

Document downloaded from:

<http://hdl.handle.net/10251/65956>

This paper must be cited as:

Solanes Galbis, JE.; Muñoz Benavent, P.; Girbés, V.; Armesto Ángel, L.; Tornero Montserrat, J. (2015). On improving robot image-based visual servoing based on dual-rate reference filtering control strategy. *Robotica*. 1-18. doi:10.1017/S0263574715000454.



The final publication is available at

<http://dx.doi.org/10.1017/S0263574715000454>

Copyright Cambridge University Press

Additional Information

On Improving Robot Image Based Visual Servoing Based On Dual-rate Reference Filtering Control Strategy*

J. Ernesto Solanes Pau Muñoz-Benavent Vicent Girbés
Leopoldo Armesto Josep Tornero †

May 13, 2015

Abstract

It is well known that the use of multi-rate control techniques have improved the performance of many systems in general, and robotic systems, in particular. The main contribution of this paper is the generalization of the Reference Filtering control strategy from a dual-rate point of view, improving its inherent properties by overcoming the problem of sensor latency. In the paper, we discuss and analyze the improvements introduced by the novel dual-rate reference filtering control strategy in terms of convergence time, reachability and robustness. More specifically, we discuss the capability to solve positioning tasks, when hardware limitations are present with large sampling rates. In addition, a comparison is made between the single-rate and the proposed dual-rate control strategies to prove the advantages of the latter approach. A complete set-up has been prepared for validation, including a 6DOF industrial manipulator, a smart camera, and embedded hardware used as a high level controller.

1 Introduction

During the last decade, a number of real-time applications with significant hardware limitations have been developed in many sectors. For example, in industry,

*This work was supported by VALi+d Program (Generalitat Valenciana), DIVISAMOS Project (Spanish Ministry, DPI-2009-14744-C03-01), PROMETEO Program (Conselleria d'Educació, Generalitat Valenciana) and SAFEBUS: Ministry of Economy and Competitivity, IPT-2011-1165-370000).

†E-mail addresses: {esolanes,pmunyo,z,vgirbes,larmesto,jtornero}@idf.upv.es

most of the manipulator's controllers are based on standard PC's, in which both memory and computational capabilities are very limited. Other similar examples are found in the military sector, [Associated Press (2013)], where drones with embedded hardware are used, and where requirements are even stronger than in the manufacturing industry.

Due to the extremely large amount of information that can be extracted using vision sensors, many researchers are focusing on improving robotic system performance by closing the control loop using vision feedback, a technique commonly known as visual servoing, [Chaumette and Hutchinson(2007)]. This paper focuses on Image Based Visual Servoing (IBVS), [Corke(2011)], one of the most commonly used visual servoing techniques; this approach is easily extensible to 3D approaches. According to IBVS's literature there are three major issues: 1) global instability in the image-based method, [Hutchinson et al. (1996)]; 2) selection of features, [Chaumette(1998)]; and 3) problems with sensor latency, [Hashimoto et al. (1998)]. This paper addresses the third problem, which results from the vision sensor requiring long time for image processing; this delay leads to dead-time in the feedback control system. Recent advances in vision technology have provided very high-speed image processing sensors (about 1 ms). As a result, several real-time applications have been developed. However, there are associated problems, including limitations on complexity of the image processing task as well as the size of the image.

[Solanes et al. (2013)] presents the Reference Filtering control strategy in order to improve the performance of classic robot image based controllers. The proposed method is a linearized approach, with Gaussian assumptions, that provides future estimation of the state, based on the ideas of [Rauch et al. (1965)]. The main idea is to use measured features as initial estimation and reference features as observation. As a result, and because the features are predicted using a local model based on the interaction matrix and the robot Jacobian, the predictor-smoother is able to provide estimations that take into account robot kinematic constraints. Once the new reference is provided, it is used by the underlying IBVS control, demonstrating better performance in terms of convergence time and reachability than the classic approach.

The main contribution of this paper is the generalization of the work presented in [Solanes et al. (2013)] from a dual-rate point of view, exploiting the dual-rate nature of many visual servoing applications, resulting in the Dual-rate Reference Filtering control strategy.

In addition, the proposed controller has been implemented using industrial and embedded systems with significant hardware limitations to prove that the new dual-rate reference filtering controller performs better (in terms of convergence time, reachability and robustness), not only than the classic single-rate IBVS controller

but even better than the single-rate reference filtering controller. For this purpose, the experimental requirements are addressed using an industrial manipulator (Kuka KR5 sixx R65) with a smart camera (VC6212 nano) in eye-in-hand configuration, controlled by an IGEPv2 board. The entire system is subject to hard real-time requirements. It is important to remark all of these hardware is commonly used in many robotic systems (i.e. manipulators, mobile robots, underwater robots, drones, etc.).

This paper is organized as follows: Section 2 provides the mathematical background of dual-rate control schemes, visual servoing and a summary of the Reference Filtering control strategy. In Section 3, we extend our previous results in [Solanes et al. (2013)] to the dual-rate case. In Section 4.2, we present a real-time implementation, which validates the proposed dual-rate reference filtering controller. Finally, conclusions are provided in Section 5.

2 Mathematical Background

2.1 Dual-rate Systems

In Dual-rate Image Based Visual Servoing (IBVS), image processing performs at a different sampling rate, commonly much lower, than the one required by the controller of the robotic system. This problem arises from the fact that image processing algorithms are much more time-consuming than the sampling periods of the rest of the electrical-mechanical system, for example an industrial manipulator. [Solanes et al. (2011)] shows how classic single-rate IBVS controllers, working at the image processing sampling rate, produce significant reduction in performance, and even system instability.

Many contributions with dual-rate or N -periodic systems (see [Armesto and Tornero(2003)] for an overview of modelling approaches) can be found using a linear system approach. In such systems, there are two different sampling rates commonly known as: the **base period** (i.e.: the fastest sampling time) and the **frame period** (i.e.: the slowest sampling time). One of the most cited and used approaches to deal with N -periodic systems is the Lifting technique, [Khargonekar et al. (1985)], [Bamieh et al. (1991)], which transforms a linear periodic system into a linear time-invariant system (LTI); therefore, standard LTI tools can be used for analysis and control design. This is achieved by expanding input and output signal vectors, grouping together all the signals in the same frame period. This implies that, in the most simplistic case of a controller, where its output is updated N times faster than its input, the output vector size will be N times bigger. The system is accordingly augmented based on the lifted input and output vectors and transformed into a discrete LTI running at the frame period. As

a consequence two different operators are commonly defined to transform a discrete N -periodic signal \mathbf{u}_t into an expanded (lifted) vector: **lifting operator** and **inverse lifting operator**.

$$\{\mathbf{u}_t, \mathbf{u}_{t+1}, \dots, \mathbf{u}_{t+N-1}\} \begin{array}{c} \xrightarrow{\text{lifting}} \\ \xleftarrow{\text{inv. lifting}} \end{array} \left[\begin{array}{c} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \vdots \\ \mathbf{u}(N-1) \end{array} \right]_k \quad (1)$$

with $t = kN$. It is also very common to use a double-index notation, such $\mathbf{u}(k, i) \equiv \mathbf{u}_{kN+i} \equiv \mathbf{u}_{t+i}$, to refer to the i -th element of the lifted vector, with $i \in [0, N-1]$.

Unfortunately, for N -periodic non-linear systems (i.e. a robot equipped with a vision system), there are very few tools that can be used. In [Solanes et al. (2011)] and [Solanes et al. (2012)], the problem was addressed by using dual-rate high order holds, [Armesto and Tornero (2003)], which allowed the extension of classic visual servoing systems to N -periodic equivalent ones for generating dual-rate controllers.

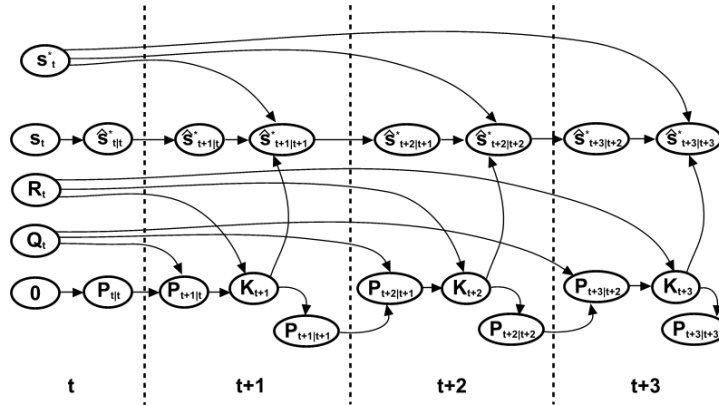
2.2 Image-Based Visual Servoing

Let us assume the **non-free flying camera** case, see [Corke(2011)], in which the camera's motion is constrained by an industrial manipulator, in this paper, a 6 degree of freedom (DOF) manipulator, where in its end-effector, the camera is attached (**eye-in-hand configuration**).

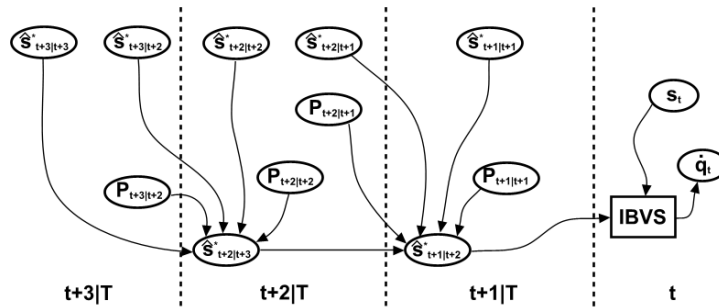
Additionally, let us denote \mathbf{f}_t and \mathbf{f}_t^* as the current and reference state, respectively. $\mathbf{s}_t = \{\mathbf{f}_1, \dots, \mathbf{f}_n\}_t \in \mathbb{R}^{3n}$ is the set of n current states and $\mathbf{s}_t^* = \{\mathbf{f}_1^*, \dots, \mathbf{f}_n^*\}_t \in \mathbb{R}^{3n}$ the set of n reference states. According to the **non-free flying camera** case, we have that $\mathbf{J}(\mathbf{s}_t, \mathbf{q}_t) = \mathbf{L}_s(\mathbf{s}_t) \cdot {}^c\mathbf{V}_r \cdot {}^r\mathbf{J}_r(\mathbf{q}_t) \in \mathbb{R}^{3 \times m}$ is the Jacobian matrix, which relates the measured features velocity with the manipulator velocity, where ${}^r\mathbf{J}_r(\mathbf{q}_t) \in \mathbb{R}^{m \times m}$ is the robot Jacobian matrix, ${}^c\mathbf{V}_r \in \mathbb{R}^{6 \times m}$ is the twist velocity matrix, $\mathbf{L}_s(\mathbf{s}_t) \in \mathbb{R}^{3n \times 6}$ is the image Jacobian (dub interaction matrix) and $\mathbf{q}_t \in \mathbb{R}^m$ and $\dot{\mathbf{q}}_t \in \mathbb{R}^m$ are the robot positions and velocities, respectively, being m the manipulator number of DOF.

Without loss of generality, this paper considers features-like-points, $\mathbf{P}_t = (X_t, Y_t, Z_t)^T \in \mathbb{R}^3$ in the Cartesian workspace, to perform the visual servoing task, being extensible to any other kind of features.

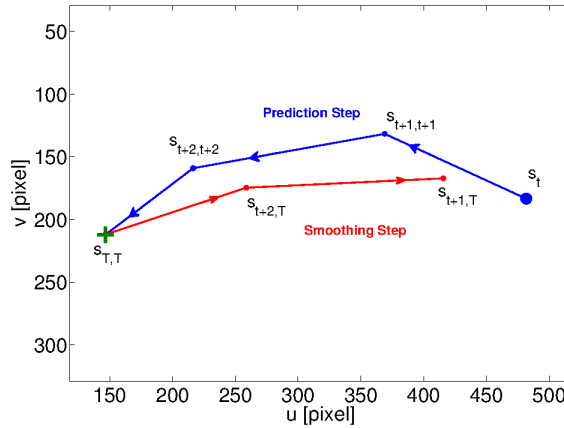
[Chaumette and Hutchinson(2006)] fully explain the advantages and drawbacks with respect to the choice of the depth parameter (Z_t), proving that the behavior



(a) Prediction step using an Extended Kalman Filter.



(b) Smoothing step using a Gaussian Smoother.



(c) Example of trajectory prediction (EKF step) and smoothing (EKS step) of one execution of EKFS algorithm for one of the measured features (big blue dot) and given reference features (green cross). Arrows indicate the direction of the estimation progression for EKF and EKS.

Figure 1: Principle of Reference Filtering control strategy (example with $T=t+3$).

of the control task has better performance if Z_t is updated (by computing or estimating it) at each sampling control period. Correspondingly, the state is extended as $\mathbf{f}_t = (x_t, y_t, Z_t)^T \in \mathbb{R}^3$, with $x_t = X_t/Z_t$ and $y_t = Y_t/Z_t$ the coordinates being expressed in meters in the camera workspace. In consequence, the classic interaction matrix described in [Chaumette and Hutchinson(2006)] is extended as,

$$\mathbf{L}_f(\mathbf{f}_t) = \begin{bmatrix} -\frac{1}{Z_t} & 0 & \frac{x_t}{Z_t} & x_t \cdot y_t & -(1 + x_t^2) & y_t \\ 0 & -\frac{1}{Z_t} & \frac{y_t}{Z_t} & 1 + y_t^2 & -x_t \cdot y_t & -x_t \\ 0 & 0 & -1 & -y_t \cdot Z_t & x_t \cdot Z_t & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 6} \quad (2)$$

One of the most common approaches for IBVS is to generate a proportional control law [Corke(2011)]:

$$\dot{\mathbf{q}}_t = -\lambda \cdot \hat{\mathbf{J}}^+(\mathbf{s}_t, \mathbf{q}_t) \cdot \mathbf{H} \cdot (\mathbf{s}_t - \mathbf{s}_t^*) \quad (3)$$

where λ is a proportional gain, $\hat{\mathbf{J}}^+(\mathbf{s}_t, \mathbf{q}_t)$ is the Jacobian pseudo-inverse matrix of $\mathbf{J}(\mathbf{s}_t, \mathbf{q}_t)$ and $\mathbf{H} \in \mathbb{R}^{3n \times 3n}$ is a block-diagonal matrix defined by $\mathbf{H} = \text{diag}(\phi)$, with $\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. The matrix \mathbf{H} in Eq. 3, is introduced in order to obtain a pure image-based controller which considers the Z parameter of error signal in the control law.

2.3 Image Based Visual Servoing with Reference Filtering

In [Todorov (2008)], a stochastic control index is presented, whose optimal result is dual to an estimation problem. [Zima et al. (2013), Armesto et al. (2014)] demonstrates and solves the implementation for such an estimation problem, which is composed of an estimation/filtering structure, following the result in [Rauch et al. (1965)]. Thus, the authors derive an optimal controller with the appropriate adjustment of the Kalman matrices.

In [Solanes et al. (2013)], those results are used to develop the Reference Filtering control strategy in order to improve the performance of the classic IBVS controller. The proposed predictor-smoother is a linearized approach with Gaussian assumption that provides future estimation over the state based on the ideas of [Rauch et al. (1965)]. However, the main idea in [Solanes et al. (2013)] is to use measured features as initial estimation and reference features as observation. As a result, and because the features are predicted using a local model based on the interaction matrix and the robot Jacobian, the predictor-smoother is able to give estimations taking into account robot kinematic constraints.

According to [Rauch et al. (1965)], two steps are needed to get the new reference features $\hat{\mathbf{s}}_{t+1|T}^*$: the first step of the algorithm is carried out by an Extended Kalman Filter, which provides a sequence of future features

$\{\hat{\mathbf{s}}_{t+1|t+1}^*, \hat{\mathbf{s}}_{t+2|t+2}^*, \dots, \hat{\mathbf{s}}_{T|T}^*\}$ subject to the open-loop model behavior. While the prediction is being carried out, the reference features $\hat{\mathbf{s}}_t^*$ (the final goal of the visual servoing task) are used as observation and the previous estimated features are used as state. In Figure 1(a) an example of this first step is shown for the particular case when $T = 3$, and in Figure 1(c) a representation in the image plane is shown, being the blue dots and line trajectory the one have in this first step. Once the prediction step provides the sequence of estimated features, a Gaussian Smoother is used, due to its well known advantages. Starting with the last state in this sequence $\hat{\mathbf{s}}_{T|T}^*$, a correction of the previous estimated features is made, as is shown in Figure 1(b), and in Figure 1(c) by the red dots and line trajectory.

The result of this step can be seen in Figure 1(c), where red dots and a solid red trajectory belong to the correction carried by the Gaussian Smoother. Once the sequence $\bar{\mathbf{s}}_{d,t+1}^* \equiv \{\hat{\mathbf{s}}_{t+1|T}^*, \hat{\mathbf{s}}_{t+2|T}^*, \dots, \hat{\mathbf{s}}_{T|T}^*\}$ is provided, $\hat{\mathbf{s}}_{t+1,T}^*$ is used as the new reference $\hat{\mathbf{s}}_{d,t+1}^*$ for an underlying IBVS controller.

Classic IBVS control based on the interaction matrix has two well known problems (if dots are taken as features): pure rotations around the optical axis, due to minimal problems with the interaction matrix, and large displacements. [Solanes et al. (2013)] shows, not only through simulation but also trough tests in real scenarios, how our Reference Filtering control strategy is able to overcome these kind of problems, also improving the convergence time with respect to the classic one.

Since several controllers are being used, let us use SR-LF-EKFS-IBVS to refer to the single-rate reference filtering controller working at low frequency (vision sensor frequency), SR-HF-EKFS-IBVS to refer to the single-rate reference filtering controller working at high frequency (controller frequency), DR-EKFS-IBVS to refer to the dual-rate reference filtering controller, SR-HF-IBVS to refer to the classic IBVS controller working at high frequency, and SR-LF-IBVS to refer to the classic IBVS controller working at low frequency.

3 Dual-rate Extended Kalman Filter/Smoother for Visual Servoing

This section describes the algorithmic structure of this novel dual-rate reference filtering controller. This structure can be seen looking at Algorithm 1: first, there is an initialization of the variables required for the dual-rate Kalman Filter (DR-KF) used in line 9; afterwards, it loops until convergence, i.e. until detected features match with reference ones. Line 10 computes filtered feature references with the dual-rate Extended Kalman Filter/Smoother implemented in Algorithm

Algorithm 1 Dual-rate Extended Kalman Filter/Smoothen for Image-Based Visual Servoing (DR-EKFS-IBVS)

```

1:  $\mathcal{I}_{t-N} = \text{GetNewImage}()$ 
2:  $\mathbf{s}_{t-N|t-N} = \text{GetFeatures}(\mathcal{I}_{t-N})$ 
3:  $\hat{\mathbf{s}}_{t-N|t-N} = \mathbf{0}$ 
4:  $\bar{\mathbf{P}}_{t-N|t-N} = \mathbf{P}_0$ 
5:  $\text{Wait}(\delta \cdot N)$ 
   // Loop until convergence
6: while  $\|\mathbf{s}_t - \mathbf{s}_t^*\| < \text{error}$  do
7:    $\mathcal{I}_t = \text{GetNewImage}()$ 
8:    $\mathbf{s}_t = \text{GetFeatures}(\mathcal{I}_t)$ 
9:    $[\bar{\mathbf{s}}_{t|t}, \hat{\mathbf{s}}_{t|t}, \bar{\mathbf{P}}_{t|t}] = \text{DR-KF}(\mathbf{s}_t, \hat{\mathbf{s}}_{t-N|t-N}, \hat{\mathbf{s}}_{t-N|t-N}, \bar{\mathbf{P}}_{t-N|t-N})$ 
10:   $\bar{\mathbf{s}}_{d,t+1}^* = \text{DR-EKFS}(\bar{\mathbf{s}}_{t|t}, \mathbf{q}_t, \mathbf{s}_t^*)$ 
11:   $\bar{\mathbf{q}}_t^* = \text{DR-IBVS}(\mathbf{s}_t, \mathbf{q}_t, \bar{\mathbf{s}}_{t|t}, \hat{\mathbf{s}}_{t|t}, \bar{\mathbf{s}}_{d,t+1}^*)$ 
   // Inverse Lifting
12:  for  $i = 0, \dots, N - 1$  do
13:     $\text{ApplyControl}(\bar{\mathbf{q}}_t^*(i))$ 
14:     $\text{Wait}(\delta)$ 
15:  end for
16: end while

```

3. The dual-rate IBVS control is used in line 11 according to Algorithm 5. The remainder of the algorithm implements the inverse lifting technique through lines 12 to 15. The algorithm uses generic call functions such as “GetNewImage”, “GetFeatures”, “ApplyControl” and “Wait” to complete the remainder of functionalities also required in standard Visual Servoing applications.

Summarizing, our algorithm consists of three clearly differentiated components; therefore, we divide this section into three parts:

- Dual-rate Kalman Filter (section 3.1)
- Dual-rate Extended Kalman Filter/Smoothen (section 3.2)
- Dual-rate Image-Based Visual Servoing (section 3.3)

3.1 Dual-rate Kalman Filter (DR-KF)

A dual-rate Kalman filter is used to estimate the position and velocity of the set of features, \mathbf{s}_t . The inherent model of the dual-rate Kalman filter is a particular

case of the formulation presented in Appendix A. In this sense, we use a periodic stochastic linear model based on the well-known constant velocity model. It is interesting to note that the 3D parameter Z is also filtered since \mathbf{s}_t includes this parameter estimated at frame-period. Accordingly,

$$\begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\dot{\mathbf{s}}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \delta \cdot \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_{t-1} \\ \hat{\dot{\mathbf{s}}}_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\delta^2}{2} \cdot \mathbf{I} \\ \delta \cdot \mathbf{I} \end{bmatrix} \mathbf{w}_{t-1} \quad (4)$$

$$\mathbf{s}_t = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\dot{\mathbf{s}}}_t \end{bmatrix} + \mathbf{v}_t \quad (5)$$

with δ being the sampling time at base-period, $\hat{\mathbf{s}}_t \in \mathfrak{R}^{3n}$ is the estimated feature vector, $\hat{\dot{\mathbf{s}}}_t \in \mathfrak{R}^{3n}$ its velocity, $\mathbf{w}_{t-1} \in \mathfrak{R}^{3n}$ is the acceleration (assumed to be a white noise $\mathbf{w}_{t-1} \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{Q}})$) and $\tilde{\mathbf{Q}} \in \mathfrak{R}^{3n \times 3n}$ a covariance matrix. In addition to this, \mathbf{s}_t is treated here as measurement and $\mathbf{v}_t \in \mathfrak{R}^{3n}$ as measurement noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{R}}) \in \mathfrak{R}^{3n \times 3n}$, with covariance matrix $\tilde{\mathbf{R}}$. To derive the lifted model, we treat the noise \mathbf{w}_{t-1} as input and therefore the full lifted input/output vectors are:

$$\bar{\mathbf{W}}_t = \begin{bmatrix} \mathbf{w}_t^T(0) & \mathbf{w}_t^T(1) & \dots & \mathbf{w}_t^T(N-1) \end{bmatrix}^T \in \mathfrak{R}^{3nN} \quad (6)$$

$$\bar{\mathbf{S}}_t = \begin{bmatrix} \mathbf{s}_t^T(0) & \mathbf{s}_t^T(1) & \dots & \mathbf{s}_t^T(N-1) \end{bmatrix}^T \in \mathfrak{R}^{3nN} \quad (7)$$

$$\bar{\mathbf{V}}_t = \begin{bmatrix} \mathbf{v}_t^T(0) & \mathbf{v}_t^T(1) & \dots & \mathbf{v}_t^T(N-1) \end{bmatrix}^T \in \mathfrak{R}^{3nN} \quad (8)$$

Since features are measured once every frame period δ , the sparse lifted vectors are (see Appendix for details):

$$\bar{\mathbf{w}}_t = \begin{bmatrix} \mathbf{w}_t^T(0) & \mathbf{w}_t^T(1) & \dots & \mathbf{w}_t^T(N-1) \end{bmatrix}^T \in \mathfrak{R}^{3nN} \quad (9)$$

$$\bar{\mathbf{s}}_t = \begin{bmatrix} \mathbf{s}_t^T(0) \end{bmatrix}^T \in \mathfrak{R}^{3n} \quad (10)$$

$$\bar{\mathbf{v}}_t = \begin{bmatrix} \mathbf{v}_t^T(0) \end{bmatrix}^T \in \mathfrak{R}^{3n} \quad (11)$$

which leads to the following lifted stochastic model:

$$\begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\dot{\mathbf{s}}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{I} & N\delta\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_{t-N} \\ \hat{\dot{\mathbf{s}}}_{t-N} \end{bmatrix} + \begin{bmatrix} \frac{(2N-1)\delta^2}{2}\mathbf{I} & \frac{(2N-3)\delta^2}{2}\mathbf{I} & \dots & \frac{\delta^2}{2}\mathbf{I} \\ \delta\mathbf{I} & \delta\mathbf{I} & \dots & \delta\mathbf{I} \end{bmatrix} \bar{\mathbf{w}}_t \quad (12)$$

$$\bar{\mathbf{s}}_t = \begin{bmatrix} \mathbf{I} & \delta\mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\dot{\mathbf{s}}}_t \end{bmatrix} + \bar{\mathbf{v}}_t \quad (13)$$

Algorithm 2 Dual-rate Kalman Filter (DR-KF)

Input: $\mathbf{s}_t, \hat{\mathbf{s}}_{t-N|t-N}, \hat{\mathbf{s}}_{t-N|t-N}$ and $\bar{\mathbf{P}}_{t-N|t-N}$

- 1: $\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{I} & N\delta\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathfrak{R}^{6n \times 6n}$
 - 2: $\bar{\mathbf{B}} = \begin{bmatrix} \frac{(2N-1)\delta^2}{2}\mathbf{I} & \frac{(2N-3)\delta^2}{2}\mathbf{I} & \dots & \frac{\delta^2}{2}\mathbf{I} \\ \delta\mathbf{I} & \delta\mathbf{I} & \dots & \delta\mathbf{I} \end{bmatrix} \in \mathfrak{R}^{6n \times 3nN}$
 - 3: $\bar{\mathbf{C}} = \begin{bmatrix} \mathbf{I} & \delta\mathbf{I} \end{bmatrix} \in \mathfrak{R}^{3n \times 6n}$
 - 4: $\begin{bmatrix} \hat{\mathbf{s}}_{t|t-N} \\ \hat{\mathbf{s}}_{t|t-N} \end{bmatrix} = \bar{\mathbf{A}} \cdot \begin{bmatrix} \hat{\mathbf{s}}_{t-N|t-N} \\ \hat{\mathbf{s}}_{t-N|t-N} \end{bmatrix}$
 - 5: $\bar{\mathbf{P}}_{t|t-N} = \bar{\mathbf{A}} \cdot \bar{\mathbf{P}}_{t|t-N} \cdot \bar{\mathbf{A}}^T + \bar{\mathbf{B}} \cdot \bar{\mathbf{Q}} \cdot \bar{\mathbf{B}}^T$
 - 6: $\mathbf{K}_t = \bar{\mathbf{P}}_{t|t-N} \cdot \bar{\mathbf{C}}^T \cdot \left[\bar{\mathbf{C}} \cdot \bar{\mathbf{P}}_{t|t-N} \cdot \bar{\mathbf{C}}^T + \bar{\mathbf{R}} \right]^{-1}$
 - 7: $\begin{bmatrix} \hat{\mathbf{s}}_{t|t} \\ \hat{\mathbf{s}}_{t|t} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{s}}_{t|t-N} \\ \hat{\mathbf{s}}_{t|t-N} \end{bmatrix} + \mathbf{K}_t \cdot \left(\mathbf{s}_t - \bar{\mathbf{C}} \cdot \begin{bmatrix} \hat{\mathbf{s}}_{t|t-N} \\ \hat{\mathbf{s}}_{t|t-N} \end{bmatrix} \right)$
 - 8: $\bar{\mathbf{P}}_{t|t} = \bar{\mathbf{P}}_{t|t-N} - \mathbf{K}_t \cdot \bar{\mathbf{C}} \cdot \bar{\mathbf{P}}_{t|t-N}$
 - 9: **for** $i = 1, \dots, N$ **do**
 - 10: $\hat{\mathbf{s}}_{t+i|t} = \hat{\mathbf{s}}_{t|t} + i \cdot \delta \cdot \hat{\mathbf{s}}_{t|t}$
 - 11: **end for**
 - 12: **return** $\bar{\mathbf{s}}_{t|t} \equiv \{\hat{\mathbf{s}}_{t|t}, \dots, \hat{\mathbf{s}}_{t+N|t}\}, \hat{\mathbf{s}}_{t|t}$ and $\bar{\mathbf{P}}_{t|t}$
-

Algorithm 2 implements the dual-rate Kalman Filter. Lines 1 to 3 define the lifted matrices, while lines 4 to 8 implement standard Kalman filter equations based on lifted matrices. Line 10 implements a prediction of estimated state for the next N iterations.

3.2 Dual-rate Extended Kalman Filter/Smoothen (DR-EKFS)

The state for EKF and EKS represents a filtered feature reference. Its initialization is made through variables $\bar{\mathbf{s}}_{t|t}, \mathbf{q}_t, \mathbf{s}_t^*$. Since they are known, the initial covariance $\mathbf{P}_{t|t} \in \mathfrak{R}^{3n \times 3n}$ is initialized to a zero covariance matrix. Reference estimation evolves during a given time horizon $T = t + h \cdot N$, where h is the number of iterations expressed in frame period units representing a prediction horizon.

At this point, two approaches have to be considered: 1) applying a full step EKFS working at control period; 2) using two different sampling rates: vision period and control period. In the following, we will prove that using 2), the com-

Algorithm 3 Dual-rate Extended Kalman Filter/Smother (DR-EKFS)

Input: $\mathbf{s}_t, \mathbf{q}_t$ and \mathbf{s}_t^* **Definitions:** $T = t + h \cdot N$

```
// Fast-sampling EKF Step
1.  $\hat{\mathbf{s}}_{t|t}^* = \mathbf{s}_t, \mathbf{P}_{t|t} = \mathbf{0}, \mathbf{C} = \mathbf{I} \in \mathbb{R}^{3n \times 3n}$ 
2. for  $\tau = t + 1, \dots, t + N$  do
3.    $\hat{\mathbf{s}}_{\tau|\tau-1}^* = \hat{\mathbf{s}}_{\tau-1|\tau-1}^*$ 
4.    $\mathbf{P}_{\tau|\tau-1} = \mathbf{P}_{\tau-1|\tau-1} + \mathbf{Q}_\delta$ 
5.    $\mathbf{K}_\tau = \mathbf{P}_{\tau|\tau-1} \cdot \mathbf{C}^T \cdot [\mathbf{C} \cdot \mathbf{P}_{\tau|\tau-1} \cdot \mathbf{C}^T + \mathbf{R}_\delta]^{-1}$ 
6.    $\hat{\mathbf{s}}_{\tau|\tau}^* = \hat{\mathbf{s}}_{\tau|\tau-1}^* + \mathbf{K}_\tau \cdot (\mathbf{s}_t^* - \mathbf{C} \cdot \hat{\mathbf{s}}_{\tau|\tau-1}^*)$ 
7.    $\mathbf{P}_{\tau|\tau} = \mathbf{P}_{\tau|\tau-1} - \mathbf{K}_\tau \cdot \mathbf{C} \cdot \mathbf{P}_{\tau|\tau-1}$ 
8. end for
// Slow-sampling EKF Step
9. for  $l = 2, \dots, h$  do
10.   $\tau = t + lN$ 
11.   $\hat{\mathbf{s}}_{\tau|\tau-N}^* = \hat{\mathbf{s}}_{\tau-N|\tau-N}^*$ 
12.   $\mathbf{P}_{\tau|\tau-N} = \mathbf{P}_{\tau-N|\tau-N} + \mathbf{Q}_{N\delta}$ 
13.   $\mathbf{K}_\tau = \mathbf{P}_{\tau|\tau-N} \cdot \mathbf{C}^T \cdot [\mathbf{C} \cdot \mathbf{P}_{\tau|\tau-N} \cdot \mathbf{C}^T + \mathbf{R}_{N\delta}]^{-1}$ 
14.   $\hat{\mathbf{s}}_{\tau|\tau}^* = \hat{\mathbf{s}}_{\tau|\tau-N}^* + \mathbf{K}_\tau \cdot (\mathbf{s}_t^* - \mathbf{C} \cdot \hat{\mathbf{s}}_{\tau|\tau-N}^*)$ 
15.   $\mathbf{P}_{\tau|\tau} = \mathbf{P}_{\tau|\tau-N} - \mathbf{K}_\tau \cdot \mathbf{C} \cdot \mathbf{P}_{\tau|\tau-N}$ 
16. end for
// Slow-sampling EKS Step
17. for  $l = h, \dots, 2$  do
18.   $\tau = t + (l - 1)N$ 
19.   $\hat{\mathbf{s}}_{\tau|T}^* = \hat{\mathbf{s}}_{\tau|\tau}^* + \mathbf{P}_{\tau|\tau} \cdot \mathbf{P}_{\tau+N|\tau}^{-1} \cdot (\hat{\mathbf{s}}_{\tau+N|T}^* - \hat{\mathbf{s}}_{\tau+N|\tau}^*)$ 
20. end for
// Fast-sampling EKS Step
21. for  $\tau = t + N - 1, \dots, t + 1$  do
22.   $\hat{\mathbf{s}}_{\tau|T}^* = \hat{\mathbf{s}}_{\tau|\tau}^* + \mathbf{P}_{\tau|\tau} \cdot \mathbf{P}_{\tau+1|\tau}^{-1} \cdot (\hat{\mathbf{s}}_{\tau+1|T}^* - \hat{\mathbf{s}}_{\tau+1|\tau}^*)$ 
23. end for
24. return  $\bar{\mathbf{s}}_{d,t+1}^* \equiv \{\hat{\mathbf{s}}_{t+1|T}^*, \dots, \hat{\mathbf{s}}_{t+N|T}^*\}$ 
```

computational time is reduced N times (in the limit case) with slightly higher mean squared error (MSE) compared with 1).

According to 2), in the filtering step of the Algorithm 3, the first N iterations of the fast-sampling EKF evolves at control period (lines 2 to 8), while for the remainder of iterations, a slow-sampling EKF evolves at vision period (lines 9 to

16). During the EKF step, the system evolves in “open-loop” since future control inputs are unknown and thus treated as Gaussian noises, for that reason lines 3 and 11 of Algorithm 3 assume null input. Target features \mathbf{s}_t^* are taken as “observations” to ensure global convergence of the trajectory (lines 6 and 14) and compared with the predicted state to perform the update step. As a summary, the EKF provides a trajectory set:

$$\{\hat{\mathbf{s}}_{t+1|t+1}^*, \hat{\mathbf{s}}_{t+2|t+2}^*, \dots, \hat{\mathbf{s}}_{t+N|t+N}^*, \hat{\mathbf{s}}_{t+2N|t+2N}^*, \dots, \hat{\mathbf{s}}_{T|T}^*\} \quad (14)$$

The Extended Kalman Smoother (EKS step) performs a backward estimation by taking into account “predicted” estimations. Here, we also perform a double-integration step, in which first iterations evolve at frame period (lines 17 to 20 of Algorithm 3), while the last N iterations evolve at base period (21 to 23). Thus, the EKS step generates estimations under the form:

$$\{\hat{\mathbf{s}}_{T|T}^*, \hat{\mathbf{s}}_{T-N|T}^*, \hat{\mathbf{s}}_{T-2N|T}^*, \dots, \hat{\mathbf{s}}_{t+N|T}^*, \dots, \hat{\mathbf{s}}_{t+1|T}^*\} \quad (15)$$

It is interesting to note that the smoothing step in [Rauch et al. (1965)] also computes state covariance during backward smoothing, but in our algorithm, the corresponding equations are not implemented since they are not used. Algorithm 3 returns the set of feature references filtered by the smoother on a lifted vector as defined in Equation (22).

The well known local model based on the interaction matrix [Allibert et al. (2010)] is used to perform the filter-smoother step. This model has to be properly linearized and discretized in order to obtain its equivalent discrete model¹. Thus, depending on the period used, we obtain:

$$\text{(based period) } \hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}_t + \mathbf{J}_\delta(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \dot{\mathbf{q}}_t \quad (16)$$

$$\text{(frame-period) } \hat{\mathbf{s}}_{t+N} = \hat{\mathbf{s}}_t + \mathbf{J}_{N\delta}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \dot{\mathbf{q}}_t \quad (17)$$

with $\mathbf{J}_\delta(\hat{\mathbf{s}}_t, \mathbf{q}_t) = \delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t)$ and $\mathbf{J}_{N\delta}(\hat{\mathbf{s}}_t, \mathbf{q}_t) = N \cdot \delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t)$. Note that \mathbf{q}_t is not considered part of the state so our method is a pure image-based visual servoing approach.

¹From discretization of a non-linear stochastic system $\dot{\mathbf{x}}_t = \mathbf{A}_c(\mathbf{x}_t)\mathbf{x}_t + \mathbf{B}_c(\mathbf{x}_t)(\mathbf{u}_t + \mathbf{w}_t)$ and $\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_t$, where $\mathbf{x}_{t+\delta} = \mathbf{A}_\delta(\mathbf{x}_t)\mathbf{x}_t + \mathbf{B}_\delta(\mathbf{x}_t)(\mathbf{u}_t + \mathbf{w}_{t:t+\delta})$, $\mathbf{y}_{t:t+\delta} = \mathbf{C}\mathbf{x}_t + \mathbf{v}_{t:t+\delta}$, $\mathbf{w}_{t:t+\delta} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\delta)$ and $\mathbf{v}_{t:t+\delta} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_\delta)$, with $\mathbf{Q}_\delta = \int_0^\delta e^{\mathbf{A}_c(\delta-\tau)} \cdot \mathbf{B}_c \cdot \mathbf{Q}_c \cdot \mathbf{B}_c^T \cdot e^{\mathbf{A}_c^T(\delta-\tau)} d\tau$ and $\mathbf{R}_\delta = \mathbf{C} \int_0^\delta \left\{ \left[\int_\tau^\delta e^{\mathbf{A}_c(s-\tau)} \mathbf{B}_c ds \right] \mathbf{Q}_c \left[\int_\tau^\delta \mathbf{B}_c^T e^{\mathbf{A}_c^T(s-\tau)} ds \right] \right\} d\tau \mathbf{C}^T + \mathbf{R}_c \delta$.

Algorithm 4 Extended Kalman Filter/Smoothen (EKFS)

Input: $\mathbf{s}_t, \mathbf{q}_t$ and \mathbf{s}_t^*
// Fast-sampling EKF Step

1. $\hat{\mathbf{s}}_{t|t}^* = \mathbf{s}_t, \mathbf{P}_{t|t} = \mathbf{0}, \mathbf{C} = \mathbf{I} \in \mathbb{R}^{3n \times 3n}$
 2. **for** $\tau = t + 1, \dots, T$ **do**
 3. $\hat{\mathbf{s}}_{\tau|\tau-1}^* = \hat{\mathbf{s}}_{\tau-1|\tau-1}^*$
 4. $\mathbf{P}_{\tau|\tau-1} = \mathbf{P}_{\tau-1|\tau-1} + \mathbf{Q}_\delta$
 5. $\mathbf{K}_\tau = \mathbf{P}_{\tau|\tau-1} \cdot \mathbf{C}^T \cdot [\mathbf{C} \cdot \mathbf{P}_{\tau|\tau-1} \cdot \mathbf{C}^T + \mathbf{R}_\delta]^{-1}$
 6. $\hat{\mathbf{s}}_{\tau|\tau}^* = \hat{\mathbf{s}}_{\tau|\tau-1}^* + \mathbf{K}_\tau \cdot (\mathbf{s}_\tau^* - \mathbf{C} \cdot \hat{\mathbf{s}}_{\tau|\tau-1}^*)$
 7. $\mathbf{P}_{\tau|\tau} = \mathbf{P}_{\tau|\tau-1} - \mathbf{K}_\tau \cdot \mathbf{C} \cdot \mathbf{P}_{\tau|\tau-1}$
 8. **end for**
 - // Fast-sampling EKS Step*
 9. **for** $\tau = T - 1, \dots, t + 1$ **do**
 10. $\hat{\mathbf{s}}_{\tau|T}^* = \hat{\mathbf{s}}_{\tau|\tau}^* + \mathbf{P}_{\tau|\tau} \cdot \mathbf{P}_{\tau+1|\tau}^{-1} \cdot (\hat{\mathbf{s}}_{\tau+1|T}^* - \hat{\mathbf{s}}_{\tau+1|\tau}^*)$
 11. **end for**
 12. **return** $\hat{\mathbf{s}}_{t+1|T}^*$
-

Algorithm 5 Dual-rate Image-Based Visual Servoing (DR-IBVS)

Input: $\mathbf{q}_t, \{\hat{\mathbf{s}}_{t|t}, \hat{\mathbf{s}}_{t+1|t}, \dots, \hat{\mathbf{s}}_{t+N|t}\}$ and $\{\hat{\mathbf{s}}_{t+1|T}^*, \dots, \hat{\mathbf{s}}_{t+N|T}^*\}$

1. **for** $i = 0, \dots, N - 1$ **do**
 2. $\dot{\mathbf{q}}_{t+i}^* = -\lambda \cdot \hat{\mathbf{J}}^+(\hat{\mathbf{s}}_{t+i|t}, \mathbf{q}_t) \cdot (\hat{\mathbf{s}}_{t+i+1|t} - \hat{\mathbf{s}}_{t+i+1|T}^*)$
 3. **end for**
 4. **return** $\{\dot{\mathbf{q}}_t^*, \dots, \dot{\mathbf{q}}_{t+N-1}^*\}$
-

Similarly, covariances of the EKF must be appropriately computed:

$$\mathbf{Q}_\delta = \delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) \quad (18)$$

$$\mathbf{Q}_{N\delta} = N\delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) \quad (19)$$

$$\mathbf{R}_\delta = \frac{\delta^3}{3} \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) + \mathbf{R}_c \delta \quad (20)$$

$$\mathbf{R}_{N\delta} = \frac{(N\delta)^3}{3} \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) + \mathbf{R}_c N\delta \quad (21)$$

3.3 Dual-rate Image Based Visual Servoing (DR-IBVS)

In order to generate a set of N control inputs, our approach is to implement a proportional control law based on interaction matrix as if it were sampled at control

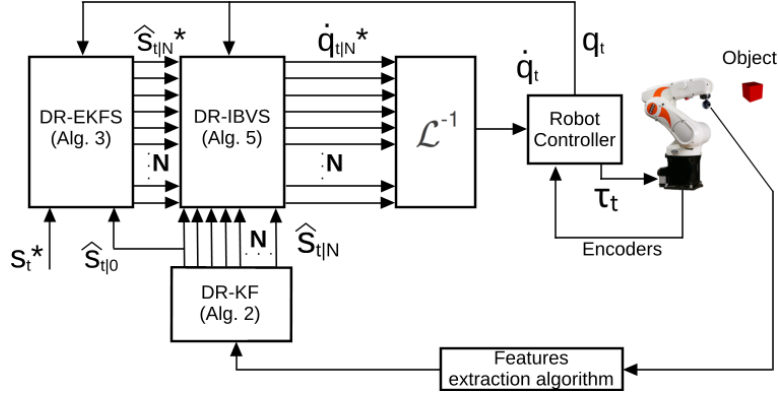


Figure 2: DR-EKFS-IBVS control diagram.

period (see line 2 of Algorithm 5) taking the form:

$$\bar{\mathbf{q}}_t^* \equiv \{\dot{\mathbf{q}}_t^*, \dot{\mathbf{q}}_{t+1}^*, \dots, \dot{\mathbf{q}}_{t+N-1}^*\} \longrightarrow \begin{bmatrix} \dot{\mathbf{q}}_t^*(0) \\ \dot{\mathbf{q}}_t^*(1) \\ \vdots \\ \dot{\mathbf{q}}_t^*(N-1) \end{bmatrix} \in \mathfrak{R}^{3nN} \quad (22)$$

Applying the inverse Lifting operator (\mathcal{L}^{-1}) to the output of Algorithm 5 the corresponding control actions at base period are provided. Summarizing, Figure 2 shows the control block diagram of the new DR-EKFS-IBVS control strategy, where DR-KF block implements Algorithm 2, DR-EKFS block implements Algorithm 3 and DR-IBVS algorithms block implements Algorithm 5; \mathcal{L}^{-1} block implements the inverse Lifting operator.

4 Results

A series of tests have been performed in order to compare the DR-EKFS-IBVS control structure with respect to its single-rate counterpart and the classic IBVS controller. For that purpose, a 6 DOF KUKA industrial manipulator (KUKA KR5 sixx R650), a camera on board and a fixed target have been used. Therefore, the base-period δ will be $10ms$, a time given by the manipulator's communication limitation. The starting position of the robot allows the camera to frame a wide area of the workspace.

The implementation has been carried out using the C++ programming language, with different open source libraries such as ViSP, [Marchand et al. (2005)],

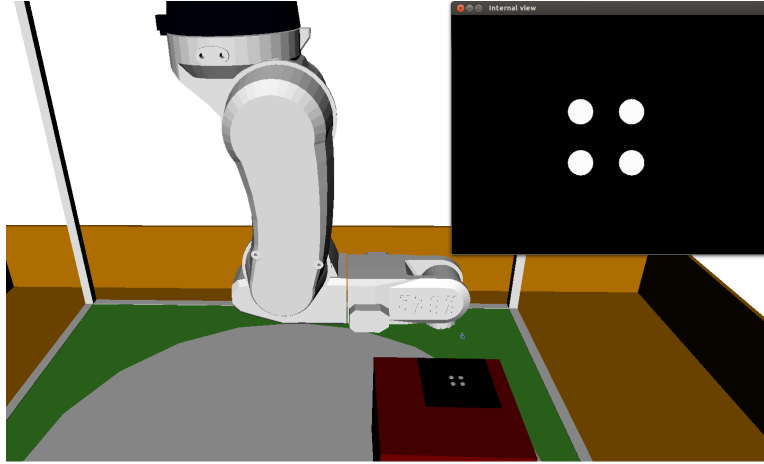


Figure 3: KUKA KR5 sixx R650 Simulator.

OpenRAVE, [Rosen Diankov and James Kuffner (2008)] or Orocos Toolchain, [Bruyninckx et al. (2003), Soetens et al. (2005)], among others.

In the following, it will be assumed that, for all tests presented, the Kalman matrices, Q and R , have been tuned up to their optimal values, which depend on the task to be solved (i.e. large translation positioning task and pure rotation positioning task).

4.1 Simulation Results

The simulation environment used in this paper is shown in Figure 3. The robot has been kinematically and dynamically modeled using the OpenRAVE and ViSP libraries. In this case, four coplanar points, forming a square of 300 mm on a side, have been used as the target. This environment will allow us to show the better performance we get by using the DR-EKFS-IBVS control strategy in critical situations.

The comparison between the DR-EKFS-IBVS, SR-HF-EKFS-IBVS and SR-LF-EKFS-IBVS controllers is carried out throughout a positioning tasks, which is a 150° pure rotation error around the camera optical axis.

Figure 4 shows the reachability and the convergence time performed by using SR-LF-IBVS, Figure 4(a) and DR-EKFS-IBVS, Figure 4(b) controllers, and varying the controller gain parameter λ and the camera's frame rate. In those sub-figures, the zone colored in red indicates that the algorithm has failed to solve the task, while degraded green shows its convergence time. The test shows that the DR-EKFS-IBVS controller is not only more robust against the delay introduced

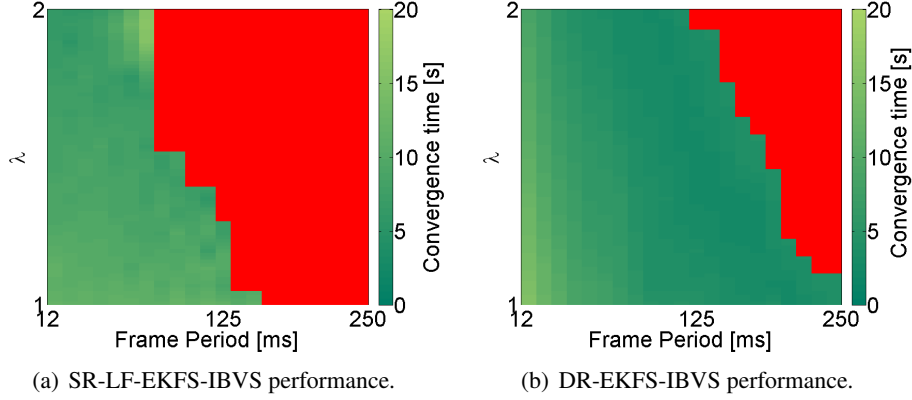


Figure 4: Task reachability and convergence time in function of the controller gain and the frame period used.

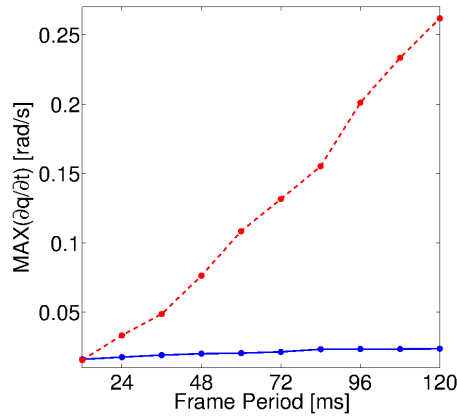


Figure 5: Maximum control action ($\dot{\mathbf{q}}_t \equiv \partial \mathbf{q}_t / \partial t$) applied: SR-LF-EKFS-IBVS (dashed red line) and DR-EKFS-IBVS (solid blue line).

by the camera's frame rate, but also converges faster than its equivalent single-rate one working at frame rate.

Contrary to single-rate controllers, whose control actions are applied once each frame period, dual-rate controllers based on Lifting technique apply N control actions each frame period. Thus, the maximum control action applied by DR-EKFS-IBVS is smaller than the one applied by SR-LF-EKFS-IBVS as can be seen in Figure 5, in which the controller gain parameter λ is tuned in order to get the same convergence time in both cases. The figure confirms that property, and allows

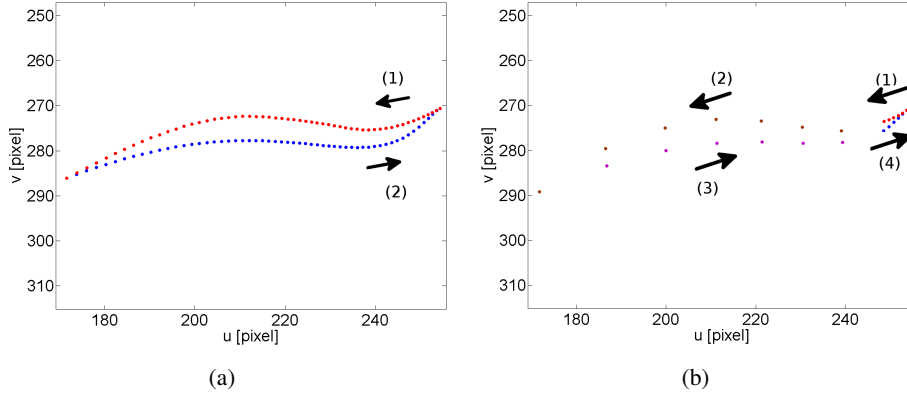
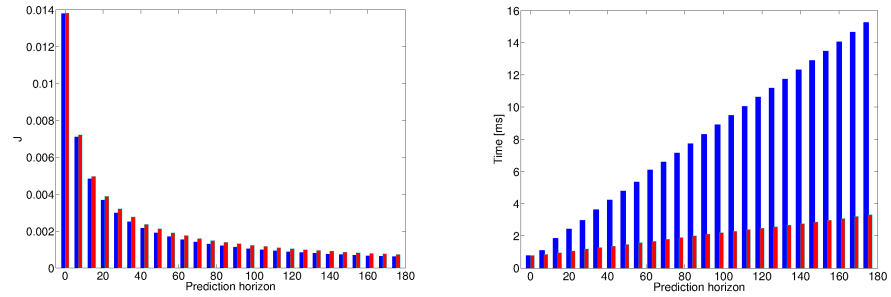


Figure 6: Comparison of the trajectories estimated by both (high frequency single-rate) EKFS and DR-EKFS. The case shown corresponds to a parameter setting of $N = 8$, being 56 the prediction horizon : high frequency prediction step (1), red dots; low frequency prediction step (2), brown dots; low frequency smoothing step (3), magenta dots; high frequency smoothing step (4), blue dots.



(a) Mean Squared Error of the estimated state and the reference state w.r.t the prediction horizon. (b) Computational cost of the algorithms w.r.t the prediction horizon.

Figure 7: Algorithm computational cost and state estimation error between (high frequency single-rate) HF-EKFS (blue bars) and DR-EKFS (red bars).

us to tune higher λ values to perform the task faster. The DR-EKFS-IBVS control strategy described by Algorithm 3 can be considered to be a discretization of the SR-HF-IBVS controller described by Algorithm 4. This is illustrated in Figure 6, where is shown the prediction and the smoothing trajectories performed in one algorithm step by using the SR-HF-IBVS (Figure 6(a)) and the DR-EKFS-IBVS (Figure 6(b)) control strategies. In addition, Figure 7

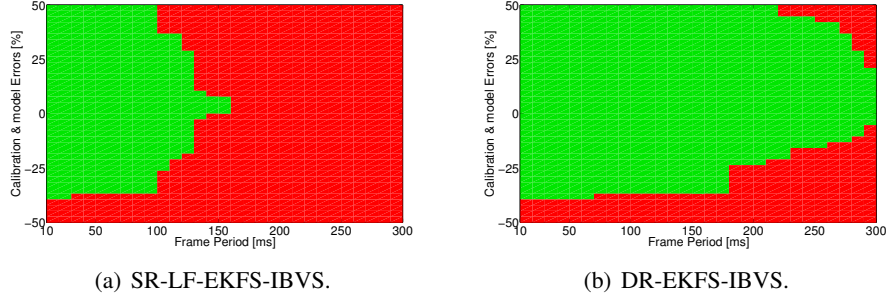


Figure 8: Analysis of the reference filtering control strategy robustness against calibration and model errors: the pictures represents the worst case solution of 25 experiments performed.

shows a more detailed comparison between the two algorithms. Concretely, Figure 7(a) illustrates the Mean Square Error (MSE) defined as the error between $\hat{\mathbf{s}}_{t+1|T}^*$ and \mathbf{s}_t^* , where, despite the higher sampling period, very little degradation can be seen. On the other hand, Figure 7(b), which shows the computational cost of both algorithms, indicates that the computational cost of Algorithm 4 is significantly reduced by an approximate ratio of 1 to N by using Algorithm 3. It is convenient to remark that the prediction horizon can be higher than the number of inter-sampling samples needed because better estimations of the algorithm are obtained, as it is shown in Figure 7(a). Thus, even though only the last N estimations of the algorithm are used, the rest plays an important role in both robustness and reachability. In this sense, we have to reach to an agreement between estimations accuracy and computational cost.

It is well known that image based control laws are very robust with respect to calibration and model errors [Chaumette and Hutchinson(2006)]. However, this is partly because depth parameter Z is usually taken constant and equal to the desired value Z^* . In spite of this, in [Chaumette(1998)] is demonstrated how the optimal performance of such control laws is achieved when Z is updated each sampling period. Due to that parameter Z has to be estimated from vision algorithms, it highly depends on the camera calibration accuracy. Most of the algorithms use for models of the object to be detected as a form to estimate such a depth. Thus, object model errors also affect to the estimation of this parameter.

Owing the difficulty to analytically analyze the robustness of multi-rate control structures (even more difficult when highly nonlinear systems, such as the one presented in this paper, are within the control loop), this analysis is carried out empirically throughout experimentation. In this case, the system has to perform a pure rotation around the optical axis positioning task of 90° . The control period

is 10ms for the dual-rate approach, remaining the same as the frame period in the case of the single-rate one. The controller characteristic parameters, such as the control gain λ , Q and R , as well as the dynamic model matrices, are modified to the needed equivalent ones for each frame period (see Appendix A and Section 3 for more details). Taking into account the comments of the previous paragraph, the depth parameter Z is modified to prove the robustness of the new dual-rate control structure with respect to its single-rate counterpart. Its optimal value in this experiment is 0.38m.

Results in Figure 8 demonstrate that the dual-rate control strategy presented in this paper is not only able to reach out the solution when high frame periods are required by the vision system but also is much more robust against camera calibration and object model errors than its single-rate counterpart. It is interesting to remark that errors which makes lower Z value affect more to the controller robustness than vice versa due to that the model is based on the image Jacobean [Allibert et al. (2010)]. This could be different if other features are chosen, as it is well known by researchers in this field. Despite this, this experiment proves that, under the same conditions, the dual-rate control strategy proposed in this paper is more robust than the single-rate one.

4.2 Experimental Results

The set-up includes a 6 DOF industrial manipulator (KUKA KR5 sixx R650). A Kuka RTI-Ethernet board is used to control the manipulator by using external hardware. This type of industrial manipulators have a watchdog security system, which means that the manipulator’s controller is waiting for a new command (a value that indicates that a new data is received). In our particular manipulator, this period is $\delta = 10\text{ms}$. If several consecutive periods are lost, then an error is occurred, the program flow is interrupted, and the manipulator is shut down.

The control is carried out by using an IGEPv2 board, with embedded hardware based on a DaVinci OMAP3 processor ARM Cortex-A8 at 1GHz. It is running on a Real-Time Operating System (RTOS) under RTLinux with Linaro 10.11 distribution. All the control algorithms described in this paper have been implemented on this system.

An industrial smart camera VC6212 nano handles the task of recording and processing images, which transmits the set of detected features to the IGEPv2 board via TCP/IP. This camera uses a 700 Mhz C64+ DSP processor which supports the VCLIB 3.0 library, which includes standard and well known steps such as image binarization, object labeling through segment run length code (RLC) and computation of object features in the labeled RLC (blob center). This camera acquires images at a maximum rate of 80ms (which means $N = 8$). Processing

has been implemented so that features are detected within the acquisition time and therefore no additional latency is introduced. In this paper the features tracked are dots. The tracker is composed by the following steps: first off the definition of dot's center of gravity is required; it is followed by the binarisation of the image; then, from the previous position of the dot's center of gravity, the algorithm goes right to detect the boundary and then follows such a boundary in order to compute the Freeman chain; if a dot is found, the algorithm checks if it looks like the previous dot (i.e. size, moment); if the dot is not found or similar, the algorithm checks out if the dot is in an image part around. This algorithm comes from [Marchand et al. (2005)] and was adapted to be implemented in the smart camera VC6212 nano.

Let us assume the following notation: ${}^{S_1}\mathbf{M}_{S_2}(X, Y, Z, R, P, Y)$, where ${}^{S_1}\mathbf{M}_{S_2}$ is the homogeneous matrix transformation of the S_2 system with respect to the S_1 one; X, Y and Z are Cartesian coordinates and R, P and Y are the roll, pitch and yaw Euler angles.

An object formed by 10 coplanar dots forming a star is used. The World's frame is noted with the sub-index w , and its origin is placed on the robot's base with positive-Z is pointing at the ceiling. The end-effector's frame is noted with the sub-index e , which initial pose with respect to the World's frame is described by ${}^wM_e(0.445, 0.0, -0.767, -3.14, 0.0, 1.57)$. The initial position of the camera's frame, noted by the sub-index c , with respect to the World's frame is described by ${}^wM_c(0.45, 0.0, -0.775, 0.0, 3.14, -1.57)$. The object frame is noted with the sub-index o (see Figure 9).

The comparison between DR-EKFS-IBVS, SR-LF-IBVS and classic SR-HF-IBVS controllers has been carried out for two positioning tasks. The first one is placing the object ${}^wM_o(0.45, 0.0, -0.967, 0.0, 0.0, -0.5236)$ with respect to the World's frame, a 30° pure rotation around the camera optical axis error task, which allows the convergence for all those controllers. Using this task we will evaluate the efficient performance of the proposed method compared to both classic SR-IBVS and our previous work. In the second task the object is placed at ${}^wM_o(0.45, 0.0, -0.967, 0.0, 0.0, 2.618)$, a configuration in which the classic IBVS fails (see [Solanes et al. (2013)] and the attached audio-visual material for more details), which will allow us to study its benefits with respect to our previous single-rate method.

Figure 10(a) shows the performance obtained using each control algorithm by solving the first positioning task. Choosing the optimal covariance matrices values for each controller and tuning the controller gain, the figure shows the minimum convergence time that is possible for each controller. It can be seen that, the best performance obtained is by using the DR-EKFS-IBVS controller while the worst is by using the classic SR-IBVS one.

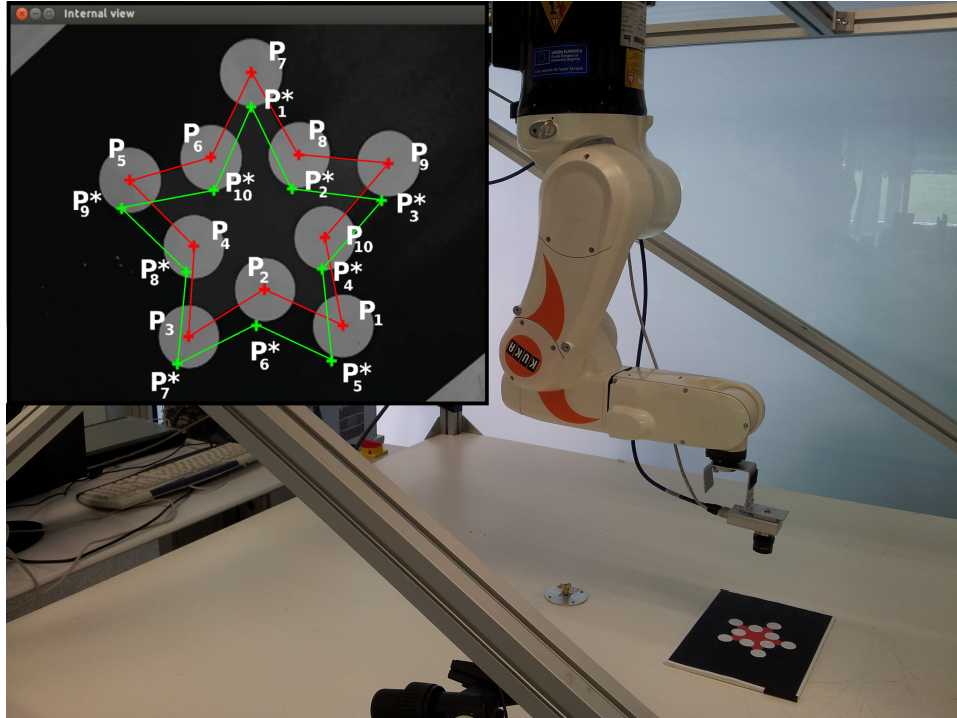
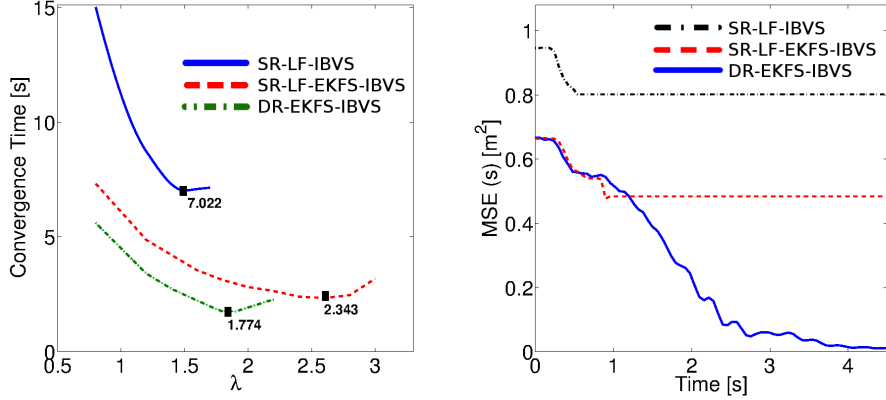


Figure 9: Experimental set-up with eye-in-hand configuration detecting ten coplanar dots forming the vertices of a star: the case represented is the second positioning task, where the object is placed at ${}^w M_o(0.45, 0.0, -0.967, 0.0, 0.0, -0.5236)$.

Focusing on the second positioning task, Figure 10(b), which represents the Mean-Squared Error (MSE) of the features, shows the failure of both classic SR-IBVS and SR-EKFS-IBVS controllers for solving such a configuration because of the sensor latency, while the DR-EKFS-IBVS controller has succeeded. Figure 11 represents the trajectories of each feature on the image plane (Figures 11(a) and 11(b)) and the control actions (Figure 11(c)) provided by the DR-EKFS-IBVS controller. This experiment can be viewed in the attached audio-visual material as well.

5 Conclusions

The main contribution of this paper is the generalization of the Reference Filtering control strategy presented in [Solanes et al. (2013)], from a dual-rate point of view, in order to overcome latency problems introduced by vision sensors in many robot



(a) Comparison of the convergence time solving a pure rotation of 30° around the optical axis: EKFS-IBVS and DR-EKFS-IBVS approaches, solid blue line corresponds to classic LF-IBVS solving the problem while the object is sited at case, dashed red line corresponds to SR-LF-EKFS-IBVS, and point-dashed green line corresponds to DR-EKFS-IBVS. In SR-LF-EKFS-IBVS and DR-EKFS-IBVS cases. The parameters used are the optimal ones in each case. (b) Comparison between SR-LF-IBVS, SR-LF-EKFS-IBVS and DR-EKFS-IBVS approaches solving the problem while the object is sited at case. The case shown corresponds to $N = 8$ and $h = 8$ and the rest of parameters used are the optimal ones in each case. The single-rate approaches fail, while DR-EKFS-IBVS approach is able to solve it.

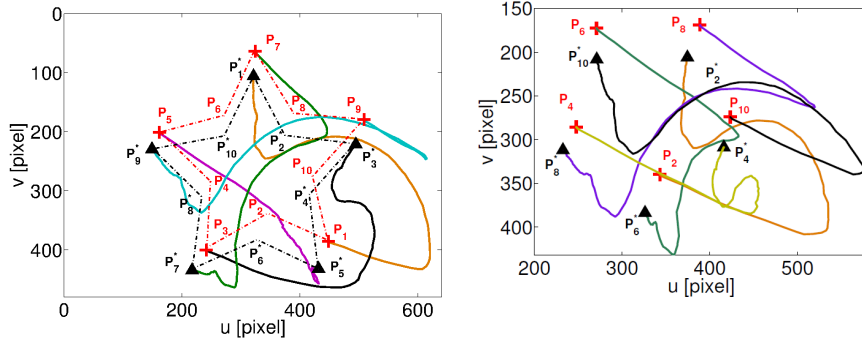
Figure 10: Analysis of the improvements reached by using the proposed dual-rate reference filtering control strategy.

visual servoing systems.

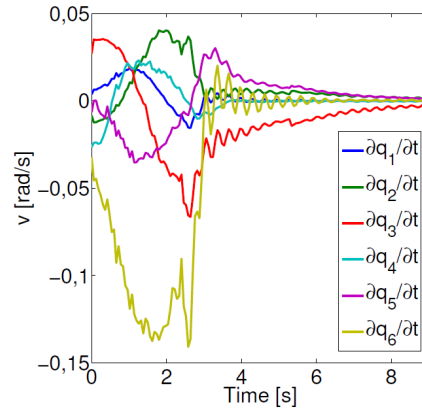
Additionally, the novel Dual-rate Reference Filtering control strategy has been implemented on industrial and embedded systems with significant hardware limitations. In particular, the experimental set-up includes a 6 DOF industrial manipulator, externally controlled by the embedded hardware, and a smart camera to record and process images.

In the paper, several experiments have been described to demonstrate the benefits of this novel dual-rate approach, not only compared with the classic IBVS controller but also compared with the single-rate reference filtering one. The dual-rate filtering controller has shown in fact, a much better performance in terms of convergence time, reachability and robustness. Therefore, it has been proved that, in some scenarios where the single-rate controllers fail because of sensor latency, the new dual-rate reference filtering controller succeeds.

Although, it has not been shown in this paper, due to lack of space, we have also validated the proposed dual-rate control structure using the well known 3D or Position-based Image Servoing (PBVS) approach, where similar results are ob-



(a) Image Plane Trajectories of the five external vertices of the star. (b) Image Plane Trajectories of the five internal vertices of the star.



(c) Control Actions ($v = \partial q_i / \partial t$ being $i = 1, \dots, 6$).

Figure 11: Problem while the object is sited at ${}^w M_o(0.441, 0.01, -0.967, 0.0, 0.0, 2.618)$ with respect to the frame reference: DR-EKFS-IBVS working on IGEP hardware can solve it, while SR-LF-IBVS and SR-LF-EKFS-IBVS cannot. The case shown corresponds to $N = 8$ and $h = 8$ and the rest of parameters used are the optimal ones in each case.

tained.

A Lifting technique

Lifted models can be derived from the general formulation presented in [Armesto and Tornero(2003)]. For a general linear periodic system, at base period δ :

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} \quad (23)$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t \quad (24)$$

with $\mathbf{x}_t \in \mathfrak{R}^n$ the state, $\mathbf{u}_t \in \mathfrak{R}^m$ the input and $\mathbf{y}_t \in \mathfrak{R}^p$ the output, the equivalent linear-time invariant (LTI) lifted model, at frame period is:

$$\mathbf{x}_t = \bar{\mathbf{A}}\mathbf{x}_{t-N} + \bar{\mathbf{B}}\bar{\mathbf{u}}_{t-N} \quad (25)$$

$$\bar{\mathbf{y}}_t = \bar{\mathbf{C}}\mathbf{x}_t + \bar{\mathbf{D}}\bar{\mathbf{u}}_t \quad (26)$$

where $\bar{\mathbf{u}}_t$ and $\bar{\mathbf{y}}_t$ are sparse lifted vectors (1):

$$\bar{\mathbf{u}}_t \prec \bar{\mathbf{U}}_t = \begin{bmatrix} \Delta_{\mathbf{u}}(k,0) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathbf{u}}(k,1) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Delta_{\mathbf{u}}(k,N-1) \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(0)} \\ \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(N-1)} \end{bmatrix}_k \quad (27)$$

$$\bar{\mathbf{y}}_t \prec \bar{\mathbf{Y}}_t = \begin{bmatrix} \Delta_{\mathbf{y}}(k,0) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathbf{y}}(k,1) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Delta_{\mathbf{y}}(k,N-1) \end{bmatrix} \begin{bmatrix} \mathbf{y}^{(0)} \\ \mathbf{y}^{(1)} \\ \vdots \\ \mathbf{y}^{(N-1)} \end{bmatrix}_k \quad (28)$$

with \prec a row-reduction operator which removes zero elements from $\bar{\mathbf{U}}_t \in \mathfrak{R}^{Nm}$ and $\bar{\mathbf{Y}}_t \in \mathfrak{R}^{Np}$, thus $\bar{\mathbf{u}}_t$ and $\bar{\mathbf{y}}_t$ are sparse vectors of the full lifted vectors $\bar{\mathbf{U}}_t$ and $\bar{\mathbf{Y}}_t$, respectively. Where a general sampling scheme for inputs and outputs is defined as follows:

$$\Delta_{\mathbf{u},t+i} = \Delta_{\mathbf{u}}(k,i) = \text{diag}\{d^{u_w}(k,i), w = 1, 2, \dots, m\} \in \mathfrak{R}^{m \times m} \quad (29)$$

$$\Delta_{\mathbf{y},t+i} = \Delta_{\mathbf{y}}(k,i) = \text{diag}\{d^{y_v}(k,i), v = 1, 2, \dots, p\} \in \mathfrak{R}^{p \times p} \quad (30)$$

with,

$$d^{u_w}(k,i) = \begin{cases} 1 & \text{if } u_w \text{ is sampled at time } t+i \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

$$d^{y_v}(k,i) = \begin{cases} 1 & \text{if } y_v \text{ is sampled at time } t+i \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

where m and p are the number of inputs and outputs, respectively. Note that we use a double-index notation to refer to the same time instant $t = k \cdot N$, being N the ratio between the base and frame periods $\bar{T} = N\delta$.

According to previous formulation, matrices $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, $\bar{\mathbf{C}}$ and $\bar{\mathbf{D}}$ are defined as follows:

$$\bar{\mathbf{A}} = \mathbf{A}^N, \bar{\mathbf{B}} \prec_{\bar{\mathbf{U}}_t}^{col} [\mathbf{A}^{N-1}\mathbf{B} \ \mathbf{A}^{N-2}\mathbf{B} \ \dots \ \mathbf{B}] \quad (33)$$

$$\bar{\mathbf{C}} \prec_{\bar{\mathbf{Y}}_t}^{row} \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix}, \bar{\mathbf{D}} \prec_{\bar{\mathbf{Y}}_t}^{row} \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0} \\ \vdots & & & \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \dots & \mathbf{CB} \end{bmatrix} \quad (34)$$

where $\prec_{\bar{\mathbf{U}}_t}^{col}$ is a column-reduction operator appending columns of the right-hand side term. It sums up columns related to zero elements of $\bar{\mathbf{U}}_t$ to columns of non-zero elements (columns are appended to their left). Similarly, $\prec_{\bar{\mathbf{Y}}_t}^{row}$ is a row-reduction operator that removes rows of the right-hand side term related to zero elements of $\bar{\mathbf{Y}}_t$.

References

- [Allibert et al. (2010)] G. Allibert, E. Courtial and F. Chaumette (2010). *Visual servoing via Nonlinear Predictive control*. In: Visual Servoing via Advanced Numerical Methods, Chesi G, Hashimoto K (eds), LNCIS 401, Springer-Verlag, pp 375–394
- [Armesto et al. (2014)] L. Armesto, V. Girbes, A. Sala, M. Zima and V Smidl (2013). *Duality-Based Nonlinear Quadratic Control: Application to Mobile Robot Trajectory-Following*. In: IEEE Transactions on Control Systems Technology (in press).
- [Armesto and Tornero(2003)] L. Armesto, J. Tornero (2003). *A general formulation for generating multi-rate models*. In: American Control Conference, pp 1146–1151
- [Armesto and Tornero (2003)] L. Armesto; J. Tomero (2003). *Dual-rate high order holds based on primitive functions*. In: American Control Conference, 2003. Proceedings of the 2003 , vol.2, no., pp.1140,1145, June 4-6, 2003 doi: 10.1109/ACC.2003.1239741
- [Associated Press (2013)] Associated Press V(2013). *Navy launches unmanned aircraft from deck of aircraft carrier for 1st time*. In: Washington Post.
- [Bamieh et al. (1991)] B. Bamieh, J. Pearson, F. BA, T. A (1991). *A lifting technique for linear periodic systems with applications to*

sampled-data control. In *Systems and Control Letters* 17:79–88, DOI [http://dx.doi.org/10.1016/0167-6911\(91\)90033-B](http://dx.doi.org/10.1016/0167-6911(91)90033-B)

- [Bruyninckx et al. (2003)] H. Bruyninckx, P. Soetens and B. Koninckx (2003). *The Real-Time Motion Control Core of the Orocos Project*. IEEE International Conference on Robotics and Automation.
- [Chaumette(1998)] F. Chaumette (1998). *Potential problems of stability and convergence in image-based and position-based visual servoing*. In: *The Confluence of Vision and Control*, LNCIS Series, No 237, Springer-Verlag, pp 66–78
- [Chaumette and Hutchinson(2006)] F. Chaumette and S. Hutchinson (2006). *Visual servo control. i. basic approaches*. *Robotics Automation Magazine*, IEEE 13(4):82–90, DOI 10.1109/MRA.2006.250573
- [Chaumette and Hutchinson(2007)] F. Chaumette, S. Hutchinson (2007). *Visual servo control, part ii: Advanced approaches*. *IEEE Robotics and Automation Magazine* 14(1):109–118
- [Corke(2011)] P. Corke (2011). *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. In *Springer Tracts in Advanced Robotics*, Vol. 73.
- [Hashimoto et al. (1998)] K. Hashimoto and T. Noritsugu (1998). *Performance and Sensitivity in Visual Servoing*. *Int. Conf. on Robotics and Automation (ICRA)*
- [Hutchinson et al. (1996)] S. Hutchinson, G. Hager and P. Corke (1996). *A tutorial on visual servo control*. *Robotics and Automation*, IEEE Transactions on 12(5):651–670, DOI 10.1109/70.538972
- [Khargonekar et al. (1985)] P. Khargonekar, K. Poolla, A. Tannenbaum (1985). *Robust control of linear time-invariant plants using periodic compensation*. In *IEEE Transactions on Automatic Control*, 30(11):1088–1096.
- [Marchand et al. (2005)] E. Marchand, F. Spindler and F. Chaumette (2005). *Visp for visual servoing: a generic software platform with a wide class of robot control skills*. In *IEEE Robotics Automation Magazine*, 12(4):40–52, DOI 10.1109/MRA.2005.1577023.
- [Rosen Diankov and James Kuffner (2008)] R. Diankov and J. Kuffner (2008). *OpenRAVE: A Planning Architecture for Autonomous Robotics*. In *Robotics Institute*, CMU-RI-TR-08-34.

- [Rauch et al. (1965)] H. E. Rauch, C. T. Striebel and F. Tung (1965). *Maximum likelihood estimates of linear dynamic systems*. in Journal of the American Institute of Aeronautics and Astronautics 3(8):1445–1450.
- [Soetens et al. (2005)] P. Soetens and H. Bruyninckx (2005). *Realtime Hybrid Task-Based Control for Robots and Machine Tools*. In IEEE International Conference on Robotics and Automation.
- [Solanes et al. (2011)] J.E. Solanes, J. Tornero, L. Armesto, V. Girbés (2011). *Multi-rate visual servoing based on dual-rate high order holds*. In: Proceedings of the 12th Annual conference on Towards autonomous robotic systems, Towards Autonomous Robotic Systems (TAROS), pp 195 – 206.
- [Solanes et al. (2012)] J.E. Solanes, L. Armesto, J. Tornero, P. Muñoz-Benavent and V. Girbes (2012). *Dual-rate non-linear high order holds for visual servoing applications*. In: Towards Autonomous Robotic Systems (TAROS).
- [Solanes et al. (2013)] J.E. Solanes, L. Armesto, J. Tornero and V. Girbes (2013). *Improving image-based visual servoing with reference features filtering*. In Int. Conference on Robotics and Automation (ICRA).
- [Todorov (2008)] E. Todorov (2008). *General duality between optimal control and estimation*. In *Decision and Control, 2008.*, CDC 2008. 47th IEEE Conference on, pages 4286–4292.
- [Zima et al. (2013)] M. Zima, L. Armesto, V. Girbes, A. Sala and V. Smidl (2013). *Extended Rauch-Tung-Striebel Controller.*, In: Conference on Decision and Control.