Adaptive and robust algorithms and tests for visual-based navigation of a space robotic manipulator

Marco Sabatini*, Riccardo Monti, Paolo Gasbarri, Giovanni B. Palmerini

Università di Roma “La Sapienza”, Italy

Abstract

Optical navigation for guidance and control of robotic systems is a well-established technique from both theoretic and practical points of view. According to the positioning of the camera, the problem can be approached in two ways: the first one, “hand-in-eye”, deals with a fixed camera, external to the robot, which allows to determine the position of the target object to be reached. The second one, “eye-in-hand”, consists in a camera accommodated on the end-effector of the manipulator. Here, the target object position is not determined in an absolute reference frame, but with respect to the image plane of the mobile camera. In this paper, the algorithms and the test campaign applied to the case of the planar multibody manipulator developed in the Guidance and Navigation Lab at the University of Rome La Sapienza are reported with respect to the eye-in-hand case. In fact, being the space environment the target application for this research activity, it is quite difficult to imagine a fixed, non-floating camera in the case of an orbital grasping maneuver. The classic approach of Image Base Visual Servoing considers the evaluation of the control actions directly on the basis of the error between the current image of a feature and the image of the same feature in a final desired configuration. Both simulation and experimental tests show that such a classic approach can fail when navigation errors and actuation delays are included. Moreover, changing light conditions or the presence of unexpected obstacles can lead to a camera failure in target acquisition. In order to overcome these two problems, a Modified Image Based Visual Servoing algorithm and an Extended Kalman Filtering for feature position estimation are developed and applied. In particular, the filtering shows a quite good performance if target’s depth information is supplied. A simple procedure for estimating initial target depth is therefore developed and tested. As a result of the application of all the novel approaches proposed, the experimental test campaign shows a remarkable increase in the robustness of the guidance, navigation and control systems.

Article info

Article history:
Received 19 January 2012
Received in revised form 21 August 2012
Accepted 8 October 2012

Keywords:
Space manipulators
Visual-based control
Robotics

1. Introduction

Space manipulators are complex systems, composed of robotic arms generally accommodated on an orbiting platform. They can be used to perform a variety of tasks, such as retrieval of spacecraft for inspection, maintenance and repair, debris removal or assistance to the astronauts during their extra-vehicular activities. In the next future autonomous orbiting robots, equipped with manipulator arms, could be used for assembling large space structures or servicing satellites that run out of propellant or need repair [1–3]. In past years the authors have studied guidance, navigation and control (GNC) issues of a space manipulator, leading to a complete model that takes issues as the unconstrained base, the orbital environment, the high flexibility of the links, the strict accuracy

* This paper was presented during the 62nd IAC in Cape Town.

* Corresponding author.

E-mail addresses: marcosabatini@hotmail.it, marco.sabatini@uniroma1.it (M. Sabatini), riccardo.monti@uniroma1.it (R. Monti), paolo.gasbarri@uniroma1.it (P. Gasbarri), giovanni.palmerini@uniroma1.it (G.B. Palmerini).
requirements into account [4–8]. In [9], the design of a preliminary test-bed to simulate in-orbit operations has been reported. GNC performance strictly depends on the determination of the kinematic state provided by the sensors; previous studies have considered typical encoders located at the joints, as well as the possibility of GNSS-based motion determination [10]. This paper instead aims to investigate, both in theory and practice, the possibilities of visual-based robot control techniques for target detection and grasping. These techniques represent an interesting alternative to already considered approach, as they can take the flexibility of the links (opposite to encoders) into account and be simpler to implement than GNSS-based approach.

Visual servoing concerns several fields of research including vision systems, robotics and automatic control; the techniques are generally classified depending on the number of cameras involved, on the position of the camera(s) with respect to the robot, on the choice of the error function to minimize in order to reposition the robot [11,12]. Visual servoing has been intensively studied during the last three decades. In [13] a pixel-based algorithm, that estimates depth and depth uncertainty at each pixel and incrementally refines these estimates over time, is introduced. In [14,15] algorithms are proposed for the solution of the robotic (eye-in-hand configuration) visual tracking and servoing problems. The visual tracking problem is solved by using the optical flow techniques to compute the vector of discrete displacements. In [16], a Position Based Visual Servoing (PBVS) is proposed. However, the servoing strategy that in the present paper is selected as best fit for the space manipulator operation is the Image Based Visual Servoing (IBVS) [17]. The difference between the two methods is that in the IBVS the control actions are evaluated taking only the image plane coordinates of the features that identify the target object into account; therefore, no information is strictly needed on the relative position of the target with respect to the camera, as instead in the PVBS. This also means that no 3D reconstruction of the object is needed, and therefore only 1 camera can be used for servoing (2 cameras would be needed for stereoscopy).

Even if some researches [18,19] have dealt with the problem of moving targets, only fixed targets are considered in this paper. The camera will be considered rigidly mounted at the end effector of the manipulator (eye-in-hand configuration). In the eye-in-hand, IBVS configuration, the control actions are evaluated comparing the image of the target in the current and in the final desired configuration. All the measurements refer to the image plane and are indeed bi-dimensional (details can be found in Section 2). Even if an information about the third dimension of the problem, i.e. the target depth, is needed, a rough approximation of this value can be typically used.

The paper validates the application of the IBVS algorithm to a space manipulator for an eye-in-hand camera first with a software simulation tool (Section 3), then by means of the experimental test-bed at the Guidance and Navigation Lab, Università di Roma “La Sapienza”. At present day, the space environment is only reproduced by removing the friction between the horizontal plane and the arm (thus canceling the gravity effect on the arm movement); however, the “shoulder” joint is connected to a fixed base as a terrestrial robot; in near future applications, the manipulator will be mounted on a free flying 2D platform already developed in the laboratory, thus approaching more closely the orbital scenario.

In the studied mission, possible causes of failure are identified in the presence of errors in the feature detection and in the actuation of the evaluated control signals. In fact, commands are executed with some time delay by the stepper motors located at the joints, as well as the image processing needed for target detection returns a noisy value of the feature in the image plane. Even if these problems do not seem to affect the IBVS in slow, limited range maneuvers, failures can occur in fast, wide range operations. In [20,21] it is stated that the first aim of any visual-servoing strategy is to avoid the features lost from the Field of View (FOV) and that the desired location may not be reached. However, avoiding both these system failures it turns out to be very difficult, especially when the initial and desired locations are distant. In the present paper, a Modified Image Based Visual Servoing (MIBVS) is introduced (Section 4) to overcome the problem of the target exiting from the camera FOV. A task assignment procedure, also present in [22], is realized. While the long links of the manipulator are controlled in such a way that the target is approached by the end effector, the short third link on which the camera is mounted quickly corrects the camera orientation to keep the target in the center of the FOV. The two tasks are conceptually performed at different velocities, with the “eye” moving much faster than the rest of the manipulator.

Another important issue that is taken into account to improve the system robustness and to decrease the possibility of failure is, the loss of the target due to the variation of lighting conditions or to the presence of obstacles between camera and target. To this aim an Extended Kalman Filter (EKF) for the estimate of the feature motion has been designed and tested (Section 5). The estimate is not only used to improve the accuracy in the knowledge of the feature kinematics state, but it can substitute the measurements when outages occur. The performance of the EKF is satisfactory if a correct dynamics for the feature in the image plane is implemented. This is only possible if an accurate estimate of the target depth is provided. Many works have focused on the possibility to estimate this information. In [23], a visual servoing approach is proposed where the target depth is observed and made available for servoing. In this paper, we propose a simple, fast pre-maneuver that is able to determine the initial target depth (Section 6). An on-line numerical propagation of the depth is then performed. The performance of the guidance, navigation and control systems developed is experimentally tested (Section 7) and it shows satisfactory performance in terms of accuracy, robustness and simplicity of implementation.

2. IBVS control theory

This section reports some fundamentals of the Imaged Based Visual Servoing (IBVS). Further details about the theory can be found in literatures [17,22].
The basic idea of IBVS consists in characterizing the image of a target to be reached by means of some key-features (points, lines, shapes already known to the observer), and by evaluating the control actions needed to reach the target only by taking the dynamics of these key-features into account. The basic key-feature is of course the point element: in fact, sets of points can be associated to an image representing the target and they will be eventually used in this paper to determine the desired control actions.

The image acquisition process is a transformation from the three-dimensional space where the target object is located, to a two-dimensional space where the image of the target object is detected. Let us introduce an inertial reference frame fixed to the camera, and an image reference frame fixed to the target object. The image plane is located at a fixed to the camera, and an image reference frame fixed to the target object is detected. Let us introduce an inertial reference frame where the origin of the camera reference frame is located, to a two-dimensional space where the image of the target object is reached by means of some control actions.

Note that $X_p, Y_p,$ and $Z_p$ are measured in meters, while $f, y$ and $z$ are measured in terms of pixels.

Since the digital images are acquired and processed as matrices, with elements sorted by row and column, the coordinates of a given point are usually given in terms of $u$ and $v$ (Fig. 1) and not in terms of $y$ and $z$. The transformation from $(u, v)$ to $(y, z)$ can be easily performed by applying

$$y = u - u_0$$
$$z = v - v_0$$

where $u_0$ and $v_0$ are the coordinates in the $(u, v)$ frame of the origin of the $(y, z)$ frame (center of the image). Of course this bias depends on the resolution of the image. All images considered in the following paragraphs will have a $640 \times 480$ resolution, meaning that $u_0 = 320$ and $v_0 = 240$.

Deriving with respect to time Eq. (2), it is possible to write, for any target point $P$ detected in the image:

$$y = f \frac{Y_p}{X_p}$$
$$z = f \frac{Z_p}{X_p}$$

(2)

(1)

which in matrix form reads as:

$$\begin{bmatrix}
y \\
z
\end{bmatrix} = \begin{bmatrix}
-\frac{y}{X_p} & f & 0 \\
-\frac{z}{X_p} & 0 & f
\end{bmatrix} \begin{bmatrix}
\dot{X}_p \\
\dot{Y}_p \\
\dot{Z}_p
\end{bmatrix} = J_1 \begin{bmatrix}
\dot{X}_p \\
\dot{Y}_p \\
\dot{Z}_p
\end{bmatrix}$$

(5)

Since $\dot{X}_p, \dot{Y}_p, \dot{Z}_p$ are the velocity components of the target point $P$ with respect to the camera, written in the camera reference frame, it is possible to link them, under the hypothesis of a fixed target, to the camera translational $(V_x, V_y, V_z)$ and rotational $(\Omega_x, \Omega_y, \Omega_z)$ velocities—written in the camera reference frame:

$$\begin{bmatrix}
\dot{X}_p \\
\dot{Y}_p \\
\dot{Z}_p
\end{bmatrix} = - \begin{bmatrix}
V_x \\
V_y \\
V_z
\end{bmatrix} - \begin{bmatrix}
\Omega_x \\
\Omega_y \\
\Omega_z
\end{bmatrix} \times \begin{bmatrix}
X_p \\
Y_p \\
Z_p
\end{bmatrix}$$

(6)

In the case of the robotic manipulator considered in this paper, the motion of the camera is planar, i.e. $V_z = 0$, $\Omega_x = \Omega_y = 0$ (see Fig. 2).
Eq. (6) therefore can be simplified as:
\[
\begin{bmatrix}
  \dot{X}_p \\
  \dot{Y}_p \\
  \dot{Z}_p \\
\end{bmatrix} = \begin{bmatrix}
  -1 & 0 & Y_p \\
  0 & -1 & -X_p \\
  0 & 0 & 0 \\
\end{bmatrix} \begin{bmatrix}
  V_x \\
  V_y \\
  \Omega_z \\
\end{bmatrix} = J_2 \begin{bmatrix}
  V_x \\
  V_y \\
  \Omega_z \\
\end{bmatrix}
\]
(7)

From Eqs. (5) and (7) it is therefore possible to refer the motion of the image of a target point to the motion of the camera:
\[
\begin{bmatrix}
  \dot{y} \\
  \dot{z} \\
\end{bmatrix} = J_1 \begin{bmatrix}
  \dot{X}_p \\
  \dot{Y}_p \\
  \dot{Z}_p \\
\end{bmatrix} = J_1 J_2 \begin{bmatrix}
  V_x \\
  V_y \\
  \Omega_z \\
\end{bmatrix}
\]
(8)

Again, the above equation can be written for any point detected in the image. In this way, a system of 2N equations (where N is the number of selected key-features) and 3 unknowns \((V_x, V_y, \Omega_z)\) is obtained:
\[
\begin{bmatrix}
  \dot{y}_1 \\
  \dot{z}_1 \\
  \vdots \\
  \dot{y}_N \\
  \dot{z}_N \\
\end{bmatrix} = L \begin{bmatrix}
  \frac{y_1 - y_{c1}}{x_{c1}} & -\frac{y_1^2}{x_{c1}} & -f \\
  \frac{z_1}{x_{c1}} & 0 & -\frac{y_1 z_1}{x_{c1}} \\
  \vdots & \vdots & \vdots \\
  \frac{y_N - y_{cN}}{x_{cN}} & -\frac{y_N^2}{x_{cN}} & -f \\
  \frac{z_N}{x_{cN}} & 0 & -\frac{y_N z_N}{x_{cN}} \\
\end{bmatrix} \begin{bmatrix}
  V_x \\
  V_y \\
  \Omega_z \\
\end{bmatrix} = J_{INT} \begin{bmatrix}
  V_x \\
  V_y \\
  \Omega_z \\
\end{bmatrix}
\]
(9)

where \(L\) is the (column) vector containing the components of the image points. The matrix \(J_{INT}\) is called interaction matrix. The above formula is maintained also in case that more complex features, instead of simple points, are selected, even if a not-so-simple interaction matrix will result.

In the IBVS approach, the control is evaluated directly on the bases of the information contained in the image captured by the camera, while no information is needed on the inertial position of the target. Considering a desired final relative position of the target with respect to the camera and labeling \(y_{1d}, z_{1d}, \ldots, y_{Nd}, z_{Nd}\) the images of the N key-features relevant to the desired configuration, the error at a certain time instant is defined as:
\[
e = \begin{bmatrix}
  y_{1d} - y_1 \\
  z_{1d} - z_1 \\
  \vdots \\
  y_{Nd} - y_N \\
  z_{Nd} - z_N \\
\end{bmatrix}
\]
(10)

Within the hypothesis that the desired target-camera relative geometry is not time-varying, the time derivative of the error is directly related to the velocity of the image points as follows:
\[
\dot{e} = \begin{bmatrix}
  \dot{y}_1 \\
  \dot{z}_1 \\
  \vdots \\
  \dot{y}_N \\
  \dot{z}_N \\
\end{bmatrix}
\]
(11)

The control actions can be now evaluated, for instance, by imposing the error to decrease in an exponential way, through an assigned gain matrix \(K\):
\[
\dot{e} = -K e \quad \Rightarrow \quad \dot{L} = -K e
\]
(12)

Combining Eqs. (9) and (12), it results as follows:
\[
K e = L = J_{INT} \begin{bmatrix}
  V_x \\
  V_y \\
  \Omega_z \\
\end{bmatrix}
\]
(13)

The motion of the camera needs to be related to the commands provided to the actuators, depending on the specific architecture of the manipulator. In the case of a 3-links (arm, forearm, rigid hand) robotic manipulator, the translation and rotation of the camera (written in the camera reference frame) can be expressed as function of the angular velocities at the manipulator’s joints by means of the geometric Jacobian \(J_{GEO}\):
\[
\begin{bmatrix}
  V_x \\
  V_y \\
  \Omega_z \\
\end{bmatrix} = J_{GEO} \begin{bmatrix}
  \dot{\theta}_1 \\
  \dot{\theta}_2 \\
  \dot{\theta}_3 \\
\end{bmatrix}
\]
(14)

The complete expression of \(J_{GEO}\) for the three links manipulator can be found in Appendix.

Finally, the joint motors can be commanded by a guidance law, to be evaluated by substituting Eq. (14) in Eq. (13):
\[
K e = J \begin{bmatrix}
  V_x \\
  V_y \\
  \Omega_z \\
\end{bmatrix} = J_{INT} J_{GEO} \begin{bmatrix}
  \dot{\theta}_1 \\
  \dot{\theta}_2 \\
  \dot{\theta}_3 \\
\end{bmatrix}
\]
(15)

These time-varying angular velocities are the control signal for the joint motors.

3. Numerical simulation of IBVS performance

A preliminary totally software-level test of the proposed IBVS algorithm is strongly advisable, both to provide confidence in the approach and to define the dynamic behavior to be expected during the tests involving hardware.

To this aim, a software tool for numerical simulation of image acquisition and manipulator control has been prepared. The coordinates in the inertial reference frame of a target body, characterized by six feature points (ordered in three rows and two columns, just like the actual experimental target, see Section 7), and of the end-effector location are introduced as input. Performing the required transformations, the relative position of the features, with respect to the camera written in the camera reference frame, can be obtained. Afterwards, the current and the desired final position of the pixels can be identified. Based on the error between the current and the desired position, the required joints angular velocities are evaluated and the manipulator kinematics are propagated.

As a first example, a case with an ideal, noiseless camera and perfect actuators is considered. The robotic
The arm is initially in a position such that the target is seen in the center of the camera Field Of View (FOV), see Fig. 3, upper left frame. Applying the IBVS control laws, the manipulator moves and the camera gets closer and closer to the target (Fig. 3 upper right and lower left frames), until it gets to the final desired relative position with respect to the target (Fig. 3 lower right frame). In the image plane, the features move along straight lines during the maneuver (Fig. 4). Since the required motion is a simple translation of the camera along its optical axis (X axis), the commanded velocities are such that only $V_x \neq 0$, while $V_y = 0$, $\Omega_z = 0$, as shown in Fig. 5. In such an ideal behavior, the components $y$ and $z$ of the error decrease in an exponential way just as expected from Eq. (12), see Figs. 6 and 7.

Then, to consider more realistic behaviors, two sources of error are introduced in the code: the inaccuracy on the feature measurement and the time delays in the joints’ actuation. The process of features extraction from an acquired image is characterized by filtering and approximations (see details in [26]) leading to a noisy measurement of the features: Fig. 8 reports the trajectories of the features on the image plane if a three-pixels random error is added to the previous measurements. Due to this error, the control actions become noisy, and, as a result, Fig. 9 shows how all the components of the target velocity ($V_x \neq 0$, $V_y \neq 0$, $\Omega_z \neq 0$) are now different from zero.

The mean value of $V_y$, $\Omega_z$ is still nearly zero, and, in the simulation test, this behavior does not prevent the maneuver to succeed. Dealing with real hardware, however, sudden variations on the commanded velocities could not be provided due to the inherent slower dynamics of the stepper motors.

Moreover, in the present implementation the three motors could have different response delays (even if the difference could be less than 0.1 s) due to different communication time with microcontrollers and drivers, and mechanical response time. This error source can lead to a camera motion so that the acquired features do not describe a rectilinear trajectory on the image plane. In Fig. 10 a slightly slower response of the wrist motor (with respect to shoulder and elbow motors) is simulated. The initial features are not in the center of the image but peripheral (see the circles in the right side of Fig. 10). The non-rectilinear trajectory seen by the camera can bring some of the features to exit from its FOV (ranging $[-320, 320]$ along $y$ and $[-240, 240]$ along $z$). While the simulation is not seriously affected by this behavior, the actual experimental maneuver would stop suddenly, due to the loss of the target, and the result is a failure of the mission.

Possible solutions to the problems highlighted by these software simulations are proposed in the remainder of the paper. First of all (see Section 4) a modification to the standard IBVS algorithm described in Section 2 is...
proposed in order to prevent the key features from exiting the camera FOV. To this aim, a dedicated control law is derived and eventually applied to the third link so that its task is no more the target approaching (a task that is reserved to the first two links) but keeping the target in the center of the image.

Afterwards (see Section 5) an Extended Kalman Filtering for the estimation of the features position in the image plane is studied and implemented. The aim is to reduce the measurement noise and to provide an effective tool for predicting the feature dynamics in case that the target is not acquired during some instants of the maneuver (sort of visual black-out) for some unexpected reason (bad light conditions, presence of an obstacle, etc.).

4. Modified IBVS algorithm (MIBVS)

This section introduces a Modified Image Based Visual Servoing (MIBVS), whose main goal is the capability to perform fast and wide-range maneuvers preventing target’s exit from the camera FOV.
The MIBVS is illustrated for the specific case of the three links, three actuators robotic manipulator operated at the Guidance and Navigation Lab. The basic idea is that different tasks can be assigned to the different actuators of the arm, even if the final goal is common, i.e. bringing the image of the features to match the desired position in the image plane. Therefore, actuators #1 and #2, i.e. shoulder and elbow motors, will be assigned the task to approach the target, while actuator #3, i.e. wrist motor, will take care of maintaining the target inside the FOV. These two tasks can also be performed at different dynamic rates: in the specific case, the wrist's actuator task is accomplished at higher rate with respect to the approaching one.

In detail, the first two motors are commanded so that they respect the classic IBVS approach, and, as a result,
also the third link will be dragged near the target:

\[
\begin{pmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2 \\
\dot{\theta}_3
\end{pmatrix} = J^{-1}Ke
\]  

(16)

Original to MIBVS; however, only the first two components of the solution, \( \theta_1 \) and \( \theta_2 \), are used to command the joint motors. The task of keeping the target in the FOV is granted by fixing for the third link a new “virtual” target, here selected as the middle point of the features points, and by fixing a desired position for this virtual target to be in the center of the image. That is:

\[
\begin{align*}
\bar{y} &= \frac{y_1 + y_4}{2} \\
\hat{y}_{\text{des}} &= 0
\end{align*}
\]  

(17)

where \( y_1, y_4 \) are the upper left and upper right features of the current target. A new, single row interaction matrix is
then associated to this virtual target:

$$\tilde{J}_{\text{INT}} = \left[ \frac{\dot{y}}{\dot{X}} - \frac{f}{X^2} - f' \right]$$  \hspace{1cm} (18)$$

The modified error for the FOV task is now $\tilde{e} = \tilde{y}_{\text{des}} - \tilde{y}$ to which the exponential law $\tilde{e} = -\tilde{K}\tilde{e}$ is applied. The value of the gain $\tilde{K}$ can be remarkably larger than $K$ in order to have faster dynamics for the third link (FOV preserving task) with respect to the first two links (target approaching task). With the above assumptions, the commanded angular velocity for the third link is now:

$$\tilde{J}_{\text{INT}}J_{\text{GEO}} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \tilde{J}_{\text{INT}} \begin{bmatrix} J_{\text{GEO}}^{12} \\ J_{\text{GEO}}^{13} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \tilde{K} \cdot \tilde{e}$$  \hspace{1cm} (19)$$

where $\dot{\theta}_1$ and $\dot{\theta}_2$ are known from Eq. 16. On account of previous steps the geometric Jacobian has been divided into two submatrices, namely $J_{\text{GEO}}^{12}$ multiplying $[\dot{\theta}_1, \dot{\theta}_2]$ and

![Fig. 10. Trajectory of acquired image features in the case of non-perfectly synchronized motors; arrows indicate when the features are exiting the FOV.](image)

![Fig. 11. Trajectory of the acquired image features from in the case of non-perfectly synchronized motors and MIBVS control applied.](image)
where $\lambda_{GEO}$ multiplying $\theta_3$. By performing simple algebra we finally obtain:

$$J_{INTGEO}^2 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + J_{INTGEO}^1 \dot{\theta}_3 = \tilde{\mathbf{K}} \tilde{e}$$

(20)

$$\dot{\theta}_3 = (J_{INTGEO}^1)^{-1} \left( \tilde{\mathbf{K}} \cdot \tilde{e} - J_{INTGEO}^2 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right)$$

(21)

It is clear that this strategy works properly only if the task of bringing to zero the error $e$ is not opposite to the task of bringing to zero the error $\tilde{e}$. A modification to overcome this limit in case that $e=0$ does not correspond to $\tilde{e}=0$ will be presented in Section 8.

Considering again the numerical simulation in Section 3 (Fig. 10), in which the target exits from the FOV, the MIBVS is now applied. Fig. 11 shows the trajectories of the features in the image plane. Once again, trajectories are not straight lines, but the faster dynamics of the third motor allows for a sudden centering of the target, followed by the slower target approaching maneuvering. With MIBVS the entire mission is performed while keeping the target in the center of the FOV, and the risk of a sudden stop due to the target’s loss is avoided.

5. Extended Kalman filter for image target tracking

A complementary tool proposed for detecting and tracking a target is the Extended Kalman Filter (EKF) [27], capable to estimate the dynamics of the features inside the image. EKF is intended to (a) improve the feature measurement accuracy, and (b) allow to continue the maneuver even if, for any unexpected reason, the acquired target is lost (i.e. no more detected) for a certain time interval. In this case, the state propagation of the state (i.e. the coordinates of the features) can substitute the estimate, preventing the maneuver to stop. Other more recent algorithms, such as the Unscented Kalman Filter, could be also considered, but, due to the simplicity and the good performance obtained by designing the EKF, the authors prefer to choose a tool (EKF) that requires a smaller computation time to provide an estimate, due to the limitation of a typical onboard hardware.

Following the classical EKF loop, the best estimate is evaluated by propagating the state according to a convenient dynamic process, and updating the result with the measurements.

The selected state vector is the vector $L$ defined in Section 2. The assumed dynamic process is a nonlinear function:

$$L = f(L, u)$$

where $u = \begin{bmatrix} V_x \\ V_y \\ \Omega_z \end{bmatrix}$.

From Eq. (9) it is possible to write (for any feature):

$$\dot{y} = -\frac{\Omega_z}{f} y^2 + \frac{V_x}{x_p} y - \frac{V_y}{x_p} f \Omega_z = f_y$$

$$\dot{z} = -\frac{\Omega_z}{f} z y + \frac{V_z}{x_p} z = f_z$$

(23)

The Jacobian matrix of the function $f(L, u)$ results:

$$F = \frac{\partial f}{\partial L} = \begin{bmatrix} \frac{\partial f_x}{\partial L_x} & \frac{\partial f_x}{\partial L_y} & \frac{\partial f_x}{\partial L_z} \\ \frac{\partial f_y}{\partial L_x} & \frac{\partial f_y}{\partial L_y} & \frac{\partial f_y}{\partial L_z} \\ \frac{\partial f_z}{\partial L_x} & \frac{\partial f_z}{\partial L_y} & \frac{\partial f_z}{\partial L_z} \end{bmatrix}$$

(24)

Following EKF rules, the propagation of the state vector $L$ and of the covariance error matrix $P$ at the time $t_k$, can be evaluated starting from the estimate at the time $t_{k-1}$.

$$L(t_k) = L(t_{k-1}) + f(L, u) \cdot dt$$

$$P(t_k) = P(t_{k-1}) + (FP(t_{k-1}) + P(t_{k-1})F^T) \cdot Q \cdot dt$$

(25)

where $Q$ is the covariance matrix associated with the not-to-be-perfect selected dynamic process. Since the state is directly measured, the measurement equation, relating the measurements $M$ to the state $L$, is simply given by:

$$M = HL$$

(26)

being $H$ an identity matrix of order $2N-N$ being the number of feature points ($N=6$ in our case).

The Kalman gain $K_{EKF}$ can be evaluated as:

$$K_{EKF} = P(t_k)H^T(HP(t_k)H^T + R)^{-1}$$

(27)

being the covariance matrix of the measurements’ error. The update of the propagated state and the covariance error matrix, in order to obtain the estimate at time $t_0$, is given by:

$$L(t_k^+) = L(t_k^-) + K(M - HL(t_k^-))$$

$$P(t_k^+) = (I_{12} - KH)P(t_k^-)$$

(28)

being $I_{12}$ an identity matrix, of order 12 in the present case.

The simulation software tool described in Section 3 is now adopted to test the EKF performance. One of the key characteristics of the IBVS depends on the fact that the actual target depth $X_p$ should be known to correctly evaluate the interaction matrix, but the algorithm works properly even if an approximate, constant, value of $X_p$ is assumed [23]. However, the performance of the EKF depends on the accuracy of the process dynamic Eq. (23). An error on $X_p$ results in an error on the estimates of $y$ and $z$. Fig. 12 reports the comparison between the measured value and the estimate value on the $z$ coordinate of one of the point features. At a given time, an outage period of the measurement is simulated. The mission is not interrupted but the control actions are evaluated on the basis of the propagation alone. However, if a constant value of $X_p$ is used (in the example $X_p=13$ cm, which is the value that we have in the final configuration), the propagation will not match the actual value, as it is possible to see from the difference between the measurement and the estimate when the outage period ends.

Fig. 13 shows the same maneuver when $X_p=X_p(t)$ is the actual time-varying value. It is possible to see that when the outage period ends, the estimate and the measurements have really close values. Hence the importance of having an accurate estimate of the value $X_p$ is evident. In the following Section 4 a simple method for estimating the target depth (and the focal length) is proposed.

6. Focal length and depth evaluation

The focal length $f$ of the camera is an intrinsic parameter that can be evaluated with the common methods
known in literature [24,25]. However, taking the first of the Eq. (23) into account:

\[
y = \frac{O_x \, y^2 + V_x \, y \, f \, f \, \Omega_z}{X_p \, f}
\]

it is possible to evaluate the focal length \( f \) with a simple, known motion of the camera. Specifically, if the motion of the camera is a simple rotation of amplitude \( \Delta \Theta \) without translation, there will be:

\[
V_x = 0, \quad V_y = 0, \quad \Omega_z = \frac{\Delta \Theta}{\Delta t}
\]

Leading to rewrite Eq. (29) as:

\[
\dot{y} = \frac{\Delta y}{\Delta t} = -\frac{1}{f} \frac{\Delta \Theta}{\Delta t} \, y^2 - f \frac{\Delta \Theta}{\Delta t}
\]
assuming the approximation of the time derivative with a finite difference with respect to time. The term \( \Delta y \) represents the difference in the image plane of the \( y \) coordinate of each feature before and after the rotation. Then, the focal length can be evaluated by solving the resulting second order algebraic equation:

\[
f^2 \Delta \Theta + f \Delta y + y^2 \Delta \Theta = 0
\]

(32)

for every point belonging to the chosen set of \( N \) features. Selecting only the feasible value (between the two solutions given by each second order equation) and averaging among the \( N \) values, a suitable estimate of the focal length of the specific camera can be obtained (\( f = 850 \) pixel in the example here considered).

A similar approach is adopted to estimate the initial target depth, even if the rotation maneuver needed to evaluate \( f \) can be performed once and for all, while the target depth must be evaluated when the maneuver starts. Once the initial value of \( X_p \) is evaluated, the time behavior of the target depth is estimated by remembering that:

\[
dX_p \quad = \quad -V_x
\]

(33)

In this “pre-mission” short maneuver, only a translation is commanded to the camera: \( \Omega_t = 0 \). Approximating the incremental displacement of the camera \( \Delta X, \Delta Y \) in the time interval \( \Delta t \) with the velocities \( V_x, V_y \), and the angular displacement \( \Delta \theta_1, \Delta \theta_2, \Delta \theta_3 \) with the angular rates \( \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3 \), it is possible to write:

\[
\begin{bmatrix}
\Delta \theta_1 \\
\Delta \theta_2 \\
\Delta \theta_3
\end{bmatrix} = J_{\text{geo}}^{-1} \begin{bmatrix}
\Delta X \\
\Delta Y \\
0
\end{bmatrix}
\]

(34)

It should be noted that, due to all the approximation performed, once these angular displacements are commanded to the joint motors, the camera is not moved to the expected \( \Delta X, \Delta Y \) values but to slightly different \( \Delta \hat{X}, \Delta \hat{Y} \), values that can be evaluated geometrically for a correct result (\( \Delta \hat{X}, \Delta \hat{Y} = \Delta \hat{X}, \Delta \hat{Y} \) only for \( \Delta t \rightarrow 0 \)). On account of this, Eq. (29) becomes:

\[
\frac{\Delta y}{\Delta t} = \frac{1}{X_p} \frac{\Delta \hat{X}}{\Delta \hat{y}} f \frac{\Delta \hat{Y}}{X_p} \Delta t
\]

(35)

With \( N \) point features, it is possible to write \( N \) times:

\[
X_p = \frac{y \Delta \hat{X} - f \Delta \hat{Y}}{\Delta y}
\]

(36)

Averaging the result, a good estimate of the target depth is obtained, as it will be shown in the following Section 7, where the experimental results will be illustrated.

7. Experimental results

Aside from numerical simulations, a test campaign has been carried out to analyze the proposed algorithm. To approach the conditions in space, a frictionless motion has been reproduced, even if limited to a planar surface. The robotic arm used for the experimental test of the proposed visual based control is depicted in Fig. 14, while the details of the test bed can be found in [26].

7.1. Depth estimate verification

The algorithm for initial target depth determination has been tested in a number of initial conditions, schematically reported in Fig. 15. From the initial configuration, the arm is commanded so that the camera (i.e., the end effector) slightly moves with displacements \( \Delta X = 0, \Delta Y = 0.1 \text{ m} \) while \( \Omega_z = 0 \) – and then comes back to its initial position. Fig. 16 reports the measurements of the depth \( X_p \) evaluated via “direct” measurement and the value estimated by the image processing technique described in previous Section 6. The two curves nearly overlap. This result not only confirms how the proposed technique to estimate the initial depth works for a wide range of values, but also that the focal length is correctly evaluated as well. In fact, from Eq. (36) it is possible to see how the value of \( X_p \) is strictly dependant on \( f \) and the obtained, correct estimate of \( X_p \) is therefore an indirect validation of the method used for evaluating \( f \).
7.2. IBVS and MIBVS strategies

As a first experimental test of the IBVS and MIBVS strategies, the same short range-slow maneuver numerically simulated in Section 3 (Fig. 3) is performed experimentally. Therefore the initial condition #1 in Fig. 17 is considered in this first test.

The behavior of the robotic arm commanded by means of IBVS is nearly ideal, as it is possible to see from Fig. 18, reporting the error on the $z$ component of the image features with respect to the desired feature position on the image plane. The curves for the numerically simulated results (dotted lines) and for the experiment results (solid lines) look pretty the same way.

The performance remarkably changes when more complex maneuvers are required. In particular, when a wider range of movements is involved under the constraint of a fast dynamics, the problems already noticed during numerical simulations (Section 3) actually cause the failure of the maneuver.

As an example, let us consider the initial condition #2 in Fig. 17. Due to the different actuation delays of the joint motors (caused by the presence of different microcontrollers, drivers, friction, and so on), the initial, fast movement of the manipulator leads to losing the target, which exits from the camera FOV, and the measurements vanish. From Fig. 19, it is possible to see that the implemented EKF still substitutes the lack of measurement with a propagated state (solid line), but since the target lost happens in a very early stage of the maneuver – and the filter has not converged yet – such a propagation is not fair enough to re-acquire the target. The maneuver therefore continues following a wrong propagated feature, and the fact that the target does not re-enter the camera FOV anymore is a clear demonstration that the mission is failed.

The same maneuver is now repeated with the MIBVS strategy. The gain value for the approaching task (see Eq. (16)) is maintained equal to the one in previous, failed test with IBVS algorithm:

$$K = 0.03I_{12}$$

The gain associated to the third actuator for the in-FOV keeping task (Eq. (20)) is instead selected as much higher:

$$\tilde{K} = 5$$

Figs. 20 and 21 show the experimental results in terms of errors on the $y$ and $z$ coordinates. It is possible to see that the errors first become symmetric, due to the successful in-FOV keeping task which maintains the target in the middle of the image, and then decrease in an exponential way. The irregularities occurring especially in the error along the $y$ coordinate are mainly due to the third joint stepper motor (wrist actuator), which is, in the current test bed, of a different type with respect to the other ones and provides a lower performance.
The advantage of implementing the pre-maneuver technique to compute an initial value of the target depth $X_p$ (paragraph VII.I) is clearly proofed by Fig. 22. The propagation in time, according to Eq. (33) leads to a final value ($X_p = 13$ cm) which is very close to the actual distance between the camera and the target in the final configuration.

Fig. 23 reports the angular rates evaluated with MIBVS strategy and commanded to the joint motors. Notice the initially much faster dynamics (high values of $\theta_3$) associated by MIBVS to the third actuator. Once target image has been constrained to the center of the FOV, the commanded actions of the third link become much slower, of the same order as the first two motors. Again, the irregular behavior of $\theta_3$ clearly noticeable in Fig. 23, is due to the low resolution of the third joint stepper motor.

### 7.3. EKF performance test

The experimental maneuver has been repeated to test the actual performance of EKF. To this aim, two target outages are artificially added (by covering the target) during manipulator motion. The maneuver does not stop, but proceeds by evaluating the control actions on the
basis of the estimate of the image features which relies uniquely on the propagated state, as no information comes from camera measurements. Fig. 24 reports the behavior along the z axis of one of the features. The estimate and the measurement can be compared: as long as a measurement (gray dot) is provided, the estimate is very close to the measurement, and, as expected, more regular as the EKF acts as a smoother. When the measurement is suddenly removed (target obscured, no measurements), the EKF only propagates the state, until a new measurement is finally re-acquired. At the end of both the outage periods the propagated state and the newly incoming measurements almost agree, meaning that the selected dynamics and tuning of the filter are correct and that the EKF is able to handle quite a long period of lacking measurements.

Fig. 25 reports the behavior of the same feature along the y axis of the image plane during the same maneuver, basically leading to the same conclusions. Furthermore, by observing the behavior during outages, it is possible to
notice how the remarkable, high frequency oscillations, due to the third actuator’s vibrations, disappear in the estimate, as the filter does not include a detailed model for the stepper motor.

7.4. Mixed IBVS – MIBVS strategy

As stated in Section 4, the MIBVS strategy shows its limitations when the task of maintaining the target in the center of the camera FOV contrasts with the task of reaching the desired final configuration of the features in the image plane. An example is provided by the case of a desired final feature position that is not centered, but shifted in the right half of the image plane (blue dot in Fig. 26). The third joint motor forces the camera to keep the target in the center of the FOV, while the first two joint motors manage to perform the target approaching, but, since it is in contrast with the third motor command, the desired configuration is not attained. Fig. 26 shows the trajectories of the features in the image plane and the remarkable final residual error between measured feature and desired feature.

The solution to this problem can be easily found by dividing the maneuver into two phases. In the first phase,
when the camera is far from the target, the errors are large and the commanded action indeed significant. The MIBVS strategy already proved more robust with respect to simple IBVS, and so it can be successfully adopted in this phase. However, when the camera gets closer to the target, and the stall condition seen in Fig. 26 could be reached, a switch to the classic IBVS should be performed. In the experimental test here presented, the switching condition is imposed basing on the values of the commanded joint angular velocities: when $\dot{\theta}_1, \dot{\theta}_2 \leq 0.5 \, ^\circ/s$ (i.e. the end of MIBVS maneuver is close) the control is switched to the IBVS algorithm. Fig. 27 reports the trajectories of the features in the image plane.

Fig. 28 shows the error with respect to the desired feature along the $y$ axis for the cases of MIBVS strategy (gray dotted line) and mixed MIBVS+IBVS strategy. It is clear how in this not-centered final configuration, the MIBVS would end up with a residual, non-zero error.
The proposed mixed strategy leads instead to a fully successful mission, with target first moving to the center of the FOV (initial phases) and then attaining its desired final configuration. As final remark we can conclude that the MIBVS is a useful tool to prevent failures during the initial phases (i.e. the loss of the target), while IBVS is needed to avoid final stalling conditions and to bring the features to the correct final desired location.

8. Final remarks

The paper discusses the application of the Image Based Visual Servoing (IBVS) strategy to space manipulators. First, software simulations to verify visual control performance are implemented. It is shown how in the ideal, frictionless environment, with perfect actuators and noiseless camera measurements, the results exactly

---

**Fig. 26.** Trajectories on the image plane of the features. A final error is still present between actual features and desired features, since the desired features are not in the center of the image.

**Fig. 27.** Trajectories on the image plane of the features. MIBVS is used to prevent instability in the initial phases, IBVS is used to achieve precise final positioning.
match the expected behavior. However, the software tool allows to introduce the characteristics of a more realistic case: the feature detection in the image plane can be affected by an error, and the actuators can be subjected to time delays in responding to a given command. As a result of these imperfections, it is shown how classic IBVS can fail.

In order to overcome these problems, an original Modified Image Based Visual Servoing (MIBVS) is proposed, also combined with an Extended Kalman Filter (EKF) for the estimation of the image features dynamics. The MIBVS special feature consists in assigning different tasks to the controlled motors located at the joints of the manipulator. Some of these actuators are still commanded to reach the desired final configuration, as in IBVS; but some other ones are controlled in such a way that the target never exits from the camera field of view. Once MIBVS and EKF have been validated in software tests, this strategy is applied to the experimental 2D frictionless manipulator test-bed of the Guidance and Navigation Lab at Sapienza – Università di Roma.

Experimental results are reported in this paper. Standard IBVS is demonstrated to fail, as expected, while dealing with fast dynamics, wide movement range situations. Instead, MIBVS can cope with stepper motors and webcam imperfections. It is also shown that the implemented EKF, completed by a simple but successful algorithm for evaluating the target depth, is a powerful tool to improve the performance, and allows to continue operations even if the target, for any reason, is not acquired by the camera. As a result, it is possible to affirm that the modification to IBVS and the novel navigation algorithm proposed have added a significant robustness with respect to the direct application of the classic approach.

Appendix

The geometric Jacobian $J_{\text{GEO}}$ of the manipulator is reported in this appendix element by element:

$J_{\text{GEO}}(1,1) = -L_1 \cos(\theta_1 + \theta_2 + \theta_3) \sin(\theta_1) - L_2 \cos(\theta_1 + \theta_2 + \theta_3) \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \cos(\theta_1 + \theta_2)$

$J_{\text{GEO}}(2,1) = L_1 \sin(\theta_1 + \theta_2 + \theta_3) \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2 + \theta_3) \sin(\theta_1 + \theta_2) + L_3$

$J_{\text{GEO}}(3,1) = 1$

$J_{\text{GEO}}(1,2) = -L_2 \cos(\theta_1 + \theta_2 + \theta_3) \sin(\theta_1 + \theta_2) - \sin(\theta_1 + \theta_2 + \theta_3) \cos(\theta_1 + \theta_2)$

$J_{\text{GEO}}(2,2) = L_2 \sin(\theta_1 + \theta_2 + \theta_3) \sin(\theta_1 + \theta_2) + L_3 + L_2 \cos(\theta_1 + \theta_2 + \theta_3) \cos(\theta_1 + \theta_2)$

$J_{\text{GEO}}(3,2) = 1$

$J_{\text{GEO}}(1,3) = 0$

$J_{\text{GEO}}(2,3) = L_3$

$J_{\text{GEO}}(3,3) = 1$

References


