MULTI-RATE VISUAL FEEDBACK ROBOT CONTROL

Ph.D. Thesis

Author:

Juan Ernesto Solanes Galbis

Supervisors:

Josep Tornero Montserrat
Leopoldo Armesto Ángel

Institute of Design and Manufacturing
Departamento de Ingeniería de Sistemas y Automática
Universitat Politècnica de València

November, 2015

MULTI-RATE VISUAL FEEDBACK ROBOT CONTROL

Ph.D. in Robotics, Automatics and Computer Engineering

Author:

Juan Ernesto Solanes Galbis

Supervised by:

Josep Tornero Montserrat
Leopoldo Armesto Ángel

Institute of Design and Manufacturing
Departamento de Ingeniería de Sistemas y Automática
Universitat Politècnica de València

November, 2015

This work has been supported by the Generalitat Valenciana (Programa VALi+d), 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)

The author was recipient of the fellowship *Formación de investigadores en fase pre-doctoral* (ACIF/2010/242) by Generalitat Valenciana and *Programa de becas de excelencia* by Universitat Politècnica de València.

This work has also been supported by the Mechanical Department of the University of California, Berkeley (USA), accepting the author in its facilities as pre-doctoral stay and allowing a close collaboration with its Robotic group as well as collaborating with FANUC Robotics in Japan. The author was recipient of the fellowship *Becas para estancias de becarios y contratados pre-doctorales en centros de investigación fuera de la Comunidad Valenciana* (BEFPI/2013/027) by Generalitat Valenciana, supporting economically the stay.

To my parents

Acknowledgement

Remember who has been at your side
when you failed because, ultimately, he
is the one responsible of your success.

If when I finished my studies at school and started the bachelor grade someone would have told me that I would get to do a Ph.D., I would have not believed him. Actually, there was a moment in the beginning of the path that I tripped down and I thought of not going on with my academy life any more.

But I am very lucky. Yes, I am. Because it is luck to have a parents who always support you, push you up to keep going on, leave you get wrong and then forgive you. It is luck when you do not believe in yourself, in your possibilities, and they are always there to show you up how wrong you are and, what is more important, whatever you do, for them, will be right.

For that, I have wanted to dedicate these few lines to thank my parents, Ángeles Maria Galbis and Juan Ernesto Solanes, all their tips, their comprehension, their unconditional support and all the cheers they got me along my entire life. Also, ask them for forgiveness and thank them for their support these years of thesis. *If I have become what I have become, and be who I am, is because both of you. Many thanks mum and dad.*

Neither I do wish to forget my two little angels, María Ángeles Solanes Galbis and Carmen Solanes Galbis, for being always at my side, helping me in whatever I need.

Academically, I have had the fortune of knowing and being able to work side by side with two great researchers and better people, my advisers Josep Tornero Montserrat and Leopoldo Armesto Ángel. I have conveyed their enthusiasm for research as well as their vision and character when solving problems. I thank you both all the hours expended in reviewing my work, your tips and critics. They have been needed to achieve the established objectives.

Moreover, I also wanted to say how impress I am of seeing that, although life has given you a very hard blow, Josep, you decided to stand up and fight back. Very few would have done what you did and this is something you have thought me too.

Of course, I must not forget to thank my friends and colleagues, Vicent Girbés Juan and Pau Muñoz Benavent, for all these years of work, mutual support and laughs. It is incredible how this period of life has linked us, doesn't it? And also, my dears colleagues from the Institute of Design and Manufacturing, Vicente Franch, Mario Lledó, Laura Arnal, Jaime Molina, Diego Perez and so many more, for the assistance provided during these years and, what it is more important, for creating a perfect work environment for achieving all our goals.

My last words are dedicated to Professor Masayoshi Tomizuka from University of Berkeley, California (USA), for let me do my research stay collaborating closely with him and his group in control, robotics and vision topics.

Agraïments

Recorda el que ha estat al teu costat
en el fracàs perquè és el màxim
responsable del teu èxit.

Si quan vaig finalitzar l'escola i comencí els meus estudis de batxillerat algú m'haguera dit que arribaria a realitzar una tesi doctoral, no l'haguera cregut. De fet, va hi haure un moment només començar els estudis universitaris que vaig ensopegar i em vaig inclús plantejar el no continuar amb els estudis.

Però soc una persona amb sort. Si, sort. Perquè és sort el tindre uns pares que sempre estiguen al teu costat, t'empenten a seguir endavant, deixen que t'equivoques i et perdonen quan ho fas. Es sort que, quan no creus amb tu, amb les teues possibilitats, estiguen aquí per a que et dones conter de lo enganyat que estàs i, el més important, de que fages el que fages, per a ells estarà be.

Per aquest motiu, he volgut dedicar estes poques línies a agrair als meus pares, Ángeles María Galbis Martínez i Juan Ernesto Solanes Galán, tots els consells, comprensió, el suport incondicional i tots els ànims que m'han donat a lo llarg de la meua vida. També el demanar-los perdó i agrair-los el haver-me aguantat aquestos últims anys de tesi. *Si he arribat on he arribat, i ser el que soc, és per vosaltres. Moltes gracies als dos, mamà i papà.*

No puc oblidar-me tampoc dels meus dos angelets, María Ángeles Solanes

Galbis i Carmen Solanes Galbis, per estar sempre ajudant-me en tot el que necessito i cuidar-me com ho fan.

Per lo que respecta a lo acadèmic, he tingut la gran fortuna de conèixer i poder treballar colze amb colze amb dos grans investigadors i millor persones, els meus directors Josep Tornero Montserrat i Leopoldo Armesto Ángel. Ells m'han transmès el seu entusiasme per la investigació a lo llarg de tota la tesi, així com la seua visió i caràcter a l'hora de solucionar els problemes que s'anaven plantejant. Als dos vos agraeixo totes les hores de revisions del meu treball que hagueu dedicat, les vostres crítiques i consells. Han sigut indispensables per l'obtenció dels resultats.

A més a més, voldria també expressar lo admirat que estic de veure que, però la vida t'ha donat un molt dur colp, Josep, tu has decidit mantenir-te en peu i seguir lluitant. Pocs ho hagueren fet al teu lloc i crec que no m'equivoco quan dic que és lo més important que m'has ensenyat.

Com no, no puc sinó agrair als meus amics i companys d'aventura, Vicent Girbés Juan i Pau Muñoz Benavent, per tots aquestos anys de treball, suport mútua i rialles que em passat. És increïble el que ens ha aplegat a unir aquest període de tesi, veritat? I com no, recordar als meus estimats companys del Institut de Disseny i Fabricació, Vicent Franch, Mario Lledó, Laura Arnal, Jaime Molina, Diego Perez, i molts més que espero que em perdonen si no els he nombrat, per tota l'ajuda prestada i, el més important, per crear un clima perfecte de treball per la consecució dels nostres objectius.

Les meues últimes línies d'agraïment van dedicades al professor de La Universitat de Berkeley, Califòrnia (USA), Masayoshi Tomizuka, per haver-me permès treballar junt amb el seu grup d'investigació en temes de control, robòtica industrial i visió artificial, que no solament m'ha enriquit acadèmicament sinó personalment.

Abstract

This thesis deals with two characteristic problems in visual feedback robot control: 1) sensor latency; 2) on providing suitable trajectories for the robot and for the measurement in the image. All the approaches presented in this work are analyzed and implemented on a 6 DOF industrial robot manipulator or/and a wheeled robot.

Focusing on the sensor latency problem, this thesis proposes the use of dual-rate high order holds within the control loop of robots. In this sense, the main contributions are:

- **Dual-rate high order holds based on primitive functions for robot control (Chapter 3):** analysis of the system performance with and without the use of this multi-rate technique from non-conventional control. In addition, as consequence of the use of dual-rate holds, this work obtains and validates multi-rate controllers, concretely dual-rate PIDs.

- **Asynchronous dual-rate high order holds based on primitive functions with time delay compensation (Chapter 3):** generalization of asynchronous dual-rate high order holds incorporating an input signal time delay compensation component, improving thus the inter-sampling estimations computed by the hold. It is provided an analysis of the properties of such dual-rate holds with time delay compensation, comparing them with estimations obtained by the equivalent dual-rate holds without this compensation, as well as their implementation and validation within the control loop of a 6 DOF industrial robot manipulator.

- **Multi-rate nonlinear high order holds (Chapter 4):** generalization of the concept of dual-rate high order hold with nonlinear estimation models, which include information about the plant to be controlled, the controller(s) and sensor(s) used, obtained from machine learning techniques. Thus, in order to obtain such a nonlinear hold, it is described a methodology non dependent of the machine technique used, although validated using artificial neural networks. Finally, an analysis of the properties of these new holds is carried out, comparing them with their equivalents based on primitive functions, as well as their implementation and validation within the control loop of an industrial robot manipulator and a wheeled robot.

With respect to the problem of providing suitable trajectories for the robot and for the measurement in the image, this thesis presents the novel reference features filtering control strategy and its generalization from a multi-rate point of view. The main contributions in this regard are:

- **Reference features filtering control strategy (Chapter 5):** a new control strategy is proposed to enlarge significantly the solution task reachability of robot visual feedback control. The main idea is to use optimal trajectories proposed by a non-linear EKF predictor-smoother (ERTS), based on Rauch-Tung-Striebel (RTS) algorithm, as new feature references for an underlying visual feedback controller. In this work it is provided both the description of the implementation algorithm and its implementation and validation utilizing an industrial robot manipulator.

- **Dual-rate Reference features filtering control strategy (Chapter 5):** a generalization of the reference features filtering approach from a multi-rate point of view, and a dual Kalman-smoother step based on the relation of the sensor and controller frequencies of the reference filtering control strategy is provided, reducing the computational cost of the former algorithm, as well as addressing the problem of the sensor latency. The implementation algorithms, as well as its analysis, are described.

Resum

La present tesis proposa solucions per a dos problemes característics dels sistemes robòtics el els que el bucle de control es tanca únicament utilitzant sensors de visió artificial: 1) la latència del sensor; 2) l'obtenció de trajectòries factibles tant per al robot com per les mesures en la imatge. Tots els mètodes propostos en aquest treball son analitzats, validats e implementats utilitzant un braç robot industrial de 6 graus de llibertat i/o un robot amb rodes.

Atenent al problema de la latència del sensor, esta tesis proposa l'ús de retenidors bi-freqüència d'ordre alt a dins del llaços de control de robots. Al respecte, les principals contribucions son:

- **Retenidors bi-freqüència d'ordre alt basats en funcions primitives a dintre dels llaços de control de robots (Capítol 3):** anàlisis del comportament del sistema amb i sense l'ús d'aquesta tècnica de control no convencional. A més a més, com a conseqüència de l'ús dels retenidors, obtenció i validació de controladors multifreqüència, concretament de PIDs bi-freqüència.

- **Retenidors bi-freqüència asíncrons d'ordre alt basats en funcions primitives amb compensació de retards (Capítol 3):** generalització dels retenidors bi-freqüència asíncrons d'ordre alt inclouen una component de compensació del retràs en la senyal d'entrada al retenidor, millorant així les estimacions inter-mostreig calculades per el retenidor. Es proporciona un anàlisis de les propietats dels retenidors

amb compensació del retràs, comparant-les amb les obtingudes per el seus predecessors sense la compensació, així com la seua implementació i validació en un braç robot industrial de 6 graus de llibertat.

- **Retenidors multi-freqüència no-lineals d'ordre alt (Capítol 4):** generalització del concepte de retenidor bi-freqüència d'ordre alt amb models d'estimació no lineals, incloent informació tant de la planta a controlar, com del controlador(s) i sensor(s) utilitzat(s), obtenint-la a partir de tècniques d'aprenentatge. Així doncs, per obtindre el retenidor no lineal, es descriu una metodologia independent de la ferramenta d'aprenentatge utilitzada, però validada amb l'ús de rets neuronals artificials. Finalment es realitza un anàlisis de les propietats d'aquestos nous retenidors, comparant-los amb els seus predecessors basats amb funcions primitives, així com la seua implementació i validació amb un braç robot de 6 graus de llibertat i amb un robot mòbil de rodes.

Per el que respecta al problema de generació de trajectòries factibles per al robot i per la mesura en la imatge, aquesta tesis proposa la nova estratègia de control basada amb el filtrat de la referència i la seua generalització des de el punt de vista multi-freqüència.

- **Estratègia de control basada amb el filtrat de la referència (Capítol 5):** una nova estratègia de control es proposada per ampliar significativament l'espai de solucions dels sistemes robòtics realimentats amb sensors de visió artificial. La principal idea es la d'utilitzar les trajectòries optimes obtingudes per una trajectòria predita per un filtre de Kalman seguit d'un suavitzat basat en l'algoritme Rauch-Tung-Striebel (RTS) com noves referències per a un control donat. En aquest treball es proporciona tant la descripció del algoritme així com la seua implementació i validació utilitzant un braç robòtic industrial de 6 graus de llibertat.

- **Estratègia de control bi-freqüència basada en el filtrat (Capítol 5):** generalització de l'estratègia de control basada am filtrat de

la referència des de un punt de vista multi freqüència, amb un filtre de Kalman multi freqüència i un Kalman-Smoother dual basat amb la relació existent entre les freqüències del sensor i del controlador, reduint així el cost computacional de l'algoritme i, al mateix temps, donant solució al problema de la latència del sensor. L'algoritme d'implementació d'aquesta tècnica, així com la seua validació utilitzant un braç robot industrial de 6 graus de llibertat, es descriu a lo llarg d'aquesta tesis.

Resumen

La presente tesis propone soluciones para dos problemas característicos de los sistemas robóticos cuyo bucle de control se cierra únicamente empleando sensores de visión artificial: 1) la latencia del sensor; 2) la obtención de trayectorias factibles tanto para el robot así como para las medidas obtenidas en la imagen. Todos los métodos propuestos en este trabajo son analizados, validados e implementados utilizando brazo robot industrial de 6 grados de libertad y/o en un robot con ruedas.

Atendiendo al problema de la latencia del sensor, esta tesis propone el uso de retenedores bi-frequencia de orden alto dentro de los lazos de control de robots. En este aspecto las principales contribuciones son:

- **Retenedores bi-frecuencia de orden alto basados en funciones primitivas dentro de lazos de control de robots (Capítulo 3):** análisis del comportamiento del sistema con y sin el uso de esta técnica de control no convencional. Además, como consecuencia del empleo de los retenedores, obtención y validación de controladores multi-frequencia, concretamente de PIDs bi-frecuencia.

- **Retenedores bi-frecuencia asíncronos de orden alto basados en funciones primitivas con compensación de retardos (Capítulo 3):** generalización de los retenedores bi-frecuencia asíncronos de orden alto incluyendo una componente de compensación del retardo en la señal de entrada, mejorando así las estimaciones inter-muestreo calculadas por el retenedor. Se proporciona un análisis de las propiedades de los retenedores con compensación del retardo, comparándolas con

las obtenidas por sus predecesores sin compensación, así como su implementación y validación en un brazo robot de 6 grados de libertad.

- **Retenedores multi-frecuencia no lineales de orden alto (Capítulo 4):** generalización del concepto de retenedor bi-frecuencia de orden alto con modelos de estimación no lineales, los cuales incluyen información tanto de la planta a controlar, como del controlador(es) y sensor(es) empleado(s), obtenida a partir de tecnicas de aprendizaje. Así pues, para obtener dicho retenedor no lineal, se describe una metodología independiente de la herramienta de aprendizaje utilizada, aunque validada con el uso de redes neuronales artificiales. Finalmente se realiza un análisis de las propiedades de estos nuevos retenedores, comparándolos con sus predecesores basados en funciones primitivas, así como su implementación y validación en un brazo robot de 6 grados de libertad y en un robot móvil con ruedas.

Por lo que respecta al problema de generación de trayectorias factibles para el robot y para la medida en la imagen, esta tesis propone la nueva estrategia de control basada en el filtrado de la referencia y su generalización desde el punto de vista multi-frecuencial.

- **Estrategia de control basada en el filtrado de la referencia (Capítulo 5):** una nueva estrategia de control se propone para ampliar significativamente el espacio de soluciones de los sistemas robóticos realimentados con sensores de visión artificial. La principal idea es utilizar las trayectorias óptimas obtenidas por una trayectoria predicha por un filtro de Kalman seguido de un suavizado basado en el algoritmo Rauch-Tung-Striebel (RTS) como nuevas referencias para un controlador dado. En este trabajo se proporciona tanto la descripción del algoritmo como su implementación y validación empleando un brazo robótico industrial.

- **Estrategia de control bi-frecuencia basada en el filtrado de la referencia (Capítulo 5):** generalización de la estrategia de control basada en filtrado de la referencia desde un punto de vista multi-

frecuencial, con un filtro de Kalman multi-frecuencia y un Kalman-Smoother dual basado en la relación existente entre las frecuencias del sensor y del controlador, reduciendo así el coste computacional del algoritmo y, al mismo tiempo, dando solución al problema de la latencia del sensor. El algoritmo de implementación de este enfoque, así como su validación empleando empleándo un brazo robot industrial, es descrito a lo largo de la presente tesis.

Contents

List of Figures

List of Tables

List of Terms

δ: Base period

$\overline{\delta}$: Frame period

t_d: Time delay

$\mathcal{L}$: Lifting operator

DR-HOH: Dual-rate high order hold

DR-ZOH: Dual-rate zero order hold

DR-FOH: Dual-rate first order hold

DR-SOH: Dual-rate second order hold

DR-SOBH: Dual-rate second order bezier hold

DR-SOTH: Dual-rate second order Taylor hold

DR-NLHOH: Dual-rate non-linear high order hold

Asyn. DR-HOH-TDC: Asynchronous dual-rate high order hold with time delay compensation

Asyn. DR-FOH-TDC: Asynchronous dual-rate first order hold with time delay compensation

Asyn. DR-SOH-TDC: Asynchronous dual-rate second order hold with time delay compensation

$\mathbf{u}_h$: Dual-rate high order hold output

$\mathbf{G}_h(s)$: Continuous-time hold transfer function

$\mathbf{G}_h(z^N)$: Discrete-time hold transfer function at frame period

$\mathbf{G}_h(z)$: Discrete-time hold transfer function at base period

$\mathbf{A}_h$, $\mathbf{B}_h$, $\mathbf{C}_h$ and $\mathbf{D}_h$: Dynamic model of dual-rate high order hold at base period.

$\bar{\mathbf{A}}_h$, $\bar{\mathbf{B}}_h$, $\bar{\mathbf{C}}_h$ and $\bar{\mathbf{D}}_h$: Dynamic model of dual-rate high order hold at frame period.

$\mathbf{P}_c(s)$: Continuous-time plant transfer function

$\mathbf{A}_c$, $\mathbf{B}_c$, $\mathbf{C}_c$ and $\mathbf{D}_c$: Continuous-time dynamic model of the plant.

$\mathbf{A}_\delta$, $\mathbf{B}_\delta$, $\mathbf{C}_\delta$ and $\mathbf{D}_\delta$: Discrete-time dynamic model of the plant at base period.

$\mathbf{A}_{N\delta}$, $\mathbf{B}_{N\delta}$, $\mathbf{C}_{N\delta}$ and $\mathbf{D}_{N\delta}$: Discrete-time dynamic model of the plant at frame period.

$\mathbf{f}$: Image feature

$\mathbf{s}$: Set of image features

$\mathbf{q}$: Robot joints

${}^rJ_r$:Robot Jacobean

${}^{\mathbf{S_1}}\mathbf{V}_{\mathbf{S_2}}$:Twist transformation matrix of S_2 with respect to S_1

${}^{\mathbf{S_1}}\mathbf{M}_{\mathbf{S_2}}$:Homogeneous transformation matrix of S_2 with respect to S_1

EKFS: Extended Kalman Filter/Smoother algorithm

DS-EKFS: Dual-sampling Extended Kalman Filter/Smoother algorithm

SR-EKFS-IBVS: Single-rate Extended Kalman Filter/Smoother Image based Visual Servoing control strategy

SR-EKFS-PBVS: Single-rate Extended Kalman Filter/Smoother Position based Visual Servoing control strategy

SR-EKFS-VFC: Single-rate Extended Kalman Filter/Smoother Visual Feedback Control Strategy

DR-EKFS-VFC: Dual-rate Extended Kalman Filter/Smoother Visual Feedback Control Strategy

DR-EKFS-VFC: Dual-sampling Extended Kalman Filter/Smoother Visual Feedback Control Strategy

DR-EKFS-IBVS: Dual-sampling Extended Kalman Filter/Smoother Image based Visual Servoing control strategy

DR-EKFS-PBVS: Dual-sampling Extended Kalman Filter/Smoother Position based Visual Servoing control strategy

DR-EKFS-P: Dual-sampling Extended Kalman Filter/Smoother Position based Visual Servoing control strategy with proportional control law

DR-EKFS-PID: Dual-sampling Extended Kalman Filter/Smoother Position based Visual Servoing control strategy with PID control law

Chapter 1

Introduction

Visual feedback robot control has been studying along the last four decades by many researchers, mostly from the areas of control and computer science, as shows Figure 1.1. It is a viable method for robot control based on the utilization of visual information extracted from images to close the robot control loop. Using vision in robot control makes it possible to solve different problems, which can be handled safely based on the sensory visual data without any contact with the environment.

Yet, until the last fifteen years, this field of research was not fully applied

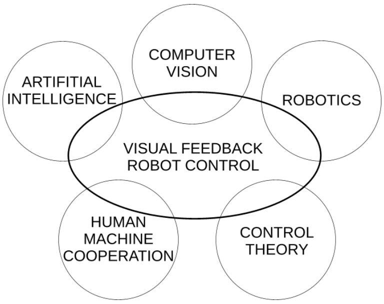


Figure 1.1: Visual feedback robot control related areas.

in real-time processes because of technology limitations. Nowadays however, the growth in robotics, computer science and sensor technologies has allowed applying all the knowledge acquired in the past, solving and improving thus all sort of applications, yielding a spectacular impact in several areas including industrial manufacturing, automotive systems, medical robots for surgery, and on.

Despite of this, there is still being very important issues that have to be considered when vision sensors are used within automatic systems. These issues include local or global stability, Branicky (1997); Chaumette (1998); Gans and Hutchinson (2007); Park et al. (2012), robustness, Malis et al. (2010); Assa and Janabi-Sharifi (2014), suitable trajectories for the robot and for the measurement in the image, Pages et al. (2006); Chesi (2009b), and the problem of the sensor latency, Usai and Di Giamberardino (2006); Fujimoto (2003); Sasajima and Fujimoto (2007).

The commented problems in visual feedback robot control have not been an obstacle to develop a wide variety of applications using all kind of robotic platforms, ranging from robot manipulators, Kragic and Christensen (2002), underwater robots, Lots et al. (2001); Heshmati-Alamdari et al. (2014), areal robots, Barajas et al. (2013); Nguyen et al. (2014), or humanoids robots, Agravante et al. (2013); Carbajal-Espinosa et al. (2014). The actual commercial applications avoid some of the commented problems by introducing high technological devices, increasing thus the price of the application, or well reducing the 3D match control problem to the more much simple 2D control problem.

This demonstrates the interest this field of research awakes not only in the academy society but in the society in general.

Motivations

Visual feedback robot control encounters different sort of issues, as the sensor latency problem or task reachability. The first one is usually avoided by the utilization of high speed cameras, increasing considerably the cost of the application and decreasing their resolution. However, there are so

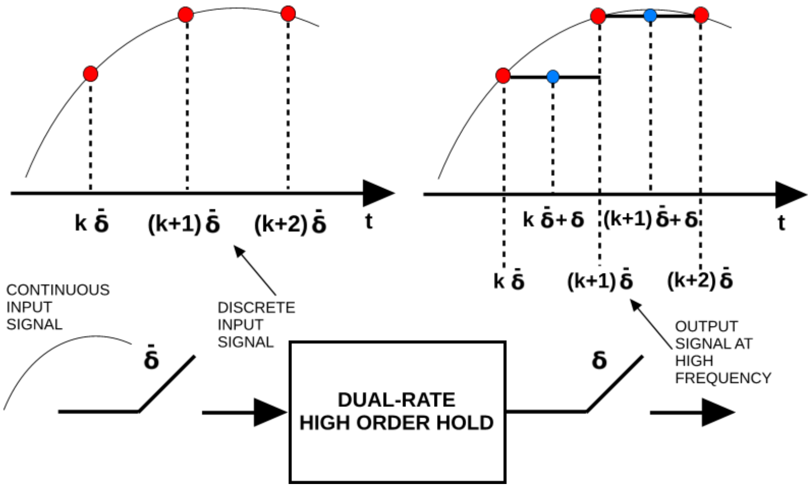


Figure 1.2: Dual-rate High Order Hold.

many systems that use limited hardware, such as drones, small robots using embedded hardware, etc., in which the sensor latency problem is a very important issue that must be taken into account when controlling them.

For the second problem, there are many approaches trying to overcome it, although the majority of them have a high computational cost that makes them difficult to be implemented in hardware with hard real-time limitations.

These sorts of problems have motivated us to accomplish the present work.

Contributions and Manuscript Outline

This manuscript is compound of 6 chapters including the introductory and the conclusions chapters. After the introductory chapter, it starts by reviewing the state of the art in multi-rate control schemes, non-linear and optimal control, and visual feedback control. Then new approaches to solve the problem of sensor latency in visual feedback robot control are proposed and their experimental evaluations are presented. Following, a novel control strategy for visual feedback robot control is formulated in order to provide suitable trajectories for the measurement in the image, when using hardware with limited resources. Then, its generalization from a multi-rate point of view is formulated and a set of algorithms are provided. Finally, the conclusions of these approaches are collected in the last chapter.

- **Chapter 2. State of the art:** In this chapter, state of the art on multi-rate control is focused on dual-rate high order holds, nonlinear control focused on optimal controllers, and visual feedback control in the light of visual features used, control schemes designed and problems that could appear, is presented.

- **Chapter 3. Visual feedback control based on dual-rate high order holds:** In this chapter, the sensor latency problem is addressed via classical dual-rate high order holds, Solanes et al. (2011), which are circuits for generating, from a sequence of inputs sampled at low sampling rates, a continuous signal or a discrete one at a higher sampling rate (see Figure 1.2). A set of experimental evaluations using a wheeled robot performing path following task based on visual feedback data controlled by the well known Pure Pursuit controller, and a 6 DOF industrial robot manipulator performing a visual feedback control positioning task by considering the most usual and simple features, which are the Cartesian coordinates of image points, are presented, Solanes et al. (2011). Then, dual-rate high order holds are generalized by considering time delays of the input signal. A general formulation and algorithm are then described obtaining the new Asynchronous Dual-rate High Order holds with Time Delay Compensation. Experimental experimentation is provided by using a 6 DOF industrial robot manipulator.

- **Chapter 4. Visual feedback control based on dual-rate nonlinear high order holds:** In this chapter, we propose a new dual-rate hold function that includes information about the system to be controlled, the specific controller used and the task performed. It is based on machine learning techniques, which uses real data from the system in order to learn the hold function. A methodology for obtaining such dual-rate nonlinear function is also presented by using artificial neural networks as learning tool. Finally, experimental results are provided demonstrating the properties of such an approach, Solanes et al. (2012b,a).

- **Chapter 5. Visual feedback control based on reference features filtering control strategy:** In this chapter, a new control strategy is proposed to enlarge significantly the solution task reachability of visual feedback robot control. The main idea is to use optimal trajectories proposed by a non-linear EKF predictor-smoother (ERTS), based on Rauch-Tung-Striebel (RTS) algorithm, Rauch et al. (1965), as new feature references for an underlying visual feedback controller. In addition to this, its generalization from a multi-rate point of view and a dual Kalman-smoother step based on the relation of the sensor and controller frequencies is provided, reducing thus the computational cost of the algorithm, as well as addressing the problem of the sensor latency. Finally, experimental results are provided showing up the properties of this new control strategy, Ernesto Solanes et al. (2013); Solanes et al. (2015).

This work has led to several publications in the robotic community. Following the order they appear in throughout this chapter, here are the contributions with the corresponding publications:

JCR Journal papers

- Solanes, J. Ernesto and Muñoz-Benavent, Pau and Girbés, Vicent and Armesto, Leopoldo and Tornero, Josep. On Improving Robot Image Based Visual Servoing Based On Dual-rate Reference Filtering Control Strategy. In *Robotica.* Vol. FirstView, Pag. 1-18, 2015.
  DOI 10.1017/S0263574715000454 ISSN 1469-8668

International conference papers and non-indexed journals

- Solanes, J. Ernesto and Armesto, Leopoldo and Tornero, Josep and Girbés, Vicent. Improving image based visual servoing with reference features filtering. In *International Conference on Robotics and Automation (ICRA 2013)*. Karlsruhe, Germany, May 6-10, 2013. Page 3083-3088.

- Solanes, J. Ernesto and Girbés, Vicent and Tornero, Josep and Armesto, Leopoldo. Non-linear Dual-rate Controller for Path Following in Non-Holonomic Mobile Robots . In *43rd International Symposium on Robotics (ISR 2012)*. Taipei, Taiwan, Aug. 29-31, 2013. Page 847-852.
  ISNB/ISSN 978-986-88683-0-4.

- Solanes, J. Ernesto and Armesto, Leopoldo and Tornero, Josep and and Muñoz-Benavent, Pau and Girbés, Vicent. Dual-Rate Non-Linear High Order Holds for Visual Servoing Applications. In *13th Towards Autonomous Robotic Systems (TAROS 2012)*. Bristol, England, Aug. 23, 2012. Page 152-163. ISNB/ISSN 978-3-642-32526-7

- Solanes, J. Ernesto and Armesto, Leopoldo and Tornero, Josep and Girbés, Vicent. Multi-rate Visual Servoing Based on Dual-rat High Order Holds. In *12th Towards Autonomous Robotic Systems (TAROS 2011)*. Shefield, England, August 31, 2011. Page 195-206.
  ISNB/ISSN 978-3-642-23231-2

In addition to these contributions, it has been collaborating with other colleges in their research and projects, giving rise to be co-author in several journal and conference paper publications.

JCR Journal Papers

- Muñoz-Benavent, Pau and Solanes, J. Ernesto and García Calandín, Luis and Tornero, Josep. PWM and PFM for visual servoing in fully decoupled approaches. In *Robotics and autonomous systems.* Vol. 65-1, Pag. 57-64, 2015. DOI 10.1016/j.robot.2014.11.011

International conference papers and non-indexed journals

- Muñoz-Benavent, Pau and Armesto, Leopoldo and Girbés Vicent and Solanes, J. Ernesto and Dols Ruiz, J. Francisco and Muñoz García, Adolfo and Tornero, Josep. *Advanced Driving Assistance Systems for an Electric Vehicle.* In Advanced Driving Assistance Systems for an Electric Vehicle. Vol. 2-4, Pag. 329-338, 2012.
DOI 10.5875/ausmt.v2i4.169

- Aguirre, Fredy and Muñoz García, Adolfo and Muñoz-Benavent, Pau and Solanes, J. Ernesto and Girbés Vicent and Colomer, Vicente and Armesto, Leopoldo and Tornero, Josep. The complete design of the ORCA300-AUV. In *4th World Maritime Technology Conf. (WMTC 2012).* Saint-Petersburg, Russia, May 29 - June 1, 2012. Page 1-18

- Girbés, Vicent and Armesto, Leopoldo and Tornero, Josep and Solanes, J. Ernesto. Continuous-Curvature Kinematic Control for Path Following Problems. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems.* San Francisco, USA, Sep. 30, 2011. Page 4335-4340. ISNB/ISSN 978-1-61284-456-5

- Girbés, Vicent and Armesto, Leopoldo and Tornero, Josep and Solanes, J. Ernesto. Smooth Kinematic Controller vs. Pure-Pursuit. In *12th Towards Autonomous Robotic Systems (TAROS 2011).* Shefield, England, Aug. 31, 2011. Page 277-288. ISNB/ISSN 978-3-642-23231-2

National conference papers

- Girbés, Vicent and Solanes, J. Ernesto and Armesto, Leopoldo and Tornero, Josep. Proyecto PISALA. Sensorización Inteligente para el Seguimiento de Líneas por Visión Artificial. In *II Workshop de Robotica (ROBOT 2009).* Barcelona, Spain, Nov. 24, 2009. Page 101-106.

In 2013, a research stay of ten months, founded by Generalitat Valenciana with the program VALi+d, was made at the Mechanical Department of the University of California in Berkeley (USA), under the supervision of Professor Masayoshi Tomizuka. In this period of research, several aspects of the Chapter 3, concretely *asynchronous dual-rate high order holds with time delay compensation*, as well as others not included in this work, such as *constrained LQR for visual feedback robot control*, were developed.

Finally, during this thesis, several final undergrad projects have been guided. One of them was awarded with the National OMRON award "Initiative to the Research and Innovation in Automatic Control", closely related to the topic of this Ph.D.

Final Undergrad Projects Guided

- *Soluciones Industriales basadas en Robótica y Visión* by Andreu Colomer Perpiñá. Guided by Josep Tornero Montserrat and J. Ernesto Solanes Galbis. June 2013, ETSID, Technical University of Valencia (Spain)

Final Undergrad Projects Awards

- Premio Omron de "Iniciación a la Investigación e Innovación en Automática" 2012/2013. *Aplicación Industrial de Multi-rate Robot Visual Servoing* by Andreu Colomer Perpiñá. Guided by Josep Tornero Montserrat and J. Ernesto Solanes Galbis

Chapter 2

State of the Art

2.1 Multi-rate Estimation and Control

Digital controllers and computer science have become almost essential in our society. Daily, new devices flood the market, from mobile phones to cars, offering new capabilities to make our life easier and more comfortable.

Industry is not alien to recent technological developments in computer science and digital controllers, which are utilized for controlling almost all mechanical systems such as robots, motors, machine tools etc., because of cost, reliability, flexibility and compactness.

Figure 2.1 shows the block diagram of a general digital control sys-

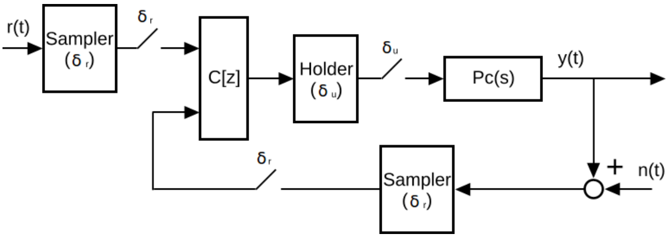


Figure 2.1: Digital control system.

tem, where $\mathbf{P}_c(s)$ is a continuous-time plant to be controlled, $\mathbf{C}(z)$ is a discrete-time controller implemented in digital computer, $\mathbf{d}(t)$ and $\mathbf{n}(t)$ are disturbance and measurement noise, respectively. Because the discrete-time controller has to deal with continuous-time signals in the digital control systems, it needs to have two samplers $\mathbf{S}$ for the reference signal $\mathbf{r}(t)$ and the output $\mathbf{y}(t)$, and one hold $\mathbf{H}$ on the input $\mathbf{u}(t)$.

Therefore, there exist three time periods δ_r, δ_y, δ_u which represent the periods of $\mathbf{r}(t)$, $\mathbf{y}(t)$, and $\mathbf{u}(t)$, respectively. The input period δ_u is generally decided by the speed of the actuator, D/A converter, or the calculation on the CPU. Moreover, the output period δ_y is also determined by the speed of the sensor or the A/D converter.

Current control systems usually hold hardware restrictions on these periods (δ_u and/or δ_y). Moreover, in case of multi-variable systems, there exist many time periods. However, the conventional digital control systems, also known as single-rate control approach, make all periods equal to the longest period for simplification.

On the other hand, the multi-rate sampling control systems have been studied from the point of view both of control theories and practical applications.

2.1.1 Periodic Modelling

A general multi-rate system can be defined by a set of m-inputs and p-outputs, as it is shown in Figure 2.2. The inputs can only be applied in a specific sampled time given by $t^{u_v} = \left\langle t^{u_v}_{r^v_1}, t^{u_v}_{r^v_2}, \ldots, t^{u_v}_{r^v_r} \right\rangle$, with $v = 1, 2, \ldots, m$, being the r-subscript of r^v_r the total number of updates of the m-input. In the same way, the outputs can only be measured in the sampled time given by $t^{y_w} = \left\langle t^{y_w}_{s^w_1}, t^{y_w}_{s^w_2}, \ldots, t^{y_w}_{s^w_s} \right\rangle$, with $w = 1, 2, \ldots, p$, and being the s-subscript of s^w_s the total number of updates of the p-output. Supposing that the input signal is sampled synchronously and considering the system outputs periodic, the frame period is obtained as the least common multiple $\bar{\delta} = lcm \left\langle t^{u_v}_j, t^{y_w} \right\rangle$, and the base period as the greatest common divisor $\delta =$

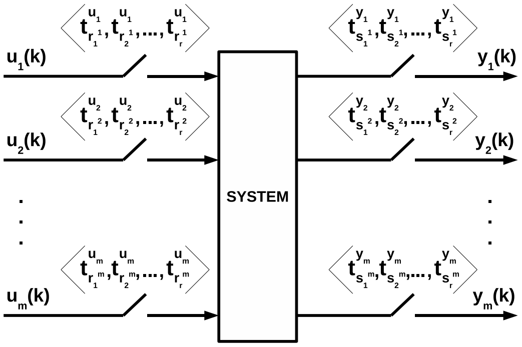

$\left\langle t^{u_1}_{r^1_1}, t^{u_1}_{r^1_2}, \ldots, t^{u_1}_{r^1_r} \right\rangle$
$\left\langle t^{y_1}_{s^1_1}, t^{y_1}_{s^1_2}, \ldots, t^{y_1}_{s^1_r} \right\rangle$
$u_1(k)$
$y_1(k)$
$\left\langle t^{u_2}_{r^2_1}, t^{u_2}_{r^2_2}, \ldots, t^{u_2}_{r^2_r} \right\rangle$
$\left\langle t^{y_2}_{s^2_1}, t^{y_2}_{s^2_2}, \ldots, t^{y_2}_{s^2_r} \right\rangle$
$u_2(k)$
$y_2(k)$
SYSTEM
$\left\langle t^{u_m}_{r^m_1}, t^{u_m}_{r^m_2}, \ldots, t^{u_m}_{r^m_r} \right\rangle$
$\left\langle t^{y_m}_{s^m_1}, t^{y_m}_{s^m_2}, \ldots, t^{y_m}_{s^m_r} \right\rangle$
$u_m(k)$
$y_m(k)$

Figure 2.2: Genear multi-rate scheme.

$gcd \left\langle t^{u_v}_j, t^{y_w} \right\rangle$. The relation between both periods is known as multiplicity number $N = \frac{\bar{\delta}}{\delta}$.

In Armesto and Tornero (2005), the authors provide a matrix based method to model multi-rate MIMO systems taking into account the sampling pattern of each input and output. The method is based on introducing a switching matrix in which its row refers to an input/output and the columns to the moment in which the referred input/output is activated, that is:

$$\mathbf{\Delta_u}(k) = diag\left\{\delta^{u_v}(k), v = 1, 2, \ldots, m\right\} \tag{2.1}$$

$$\mathbf{\Delta_y}(k) = diag\left\{\delta^{y_w}(k), w = 1, 2, \ldots, p\right\} \tag{2.2}$$

$$\delta^{u_v}(k) = \begin{Bmatrix} 1 & mod(k, N) = r^v_j \\ 0 & mod(k, N) \neq r^v_j \end{Bmatrix}, j = 1, 2, \ldots, r^v_r \tag{2.3}$$

$$\delta^{y_w}(k) = \begin{Bmatrix} 1 & mod(k, N) = s^w_i \\ 0 & mod(k, N) \neq s^w_i \end{Bmatrix}, i = 1, 2, \ldots, s^w_s \tag{2.4}$$

with $t^{u_v}_j = r^v_j \cdot \delta$ and $t^{y_w}_i = s^w_i \cdot \delta$.

The main advantage of this modeling technique is the use of a model at base period and therefore is oriented to real-time systems, which requires information processed at high frequency. Using this technique it is possible to design different elements of the control loop easily, such as controllers, holds, samplers, multi-rate observers, etc.

2.1.2 Multi-rate Estimation based on Holds

Historically, the generalized hold is firstly introduced in Chammas and Leondes (1978). In the generalized hold approach, shown in Figure 2.3(a), the control input is generated from $\mathbf{u}(k)$ by:

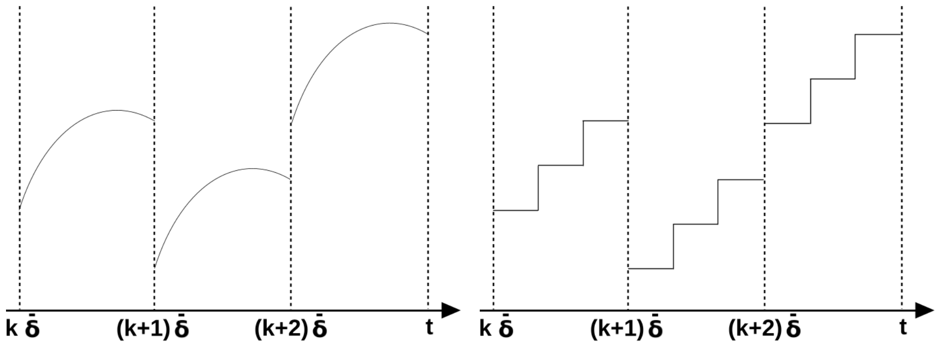
$$\mathbf{u}_h(t) = \sum_{k=0}^{\infty} h\left(t - k\cdot \bar{\delta}\right) \mathbf{u}(k) \tag{2.5}$$

where $h(t)$ is an arbitrary hold function and $\bar{\delta}$ is the frame period. This function is also called the Chammas-Leondes' generalized hold Chammas and Leondes (1978).

Utilizing the hold function as a design parameter, it is possible to assign all poles only by the gain output feedback without the state observer. This approach was extended to feed-forward control in Kabamba (1987), where simultaneous pole assignment, exact model matching, decoupling, and optimal noise rejection are successfully realized.

However, in this method, an arbitrary waveform is assumed to be generated as the hold function $h(t)$. In practice, it is very difficult to make arbitrary hold function because $h(t)$ is generally composed of exponential and sinusoidal functions. Thus, in Chammas and Leondes (1979); Kaczorek (1985); Araki and Hagiwara (1985), the pole assignment method was modified to the multi-rate hold, in which the control input is piecewise constant, as shown in Figure 2.3(b). Because this scheme is easily implemented by ordinary D/A converters, the multi-rate hold is a practical solution of the generalized hold.

Moreover, it is possible to assign not only poles but also zeros by the generalized holds, Kabamba (1987), and multi-rate holds, Araki and Hagi-



(a) Generalized hold.

(b) Controller working at high frequency

Figure 2.3: Multi-rate holds

wara (1985); Moore et al. (1989); Mita et al. (1990). These results have brought great advantages because the discrete-time plant usually has unstable zeros in the conventional single-rate digital control system, and it is impossible to allocate zeros by feedback control.

Other method to obtain practical solution to the generalized hold was introduced by Tornero and Tomizuka (2000); Tornero et al. (1999b). The approach was coined as Dual-rate High Order Holds (DR-HOHs), and can be seen as a generalization of conventional holds such as impulse hold (IH), zero order hold (ZOH), first order hold (FOH), and second order hold (SOH), which are in practice the most commonly used. Table 2.1 shows, for the three most used hold circuits, the continuous transfer function $\mathbf{G}_h(s)$, the discrete transfer function $\mathbf{G}_{h,i}(z^N)$ with the input hold at frame period $\bar{\delta}$, and dual-rate discrete transfer function, $\mathbf{G}_{h,i}(z, z^N)$, and the discrete-time algorithm, being $i = 0, 1, \ldots, N-1$. By following the procedure in Tornero and Tomizuka (2000), other hold circuits and FIR filters can be obtained.

Furthermore, Armesto and Tornero (2003a, 2005) introduced a methodology to generate dual-rate high order holds by using all kind of primitive functions. Moreover in Armesto and Tornero (2005) an attempt to introduce the dynamics of the system into the holder model was described. Due to

HOLD	IH	ZOH	FOH
$\mathbf{G}_h(s)$	1	$\frac{1-e^{-N\cdot\delta\cdot s}}{s}$	$(1-e^{-N\delta s})^2\cdot(\frac{1}{s}+\frac{1}{N\delta s^2})$
$\mathbf{G}_{h,i}(\bar{z})$	1	1	$(\frac{\frac{N+i}{N}\bar{z}-\frac{i}{N}}{\bar{z}})$
$\mathbf{G}_{h,i}(z,\bar{z})$	$\{_{0 \quad iff \quad otherwise}^{N \quad iff \quad i=0}$	$(\frac{\bar{z}-1}{\bar{z}})\cdot(\frac{z}{z-1})$	$(\frac{\bar{z}-1}{\bar{z}})^2\cdot(\frac{z^2-\frac{N-1}{N}\cdot z}{(z-1)^2})$
Algorithm at δ	$\mathbf{u}_h(k,i)=\mathbf{u}(k,0)\cdot\frac{1}{\delta}\cdot\Delta_i$	$\mathbf{u}_h(k,i)=\mathbf{u}(k,0)$	$\mathbf{u}_h(k,i)=\mathbf{u}(k,0)+\frac{i}{N}[\mathbf{u}(k,0)-\mathbf{u}(k-1,0)]$

Table 2.1: Dual-rate High Order holds.

the relevance of such holds along this thesis, below we give a more detailed explanation of this approach.

2.1.2.1 Dual-rate High Order Holds based on Primitive Functions

In Armesto and Tornero (2003a) a general formulation valid for generating DR-HOH circuits based on Lagrange extrapolator, Bezier parametric equations and Taylor series as primitive functions was presented. The concept of primitive function is used to generate the set of signals at base-period (δ) every frame-period ($\bar{\delta}$), Tornero and Tomizuka (2002a) and Armesto and Tornero (2003a).

The primitive function uses the input values inducing an order of function complexity:

$$\mathbf{u}_h(t)=\sum_{l=0}^{n}\mathbf{f}_{n,l}(t,t_k)\cdot\mathbf{u}(t_{k-l}) \tag{2.6}$$

where $t_k=k\bar{\delta}$ is the sampling-time for the last available input and $t_{k-l}=t_l-l\bar{\delta}$ the time previous frame periods. $\mathbf{u}_h(t)$ is the hold output evaluated at time t based on inputs from t_k to t_{k-l}. Therefore, every base period, the output is:

$$\mathbf{u}_h(t_k+i\cdot\delta)=\sum_{l=0}^{n}\mathbf{f}_{n,l}(i\cdot\delta)\cdot\mathbf{u}(t_{k-l}) \tag{2.7}$$

where $i=0,1,\ldots,N-1$. The notation for discrete sequence of input/output values uses double index indicating the indexes for frame and base periods, that is $\mathbf{u}(k,i) = \mathbf{u}(k\cdot\bar{\delta}+i\cdot\delta)$. Therefore the hold can be expressed in discrete time as:

$$\mathbf{u}_h(k,i) = \sum_{l=0}^{n} \mathbf{f}^*_{n,l}(i)\cdot\mathbf{u}(k-l,0) \tag{2.8}$$

with $\mathbf{f}^*_{n,l}(i) = \mathbf{f}_{n,l}(i\cdot\delta)$.

On the other hand, the output for each update can be expressed as:

$$\mathbf{u}_h(t_k + i\cdot\delta) = \sum_{l=0}^{n} \mathbf{f}_{n,l}(i\cdot\delta)\cdot\mathbf{u}(t_{k-l}) \tag{2.9}$$

that can be rewritten as:

$$\mathbf{u}_h((k+1/N)\bar{\delta}) = \sum_{l=0}^{n} \mathbf{f}_{n,l}(i\cdot\delta)\cdot\mathbf{u}((k-l)\bar{\delta}) \tag{2.10}$$

The discrete function for each output is expressed as:

$$\mathbf{u}_h((k+1/N)) = \sum_{l=0}^{n} \mathbf{f}^*_{n,l}(i)\cdot\mathbf{u}((k-l)) \tag{2.11}$$

Applying the lifting technique Bamieh et al. (1991) to the output vector, it takes the form:

$$\bar{\mathbf{U}}_h((k+1/N)) = \begin{bmatrix} \mathbf{u}_h(0) \\ \mathbf{u}_h(1) \\ \dots \\ \mathbf{u}_h(N-1) \end{bmatrix} = \begin{bmatrix} \sum_{l=0}^{n} \mathbf{f}^*_{n,l}(0)\mathbf{u}((k-l)) \\ \sum_{l=0}^{n} \mathbf{f}^*_{n,l}(1)\mathbf{u}((k-l)) \\ \dots \\ \sum_{l=0}^{n} \mathbf{f}^*_{n,l}(N-1)\mathbf{u}((k-l)) \end{bmatrix} \tag{2.12}$$

And its $\mathcal{Z}$-transform is:

$$\bar{\mathbf{U}}_h(z^N) = \bar{\mathbf{G}}_h(z^N)\cdot\bar{\mathbf{U}}(z^N) \tag{2.13}$$

$$\bar{\mathbf{G}}_h(z^N)= \begin{bmatrix} \sum\limits_{l=0}^{n} \mathbf{f}^*_{n,l}(0) z^{N-l} \\ \sum\limits_{l=0}^{n} \mathbf{f}^*_{n,l}(1) z^{N-l} \\ \dots \\ \sum\limits_{l=0}^{n} \mathbf{f}^*_{n,l}(N-1) z^{N-l}) \end{bmatrix} \tag{2.14}$$

In addition, the transfer function of a dual-rate hold can be obtained in the recursive form applying Equations 2.11 and 2.12, obtaining the matrices in estate space representation expressed at frame period:

$$\bar{\mathbf{G}}_h(z^N) = \bar{\mathbf{C}}_h \cdot \left(z^N \mathbf{I} - \bar{\mathbf{A}}_h\right)^{-1} \cdot \bar{\mathbf{B}}_h + \bar{\mathbf{D}}_h \tag{2.15}$$

with,

$$\bar{\mathbf{A}}_h(k) = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \mathbf{I} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & \mathbf{I} & 0 \end{bmatrix} \quad \bar{\mathbf{B}}_h = \begin{bmatrix} \mathbf{I} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.16}$$

$$\bar{\mathbf{C}}_h(k) = \begin{bmatrix} \mathbf{f}^*_{n,1}(0) & \dots & \mathbf{f}^*_{n,l}(0) & 0 \\ \mathbf{f}^*_{n,1}(1) & \dots & \mathbf{f}^*_{n,l}(1) & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{f}^*_{n,1}(N-1) & \dots & \mathbf{f}^*_{n,l}(N-1) & 0 \end{bmatrix} \quad \bar{\mathbf{D}}_h = \begin{bmatrix} \mathbf{f}^*_{n,0}(0) \\ \mathbf{f}^*_{n,0}(1) \\ \vdots \\ \mathbf{f}^*_{n,0}(N-1) \end{bmatrix} \tag{2.17}$$

HOLD	**TRANSFER FUNCTION ENTRIES** $\bar{\mathbf{G}}_h(z^N)$
ZOH	$\mathbf{I}$
FOH **FBOH** **FOTH**	$\frac{\left(\frac{N+i}{N}\right)z^N-\frac{i}{N}}{z^N}\cdot I$
SOH	$\left(\frac{\left(1+\frac{3i}{2N}+\frac{1}{2}\left(\frac{i}{N}\right)^2\right)z^N}{z^N}+\frac{\frac{2i}{N}+\left(\frac{i}{N}\right)^2}{z^N}+\frac{\frac{1}{2}\left(\frac{i}{N}+\left(\frac{i}{N}\right)^2\right)z^{-N}}{z^N}\right)\cdot I$
SOBH	$\left(\frac{\left(1+\frac{i}{2N}\right)^2 z^N}{z^N}+\frac{\frac{i}{N}+\left(\frac{i}{N}\right)^2}{z^N}+\frac{\left(\frac{-i}{2N}\right)^2 z^{-N}}{z^N}\right)\cdot I$
SOTH	$\left(\frac{\left(1+\frac{i}{N}+\frac{1}{2}\left(\frac{i}{N}\right)^2\right)z^N}{z^N}+\frac{\left(1+\frac{i}{N}\right)\left(\frac{-i}{2N}\right)}{z^N}+\frac{\frac{1}{2}\left(\frac{i}{N}\right)^2 z^{-N}}{z^N}\right)\cdot I$

Table 2.2: DR-HOHs based on polynomial functions (transfer function representation).

with the LTI transfer function expressed at frame period:

$$\bar{\mathbf{G}}_h(z^N) = \begin{bmatrix} \mathbf{I} \\ \sum_{l=0}^{n} \mathbf{f}^*_{n,l}(1) \cdot z^{N-l} \\ \vdots \\ \sum_{l=0}^{n} \mathbf{f}^*_{n,l}(N-1) \cdot z^{N-l} \end{bmatrix} \tag{2.18}$$

Tables 2.2 and 2.3 show the DR-HOHs transfer function and discrete algorithm, respectively, by using Lagrange extrapolation (denoted as HOH), Bezier based functions (denoted as HOBH) and Taylor series based functions (denoted as HOTH).

Moreover, in Armesto and Tornero (2005) a state space representation of the DR-HOHs is formulated. This representation can be easily used whether into SISO or MIMO systems. As a resume, a DR-HOH can be formulated

HOLD	DISCRETE ALGORITHM $\mathbf{u}_h(k,i)$
ZOH	$\mathbf{u}(k,0)$
FOH	$\left(\frac{N+i}{N}\right)\cdot\mathbf{u}(k)-\frac{i}{N}\cdot\mathbf{u}((k-1))$
FBOH	
FOTH	
SOH	$\left(1+\frac{3i}{2N}+\frac{1}{2}\left(\frac{i}{N}\right)^2\right)\cdot\mathbf{u}(k,0)+\left(\frac{2i}{N}+\left(\frac{i}{N}\right)^2\right)\cdot\mathbf{u}(k-1,0)+\frac{1}{2}\left(\frac{i}{N}+\left(\frac{i}{N}\right)^2\right)\cdot\mathbf{u}(k-2,0)$
SOBH	$\left(1+\frac{i}{2N}\right)^2\cdot\mathbf{u}(k,0)+2\left(1+\frac{i}{N}\right)\left(\frac{-i}{2N}\right)\cdot\mathbf{u}(k-1,0)+\left(\frac{-i}{2N}\right)^2\cdot\mathbf{u}(k-2,0)$
SOTH	$\left(1+\frac{i}{N}+\frac{1}{2}\left(\frac{i}{N}\right)^2\right)\cdot\mathbf{u}(k,0)+\left(1+\frac{i}{N}\right)\left(\frac{-i}{2N}\right)\cdot\mathbf{u}(k-1,0)+\frac{1}{2}\left(\frac{i}{N}\right)^2\cdot\mathbf{u}(k-2,0)$

Table 2.3: MR-HOH based on polynomial functions (discrete algorithm representation).

as follows,

$$\begin{aligned}\nu_u(k+1)&=\mathbf{A}_h(k)\cdot\nu_u(k)+\mathbf{B}_h(k)\mathbf{u}(k)\\ \mathbf{u}_h(k)&=\mathbf{C}_h(k)\cdot\nu_u(k)+\mathbf{D}_h(k)\cdot\mathbf{u}(k)\end{aligned} \tag{2.19}$$

with,

$$\mathbf{A}_h(k)=\begin{bmatrix} I-\Delta_u(k) & 0 & \dots & 0 \\ \Delta_u(k) & I-\Delta_u(k) & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & \Delta_u(k) & I-\Delta_u(k) \end{bmatrix} \quad \mathbf{B}_h(k)=\begin{bmatrix} \Delta_u(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.20}$$

$$\mathbf{C}_h(k)=\begin{bmatrix} \mathbf{F}_{n,0}(i)\left[I-\Delta_u(k)\right]+\mathbf{F}_{n,1}(i)\Delta_u(k) \\ \mathbf{F}_{n,1}(i)\left[I-\Delta_u(k)\right]+\mathbf{F}_{n,2}(i)\Delta_u(k) \\ \dots \\ \mathbf{F}_{n,l}(i)\left[I-\Delta_u(k)\right] \end{bmatrix}^T \tag{2.21}$$

$$\mathbf{D}_h(k)=\mathbf{F}_{n,0}(i)\Delta_u(k) \tag{2.22}$$

where $\Delta_u(k)$ is defined as in Equation 2.1 and $\mathbf{F}=\{\mathbf{F}_{n,0},\mathbf{F}_{n,1},\dots,\mathbf{F}_{n,n}\}$ is the hold model, which is fully explained in Armesto and Tornero (2005).

2.1.2.2 Dual-rate Dynamic Model based Holds.

It is important to remark that even though holds based on primitive functions have proved a very good performance working on nonlinear systems such as robot manipulators, Solanes et al. (2011), mobile robots, Armesto and Tornero (2004); Armesto et al. (2004), among others, this performance is highly dependent on the primitive function used.

A first attempt to include the system dynamics into the hold prediction model was described in Armesto and Tornero (2005). It is about the use of its dynamic model as tool used by the hold to provide the N-1 prediction samples. In this sense, and if the dynamic model fits with the plant, such predictions may perform better than the ones obtained by using primitive functions.

Without loss of generality, let's use the equation of a zero order hold, which is:

$$\mathbf{G}_h(s) = \left(1 - e^{-NTs}\right) \frac{1}{s} \tag{2.23}$$

As it is said before, the aim of this approach is to include the dynamic model of the system inside the hold model. If $\mathbf{G_p(s)}$ is the transfer function of the plant, then the expression of the hold model is:

$$\mathbf{G}_h(s) = \left(1 - e^{-NTs}\right) \frac{\mathbf{G}_p(s)}{s} \tag{2.24}$$

Now, if a first order dynamic model is utilized, the expression given by Equation 2.24 becomes:

$$\mathbf{G}_h(s) = \left(1 - e^{-NTs}\right) \frac{K}{s(\tau s + 1)} \tag{2.25}$$

where the parameter K refers to the system gain and τ its response time.

Since an implementation point of view is worthy to obtain the time-domain hold expression, as long as it has been done in the case of the dual-rate holds based on primitive functions. Thus, using the inverse Laplace transform to Equation 2.25:

$$
\begin{aligned}
&\mathbf{G}_h(s) = K\cdot\left(1-e^{-N\cdot\delta\cdot s}\right)\cdot\tfrac{1}{s(\tau s+1)} = K\cdot\left[\tfrac{1}{1/\tau}\cdot\tfrac{1/\tau}{s(s+1/\tau)} - \tfrac{e^{-N\cdot T\cdot s}}{s(s+1/\tau)}\right]\\
&\begin{cases}
\mathcal{L}^{-1}\left(\tfrac{1/\tau}{s(s+1/\tau)}\right) \rightarrow (1-e^{-1/\tau t})\cdot u(t)\\
\mathcal{L}^{-1}\left(\tau(\tfrac{e^{-N\cdot\delta\cdot s}}{s} - \tfrac{e^{-N\cdot\delta\cdot s}}{s+1/\tau})\right) \rightarrow \tau(1-e^{-1/\tau(t-N\delta)})u(t-N\cdot\delta)
\end{cases}
\end{aligned}
\tag{2.26}
$$

$$
\mathbf{u}_h(k\cdot N\cdot\delta+t) = K\cdot\Big((1-e^{-1/\tau t})\cdot u(t) - \tau(1-e^{-1/\tau(t-k\cdot N\delta)})\cdot u(t-k\cdot N\cdot\delta)\Big) \tag{2.27}
$$

In Armesto and Tornero (2005) several examples are shown demonstrating their properties using linear examples.

2.1.3 Multi-rate Control

The multi-rate estimation techniques seen along this chapter can be used to design new multi-rate controllers, Fujimoto et al. (1999); Albertos et al. (1990). In this regard, the dual-rate high order holds shown in Section 2.1.2 were used in Tornero and Tomizuka (2000); Tornero et al. (2001b) to obtain equivalent dual-rate controllers. Along this thesis, several examples will demonstrate how the discretization of controllers based on dual-rate high order holds improve the performance of the overall system, getting back most of the continuous controller properties.

In general, dual-rate controllers are composed by three different elements: a discrete controller working at high frequency, a dual-rate high order hold, and a discrete controller working at low frequency. In this sense, dual-rate hold circuit has two main functions: 1) being the interface in between both frequencies, the high and the low one; 2) provide samples at the instants where there is not measured data available.

The general expression of a multi-rate controller is given by:

$$
\left[\bar{\mathbf{U}}_0(z^N) \quad \ldots \quad \bar{\mathbf{U}}_{N-1}(z^N)\right]^T = \left[\bar{\mathbf{G}}_{c,DR,0}(z^N) \quad \ldots \quad \bar{\mathbf{G}}_{c,DR,N-1}(z^N)\right]^T \cdot \bar{\mathbf{E}}(z^N) \tag{2.28}
$$

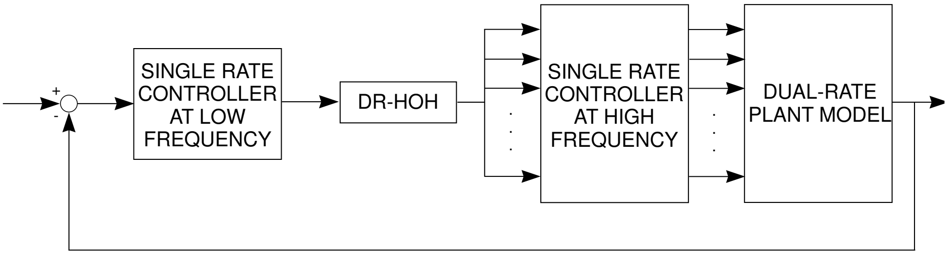

+
-
SINGLE RATE CONTROLLER AT LOW FREQUENCY
DR-HOH
SINGLE RATE CONTROLLER AT HIGH FREQUENCY
DUAL-RATE PLANT MODEL

Figure 2.4: Multi-rate control structure.

where,

$$\begin{bmatrix} \bar{\mathbf{G}}_{c,DR,0}(z^N) \\ \vdots \\ \bar{\mathbf{G}}_{c,DR,N-1}(z^N) \end{bmatrix} =$$

$$= \begin{bmatrix} \mathbf{G}_{c,HF,0,0}(z) & \dots & \mathbf{G}_{c,HF,0,N-1}(z) \\ \vdots & \ddots & \vdots \\ \mathbf{G}_{c,HF,N-1,0}(z) & \dots & \mathbf{G}_{c,HF,N-1,N-1}(z) \end{bmatrix} \cdot \begin{bmatrix} \bar{\mathbf{G}}_{h,0}(z^N) \\ \vdots \\ \bar{\mathbf{G}}_{h,N-1}(z^N) \end{bmatrix} \cdot \bar{\mathbf{G}}_{c,LF}(z^N) \tag{2.29}$$

being $\bar{\mathbf{G}}_{c,DR,0}(z^N)$, the dual-rate controller transfer function, $\mathbf{G}_{c,HF}(z)$ the lifted model of the controller transfer function at high frequency, $\bar{\mathbf{G}}_{h,0}(z^N)$ the hold transfer function, and $\bar{\mathbf{G}}_{c,LF}(z^N)$ the low frequency controller transfer function. The general scheme of this approach can be seen in Figure 2.4.

In the following, among the vast variety of controllers, multi-rate PID and LQR versions are going to be described due to their use along this work.

2.1.3.1 Multi-rate PID

In Tornero et al. (2001b) was presented the multi-rate PID controller. This work assumes that a continuous PID controller is already designed in order to meet continuous time-domain and/or frequency-domain specifications. Then, authors take the controller described in a multiplicative form:

$$G_{PID}(s) = K_p \cdot \underbrace{\left(1 + \frac{1}{\tau_i s}\right)}_{PI} \cdot \underbrace{(1 + \tau_d s)}_{PD} \tag{2.30}$$

where K_p is the proportional gain, and τ_i and τ_d are the integral and derivative time-constants, respectively. It is possible to implement a dual-rate PID controller separating the dynamics in two parts, Tornero et al. (2001b): a PI at low frequency, a PD at high frequency and a DR-HOH interfacing both controllers.

$$\mathbf{G}_{PID}(z, z^N) = \underbrace{K_p \cdot \frac{z^N - \left(1 - \frac{N\delta}{\tau_i}\right)}{z^N - 1}}_{\text{PI controller at low freq.}} \cdot \underbrace{\mathbf{G}_h(z, z^N)}_{\text{DR-HOH}} \cdot \underbrace{\frac{z - \frac{\tau_d/\delta}{1+\tau_d/\delta}}{z}}_{\text{PD controller at high freq.}} \tag{2.31}$$

Some experiments using linear systems demonstrate the improvements of such a dual-rate controller, not only with respect to its equivalent low frequency, but with respect to the scheme based on a dual-rate hold and a PID designed at high frequency.

2.1.3.2 Multi-rate Linear Quadratic Regulator

In Colaneri and de Nicolao (1995); Tornero et al. (1999a, 2001a), authors present the multi-rate version of optimal controllers such as LQG. Given the following stochastic system defined in the continuous-time domain:

$$\begin{aligned} \mathbf{x}(s+1) &= \mathbf{A}_c \cdot \mathbf{x}(s) + \mathbf{B}_c \cdot \mathbf{u}(s) \\ \mathbf{y}(s) &= \mathbf{C}_c \cdot \mathbf{x}(s) + \mathbf{D}_c \cdot \mathbf{u}(s) \end{aligned} \tag{2.32}$$

being the equivalent discrete-time domain system at δ defined as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_\delta \cdot \mathbf{x}(k) + \mathbf{B}_\delta \cdot \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_\delta \cdot \mathbf{x}(k) + \mathbf{D}_\delta \cdot \mathbf{u}(k) \end{aligned} \tag{2.33}$$

Assuming the following single-rate index:

$$J_{SR} = \frac{1}{2}\sum_{k=0}^{\infty} \begin{bmatrix}\mathbf{x}^T(k) & \mathbf{u}^T(k)\end{bmatrix} \begin{bmatrix} \overset{*}{\mathbf{Q}} & \overset{*}{\mathbf{M}} \\ \overset{*}{\mathbf{M}}^T & \overset{*}{\mathbf{R}} \end{bmatrix} \begin{bmatrix}\mathbf{x}(k) \\ \mathbf{u}(k)\end{bmatrix} \tag{2.34}$$

where $\mathbf{x}(k)$ is the system state vector, $\mathbf{u}(k)$ the system input vector and $\overset{*}{\mathbf{Q}}$, $\overset{*}{\mathbf{R}}$ y $\overset{*}{\mathbf{M}}$ have been obtained from the equivalent continuous index:

$$J_c = \frac{1}{2}\int_0^{\infty} \begin{bmatrix}\mathbf{x}^T(t) & \mathbf{u}^T(t)\end{bmatrix} \cdot \begin{bmatrix}\mathbf{Q}_c & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_c\end{bmatrix} \cdot \begin{bmatrix}\mathbf{x}(t) \\ \mathbf{u}(t)\end{bmatrix} dt \tag{2.35}$$

with:

$$\begin{aligned} \overset{*}{\mathbf{Q}} &= \int_0^{\bar{\delta}} e^{\mathbf{A}_c^T t}\mathbf{Q}_c e^{\mathbf{A}_c t} \\ \overset{*}{\mathbf{M}} &= \int_0^{\bar{\delta}} e^{\mathbf{A}_c^T t}\mathbf{Q}_c \left[\int_0^t e^{\mathbf{A}_c \tau \mathbf{B}_c} d\tau\right] dt \\ \overset{*}{\mathbf{R}} &= \int_0^{\bar{\delta}} \left[\int_0^t \mathbf{B}_c^T e^{\mathbf{A}_c^T \tau} d\tau\right] \mathbf{Q}_c \left[\int_0^t e^{\mathbf{A}_c \tau \mathbf{B}_c} d\tau\right] dt + \mathbf{R}_c \bar{\delta} \end{aligned} \tag{2.36}$$

In order to obtain the equivalent multi-rate index, an estimator like interface is required. In this regard, in Tornero et al. (1999a, 2001a) dual-rate high order holds where used for this purpose. Thus, the multi-rate index is:

$$\begin{aligned} J_{MR} &= \tfrac{1}{2}\sum_{k=0}^{\infty} \begin{bmatrix}\mathbf{x}(k) & \mathbf{u}_h(k)\end{bmatrix} \begin{bmatrix} \overset{*}{\mathbf{Q}} & \overset{*}{\mathbf{M}} \\ \overset{*}{\mathbf{M}}^T & \overset{*}{\mathbf{R}} \end{bmatrix} \begin{bmatrix}\mathbf{x}(k) \\ \mathbf{u}_h(k)\end{bmatrix} = \\ &\tfrac{1}{2}\sum_{k=0}^{\infty} \begin{bmatrix}\mathbf{x}_{MR}^T(k) & \mathbf{u}^T(k)\end{bmatrix} \begin{bmatrix}\mathbf{Q}_{MR}(k) & \mathbf{M}_{MR}(k) \\ \mathbf{M}_{MR}^T(k) & \mathbf{R}_{MR}(k)\end{bmatrix} \begin{bmatrix}\mathbf{x}_{MR}(k) \\ \mathbf{u}(k)\end{bmatrix} \end{aligned} \tag{2.37}$$

where:

$$\begin{aligned} \mathbf{Q}_{MR}(k) &= \begin{bmatrix} \overset{*}{\mathbf{Q}} & \overset{*}{\mathbf{M}}\,\mathbf{C}_\delta(k) \\ \mathbf{C}_\delta^T(k)\,\overset{*}{\mathbf{M}} & \mathbf{C}_\delta^T(k)\,\overset{*}{R}\,\mathbf{C}_\delta(k) \end{bmatrix} \quad \mathbf{M}_{MR}(k) = \begin{bmatrix} \overset{*}{\mathbf{M}}\,\mathbf{D}_h(k) \\ \mathbf{C}_\delta^T(k)\,\overset{*}{\mathbf{R}}\,\mathbf{C}_\delta(k) \end{bmatrix} \\ \mathbf{R}_{MR}(k) &= \mathbf{D}_h^T(k)\,\overset{*}{\mathbf{R}}\,\mathbf{D}_h(k) \end{aligned} \tag{2.38}$$

Finally, by using the Riccati equation, the optimal multi-rate control is obtained as follows:

$$
\begin{aligned}
\overset{*}{\mathbf{A}}_{MR}(k) &= \mathbf{A}_{MR}(k) - \mathbf{B}_{MR}(k)\mathbf{R}_{MR}^{-1}(k)\mathbf{M}_{MR}^{T}(k) \\
\mathbf{S}_{MR}(k) &= \overset{*}{\mathbf{A}}_{MR}(k)\mathbf{S}_{MR}(k-1)\left[\overset{*}{\mathbf{A}}_{MR}(k) + \mathbf{K}_{MR}(k)\right] + \mathbf{Q}_{MR}(k) - \\
&\quad - \mathbf{M}_{MR}(k)\mathbf{R}_{MR}^{-1}(k)\mathbf{M}_{MR}^{T}(k) \\
\mathbf{K}_{MR} &= \left[\mathbf{R}_{MR}(k) + \mathbf{B}_{MR}^{T}(k)\mathbf{S}_{MR}(k+1)\mathbf{B}_{MR}(k)\right]^{-1} \cdot \\
&\quad \cdot \mathbf{B}_{MR}(k)\mathbf{S}_{MR}(k+1)\overset{*}{\mathbf{A}}_{MR}(k) \\
\mathbf{u}(k) &= -\left(\mathbf{R}_{MR}^{-1}(k)\mathbf{M}_{MR}^{T} + \mathbf{K}_{MR}(k)\right)\mathbf{x}(k)
\end{aligned}
\tag{2.39}
$$

On the other hand, it is possible to obtain the equivalent multi-rate lifted LQR to the single-rate one by using also dual-rate high order holds like interface, Armesto and Tornero (2005, 2006). Thus, let us define the multi-rate lifted index as:

$$
\bar{J}_{MR} = \frac{1}{2}\sum_{k=0}^{\infty}\begin{bmatrix}\bar{\mathbf{x}}_{MR}^{T}(kN) & \bar{\mathbf{u}}^{T}(kN)\end{bmatrix}\begin{bmatrix}\bar{\mathbf{Q}}_{MR} & \bar{\mathbf{M}}_{MR} \\ \bar{\mathbf{M}}_{MR}^{T} & \bar{\mathbf{R}}_{MR}\end{bmatrix}\begin{bmatrix}\bar{\mathbf{x}}_{MR}(kN) \\ \bar{\mathbf{u}}(kN)\end{bmatrix}
\tag{2.40}
$$

with:

$$
\begin{aligned}
\bar{\mathbf{Q}}_{MR} &= \overset{\bar{*}}{\mathbf{A}}_{\delta}^{\;T}\,\overset{\bar{*}}{\mathbf{Q}}\overset{\bar{*}}{\mathbf{A}}_{\delta} \\
\bar{\mathbf{M}}_{MR} &= \begin{bmatrix}\overset{\bar{*}}{\mathbf{A}}_{\delta}^{\;T}\,\overset{\bar{*}}{\mathbf{M}}\overset{\bar{*}}{\mathbf{B}}_{\delta}\bar{\mathbf{C}}_{h} & \overset{\bar{*}}{\mathbf{A}}_{\delta}^{\;T}\,\overset{\bar{*}}{\mathbf{M}}\overset{\bar{*}}{\mathbf{B}}_{\delta}\bar{\mathbf{D}}_{h}\end{bmatrix} \\
\bar{\mathbf{R}}_{MR} &= \begin{bmatrix}\bar{\mathbf{C}}_{h}^{T}\overset{\bar{*}}{\mathbf{B}}_{\delta}^{\;T}\,\overset{\bar{*}}{\mathbf{R}}\overset{\bar{*}}{\mathbf{B}}_{\delta}\bar{\mathbf{C}}_{h} & \bar{\mathbf{C}}_{h}^{T}\overset{\bar{*}}{\mathbf{B}}_{\delta}^{\;T}\,\overset{\bar{*}}{\mathbf{R}}\overset{\bar{*}}{\mathbf{B}}_{\delta}\bar{\mathbf{D}}_{h} \\ \bar{\mathbf{D}}_{h}^{T}\overset{\bar{*}}{\mathbf{B}}_{\delta}^{\;T}\,\overset{\bar{*}}{\mathbf{R}}\overset{\bar{*}}{\mathbf{B}}_{\delta}\bar{\mathbf{C}}_{h} & \bar{\mathbf{D}}_{h}^{T}\overset{\bar{*}}{\mathbf{B}}_{\delta}^{\;T}\,\overset{\bar{*}}{\mathbf{R}}\overset{\bar{*}}{\mathbf{B}}_{\delta}\bar{\mathbf{D}}_{h}\end{bmatrix}
\end{aligned}
\tag{2.41}
$$

and:

$$
\begin{aligned}
&\bar{\overset{*}{\mathbf{Q}}} = diag\left\{\overset{*}{\mathbf{Q}}, \dots, \overset{*}{\mathbf{Q}}\right\} \qquad \bar{\overset{*}{\mathbf{M}}} = diag\left\{\overset{*}{\mathbf{M}}, \dots, \overset{*}{\mathbf{M}}\right\}\\
&\bar{\overset{*}{\mathbf{R}}} = diag\left\{\overset{*}{\mathbf{R}}, \dots, \overset{*}{\mathbf{R}}\right\}\\
&\quad \bar{\overset{*}{\mathbf{A}}}_\delta = \begin{bmatrix} I \\ \mathbf{A}_\delta \\ \vdots \\ \mathbf{A}_\delta^N \end{bmatrix} \qquad \bar{\overset{*}{\mathbf{B}}}_\delta = \begin{bmatrix} 0 & \dots & 0 & 0 \\ \mathbf{B}_\delta & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{A}_\delta^{N-1}\mathbf{B}_\delta & \dots & \mathbf{B}_\delta & 0 \end{bmatrix}
\end{aligned}
\tag{2.42}
$$

Finally, by using the Riccati equation, the optimal multi-rate control is obtain as follows:

$$
\begin{aligned}
\bar{\mathbf{S}}_{MR}(kN) = \bar{\mathbf{Q}}_{MR} - \bar{\mathbf{M}}_{MR}\bar{\mathbf{R}}_{MR}^{-1}\bar{\mathbf{M}}_{MR}^T + \left[\bar{\mathbf{A}}_{MR}^T - \bar{\mathbf{M}}_{MR}^T\bar{\mathbf{R}}_{MR}^{-1}\bar{\mathbf{B}}_{MR}^T\right]\\
\bar{\mathbf{S}}_{MR}(kN+N)\left[\bar{\mathbf{A}}_{MR} - \bar{\mathbf{B}}_{MR}^T\left(\bar{\mathbf{R}}_{MR}^{-1}\bar{\mathbf{M}}_{MR}^T + \bar{k}_{MR}(kN)\right)\right]\\
\bar{\mathbf{K}}_{MR}(kN) = \left[\bar{\mathbf{R}}_{MR} + \bar{\mathbf{B}}_{MR}^T\bar{\mathbf{S}}_{MR}(kN+N)\bar{\mathbf{B}}_{MR}\right]^{-1}\\
\bar{\mathbf{B}}^T\bar{\mathbf{S}}_{MR}(kN+N)\left(\bar{\mathbf{A}}_{MR} - \bar{\mathbf{B}}_{MR}\bar{\mathbf{R}}_{MR}^{-1}\bar{\mathbf{M}}_{MR}^T\right)\\
\bar{\mathbf{u}}(kN+N) = -\left(\bar{\mathbf{R}}_{MR}^{-1}\bar{\mathbf{M}}_{MR}^T + \bar{\mathbf{K}}_{MR}(kN)\right)\bar{\mathbf{x}}(kN)
\end{aligned}
\tag{2.43}
$$

2.2 Nonlinear Optimal Control

Optimal control theory has well developed tools for optimizing a number of performance indexes that embody desirable objectives for time-invariant systems. In contrast, nonlinear optimal control (optimization constrained by a nonlinear dynamical system) is still a developing field. While its roots were laid down in the 50's with the introduction of dynamic programming (leading to Hamilton-Jacobi-Bellman partial differential equations, Bellman (1954)) and the Pontryagin maximum principle (a generalization of the Euler-Lagrange equations deriving from the calculus of variations, Pontryagin (1959)), these were more theoretical contributions rather than practical design techniques.

From these beginnings, numerous design methodologies for nonlinear optimal control have been developed, often following different paths and techniques. Today, it appears as a fragmented field. The nonlinear optimal control problem is now attacked on many different fronts: by extending the linear theory, utilizing generalizations of the Lyapunov methodology, and brute force computation, to name a few.

In the following we described three different approaches: receding horizon control, iterative linear quadratic regulator and extended Rauch-Tung-Striebel controller.

2.2.1 Receding Horizon Control

Receding horizon, moving horizon, or model predictive control places importance on optimal performance, Kwon and Pearson (1977); Mayne and Michalska (1990); Camacho and Bordons (2004). These techniques apply a receding horizon implementation in an attempt to approximately solve the optimal control problem through on-line computation. The receding horizon methodology is to solve a trajectory optimization emanating from the current state, and implement the resulting open-loop solution until a new state update is received and the process is repeated.

In Sauvee et al. (2006), other image based visual servoing scheme based on nonlinear model predictive control was proposed considering the direct dynamic model of the robot, its joint and torque limits, the camera projection model and the visibility constraint. The authors exhibit the efficiency and the robustness of the proposed solution to control a 6 degrees of freedom mechanical system under simulation scenario.

In Allibert et al. (2010) visual servoing task is formulated into a nonlinear optimization problem in the image plane. The authors claim that with predictive control it is easy to take into account 2D and 3D constraints. The approach was proved in simulation scenario with a 6 degrees of freedom (DOF) free-flying camera, highlighting the capabilities and the efficiency of this approach by a comparison with the classical image-based visual servoing.

In Max and Lantos (2014), approximately time-optimal control of automatically driven cars modeled with gear shift as discrete control input beside the continuous ones in a path between two path boundaries is adresed using model predictive control. The path boundaries are defined by their corner points that may be the result of image processing in real time or prescribed in advance. It is divided into sections for which separate optimum control problems are solved in a nonlinear moving horizon predictive control fashion increasing the semi-online character of the approach.

Although receding horizon control has proven quite successful in some nonlinear systems such as the one commented, stability guaranty still being a concern for some time.

2.2.2 Iterative Linear Quadratic Regulator

Iterative linear quadratic regulator (iLQR) method for optimal feedback control of nonlinear systems uses iterative linearization of the nonlinear system around a nominal trajectory, and computes a locally optimal feedback control law via a LQR, Li and Todorov (2004). In the literature it is possible to find out some contributions using iLQR.

In Li and Todorov (2004), the presentation of iLQR method for locally-optimal feedback control of nonlinear dynamical systems was presentei and applied to a musculo-skeletal arm model with 10 state dimensions and 6 controls, and used to compute energy-optimal reaching movements, demonstrating that the new method converged substantially faster and finding slightly better solutions.

The algorithm considers a discrete time nonlinear dynamical system with state variable $\mathbf{x}_k \in \mathcal{R}^{n_x}$ and control $\mathbf{u}_k \in \mathcal{R}^{n_u}$:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \tag{2.44}$$

and the quadratic cost function: $\mathbf{u}_k \in \mathcal{R}^{n_u}$

$$J_0 = \frac{1}{2}(\mathbf{x}_N - \mathbf{x}^*)^T \cdot \mathbf{Q}_f \cdot (\mathbf{x}_N - \mathbf{x}^*)^T + \frac{1}{2}\sum_{K=0}^{N-1}(\mathbf{x}_k{}^T \cdot \mathbf{Q} \cdot \mathbf{x}_k + \mathbf{u}_k^{\cdot}\mathbf{R} \cdot \mathbf{u}_k) \tag{2.45}$$

where $\mathbf{x}_N$ describes the final state (each movement lasts N steps), $\mathbf{x}^*$ is the given target state. The state cost-weighting matrices $\mathbf{Q}$ and $\mathbf{Q}_f$ are symmetric positive semi-definite, the control cost-weighting matrix $\mathbf{R}$ is positive definite. All these matrices are assumed to have proper dimensions. Note that when the true cost is not quadratic, we can still use a quadratic approximation to it around a nominal trajectory.

Each algorithm iteration starts with a nominal control sequence $\mathbf{u}_k$, and a corresponding nominal trajectory $\mathbf{x}_k$ obtained by applying $\mathbf{u}_k$ to the dynamical system in open loop. When good initialization is not available, $\mathbf{u}_k$ is initialized to 0. The iteration produces an improved sequence $\mathbf{u}_k$ by linearizing the system dynamics around $\mathbf{x}_k$, $\mathbf{u}_k$, and solving a modified LQR problem. The process is repeated until convergence is.

In Zhang et al. (2012) an iLQR method for trajectory tracking control of a wheeled mobile robot system was presented. The proposed scheme involves a kinematic model linearization technique, a global trajectory generation algorithm, and trajectory tracking controller design. A lattice planner, which searches over a $3D$ configuration space (x, y, θ), was adopted to generate the global trajectory. The iLQR method is used to design a local trajectory tracking controller. According to the experiment results, the proposal improved the control sequences $(\mathbf{v}, \boldsymbol{\omega})$ iteratively and produced slightly better results.

2.2.3 Extended Rauch-Tung-Striebel Controller

Zima et al. (2013); Armesto et al. (2015) presented a non-iterative linearization-based controller for nonlinear unconstrained systems, coined as Extended Rauch-Tung-Striebel (ERTS). Due to the relation of this work in this thesis, let us describe the basis and algorithm presented in Armesto et al. (2015).

Consider an stochastic nonlinear dynamic system modelled as Markov process with known transition probability depending on the actual state $\mathbf{x}_t$ and the control action $\mathbf{u}_t$:

$$\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) \tag{2.46}$$

For an arbitrary stochastic control given by distribution $\pi_t(\mathbf{u}_t)$, the resulting distribution of $\mathbf{x}_{t+1}$ is

$$\mathbf{x}_{t+1} \sim p_\pi(\mathbf{x}_{t+1}|\mathbf{x}_t) = \int_{\mathbb{R}^{n_u}} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)\pi_t(\mathbf{u}_t)\mathrm{d}\mathbf{u}_t \tag{2.47}$$

Consider also obtaining a *stochastic* controller which optimizes the following expected[1] loss:

$$J(\mathbf{x}_0, \bar{\mathbf{s}}_{0:N}, \pi_{0:N-1}) = \mathrm{E}\left\{ q_N(\mathbf{x}_N, \bar{\mathbf{s}}_N) + \sum_{t=0}^{N-1} l_t(\mathbf{x}_t, \bar{\mathbf{s}}_t, \pi_t) \right\} \tag{2.48}$$

where sequence $\bar{\mathbf{s}}_{0:N} \triangleq \bar{\mathbf{s}}_0, \ldots, \bar{\mathbf{s}}_N$ stands for the desired quantities related to states (in fact, reference trajectories for some outputs), $q_N(\mathbf{x}_N, \bar{\mathbf{s}}_N)$ is an arbitrary function for the final cost and the intermediate loss is $l_t(\mathbf{x}_t, \bar{\mathbf{s}}_t, \pi_t)$.

Similarly as in Todorov (2008), consider a cost function of the form:

$$l_t(\mathbf{x}_t, \bar{\mathbf{s}}_t, \pi_t) = q_t(\mathbf{x}_t, \bar{\mathbf{s}}_t) + \mathrm{KL}\big(p_\pi(\mathbf{x}_{t+1}|\mathbf{x}_t) || \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t)\big) \tag{2.49}$$

where $\bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t)$ is a user-defined *reference dynamics* and $q_t(\mathbf{x}_t, \bar{\mathbf{s}}_t)$ is an arbitrary function over the state satisfying

$$\int_{\mathbb{R}^{n_s}} e^{-q_t(\mathbf{x}_t, \bar{\mathbf{s}}_t)} \mathrm{d}\bar{\mathbf{s}}_t = 1 \tag{2.50}$$

The KL function stands for Kullback-Leibler divergence Kullback and Leibler (1951). As discussed there, such KL term acts as a penalization on the control input.

Under these assumptions, a controller was derived from the duality between optimal control and estimation, established by Todorov in Kullback-Leibler divergence setting. This was achieved by considering a nonlinear model with affine control input $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t) + \mathbf{B}_t(\mathbf{x}_t)\mathbf{u}_t$ for known vector-valued function $\mathbf{f}(\mathbf{x}_t)$, matrix $\mathbf{B}(\mathbf{x}_t)$, and a quadratic index:

$$J = \frac{1}{2}\mathbf{e}_N^T \mathbf{Q}_N \mathbf{e}_N + \frac{1}{2}\sum_{t=0}^{N-1}(\mathbf{e}_t^T \mathbf{Q}_t \mathbf{e}_t + \mathbf{u}_t^T \mathbf{R}_t \mathbf{u}_t) \tag{2.51}$$

[1] the expectation is taken over realizations of the random variables $\mathbf{x}_{1:N}$.

for $\mathbf{e}_t = \bar{\mathbf{s}}_t - \mathbf{h}(\mathbf{x}_t)$ and known $\mathbf{Q}_t$, $\mathbf{R}_t$ and known vector-valued function $\mathbf{h}(\mathbf{x}_t)$.

In Todorov (2008); Kárný (1996) was demonstrated that there is a related *stochastic* KL optimal control problem for an arbitrary *deterministic* quadratic-cost problem which, as a consequence, can be solved optimally via the duality.

In order to accommodate to KL control approach, in Zima et al. (2013); Armesto et al. (2015) a *fictitious* Gaussian noise was added, with variance $\mathbf{V}_t$ and, also, a fictitious target stochastic dynamics was proposed.

Hence, the original dynamics (2.46) is restricted to

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) \triangleq \mathcal{N}(\mathbf{f}(\mathbf{x}_t) + \mathbf{B}(\mathbf{x}_t)\mathbf{u}_t, \mathbf{V}_t) \tag{2.52}$$

and the reference dynamics is set to

$$\bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t) \triangleq \mathcal{N}(\bar{\mathbf{f}}(\mathbf{x}_t), \bar{\mathbf{V}}_t) \tag{2.53}$$

Function $\bar{\mathbf{f}}(\mathbf{x}_t)$, and matrices $\mathbf{V}_t, \bar{\mathbf{V}}_t$ are derived from the cost function to be analogous to the Riccati equation solution Todorov (2008), in particular $\bar{\mathbf{V}}_t = \mathbf{B}(\mathbf{x}_t)\mathbf{R}_t^{-1}\mathbf{B}^T(\mathbf{x}_t)$, and reference function is taken as the original passive dynamics $\bar{\mathbf{f}}(\mathbf{x}_t) = \mathbf{f}(\mathbf{x}_t)$.

The q_t term in (2.49) is stated as the sum of quadratic state-dependent terms

$$q_t(\mathbf{x}_t, \bar{\mathbf{s}}_t) \triangleq \frac{1}{2}(\bar{\mathbf{s}}_t - \mathbf{h}(\mathbf{x}_t))^T\mathbf{Q}_t(\bar{\mathbf{s}}_t - \mathbf{h}(\mathbf{x}_t)) + c_t \tag{2.54}$$

for constant c_t chosen in order to fulfill the restriction $\int_{\mathbb{R}^{n_s}} e^{-q_t(\mathbf{x}_t, \bar{\mathbf{s}}_t)} d\bar{\mathbf{s}}_t = 1$, and the rightmost term in the referred formula is set as the KL divergence term between (2.52) and (2.53). This choice of matrices makes (2.48) identically equal to the quadratic index in (2.51); in the case under consideration, for $\bar{\mathbf{f}}(\mathbf{x}_t) = \mathbf{f}(\mathbf{x}_t)$, which shows the equivalence in such a case between the standard LQR and KL cost solutions.

The estimation problem dual to the previous control problem requires computing the smoothed probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{y}_{t+1:N} = \bar{\mathbf{s}}_{t+1:N})$, $t = 0, \ldots, N-1$, where the desired states $\bar{\mathbf{s}}_{t+1:N}$ from the original control problem are now considered as the observed measurements of an output $\mathbf{y}_t$ from the model in the dual problem defined (using (2.53) and (2.54)) as

$$\mathbf{x}_{t+1} \sim \mathcal{N}(\bar{\mathbf{f}}(\mathbf{x}_t), \bar{\mathbf{V}}_t) \tag{2.55}$$

$$\mathbf{y}_t \sim \mathcal{N}(\mathbf{h}(\mathbf{x}_t), \mathbf{Q}_t^{-1}) \tag{2.56}$$

The control algorithm proposed in Zima et al. (2013); Armesto et al. (2015) uses an approximation of the above conditional probability computed by Rauch-Tung-Striebel (RTS) smoother, Rauch et al. (1965). The RTS smoother is optimal for linear models with Gaussian noise, and uses the following steps, given a known $\mathbf{x}_t$ by assuming:

- forward pass realized by Kalman filter computing $p(\mathbf{x}_\tau|\mathbf{x}_t, \mathbf{y}_{t:\tau} = \bar{\mathbf{s}}_{t:\tau})$ for $\tau = t, \ldots, N$.
- backward pass computing the $p(\mathbf{x}_\tau|\mathbf{x}_t, \mathbf{y}_{t:N} = \bar{\mathbf{s}}_{t:N})$ for $\tau = N - 1, \ldots, t+1$.

The resulting smoothed distribution is a Gaussian one Rauch et al. (1965).

If functions $\bar{\mathbf{f}}(\mathbf{x}_t), \mathbf{h}(\mathbf{x}_t)$ are nonlinear, RTS can be used for the linearized model at each trajectory point; however, optimality of the proposed estimate is no longer guaranteed.

Notwithstanding, this is analogue to the Extended Kalman filter, successfully used in many control and robotic applications. Inspired on that, the nonlinear version of the RTS smoother is denoted as ERTS in Leondes et al. (1970).

Once the estimation problem has been solved by either of the two above proposals, duality indicates that the optimal control action should fulfil $p_\pi(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\hat{\mathbf{x}}_{t+1|t+N}, \mathbf{P}_{t+1|t+N})$, taking the Gaussian result of the smoother. Hence, both mean and variance should be matched, computing a deterministic control $\mathbf{u}_t$ in (2.52) to match the estimate mean,

$$\mathbf{f}(\mathbf{x}_t) + \mathbf{B}(\mathbf{x}_t)\mathbf{u}_t \approx \hat{\mathbf{x}}_{t+1|t+N} \tag{2.57}$$

and setting $\mathbf{V}_t$ equal to the estimated variance (recall that $\mathbf{V}_t$ is a "fictitious" variance as the original problem is deterministic):

$$\mathbf{V}_t = \mathbf{P}_{t+1|t+N} \tag{2.58}$$

As was mentioned above, $\mathbf{V}_t$ is related to the standard LQR backwards-in-time recursion of the Riccati equation solution.

If $\mathbf{f}(\mathbf{x}_t) \equiv \mathbf{A}_t\mathbf{x}_t$ in (2.57), we would be in the *linear* LQR case so (2.57) could be solved for *exact* equality. However, for nonlinear cases we have only an approximation of $p(\mathbf{x}_{t+1|N})$ proposed by the smoother because $\mathbf{A}_t\mathbf{x}_t$ involves some linearization error in the generated system trajectories. Hence, using the left pseudo-inverse of $\mathbf{B}_t \triangleq \mathbf{B}(\mathbf{x}_t)$, denoted as $\boldsymbol{\beta}_t = (\mathbf{B}_t^T\mathbf{B}_t)^{-1}\mathbf{B}_t^T$, we have a control action given by:

$$\mathbf{u}_t = \boldsymbol{\beta}_t(\hat{\mathbf{x}}_{t+1|t+N} - \mathbf{f}(\mathbf{x}_t)) \tag{2.59}$$

as the proposal for the nonlinear control law[2] trying to fit (2.57) in a Least-Squares sense.

From the above considerations, the resulting control algorithm (Algorithm 1) in Zima et al. (2013); Armesto et al. (2015), denoted as Extended Rauch-Tung-Striebel (ERTS) controller, is composed of two parts: 1) computing $\hat{\mathbf{x}}_{t+1|t+N}$ via ERTS smoother, and 2) obtaining the approximation of the optimal control (2.59). The state $\mathbf{x}_t$ is assumed to be known, so the proposed controller is a deterministic state feedback one.

It should be noted that the backward smoothing step does *not* compute matrices $\mathbf{P}_{\tau|t+N}$: expression (2.58) is only needed in the formal problem solution but, in fact, the value of $\mathbf{V}_t$ is not needed to solve (2.57), so its computation is omitted. Also, note that in order to provide a valid solution for the smoothing algorithm, matrix $\mathbf{P}_{\tau+1|\tau}$ must be full-rank, otherwise, its Moore-Penrose pseudo-inverse should be used on line 12. Indeed, the pseudo-inverse would provide zero correction on line 13 of the algorithm in the state directions in which the fictitious noise does have zero variance. From a duality point of view, that amounts to requiring zero control action effect on uncontrollable states, as intuitively expected.

[2] If the KL control problem were set with user-defined $\mathbf{V}_t$ instead of (2.58), a variant of the approach resulting in a probability distribution over control actions would ensue Zima et al. (2013). However, artificially introducing noise in the control action is not relevant in most optimal control applications, so the issue is pursued no further.

Algorithm 1 ERTS Controller

```
*[Initialization]*
$\hat{\mathbf{x}}_{t|t} = \mathbf{x}_t$, $\mathbf{P}_{t|t} = \mathbf{0}$
*[Prediction]*
for $\tau = t+1, \ldots, t+N$ do
    $\hat{\mathbf{x}}_{\tau|\tau-1} = \bar{\mathbf{f}}(\hat{\mathbf{x}}_{\tau-1|\tau-1})$
    *[linearization]*
    $\mathbf{A}_{\tau-1} = \left.\frac{\partial \bar{\mathbf{f}}}{\partial \mathbf{x}_{\tau-1}}\right|_{\mathbf{x}_{\tau-1}=\hat{\mathbf{x}}_{\tau-1|\tau-1}}$
    $\mathbf{H}_\tau = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_\tau}\right|_{\mathbf{x}_\tau=\hat{\mathbf{x}}_{\tau|\tau-1}}$
    $\mathbf{P}_{\tau|\tau-1} = \mathbf{A}_{\tau-1}\mathbf{P}_{\tau-1|\tau-1}\mathbf{A}^T_{\tau-1} + \mathbf{B}_{\tau-1}\mathbf{R}^{-1}_{\tau-1}\mathbf{B}^T_{\tau-1}$
    $\mathbf{K}_\tau = \mathbf{P}_{\tau|\tau-1}\mathbf{H}^T_\tau(\mathbf{H}_\tau\mathbf{P}_{\tau|\tau-1}\mathbf{H}^T_\tau + \mathbf{Q}^{-1}_\tau)^{-1}$
    $\mathbf{P}_{\tau|\tau} = (\mathbf{I} - \mathbf{K}_\tau\mathbf{H}_\tau)\mathbf{P}_{\tau|\tau-1}$
    $\hat{\mathbf{x}}_{\tau|\tau} = \hat{\mathbf{x}}_{\tau|\tau-1} + \mathbf{K}_\tau(\bar{\mathbf{s}}_\tau - \mathbf{h}(\hat{\mathbf{x}}_{\tau|\tau-1}))$
end for
*[Smoothing]*
for $\tau = t+N-1, t+N-2, \ldots, t+1$ do
    $\mathbf{L}_\tau = \mathbf{P}_{\tau|\tau}\mathbf{A}^T_\tau\mathbf{P}^{-1}_{\tau+1|\tau}$
    $\hat{\mathbf{x}}_{\tau|t+N} = \hat{\mathbf{x}}_{\tau|\tau} + \mathbf{L}_\tau(\hat{\mathbf{x}}_{\tau+1|t+N} - \hat{\mathbf{x}}_{\tau+1|\tau})$
end for
*[Control computation]*
$\mathbf{u}_t = (\mathbf{B}^T_t\mathbf{B}_t)^{-1}\mathbf{B}^T_t(\hat{\mathbf{x}}_{t+1|t+N} - \mathbf{f}(\mathbf{x}_t))$
```

2.3 Visual Feedback Control

Visual feedback control, also known as visual servoing, is a field which involves different disciplines such as computer vision, robotics and automatic control. The concept was introduced in the late 80's early 90's in order to control robots thanks to cameras, Weiss et al. (1987); Feddema et al. (1991); Hutchinson et al. (1996); Chaumette and Hutchinson (2006). Regardless the robot or task used for, the goal is to control the robot thanks to visual data gathered by a camera. In order to be successful, the control task must therefore link the motion of the robot with the visual data. It is a well known flexible and robust technique which increase the accuracy and the versatility of a vision-based robotic system.

Although in the beginning the approaches were thought to be applied to industrial manipulators to perform positioning or tracking tasks, during the last decade researchers have been applying visual feedback control to other kind of robots (i.e. wheeled robots, Fomena et al. (2009); Pasteau et al. (2014); Cherubini and Chaumette (2013), under-actuated systems like quadrotors, Grabe et al. (2015); Bourquardez et al. (2009); Ozawa and Chaumette (2011), or underwater vehicles, Lots et al. (2001); Heshmati-Alamdari et al. (2014)), and tasks, (i.e. tele-Echography Li et al. (2012); Krupa et al. (2015)). In fact, there is a wide realm of applications in many robotics domains.

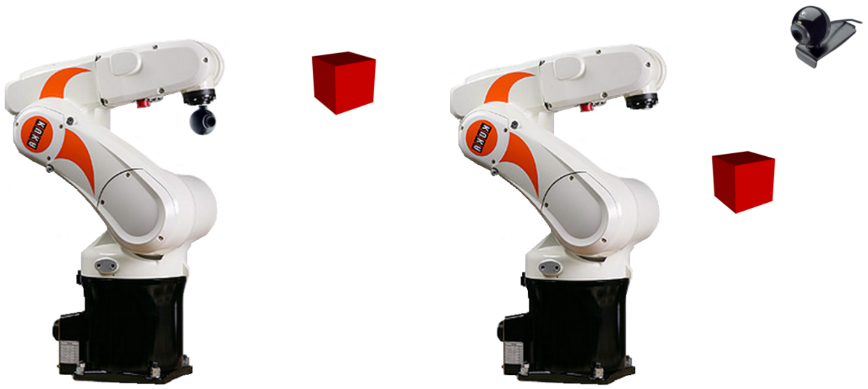
In the following, a review in visual feedback control is presented by focusing on the methods and configurations used along this thesis.

2.3.1 Camera-robot Configurations

Mainly, there exist two configurations to combine camera(s) and robot(s). The first one is called *eye-in-hand*, consisting in camera(s) attached to the robot end-effector (see Figure 2.5(a)). In this configuration, the transformation between camera(s) and the robot frames is constant during all the control task. The second configuration is known as *eye-to-hand*, in which camera(s) is(are) placed somewhere into the workspace looking at the target, the manipulator or both of them (see Figure 2.5(b)). Contrary to the former configuration, in this case the transformation between camera(s) and manipulator frames has to be compute in each control iteration. Hybrid configurations can be constructed when eye-to-hand and eye-in-hand configurations are used, Allen et al. (1993). More details concerning camera(s)/robot configurations can be found in Flandin et al. (2000).

2.3.2 Task Functions in Visual Feedback Control

In visual feedback control a task is defined as the interaction of a robot with its environment using vision sensors. In order to apply motions with respect to a specified reference frame (i.e. camera frame $\mathcal{F}_c$, end-effector



(a) Eye-in-hand

(b) Eye-to-hand

Figure 2.5: Example of camera-robot configurations.

frame $\mathcal{F}_e$, tool frame $\mathcal{F}_t$, manipulated object frame $\mathcal{F}_o$, or robot frame $\mathcal{F}_r$), a transformation between different frames is required to transform the motion from one frame to another, Spong et al. (2006); Dombre and W. (2007). In order to perform a task, features from the target must be extracted from different points of views, depending on the target and the task to be carried out.

2.3.2.1 Camera Space Control

Let us denote $\mathbf{s}^*$ the set of desired features, $\mathbf{s}$ the set of current features, and ${}^c\mathbf{P}_o(t)$ the relative pose between the camera and the object frames at instance t. With this, it can be defined the task function $\mathbf{e}$ as:

$$\mathbf{e} = \mathbf{s}({}^c\mathbf{P}_o(t)) - \mathbf{s}^* \tag{2.60}$$

In this case, the variation of visual features, $\mathbf{s}$, related to the relative movements between the camera and the scene is given by:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial {}^c\mathbf{P}_o} \cdot \frac{\partial {}^c\mathbf{P}_o}{\partial t} + \frac{\partial \mathbf{s}}{\partial t} \tag{2.61}$$

being $\frac{\partial s}{\partial t}$ the variation of $\mathbf{s}$ due to the object own motion. Taking $\frac{\partial \mathbf{s}}{\partial {}^c\mathbf{P}_o} \equiv \mathbf{L_s}$, which is defined in the literature as the Interaction matrix, Hutchinson et al. (1996); Dombre and W. (2007), and $\frac{\partial {}^c\mathbf{P}_o}{\partial t} \equiv \mathbf{v}_c$ the velocity of the camera's frame, obtaining thus the relation used for object tracking tasks:

$$\dot{\mathbf{s}} = \mathbf{L_s} \cdot \mathbf{v}_c + \frac{\partial \mathbf{s}}{\partial t} \tag{2.62}$$

If the target is considered stationary during all the task, then $\frac{\partial \mathbf{s}}{\partial t} = 0$, and the relation used for positioning tasks is:

$$\dot{\mathbf{s}} = \mathbf{L_s} \cdot \mathbf{v}_c \tag{2.63}$$

2.3.2.2 Joints Space Control

In this case, the relation between $\dot{\mathbf{s}}$ and the velocity of the joint variables $\dot{\mathbf{q}}$ is given in Dombre and W. (2007) as follows:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial {}^c\mathbf{P}_e} \cdot \frac{\partial {}^c\mathbf{P}_e}{\partial {}^e\mathbf{P}_o} \cdot \frac{\partial {}^e\mathbf{P}_o}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{s}}{\partial t} \tag{2.64}$$

where ${}^c\mathbf{P}_e(t)$ and ${}^e\mathbf{P}_o(t)$ are the relative pose between the camera and the end-effector frames, and between the end-effector and the object frames, respectively. Taking $\frac{\partial {}^e\mathbf{P}_o}{\partial \mathbf{q}} \equiv {}^e\mathbf{J}_r$, which is the robot Jacobian, and $\frac{\partial {}^c\mathbf{P}_e}{\partial {}^e\mathbf{P}_o} \equiv {}^e\mathbf{J}_r \equiv {}^c\mathbf{V}_e$ coined as twist transformation matrix defined by Dombre and W. (2007):

$${}^c\mathbf{V}_e = \begin{bmatrix} {}^c\mathbf{R}_e & [{}^c\mathbf{t}_e]_\times \cdot {}^c\mathbf{R}_e \\ \mathbf{0}_{3\times3} & {}^c\mathbf{R}_e \end{bmatrix} \tag{2.65}$$

where ${}^c\mathbf{R}_e$ and ${}^c\mathbf{t}_e$ are the rotation matrix and the translation vector from the end-effector frame $\mathcal{F}_e$ to the camera frame $\mathcal{F}_c$. Thus, thus the relation used for tracking object purposes is:

$$\dot{\mathbf{s}} = \mathbf{L_s} \cdot {}^c\mathbf{V}_e \cdot {}^e\mathbf{J}_r \cdot \dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \tag{2.66}$$

If the target is considered stationary during all the task, then $\frac{\partial \mathbf{s}}{\partial t} = 0$ and the relation used for positioning tasks is:

$$\dot{\mathbf{s}} = \mathbf{L_s} \cdot {}^c\mathbf{V}_e \cdot {}^e\mathbf{J}_r \cdot \dot{\mathbf{q}} \tag{2.67}$$

2.3.3 Features Selection

In visual feedback control the set of vision features extracted from the target are the input to the control scheme. A feature can be any part/property of the scene that can be extracted from the image. It is a crucial aspect to select a good set of visual features since the performance and robustness of the controller depends on great measure of it, Janabi-Sharifi and Wilson (1997); J.T Feddema (1991); Chaumette (1998). The number of degrees of freedom (DOF) to be controlled by the employed control scheme determines the minimum number of independent features required.

Three classes of image features are commented in this section: geometric features, luminance/photometric features, and moment features, paying more attention to the former ones, due to their relevance in this work[3].

2.3.3.1 Geometric Features

This approach is based on the geometrical description of elements of a scene, in the $2D$ case, or in relating the robot frame to a target frame in the $3D$ case. Hybrids features (both $2D$ and $3D$ visual features) can also be used to perform visual feedback control tasks. Following, a brief description of the most commonly used geometrical features is provided:

- $2D$ **visual features:** the most common 2D visual features are point coordinates, parameters representing straight lines or ellipses, region of interest, and contours, J.T Feddema (1991); Espiau et al. (1991); Janabi-Sharifi and Wilson (1997); Gans et al. (2003). In the particular case of image points, Cartesian, polar or cylindrical coordinates are generally used, Iwatsuki and Okiyama (2002). In all cases, camera calibration to obtain both intrinsic and extrinsic camera parameters is required, Malis and Cipolla (2002); Pomares et al. (2007).

- $3D$ **visual features:** in this case, object model and image measurements are used to compute or estimate the relative pose between ob-

[3] For a more detailed survey about classes of image features used in visual servoing, see Marey (2010)

ject and camera frames in the Cartesian workspace, or to reconstruct the $3D$ coordinates. In Martinet et al. (1996); Wilson et al. (1996); Deng et al. (2003), the $3D$ coordinates of the points of the object are used as the feature vector. A priori knowledge about the camera calibration parameters is required. In Cervera et al. (2003), $3D$ visual feedback control, orientation in pose vector can be represented by the total format, roll-pitch-yaw or axis-angle formats, Wang and Wilson (1992), or quaternion formulate, Hu et al. (2010).

- **Hybrid visual features:** a mixture composed of both kinds of $2D$ and $3D$ features is presented in Malis et al. (1999); Deng et al. (2002); Cervera et al. (2003); Marchand et al. (2005), and polar and Cartesian parametrizations of image points coordinates are presented in Corke et al. (2009).

- **Gaussian mixture features:** in Abdul Hafez et al. (2008), an approach that removes both features tracking and features correspondence was presented. Collectively features points extracted from the image are modelled as a mixture of Gaussian. Using Lyapunov theory, a control signal is derived to minimize a distance function between the two Gaussian mixtures. The distance function is given in a closed form, and its gradient is efficiently computed and used to control the system. As presented in Abdul Hafez et al. (2008), three degrees of freedom can be controlled using this approach, while six degrees of freedom can be considered under the assumptions that a depth distribution is available.

- **Redundant features:** using redundant features can improve the performance of control task and increase the positioning accuracy by improving the corresponding minimum singular value of the extended image Jacobian, Hashimoto et al. (1996). However, processing a large feature set can sometimes be computationally infeasible.

2.3.3.2 Luminance/Photometric Features

This approach is based on considering the whole image as a feature set defined by pixel intensities. Photometric features does not rely on complex image processing such as feature extraction, matching, and tracking process, contrary to utilizing geometric visual features such as points, straight lines, pose, homography, etc. In addition, it is not very sensitive to partial occlusions and to coarse approximations of the depths required to compute the interaction matrix.

In Kallem et al. (2007), a framework was proposed, in which spatial sampling kernels were used to design feedback controllers for visual servoing. The use of such kernels provided natural hooks for Lyapunov theory, thus unifying tracking and control and providing a framework for optimizing a particular servoing task. The proposed control law is performed in two stages, translation first and then rotation, accomplishing thus the goal task. The proposed method was valid only for 4DOF robots.

In Bakthavatchalam et al. (2013), authors proposed a new type of visual features for visual servoing, coined as photometric moments. The authors claimed that using this set of features it is possible to avoid binary image or spatial segmentation steps, liberating thus the visual servoing process from the crutches of image processing and feature tracking. Bakthavatchalam et al. (2013) also proves that with photometric moments a large convergence domain is obtained. The analytical form of the interaction matrix was also developed in closed form for these features. Results from experiments carried out with photometric moments validated the proposed modelling and control schemes, showing that they performed well for large camera displacements and were endowed with a large convergence domain.

2.3.3.3 Moment Features

Image moments have been widely used in computer vision for decades, especially in pattern recognition and 3D reconstruction. As it is pointed out in Chaumette (2004), image moments provide a generic representation of any object, with simple or complex shapes, that can be segmented in an image.

In Chaumette (2004), the analytical form of the interaction matrix related to any moment that can be computed from segmented images was determined, using Green's theorem derivation method. This result was then applied to classical geometrical primitives. Then, the selection of six combinations of moments to control the 6DOF of the system was described and analyzed. In this work the author assumed a planar object and that the configurations such that the object and the camera planes were parallel at the desired pose. Experimental results demonstrated that a correct behavior of the system was obtained if either a simple symmetrical object or a planar object with complex and unknown shape were considered.

Later, in Liu et al. (2009), a method consisting of an ameliorative image pre-processing algorithm and an improved image based visual servoing (IBVS) using image moments algorithm, were proposed. In this sense, choosing image moments as the image features, there was an increase regarding to the stability of the visual servoing system and in the applied range of objects. In this approach, authors proposed a decoupled Jacobian matrix for controlling the motion of the camera with individual image features, based on this control sequence:

- *Stage 1 - Teaching:* the robot is taught with the desired position of object.

- *Stage 2 - Move the optical axis to the object centroid:* the optical axis is moved onto the object centroid in order to estimate the depth of the object centroid.

- *Stage 3 - Adjust the rotation corresponding to x and y axes:* this stage allows decoupling the Jacobian matrix.

- *Stage 4 - Adjust the object size and orientation:* this final stage completes the task function goal keeping constant image features related to x and y axes (this is possible because their derivation from invariant moments).

Results under simulated scenario showed important improvements in terms of stability.

Then, in Bakthavatchalam et al. (2014), was introduced the concept of tunable visual features for moments based visual servoing schemes. The authors also proposed two metrics for tuning them for optimal behavior: the first metric ensures optimal response of the control to errors in the image space, while the second metric ensures orthogonality between the interaction matrix components (vectors) related to the control of x and y rotational motions. Visual features based on moment invariants were then computed from these optimally tuned shift points. The visual features built from these moment invariants were used to control the x and y rotations in a 6DOF visual servoing control scheme. In short, the "tunable" property of shift points was exploited to solve the important problem of controlling rotational motions around the x and y camera axes. These ideas were validated in both virtual and experimental scenarios.

2.3.4 The Interaction Matrix

The analytical form of the interaction matrix is based on the type of the camera and the projection model used, Hutchinson et al. (1996); Fomena et al. (2009). The most common geometric model is the perspective projection one. In this model, the center of projection is considered at the origin of the camera frame $\mathcal{F}_c$, and the image plane is at f, being f the camera focal length.

In the following, let us obtain the interaction matrix form for the particular case of points like features due to its relevance along this work.

By considering a $3D$ point with coordinates $\mathbf{P} = [X, Y, Z, 1]^T$ in the camera frame $\mathcal{F}_c$, and using a perspective projection model, the point $\mathbf{P}$ is projected on a $2D$ point $\mathbf{p}$, of coordinates $\mathbf{p} = [x, y, 1]^T$ in the image plane, such that:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix} = \begin{bmatrix} (u - c_u)/f\alpha \\ (v - c_v)/f \end{bmatrix} \tag{2.68}$$

where u and v are the image point coordinates in pixel unit, $\mathbf{c} = (c_x, c_y)$ is the coordinates of the principle point, f is the focal length of the camera lens, and α is the ratio of pixel dimension.

By taking the derivative of 2.68, we obtain:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{X}/Z - X\dot{Z}/Z^2 \\ \dot{Y}/Z - Y\dot{Z}/Z^2 \end{bmatrix} \tag{2.69}$$

If the spatial velocity of the camera is given by $\mathbf{v}_c = (\mathbf{v}, \boldsymbol{\omega})$, where $\mathbf{v}$ and $\boldsymbol{\omega}$ are the instantaneous linear and angular velocities of the origin of the camera frame $\mathcal{F}_c$ respectively, then the velocity of a $3D$ point $\mathbf{P}$ related to the camera velocity is defined using the fundamental kinematic equation $\dot{\mathbf{P}} = -\mathbf{v} - \boldsymbol{\omega} \times \mathbf{P}$, such that:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} v_x - \omega_y \cdot Z + \omega_z \cdot Y \\ v_y - \omega_z \cdot X + \omega_x \cdot Z \\ v_z - \omega_x \cdot Y + \omega_y \cdot X \end{bmatrix} \tag{2.70}$$

Injecting the values from 2.70 into 2.69 and grouping we obtain the classical result, Weiss et al. (1987):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & x \cdot y & -(1 + x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1 + y^2 & -x \cdot y & -x \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \tag{2.71}$$

which can be rewritten as,

$$\dot{\mathbf{p}} = \mathbf{L_p} \cdot \mathbf{v} \tag{2.72}$$

where $\mathbf{L_p}$ is the interaction matrix related to $\mathbf{p}$. If there is a set of n features, $\mathbf{s} = \langle \mathbf{p}_1, ..., \mathbf{p}_n \rangle$, the interaction matrix is obtained by staking $\mathbf{L_{p_i}}$ for all $\mathbf{p_i} \in \mathbf{s}$:

$$\mathbf{L_s} = \begin{bmatrix} \mathbf{L_{p_1}} \\ \vdots \\ \mathbf{L_{p_n}} \end{bmatrix} \tag{2.73}$$

2.3.5 Visual Feedback Control Laws

In the control design phase, a number of properties should be considered such as local and global stability, robust behavior with respect to measurement and modelling errors, local or global exponential decrease, order of convergence, absence of local minima and singularities, obtaining suitable robot trajectories, and finally the degree of decoupling between the visual information and the controlled degrees of freedom.

Taking that in consideration, a huge amount of control laws have been proposed along the time: **classical approaches**, 2D and 3D visual feedback control; **enhanced visual feedback control approaches**, such as "*sliding mode control*", Zanne et al. (2000), "*decoupled/partitioned control*", Corke and Hutchinson (2001), "*control based on trajectory planning*", Mezouar and Chaumette (2002); Chesi (2009b), "*Origin-shift in cylindrical coordinates*", Iwatsuki and Okiyama (2005), "*Varying-feature-set control schemes*", Mansard et al. (2009); **Hybrid and switching strategies**, such as "*Hybrid approaches*", Malis et al. (1999); Chaumette and Malis (2000); Malis and Cipolla (2002); Malis and Chaumette (2000); Benhimane and Malis (2007); Kyrki et al. (2004b); Hafez et al. (2007), "*Switching approaches*", Gans and Hutchinson (2003a,b, 2007); Toibero et al. (2009); Deng et al. (2005); Xie et al. (2009), "*Hybrid and/or switching strategies involving other approaches*", Hashimoto and Noritsugu (2000a); Chesi et al. (2004); Taylor and Kleeman (2004); Wang et al. (2010), and on.

In the following, classical 2D and 3D visual feedback control laws are detailed due to their relevance along this thesis.

2.3.5.1 2D Visual Feedback Control

$2D$ visual feedback control, or image based visual servoing (IBVS), uses features extracted directly from the image as input of the control law without any supplementary estimation step. IBVS does not need to estimate the pose at each iteration which helps to provide a robust positioning control against calibration and modelling errors.

IBVS is characterized by several advantages. First of all, when points are used as set of features, this set is controlled directly in the image plane to move the features approximately around straight lines. As a consequence, the features can be constrained in order to keep them within the camera field of view during all the control task, Chaumette (1998). In addition to this, camera calibration and modelling errors do not affect the convergence of the algorithm but the time of such a convergence, Hutchinson et al. (1996); Hager (1997); Mezouar and Chaumette (2002). IBVS is also less sensitive to image noises rather than other approaches.

However, some issues concerned to IBVS must be considered. Firstly, it is hard to predict the trajectory of the end effector and robot may reach its joint limits. Secondly, the end-effector translational and rotational motions are not directly controlled and the usual coupling existing between these motion makes it difficult to plan a pure rotation or a pure translation (i.e. camera retreat problem, Chaumette (1998)). Also, since the system is usually highly coupled, the analytical domain of system stability is hardly reachable in the presence of camera modelling errors. Furthermore, classical IBVS is only locally asymptotically stable, and may fail in the presence of large displacement to be realized, Chaumette (1998); Cervera et al. (2003), which needs a path planning step to split a large displacement up in smaller local movements, Mezouar and Chaumette (2002). Finally, potential failure occurs when IBVS is subject to image singularities or local minima, Chaumette (1998).

In IBVS, providing some information about the depth of the object in the camera frame is usually necessary for computations required to obtain the interaction matrix. Since the stability region for the error in depth estimation is not very large, Malis et al. (2010), it is necessary to accurately estimate the depth. For static objects, this estimation of the depth value can be obtained from the measurement of the current values of the feature points x and y, and their image motion $\dot{x}$ and $\dot{y}$, and of the camera velocity, Matthies et al. (1989); Geraud et al. (1998). The depth parameters of planar and volumetric parametric primitives like points, lines, cylinders, spheres, etc. can be also obtained, Collewet et al. (2008). Another depth estimation method for static points without the explicit need for image motion estimation can be found in De Luca et al. (2008). In Xie et al. (2009), a laser pointer is used and the depth estimation can be achieved through a triangulation method.

2.3.5.2 3D Visual Feedback Control

In $3D$ visual feedback control, or position based visual servoing (PBVS), the task function is expressed in the Cartesian space. The translation and the rotation of the camera are explicitly reconstructed using pose estimation, Hutchinson et al. (1996); Wilson et al. (1996); Martinet et al. (1996); Deng et al. (2003). PBVS is known to have global asymptotic stability referring to the ability of a controller to stabilize the pose of the camera from any initial condition if $3D$ estimation is perfect. The analytical proof is evident if the pose is perfect, otherwise it is very hard to reach it out. When accurate $3D$ estimation is employed, decoupling rotation and translation is obtained. Errors in calibration are propagated to errors in the $3D$ world, so it is required to ensure robustness of PBVS, Kyrki et al. (2004a).

No mechanism in PBVS ensures keeping the features visible within the camera field of view (FOV) when the translation is defined in the desired fixed end-effector frame, Wilson et al. (1996). While, if the translation is expressed in the camera frame, the trajectory in the image plane is improved under large camera rotation motion, and features can be kept in the image plane for small rotation, Deng et al. (2003). In PBVS, the task function to be regulated is usually defined as the error between current and desired poses. The pose can also be selected as the pose of the camera or the end-effector with respect to the object or any other reference frame in the workspace. When the pose between the camera and the object is considered, the task function is given by $\mathbf{e} =^{c*} P_c$. After executing the task, the camera reaches the desired position and the task function $\mathbf{e} =^{c*} P_c * = 0$.

One of the main problems in $3D$ visual feedback control is the determination of the relative position and orientation of the observed object, that is the pose with respect to the camera. For real-time pose estimation of the object, image measurements are combined with the known object CAD description. The pose can be estimated using image points, Haralick et al. (1989); DeMenthon and Davis (1995a); Liu and Wong (1999); Ansar and Daniilidis (2003); Chesi (2009a), using point and line correspondence, Dornaika and Garcia (1999), using point to region correspondence, McInroy and Qi (2008), using curves, Safaee-Rad et al. (1992), or using other different geometrical features as in virtual visual servoing, Marchand and Chaumette (2002). For obtaining more accurate pose estimation, different filters are usually used to estimate its translational and rotational parameters, Ficocelli and Janabi-Sharifi (2001); Lippiello et al. (2004); Shademan and Janabi-Sharifi (2005), and recently Janabi-Sharifi et al. (2011).

In response to the difficulties appearing in IBVS and PBVS, several methods that do not rely solely on the interaction matrix have been devised to improve the behavior of the visual feedback control laws.

2.3.6 Problems in Visual Feedback Control

Selecting a suitable set of visual features and designing good control schemes should be taken into account for avoiding system failures. Usual problems in visual feedback control that are directly influenced by this selection, and can be enhanced by a good selection, are: local minima, singularity and visibility problems.

2.3.6.1 Local Minima

Generally, local minima occurs only with very specific configurations, Chaumette (1998); Gans and Hutchinson (2003a). Getting trapped in a local minima, camera velocity is null $\mathbf{v} = \mathbf{0}$, while the feature errors have not been minimized, such that $\mathbf{s}-\mathbf{s}^* \neq \emptyset \in Ker(\mathbf{L}^+)$. This results in converging to a final pose that is different from the desired one. When $\mathbf{s}$ is composed in three image points and $\mathbf{L_s}$ is full rank, then we have $Ker(\mathbf{L}^+) = 0$, implying that there is no local minima. However, when three points are used, the same image of the three points can be seen from four different camera poses, which means four camera poses exist such that $\mathbf{s} = \mathbf{s}^*$, which corresponds to four global minima. When at least four points are used, unique pose can theoretically be obtained. However, $\dim(\mathbf{L}^+) = 6 \times 6$, implying that dim $Ker(\mathbf{L}^{+}) = 0$. Using four points, the control law tries to enforces 8 constraints on the image trajectory while the system has only six degrees of freedom. In that case, due to the existence of unrealizable motions in the image that may be computed by the control law, a local minima may be reached, Chaumette (1998).

Several control strategies have been used to avoid local minima in visual servoing. For example, and as already said before, in Deng et al. (2005); Gans and Hutchinson (2007), a hybrid motion control strategy that considers the local minima problem is presented while in Mezouar and Chaumette (2002) a path planning strategy is developed.

2.3.6.2 Singularity

When the interaction matrix is singular causing a task singularity, the velocity tends to infinity and the system is unstable. It may become singular if image points are chosen as visual features. For instance, when four points are used and the required camera motion is defined by a pure rotation of 180° around its optical axis, the image trajectory obtained of each point is such that the points move concurrently in a straight line at the principal point, where the interaction matrix is singular, Chaumette (1998). For the considered motion, the choice of image points coordinates is really inadequate. Indeed, if the four points in the image are replaced by cylindrical parameters (ϕ, θ), the singularity can be avoided when the same initial position is used, showing a pure rotation motion around the optical axis of the camera.

In PBVS, Wilson et al. (1996); Martinet et al. (1996); Deng et al. (2003), most of representations avoid the problems of local minima and/or singularities of the corresponding interaction matrices depending on the chosen $\mathbf{e}$. This problem can also be solved by using potential function, Hashimoto and Noritsugu (2000b), partitioning approach, Corke and Hutchinson (2001), switching approach, Deng et al. (2005), hybrid approach, Malis et al. (1999), and PBVS, Wilson et al. (1996).

2.3.6.3 Feature Visibility

Using classical $2D$ and $3D$ visual feedback control and assuming a bad calibration and a large initial camera displacement, the target may leave the camera field of view, Malis et al. (1999); Chesi and Vicino (2004). That is why it is desirable to have servoing controls able to keep features in the camera field of view to obtain reliable feedback signal during the servoing process. To minimize the probability that the object leaves the FOV, a repulsive potential field can be adopted, Chesi and Vicino (2004), a path planning strategy, Mezouar and Chaumette (2002), switching strategies, Gans and Hutchinson (2007), as well as using structure light, Xie et al. (2009).

2.3.7 Multi-rate Visual Feedback Control

Recently, many researches have focused their attention on the sensor latency problem. This problem arises when sensors are much more time-consuming than the sampling periods of the rest of the electrical-mechanical system. Vision sensors are one example of this. Although large amount of information can be extracted using vision sensors, their require long time for image processing.

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.

It is well known how multi-rate control and estimation approaches have improved the performance of many kind of systems. Although many contributions with dual-rate or N-periodic systems, Armesto and Tornero (2003b), can be found using a linear system approach, for N-periodic non-linear systems (i.e. a robot equipped with a vision system) there are very few tools. Yet, there are some attempts of solving the problem of sensor latency for visual feedback controlled systems using multi-rate techniques.

For example, in Usai and Di Giamberardino (2006) a multi-rate strategy, based on motion planing under multi-rate digital control, Monaco and Normand-Cyrot (1992), for the control of a non-holonomic mobile robot using a visual feedback control approach, was presented. Such strategy is based on a mathematical model computed using two virtual image features, which guarantees their presence in the camera field of view during all the robot motion, assuming the robot moves throughout an environment without any obstacles, and features belong to the ground plane. Such a hypothesis allows one to have an estimation of the distance of the 3D features from the camera.

In Fujimoto (2003), problems of sensor latency and disturbances in visual feedback control systems are addressed using inter-sampling disturbance rejection with switching scheme. In the controller, multi-rate inter-sample disturbance rejection algorithm is utilized, which was proposed by authors for general digital control system with restricted sampling frequency. The proposed feed-forward scheme with open-loop estimation and switching function enables the disturbance rejection without any sacrifice of the closed-loop characteristics. Therefore, a new precise formulation of delay problems in visual feedback control was established as the image processing latency, the difference between sampling period of camera signal and control period of joint servo system, and delay of inner-loop joint servo system. The authors demonstrated that by introducing novel multi-loop control schemes and depth identification, the proposed inter-sample disturbance rejection controller becomes applicable to the complicated visual feedback control problem of 6 DOF manipulator with moving object points.

Closely to this work, in Sasajima and Fujimoto (2007) the problem of sensor latency was addressed by separating this latency into two problems. The first problem is that the vision sensor requires long time for image processing, and this delay works as dead-time to feedback control system. The second problem is that the sampling period of vision sensor such as a CCD camera is comparatively long (over 33 ms), while the control period of joint servo is short (less than 1 ms). For the first problem of time delay modelled as $e^{-T_d \cdot s}$, the movement of the object is modelled as output disturbance, and it is estimated and predicted by full-order disturbance observer.

For the second problem about difference between sampling period and control period, this work formulates this issue explicitly by establishing novel precise model of the delay problem. Two multi-rate repetitive controllers were proposed to reject high-order repetitive disturbance. By simulations and experiments, perfect tracking controller with periodic signal generator (RPTC) demonstrated higher disturbance rejection performance than observer based controller (RIDR). RPTC was enable to track moving objects with high frequency element in visual servo system. Moreover, the computation cost of RPTC was superior to RIDR because it could be realized by the periodic signal generator and low order perfect tracking controller.

Multi-rate Kalman filter has also been used as an attempt to overcome the problem of sensor latency, Hollmuler and Harle (1999). In this case the Kalman model estimates both robot and object movements to perform a visual feedback control tracking task.

Chapter 3

Visual Feedback Control based on Dual-rate High Order Holds

3.1 Motivation

Sensors and actuators in robotic applications are based on different technologies, showing different physical properties. In this sense, it is very common to find robots with sensors and actuators working at different sampling rates. This is a problem that arises from inherent technological limitations of each type of sensor, communication channels, processing time, etc. The naive solution to overcome this problem, known as sensor latency problem, is to decrease the overall sampling period to the slowest one. However, it is well known that this approach may decrease the overall system performance, since high frequency dynamics are missed due to the temporal discretization as a consequence of the Nyquist sampling constraint.

Significant improvements can be obtained for robot systems when introducing multi-rate techniques, Chang et al. (2009); Anderson (2007); Reddy et al. (2007). During the last three decades special attention has been paid to dual-rate sampled-data systems, and it is possible to find many contributions dealing with estimating and control of multi-rate systems Tornero and Tomizuka (2002b), but few of them dealing with the design of nonlinear multi-rate estimators or controllers for nonlinear systems, Chang et al. (2009); Ahrens et al. (2009).

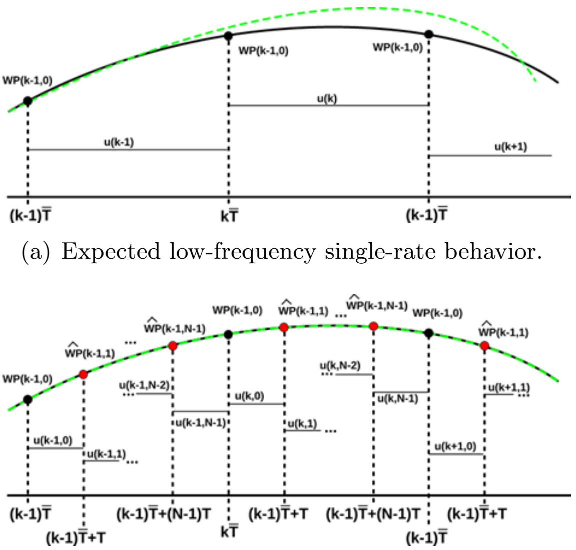
Perhaps, one of the most used tools in estimation is the Kalman filter and its generalization from a multi-rate point of view, the multi-rate Kalman filter, Hollmuler and Harle (1999); Smyth and Wu (2007). This estimation technique uses linear model and well defined covariance matrices in order to perform good estimations. It is well known its robustness dealing with signal noises, although this property depends on grate measure of the covariance matrices sometimes complicated to set up, even more if we deal with complex MIMO nonlinear systems.

Owing to their easy implementation and the good performance shown for signal estimation in the past, this chapter proposes dual-rate high order holds, Armesto and Tornero (2003a, 2005), to control nonlinear systems (i.e. robot manipulators), assuming small noisy signals. Concretely, two different robotic platforms are used along the chapter to study the properties of using such extrapolators: a wheeled robot performing a path following task; a 6 DOF industrial robot manipulator performing error positioning and tracking tasks. Both of them use vision systems to close the control loop.

In addition to that, in this chapter a more general dual-rate high order hold is presented, incorporating time delay compensation within its structure. This is the most general case of hold since signal asynchronies and time delay are considered to perform the inter-sampling estimations. Formulations, as well as the implementation algorithm, are described. This approach is then validated by both simulation and experimental scenarios for the particular case of estimating data from vision systems to control robots.

3.2 Visual Feedback Wheeled Robot Control based on Dual-rate High Order Holds.

The problem of maintaining accurate motion along a specified path is one of the most interesting control tasks in the field of mobile robotics. The



(a) Expected low-frequency single-rate behavior.

(b) Expected dual-rate control behavior.

Figure 3.1: Due to the low-frequency actuation in Figure (a) the robot deviates from the desired path, while the dual-rate control in Figure (b) can potentially correct such deviations at a higher sampling rate. Measured way-points (WP) (black points), estimated WP (red points), ideal behavior (black line) and robot behavior (dashed green line).

accuracy has a direct influence on the maximum velocity that the robot can achieved with a stable control. As in every control approach of sampled-data systems, the sampling time of sensors and actuators has also a great influence on the maximum following speed.

In classical control theory, the sampled period is taken from the slowest element of the system, which in robotics usually comes from sensors (i.e., GPS, vision, etc.). In order to provide a solution to this problem, multi-rate approaches can be used, where the aim is to reproduce, as ideal behavior, the high-frequency single-rate response.

Figure 3.1 shows the advantages of using the knowledge of inter-sampling estimated way-points for controlling a robot. Without inter-sampling knowledge, the robot applies a constant control input that may cause an excessive deviation between the reference path and the robot path. However, if such way-points were accessible at inter-sampling time instants, the controller could provide a better path following performance. This problem arises from designing controllers at too low frequency sampling rates, which could even lead to non-stabilization.

Therefore, the inclusion of dual-rate high order holds for estimating the unknown inter-sampling way-points should improve the overall performance of such applications. In the following, we describe the set-up used for validating our approach.

SUMMIT mobile platform is a robot with Ackerman configuration with limited curvature:

$$\kappa_{max} = \frac{1}{R_{min}} = \frac{\tan\alpha}{L} \tag{3.1}$$

where κ_{max} is the maximum curvature of the robot, R_{min} is the minimum turning radius, α is the steering angle and L is the distance between axles. All the characteristics and interesting information regarding to this platform is in Appendix A Section A.1.2.

Figure 3.2 shows the general proposed dual-rate control diagram based on dual-rate high order holds. Remark that in this case the controller is designed at high frequency, although if the dynamics of the controller allow it, it is possible to implement dual-rate controllers, as it will be proved in Section 3.3 (see Tornero and Tomizuka (2000); Tornero et al. (2001b) for implementation details).

Hence, the controller used is the well-known Pure Pursuit, Morales et al. (2009). This method requires selecting a point along the path that is separated a given look-ahead distance from the robot. The steering angle is proportional to the look-ahead point translated to the reference frame of the robot. In addition, it is used SLAM technique, Grisetti et al. (2007), stereo vision processing and a way-point selection module which provides estimations of the robot pose and valid way-point based on estimation on several sensors, such as IMU (odometry) and vision system.

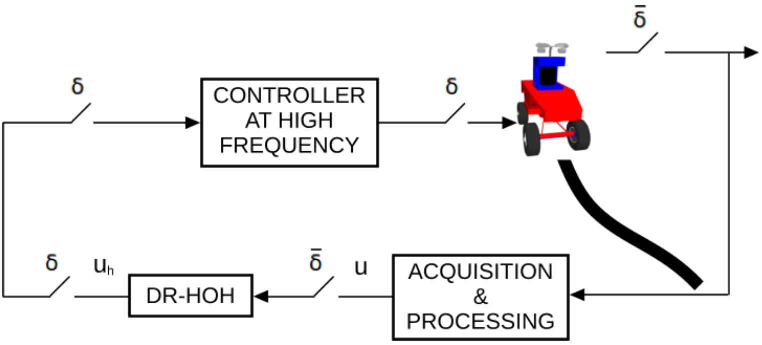


Figure 3.2: Dual-rate control block diagram for SUMMIT platform based on DR-HOHs.

Considering the physical constraints of SUMMIT and the shape of the path to follow (a regular path inscribed into a rectangle of 5.2 by 4.9 meters of size), the following control parameters are chosen: look-ahead (LA) 1.5 m, nominal velocity (v) 1.5 m/s, base-period (control period) 50 ms, (20 Hz) and frame-period (measure/way-points acquisition) $\bar{\delta}=500$ ms (2 Hz).

In the following the validation under simulated scenario of the proposed dual-rate approach is discussed.

A dynamic simulator of the SUMMIT mobile platform was developed for test and validation of the algorithms before being tested into the real SUMMIT. As resume, it uses open source tools such as ROS, GACEBO, ViSP, OpenCV, OROCOS, among others. All these tools allow us to take into account the robot dynamics and wheeled slippage, among much other non-linearity, and simulate sensors and dual-rate control like if we were working directly with the experimental platform (more details can be seen in Appendix A Section A.1.2).

Figure 3.3 shows the different ROS processing nodes and their interaction between them. In this case, the “Acquisition and processing block” of Figure 3.2, is composed by “Image processing”, “Data fusion” and “Way-point selector” nodes. These nodes are in charge to provide the next way-point to follow from image data of a stereo vision system, and the current robot

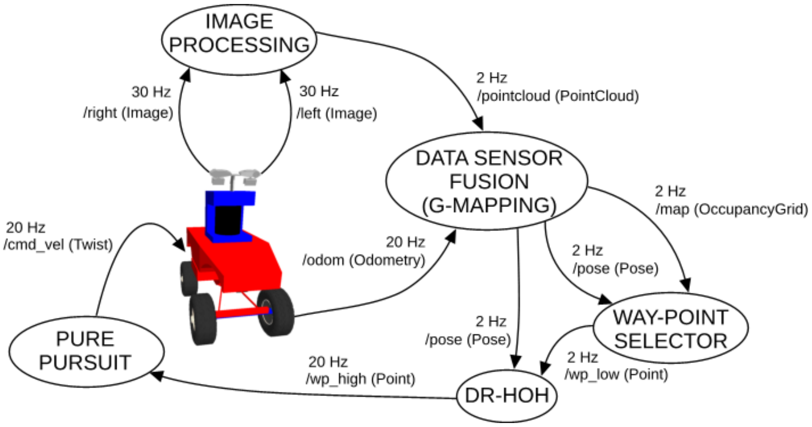
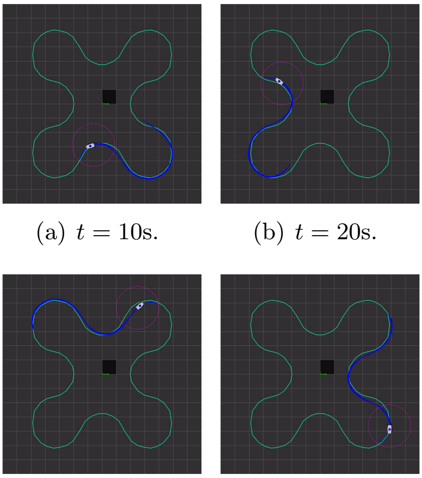


Figure 3.3: Proposed ROS-based software architecture: stereo vision node provides two images in topics (/left) and (/right), stereo processing produces a /pointcloud topic (converted into a laser scan), G-mapping produces an occupancy grid (/map) and the robot pose (/pose), way-points selector determines a valid way-point (/wp_low) at low-frequency, /DR-HOH generates inter-sampling way-points (/wp_high) and pure-pursuit computes appropriate robot command (/cmd_vel).

pose, both signals working at 2 Hz (low sampling rate). The robot control (SUMMIT driver) is performed at a higher sampling rate (20Hz). Here the path and robot pose are assumed to be obtained from real sensors.

The advantage of the analysis under simulation is that it is possible to simulate how the system will behave if it could work at ideal high frequency and thus, quantify the improvements of our approaches. Therefore, Figure 3.4 shows the ideal trajectory performed by SUMMIT robot when the controller can work at high frequency. Compared with the trajectory performed by the robot when its controller works at single-rate low frequency (Figure 3.5), it is clear that its behavior has degraded until the point that the robot may lose the path. On the other hand, using the proposed dual-rate approach based on dual-rate high order holds, the system behaves much better



(a) $t = 10$s.

(b) $t = 20$s.

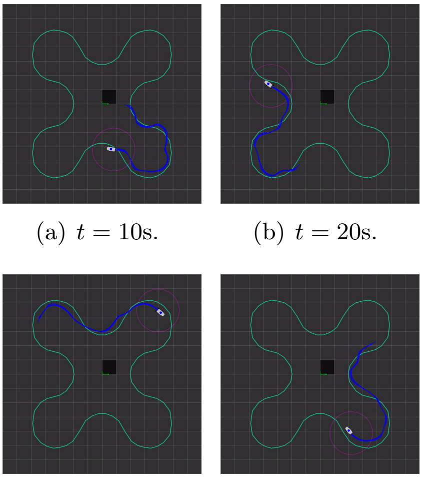
(c) $t = 30$s.

(d) $t = 40$s.

Figure 3.4: SUMMIT mobile robot working at high frequency.

than the classical single-rate working at low frequency, as it is shown in Figure 3.6 when DR-FOHs are used, and Figure 3.7 when DR-SOHs are used. Figure 3.8(a) shows a comparison using the single-rate at high frequency as ideal behavior and comparing this ideal with the result produced by the classic single-rate at low frequency. Similarly, Figure 3.8(e) compares the results of using dual-rate approach with DR-SOH. Certainly, classic single-rate behavior working at low frequency produce a bad performance of the system, even oscillating and losing the path to follow. On the other hand, the dual-rate approach presents a better performance. This can be seen in the control action shown in Figures 3.8(b) and 3.8(f), where even though DR-SOH sometimes produces strong actions, they are more similar than the ones produced by the ideal single-rate control working at high frequency.

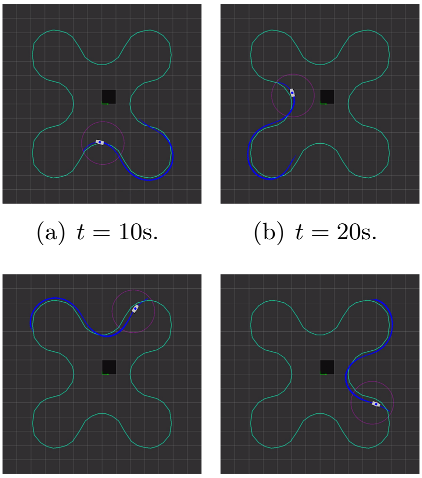
Table 3.1 provides a more accurate analysis based on a benchmarking with different metrics, Yuste et al. (2010). As expected, the average following error and its maximum value are much bigger in the single-rate approach, while using a dual-rate framework improves the performance.



(a) $t = 10$s. (b) $t = 20$s.

(c) $t = 30$s. (d) $t = 40$s.

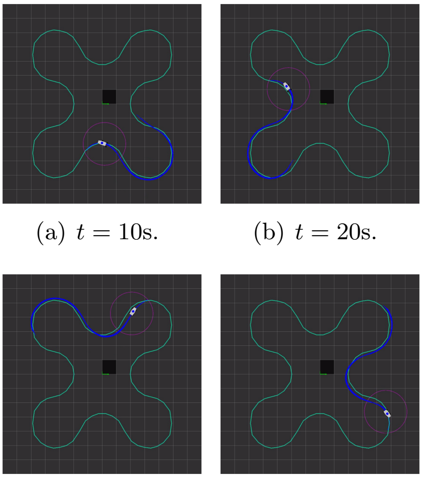
Figure 3.5: SUMMIT mobile robot working at low frequency.



(a) $t = 10$s. (b) $t = 20$s.

(c) $t = 30$s. (d) $t = 40$s.

Figure 3.6: SUMMIT mobile robot using DR-FOHs.



(a) $t = 10$s.

(b) $t = 20$s.

(c) $t = 30$s.

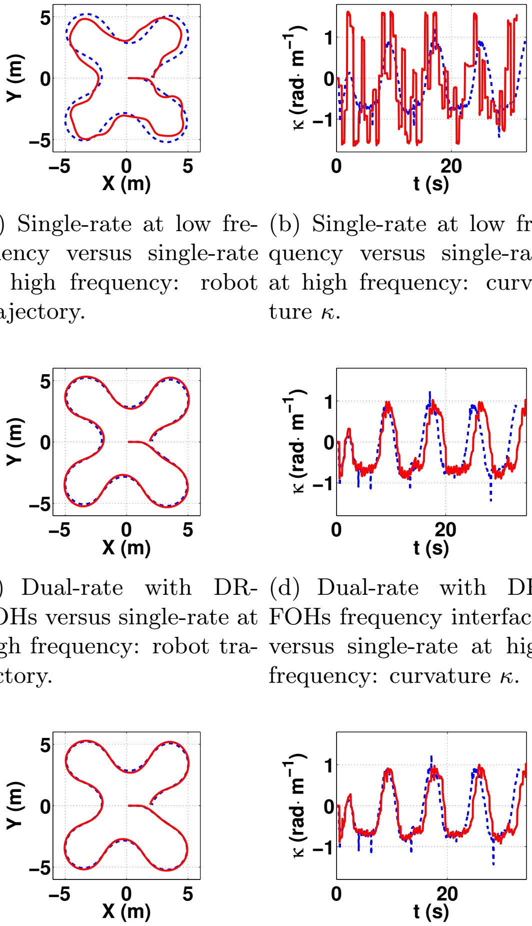
(d) $t = 40$s.

Figure 3.7: SUMMIT mobile robot using DR-SOHs.

FEATURE	**METRIC**	**SR(δ)**	**SR($\bar{\delta}$)**	**DR-FOH**	**DR-SOH**
Average Error [m]	$\frac{1}{N}\sum_t e_t$	0.1458	0.3097	0.1803	0.1585
Maximum Error [m]	$\max_t e_t$	1.8409	1.8751	1.8576	1.8504
Average Curvature [rad·m^{-1}]	$\frac{1}{N}\sum_t \kappa_t$	0.5566	0.8434	0.5752	0.5760
Maximum Curvature [rad·m^{-1}]	$\max_t \lvert\kappa_t\rvert$	1.0050	1.6520	1.0298	1.0233
Average Sharpness [rad·m^{-1}]	$\frac{1}{N}\sum_t \sigma_t$	3.5276	11.4703	4.4502	3.7886
Maximum Sharpness [rad·m^{-1}]	$\max_t \lvert\sigma_t\rvert$	64.4824	288.2119	89.6029	78.9596
Normalized Bending Energy [m^{-1}]	$\frac{1}{N}\sum_t \kappa_t^2 \cdot v_t \Delta t$	0.0282	0.0741	0.0292	0.0289
Normalized Abruptness [rad·m^{-1}·s^{-1}]	$\frac{1}{N}\sum_t \sigma_t^2 \cdot v_t \Delta t$	4.1236	106.5804	5.5357	4.6746

Table 3.1: Metrics of path following comparing Single-rate and Dual-rate approaches.

On the other hand the Bending Energy and the Abruptness demonstrates that the control with a DR-SOH is smoother and needs less energy to follow the path because it does not have to modify the robot direction so many times, due to the fact that the intermediate way-points are estimated more accurately rather than conventional single-rate control.



(a) Single-rate at low frequency versus single-rate at high frequency: robot trajectory.

(b) Single-rate at low frequency versus single-rate at high frequency: curvature κ.

(c) Dual-rate with DR-FOHs versus single-rate at high frequency: robot trajectory.

(d) Dual-rate with DR-FOHs frequency interfaces versus single-rate at high frequency: curvature κ.

(e) Dual-rate with DR-SOHs versus single-rate at high frequency: robot trajectory.

(f) Dual-rate with DR-SOHs versus single-rate at high frequency: curvature κ.

Figure 3.8: Comparison of SUMMIT performance when using conventional and dual-rate approaches: single-rate at low and high frequency and dual-rate second order holds.

3.3 Visual Feedback Robot Manipulator Control based on Dual-rate High Order holds.

In the following, let us assume this notation: ${}^{S_1}\mathbf{M}_{S_2}(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$, where ${}^{S_1}\mathbf{M}_{S_2}$ is the homogeneous matrix relating frame S_2 with respect to frame S_1; X, Y and Z are Cartesian coordinates in meters, and θ_y, θ_y and θ_z are the three Euler angular coordinates in radians. In addition, the world frame is denoted with sub-index w, the camera frame is denoted with sub-index c, the robot end-effector frame is denoted with sub-index e, and the object frame is denoted with sub-index o

The proposed dual-rate approach can be implemented in 2D and 3D workspaces. This section only provides experiments working into the 2D workspace, although the conclusions can be extended to the control on the 3D workspace.

2D visual feedback control schemes (IBVS), Corke (2011); Chaumette and Hutchinson (2006), use the image-plane coordinates of a set of points to define the set $\mathbf{s}_t$. Let us denote $\mathbf{f}_t$ and $\mathbf{f}_t^*$ as the current and reference image state, respectively. $\mathbf{s}_t = \{\mathbf{f}_1, ..., \mathbf{f}_n\}_t \in \Re^{3n}$ is the set of n current state, and $\mathbf{s}_t^* = \{\mathbf{f}_1^*, ..., \mathbf{f}_n^*\}_t \in \Re^{3n}$ the set of n reference state.

Without loss of generality, this section considers features-like-points, $\mathbf{P}_t = (X_t, Y_t, Z_t)^T \in \Re^3$ in the Cartesian workspace, to perform vision feedback control tasks, being extensible to any other kind of features. In this section, the depth parameter Z_t is getting updated, Chaumette and Hutchinson (2006), by using Dementhon and Davis pose estimation algorithm, DeMenthon and Davis (1995b) (see Chapter 2 Section 2.3.3.1 for further details about how to obtain the interaction matrix).

The **non-free flying camera** case (see Corke (2011)) is used, in which the motion of the camera is constrained by a manipulator, in our particular case a 6 degree of freedom (DOF) manipulator, where in its end-effector the camera is attached (**eye-in-hand configuration**).

According to the **non-free flying camera** case $\mathbf{J}(\mathbf{s}_t, \mathbf{q}_t) = \mathbf{L_s}(\mathbf{s}_t) \cdot {}^{\mathbf{c}}\mathbf{V_r} \cdot {}^{\mathbf{r}}\mathbf{J_r}(\mathbf{q}_t) \in \Re^{3 \times m}$ is the Jacobean matrix, which relates the measured features velocity with the manipulator velocity, where ${}^{\mathbf{r}}\mathbf{J_r}(\mathbf{q}_t) \in \Re^{m \times m}$ is the robot

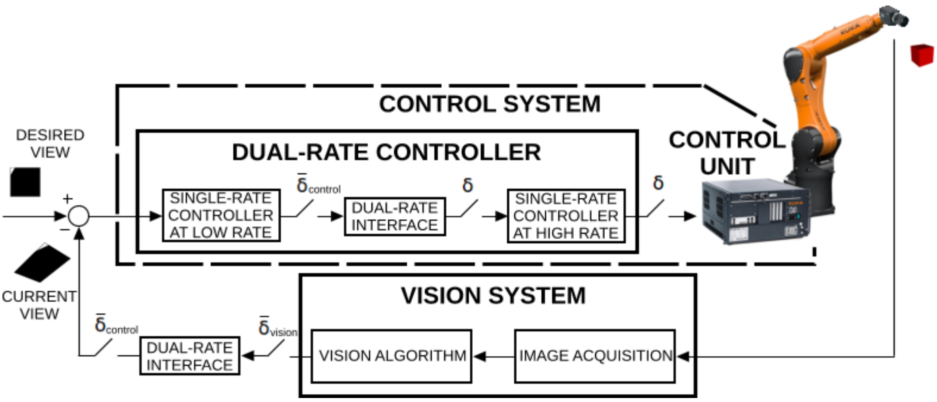


Figure 3.9: Visual feedback robot control block diagram.

Jacobean matrix, ${}^{\mathbf{c}}\mathbf{V_r} \in \Re^{6\times m}$ is the twist velocity matrix, $\mathbf{L_s}(\mathbf{s}_t) \in \Re^{3n\times 6}$ is the image Jacobean (known as interaction matrix) and $\mathbf{q}_t \in \Re^m$ and $\dot{\mathbf{q}}_t \in \Re^m$ are the robot positions and velocities, respectively, being m the manipulator number of DOF.

Thus, the general control law is given by:

$$\dot{\mathbf{q}}_t = \hat{\mathbf{J}}^+(\mathbf{s}_t, \mathbf{q}_t)\cdot\dot{\mathbf{s}}_t \tag{3.2}$$

Figure 3.9 shows the general multi-rate control block diagram. Regardless of the robot used, and focusing on the working frequency, it is possible to distinguish to subsystems out: *Vision System* and *Control System*.

In *Vision System* two related tasks are contemplated: *Image Acquisition* is in charge of getting images with a sampling period $\bar{\delta}_{acq}$; *Vision Algorithm*, containing image processing algorithms, running at $\bar{\delta}_{proc}$. Thus, $\bar{\delta}_{vision} = \bar{\delta}_{acq} + \bar{\delta}_{proc}$ denotes the periodic sampling time of the *Vision System*.

With respect to the *Control System*, it is composed by a high level controller that, in the general multi-rate case, is a dual-rate controller, although it can also be a single-rate one, and a low level controller that is in charge to control the motors directly. $\bar{\delta}_{control}$ will be the sampling period of the low frequency controller counterpart (if applicable) and δ the minimum period admissible by the manipulator, which will be the same for the high frequency controller counterpart.

Along this section two controllers are used: a proportional control law, Hashimoto (1993); Corke (2011), and a PID controller, Lots et al. (2001).

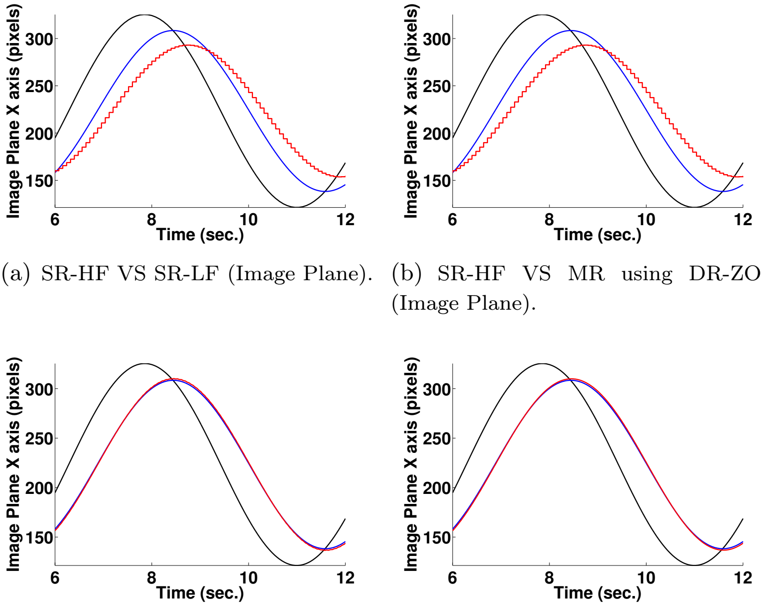
In this case, it was developed a simulator based on Robotics, Corke (1996), and Machine Vision, Corke (2005), toolboxes developed in Matlab/Simulink.

3.3.1 Results under Simulation

Several experiments have been carried out to analyze some aspects of our approach, which is impossible with real systems. For all of them, the proportional controller is used. Among all the alternatives of placing dual-rate holds within the control loop, in this test they are placed between the *Vision System* and the *Control System*. Thus, the controller works at high frequency and the features are being extrapolated using dual-rate holds. The sampling period of the *Vision System* is $\bar{\delta}_{vision} = 84ms$, being the controller period the same as the base period $\bar{\delta}_{control} = \delta = 10$ ms. Furthermore, a tracking task is performed, where an object moves with constant velocity.

Figure 3.10 shows performances obtained by the single-rate and the proposed dual-rate approach based on DR-HOHs: Figure 3.10(a) compares single-rate low frequency with the ideal at high frequency; Figure 3.10(b) compares dual-rate approach using DR-ZOH with the ideal at high frequency; Figure 3.10(c) compares dual-rate approach using DR-FOH with the ideal at high frequency; Figure 3.10(d)compares dual-rate approach using DR-SOH with the ideal at high frequency. Note how when dual-rate higher order holds are used, the system behaves much more similar to the ideal high frequency one. Due to the trajectory followed by the object, both first and second function fit quite the same, although this result cannot be generalized.

As expected, the multi-rate approach based on dual-rate high order holds provides better system performance than low frequency single-rate. In fact,



(a) SR-HF VS SR-LF (Image Plane).

(b) SR-HF VS MR using DR-ZOH (Image Plane).

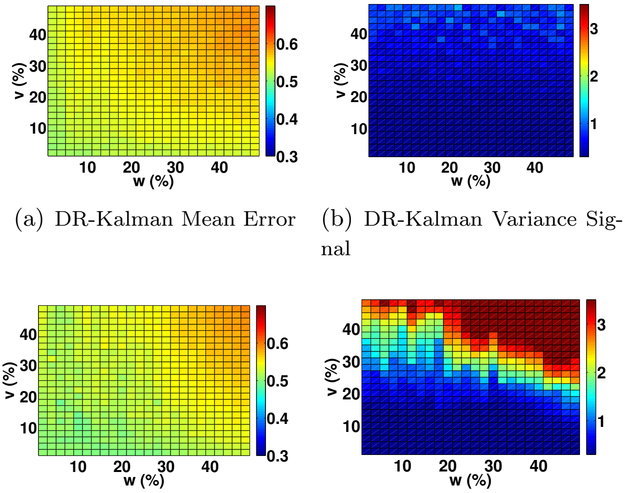
(c) SR-HF VS MR using DR-FOH (Image Plane).

(d) SR-HF VS MR using DR-SOH (Image Plane).

Figure 3.10: Dual-rate visual feedback control using DR-HOHs vs single-rate visual feedback control: true object position, solid black line; high frequency control, solid blue line; low frequency control and control using DR-HOHs, solid red line.

these responses look like the ideal high frequency response.

In addition to this, the next experiment analyzes the effect of the noise, and thus the robustness of this approach, compared with the one obtained using a dual-rate Kalman filter, Armesto and Tornero (2006), Steffes (2014). It is obvious that, dual-rate high order holds do not perform filtering and therefore they are not as robust as Kalman filter while dealing with noises. The aim of the analysis is to determine the maximum level of noise that can be allowed with dual-rate high order holds. Figure 3.11 shows the



(a) DR-Kalman Mean Error (b) DR-Kalman Variance Signal

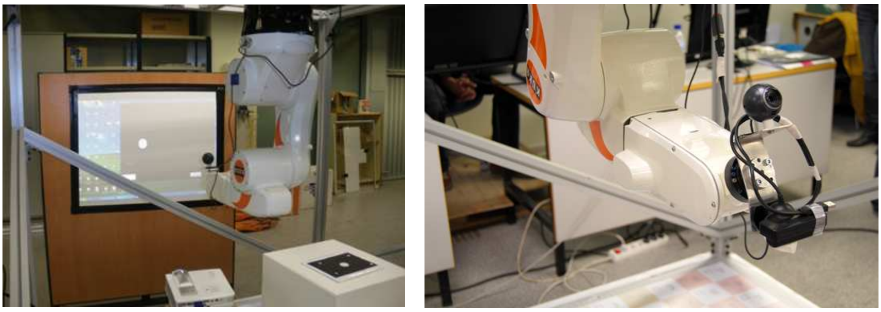
(c) DR-FOH Mean Error (d) DR-FOH Variance Signal

Figure 3.11: DR-HOH versus DR-Kalman Filter

results obtained with both dual-rate estimators in terms of MSE and its covariance, while varying the percentage of contained error on the system (robot pose) and measurement (vision system). For measurement errors less than 15%, the performance of both estimators is very similar. For higher errors Kalman behaves better because its filter nature, which is not the case of dual-rate high order holds.

3.3.2 Experimental Results

In this section, Kr5 cell platform is used to validate the proposed approach (see Appendix A, Section A.1.1.1 for a more detail information). A Logitech web-cam mounted on the robot end-effector is used as sensor. The resolution of the camera is set 640 × 480 pixels, which allows work at 30 fps. The object is a single circular shape, and the task goal is to center its center of mass in the middle of the image plane. The object is moving along a square



(a) Kuka KR5 sixx R650 cell. (b) Kuka industrial robot end-effector.

Figure 3.12: Experimental setup.

trajectory with a velocity of 13 cm/s.

As controller, a workstation[1] is used running on Ubuntu 10.04 O.S., and using Visual Servoing Platform, Marchand et al. (2005)), as open source library to implement the needed vision processing algorithm. More detailed information regarding the communication within each system, as well as hardware description, is given in Appendix A.

The initial pose of the end-effector is ${}^{w}\mathbf{M}_e(0.375, 0.0, 0.695, 0, \pi/2, 0)$. The camera is at ${}^{o}\mathbf{M}_c(0.0, 0.0, -1.650, 0.0, 0.0, 0.0)$ with respect to the object, being the transformation between the end-effector and the camera ${}^{e}\mathbf{M}_c(-0.080, 0.0, 0.050, -2.235, -2.182, 0.056)$ (see Figure 3.12).

Given that we don't have direct access to join torques, indirect visual servoing has been implemented according to Hashimoto et al. (1996).

Table 3.2 shows the obtained results while performing the described tracking task using single-rate and the proposed dual-rate approach. Concretely, the conventional low frequency single-rate visual feedback robot control approach is compared with the proposed dual-rate one using DR-ZOHs, DR-FOHs, DR-SOHs, DR-SOBHs and DR-SOTH. It is shown that quadratic errors produced by the proposed dual-rate approach are less than

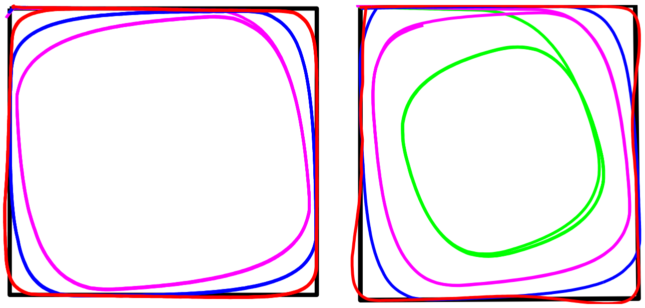
[1] ASUS: Intel Core i7-2670QM, 8GB of RAM @ 2.2GHz and a GeForce GTX 560M 2GB

	$e^2(s*-s)$ (m)	t_s (ms)	$\delta(\%)$
SINGLE-RATE IBVS	0.03441748	–	$Under-damped$
Dual-rate IBVS (DR-ZOH)	0.03431598	–	$Under-damped$
Dual-rate IBVS (DR-FOH)	0.01741782	550	12
Dual-rate IBVS (DR-SOH)	0.01768391	530	9
Dual-rate IBVS (DR-SOBH)	0.01851924	580	9
Dual-rate IBVS (DR-SOTH)	0.01799218	570	9
Dual-rate IBVS (KALMAN)	0.01874451	580	$Under-damped$

Table 3.2: Single-rate versus dual-rate IBVS controller: object moving at 13 cm/s.

the one produced by the system controlled at low frequency. This fact is highlighted in Figure 3.13(a), where the trajectory followed by the center of the camera is drawn, showing that the best result is given when a dual-rate first order hold is used. Remark that this is because the first order estimating function fits better to the trajectory made by the object rather than the second order model function. Although in this particular set-up has been obtained a better performance in terms of convergence time rather than using a dual-rate Kalman filter, the latter is more robust against system and measure noises, as it was shown in the previous section.

The comparison between performances obtained by using a dual-rate controller is shown in Figure 3.13(b). In this case a visual feedback PID, Lots et al. (2001), is chosen as controller. According with Tornero et al. (2001b), a continuous PID is designed and then it is obtained its equivalent at low frequency, at high frequency and the dual-rate equivalent controller (see Chapter 2, Section 2.1.3 and Tornero et al. (2001b) for further details about how obtain a dual-rate PID). Figure 3.13(b) shows the system performance in each one of the described cases, where the better behavior is achieved with the dual-rate PID. The improvement with respect to the conventional single-rate approach reaches a 75% in terms of quadratic errors.



(a) Using a P controller: singel-rate at low frequency, solid magenta line; dual-rate using DR-FOHs, solid red line; dual-rate using DR-SOHs, solid blue line

(b) Using a PID controller: single-rate at low frequency, solid green line; dual-rate with PID designed at low frequency, solid magenta line; dual-rate with PID designed at high frequency, solid blue line; dual-rate with dual-rate equivalent PID, solid red line.

Figure 3.13: Visual feedback robot control: single-rate vs dual-rate approaches.

3.4 Asynchronous Dual-rate High Order Holds with Time Delay Compensation

At this point, dual-rate high order holds have demonstrated how their inclusion within the control loop improves the overall system behavior, not only working with linear systems but with nonlinear systems.

However, the control of such systems can be improved by considering signal asynchronies, which is more realistic when dealing with robotic platforms. That is, even although the control actions have to be given periodically, the data from sensors might be used at the moment which is available. This concept was introduced to dual-rate holds in Armesto et al. (2008), improving in grate measure the performance given by classical dual-rate

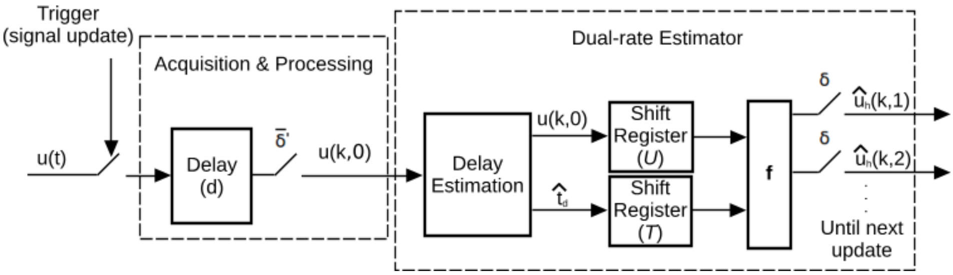


Figure 3.14: Dual-rate Asynchronous High Order Holds with Time Delay Compensation block diagram.

holds.

Other aspect to consider is signal delays, as happens in the case of vision systems, where the required time by the sensor for acquiring + processing + sending results to other systems includes delays between the moment that the picture was taken and the moment the controller has access to such results.

The contribution of this section is the generalization of dual-rate asynchronous holds presented in Armesto et al. (2008), by adding the Time Delay Compensation, coining the approach as dual-rate asynchronous high order holds with time delay compensation (Async. DR-HOH-TDC).

3.4.1 Concepts and Formulation

Let us define the time delay as t_d. Since sometimes it is difficult to compute the exact value of such delay when working with real systems, for now and the rest of the chapter signals depending on the delay will be treated as estimations and denote with $\hat{\cdot}$. Taking this notation, the general expression of a hold is:

$$\hat{\mathbf{u}}_h(t) = \sum_{l=0}^{n} \mathbf{f}_l \left(t, \mathbf{u}(\hat{t}_{j-l}), \hat{t}_{j-l} \right) \tag{3.3}$$

Algorithm 2 Dual-rate Asynchronous High Order Holds with Time Delay Compensation

```
 1: Input: $\mathbf{u}(t)$,$\mathcal{U}_j, \mathcal{T}_j, t_j$,Other information about $t_{d_j}, t$
 2: $[\mathbf{u}(t), \hat{t}_j]$ =DelayEstimation($\mathbf{u}(t), t_j$,Other information about $t_{d_j}$)
 3: if $\mathbf{u}(t)$ is updated then
 4:     shift out $\mathbf{u}(\hat{t}_{j-n})$ and shift in $\mathbf{u}(\hat{t}_j)$ from $\mathcal{U}_j$
 5:     shift out $\hat{t}_{j-n}$ and shift in $\hat{t}_j$ from $\mathcal{T}_j$
 6:     $\hat{\mathbf{u}}_{\mathbf{h}}(t) = \mathbf{u}(\hat{t}_j)$
 7: else
 8:     $\hat{\mathbf{u}}_h(t) = 0$
 9:     for $l = 0$ to $n$ do
10:         retrieve $\mathbf{u}_{\hat{t}_j}$ from $\mathcal{U}_j$
11:         retrieve $\hat{t}_j$ from $\mathcal{T}_j$
12:         $\hat{\mathbf{u}}_h(t) = \mathbf{u}_h(t) + \mathbf{f}_l\left(t, \mathbf{u}(\hat{t}_{j-l}), \hat{t}_{j-l}\right)$
13:     end for
14: end if
15: return $\hat{\mathbf{u}}_h(t)$, $\mathcal{U}_j$ and $\mathcal{T}_j$
```

where $\mathbf{u}(\hat{t}_j)$ denotes the hold input signal at the delay compensated instant $\hat{t}_j = t_j - \hat{t}_{d_j}$, being $\hat{t}_{j-n} \leq \ldots \leq \hat{t}_j \leq t$, and $\hat{t}_{d_j}$ the estimated time delay at j.

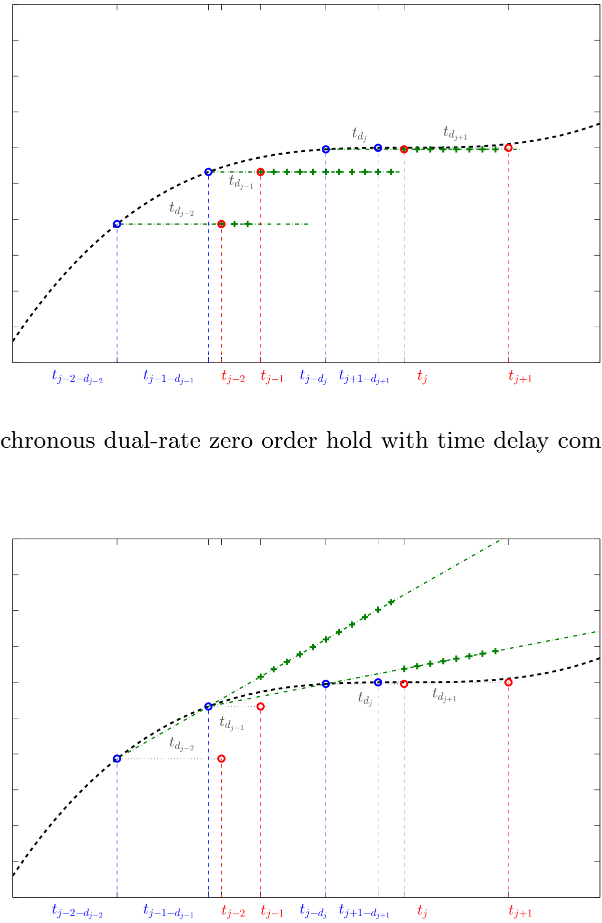
Figure 3.14 shows the block diagram of the generalized dual-rate hold. Asynchronism is denoted by the variable frame period $\bar{\delta}'$, meanwhile the base period (hold output) is denoted δ. The hold incorporates a delay estimator which, if it is possible, receives time data information from the acquisition/processing block. Thus, having the estimated delay and the signal value, it is possible to compute more accurate inter-sampling estimations.

In addition, the pseudo-code for implementing the proposed dual-rate hold with time delay compensation is given in Algorithm 2, which is the generalization of the algorithm presented in Armesto et al. (2008), corresponding to the control diagram block of Figure 3.14. In line 2, delay estimation is needed and performed with input $\mathbf{u}(t)$ at t_j and the corresponding time information given by sensors. Once the time is compensated obtaining $\hat{t}_j$, both $\mathcal{U}_j = \{\mathbf{u}(\hat{t}_j), \ldots, \mathbf{u}(\hat{t}_{j-n})\}$ and $\mathcal{T}_j = \{\hat{t}_j, \ldots, \hat{t}_{j-n}\}$ are updated with $\mathbf{u}(\hat{t}_j)$ and $\hat{t}_j$, respectively, and shifting out the last part information $\mathbf{u}(\hat{t}_{j-n})$ and $\hat{t}_{j-n}$ (see lines 4 and 5). The rest of the algorithm (from line 8 to 13) is the application of Equation 3.3, getting finally the corresponding inter-sampling estimations, returning in line 15 $\hat{\mathbf{u}}_h(t)$, $\mathcal{U}_j$ and $\mathcal{T}_j$.

From now to the end of this chapter, holds to be used are the ones based on primitive functions, Armesto and Tornero (2003a). Figure 3.15 shows the performance obtained by Async. DR-ZOH-TDC and DR-FOH-TDC, while Figure 3.16 shows a comparison of Lagrange, Bezier and Taylor DR-SOHs, denoted as (DR-ASOH-TDC), (DR-ABSOH-TDC) and (DR-ATSOH-TDC), respectively.

In addition, Figure 3.17 compares both Async. DR-SOH and DR-SOH-TDC. With compensating the time delay the hold is able to provide better estimations than without such compensation.

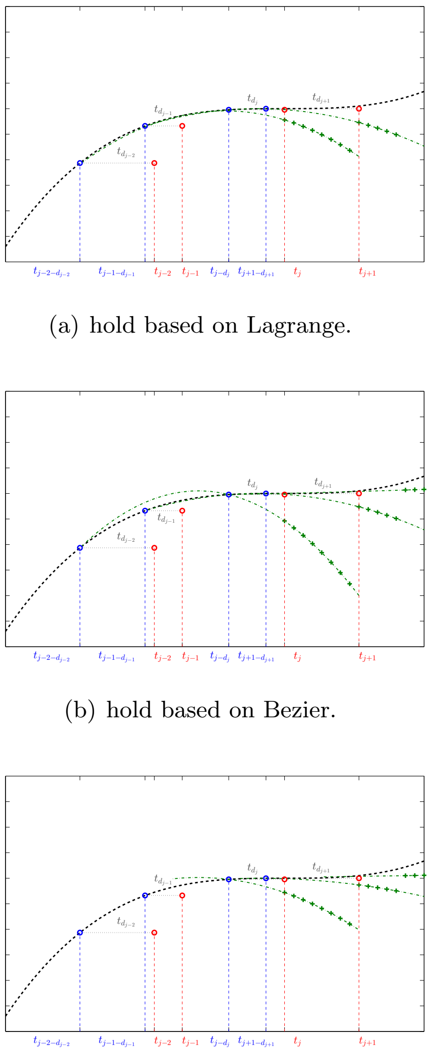
In the following, without loss of generality, we comment the results obtained when we use the new Async. DR-HOH-TDC for estimating data from vision systems in order to control robots such as industrial robot manipulators. Even that this is a particular example, the results obtained are more than relevant because this system is highly nonlinear, validating thus the presented approach.



(a) Asynchronous dual-rate zero order hold with time delay compensation.

(b) Asynchronous dual-rate first order hold with time delay compensation.

Figure 3.15: Performances of Async. DR-ZOH-TDC and Async. DR-FOH-TDC.



(a) hold based on Lagrange.

(b) hold based on Bezier.

(c) hold based on Taylor.

Figure 3.16: Performances of Async. DR-SOH-TDC.

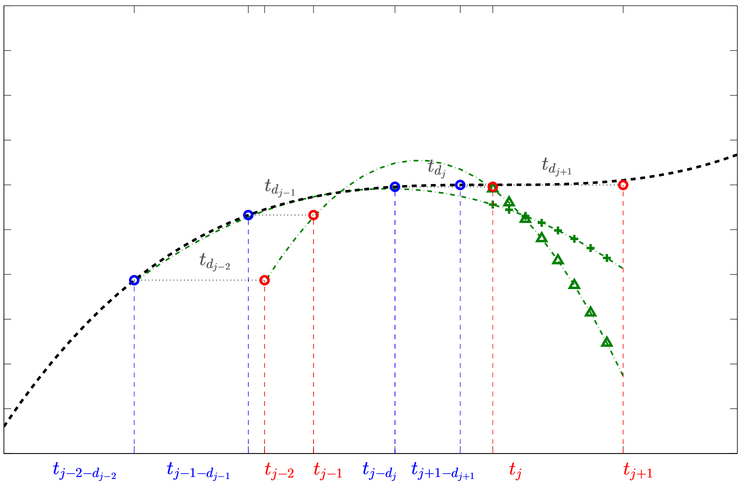


Figure 3.17: Comparison between the estimations carried out by Async. Dual-rate SOH with time delay compensation (green cross samples) and without (green triangular samples)

3.4.2 Data Estimation from Vision Systems

Figure 3.18 shows a typical industrial visual feedback robot control structure. There is a controller in charge to get data from the vision sensor in order to compute the correspondent control action to be sent to the robot. The robot exchanges data with the external controller, while the vision system only sends data without receive any feedback from the external controller. Coming up next in Sections 3.4.2.1 and 3.4.2.2, the robot system is not considered.

3.4.2.1 Simulated Results

Let us suppose the following virtual set-up: a stationary 5 MP camera (2048 × 2444 pixel) ceiling mounted, in overhead position, and looking at a

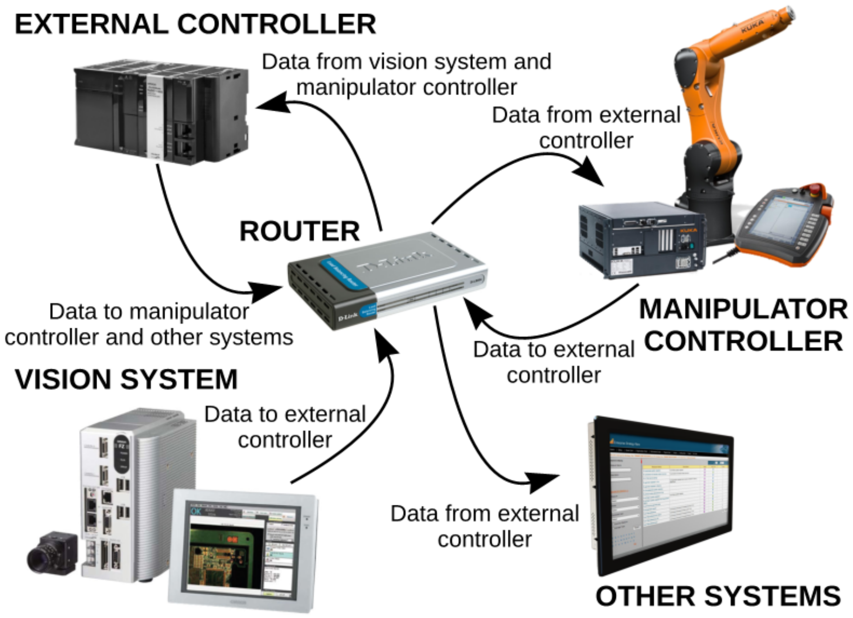

EXTERNAL CONTROLLER
Data from vision system and manipulator controller
Data from external controller
ROUTER
MANIPULATOR CONTROLLER
Data to manipulator controller and other systems
Data to external controller
VISION SYSTEM
Data to external controller
Data from external controller
OTHER SYSTEMS

Figure 3.18: Industrial control structure based on vision systems.

target, which moves in a circular path of 10 cm of radio and a velocity of 4 cm/s. The minimum frame period of the camera is 100 ms, while the worst case will be variable depending on the experiment performed. The control task goal is to track the object.

Figure 3.19 shows the estimations performed by several dual-rate holds. Concretely, the comparison between classical synchronous dual-rate second order hold (Sync. DR-SOH), asynchronous dual-rate second order hold (Async. DR-SOH) and the proposed asynchronous dual-rate second order hold with time delay compensation (DR-SOH-TDC). The reason of using a second order hold is because the hold function model fits with the signal to be estimated. Comparing the three of them it can be concluded that the best results are obtained by the Async. DR-SOH-TDC.

Figure 3.20 shows the main property of this approach. When delays are

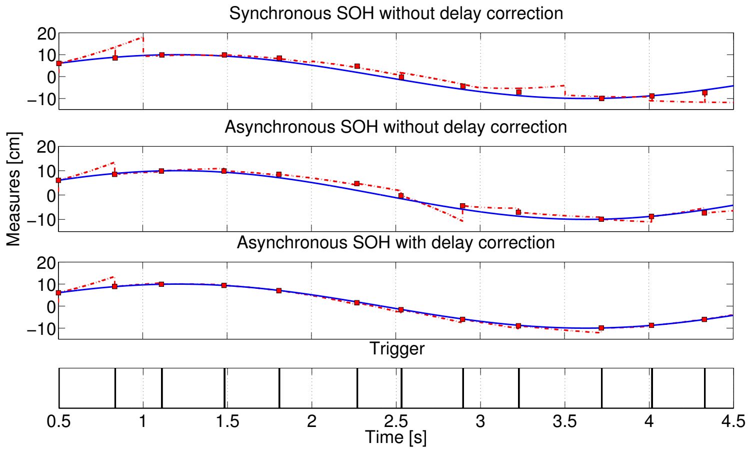


Figure 3.19: Comparison of the performance carried out by using classical asynchronous approach with and without time delay compensation.

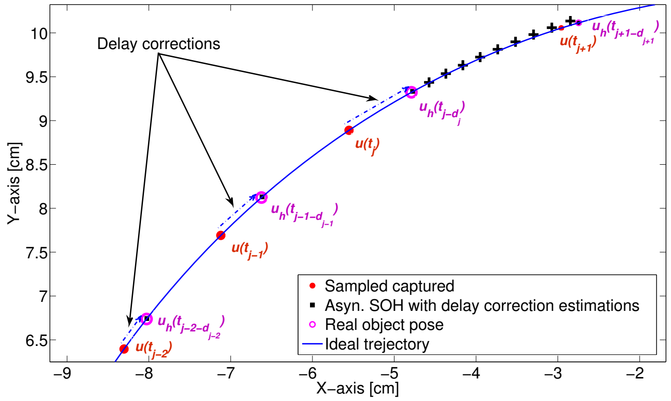


Figure 3.20: Asynchronous dual-rate second order hold with time delay compensation performance into the Cartesian workspace.

significant, classical dual-rate holds are performing their estimations using past information. By introducing the TDC component, the estimations performed by dual-rate holds are based on actual (an error factor of time delay compensation was introduced to fit better with the reality) information and thus the improvement in their estimation is evident.

Figure 3.21 shows the MSE of the estimations with respect the ideal trajectory using Async. DR-HOHs with and without TDC approach. Note how, independently of the hold order chosen, the best results are always given by the proposed dual-rate holds with time delay compensation.

In addition to this, Figure 3.22 shows the histogram analysis of the MSE indicating how is the data disparity when the TDC component is not included and when it is included. The result shows a big improvement when the TDC component is included with a disparity below 7%. However, without the TDC component, the MSE disparity is in between 57% and 87%. These results confirm that with the new DR-HOH-TDC better estimations are obtained.

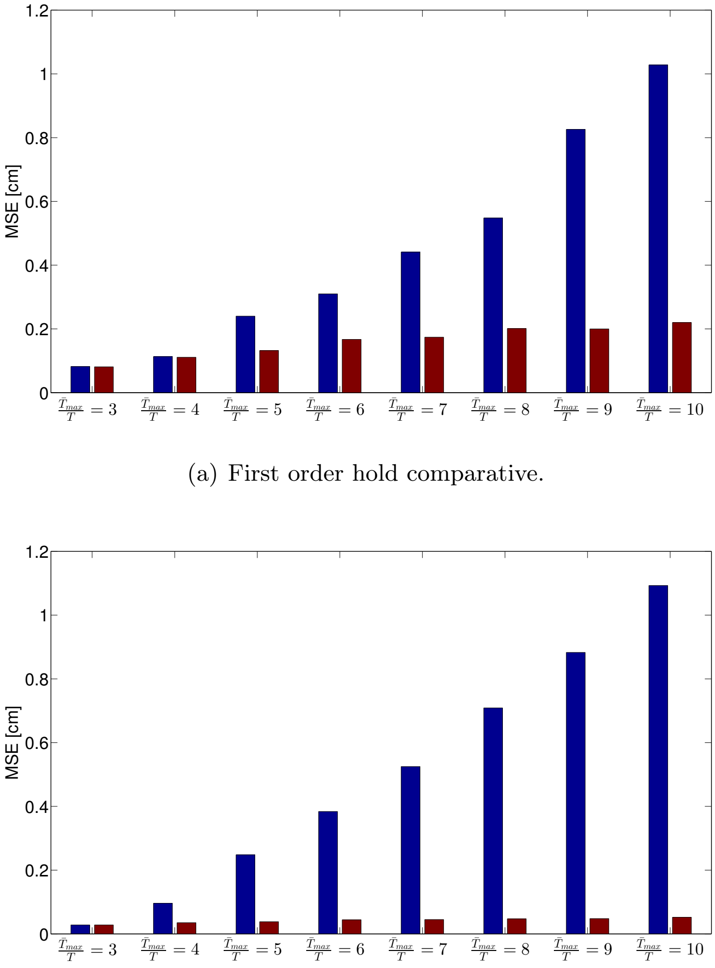
Remark 1 *Input signal delays produce noise effect to high order holds.*

The proof of remark 1 is given in Figure 3.21. Note that, estimations produced by Async. DR-SOH are worst than the ones performed by a first order hold. This is because the signal stored by the hold, due to delays, is becoming noisy and, as higher order model function is used, much bad data is utilized for computing the estimations, giving worst results. On the other hand, note that if delays are compensated, the second order hold produce better estimations rather than the first order hold, in this particular case because the second order function fits better with the input signal.

Remark 2 *Time delay compensator component acts like a low pass filter to high order holds.*

The proof of Remark 2 is given empirically in Section 3.4.2.2.

The comparison between noisy signals and asynchronous delayed signals that Remark 1 postulates is demonstrated in Figures 3.23 and 3.24, where the MSE obtained by using DR-FOH-TDC and DR-SOH-TDC is compared



(a) First order hold comparative.

(b) Second order hold comparative.

Figure 3.21: Asynchronous Dual-rate High Order Holds estimation MSE with and without time delay compensation approach.

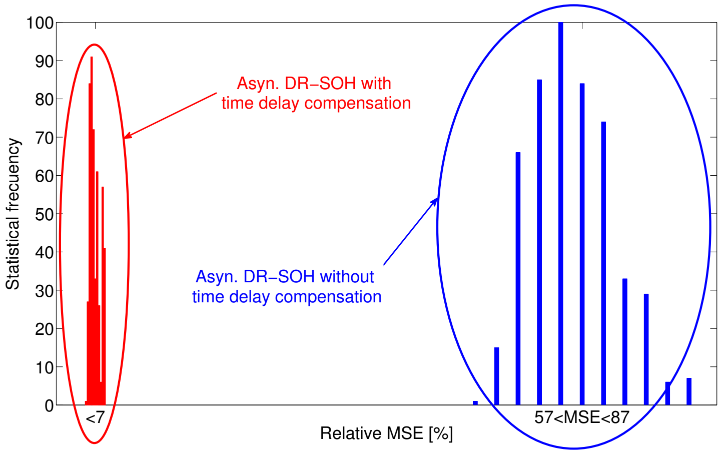
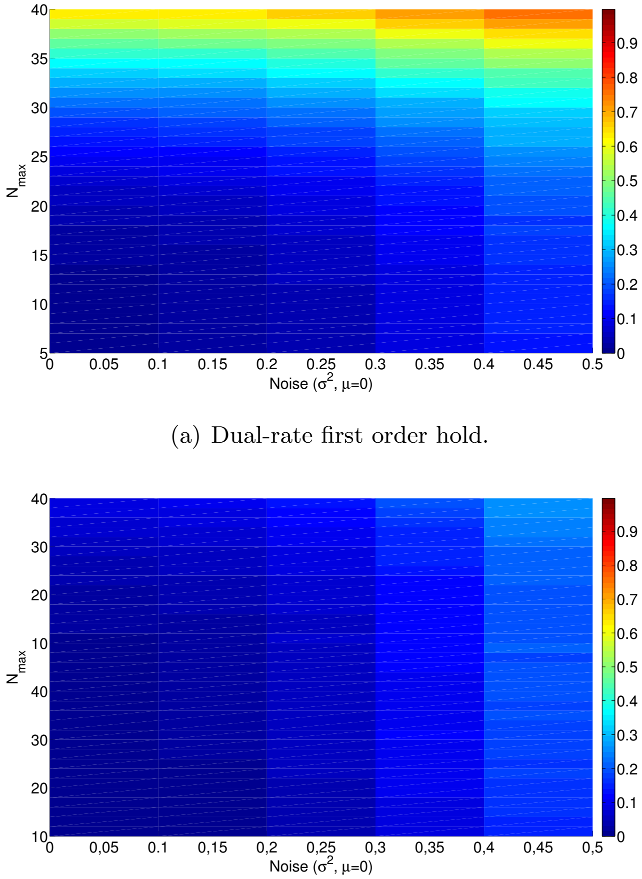


Figure 3.22: Histogram comparison between DR.SOH with and without TDC.

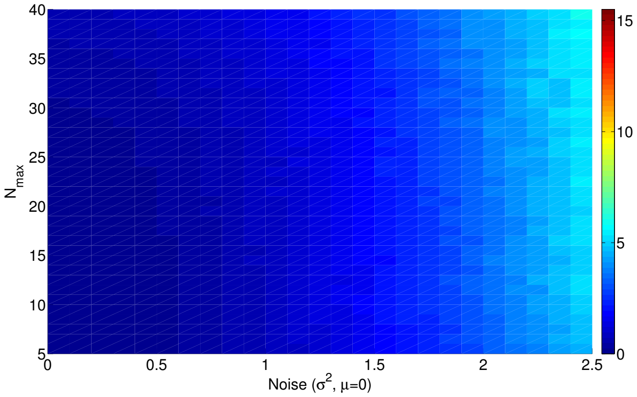
in function of the level of noise introduced with the hold input signal. Again, it can be seen that, when the noise level is low, DR-SOH-TDC obtains better estimations rather than DR-FOH-TDC because its model function fits better with the trajectory followed by the target. Contrary to this, if the noise level is high, then is the DR-FOH-TDC the one which produces better estimations rather than DR-SOH-TDC.



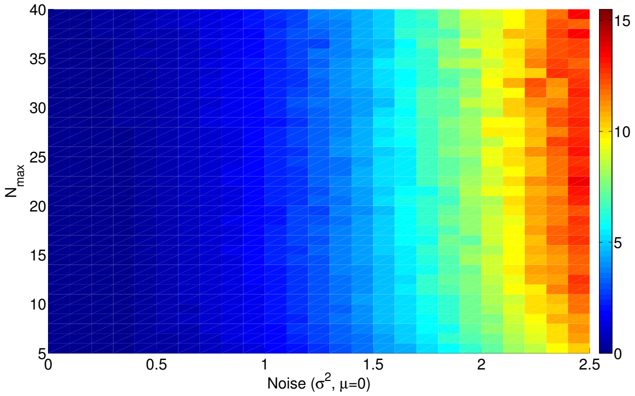
(a) Dual-rate first order hold.

(b) Dual-rate second order hold.

Figure 3.23: Async. DR-HOH-TDC performance against low noisy input signals.



(a) Dual-rate first order hold.



(b) Dual-rate second order hold.

Figure 3.24: Async. DR-HOH-TDC performance against high noisy input signals.

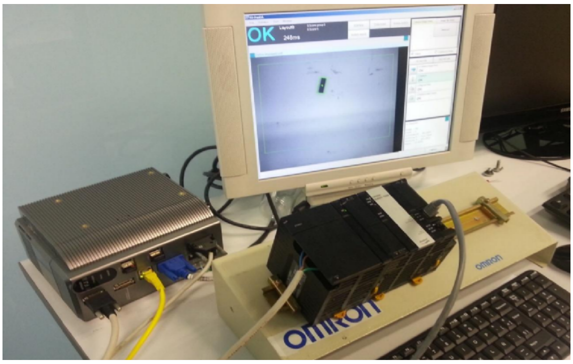


Figure 3.25: Experimental set-up: PC-based Xpectia vision system, 5 MP camera and a PLC.

3.4.2.2 Experimental Results

Let us define the set-up used in this section (Figure 3.25): a PC-based industrial vision system Xpectia, a 5 MP industrial camera (resolution of 2448×2044) working at 12 fps, a PLC as external controller, an automatic rolling disc (see Appendix A.1.1.3) and a blue rectangle as target. The camera communicates with Xpectia via camera-link, and Xpectia uses Ethernet UDP/IP protocol to communicate with the PLC. Within the PLC, all the presented approaches are implemented (see Appendix A). A study of the worst case given by the detection algorithm to be used fixes the frame period in $\bar{\delta} = 300$ ms, being the base period $\delta = 10$ ms.

The goal of this experiment is to detect the Cartesian pose of a moving object and send this pose out, as well as the acquisition+processing runtime to the external computer. In this sense, a 3D object pose detection algorithm, based on image model of the object already implemented in Xpectia, is used. This algorithm has the following working modes, depending on the precision or the searching window required by the application:

- *Matching model using* 360^o *(P1)*: if this option is activated, the matching between the object and the model is carried out not only along

Table 3.3: Comparison of Asyn DR-HOHs performances with and without TDC

				MSE			
P1	**P2**	**ALG. TIME [ms]**	**VELOCITY [rpm]**	**WITHOUT TDC**		**WITH TDC**	
				FOH	**SOH**	**FOH**	**SOH**
ON	**OFF**	**300**	9	3.23	3.04	2.64	1.79
			7.5	3.07	2.80	1.96	1.31
			6	2.56	2.49	1.76	1.29
			4.5	2.21	2.19	1.35	1.11
OFF	**OFF**	**88**	9	1.60	1.59	0.80	0.83
			7.5	1.47	1.46	0.76	0.82
			6	1.30	1.30	0.65	0.77
			4.5	1.13	1.13	0.53	0.68
	ON	**120**	7.5	1.64	1.63	0.87	0.79
			6	1.55	1.54	0.79	0.74
			4.5	1.46	1.46	0.72	0.72

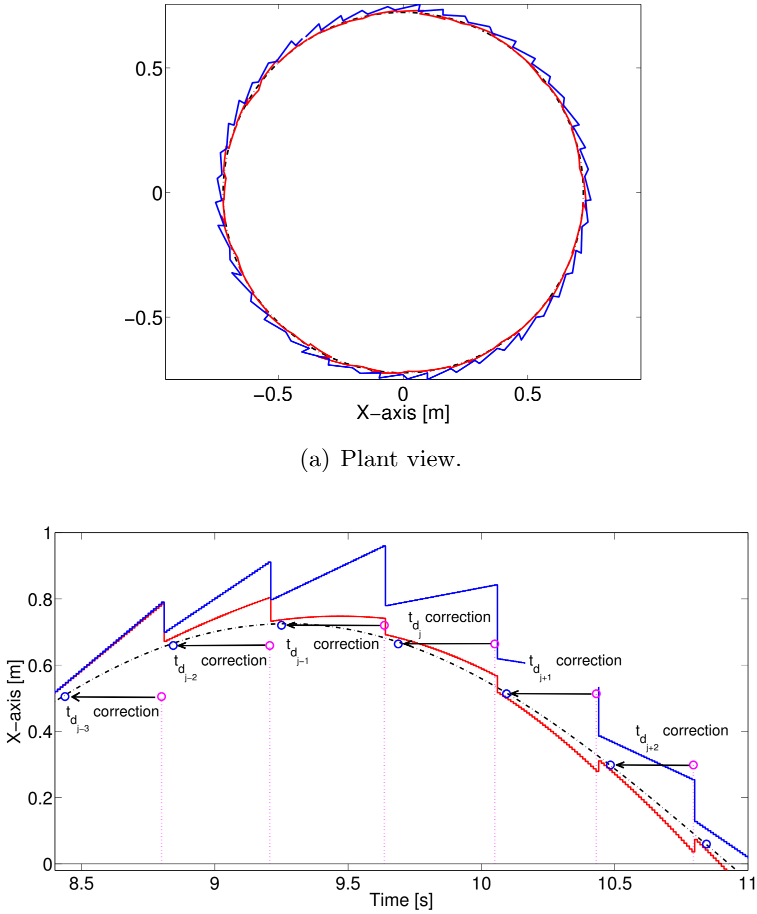
the whole image size, but also considering rotation of the object. This option is high computational consuming.

- *Precision (P2)*: this option assures better or worst detection of the object. If this precision option is set on, the algorithm is more robust, but also more time consuming.

Other factor to be considered is the velocity of the moving object, which means that as much velocity more problems the algorithm will get to detect it.

Table 3.3 shows the results obtained doing this experiment. The best performances are obtained by the proposed Async. DR-HOH-TDC, independently of the algorithm configuration. It is interesting to remark that, the worst case, which is when the algorithm parameter settings are all OFF, first order hold produces better estimations rather second order hold. This result confirms what we proved in said in earlier sections.

Figure 3.26 shows the comparison between the estimations given by DR-



(a) Plant view.

(b) Zoom view.

Figure 3.26: Application example of Async. DR-HOH-TDC: object trajectory, dashed black line; using DR-SOH-TDC, solid red line; using DR-FOH-TDC, solid blue line;.

FOH-TDC and the ones given by DR-SOH-TDC. Due to the path followed by the object, second order hold model function fits better rather than first order model function.

3.4.3 Visual Feedback Manipulator Control using Async. DR-HOH-TDC

In earlier sections, Async. DR-HOHs-TDC have been successfully validated in open loop. The aim of this section however, is the experimental validation of the proposed Async. DR-HOH-TDC, when they are include into the control loop of a visual feedback robot control system.

The experimentation is carried out by using the Agilus cell (see Appendix A for more details), which includes: a 6 DOF Agilus industrial robot manipulator; a camera Logitech web-cam C300 attached to the robot-end effector; a workstation[2] as control system; and a PC-Based vision system.

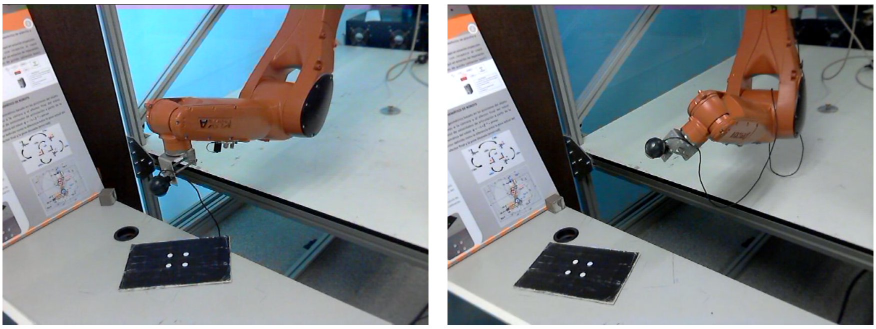
The camera is connected via USB to the PC-based vision system, which functions are: the image acquisition and processing, and the communication of the results via protocol UDP/IP to the control system.

Within the control system, all the proposed algorithms are implemented, including all the dual-rate extrapolators seen along this chapter and the 2D visual feedback robot control, Corke (2011).

In this case, four coplanar dots forming a square of $0.300m$ of side are used as target. Along this section, several experiments are analyzed, keeping the target stationary, performing thus an error position control task.

Regarding to the experimentation set-up, two different initial camera positions with respect to the target are studied: in the first, the object frame is placed at ${}^{w}\mathbf{M}_{o}(0.655, 0.023, -0.649, -2.577, 1.608, -0.194)$ with respect to the world frame, the robot end-effector frame is at ${}^{w}\mathbf{M}_{e}(0.621, -0.101, -0.628, 1.334, 2.392, 0.626)$ with respect to the world frame, being the relation between camera and end-effector frames ${}^{e}\mathbf{M}_{c}(-0.080, 0.050, -2.235, -2.182, 0.056)$ (see Figure 3.27(a)); in the second, the object frame is placed

[2] ASUS: Intel Core i7-2670QM, 8GB of RAM @ 2.2GHz and a GeForce GTX 560M 2GB working under Ubuntu 12.04 prompted with a real-time kernel



(a) Set-up for Test 1: ${}^{w}\mathbf{M}_e(0.621, -0.101, -0.628, 1.334, 2.392, 0.626)$.

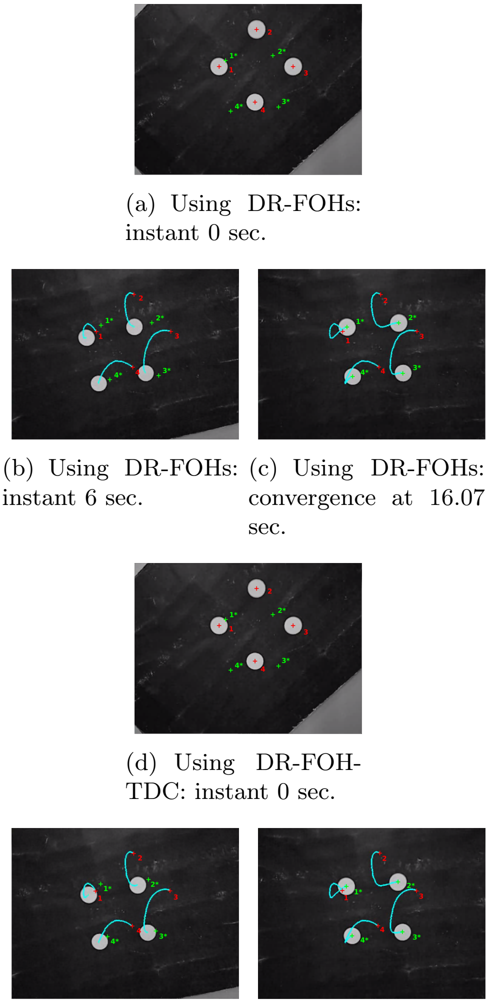
(b) Set-up for Test 2: ${}^{w}\mathbf{M}_e(0.506, 0.037, -0.973, 0.861, 2.724)$.

Figure 3.27: Visual feedback manipulator control for estationary objects: ${}^{w}\mathbf{M}_o(0.655, 0.023, \ -0.649, -2.577, 1.608, -0.194)$ and ${}^{e}\mathbf{M}_c(-0.080, 0.050, -2.235, -2.182, 0.056)$.

at ${}^{w}\mathbf{M}_c(0.621, -0.101, -0.628, 1.334, 2.392, 0.626)$ with respect to the world, the robot end-effector frame is at ${}^{w}\mathbf{M}_e(0.506, 0.037, -0.973, 0.861, 2.724)$ with respect to the world frame, with the same relation between the camera and the end-effector frames as the case before (see Figure 3.27(b)).

Figure 3.28 shows the image plane trajectory followed by each feature in three different sampling times, comparing in this case Async. DR-FOHs and Async. DR-FOH-TDC. Related to this are the results shown in Figure 3.29, in which a comparison of the MSE and control actions of the system using Async. DR-FOH and Asyn DR-FOH-TDC, respectively, demonstrates an improvement of around 1 second in terms of convergence time carried out by the last ones.

In the same way, Figure 3.30 shows the image plane trajectory followed by each feature in three different sampling times, but this time using Async. DR-FOHs and Async. DR-SOH-TDC. Figure 3.31 shows the MSE and control actions of the system using Async. DR-SOH and Async. DR-SOH-TDC, respectively. As in the previous case, the convergence time is



(a) Using DR-FOHs: instant 0 sec.

(b) Using DR-FOHs: instant 6 sec.

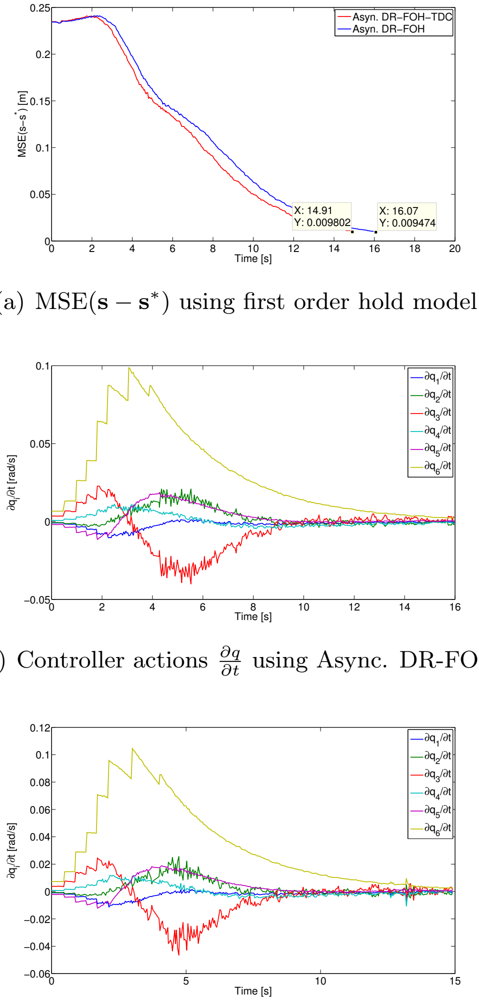
(c) Using DR-FOHs: convergence at 16.07 sec.

(d) Using DR-FOH-TDC: instant 0 sec.

(e) Using DR-FOH-TDC: instant 6 sec.

(f) Using DR-FOH-TDC: convergence at 14.91 sec.

Figure 3.28: Test 1 - Features trajectory performance: comparative between Async. DR-FOH and Async. DR-FOH-TDC.

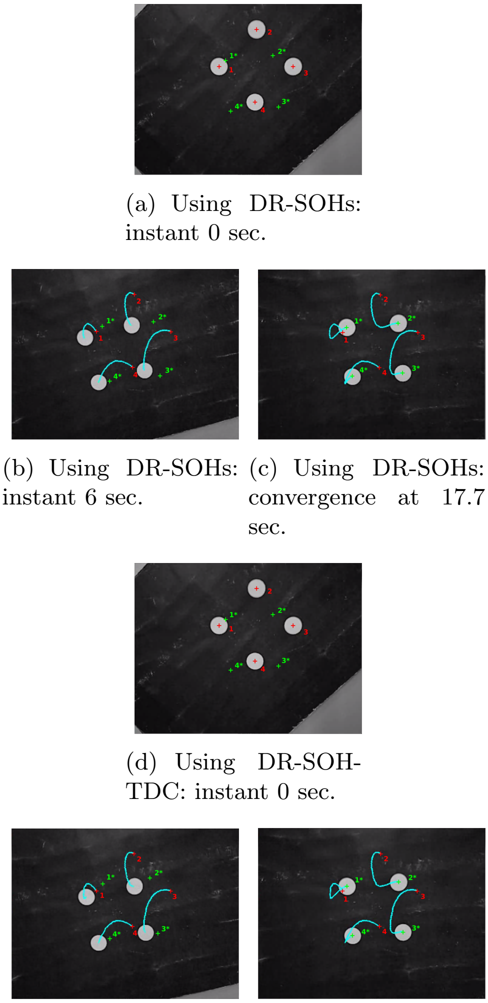


(a) MSE$(\mathbf{s} - \mathbf{s}^*)$ using first order hold model.

(b) Controller actions $\frac{\partial q}{\partial t}$ using Async. DR-FOH.

(c) Controller actions $\frac{\partial q}{\partial t}$ using Async. DR-FOH-TDC.

Figure 3.29: Test 1: comparative between Async. DR-FOHs and Async. DR-FOHs-TDC.



(a) Using DR-SOHs: instant 0 sec.

(b) Using DR-SOHs: instant 6 sec.

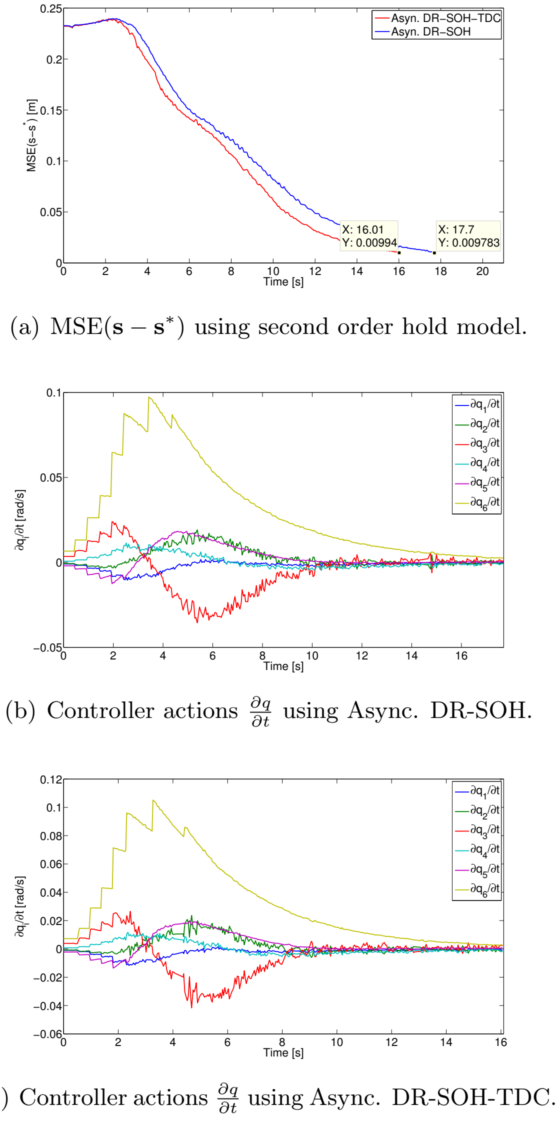
(c) Using DR-SOHs: convergence at 17.7 sec.

(d) Using DR-SOH-TDC: instant 0 sec.

(e) Using DR-SOH-TDC: instant 6 sec.

(f) Using DR-SOH-TDC: convergence at 16.01 sec.

Figure 3.30: Test 1 - features trajectory performance: comparative between Async. DR-SOH and Async. DR-SOH-TDC.



(a) MSE($\mathbf{s}-\mathbf{s}^*$) using second order hold model.

(b) Controller actions $\frac{\partial q}{\partial t}$ using Async. DR-SOH.

(c) Controller actions $\frac{\partial q}{\partial t}$ using Async. DR-SOH-TDC.

Figure 3.31: Test 1: comparative between Async. DR-SOH and Async. DR-SOH-TDC.

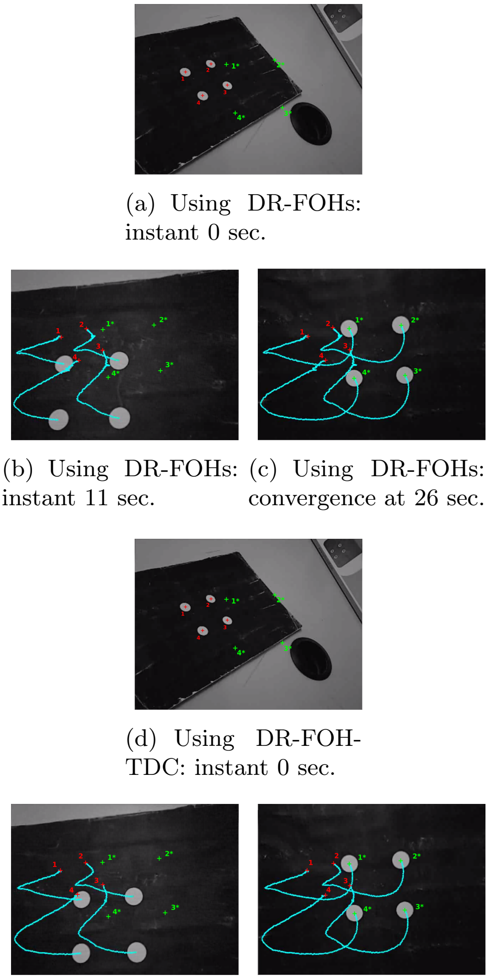
HOLD \ λ	**0.025**	**0.050**	**0.075**	**0.100**	**0.125**	**0.150**	**0.175**	**0.200**	**0.225**	**0.250**	**0.275**	**0.300**
S-DR-ZOH	90.84	52.32	39.76	28.17	23.92	19.38	21.32	–	–	–	–	–
S-DR-FOH	83.57	51.30	34.83	26.92	20.04	18.74	17.55	16.02	17.33	20.39	–	–
S-DR-SOH	85.05	51.15	33.23	25.02	19.46	18.32	20.56	23.32	–	–	–	–
A-DR-ZOH	85.18	49.20	32.36	24.63	19.08	18.92	18.62	20.00	20.15	–	–	–
A-DR-FOH	79.61	46.12	30.66	22.76	18.36	17.13	16.33	14.47	12.03	9.97	8.57	9.27
A-DR-SOH	80.02	46.08	29.10	22.95	18.37	17.17	17.50	18.89	21.88	–	–	–
A-DR-ZOH-TDC	85.18	49.20	32.36	24.63	19.08	20.22	18.69	20.00	20.15	–	–	–
A-DR-FOH-TDC	78.48	45.49	28.85	21.00	17.12	16.53	15.03	12.83	11.45	8.78	8.00	9.12
A-DR-SOH-TDC	79.01	44.07	27.67	20.79	17.00	16.51	16.45	17.55	19.32	–	–	–

Table 3.4: Test 1: Comparison between Sync. DR-HOHs, Async. DR-HOHs and Async. DR-HOH-TDC in function of the controller's gain (λ). The resulting mean of 5 experiments are shown in this table

improved around 1 second with respect to the one obtained using Async. DR-SOHs. However, the convergence time is worse compared with the one obtained by using DR-FOH-TDC, which confirms the results commented in earlier sections about that in the presence of lightly noisy signals, as higher the hold order worse estimations are obtained, proving thus experimentally Remark 2.

In addition to this, Table 3.4 shows a comparison of the convergence time with respect to the controller grain (λ) between all the holds used along this chapter, where "S" denotes Synchronous and "A" denoting Asynchronous. It is interesting to see how the new approach converges faster than the rest, keeping still its robustness, other empirical proof of Remark 2.

The second test is one that classical single-rate control at low frequency cannot solve because of the sensor latency. Due to the last results, this time only first order hold model functions are used. Figure 3.32 shows the comparative of the image trajectories performed by Async. DR-FOH and Async. DR-FOH-TDC, while Figure 3.33 their respective MSE and control actions are represented. Note that the convergence time has improved around 2 seconds without noticing any bad behavior of the system.



(a) Using DR-FOHs: instant 0 sec.

(b) Using DR-FOHs: instant 11 sec.

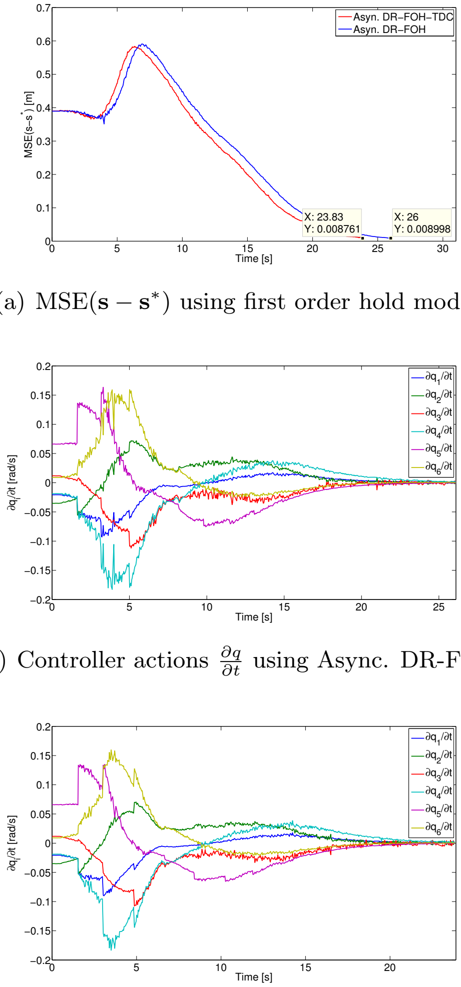
(c) Using DR-FOHs: convergence at 26 sec.

(d) Using DR-FOH-TDC: instant 0 sec.

(e) Using DR-FOH-TDC: instant 11 sec.

(f) Using DR-FOH-TDC: convergence at 23.83 sec.

Figure 3.32: Test 2 - Features trajectory performance: comparative between Async. DR-FOH and Async. DR-FOH-TDC.



(a) MSE($\mathbf{s} - \mathbf{s}^*$) using first order hold model.

(b) Controller actions $\frac{\partial q}{\partial t}$ using Async. DR-FOH.

(c) Controller actions $\frac{\partial q}{\partial t}$ using Async. DR-FOH-TDC.

Figure 3.33: Test 2: comparative between Async. DR-FOHs and Async. DR-FOHs-TDC.

3.5 Summary

The first proposal of this chapter is the use of dual-rate high order holds to improve the performance of nonlinear systems such as robots controlled data from vision sensors.

This approached has been validated first, in a wheeled robot performing a path following task, and then in a 6 DOF industrial robot manipulator performing a target tracking task.

An analysis of similarities and differences between single-rate working at low and single-rate working at high frequency as well as using the new dual-rate approach has been made, showing that the behavior of the system using dual-rate holds is more similar to the ideal working at high frequency and therefore much better than classic control at low frequency.

In addition to this, a comparison under simulation using dual-rate holds and using dual-rate Kalman filter has demonstrated that, for measurement errors less than 15%, the performance of both estimators are very similar. For higher errors Kalman Filter based extrapolator performs better because its inherent filtering effect.

Moreover, a comparison between dual-rate visual feedback robot control using DR-HOHs, and using DR-Kalman Filter as Estimator, has been made experimentally, which results shows that, in this particular case, the error is lower using DR-HOHs, although DR-Kalman is more robust.

The second part of the chapter formulates the extension of asynchronous dual-rate high order holds incorporating a time delay compensator, coining this approach Asynchronous Dual-rate High Order Hold with Time Delay Compensation Async. DR-HOH-TDC.

A wide analysis has been made in order to highlight the properties of such new approach. Under simulation, Async. DR-HOH-TDC has shown a much better performance rather than their homologous Async. DR-HOH. Results have also shown how the noise affects to the hold estimations and how this new approach improves their robustness.

Then, the new holds have been tested by including them within the control loop of a visual feedback robot controlled. In this case a 6 DOF

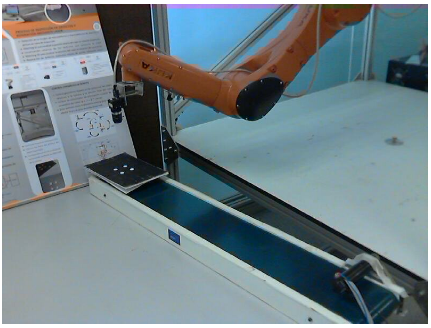


Figure 3.34: Tracking moving targets Agilus cell platform set-up.

manipulator has been used performing both error positioning and target tracking tasks. In all the experiments, the new approach has shown the property to improve more the overall system performance rather than using holds without TDC approach.

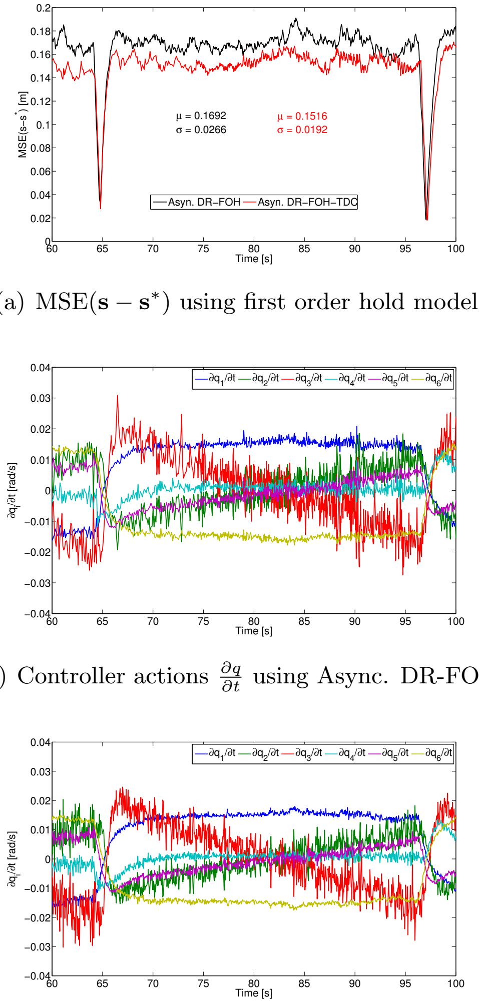
3.6 Discussion

Visual feedback robot control performing a tracking task was also analyzed. In this case, a conveyor is used to perform the experiment, in which the target is placed on it and moves with a velocity of 0.375 m/s (see Figure 3.34). The controller proportional gain used in this case takes a value of $\lambda = 0.200$, and Async. DR-FOHs with and without TDC are compared.

Figure 3.35 shows the analysis results. There is an improvement when the proposed DR-FOH-TDC is used instead the classical equivalent one, although in this case, the noise produced by the robot movement produces not desirable noisy control actions. This example shows the lack of using DR-HOHs when high noisy signals are estimated.

Nowadays we are working on incorporating a Kalman filter into the hold model in order to be able to deal with noisy signals. The preliminary results under simulation scenario show a great improvement as it can be seen in Figure 3.36, where DR-FOH-TDC with Kalman Filter is more robust against noises rather than the homologous DR-FOH-TDC.

Figure 3.35 shows the results of this experimentation, in which DR-FOH-TDC produces a better performance in terms of MSE and standard deviation rather than the analogous without TDC approach.

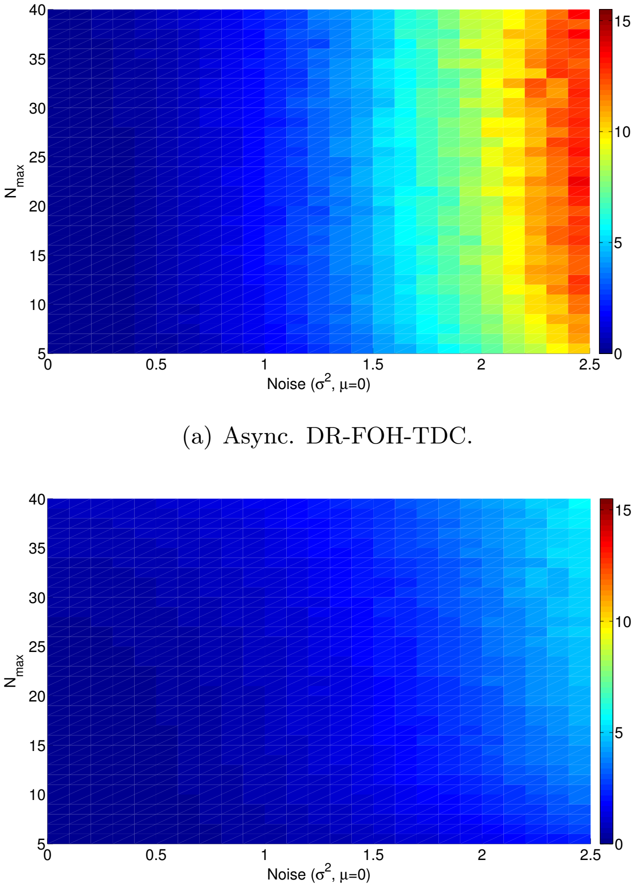


(a) MSE($\mathbf{s} - \mathbf{s}^*$) using first order hold model.

(b) Controller actions $\frac{\partial q}{\partial t}$ using Async. DR-FOH.

(c) Controller actions $\frac{\partial q}{\partial t}$ using Async. DR-FOH-TDC.

Figure 3.35: Object tracking: comparative between Async. DR-FOHs and Async. DR-FOHs-TDC.



(a) Async. DR-FOH-TDC.

(b) Async. DR-FOH-TDC including a Kalman filter.

Figure 3.36: Async. DR-FOH-TDC performance against high noisy input signals including a Kalman filter.

Chapter 4

Visual Feedback Control based on Dual-rate Nonlinear High Order Holds

4.1 Motivation

In Chapter 3, dual-rate high order holds based on primitive functions were used to provide high frequency estimations from low frequency data coming from nonlinear systems, such as an industrial manipulator or a wheeled robot. Although the performance obtained with this approach was satisfactory compared with the same system controlled at low frequency, these holds do not consider system kinematic and dynamic constraints, controller used or task performed, in order to estimate the high frequency data.

The motivation of this chapter is to provide a nonlinear function model for dual-rate holds which includes all the commented system and application aspects. For this regard, this chapter proposes the use of machine learning techniques in order to learn the specific nonlinear hold function model, coining this approach as dual-rate nonlinear high order holds.

A methodology for obtaining the nonlinear hold estimation function is firstly detailed. Although the proposed methodology is particularized to the use of artificial neural networks along this chapter, it is independent of the machine learning tool to be used. The proposed approach is then validated and compared with the classical one in several examples and robotic platforms. Concretely, a 6 DOF industrial manipulator performing a visual feedback control tracking task, and a wheeled robot performing a path following task.

4.2 Multi-rate Nonlinear Holds

In Armesto and Tornero (2003a), the expression of a hold was provided according to the primitive function used (see Chapter 2, Equation 2.6). A more general expression is given by:

$$\mathbf{u}_h(k,i)=\mathcal{F}_i(t,t_k,\mathbf{u}(t_k),...,\mathbf{u}(t_{k-n})) \tag{4.1}$$

where $\mathcal{F}_i$ is each of the elements of a non-linear function-valued vector $\mathcal{F}$ that represents the mapping between low-frequency sampled signals and inter-sampling instants. Therefore, the hold needs to predict inter-sampling signal values based on a set of previously known inputs sampled at low-sampling frequency.

Let us assume $\mathcal{F}:\mathcal{I}\to\mathcal{I}_h$ the mapping between the high-frequency discrete sequence $\mathcal{I}_h=\{\mathbf{u}_h(k,0),\cdots,\mathbf{u}_h(k,N-1)\}$ and the low-frequency signals $\mathcal{I}=\{\mathbf{u}(k-n,0),\cdots,\mathbf{u}(k-1,0),\mathbf{u}(k,0)\}$. Since in $\mathcal{I}_h$ is included future-time instants and inter-sampling values that are not accessible in real-time processes, $\hat{\mathcal{I}}_h$ is denoted as the hold-estimated mapping. Thus, the aim is to find an appropriate mapping such as $\mathcal{I}_h\approx\hat{\mathcal{I}}_h$ for every time instant $t_k=k\bar{\delta}$ and input vector $\mathcal{I}$. Once the packed vector $\hat{\mathcal{I}}_h$ is generated, it is unpacked using the inverse lifting operator $\mathcal{L}^{-1}$, Bamieh et al. (1991), which produces the hold output "*continuized*" signal $\mathbf{u}_h(t)$.

It is easy to reach out to an extension of this formulation from a multi-rate point of view, where different inputs can be considered working at different frequencies. Let us denote "NS" as the number of holds needed by the system (usually it depends on the number of sensors but could be

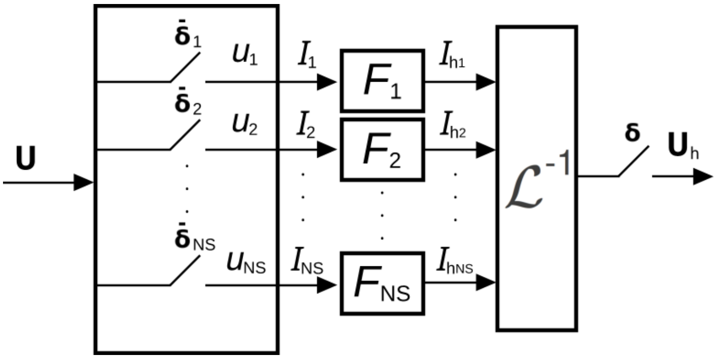


Figure 4.1: Multi-rate Nonlinear High Order Hold Hold block diagram.

also actuators or other kind of systems), $\{t_{k_1}, \ldots, t_{k_{NS}}\}$ the updating instants, $\{\bar{\delta}_1, \ldots, \bar{\delta}_{NS}\}$ the frame period, and δ de common base period, being $\{N_1, \ldots, N_{NS}\}$ the corresponding multiplicity number for each input. Thus, for each sensor, a set of input and output datasets are obtained, and gathering them together, a set of sets of input dataset $\{\mathcal{I}^1, \ldots, \mathcal{I}^{\mathrm{NS}}\}$, and output dataset $\left\{\hat{\mathcal{I}}_h^1, \ldots, \hat{\mathcal{I}}_h^{\mathrm{NS}}\right\}$ is obtained. Then, the order of the hold function can also be varying, and thus $\{n_1, \ldots, n_{NS}\}$ is denoted as the set of "NS" hold orders. The general multi-rate hold (see Figure 4.1) takes then the following form:

$$\mathbf{U}_h = \begin{bmatrix} \mathbf{u}_h(k_1, i_1) \\ \vdots \\ \mathbf{u}_h(k_{\mathrm{NS}}, i_{\mathrm{NS}}) \end{bmatrix} = \begin{bmatrix} \mathcal{F}_{i_1}(t, t_{k_1}, \mathbf{u_1}(t_{k_1}), ..., \mathbf{u_1}(t_{k_1 - n_1})) \\ \vdots \\ \mathcal{F}_{i_{\mathrm{NS}}}(t, t_{k_{\mathrm{NS}}}, \mathbf{u}_{\mathrm{NS}}(t_{k_{\mathrm{NS}}}), ..., \mathbf{u}_{\mathrm{NS}}(t_{k_{\mathrm{NS}} - n_{\mathrm{NS}}})) \end{bmatrix} \tag{4.2}$$

Analytical methods can be used in order to obtain the mapping $\mathcal{F}$. In this regard, in Armesto and Tornero (2005), a first attempt to incorporate the dynamics of the system was presented, although only proved using linear systems (first and second order models). Obtaining a precise nonlinear model analytically is always a hard and tedious work, even more when working with industrial systems, which do not allow knowing all the information about their systems, or simply, it is not possible to get access to the system nonlinear characteristic parameters.

This is the main reason that nowadays machine learning techniques, Sun et al. (1996); Kim et al. (2012); Zhao et al. (2015), are widely used in the field of robotics. The robot hardware is progressively becoming more complex, which leads to a growth of interest in applying machine learning and statistics approaches within the robotic community. At the same time, there has been a growth within the machine learning community in using robots as motivating applications for new algorithms and formalisms.

In this chapter, the use of supervised machine learning methods, Kotsiantis (2007), such artificial neural networks, Misra and Saha (2010), is proposed, which have shown to be one of the most adequate methods as it is shown hereinafter.

The non-linear function-valued vector is learnt from a set of input data working at low frequency and output data working at high frequency. Owing to the fact that the high frequency data is not available, two solutions are proposed:

- *Having access to high frequency data using high technological hardware*: this might be possible by using an advanced sensor (likely more expensive) providing data that we can use during the training phase. The expensive sensor can be later replaced with a cheaper one, which is particularly interesting if the solution has to be reproduced or is industrialized. For example, a camera with Gigabit Ethernet might have a different bandwidth than a USB camera, while they might provide images of a similar quality, since the camera interface is independent from the camera sensor.

- *High frequency data obtained by virtual environments*: nowadays, the use of dynamic simulators is common in all research and industrial fields but in robotics even more. For this reason, there exists a vast variety of available open-source tools which can be applied in specific applications. In this kind of simulators one can perform tasks almost like in the real systems.

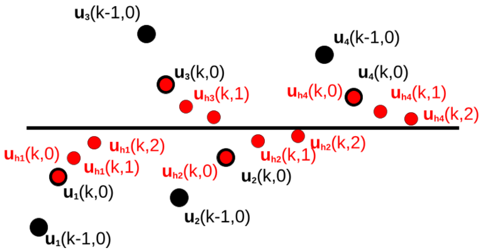


Figure 4.2: Method 1 for collecting data.

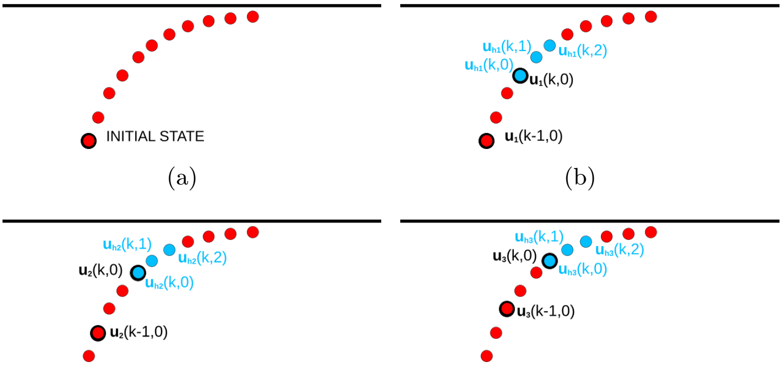
All two methods include kinematic and dynamic information of the system working in close loop. Thus, the non-linear function-valued vector has specific information not only about the system, controller and sensors but also the task performed. As result, better estimations might be obtained.

In this chapter all the experiments presented are performed using the second proposal, which is by using synthetic dataset obtained in simulation. Following, a methodology for obtaining dual-rate non-linear high order holds is described, which is independent of the method used for obtaining datasets at high frequency.

4.2.1 Methodology

In this section, two possible methods for collecting datasets are described. The first one, shown in Figure 4.2, is to do small "jumps" around all the workspace, letting the system evolve N-instants ($N = \frac{\bar{\delta}}{\delta}$) in each "jump". The second one, shown in Figure 4.3, is the same but letting the system more instants to evolve. In this sense, less "jumps" are needed, although there is major probability of repeated data, which can be problematic for the learning procedure.

In both cases, the set of inputs is set as $\mathcal{I}=\{[\mathbf{u}_1(k,0),\mathbf{u}_1(k-1,0)],[\mathbf{u}_2(k,0),\mathbf{u}_2(k-1,0)],[\mathbf{u}_3(k,0),\mathbf{u}_3(k-1,0)],\ldots\}$, and the set of outputs as $\mathcal{I}_h = \{[\mathbf{u}_{h_1}(k,0),\mathbf{u}_{h_1}(k,1),\mathbf{u}_{h_1}(k,2),...],[\mathbf{u}_{h_2}(k,0),\mathbf{u}_{h_2}(k,1),\mathbf{u}_{h_2}(k,2),...],[\mathbf{u}_{h_3}(k,0),\mathbf{u}_{h_3}(k,1),\mathbf{u}_{h_3}(k,2),...],\ldots\}$.



(a) (b)

(c) (d)

Figure 4.3: Method 2 of casting datasets.

In Algorithm 3, the steps for obtaining dual-rate nonlinear high order holds are described. The algorithm inputs are the set of datasets at low and high frequency collected previously and well sequenced. Then two more steps are needed: the training step (from line 5 to line 8), in which machine learning tools are used to learn the non-linear function-valued vector for each l-hold needed; the validation step (from line 9 to line 20), in which the obtained non-linear function-valued vectors are tested and validated. Remark that it is necessary to use different datasets than the one utilized during the training step, otherwise results will be not significant. If the test step is passed, then the non-linear function-valued vector is ready to be used, otherwise the learning procedure has to be reviewed and performed again.

Algorithm 3 Multi-rate NLHOHs setting-up methodology.

```
1: *Obtaining needed datasets:*
2: Set of input $\left\{\mathcal{I}^1, \ldots, \mathcal{I}^{\mathrm{NS}}\right\}_{\mathrm{TRAINING}}$ and output $\left\{\hat{\mathcal{I}}_h^1, \ldots, \hat{\mathcal{I}}_h^{\mathrm{NS}}\right\}_{\mathrm{TRAINING}}$ datasets for learning
3: Set of input $\left\{\mathcal{I}^1, \ldots, \mathcal{I}^{\mathrm{NS}}\right\}_{\mathrm{TEST}}$ dataset for testing
4: *Learning step:*
5: **for** $l = 1, \ldots,$ NS **do**
6:    Learn $\mathcal{F}_l$
7: **end for**
8: *Testing step:*
9: **for** $l = 1, \ldots,$ NS **do**
10:    Test $\mathcal{F}_l$ using $\mathcal{I}_l$ and compare with ideal ones
11: **end for**
12: **if** Testing step OK **then**
13:    **return** $\{\mathcal{F}_1, \ldots, \mathcal{F}_{\mathrm{NS}}\}$
14: **else**
15:    **if** First time **then**
16:       Go to step 5
17:    **else**
18:       Go to step1
19:    **end if**
20: **end if**
```

4.2.2 Example

In this example, the control of a DC Motor is carried out by using a LQR controller. In this sense, the simplified model of a DC Motor is used, which is:

$$\begin{array}{c} \dot{\omega} = -\frac{f}{J}\omega + \frac{K}{J}u \\ \dot{\theta} = \omega \end{array} \tag{4.3}$$

where u is the control input, and $y_1 = \theta$ and $y_2 = \omega$ are the position and velocity of the motor respectively. Its representation in state space takes the form:

$$\begin{aligned}\begin{bmatrix}\dot{\theta}\\ \dot{\omega}\end{bmatrix} &= \begin{bmatrix}0 & 1\\ 0 & -\frac{f}{J}\end{bmatrix}\begin{bmatrix}\theta\\ \omega\end{bmatrix} + \begin{bmatrix}0\\ \frac{K}{J}\end{bmatrix}u + w\\ \begin{bmatrix}y_1\\ y_2\end{bmatrix} &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\theta\\ \omega\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}u + v\end{aligned} \tag{4.4}$$

where w is the process noise, which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $\mathbf{Q_w}$, and v is the observation noise which is assumed to be zero mean Gaussian white noise with covariance $\mathbf{R}_v$.

Thus, the discrete-time domain equivalent system at $\bar{\delta}$ is:

$$\begin{aligned}\begin{bmatrix}\theta(k+1)\\ \omega(k+1)\end{bmatrix} &= \begin{bmatrix}1 & \tau(1-e^{-\bar{\delta}/\tau})\\ 0 & e^{-\bar{\delta}/\tau}\end{bmatrix}\begin{bmatrix}\theta(k)\\ \omega(t)\end{bmatrix} + \begin{bmatrix}K(\bar{\delta}-\tau(1-e^{-\bar{\delta}/\tau}))\\ K(1-e^{-\bar{\delta}/\tau})\end{bmatrix}u(k)+w(k)\\ \begin{bmatrix}y_1(k)\\ y_2(k)\end{bmatrix} &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\theta(k)\\ \omega(k)\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}u(k) + v(k)\end{aligned} \tag{4.5}$$

The concrete values used in this example are: $\delta = 0.2$ and $\bar{\delta} = 2$, being $N = 5$. Furthermore, the DC Motor parameters taken for this example are $\tau = 0.25seg.$ and $K = 0.25$. Then, the continuous model described by Equation 4.4 is:

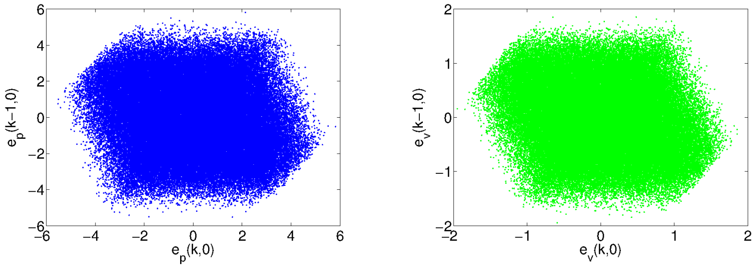
$$\begin{aligned}\begin{bmatrix}\dot{\theta}\\ \dot{\omega}\end{bmatrix} &= \begin{bmatrix}0 & 1\\ 0 & -4\end{bmatrix}\begin{bmatrix}\theta\\ \omega\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u + w\\ \begin{bmatrix}y_1\\ y_2\end{bmatrix} &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\theta\\ \omega\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}u + v\end{aligned} \tag{4.6}$$

Lastly, the continuous-time domain process and observation noise covariances are:

$$Q_c = \begin{bmatrix}5 & 0\\ 0 & 5\end{bmatrix} \quad R_c = 1 \tag{4.7}$$

and a unitary gain matrix $\mathbf{G}_c \in \mathcal{R}^{2x2}$ is taken.

Using MATLAB control toolbox, a LQR is designed in the continuous-time domain, obtaining then the equivalents discrete LQ regulators at low and high frequency.



(a) Motor position dataset distribution.

(b) Motor velocity dataset distribution.

Figure 4.4: Dataset distribution.

A first order hold is used in this example. In addition, the signal to be estimated is the position and velocity errors. Therefore, the set of inputs are set as $\mathcal{I}(k) = \{\mathbf{e_p}(k,0), \mathbf{e_p}(k-1,0), \mathbf{e_v}(k,0), \mathbf{e_v}(k-1,0)\}$, meanwhile the set of outputs are $\mathcal{I}(k) = \{\mathbf{e_p}(k,0), \ldots, \mathbf{e_p}(k,N-1), \mathbf{e_v}(k,0), \ldots, \mathbf{e_v}(k,N-1)\}$.

Figure 4.4(a) shows the collected data uniformly distributed in the range $[-2.5, 2.5]$ radians, and Figure 4.4(b) shows the equivalent velocities for the working point of our system. For this case, the second method for collecting data has been used (see Figure 4.3).

Matlab artificial neural network toolbox, concretely Levenberg-Marquardt back-propagation algorithm, is used to learn the non-linear function-valued vector. A network structure of two-layer feed-forward network with sigmoid hidden neurons and linear output neurons using 10 hidden neurons was used. The validation is carried out using a different dataset.

Figure 4.5 shows the comparison between low and high frequency equivalent LQ regulators. In this case, sampling at low frequency produces a non-desirable performance, while if the sampling is at high frequency the system behaves adequately.

In Figures 4.6 and 4.7, the comparison of the system working at ideal

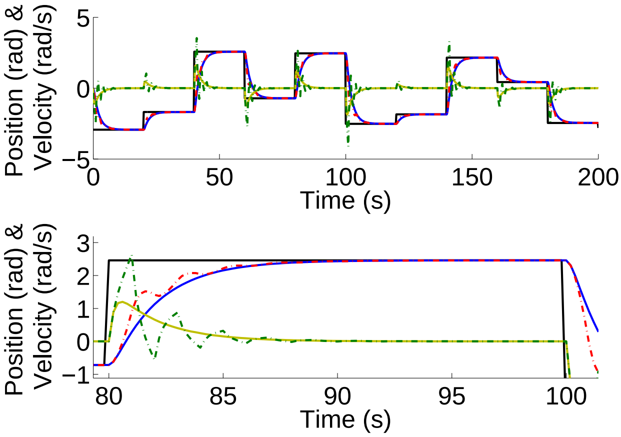


Figure 4.5: Low frequency single-rate control vs control based on dual-rate nonlinear high order holds. Motor position estate: system working at low frequency (dashed red line); system using DR-NLFOHs (solid blue line). Motor Velocity state: system working at low frequency (dashed green line); system using DR-NLFOHs (solid yellow line).

high frequency and using dual-rate approaches are represented. Figure 4.6 show the system behavior obtained by using a classical dual-rate first order hold. In this case, the system has improved considerably its performance with respect to the same system controlled at low frequency. Yet, the comparison with the system controlled at high frequency highlights that there is still appearing some oscillations. On the other hand, by using dual-rate nonlinear first order holds (Figure 4.7), the performance is almost exactly than the ideal single-rate working at high frequency, which proves the improvement obtained with the new approach.

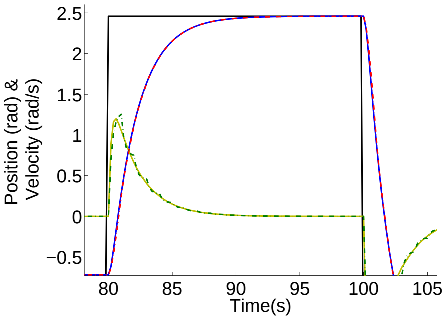


Figure 4.6: Control based on dual-rate high order holds vs control based on dual-rate nonlinear high order holds. Motor position estate: system using DR-FOHs (dashed red line); system using DR-NLFOHs (solid blue line). Motor Velocity state: system using DR-HOHs (dashed green line); system using DR-NLFOHs (solid yellow line).

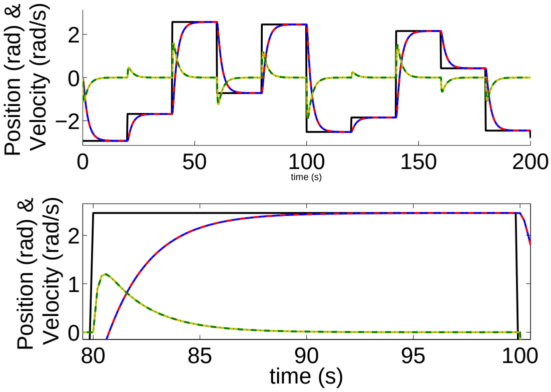


Figure 4.7: Ideal high frequency single-rate control vs control based on dual-rate nonlinear high order holds. Motor position estate: system working at high frequency (dashed red line); system using DR-NLFOHs (solid blue line). Motor Velocity state: system working at high frequency (dashed green line); system using DR-NLFOHs (solid yellow line).

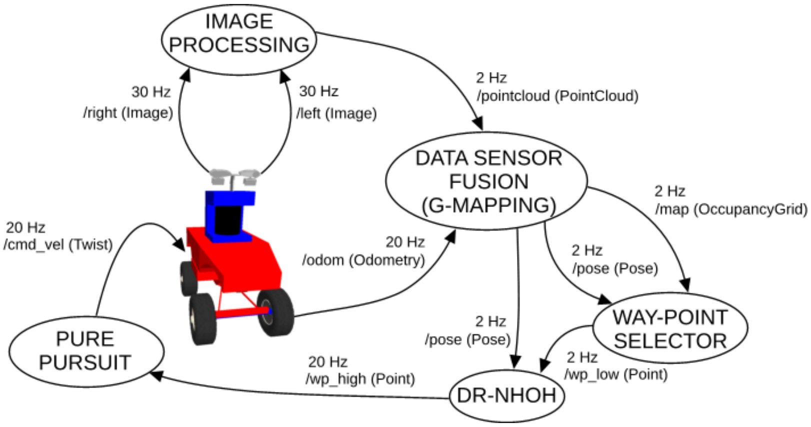


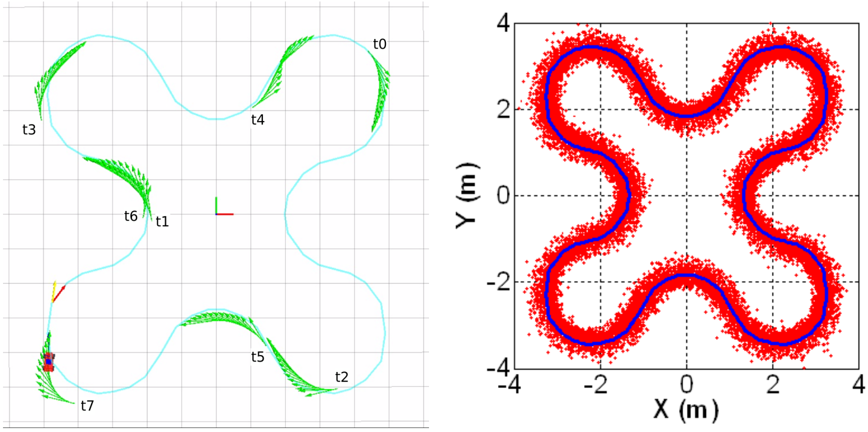
Figure 4.8: SUMMIT virtual environment based on ROS and Gacebo.

4.3 Visual Feedback Wheeled Robot Control based on Dual-rate Nonlinear High Order Holds

The path following control problem described in Chapter 3 Section 3.2 is here analyzed but using the proposed dual-rate nonlinear high order holds (Figure 4.8). In this case, we have considered the same path than in Chapter 3 but here, in order to highlight the properties of the new approach, it is inscribed into a rectangle of 3.2 by 2.9 meters of size with the following control parameters: look-ahead (LA) 1 m, nominal velocity (v) 1.5 m/s, base-period (control period) 50 ms, (20 Hz) and frame-period (measure/way-points acquisition) $\bar{\delta}=500$ ms (2 Hz).

For this application, dual-rate second order holds are used. In addition, way-points (q_T) are estimated by the hold from past measured way-points and the actual robot pose (q_R). Thus, the hold input takes the form $\{q_T(k-2,0), q_T(k-1,0), q_T(k,0), q_R(k,0)\}$, and the hold output $\{q_T(k,0), q_T(k,1), \cdots, q_T(k,N-1)\}$. Note that in this example $\bar{\delta} = 500$ ms and $\delta = 50$ ms, being $N = \frac{\bar{\delta}}{\delta} = 10$.

Figure 4.9 shows the collected data uniformly distributed along the path.



(a) Snapshot of the dataset process.

(b) Final dataset obtained.

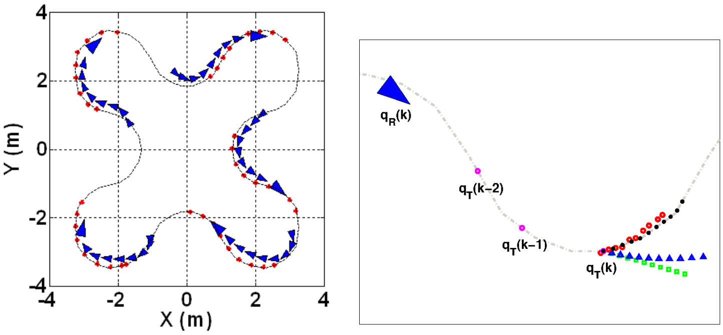
Figure 4.9: Dataset at high frequecy acquition example.

For this case, the first method for collecting data has been used (see Figure 4.2).

Without loss of generality, the results for one specific path are presented in this chapter, although it can be easily extended in order to learn more situations that allow the dual-rate nonlinear hold to estimate way-points for paths a priori unknown.

Although other libraries such as FANN library, Nissen (2003), were used to learn the non-linear function-valued vector, at the end the best results were obtained with Levenberg-Marquardt back-propagation algorithm developed in Matlab. A network structure of two-layer feed-forward network with sigmoid hidden neurons and linear output neurons was used. In this case, using a number of 20 hidden neurons was enough to get good estimations.

Figure 4.10(a) shows the validation of the dual-rate nonlinear second order hold, whereas Figure 4.10(b) compares the estimations performed by classic and nonlinear holds. In that figure, the robot is located in $\mathbf{q}_R(k,0)$, and based on that position, the way-point $\mathbf{q}_T(k,0)$ has been selected based



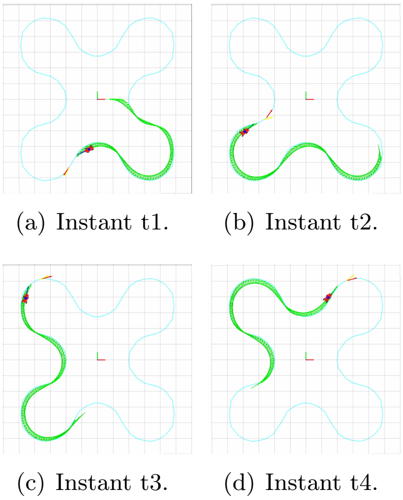
(a) Validation of ANN training: DR-NLSOH way-points estimations (red dots), movement of the mobile robot (blue triangles).

(b) Comparison between different dual-rate estimations: ideal way-points (black dots), DR-FOH way-points estimation (green squares), DR-SOH way-points estimations (blue triangles), DR-NLSOH way-points estimation (red circles). The robot is represented by a big blue triangle, while previously selected way-points are represented by magenta circles.

Figure 4.10: Validation and comparison of the estimations given by the obtained dual-rate nonlinear hold.

on the look-ahead distance criterion. Previous selected way-points are also considered $\mathbf{q}_T(k-1,0)$ and $\mathbf{q}_T(k-2,0)$ since a second order hold is used. As result, the figure shows that the best estimations are produced by the proposed dual-rate nonlinear second order hold.

Figure 4.11 shows the trajectory followed by SUMMIT along the path if the controller is computing control actions at high frequency. Remark that this is the ideal behavior of the robot, although it is not possible to implement such a high frequency controller because technological limitations. On the other hand, Figure 4.12 shows how the system performs when the controller is computing actions at low frequency. This comparison highlights that the low frequency controlled system performs worse rather than the


(a) Instant t1. (b) Instant t2.

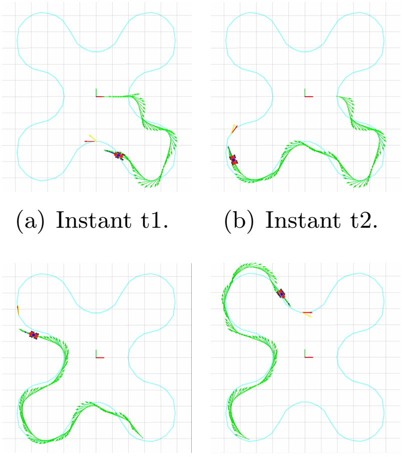
(c) Instant t3. (d) Instant t4.

Figure 4.11: SUMMIT mobile robot working at high frequency.

system controlled at ideal high frequency. Nevertheless, Figure 4.13 shows that the performance obtained by the same system when the dual-rate second order hold is set on it, is very similar to the one obtained by the systems controlled at ideal high frequency.

Similarly, Figure 4.14(c) and Figure 4.14(e) shows the system performance when dual-rate approaches with DR-SOH and DR-NLSOHs are used, respectively. Certainly, the classic single-rate behavior working at low frequency produce a bad performance of the system, while dual-rate approaches present better results, independently the holder used. However, the behavior of the system obtained by DR-NLSOHs is much more similar to the ideal one rather than the obtained using DR-SOHs. This result is also shown in Figures 4.14(d)) and 4.14(f))), where the controller actions of both the ideal single-rate at high frequency and DR-NLSOH are very similar, while the ones produced by using DR-SOH are different, sometimes stronger, increasing thus the abruptness.

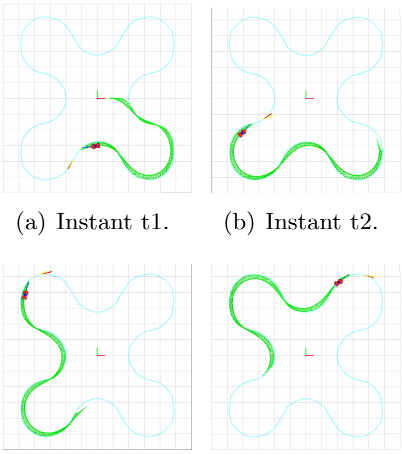
Table 4.1 provides a more accurate analysis based on a benchmarking



(a) Instant t1. (b) Instant t2.

(c) Instant t3. (d) Instant t4.

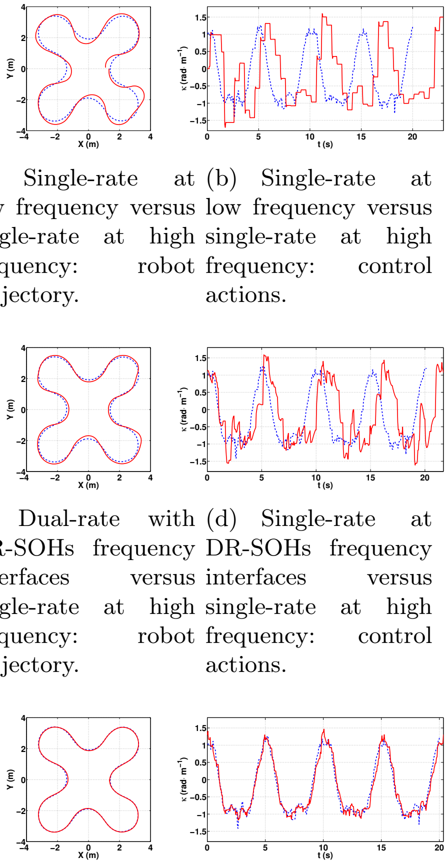
Figure 4.12: SUMMIT mobile robot working at low frequency.



(a) Instant t1. (b) Instant t2.

(c) Instant t3. (d) Instant t4.

Figure 4.13: SUMMIT mobile robot working at high frequency using DR-NLSOHs.



(a) Single-rate at low frequency versus single-rate at high frequency: robot trajectory.

(b) Single-rate at low frequency versus single-rate at high frequency: control actions.

(c) Dual-rate with DR-SOHs frequency interfaces versus single-rate at high frequency: robot trajectory.

(d) Single-rate at DR-SOHs frequency interfaces versus single-rate at high frequency: control actions.

(e) Single-rate at DR-NLSOHs frequency interfaces versus single-rate at high frequency: robot trajectory.

(f) Single-rate at DR-NLSOHs frequency interfaces versus single-rate at high frequency: control actions.

Figure 4.14: Comparison of SUMMIT performance when using conventional and multi-rate approaches (DR-HOhs based on primitive functions and DR-NLHOHs based on learning).

METRIC	SR(δ)	SR($\bar{\delta}$)	DR-SOH	DR-NLSOH
$\frac{1}{N}\sum_t e_t$ [m]	0.0515	0.1036	0.0555	0.0532
$\max_t e_t$ [m]	0.1077	0.4405	0.1882	0.1173
$\frac{1}{N}\sum_t \kappa_t$ [rad/m]	0.7555	0.8198	0.7944	0.7571
$\max_t \lvert\kappa_t\rvert$ [rad/m]	1.4147	1.7045	1.6125	1.4666
$\frac{1}{N}\sum_t \sigma_t$ [rad/m]	1.3411	1.2863	2.0351	1.3791
$\max_t \lvert\sigma_t\rvert$ [rad/m]	9.8470	22.4286	16.5394	10.5302
$\frac{1}{N}\sum_t \kappa_t^2 \cdot v_t \Delta t$ [m^{-1}]	0.0499	0.0632	0.0577	0.0502
$\frac{1}{N}\sum_t \sigma_t^2 \cdot v_t \Delta t$ [rad/mxs]	0.3179	1.1990	1.0364	0.3467

Table 4.1: Metrics of path following comparing Single-rate and Dual-rate approaches.

with different safety-related metrics, Yuste et al. (2010). As expected, the average following error and its maximum value are much bigger on the single-rate approach, while using a dual-rate framework improves the performance. Furthermore, this result demonstrates that the DR-NLSOH is much closer to the ideal case rather than the DR-SOH. Moreover, it can be shown that both maximum of curvature and sharpness are lower and much more similar between the ideal case and the DR-NLSOH than with respect the other approaches, which in general implies more safety and robust behavior.

In addition to this, the Bending Energy and the Abruptness demonstrates that the control with a DR-NLSOH is smoother and needs less energy to follow the path because it does not have to modify the robot direction so many times, due to the fact that the intermediate way-points are estimated more accurately than with DR-SOH.

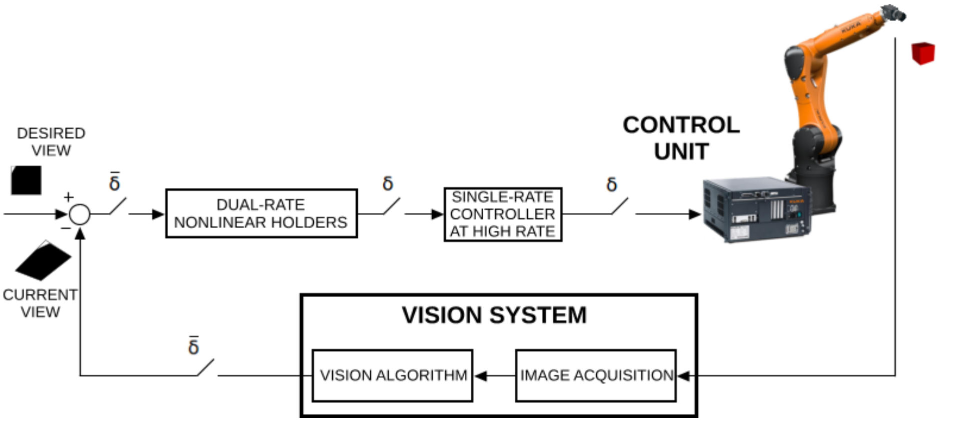


Figure 4.15: Visual feedback robot control using DR-NLHOHs block diagram.

4.4 Visual Feedback Robot Manipulator Control based on Dual-rate Nonlinear High Order Holds

The proposed dual-rate nonlinear holds are used in this section to estimate the intersampling image features in order to perform a visual feedback robot control task, in which the controller is working at high frequency. In this chapter, we focus on the 2D visual feedback control approach with the camera in eye-in hand configuration. Remark that the proposed approach can be used to estimated all kind of signals coming from different systems and workspaces.

Figure 4.15 shows the dual-rate block diagram, which represents a particular case of the full dual-rate scheme shown in Chapter 3, Section 3.3, and also in Solanes et al. (2011). Differently to there, in this section only two sampling rates are considered: the frame period relative to the vision system ($\bar{\delta} = N \cdot \delta$), and the base period relative to the control system (δ). The signal to be estimated is the error between current and desired features of the detected object. Without loss of generality, this section uses features-like-points, being extensible for any class of features.

Kr5 cell platform is used in this section with the same set-up already explained in Chapter 3, Section 3.3.2. In addition, the corresponding virtual Kr5 cell platform has been used for generating the datasets. This platform includes robot physical constraints such as the kinematics and dynamics of each element and it is implemented using OpenRAVE library, Diankov and Kuffner (2008). In this sense, physical joint limits of the robot have been obtained from manufacture's specifications, and inertial parameters have been obtained through finite element software, in this case NX7.5, assuming that the center of gravity of each element is on the geometric center of the element, each element has an uniform density of the mass, and assuming the manufacture's material specification. In addition, it is used the distortion model of the camera, Puget and Skordas (1990), which parameters have been obtained based on calibration tools included in ViSP library, Marchand et al. (2005), taking thus into account camera non-linearities.

Due to the particular real-time needs of the Kr5 robot (see Appendix A for further details), a base period of 10 ms is chosen. On the other hand, our vision system runs at 70 ms, becoming this as frame period $\bar{\delta}$. The relation between both periods is $N = \bar{\delta}/\delta = 7$.

Owing to the results obtained in Chapter 3, where first order hold worked better than second order hold in this particular set up, this section uses first order holds.

The error between the desired and the current features, $\mathbf{e}(k,0) = \mathbf{s}(k,0) - \mathbf{s}^*(k,0)$ is signal to be estimated by the hold. Thus, $\mathbf{e}(k,0) \sim \mathcal{M}(-0.2, 0.2)$ meters is randomly generated, where $\mathcal{M}$ refers to uniform distribution, by moving every 2 frame periods an object to be followed. The input sequence takes the form $\mathbf{I}(k,0) = \{\mathbf{e}(k-1,0), \mathbf{e}(k,0)\}$, being the target sequence $\hat{\mathbf{I}}(k,i) = \{\mathbf{e}(k,0), \ldots, \mathbf{e}(k,N-1)\}$.

In this case, Levenberg-Marquardt back-propagation algorithm implemented in Matlab with a network structure of two-layer feed-forward network with sigmoid hidden neurons and linear output neurons is used with 20 hidden neurons, in order to learn the non-linear function-valued vector.

Figure 4.16 shows, once the dual-rate nonlinear hold has been val-

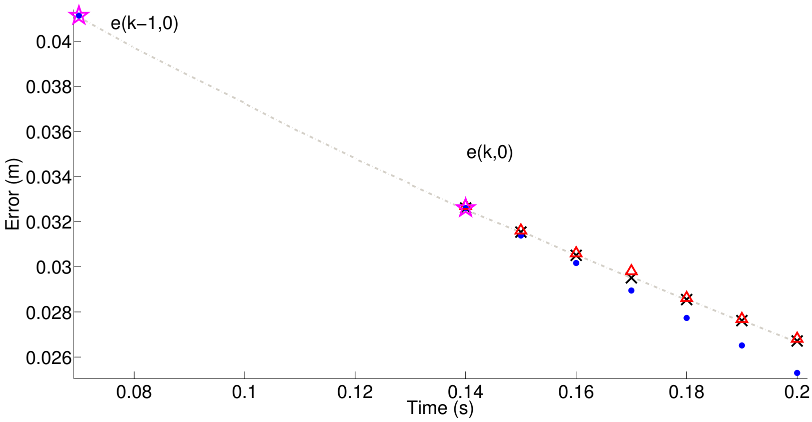
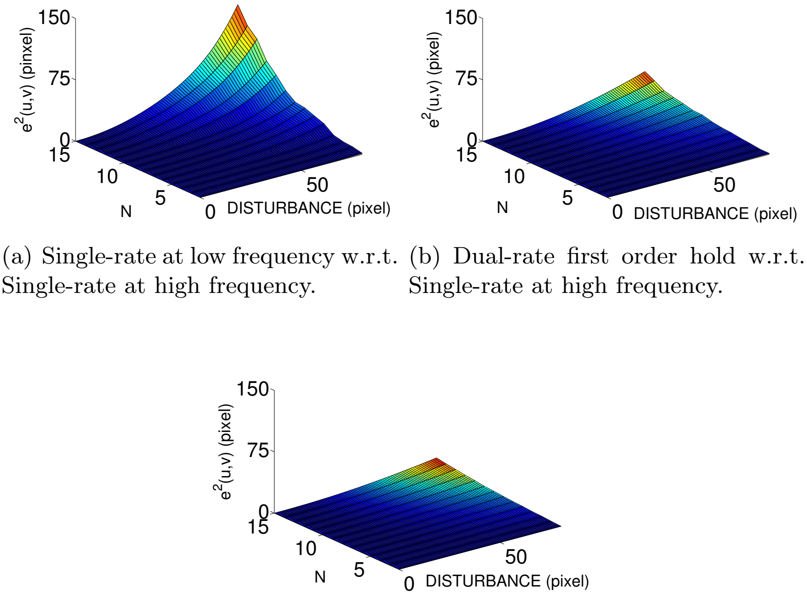


Figure 4.16: Comparison between dual-rate high order hold estimations: ideal (black crosses), dual-rate first order holds (blue dots), dual-rate non-linear first order hold (red triangles), $\mathbf{I}(k) = \{\mathbf{e}(k-1,0), \mathbf{e}(k,0)\}$ (magenta diamonds).

idated using different dataset than the one used for training purposes, the comparative between the estimations provided by DR-FOH and its equivalent DR-NLFOH. It can be seen that estimations provided by DR-NLFOH are almost match to the ideal ones, while DR-FOH estimations differ much more.

4.4.1 Results

In order to demonstrate how similar is the performance of the dual-rate approach with respect to the ideal single-rate at high frequency, a comparative analysis is shown under simulated scenario. The sampling base period has been kept constant and equal to 10 ms as in the real case. The frame-period is changed in order to analyze how the ratio N affects to the estimations. For every experiment, a glitch displacement of the object is done on random directions and magnitudes. The aim of this experimentation is to analyze



(a) Single-rate at low frequency w.r.t. Single-rate at high frequency.

(b) Dual-rate first order hold w.r.t. Single-rate at high frequency.

(c) Dual-rate non-linear hold w.r.t. Single-rate at high frequency.

Figure 4.17: Disturbance Analysis: Mean Square Error [pixel] with respect to single-rate at high frequency.

the behavior on a situation with sudden changes, where the feature extraction fails by detecting an incorrect object. Results in Figure 4.17 show that DR-NLFOHs provide a solution similar to the single-rate at high frequency.

The validation of the obtained dual-rate non-linear holds is carried out by using the real Kr5 platform. Two different kind of experiments are performed: firstly, a stationary object formed by four dots is used to perform an error positioning task (locating the robot end-effector in a specific pose relative to the object); the second set-up is carried out by using a moving object formed by a single dot projected on a board, with an over-head

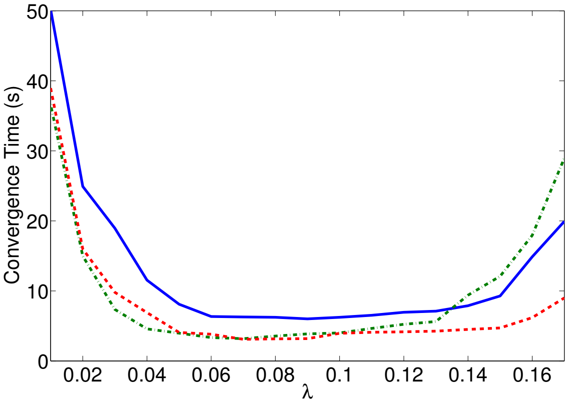
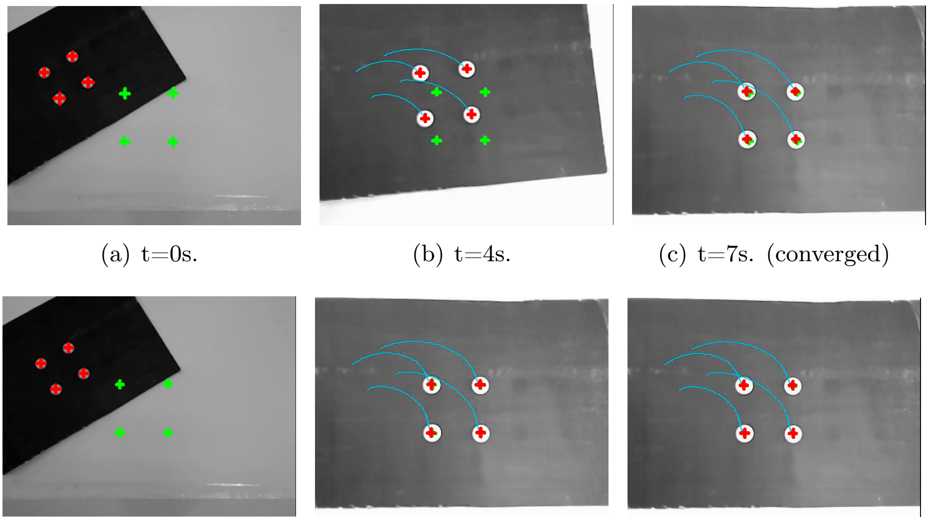


Figure 4.18: IBVS convergence time: single-rate at low frequency (blue solid line), dual-rate first order holds (green dashed-dot line), dual-rate non-linear holds (red dashed line).

projector (this set-up is fully described in Chapter 3 Section 3.3). The object in this experiment is moving following a square trajectory at different velocities.

Figure 4.18 shows a comparison of the convergence time in function of the proportional gain (λ) for the single-rate approach and conventional dual-rate high order holds described in this chapter. An experiment finishes when the error between desired and current features is less than 0.01 meters in normalized coordinates, or the controller produces instabilities such as system overshooting greater than 20%. Moreover, each result is obtained as a mean of 10 experiments. As conclusion, our dual-rate approach introduces higher stability margin.

Figure 4.19 shows a sequence of frames at different time instants in order to compare single-rate at low frequency and dual-rate using DR-NLFOHs. The dual-rate approach using DR-NLFOHs provides a good performance of features trajectories and the algorithm convergence is around two times faster than conventional single-rate approach and nearly the same conver-



(a) t=0s.

(b) t=4s.

(c) t=7s. (converged)

(d) t=0s.

(e) t=4s. (converged)

(f) t=4s. (converged)

Figure 4.19: Image plane trajectories with $\lambda = 0.08$: (a), (b) and (c) single-rate at low frequency and (d), (e) and (f) dual-rate non-linear holds; current features (red crosses) and Desired features (green crosses).

gence time than DR-FOH, but with higher stability margin.

Moreover, Figure 4.20 shows the object following experiment as described above. Once again, DR-NLFOHs present less quadratic error than the system with classical dual-rate first order holds (see Table 4.2), due to their better predictions, especially when the object changes its trajectory. At all events, the performed visual feedback control using dual-rate approach has an important improvement in the object tracking with respect to single-rate cases at low sampling frequencies.

4.5 Summary

In this chapter a generalization of dual-rate high order holds has been presented in order to take into account the dynamic of non-linear systems and

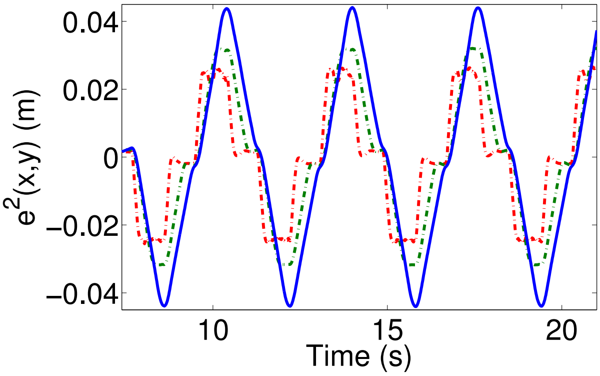


Figure 4.20: $e^2(x, y)$ with optimal gain (λ): single-rate at low frequency (blue solid line), dual-rate first order hold (green dashed line), dual-rate non-linear hold (red dot-dashed line).

Optimal λ		SR-LF	DR-FOH	DR-SOH	DR-SOBH	DR-SOTH	DR-NLHOH
$v_{obj} = 10cm/s$	MSE(m)	0.02987	0.02078	0.02013	0.02192	0.02102	0.01998
$v_{obj} = 13cm/s$	MSE(m)	0.04135	0.03248	0.03652	0.03715	0.03773	0.024183
$v_{obj} = 16cm/s$	MSE(m)	0.06223	0.03938	0.04032	0.04193	0.04109	0.03389

Table 4.2: Single-rate VS dual-rate approaches: object moving at different velocities following a square trajectory.

controllers. Using machine learning techniques, and in particular artificial neural networks (ANN), a new concept of dual-rate non-linear hold (DR-NLHOH) has been formulated.

In that sense, the chapter establishes a general methodology for computing DR-NLHOHs based on artificial neural networks, based on datasets set from virtual environments.

The approach has been validated in two different platforms: a wheeled robot performing a path following task, and an industrial robot manipulator performing a visual feedback control error positioning and tracking tasks.

An analysis of similarities and differences between single-rate working at low and single-rate working at high frequency has been made, showing that the behavior of the system using DR-NLHOHs is quite similar to the ideal working at high frequency and therefore much better than classic control at low frequency.

In addition, the comparison between DR-NLHOHs and classic DR-HOHs has showed that, although both of them improve the classic single-rate behavior, the system using DR-NLHOHs produces better estimations rather than DR-HOHs.

Chapter 5

Visual Feedback Control based on Reference Features Filtering Control Strategy

5.1 Motivation

The problem of finding suitable trajectories for the measurement in the image still being nowadays, an important issue in robot visual feedback control.

In Todorov (2008), a stochastic optimal control is presented, whose result is dual to an estimation problem. Zima et al. (2013) demonstrates and solves the implementation for such an estimation problem, solving non-linear LQR-like problems more efficiently than iterative strategies such as iLQR, Todorov (2005). The authors derive an optimal controller, with the appropriate setting of Kalman matrices dual to a LQR problem, using a non-linear EKF predictor-smoother (ERTS), based on Rauch-Tung-Striebel (RTS) algorithm, Rauch et al. (1965), which provides an optimal state trajectory. Such trajectory is used to compute the optimal control law based on Least Squares (LS).

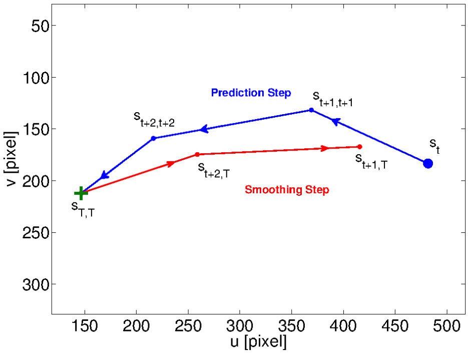


Figure 5.1: Principle of Reference Filtering control strategy (example with T=t+3).

Our concerned is not in providing an optimal controller, but robust solutions for visual feedback robot control problems using well-known controllers, being the main idea of the approach presented in this chapter to use optimal trajectories proposed by ERTS as reference.

In addition, the generalization of the reference features filtering algorithm proposed, from a dual-rate point of view, is presented in this chapter, giving a solution to the sensor latency problem.

5.2 Reference Features Filtering Control Strategy

The idea behind the proposed approach is to use the solution given by ERTS algorithm as reference for an underlying controller. By doing this, 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 subject to the open-loop model behavior using reference as observation, as shown in blue Figure 5.1. Then, a Gaussian Smoother is used to provide an optimal trajectory, as shown in the red dots and line trajectory of Figure 5.1.

The algorithm structure of the proposed reference filtering control strategy contains three main components. As a consequence, the explanation of the algorithm is divided into three parts:

- Kalman Filter (subsection 5.2.1)
- Extended Kalman Filter/Smoother (subsection 5.2.2)
- Visual Feedback Control (subsection 5.2.3)

In Algorithm 4 there is an initialization of state (line 3) and its covariance matrix (line 4); afterwards, it loops, from line 6 to 14 until convergence, i.e. until detected features match with reference ones. Line 10 computes filtered feature references with the dual-sampling Extended Kalman Filter/Smoother. A generic underlying visual feedback controller (VCF) is used in line 11 to generate the correspondent control actions for each sampling instant. The algorithm uses generic call functions such as “GetNewImage”, “GetFeatures”, “ApplyControl” and “Wait” to complete the remainder of functionalities also required in standard visual feedback control applications.

As reminder, along this chapter the base period (control period) is denoted as δ and the frame period is denoted as $\bar{\delta} = N\delta$.

5.2.1 Kalman Filter (KF)

A conventional Kalman filter (KF) is used to estimate the position and velocity of the set of features, $\mathbf{s}_t$. In this sense, a periodic stochastic linear model based on the well-known constant velocity model is employed.

$$\begin{bmatrix}\hat{\mathbf{s}}_t\\ \hat{\dot{\mathbf{s}}}_t\end{bmatrix} = \begin{bmatrix}\mathbf{I} & N\delta\cdot\mathbf{I}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{s}}_{t-1}\\ \hat{\dot{\mathbf{s}}}_{t-1}\end{bmatrix} + \begin{bmatrix}\frac{(N\delta)^2}{2}\cdot\mathbf{I}\\ N\delta\cdot\mathbf{I}\end{bmatrix}\mathbf{w}_{t-1} \tag{5.1}$$

$$\mathbf{s}_t = \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{s}}_t\\ \hat{\dot{\mathbf{s}}}_t\end{bmatrix} + \mathbf{v}_t \tag{5.2}$$

with $\hat{\mathbf{s}}_t$ is the estimated feature vector, $\hat{\dot{\mathbf{s}}}_t$ its velocity and $\bar{\mathbf{P}}_t$ its joint covariance matrix, $\mathbf{w}_{t-1}$ is the acceleration (assumed to be a white noise

Algorithm 4 Reference filtering control strategy for robot visual feedback control

```
$\mathcal{I}_t$ = GetNewImage()
$\mathbf{s}_{t|t}$ = GetFeatures($\mathcal{I}_t$)
$\hat{\dot{\mathbf{s}}}_{t|t} = \mathbf{0}$
$\bar{\mathbf{P}}_{t|t} = \mathbf{P}_0$
$\mathbf{R}_{N\delta}$, $\mathbf{Q}_{N\delta}$
// Loop until convergence
while $\|\mathbf{s}_t - \mathbf{s}_t^*\| <$ error do
    $\mathcal{I}_t$ = GetNewImage()
    $\mathbf{s}_t$ = GetFeatures($\mathcal{I}_t$)
    $\left[\bar{\hat{\mathbf{s}}}_{t|t}, \hat{\dot{\mathbf{s}}}_{t|t}, \bar{\mathbf{P}}_{t|t}\right]$=KF($\mathbf{s}_t, \hat{\mathbf{s}}_{t|t}, \hat{\dot{\mathbf{s}}}_{t|t}, \bar{\mathbf{P}}_{t|t}$)
    $\bar{\hat{\mathbf{s}}}^*_{d,t+1}$=EKFS($\bar{\hat{\mathbf{s}}}_{t|t}, \mathbf{s}_t^*, \mathbf{q}_t, \mathbf{R}_{N\delta}, \mathbf{Q}_{N\delta}$)
    $\dot{\mathbf{q}}_t^*$ = VFC($\mathbf{q}_t, \bar{\hat{\mathbf{s}}}_{t|t}, \bar{\hat{\mathbf{s}}}^*_{d,t+1}$)
    ApplyControl($\dot{\mathbf{q}}_t^*$)
    Wait($N\delta$)
end while
```

$\mathbf{w}_{t-1} \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{Q}})$) and $\tilde{\mathbf{Q}}$ a covariance matrix. In addition to this, $\mathbf{s}_t$ is treated here as measurement and $\mathbf{v}_t$ as measurement noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{R}})$, with covariance matrix $\tilde{\mathbf{R}}$. It is convenient to remark that if the control is performed into the image plane (2D), the upper index n of Table 5.1 represents the number of detected features. On the contrary, if the control is performed in the Cartesian space (3D), the dimensions of previous matrices are constant because the state is related to the translation and rotation of the object frame, $\mathcal{F}_o$.

5.2.2 Extended Kalman Filter/Smoother (EKFS)

The state for EKFS represents a filtered feature reference. Its initialization is made through variables $\bar{\hat{\mathbf{s}}}_{t|t}$, $\mathbf{q}_t$, $\mathbf{s}_t^*$. Since they are known, the initial covariance $\mathbf{P}_{t|t}$ is initialized to a zero covariance matrix. Reference estimation

Table 5.1: Variable dimensions w.r.to the workspace in which the control task is being performed.

variable	Dimensions in 2D	Dimensions in 3D
$\hat{\mathbf{s}}_t$	$\Re^{3n}$	$\Re^{6}$
$\bar{\mathbf{P}}_t$	$\Re^{6n\times 6n}$	$\Re^{6\times 6}$
$\mathbf{w}_{t-1}$	$\Re^{3n}$	$\Re^{6}$
$\tilde{\mathbf{Q}}$	$\Re^{3n\times 3n}$	$\Re^{6\times 6}$
$\mathbf{v}_t$	$\Re^{3n}$	$\Re^{6}$
$\tilde{\mathbf{R}}$	$\Re^{3n\times 3n}$	$\Re^{6\times 6}$
$\tilde{\mathbf{I}}$	$\Re^{3n\times 3n}$	$\Re^{6\times 6}$

evolves during a given time horizon $T = t+h$, where h is the number of iterations expressed in frame period units representing a prediction horizon.

During the EKF step, the system evolves in "open-loop" since future control inputs are unknown and thus treated as Gaussian noises, and for that reason Algorithm 5 assumes null input. Target features $\mathbf{s}^*_t$ are taken as "observations" to ensure global convergence of the trajectory and compared with the predicted state to perform the update step. As summary, the EKF provides a trajectory set:

$$\{\hat{\mathbf{s}}^*_{t+1|t+1}, \hat{\mathbf{s}}^*_{t+2|t+2}, ..., \hat{\mathbf{s}}^*_{t+T|t+T}\} \tag{5.3}$$

The Extended Kalman Smoother (EKS step) performs a backward estimation by taking into account "predicted" estimations. Thus, the EKS step generates estimations under the form:

$$\{\hat{\mathbf{s}}^*_{T|T}, \hat{\mathbf{s}}^*_{T-1|T}, \hat{\mathbf{s}}^*_{T-2|T}, ..., \hat{\mathbf{s}}^*_{t+1|T}\} \tag{5.4}$$

It is interesting to note that the smoothing step in Rauch et al. (1965) also computes state covariance during backward smoothing, however in the

Algorithm 5 Extended Kalman Filter/Smoother (EKFS)

```
Input: $\mathbf{s}_t$, $\mathbf{s}_t^*$, $\mathbf{q}_t$, $\mathbf{R}_{N\delta}$ and $\mathbf{Q}_{N\delta}$
Definitions: $T = t + h$
    // EKF Step
    $\hat{\mathbf{s}}^*_{t|t} = \mathbf{s}_t$, $\mathbf{P}_{t|t} = \mathbf{0}$, $\mathbf{C} = \mathbf{I}$
    for $\tau = t+1, \ldots, T$ do
        $[\hat{\mathbf{s}}^*_{\tau|\tau}, \mathbf{P}_{\tau|\tau}] = EKF(\mathbf{s}_t^*, \hat{\mathbf{s}}^*_{\tau-1|\tau-1}, \mathbf{P}_{\tau-1|\tau-1}, \mathbf{Q}_{N\delta}, \mathbf{R}_{N\delta})$
    end for
    // EKS Step
    for $\tau = T-1, \ldots, t+1$ do
        $\hat{\mathbf{s}}^*_{\tau|T} = \hat{\mathbf{s}}^*_{\tau|\tau} + \mathbf{P}_{\tau|\tau} \cdot \mathbf{P}^{-1}_{\tau+1|\tau} \cdot (\hat{\mathbf{s}}^*_{\tau+1|T} - \hat{\mathbf{s}}^*_{\tau+1|\tau})$
    end for
    return $\bar{\hat{\mathbf{s}}}^*_{d,t+1} \equiv \{\hat{\mathbf{s}}^*_{t+1|T}, ..., \hat{\mathbf{s}}^*_{t+N|T}\}$
```

proposed algorithm, the corresponding equations are not implemented since they are not necessary.

The well known local model based on the interaction matrix, Allibert and Courtial (2009); Allibert et al. (2010), is used to perform the filter-smoother step. This model is linearized and "discretized" upon the sampling time[1]. Thus,

$$\hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}_t + \mathbf{J}_{N\delta}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \dot{\mathbf{q}}_t \tag{5.5}$$

with $\mathbf{J}_{N\delta}(\hat{\mathbf{s}}_t, \mathbf{q}_t) = N\delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t)$.

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

Similarly, covariances of EKF must be appropriately computed:

$$\mathbf{Q}_{N\delta} = N\delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) \tag{5.6}$$

$$\mathbf{R}_{N\delta} = \frac{(N\delta)^3}{3} \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) + \mathbf{R}_c N\delta \tag{5.7}$$

Algorithm 5 returns the set of reference features filtered by the smoother.

5.2.3 Visual Feedback Control

As mentioned hereinafter, the generic visual feedback controller in Algorithm 4 can be implemented in 2D and 3D workspaces. This chapter provides analysis and examples using well-know P and PID controllers. Let us summarize the setting-up of states $\mathbf{s}_t$, as well as the controller used in each workspace.

2D visual feedback control schemes (IBVS), Corke (2011); Chaumette and Hutchinson (2006), use the image-plane coordinates of a set of points to define the set $\mathbf{s}_t$. Let us denote $\mathbf{f}_t$ and $\mathbf{f}_t^*$ as the current and reference image state, respectively. $\mathbf{s}_t = \{\mathbf{f}_1, ..., \mathbf{f}_n\}_t \in \Re^{3n}$ is the set of n current state and $\mathbf{s}_t^* = \{\mathbf{f}_1^*, ..., \mathbf{f}_n^*\}_t \in \Re^{3n}$ the set of n reference state.

Without loss of generality, this chapter considers features-like-points, $\mathbf{P}_t = (X_t, Y_t, Z_t)^T \in \Re^3$ in the Cartesian workspace, to perform vision feedback control tasks, being extensible to any other kind of features. In this chapter, the depth parameter Z_t is getting updated, Chaumette and Hutchinson (2006), by using Dementhon and Davis pose estimation algorithm, DeMenthon and Davis (1995b). As a consequence, the estimation of Z_t has to be estimated so that the state is extended as $\mathbf{f}_t = (x_t, y_t, Z_t)^T \in \Re^3$, with $x_t = X_t/Z_t$ and $y_t = Y_t/Z_t$ the coordinates being expressed in meters in the camera workspace and the interaction matrix described in Chaumette and Hutchinson (2006) is extended as,

$$\mathbf{L}_\mathbf{f}(\mathbf{f}_t) = \begin{bmatrix} -\frac{1}{Z_t} & 0 & \frac{x_t}{Z_t} & x_t \cdot y_t & -(1+x_t^2) & y_t \\ 0 & -\frac{1}{Z_t} & \frac{y_t}{Z_t} & 1+y_t^2 & -x_t \cdot y_t & -x_t \\ 0 & 0 & -1 & -y_t \cdot Z_t & x_t \cdot Z_t & 0 \end{bmatrix} \in \Re^{3x6} \tag{5.8}$$

being $\mathbf{L_s}(\mathbf{s}_t) = [\mathbf{L_f}_1 \cdots \mathbf{L_f}_n]^T \in \Re^{nx6}$.

On the other hand, 3D visual feedback control schemes, Martinet et al. (1996); Chaumette and Hutchinson (2006); Siciliano and Khatib (2008); Corke (2011), also known as position based visual servoing (PBVS), use the pose of the camera with respect to some reference coordinate frame to define $\mathbf{s}_t$.

Let us denote $\mathcal{F}_c$ the current camera frame, $\mathcal{F}_c^*$ the desired camera frame, and $\mathcal{F}_o$ the object frame. Here the standard notation of using a leading superscript to denote the frame with respect to which a set of coordinates is adopted. Furthermore, let ${}^{\mathbf{c}^*}\mathbf{t_c}$ and ${}^{\mathbf{c}^*}\mathbf{R_c}$ be the translation and rotation matrix that gives the orientation of the current camera frame relative to the desired frame. Thus, in case the interaction matrix is,

$$\mathbf{L_s}(\mathbf{s}_t) = \begin{bmatrix} {}^{\mathbf{c}^*}\mathbf{R_c} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix} \in \Re^{6x6} \tag{5.9}$$

in which $\mathbf{L}_{\theta\mathbf{u}}$ is given by:

$$\mathbf{L}_{\theta\mathbf{u}} = \mathbf{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{sinc(\theta)}{sinc^2(\frac{\theta}{2})}\right) \cdot [\mathbf{u}]_\times^2 \in \Re^{3x3} \tag{5.10}$$

where $\mathbf{I}_3$ is the 3x3 identity matrix, $sinc(x)$ is the sinus cardinal defined such that $x \cdot sinc(x) = \sin x$, $sinc(0) = 1$ and $\mathbf{s}_t$ is $(\mathbf{t}, \theta\mathbf{u})$.

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

According to the **non-free flying camera** case $\mathbf{J}(\mathbf{s}_t, \mathbf{q}_t) = \mathbf{L_s}(\mathbf{s}_t) \cdot {}^{\mathbf{c}}\mathbf{V_r} \cdot {}^{\mathbf{r}}\mathbf{J_r}(\mathbf{q}_t) \in \Re^{3 \times m}$ is the Jacobean matrix, which relates the measured features velocity with the manipulator velocity, where ${}^{\mathbf{r}}\mathbf{J_r}(\mathbf{q}_t) \in \Re^{m \times m}$ is the robot Jacobean matrix, ${}^{\mathbf{c}}\mathbf{V_r} \in \Re^{6 \times m}$ is the twist velocity matrix, $\mathbf{L_s}(\mathbf{s}_t) \in \Re^{3n \times 6}$ the interaction matrix and $\mathbf{q}_t \in \Re^m$ and $\dot{\mathbf{q}}_t \in \Re^m$ are the robot positions and velocities, respectively, being m the manipulator number of DOF.

Thus, the general control law is given by:

$$\dot{\mathbf{q}}_t = \hat{\mathbf{J}}^+(\mathbf{s}_t, \mathbf{q}_t) \cdot \dot{\mathbf{s}}_t \tag{5.11}$$

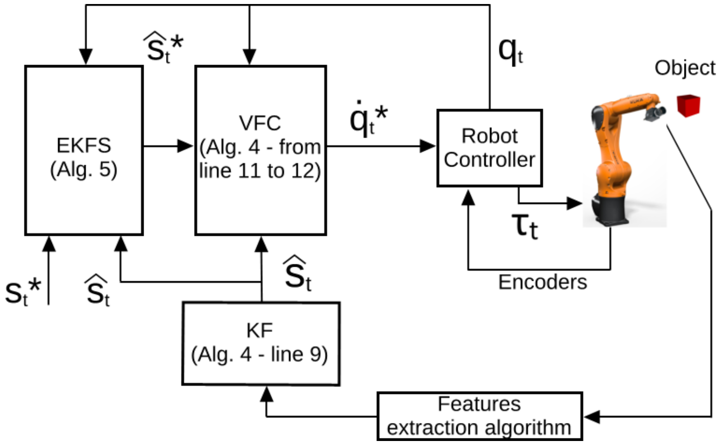


Figure 5.2: Single-rate Reference Features Filtering strategy control block diagram.

To obtain a pure image-based control, which is to reduce the interaction matrix given in equation (5.8), an additional $\mathbf{H} \in \Re^{3n\times3n}$ block-diagonal matrix defined by $\mathbf{H} = \mathrm{diag}(\phi)$, with $\phi = \left[\begin{smallmatrix}1&0&0\\0&1&0\\0&0&0\end{smallmatrix}\right]$, is introduced in (5.11), obtaining the 2D visual feedback controller used in this chapter:

$$\dot{\mathbf{q}}_t = \hat{\mathbf{J}}^+(\mathbf{s}_t, \mathbf{q}_t) \cdot \mathbf{H} \cdot \dot{\mathbf{s}}_t \tag{5.12}$$

Along this chapter two controllers are used: a proportional control law, Hashimoto (1993); Corke (2011), and a PID controller, Lots et al. (2001).

Finally, Figure 5.2 shows the control block diagram of the proposed dual-rate reference filtering control strategy approach.

5.3 Dual-rate Reference Filtering Control Strategy

In this section, the reference filtering approach of Algorithm 4 is extended from a dual-rate point of view. In this sense, besides providing more suitable trajectories for the measurements in the image, the following generalization will allow us to deal with the problem of the sensor latency.

As occurred in the single-rate case, Algorithm 6 contains three main components. As a consequence, the explanation of the algorithm is divided into three parts:

- Dual-rate Kalman Filter (subsection 5.3.1)
- Dual-sampling Extended Kalman Filter/Smoother (subsection 5.3.2)
- Visual Feedback Control (subsection 5.3.3)

In the proposed algorithm 6 there is an initialization of state (line 3) and its covariance matrix (line 3); afterwards, it loops, from line 6 to 16 until convergence, i.e. until detected features match with reference ones. Line 10 computes filtered feature references with the Dual-Sampling Extended Kalman Filter/Smoother (DS-EFFS). A generic underlying visual feedback controller (VCF) is used in line 12 to generate the correspondent control actions for each sampling instant (code from 11 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 feedback control applications.

5.3.1 Dual-rate Kalman Filter (DR-KF)

A dual-rate Kalman filter, Armesto and Tornero (2006); Steffes (2014), 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 Armesto and Tornero (2006). In this sense, the periodic stochastic linear model given in Equation 5.1 sampled at high frequency (δ) takes the form of:

$$\begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\dot{\mathbf{s}}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \delta\cdot\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_{t-1} \\ \hat{\dot{\mathbf{s}}}_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\delta^2}{2}\cdot\mathbf{I} \\ \delta\cdot\mathbf{I} \end{bmatrix} \mathbf{w}_{t-1} \tag{5.13}$$

$$\mathbf{s}_t = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\dot{\mathbf{s}}}_t \end{bmatrix} + \mathbf{v}_t \tag{5.14}$$

Algorithm 6 Dual-rate Reference Filtering for visual feedback control

```
$\mathcal{I}_{t-N}$ = GetNewImage()
$\mathbf{s}_{t-N|t-N}$ = GetFeatures($\mathcal{I}_{t-N}$)
$\hat{\dot{\mathbf{s}}}_{t-N|t-N} = \mathbf{0}$
$\bar{\mathbf{P}}_{t-N|t-N} = \mathbf{P}_0$
Wait($\delta \cdot N$)
// Loop until convergence
while $\|\mathbf{s}_t - \mathbf{s}^*_t\| < \text{error}$ do
  $\mathcal{I}_t$ = GetNewImage()
  $\mathbf{s}_t$ = GetFeatures($\mathcal{I}_t$)
  $\left[\bar{\hat{\mathbf{s}}}_{t|t}, \hat{\dot{\mathbf{s}}}_{t|t}, \bar{\mathbf{P}}_{t|t}\right]$=DR-KF($\mathbf{s}_t, \hat{\mathbf{s}}_{t-N|t-N}, \hat{\dot{\mathbf{s}}}_{t-N|t-N}, \bar{\mathbf{P}}_{t-N|t-N}$)
  $\bar{\hat{\mathbf{s}}}^*_{d,t+1}$=DS-EKFS($\bar{\hat{\mathbf{s}}}_{t|t}, \mathbf{q}_t, \mathbf{s}^*_t$)
  for $i = 0, \ldots, N-1$ do
    $\dot{\mathbf{q}}^*_t$ = VFC($\mathbf{q}_{t+i}, \bar{\hat{\mathbf{s}}}_{t+i|t+i}, \bar{\hat{\mathbf{s}}}^*_{d,t+i+1}$)
    ApplyControl($\dot{\mathbf{q}}^*_t$)
    Wait($\delta$)
  end for
end while
```

with δ being the sampling time at base-period, $\hat{\mathbf{s}}_t$ is the estimated feature vector, $\hat{\dot{\mathbf{s}}}_t$ its velocity and $\bar{\mathbf{P}}_t$ its joint covariance matrix, $\mathbf{w}_{t-1}$ is the acceleration (assumed to be a white noise $\mathbf{w}_{t-1} \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{Q}})$) and $\tilde{\mathbf{Q}}$ a covariance matrix. In addition to this, $\mathbf{s}_t$ is treated here as measurement and $\mathbf{v}_t$ as measurement noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{R}})$, with covariance matrix $\tilde{\mathbf{R}}$ (see Table 5.1 for matrices and vectors dimension).

To derive the lifted model, the following lifted vectors are considered:

$$\bar{\mathbf{W}}_t = \left[\mathbf{w}_t^T(0) \quad \mathbf{w}_t^T(1) \quad \ldots \quad \mathbf{w}_t^T(N-1)\right]^T \in \Re^{3nN} \tag{5.15}$$

$$\bar{\mathbf{S}}_t = \left[\mathbf{s}_t^T(0) \quad \mathbf{s}_t^T(1) \quad \ldots \quad \mathbf{s}_t^T(N-1)\right]^T \in \Re^{3nN} \tag{5.16}$$

$$\bar{\mathbf{V}}_t = \left[\mathbf{v}_t^T(0) \quad \mathbf{v}_t^T(1) \quad \ldots \quad \mathbf{v}_t^T(N-1)\right]^T \in \Re^{3nN} \tag{5.17}$$

Since features are measured once every frame period δ, the sparse lifted vectors are (see Armesto and Tornero (2006) for details):

$$\bar{\mathbf{w}}_t = \left[\mathbf{w}_t^T(0) \quad \mathbf{w}_t^T(1) \quad \dots \quad \mathbf{w}_t^T(N-1)\right]^T \in \Re^{3nN} \tag{5.18}$$

$$\bar{\mathbf{s}}_t = \left[\mathbf{s}_t^T(0)\right]^T \in \Re^{3n} \tag{5.19}$$

$$\bar{\mathbf{v}}_t = \left[\mathbf{v}_t^T(0)\right]^T \in \Re^{3n} \tag{5.20}$$

leading to the following lifted stochastic model:

$$\begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\dot{\mathbf{s}}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{I} & N\delta\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_{t-N} \\ \hat{\dot{\mathbf{s}}}_{t-N} \end{bmatrix} + \begin{bmatrix} \frac{(2N-1)\delta^2}{2}\mathbf{I} & \frac{(2N-3)\delta^2}{2}\mathbf{I} & \dots & \frac{\delta^2}{2}\mathbf{I} \\ \delta\mathbf{I} & \delta\mathbf{I} & \dots & \delta\mathbf{I} \end{bmatrix} \bar{\mathbf{w}}_t \tag{5.21}$$

$$\bar{\mathbf{s}}_t = \begin{bmatrix} \mathbf{I} & \delta\mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\dot{\mathbf{s}}}_t \end{bmatrix} + \bar{\mathbf{v}}_t \tag{5.22}$$

5.3.2 Dual-sampling Extended Kalman Filter/Smoother(DS-EKFS)

The state for EKFS represents a filtered feature reference. Its initialization is made through variables $\bar{\hat{\mathbf{s}}}_{t|t}$, $\mathbf{q}_t$, $\mathbf{s}_t^*$. Since they are known, the initial covariance $\mathbf{P}_{t|t}$ 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. By using the dual-sampling approach, the computational time is reduced N times (in the limit case) compared to the fixed step approach.

The dual-sampling approach is shown in Algorithm 7, where the first N iterations of the fast-sampling Extended Kalman Filter (EKF) evolves at control period (lines 2 to 4), while for the remainder of iterations, a slow-sampling EKF evolves at vision period (lines 5 to 8). During the EKF step, the system evolves in "open-loop" since future control inputs are unknown and thus treated as Gaussian noises, and for that reason Algorithm 7 assumes null input. Target features $\mathbf{s}_t^*$ are taken as "observations" to ensure

Algorithm 7 Dual-sampling Extended Kalman Filter/Smoother (DS-EKFS)

```
Input: $\mathbf{s}_t$, $\mathbf{q}_t$ and $\mathbf{s}^*_t$
Definitions: $T = t + h \cdot N$
    // Fast-sampling EKF Step
    $\hat{\mathbf{s}}^*_{t|t} = \mathbf{s}_t$, $\mathbf{P}_{t|t} = \mathbf{0}$, $\mathbf{C} = \mathbf{I} \in \Re^{3n \times 3n}$
    for $\tau = t+1, \dots, t+N$ do
        $[\hat{\mathbf{s}}^*_{\tau|\tau}, \mathbf{P}_{\tau|\tau}] = EKF(\mathbf{s}^*_t, \hat{\mathbf{s}}^*_{\tau-1|\tau-1}, \mathbf{P}_{\tau-1|\tau-1}, \mathbf{Q}_\delta, \mathbf{R}_\delta)$
    end for
    // Slow-sampling EKF Step
    for $l = 2, \dots, h$ do
        $\tau = t + lN$
        $[\hat{\mathbf{s}}^*_{\tau|\tau}, \mathbf{P}_{\tau|\tau}] = EKF(\mathbf{s}^*_t, \hat{\mathbf{s}}^*_{\tau-N|\tau-N}, \mathbf{P}_{\tau-N|\tau-N}, \mathbf{Q}_{N\delta}, \mathbf{R}_{N\delta})$
    end for
    // Slow-sampling EKS Step
    for $l = h, \dots, 2$ do
        $\tau = t + (l-1)N$
        $\hat{\mathbf{s}}^*_{\tau|T} = \hat{\mathbf{s}}^*_{\tau|\tau} + \mathbf{P}_{\tau|\tau} \cdot \mathbf{P}^{-1}_{\tau+N|\tau} \cdot (\hat{\mathbf{s}}^*_{\tau+N|T} - \hat{\mathbf{s}}^*_{\tau+N|\tau})$
    end for
    // Fast-sampling EKS Step
    for $\tau = t+N-1, \dots, t+1$ do
        $\hat{\mathbf{s}}^*_{\tau|T} = \hat{\mathbf{s}}^*_{\tau|\tau} + \mathbf{P}_{\tau|\tau} \cdot \mathbf{P}^{-1}_{\tau+1|\tau} \cdot (\hat{\mathbf{s}}^*_{\tau+1|T} - \hat{\mathbf{s}}^*_{\tau+1|\tau})$
    end for
    return $\bar{\hat{\mathbf{s}}}^*_{d,t+1} \equiv \{\hat{\mathbf{s}}^*_{t+1|T}, \dots, \hat{\mathbf{s}}^*_{t+N|T}\}$
```

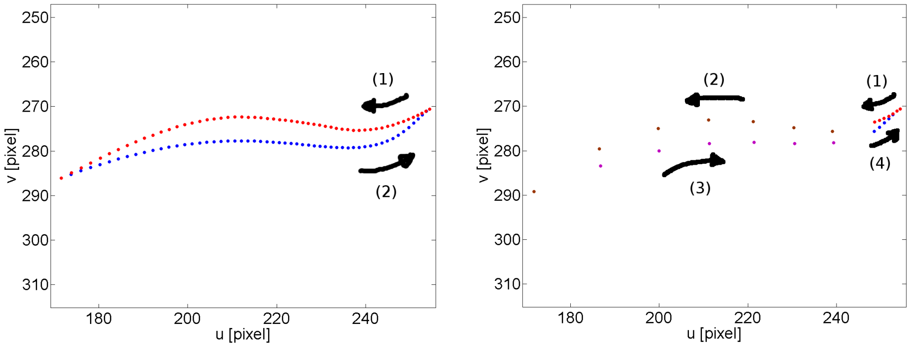


Figure 5.3: Comparison of the trajectories estimated by both (high frequency single-rate) EKFS and DR-EKFS. The case shown corresponds to a parameter setting of $N = 8$, being 56 the prediction horizon : high frequency prediction step (1), red dots; low frequency prediction step (2), brown dots; low frequency smoothing step (3), magenta dots; high frequency smoothing step (4), blue dots.

global convergence of the trajectory and compared with the predicted state to perform the update step. As summary, the EKF provides a trajectory set:

$$\{\hat{\mathbf{s}}^*_{t+1|t+1}, \hat{\mathbf{s}}^*_{t+2|t+2}, ..., \hat{\mathbf{s}}^*_{t+N|t+N}, \hat{\mathbf{s}}^*_{t+2N|t+2N}, ..., \hat{\mathbf{s}}^*_{T|T}\} \tag{5.23}$$

The Extended Kalman Smoother (EKS step) performs a backward estimation by taking into account “predicted” estimations. Here, a double-integration step is performed, in which first iterations evolve at frame period (lines 9 to 12 of Algorithm 7) backward in time, while the last N iterations evolve at base period (13 to 15). Thus, the EKS step generates estimations under the form:

$$\{\hat{\mathbf{s}}^*_{T|T}, \hat{\mathbf{s}}^*_{T-N|T}, \hat{\mathbf{s}}^*_{T-2N|T}, ..., \hat{\mathbf{s}}^*_{t+N|T}, ..., \hat{\mathbf{s}}^*_{t+1|T}\} \tag{5.24}$$

It is interesting to note that the smoothing step in Rauch et al. (1965) also computes state covariance during backward smoothing, however in the proposed algorithm, the corresponding equations are not implemented since they are not necessary.

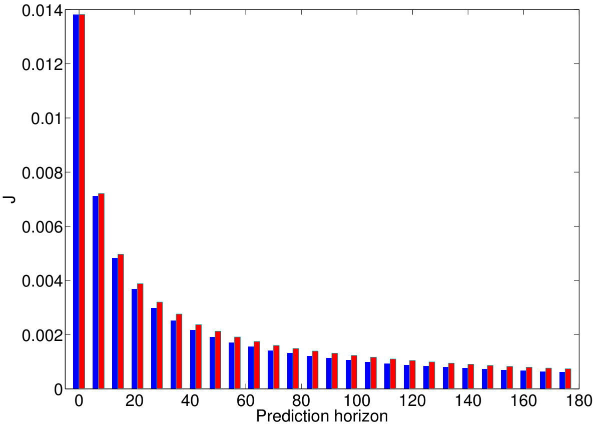
The double step reference filtering control strategy described by Algorithm 7 can be considered to be a discretization of the single-rate one described by Algorithm 5. This is illustrated in Figure 5.3, where is shown the prediction and the smoothing trajectories performed in one algorithm step by using the SR-HF-IBVS (Figure 5.3(a)) and the DR-EKFS-IBVS (Figure 5.3(b)) control strategies. In addition, Figure 5.4 shows a more detailed comparison between the two algorithms. Concretely, Figure 5.4(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 5.4(b), which shows the computational cost of both algorithms, indicates that the computational cost of Algorithm 5 is significantly reduced by an approximate ratio of 1 to N by using Algorithm 7. 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 5.4(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.

The well known local model based on the interaction matrix, Allibert and Courtial (2009); Allibert et al. (2010), is used to perform the filter-smoother step. This model is linearized and “discretized” upon the sampling time[2]. Thus, depending on the period used, obtaining:

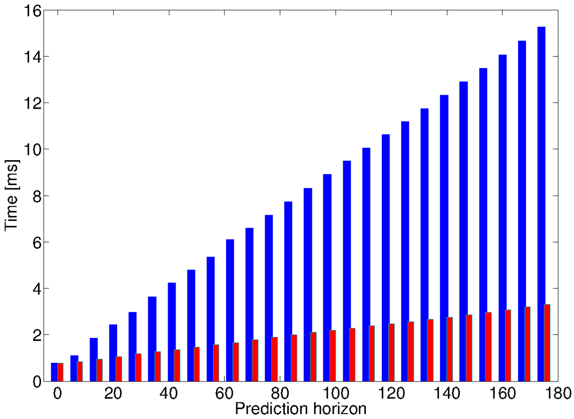
$$\text{(based period) } \hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}_t + \mathbf{J}_\delta(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \dot{\mathbf{q}}_t \tag{5.25}$$

$$\text{(frame-period) } \hat{\mathbf{s}}_{t+N} = \hat{\mathbf{s}}_t + \mathbf{J}_{N\delta}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \dot{\mathbf{q}}_t \tag{5.26}$$

[2] From “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 $\mathbf{Q}_\delta = \int_0^\delta e^{\mathbf{A}_c(\delta-\tau)} \cdot \mathbf{B}_c \cdot \mathbf{Q}_c \cdot \mathbf{B}_c^T \cdot e^{\mathbf{A}_c^T(\delta-\tau)} \mathrm{d}\tau$ and $\mathbf{R}_\delta = \mathbf{C} \int_0^\delta \left\{ \left[\int_\tau^\delta e^{\mathbf{A}_c(s-\tau)} \mathbf{B}_c ds \right] \mathbf{Q}_c \left[\int_\tau^\delta \mathbf{B}_c^T e^{\mathbf{A}_c^T(s-\tau)} ds \right] \right\} \mathrm{d}\tau \mathbf{C}^T + \mathbf{R}_c \delta$.



(a) Cost index w.r.t the prediction horizon.



(b) Computational cost of the algorithms w.r.t the prediction horizon.

Figure 5.4: Algorithm computational cost and cost index between (high frequency single-rate) HF-EKFS (blue bars) and DR-EKFS (red bars).

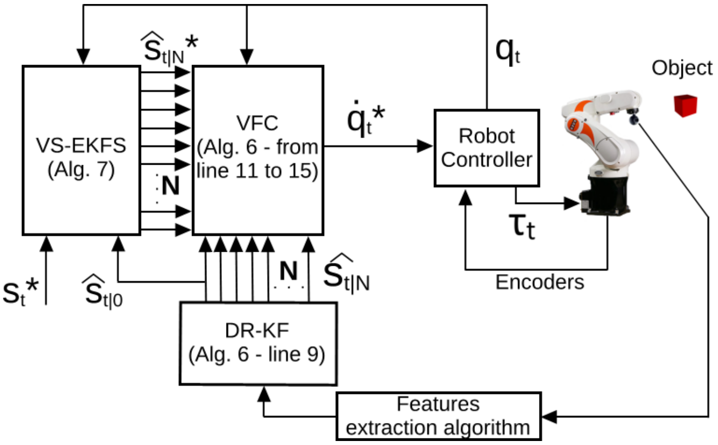


Figure 5.5: Dual-sampling reference filtering control strategy for visual feedback robot control block diagram.

with $\mathbf{J}_\delta(\hat{\mathbf{s}}_t, \mathbf{q}_t) = \delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t)$ and $\mathbf{J}_{N\delta}(\hat{\mathbf{s}}_t, \mathbf{q}_t) = N \cdot \delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t)$.

Similarly, covariances of EKF must be appropriately computed:

$$\mathbf{Q}_\delta = \delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) \tag{5.27}$$

$$\mathbf{Q}_{N\delta} = N\delta \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) \tag{5.28}$$

$$\mathbf{R}_\delta = \frac{\delta^3}{3} \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) + \mathbf{R}_c \delta \tag{5.29}$$

$$\mathbf{R}_{N\delta} = \frac{(N\delta)^3}{3} \cdot \mathbf{J}(\hat{\mathbf{s}}_t, \mathbf{q}_t) \cdot \mathbf{Q}_c \cdot \mathbf{J}^T(\hat{\mathbf{s}}_t, \mathbf{q}_t) + \mathbf{R}_c N\delta \tag{5.30}$$

Algorithm 7 returns the set of reference features filtered by the smoother on a lifted vector.

5.3.3 Visual Feedback Control

Here the same explanation seen earlier in Section 5.2.3.

Finally, Figure 5.5 shows the control block diagram of the proposed dual-rate reference filtering control strategy approach.

5.4 Analysis and Results

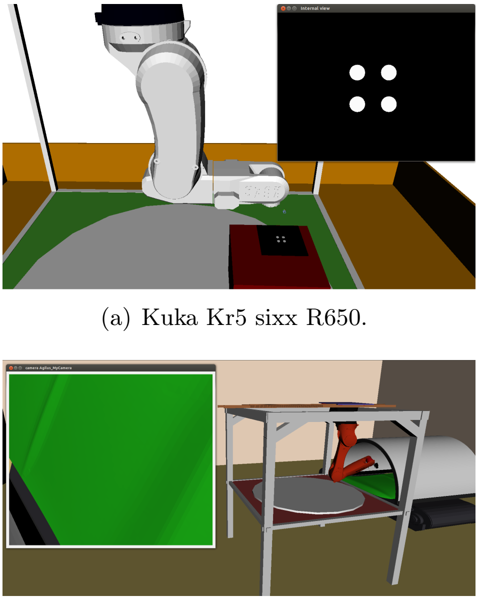
A series of experiments have been made up in order to analyze the properties and improvements of the proposed dual-rate reference filtering control structure with respect to its single-rate counterpart, in both 2D and 3D workspaces.

In the following, let us assume this notation: ${}^{S_1}\mathbf{M}_{S_2}(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$, where ${}^{S_1}\mathbf{M}_{S_2}$ is the homogeneous matrix relating the coordinate system S_2 with respect to the S_1 one; X, Y and Z are Cartesian coordinates and θ_y, θ_y and θ_z are the three angular coordinates.

Results described along this section have been carried out in both simulation and real scenarios. Working under simulation scenario provides us the capability of carrying out tests in extreme conditions. On the other hand, implementing all the approaches presented in this chapter using industrial robot manipulators, vision system, industrial cameras, Ethernet communications, etc. will validate results obtained under simulation.

The simulation environment has been implemented using the C++ programming language, with different open source libraries such as ViSP, Marchand et al. (2005), OpenRAVE, Diankov and Kuffner (2008) or Orocos Toolchain, Bruyninckx et al. (2003); Soetens and Bruyninckx (2005), among others. This environment allows us to show the better performance it is possible to get by using the novel dual-rate reference filtering control strategy in critical situations. The robots (Kr5 and Agilus) have been kinematically and dynamically modeled using OpenRAVE and ViSP libraries. The intrinsic and extrinsic camera parameters used are the ones obtained by calibrating the different cameras used for. In addition Orocos Tool-chain is used to assure the task schedule of the different components working under Linux prompted with a real-time kernel (see Annex A for more details). Figure 5.6 shows a snapshot of the simulation scenarios used in this chapter.

In the following, several tests are performed in order to analyze the properties of the proposed dual-rate visual feedback control structure. Without loss of generality, four coplanar points, forming a square of $300mm$ on a



(a) Kuka Kr5 sixx R650.

(b) Kuka Agilus R900 sixx.

Figure 5.6: Virtual robot platforms.

side, have been used as target.

5.4.1 Simulation Results

With respect to the controller parameters, in this subsection a proportional control law has been used and **Q** and **R** have been tuned up to their optimal values, which are the ones that minimize the reachability time and maximize task solution reachability.

Moreover, and because the control task failure in 2D and 3D spaces are not produced by the same causes, two types of tests, depending on where the control task is carried out, have been performed:

1) If the control is carried out into the 2D space, there are two problems, pure rotations around the optical axis and large translations. The

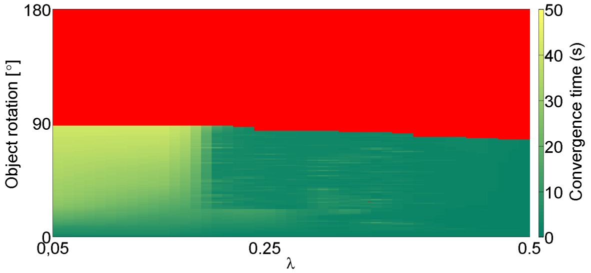


Figure 5.7: Reachability regions and convergence time analysis for different rotation angles using classical image based control. The red region indicates that IBVS was unable to converge.

first analysis achieved in this section is by using the single-rate proposal of Section 5.2. In this very first experiment, it is analyzed the benefits of our proposal with respect to classical 2D visual feedback controller[3].

The case of pure rotations is one of the most representative and challenging cases in visual feedback control tasks and is cause bay the failure when working with real robots due to the camera retreat, which violates the limit joint constraints. The case of large translation as well as cases where features leave the camera field of view is similar to the 3D case, depending on the situation of the object with respect to the camera.

Figure 5.7, where the zone colored in red indicates that the algorithm has failed to solve the task, while degraded green shows its convergence time, shows how classical single rate 2D visual feedback control (SR-IBVS) cannot solve complex situations where the rotation around the optical axis is higher than 90° degrees (red region), despite of the value for the control gain λ. On the contrary, our novel reference filtering control strategy working at low frequency (SR-EFKS) is able to solve rotations up to 170° degrees, as depicted in Figure 5.8. In addition to this, the convergence time is

[3] Results have been obtained with a ASUS laptop with Ubuntu 12.04 as OS and hardware configuration: Intel Core i7-2670QM, 8GB of RAM @ 2.2GHz and a GeForce GTX 560M 2GB

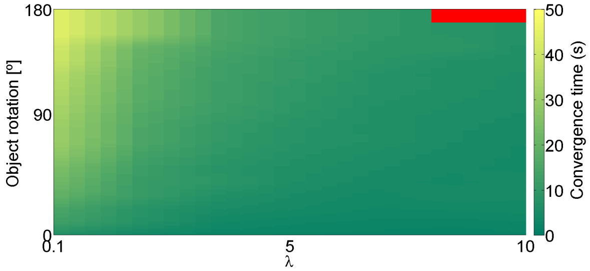


Figure 5.8: Reachability regions and convergence time analysis for different rotation angles using SR-EFKS-IBVS controller: problem reachability and convergence time in function of λ. Red regions indicate control convergence failure.

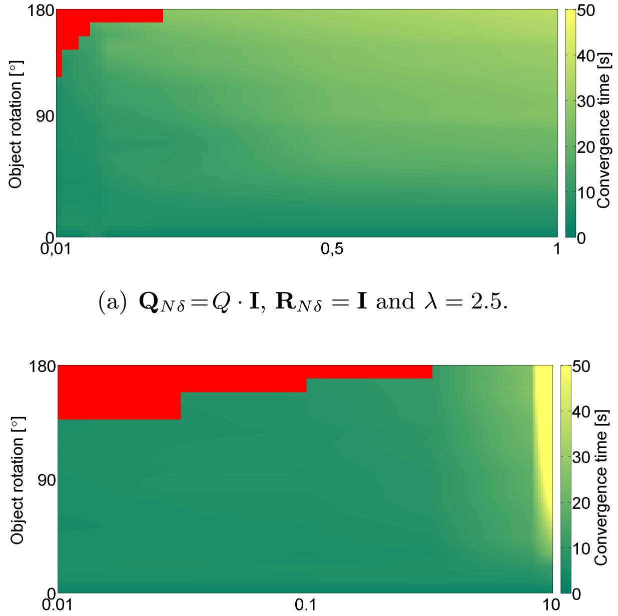
about three times smaller compared to classical IBVS. Previous figures show that values of λ are higher in SR-EKFS-IBVS than classical IBVS without become unstable.

In general, the covariance matrix $\mathbf{Q}_{N\delta}$ will have little influence, while the smoothed trajectory is more sensible to different values on covariance matrix $\mathbf{R}_{N\delta}$. If the covariance matrix $\mathbf{Q}_{N\delta}$ is too large, then the dynamics of the filter/smoother become slower than the closed-loop dynamics and thus convergence time increases exponentially as shown in Figure 5.9(a) for larger values of $\mathbf{Q}_{N\delta}$. If the covariance matrix $\mathbf{R}_{N\delta}$ is too large, then the trajectories modified by EKFS smoother converge too quick to the reference features and thus the "smoothing" effect is cancel out, which implies that reachability is similar to the classic IBVS case.

Horizon prediction parameter N_p has little influence over the reachability, while clearly affects to the convergence time as shown in Figure 5.10.

The results under simulation scenario for the case of large translation are pretty similar to the ones of the case of pure rotation. An example of this case is shown experimentally in Section 5.4.2.

Now, the following test highlights the benefits of using the dual-rate



(a) $\mathbf{Q}_{N\delta} = Q \cdot \mathbf{I}$, $\mathbf{R}_{N\delta} = \mathbf{I}$ and $\lambda = 2.5$.

(b) $\mathbf{Q}_{N\delta} = \mathbf{I}$, $\mathbf{R}_{N\delta} = R \cdot \mathbf{I}$ and $\lambda = 2.5$.

Figure 5.9: Influence of covariance matrices magnitudes over reachability and convergence time. The red regions indicate that SR-EKFS-IBVS was unable to converge.

reference features filtering algorithm 6 with respect to its single-rate counterpart, Algorithm 4. Figure 5.11 shows, for the case of a pure rotation of 150$^{\text{o}}$, how the variation of the controller gain parameter (λ) and the camera frame rate affects to the solution reachability and the convergence time. The test shows the improvement obtained by the proposed dual-rate reference filtering control structure (Figure 5.11(b)) compared with its single-rate counterpart (Figure 5.11(a)). In can be seen that the dual-rate approach is more robust against the delay introduced by the camera's frame rate, but also converges faster than its equivalent single-rate one working at frame

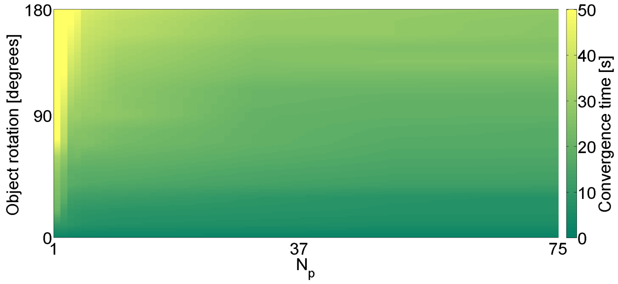


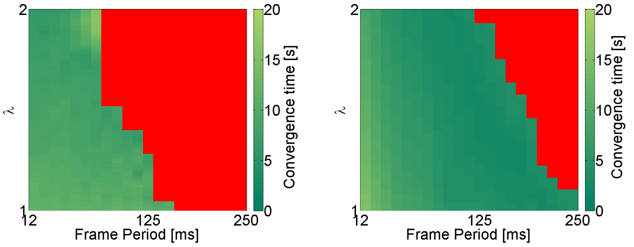
Figure 5.10: Influence of the prediction horizon N_p over reachability and computational time with ($\lambda = 2.5$ $\mathbf{R}_{N\delta} = \mathbf{I}$ and $\mathbf{Q}_{N\delta} = \mathbf{I}$).

rate.

2) If the control is carried out directly in the 3D space, there are not typical cases of failure but the ones in which during the control task features leave the camera field of view, Chaumette (1998). For this reason, initially the object is placed on 9 different places in the Cartesian workspace. Such places have been chosen because, experimentally, they produce task control failures when using classic single-rate approaches. For each one of them, its local frame is rotated, ${}^{w}\mathcal{F}_c$, from -20^o to 20^o in roll and pitch, and 360^o in yaw. All these rotations draw a spherical cap. Because of the difficulty on representing such 3D space of solutions, for better readability and without loss of generality, it has been impose that, if taking one roll-pitch combination any yaw has produced a failure, then that roll-pitch pair is considered as failure.

In Figure 5.12 the representation of the commented test is shown. The camera frame-period is $84ms$. A proportional control law has been used as underlying controller, and $\mathbf{Q}_\delta$ and $\mathbf{R}_\delta$ have been tuned up to their optimal values, which are the ones that minimize the convergence time and maximize task solution reachability.

The figure demonstrates that, the proposed dual-rate reference filtering control strategy provides a better performance than its single-rate counterpart working at low frequency. That is, there are more reachable situations.



(a) Single-rate approach working at low frequency.

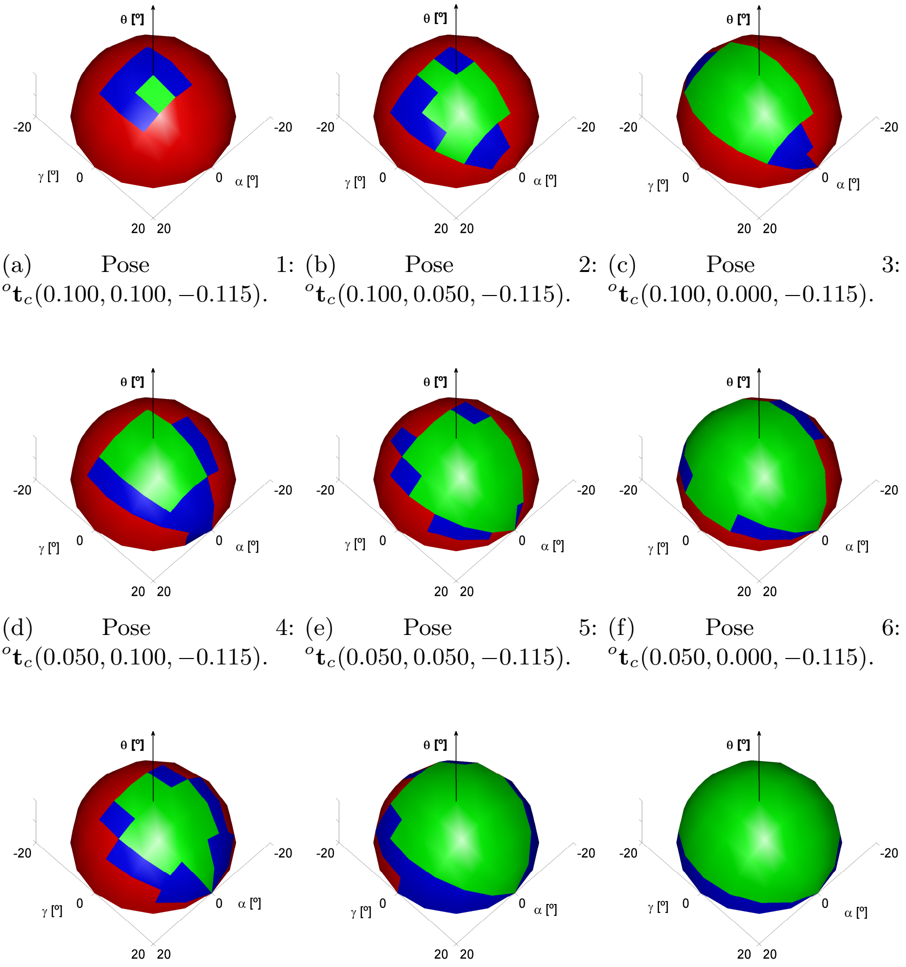
(b) Dual-rate approach

Figure 5.11: 2D visual feedback control with reference filtering strategy: analysis of the reachability and convergence time.

Owing the difficulty to analytically analyze the robustness of multi-rate control structures (even more difficult when nonlinear systems, such as the one presented in this chapter, are within the control loop), this analysis is carried out empirically throughout experimentation.

In this case (and for both 2D/3D control workspaces), the task to be solved is a pure rotation around the optical axis 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, $\mathbf{Q}_\delta$ and $\mathbf{R}_\delta$, as well as the dynamic model matrices, are modified to the needed equivalent ones for each frame period (see Section 5.3.2 for more details). $\mathbf{Q}_\delta$ and $\mathbf{R}_\delta$ have been tuned up to their optimal values, which are the ones that minimize the reachability time and maximize task solution reachability.

One easy form to introduce disturbances to visual feedback control schemes is to modify the value of the depth parameter (Z_t). Usually, this parameter has to be estimated from vision algorithms, meaning that it highly depends on the camera calibration accuracy. Moreover, most of pose estimation algorithms, such as the used in this chapter, are model based. Object model errors also affect the estimation of Z_t. Knowing this, in this experiment Z will vary to show up how such a variation affects the proposed



(a) Pose 1: ${}^{o}\mathbf{t}_c(0.100, 0.100, -0.115)$.

(b) Pose 2: ${}^{o}\mathbf{t}_c(0.100, 0.050, -0.115)$.

(c) Pose 3: ${}^{o}\mathbf{t}_c(0.100, 0.000, -0.115)$.

(d) Pose 4: ${}^{o}\mathbf{t}_c(0.050, 0.100, -0.115)$.

(e) Pose 5: ${}^{o}\mathbf{t}_c(0.050, 0.050, -0.115)$.

(f) Pose 6: ${}^{o}\mathbf{t}_c(0.050, 0.000, -0.115)$.

(g) Pose 7: ${}^{o}\mathbf{t}_c(0.000, 0.100, -0.115)$.

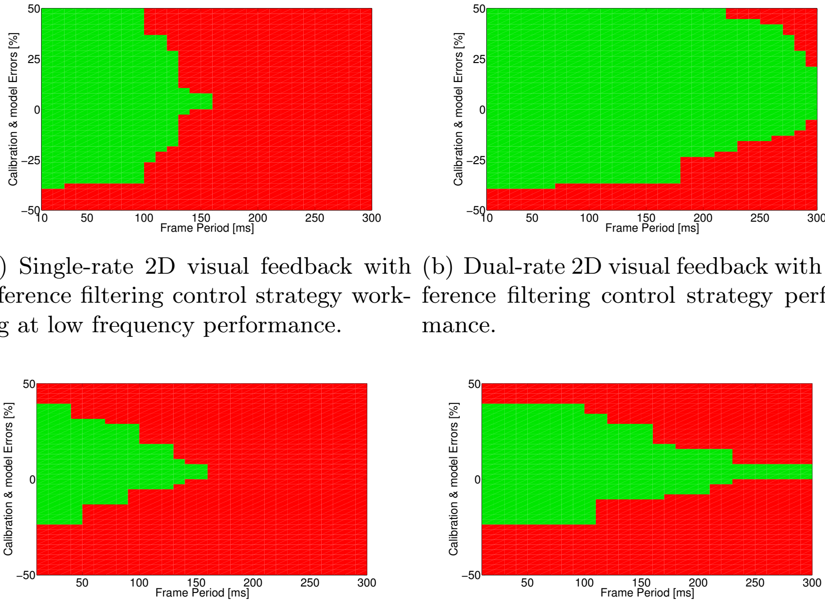
(h) Pose 8: ${}^{o}\mathbf{t}_c(0.000, 0.050, -0.115)$.

(i) Pose 9: ${}^{o}\mathbf{t}_c(0.000, 0.000, -0.115)$.

Figure 5.12: Analysis of solution reachability using 3D visual feedback with reference filtering control strategy. A failure is represented with red color, DR-EKFS-PBVS in blue and SR-EKFS-PBVS in green.

dual-rate reference filtering control structure, comparing its performance with the given by its single-rate counterpart. Its optimal value is $0.115m$.

Results in Figure 5.13 demonstrate that the dual-rate control strategy presented in this chapter 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 of this, this experiment proves that, under the same conditions, the dual-rate control strategy proposed in this chapter is more robust than the single-rate one.



(a) Single-rate 2D visual feedback with reference filtering control strategy working at low frequency performance.

(b) Dual-rate 2D visual feedback with reference filtering control strategy performance.

(c) Single-rate 3D visual feedback with reference filtering control strategy working at low frequency performance.

(d) Dual-rate 3D visual feedback with reference filtering control strategy performance.

Figure 5.13: Robustness comparison of both single-rate and dual-rate reference filtering approaches (25 experimets were performed).

5.4.2 Experimental Results

The proposed approach has been implemented and validated using different hardware. Specifically: two industrial robot manipulators of 6 DOF have been used, Kuka Kr5 sixx R650 and Kuka Agilus R900 sixx; as vision system a webcam connected via USB to a workstation, a smart camera connected via Ethernet IP and a 5MP monochrome camera connected to an industrial PC-based Xpectia vision system from OMRON Electronics; as controllers, a workstation and an IGEP board (see Annex A for further information).

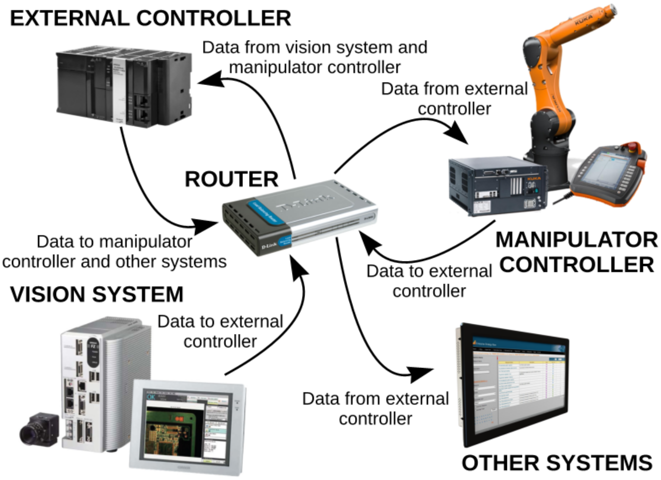

EXTERNAL CONTROLLER
Data from vision system and manipulator controller
Data from external controller
ROUTER
Data to manipulator controller and other systems
MANIPULATOR CONTROLLER
Data to external controller
VISION SYSTEM
Data to external controller
Data from external controller
OTHER SYSTEMS

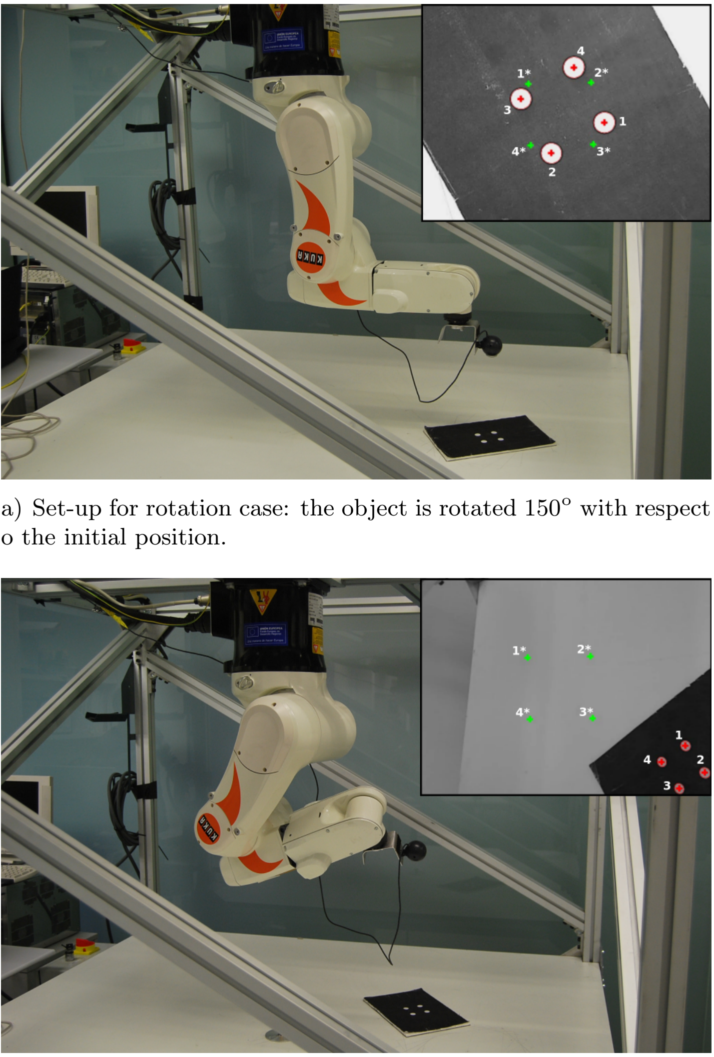
Figure 5.14: Industrial robot visual feedback control scheme.

Besides the hardware used, Figure 5.14 shows their common control block diagram. Concretely, for the experiments shown in this section, vision system works at $\bar{\delta} = 84$ ms, and the external controller at $\delta = 10$ ms.

In the first experiment, the validation of the Algorithm 4 is presented. In this case single-rate reference features filtering control strategy is compared with classic 2D visual feedback controller, both of them working at frame period. In addition, the comparison between our SR-EKFS-IBVS approach and the same but without the use of the Gaussian smoother, SR-EK-IBVS, is also provided. The hardware used in this case is: robot Kuka Kr5 sixx R650, web-cam Logitech C300 mounted at the robot end-effector, and a workstation as controller. All these hardware are communicated via protocol TCP/IP throughout the RSI.EthernetXML technology from KUKA Robots (see Annex A for a more detailed information regarding KUKA-External hardware real-time communication).

The experimentation is based on pure rotation around camera optical axis studied in Section 5.4.1 and, in addition, to large displacements.

Figure 5.15(a) depicts the robot initial configuration for the rotational



(a) Set-up for rotation case: the object is rotated 150° with respect to the initial position.

(b) Set-up for large displacement problem: the object is rotated and translated with respect to the initial position.

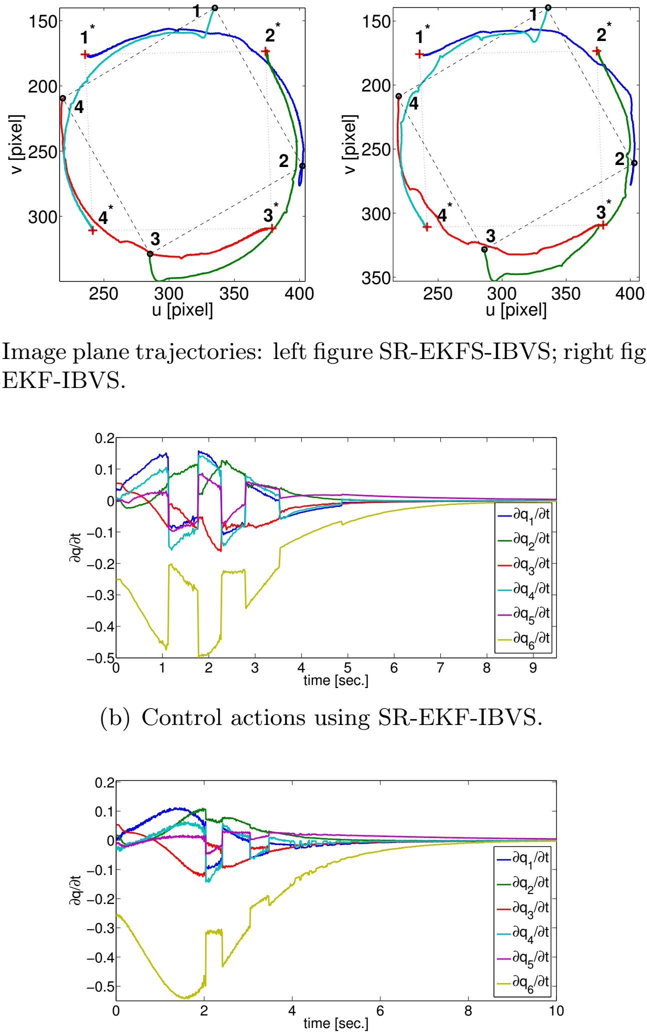
Figure 5.15: Experimental set-up on KUKA KR5 sixx R650 cell platform.

case, where the initial pose is turned an angle of 150° degrees with respect Z axis and the distance of 0.205 m below the camera reference system. On the other hand, a large displacement case is shown in Figure 5.15(b). Let us denote the world frame with the sub-index w, which origin is placed on the base of the robot, with positive-Z is pointing at the ceiling. Thus, the camera, denoted with the sub-index c is at ${}^{w}\mathbf{M}_c(0.280, 0.046, -0.524, 3, 093, 0.215, 0.878)$, and the object, denoted with the sub-index o, is at ${}^{c}\mathbf{M}_o(0.312, 0.202, 0.819, 0.0986, 0.685, 0.957)$. Finally, the robot end-effector and the camera are related by ${}^{e}\mathbf{M}_c(-0.080, 00.050, -2.235, -2.182, 0.056)$. This problem is even harder to solve since it implies large scale variations, rotations and Cartesian shifts.

Parametrization setting for both experiments is $\lambda = 4$, $\mathbf{Q}_{N\delta} = \mathbf{I}$ and $\mathbf{R}_{N\delta} = 10 \cdot \mathbf{I}$. Both cases show a comparison between SR-EKF-IBVS and SR-EKFS-IBVS has made in order to compare two filtering strategies and how affect them to the system behavior.

Figures 5.16(a) and 5.17(a) show image plane trajectories described with the SR-EKF-IBVS and SR-EKFS-IBVS algorithms, using the same covariance matrices and the control gain. In the rotation problem, the trajectories result are quite similar but the control action are more smoothed in the case of SR-EKFS-IBVS than in the SR-EKF-IBVS, providing a better movement of the robot joints (see Figures 5.16(b) and 5.16(c)). In large displacement problem, both of them solve it but the behavior in the image plane is different, see Figure 5.17(a). This is because, in SR-EKFS-IBVS, the smooth part of the filter is considering the Z parameter during more prediction ($T = t + N_p$) than SR-EKF-IBVS (that only predicts the $T = t + 1$), that in this problem is quite different from the desired Z^*, and this aspect produces different trajectories filtered. An important aspect is that, although both algorithms have similar magnitudes of the control actions in the beginning, SR-EKFS-IBVS has more smoothed control actions than SR-EKF-IBVS with the same convergence time, due to the trajectory predicted (see Figures 5.17(b) and 5.17(c)).

Although the SR-EKFS-IBVS behavior can provide better behaviors

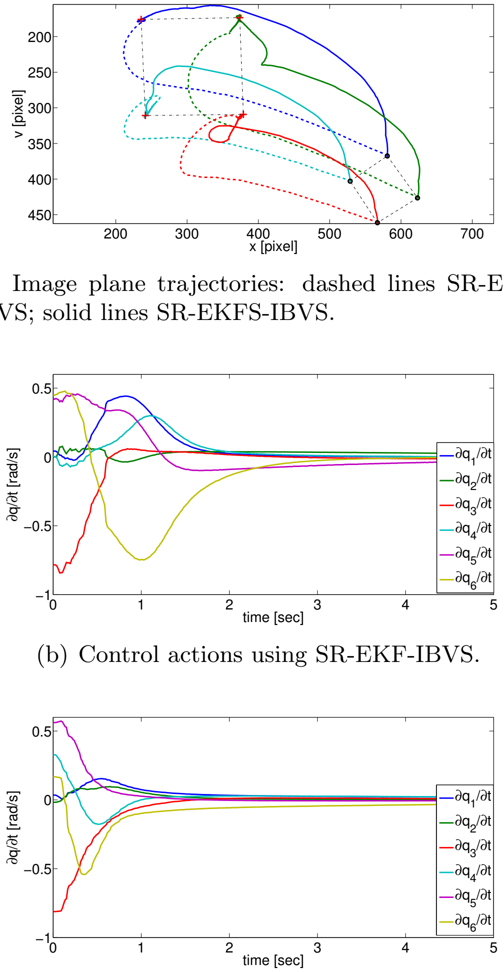


(a) Image plane trajectories: left figure SR-EKFS-IBVS; right figure SR-EKF-IBVS.

(b) Control actions using SR-EKF-IBVS.

(c) Control actions using SR-EKFS-IBVS.

Figure 5.16: Rotation problem (Object rotation 150°): comparative between SR-EKF-IBVS and SR-EKFS-IBVS both with the same parameters ($Q = I$, $R = 10.0 \cdot I$ and $\lambda = 4.0$)



(a) Image plane trajectories: dashed lines SR-EKF-IBVS; solid lines SR-EKFS-IBVS.

(b) Control actions using SR-EKF-IBVS.

(c) Control actions using SR-EKFS-IBVS.

Figure 5.17: Large translation problem: comparative between SR-EKF-IBVS and SR-EKFS-IBVS both with the same parameters ($Q = I$, $R = 10.0 \cdot I$ and $\lambda = 4.0$)

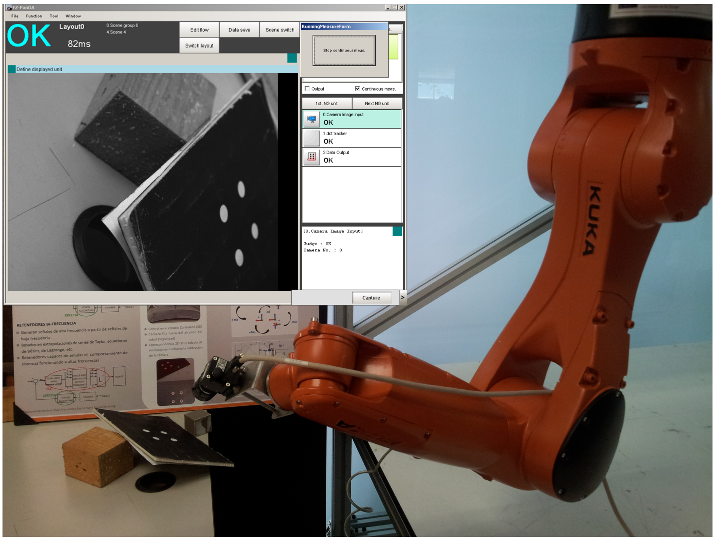
Table 5.2: Pure Rotation Problem (Object Rotation 150°).

	Algorithm Time [ms]	Convergence Time [s]
SR-EKF-IBVS	0.2971	20.295
SR-EKFS-IBVS $N_p = 10$	1.4735	18.943
SR-EKFS-IBVS $N_p = 20$	2.9906	18.317
SR-EKFS-IBVS $N_p = 30$	4.34	17.865

than SR-EKF-IBVS, its convergence time depends on the parameter N_p (see Table 5.2). The SR-EKFS-IBVS filtering time grows linearly with N_p and this implies that the convergence time of the whole algorithm (time between the start pose until the system reach the desired pose) is greater.

The next experimentation is carried out by using the following hardware: Agilus R900 sixx manipulator is equipped with a robot controller that allows external real-time communication using the Ethernet UDP protocol; the Xpectia vision system is a PC based system that allows an easy implementation of industrial visual applications. However, the pre-programmed algorithms are simple (i.e. 2D model-based object detection, bar-codes detection, etc.), so new functionalities like feature detection, pose estimation and camera calibration algorithms has been added. On it, the block "Features extraction algorithm" from Figure 5.5; The workstation controller receives the current robot joints positions from the robot controller and the updated features pose estimation from the vision system. All the control approaches presented in this paper have been implemented in this external controller. The resulting control action, i.e. target robot joints positions, is sent then to the robot controller. On it, the blocks DR-KF, DS-EKFS and VFC from Figure 5.5.

The end effector frame is denoted with the sub-index e, being in this experiment placed at ${}^{w}\mathbf{M}_e(0.506, 0.037, -0.973, 0.861, 2.724)$. The camera frame is in ${}^{w}\mathbf{M}_c(0.489, -0.294, -0.586, 1.506, 2.523, 0.406)$ with respect to the world frame. The object frame will vary depending on the test performed. Finally, the end-effector and the camera are related by ${}^{e}\mathbf{M}_c(-0.080, 00.050, -2.235, -2.182, 0.056)$ (see Figure 5.18).



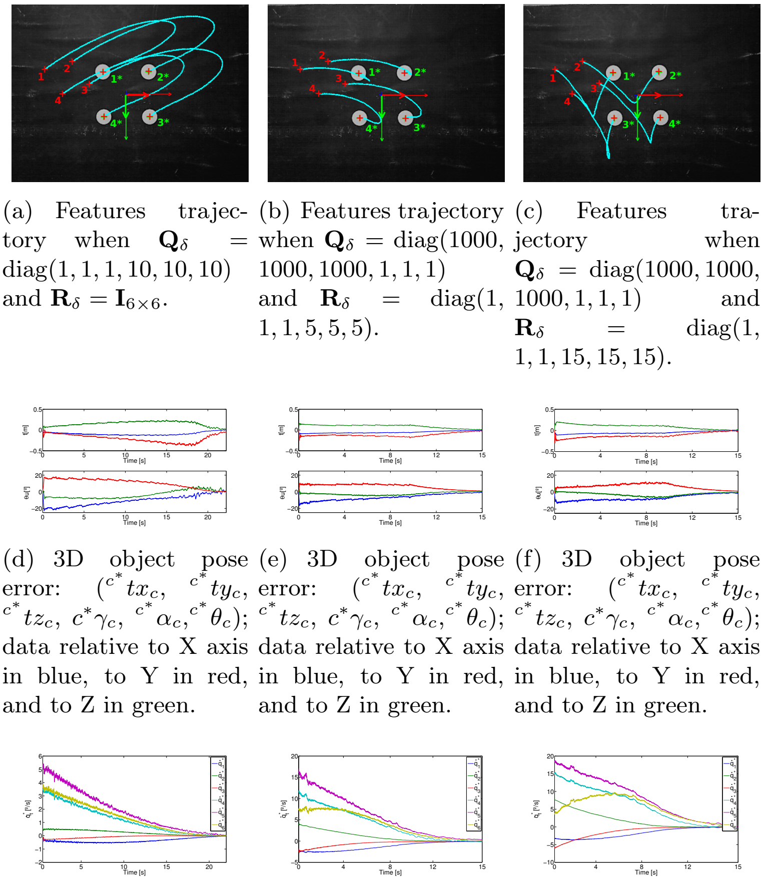
```
[0.Camera Image Input]

Judge : OK
Camera No. : 0
```

Figure 5.18: Experimental set-up with eye-in-hand camera-robot configuration detecting four coplanar dots forming the vertices of a square of 0.1 meter of side.

The influence of the matrices $\mathbf{R}_\delta$ and $\mathbf{Q}_\delta$, when a proportional controller is used, is analyzed, in order to highlight the potential of the proposed dual-rate reference filtering control scheme. In this case, the control is performed directly in the Cartesian workspace and the controller gain (λ) is taken constant and equal to 4.5.

The object is in ${}^{w}\mathbf{M}_o(0.655, 0.023, -0.649, -2.577, 1.608, -0.194)$, being the distance between the camera and the object frames $0.115m$, and the goal to center the object within the image. This task cannot be performed by single-rate controller working at low frequency due to that the vision system latency makes the features leave the camera field of view while the task is being performed.

Figure 5.19 shows the obtained results. It is interesting to remark how the election of $\mathbf{Q}_\delta$ and $\mathbf{R}_\delta$ affects the system behavior, by solving many situ-



(a) Features trajectory when $\mathbf{Q}_\delta = \text{diag}(1,1,1,10,10,10)$ and $\mathbf{R}_\delta = \mathbf{I}_{6\times 6}$.

(b) Features trajectory when $\mathbf{Q}_\delta = \text{diag}(1000, 1000, 1000, 1, 1, 1)$ and $\mathbf{R}_\delta = \text{diag}(1, 1, 1, 5, 5, 5)$.

(c) Features trajectory when $\mathbf{Q}_\delta = \text{diag}(1000, 1000, 1000, 1, 1, 1)$ and $\mathbf{R}_\delta = \text{diag}(1, 1, 1, 15, 15, 15)$.

(d) 3D object pose error: $({}^{c^*}tx_c$, ${}^{c^*}ty_c$, ${}^{c^*}tz_c$, $c^*\gamma_c$, ${}^{c^*}\alpha_c, {}^{c^*}\theta_c)$; data relative to X axis in blue, to Y in red, and to Z in green.

(e) 3D object pose error: $({}^{c^*}tx_c$, ${}^{c^*}ty_c$, ${}^{c^*}tz_c$, $c^*\gamma_c$, ${}^{c^*}\alpha_c, {}^{c^*}\theta_c)$; data relative to X axis in blue, to Y in red, and to Z in green.

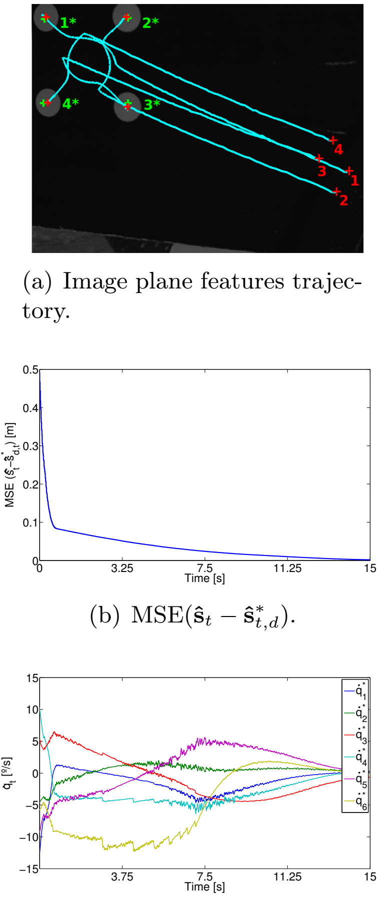
(f) 3D object pose error: $({}^{c^*}tx_c$, ${}^{c^*}ty_c$, ${}^{c^*}tz_c$, $c^*\gamma_c$, ${}^{c^*}\alpha_c, {}^{c^*}\theta_c)$; data relative to X axis in blue, to Y in red, and to Z in green.

(g) Control actions $\partial\mathbf{q}_i/\partial t$.

(h) Control actions $\partial\mathbf{q}_i/\partial t$.

(i) Control actions $\partial\mathbf{q}_i/\partial t$.

Figure 5.19: Analysis of the influence of $\mathbf{Q}_\delta$ and $\mathbf{R}_\delta$ matrices: 3D workspace, proportional control law ($\lambda = 4.5$); $\bar{\delta} = 84ms$, $\delta = 10ms$; ${}^w\mathbf{M}_o(0.655, 0.023, -0.649, -2.577,\ 1.608, -0.194)$, ${}^w\mathbf{M}_c(0.49, -0.294, -0.586, 1.506, 2.523, 0.406)$, ${}^o\mathbf{M}_c^*(0.0, 0.0, -0.115, 0.0,\ 0.0, 0.0)$, and ${}^w\mathbf{M}_e^*(0.60, -0.068, -0.638, 0.132, 0.041, -2.73)$.

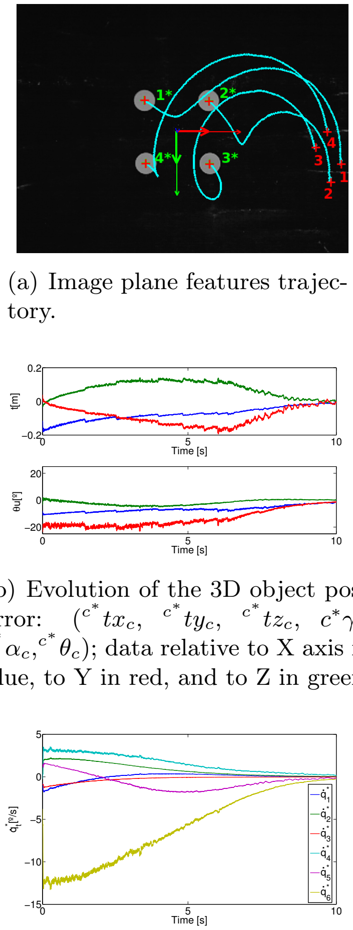


(a) Image plane features trajectory.

(b) MSE($\hat{\mathbf{s}}_t - \hat{\mathbf{s}}^*_{t,d}$).

(c) Control actions $\partial \mathbf{q}_i/\partial t$.

Figure 5.20: 2D workspace task, using the DR- reference filtering control strategy (both single-rate and classical dual-rate approaches fail): ${}^{w}\mathbf{M}_o(0.621, -0.101, -0.628, 1.334,\ 2.392, 0.626)$; ${}^{w}\mathbf{M}_c($ $0.489, -0.294, -0.586, 1.506, 2.523, 0.406)$, ${}^{o}\mathbf{M}_c(0, 0, -0.115,\ 0.0, 0.0, 0.0)$, ${}^{w}\mathbf{M}^*_e(0.57, -0.176, -0.638, 0.158, -0.568, 0.543)$; $\mathbf{Q}_\delta = \mathbf{I}_{12\times12}$ and $\mathbf{R}_\delta = 10 \cdot \mathbf{I}_{6\times6}$; $\lambda = 2.5$.



(a) Image plane features trajectory.

(b) Evolution of the 3D object pose error: $({}^{c^*}tx_c,\ {}^{c^*}ty_c,\ {}^{c^*}tz_c,\ {}^{c^*}\gamma_c,\ {}^{c^*}\alpha_c, {}^{c^*}\theta_c)$; data relative to X axis in blue, to Y in red, and to Z in green.

(c) Control actions $\partial \mathbf{q}_i/\partial t$.

Figure 5.21: 3D workspace task, using the dual-rate reference filtering control strategy (both single-rate and classical dual-rate approaches fail): ${}^{w}\mathbf{M}_o(0.621, -0.1, -0.628, 1.334, 2.392, 0.626)$; ${}^{w}\mathbf{M}_c(0.489, -0.294, -0.586, 1.506, 2.523, 0.406)$, ${}^{o}\mathbf{M}_c(0.0, 0.0, -0.1, 0.0, 0.0, 0.0)$, ${}^{w}\mathbf{M}_e^*(0.57, -0.25, -0.60,\ 0.09, -0.37, 0.51)$, $\mathbf{Q}_\delta = 1100 \cdot \mathbf{I}_{6\times 6}$ and $\mathbf{R}_\delta = \mathrm{diag}(1, 1, 1, 15, 15, 15)$; $\lambda = 4.5$.

ations, where classic single and dual rate control strategies fail. In addition to this, such improvement does not carry a loss of time reaching out the solution, neither a high computational cost, which would make the approach not suitable for real-time applications.

Figures 5.20 and 5.21 show two examples only solved by using the proposed dual-rate reference filtering approach. In Figure 5.20 the control is performed directly into the image plane using a proportional controller working at high frequency. The object is placed at ${}^{w}\mathbf{M}_{o}(0.621, -0.101, -0.628, 1.334, 2.392, 0.626)$, being the goal to position the end-effector at ${}^{w}\mathbf{M}_{e}^{*}(0.570, -0.176, -0.638, 0.158, -0.568, 0.543)$. The failure in classic approaches is because its tendency to perform straight lines in the image plane. This behavior produces that features leave the camera field of view. The failure of the single-rate control using the reference filtering approach is due to the latency in getting new data from the vision system.

Closely to the last result is the one shown in Figure 5.21, where the difference is that the control is performed directly into the 3D workspace, also using a proportional controller working at high frequency. The object position is the same that the case before but the desired end effector position is ${}^{w}\mathbf{M}_{e}^{*}(0.568, -0.248, -0.604, 0.087, -0.366, 0.514)$. As the 2D case, the failure of classic approaches is because the controller performs a straight trajectory in the Cartesian workspace, losing the features before the goal is reached. The failure of the single-rate control using the reference filtering approach is, as in the previous case, due to the latency in getting new data from the vision system.

For last, two cases of study are studied in order to compare the proposed 2D and 3D dual-rate reference filtering control strategies with classic single and dual rate respective controllers, as well as with their single-rate counterparts: in case 1, all the controllers used are able to solve the problem, meanwhile in case 2, only the proposed dual-rate with reference filtering approach has had the capability to provide a solution. The indicators chosen has been the typical rise and settling times, overshoot, the maximum effort defined as $max_t\sqrt{\sum_{i=1}^{m}\left(\frac{\partial q_i}{\partial t}\right)^2}$ and the Integrating Square Control Effort defined as

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

Table 5.3, where the reference filtering control strategy approach has been denoted as EKFS for simplification, shows that the system obtains the best performances when the proposed dual-rate visual feedback control strategy is used. In can be seen how all measured indicators show the better performance of dual-rate control strategies against their single-rate ones, and in particular, how the implemented algorithms DR-EKFS-P and DR-EKFS-PID (in IBVS and PBVS approaches) are clearly superior to other studied cases.

Table 5.3: Comparative of single-rate and dual-rate approaches using several controllers.

APPROACH	CONTROLLER		TIME DOMAIN RESPONSE									
			RISE TIME [s]		SETTLING TIME [s]		OVERSHOOT [%]		MAX EFFORT [°/s]		ISCE [°]	
			CASE 1	CASE 2	CASE 1	CASE 2	CASE 1	CASE 2	CASE 1	CASE 2	CASE 1	CASE 2
IBVS	SR	**P**	6.517	-	11.829	-	2	-	11.791	-	44.625	-
		PID	6.208	-	11.056	-	3	-	11.305	-	41.552	-
		EKFS-P	5.822	-	10.233	-	2	-	10.338	-	39.001	-
		EKFS-PID	5.513	-	10.012	-	3	-	10.155	-	36.976	-
	DR	**P**	5.219	-	8.939	-	2	-	9.102	-	24.763	-
		PID	4.927	-	8.071	-	2	-	8.188	-	15.239	-
		EKFS-P	4.318	5.333	7.634	10.015	0	0	7.522	9.130	13.777	16.229
		EKFS-PID	4.018	5.112	7.183	9.899	0	0	7.161	8.882	13.069	15.401
PBVS	SR	**P**	7.526	-	13.271	-	5	-	14.054	-	52.991	-
		PID	7.056	-	12.988	-	3	-	13.721	-	50.327	-
		EKFS-P	6.418	-	11.638	-	0	-	11.428	-	43.910	-
		EKFS-PID	5.481	-	10.673	-	0	-	10.307	-	38.179	-
	DR	**P**	5.928	-	10.301	-	2	-	10.492	-	39.871	-
		PID	5.376	-	9.834	-	5	-	10.015	-	36.317	-
		EKFS-P	4.911	6.521	8.334	12.412	0	5	8.325	11.4284	16.625	19.280
		EKFS-PID	4.562	5.876	7.929	11.811	2	0	7.816	10.041	14.957	17.154

5.5 Summary

In this chapter, a novel control strategy based on the ideas of Todorov (2008) and Zima et al. (2013) has been presented. Instead of proving dualities between estimation and optimal control, this chapter proposes the use of the optimal estate trajectory resulting of the Rauch-Tung-Striebel (RTS) algorithm, as a new set of reference features to be used by an underlying visual feedback control algorithm. Thus, the chapter shows how using this approach, the solution reachability of classical visual feedback controllers grows exceedingly and thus providing a more suitable trajectories for the measurement in the image.

In addition to this, its generalization from a dual-rate point of view has been presented, increasing thus its inherent properties and overcoming the problem of the sensor latency.

A general formulation has been also given in order to perform our approch in both 2D and 3D workspaces.

A wide analysis has been described along the chapter, highlighting the properties of the proposed approaches. Regardless the workspace in which the control task is performed, the results show that the system behaves much better in terms of solution reachability, convergence time and robustness using the new dual-rate reference filtering control strategy compared to classic approaches.

Finally, a variety of experiments and simulations has been analyzed, demonstrating the benefits of the proposed dual-rate reference filtering control strategy and also using different controllers working at high frequency. In particular, the proposed algorithm has been validated with a complete industrial set-up with significant hardware limitations and performing several positioning error tasks. It is therefore shown those cases where classic approaches tend to fail, while the dual-rate approach still succeeds.

Chapter 6

Conclusions and Perspectives

6.1 Contributions

The research described in this thesis has focused on two main issues related with sensor-based robot visual feedback control. The first issue, presented in Chapters 3 and 4, concerned the sensor latency problem, while the second issue, presented in Chapter 5, concerned the domain of suitable trajectories for convergence assurance.

6.1.1 Sensor Latency Issue

In Chapter 3 we proposed the use of dual-rate high order holds Armesto and Tornero (2003a) as a solution to the sensor latency problem when we have to deal with highly non-linear systems and no particular knowledge of the system is assumed.

In particular, we have validated this approach using two different robotic platforms: a wheeled robot, and a 6 DOF industrial robot manipulator. In both cases, results show that the use of dual-rate high order holds improves the behavior of the overall system due to the fact that the controller is working at high frequency instead the classical low frequency one.

Chapter 3 also extends the work in Armesto et al. (2008), by incorporating time delay compensation, coined as asynchronous dual-rate high order holds with time delay compensation (Asyn. DR-HOH-TDC). A wide set of experiments were shown in both virtual and real scenarios were presented. The proposed Asyn. DR-HOH-TDC was validated introducing them within the control loop of a robot visual feedback control system. Concretely, a 6 DOF industrial robot manipulator performing error positioning and target tracking tasks. The results shown that Asyn. DR-HOH-TDC provided an important improvement in terms of convergence time (around 1 to 2 seconds) with respect to Asyn. DR-HOHs approach. From such experiments some additional conclusions could be extracted:

- Asyn. DR-HOH-TDC provides better input signal estimations rather than its homologous Asyn. dual-rate high order holds (Asyn. DR-HOH), in terms of MSE and typical deviation.
- Asyn. DR-HOH-TDC is more robust against input signals mixed with white noises.

Dual-rate holds presented in Chapter 4 learn the system and controller dynamics in order to provide better estimations, compared to all the approaches presented in Chapter 3. The holds in Chapter 4 use machine learning techniques to generate dual-rate nonlinear hold functions for the particular controlled system. In particular, we use artificial neural networks and synthetic datasets in order to learn a nonlinear hold function in closed loop. By doing this, the new nonlinear hold has information not only of the plant to be controlled, but also the dynamics incorporated by the controller and sensors. Nonlinear hold functions have been also validated in two different platforms: a wheel robot, and a 6 DOF industrial robot manipulator. The results showed an improvement not only in terms of convergence time, but also in terms of robustness.

6.1.2 On Providing Suitable Trajectories Issue

Chapter 5 presented a novel control strategy based on the ideas of Todorov (2008) and Zima et al. (2013). The key idea is to use an optimal trajectory, derived from an optimal non-linear controller based on Rauch-Tung-Striebel (RTS) algorithm, as a new set of reference features to be used by an underlying visual feedback control algorithm. The chapter has shown that by using this approach, the solution reachability of classical visual feedback controllers grew exceedingly and thus, more suitable trajectories for the measurement in the image were provided. Therefore, this new approach was coined "*reference features filtering control strategy*". This approach has been extended to dual-rate systems, by including a dual-rate Kalman filter and a dual-sampling estimation strategy.

A wide analysis in both, 2D and 3D workspaces and using several underlying controllers, was described highlighting the properties of this approach. Regardless the workspace and the underlying controller used, the results shown a better behavior in terms of solution reachability, convergence time and robustness using the new reference features filtering control strategy compared to classic approaches. Comparatives between single-rate and dual-rate approaches have been also provided. In particular, the proposed dual-rate algorithm has been validated with a complete industrial set-up with significant hardware limitations and performing several positioning error tasks, showing cases where classic approaches tend to fail, while the dual-rate proposed approach still succeeded.

6.2 Perspectives

The work performed in this thesis and the contributions obtained open several research issues to be performed in the future concerning visual feedback control schemes.

The first one is the industrial transference of the learnt knowledge to the industry. In this sense, several known brands have been interested on the approaches presented in this thesis (i.e. KUKA Robotics, in the use of their real-time technology to make their applications faster and more robust, and OMRON Electronics, in the use of their vision system within industrial robot manipulators from different brands).

In addition to this, automotive industry sector is very interested in using visual feedback controlled systems for improving the performance of different tasks such as the windscreen sealing. This application is nowadays carried out stopping the body car and using pre-established programmed robot trajectories. This approach incorporates calibration errors and delays. Using visual feedback and some of the approaches presented in this thesis, applications like the one described would be solved more efficiently, reducing the calibration problems and solving the problem while the body car still moving.

Although including dual-rate high order holds in robot visual feedback control applications improved the overall system performance, the noise issue is yet a scourge of this approach. In the case of the novels dual-rate nonlinear high order holds, this is been already solved by introducing a Kalman filter within process of learning.

Moreover, other important aspect is the joint and camera field of view constraints. These issues are already being studying by the use of the novel dual-rate constrained LQR approach, which has demonstrated interesting preliminary results under simulation. However, the time required by the algorithm for finding a solution still being very long in order to be implemented in real-time applications, such as the ones presented in this work.

In the same way, the reference features filtering approach could include joint and camera field of view constraints, assuring thus that the proposed trajectories are reliable.

Appendix A

Experimental Platforms

The work presented along this thesis has been strongly marked by results obtained using not only laboratory hardware but industrial one. In that sense, validation of the different approaches, as well as demonstrations shown up to experts coming up directly from industry, are more plausible and relevant rather than using academic hardware. However, the setting-up process is very time consuming, requiring technical help from experts that takes long time to come or never do. The positive thing is that, in some sense, you become an expert by yourself, finally knowing better than anyone about the

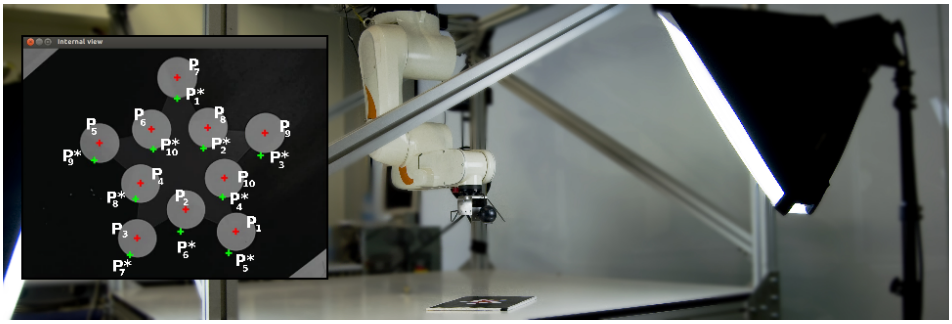


Figure A.1: Industrial robot manipulator cell.

hardware, its properties and limitations.

In addition to this, and as commented in chapters above, we implemented one virtual environment for each platform used. Experience acquired during these years of thesis demonstrated us that, it was a loss of time to develop one virtual environment using software like Matlab/Simulink if later such algorithms had to be translated to other programming languages like C++ used by the real platforms. Hence, we decided to implement a complete software architecture in which we are able to perform virtual or real experiments by only switching by console its running mode. Figure A.2 shows the general diagrammed flow of the mentioned software architecture, which is based on Orocos Toolchain, Soetens (2006). First, several initialization files have to be set up:

- *"Global.xml":* all aspects concerning with the running mode are contemplated into this file. Hence, components to be launched out (i.e. server, control, vision, simulator and reporter), their work frequencies, the robot to be used (for kinematic and dynamic aspects), the launching mode (in simulation or real scenarios), etc. have to be set up before the program is started.

- *"Control.xml":* general aspects of the controller has to be specified in this file: control parameters, estimator to be used, control approach, and on.

- *"Vision.xml":* this file includes all possible options to be set regarding to the camera and the vision algorithm.

- *"Reports.xml":* in this file it is possible to store data from a set of possible options such as the vision and controller outputs, estimations, acquired data, etc.

There are two possibilities depending on the mode selected: virtual environment of real platform.

- *"Virtual environment":* in this mode, nor the server component neither the camera driver are needed. This component is in charge to

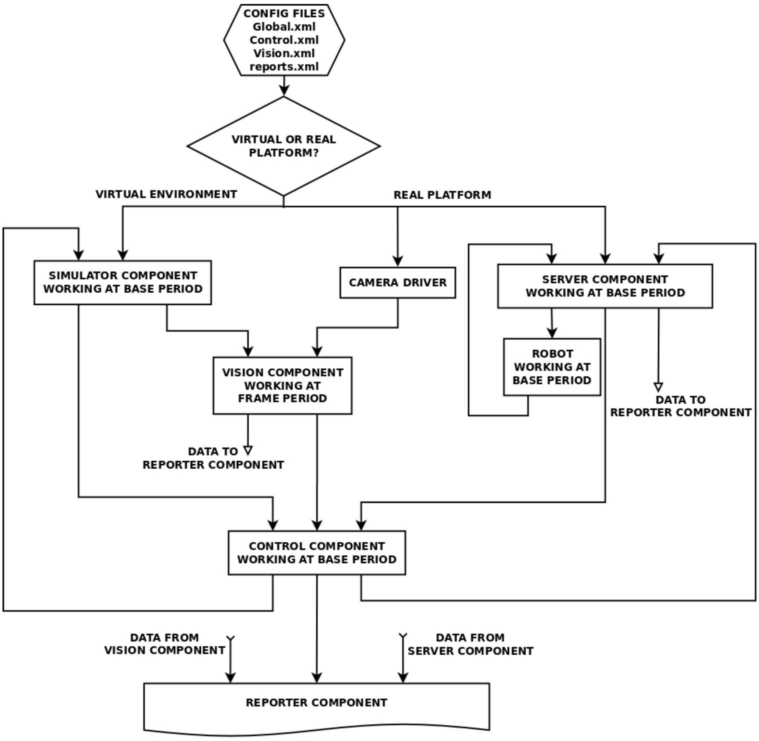
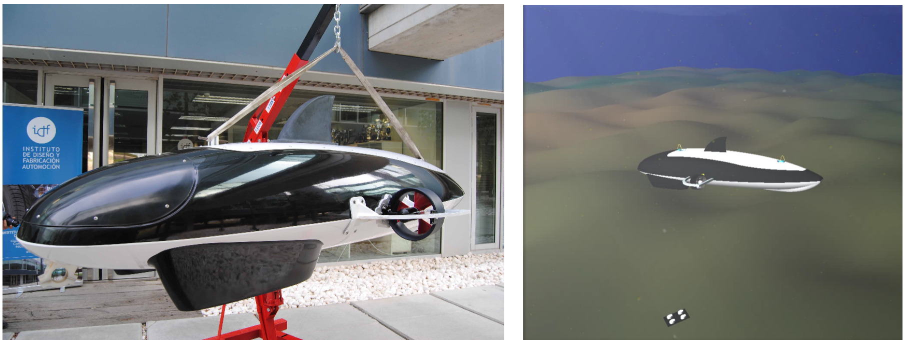

CONFIG FILES
Global.xml
Control.xml
Vision.xml
reports.xml
VIRTUAL OR REAL PLATFORM?
VIRTUAL ENVIRONMENT
REAL PLATFORM
SIMULATOR COMPONENT WORKING AT BASE PERIOD
CAMERA DRIVER
SERVER COMPONENT WORKING AT BASE PERIOD
ROBOT WORKING AT BASE PERIOD
VISION COMPONENT WORKING AT FRAME PERIOD
DATA TO REPORTER COMPONENT
DATA TO REPORTER COMPONENT
CONTROL COMPONENT WORKING AT BASE PERIOD
DATA FROM VISION COMPONENT
DATA FROM SERVER COMPONENT
REPORTER COMPONENT

Figure A.2: Algorithm structure developed to switch out in between virtual and real platforms.

render the entire virtual environment, as well as to provide sensors feedback. It receives data from the control component. Depending on the platform simulated, different open source libraries have been used, such as OpenRave (which utilizes Coin library) Diankov and Kuffner (2008), Ogre Ogre3D (2007) or Gazebo Koenig and Howard (2004); Aguero et al. (2015). In Section A.2 Industrial Robotic Cells and SUMMIT mobile platform are described due to their relevance along this work. Yet, during the years of thesis other platforms and simulators were worked with. One example is ORCA300 shown in



(a) Real platform.

(b) Virtual environment.

Figure A.3: ORCA300 platform.

Figure A.3.

- *"Real platform":* when this option is set, the server is in charge to communicate with the respective platform, getting data from it and exchanging data with control and vision components.

Vision and control components are totally independent of the running mode selected. They can work at the same rate or at different one. Control component is set out as the highest priority task (in fact is selected a little bit lower in order to do not loose other system functions such as the Ethernet communications), while vision component task priority is set low.

Lastly, reports are necessary to data analysis. To do this, other component, coined as reporter, is used. Its priority is the lower one.

In the following, real platforms and virtual environments, robots and other hardware discussed during chapters above, are described. All the hardware mentioned is located at the Institute of Design and Manufacturing (8G Ciutat Politècnica de la Inuvestigaciò building) of the Universitat Politèctica de València (UPV).

A.1 Real Platforms

A.1.1 Industrial Robotic Cells

Robotic cells are fully used in industry. They consist on a robotic environment surrounded by a fully sensory fence blocking up humans pass and saving them from the robot while it is on running mode.

Such cells are provided by a set of sensors which only purpose is the human safeness. Such sensors block immediately the automatic mode of the robotic system if something goes wrong. In addition, the robotic system cannot run until everything indicates that the safeness is totally guaranteed.

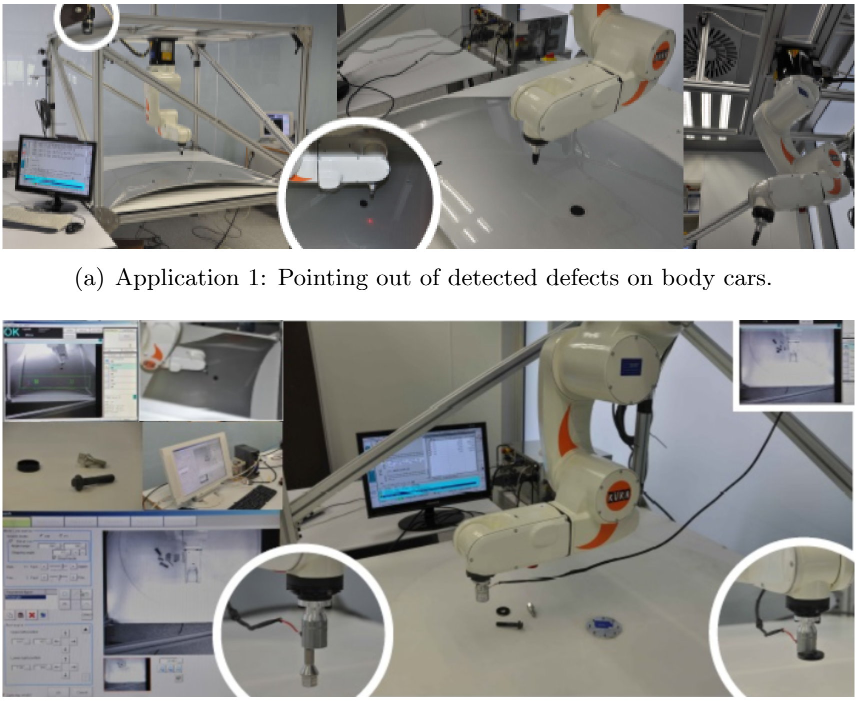
With all of this in mind, in 2010 (starting point of this work) it was designed a cell for small robots (see Figure A.4). It was built up using aluminum bars well bracing for holding still a robot ceiling mounted inside it. The structure is lightly bigger than the robot workspace in order to safe humans from it. It is also equipped by a middle platform with an automatic disc and a conveyor to perform object inspection and tracking tasks.

Two robot manipulators were set in this structure: from 2010 to 2014, robot Kuka Kr5 sixx R650, and from 2014 on, robot Kuka Agilus R900 sixx. In sections A.1.1.1 and A.1.1.2 a more detailed description of both robots is done, as well as how to externally communicate with them to perform real-time applications.

A.1.1.1 Robot KR5 sixx R650

Kuka Kr5 sixx R650 is a compact, light-weight, fast and reliable robot for its size (see Figure A.5(a)). This robot was thought by KUKA Robotics to be a link between laboratory and robust industrial robots. At the end, and due to its capabilities, more close to the industrial demands rather than the laboratory needs, it was utilized in some many applications around the world. In Table A.1, the main technical specifications of Kr5 robot manipulator are shown.

In our case, this robot is ceiling mounted and equipped with a camera



(a) Application 1: Pointing out of detected defects on body cars.

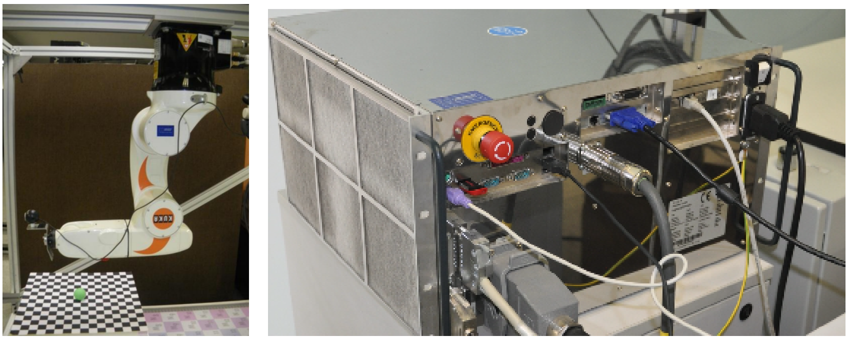
(b) Application 2: Grasping with magnetic tool.

Figure A.4: Robot Kr5 sixx R650 cell.

attached at its end-effector for inspecting or tracking applications.

The compact KUKA KR C2 sr (see Figure A.5(b)) is a made for small robots who need a high performance controller. The high-quality stainless steel housing with a 19" format, and connections routed to the front, offer maximum flexibility and ease of use. This extremely reliable controller makes use of service-proven drive technology and standard PC components.

It features three free PCI slots to allow use of all current and future KUKA PC plug-in cards for small robots. It includes a wide range of connection options, such as a 100-Mbit Ethernet controller and DeviceNet master, to ensure reliable system integration.



(a) Robot Manipulator.

(b) Robot Controller.

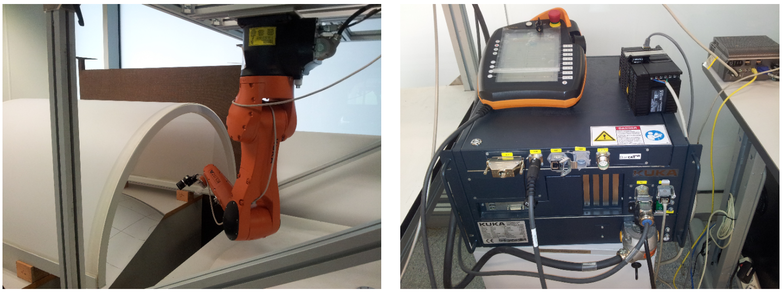
Figure A.5: Robot Kr5 sixx R650 and KRC2 controller.

The KCP, KUKA Control Panel, is used to communicate with the KR C2 sr controller. The KCP teach pendant features a full graphics color display and a Windows interface for easy programming.

KUKA KR C2 sr is also equipped by Kuka real-time technology, which is fully explained along this chapter.

Model	KR5 sixx R650
Payload	5 kg
Max. reachability	650 mm
Max. speed	8.2 m/s
Controller	KR C2sr
Number of axes	6
Repeatability	<±0.02 mm
Weight	28 kg
Mounting positions	Floor, ceiling

Table A.1: Kuka Kr5 sixx R650 specifications.



(a) Robot Manipulator. (b) Robot Controller.

Figure A.6: Robot Kuka Agilus R900 sixx and KRC4 controller.

Model	Agilus R900 sixx
Payload	6 kg
Max. reachability	901 mm
Max. speed	13 m/s
Controller	KR C4sr
Number of axes	6
Repeatability	<±0.03 mm
Weight	52 kg
Mounting positions	Floor, ceiling, wall

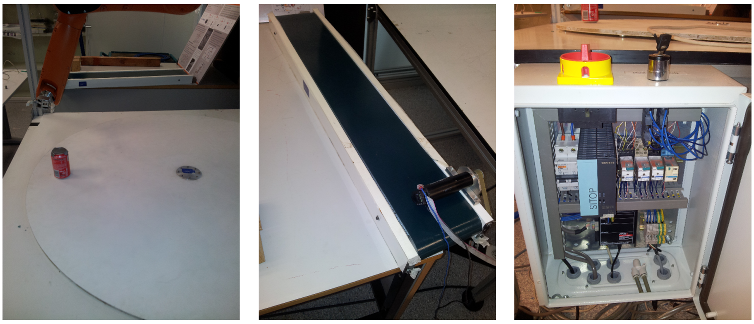
Table A.2: Kuka Agilus R900 sixx specifications.

A.1.1.2 Robot Agilus R900 sixx

Contrary to the Kr5, Kuka Agilus is at all effects one robot thought to work in industrial environments (see Figure A.6(a)). Technologically talking, is more robust, can reach very high speeds and has a short cycle. It is well suited to perform all kind of tasks, from pick & place to welling, and equipped with the last Kuka technologies. It can perform tasks around the 50% faster than the Kr5.

Table A.2 shows the main technical specification of this industrial robot.

The KR C4 compact shown in Figure A.6(b), offers the high perfor-



(a) Disc. (b) Conveyor. (c) Control distribution panel.

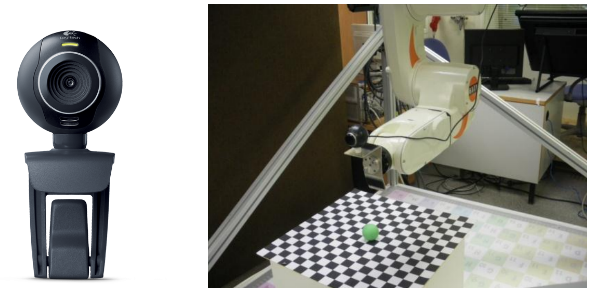
Figure A.7: Automatic externally controlled disc and conveyor.

mance and reliability of the KR C4 technology in a compact design. Its flexible configuration and expansion capability make it a real all-rounder. The number of hardware components, cables and connectors has been significantly reduced and replaced by software-based solutions. The robust, high-quality controller is designed for low maintenance; the temperature-controlled technology only switches on briefly when needed, and is barely audible.

The KUKA smartPAD is used to communicate with the KR C4 compact controller. Manage the robot with it is simply easier rather than it was using the previous PAD. Intelligent, interactive dialogues provide the user with those operator control elements that are currently required. This makes work easier, faster and more efficient.

A.1.1.3 Supplementary Tools

This robotic cell is well suited with different tools and automatic systems to perform all kind of tasks ranging from inspection, pick & place or grasping, among others.



(a) Vision Sensor (b) Eye-in-hand configuration.

Figure A.8: Logitech Web-cam C300.

Linear and Circular Conveyors

Tracking and grasping applications are very usual in industry. There is a huge demand nowadays in incorporating vision sensors and robots to solve problems dealing with moving targets.

In order to perform our approaches and then show the results achieved, a linear belt (Figure A.7(b)) and a circular (Figure A.7(a)) conveyors were designed, built and automatized to be used in this cell.

Vicent Franch, technical in chief at the Institute of Design and manufacturing, was in charge of the mechanical aspects of those systems, while J. Ernesto Solanes, PhD. student, was in charge of the electrical and control aspects.

The controller used, as well as the DC motors, is from MAXON Motors. This controller also allows controlling one system at a time, so the electrical cabinet (Figure A.7(c)) has a switch that allows commuting from one system to the other, electrically.

In addition, it was implemented a simple software in C++ which communicates via USB, RS232 and I2C, allowing managing the different running modes (position, velocity, ramp, direction, etc). This program works under Windows and also under LINUX.

Total Pixels	1300000 pixels
Max. digital video resolution	1280 x 1024
Video Capture	1280 x 1024 @ 30 fps
Features	Right-Sound technology, USB 2.0 compatibility, snap-shot button
Video input device	Color

Table A.3: Logitech web-cam C300 specifications.

CPU	Core i5 2.4 GHz
O.S	Windows XP Embedded
.NET Framework	3.5 or higher
Dual-task	Yes
Max. number of cameras	2
Connectors	USB 2.0 compatibility, Camera Link, Serial Port, Ethernet
Camera	2
Image sensor	5 MP monochrome camera FZ-S5M2 (2248 x 2044) at 12 fps

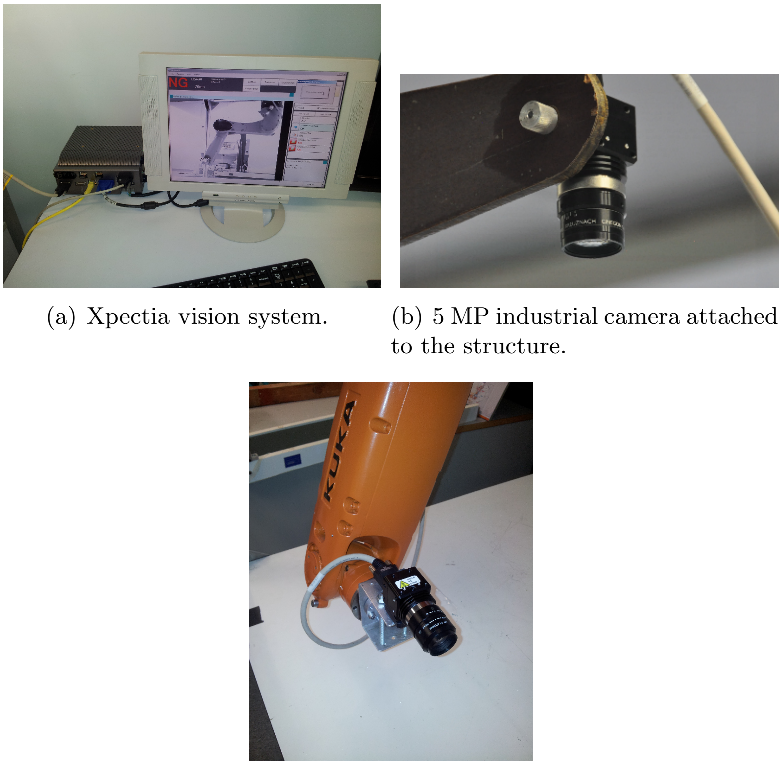
Table A.4: Industrial PC-based vision system Xpectia specifications.

Vision System

Along this thesis, different kind of vision sensors (i.e. web-cam, camera-link industrial cameras, EthernetIP industrial cameras, smart cameras, stereo vision system, embedded vision sensors (i.e. CMUcam)), have been used in order to learn about different techniques and visual approaches. This section focus its attention on two of them, which are the ones used in the experimentation described in above chapters: Logitech web-cam C300, and industrial PC-based vision system compounded by a 5 MP monochrome camera FZ-S5M2 and the Xpectia FJ from OMRON Electronics.

Logitech C300 web-cam (Figure A.8) was selected because its compatibility with Linux. It allows recording images at 30 fps with a resolution of 640x480, with a good image refresh. The communication with the controller is via USB. Table A.3 shows the main characteristics of this sensor.

Xpectia FJ vision system (see Figure A.9(a)) combines the benefits of a compact system with the power and flexibility of an industrial PC platform. The system enables to develop tailored vision solutions quickly. The pro-



(a) Xpectia vision system.

(b) 5 MP industrial camera attached to the structure.

(c) 5 MP industrial camera attached to the robot end-effector.

Figure A.9: Industrial PC-based vision system Xpectia and 5 MP camera configurations.

grams are developed in the "Application Producer", programming in C++, and its HMI capability using Visual Basic. This system provides only drivers for OMRON cameras, being very complicated to use other cameras apart from these.

As vision sensor, the 5 MP monochrome camera FZ-S5M2 is used. The resolution of this camera is 2248x2044, running at 12 fps. The connection with Xpectia is via camera-link.

The main characteristics of this vision system are shown in Table A.4.

Manipulation Tools

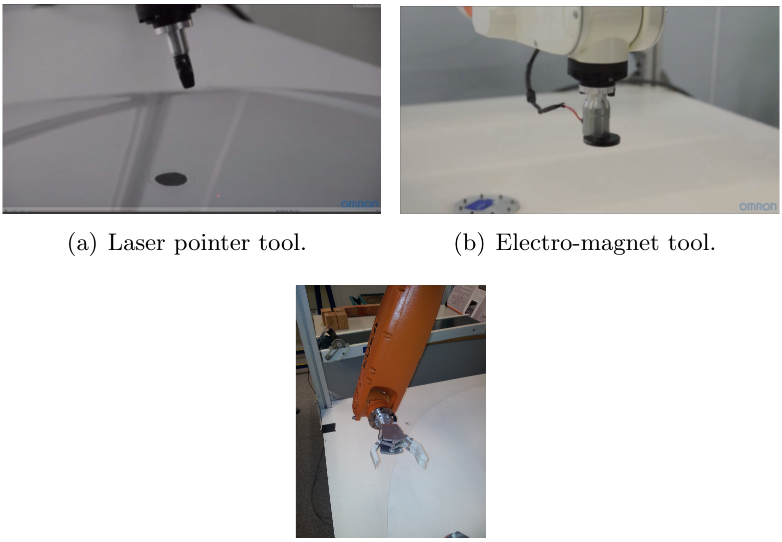
In the results presented along the above chapters, only error positioning and tracking tasks without any kind of manipulation have been described. However, during the years of this thesis, some manipulation approaches have been developed using mainly three tools: a laser pointer (Figure A.10(a)), an electro-magnet tool (Figure A.10(b)), and a grip (Figure A.10(c)).

In Section A.1.1.6, a set of applications using some of those manipulation tools is shown.

A.1.1.4 On Externally Control of Kuka Robots

Industrial robot manipulators are usually controlled with classical DIO cards. However, that is because the automatic program is developed offline and repeated once and again, the most of the cases in open loop. It is possible to control Kuka robots from external devices in two ways: using "KUKA.OPC" technology, or Kuka "*KUKA.Ethernet RSI XML*" technology.

"KUKA.OPC" technology is commonly used within the industry. The problem is that real-time cannot be assured since the KCP layer has higher priority rather than this technology. On the contrary, "*KUKA.Ethernet RSI XML*" technology assures real-time since no-task has a higher priority than this one. The reason because the former one is more used, even though real-time requirements are not assured, is because the majority of industrial applications do not need to be control at real-time.



(a) Laser pointer tool.

(b) Electro-magnet tool.

(c) Grip tool.

Figure A.10: Manipulation tools.

However, in order to validate our approaches, real-time has to be guaranteed, so "*KUKA.Ethernet RSI XML*" technology is used to perform all the validations presented along this work.

Real-time Control of Kuka Robots

The *KUKA.Ethernet RSI XML* is an add-on technology package with the following functions:

- Cyclical data transmission from the robot controller to an external system in the interpolation cycle of $3 - 12$ms (e.g. position data, axis angle, operating mode, etc.)
- Cyclical data transmission from an external system to the robot controller in the interpolation cycle of $3 - 12$ms (e.g. sensor data)
- Influencing the robot in the interpolation cycle of $3 - 12$ ms
- Direct intervention in the path planning of the robot

The characteristics of the package are the following:

- Reloadable RSI-object for communication with an external system, in conformity with *KUKA.RobotSensorInterface* (RSI)
- Communications module with access to standard Ethernet.
- Freely definable inputs and outputs of the communication object.
- Data exchange time-out monitoring.
- Expandable data frame that is sent to the external system. The data frame consists of a fixed section that is always sent and a freely definable section.

The *KUKA.Ethernet RSI XML* enables the robot controller to communicate with the external system via a real-time-capable point-to-point network link. This technology has suffered several changes between the version 2.1 used by the robot Kr5 and the version 3.1 used by robot Agilus. On e of the changes is that, meanwhile in version 2.8 the exchanged data could be transmitted via the Ethernet TCP/IP or UDP/IP protocol as XML strings, in the new version 3.1 this XML strings transmission is only allowed via UDP/IP protocol. In the case of the Kr5, the Ethernet TCP/IP was used, the UDP/IP protocol is the one already working on Agilus.

Programming of the *KUKA.Ethernet RSI XML* package is based on creating and linking RSI-objects. RSI-objects are small pieces of pre-programmed code that can be executed and has additional functionalities than the normal KRL-code. To be able to communicate externally through Ethernet, a specific standard object (ST_ETHERNET or ST_COROB) needs to be created. The code line for creating for example the ST_ETHERNET is typically: ***err=ST_ETHERNET(A,B,config_file.xml)***, where err is a type of string used by the RSI XML (called RSIERR) containing the error code produced when creating the object (normally #RSIOK when it works), A is an integer value that contains the specific RSI-object ID so that it is possible to locate and refer to, B is an integer value for the container to which you want the RSI-object to belong in order to create a group of different objects containing to the same container, config_file.xml is a configuration file located in the INIT folder (path *C:/KRC/ROBOTER/INIT*) on the robot controller that specifies what should be sent and received by the robot controller. The content of this file will be explained further down.

ST_ETHERNET and ST_COROB are objects that can be influenced by external signals, and also send data back to the external system in form of XML files, containing different tags with data. The data can be for example information about the robot's actual axis positions, Cartesian actual positions etc. This data shall send to the server and back within each interpolation cycle of 12ms. ST_ETHERNET has the same functionality as ST_COROB but with additional functionalities; one of these object always

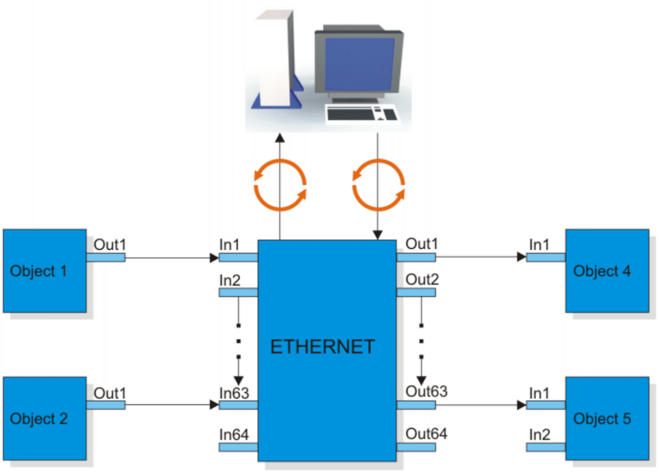


Figure A.11: Functional principle of data exchange.

need to be created and correctly linked in the KRL code in order to establish communication with the external system. For this thesis, the communication object ST_ETHERNET was used. When one of these objects is created and linked correctly, the robot controller connects to the external system as a client.

There are different types of RSI-objects and depending on what you want to do, you have to create and link the correct objects to each other. Besides the standard Ethernet card, an additional card (3COM) was needed, to be able to handle the speed of the transferred data.

The robot controller initiates the cyclical data exchange with a KRC data packet and transfers further KRC data packets to the external system in the interpolation cycle of 12ms. This communication cycle is called an IPO-cycle (Input Process Output), and can be seen in Figure A.11 above. The external system must respond to the KRC data packets with a data packet of its own.

To be able to influence the robot, one needs to initiate an RSI-object for the movements. There are mainly two objects used for this: First, an

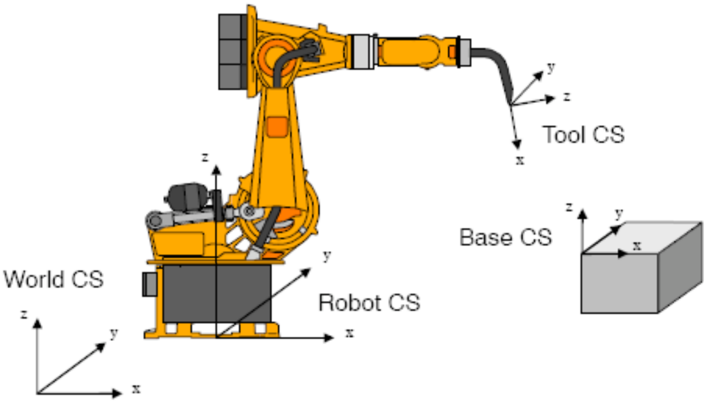
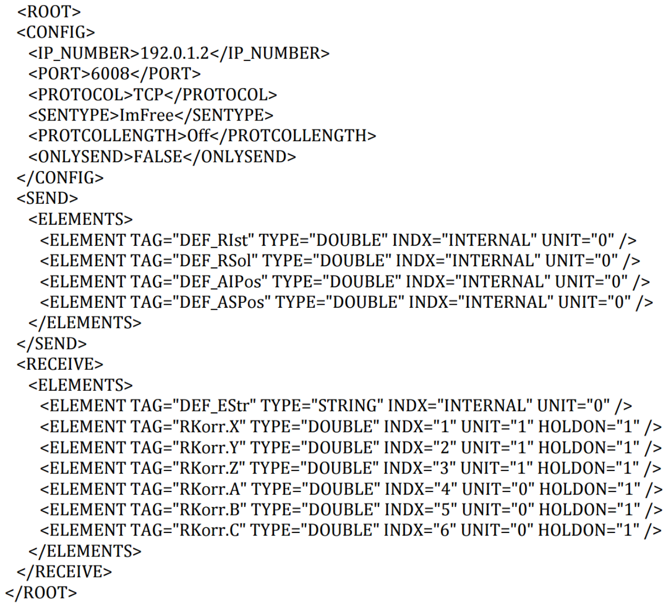


Figure A.12: Definition of the different coordinate systems.

object called ST_AXISCORR(A,B) for specific movements in axis A1 to A6, where A is the specific ID of the created object, and B is the container that the object belongs to; The second object is called ST_PATHCORR(A,B) for movements in Cartesian coordinates, where A, B are the same as for the ST_AXISCORR object.

A coordinate system is also needed (normally BASE, TCP, WORLD) as a reference for the movements. This is done by creating a RSI_object called ST_ON(A,B), where the parameter A is a string containing the coordinate system that is supposed to be used (expressed as #BASE, #TCP or #WORLD), and B is an integer value, 0 if the correction values sent to the robot shall be absolute, or 1 if they shall be relative.

A schematic picture of the data exchange sequence with the different RSI_objects is shown in Figure A.11 above. When the robot is delivered from the factory, the BASE coordinate system is the same as the WORLD and both are located in the base of the robot by default. BASE is normally moved to the base of the work piece on which the robot is working. The differences between the different coordinate systems (CS) can be seen in Figure A.12 below. NOTE: For the robot used in this thesis, all three systems; WORLD CS, ROBOT CS and BASE CS have the same origin



```
 <ROOT>
 <CONFIG>
   <IP_NUMBER>192.0.1.2</IP_NUMBER>
   <PORT>6008</PORT>
   <PROTOCOL>TCP</PROTOCOL>
   <SENTYPE>ImFree</SENTYPE>
   <PROTCOLLENGTH>Off</PROTCOLLENGTH>
   <ONLYSEND>FALSE</ONLYSEND>
 </CONFIG>
 <SEND>
   <ELEMENTS>
     <ELEMENT TAG="DEF_RIst" TYPE="DOUBLE" INDX="INTERNAL" UNIT="0" />
     <ELEMENT TAG="DEF_RSol" TYPE="DOUBLE" INDX="INTERNAL" UNIT="0" />
     <ELEMENT TAG="DEF_AIPos" TYPE="DOUBLE" INDX="INTERNAL" UNIT="0" />
     <ELEMENT TAG="DEF_ASPos" TYPE="DOUBLE" INDX="INTERNAL" UNIT="0" />
   </ELEMENTS>
 </SEND>
 <RECEIVE>
   <ELEMENTS>
     <ELEMENT TAG="DEF_EStr" TYPE="STRING" INDX="INTERNAL" UNIT="0" />
     <ELEMENT TAG="RKorr.X" TYPE="DOUBLE" INDX="1" UNIT="1" HOLDON="1" />
     <ELEMENT TAG="RKorr.Y" TYPE="DOUBLE" INDX="2" UNIT="1" HOLDON="1" />
     <ELEMENT TAG="RKorr.Z" TYPE="DOUBLE" INDX="3" UNIT="1" HOLDON="1" />
     <ELEMENT TAG="RKorr.A" TYPE="DOUBLE" INDX="4" UNIT="0" HOLDON="1" />
     <ELEMENT TAG="RKorr.B" TYPE="DOUBLE" INDX="5" UNIT="0" HOLDON="1" />
     <ELEMENT TAG="RKorr.C" TYPE="DOUBLE" INDX="6" UNIT="0" HOLDON="1" />
   </ELEMENTS>
 </RECEIVE>
</ROOT>
```

Figure A.13: The structure of *ERXconfig.xml*.

located on the base of the robot.

Though it was discovered that when starting the WORLD coordinate system, the origin where set on where the robot were standing when starting the system. This only applies for the communication to the robot controller, so that the starting position shall always be set as 0 before starting to move the robot. The response sent back to the external system was in real coordinates with the WORLD coordinate system starting at its standard position on the base of the robot.

When the robot controller communicates with the external system it interchanges XML Strings. The content in the XML strings for the demo program provided by KUKA, is decided and defined in a file called ERX_config.xml (the configuration file, mentioned above), which is located in the robot controller, inside the INIT folder.

CPU	Intel Core i7-2670QM
RAM	8GB @ 2.2GHz
GPU	GeForce GTX 560M 2GB
O.S	Ubuntu 10.04 and 12.04 prompted with a real-time kernel

Table A.5: ASUS laptop workstation.

The IP address and port to which the robot controller will connect, when establishing the connection, are set in this file. The sub-tags under <SEND> are what the robot controller sends to the external system. The most important tags for sending to the external system for in this thesis are the tags called DEF_RIst, which is the real position of the robot's end-effector, and DEF_RSol that is the set position received from the external system. A typical example of the content in this file is shown in Figure A.13.

The DEF_AIPos and DEF_ASPos are real axis position and set axis position respectively. Under the tag RECEIVE is described what the robot controller expects to receive from the external system. In this example only corrections in six values (X, Y, Z, A, B and C) are included, tagged as RKorr.

High Level Controllers

As in the case of vision sensors, different kind of controllers has been tested up along the realization of this thesis: PCs, industrial workstation, PLC and embedded hardware (i.e. PANDA board, IGEPv2). Here, it is described only the ones used in the validations presented in above chapters, which are a laptop workstation and the Machine Controller PLC NJ501-1500. Tables A.5 and A.6 shows the main characteristics of both of them.

A.1.1.5 Software

Depending on the hardware and the concrete application to be developed, along the years of this thesis several software platforms have been learnt

Dimensions (HxDxW)	90 mm x 90 mm x 90 mm
Memory capacity for variables	2 MB: Retained; 4 MB: Not retained
Programming	SYSMAC-SE200D
CPU	Intel Atom Processor
Connection	USB 2.0-compliant, Ethernet, EtherCAT, DIO
Protocol	EtherCAT, EthernetIP, Serial

Table A.6: Machine Controller PLC NJ501-1500.

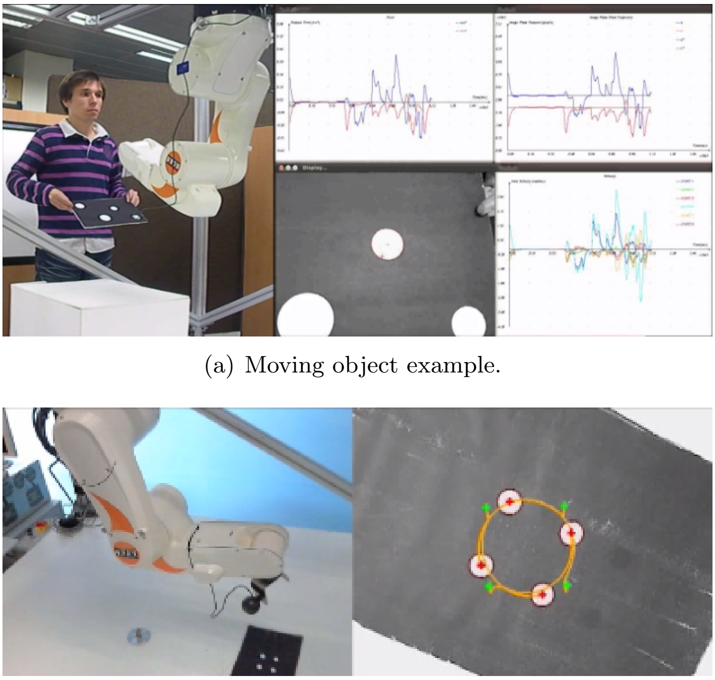
and used, ranging from commercial software (i.e. Matlab/Simulink, Matrox Imaging Library, OMRON "Application Producer"), or open source libraries (i.e. Orocos Toolchain, Soetens (2006), ViSP, Marchand et al. (2005), openCV, Bradski (2000), ROS, Quigley et al. (2009)).

A.1.1.6 Applications Highlights

During the process of this research, several applications have been developed in this platform. Here we highlight the most relevant from the point of view of this thesis, since whether the author or the approaches have had an important role on their success.

Visual feedback control of robots was not a research topic at the Institute of Design and Manufacturing. In fact, this thesis has opened this new research line with promising results that can be transferred to the industry in a near future. Figure A.14 shows two applications, differentiates by if the object is moving or stationary.

The first contribution of this thesis, visual feedback control based to dual-rate high order holders, was implemented successfully in industrial hardware to solve the latency problem produced by the vision system Xpectia when controlling KUKA industrial robot manipulators. This project won the Annual OMRON Price "Iniciación a la Investigación e Innovación en Automática" in 2012/13. The application consisted in pointing at defects on body cars previously detected by a vision system (see Figure A.15). The inclusion of the multi-rate approach shown a very important improvement of the system concerning to the convergence time and also with the smoothness



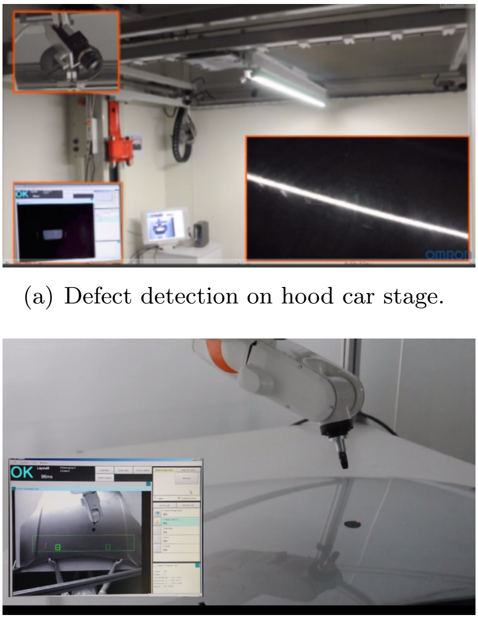
(a) Moving object example.

(b) Stationary object example.

Figure A.14: Visual feedback control.

of the motion of the robot manipulator. In addition, a grasping application was also performed demonstrating that, using the same hardware but with the proposed dual-rate approach, the manipulator was able to grasp objects moving at higher velocity, where classic single-rate approaches failed.

Other interesting application developed under the guidance and help of the researcher Aldolfo Muñoz, was a real-time user control of industrial manipulators via multi-touch screens (see Figure A.17). In this case, the robot was controlled by a non-expert user via an interface thought precisely to help such users to interact with robots without any knowledge of robot theory.



(a) Defect detection on hood car stage.

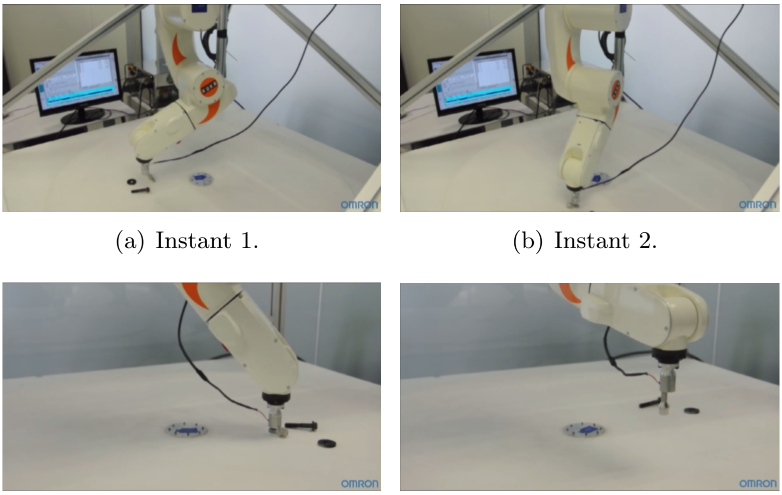
(b) Defect tracking on car hood stage.

Figure A.15: OMRON prize 2012/2013 - Visual feedback control based on dual-rate holders: tracking application.

Several Human Machine Interfaces (e.g. Figures A.1.1.6 and A.1.1.6) were tested with non-expert individuals and, taking into account their comments, the final interface was the one shown in Figures A.1.1.6 and A.1.1.6.

A.1.2 SUMMIT Wheeled Robot Platform

The mobile platform SUMMIT is a mobile robot which has high mobility and it is thought for both indoor and outdoor usage. Its mechanics are equivalent to one 4x4 and uses high quality chassis manufactured with aluminum. The robot can be remote-controlled or autonomous since it has been included a stereo vision system, a laser, RTK-DGPS, inertial sensor,



(a) Instant 1.

(b) Instant 2.

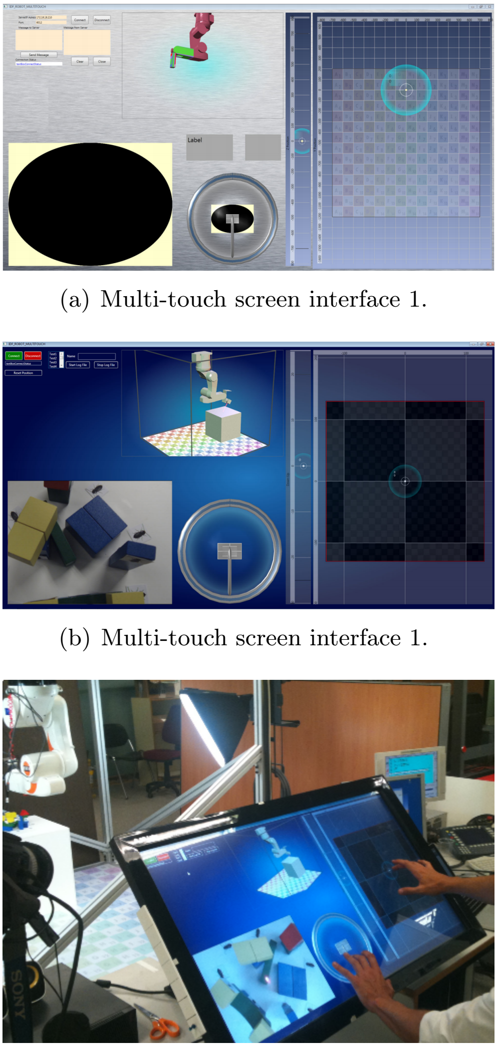
(c) Instant 3.

(d) Instant 4.

Figure A.16: OMRON prize 2012/2013 - Visual feedback control based on dual-rate holders: grasping application.

Dimensions	567x389x323
Weight	12.9 kg
Loading capacity	5 Kg
Max velocity	3 m/s
Safety level	IP54
Autonomy	240 minutes
Traction system	4 wheels
Batteries	4x3.3V LiFePO4
Motors	2x brushless
Temperature range	from 0°C to 50°C
Interior Connectivity	USB, RS232, GPIO y RJ45
Exterior connectivity	USB
Communication	WiFi 802.11n
Kinematics	Ackerman

Table A.7: SUMMIT main characteristics.



(a) Multi-touch screen interface 1.

(b) Multi-touch screen interface 1.

(c) Platform set-up.

Figure A.17: Real-time control of industrial manipulators via multi-touch screens.

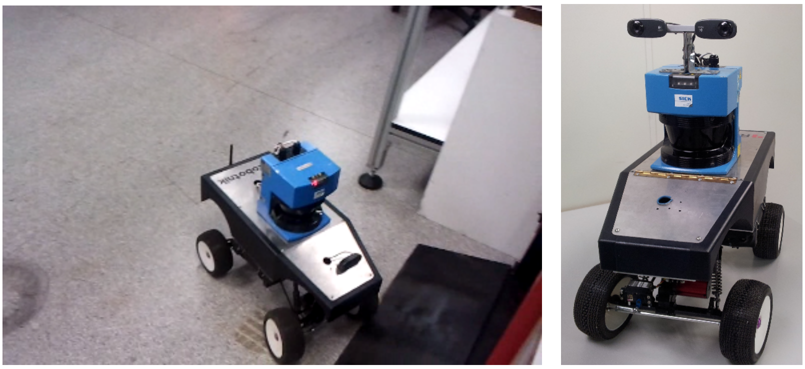

(a) (b)

Figure A.18: SUMMIT mobile platform.

among others.

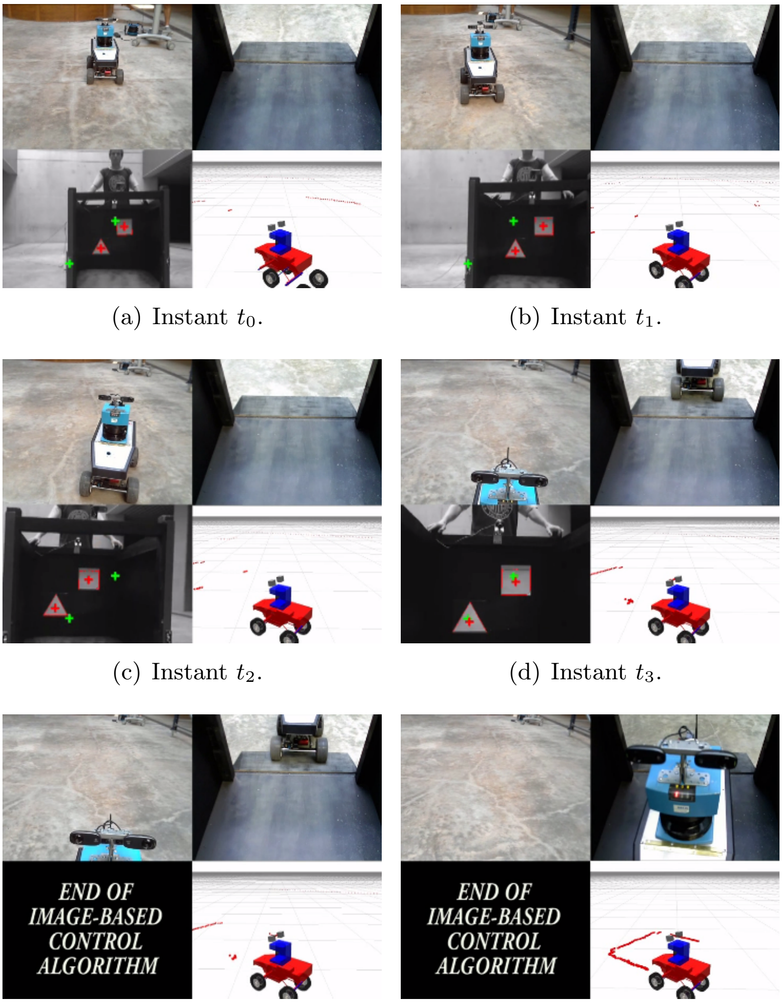
The control diagram is open and modular, based on ROS, Quigley et al. (2009), which allows whether programming or simulating the algorithms off-line and then be tested on-line.

The main mechanical characteristics of this robot are summarized in Table A.7.

This platform has been used mainly for line and path following tasks. One application in the context of this thesis was a moving robot docking using vision feedback control. Figure A.19 shows how SUMMIT docks successfully inside the moving dock in this case. Still, several issues regarding to the loss of the image features while the docking is being performed, are being studied at the time this manuscript is written.

A.2 Virtual Platforms

Several virtual platforms have been developed through the course of this thesis. However, in the present manuscript has been presented the results obtained by using two of them: Industrial Robotic Cell Simulator and SUM-



(a) Instant t_0.

(b) Instant t_1.

(c) Instant t_2.

(d) Instant t_3.

(e) Instant t_4.

(f) Instant t_5.

Figure A.19: Visual servoing application: wheeled robot docking with moving dock.

MIT Mobile Platform Simulator. Following a brief description of each simulator is given.

A.2.1 Industrial Robotic Cell Simulator

This simulator was developed by J. Ernesto Solanes on Linux O.S, concretely Ubuntu 12.04, using open source libraries, being the most relevant:

- **ViSP, Marchand et al. (2005):** standing for Visual Servoing Platform is a modular cross platform library that allows prototyping and developing applications using visual tracking and visual servoing techniques at the heart of the researches done by Inria Lagadic team. It provides a set of visual features that can be tracked using real-time image processing or computer vision algorithms. An overview of its main characteristics can be seen in Figure A.20.

  Several vision algorithms from this framework have been used to perform some of the tests shown along this thesis.

- **OpenRave, Diankov and Kuffner (2008):** OpenRAVE provides an environment for testing, developing, and deploying motion planning algorithms in real-world robotics applications (see Figure A.21 for an example). The main focus is on simulation and analysis of kinematic and geometric information related to motion planning. OpenRAVE's stand-alone nature allows is to be easily integrated into existing robotics systems.

  It provides many command line tools to work with robots and planners, and the run-time core is small enough to be used inside controllers and bigger frameworks. An important target application is industrial robotics automation.

  In this thesis, simulation and rendering functions are used, while kinematics, dynamics, vision and control algorithms are programmed externally to this library.

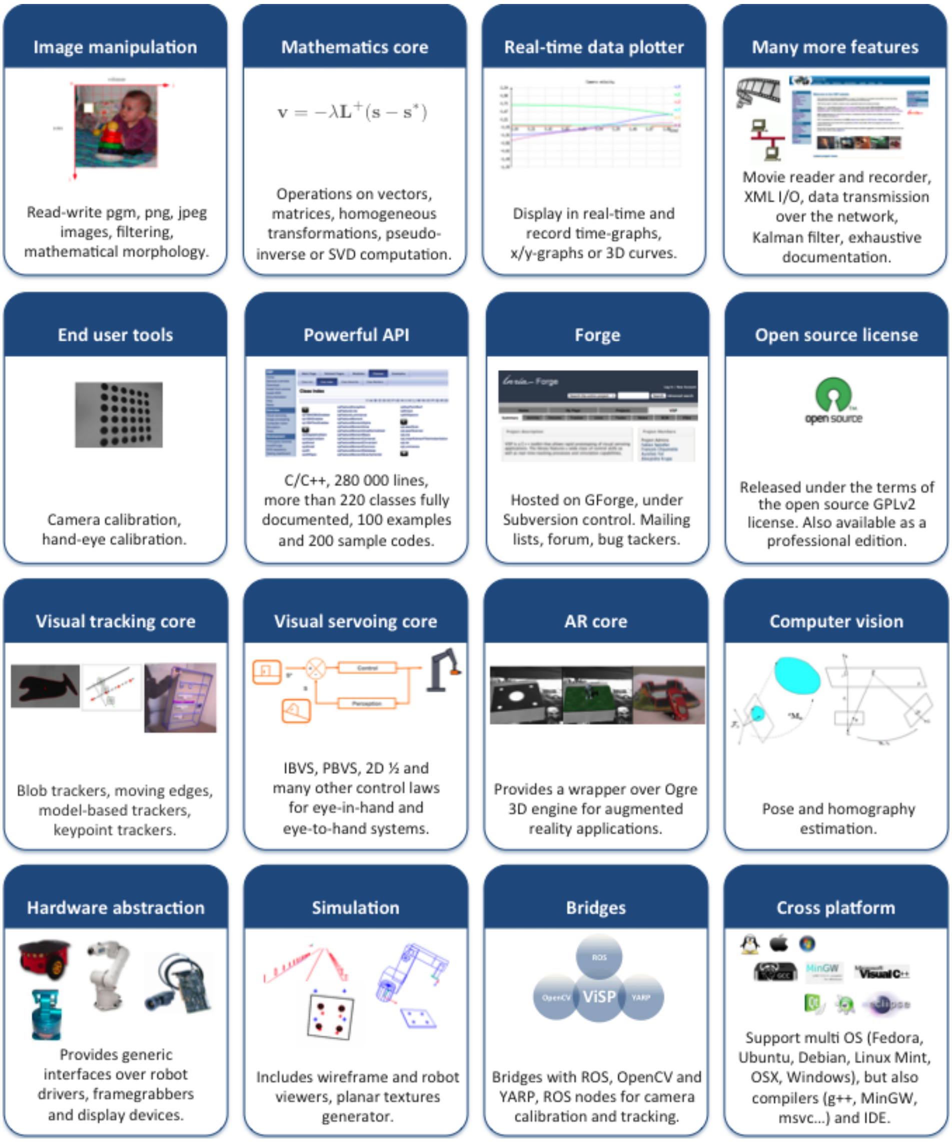


Image manipulation

Read-write pgm, png, jpeg images, filtering, mathematical morphology.

Mathematics core

$$\mathbf{v} = -\lambda \mathbf{L}^{+}(\mathbf{s} - \mathbf{s}^{*})$$

Operations on vectors, matrices, homogeneous transformations, pseudo-inverse or SVD computation.

Real-time data plotter

Display in real-time and record time-graphs, x/y-graphs or 3D curves.

Many more features

Movie reader and recorder, XML I/O, data transmission over the network, Kalman filter, exhaustive documentation.

End user tools

Camera calibration, hand-eye calibration.

Powerful API

C/C++, 280 000 lines, more than 220 classes fully documented, 100 examples and 200 sample codes.

Forge

Hosted on GForge, under Subversion control. Mailing lists, forum, bug tackers.

Open source license

Released under the terms of the open source GPLv2 license. Also available as a professional edition.

Visual tracking core

Blob trackers, moving edges, model-based trackers, keypoint trackers.

Visual servoing core

IBVS, PBVS, 2D ½ and many other control laws for eye-in-hand and eye-to-hand systems.

AR core

Provides a wrapper over Ogre 3D engine for augmented reality applications.

Computer vision

Pose and homography estimation.

Hardware abstraction

Provides generic interfaces over robot drivers, framegrabbers and display devices.

Simulation

Includes wireframe and robot viewers, planar textures generator.

Bridges

Bridges with ROS, OpenCV and YARP, ROS nodes for camera calibration and tracking.

Cross platform

Support multi OS (Fedora, Ubuntu, Debian, Linux Mint, OSX, Windows), but also compilers (g++, MinGW, msvc...) and IDE.

Figure A.20: ViSP main funtionalities, Marchand et al. (2005).

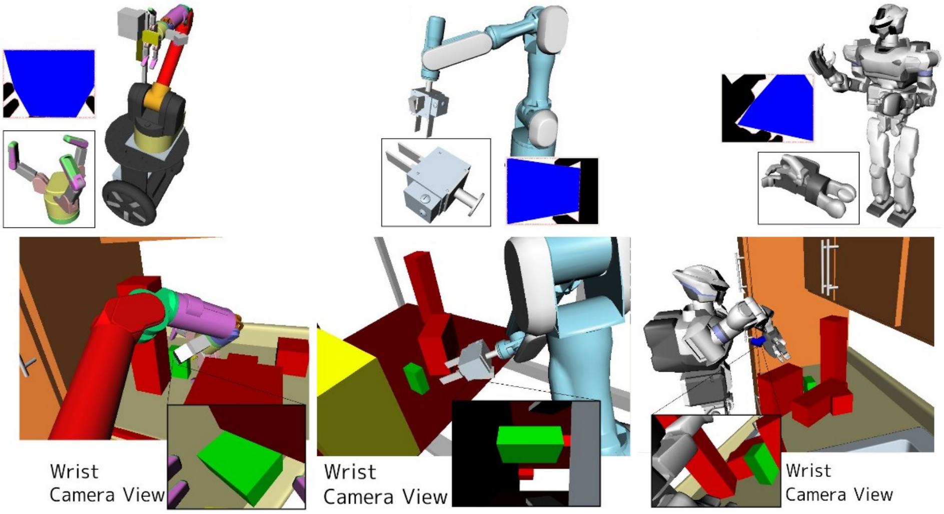


Figure A.21: OpenRave visibility example, Diankov and Kuffner (2008).

Apart from these libraries, a kinematic/dynamic specific library for 6DOF Kuka industrial robot manipulators was developed, in charge to get access to all possible kinematic transformations, robot Jacobean, and on, as well as the required robot dynamic parameters needed by the controller.

Therefore, Figure A.22 shows the virtual environment developed for Kuka Kr5 sixx R650, and Figure A.23 the one for Kuka Agilus R900 sixx.

A.2.2 SUMMIT Mobile Platform Simulator

This simulator was developed by Vicent Girbés on Linux O.S., concretely Ubuntu 10.04, using open source libraries, being the most relevant:

- **Gazebo, Koenig and Howard (2004); Aguero et al. (2015):** is a well-designed simulator that makes possible to rapidly test algorithms, design robots, and perform regression testing using realistic scenarios. Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. It

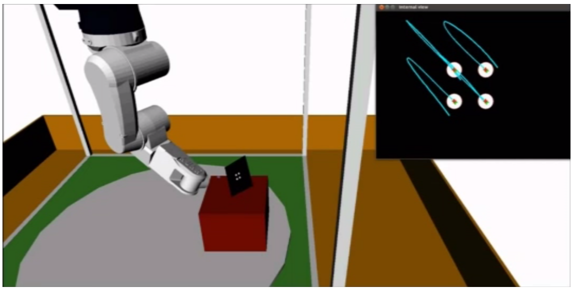


Figure A.22: Kr5 virtual environment.

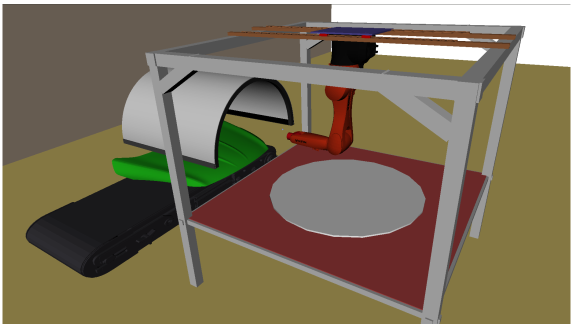


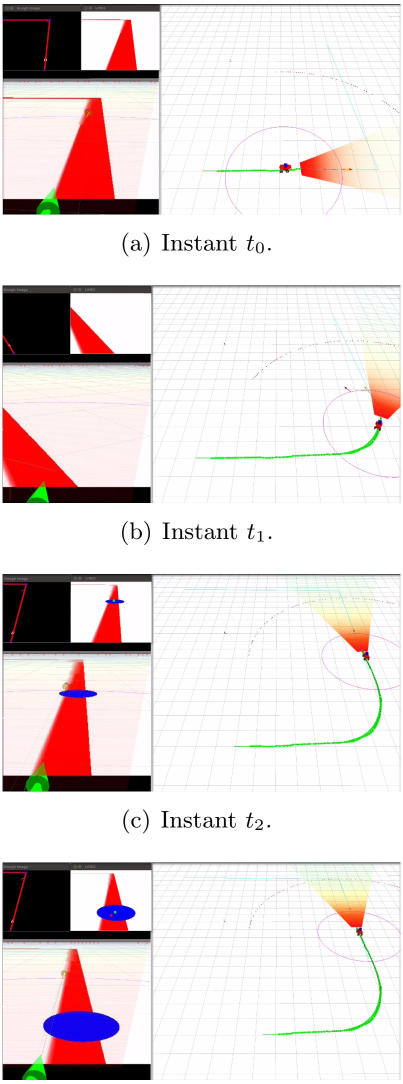
Figure A.23: Agilus virtual environment.

is a robust physics engine, has high-quality graphics, and convenient programmatic and graphical interfaces.

- **OpenCV, Bradski (2000):** It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

Some ROS stacks have also been used to develop this simulator, such as the Navigation stack.

In Figure A.24 is shown a set of snapshots of this simulator. In this case, SUMMIT robot performs a line following task stopped by the detection of a visual mark.



(a) Instant t_0.

(b) Instant t_1.

(c) Instant t_2.

(d) Instant t_3.

Figure A.24: wheeled robot docking with moving dock.

Bibliography

Abdul Hafez, A., Achar, S., and Jawahar, C. (2008). Visual servoing based on gaussian mixture models. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3225–3230.

Agravante, D., Pages, J., and Chaumette, F. (2013). Visual servoing for the reem humanoid robot's upper body. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5253–5258.

Aguero, C., Koenig, N., Chen, I., Boyer, H., Peters, S., Hsu, J., Gerkey, B., Paepcke, S., Rivero, J., Manzo, J., Krotkov, E., and Pratt, G. (2015). Inside the virtual robotics challenge: Simulating real-time robotic disaster response. *Automation Science and Engineering, IEEE Transactions on*, 12(2):494–506.

Ahrens, J., Tan, X., and Khalil, H. (2009). Multirate sampled-data output feedback control with application to smart material actuated systems. *Automatic Control, IEEE Transactions on*, 54(11):2518 –2529.

Albertos, P., Salt, J., and Tornero, J. (1990). Dual-rate adaptative control. *Automatica*, 32(7):1027–1030.

Allen, P., Timcenko, A., Yoshimi, B., and Michelman, P. (1993). Automated tracking and grasping of a moving object with a robotic hand-eye system. *Robotics and Automation, IEEE Transactions on*, 9(2):152–165.

Allibert, G. and Courtial, E. (2009). What can prediction bring to image-based visual servoing ? In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5210–5215.

Allibert, G., Courtial, E., and Chaumette, F. (2010). Visual servoing via nonlinear predictive control. In Chesi, G. and Hashimoto, K., editors, *Visual Servoing via Advanced Numerical Methods*, pages 375–394. LNCIS 401, Springer-Verlag.

Anderson, D. (2007). Multirate and nonlinear controllers for low-cost laser tracking systems. In *17th IFAC Symposium on Automatic Control in Aerospace.*

Ansar, A. and Daniilidis, K. (2003). Linear pose estimation from points or lines. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):578–589.

Araki, M. and Hagiwara, T. (1985). Pole assignment by multirate sampled-data output feedback. In *Decision and Control, 1985 24th IEEE Conference on*, pages 189–193.

Armesto, L., Chroust, S., Vincze, M., and Tornero, J. (2004). Multi-rate fusion with vision and inertial sensors. In *Int. Conf. on Robotics and Automation*, pages 193–199.

Armesto, L., Girbes, V., Sala, A., Zima, M., and Smidl, V. (2015). Duality-based nonlinear quadratic control: Application to mobile robot trajectory-following. *Control Systems Technology, IEEE Transactions on*, PP(99):1–1.

Armesto, L., Ippoliti, G., Longhi, S., and Tornero, J. (2008). Probabilistic self-localization and mapping - an asynchronous multirate approach. *Robotics and Automation Magazine*, 15(2):77–88.

Armesto, L. and Tornero, J. (2003a). Dual-rate high order holds based on primitive functions. In *American Control Conf.*, pages 1140–1145.

Armesto, L. and Tornero, J. (2003b). A general formulation for generating multi-rate models. In *American Control Conf.*, pages 1146–1151.

Armesto, L. and Tornero, J. (2004). Simultaneous localization and mapping based on multi-rate fusion of laser and encoders measurements. In II, editor, *International Conference on Informatics for Control, Automation and Robotics*, pages 435–439.

Armesto, L. and Tornero, J. (2005). *Técnicas de Control y Fusión Sensorial Multifrecuenciales y su Aplicación a la Robótica Móvil.*

Armesto, L. and Tornero, J. (2006). Linear quadratic gaussian regulators for multi-rate sampled-data stochastic systems. In Andrade-Cetto, J., Ferrier, J.-L., Pereira, J. D., and Filipe, J., editors, *ICINCO-SPSMC*, pages 67–74. INSTICC Press.

Assa, A. and Janabi-Sharifi, F. (2014). Robust model predictive control for visual servoing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2715–2720.

Bakthavatchalam, M., Chaumette, F., and Marchand, E. (2013). Photometric moments: New promising candidates for visual servoing. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5241–5246.

Bakthavatchalam, M., Tahri, O., and Chaumette, F. (2014). Improving moments-based visual servoing with tunable visual features. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6186–6191.

Bamieh, B., Pearson, J., B.A., F., and A., T. (1991). A lifting technique for linear periodic systems with applications to sampled-data control. *Systems and Control Letters*, 17:79–88.

Barajas, M., Davalos-Viveros, J., Garcia-Lumbreras, S., and Gordillo, J. (2013). Visual servoing of uav using cuboid model with simultaneous tracking of multiple planar faces. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 596–601.

Bellman, R. (1954). The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6):503–515.

Benhimane, S. and Malis, E. (2007). Homography-based 2d visual tracking and servoing. *Int. J. Rob. Res.*, 26(7):661–676.

Bourquardez, O., Mahony, R., Guenard, N., Chaumette, F., Hamel, T., and Eck, L. (2009). Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Trans. on Robotics*, 25(3):743–749.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools.*

Branicky, M. (1997). Stability of hybrid systems: state of the art. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 1, pages 120–125 vol.1.

Bruyninckx, H., Soetens, P., and Koninckx, B. (2003). The real-time motion control core of the orocos project. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 2766–2771 vol.2.

Camacho, E. and Bordons, C. (2004). *Model Predictive Control.* Advanced Textbooks in Control and Signal Processing. Springer London.

Carbajal-Espinosa, O., Osuna-Gonzalez, G., Gonzalez-Jimenez, L., Loukianov, A., and Bayro Corrochano, E. (2014). Visual servoing and robust object manipulation using symmetries and conformai geometric algebra. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 1051–1056.

Cervera, E., Pobil, A. P. D., Berry, F., and Martinet, P. (2003). Improving image-based visual servoing with three-dimensional features. *I. J. Robotic Res.*, 22(10-11):821–840.

Chammas, A. B. and Leondes, C. T. (1978). On the design of linear time invariant systems by periodic output feedback part i. discrete-time pole assignment. *International Journal of Control*, 27(6):885–894.

Chammas, A. B. and Leondes, C. T. (1979). Pole assignment by piecewise constant output feedback. *International Journal of Control*, 29(1):31–38.

Chang, H. J., Kim, P. J., Song, D. S., and Choi, J.-Y. (2009). Optical image stabilizing system using multirate fuzzy pid controller for mobile device camera. *Consumer Electronics, IEEE Transactions on*, 55(2):303–311.

Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In Kriegman, D., Hager, G., and Morse, A., editors, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag.

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

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

Chaumette, F. and Malis, E. (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*, volume 1, pages 630 –635 vol.1.

Cherubini, A. and Chaumette, F. (2013). Visual navigation of a mobile robot with laser-based collision avoidance. *Int. Journal of Robotics Research*, 32(2):189–209.

Chesi, G. (2009a). Estimation of the camera pose from image point correspondences through the essential matrix and convex optimization. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 35–40.

Chesi, G. (2009b). Visual servoing path planning via homogeneous forms and lmi optimizations. *Robotics, IEEE Transactions on*, 25(2):281–291.

Chesi, G., Hashimoto, K., Prattichizzo, D., and Vicino, A. (2004). Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. *Robotics, IEEE Transactions on*, 20(5):908–914.

Chesi, G. and Vicino, A. (2004). Visual servoing for large camera displacements. *Robotics, IEEE Transactions on*, 20(4):724–735.

Colaneri, P. and de Nicolao, G. (1995). Multirate LQG control of continuous-time stochastic systems. *Automatica*, 31:591–596.

Collewet, C., Marchand, E., and Chaumette, F. (2008). Visual servoing set free from image processing. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 81–86.

Corke, P. (1996). A robotics toolbox for matlab. *Robotics Automation Magazine, IEEE*, 3(1):24 –32.

Corke, P. (2005). The machine vision toolbox: a matlab toolbox for vision and vision-based control. *Robotics Automation Magazine, IEEE*, 12(4):16 – 25.

Corke, P. (2011). *Robotics, Vision and Control - Fundamental Algorithms in MATLABÂ®*, volume 73 of *Springer Tracts in Advanced Robotics*. Springer.

Corke, P. and Hutchinson, S. (2001). A new partitioned approach to image-based visual servo control. *Robotics and Automation, IEEE Transactions on*, 17(4):507–515.

Corke, P., Spindler, F., and Chaumette, F. (2009). Combining cartesian and polar coordinates in ibvs. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5962–5967.

De Luca, A., Ferri, M., Oriolo, G., and Giordano, P. (2008). Visual servoing with exploitation of redundancy: An experimental study. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3231–3237.

DeMenthon, D. F. and Davis, L. S. (1995a). Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141.

DeMenthon, D. F. and Davis, L. S. (1995b). Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141.

Deng, L., Janabi-Sharifi, F., and Wilson, W. (2002). Hybrid strategies for image constraints avoidance in visual servoing. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 348–353 vol.1.

Deng, L., Janabi-Sharifi, F., and Wilson, W. J. (2005). Hybrid motion control and planning strategies for visual servoing. *Industrial Electronics, IEEE Transactions on*, 52(4):1024–1040.

Deng, L., Wilson, W., and Janabi-Sharifi, F. (2003). Dynamic performance of the position-based visual servoing method in the cartesian and image spaces. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 510–515 vol.1.

Diankov, R. and Kuffner, J. (2008). OpenRAVE: A planning architecture for autonomous robotics. Technical report, Robotics Institue, Carnegie Mellon University. `http://openrave.org/`.

Dombre, E. and W., K. (2007). Robot manipulator modeling, performance analysis and control. In *ISTE,Cambridge, Massachusetts.*

Dornaika, F. and Garcia, C. (1999). Pose estimation using point and line correspondences. *Real-Time Imaging*, 5(3):215 – 230.

Ernesto Solanes, J., Armesto, L., Tornero, J., and Girbes, V. (2013). Improving image-based visual servoing with reference features filtering. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3083–3088.

Espiau, B., Chaumette, F., and Rives, P. (1991). A new approach to visual servoing in robotics. In *Geometric Reasoning for Perception and Action 91*, pages 106–136.

Feddema, J., Lee, C., and Mitchell, O. (1991). Weighted selection of image features for resolved rate visual feedback control. *Robotics and Automation, IEEE Transactions on*, 7(1):31–47.

Ficocelli, M. and Janabi-Sharifi, F. (2001). Adaptive filtering for pose estimation in visual servoing. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 19–24 vol.1.

Flandin, G., Chaumette, F., and Marchand, E. (2000). Eye-in-hand/eye-to-hand cooperation for visual servoing. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 3, pages 2741–2746 vol.3.

Fomena, R., Yoon, H. U., Cherubini, A., Chaumette, F., and Hutchinson, S. (2009). Coarsely calibrated visual servoing of a mobile robot using a catadioptric vision system. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5432–5437.

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*, volume 5, pages 5408 – 5413 Vol.5.

Fujimoto, H., Kawamura, A., and Tomizuka, M. (1999). Generalized digital redesign method for linear feedback system based on n-delay control. *Mechatronics, IEEE/ASME Transactions on*, 4(2):101–109.

Gans, N. and Hutchinson, S. (2003a). An asymptotically stable switched system visual controller for eye in hand robots. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 735–742 vol.1.

Gans, N. and Hutchinson, S. (2003b). An experimental study of hybrid switched system approaches to visual servoing. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 3061–3068 vol.3.

Gans, N. and Hutchinson, S. (2007). Stable visual servoing through hybrid switched-system control. *Robotics, IEEE Transactions on*, 23(3):530–540.

Gans, N. R., Hutchinson, S., and Corke, P. I. (2003). Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control. *I. J. Robotic Res.*, 22(10-11):955–984.

Geraud, B., Le Besnerais, G., and Foulon, G. (1998). Determination of a dense depth map from an image sequence: application to aerial imagery. *Proc. SPIE*, 3500:345–356.

Grabe, V., BÃ¼lthoff, H.-H., Scaramuzza, D., and Robuffo Giordano, P. (2015). Nonlinear ego-motion estimation from optical flow for online control of a quadrotor uav. *The International Journal of Robotics Research, IJRR.*

Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46.

Hafez, A., Cervera, E., and Jawahar, C. (2007). Optimizing image and camera trajectories in robot vision control using on-line boosting. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 352–357.

Hager, G. (1997). A modular system for robust positioning using feedback from stereo vision. *Robotics and Automation, IEEE Transactions on*, 13(4):582–595.

Haralick, R., Joo, H., Lee, D., Zhuang, S., Vaidya, V., and Kim, M. (1989). Pose estimation from corresponding point data. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(6):1426–1446.

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., Ebine, T., and Kimura, H. (1996). Visual servoing with hand-eye manipulator-optimal control approach. *Robotics and Automation, IEEE Transactions on*, 12(5):766–774.

Hashimoto, K. and Noritsugu, T. (2000a). Enlargement of stable region in visual servo. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 4, pages 3927–3932 vol.4.

Hashimoto, K. and Noritsugu, T. (2000b). Potential problems and switching control for visual servoing. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 1, pages 423–428 vol.1.

Heshmati-Alamdari, S., Eqtami, A., Karras, G., Dimarogonas, D., and Kyriakopoulos, K. (2014). A self-triggered visual servoing model predictive control scheme for under-actuated underwater robotic vehicles. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3826–3831.

Hollmuler, W. and Harle, N. (1999). A kalman-filter approach for the multirate-control problem in visual servoing systems. In *Australian Robotics and Automation Association, ARAA '99*.

Hu, G., Gans, N., Fitz-Coy, N., and Dixon, W. (2010). Adaptive homography-based visual servo tracking control via a quaternion formulation. *Control Systems Technology, IEEE Transactions on*, 18(1):128–135.

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

Iwatsuki, M. and Okiyama, N. (2002). A new formulation of visual servoing based on cylindrical coordinate system with shiftable origin. In *Intelli-*

gent Robots and Systems, 2002. IEEE/RSJ International Conference on, volume 1, pages 354–359 vol.1.

Iwatsuki, M. and Okiyama, N. (2005). A new formulation of visual servoing based on cylindrical coordinate system. *Robotics, IEEE Transactions on*, 21(2):266–273.

Janabi-Sharifi, F., Deng, L., and Wilson, W. J. (2011). Comparison of basic visual servoing methods. *Mechatronics, IEEE/ASME Transactions on*, 16(5):967–983.

Janabi-Sharifi, F. and Wilson, W. (1997). Automatic selection of image features for visual servoing. *Robotics and Automation, IEEE Transactions on*, 13(6):890–903.

J.T Feddema, C.S.G. Lee, O. M. (1991). *Weighted selection of image features for resolved rate visual feedback control.* IEEE Transactions on Robotics and Automation.

Kabamba, P. (1987). Control of linear systems using generalized sampled-data hold functions. *Automatic Control, IEEE Transactions on*, 32(9):772–783.

Kaczorek, T. (1985). Pole placement for linear discrete-time systems by periodic output feedbacks. *Systems and Control Letters*, 6(4):267 – 269.

Kallem, V., Dewan, M., Swensen, J., Hager, G., and Cowan, N. (2007). Kernel-based visual servoing. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1975–1980.

Kárný, M. (1996). Towards fully probabilistic control design. *Automatica*, 32(12):1719–1722.

Kim, J., Kumar, N., Panwar, V., Borm, J.-H., and Chai, J. (2012). Adaptive neural controller for visual servoing of robot manipulators with camera-in-hand configuration. *Journal of Mechanical Science and Technology*, 26(8):2313–2323.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan.

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Kragic, D. and Christensen, H. I. (2002). Survey on visual servoing for manipulation. Technical report, COMPUTATIONAL VISION AND ACTIVE PERCEPTION LABORATORY.

Krupa, A., Folio, D., Novales, C., Vieyres, P., and Li, T. (2015). Robotized tele-echography: an assisting visibility tool to support expert diagnostc. *IEEE Systems Journal*.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

Kwon, W. and Pearson, A. (1977). A modified quadratic cost problem and feedback stabilization of a linear system. *Automatic Control, IEEE Transactions on*, 22(5):838–842.

Kyrki, V., Kragic, D., and Christensen, H. (2004a). Measurement errors in visual servoing. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1861–1867 Vol.2.

Kyrki, V., Kragic, D., and Christensen, H. (2004b). New shortest-path approaches to visual servoing. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 349–354 vol.1.

Leondes, C., Peller, J., and Stear, E. B. (1970). Nonlinear smoothing theory. *Systems Science and Cybernetics, IEEE Transactions on*, 6(1):63–71.

Li, T., Kermorgant, O., and Krupa, A. (2012). Maintaining visibility constraints during tele-echography with ultrasound visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'12*, pages 4856–4861, Saint Paul, USA.

Li, W. and Todorov, E. (2004). Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO 2004, Proceedings of the First International Conference on Informatics in Control, Automation and Robotics, Setúbal, Portugal, August 25-28, 2004*, pages 222–229.

Lippiello, V., Siciliano, B., and Villani, L. (2004). Visual motion estimation of 3d objects: an adaptive extended kalman filter approach. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 957–962 vol.1.

Liu, M. and Wong, K. (1999). Pose estimation using four corresponding points. *Pattern Recognition Letters*, 20(1):69 – 74.

Liu, S., Xie, W.-F., and Su, C.-Y. (2009). Image-based visual servoing using improved image moments. In *Information and Automation, 2009. ICIA '09. International Conference on*, pages 577–582.

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

Malis, E. and Chaumette, F. (2000). 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97.

Malis, E., Chaumette, F., and Boudet, S. (1999). 2-1/2-d visual servoing.

Malis, E. and Cipolla, R. (2002). Camera self-calibration from unknown planar structures enforcing the multiview constraints between collineations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1268–1272.

Malis, E., Mezouar, Y., and Rives, P. (2010). Robustness of image-based visual servoing with a calibrated camera in the presence of uncertainties in the three-dimensional structure. *Robotics, IEEE Transactions on*, 26(1):112–120.

Mansard, N., Remazeilles, A., and Chaumette, F. (2009). Continuity of varying-feature-set control laws. *Automatic Control, IEEE Transactions on*, 54(11):2493–2505.

Marchand, E. and Chaumette, F. (2002). Virtual visual servoing: a framework for real-time augmented reality.

Marchand, E., Spindler, F., and Chaumette, F. (2005). Visp for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4):40–52. `http://www.irisa.fr/lagadic/visp/visp.html`.

Marey, M. (2010). *Contributions to control modeling in visual servoing, task redundancy, and joint limits avoidance.* PhD Thesis Université de Rennes 1, Mention informatique, December 2010.

Martinet, P., Gallice, J., and Khadraoui, D. (1996). Vision based control law using 3d visual features. In *Committees, Econometrica*, pages 497–502.

Matthies, L. H., Szeliski, R., and Kanade, T. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *Int. J. Computer Vision*, 3(3):209–236.

Max, G. and Lantos, B. (2014). Nonlinear moving horizon predictive control of ground vehicles. In *Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on*, pages 421–426.

Mayne, D. and Michalska, H. (1990). Receding horizon control of nonlinear systems. *Automatic Control, IEEE Transactions on*, 35(7):814–824.

McInroy, J. and Qi, Z. (2008). A novel pose estimation algorithm based on points to regions correspondence. *Journal of Mathematical Imaging and Vision*, 30(2):195–207.

Mezouar, Y. and Chaumette, F. (2002). Path planning for robust image-based control. *Robotics and Automation, IEEE Transactions on*, 18(4):534–549.

Misra, J. and Saha, I. (2010). Artificial neural networks in hardware: A survey of two decades of progress. *Neurocomputing*, 74(1-3):239 – 255. Artificial Brains.

Mita, T., Chida, Y., Kaku, Y., and Numasato, H. (1990). Two-delay robust digital control and its applications-avoiding the problem on unstable limiting zeros. *Automatic Control, IEEE Transactions on*, 35(8):962–970.

Monaco, S. and Normand-Cyrot, D. (1992). An introduction to motion planning under multirate digital control. In *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, pages 1780–1785 vol.2.

Moore, K., Bhattacharyya, S., and Dahleh, M. (1989). Arbitrary pole and zero assignment with n-delay input control using stable controllers. In *Decision and Control, 1989., Proceedings of the 28th IEEE Conference on*, pages 1253–1258 vol.2.

Morales, J., Martinez, J., Martinez, M., and Mandow, A. (2009). Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2d laser scanner. *EURASIP Journal on Advances in Signal Processing*, 2009(1):935237.

Nguyen, T., Mann, G., and Gosine, R. (2014). Vision-based qualitative path-following control of quadrotor aerial vehicle. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 412–417.

Nissen, S. (2003). Implementation of a fast artificial neural network library (fann). *Report, Department of Computer Science University of Copenhagen (DIKU)*, 31.

Ogre3D (2007). Website. http://www.ogre3d.org.

Ozawa, R. and Chaumette, F. (2011). Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach. In *IEEE Int. Conf. on Robotics and Automation, ICRA'11*, pages 5670–5676, Shanghai, China.

Pages, J., Collewet, C., Chaumette, F., and Salvi, J. (2006). Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light. *Robotics, IEEE Transactions on*, 22(5):1000–1010.

Park, D.-H., Kwon, J.-H., and Ha, I.-J. (2012). Novel position-based visual servoing approach to robust global stability under field-of-view constraint. *Industrial Electronics, IEEE Transactions on*, 59(12):4735–4752.

Pasteau, F., Narayanan, V. K., Babel, M., and Chaumette, F. (2014). A visual servoing approach for autonomous corridor following and doorway passing in a wheelchair. *Robotics and Autonomous Systems*, 26(0):–.

Pomares, J., Chaumette, F., and Torres, F. (2007). Adaptive visual servoing by simultaneous camera calibration. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2811–2816.

Pontryagin, L. S. (1959). Optimal control processes. *Uspehi Mat. Nauk.*

Puget, P. and Skordas, T. (1990). An optimal solution for mobile camera calibration. In Faugeras, O., editor, *Computer Vision ECCV 90*, volume 427 of *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg.

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Rauch, H. E., Striebel, C. T., and Tung, F. (1965). Maximum likelihood estimates of linear dynamic systems. *Journal of the American Institute of Aeronautics and Astronautics*, 3(8):1445–1450.

Reddy, G., Bandyopadhyay, B., and Tiwari, A. (2007). Multirate output feedback based sliding mode spatial control for a large phwr. *Nuclear Science, IEEE Transactions on*, 54(6):2677–2686.

Safaee-Rad, R., Tchoukanov, I., Benhabib, B., and Smith, K. (1992). 3d-pose estimation from a quadratic-curved feature in two perspective views. In *Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, pages 341–344.

Sasajima, K. and Fujimoto, H. (2007). 6 dof multirate visual servoing for quick moving objects. In *American Control Conference, 2007. ACC '07*, pages 1538–1543.

Sauvee, M., Poignet, P., Dombre, E., and Courtial, E. (2006). Image based visual servoing through nonlinear model predictive control. In *Decision and Control, 2006 45th IEEE Conference on*, pages 1776–1781.

Shademan, A. and Janabi-Sharifi, F. (2005). Sensitivity analysis of ekf and iterated ekf pose estimation for position-based visual servoing. In *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, pages 755–760.

Siciliano, B. and Khatib, O. (2008). *Springer Handbook of Robotics*. Springer.

Smyth, A. and Wu, M. (2007). Multi-rate kalman filtering for the data fusion of displacement and acceleration response measurements in dynamic system monitoring. *Mechanical Systems and Signal Processing*, 21(2):706 – 723.

Soetens, P. (2006). *A Software Framework for Real-Time and Distributed Robot and Machine Control*. PhD thesis, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium. `http://www.mech.kuleuven.be/dept/resources/docs/soetens.pdf`.

Soetens, P. and Bruyninckx, H. (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*, pages 259–264.

Solanes, J. E., Armesto, L., Tornero, J., and Girbés, V. (2012a). Non-linear dual-rate controller for path following in nonholonomic mobile robots. In *International Conference on Systems and Robotics (ISR), 2010 IEEE/RSJ International Conference on.*

Solanes, J. E., Armesto, L., Tornero, J., Muñoz-Benavent, P., and Girbes, V. (2012b). Dual-rate non-linear high order holds for visual servoing applications. In *Towards Autonomous Robotic Systems (TAROS).*

Solanes, J. E., Muñoz Benavent, P., Girbés, V., Armesto, L., and Tornero, J. (2015). On improving robot image-based visual servoing based on dual-rate reference filtering control strategy. *Robotica*, FirstView:1–18.

Solanes, J. E., Tornero, J., Armesto, L., and Girbés, V. (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), pages 195 – 206.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot modeling and control.* John Wiley & Sons, Hoboken (N.J.).

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

Sun, F., Sun, Z., Zhang, R., and Zhang, F. (1996). A multirate adaptive composite controller for flexible-link robots using neural networks. *IEEE International Conf. on Systems, Man and Cybernetics*, 2:1036–1041.

Taylor, G. and Kleeman, L. (2004). Hybrid position-based visual servoing with online calibration for a humanoid robot. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 686–691 vol.1.

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*, pages 4286–4292.

Toibero, J., Soria, C., Roberti, F., Carelliz, R., and Fiorini, P. (2009). Switching visual servoing approach for stable corridor navigation. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6.

Tornero, J., Albertos P., and Salt J. (1999a). Periodic optimal control of multirate sampled data systems. In *14th World Congress of IFAC*, pages 211–216, China.

Tornero, J., Gu, Y., and Tomizuka, M. (1999b). Analysis of multi-rate discrete equivalent of continuous controller. In *American Control Conference*, pages 2759–2763.

Tornero, J., Piza, R., Albertos, P., and Salt, J. (2001a). Multirate LQG controller applied to self-location and path tracking in mobile robots. In *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 625–630.

Tornero, J. and Tomizuka, M. (2000). Dual-rate high order hold equivalent controllers. In *American Control Conference*, pages 175–179.

Tornero, J. and Tomizuka, M. (2002a). Modeling, analysis and design tools for dual-rate systems. In *American Control Conference*, pages 4116–4121.

Tornero, J. and Tomizuka, M. (2002b). Modeling, analysis and design tools for dual-rate systems. In *American Control Conference, 2002. Proceedings of the 2002*, volume 5, pages 4116–4121 vol.5.

Tornero, J., Tomizuka, M., Camina, C., Ballester, E., and Piza, R. (2001b). Design of dual-rate pid controllers. In *Proceedings of the IEEE International Conference on Control Applications*, pages 859–865. IFAC Press.

Usai, A. and Di Giamberardino, P. (2006). Nonholonomic mobile robot visual servoing: A multirate digital approach. In *Proceedings of the 8th WSEAS International Conference on Automatic Control, Modeling &#38; Simulation*, ACMOS'06, pages 134–139, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).

Wang, J. and Wilson, W. J. (1992). 3d relative position and orientation estimation using kalman filter for robot control. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2638–2645 vol.3.

Wang, Y., Lang, H., and de Silva, C. (2010). A hybrid visual servo controller for robust grasping by wheeled mobile robots. *Mechatronics, IEEE/ASME Transactions on*, 15(5):757–769.

Weiss, L., Sanderson, A., and Neuman, C. P. (1987). Dynamic sensor-based control of robots with visual feedback. *Robotics and Automation, IEEE Journal of*, 3(5):404–417.

Wilson, W. J., Williams Hulls, C., and Bell, G. (1996). Relative end-effector control using cartesian position based visual servoing. *Robotics and Automation, IEEE Transactions on*, 12(5):684–696.

Xie, W.-F., Li, Z., Tu, X.-W., and Perron, C. (2009). Switching control of image-based visual servoing with laser pointer in robotic manufacturing systems. *Industrial Electronics, IEEE Transactions on*, 56(2):520–529.

Yuste, H., Armesto, L., and Tornero, J. (2010). Benchmark tools for evaluating agvs at industrial environments. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2657–2662.

Zanne, P., Morel, G., and Piestan, F. (2000). Robust vision based 3d trajectory tracking using sliding mode control. In *Robotics and Automation,* *2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 3, pages 2088–2093 vol.3.

Zhang, H., Gong, J., Jiang, Y., Xiong, G., and Chen, H. (2012). An iterative linear quadratic regulator based trajectory tracking controller for wheeled mobile robot. *Journal of Zhejiang University - Science C*, 13(8):593–600.

Zhao, Y.-M., Xie, W.-F., Liu, S., and Wang, T. (2015). Neural network-based image moments for robotic visual servoing. *Journal of Intelligent & Robotic Systems*, 78(2):239–256.

Zima, M., Armesto, L., Girbes, V., Sala, A., and Smidl, V. (2013). Extended rauch-tung-striebel controller. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 2900–2905.

–Pippin: *“I didn’t think it would end this way.”*
–Gandalf: *“End? No, the journey doesn’t end here.*
Death is just another path, one that we all must take.
The grey rain-curtain of this world rolls back, and
all turns to silver glass, and then you see it.”
–Pippin: *“What? Gandalf? See what?”*
–Gandalf: *“White shores, and beyond, a*
far green country under a swift sunrise.”
–Pippin: *“Well, that isn’t so bad.”*
–Gandalf: *“No. No, it isn’t.”*

The Lord of the Rings
J.R.R. Tolkien