

Force and Visual Control for Safe Human–robot Interaction

Bruno Siciliano, Luigi Villani, Vincenzo Lippiello, Agostino De Santis

Abstract Unlike the industrial robotics domain where the workspace of machines and humans can be segmented, applications of intelligent machines that work in contact with humans are increasing, which involve e.g. haptic interfaces and tele-operators, cooperative material-handling, power extenders and such high-volume markets as rehabilitation, physical training and entertainment. Force and vision play a fundamental role to increase the autonomy of a robotic system, especially in the presence of humans. Vision provides global information on the surrounding environment to be used for motion planning and obstacle avoidance, while force allows adjusting the robot motion so that the local constraints imposed by the environment are satisfied. In order to avoid dangerous collisions and ensure a safe interaction, suitable control strategies based on force and visual feedback can be used while tracking human motion. This paper surveys such strategies and presents some experimental results in a number of significant case studies.

1 Introduction

The extension of application domains of robotics from factories to human environments leads to implementing proper strategies for close interaction between people and robots. On the one hand, small-scale industrial robots have to learn to get along with human co-workers in factories, while, on the other hand, service robots are a solution for automating common daily tasks in domestic environments, due to lack or high cost of human workforce.

The size of an industrial robot, or the necessary autonomous behaviour of a service robot, can result in dangerous situations for humans co-habiting the robot operational domain. Therefore, physical issues must be carefully considered, since “nat-

PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Naples, Italy; e-mail: {bruno.siciliano, luigi.villani, vincenzo.lippiello, agodesa}@unina.it.

ural” or unexpected behaviours of people during interaction with robots can result in injuries, which may be severe, when considering the current mechanical structure of robots available on the market [1].

One possible issue to consider, in order to increase safety, is the proper use of the two main “senses”: vision and touch. Vision-and-force-based control for physical interaction may include collision avoidance, control of close interaction and fusion with other sensory modes, which all may lead to improving robots performance, without necessarily considering a novel mechanical design.

However, the need for safety suggests complementing the control system with the adoption of compliant components in the structure. Compliance can be introduced at the contact point by a soft covering of the whole arm with visco-elastic materials or by adopting compliant transmissions at the robot joints. Increasing in this way the robot mechanical compliance while reducing its overall apparent inertia can be realized through different elastic actuation/transmission arrangements which include: relocation of actuators close to the robot base and transmission of motion through steel cables and pulleys, combination of harmonic drives and lightweight link design, and use of parallel and distributed macro-mini [2] or variable-impedance [3] actuation. Other improvements for anticipating and reacting to collisions can be achieved through the use of combinations of external/internal robot sensing, electronic hardware and software safety procedures, which intelligently monitor, supervise, and control manipulator operation.

This paper focuses on techniques for augmenting safety by means of control systems. Human-like capabilities in close interaction can be considered as mimicking human sensing and actuation. This leads to considering fully integrated vision-and-force-based control. Thanks to 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 robot allows adjusting the motion so that the local constraints imposed by the environment during the interaction are satisfied. Moreover, a vision system may substitute the complex infrastructure needed for “intelligent environments” [4] to detect and track people in the operational domain.

In the last years, several papers on this subject have been presented. Some of them combine force or vision in the same feedback control loop, such as hybrid visual/force control [5], shared and traded control [6, 7] or visual impedance control [8, 9]. These algorithms improve classical interaction control schemes [10], e.g., impedance control, hybrid force/position control, parallel force/position control, where only force and joint position measurements are used.

The approach presented here, for the part concerning interaction control based on force and vision, was developed in previous publications [11, 12]. The collision avoidance technique is based on the “skeleton algorithm” reported in [13].

2 Modeling

For physical human–robot interaction (pHRI) it is necessary to model, or track human motion, to formulate a model of robot motion and of the objects to interact with. 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). Some modeling assumption concerning the human user, the environment, the robot and the camera are illustrated below.

2.1 Human User

Positioning of critical parts of a human body may be addressed, like for robots, considering the kinematics of the structure. However, joint measures are not available on the human body; therefore, exteroceptive sensing by means of cameras is used, obtaining the position in the space of some relevant features (hands, head etc.). This leads to finding a simplified kinematic model, to be updated in real time, with the novel “skeleton algorithm” [13]. This algorithm considers a skeleton, composed of segments, as a simplified model of a human (or a robot or even an object), exploiting the simple geometric structures in order to compute the distances between the segments, which can be used for collision avoidance, considering all the points of the articulated structure of humans and robots which may collide. For every link of the skeleton of a human figure, the closest point to the robot or the object to be avoided is computed. The distance information between the two closest points of human and obstacle can be used to avoid a collision, using “spheres” located in the selected closest points as protective hulls: these spheres can have a finite or infinite radius and can be the source of repelling forces shaped as effects of virtual springs or potential fields. Summarizing, the steps of the algorithm are:

- Create a skeleton of the human body, by using vision, and of the robot, by using direct kinematics in order to find the extremal point of the segments.
- Compute the distances between the different segments, finding also the two closest points for each pair of links.
- Define intensity and shape of repelling forces between these two points and use them as reference values in the position/force control system.

Almost all structures can be encapsulated by a finite skeleton with spheres, as sketched in Fig. 1 for the DLR arm [14]. The position of the closest point on each link (continuous curves) varies continuously, preserving continuity of reference values for any kind of control scheme. The key point of the proposed approach is that only the two closest points (on each link) of the structure are considered each time, leading to a simple generation of the Cartesian desired velocity (or force) for only one of these points, which eventually is transformed in the corresponding joint trajectory via proper inverse kinematics (or kinetics). Any point on the structure can be considered as a control point. To simplify the problem, there is also the possibility to

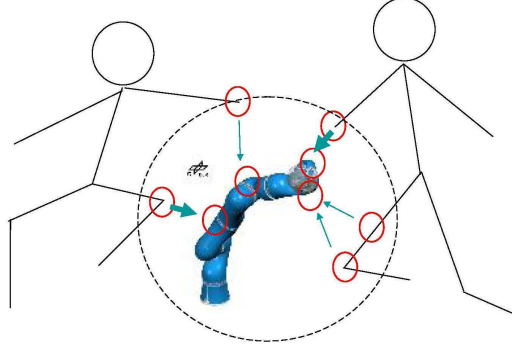


Fig. 1 Exemplification of the skeleton algorithm for the DLR lightweight arm

choose only a subset of control points, e.g., the articulation of the robot [15]. Moreover, it is possible to use an inverse kinematics, an impedance control or whatever is desired, since the algorithm just adds with continuity repelling forces or velocity, preserving stability of the control loops used for the system.

2.2 Environment

The position and orientation of a frame $O_o-x_o y_o z_o$ attached to the environment (assumed as 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. In the case that a minimal representation of the orientation is adopted, e.g., Euler angles, $p = 3$, while $p = 4$ 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 $\tilde{\mathbf{p}} = \mathbf{H}_o(\mathbf{x}_o) {}^o\tilde{\mathbf{p}}$, 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 ${}^o\mathbf{n}$.

2.3 Robot

The case of an n -joint 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 tip $\mathbf{v}_{P_q}$ can be expressed as $\mathbf{v}_{P_q} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ where $\mathbf{J}(\mathbf{q})$ 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, also known as cross-product matrix. 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 with the object, the normal component of the relative velocity ${}^o\dot{\mathbf{p}}_q$ is null, i.e., ${}^o\mathbf{n}^T({}^o\mathbf{p}_q){}^o\dot{\mathbf{p}}_q = 0$.

2.4 Camera

A frame $O_c\text{--}x_c y_c z_c$ 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

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

where ${}^c\boldsymbol{\nu}_{o,c} = [{}^c\dot{\boldsymbol{o}}_o^T \quad {}^c\boldsymbol{\omega}_{o,c}^T]^T$ 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

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

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

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

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

3 Use of Vision, Force and Joint Positions Measurements

When the robot moves in free space, the unknown object pose and the position of the head of a human user can be estimated online by using the data provided by the camera; when the robot is in contact with the object, also the force measurements and the joint position measurements are used. Joint values are used for evaluating the position of the links for collision avoidance. The equations mapping the measurements to the unknown position and orientation of the object are derived below.

3.1 Vision

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

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

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, the computation of the Jacobian matrix $\boldsymbol{J}_f = \partial \boldsymbol{g}_f / \partial \boldsymbol{x}_o$ is required. To this end, the time derivative of Eq. (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 on the right-hand side is null for eye-to-hand cameras. On the other hand, the time derivative of Eq. (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 [16]. By taking into account the velocity composition in Eq. (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 [16]. In view of Eq. (4), comparing Eq. (7) with Eq. (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

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

with ${}^o\mathbf{p}_q = \mathbf{R}_o^T(\mathbf{p}_q - \mathbf{o}_o)$. For the estimation of the object pose, the computation of the Jacobian matrix $\mathbf{J}_h = \partial \mathbf{g}_h / \partial \mathbf{x}_o$ is required. To this end, the time derivative of Eq. (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

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

where ${}^o\mathbf{N}({}^o\mathbf{p}_q) = \partial {}^o\mathbf{n} / \partial {}^o\mathbf{p}_q$ depends on the surface curvature and ${}^o\dot{\mathbf{p}}_q$ can be computed as

$${}^o\dot{\mathbf{p}}_q = \mathbf{R}_o^T(\dot{\mathbf{p}}_q - \dot{\mathbf{o}}_o + \mathbf{S}(\mathbf{p}_q - \mathbf{o}_o) \boldsymbol{\omega}_o).$$

Hence, by replacing the above expression in Eq. (12) and taking into account the equality $\dot{\mathbf{R}}_o^o \mathbf{n}({}^o\mathbf{p}_q) = -\mathbf{S}(\mathbf{n}_h)\boldsymbol{\omega}_o$, the following equality holds:

$$\dot{\mathbf{n}}_h = \mathbf{N}\dot{\mathbf{p}}_q - \mathbf{N}\dot{\mathbf{o}}_o - (\mathbf{S}(\mathbf{n}_h) - \mathbf{N}\mathbf{S}(\mathbf{p}_q - \mathbf{o}_o))\boldsymbol{\omega}_o \quad (13)$$

where $\mathbf{N} = \mathbf{R}_o^o \mathbf{N}({}^o\mathbf{p}_q) \mathbf{R}_o^T$. In view of Eq. (4), comparing Eq. (11) with Eq. (13) yields

$$\mathbf{J}_h = -[\mathbf{N} \quad \mathbf{S}(\mathbf{n}_h) - \mathbf{N}\mathbf{S}(\mathbf{p}_q - \mathbf{o}_o)] \mathbf{L}(\mathbf{x}_o). \quad (14)$$

3.3 Joint Positions

The measurement of the joint position vector $\mathbf{q}$ can be used not only to evaluate the configuration of the robot, which can possibly collide with a user, but also to evaluate the position of the point P of the object when in contact with the robot tip point P_q , using the direct kinematics equation. In particular, it is significant to compute the scalar

$$\delta_{hq} = \mathbf{n}_h^T \mathbf{p}_q = g_{hq}(\mathbf{x}_o, \mathbf{p}_q), \quad (15)$$

using also the force measurements via Eq. (9). This quantity represents the component of the position vector of the robot tip point along the constrained direction $\mathbf{n}_h$. The corresponding infinitesimal displacement is the same for the robot and for the object (assuming that contact is preserved). This is not true for the other components of the tip position, which do not provide any useful information on the object motion because the robot tip may slide on the object surface; therefore, these components are not useful to estimate the pose of the object.

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

$$\dot{\delta}_{hq} = \frac{\partial g_{hq}}{\partial \mathbf{x}_o} \dot{\mathbf{x}}_o + \frac{\partial g_{hq}}{\partial \mathbf{p}_q} \dot{\mathbf{p}}_q, \quad (16)$$

but also as $\dot{\delta}_{hq} = \dot{\mathbf{n}}_h^T \mathbf{p}_q + \mathbf{n}_h^T \mathbf{R}_o({}^o\dot{\mathbf{p}}_q + \boldsymbol{\Lambda}({}^o\mathbf{p}_q) {}^o\boldsymbol{\nu}_o)$, where the expression of the absolute velocity of the point P_q in Eq. (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

$$\dot{\delta}_{hq} = \mathbf{p}_q^T \dot{\mathbf{n}}_h + \mathbf{n}_h^T \boldsymbol{\Lambda}(\mathbf{p}_q - \mathbf{o}_o) \boldsymbol{\nu}_o. \quad (17)$$

Hence, by comparing Eq. (16) with Eq. (17) and taking into account Eqs. (12), (14) and (4), the expression below can be found

$$\mathbf{J}_{hq} = \mathbf{p}_q^T \mathbf{J}_h + \mathbf{n}_h^T \boldsymbol{\Lambda}(\mathbf{p}_q - \mathbf{o}_o) \mathbf{L}(\mathbf{x}_o). \quad (18)$$

4 Vision-Based Pose Estimation

4.1 Human Operator's Pose Estimation

In order to use the skeleton algorithm [13] for collision avoidance, simple fixed cameras are employed to detect the positions of face and hands of an operator in the operational space of the robot. In assembly tasks in cooperation with the robot, the operator does not move fast, thus simplifying the tracking by means of cameras. In preliminary experiments, markers are used to help the detection and tracking. The detected positions of the human operator are to be tracked in order to keep a safety volume around him/her, repelling the robot when it approaches too much. Cameras mounted on the robot can be used as well. Potential fields or optimization techniques are then to be designed, in order to create modifications to the robot trajectory aimed at avoiding dangerous approaches. Simple virtual springs or more complex modifications to trajectories, using null-space motion if possible, can be adopted also while using an interaction control with an object, which is considered below. The shape of the computed repelling force or velocity must preserve continuity of reference values for the robot controllers.

4.2 Object Pose Estimation

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

To this end, 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 modeling 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 $\boldsymbol{\zeta}_{hq,k} = \boldsymbol{\delta}_k + \boldsymbol{\mu}_{hq,k}$, and $\boldsymbol{\mu}$ is the measurement noise, which is assumed to be zero-mean Gaussian noise with covariance $\boldsymbol{\Pi}$.

By taking into account Eqs. (5), (10) and (15), the output equation of the EKF can be written as

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

with $[\mu_{f,k}^T \ \mu_{h,k}^T \ \mu_{hq,k}^T]^T$ and $\mathbf{g}(\mathbf{w}_k) = [\mathbf{g}_f^T(\mathbf{w}_k) \ \mathbf{g}_h^T(\mathbf{w}_k) \ \mathbf{g}_{hq}^T(\mathbf{w}_k)]^T$, 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 the null matrix of proper dimensions corresponding to the partial derivative of $\mathbf{g}$ with respect to the velocity variables. In view of Eqs. (8), (14) and (18), the Jacobian matrix $\partial \mathbf{g}(\mathbf{w})/\partial \mathbf{x}_o$ has the expression $\partial \mathbf{g}(\mathbf{w})/\partial \mathbf{x}_o = [\mathbf{J}_f^T \ \mathbf{J}_h^T \ \mathbf{J}_{hq}^T]^T$. The equations of the recursive form of the EKF, being standard, are omitted here for brevity, but can be found, e.g., in [17].

5 Interaction Control

The proposed algorithm can be used to estimate online the pose of an object in the workspace; this allows computing 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 a 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

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

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 $\boldsymbol{\lambda}$ is the lagrangian multiplier associated to the constraint.

The online computation of the constraint equations can be suitably exploited for interaction control. The cases of hybrid force/position control and of impedance control are considered below.

5.1 Hybrid Force/Position Control

Following the formulation introduced in [18] the configuration of the robot subject to the constraint $\varphi(\mathbf{q}) = 0$ can be described in terms of a suitable vector $\mathbf{r}$ of $(n - 1)$ independent variables. From the implicit function theorem, this vector can be defined as $\mathbf{r} = \boldsymbol{\psi}(\mathbf{q})$, where $\boldsymbol{\psi}(\mathbf{q})$ is any $((n - 1) \times 1)$ twice differentiable vector function such that the $(n - 1)$ components of $\boldsymbol{\psi}(\mathbf{q})$ and function $\varphi(\mathbf{q})$ are linearly

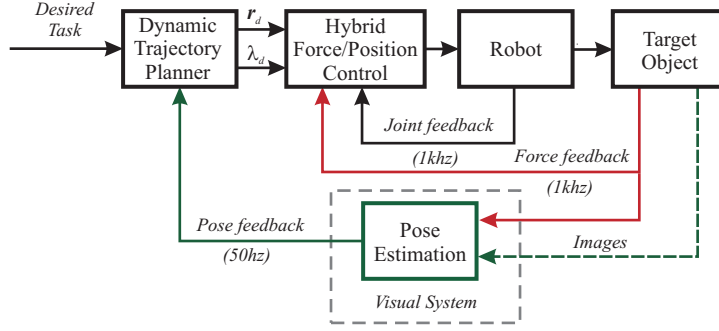


Fig. 2 Hybrid force/position control

independent at least locally in a neighborhood of the operating point. This implies the existence of an $(n \times 1)$ vector function $\rho(\mathbf{r})$ such that $\mathbf{q} = \rho(\mathbf{r})$, at least locally. Moreover, the Jacobian $\mathbf{J}_\rho = \partial \rho / \partial \mathbf{r}$ of the differential mapping $\dot{\mathbf{q}} = \mathbf{J}_\rho(\mathbf{r})\dot{\mathbf{r}}$ is a full-rank matrix. The matrices

$$\mathbf{S}_f = \mathbf{J}^{-T}(\mathbf{q})\mathbf{J}_\phi^T(\mathbf{q}), \quad \mathbf{S}_v = \mathbf{J}(\mathbf{q})\mathbf{J}_\rho(\mathbf{r}),$$

by virtue of the equalities $\mathbf{h} = \mathbf{S}_f\boldsymbol{\lambda}$ and $\mathbf{v}_{P_q} = \mathbf{S}_v\dot{\mathbf{r}}$, play the role of selection matrices.

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

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\boldsymbol{\alpha} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{J}_\phi^T(\mathbf{q})\mathbf{S}_f\mathbf{h}_\lambda,$$

with $\boldsymbol{\alpha} = \mathbf{J}^{-1}(\mathbf{q})(\mathbf{S}_v\boldsymbol{\alpha}_v + \dot{\mathbf{S}}_v\dot{\mathbf{r}} - \dot{\mathbf{J}}\dot{\mathbf{q}})$,

$$\boldsymbol{\alpha}_v = \ddot{\mathbf{r}}_d + \mathbf{K}_{Dr}(\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) + \mathbf{K}_{Pr}(\mathbf{r}_d - \mathbf{r}) \quad (19)$$

and

$$\mathbf{h}_\lambda = \lambda_d + k_{I\lambda} \int_0^t (\lambda_d(\tau) - \lambda(\tau)) d\tau, \quad (20)$$

where $\mathbf{K}_{Dr}$, $\mathbf{K}_{Pr}$ and $k_{I\lambda}$ are suitable feedback gains.

From the block scheme of Figure 2 it can be observed that the algorithm has an 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).

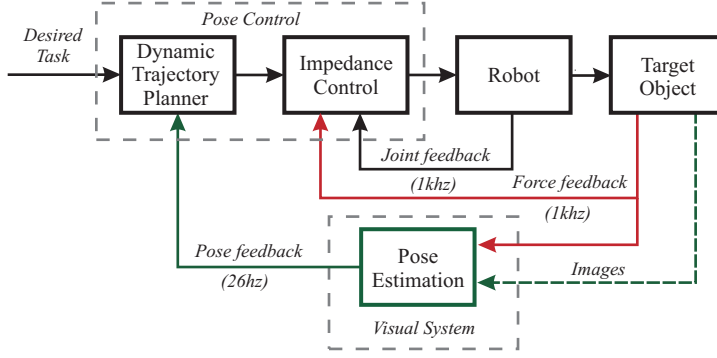


Fig. 3 Position-based visual impedance control

5.2 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, a position-based impedance control algorithm, based on the concept of compliant frame [10], can be adopted (see Figure 3).

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.

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

$$\mathbf{M}_p \Delta \ddot{\mathbf{p}}_{dr} + \mathbf{D}_p \Delta \dot{\mathbf{p}}_{dr} + \mathbf{K}_p \Delta \mathbf{p}_{dr} = \mathbf{h}, \quad (21)$$

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 Eq. (21), in terms of an orientation error based on the (3×1) vector ${}^r \epsilon_{dr}$, 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 $\mathbf{M}_o$, $\mathbf{D}_o$ and $\mathbf{K}_o$ respectively. More details about the geometrically consistent impedance based on the unit quaternion can be found in [10].

It is interesting to notice that the above scheme is able to manage both contact and free-space motion phases. In fact, during free space motion, the position-based



Fig. 4 Experimental setup

impedance control is equivalent to a pose control and the whole scheme corresponds to a classical position-based visual servoing algorithm [19]. For this reason, this scheme can be classified as position-based visual impedance control.

6 Case Studies

The proposed approach has been tested in a number of simulation and experimental case studies. The setup (Figure 4) is composed of a 6-DOF industrial robot Co-mau SMART-3 S with an open control architecture based on RTAI-Linux operating system. A six-axis force/torque sensor ATI FT30-100 with force range of ± 130 N and torque range of ± 10 N·m is mounted at the arm's wrist, providing readings of six components of generalized force at 1 ms. A visual system composed of a PC equipped with a Matrox GENESIS board and a Sony 8500CE B/W camera is available. The camera is fixed and calibrated with respect to the base frame of the robot.

6.1 Interaction with an Object

Experimental tests have been carried out in order to assess the performance of the impedance control scheme. The environment is a planar wooden horizontal surface, with an estimated stiffness (along $^o n$) of about 46000 N/m. The object features are 8 landmark points lying on the plane at the corners of two rectangles of 0.1×0.2 m size (as in Figure 4).

The impedance parameters are chosen as: $M_p = 40I_3$, $D_p = 26.3I_3$ and $K_p = 1020I_3$, $M_o = 15I_3$, $D_o = 17.4I_3$ and $K_o = 3I_3$; a 1 ms sampling time has been

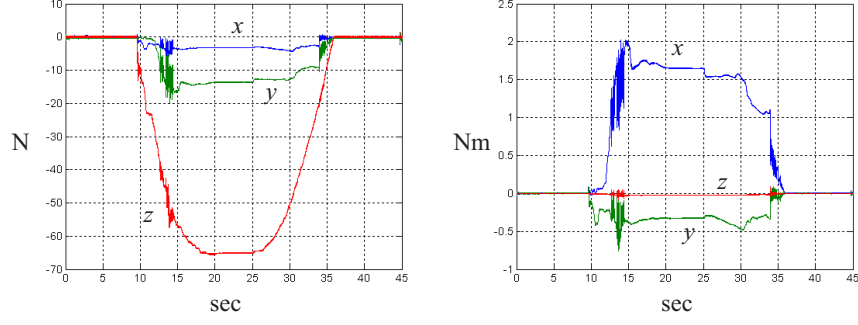


Fig. 5 Measured force (left) and moment (right) in the first experiment

selected for the impedance and pose control. Notice that the stiffness of the object is much higher than the positional stiffness of the impedance, so that the environment can be considered rigid.

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

In the EKF, the non-null elements of the matrix $\mathbf{II}$ have been set equal to 2.5 pixel^2 for $\mathbf{f}$, $5 \cdot 10^{-4}$ for $\mathbf{n}_h$ and 10^{-6} m^2 for δ_{hq} . These values have been set on the basis of the calibration errors of the camera. The state noise covariance matrix has been selected so as to give a rough measure of the errors due to the simplification introduced in the model (constant velocity), by considering only velocity disturbance, i.e.

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

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

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

In the first experiment only the visual measurements are used. The time history of the measured force and moment in the sensor frame are reported in Figure 5. Notice that the force is null during the motion in free space and becomes different from zero after contact. The impedance control keeps the force limited during the transient while, at steady state, the force component along the z axis reaches a value

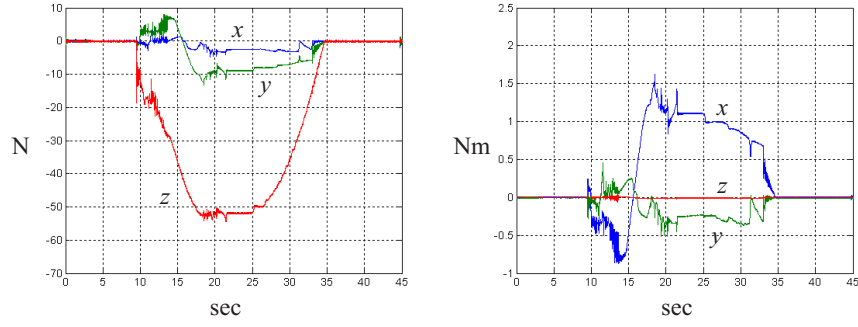


Fig. 6 Measured force (left) and moment (right) in the second experiment

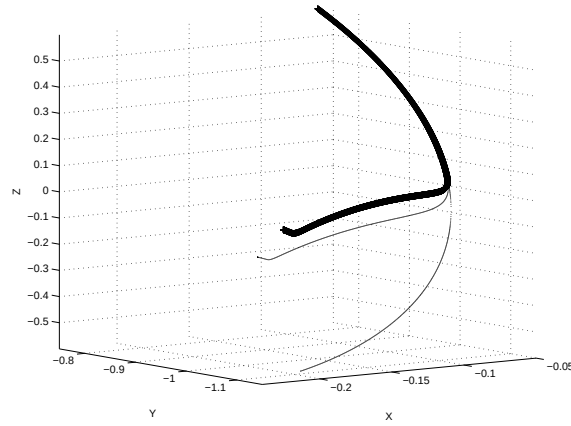


Fig. 7 Trajectory modifications for collision avoidance

of about -65 N, which is different from the desired value of -50 N. This is caused by the presence of estimation errors on the position of the plane due to calibration errors of the camera. Moreover, the moment measured by the sensor is different from zero due to the misalignment between the estimated and the real normal direction of the plane.

The same task is repeated using also the contact force and the joint position measurements for object pose estimation; the results are reported in Figure 6. It can be observed that the benefit of using additional measurements in the EKF results in a force along the vertical direction which is very close to the desired value, due to the improved estimation of the position of the plane; moreover, the contact moment is also reduced because of the better estimation of the normal direction of the plane.

6.2 Vision-Based Head Avoidance

During a task involving interaction with an object, there is the possibility that a human operator is present in the workspace. In such a case, the robot has to reconfigure in order to avoid the body of the operator, tracked by a camera. In a simple case, it is possible to consider the head and the arms of a person present in the workspace as a source of a repelling elastic force. A volume is created around the head and the arms: the robot is pushed with continuous reference values given to force or velocity for a point on each link which is the closest to the “safety volume” considered.

Results of an experiment with the Comau SMART 3S industrial robot are reported in Fig. 7. The planned trajectory (dotted line) is abandoned for the presence of the arm (segment parallel to the axis x , with $y = 1$ and $z = 0.5$). The bold trajectory is the path followed with an elastic constant $K = 0.5$ for planning the desired velocity v of the closest points with the formula $v = K(d_{ref} - d)$ for $d < d_{ref}$, where d_{ref} is the radius of the protective sphere and d is the distance between the robot links and the center of such a sphere. The thin path in Fig. 7 is tracked for $K = 0.5$. This simple case study shows the robustness of the skeleton algorithm, which gives continuous references to different inverse kinematics schemes (one for each robot link) in order to push the robot in a certain direction during any kind of operation and with any kind of motion/force controller.

7 Conclusions

The integration of force and visual control to achieve safe human-robot interaction has been presented. A hybrid force/position control scheme and a position-based visual impedance control scheme were introduced, employing a pose estimation algorithm on the basis of visual, force and joint position data. The addition of collision-avoidance facilities with the “skeleton algorithm” gives the opportunity of sharing the workspace with a human operator.

Acknowledgements The research leading to these results has been supported by the DEXMART Large-scale integrating project, which has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-216239. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

References

1. De Santis A, Siciliano B, De Luca A, Bicchi A (2008) An atlas of physical Human–Robot Interaction. *Mechanism and Machine Theory* 43:253–270

2. Zinn M, Khatib O, Roth O, Salisbury JK (2004) Playing it safe. *IEEE Robotics and Automation Magazine* 11(2):12–21
3. Bicchi A, Tonietti G (2004) Fast and soft-arm tactics. *IEEE Robotics and Automation Magazine* 11(2):22–33
4. Hashimoto H (2005). Intelligent interactive spaces — Integration of IT and robotics. *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts* 85–90
5. Hosoda K, Igarashi K, Asada M (1998) Adaptive hybrid control for visual and force servoing in an unknown environment. *IEEE Robotics and Automation Magazine* 5(4):39–43
6. Nelson BJ, Morrow JD, Khosla PK (1995) Improved force control through visual servoing. *Proceedings of American Control Conference* 380–386
7. Baeten J, De Schutter J (2004) *Integrated Visual Servoing and Force Control. The Task Frame Approach* Springer, Heidelberg
8. Morel G, Malis E, Boudet S (1998) Impedance based combination of visual and force control. *Proceedings of IEEE International Conference on Robotics and Automation* 1743–1748
9. Olsson T, Johansson R, Robertsson A (2004) Flexible force-vision control for surface following using multiple cameras. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and System* 798–803
10. Siciliano B, Villani L (1999) *Robot Force Control* Kluwer, Dordrecht
11. Lippiello V, Siciliano B, Villani L (2007) A position-based visual impedance control for robot manipulators. *Proceedings of IEEE International Conference on Robotics and Automation* 2068–2073
12. Lippiello V, Siciliano B, Villani L (2007) Robot force/position control with force and visual feedback. *Proceedings of European Control Conference* 3790–3795
13. De Santis A, Albu-Schaeffer A, Ott C, Siciliano B, Hirzinger G (2007) The skeleton algorithm for self-collision avoidance of a humanoid manipulator. *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*
14. Hirzinger G, Albu-Schaeffer A, Hahnle M, Schaefer I, Sporer N (2001) On a new generation of torque controlled light-weight robots. *Proceedings of IEEE International Conference on Robotics and Automation* 3356–3363
15. De Santis A, Pierro P, Siciliano B (2006) The virtual end-effectors approach for human-robot interaction. In: Lenarčič J, Roth B (eds) *Advances in Robot Kinematics*. Springer, Heidelberg
16. Espiau B, Chaumette F, Rives P (1996) A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation* 12:313–326
17. Lippiello V, Villani L (2003) Managing redundant visual measurements for accurate pose tracking. *Robotica* 21:511–519
18. Villani L, De Schutter J (2008) Force control. In: Siciliano B, Khatib O (eds) *Springer Handbook of Robotics*. Springer, Heidelberg
19. Lippiello V, Siciliano B, Villani L (2007) Position-based visual servoing in industrial multi-robot cells using a hybrid camera configuration. *IEEE Transactions on Robotics* 23:73–86