A General Visual-Impedance Framework for Effectively Combining Vision and Force Sensing in Feature Space

Alexander Antonio Oliva[1], Paolo Robuffo Giordano[2] and François Chaumette[1]

***Abstract*—Robotic systems are increasingly used to work in dynamic and/or unstructured environments and to operate with a high degree of safety and autonomy. Consequently, they are often equipped with external sensors capable of perceiving the environment (e.g. cameras) and the contacts that may arise (e.g. force/torque sensors). This paper proposes a general framework for combining force and visual information in the visual feature space. By leveraging recent results on the derivation of visual servo dynamics, we generalize the treatment regardless of the visual features chosen. Vision and force sensing are coupled in the feature space, avoiding both the convergence to a local minimum and the arising of inconsistencies at the actuation level. Any task space direction is simultaneously controlled by both vision and force. Compliance against interaction forces is achieved in feature space along the features defining the visual task. Experiments on a real platform are carried out to evaluate the effectiveness of the proposed framework.**

***Index Terms*—Visual Servoing; Force Control; Compliance and Impedance Control; Sensor-based Control.**

I. INTRODUCTION

For many real-world robotic applications, interaction with the environment is a fundamental requirement and the ability of robots in managing this interaction often determines the successful execution of a task. For example, assembly or polishing requires to control the exchanged forces at contact by regulating them to a specific value. The contact *wrench* between the end-effector and the environment is the most complete and effective quantity describing the state of the interaction, which is naturally described in the operational space (often the end-effector frame) [1].

For implementing an interaction control, either an exact knowledge of the location and geometry of the environment is required for accurately planning the task trajectory, or the robot needs to be equipped with force sensing capabilities in order to better adapt to uncertainties and avoid high contact forces along the constrained directions. An effective way to deal with constrained motions is via active compliance, which can be achieved through impedance control [2] by imposing a *mass-spring-damper* behavior of the robot in contact with the environment. This scheme, as well as *compliance* or *stiffness* control, belongs to the category of *indirect* force control methods [3], since they achieve *open-loop* force control via *closed-loop* position control. Many other approaches have been explored in the past decades for including force sensing capabilities inside a motion control scheme, such as the widely studied Hybrid Position/Force [4] that is based on the task description [5], the *inner/outer* [6], or the parallel scheme [7]. The ability of these methods to perform *closed-loop* force control via explicit closure of a force feedback loop places them among the *direct* force control methods [3].

Motivated by the desire of reducing or avoiding the need for accurate environmental modeling, many robotic systems have been equipped with external exteroceptive sensors, like cameras, to perceive the surroundings. This allows such systems to perform the required tasks without having an accurate preliminary knowledge of the environment.

Given the complementarity of vision and force sensing, these two sensing modalities are often used together, especially in the context of physical human-robot interaction [8]. Cameras are capable of providing a rich description of the scene, while force sensing can provide local information about the contact itself. However, due to the very different nature of these two sensing modalities (*e.g.*, difference in the measured physical quantities, data rates, delays, etc.), obtaining an effective combined use of vision and force is not straightforward. To this end, several of the above-mentioned motion/force control schemes have been revisited and expanded mostly by replacing the motion control loop with vision. Those schemes share therefore the same capabilities and drawbacks of their position-based counterparts. For instance, the hybrid vision/force scheme presented in [9] aims at controlling force along constrained directions while vision controls the motion of the remaining ones. The task geometry needs to be known *a priori* in order to properly design the controller via a *selection* matrix for ensuring orthogonality between vision and force controlled directions. This sensory separation does not fully exploit the complementarity of vision and force sensing.

In [10] a position-based impedance controller is used to achieve compliance in Cartesian space while an external vision control loop is closed around the former. The main disadvantage of this control scheme is that it can get stuck in a local minimum where the simultaneous convergence towards the force and visual reference will not be reached [11]. Recently, three image-based visual impedance control laws

Manuscript received: October, 15, 2020; Revised February, 1, 2021; Accepted March, 3, 2021.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments.

[1]A. A. Oliva and F. Chaumette are with Inria, Univ Rennes, CNRS, IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France. alexander.oliva@inria.fr, francois.chaumette@inria.fr

[2]P. Robuffo Giordano is with CNRS, Univ Rennes, Inria, IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France. paolo.robuffo_giordano@irisa.fr

Digital Object Identifier (DOI): see top of this page.

have been proposed [12]. Although the presented first- and second-order controllers rely on the regulation of the visual error, compliance of the end-effector carrying the object is achieved along/about its Cartesian directions.

Constraint-based methods are an alternative way to integrate vision and force sensing. Those frameworks allow over-specifying constraints along a given direction. The conflicts are resolved by adding weights to such constraints imposing a compliant behavior [13], [14]. This weighted addition of contribution may however lead to convergence to a local minimum, as in [10].

To the best of our knowledge, the literature lacks a general framework capable of blending visual servoing (VS) and force regulation simultaneously, regardless of the choice of the visual features and defined directly in the feature space (i.e., the task space for vision systems). Moreover, being the robot motion guided by vision and the tasks defined in the image plane, we aim at achieving compliance along the directions defining the visual task, i.e. along the visual features, rather than in the Cartesian or joint space as done in the literature.

II. RELATED WORKS

Mezouar *et al.* [11] developed the External/Hybrid visual-force control scheme to overcome the drawbacks of the Hybrid and Impedance-based vision-force schemes, for which they have provided an exhaustive comparative analysis. In their approach, the external *wrench* is transformed into a displacement of the image feature reference. This transformation is equivalent to an undamped spring which, as we show in Section V, can start oscillating without ever reaching the convergence of the task or, in more serious cases, damaging the tool. We then seek for a higher order relation linking forces and features motion. One of the first works aiming at figuring out this relationship for vision driven robotic systems is [15], in which the VS dynamics for a ball target is derived and, by exploiting a general definition and pose invariance of the Lagrangian function in the feature space, authors yields to an *ad hoc* simplified model dynamics for the features and the system considered. Vice versa, we are interested in the full Lagrangian model of the manipulator and in a generalized treatment that is independent of the type of visual feature.

Carelli *et al.* [16] proposed a Hybrid force- and vision-based impedance controller to perform a *peg-in-hole* task. Pose-based VS (PBVS) is used to guide the end-effector towards the hole. Interaction forces are fed back to correct small errors of visual guidance by modifying the visual reference along the horizontal plane. Pure force control is used along the vertical axis. The use of selection matrices does not allow to use vision along the vertical axis. The interaction forces are fed back simply by changing the point of application through a coordinate transformation limiting its applicability only to point features.

In this work, we instead leverage results on the derivation of the VS dynamics [17], [18] that do not depend on the particular choice of the visual features: this allows us to derive a general framework that can effectively combine vision and force sensing directly in feature space. This differs from previous works on this topic since the derivation of second-order models has often been formulated *ad hoc* from the definition of the considered features. Furthermore, thanks to this formulation, the projection of the *wrench* applied on the camera into the feature space can also be generalized. This projection allows us to design a controller that makes the closed-loop system to behave as an equivalent mass-spring-damper system fully defined in the feature space and with an isotropic response to external forces (*i.e.*, the interaction does not depend on the manipulator configuration). Moreover, a feature admittance law is coupled with a force control law (FCL) to ensure precise force regulation. Finally, experiments are carried out on a real platform to validate the effectiveness of the proposed approach. A comparison with the external hybrid method in a critical situation is also discussed for further illustrating the benefits of our approach.

The advantages of the proposed framework are several. First of all, it allows to treat the combination of vision and force sensing in a unified way regardless of both the camera configuration (i.e., eye-in-hand or eye-to-hand) and of the chosen visual features. Indeed, any visual feature suitable for either Image-Based Visual-Servoing (IBVS) (e.g. image points, slopes and offsets of lines, image moments, etc) or (PBVS) (i.e. Position and orientation (pose) of objects observed by the camera) can be used. Furthermore, since the vision-force coupling is obtained in the feature space, both convergence towards a local minimum and inconsistencies at actuation level are avoided. Besides that, compliance is achieved in feature space along/about the features defining the visual task resulting in an intuitive choice of the admittance parameters. Unlike in the Hybrid approach, any task direction is fully controlled using both vision and force sensing, and both physical and fictitious forces (i.e. generated in the image plane) can be handled in this framework to account for e.g., the physical interaction with the environment or a visually generated repulsive force field in a collision avoidance tasks. Finally, force regulation can also be achieved.

The rest of this paper is organized as follows. In Sec. III a description of the system *kinematics and dynamics* is given. The *wrench* projection in feature space as well as the derivation of the framework is explained in Sec. IV. In Sec. V experimental results are discussed while in Sec. VI, we present some conclusions and an overview of possible future works.

III. PRELIMINARIES

Without loss of generality we consider the fixed world frame of reference, named Σ_w and with the z-axis pointing upwards, to be coincident with the robot base frame Σ_b. The manipulator is a succession of links and actuated joints whose configuration is described by the joint vector $\boldsymbol{q} \in \mathbb{R}^n$. The end-effector frame is denoted with Σ_e. The robot is equipped with a wrist-mounted force/torque sensor whose reference frame is denoted with Σ_{ft}. Moreover, although we assume here to be using a camera in a *eye-in-hand* configuration (frame Σ_c), the development for the *eye-to-hand* case is formally analogous. Σ_d represents the desired camera frame. Finally, an object is visually tracked and the pose of its reference Σ_o in the

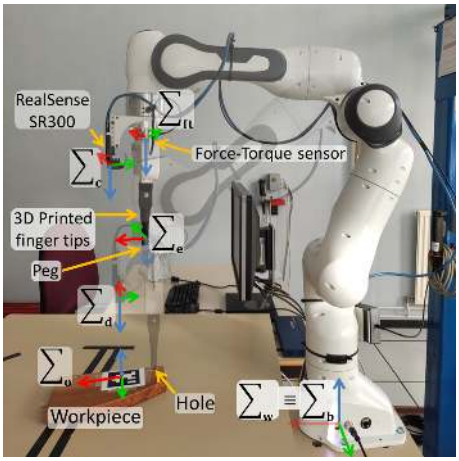


Fig. 1: System setup and reference frames.

camera frame is estimated, or descriptive image features are extracted from the image through computer vision algorithms. Any required geometrical transformation between the above-defined reference frames is supposed to be known, except for those concerning the object frame. An overview of the system setup and main reference frames defining the system is shown in Fig. 1.

A. Manipulator Dynamics

The dynamic model of a robotic arm can be written using the *Lagrange* formulation in the joint space as described in [3]:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau_e} = \boldsymbol{\tau} \tag{1}$$

where $\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^n$ are respectively the generalized joint velocities and accelerations. $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the symmetric and positive definite inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the matrix of centrifugal and Coriolis effects, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the configuration dependent vector of gravitational forces, $\boldsymbol{\tau_e} = {}^{\boldsymbol{e}}\boldsymbol{J_e}^{\top}\,{}^{\boldsymbol{e}}\boldsymbol{h_e} \in \mathbb{R}^n$ is the joint vector corresponding to the external *wrench* ${}^{\boldsymbol{e}}\boldsymbol{h_e} \in \mathbb{R}^6$ acting on the end-effector frame and ${}^{\boldsymbol{e}}\boldsymbol{J_e} \in \mathbb{R}^{6\times n}$ is the robot *Jacobian*, being both expressed in end-effector frame. Finally, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of joint actuation torques.

B. Visual Servoing Kinematics and Dynamics

In the following we recall the derivation of the visual-servo dynamics which relates the motion of visual features with the joint motion at acceleration level. This relationship will then be used, together with the visual-servo kinematics, to express the dynamic model of the manipulator in feature space.

Let us consider a vector $\boldsymbol{s} \in \mathbb{R}^k$ of k image features, e.g., the positions of points in the image plane or parameters of lines. The *kinematic* relationship between the motion of the visual features and the relative *twist* ${}^{\boldsymbol{c}}\boldsymbol{v} \in \mathbb{R}^6$, which is the difference between camera (${}^{\boldsymbol{c}}\boldsymbol{v_c}$) and object (${}^{\boldsymbol{c}}\boldsymbol{v_o}$) twists in the camera frame, is given by the following differential relation:

$$\dot{\boldsymbol{s}} = \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\boldsymbol{v} = \boldsymbol{L_s}({}^{\boldsymbol{c}}\boldsymbol{v_c} - {}^{\boldsymbol{c}}\boldsymbol{v_o}) \tag{2}$$

being $\boldsymbol{L_s} \in \mathbb{R}^{k\times 6}$ the well-known interaction matrix [19] (in this derivation we will assume that it has full rank, thus constraining the 6 degrees of freedom (DoF) of the camera motion). The explicit expression of the interaction matrix for many types of visual features can be found in [19]. When the relative motion between the camera and the object is only due to the robot motion (*i.e.* ${}^{\boldsymbol{c}}\boldsymbol{v_o} = 0$), the previous equation can be rewritten to link the features velocities with the joint velocities through the robot *Jacobian* as:

$$\dot{\boldsymbol{s}} = \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\mathbb{T}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\boldsymbol{J_e}\dot{\boldsymbol{q}} \tag{3}$$

being ${}^{\boldsymbol{c}}\mathbb{T}_{\boldsymbol{e}} \in \mathbb{R}^{6\times 6}$ the Twist-transformation matrix that transforms the *twist* from the end-effector to the camera frame.

By time differentiation of (2) one obtains the expression of the feature accelerations:

$$\ddot{\boldsymbol{s}} = \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\dot{\boldsymbol{v}} + \dot{\boldsymbol{L}}_{\boldsymbol{s}}\,{}^{\boldsymbol{c}}\boldsymbol{v}$$

or when differentiating (3):

$$\ddot{\boldsymbol{s}} = \dot{\boldsymbol{L}}_{\boldsymbol{s}}\,{}^{\boldsymbol{c}}\mathbb{T}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\boldsymbol{J_e}\dot{\boldsymbol{q}} + \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\dot{\mathbb{T}}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\boldsymbol{J_e}\dot{\boldsymbol{q}} + \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\mathbb{T}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\dot{\boldsymbol{J}}_{\boldsymbol{e}}\dot{\boldsymbol{q}} + \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\mathbb{T}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\boldsymbol{J_e}\ddot{\boldsymbol{q}} \tag{4}$$

We can rewrite (4) in a more compact form as:

$$\ddot{\boldsymbol{s}} = \boldsymbol{J_s}\ddot{\boldsymbol{q}} + \boldsymbol{h_q} \tag{5}$$

where $\boldsymbol{J_s} = \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\mathbb{T}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\boldsymbol{J_e}$ denotes the so called *Feature Jacobian* [20], [17] and:

$$\boldsymbol{h_q} = (\dot{\boldsymbol{L}}_{\boldsymbol{s}}\,{}^{\boldsymbol{c}}\mathbb{T}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\boldsymbol{J_e} + \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\dot{\mathbb{T}}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\boldsymbol{J_e} + \boldsymbol{L_s}\,{}^{\boldsymbol{c}}\mathbb{T}_{\boldsymbol{e}}\,{}^{\boldsymbol{e}}\dot{\boldsymbol{J}}_{\boldsymbol{e}})\dot{\boldsymbol{q}}$$

IV. VISUAL IMPEDANCE CONTROL FRAMEWORK

In this section we start by deriving the model of the manipulator in interaction with the environment in the feature space up to the definition of a general framework for combining vision and force sensing in feature space.

A. Impedance Control in Feature Space

Impedance control aims to achieve a dynamic behaviour for the robot end-effector equivalent to a *mass-spring-damper* system subjected to an external force. Our purpose here is to replicate this behaviour between the current and desired visual features lying on the image plane. Let Σ_c be the current and Σ_d the desired camera frames and $\boldsymbol{e_s} = \boldsymbol{s}^d - \boldsymbol{s} \in \mathbb{R}^k$ be the error vector between the desired and current visual features. We formally want to obtain the following behaviour in feature space:

$$\boldsymbol{M_s}\ddot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{D_s}\dot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{K_s}\boldsymbol{e_s} = \boldsymbol{f_s} \tag{6}$$

where $\boldsymbol{M_s}, \boldsymbol{D_s}$ and $\boldsymbol{K_s}$ are positive definite $k \times k$ matrices representing the relative per unit mass/inertia (*p.u.m.i.*) virtual mass, dampers and stiffness of the impedance equation while $\boldsymbol{f_s} \in \mathbb{R}^k$ is the vector of *p.u.m.i.* virtual forces (accelerations) acting on the features.

Equation (1) is a set of n nonlinear and coupled second-order differential equations for which the *inverse dynamics control* is a well-known control strategy for trajectory tracking aiming at linearizing and decoupling the manipulator dynamics via feedback linearization under the assumption of perfect *model* knowledge. We can rearrange (1) as:

$$\ddot{\boldsymbol{q}} = \boldsymbol{B}(\boldsymbol{q})^{-1}(\boldsymbol{\tau} - \boldsymbol{b} - {}^{\boldsymbol{e}}\boldsymbol{J_e}^{\top}\,{}^{\boldsymbol{e}}\boldsymbol{h_e}) \tag{7}$$

having compacted in $\boldsymbol{b} = (\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}))$. After replacing (7) into (5), we find the dynamic equation that governs the features motion:

$$\boldsymbol{J_s B}(\boldsymbol{q})^{-1}\boldsymbol{\tau} = \ddot{\boldsymbol{s}} - \boldsymbol{h_q} + \boldsymbol{J_s B}(\boldsymbol{q})^{-1}\boldsymbol{b} + \boldsymbol{J_s B}(\boldsymbol{q})^{-1}{}^{e}\boldsymbol{J}_{e}^{\top}{}^{e}\boldsymbol{h_e}. \tag{8}$$

Taking into account the expression of the *feature Jacobian* $\boldsymbol{J_s}$, the wrench-transformation matrix that projects the external *wrench* acting on the camera into the end-effector (${}^{e}\boldsymbol{h_e} = {}^{e}\mathbb{F}_{c}{}^{c}\boldsymbol{h_c}$) and recalling that ${}^{c}\mathbb{T}_{e}^{\top} = {}^{e}\mathbb{F}_{c}$ [3], we can rearrange the last member of (8) as:

$$\boldsymbol{L_s}{}^{c}\mathbb{T}_{e}{}^{e}\boldsymbol{J_e B}(\boldsymbol{q})^{-1}{}^{e}\boldsymbol{J}_{e}^{\top}{}^{c}\mathbb{T}_{e}^{\top}{}^{c}\boldsymbol{h_c} = \boldsymbol{L_s B_c}(\boldsymbol{q})^{-1}{}^{c}\boldsymbol{h_c} \tag{9}$$

where $\boldsymbol{B_c}^{-1} = {}^{c}\mathbb{T}_{e}{}^{e}\boldsymbol{J_e B}(\boldsymbol{q})^{-1}{}^{e}\boldsymbol{J}_{e}^{\top}{}^{c}\mathbb{T}_{e}^{\top}$ is the inverse of the manipulator inertia matrix projected in the camera frame. Equation (9) highlights how the *wrench* acting on the camera frame (expressed in its own coordinates ${}^{c}\boldsymbol{h_c}$) projects in feature space as virtual *p.u.m.i.* forces (accelerations) acting on the image features. By renaming some terms in (8) as:

$$\begin{aligned} \boldsymbol{f_s} &= \boldsymbol{J_s B}(\boldsymbol{q})^{-1}\boldsymbol{\tau} \\ \boldsymbol{b_s} &= \boldsymbol{J_s B}(\boldsymbol{q})^{-1}\boldsymbol{b} \\ \boldsymbol{f}_{\boldsymbol{s_{ext}}} &= \boldsymbol{L_s B_c}(\boldsymbol{q})^{-1}{}^{c}\boldsymbol{h_c} \end{aligned} \tag{10}$$

one obtains the manipulator model in feature space:

$$\ddot{\boldsymbol{s}} - \boldsymbol{h_q} + \boldsymbol{b_s} + \boldsymbol{f}_{\boldsymbol{s_{ext}}} = \boldsymbol{f_s}. \tag{11}$$

As we can see from the previous equation, the system is not endowed of inertia (behaves as a mechanical system with unitary mass/inertia). The new control input for the manipulator model in feature space is the vector of *p.u.m.i.* virtual forces $\boldsymbol{u} = \boldsymbol{f_s}$. At this stage we suppose not to have any force measurement. By then choosing our feature space control input $\boldsymbol{u}$ as to compensate for any dynamic term in (11), we have:

$$\boldsymbol{u} = \boldsymbol{w} - \boldsymbol{h_q} + \boldsymbol{b_s} \tag{12}$$

and by injecting it back into (11) yields to:

$$\boldsymbol{w} = \ddot{\boldsymbol{s}} + \boldsymbol{f}_{\boldsymbol{s_{ext}}} \tag{13}$$

in which $\boldsymbol{w} \in \mathbb{R}^k$ represents a resolved acceleration in feature space and constitutes the new control input that has to be opportunely designed. A natural choice for $\boldsymbol{w}$ is a PD controller with acceleration *feed-forward*:

$$\boldsymbol{w} = \ddot{\boldsymbol{s}}^{*} + \boldsymbol{D_s}\dot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{K_s e_s}$$

that replaced in (13) leads to the *closed-loop dynamics* of the system:

$$\ddot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{D_s}\dot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{K_s e_s} = \boldsymbol{f}_{\boldsymbol{s_{ext}}} \tag{14}$$

The behavior of the closed-loop system is as we were looking for i.e., as (6) but with $\boldsymbol{M_s}$ as the identity matrix. The resulting joint space controller is:

$$\boldsymbol{u_\tau} = (\boldsymbol{J_s B}(\boldsymbol{q})^{-1})^{\dagger}(\ddot{\boldsymbol{s}}^{*} + \boldsymbol{D_s}\dot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{K_s e_s} + \boldsymbol{J_s B}(\boldsymbol{q})^{-1}\boldsymbol{b} - \boldsymbol{h_q}) \tag{15}$$

where $(.)^{\dagger}$ is the pseudo-inverse of the matrix in the argument. Even though we have considered the manipulator interacting with the environment in our derivation, if we do not have the force measurement, we arrive at an indirect control of the force through a position controller such as the one derived in [17], [18].

The drawback of this controller is that the closed-loop system exhibits a configuration-dependent compliant behavior due to the presence of the inertia matrix of the manipulator in the external *wrench* projected in the feature space (see (10)). An easy way to overcome this issue and render the manipulator behaviour isotropic, is to measure the external forces being applied on the robot and fully compensate for them. This will make the manipulator infinitely rigid with respect to the measured forces but, thanks both to the fact that we have this measure and that we know how to project the wrenches into the feature space, we can impose the desired system behavior during the interaction. Redefining the feature space controller (12) as:

$$\boldsymbol{u} = \boldsymbol{w} - \boldsymbol{h_q} + \boldsymbol{b_s} + \boldsymbol{L_s B_c}(\boldsymbol{q})^{-1}{}^{c}\boldsymbol{h_c}$$

and plugging it again into (11) we obtain

$$\ddot{\boldsymbol{s}} = \boldsymbol{w}$$

For control purposes it is convenient to impose a constant apparent inertia matrix of the camera in order to have an homogeneous behavior along the Cartesian directions of the camera frame and manage the compliance along the visual feature by opportunely tuning the impedance parameters. This

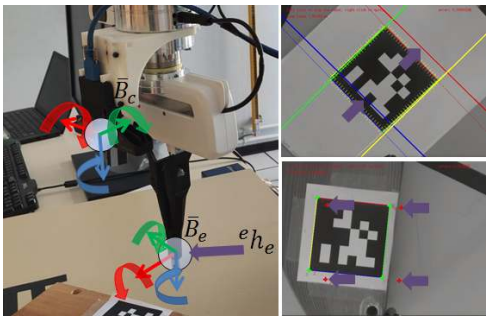


Fig. 2: By assigning a constant apparent inertia to the end-effector or camera frame, we impose the desired interaction behavior along/about the Cartesian directions. We choose an homogeneous inertial behavior so that the end-effector is equivalent to a sphere of mass. Applied forces on the end-effector will be homogeneously distributed among the features.

can be done by choosing as resolved accelerations in feature space the controller:

$$\boldsymbol{w} = \ddot{\boldsymbol{s}}^{*} + \boldsymbol{D_s}\dot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{K_s e_s} - \boldsymbol{L_s}\bar{\boldsymbol{B}}_{\boldsymbol{c}}{}^{-1}{}^{c}\mathbb{F}_{e}{}^{e}\boldsymbol{h_e} \tag{16}$$

being $\bar{\boldsymbol{B}}_{\boldsymbol{c}} = diag(m_{c_d}, m_{c_d}, m_{c_d}, J_{c_d}, J_{c_d}, J_{c_d})$ in which m_{c_d} $[kg]$ and J_{c_d} $[kg.m^2]$ are respectively the desired apparent mass and inertia that the camera should exhibit. Having to deal with contacts with the tool, it can be useful to exhibit isotropy on the end-effector frame by choosing a constant inertia matrix $\bar{\boldsymbol{B}}_{\boldsymbol{e}}$, as done before for $\bar{\boldsymbol{B}}_{\boldsymbol{c}}$, and then project it in the camera frame as $\bar{\boldsymbol{B}}_{\boldsymbol{c}}{}^{-1} = {}^{c}\mathbb{T}_{e}\bar{\boldsymbol{B}}_{\boldsymbol{c}}{}^{-1}{}^{c}\mathbb{T}_{e}^{\top}$. The resulting joint space controller is therefore:

$$\begin{aligned} \boldsymbol{u_\tau} = (\boldsymbol{J_s B}(\boldsymbol{q})^{-1})^{\dagger}(&\ddot{\boldsymbol{s}}^{*} + \boldsymbol{D_s}\dot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{K_s e_s} \\ &+ \boldsymbol{J_s B}(\boldsymbol{q})^{-1}\boldsymbol{b} - \boldsymbol{h_q} + \boldsymbol{f}_{\boldsymbol{s_{ext}}} \\ &- \boldsymbol{L_s}\bar{\boldsymbol{B}}_{\boldsymbol{c}}{}^{-1}{}^{c}\mathbb{F}_{e}{}^{e}\boldsymbol{h_e}) \end{aligned} \tag{17}$$

The closed loop dynamics has the same expression as in (14), but now the projected external forces in feature space do not depend on the manipulator configuration anymore. This translates in a more intuitive tuning of the admittance parameters since this controller would make the manipulator under interaction to behave in the same manner, no matter its configuration while preserving the same stability properties of the previous controller (15).

B. Feature Space Admittance

Despite its theoretical appeal, the feature impedance control cannot be implemented as is with classical cameras due to their intrinsic technological limitations, (*e.g.,* slow frame rate compared to the faster force/torque sensing), which makes it impossible to control a manipulator at torque level. The low control bandwidth resulting from the use of low-rate cameras is in contrast with the need of high gains in the impedance controller for ensuring good tracking performances and disturbance rejection. It follows that there exists a trade-off between the trajectory tracking accuracy and the compliance given to the robot. This limitation motivates the definition of the admittance control in feature space.

In Section IV-A it was shown how the external *wrench* projects into the feature space as virtual *p.u.m.i.* forces that act on the visual features. Exploiting this relationship and the concept of compliant frame (Σ_*), we can link the desired reference (Σ_d) and the compliant frame defined before with a second-order relation that will be used to feed the vision control loop. For the case of PBVS, the admittance is equivalent to the one defined in the task frame, as reported in [3], and depends on the chosen representation of the orientation (e.g., Euler angles, axis/angle, etc.). In our case the task frame is the camera frame rather than the end-effector frame. For IBVS, we extend to the feature space the admittance law linking the desired $\boldsymbol{s}^d(t)$ and compliant feature $\boldsymbol{s}^*(t)$ vectors. We will denote this error with $\bar{\boldsymbol{e}}_{\boldsymbol{s}} = (\boldsymbol{s}^d - \boldsymbol{s}^*)$ and from now on $\boldsymbol{e}_{\boldsymbol{s}} = (\boldsymbol{s}^* - \boldsymbol{s})$ (see Fig. 3):

$$\boldsymbol{M}_{\boldsymbol{s}}(\ddot{\boldsymbol{s}}^d - \ddot{\boldsymbol{s}}^*) + \boldsymbol{D}_{\boldsymbol{s}}(\dot{\boldsymbol{s}}^d - \dot{\boldsymbol{s}}^*) + \boldsymbol{K}_{\boldsymbol{s}}(\boldsymbol{s}^d - \boldsymbol{s}^*) = \bar{\boldsymbol{f}}_{\boldsymbol{s}_{ext}} \quad (18)$$

with $\bar{\boldsymbol{f}}_{\boldsymbol{s}_{ext}}$ the external *wrench* projected in feature space with the constant inertia as in (16). When no external forces act on the end-effector, the compliant and the desired reference frames are coincident ($\Sigma_* \equiv \Sigma_d$). From (18) we then isolate $\ddot{\boldsymbol{s}}^*$ and integrate it twice in order to obtain $\dot{\boldsymbol{s}}^*$ and $\boldsymbol{s}^*$. It is evident that modifying the visual reference accordingly to the sensed force, we are not facing anymore a regulation problem but a trajectory tracking one, for which an appropriate visual control law is necessary in order to better track the moving visual reference [20]. Since we dispose of both the desired features and camera velocities, we can use this information to implement the following visual control law (VCL):

$${}^c\boldsymbol{v}_{\boldsymbol{c}} = \lambda \boldsymbol{L}_{\boldsymbol{s}}{}^{\dagger}\boldsymbol{e}_{\boldsymbol{s}} + \boldsymbol{L}_{\boldsymbol{s}}{}^{\dagger}(\dot{\boldsymbol{e}}_{\boldsymbol{s}} + \boldsymbol{J}_{\boldsymbol{s}}\dot{\boldsymbol{q}}) \quad (19)$$

where λ is a scalar positive gain.

It is interesting to notice that depending on both the type of visual feature chosen and the values assigned to the admittance parameters for each feature, we can have a different compliant behavior and this can be exploited to our advantage.

C. Force regulation

For many robotic applications the regulation of the exchanged forces with the environment is of particular importance for the correct execution of the task itself. It should be stressed that since the impedance/admittance control is an indirect force control scheme, the amount of exerted force is a consequence of the positional error and the chosen control stiffness. In particular, for the visual-feature impedance/admittance control, from (18) at the equilibrium (*i.e.*, $\ddot{\boldsymbol{e}}_{\boldsymbol{s}} = \dot{\boldsymbol{e}}_{\boldsymbol{s}} = 0$) and referring directly to the external *wrench* in the end-effector frame, we have:

$$\bar{\boldsymbol{e}}_{\boldsymbol{s}} = \boldsymbol{K}_{\boldsymbol{s}}{}^{-1}\boldsymbol{L}_{\boldsymbol{s}}\bar{\boldsymbol{B}}_{\boldsymbol{c}}{}^{-1}{}^{c}\mathbb{F}_{e}{}^{e}\boldsymbol{h}_{\boldsymbol{e}}$$

Our intention is to regulate the amount of exchanged force along/about a certain direction to a specific value. To this purpose we use, together with the visual-feature admittance, a PI controller based on the force error in Cartesian space (where the *wrench* is naturally defined) projected into the feature space. We can replace then the second member of (18) with the following (FCL):

$$\begin{aligned} \bar{\boldsymbol{f}}^*_{\boldsymbol{s}_{ext}} &= \boldsymbol{L}_{\boldsymbol{s}}\bar{\boldsymbol{B}}_{\boldsymbol{c}}{}^{-1}{}^{c}\mathbb{F}_{e}{}^{e}\boldsymbol{h}_{\boldsymbol{e}}{}^{*} \quad \text{with} \\ {}^{e}\boldsymbol{h}_{\boldsymbol{e}}{}^{*} &= \boldsymbol{K}_{\boldsymbol{f_P}}({}^{e}\boldsymbol{h}_{\boldsymbol{e}}{}^{d} - {}^{e}\boldsymbol{h}_{\boldsymbol{e}}) + \boldsymbol{K}_{\boldsymbol{f_I}}\int({}^{e}\boldsymbol{h}_{\boldsymbol{e}}{}^{d} - {}^{e}\boldsymbol{h}_{\boldsymbol{e}})\cdot dt \end{aligned} \quad (20)$$

being ${}^{e}\boldsymbol{h}_{\boldsymbol{e}}{}^{d}$ the desired *wrench* in the end-effector frame, $\boldsymbol{K}_{\boldsymbol{f_P}}$ and $\boldsymbol{K}_{\boldsymbol{f_I}}$ two 6×6 positive gain matrices for the proportional and integral force error terms respectively. The presence of the integral term in the controller creates a dominance of the force loop over the internal positional loop.

It is of course clear that such a desired force/torque to be regulated has to be specified accordingly to the task geometry (along/about a constrained direction) otherwise velocities will arise along/about unconstrained direction.

D. The Extended External Hybrid Vision-Force Scheme

Once all the parts of the framework have been explained, it is possible to put them all together. Figure 3 shows the block diagram of the proposed controller: it presents a hierarchical structure in which a FCL, together with a feature space admittance law, is closed around an inner vision control loop implementing the VCL in (19). The reference trajectory

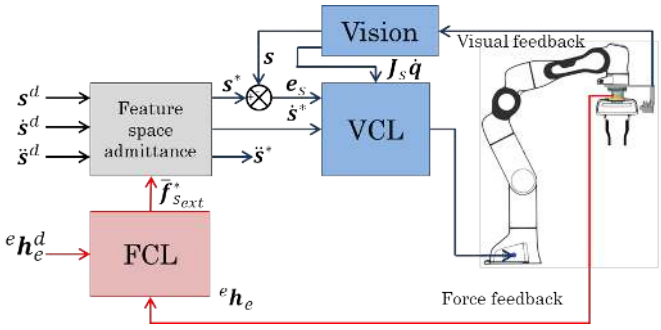


Fig. 3: Extended external hybrid vision/force control scheme. A feature space admittance is used as reference modifier to achieve compliance along visual features directions. Force control law ensures force regulation when the integral term is activated.

$\boldsymbol{s}^d(t)$ feeds the admittance (18), which opportunely modifies it proportionally to the output of the FCL implementing (20). We named the resulting controller as Extended External Hybrid vision-force scheme since it presents the same structure of the External Hybrid. The main advantage over the latter is that our framework offers more flexibility in the choice of the compliant behavior we want to implement, thanks to a greater number of parameters in the admittance that can be tuned, besides to give the possibility to achieve force regulation.

V. EXPERIMENTAL RESULTS

In this section, we show the results of our proposed *Extended External Hybrid* controller executing a *Peg-in-Hole* task. We then compare it to the External Hybrid controller [11] in a critical situation and conclude with a series of peg insertions against pure VS to highlight the greater potential and the range of applicability that comes from a small increase of the controller's complexity. For a better understanding of the work done we suggest to refer to the accompanying video in which, in addition to the experiments in the paper, we show a trial using lines as visual features and another trial that simultaneously copes with physical and fictitious forces to maintain the visual references in the field of view.

A. Experimental Setup

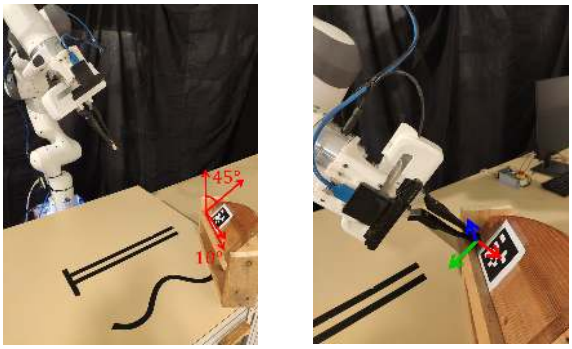
The setup consists of a lightweight Franka Emika Panda arm with 7 revolute joints and its control software running on a Desktop PC with an Arch-Linux distribution patched with a *Preemptive* RT-Kernel 5.4.52-rt31. A wrist mounted six axis force/torque sensor Alberobotics FT45 with force and torque ranges of $F_z \pm 1000N, F_x, F_y \pm 500N$ and $M_{x,y,y} \pm 20Nm$ respectively is present as well as a RealSense RGB-D SR300 camera, mounted in a *eye-in-hand* configuration (see Fig. 1).

Force/torque sensors are capable of measuring any kind of force, being them gravitational, inertial or contact forces. We are interested only in the contact forces. Since force measurements modify the visual reference, an accurate gravitational and inertial effects compensation should be performed. If the payload is not well estimated, spurious force/torque readings remain that cannot be compensated for without resorting to online estimation strategies of the payload parameters. These spurious readings will be interpreted as forces/torques acting on the features reference and will move them according to the admittance stiffness.

To make the system more robust against misestimation of the payload parameters (*i.e.* mass, inertia, and center of mass) or inertial effects, we can choose higher values of the virtual *p.u.m.i.* stiffness of the admittance at the expense of a more rigid system. In our setup, in addition to the standard gripper, we have the camera, the force/torque sensor and some 3D printed parts whose weight has to be estimated. An efficient way to estimate these parameters is reported in [21] that rely on the identified coefficients of the robot dynamics that can be found in [22] for our robot. As we use a wrist-mounted force/torque sensor, we need to compensate only for the payload downstream the sensors flange. In all the reported experiments, 3^{rd}-order Butterworth low-pass filters with a *cut-off* frequency of $2\,Hz$ are used.

B. Peg-in-Hole Experiment

The task consists in inserting a *peg* into a *hole* present in a wood workpiece, both with a diameter of $1cm$. The depth of the *hole* is of $2cm$ while the length of the *peg* is $4cm$. An AprilTag of the 36h11 family of size $6.45cm$ is applied to the surface of the workpiece. The latter is supported by a wooden structure fixed to the work table, which holds an inclined groove of about $45°$ with respect to the horizontal plane on one side and about $10°$ on the other (see Fig. 4(a)).



(a) (b)

Fig. 4: Peg-in-hole experiment setup: (a) The robot at the initial configuration. (b) Forces exerted by the end-effector on the environment during phase 2, the task requires to apply $[5, -5, -15]N$ along the end-effector's x-(blue), y-(green) and z-axis (red arrow) respectively.

The experiment consists of two phases. In the former the robot has to insert the *peg* in the *hole* without computing any trajectory while in the latter it pushes the *peg* towards the bottom of the *hole* and simultaneously applies lateral forces such that the workpiece slides into the groove until a desired force is reached along both directions. The task is executed using our proposed Extended Exteral Hybrid controller (see Fig. 3) for which the VCL gain is set to $\lambda = 1.5\,s^{-1}$. The vision system guides the camera towards a desired final position of the features in the image plane. This final position corresponds to the position in which the peg, held by the robot gripper, is inserted in the hole at a depth of about $1.5\,cm$. Using Visp [23], the AprilTag on the workpiece is tracked and the coordinates of its corners are used as visual features $\boldsymbol{s} \in \mathbb{R}^k$, with $k = 8$. The expression of the interaction matrix $\boldsymbol{L_s} \in \mathbb{R}^{8\times 6}$ associated to these features can be found in [20].

During the first phase of the task, the FCL only projects the external *wrench* into the feature space allowing the system to accommodate for unmodeled interaction forces. So, accordingly with (20), we have: $\bar{\boldsymbol{f}}^*_{\boldsymbol{s_{ext}}} = -\boldsymbol{L_s}\bar{\boldsymbol{B}}_{\boldsymbol{c}}{}^{-1}\,{}^{\boldsymbol{c}}\boldsymbol{h_c}$, meaning that $\boldsymbol{K_{f_P}} = \mathbb{I}_8$, the identity matrix of dimension 8×8, $\boldsymbol{K_{f_I}} = 0\,s^{-1}$ and ${}^{\boldsymbol{e}}\boldsymbol{h_e}{}^d = 0$. The admittance parameters have been chosen in order to have small interaction forces and sufficient damping to attenuate the oscillations that can be triggered while maintaining the overall reactiveness of the system ($\boldsymbol{M_s} = \mathbb{I}_8\,\frac{kg}{kg}$, $\boldsymbol{K_s} = diag(300)\frac{N/m}{kg}$, $\boldsymbol{D_s} = diag(200)\frac{N/m.s^{-1}}{kg}$). During the second phase, that is the force regulation phase, the FCL fully implements eq. (20) with $\boldsymbol{K_{f_P}} = diag(0.2)$, $\boldsymbol{K_{f_I}} = diag(5)\,s^{-1}$ so as to guarantee fast force convergence and limited overshoot while keeping the system stable. The desired wrench is set to ${}^{\boldsymbol{e}}\boldsymbol{h_e}{}^d =$

$[5\,N\ -5\,N\ -15\,N\ 0\ 0\ 0]^\top$ and is transformed to the camera frame as ${}^{c}\boldsymbol{h_c}^{d} = {}^{c}\mathbb{F}_{e}\,{}^{e}\boldsymbol{h_e}^{d}$. For both phases, the desired apparent constant inertia we want the camera to exhibit is $\bar{\boldsymbol{B}}_{\boldsymbol{c}} = {}^{c}\mathbb{T}_{e}\bar{\boldsymbol{B}}_{\boldsymbol{e}}{}^{-1}{}^{c}\mathbb{T}_{e}^{\top}$ with $\bar{\boldsymbol{B}}_{\boldsymbol{e}} = diag(1\,\mathbb{I}_3\,kg, 0.1\,\mathbb{I}_3\,kg.m^2)$.

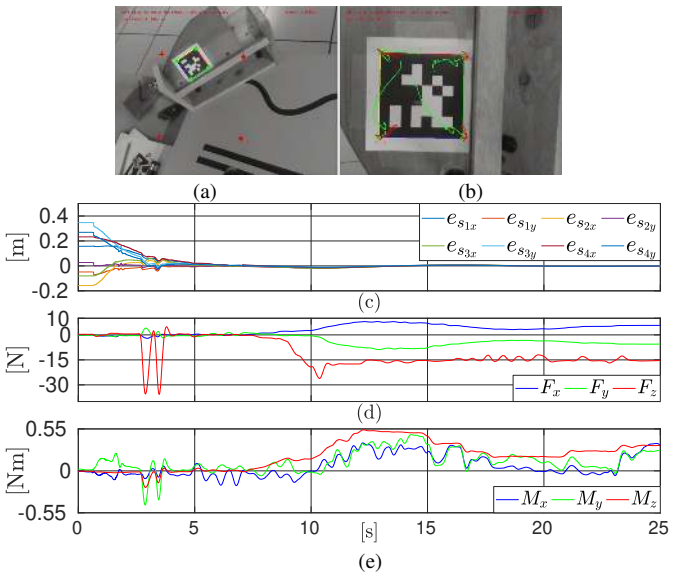


Fig. 5: Peg-in-hole experiment: (a): Initial camera view of the experiment. The green crosses are the current features while the red crosses are the references. (b): Final camera view. It is possible to appreciate (in green) the trajectory of the features in the image plane. (c): Feature errors ($\boldsymbol{s}^* - \boldsymbol{s}$), (d) End-effector forces and (e) moments.

The experimental setup and the framework's behaviour during the task execution are shown in Figs. 4 and 5 respectively. As we can appreciate from the plots in Fig. 5c and the image points trajectory (almost straight) in Fig. 5b, the visual task begins to converge exponentially until it hits the structure, not being aware of it since there is neither a previous knowledge of the environment nor a pre-calculated trajectory. The impact force in the approach direction is approximately $35\,N$ and it projects into the feature space as virtual *p.u.m.i.* forces that pull the four point features reference towards the target centroid causing the robot to slow down along this direction while it continues converging along the others. The robot then approaches the *hole* and the *peg* is successfully inserted. These two collisions are perfectly visible both in Fig 5b (small saw-teeth close to the desired features) and Fig 5d (two peaks along z-axis). Once the *peg* is within the *hole*, we observe that the visual error converges allowing the system to autonomously switch to the second phase in which the integral term in the FCL is activated. The lateral force pushes the workpiece into the groove and against the wall of the structure. The integral term makes the output of the FCL to increase until the force error is nullified. This creates a dominance of the force loop over the internal vision loop.

C. Extended External Hybrid vs External Hybrid

The external hybrid approach [11] has been shown to be successful where impedance- and hybrid-based vision/force schemes have failed. Then follows our interest in comparing this control scheme with the proposed one, as we inherit its structure and extend it, eliminating its drawbacks.

In [11], a Cartesian displacement $d\boldsymbol{X} \in \mathbb{R}^6$ is firstly computed as $d\boldsymbol{X} = \boldsymbol{K}^{-1}({}^{e}\boldsymbol{h_e}^{d} - {}^{e}\boldsymbol{h_e})$ being $\boldsymbol{K} \in \mathbb{R}^{6\times6}$ the contact stiffness. This displacement is then projected in feature space as $d\boldsymbol{s} = \boldsymbol{L_s}\boldsymbol{L_X}^{-1}d\boldsymbol{X}$, with $\boldsymbol{L_X}$ the interaction matrix of the pose parameters. Finally, the compliant reference is obtained by adding the computed feature space displacement to the desired reference $\boldsymbol{s}^* = \boldsymbol{s}^d + d\boldsymbol{s}$.

To evaluate their relative performance over the task, we have executed different trials with both methods, each starting and ending from a different position. This time the workpiece is clamped on the work surface, preventing it from moving. The results of one of these experiments are in Fig. 6.

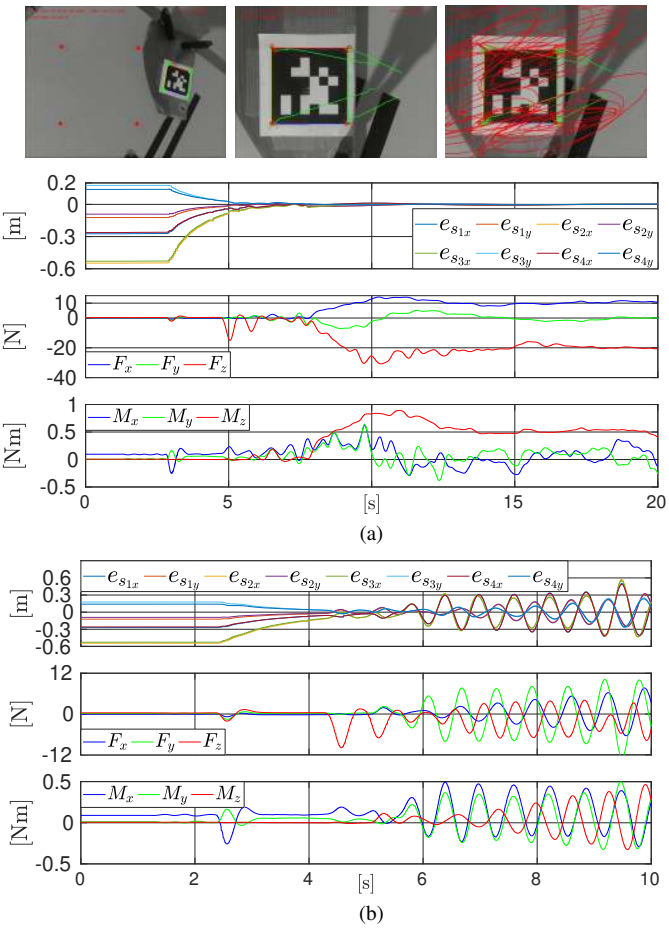


Fig. 6: Peg-in-hole comparison: (a) the results obtained with our method and, (b) those obtained with the external hybrid [11]. The images show the initial and final camera views with the current and reference feature trajectories in green and red respectively. The plots report, from top to bottom: the visual feature errors ($\boldsymbol{s}^* - \boldsymbol{s}$), measured end-effector forces and moments. ${}^{e}\boldsymbol{h_e}^{d} = [5N\,,0\,,-20N\,,0\,,0\,,0]^\top$

To make the comparison as fair as possible, we choose the same gain $\lambda = 1.5\,s^{-1}$ for both controllers and the proportional gain of the external hybrid, *i.e.*, the contact stiffness, in such a way that both methods achieve the same displacement of the visual reference when the same force is applied leading to choose $\boldsymbol{K} = (\boldsymbol{L_X}\boldsymbol{L_s}^{\dagger}\boldsymbol{K_s}{}^{-1}\boldsymbol{L_s}\bar{\boldsymbol{B}}_{\boldsymbol{c}}{}^{-1}{}^{c}\mathbb{F}_{e})^{-1}$. On the other hand, $\bar{\boldsymbol{B}}_{\boldsymbol{c}}$, $\boldsymbol{K_{f_P}}$, $\boldsymbol{K_{f_I}}$ and $\boldsymbol{M_s}$ are as in the previous experiment, for both phases, while for this experiment we impose

a less stiff behaviour by choosing $\boldsymbol{K_s} = diag(200)\frac{N/m}{kg}$ and $\boldsymbol{D_s} = diag(150)\frac{N/m.s^{-1}}{kg}$ for the first phase.

At the beginning, both methods starts converging towards the target with exactly the same behaviour. In fact, as long as there are no interaction forces, the reference is not modified and the task is a pure VS. When the *peg* approaches the *hole*, it hits the border before entering. As shown in Fig. 6, for both the external hybrid and our method, the impact force along the approach direction stays quite limited even though for our method it is slightly higher. This is due to the presence of the damping term. Both methods succeed in the insertion of the *peg* tip but the increase of lateral forces once the *peg* is inside the workpiece, triggers the oscillation of the non-damped spring of the stiffness control for the external hybrid method. If these oscillations remain limited, they do not allow to reach the convergence of the visual reference while, in the event they explode as it is the case, it can lead to damage the manipulated object or even the robot tool. This experimental result has shown that endowing the feature's motion with a dynamic capable of damping their velocity, improves the performance of the system, managing to dampen the oscillations that can be triggered.

D. Further Experiments

We have conducted a series of insertion tasks to evaluate the success rate of our proposed extended external hybrid method. Starting from 10 different initial conditions covering almost all the dexterous work-space of the robot, we tested the efficiency of our method in inserting the *peg* in the *hole* against the task being executed by pure VS.

As expected, being able to account for interaction forces, our method succeeded $9/10$ times, while the pure VS only $5/10$. It should be noticed that the last experiment was purposely started from a configuration for which the approach direction will converge before the other ones. This is to highlight that no trajectory planning or insertion strategy was used. The controller simply nullifies the visual error. The method succeeds in every case the pure VS has failed.

VI. CONCLUSIONS

We presented a general framework for both IBVS and PBVS regardless of the visual feature chosen or camera mount. It allows for potentially 6 DoF force regulation. External wrenches are projected in feature space in a general fashion and compliance is achieved along the directions defined by the visual task. This can open a new research line that uses machine learning to learn how to dynamically tune the admittance parameters based on the provided images.

We are interested in performing high-speed interaction tasks and this means taking robot dynamics into account by implementing eq. (17), that controls the robot at torque level. To this end we will focus on increasing the camera rate by predicting the state of the features e.g., with a Kalman filter designed in feature space. We are also interested into studying the resulting system energy in feature space to design passivity-based controllers capable of passivize the destabilizing effects of low camera rates and time delays.

REFERENCES

[1] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[2] N. Hogan, "Impedance control: An approach to manipulation: Part I-Theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, 03 1985.

[3] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modeling, Planning and Control*, 3rd ed. London: Springer, 2008.

[4] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 06 1981.

[5] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[6] J. De Schutter and H. Van Brussel, "Compliant robot motion ii. a control approach based on external control loops," *The International Journal of Robotics Research*, vol. 7, no. 4, pp. 18–33, 1988.

[7] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 361–373, 1993.

[8] A. Cherubini and D. Navarro-Alarcon, "Sensor-based control for collaborative robots: Fundamentals, challenges, and opportunities." *Front. Neurorobot.*, 2021.

[9] J. Baeten and J. De Schutter, *Integrated Visual Servoing and Force Control - The Task Frame Approach*. Springer, 01 2003.

[10] G. Morel, E. Malis, and S. Boudet, "Impedance based combination of visual and force control," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 2, 1998, pp. 1743–1748 vol.2.

[11] Y. Mezouar, M. Prats, and P. Martinet, "External hybrid vision/force control," *Intl. Conference on Advanced Robotics*, 2007.

[12] V. Lippiello, G. A. Fontanelli, and F. Ruggiero, "Image-based visual-impedance control of a dual-arm aerial manipulator," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1856–1863, 2018.

[13] J. D. Schutter, T. D. Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *Int. J. Robotics Research*, pp. 433–455, 2007.

[14] C. Cai and N. Somani, "Visual servoing in a prioritized constraint-based torque control framework," in *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2018, pp. 309–314.

[15] Hong Zhang and J. P. Ostrowski, "Visual servoing with dynamics: control of an unmanned blimp," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, 1999, pp. 618–623 vol.1.

[16] R. Carelli, E. Oliva, C. Soria, and O. Nasisi, "Combined force and visual control of an industrial robot," *Robotica*, vol. 22, no. 2, p. 163–171, 2004.

[17] F. Fusco, O. Kermorgant, and P. Martinet, "A comparison of visual servoing from features velocity and acceleration interaction models," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4447–4452.

[18] S. Vandernotte, A. Chriette, P. Martinet, and A. S. Roos, "Dynamic sensor-based control," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016, pp. 1–6.

[19] 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, 1992.

[20] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," *Handbook of Robotics, 2nd edition*, pp. 841–866, 2016.

[21] C. Gaz and A. De Luca, "Payload estimation based on identified coefficients of robot dynamics — with an application to collision detection," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3033–3040.

[22] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, and A. De Luca, "Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Lett.*, 2019.

[23] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, December 2005.