
A Framework for Force and Visual Control of Robot Manipulators

Vincenzo Lippiello, Bruno Siciliano, Luigi Villani

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

Abstract. A framework for force and visual control of robot manipulators in contact with a partially known environment is proposed. The environment is modelled as 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 as well as on force measurements. This information is used by a force/position controller. The resulting control scheme has a inner/outer structure where the outer loop performs pose estimation and the inner loop is devoted to interaction control.

1 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, visual perception allows to achieve global information on the surrounding environment to 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.

Several approaches to robot manipulators control combining force and vision feedback have been developed so far, e.g., hybrid visual/force control [1], shared and traded control [2, 3] or visual impedance control [4, 5]. These algorithms improve classical interaction control schemes [6], e.g., impedance control, hybrid force/position control, parallel force/position control, where only force and joint position measurements are used.

In general, interaction control approaches require knowledge of the geometry of the environment in the form of constraints imposed to the end-effector motion. Some of these approaches, like hybrid force/position control [7], make explicit use of the constraint equations for the design of the control law; other

approaches, like impedance control [8] and parallel force/position control [9], exploit the available knowledge of the geometry of the environment for the selection of the control gains and the choice of the reference force and position trajectories.

In the framework presented in this paper, 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 for computing online the constraint equations used by standard interaction control laws.

The pose estimation algorithm is an extension of the visual tracking scheme proposed in [10] 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.

2 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.

2.1 Object

The position and orientation of a frame $O_o-x_o y_o z_o$ attached to a rigid object with respect to a base coordinate frame $O-xyz$ can be expressed in terms of the coordinate vector of the origin $\mathbf{o}_o$ 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. 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 $\tilde{\mathbf{p}} = \mathbf{H}_o(\mathbf{x}_o) {}^o\tilde{\mathbf{p}}$, where ${}^o\tilde{\mathbf{p}}$ 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.

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 $\varphi({}^o\mathbf{p}) = 0$. Hence, 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 surface equation $\varphi({}^o\mathbf{p}_q) = 0$. Moreover, the (3×1) contact force ${}^o\mathbf{h}$ is aligned to the normal unit vector ${}^o\mathbf{n}$.

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

$${}^o\mathbf{v}_{P_q} = {}^o\dot{\mathbf{p}}_q + \mathbf{A}({}^o\mathbf{p}_q){}^o\boldsymbol{\nu}_o, \quad (2)$$

with $\mathbf{A}(\cdot) = [\mathbf{I}_3 \quad -\mathbf{S}(\cdot)]$, where $\mathbf{I}_3$ is the (3×3) identity matrix and $\mathbf{S}(\cdot)$ denotes the (3×3) skew-symmetric matrix operator. In Eq. (2), ${}^o\dot{\mathbf{p}}_q$ 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 $\boldsymbol{\omega}_o$. 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., ${}^o\mathbf{n}^T({}^o\mathbf{p}_q){}^o\dot{\mathbf{p}}_q = 0$.

2.3 Camera

A frame $O_c-x_c y_c z_c$ attached to the camera 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 $[X \ Y]^T = \lambda_c [x/z \ y/z]^T$ where λ_c is the lens focal length.

The homogeneous coordinate vector of P with respect to the camera frame can be expressed as ${}^c\tilde{\mathbf{p}} = {}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c){}^o\tilde{\mathbf{p}}$. 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 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 ${}^c\boldsymbol{\nu}_c = [{}^c\mathbf{v}_{O_c}^T \quad {}^c\boldsymbol{\omega}_c^T]^T$. It can be shown that the velocity screw ${}^c\boldsymbol{\nu}_o = [{}^c\mathbf{v}_{O_o}^T \quad {}^c\boldsymbol{\omega}_o^T]^T$ corresponding to the absolute motion of the object frame can be expressed as

$${}^c\boldsymbol{\nu}_o = {}^c\boldsymbol{\nu}_{o,c} + \boldsymbol{\Gamma}({}^c\mathbf{o}_o){}^c\boldsymbol{\nu}_c \quad (3)$$

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

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

being $\mathbf{O}_3$ 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

$${}^r\boldsymbol{\nu}_s = {}^r\mathbf{L}(\mathbf{x}_s)\dot{\mathbf{x}}_s \quad (4)$$

where ${}^r\mathbf{L}(\cdot)$ is a Jacobian matrix depending on the particular choice of coordinates for the orientation.

3 Object pose estimation

When the robot moves in free space, the unknown object pose can be estimated online by using vision; 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.

3.1 Vision

Vision is used to measure the image features, characterized by a set of scalar parameters f_j grouped in a vector $\mathbf{f} = [f_1 \cdots f_k]^T$. 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

$$\mathbf{f} = \mathbf{g}_f({}^c\mathbf{H}_o(\mathbf{x}_o, \mathbf{x}_c)), \quad (5)$$

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 (5) can be computed in the form

$$\dot{\mathbf{f}} = \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}_c} \dot{\mathbf{x}}_c, \quad (6)$$

where the second term in the right hand side is null for eye-to-hand cameras. On the other hand, the time derivative of (5) can be expressed also in the form $\dot{\mathbf{f}} = \mathbf{J}_{o,c} {}^c \boldsymbol{\nu}_{o,c}$ 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 [11]. Taking into account the velocity composition (3), vector $\dot{\mathbf{f}}$ can be rewritten in the form

$$\dot{\mathbf{f}} = \mathbf{J}_{o,c} {}^c \boldsymbol{\nu}_o - \mathbf{J}_c {}^c \boldsymbol{\nu}_c \quad (7)$$

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. (4), the comparison of (7) with (6) yields

$$\mathbf{J}_f = \mathbf{J}_{o,c} {}^c \mathbf{L}(\mathbf{x}_o). \quad (8)$$

3.2 Force

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.,

$$\mathbf{n}_h = \mathbf{h} / \|\mathbf{h}\|. \quad (9)$$

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

$$\mathbf{n}_h = \mathbf{R}_o {}^o \mathbf{n}({}^o \mathbf{p}_q) = \mathbf{g}_h(\mathbf{x}_o, \mathbf{p}_q), \quad (10)$$

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 (10) can be expressed as

$$\dot{\mathbf{n}}_h = \frac{\partial \mathbf{g}_h}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial \mathbf{g}_h}{\partial \mathbf{p}_q} \dot{\mathbf{p}}_q, \quad (11)$$

but also as

$$\dot{\mathbf{n}}_h = \dot{\mathbf{R}}_o {}^o \mathbf{n}({}^o \mathbf{p}_q) + \mathbf{R}_o {}^o \mathbf{N}({}^o \mathbf{p}_q) \dot{{}^o \mathbf{p}}_q, \quad (12)$$

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 (2). Hence, comparing (11) with (12) and taking into account (4) 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:

$$\mathbf{J}_h = -[\mathbf{N} \quad \mathbf{S}(\mathbf{n}_h) - \mathbf{N} \mathbf{S}(\mathbf{p}_q - \mathbf{o}_o)] \mathbf{L}(\mathbf{x}_o), \quad (13)$$

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

3.3 Joint position

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. In particular, it is significant computing the scalar

$$\delta_{hq} = \mathbf{n}_h^T \mathbf{p}_q = g_{hq}(\mathbf{x}_o, \mathbf{p}_q), \quad (14)$$

using also the force measurements via Eq. (9). For the estimation of the object pose it is required the computation of the Jacobian matrix $\mathbf{J}_{hq} = \partial g_{hq} / \partial \mathbf{x}_o$. The time derivative of δ_{hq} can be expressed as

$$\dot{\delta}_{hq} = \frac{\partial g_{hq}}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial g_{hq}}{\partial \mathbf{p}_q} \dot{\mathbf{p}}_q, \quad (15)$$

but also as $\dot{\delta}_{hq} = \dot{\mathbf{n}}_h^T \mathbf{p}_q + \mathbf{n}_h^T \mathbf{R}_o({}^o\dot{\mathbf{p}}_q + \mathbf{A}({}^o\mathbf{p}_q){}^o\boldsymbol{\nu}_o)$ where the expression of the absolute velocity of the point P_q in (2) has been used. Using the identity ${}^o\mathbf{n}^T({}^o\mathbf{p}_q){}^o\dot{\mathbf{p}}_q = 0$, this equation can be rewritten as

$$\dot{\delta}_{hq} = \mathbf{p}_q^T \dot{\mathbf{n}}_h + \mathbf{n}_h^T \mathbf{A}(\mathbf{p}_q - \mathbf{o}_o) \boldsymbol{\nu}_o. \quad (16)$$

Hence, comparing (15) with (16) and taking into account (12), (13) and (4), the following expression can be found

$$\mathbf{J}_{hq} = \mathbf{p}_q^T \mathbf{J}_h + \mathbf{n}_h^T \mathbf{A}(\mathbf{p}_q - \mathbf{o}_o) \mathbf{L}(\mathbf{x}_o). \quad (17)$$

3.4 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. Hence, the discrete-time dynamic model can be written as $\mathbf{w}_k = \mathbf{A}\mathbf{w}_{k-1} + \boldsymbol{\gamma}_k$, where $\boldsymbol{\gamma}$ is the dynamic modelling error described by zero mean Gaussian noise with covariance $\mathbf{Q}$ and $\mathbf{A}$ 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 EKF, in the case that all the available data can be used, is the vector $\boldsymbol{\zeta}_k = [\boldsymbol{\zeta}_{f,k}^T \quad \boldsymbol{\zeta}_{h,k}^T \quad \boldsymbol{\zeta}_{hq,k}^T]^T$ of the measurements at time kT_s , where $\boldsymbol{\zeta}_{f,k} = \mathbf{f}_k + \boldsymbol{\mu}_{f,k}$, $\boldsymbol{\zeta}_{h,k} = \mathbf{h}_k + \boldsymbol{\mu}_{h,k}$, and $\boldsymbol{\zeta}_{hq,k} = \delta_k + \mu_{hq,k}$, being $\boldsymbol{\mu}$ the measurement noise, which is assumed to be zero mean Gaussian noise with covariance $\boldsymbol{\Pi}$.

Taking into account the Eqs. (5), (10), and (14), the output equation of the EKF can be written as $\zeta_k = \mathbf{g}(\mathbf{w}_k) + \boldsymbol{\mu}_k$, with $[\boldsymbol{\mu}_{f,k}^T \quad \boldsymbol{\mu}_{h,k}^T \quad \boldsymbol{\mu}_{hq,k}^T]^T$ and $\mathbf{g}(\mathbf{w}_k) = [\mathbf{g}_f^T(\mathbf{w}_k) \quad \mathbf{g}_h^T(\mathbf{w}_k) \quad \mathbf{g}_{hq}^T(\mathbf{w}_k)]^T$, where only the explicit dependence on the state vector $\mathbf{w}_k$ has been evidenced. The EKF 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. The Jacobian matrix $\partial \mathbf{g}(\mathbf{w})/\partial \mathbf{x}_o$, in view of (8), (13), and (17) has the expression $\partial \mathbf{g}(\mathbf{w})/\partial \mathbf{x}_o = [\mathbf{J}_f^T \quad \mathbf{J}_h^T \quad \mathbf{J}_{hq}^T]^T$. The equations of the recursive form of the EKF are standard and are omitted here and can be found, e.g., in [10].

4 Interaction control

The proposed algorithm can be used to estimate online the pose of an object in the workspace; this allows to compute the surface equation 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. 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 is $\varphi(\mathbf{q}) = 0$; moreover $\mathbf{J}_\varphi(\mathbf{q})\dot{\mathbf{q}} = 0$, where $\mathbf{J}_\varphi = \partial \varphi/\partial \mathbf{q}$ is a $(1 \times n)$ vector.

The dynamic model of the manipulator in contact with the environment is

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

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.

The online computation of the constraint equations can be suitably exploited for interaction control. In the following, the case of the hybrid force/position control and of the impedance control are considered.

4.1 Hybrid force/position control

According to the hybrid force/position control strategy, the knowledge of the constraint equation allows to apply the change of coordinates

$$\mathbf{r} = \begin{bmatrix} r_F \\ r_P \end{bmatrix} = \begin{bmatrix} \varphi(\mathbf{q}) \\ \boldsymbol{\psi}(\mathbf{q}) \end{bmatrix} \quad (18)$$

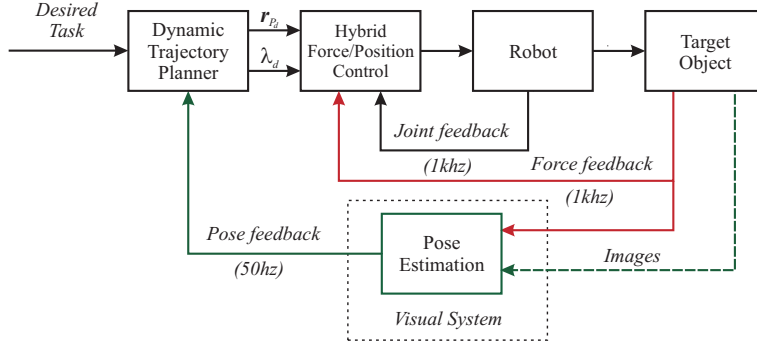


Fig. 1. Block scheme of the complete hybrid force/position control algorithm

where $\psi(\mathbf{q})$ 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 ψ are linearly independent at least locally in a neighborhood of the operating point. Hence, the inverse transformation of (18) is well defined; moreover, differentiation of (18) gives $\dot{\mathbf{r}} = \mathbf{J}_r(\mathbf{q})\dot{\mathbf{q}}$, where the matrix $\mathbf{J}_r(\mathbf{q})$ is nonsingular.

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 (see, e.g., [12]), by choosing the control torque $\boldsymbol{\tau}$ as

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

with $\boldsymbol{\alpha}_r = \mathbf{J}_r^{-1}(\mathbf{q})(\mathbf{a}_r - \dot{\mathbf{J}}_r(\mathbf{q})\dot{\mathbf{q}})$, $\mathbf{a}_r = [0 \quad \mathbf{a}_P^T]^T$,

$$\mathbf{a}_P = \ddot{\mathbf{r}}_{Pd} + \mathbf{K}_{Dr}(\dot{\mathbf{r}}_{Pd} - \dot{\mathbf{r}}_P) + \mathbf{K}_{Pr}(\mathbf{r}_{Pd} - \mathbf{r}_P)$$

and

$$h_\lambda = \lambda_d + k_{I\lambda} \int_0^t (\lambda_d(\tau) - \lambda(\tau))d\tau,$$

where $\mathbf{K}_{Dr}$, $\mathbf{K}_{Pr}$ and $k_{I\lambda}$ are suitable feedback gains. The scalar λ can be computed from the measured contact force in the form $\lambda = \mathbf{J}_\varphi^{T\dagger}(\mathbf{q})\mathbf{J}^T(\mathbf{q})\mathbf{h}$, where $\mathbf{J}_\varphi^{T\dagger}$ is a weighted pseudo-inverse of $\mathbf{J}_\varphi^T$ and $\mathbf{J}$ is the robot Jacobian.

From the block scheme of Fig. 1 it can be observed that the algorithm has a inner/outer structure, where the inner loop implements hybrid control whereas the outer loop computes the estimation of the object pose as well as the desired force and motion trajectories, on the basis of force, joint and visual measurements. Usually, the outer loop runs at a frequency lower than the inner loop, due to the limitation in the maximum camera frame rate (between 25 Hz and 60 Hz for low-cost cameras used in standard applications).

4.2 Impedance Control

The above scheme can be easily modified by replacing the inner hybrid force/position control with a different interaction control scheme. For instance,

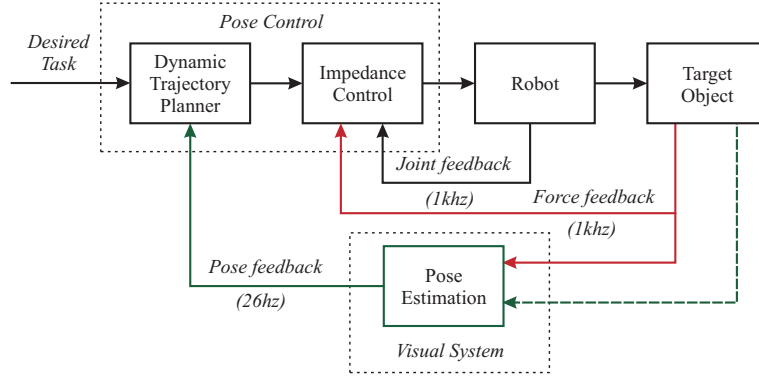


Fig. 2. Position-based visual impedance control.

a position-based impedance control algorithm, based on the concept of compliant frame [6], can be adopted (see Fig. 2).

In detail, on the basis of the current estimate of the constraint equation, the Dynamic Trajectory Planner generates a pose trajectory for a desired end-effector frame specified in terms of the position of the origin $\mathbf{p}_d$ and orientation matrix $\mathbf{R}_d$. Moreover, 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 contain 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 motion controller of the robot end effector. The details about the impedance equations can be found in [6].

It is interesting noticing that the above scheme is able to manage both contact and free-space motion phases. In fact, during free space motion, the position-based impedance control is equivalent to a pose control and the whole scheme corresponds to a classical position-based visual servoing algorithm [13]. For this reason, this scheme can be classified as position-based visual impedance control.

5 Conclusion

A framework for force and visual control of robot manipulators was proposed in this paper. The environment is assumed as a rigid object of known geometry but unknown and possibly time-varying position and orientation. An estimation algorithm is adopted, based on visual, force and joint position data, which allows to compute the actual position and orientation of the object so as to update in real time the constraint equations for the robot end effector. The re-

sulting control has a inner/outer structure where the outer loop performs pose estimation and the inner loop is an interaction control scheme. Simulation and experimental results presented in previous papers [14, 15] have demonstrated the superior performance of the proposed approach with respect to algorithms based on force measurements or visual measurements separately.

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 System*, pp. 798–803.
6. B. Siciliano and L. Villani, *Robot Force Control*. Boston, Ma: Kluwer Academic Publishers, 1999.
7. 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.
8. N. Hogan, "Impedance control: An approach to manipulation, Parts I–III," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–24, 1985.
9. S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Trans. on Robotics and Automation*, vol. 9, pp. 361–373, 1993.
10. V. Lippiello and L. Villani, "Managing redundant visual measurements for accurate pose tracking," *Robotica*, vol. 21, pp. 511–519, 2003.
11. B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. 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.
13. V. Lippiello, B. Siciliano, and L. Villani, "Position-based visual servoing in industrial multi-robot cells using a hybrid camera configuration," *IEEE Trans. on Robotics*, pp. 73–86, 2007.
14. V. Lippiello, B. Siciliano, and L. Villani, "A position-based visual impedance control for robot manipulators," *2007 IEEE Int. Conf. on Robotics and Automation*, pp. 2068–2073.
15. V. Lippiello, B. Siciliano, and L. Villani, "Robot force/position control with force and visual feedback," *European Control Conference 2007*, Kos, Greece.