

Robot Force/Position Control with Force and Visual Feedback

Vincenzo Lippiello, Bruno Siciliano, and Luigi Villani

Abstract—In this paper, a force/position control law for a robot manipulator in contact with a partially known environment is proposed. The environment is a rigid object of known geometry but of unknown and time-varying pose. An algorithm for online estimation of the object pose is adopted, based on visual data provided by a camera as well as on forces measured during the interaction with the environment. This information is used by a hybrid force/position control scheme. Simulation 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 during the interaction with the environment. In fact, a robot may achieve global information on the environment using vision; 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 are satisfied.

In recent years, several approaches where force and vision measurements are combined in the same feedback control loop have been proposed, as hybrid visual/force control [1], shared and traded control [2], [3] or visual impedance control [4], [5], [6]. These algorithms improve classical interaction control schemes [7], e.g., impedance control, hybrid force/position control, parallel force/position control, where only force and joint position measurements are used.

The approach adopted in this work is based on the classical constrained hybrid force/position control [8], which requires exact knowledge of the geometry of the environment in the form of constraints imposed to the end-effector motion. This hypothesis is relaxed here, in the sense that the geometry of the environment is assumed to be known, but its position and orientation with respect to the robot end-effector are unknown. The relative pose is estimated online from all the available sensor data, i.e., visual, force and joint position measurements, using the Extended Kalman Filter (EKF). The estimated pose is then exploited to compute the constraint equations in the hybrid force/position control law.

The pose estimation algorithm is an extension of the visual tracking scheme proposed in [9] 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.

PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, Via Claudio 21, 80125 Napoli, ITALY
{lippiell, siciliano, lvillani}@unina.it

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) 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 $\mathbf{o}_o = [x_o \ y_o \ z_o]^T$ 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:

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

where $\mathbf{0}$ 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 a twice 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 $n \geq 3$. Assuming that the object frame is known, 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

$$\mathbf{v}_{P_q} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (4)$$

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

$${}^o\mathbf{v}_{P_q} = {}^o\dot{\mathbf{p}}_q + \mathbf{\Lambda}({}^o\mathbf{p}_q){}^o\boldsymbol{\nu}_o, \quad (5)$$

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. (5), ${}^o\dot{\mathbf{p}}_q$ 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 referred to 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.:

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

C. Camera

A frame $O_c\text{-}x_cy_cz_c$ 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

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

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

$${}^c\tilde{\mathbf{p}} = {}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c){}^o\tilde{\mathbf{p}} \quad (8)$$

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 (5), the absolute velocity of the origin O_o of the object frame can be computed as

$${}^c\mathbf{v}_{O_o} = {}^c\dot{\mathbf{o}}_o + \mathbf{\Lambda}({}^c\mathbf{o}_o){}^c\boldsymbol{\nu}_c, \quad (9)$$

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

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

where ${}^c\boldsymbol{\omega}_{o,c}$ 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

$${}^c\boldsymbol{\nu}_o = {}^c\boldsymbol{\nu}_{o,c} + \mathbf{\Gamma}({}^c\mathbf{o}_o){}^c\boldsymbol{\nu}_c \quad (11)$$

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 ${}^r\boldsymbol{\nu}_s$ 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

$${}^r\boldsymbol{\nu}_s = {}^r\mathbf{L}(\mathbf{x}_s)\dot{\mathbf{x}}_s \quad (12)$$

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 [10].

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

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

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, it is required the computation of the Jacobian matrix $\mathbf{J}_f = \partial \mathbf{g}_f / \partial \mathbf{x}_o$. To this purpose, the time derivative of (13) can be computed in the form

$$\dot{\mathbf{f}} = \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}_c} \dot{\mathbf{x}}_c, \quad (14)$$

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

$$\dot{\mathbf{f}} = \mathbf{J}_{o,c} {}^c\boldsymbol{\nu}_{o,c} \quad (15)$$

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 [10].

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

$$\dot{\mathbf{f}} = \mathbf{J}_{o,c} {}^c\boldsymbol{\nu}_o - \mathbf{J}_c {}^c\boldsymbol{\nu}_c \quad (16)$$

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

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

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.,

$$\mathbf{n}_h = \frac{\mathbf{h}}{\|\mathbf{h}\|}. \quad (18)$$

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

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

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 $\mathbf{J}_h = \partial \mathbf{g}_h / \partial \mathbf{x}_o$. To this purpose, the time derivative of (19) can be expressed as

$$\dot{\mathbf{n}}_h = \frac{\partial \mathbf{g}_h}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial \mathbf{g}_h}{\partial \mathbf{p}_q} \dot{\mathbf{p}}_q. \quad (20)$$

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

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

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

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

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

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

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

For the estimation of the object pose it is required the computation of the Jacobian matrix $\mathbf{J}_{hq} = \partial g_{hq} / \partial \mathbf{x}_o$. As in the previous subsection, the time derivative of δ_{hq} can be expressed as

$$\dot{\delta}_{hq} = \frac{\partial g_{hq}}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial g_{hq}}{\partial \mathbf{p}_q} \dot{\mathbf{p}}_q. \quad (24)$$

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

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

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

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

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

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

C. Extended Kalman Filter

The pose vector $\mathbf{x}_o$ 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 $\mathbf{w} = [\mathbf{x}_o^T \quad \dot{\mathbf{x}}_o^T]^T$. 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

$$\mathbf{w}_k = \mathbf{A}\mathbf{w}_{k-1} + \gamma_k, \quad (27)$$

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

$$\zeta_k = [\zeta_{f,k}^T \quad \zeta_{h,k}^T \quad \zeta_{hq,k}^T]^T,$$

where $\zeta_{f,k} = \mathbf{f}_k + \boldsymbol{\mu}_{f,k}$, $\zeta_{h,k} = \mathbf{h}_k + \boldsymbol{\mu}_{h,k}$, 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:

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

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

$$\mathbf{g}(\mathbf{w}_k) = [\mathbf{g}_f^T(\mathbf{w}_k) \quad \mathbf{g}_h^T(\mathbf{w}_k) \quad g_{hq}^T(\mathbf{w}_k)]^T \quad (28)$$

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 (17), (22), and (26) 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.

IV. HYBRID FORCE/POSITION CONTROL

The proposed algorithm can be used to estimate online the pose of an object in the workspace; this allows to compute the constraint (2) with respect to the base frame in the form

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

where the last equality holds in view of the direct kinematic equation of the robot manipulator (3). In the following, it is assumed that the object does not move; the general case of moving object is more complex but can be analyzed in a similar way. Hence, the constraint equation can be rewritten in the form

$$\varphi(\mathbf{q}) = 0, \quad (29)$$

hence $\mathbf{J}_\varphi(\mathbf{q})\dot{\mathbf{q}} = 0$, where $\mathbf{J}_\varphi = \partial \varphi/\partial \mathbf{q}$ is a $(1 \times n)$ vector.

The dynamic model of a robot manipulator in contact with the environment can be written in the Lagrangian form as

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} - \mathbf{J}_\varphi^T(\mathbf{q})\lambda,$$

where $\mathbf{B}$ is the $(n \times n)$ symmetric and positive definite inertia matrix, $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$ is the $(n \times 1)$ vector taking into account Coriolis, centrifugal, friction and gravity torques, $\boldsymbol{\tau}$ is the $(n \times 1)$ vector of the joint torques, and λ is the lagrangian multiplier associated to the constraint (29). The scalar λ can be computed from the measured contact force in the form

$$\lambda = \mathbf{J}_\varphi^\dagger(\mathbf{q})\mathbf{J}^T(\mathbf{q})\mathbf{h},$$

where $\mathbf{J}_\varphi^\dagger(\mathbf{q})$ is a pseudo-inverse of $\mathbf{J}_\varphi^T(\mathbf{q})$ and $\mathbf{J}(\mathbf{q})$ is the robot Jacobian matrix defined in (4).

According to the hybrid force/position control strategy, it is useful to apply the change of coordinates

$$\mathbf{r} = \begin{bmatrix} r_F \\ \mathbf{r}_P \end{bmatrix} = \begin{bmatrix} \varphi(\mathbf{q}) \\ \boldsymbol{\psi}(\mathbf{q}) \end{bmatrix} \quad (30)$$

where $\boldsymbol{\psi}(\mathbf{q})$ is a suitable $((n-1) \times 1)$ vector function, assumed to be twice differentiable and such that $\varphi(\mathbf{q})$ and the $n-1$ components of $\boldsymbol{\psi}$ are linearly independent at least locally in a neighborhood of the operating point. Hence, the inverse transformation of (30) is well defined; moreover, differentiation of (30) gives $\dot{\mathbf{r}} = \mathbf{J}_r(\mathbf{q})\dot{\mathbf{q}}$, where the matrix $\mathbf{J}_r(\mathbf{q})$ is nonsingular. Notice that, in the new coordinates $\mathbf{r}$, the constraint equation (29) is equivalent to $r_F = 0$. Moreover, the vector $\mathbf{r}_P$ allows to specify any end-effector position and orientation which satisfies the constraint. On the other hand, the force vector in the new coordinates $\mathbf{r}$ is

$$\mathbf{F} = \begin{bmatrix} \lambda \\ \mathbf{0} \end{bmatrix}$$

and $\dot{\mathbf{r}}$ and $\mathbf{F}$ are orthogonal, i.e., $\mathbf{F}^T \dot{\mathbf{r}} = 0$ for any $\dot{\mathbf{r}}$ that satisfy the constraint (such that $\dot{r}_F = 0$).

In sum, a force/position control task can be assigned by specifying the desired force $\lambda_d(t)$ and the $n-1$ components of the vector $\mathbf{r}_{Pd}(t)$. An inverse dynamics control law can be adopted, by choosing the control torque $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\alpha_r + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{J}_\varphi^T(\mathbf{q})\mathbf{h}_\lambda,$$

where α_r and h_λ are control inputs to be properly designed. It is possible to recognize that the choice (see, e.g., [12])

$$\alpha_r = J_r^{-1}(q)(a_r + \dot{J}_r(q)) \quad a_r = \begin{bmatrix} 0 \\ a_P \end{bmatrix}$$

leads to $\ddot{r} = a_r$, where a_r attains the meaning of a resolved acceleration in terms of the new coordinate vector r . Notice that, by virtue of the choice of a_r , this equation corresponds to the equations $\ddot{r}_F = 0$ and $\ddot{r}_P = a_P$. The solution of the first equation is $r_F = 0$ and $\dot{r}_F = 0$, assuming that the system is initialized on the constraint. The resolved acceleration can be chosen as

$$a_P = \ddot{r}_{Pd} + K_{Dr}(\dot{r}_{Pd} - \dot{r}_P) + K_{Pr}(r_{Pd} - r_P), \quad (31)$$

while the control input h_λ can be chosen as

$$f_\lambda = \lambda_d + k_{I\lambda} \int_0^t (\lambda_d(\tau) - \lambda(\tau)) d\tau \quad (32)$$

where K_{Dr} , K_{Pr} and $k_{I\lambda}$ are suitable feedback gains. Details concerning the stability of the constrained hybrid force/position control can be found, e.g., in [12].

V. CASE STUDY

A planar object surface is considered, described by the equation ${}^o n^T o p = 0$, 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 ${}^o n$. During the interaction with the robot, the normal vector n remains constant in the base frame while the plane is assumed to be elastically compliant along n according to a simple elastic law. The contact force of the object on the robot's tip at p_q is given by

$$h = \begin{cases} k n n^T (p_o - p_q) & \text{if } n^T (p_o - p_q) \geq 0 \\ 0 & \text{if } n^T (p_o - p_q) < 0 \end{cases}$$

where p_q is on the plane when $h \neq 0$ while p_o is a constant vector representing the position of a point of the plane when $h = 0$. 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. The robot has a force/torque sensor mounted at the wrist. Neglecting the weight and inertia of the tool, the force at the robot's tip p_q and that measured at the robot wrist by the force sensor are the same, while a moment is measured by the sensor if the stick is not aligned to ${}^o n$.

A camera is mounted on the robot end effector. It is assumed that the intrinsic parameters of the camera are affected by a 3% error, while the extrinsic parameters are supposed to be known with a tolerance of ± 1 cm for the position and ± 3 deg for the orientation. The object features are 4 points at the corners of a square of 10 cm side.

The control parameters in (31) and (32) are chosen as: $K_{Pr} = \text{diag}\{250, 250, 250, 15, 15, 15\}$, $K_{Dr} =$

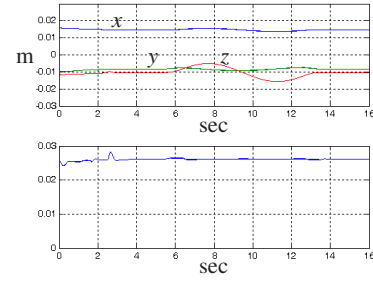


Fig. 1. Pose estimation error in the first simulation. Top: position error; bottom: orientation error.

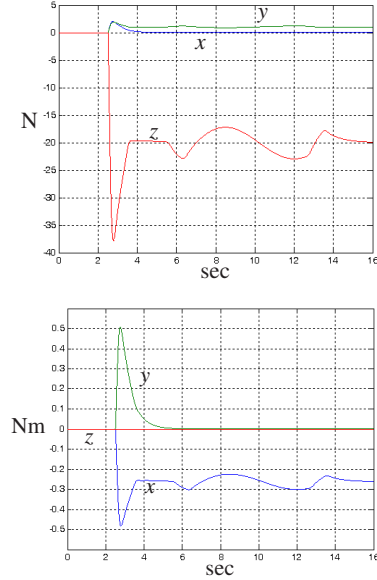


Fig. 2. Measured force (top) and moment (bottom) in the first simulation.

$\text{diag}\{220, 220, 220, 10, 10, 10\}$, $k_{I\lambda} = 0.05$; a 1 ms sampling time has been selected.

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 the estimated rest position of the plane. During this phase, only the end-effector position and orientation are controlled, according to the law (31') applied to all the task directions. When force is detected, the motion controller is replaced by the hybrid force/position control law (31) and (32), with a constant position and orientation and a desired force of -20 N along the z_o axis. After 2 sec, a desired circular trajectory on the contact plane with 10 cm radius and 8 sec duration is commanded for the end-effector position, while the desired orientation is unchanged and the desired force is kept constant to -20 N.

In the EKF, the non null elements of the matrix Π has been set equal to $625 \cdot 10^{-11}$ for f , 10^{-5} for n_h and 10^{-6} 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.

$$Q = \text{diag}\{0, 0, 0, 0, 0, 0, 0, 5, 5, 0.5, 10^2, 10^3, 10^3, 10^3\} \cdot 10^{-11}.$$

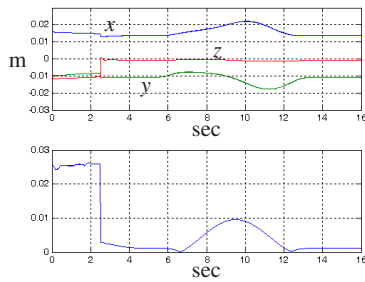


Fig. 3. Pose estimation error in the second simulation. Top: position error; bottom: orientation error.

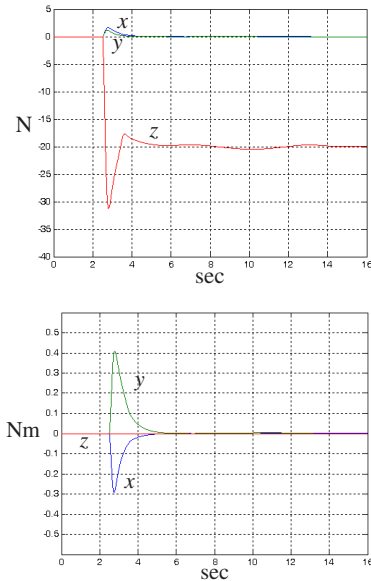


Fig. 4. Measured force (top) and moment (bottom) in the second simulation.

Notice that the unit quaternion has been used for the orientation in the EKF. 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. 1. The position error is computed as the difference between the real position of the origin of the object frame and the estimated position referred to the object frame; the orientation error is defined as the norm of the vector part of the quaternion 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 initial value of the pose estimation error is non null, due to camera calibration errors and remains different from zero during all the task.

The time history of the force and moments measured by the sensor are reported in Fig. 2. Notice that the normal

component of the measured force (the z component), after a transient, reaches a value close to the desired one. However, the y component of the force remains non null and a moment different from zero can be observed during the task execution, due to the error in the estimation of z_o .

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. 3 and Fig. 4. Before the contact (i.e. before time $t_c \simeq 2.2$ s), the estimation errors 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 component of the position (aligned to z_o) and for the orientation. The peak of the measured force during the transient is lower than before and the value of the normal component of the force, during the execution of the circle, remains very close to the desired value of -20 N. Finally, all the other components of the measured force, as well as the measured moment, become null after a short transient, meaning that the stick remains aligned to z_o .

VI. CONCLUSION

A 6-DOF hybrid force/position 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. Simulation results have demonstrated the superior performance of the proposed control scheme with respect to an algorithm using only visual measurements for the estimation of the object pose.

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 Conf.*, 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 Systems*, pp. 798–803.
- [6] V. Lippiello, B. Siciliano, L. Villani, "Robot interaction control using force and vision," *2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1470–1475.
- [7] B. Siciliano and L. Villani, *Robot Force Control*. Boston, Ma: Kluwer Academic Publishers, 1999.
- [8] T. Yoshikawa, T. Sugie, and M. Tanaka, "Dynamic hybrid position/force control of robot manipulators—Controller design and experiment," *IEEE J. of Robotics and Automation*, vol. 4, pp. 699–705, 1988.
- [9] V. Lippiello and L. Villani, "Managing redundant visual measurements for accurate pose tracking," *Robotica*, vol. 21, pp. 511–519, 2003.
- [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*, pp. 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, no. 3, pp. 313–326, 1996.
- [12] C. Canudas de Wit, B. Siciliano, and G. Bastin (Eds.), *Theory of Robot Control*, Springer-Verlag, London, 1996.