Hybrid force-vision control for da Vinci simulator

Cappella Luigi Cirillo Michele Rinaldi Riccardo

Abstract

In robotic manipulation tasks where it is asked to approach a surface (e.g., surgical operations) it is usually necessary to guarantee short approaching times, stability, low impact forces, avoid any bounce between the end-effector and the surface. A simple force damping control is trivial to implement, but it can show instability problems and/or very low approaching velocities, depending on the values of the gains. To overcome these limitations a guarded move strategy could be considered, but this fails when significative inertial effects occur. A resolutive solution paradigm seems to be the one based on a combination of vision-based and force-based control techniques, that is becoming one of the favorite approaches for these tasks. The contribution of our work is twofold: first, we analyse and resume part of the literature about the aforementioned vision/force-based solution paradigm, by presenting the control schemes described into [1][2][3]; second, we implement the control scheme proposed in [1] using MATLAB and test it with the Research Kit V-REP simulator from [4].

1 Hybrid vision/force-based control schemes, a survey.

As said, we analysed the content of works [1],[2] and [3]. Document [1] provides two main contributions: first, it provides a taxonomy of the current mixed vision/force based control schemes, underlining the pro and cons of each; second, it proposes a new scheme that basically arranges the two pure feedback control law in a nested hierarchy. Paper [2] introduces a way to switch conveniently between vision-based and force-based control modes, basing on which of the two is the most suitable in any specific moment. Document [3] simply shows a non-conventional vision/force control strategy using an ultrasound probe in place of the camera in the role of vision sensor.

1.1 A taxonomy of joined visual/force control schemes.

According to [1], there are essentially three effective way to combine visual-based and force-based control techniques, that are depicted in Fig. [1]:

- ***The hybrid-based control scheme.*** This scheme selects for any axis direction the best control law between a vision-based one and a force-based one. Therefore, for this case the basic idea is not to merge the two control laws, but instead to select in a mutual exclusive manner the best one at any instant.

- ***The impedance-based control scheme.*** This scheme uses essentially a pure visual-based control law corrected with an output term originating from the force error information. The name of this scheme arises from the fact that this additional force-based term is obtained converting the force error information in a velocity quantity using the well-known mechanical impedance law $F = ZV$.

- ***The external hybrid vision/force control scheme.*** This scheme represents the main contribution of [1], and it is the one that we implemented. Therefore, the detailed description of the scheme is postponed in the next ad hoc section. However, very concisely, this scheme uses a pure visual-based control law corrected with an input term originating from the force error information.

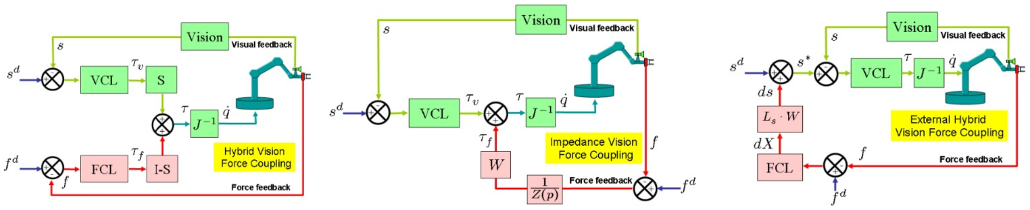


Figure 1. The three main vision/force combination schemes: hybrid-based control scheme (left), impedance-based control (center) and external hybrid vision/force control (right).

We remark that all these combination schemes provide a generalized Cartesian velocity vector τ for the end-effector, that has eventually to be converted in the actual joint velocity $\dot{q}$ by means of standard, often built–in procedure. This design choice is very common, in particular for the context of the visual servoing, where it is named *dynamic look-and-move* system [5].

Authors in [1] even made a comparison between the three schemes presented above. We resume here the most relevant conclusions. The hybrid-based control scheme suffers from a very high task dependency that makes this control adequate just for a limited range of scenarios. This because, for example, the task model parameters must be exactly known. The impedance-based control fails when the additional output term cancels the one provided by the pure visual controller. In this situation it can still happen that convergence is not achieved or that it is achieved with an unsatisfactory oscillatory behaviour. The external hybrid vision/force control scheme overcomes all the presented limitations, and results in a very successful solution.

1.2 An hybrid-based control law using resolvability.

In the paper [2], force and vision sensing modalities are combined using the hybrid-based control scheme depicted in Fig. 1, leftmost panel. We recall that in this scheme a pure vision-based and a pure force-based control laws are selected in a mutual exclusive manner. In this particular case the authors made the following choices:

- The pure vision-based law is the following one:

$$\tau_v = -(L^\top QL + H)^{-1} L^\top Q(s - s^d) \tag{1}$$

  where s is the measured image feature vector representing the objects being servoed, s^d is the desired feature vector, L is the interaction matrix and Q, H are weighting parameters.

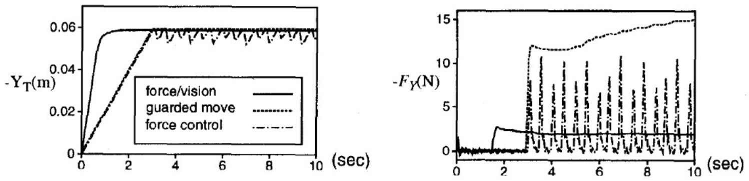
- The pure force-based law is the following one:

$$\tau_f = K\,(f^d - f) \tag{2}$$

  where f is the measured generalized force from the sensor, f^d is the desired generalized force, and K is a matrix control gain.

- The criterion for selecting the best control law is based on a new quantity inspired to manipulability that is called *resolvability*. In a nutshell, this quantity measures how much a displacement in task space can be appreciated in sensor space along the three axes x, y and z. Therefore, we actually have two resolvabilities, one for the vision sensor and the other for the force sensor. For both sensors, the higher is the resolvability the better is the related control law. At a certain instant, we select for any axis direction the control law having the greatest resolvability along that direction.

Authors in [2] even provide some comparative experimental results between the proposed strategy and two other classical techniques, that are the damping force control law and the guarded move solution. The results, that are depicted in Fig 2, refer to the case of reaching a surface and maintaining a 2N force when in contact. Three key points can be noticed: i) the proposed strategy approaches the surface faster, ii) the damping force control law results in highly unstable contact response, and iii) the guarded move solution shows too high impact forces.



(a) Y-axes measurements. (b) Force measurements.

Figure 2. Depiction of the experimental comparison made in [2].

1.3 An hybrid-based control law using ultrasound probes.

Like in [2], even authors in document [3] use the hybrid-based control scheme seen in the first section and depicted in Fig. 1, leftmost panel, but this time they consider a less common Ultrasound (US) probe in the role of visual feedback, and made different design choices:

- The pure vision-based law is the following one:

$$\tau_v = -k\hat{L}^\dagger(s^d - s) \tag{3}$$

  where k is a positive gain, $\hat{L}^\dagger$ the pseudoinverse of an estimated interaction matrix and $(s^d - s)$ is the feature error.

- The pure force-based law is the following one:

$$\tau_f = -K\frac{(f^d - f)}{\hat{S}} \tag{4}$$

  where K is the control gain, $(f^d - f)$ is the contact force error expressed w.r.t. the probe frame, $\hat{S}$ is an estimate of the contact stiffness.

- The criterion for selecting the best control law is very raw. In fact, the force control is statically preferred along the y-axis of the probe frame, while the remaining directions are controlled by the visual control.

The authors considered two different kinds of US probes, that are shown in Fig 3. The former, depicted on the left, is a 2D probe, that is able to detect just a planar section of the scene, whereas the latter, depicted on the right, is a 3D probe, and detects an entire volume of interest.

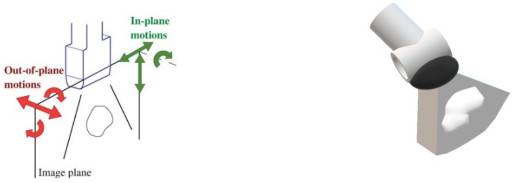


Figure 3. The two types of US probes considered in [3].

Depending on the probe that is used, the estimation of the interaction matrix, $\hat{L}$, has different expressions. In particular:

→ For the 2D probe it is:

$$J = [\ \nabla I_x \ \ \nabla I_y \ \ \nabla I_z \ \ y\nabla I_z - x\nabla I_z \ \ x\nabla I_y - y\nabla I_x\ ] \tag{5}$$

where ∇I_x, ∇I_y and ∇I_z are the US intensity gradient components.

→ For the 3D probe it is:

$$J = [\ \nabla I_x \ \ \nabla I_y \ \ \nabla I_z \ \ y\nabla I_z - z\nabla I_y \ \ -x\nabla I_z + z\nabla I_x \ \ x\nabla I_y - y\nabla I_x\ ] \tag{6}$$

From these expressions we note that to compute the estimation L we need to estimate the intensity gradient, namely ∇I_x, ∇I_y and ∇I_z. For this aim one can use the standard methods present in the literature, like the least sum of squares method.

Let now present the experimental results of the document. In a nutshell, 2D probes represent the preferred choice for tracking tasks thanks to their high frame rate, whereas 3D probes are a better solution for positioning tasks thanks to the larger convergence domain. Let first consider the tracking task. In Fig. 4 are shown some results achieved with a 2D probe and by using a least sum of squares method to estimate the gradient. In particular: in image (a) the region of interest is highlighted in a cyan box; in plot (b) the visual error is reported; images (c) – (d) and images (e) – (f) show the US image (left) and the US image error (right) in the situation when the tracking error reaches its maximum and minimum, respectively.

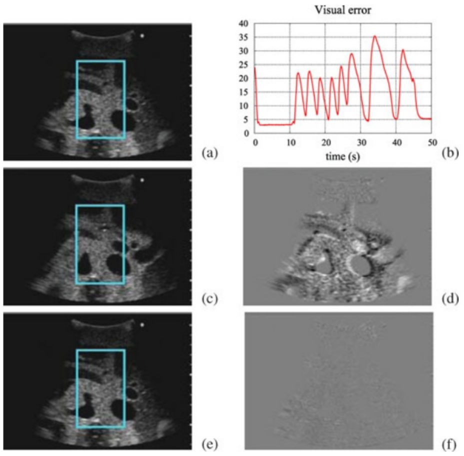


Figure 4. Depiction of the experimental results for the tracking task with a 2D probe made in [3].

Let now consider the positioning task. In Fig. 5 we show some results achieved with a 3D probe and by using a 3D weighted filters method to estimate the gradient. In particular: i) the two images positioned above represents the initial US image (left) and the initial US image error (right); ii) the same is done with the centered images for the final reached state; iii) the two images positioned below represent the features error (left) and the Cartesian error (right) convergence during the simulation time.

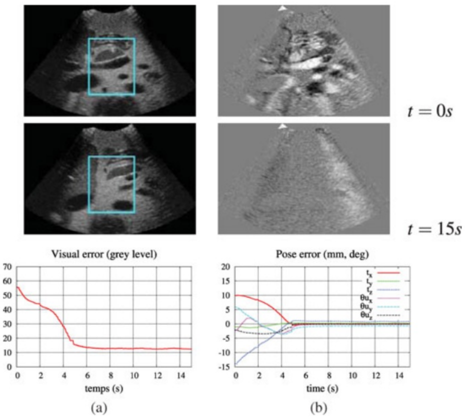


Figure 5. Depiction of the experimental results for the positioning task with a 3D probe made in [3].

2 The implemented control scheme.

The subject of this section is the main contribution of [1], namely the external hybrid vision/force control scheme, that is represented in Fig. 6. It is the control scheme that we implemented using MATLAB and V-REP, and now we will describe it in detail.

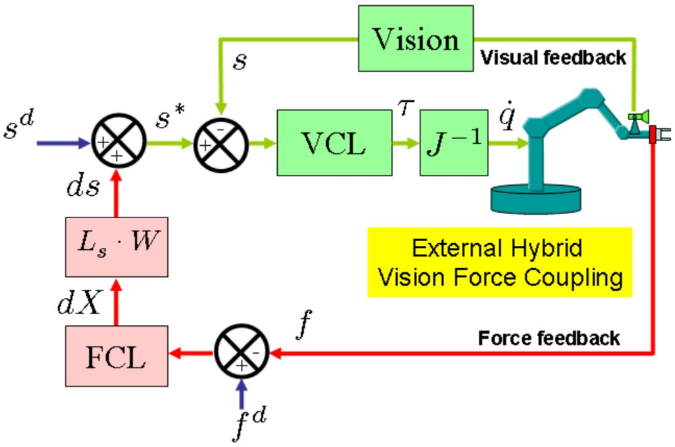


Figure 6. Block-wise representation of the the external hybrid vision/force control scheme.

Firstly, we note that this scheme:

→ is an instance of the ***eye-in-hand***, ***image-based***, ***dynamic look-and-move*** design priciple [5],

→ takes in input f^d, f, s^d and s, namely the desired and the actual contact force and image features, respectively,

→ provides in output the generalized end-effector velocity vector according to the following control law:

$$\tau = \hat{L}^\dagger K(s^* - s), \quad \text{with} \quad s^* = s^d - \hat{L}\,WC\,(f^d - f) \tag{7}$$

where $\hat{L}$ is an estimation of the interaction matrix and K, W are standard error-proportional gain matrices and C is a compliance matrix.

2.1 An interpretation.

For the expression 7 we see a clear interpretation, that we now try to explain. Let recall that the generalized end-effector velocity τ w.r.t. the camera frame and the image features velocity $\dot{s}$ w.r.t. the image frame are related linearly by the interaction L, namely

$$\dot{s} = L\tau \tag{8}$$

Using the standard control strategy $\dot{s} = K(s^* - s)$, where s^* is the desired trajectory, we obtain the system

$$K(s^* - s) = L\tau \tag{9}$$

that admits as solution

$$\tau = L^\dagger K(s^* - s) \tag{10}$$

However, because L is in general not perfectly know, we need to use in place of it an estimation $\hat{L}$, leading to the first part of 7. Now we turn to interpret the last expression, namely $s^* = s^d + \hat{L}\, C\, (f^d - f)$. This expression computes the desired image feature vector, and we note that this quantity is such that ***(a)*** when the contact force corresponds to the one desired, namely $f = f^d$, it assumes a "default" value s^d, otherwise ***(b)*** this default value is modified according to the following. ***(b.1)*** In order to reduce the force error, the end-effector should move along the direction of $-(f^d - f)$ by a distance proportional to $\|f^d - f\|$, given by the compliance matrix C and scaled suitably by the gain matrix W; briefly, in order to reduce the force error, the end-effector should be moved from the current pose with the following displacement:

$$\Delta\rho \triangleq -WC\,(f^d - f) \tag{11}$$

(b.2) Let ρ^d be the pose from which the image features s are seen equal to the desired ones s^d, in order to induce such a motion, the desired image features can be changed to become the ones that are seen from $\rho^d - \Delta\rho$. To the aim we need to modify s^d in $s^d + \Delta s$, and from 8 we have

$$\begin{aligned} \Delta s &= L\Delta\rho \\ &= -LWC\,(f^d - f) \end{aligned} \tag{12}$$

2.2 Further technical details.

The expression for the matrix L depends heavily on the nature of the image features. First, let assume that the camera implements the standard Pinhole model, namely if the camera employs the following mapping:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \triangleq \begin{pmatrix} f\,\frac{X}{Z} \\ f\,\frac{Y}{Z} \\ 1 \end{pmatrix} \tag{13}$$

where f is the focal length, namely the depth of the image plane w.r.t. the camera frame. Then, let consider the case when the image features are a set of n landmark points, namely the case when the vector s is a $2n$-dimentional vector $\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{pmatrix}$, where u_i and v_i are the coordinates of the generic point in the image plane. In this case the interaction matrix is equal to:

$$L = \begin{pmatrix} \frac{f}{Z_1} & 0 & \frac{u_1}{Z_1} & \frac{u_1 v_1}{Z_1} & -\frac{f^2 + u_1^2}{f} & v_1 \\ 0 & \frac{f}{Z_1} & \frac{v_1}{Z_1} & \frac{f^2 + v_1^2}{f} & -\frac{u_1 v_1}{Z_1} & -u_1 \\ \frac{f}{Z_2} & 0 & \frac{u_2}{Z_2} & \frac{u_2 v_2}{Z_2} & -\frac{f^2 + u_2^2}{f} & v_2 \\ 0 & \frac{f}{Z_2} & \frac{v_2}{Z_2} & \frac{f^2 + v_2^2}{f} & -\frac{u_2 v_2}{Z_2} & -u_2 \\ & & & \vdots & & \\ \frac{f}{Z_n} & 0 & \frac{u_n}{Z_n} & \frac{u_n v_n}{Z_n} & -\frac{f^2 + u_n^2}{f} & v_n \\ 0 & \frac{f}{Z_n} & \frac{v_{1n}}{Z_n} & \frac{f^2 + v_n^2}{f} & -\frac{u_n v_n}{Z_n} & -u_n \end{pmatrix} \tag{14}$$

where $Z_1, \ldots, Z_n$ are the depths of the n landmark points w.r.t. the camera frame. Obviously these depths cannot be measured by the camera sensor, and therefore if there is not any other mechanism that measures them (e.g. a two-camera stereo system or a range sensor) then the expression of L is not known, as we said above. In this case, L needs to be estimated. More technical details can be found in [5].

3 Experimental setting.

As said, we implemented the control scheme described in the previous section by interfacing MATLAB and V-REP by means the standard API. In particular, a MATLAB script implements the control unit, and a V-REP scene provided by [4] implements the da Vinci robot.

3.1 The da Vinci simulator.

Even if complete details can be found in [4], we now briefly describe the considered V-REP model of the da Vinci, shown in Fig 7. The robot consists of three actuated robot arms, attached to a non-actuated common base. The central arm is named Endoscopic Camera Manipulator (ECM), whereas the lateral arms are identical and are named Patient Side Manipulators (PSMs).

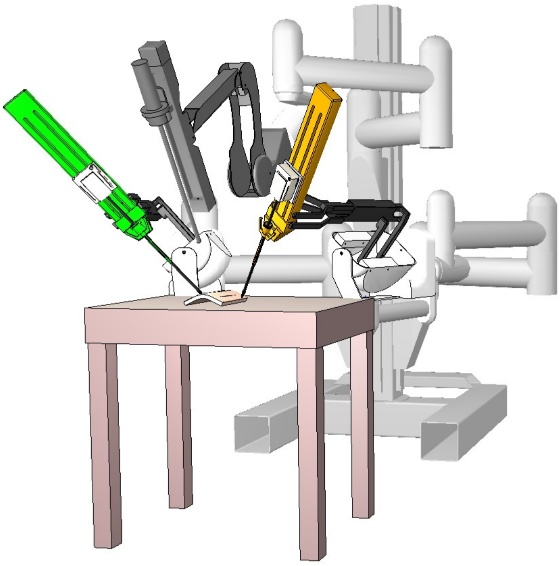


Figure 7. The da Vinci Research Kit V-REP simulator.

In Fig. 8 it is shown the kinematic structutured for both the PSMs and the ECM. We note that the PSMs consist of an RRP kinematic chain followed by a spherical wrist. Conversely, the ECM consists of an RRP kinematic chain followed by a single R joint that permits to orient the image plane.

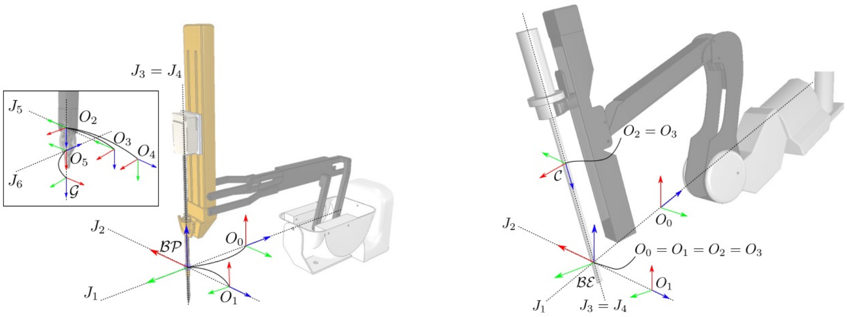


Figure 8. PSM (left) and ECM (right) kinematic description.

For our experiments we not consider the ECM, since for control purposes it is useful for an eye-to-end visual servoing context, that is not the one considered by our reference paper [1], that in fact consider an eye-in-hand scenario. Furthermore, we test our control law just on the left PSM because, due to the perfect symmetry between the PSMs, the same identical results would have been obtained using the other.

3.2 Experiment results.

We prepared two experiments. The first experiment aims to show the ability of our control law to put the end-effector of the PSM at any desired pose in the workspace with a short approaching time; in particular, it shows that the considered control law could be employed in a suturing operation. The second experiment aims to show how the robot reacts to an undesired and sudden increase of the contact force; it shows that the considered control law will not cause any hurt to the patient under these circumstances.

The first experiment: the suturing.

During the experiment, the control law will:

1. put the PSM end-effector pose "at home", as depicted in Fig 9, left side;
2. put the PSM end-effector pose at the center of the first landmark, namely the rightmost set of grey spheres in Fig. 9, right side;
3. put again the PSM end-effector pose "at home";
4. put the PSM end-effector pose at the center of the second landmark;
5. put again the PSM end-effector pose "at home";
6. . . .
7. put the PSM end-effector pose at the center of the last landmark, namely the leftmost set of grey spheres in Fig. 9, right side.

In order to let the reader to visualize the experiment, some screenshots of are collected into Fig. 10.

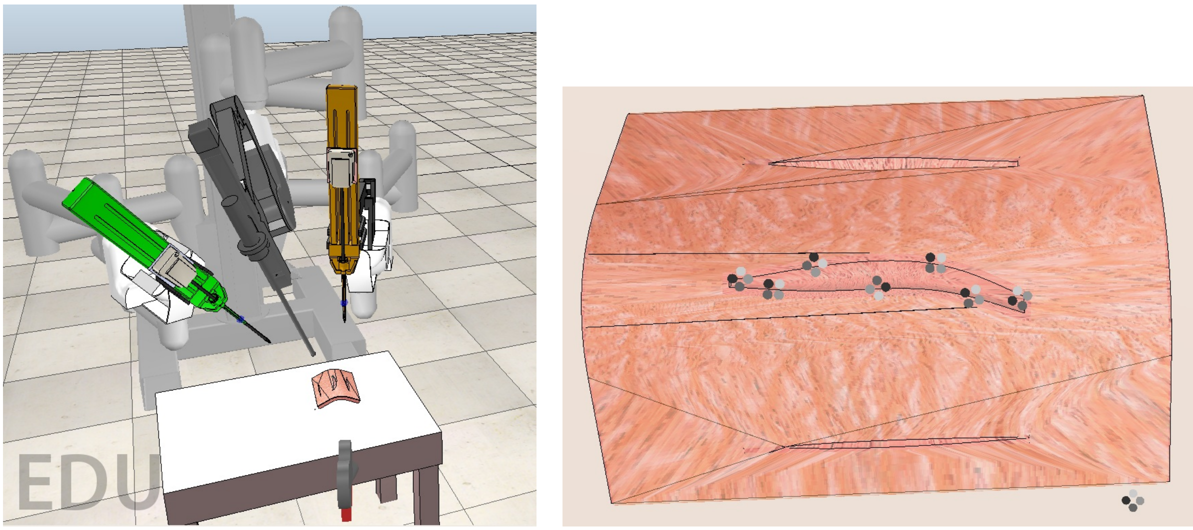


Figure 9. First experiment, the pose "at home" (left) and the landmarks (right).

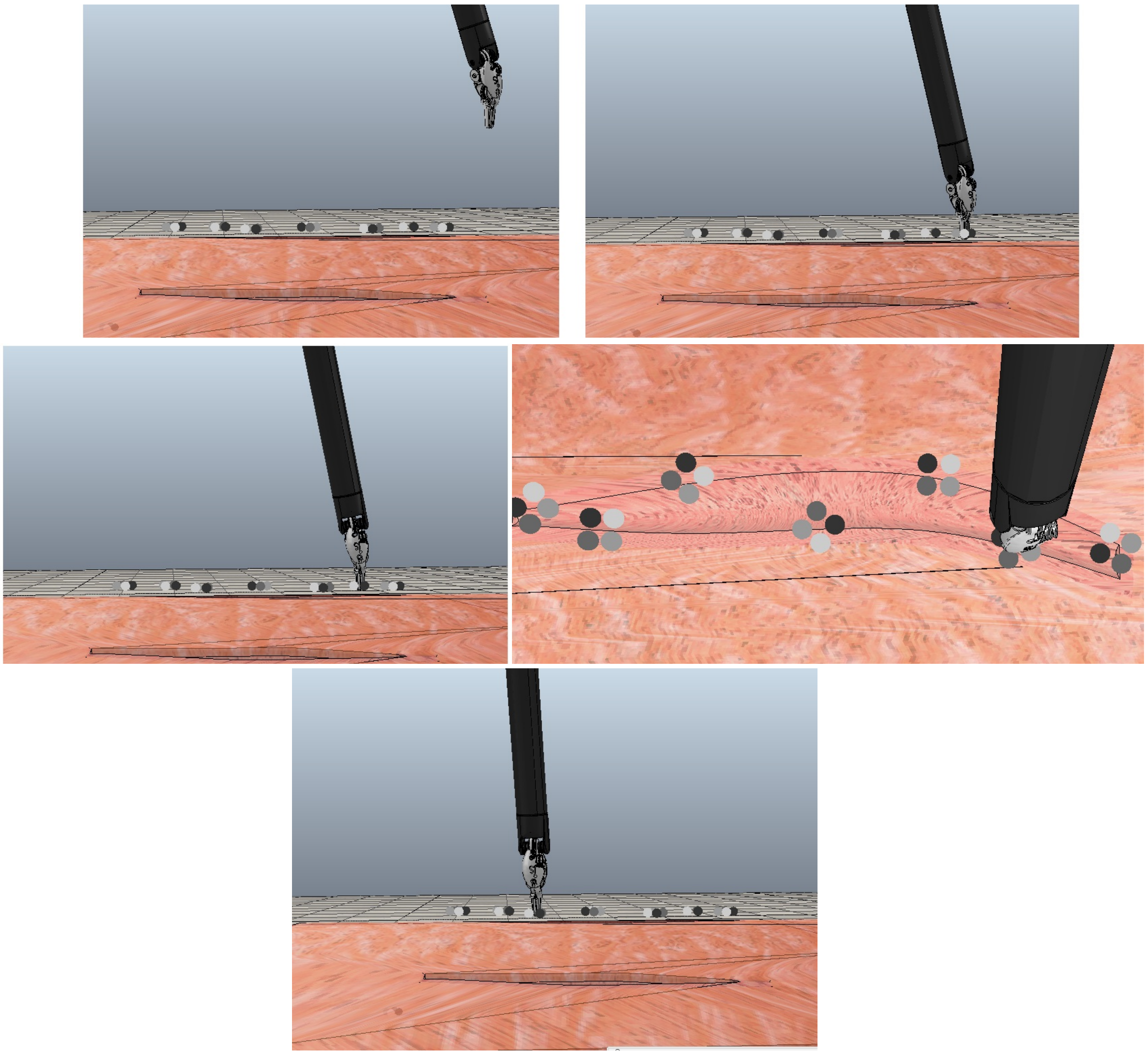


Figure 10. First experiment, some screenshots of the path followed by the end-effector.

Until now, during the considered path, the robot will never collide with any surface, therefore we have a pure vision-based control law. The error convergence for this case in shown in Fig. 11.

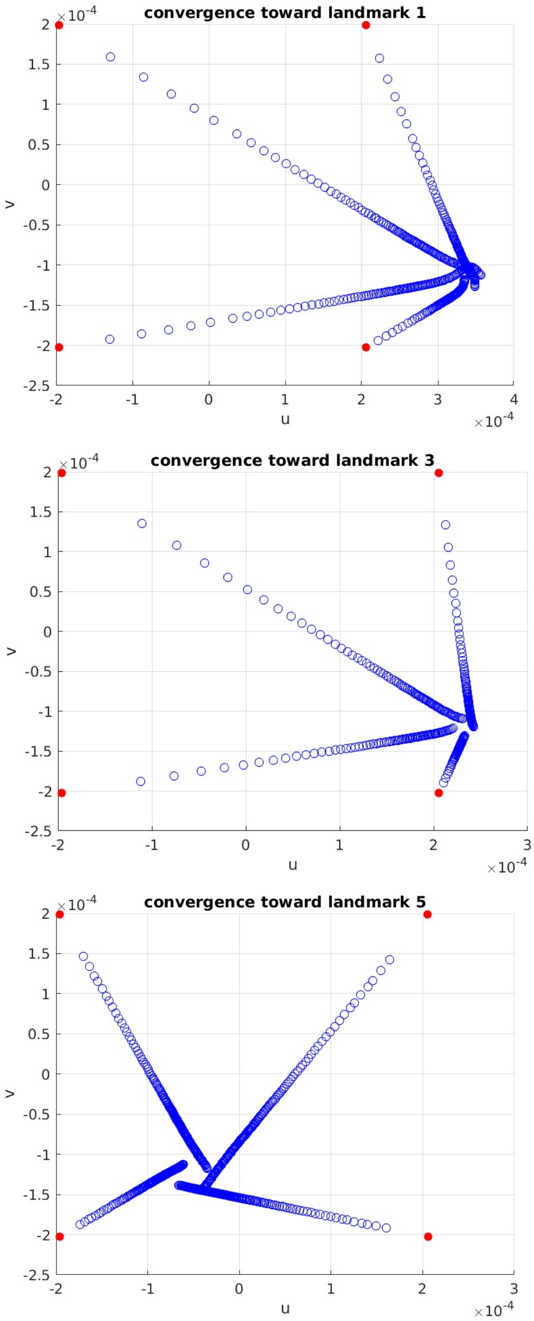


Figure 11. First experiment, the image error convergence for some landmarks.

The second experiment: testing the reaction to the force.

Now, we verify that the correction of the s^d into s^* is actually the one expected by our interpretation (see previous section). To the aim, we slightly modify the previous experiment in order to create the case when the end-effector approaches vertically to a landmark that is too deep and cannot be reached without forcing toward the patient skin. We observe the following expected behaviour:

1. as soon as the end effector touches the skin surface, it perceives a force and rapidly moves away the normal of the contact point,
2. by moving away, the force vanishes and the robot try again to reach the target pose, returning to step 1,
3. after a certain number of steps, the robot eventually converges.

In order to let the reader to visualize the experiment, some screenshots of are collected into Fig. 10.

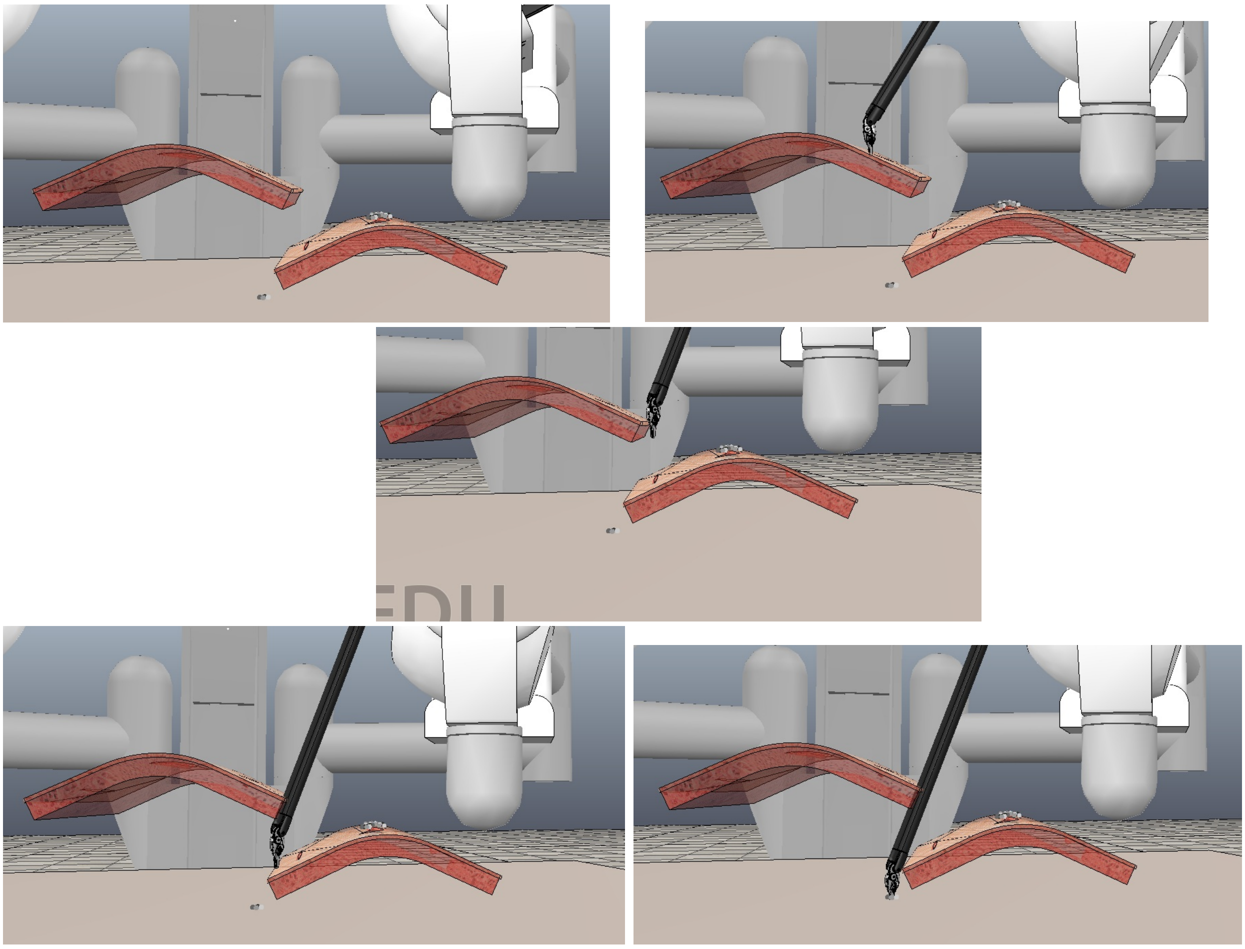


Figure 12. The second experiment, some screenshots of the path followed by the end-effector.

Appendix A An eye-to-hand variant.

In this section we design the eye-to-hand variant of the technique considered up to now. Essentially, there are three main difficulties:

1. In any eye-to-end case, the feature vector s has to be related to some features of the end-effector. In fact, because the camera is fixed, the feature error can change during the time only if the features are moving. For this reason, as shown in Figure (14), we introduce new four gray balls on the end effector, and we define the vector s as the pixel coordinates of the centers of these new balls during the time. This key difference is also shown by the authors in [6], that propose the following image:

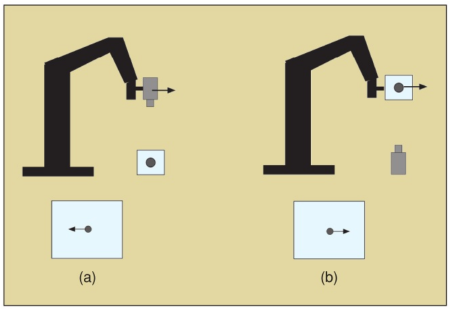


Figure 13. The image reported in [6]. It shows the key differences between the eye-in-hand and the eye-on-hand paradigms. In the former is reported on the left side, and the latter on the right side. Here, the features in s are represented with the icon ⊡, and we note that in the former case they are extrapolated by the scene, whereas in the latter case they are attached to the end effector.

2. In this new scenario, the features could not be seen all together from the camera. In fact, one of the four new gray balls attached to the end effector could be hidden by the end effector itself. See, for example, again the latter image in Figure (14). Therefore, at each iteration the control algorithm needs to recognize which gray balls are visible from the camera, and then build the vector s only for them. However, in order to uniquely determine the output of the control law, the camera actually needs to see only three non collinear balls. Since in most cases at least three of the four balls are visible, we practically have a sufficient number of features at each step of the control algorithm.

3. As it can be noticed by Figure (15), the considered task implies that the desired final pose of the end effector relative to a landmark is the same for all the landmarks. Due to this fact, in the eye-in-hand case the desired features s^d are the same for all the landmarks. However, since for the eye-to-hand case the camera is fixed, not mounted on the end effector, this is not true for the new settings. Therefore, now we have different values of $s_1^d, \ldots, s_\ell^d$ for each of the ℓ landmarks in the scene. In particular, the values of s_i^d, for $i \in [1, \ldots, \ell]$, are the pixel coordinates (u, v)'s of the four gray balls of the $i-$th landmark.

Other than these facts, according to [6], the mathematical expression of the control law is the same, unless from the following key fact:

> ***instead of moving the camera with the velocity $\boldsymbol{\tau} = \boldsymbol{L}^{\dagger}\boldsymbol{K}(\boldsymbol{s}^* - \boldsymbol{s})$ expressed w.r.t. the camera frame, we have now to move the end effector with the opposite velocity $-\boldsymbol{\tau}$, again expressed w.r.t. the camera frame***

Practically, the algorithm performs the following procedure:

- Firstly, it computes τ in the same manner of the eye-in-hand case, as reported in (10),
- Then, it moves the end effector of $-\tau$.

Remark. The procedure described above presupposes that the internal control law is able to handle the fact that the end effector velocity $-\tau$ is expressed in terms of the camera frame, and not (as in standard situation) w.r.t. the world frame or the end effector one. If the internal control law is not able to do that, we need to translate $-\tau$ in terms of one between these frames. However, this translation procedure is well known in literature [7], and is formally described by the following statement.

Theorem 1.

Let A and A' be two coordinates frames differing each other by a rototraslation described by the rotation matrix R and the translation vector t. The generic position vector p in A can be translated into the respective position p' in A' according to:

$$p' = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} p \tag{15}$$

Furthermore, the generalized velocities vector τ in A can be translated into the respective vector τ' in A' according to:

$$\tau' = \begin{pmatrix} R & [t]_\times R \\ 0 & R \end{pmatrix} \tau \tag{16}$$

where $[t]_\times$ is the skew symmetric matrix defined as

$$[t]_\times \triangleq \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{17}$$

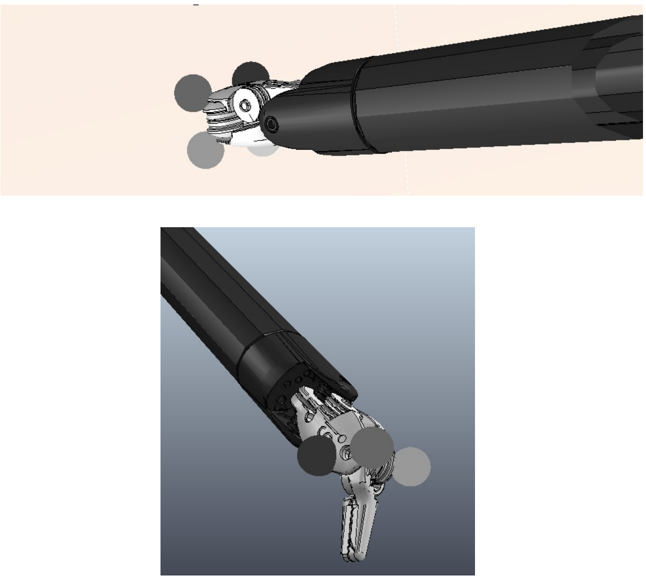


Figure 14. The four new gray balls attached to the end-effector.

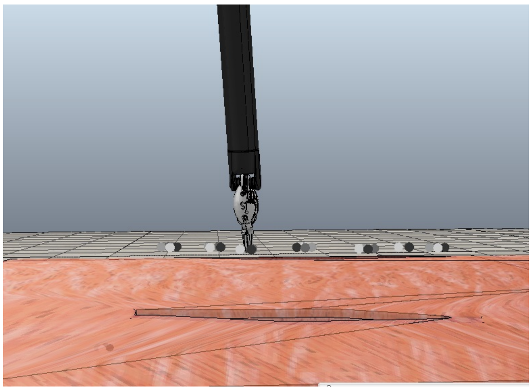


Figure 15. The desired final pose of the end effector when appraching to a landmark. Note that, relatively to the generic landmark, it is the same for all the landmarks.

Bibliography.

[1] Youcef Mezouar;Mario Prats;Philippe Martinet. *External Hybrid Vision/Force Control*. ICAR 2007, The 13th International Conference on Advanced Robotics August 21-24, 2007, Jeju, Korea

[2] B.J. Nelson; P.K. Khosla. *Force and vision resolvability for assimilating disparate sensory feedback.* IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 12, NO. 5, OCTOBER 1996.

[3] Caroline Nadeau; Alexandre Krupa. *Intensity-Based Ultrasound Visual Servoing: Modeling and Validation With 2-D and 3-D Probes* IEEE TRANSACTIONS ON ROBOTICS, VOL. 29, NO. 4, AUGUST 2013.

[4] G.A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, B. Siciliano, *"A V-REP Simulator for the da Vinci Research Kit Robotic Platform"*, BioRob, 2018. Avaliable online at: *https://github.com/unina-icaros/dvrk-vrep*

[5] S. Hutchinson, G.D. Hager, P.I. Corke, “A tutorial on visual servo control", *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651-670, 1996.

[6] F. Chaumette and S. Hutchinson, *“Visual servo control. II. Advanced approaches [Tutorial]*” in *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109-118, March 2007.

[7] R. Paul, “*Robot Manipulators: Mathematics, Programming and Control.*” Cambridge, MA: MIT Press, 1982