^c\mathbf{o}_o){}^c\boldsymbol{\nu}_c \quad (10)$$

where ${}^c\boldsymbol{\nu}_o = [{}^c\mathbf{v}_{O_o}^T \quad {}^c\boldsymbol{\omega}_o^T]^T$ is the velocity screw corresponding to the absolute motion of the object frame, ${}^c\boldsymbol{\nu}_{o,c} = [{}^c\dot{\mathbf{o}}_o^T \quad {}^c\boldsymbol{\omega}_{o,c}^T]^T$ is the velocity screw corresponding to the relative motion of the object frame with respect to camera frame, and the matrix $\mathbf{\Gamma}(\cdot)$ is defined as

$$\mathbf{\Gamma}(\cdot) = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{S}(\cdot) \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix},$$

where $\mathbf{O}_3$ 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 $\mathbf{x}_s$ representing the pose of frame s through the equation

$${}^r\nu_s = {}^r\mathbf{L}(\mathbf{x}_s)\dot{\mathbf{x}}_s \quad (11)$$

where ${}^r\mathbf{L}(\cdot)$ is a Jacobian matrix depending on the particular choice of coordinates for the orientation. The expressions of ${}^r\mathbf{L}(\cdot)$ for different kinds of parametrization of the orientation can be found, e.g., in [9].

III. USE OF VISION, FORCE AND JOINT MEASUREMENTS

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 object, also the force measurements and the joint position measurements are used. In the following, the equations mapping the measurements to the unknown position and orientation of the object are derived.

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 $\mathbf{f} = [f_1 \cdots f_k]^T$, 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

$$\mathbf{f} = \mathbf{g}_f({}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c)), \quad (12)$$

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, it is required the computation of the Jacobian matrix

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

To this purpose, the time derivative of (12) can be computed in the form

$$\dot{\mathbf{f}} = \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}_c} \dot{\mathbf{x}}_c, \quad (13)$$

where the second term in the right hand side is null for eye-to-hand cameras. On the other hand, the time derivative of (12) can be expressed also in the form

$$\dot{\mathbf{f}} = \mathbf{J}_{o,c} {}^c\nu_{o,c} \quad (14)$$

where the matrix $\mathbf{J}_{o,c}$ 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 $\mathbf{J}_{o,c}$ depends on the choice of the image features; examples of computation can be found in [9].

Taking into account the velocity composition (10), Eq. (14) can be rewritten in the form

$$\dot{\mathbf{f}} = \mathbf{J}_{o,c} {}^c\nu_o - \mathbf{J}_c {}^c\nu_c \quad (15)$$

where $\mathbf{J}_c = \mathbf{J}_{o,c} \Gamma({}^c\mathbf{o}_o)$ is the Jacobian corresponding to the contribution of the absolute velocity screw of the camera frame, known in the literature as interaction matrix [10]. Considering Eq. (11), the comparison of (15) with (13) yields

$$\mathbf{J}_f = \mathbf{J}_{o,c} {}^c\mathbf{L}(\mathbf{x}_o). \quad (16)$$

B. Force and joint measurements

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

$$\mathbf{n}_h = \frac{\mathbf{h}}{\|\mathbf{h}\|}. \quad (17)$$

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

$$\mathbf{n}_h = \mathbf{R}_o {}^o\mathbf{n}({}^o\mathbf{p}_q) = \mathbf{g}_h(\mathbf{x}_o, \mathbf{p}_q), \quad (18)$$

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

For the estimation of the object pose, it is required the computation of the Jacobian matrix

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

To this purpose, the time derivative of (18) can be expressed as

$$\dot{\mathbf{n}}_h = \frac{\partial \mathbf{g}_h}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial \mathbf{g}_h}{\partial \mathbf{p}_q} \dot{\mathbf{p}}_q. \quad (19)$$

On the other hand, the time derivative of (18) can be computed also in the form

$$\dot{\mathbf{n}}_h = \dot{\mathbf{R}}_o {}^o\mathbf{n}({}^o\mathbf{p}_q) + \mathbf{R}_o {}^o\mathbf{N}({}^o\mathbf{p}_q) \dot{{}^o\mathbf{p}}_q, \quad (20)$$

where ${}^o\mathbf{N}({}^o\mathbf{p}_q) = \partial {}^o\mathbf{n} / \partial {}^o\mathbf{p}_q$ depends on the surface curvature and $\dot{{}^o\mathbf{p}}_q$ can be computed from (4). Hence, comparing (19) with (20) and taking into account (11) and the equality $\dot{\mathbf{R}}_o {}^o\mathbf{n}({}^o\mathbf{p}_q) = -\mathbf{S}(\mathbf{n}_h)\omega_o$, the following expression can be found:

$$\mathbf{J}_h = -[\mathbf{N} \quad \mathbf{S}(\mathbf{n}_h) - \mathbf{N}\mathbf{S}(\mathbf{p}_q - \mathbf{o}_o)]\mathbf{L}(\mathbf{x}_o), \quad (21)$$

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

The measurement of the joint position vector $\mathbf{q}$ 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 (3). In particular, it is significant computing the scalar

$$\delta_{hq} = \mathbf{n}_h^T \mathbf{p}_q = g_{hq}(\mathbf{x}_o, \mathbf{p}_q), \quad (22)$$

using also the force measurements via Eq. (17).

For the estimation of the object pose it is required the computation of the Jacobian matrix

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

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

$$\dot{\delta}_{hq} = \frac{\partial g_{hq}}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial g_{hq}}{\partial \mathbf{p}_q} \dot{\mathbf{p}}_q. \quad (23)$$

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

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

where the expression of the absolute velocity of the point P_q in (4) has been used. Using the identity (5), the above equation can be rewritten as

$$\dot{\delta}_{hq} = \mathbf{p}_q^T \dot{\mathbf{n}}_h + \mathbf{n}_h^T \mathbf{\Lambda}(\mathbf{p}_q - \mathbf{o}_o) \boldsymbol{\nu}_o. \quad (24)$$

Hence, comparing (23) with (24) and taking into account (20), (21) and (11), the following expression can be found

$$\mathbf{J}_{hq} = \mathbf{p}_q^T \mathbf{J}_h + \mathbf{n}_h^T \mathbf{\Lambda}(\mathbf{p}_q - \mathbf{o}_o) \mathbf{L}(\mathbf{x}_o). \quad (25)$$

IV. OBJECT POSE ESTIMATION

In this Section, the problem of the estimation of the pose vector $\mathbf{x}_o$ of the object with respect to the base frame using visual, force and joint position measurements. The proposed solution is based on the EKF [8].

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 $\mathbf{w} = [\mathbf{x}_o^T \quad \dot{\mathbf{x}}_o^T]^T$. 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 γ described by zero mean Gaussian noise with covariance $\mathbf{Q}$. The discrete-time dynamic model can be written as

$$\mathbf{w}_k = \mathbf{A}\mathbf{w}_{k-1} + \gamma_k, \quad (26)$$

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

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

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

$$\boldsymbol{\zeta}_k = [\boldsymbol{\zeta}_{f,k}^T \quad \boldsymbol{\zeta}_{h,k}^T \quad \boldsymbol{\zeta}_{hq,k}^T]^T,$$

where $\boldsymbol{\zeta}_{f,k} = \mathbf{f}_k + \boldsymbol{\mu}_{f,k}$, $\boldsymbol{\zeta}_{h,k} = \mathbf{h}_k + \boldsymbol{\mu}_{h,k}$, and $\boldsymbol{\zeta}_{hq,k} = \delta_k + \boldsymbol{\mu}_{hq,k}$, 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. (12), (18), and (22), the output model of the Kalman filter can be written in the form:

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

where $[\boldsymbol{\mu}_{f,k}^T \quad \boldsymbol{\mu}_{h,k}^T \quad \boldsymbol{\mu}_{hq,k}^T]^T$ and

$$\mathbf{g}(\mathbf{w}_k) = [\mathbf{g}_f^T(\mathbf{w}_k) \quad \mathbf{g}_h^T(\mathbf{w}_k) \quad \mathbf{g}_{hq}^T(\mathbf{w}_k)]^T \quad (27)$$

where only the explicit dependence on the state vector $\mathbf{w}_k$ 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

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

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

The Jacobian matrix $\partial \mathbf{g}(\mathbf{w}) / \partial \mathbf{x}_o$, in view of (16), (21), and (25) has the expression

$$\frac{\partial \mathbf{g}(\mathbf{w})}{\partial \mathbf{x}_o} = [\mathbf{J}_f^T \quad \mathbf{J}_h^T \quad \mathbf{J}_{hq}^T]^T.$$

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

V. INTERACTION CONTROL

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. (2) with respect to the base frame in the form

$$\varphi(\mathbf{R}_o^T(\mathbf{p}_q - \mathbf{o}_o)) = 0.$$

This information can be suitably exploited to implement any kind of interaction control strategy. In this paper, an impedance control is adopted, according to a position-based control scheme [7].

In detail, a position and orientation control is adopted for the robot end-effector, and a pose trajectory for a desired frame d is specified in terms of $\mathbf{p}_d$ and $\mathbf{R}_d$. To manage the interaction with the environment, a compliant frame r is introduced, specified in terms of $\mathbf{p}_r$ and $\mathbf{R}_r$. Then, a mechanical impedance between the desired and the compliant frame is considered, so as to keep limited the values of the interaction force $\mathbf{h}$ and moment $\mathbf{m}$. 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 pose control of the robot end effector.

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

$$\mathbf{M}_p \Delta \ddot{\mathbf{p}}_{dr} + \mathbf{D}_p \Delta \dot{\mathbf{p}}_{dr} + \mathbf{K}_p \Delta \mathbf{p}_{dr} = \mathbf{h}, \quad (28)$$

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

The orientation of the reference frame $\mathbf{R}_r$ is computed via a geometrically consistent impedance equation similar to (28), 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 ${}^r\mathbf{R}_d = \mathbf{R}_r^T \mathbf{R}_d$. The corresponding

mass, damping and inertia matrices are M_o , D_o and K_o respectively. More details about the geometrically consistent impedance based on the unit quaternion can be found in [7].

Notice that, when the robot moves in free space, the proposed scheme is equivalent to a position-based visual servoing [6]. Hence, it can be classified as a position-based visual impedance control.

VI. CASE STUDY

A planar object surface is considered, described by the equation

$${}^o\mathbf{n}^T {}^o\mathbf{p} = 0,$$

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 ${}^o\mathbf{n}$. During the interaction with the robot, the normal vector $\mathbf{n}$ remains constant in the base frame while the plane is elastically compliant along $\mathbf{n}$ according to a simple elastic law. The contact force of the object on the robot's tip at $\mathbf{p}_q$ is given by

$$\mathbf{h} = \begin{cases} k\mathbf{n}\mathbf{n}^T(\mathbf{p}_o - \mathbf{p}_q) & \text{if } \mathbf{n}^T(\mathbf{p}_o - \mathbf{p}) \geq 0 \\ \mathbf{0}_3 & \text{if } \mathbf{n}^T(\mathbf{p}_o - \mathbf{p}) < 0 \end{cases}$$

where $\mathbf{p}_q$ is on the plane when $\mathbf{h} \neq \mathbf{0}_3$ while $\mathbf{p}_o$ is a constant vector representing the position of a point of the plane when $\mathbf{h} = \mathbf{0}_3$. The scalar k , representing the stiffness of the surface, has been set to 10000 N/m.

An industrial robot Comau SMART-3 S is considered for the simulation case study, using MATLAB/Simulink. The robot has a six-degree-of-freedom anthropomorphic geometry (see [7] for the kinematic and dynamic model).

The end-effector tool is a rigid stick of 25 cm length ending with a circular disk of 5 cm radius. The end-effector frame has its origin at the center of the disk and its approach axis normal to the disk surface and pointing outwards. During the interaction, when the disk surface and the plane are not parallel, the robot's tip point P_Q is assumed to be the instantaneous contact point of the external contour of the disk with the plane. In the case that the disk and the plane are parallel and in contact, the instantaneous contact point is chosen as the center of the disk.

The robot has a force/position sensor mounted at the wrist. Neglecting the weight and inertia of the tool, the force at the robot's tip point P_Q and that at the origin of the end-effector frame are the same, while a moment is present at the origin of the end-effector frame due to the contact with the external contour of the disk. Notice that both the impedance equation and the pose control law are formulated for the end-effector frame.

A camera is mounted on the robot end effector. It is assumed that the intrinsic parameters of the camera are affected by a 2% error, while the extrinsic parameters are known. The object features are 4 landmark points lying on the plane at the corners of a square of 10 cm side.

A sketch of the end-effector in contact with the plane is reported in Fig. 1.

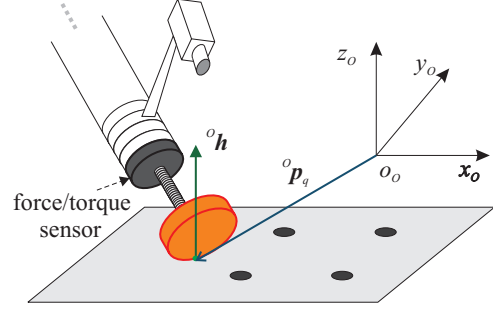


Fig. 1. Sketch of the end-effector in contact with the plane.

The impedance parameters are chosen as: $M_p = 9\mathbf{I}_3$, $D_p = 5\mathbf{I}_3$ and $K_p = 700\mathbf{I}_3$, $M_o = 0.4\mathbf{I}_3$, $D_o = 5\mathbf{I}_3$ and $K_o = 2\mathbf{I}_3$; a 2 ms sampling time has been selected for the impedance and the pose controller.

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 disk surface parallel to the $x_o y_o$ -plane. The final position is:

$${}^o\mathbf{p}_f = {}^o\mathbf{p}_i - {}^o\mathbf{n}({}^o\mathbf{n}^T {}^o\mathbf{p}_i - \delta),$$

where ${}^o\mathbf{p}_i = [0.5 \ 0 \ 0]^T$ m is the initial position of the end effector and $\delta = 0.033$ m is chosen to have a normal force of about 22 N at the equilibrium, with the available estimate of the environment stiffness. A trapezoidal velocity profile time-law is adopted, with a cruise velocity of 0.023 m/s. The absolute trajectory is computed from the desired relative trajectory using the current object pose estimation.

In the EKF, the non null elements of the matrix $\mathbf{\Pi}$ has been set equal to $625 \cdot 10^{-12}$ for $\mathbf{f}$, 10^{-7} for $\mathbf{n}_h$ and $6.5 \cdot 10^{-5}$ for δ_{hq} . The state noise covariance matrix has been selected so as to give a rough measure of the errors due to the simplification introduced on the model (constant velocity), by considering only velocity disturbance, i.e.

$$\mathbf{Q} = \text{diag}\{0, 0, 0, 0, 0, 0, 0, 5, 5, 0.5, 10^2, 10^3, 10^3, 10^3\} \cdot 10^{-12}.$$

Notice that the unit quaternion has been used for the orientation in the EKF, to avoid any occurrence of representation singularities. Moreover a 20 ms sampling time has been set for the estimation algorithm, corresponding to the typical camera frame rate of 50 Hz.

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

In the first simulation only the visual measurements are used. The object pose estimation errors are reported in Fig. 2. The position error is computed as the difference between the real position of the origin of the object frame and the estimated position; the orientation error is defined as the norm of the vector part of the quaternion that can be extracted from the rotation matrix representing the mutual orientation of the real object frame with respect to the estimated frame. The task starts at time $t_o = 0$ s, when an estimate of the object pose is available from visual measurements; notice that the

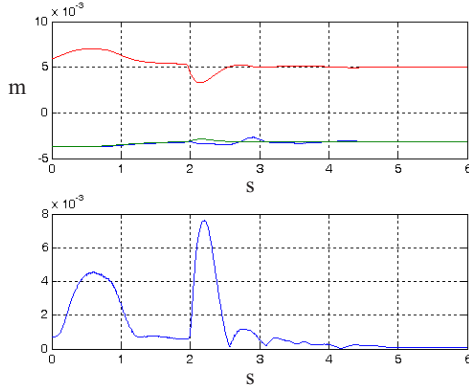


Fig. 2. Pose estimation error in the first simulation. Top: position error; bottom: orientation error.

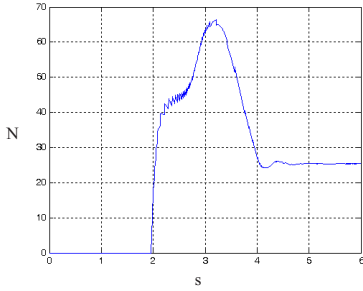


Fig. 3. Contact force in the first simulation.

initial value of the pose estimation error is non null, due to the camera calibration error. From t_0 to $t_1 \simeq 2$ s the error varies slowly due to the robot motion. At time t_1 the disk comes into contact with the plane; the abrupt change of robot velocity causes an increment of the estimation error that, after a transient, becomes constant and approximatively equal to the initial value.

The time history of the contact force in the object frame is reported in Fig. 3. Notice that the contact force is null during the motion in free space and becomes different from zero after the contact at time t_1 . The impedance control keeps the force limited during the transient while, at steady state, the force reaches a value of about 26 N, which is different from the desired value due to the presence of the estimation error along to the z_o -axis.

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. 4 and Fig. 5. Before the contact (i.e. before time t_1), the results are the same as in the previous simulation. After the contact, the benefit of using additional measurements in the EKF produces a significant reduction of the pose estimation error, especially for the z_o component and for the orientation. Moreover, the peak of the contact force is lower than before and the force value at steady state is near to the expected value of 22 N.

VII. CONCLUSION

A 6-DOF position-based visual impedance control scheme was proposed in this paper. The environment is a rigid object

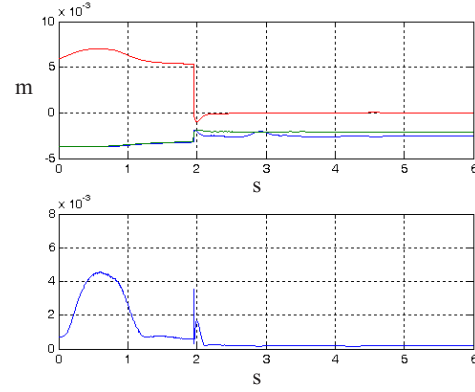


Fig. 4. Pose estimation error in the second simulation. Top: position error; bottom: orientation error.

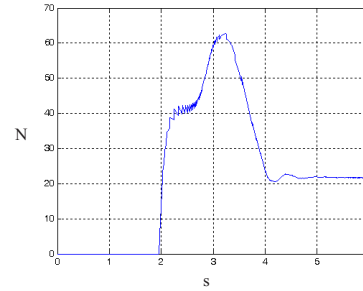


Fig. 5. Contact force in the second simulation.

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*, 1995, 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*, May 1998, 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 Systems*, October 2004, 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, no. 5, 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, 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*, May 2006.
- [10] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1996.