INT. JOURNAL OF COMPUTER INTEGRATED MANUFACTURING

Generalization of Reference Filtering Control Strategy for 2D/3D Visual Feedback Control of Industrial Robot Manipulators

J. Ernesto Solanes[a], Pau Muñoz-Benavent[a], Leopoldo Armesto[a], Luis Gracia[a] and Josep Tornero[a]

[a]Instituto de Diseño y Fabricación, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain

ARTICLE HISTORY
Compiled July 30, 2021

ABSTRACT

This paper develops the application of the Dual Rate Dual Sampling Reference Filtering Control Strategy to 2D and 3D visual feedback control. This strategy allows to overcome the problem of sensor latency and to address the problem of control task failure due to visual features leaving the camera field of view. In particular, a Dual Rate Kalman Filter is used to generate inter-sample estimations of the visual features to deal with the problem of vision sensor latency, whereas a Dual Rate Extended Kalman Filter Smoother is used to generate more convenient visual features trajectories in the image plane. Both 2D and 3D visual feedback control approaches are widely analyzed through the paper, as well as the overall system performance using different visual feedback controllers, providing a set of results that highlight the improvements in terms of solution reachability, robustness and time domain response. The proposed control strategy has been validated on an industrial system with hard real-time limitations, consisting of a 6 DOF industrial manipulator, a 5 MP camera and a PLC as controller.

KEYWORDS
Visual Servoing; multi rate control; industrial robot manipulator

1. Introduction

Recent advances in computer science, control theory, robotics and technology have allowed the application of robot visual feedback control or visual servoing for solving complex tasks (Hutchinson, Hager, and Corke (1996); Chaumette and Hutchinson (2007); Corke (2011)). For instance, in the food industry, due to the high variety of products and features (e.g., shape, size, color, etc.), visual servoing is applied for grasping and placing tasks (Wu et al. (2017)). In medicine, visual information from endoscopies or Computed Tomography (CT) scans is used to manoeuvre robot manipulators during surgeries (Gangloff, Nageotte, and Poignet (2013); Azizian et al. (2014, 2015)). In the manufacturing industry, visual servoing is used in many applications such as welding, assembly, painting, etc. (Li et al. (2019); Muñoz-Benavent et al. (2019); Chen et al. (2019)).

In general, any industrial visual servoing system has, at least, the following main elements (see Fig. 1): a vision system in charge of extracting the required information

Corresponding Author: J. Ernesto Solanes (email: esolanes@idf.upv.es)

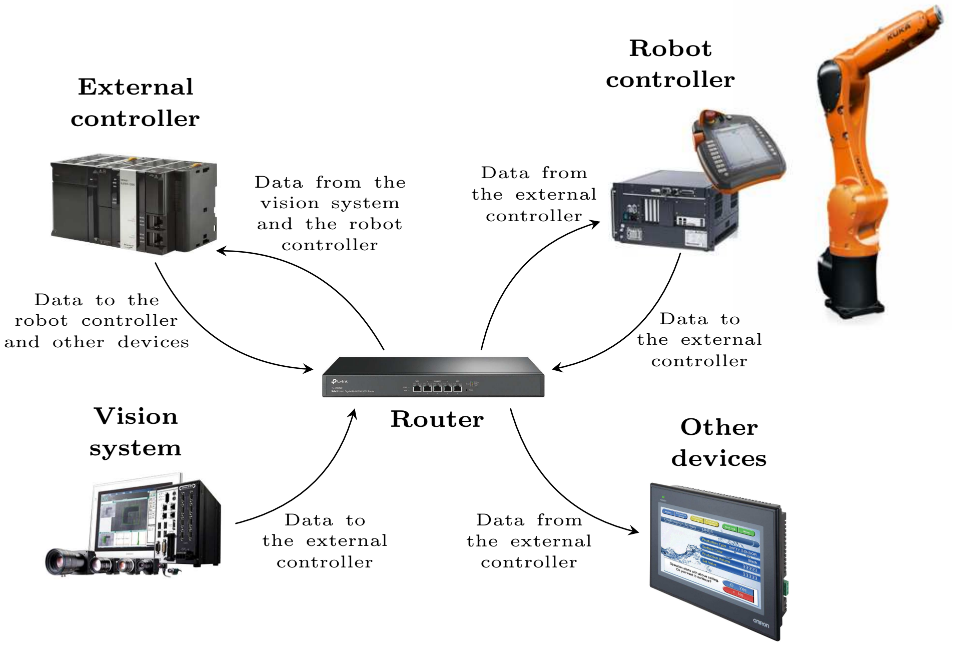

External controller
Robot controller
Data from the vision system and the robot controller
Data from the external controller
Data to the robot controller and other devices
Data to the external controller
Router
Vision system
Other devices
Data to the external controller
Data from the external controller

Figure 1. Industrial robot visual feedback control scheme.

from the environment, coined as visual features vector; an external controller, usually a PLC or PC-based industrial workstation, where the visual control algorithms and other auxiliary algorithms (e.g., communications with the factory servers) are implemented; an industrial robot; and other devices, such as screens or factory displays. All of these elements are connected through an industrial router.

Visual servoing has still several unsolved issues, as is the case of the vision system latency (Yu et al. (2017); Xiao and Chen (2020)), i.e., the total time required to obtain the visual information. Nowadays, the main cause of latency in vision systems is due to image processing algorithms, which are often computationally intensive, limiting the refresh rate of the commands provided to the robot system (Castelli et al. (2017); Zhang et al. (2017)). As a consequence, there is a deterioration of the whole system performance, assuming slow robot motion in most cases (Hashimoto and Noritsugu (1998); Solanes et al. (2011)). Another issue of classical visual servoing approaches is stability, since local stability is achieved but global stability cannot be guaranteed (Chaumette (1998)).

Several approaches can be found in the literature to deal with the above mentioned issues, as discussed below.

Regarding the visual sensor latency issue, a multi rate inter-sample disturbance rejection algorithm was presented in (Fujimoto (2003)) to ensure perfect disturbance rejection at M-inter sample points in the steady state. Moreover, dual rate high order holds were used in (Solanes et al. (2011)) to estimate the set of visual features vectors in order to compensate the vision delay in visual servoing mechanisms. Furthermore, a dual rate adaptive fading Kalman filter algorithm with delay compensation was presented in (Wang et al. (2019)) to compensate the visual information delay and achieve the accurate time sequential coordination of encoder and visual feedback in visual servoing systems. Finally, a dynamic visual tracking control system for robot manipulators was proposed in (Qu et al. (2020)) using the dual rate adaptive fading

Kalman filter.

Regarding the global stability issue, most of the approaches are focused on the choice of the visual features vector. For instance, (Chaumette (2002, 2004) and Tahri et al. (2015)) proposed the use of image moments due to their ability to represent object regions, although the rotational motions around the X- and Y-axes simultaneously with the translational motions along the same axes remains a key issue. Other approaches are focused on introducing image or joint constraints in order to avoid reaching the joints limits (Marchand, Rizzo, and Chaumette (1996); Han and Park (2013)) or the loss of the visual features vector (Muñoz-Benavent et al. (2018); Qiu, Hu, and Liang (2019)). A different approach was presented in (Solanes et al. (2013)), which used the duality of LQR-like controllers (Armesto et al. (2015)) and an Extended Kalman Filter Smoother (Todorov (2005, 2008); Zima et al. (2013)) to develop a novel Reference Filtering Control Strategy (RFCS) able to significantly increase the solution reachability of the classic Image Based Visual Servoing (IBVS) controller. This approach was extended in (Solanes et al. (2016)) using a dual rate Kalman filter for the classic IBVS controller in order to deal with the vision sensor latency problem.

The present work develops a general formulation of the dual rate RFCS to be applied for all existing visual servoing controllers, whether the control task is performed in the 3D Cartesian space or in the 2D image plane. The new Dual Rate Dual Sampling Reference Filtering Control Strategy (DR-DS-RFCS) is based on: a dual rate Kalman filter to estimate the set of features vectors during the vision sensor latency; and an Extended Kalman Filter Smoother (EKFS), improved with a dual sampling approach based on the ratio between the vision and control periods, to generate the smoothed references.

In order to show the applicability and benefits of the proposed approach, simulation and real experiments are conducted for well-known visual servoing controllers in 2D and 3D workspaces using a 6R serial industrial manipulator for positioning tasks (e.g., pick-and-place or sanding tasks). The results obtained in this experimentation clearly show the improvements of the proposed approach in terms of solution reachability, robustness and time domain response.

The paper is organized as follows. Next section introduces some preliminaries and the materials used in the subsequent sections, including the proposed method (DR-DS-RFCS) in Section 2.6. Then, Section 3 presents several simulation results to evaluate the performance of DR-DS-RFCS with respect to its single rate counterpart. The feasibility and robustness of the proposed method is shown in Section 4 using a 6R serial industrial robot, which includes a comparison with several single and dual rate visual servoing controllers. Finally, a discussion about the proposed method is provided in Section 5, whereas some concluding remarks are given in Section 6.

2. Materials and Methods

2.1. *Robot kinematics*

The robot kinematics at time instant τ and its derivative are given by:

$$\mathbf{p}(\tau) = \mathbf{l}(\mathbf{q}(\tau)) \tag{1}$$

$$\dot{\mathbf{p}}(\tau) = \frac{\partial \mathbf{l}(\mathbf{q}(\tau))}{\partial \mathbf{q}(\tau)} \dot{\mathbf{q}}(\tau) = \mathbf{J}_r(\mathbf{q}(\tau))\, \dot{\mathbf{q}}(\tau), \tag{2}$$

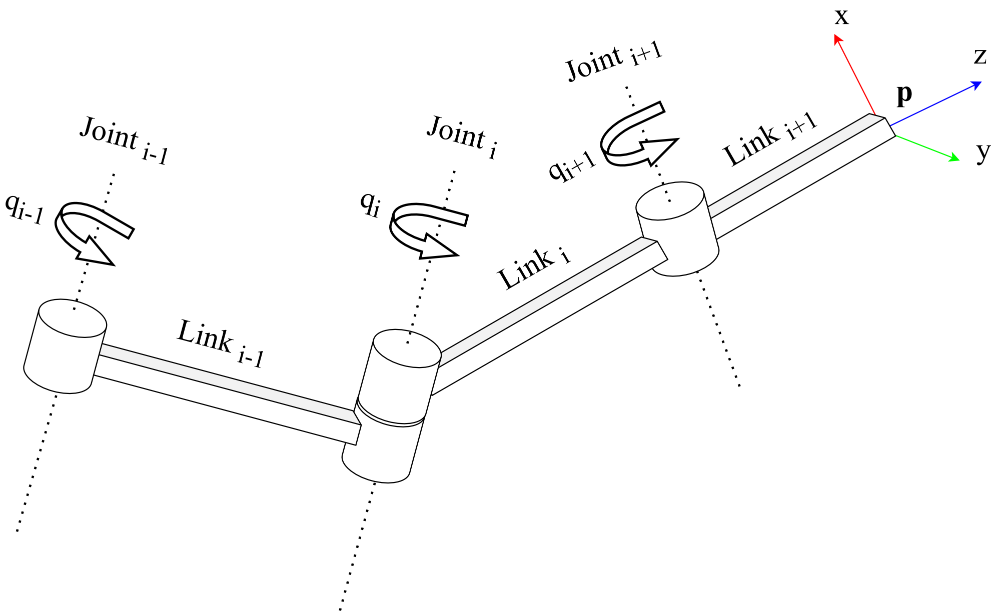


Figure 2. Schematic representation of the joint variables q_i and pose vector $\mathbf{p}$ used in the robot kinematics.

where $\mathbf{p} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^{\mathrm{T}}$ is the robot *pose* (orientation is given by roll α, pitch β and yaw γ angles), vector $\mathbf{q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^{\mathrm{T}}$ is the robot *configuration* (see Fig. 2), n is the number of robot joints, vector $\mathbf{l}$ is the robot kinematic function and matrix $\mathbf{J}_r$ is the robot Jacobian at time instant τ (Chiaverini, Oriolo, and Walker (2008)).

2.2. *Low-level controller*

This study assumes that the low-level controller developed by the robot manufacturer is able to achieve a particular joint velocity from the commanded velocity $\dot{\mathbf{q}}_c(\tau)$ with negligible (fast) dynamics. Notice that this low-level controller should take into account the robot dynamic model.

2.3. *N-periodic systems*

Systems with elements or parts working at different periods are called multi rate systems (Apostolakis (1996)). The N-periodic or dual rate systems are those multi rate systems that are composed of two parts (e.g., visual servoing robotic systems): a part working at a higher period, coined as frame period Δ; and a part working at a lower period, coined as base period δ. The relation between the base and frame periods is $\Delta = N \cdot \delta$.

The Lifting technique (Khargonekar, Poolla, and Tannenbaum (1985), Bamieh et al. (1991)) is commonly used in order to deal with dual rate systems. In this approach, a linear periodic model of the dual rate system is converted into a linear time invariant model (LTI), making it possible to use standard LTI techniques. The input and output system states are expanded in order to have both the same sampling period. Therefore, if the system output is updated N times faster than its input, the generated lifted output vector will be N times larger after the application of the Lifting technique.

The dual rate Kalman filter used in this work is based on the Lifting technique, see (Tornero and Armesto (2003)) for more details.

2.4. *Visual servoing*

The task in visual servoing is to control the pose of the robot end-effector, relative to the target, using visual features extracted from the image (Corke (2011)). There are two basic approaches for visual servoing: Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS). On the one hand, PBVS uses the visual features, a calibrated camera and a known geometric model of the target to determine the pose of the target with respect to the camera. Then, the robot control is performed in the Cartesian space (typically $\mathbb{R}^3$) to move the robot end-effector towards the mentioned pose. On the other hand, IBVS omits the pose estimation step, using directly the visual features, and performs the robot control in the image coordinate space $\mathbb{R}^2$.

This paper proposes a novel control approach that can be applied to both PBVS and IBVS basic approaches, although it can also be applied to other approaches such as the Hybrid Visual Servoing in (Hafez and Jawahar (2007) and Hafez, Cervera, and Jawahar (2008)), or the task control in the operational space in (Khatib (1987)) and (Cai (2017)). For the sake of brevity, this paper presents results only for PBVS and IBVS approaches, using the well-known P and PID controllers.

2.4.1. Image Based Visual Servoing

Without loss of generality, this paper assumes that the camera is rigidly attached to the robot, coined as *eye-in-hand* configuration, see (Chaumette and Hutchinson (2006)) and (Corke (2011)) for more details. In addition, this work considers points of the target object as the visual features. Therefore, considering a point $\mathbf{P}(\tau) = [X \quad Y \quad Z]^T \in \mathbb{R}^3$ in the Cartesian workspace, the visual feature vector is expressed as $\mathbf{f}(\tau) = [x \quad y]^T$, being $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$ the perspective projection for normalized coordinates.

The relation between the camera velocity, $\boldsymbol{v}(\tau) = \begin{bmatrix} v_x(\tau) & v_y(\tau) & v_z(\tau) & \omega_x(\tau) & \omega_y(\tau) & \omega_z(\tau) \end{bmatrix}^T$, and the visual feature movement in the image plane in normalized coordinates, $\dot{\mathbf{f}}(\tau) = [\dot{x}(\tau) \quad \dot{y}(\tau)]^T$, is given by (Chaumette and Hutchinson (2006); Corke (2011)):

$$\begin{aligned}\begin{bmatrix} \dot{x}(\tau) \\ \dot{y}(\tau) \end{bmatrix} &= \begin{bmatrix} -\frac{1}{Z(\tau)} & 0 & \frac{x(\tau)}{Z(\tau)} & x(\tau)\cdot y(\tau) & -(1+x^2(\tau)) & y(\tau) \\ 0 & -\frac{1}{Z(\tau)} & \frac{y(\tau)}{Z(\tau)} & 1+y^2(\tau) & -x(\tau)\cdot y(\tau) & -x(\tau) \end{bmatrix} \cdot \begin{bmatrix} v_x(\tau) \\ v_y(\tau) \\ v_z(\tau) \\ \omega_x(\tau) \\ \omega_y(\tau) \\ \omega_z(\tau) \end{bmatrix} \\ &= \mathbf{L}_f(\tau)\cdot\boldsymbol{v}(\tau), \end{aligned} \tag{3}$$

being $\mathbf{L}_f(\tau)$ the image Jacobian or interaction matrix at time instant τ.

Since the approach proposed in this work needs to update $Z(\tau)$ parameter, the visual feature vector is extended to $\mathbf{f}(\tau) = \begin{bmatrix} x(\tau) & y(\tau) & Z(\tau) \end{bmatrix}^T$. Therefore, the relation between the camera velocity, $\boldsymbol{v}(\tau)$, and the extended visual feature vector velocity,

$\dot{\mathbf{f}}(\tau) = [\dot{x}(\tau) \quad \dot{y}(\tau) \quad \dot{Z}(\tau)]^T$, is given by:

$$\begin{bmatrix} \dot{x}(\tau) \\ \dot{y}(\tau) \\ \dot{Z}(\tau) \end{bmatrix} = \begin{bmatrix} -\frac{1}{Z(\tau)} & 0 & \frac{x(\tau)}{Z(\tau)} & x(\tau) \cdot y(\tau) & -(1+x^2(\tau)) & y(\tau) \\ 0 & -\frac{1}{Z(\tau)} & \frac{y(\tau)}{Z(\tau)} & 1+y^2(\tau) & -x(\tau) \cdot y(\tau) & -x(\tau) \\ 0 & 0 & -1 & -y(\tau) \cdot Z(\tau) & x(\tau) \cdot Z(\tau) & 0 \end{bmatrix} \cdot \begin{bmatrix} v_x(\tau) \\ v_y(\tau) \\ v_z(\tau) \\ \omega_x(\tau) \\ \omega_y(\tau) \\ \omega_z(\tau) \end{bmatrix}$$

$$\dot{\mathbf{f}}(\tau) = \mathbf{L}_f(\tau) \cdot \boldsymbol{v}(\tau). \tag{4}$$

In general, n visual features should be considered to guarantee the convergence of the control (Corke (2011)). Therefore, the visual features vector $\mathbf{s}(\tau)$ and the interaction matrix $\mathbf{L}_s(\tau)$ are defined as:

$$\begin{aligned} \mathbf{s}(\tau) &= \begin{bmatrix} \mathbf{f}_1^T(\tau) & \mathbf{f}_2^T(\tau) & \dots & \mathbf{f}_n^T(\tau) \end{bmatrix}^T \\ \mathbf{L}_s(\tau) &= \begin{bmatrix} \mathbf{L}_{f1}(\tau) & \mathbf{L}_{f1}(\tau) & \dots & \mathbf{L}_{fn}(\tau) \end{bmatrix}^T, \end{aligned} \tag{5}$$

leading to:

$$\dot{\mathbf{s}}(\tau) = \mathbf{L}_s(\tau) \cdot \boldsymbol{v}(\tau). \tag{6}$$

The expression given by (6) indicates the movement of the visual features vector $\mathbf{s}(\tau)$ due to the camera movement in the Cartesian workspace. Therefore, the visual features motion can be controlled by:

$$\boldsymbol{v}(\tau) = \mathbf{L}_s^{\dagger}(\tau) \cdot \dot{\mathbf{s}}(\tau), \tag{7}$$

where the symbol † represents the Moore-Penrose pseudo-inverse (Rakha (2004)). In order to control the visual features velocity, the simplest strategy consists in using the following linear controller $\dot{\mathbf{s}}(\tau) = -\lambda \cdot (\mathbf{s}(\tau) - \mathbf{s}^*(\tau))$, which drives the visual features vector $\mathbf{s}$ towards the desired value $\mathbf{s}^*$ (Hashimoto (1993); Corke (2011))[1]. Therefore, Eq. (7) can be rewritten as:

$$\boldsymbol{v}(\tau) = -\lambda \cdot \mathbf{L}_s^{\dagger}(\tau) \cdot (\mathbf{s}(\tau) - \mathbf{s}^*(\tau)). \tag{8}$$

Assuming that the camera is mounted on the last link of the robot system, being ${}^e\mathbf{T}_c = \begin{bmatrix} {}^e\mathbf{R}_c & {}^e\mathbf{t}_c \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4x4}$ the homogeneous transformation matrix that relates the camera pose with respect to the pose of the last link of the robot (usually the end-effector)[2], the velocity vector of the robot joints is obtained as follows:

$$\begin{aligned} \dot{\mathbf{q}}(\tau) &= -\lambda_{IBVS} \cdot (\mathbf{L}_s(\tau) \cdot \mathbf{V} \cdot \mathbf{J}_r(\tau))^{\dagger}(\mathbf{s}(\tau) - \mathbf{s}^*(\tau)) \\ &= -\lambda_{IBVS} \cdot \mathbf{J}(\tau)^{\dagger}(\mathbf{s}(\tau) - \mathbf{s}^*(\tau)), \end{aligned} \tag{9}$$

[1] In this work, a PID controller is also used, see (Lots et al. (2001)) for further details.

[2] The present work adopts the standard notation of using a leading superscript to denote the frame with respect to which a set of coordinates is referenced, which is also denoted as a subscript

where $\mathbf{J}_r(\tau)$ is the robot Jacobian, $\mathbf{V} = \begin{bmatrix} {}^e\mathbf{R}_c & [{}^e\mathbf{t}_c]^e_\times \mathbf{R}_c \\ \mathbf{0} & {}^e\mathbf{R}_c \end{bmatrix} \in \mathbb{R}^{6\times n}$ is the twist velocity matrix[3] and n is the number of joints of the robot system.

Note that the interaction matrix in (4) uses 3D information, i.e., the visual feature depth Z_i. Therefore, the controller given by (9) is not a pure IBVS. Thus, in order to cancel the effect of the 3D information, an additional block-diagonal matrix $\mathbf{H} = \text{diag}(\phi_1, \ldots, \phi_i, \ldots, \phi_m)$, with $\phi_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and m the number of visual features, is introduced in Eq. (9), yielding the IBVS controller used in this work:

$$\dot{\mathbf{q}}(\tau) = -\lambda_{IBVS} \cdot \mathbf{J}^\dagger(\tau) \cdot \mathbf{H} \cdot (\mathbf{s}(\tau) - \mathbf{s}^*(\tau)). \tag{10}$$

The convergence of the control task is only guaranteed when the actual visual features are close to the reference visual features (local convergence). Therfore, task failure may occur when solving large translations. Moreover, the interaction matrix $\mathbf{L}_s$ is singular for pure rotation movements of 180° around the optical axis (Solanes et al. (2013)).

2.4.2. Position Based Visual Servoing

In Position Based Visual Servoing (PBVS) the pose of the target $\mathcal{F}_o$ with respect to the pose of the camera $\mathcal{F}_c$ is estimated in each time instant τ (Martinet, Gallice, and Khadraoui (1996); Chaumette and Hutchinson (2006); Siciliano and Khatib (2008); Corke (2011)). The control task goal is to determine the motion required to move the camera from its initial pose $\mathcal{F}_c$ to the desired pose $\mathcal{F}_c^*$. Thus, the homogeneous transformation matrix that relates the camera pose in each time instant τ with respect to the desired camera pose ${}^{c^*}\mathbf{T}_c(\tau)$ can be described by its translation vector ${}^{c^*}\mathbf{t}_c(\tau) \in \mathbb{R}^{3x1}$ and rotation matrix ${}^{c^*}\mathbf{R}_c(\tau) \in \mathbb{R}^{3x3}$.

The state $\mathbf{s}(\tau)$ can be defined as $(\mathbf{t}(\tau), \theta(\tau)\mathbf{u}(\tau))$, where $\mathbf{t}(\tau)$ is a translation vector and $\theta(\tau)\mathbf{u}(\tau)$ gives the angle/axis parametrization for the rotation. If $\mathbf{t}(\tau)$ is defined relative to the target frame $\mathcal{F}_o$, then $\mathbf{s}(\tau) = (\mathbf{t}(\tau), \theta(\tau)\mathbf{u}(\tau))$, $\mathbf{s}^*(\tau) = (\mathbf{t}^*(\tau), \mathbf{0})$ and $\mathbf{e}(\tau) = (\mathbf{t}(t) - \mathbf{t}^*(t), \theta(t)\mathbf{u}(t))$. In this case, the interaction matrix $\mathbf{L_e}(t)$ results in:

$$\mathbf{L_e}(\tau) = \begin{bmatrix} {}^{\mathbf{c}^*}\mathbf{R_c}(\tau) & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{\theta\mathbf{u}}(\tau) \end{bmatrix} \in \mathbb{R}^{6x6}, \tag{11}$$

where:

$$\mathbf{L}_{\theta\mathbf{u}}(\tau) = \mathbf{I}_3 - \frac{\theta(\tau)}{2}[\mathbf{u}(\tau)]_\times + \left(1 - \frac{sinc(\theta(\tau))}{sinc^2(\frac{\theta(\tau)}{2})}\right) \cdot [\mathbf{u}(\tau)]^2_\times \in \mathbb{R}^{3x3}, \tag{12}$$

being $\mathbf{I}_3$ the 3×3 identity matrix and $sinc(x)$ the sinus cardinal defined such that $x \cdot sinc(x) = \sin x$ and $sinc(0) = 1$ (Chaumette and Malis (2000)).

Following the developments presented in Section 2.4.1 for IBVS, the robot controller for PBVS results in (Chaumette and Hutchinson (2006)):

$$\dot{\mathbf{q}}(\tau) = -\lambda_{PBVS} \cdot (\mathbf{L}(\tau) \cdot \mathbf{V} \cdot \mathbf{J}_r(\tau))^\dagger \mathbf{e}(\tau) = -\lambda_{PBVS} \cdot \mathbf{J}(\tau)^\dagger \mathbf{e}(\tau). \tag{13}$$

[3] Note that, since the camera is rigidly attached to the last link of the robot system, matrix $\mathbf{V}$ is constant.

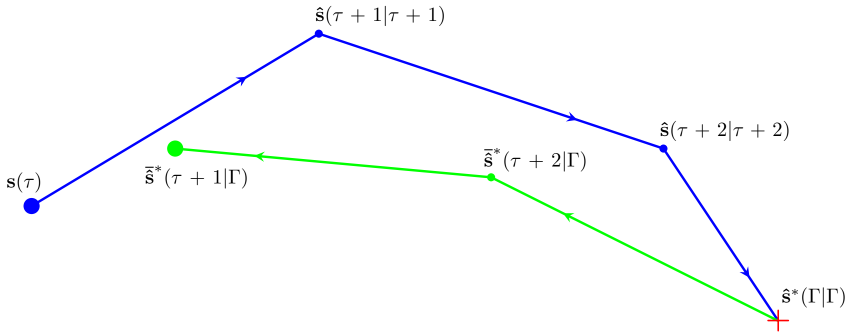

$\hat{\mathbf{s}}(\tau+1|\tau+1)$
$\hat{\mathbf{s}}(\tau+2|\tau+2)$
$\mathbf{s}(\tau)$
$\bar{\hat{\mathbf{s}}}^*(\tau+1|\Gamma)$
$\bar{\hat{\mathbf{s}}}^*(\tau+2|\Gamma)$
$\hat{\mathbf{s}}^*(\Gamma|\Gamma)$

Figure 3. Schematic representation of the RFCS (Solanes et al. (2013)).

Note that, since the control task is carried out in the 3D workspace, the trajectory of the visual features vector $\mathbf{s}(\tau)$ is not known a priory. In fact, the features could leave the camera field of view before reaching the convergence, what would produce the control failure.

2.5. *Reference Filtering Control Strategy (RFCS)*

Authors in (Solanes et al. (2013)) proposed to modify the reference visual features used in classical IBVS to improve the task solution reachability. This strategy, coined as Reference Filtering Control Strategy (RFCS), is depicted in Fig. 3 and is based on the ideas in (Zima et al. (2013)). The actual visual features vector $\mathbf{s}(\tau)$ (starting position) and the reference visual features vector $\mathbf{s}^*(\tau)$ (ending position) are assumed to be known. The first step is to predict the state trajectory from the starting position to the ending position given a prediction horizon $\Gamma = \tau + h$, being h the total number of iterations expressed in the frame period. During the filtering, the system evolves in “open-loop” mode at the prediction step since control inputs are unknown. Thus, the Kalman filter provides a trajectory $\{\hat{\mathbf{s}}(\tau+1|\tau+1), \ldots, \hat{\mathbf{s}}(\Gamma|\Gamma)\}$, see the blue dots and trajectory in Fig. 3, where $h = 2$. Subsequently, a Kalman Smoother (KS) generates a trajectory that takes into account “future” observations under the form $\{\bar{\hat{\mathbf{s}}}^*(\Gamma|\Gamma), \ldots, \bar{\hat{\mathbf{s}}}^*(\tau+1|\Gamma)\}$, see the green dots and trajectory in Fig. 3. At the final step, the algorithm establishes $\bar{\hat{\mathbf{s}}}^*(\tau+1|\Gamma)$ as “reference” visual features vector. Note that the new reference visual features vector is based on the knowledge about the evolution of the system from the starting position to the ending or reference position and, hence, the resulting trajectories represent the predicted behavior.

2.6. *Dual Rate Dual Sampling Reference Filtering Control Strategy method (DR-DS-RFCS)*

2.6.1. Overview of the method

Authors in (Solanes et al. (2016)) proposed the DR-DS-RFCS for the classic IBVS controller. Fig. 4 shows the general case of the DR-DS-RFCS method, where a visual feedback controller (VFC) working at base period δ is used along two components:

- The Dual Rate Kalman Filter component (DR-KF), which provides a set of estimated visual features vectors at base period δ.

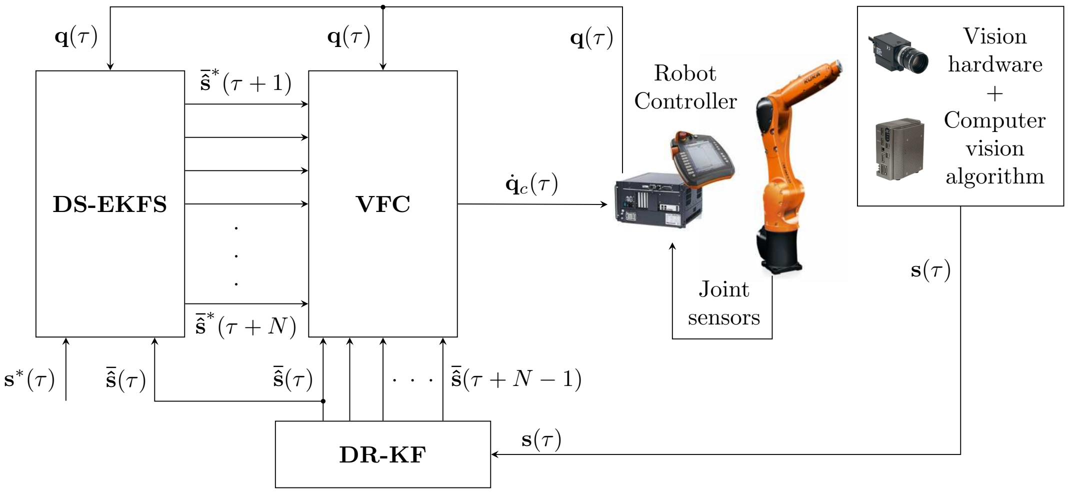

$\mathbf{q}(\tau)$
$\mathbf{q}(\tau)$
$\mathbf{q}(\tau)$
Robot Controller
Vision hardware + Computer vision algorithm
$\bar{\hat{\mathbf{s}}}^*(\tau+1)$
DS-EKFS
VFC
$\dot{\mathbf{q}}_c(\tau)$
$\bar{\hat{\mathbf{s}}}^*(\tau+N)$
Joint sensors
$\mathbf{s}(\tau)$
$\mathbf{s}^*(\tau)$
$\bar{\hat{\mathbf{s}}}(\tau)$
$\bar{\hat{\mathbf{s}}}(\tau)$
$\cdots$ $\bar{\hat{\mathbf{s}}}(\tau+N-1)$
DR-KF
$\mathbf{s}(\tau)$

Figure 4. Overview of the proposed approach.

- The Dual Sampling Extended Kalman Filter Smoother component (DS-EKFS), which provides a set of smoothed filtered reference visual features vectors at base period δ.

The input to the DR-KF is the visual features vector $\mathbf{s}(\tau)$ given by the "*features extraction algorithm*" working at frame period Δ. Note that the dimension and units of the visual features vector will be in accordance to the workspace, 2D or 3D, described in Section 2.4.

The inputs to the DS-EKFS are the first visual features vector $\bar{\hat{s}}(\tau)$ given by the DR-KF, the reference visual features vector $\mathbf{s}^*(\tau)$ and the actual robot state vector $\mathbf{q}(\tau)$.

The inputs to the VFC are the set of estimated visual features vectors $\bar{\hat{s}}(\tau)$ given by the DR-KF, the set of smoothed filtered reference visual features vectors $\bar{\hat{s}}^*(\tau)$ provided by the DS-EKFS and the actual robot state vector $\mathbf{q}(\tau)$. The commanded joint velocity vector $\dot{\mathbf{q}}_c(\tau)$ is computed at base period and serves as the input to the robot controller.

2.6.2. DR-KF component

A Dual Rate Kalman Filter (DR-KF), with the well-known constant velocity model, is used to estimate the position and velocity of the set of visual features vectors $\hat{\mathbf{s}}(\tau)$ (Armesto and Tornero (2006); Steffes (2014); Solanes et al. (2016)). The lifted stochastic model is given by:

$$\begin{bmatrix}\hat{\mathbf{s}}(\tau)\\ \dot{\hat{\mathbf{s}}}(\tau)\end{bmatrix} = \begin{bmatrix}\mathbf{I} & N\delta\mathbf{I}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{s}}(\tau-N)\\ \dot{\hat{\mathbf{s}}}(\tau-N)\end{bmatrix}+\begin{bmatrix}\frac{(2N-1)\delta^2}{2}\mathbf{I} & \frac{(2N-3)\delta^2}{2}\mathbf{I} & \dots & \frac{\delta^2}{2}\mathbf{I}\\ \delta\mathbf{I} & \delta\mathbf{I} & \dots & \delta\mathbf{I}\end{bmatrix}\bar{\mathbf{w}}(\tau) \tag{14}$$

$$\{\bar{\hat{\mathbf{s}}}(\tau),\bar{\hat{\mathbf{s}}}(\tau+1),\dots,\bar{\hat{\mathbf{s}}}(\tau+N-1)\} = \begin{bmatrix}\mathbf{I} & \delta\mathbf{I}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{s}}(\tau)\\ \dot{\hat{\mathbf{s}}}(\tau)\end{bmatrix} + \bar{\mathbf{v}}(\tau), \tag{15}$$

where $\bar{\mathbf{w}}(\tau) = \begin{bmatrix}\mathbf{w}^T(\tau) & \mathbf{w}^T(\tau+1) & \dots & \mathbf{w}^T(\tau+N-1)\end{bmatrix}^T$ and $\bar{\mathbf{v}}(\tau) = \begin{bmatrix}\mathbf{v}^T(\tau) & \mathbf{v}^T(\tau+1) & \dots & \mathbf{v}^T(\tau+N-1)\end{bmatrix}^T$ are the lifted noise matrices, assuming $\mathbf{w} \sim \mathcal{N}(\mathbf{0},\tilde{\mathbf{Q}})$ to be the acceleration noise with covariance matrix $\tilde{\mathbf{Q}}$ and

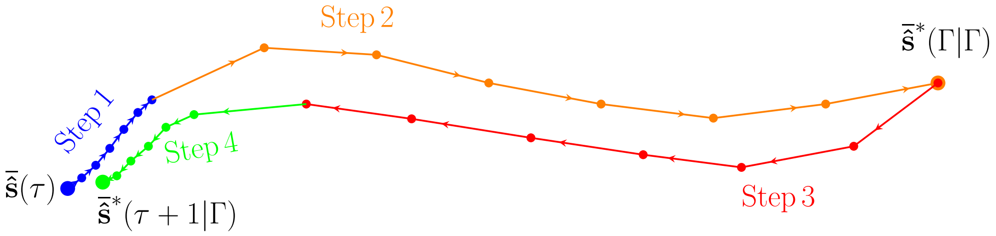

Step 2
$\bar{\hat{\mathbf{s}}}^*(\Gamma|\Gamma)$
Step 1
Step 4
$\bar{\hat{\mathbf{s}}}(\tau)$
$\bar{\hat{\mathbf{s}}}^*(\tau+1|\Gamma)$
Step 3

Figure 5. Dual Sampling Extended Kalman Filter Smoother component (Solanes et al. (2016)) with $N = 7$ and $h = 49$. Fast-EKF step (S1), in blue; Slow-EKF step (S2), in orange; Slow-EKS step (S3), in red; Fast-EKS step (S4), in green.

$\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{R}})$ to be the measurement noise with covariance matrix $\tilde{\mathbf{R}}$, see (Armesto and Tornero (2006)) for further details.

2.6.3. DS-EKFS component

The DS-EKFS approach is represented in Fig. 5 and consists of four steps (Solanes et al. (2016)):

S1) **Fast-EKF step:** in this step, the system evolves in *open-loop* using the well-known stochastic model based on the interaction matrix (Allibert and Courtial (2009); Allibert, Courtial, and Chaumette (2010)) at base period δ during N samples. The control input is assumed to be zero since its future values are unknown. The reference visual features vector $\mathbf{s}^*(\tau)$ is used as "*observation*" input to guarantee the trajectory convergence. In addition, $\mathbf{s}^*(\tau)$ is compared with the predicted state for the update stage of the EKF. Note that covariance matrices $\mathbf{R}$ and $\mathbf{Q}$ of the EKF must be appropriately discretized, see (Armesto and Tornero (2006)) for more details. In sum, the Fast-EKF step predicts the following set of visual features vectors (see (Solanes et al. (2013)) for further details):

$$\{\hat{\mathbf{s}}^*(\tau+1|\tau+1), \hat{\mathbf{s}}^*(\tau+2|\tau+2), \ldots, \hat{\mathbf{s}}^*(\tau+N|\tau+N)\}. \tag{16}$$

S2) **Slow-EKF step:** in this step, the system evolves at frame period until the given prediction horizon T. Thus, the Slow-EKF step provides the following predicted set of visual features vectors:

$$\{\hat{\mathbf{s}}^*(\tau+N|\tau+N), \hat{\mathbf{s}}^*(\tau+2N|\tau+2N), \ldots, \hat{\mathbf{s}}^*(\Gamma|\Gamma)\}. \tag{17}$$

S3) **Slow-EKS step:** in this step, a backward estimation working at frame period $N\delta$ is performed using the predicted set of visual features vectors given by the Slow-EKF step. Note that the smoothing counterpart of the algorithm proposed in (Rauch, Striebel, and Tung (1965)) computes the state covariance during the backward smoothing. However, this state covariance is not necessary in the proposed approached and can be omitted (see (Armesto et al. (2015)) for more details). Therefore, the Slow-EKS step provides the following smoothed set of reference visual features vectors:

$$\{\hat{\mathbf{s}}^*(\Gamma|\Gamma), \hat{\mathbf{s}}^*(\Gamma - N|\Gamma), \ldots, \hat{\mathbf{s}}^*(\tau + N|\Gamma)\}. \tag{18}$$

S4) **Fast-EKS step:** in this step, a backward estimation at base period δ is performed taking into account the predicted set of visual features vectors given by the Slow-EKF step. Therefore, the Fast-EKS step provides the following solution of smoothed predicted set of reference visual features vectors:

$$\{\hat{\mathbf{s}}^*(\tau + N|\Gamma), \hat{\mathbf{s}}^*(\Gamma - N - 1|\Gamma), \ldots, \hat{\mathbf{s}}^*(\tau + 1|\Gamma)\}. \tag{19}$$

2.6.4. Implementation of the controller

The pseudo-code of the proposed algorithm is given below, which is executed at a base period of δ seconds and considers several parameters and functions:

- The initial state covariance prediction $\mathbf{P}_0$ for the DR-KF component is set to zero, since the state $\mathbf{s}(\tau)$ is known.
- The continuous covariances matrices $\mathbf{Q}$ and $\mathbf{R}$ are adjustable parameters and are initialized at the begging of the algorithm.
- Function *GetDiscreteRQmatrices*() returns the discretized $\mathbf{R}_\delta$ and $\mathbf{Q}_\delta$ at base period δ, and $\mathbf{R}_{N\delta}$ and $\mathbf{Q}_{N\delta}$ at frame period $N\delta$, from the continuous covariances matrices $\mathbf{R}$ and $\mathbf{Q}$.
- Function *GetRobotStateFromController*() returns the current robot state $\mathbf{q}(\tau)$ at base period δ.
- Function *GetRobotJacobian*($\mathbf{q}(\tau)$) computes and returns the geometric Jacobian matrix of the robot at base period δ.
- Function *GetFeatureVector*() returns the set of visual features vectors $\mathbf{s}(\tau)$ at frame period $N\delta$. Note that depending on the control task workspace, the dimension and units of $\mathbf{s}(\tau)$ will change.
- Function *VFC*($\dot{\mathbf{q}}(\tau+i), \bar{\hat{\mathbf{s}}}(\tau+i), \bar{\hat{\mathbf{s}}}^*(\tau+i)$) computes the commanded joint velocity vector.
- Function *SendToJointControllers*($\dot{\mathbf{q}}_c(\tau + i)$) commands the desired velocities to the low-level controller.

3. Simulation

The DR-DS-RFCS approach described in Section 2.6 was simulated using the virtual environment depicted in Fig. 6 in order to determine its benefits and drawbacks when dealing with extreme situations, which are difficult or impossible to implement in the real scenario.

The virtual environment was composed of a 6R serial robot manipulator, a camera mounted on the robot end-effector and a square of side 300mm as virtual target, where four coplanar points were used as visual features.

The simulator was run in a PC with Intel Core i7-6700HQ processor at 2.6 GHz clock frequency, 16 GB of RAM and using Xubuntu 16.04 as Operating System.

The following open source libraries were used:

Algorithm 1 Algorithm executed at sampling time of δ seconds

```
$\bar{\mathbf{P}}(\tau-N|\tau-N) \leftarrow \mathbf{P}_0$
$\hat{\mathbf{s}}(\tau-N|\tau-N) \leftarrow \mathbf{0}$
$[\mathbf{R}_\delta, \mathbf{Q}_\delta, \mathbf{R}_{N\delta}, \mathbf{Q}_{N\delta}] \leftarrow$ GetDiscreteRQmatrices($\mathbf{Q}, \mathbf{R}$)
while $\|\mathbf{s}(\tau) - \mathbf{s}^*(\tau)\| <$ error do
    $\mathbf{q}(\tau) \leftarrow$ GetRobotStateFromController()
    $\mathbf{J}(\tau) \leftarrow$ GetRobotJacobian($\mathbf{q}(\tau)$)
    $\mathbf{s}(\tau) \leftarrow$ GetFeatureVector()
    $\bar{\hat{\mathbf{s}}}(\tau) \leftarrow$ DR-KF($\mathbf{s}(\tau), \hat{\dot{\mathbf{s}}}(\tau-N|\tau-N), \tilde{\mathbf{R}}, \tilde{\mathbf{Q}}, \bar{\mathbf{P}}(\tau-N|\tau-N)$)
    $\bar{\hat{\mathbf{s}}}^*(\Gamma|\Gamma) \leftarrow$ DS-EKFS($\bar{\hat{\mathbf{s}}}(\tau), \mathbf{q}(\tau), \mathbf{s}^*(\tau), \mathbf{R}_\delta, \mathbf{Q}_\delta, \mathbf{R}_{N\delta}, \mathbf{Q}_{N\delta}$,N,h)
    for $i \leftarrow 0$ to $N-1$ do
        $\mathbf{q}(\tau+i) \leftarrow$ GetRobotStateFromController()
        $\mathbf{J}(\tau+i) \leftarrow$ GetRobotJacobian($\mathbf{q}(\tau+i)$)
        $\dot{\mathbf{q}}_c(\tau+i) \leftarrow$ VFC($\mathbf{q}(\tau+i), \mathbf{J}(\tau+i), \bar{\hat{\mathbf{s}}}(\tau+i), \bar{\hat{\mathbf{s}}}^*(\tau+i+1|\Gamma)$)
        SendTotJointControllers($\dot{\mathbf{q}}_c(\tau+i)$)
    end for
end while
```

Algorithm 2 DS-EKFS

```
Require: $\mathbf{q}(\tau)$, $\mathbf{J}(\tau)$, $\mathbf{s}^*(\tau)$, $\mathbf{Q}_\delta$, $\mathbf{R}_\delta$, $\mathbf{Q}_{N\delta}$, $\mathbf{R}_{N\delta}$, $N$ and $h$
$\mathbf{P}(\tau|\tau) \leftarrow \mathbf{0}$
for $\nu \leftarrow \tau+1$ to $\tau+N$ do                    // Fast-EKF step (S1)
    $[\hat{\mathbf{s}}^*(\nu|\nu), \mathbf{P}(\nu|\nu)] \leftarrow EKF(\mathbf{s}^*(\tau), \hat{\mathbf{s}}^*(\nu-1|\nu-1), \mathbf{P}(\nu-1|\nu-1), \mathbf{Q}_\delta, \mathbf{R}_\delta)$
end for
for $l \leftarrow 2$ to $h$ do                                // Slow-EKF step (S2)
    $\nu \leftarrow \tau + lN$
    $[\hat{\mathbf{s}}^*(\nu|\nu), \mathbf{P}(\nu|\nu)] \leftarrow EKF(\mathbf{s}^*(\tau), \hat{\mathbf{s}}^*(\nu-N|\nu-N), \mathbf{P}(\nu-N|\nu-N), \mathbf{Q}_{N\delta}, \mathbf{R}_{N\delta})$
end for
for $l \leftarrow h$ to 2 do                                  // Slow-EKS step (S3)
    $\nu \leftarrow \tau + (l-1)N$
    $\hat{\mathbf{s}}^*(\nu|\Gamma) \leftarrow \hat{\mathbf{s}}^*(\nu|\nu) + \mathbf{P}(\nu|\nu) \cdot \mathbf{P}^{-1}(\nu+N|\nu) \cdot (\hat{\mathbf{s}}^*(\nu+N|\Gamma) - \hat{\mathbf{s}}^*(\nu+N|\nu))$
end for
for $\nu \leftarrow \tau+N-1$ to $\tau+1$ do                  // Fast-EKS step (S4)
    $\hat{\mathbf{s}}^*(\nu|\Gamma) \leftarrow \hat{\mathbf{s}}^*(\nu|\nu) + \mathbf{P}(\nu|\nu) \cdot \mathbf{P}^{-1}(\nu+1|\nu) \cdot (\hat{\mathbf{s}}^*(\nu+1|\Gamma) - \hat{\mathbf{s}}^*(\nu+1|\nu))$
end for
return $\bar{\hat{\mathbf{s}}}^*(\tau+1) \equiv \{\hat{\mathbf{s}}^*(\tau+1|\Gamma), ..., \hat{\mathbf{s}}^*(\tau+N|\Gamma)\}$
```

- Visual Servoing Platform library (Marchand, Spindler, and Chaumette (2005)), in order to implement the image visual features extraction algorithm.
- OpenRAVE (Diankov and Kuffner (2008)), in order to generate the virtual environment.
- Orocos Toolchain (Bruyninckx, Soetens, and Koninckx (2003); Soetens and Bruyninckx (2005)), in order to establish the communications between the controller thread and the image processing thread.

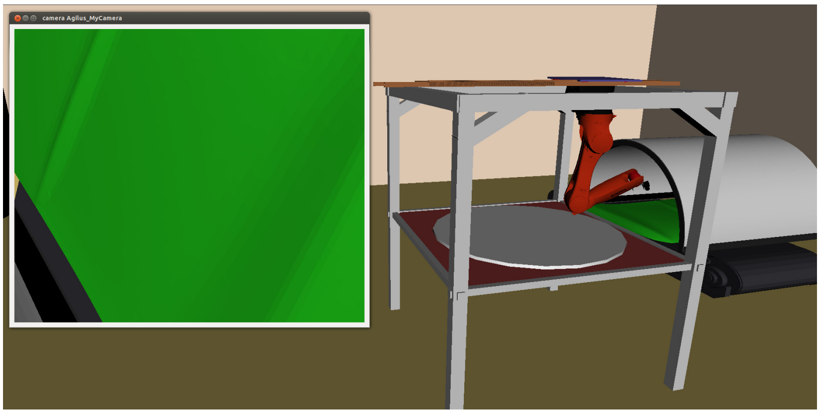


Figure 6. Simulator of the visual servoing system.

3.1. *Sensibility analysis of the covariance matrices*

The Extended Kalman Filter (EKF) is a widely adopted tool in visual servoing to estimate the measured features (Janabi-Sharifi and Marey (2010); Van et al. (2015); Saltus et al. (2020)). It is well-known that the estimation performance of the EKF depends on the noise covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$. Therefore, a sensibility analysis is conducted below to determine the effect of the noise covariance matrices in the Reference Filtering Control Strategy (RFCS) proposed in section 2.5.

In order to determine the benefits of the proposed approach, the RFCS has been compared with its EKF counterpart (Brown et al. (2020)). In particular, the classical image based visual servoing was used with a gain value λ of 0.5. Moreover, the frame period was set to 84ms and the target was placed on the camera optical axis in order to perform pure rotations of the target with respect to the camera.

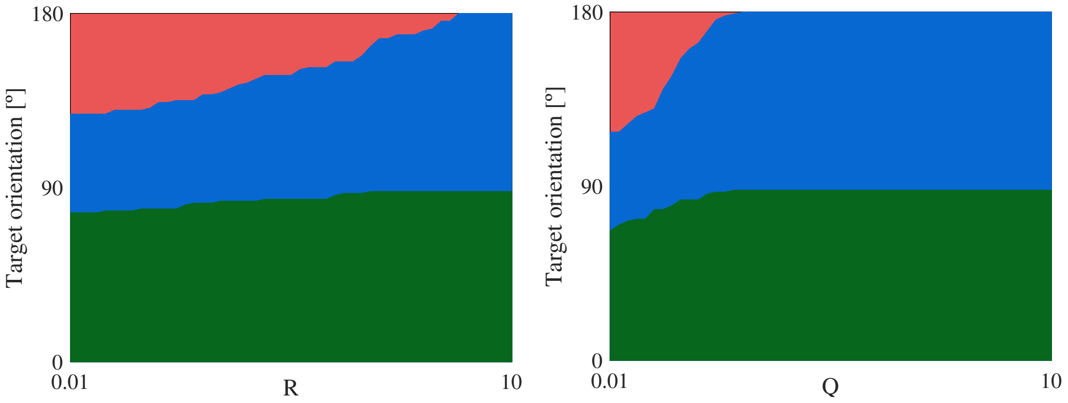
in red; task success reached by DR-DS-RFCS, in blue; task success reached by both SR-SS-RFCS and DR-DS-RFCS, in green.

Fig. 7 shows the results of the experiment: the region of control task success for both the EKF estimator and RFCS approach is depicted in green; the region of control task success for only the RFCS approach is depicted in blue; and the region of control task failure for both the EKF estimator and RFCS approach is depicted in red. On the one hand, Fig. 7(a) shows the results obtained when the noise covariance matrix $\mathbf{Q}$ is equalt to the identity matrix $\mathbf{I}$ and the noise covariance matrix $\mathbf{R}$ varies from $0.01 \cdot \mathbf{I}$ to $10 \cdot \mathbf{I}$. On the other hand, Fig. 7(b) shows the results obtained when the noise covariance matrix $\mathbf{R}$ is equal to $\mathbf{I}$ and the noise covariance matrix $\mathbf{Q}$ varies from $0.01 \cdot \mathbf{I}$ to $10 \cdot \mathbf{I}$. Note that the control task success region is improved with the proposed RFCS regardless of the values used for the noise covariance matrices.

Note also that, as the control task success depends on values used for the noise covariance matrices, the setup of these parameters is important. For this purpose, several approaches (Brown et al. (2020)) can be used to optimally estimate the values of the noise covariance matrices. However, this is out of the scope of this work.

3.2. *Single-rate versus dual-rate single sampling reference filtering control strategies*

In the following, the single-rate single sampling reference filtering control strategy (SR-SS-RFCS) proposed in (Solanes et al. (2013)) and the dual-rate dual sampling



(a) Performance with respect to **R** covariance matrice. (b) Performance with respect to **Q** covariance matrice.

Figure 7. Comparison of the effect of the noise covariance matrices, **R** and **Q** in the RFCS and EKF estimators: task success for both EKF and RFCS, green region; task success for only RFCS, blue region; and task failure for both EKF and RFCS, red region.

reference filtering control strategy (DR-DS-RFCS) are compared in terms of control robustness and task solution reachability. Without loss of generality, classic IBVS and PBVS controllers are used in the VFC component, although other control approaches could be used, such as the 2D 1/2 Visual Servoing in Chaumette and Malis (2000).

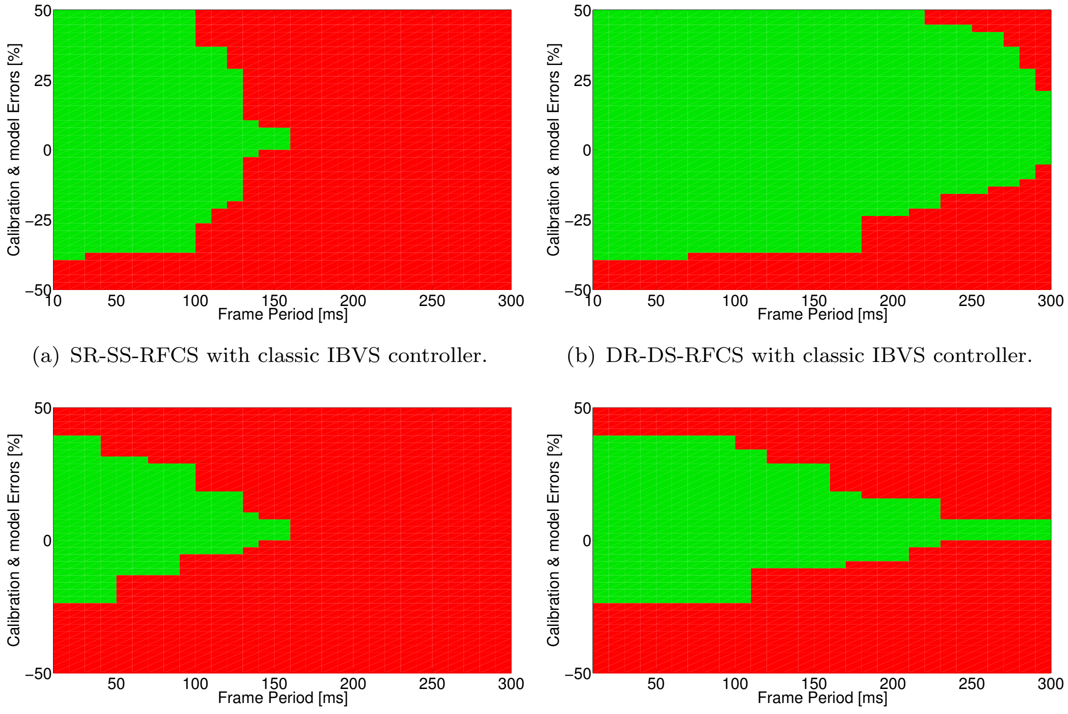
3.2.1. Conditions for the simulation

- The base period δ was set to 12 milliseconds in order to reproduce the periods imposed by the hardware used in the real experimentation detailed in Section 4.
- The total number of iterations h for the prediction horizon T was set to 49.
- Classic IBVS and PBVS controllers described in Section 2.4.1 and Section 2.4.2 were used in the VFC component (see Section 2.6.4).
- The gain parameters λ_{IBVS} in Eq. (10) and λ_{PBVS} in Eq. (13) were empirically tuned to guarantee the control task stability and the fastest task convergence.
- The covariance matrices $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{Q}}$ of the DR-KF were empirically tuned to obtain the best estimation of the image features.
- The covariance matrices $\mathbf{R}$ and $\mathbf{Q}$ of the DS-EKFS were empirically tuned to minimize the convergence time and maximize the task solution reachability.

3.2.2. Robustness analysis

In order to determine the benefits of the proposed approach in terms of control robustness improvement when calibration errors, model errors and large frame periods are present, the target was fixed in a position of the 3D workspace, which ensures the control task success when classic single rate IBVS and PBVS controllers under ideal conditions are used. Calibration and model errors were introduced through $Z(\tau)$, see (Solanes et al. (2013, 2016)) for further details.

Fig. 8 shows the result of this experiment. On the one hand, Fig. 8(a) and Fig. 8(b) show the performance of the SR-SS-RFCS and DR-DS-RFCS, respectively, when the classic IBVS is used in the VFC component. On the other hand, Fig. 8(c) and Fig. 8(d) show the performance of the SR-SS-RFCS and DR-DS-RFCS, respectively, when the classic PBVS is used in the VFC component. Green areas in Fig. 8 represent cases of success in the control task, whilst red areas represent cases of task failure. Note that,



(a) SR-SS-RFCS with classic IBVS controller.

(b) DR-DS-RFCS with classic IBVS controller.

(c) SR-SS-RFCS with classic PBVS controller.

(d) DR-DS-RFCS with classic PBVS controller.

Figure 8. Analysis of control task robustness when calibration errors, model errors and large frame periods are present (results showing the mean of 25 experiments for each case).

Table 1. Starting positions of the target with respect to the camera in the reachability analysis.

	Pose 1-3			Pose 4-6			Pose 7-9		
X(m)	0.1			0.05			0		
Y(m)	0.1	0.05	0	0.1	0.05	0	0.1	0.05	0
Z(m)	-0.115								

in both IBVS and PBVS, the proposed DR-DS-RFCS improved the robustness of its single rate counterpart, specially for large frame periods.

3.2.3. Reachability analysis

This simulation was conducted to determine the benefits of the DR-DS-RFCS in terms of task solution reachability using the classic PBVS controller. The classic PBVS was chosen because, since it performs the control task in the 3D workspace, it has more task reachability issues than classic IBVS. Moreover, the frame period was chosen to be 84 milliseconds (i.e., $N = 7$) to reproduce the real experimentation setup, which is detailed below in Section 4. Furthermore, no calibration or model errors were considered, whereas the target was placed in 9 different starting positions with respect to the camera frame, see Table 1. For each position, the target was rotated around roll α, pitch β and yaw γ angles in the following ranges: $\alpha = [-20, 20]\,^{\circ}$, $\beta = [-20, 20]\,^{\circ}$ and $\gamma = [0, 360]\,^{\circ}$.

Failure in the task solution reachability was considered when one or more visual features leave the camera field of view during the control task. Fig. 9 shows the result of this experiment. Those positions and orientations of the target that could not be properly solved by either of the two controllers are represented in red. The blue color represents those positions and orientations of the target whose solution was only reached by the DR-DS-RFCS. The green color represents all those positions and orientations of the target that were properly solved by both controllers, SR-SS-RFCS and DR-DS-RFCS. Note that this results evidence that the proposed DR-DS-RFCS increases the range of task solution reachability with respect to that obtained with the SR-SS-RFCS approach.

4. Real experimentation

The set-up used for the real experimentation consists of (see Fig. 10): a KUKA KR6 R900 sixx (Agilus) in ceiling-mounted configuration, with integrated low-level joint controller; a 5MP monochrome camera from OMRON Electronics mounted on the robot end-effector; a PC-based Xpectia vision system from OMRON Electronics; a PLC NJ501-1500 controller from OMRON Electronics; and a target that consists of a black square of side 300mm with four white points as visual features.

The controller was implemented in a PLC NJ501-1500 controller from OMRON Electronics. The visual features extraction algorithm was implemented in a PC-based Xpectia vision system from OMRON Electronics and the camera was connected directly using the Camera Link protocol. The robot controller was equipped with the KUKA robot sensor interface, allowing real-time communication between the robot controller and external sensors or controllers via UDP protocol. Thus, Agilus robot, PLC and Xpectia were connected to an industrial router and communicate via UPD protocol. The base period δ was set to 12ms due to robot specification, whereas the frame period $N\delta$ was set to 84ms due to the visual features extraction algorithm.

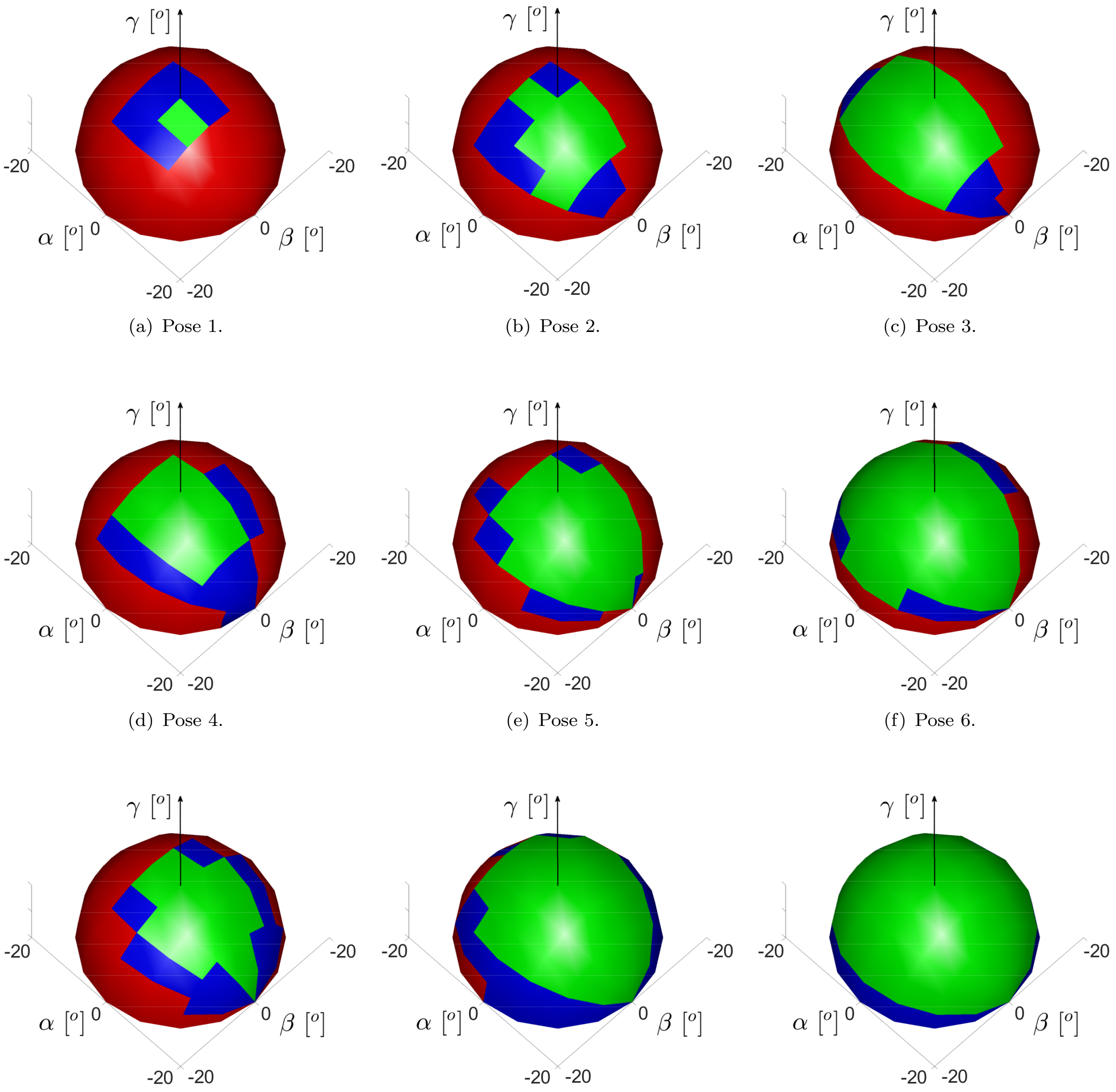
A first experiment was conducted in order to validate the benefits of the DR-DS-RFCS in terms of task solution reachability using the classic PBVS controller. A case of task failure for both the classic PBVS and SR-SS-RFCS was chosen. The PBVS controller given by Eq. (13) was used in the VFC component, being $\lambda_{IBVS} = 4.5$ the optimal value obtained experimentally. The values of the covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ used in the DS-EKFS component (Section 2.6.3) were set to $\mathbf{Q} = 1100 \cdot \mathbf{I}_{6\times 6}$ and $\mathbf{R} = \text{diag}(1, 1, 1, 15, 15, 15)$. Moreover, the total number of iterations h for the prediction horizon T was set to 49.

Fig. 11 shows the results obtained for the first experiment, where it can be seen that the proposed DR-DS-RFCS was able to reach the reference values for the visual features despite that the single rate approaches were not able to.

A second experiment was conducted in order to compare and analyze the performance of single rate, dual rate, SR-SS-RFCS and DR-DS-RFCS approaches. For each approach, classic and PID controllers were tested, both in IBVS and PBVS. The starting pose of the camera, the pose of the target and the reference visual features were the same for all the mentioned cases. The control parameters for each controller were set experimentally to obtain the best performance in each case.

The following indicators are considered for comparison purposes: the rise time; the settling time; the overshoot; the maximum control effort (MCE)[4]; and the Integrating

[4] MCE: $max_t \sqrt{\sum_{i=1}^{m} \left(\frac{\partial q_i}{\partial t}\right)^2}$



(a) Pose 1. (b) Pose 2. (c) Pose 3.

(d) Pose 4. (e) Pose 5. (f) Pose 6.

(g) Pose 7. (h) Pose 8. (i) Pose 9.

Figure 9. Analysis of solution reachability: task failure, in red; task success reached by DR-DS-RFCS, in blue; task success reached by both SR-SS-RFCS and DR-DS-RFCS, in green.

Square Control Effort (ISCE)[5].

Table 2 shows the results of the above indicators for all the mentioned approaches and controllers. Note that the best behavior was obtained by the DR-DS-RFCS with a PID: for the case of the task carried out in the 2D workspace (i.e., IBVS), the rise time was 4.1 seconds, the settling time was 7.2 seconds, the overshoot was zero, the

[5] ISCE: $\int_0^t \left(\sqrt{\sum_{i=1}^m \left(\frac{\partial q_i}{\partial t}\right)^2}[\infty] - \sqrt{\sum_{i=1}^m \left(\frac{\partial q_i}{\partial t}\right)^2}[t] \cdot dt \right)$

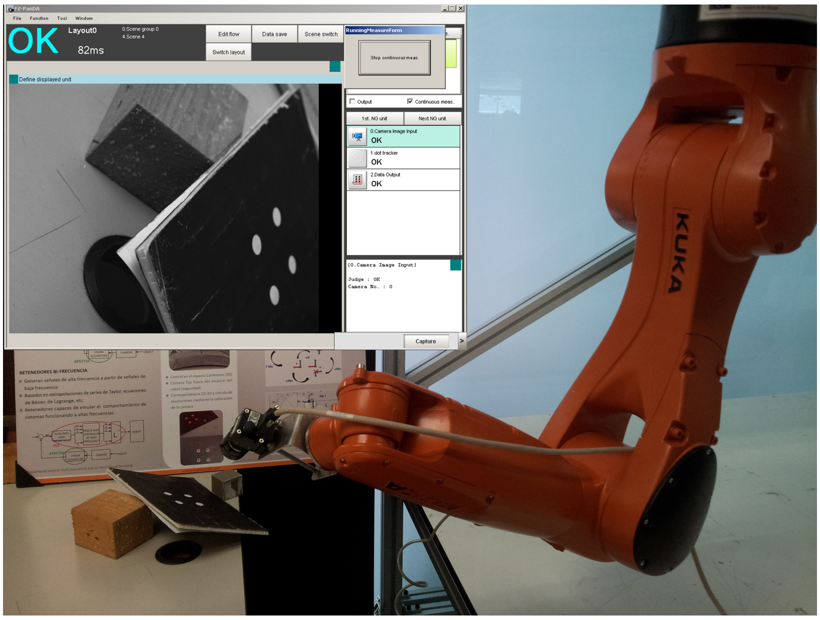


Figure 10. Setup used for the experiments: a 6R industrial robot with a camera mounted on the end-effector and a target that consists of a black square of side 300mm with four white points as visual features.

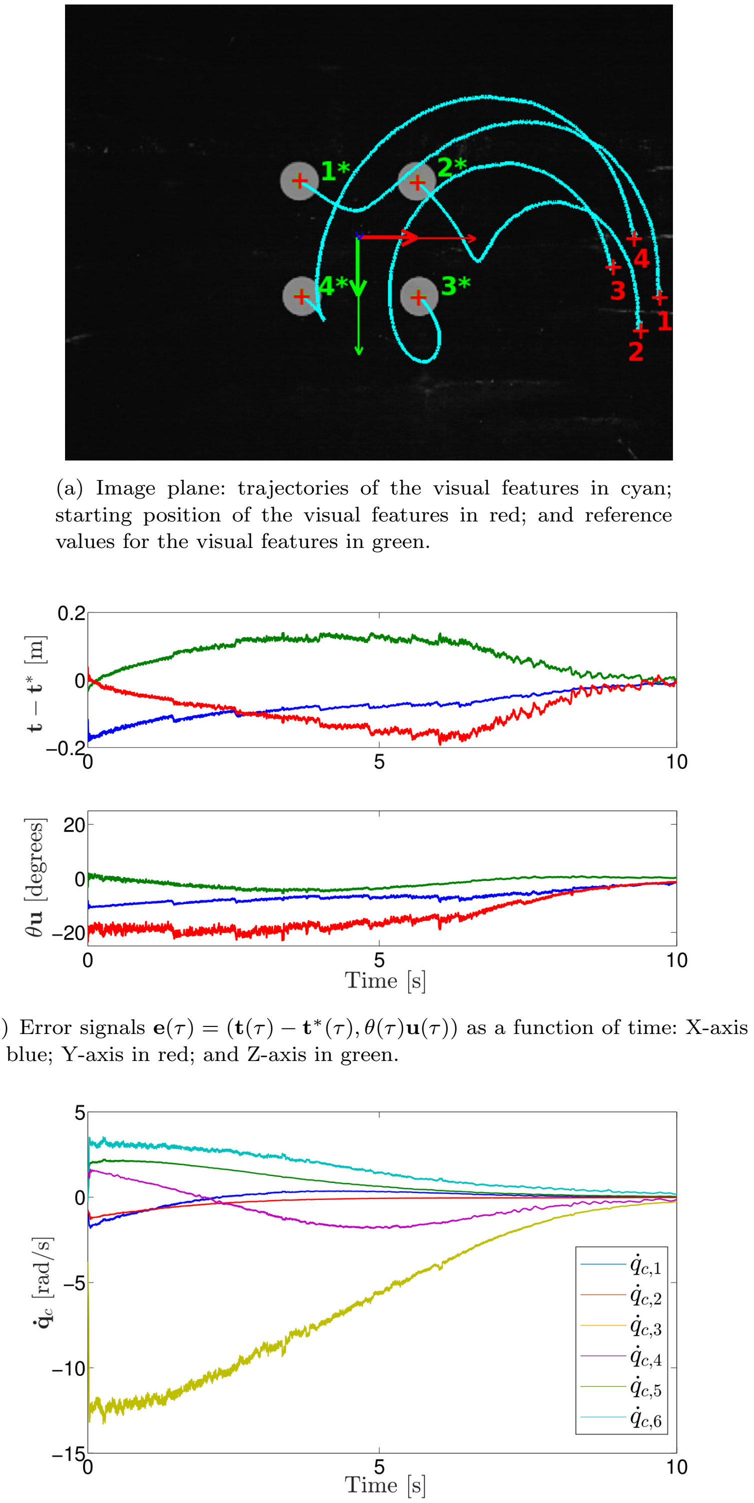
maximum effort was 7.2 degrees/seconds and the ISCE was 13.1 degrees; wereas for the case of the task carried out in the 3D workspace (i.e., PBVS), the rise time was 4.6 seconds, the settling time was 7.9 seconds, the overshoot was the 2%, the maximum effort was 7.8 degrees/seconds and the ISCE was 14.9 degrees. Therefore, it can be concluded that the proposed DR-DS-RFCS improves the results obtained with single rate and classic approaches.

5. Discussion

The proposed DR-DS-RFCS showed a significant improvement in terms of solution reachability with respect to the classic IBVS and PBVS controllers, as demonstrated in the simulation and experimentation above. It also demonstrated better performance than its single rate counterpart, i.e., the SR-SS-RFCS. Nevertheless, the control task solution is not always guaranteed since it depends on the visual servoing controller used in the VFC component and the visual features used.

One possibility to mitigate to some extent this problem consists of adding constraints to the control task. For instance, image constraints would prevent the visual features from leaving the field of view of the camera. These image constraints would be part of the DR-KF and DS-EKFS components. In addition, 3D constraints would prevent the robot from reaching joints limits or forbidden configurations. In this case, the 3D constraints would be part of the DS-EKFS and VFC components.

It should be mentioned that the proposed method is considered to be part of the solution to the problem of automatic detection and repair of defects on painted car body surfaces (Molina et al. (2017); Muñoz et al. (2019); Gracia et al. (2018); Solanes et al. (2018)). The application consists of a network of cameras, 24 in the case of the Mercedes-Benz factory in Vitoria (Spain), used to detect and locate all the defects on the car body surface, among other properties such as type, size, etc. This visual infor-



(a) Image plane: trajectories of the visual features in cyan; starting position of the visual features in red; and reference values for the visual features in green.

(b) Error signals $\mathbf{e}(\tau) = (\mathbf{t}(\tau) - \mathbf{t}^*(\tau), \theta(\tau)\mathbf{u}(\tau))$ as a function of time: X-axis in blue; Y-axis in red; and Z-axis in green.

(c) Velocity commands $\dot{\mathbf{q}}_c$ as a function of time.

Figure 11. Control task performance using the DR-DS-RFCS with PBVS. In this case, both classic PBVS and SR-SS-RFCS with PBVS failed.

Table 2. Comparison of different visual servoing controllers: best performance highlighted in yellow.

APPROACH		CONTROLLER	TIME DOMAIN RESPONSE				
			RISE TIME [s]	SETTLING TIME [s]	OVERSHOOT [%]	MCE [degrees/s]	ISCE [degrees]
IBVS	SR	CLASSIC	6.5	11.8	2.0	11.8	44.6
		PID	6.2	11.1	3.0	11.3	41.6
		RFCS with CLASSIC	5.8	10.2	2.0	10.3	39.0
		RFCS with PID	5.5	10.0	3.0	10.2	37.0
	DR	CLASSIC	5.2	8.9	2.0	9.1	24.8
		PID	4.9	8.1	2.0	8.2	15.2
		RFCS with CLASSIC	4.3	7.6	**0.0**	7.5	13.8
		RFCS with PID	**4.1**	**7.2**	**0.0**	**7.2**	**13.1**
PBVS	SR	CLASSIC	7.6	13.3	5.0	14.1	53.0
		PID	7.1	12.9	3.0	13.7	50.3
		RFCS with CLASSIC	6.5	11.7	2.0	11.4	43.9
		RFCS with PID	5.5	10.7	**2.0**	10.3	38.2
	DR	CLASSIC	5.9	10.3	**2.0**	10.5	39.9
		PID	5.4	9.8	5.0	10.0	36.3
		RFCS with CLASSIC	4.9	8.3	**2.0**	8.3	16.6
		RFCS with PID	**4.6**	**7.9**	**2.0**	**7.8**	**14.9**

mation is intended to be used to command several robot manipulators equipped with sanding tools in order to automatically repair the defects detected by the automatic detection system.

6. Conclusions

A generalization of the Reference Filtering Control Strategy has been developed in this work for 2D and 3D visual servoing of industrial robot manipulators using dual rate and dual sampling estimation techniques. In particular, a Dual Rate Kalman Filter was used to generate inter-sample estimations of the visual features to deal with the problem of vision sensor latency, whereas a Dual Rate Extended Kalman Filter Smoother was used to generate more convenient visual features trajectories in the image plane.

The effectiveness of the proposed method was shown in simulation for several complex case studies. In particular, the simulation results showed a significant improvement in terms of robustness and solution reachability of the control task in both 2D and 3D workspaces.

Furthermore, the feasibility of the proposal was demonstrated with real experimentation using a conventional 6R robot arm, a PLC and an industrial vision system. Moreover, several single rate and dual rate controllers were also implemented for comparison purposes in order to show the benefits of the proposed approach.

As further work, it is proposed to analyze in depth the stability and robustness when other typical issues of visual servoing systems are present, such as command communication delays or loss of visual features during the control task.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported in part by the Spanish Government under the projects PID2020-116585GB-I00 and PID2020-117421RB-C21, and in part by the Generalitat Valenciana under the project GV/2021/181.

References

Allibert, G., and E. Courtial. 2009. “What can prediction bring to Image-Based Visual Servoing?” In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct, 5210–5215.

Allibert, Guillaume, Estelle Courtial, and Francçois Chaumette. 2010. “Visual Servoing via Nonlinear Predictive Control.” In *Visual Servoing via Advanced Numerical Methods*, edited by Graziano Chesi and Koichi Hashimoto, Vol. 401 of *Lecture Notes in Control and Information Sciences*, 375–393. Springer London.

Apostolakis, Ioannis S. 1996. “Algorithmic Techniques in Estimation and Control for Multirate Sampled Digital Control Systems.” In *Stochastic Digital Control System Techniques*, edited by Cornelius T. Leondes, Vol. 76 of *Control and Dynamic Systems*, 1 – 31. Academic Press.

Armesto, L., V. Girbes, A. Sala, M. Zima, and V. Smidl. 2015. “Duality-Based Nonlinear Quadratic Control: Application to Mobile Robot Trajectory-Following.” *Control Systems Technology, IEEE Transactions on* PP (99): 1–1.

Armesto, Leopoldo, and Josep Tornero. 2006. “Linear quadratic gaussian regulators for multirate sampled-data stochastic systems.” In *ICINCO-SPSMC*, edited by Juan Andrade-Cetto, Jean-Louis Ferrier, José Dias Pereira, and Joaquim Filipe, 67–74. INSTICC Press.

Azizian, Mahdi, Mahta Khoshnam, Nima Najmaei, and Rajni V. Patel. 2014. “Visual servoing in medical robotics: a survey. Part I: endoscopic and direct vision imaging – techniques and applications.” *The International Journal of Medical Robotics and Computer Assisted Surgery* 10 (3): 263–274.

Azizian, Mahdi, Nima Najmaei, Mahta Khoshnam, and R. V. Patel. 2015. “Visual servoing in medical robotics: a survey. Part II: tomographic imaging modalities–techniques and applications.” *The international journal of medical robotics + computer assisted surgery : MRCAS* 11 1: 67–79.

Bamieh, B., J.B. Pearson, Francis B.A., and Tannenbaum A. 1991. “A lifting technique for linear periodic systems with applications to sampled-data control.” *Systems and Control Letters* 17: 79–88.

Brown, Jasper, Daobilige Su, He Kong, Salah Sukkarieh, and Eric C. Kerrigan. 2020. “Improved noise covariance estimation in visual servoing using an autocovariance least-squares approach.” *Mechatronics* 68: 102381.

Bruyninckx, H., P. Soetens, and B. Koninckx. 2003. “The real-time motion control core of the Orocos project.” In *Robotics and Automation, 2003. Proceedings. ICRA ’03. IEEE International Conference on*, Vol. 2, Sept, 2766–2771 vol.2.

Cai, Caixia. 2017. “6D Visual Servoing for Industrial Manipulators applied to Human-Robot Interaction Scenarios.” .

Castelli, Francesco, Stefano Michieletto, Stefano Ghidoni, and Enrico Pagello. 2017. “A machine learning-based visual servoing approach for fast robot control in industrial setting.” *International Journal of Advanced Robotic Systems* 14 (6): 1729881417738884.

Chaumette, F. 2002. “A first step toward visual servoing using image moments.” In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, 378–383 vol.1.

Chaumette, F. 2004. “Image moments: a general and useful set of features for visual servoing.” *IEEE Transactions on Robotics* 20 (4): 713–723.

Chaumette, F., and S. Hutchinson. 2006. “Visual servo control. I. Basic approaches.” *Robotics Automation Magazine, IEEE* 13 (4): 82–90.

Chaumette, F., and S. Hutchinson. 2007. “Visual servo control. II. Advanced approaches [Tutorial].” *Robotics Automation Magazine, IEEE* 14 (1): 109–118.

Chaumette, F., and E. Malis. 2000. “2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings.” In *Robotics and Automation, 2000. Proceedings. ICRA ’00. IEEE International Conference on*, Vol. 1, 630 –635 vol.1.

Chaumette, François. 1998. “Potential problems of stability and convergence in image-based and position-based visual servoing.” In *The Confluence of Vision and Control*, 66–78. LNCIS Series, No 237, Springer-Verlag.

Chen, Wei, Xinxin Wang, Hao Liu, Yang Tang, and Junjie Liu. 2019. “Optimized Combination of Spray Painting Trajectory on 3D Entities.” *Electronics* 8 (1): 74.

Chiaverini, S., G. Oriolo, and I.D. Walker. 2008. “Kinematically redundant manipulators.” *Springer Handbook of Robotics* 245–268.

Corke, Peter. 2011. *Robotics, Vision and Control - Fundamental Algorithms in MATLAB®*. Vol. 73 of *Springer Tracts in Advanced Robotics*. Springer.

Diankov, Rosen, and James Kuffner. 2008. *OpenRAVE: A Planning Architecture for Autonomous Robotics*. Technical Report CMU-RI-TR-08-34. Pittsburgh, PA: Robotics Institute.

Fujimoto, H. 2003. “Visual servoing of 6 DOF manipulator by multirate control with depth identification.” In *Decision and Control, 2003. Proceedings. 42nd IEEE Conf. on*, Vol. 5, 5408 – 5413.

Gangloff, Jacques, Florent Nageotte, and Philippe Poignet. 2013. *Vision-based Control*, Chap. 6, 177–232. John Wiley & Sons, Ltd.

Gracia, Luis, J. Ernesto Solanes, , Pau Muñoz-Benavent, Jaime Valls Miro, Carlos Perez-Vidal, and Josep Tornero. 2018. “Adaptive Sliding Mode Control for Robotic Surface Treatment Using Force Feedback.” *Mechatronics* 52: 102 – 118.

Hafez, A. H. A., E. Cervera, and C. V. Jawahar. 2008. “Hybrid Visual Servoing by Boosting IBVS and PBVS.” In *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, April, 1–6.

Hafez, A. H. Abdul, and C.V. Jawahar. 2007. “Visual Servoing by Optimization of a 2D/3D Hybrid Objective Function.” In *Proceedings 2007 IEEE Int. Conference on Robotics and Automation*, apr, 1691–1696. IEEE.

Han, Hyejin, and Jaeheung Park. 2013. “Robot Control near Singularity and Joint Limit Using a Continuous Task Transition Algorithm.” *International Journal of Advanced Robotic Systems* 10 (10): 346.

Hashimoto, K. 1993. *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific series in robotics and automated systems. World Scientific.

Hashimoto, K., and T. Noritsugu. 1998. “Performance and sensitivity in visual servoing.” In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, Vol. 3, May, 2321–2326 vol.3.

Hutchinson, S., G.D. Hager, and P.I. Corke. 1996. “A tutorial on visual servo control.” *Robotics and Automation, IEEE Transactions on* 12 (5): 651–670.

Janabi-Sharifi, F., and M. Marey. 2010. “A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing.” *IEEE Transactions on Robotics* 26 (5): 939–947.

Khargonekar, P.P., K. Poolla, and A. Tannenbaum. 1985. “Robust control of linear time-invariant plants using periodic compensation.” *Automatic Control, IEEE Transactions on* 30 (11): 1088–1096.

Khatib, O. 1987. “A unified approach for motion and force control of robot manipulators: The operational space formulation.” *IEEE Journal on Robotics and Automation* 3 (1): 43–53.

Li, J., Z. Chen, G. Rao, and J. Xu. 2019. “Structured Light-Based Visual Servoing for Robotic Pipe Welding Pose Optimization.” *IEEE Access* 7: 138327–138340.

Lots, J.-F., D.M. Lane, E. Trucco, and F. Chaumette. 2001. “A 2D visual servoing for underwater vehicle station keeping.” In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. 3, 2767–2772 vol.3.

Marchand, E., A. Rizzo, and F. Chaumette. 1996. “Avoiding robot joint limits and kinematic singularities in visual servoing.” In *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 1, 297–301 vol.1.

Marchand, E., F. Spindler, and F. Chaumette. 2005. “ViSP for visual servoing: a generic software platform with a wide class of robot control skills.” *Robotics Automation Magazine, IEEE* 12 (4): 40–52.

Martinet, P., J. Gallice, and D. Khadraoui. 1996. “Vision Based Control Law using 3D Visual Features.” In *Committees, Econometrica*, 497–502.

Molina, Jaime, J. Ernesto Solanes, Laura Arnal, and Josep Tornero. 2017. “On the detection of defects on specular car body surfaces.” *Robotics and Computer-Integrated Manufacturing* 48: 263 – 278.

Muñoz, Adolfo, Xavier Mahiques, J. Ernesto Solanes, Ana Martí, Luis Gracia, and Josep Tornero. 2019. “Mixed reality-based user interface for quality control inspection of car body surfaces.” *Journal of Manufacturing Systems* 53: 75 – 92.

Muñoz-Benavent, Pau, Luis Gracia, J. Ernesto Solanes, Alicia Esparza, and Josep Tornero. 2018. “Robust fulfillment of constraints in robot visual servoing.” *Control Engineering Practice* 71: 79 – 95.

Muñoz-Benavent, Pau, J. Ernesto Solanes, Luis Gracia, and Josep Tornero. 2019. “Robust auto tool change for industrial robots using visual servoing.” *International Journal of Systems Science* 50 (2): 432–449.

Qiu, Z., S. Hu, and X. Liang. 2019. “Model Predictive Control for Uncalibrated and Constrained Image-Based Visual Servoing Without Joint Velocity Measurements.” *IEEE Access* 7: 73540–73554.

Qu, J., F. Zhang, Y. Tang, and Y. Fu. 2020. “Dynamic Visual Tracking for Robot Manipulator Using Adaptive Fading Kalman Filter.” *IEEE Access* 8: 35113–35126.

Rakha, Medhat A. 2004. “On the Moore-Penrose Generalized Inverse Matrix.” *Appl. Math. Comput.* 158 (1): 185–200.

Rauch, H. E., C. T. Striebel, and F. Tung. 1965. “Maximum Likelihood Estimates of Linear Dynamic Systems.” *Journal of the American Institute of Aeronautics and Astronautics* 3 (8): 1445–1450.

Saltus, R., I. Salehi, G. Rotithor, and A. P. Dani. 2020. “Dual Quaternion Visual Servo Control.” In *2020 59th IEEE Conference on Decision and Control (CDC)*, 5956–5961.

Siciliano, Bruno, and Oussama Khatib. 2008. *Springer Handbook of Robotics.* Springer.

Soetens, P., and H. Bruyninckx. 2005. “Realtime Hybrid Task-Based Control for Robots and Machine Tools.” In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, April, 259–264.

Solanes, J. Ernesto, Leopoldo Armesto, Josep Tornero, and Vicent Girbés. 2013. “Improving image-based visual servoing with reference features filtering.” *2013 IEEE International Conference on Robotics and Automation* 3083–3088.

Solanes, J. Ernesto, Luis Gracia, Pau Muñoz-Benavent, Alicia Esparza, Jaime Valls Miro, and Josep Tornero. 2018. “Adaptive robust control and admittance control for contact-driven robotic surface conditioning.” *Robotics and Computer-Integrated Manufacturing* 54: 115 – 132.

Solanes, J. Ernesto, Pau Muñoz-Benavent, Vicent Girbés, Leopoldo Armesto, and Josep Tornero. 2016. “On improving robot image-based visual servoing based on dual-rate reference filtering control strategy.” *Robotica* 34 (12): 2842–2859.

Solanes, J. Ernesto, Josep Tornero, Leopoldo Armesto, and Vicent Girbés. 2011. *Multi-rate Visual Servoing Based on Dual-Rate High Order Holds*, Vol. 6856, Chap. 18, 195–206. Berlin, Heidelberg: Springer Berlin Heidelberg.

Steffes, S. 2014. “Computationally Distributed Real-Time Dual Rate Kalman Filter.” *Journal of Guidance Control Dynamics* 37: 1064–1068.

Tahri, O., A. Y. Tamtsia, Y. Mezouar, and C. Demonceaux. 2015. “Visual Servoing Based on Shifted Moments.” *IEEE Transactions on Robotics* 31 (3): 798–804.

Todorov, E. 2005. “Stochastic Optimal Control and Estimation Methods Adapted to the Noise Characteristics of the Sensorimotor System.” *Neural Computation* 17 (5): 1084–1108.

Todorov, E. 2008. “General duality between optimal control and estimation.” In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, Dec, 4286–4292.

Tornero, J., and Leopoldo Armesto. 2003. “A general formulation for generating multi-rate models.” In *American Control Conference, 2003. Proceedings of the 2003*, Vol. 2, June, 1146–1151.

Van, Mien, Denglu Wu, Shuzi Sam Ge, and Hongliang Ren. 2015. “Condition Monitoring for Image-Based Visual Servoing Using Kalman Filter.” In *Advances in Visual Computing*, edited by George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Ioannis Pavlidis, Rogerio Feris, Tim McGraw, Mark Elendt, Regis Kopper, Eric Ragan, Zhao Ye, and Gunther Weber, Cham, 842–850. Springer International Publishing.

Wang, Ziyue, Shihua Gong, Delong Li, Diyi Zhou, and Huaiqing Lu. 2019. “LED chip accurate positioning control based on visual servo using dual rate adaptive fading Kalman filter.” *ISA Transactions* 87: 163 – 173.

Wu, H., T. T. Andersen, N. A. Andersen, and O. Ravn. 2017. “Application of visual servoing for grasping and placing operation in slaughterhouse.” In *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, April, 457–462.

Xiao, Hui, and Xu Chen. 2020. “Robotic target following with slow and delayed visual feedback.” *International Journal of Intelligent Robotics and Applications* 4 (4): 378–389.

Yu, Xiaowen, Thomas Baker, Yu Zhao, and Masayoshi Tomizuka. 2017. “Visual Servo for

Fast Glass Handling by Industrial Robot with Large Sensor Latency and Low Sampling Rate**This project is supported by Lens Technology, Changsha, China.” *IFAC-PapersOnLine* 50 (1): 4594 – 4601. 20th IFAC World Congress.

Zhang, J., V. Queiroz, Z. Wu, P. Ciou, S. Hsu, S. Tseng, and L. Fu. 2017. “Visual servoing with time-delay compensation for humanoid mobile manipulator.” In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2291–2296.

Zima, M., L. Armesto, V. Girbes, A. Sala, and V. Smidl. 2013. “Extended Rauch-Tung-Striebel controller.” In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec, 2900–2905.