



Visual Servoing Multifrecuencia

Tesina Máster en Automática e Informática Industrial

Departamento de Ingeniería de Sistemas y Automática (DISA)

Departamento de Ingeniería de Sistemas y Computadores (DISCA)



**Ingeniería de Sistemas
y Automática**



Autor: D. Juan Ernesto Solanes Galbis

Director: Dr. Josep Tornero Montserrat

Codirector: Dr. Leopoldo Armesto Ángel

Diciembre 2010, Valencia

Resumen

El estudio realizado en la presente tesina de Máster trata sobre las técnicas clásicas del Visual Servoing en robots industriales, aplicando una de las técnicas de control multifrecuencia para la mejora de su comportamiento en el seguimiento de objetos en movimiento. Concretamente, se centra en temas como:

- **Estudio de las técnicas clásicas del Visual Servoing:** se realiza un estudio del estado del arte existente en el ámbito del Visual Servoing. Además, se analizan los pros y los contras de cada una de las técnicas de control clásicas en robots manipuladores.
- **Modelado y Análisis de Sistemas Multifrecuencia:** con objeto de mejorar el comportamiento de los sistemas clásicos en el seguimiento de objetos en movimiento, se estudian las principales técnicas de control multifrecuencia (*lifting*, *operadores de Kranc*, etc.). Se realiza un estudio detallado para un sistema lineal (ver Anexo 1 adjunto a la memoria)
- **Retenedores Multifrecuencia:** análisis de una metodología de control multifrecuencia que permite la implementación de la multifrecuencia sin necesidad de obtener un modelo teórico del mismo. Dentro de la gran variedad de interfaces que se han estudiado, se emplean los *DR-IH*, *DR-ZOH*, *DR-FOH* y *DR-SOH*.
- **Calibración de cámaras en manipuladores:** se estudian distintas técnicas de calibración *eye-in-hand*, así como de calibración de cámaras en general. Se utilizan herramientas tales como el *Calibration Toolbox* o la herramienta de calibración de la librería *OpenCV*.
- **Visión por computador:** Se realizan distintos algoritmos para la detección de objetos tales como una pelota, de los vértices de un cuadrado, etc. Así mismo, se utilizan algoritmos dados en la librería *Visual Servoing Platform* [81] para la detección de un punto.
- **Control del Kuka KR5 sixx R650:** Todo el estudio teórico bajo simulación es verificado en una aplicación real, con un robot industrial del comerciante Kuka montado en una plataforma del Instituto de Diseño y Fabricación. Toda la puesta a punto (*setup*) del robot, comunicaciones con sistemas externos, etc., se ha realizado por primera vez para el desarrollo experimental de esta tesina.

Las problemáticas a resolver mediante la combinación de las técnicas Visual Servoing y del diseño multifrecuencia son muy variadas. No obstante, se ha centrado la tesina en el problema relacionado con el tracking de objetos moviéndose a gran velocidad por medio de robots industriales.

Vimos en esta temática numerosos procesos industriales, en que la resolución de este problema provocaría un aumento en los tiempos de producción, lo que en definitiva equivale a un aumento de los beneficios.

En el ámbito de control multifrecuencia se han estudiado las distintas técnicas, centrándonos en los retenedores multifrecuencia basados en funciones primitivas, habiéndose implementado los de tipo ZOH, FOH, TH y SOH.

La elección de esta técnica frente a otras basadas en modelos de los procesos se justifica por las altas no linealidades asociados a los procesos involucrados. Estas no linealidades derivan del uso de brazos robots con articulaciones de revolución, pero también de los modelos y algorítmica asociada a los procesos de adquisición y procesamiento de imágenes.

En la parte del *Visual Servoing* se ha analizado e implementado los tres métodos tradicionales presentes en la literatura, con objeto de comparar las distintas técnicas, al objeto de seleccionar la más adecuada a la problemática planteada.

En concreto, se ha elegido la técnica Image-Based Visual Servoing (IBVS) por ser la más empleada tanto en la literatura como en los procesos reales. El motivo es que al realizarse el control en el espacio de trabajo del plano de imagen no existen problemas derivados a una mala calibración del sistema.

Finalmente, destacar que a lo largo de la tesina, se han desarrollado distintas aplicaciones, tanto para el estudio bajo simulación, como el estudio en un sistema real, que corroboran los resultados obtenidos.

Notación

Escalares

u : coordenada x del plano de imagen

v : coordenada y del plano de imagen

ϕ : velocidad angular eje x

ψ : velocidad angular eje y

γ : velocidad angular eje z

T : periodo de muestreo a frecuencia alta (periodo base)

\bar{T} : periodo de muestreo a frecuencia baja (metaperiodo)

k : instante de muestreo en periodo T

\bar{k} : instante de muestreo en periodo \bar{T}

λ : ganancia del controlador IBVS

f : distancia focal

d : distancia al plano del objeto

N : orden de multiplicidad

c_u : coordenada en x del punto principal

c_v : coordenada en y del punto principal

α : ratio de dimensión de los píxeles

β : parámetro intrínseco

Z : distancia del plano de la cámara a la característica del objeto

t_j^i : instantes de muestreo

Vectores

x : representación del punto en coordenadas cartesianas

s : características actuales del objeto

e : error entre la pose deseada y la pose medida

u : acciones de control

t : vector de translación

ω_i : velocidad angular

V_i : velocidad de translación

s^* : característica deseada del objeto

\dot{q} : velocidad en espacio de articulaciones

Ω_q : velocidad angular en espacio de articulaciones

V_q : velocidad de translación en espacio de articulaciones

n : plano normal

θ : ángulo de rotación

y : salida del sistema en ecuación de diferencias

E^- : vector retardo

E^+ : vector adelanto

u_h : salida de los Dual-Rate High Order Holds (DR-HOH)

y_s : salida de los Dual-Rate High Order Samples (DR-HOS)

p : punto en coordenadas cartesianas

V_c : acción de control del esquema IBVS

Matrices

R : matriz de rotación homogénea

X : matriz homogénea en coordenadas cartesianas

J : matriz Jacobiana del sistema robotizado

A : parámetros intrínsecos del objeto

Λ : operador retardo

W : operador lifting

L_s : matriz de interacción

\hat{L}_s^+ : pseudo-inversa de la matriz de interacción

Funciones de Transferencia

G_f : sistema N-periódico lineal y discreto

G_g : sistema N-periódico lineal y discreto

G_{fg} : sistema LTI

G_{cl} : matriz de transferencia en bucle cerrado

G : sistema N-periódico lineal y discreto

G_p : matriz de transferencia de la planta

G_R : matriz de transferencia del regulador

G_C : matriz de transferencia del controlador

$G_{open-loop}$: matriz de transferencia en bucle abierto

Índice general

CAPÍTULO 1. Intoducción	22
CAPÍTULO 2. Estado del arte	30
2.1. Visual Servoing	31
2.1.1. Visión general del Visual Servoing	31
2.1.2. Conceptos generales.....	32
2.1.3 Definiciones y notaciones	35
2.1.4. Clasificación de los sistemas conocido el modelo.....	37
2.1.4.1 Control basado en la posición (PBVS)	37
2.1.4.2. Control basado en la imagen (IBVS)	39
2.1.4.3. 2 ½ D Visual Servoing (HBVS)	43
2.2. Sistemas Multifrecuencia	45
2.2.1. Modelado de sistemas muestreados multifrecuencia	46
2.2.1.1. Modelos en representación externa.....	46
2.2.1.1.1. Descomposición en frecuencia	46
2.2.1.1.2. Descomposición vectorial	46
2.2.1.1.3. Operadores Skip-Expand.....	47
2.2.1.2. Modelos en representación interna	49
2.2.1.2.1. Lifting discreto.....	49
2.2.1.2.2. Modelado de Thompson	51
2.2.1.2.3. Bloque multifrecuencia Input/Output (BMIO).....	53
2.2.1.2.4. Modelado Peródico	53
2.2.2. High-Order-Holds y High-Order-Samplers multifrecuencia.....	54
2.2.2.1. Retenedor Bi-frecuencia.....	55
2.2.2.2. Muestreador Bi-frecuencia	60
2.2.3. Modelo de la planta Bi-frecuencia	61
2.2.4. Controladores equivalentes de orden alto Bi-frecuencia	64
2.2.4.1. Estimación de la señal de control	66
2.2.4.2. Estimación de la señal de error	67
2.2.5. Lugar de las raíces Multifrecuencia.....	69
CAPÍTULO 3. Image-Based Visual Servoing	74
3.1. Conceptos y formulaciones	75
3.1.1. La matriz de Interacción.....	78
3.1.2. Aproximaciones de la matriz de Interacción.....	80
3.1.3. Interpretación geométrica del IBVS.....	80
3.2. Estudio del comportamiento del sistema mediante simulación	82
3.2.1. Robot Kuka KR5 sixx R650	83
3.2.2. Calibración de la Cámara Logitech C300	87
3.2.2.1. Procedimiento de calibración	88
3.2.2.1.1. Parámetros intrínsecos	92
3.2.2.1.2. Parámetros extrínsecos	92
3.2.3. Simulador IBVS Kuka KR5 sixx R650	94
3.2.3.1. Comportamiento del sistema.....	96
3.2.3.1.1. Objeto Estático.....	96
3.2.3.1.2. Objeto dinámico	102

3.2.3.1.3. Seguimiento de una pelota	103
3.2.3.1.4. Estudio con la pelota en movimiento oscilatorio	104
3.3. Estudio del comportamiento del sistema en implementación real.	112
3.3.1. Sistema Visual Servoing.....	112
3.3.1.1. Configuración Hardware del sistema Visual Servoing.....	114
3.3.1.1.1. Comunicación mediante la tarjeta RSI-Ethernet.....	115
3.3.1.1.1.1. Principio de funcionamiento	116
3.3.1.1.1.2. Objeto RSI ST_ETHERNET	118
3.3.1.1.1.2.1. Creación de un objeto ST_ETHERNET	119
3.3.1.1.1.2.2. Configuración del ST_ETHERNET	120
3.3.1.1.1.2.3. Definición archivo de configuración.....	121
3.3.1.1.1.2.4. Parámetros de comunicación	121
3.3.1.1.1.2.5. Objetos de entrada del ST_ETHERNET	123
3.3.1.1.1.2.6. Objetos de salida del ST_ETHERNET	124
3.3.1.2. Configuración Software del Sistema Visual Servoing.....	125
3.3.2. Análisis del comportamiento del IBVS	126
3.3.2.1. Objeto estático	126
3.3.2.2. Objeto dinámico	129
3.4. Conclusiones	136
CAPÍTULO 4. Visual Servoing Multifrecuencia	137
4.1. Sistema muestreado multifrecuencia	139
4.2. Control de dos grados de libertad con un PID y interfaces multifrecuencia	139
4.2.1. Sistema muestreado multifrecuencia	140
4.2.1.1. Pruebas realizadas con estimación de la señal de error de forma síncrona.	144
4.2.1.2. Pruebas realizadas con estimación de la señal de control de forma síncrona.	151
4.2.1.3. Comparativa entre ambos enfoques	157
4.2.2. Sistemas Multifrecuenciales asíncronos.	164
4.2.2.1. Comparativa entre el enfoque multifrecuencia síncrono y el asíncrono.	166
4.2.3. Conclusiones	172
4.3. Sistema Image-Based Visual Servoing Multifrecuencia	173
4.3.1. Análisis del comportamiento bajo la plataforma de simulación	173
4.3.1.1. Comparativa entre los enfoques	184
4.3.2. Análisis del comportamiento bajo la plataforma real	190
4.3.3. Comparativa con IBVS clásico.....	206
4.4. Conclusiones	208
CAPÍTULO 5. Conclusiones y trabajos futuros.....	216
ANEXOS.....	225
ANEXO 1. Ejemplo Lineal	228
ANEXO 2. LENGUAJE RSI-XML	266
ANEXO 3. Archivo XML para comunicación con ST_ETHERNET	304
ANEXO 4. PROGRAMA DEL CONTROLADOR DEL ROBOT	310
Bibliografía	217

Índice de figuras y tablas

Figura 1. Sistema hardware de visión artificial para el control de calidad. 24

Figura 2. Software de visión para control de calidad en la industria del automóvil..... 24

Figura 3. Aplicaciones industriales del *Visual Servoing* ABB Flex-Pick de tortitas..... 24

Figura 4. Aplicaciones industriales del *Visual Servoing*, ABB Flex-Pick de salchichas..... 24

Figura 5. Izquierda: Control del robot en lazo abierto. La idea es detectar el objeto y a partir de la pose detectada calcular la secuencia de acciones de control que permitan al robot llegar a la posición deseada. Derecha: Control en lazo cerrado. La idea es, una vez detectado el objeto el control siempre va a tener en cuenta en cada iteración el estado del objeto con respecto al robot..... 33

Figura 6. Procesos envueltos en un sistema *Visual Servoing*. El movimiento euclideo relativo del robot viene determinado por las acciones del controlador en el espacio de trabajo del robot T. “x” representa el vector de estado, s es el vector de características medidas y s* es el vector de características deseadas que son medidas en el espacio del sensor S. 35

Figura 7. Clasificación según el Modelo de Estimación Visual-Motor. Los sistemas donde el modelo es conocido a-priori utilizan el modelo cinemático del robot, los parámetros de la cámara con diferentes niveles de calibración entre la cámara y el robot para estimar la posición deseada del mismo. Por otro lado, hay sistemas que estiman el modelo Visual-Motor ya sea por técnicas de “*Machine Learning*” o analíticamente, permitiendo el control sin conocimiento de la geometría del robot..... 36

Figura 8. Configuraciones Cámara-Robot utilizadas en el *Visual Servoing* (de izquierda a derecha): Configuración 1 *monocular eye-in-hand*, Configuración 2 *stereo eye-in-hand*, Configuraciones 3 y 4 *monocular y estéreo stand-alone* y Configuración 5 sistema de cámaras redundantes. 37

Figura 9. Dos ejemplos de PBVS: en la imagen de la izquierda, ejemplo de configuración de la cámara eye-in-hand donde la cámara-robot pasa de la configuración ${}^c X_o$ (posición actual) a la ${}^c X_o^*$ posición deseada; en la derecha, una cámara en configuración monocular stand-alone. 38

Figura 10. Diagrama de bloques del enfoque PBVS: la pose del objeto se estima, ${}^c X_o$, y se compara con la referencia (pose deseada), ${}^c X_o^*$. El error se utiliza luego para calcular el screw vector, $\dot{q} = [v \ \omega]^T$ de esta forma el robot minimiza el error. 39

Figura 11. Ejemplo de IBVS. Se asume el caso de una cámara estática y un objeto móvil. Un número de características del objeto se utiliza para generar el vector s. El vector de características deseadas se denota como s*. El error se define como la diferencia entre ambos vectores, $e(s) = s - s^*$. Esta función error se actualiza en cada bucle de control y se utiliza junto a la matriz de interacción para estimar la acción de control del robot..... 41

Figura 12. Una cámara observando un plano desde dos posiciones: hay una transformación lineal entre las coordenadas de un punto en dos imágenes de una cámara. Esta transformación es una matriz 3x3 de 8 grados de libertad, M , que debe estimarse utilizando cuatro o más correspondencias del punto. La homografía se puede escribir en términos de los parámetros internos de la cámara, A , el desplazamiento entre los puntos de vista, $x(R, t)$ y la ecuación del plano (n, d) dada por $f = Mf^* = A(R + tn^T/d)A^{-1}f^*$. Si los parámetros internos, A , son conocidos, la rotación R y el plano normal n pueden obtenerse a partir de la homografía. La translación t y la distancia al plano d pueden obtenerse solamente mediante un escalado [54]. 43

Figura 13. Diagrama de bloques del enfoque 2 ½ D Visual Servoing propuesto por [40]. La tarea de posicionamiento en este caso puede describirse de la siguiente forma: $e = \begin{bmatrix} s - s^* & u^T \theta \end{bmatrix}^T$ donde u y θ representa el eje de rotación y el ángulo de rotación obtenidos de la matriz R	44
Figura 14. Descomposición frecuencial.....	46
Figura 15. Descomposición Vectorial.....	47
Figura 16. Operador Expand.....	48
Figura 17. Operador Skip.....	48
Figura 18. Notación compacta de la descomposición vectorial de conexiones.....	52
Figura 19. Operador de Kranc.....	52
Figura 20. Esquema general del sistema multifrecuencia.....	54
Figura 21. Retenedor de Orden Alto Bi-frecuencia (DR-HOH).....	56
Figura 22. Retenedor bifrecuencia de orden cero (DR-ZOH).....	57
Figura 23. Retenedor bifrecuencia de primer orden (DR-FOH).....	58
Figura 24. Retenedor bifrecuencia de primer orden (DR-SOH).....	59
Figura 25. Representaciones de los DR-HOH (ejemplo con el DR-FOH).....	59
Figura 26. Muestreador de Orden Alto Bi-frecuencia (DR-HOS).....	60
Figura 27. Controlador general bifrecuencia.....	65
Figura 28. Lugar de las raíces continuo: pintada en blanco, la zona estable (izquierda del eje 0); en amarillo, la zona inestable (derecha del eje 0).....	69
Figura 29. Lugar de las raíces en el dominio discreto: en blanco, zona estable (dentro del círculo unidad); en amarillo, zona inestable (fuera del círculo unidad).....	70
Figura 30. Lugar de las raíces multifrecuencia N=1 con FOH.....	71
Figura 31. Lugar de las raíces multifrecuencia N=2 con FOH.....	71
Figura 32. Lugar de las raíces multifrecuencia N=3 con FOH.....	71
Figura 33. Lugar de las raíces multifrecuencia N→∞ con FOH.....	71
Figura 34. Variación del Lugar de las Raíces en función de N.....	72
Figura 35. Lugar de las Raíces del sistema empleando distintos retenedores de orden alto bi-frecuencia.....	72
Figura 36. Esquema de control Directo del sistema Visual Servoing.....	75
Figura 37. Esquema de control Indirecto del sistema Visual Servoing.....	75
Figura 38. Esquema de control Indirecto para el Image-Based Visual Servoing.....	76

Figura 39. Esquema de control Directo para el Image-Based Visual Servoing.....	76
Figura 40. Interpretación geométrica del IBVS: el movimiento va de los puntos marrones a los puntos rojos. En verde, el movimiento de los puntos en el plano de imagen cuando se trabaja con L_s^+ ; en azul cuando se trabaja con $L_{s^*}^+$; en negro, el movimiento de los puntos cuando se trabaja con $(L_s/2 + L_{s^*}/2)^+$	81
Figura 41. Robot manipulador industrial Kuka KR5 sixx R650.....	83
Figura 42. Características geométricas del Kuka KR5 sixx R650.....	84
Figura 43. Sistemas de Referencia de cada articulación del Kuka KR5 sixx R650 con el convenio clásico D-H... ..	86
Figura 44. Simulador Kuka KR5 sixx R650 con Matlab/ Simulink.....	87
Figura 45. Cámara Webcam Logitech C300.....	88
Figura 46. Montaje de calibración. Webcam Logitech C300 y plantilla de calibración.....	89
Figura 47. Imágenes tomadas para la calibración de la cámara.....	89
Figura 48. Imagen preparada para la detección del área donde se encuentra el patrón de calibración.....	90
Figura 49. Selección manual del patrón a analizar.....	90
Figura 50. Extracción de las esquinas en el patrón elegido. A la izquierda la extracción realizada; a la derecha el error cometido en la estimación.....	91
Figura 51. Resultado de la detección de las esquinas en el patrón.....	91
Figura 52. Parámetros extrínsecos de la cámara fija respecto a la escena.....	93
Figura 53. Parámetros extrínsecos manteniendo la escena fija y la cámara móvil.....	93
Figura 54. Diagrama Simulink del Simulador IBVS para el Kuka KR5 sixx R650.....	94
Figura 55. Diagrama Simulink del interior del bloque <i>Simulated point pose</i>	95
Figura 56. Representación en el plano de imagen del objeto utilizado para las simulaciones de objeto fijo: izquierda posición inicial del objeto; derecha posición deseada del objeto.....	96
Figura 57. Trayectoria de los puntos desde la posición inicial a la final con el enfoque de $\hat{L}_s^+ = L_s^+$	97
Figura 58. Trayectoria seguida por la cámara en el mundo cartesiano. Se comprueba que la realización de una trayectoria en línea recta en el plano de imagen se transforma en una trayectoria curva en el plano cartesiano.....	98
Figura 59. Error en función del tiempo de las características del objeto.....	98
Figura 60. Trayectoria seguida en el plano de imagen desde la configuración inicial con $\hat{L}_i^+ = L_{i^*}^+$: puntos en negro, hasta la configuración final, puntos en rojo.....	99
Figura 61. Trayectoria seguida por la cámara en el espacio cartesiano durante la tarea de <i>matching</i> con $\hat{L}_i^+ = L_{i^*}^+$	100

Figura 62. Error de convergencia de las características del objeto en función del tiempo con $\hat{L}_s = L_s^*$	100
Figura 63. Trayectoria seguida en el plano de imagen con $\hat{L}_s^* = \frac{1}{2}(L_s + L_{s^*})$	101
Figura 64. Error de convergencia en función del tiempo de las características del objeto par el caso $\hat{L}_s = \frac{1}{2}(L_s + L_{s^*})$	102
Figura 65. Posición inicial y deseada del centro de masas de la pelota: la imagen de la izquierda muestra la posición inicial y la imagen de la derecha muestra la posición deseada del centro de masas de la pelota.	103
Figura 66. Trayectoria seguida desde la posición inicial a la deseada en el plano de imagen.	103
Figura 67. Trayectoria seguida por la cámara en el espacio cartesiano para el seguimiento de la pelota.	104
Figura 68. Pelota moviéndose horizontalmente.	105
Figura 69. Trayectoria de la pelota en función del tiempo desde el punto de vista del plano de imagen.	106
Figura 70. Seguimiento de la pelota en movimiento en función del tiempo.	106
Figura 71. Trayectoria de la cámara en el plano cartesiano.	107
Figura 72. Secuencia de imágenes del robot siguiendo la pelota.	107
Figura 73. Trayectoria de las articulaciones del robot en función del tiempo.	108
Figura 74. Pelota moviéndose verticalmente.	109
Figura 75. Trayectoria de cada una de las componentes del centro de masas de la pelota a lo largo de la simulación.	109
Figura 76. Seguimiento de la pelota moviéndose verticalmente.	110
Figura 77. Trayectoria seguida por la cámara en el espacio cartesiano.	110
Figura 78. Secuencia de imágenes sobre el comportamiento del robot para el seguimiento de la pelota en el caso de movimiento vertical.	111
Figura 79. Trayectoria de cada articulación del robot a lo largo del experimento de seguimiento de una pelota con movimiento vertical.	111
Figura 80. Robot Kuka KR5 sixx R650 para aplicación de Visual Servoing.	113
Figura 81. Portátil DELL XPS para aplicación de Visual Servoing.	113
Figura 82. Sistema Visual Servoing para experimentación: la mayor parte de los experimentos se realizarán utilizando un proyector para poder tener una referencia real del objeto y poder luego estudiar los resultados obtenidos.	114
Figura 83. Configuración hardware del sistema Visual Servoing.	114
Figura 84. Unidad de control KC 2 del Kuka KR5 sixx R650.	115

Figura 85. Principio de funcionamiento del intercambio de datos.	116
Figura 86. Secuencia de intercambio de datos.	117
Figura 87. Estructura XML cuando se envían datos con el objeto ST_COROB.	118
Figura 88. Estructura XML cuando se reciben datos con el objeto ST_COROB.	118
Figura 89. Objeto RSI ST_ETHERNET.	119
Figura 90. Ejemplo de código a realizar en el programa KRL.	120
Figura 92. Ejemplo de modificación de los parámetros del objeto ST_ETHERNET.	121
Figura 92. Estructura del fichero XML.	121
Figura 93. Estructura del fichero XML con la configuración de los parámetros de comunicación.	122
Figura 94. Estructura XML con la configuración de parámetros del objeto de entrada.	123
Figura 95. Estructura XML con la configuración de parámetros del objeto de salida.	125
Figura 96. Escritorio del sistema operativo Ubuntu 10.04	125
Figura 97. Entorno de trabajo Netbeans 6.8 para LINUX.	126
Figura 98. Detección del centro de masas del objeto en el plano de imagen.	127
Figura 99. Error de las características detectadas a lo largo de la ejecución del algoritmo IBVS.	127
Figura 100. Estructura de sujeción del Kuka KR5 sixx R650.	128
Figura 101. Acciones de control a lo largo del experimento para cada uno de los ejes.	128
Figura 102. Referencias utilizadas para obtener los resultados reales.	129
Figura 103. Errores de las características detectadas en el plano de imagen (trayectoria Triángulo con periodo 30 segundos).....	130
Figura 104. Acciones de control para cada una de las articulaciones (trayectoria Triángulo con periodo 30 segundos).....	131
Figura 105. Errores de las características detectadas en el plano de imagen (Trayectoria triángulo con periodo 10 segundos).....	131
Figura 106. Trayectorias seguidas por el efector final del robot al seguir la trayectoria del objeto. En azul el objeto siguiendo una trayectoria triangular con 30 segundos de periodo. En verde el objeto siguiendo la misma trayectoria pero con el objeto moviéndose a 10 segundos.	132
Figura 107. Acciones de control aplicadas para el seguimiento de la trayectoria Triangular con periodo de 10 segundos.	132
Figura 108. Errores de las características detectadas en el plano de imagen (trayectoria Triángulo con periodo 5 segundos).	133
Figura 109. Trayectorias seguidas por el efector final en función de la velocidad del objeto: azul con periodo de 30 segundos, en verde con periodo de 10 segundos y en rojo con periodo de 5 segundos.	134

Figura 110. Acciones de control aplicadas para el seguimiento de la trayectoria Triangular con periodo de 5 segundos.....	134
Figura 111. Comparativa del comportamiento del sistema frente a cambios de la trayectoria del objeto y de su velocidad.....	135
Figura 112. Comparativa del comportamiento del sistema frente a cambios de la trayectoria del objeto y de su velocidad (Error Cuadrático).....	136
Figura 113. Diagrama de bloques multifrecuencia del sistema Visual Servoing.....	140
Figura 114. Diagrama de bloques multifrecuencia del sistema Visual Servoing simplificado.....	141
Figura 115. Máxima simplificación del sistema muestreado multifrecuencia.....	141
Figura 116. Metaperiodo del sistema de Visión (Adquisición más procesado de imagen).....	142
Figura 117. Diagramas de bloques de los enfoques de estimación de la señal de error (Derecha) y estimación de la señal de control (Izquierda).....	142
Figura 118. Diagrama de flujo de las interfaces DR-HOH.....	143
Figura 119. Estimación de la señal de error.....	144
Figura 120. Comparativa para trayectoria triángulo.....	145
Figura 121. Error Cuadrático para trayectoria triángulo.....	146
Figura 122. Comparativa para trayectoria triángulo.....	147
Figura 123. Error Cuadrático para trayectoria triángulo.....	148
Figura 124. Comparativa para trayectoria círculo.....	149
Figura 125. Error Cuadrático para trayectoria círculo.....	149
Figura 126. Comparativa para trayectoria círculo.....	150
Figura 127. Comparativa para trayectoria círculo.....	151
Figura 128. Diagrama de bloques del enfoque de estimación de la señal de control.....	151
Figura 129. Comparativa del error utilizando el enfoque de estimación de la señal de control para la trayectoria triángulo.....	152
Figura 130. Comparativa del error cuadrático utilizando el enfoque de estimación de la señal de control para la trayectoria triángulo.....	153
Figura 131. Comparativa del error utilizando el enfoque de estimación de la señal de control para la trayectoria cuadrado.....	154
Figura 132. Comparativa del error cuadrático utilizando el enfoque de estimación de la señal de control para la trayectoria cuadrado.....	154
Figura 133. Comparativa del error utilizando el enfoque de estimación de la señal de control para la trayectoria círculo.....	155

Figura 134. Comparativa del error cuadrático utilizando el enfoque de estimación de la señal de control para la trayectoria círculo.....	155
Figura 135. Comparativa del error utilizando el enfoque de estimación de la señal de control para la trayectoria infinito.....	156
Figura 136. Comparativa del error cuadrático utilizando el enfoque de estimación de la señal de control para la trayectoria infinito.....	157
Figura 137. Diagrama de bloques genérico multifrecuencia.....	157
Figura 138. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria triángulo empleando como interfaz un DR-ZOH.....	158
Figura 139. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria triángulo empleando como interfaz un DR-FOH.....	158
Figura 140. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria triángulo empleando como interfaz un DR-SOH.....	159
Figura 141. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria cuadrado empleando como interfaz un DR-ZOH.....	159
Figura 142. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria cuadrado empleando como interfaz un DR-FOH.....	160
Figura 143. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria cuadrado empleando como interfaz un DR-SOH.....	160
Figura 144. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria círculo empleando como interfaz un DR-ZOH.....	161
Figura 145. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria círculo empleando como interfaz un DR-FOH.....	161
Figura 146. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria círculo empleando como interfaz un DR-SOH.....	162
Figura 147. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria infinito empleando como interfaz un DR-ZOH.....	162
Figura 148. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria infinito empleando como interfaz un DR-FOH.....	163
Figura 149. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria infinito empleando como interfaz un DR-SOH.....	163
Figura 150. Ejecución de las tareas en un sistema real.....	164
Figura 151. Enfoque multifrecuencia asíncrono.....	165
Figura 152. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error cuadrático para la trayectoria triángulo con DR-ZOH.....	166
Figura 153. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error cuadrático para la trayectoria triángulo con DR-FOH.....	167

Figura 154. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error cuadrático para la trayectoria triángulo con DR-SOH.....	167
Figura 155. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria cuadrado con DR-ZOH.....	168
Figura 156. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria cuadrado con DR-FOH.....	168
Figura 157. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria cuadrado con DR-SOH.....	169
Figura 158. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria círculo con DR-ZOH.....	169
Figura 159. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria círculo con DR-FOH.....	170
Figura 160. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria círculo con DR-SOH.....	170
Figura 161. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria infinito con DR-ZOH.....	171
Figura 162. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria infinito con DR-FOH.....	171
Figura 163. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria infinito con DR-SOH.....	172
Figura 164. Objeto visto en el plano de imagen: pose actual del objeto (imagen de la izquierda), pose deseada del objeto (imagen de la derecha).....	173
Figura 165. Esquema multifrecuencia del sistema Visual Servoing.....	174
Figura 166. Comportamiento del sistema en función del metaperiodo del elemento más lento del sistema (la visión).....	174
Figura 167. Esquema de control bi-frecuencia del sistema Visual Servoing.....	175
Figura 168. Seguimiento del objeto utilizando DR-ZOH. En las figuras se muestra el punto 3 en el plano de imagen.....	175
Figura 169. Comportamiento del sistema con un DR-FOH.....	176
Figura 170. Esquema multifrecuencia del sistema Visual Servoing.....	176
Figura 171. Seguimiento del un objeto cambiando los valores de N_1 and N_2 utilizando DR-ZOH. En las figuras se muestra el eje X del punto 3 en el plano de imagen.....	177
Figura 172. Acciones de control del eje A3 cambiando los valores de N_1 and N_2 utilizando DR-ZOH.....	177
Figura 173. Seguimiento del un objeto cambiando los valores de N_1 and N_2 utilizando DR-FOH. En las figuras se muestra el eje X del punto 3 en el plano de imagen.....	178
Figura 174. Acciones de control del eje A3 cambiando los valores de N_1 and N_2 utilizando DR-FOH.....	178

Índice de figuras

Figura 175. Trayectoria en el eje X de los puntos en el plano de imagen a lo largo del tiempo: P1 rojo, P2 azul, P3 verde y P4 negro.	179
Figura 176. Trayectoria en el eje Y de los puntos en el plano de imagen a lo largo del tiempo: P1 rojo, P2 azul, P3 verde y P4 negro.	179
Figura 177. Zoom de las trayectorias en el eje X de los puntos 2 y 3.	179
Figura 178. Zoom de las trayectorias en el eje Y de los puntos 2 y 3.	179
Figura 179. Acción de control para la articulación 1.....	180
Figura 180. Zoom de la acción de control para la articulación 1.....	180
Figura 181. Acción de control para la articulación 2.....	180
Figura 182. Zoom de la acción de control para la articulación 2.....	180
Figura 183. Acción de control para la articulación 3.....	180
Figura 184. Zoom de la acción de control para la articulación 3.....	180
Figura 185. Acción de control para la articulación 4.....	181
Figura 186. Zoom de la acción de control para la articulación 4.....	181
Figura 187. Acción de control para la articulación 5.....	181
Figura 188. Zoom de la acción de control para la articulación 5.....	181
Figura 189. Acción de control para la articulación 6.....	181
Figura 190. Zoom de la acción de control para la articulación 6.....	181
Figura 191. Trayectoria en el eje X de los puntos en el plano de imagen a lo largo del tiempo: P1 rojo, P2 azul, P3 verde y P4 negro.	182
Figura 192. Zoom de la trayectoria en el eje X de los puntos en el plano de imagen a lo largo del tiempo: P1 rojo, P2 azul, P3 verde y P4 negro.	182
Figura 193. Acción de control para la articulación 1.....	183
Figura 194. Zoom de la acción de control para la articulación 1.....	183
Figura 195. Acción de control para la articulación 2.....	183
Figura 196. Zoom de la acción de control para la articulación 2.....	183
Figura 197. Acción de control para la articulación 3.....	183
Figura 198. Zoom de la acción de control para la articulación 3.....	183
Figura 199. Acción de control para la articulación 4.....	184
Figura 200. Zoom de la acción de control para la articulación 4.....	184
Figura 201. Acción de control para la articulación 5.....	184

Figura 202. Zoom de la acción de control para la articulación 5.....	184
Figura 203. Acción de control para la articulación 6.....	184
Figura 204. Zoom de la acción de control para la articulación 6.....	184
Figura 205. Trayectoria del punto 1 en el eje X del plano de imagen	185
Figura 206. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen	185
Figura 207. Trayectoria del punto 2 en el eje X del plano de imagen	185
Figura 208. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen	185
Figura 209. Trayectoria del punto 3 en el eje X del plano de imagen	185
Figura 210. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen	185
Figura 211. Trayectoria del punto 4 en el eje X del plano de imagen	186
Figura 212. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen	186
Figura 213. Trayectoria del punto 1 en el eje Y del plano de imagen	186
Figura 214. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen	186
Figura 215. Trayectoria del punto 2 en el eje Y del plano de imagen	186
Figura 216. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen	186
Figura 217. Trayectoria del punto 3 en el eje Y del plano de imagen	187
Figura 218. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen	187
Figura 219. Trayectoria del punto 4 en el eje Y del plano de imagen	187
Figura 220. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen	187
Figura 221. Acción de control para la articulación 1.....	188
Figura 222. Zoom de la acción de control para la articulación 1.....	188
Figura 223. Acción de control para la articulación 2.....	188
Figura 224. Zoom de la acción de control para la articulación 2.....	188
Figura 225. Acción de control para la articulación 3.....	188
Figura 226. Zoom de la acción de control para la articulación 3.....	188
Figura 227. Acción de control para la articulación 4.....	189
Figura 228. Zoom de la acción de control para la articulación 4.....	189
Figura 229. Acción de control para la articulación 5.....	189
Figura 230. Zoom de la acción de control para la articulación 5.....	189

Figura 231. Acción de control para la articulación 6.....	189
Figura 232. Zoom de la acción de control para la articulación 6.....	189
Figura 233. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 30 segundos y con DR-ZOH.....	190
Figura 234. Trayectoria efectuada por el Efecto final del Kuka KR5 sixx R650 con DR-ZOH y trayectoria triángulo con periodo 30 segundos.....	191
Figura 235. Acciones de control para la trayectoria triángulo de periodo 30 segundos con DR-ZOH.....	191
Figura 236. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 10 segundos y con DR-ZOH.....	192
Figura 237. Trayectoria efectuada por el Efecto final del Kuka KR5 sixx R650 con DR-ZOH y trayectoria triángulo con periodo 10 segundos.....	192
Figura 238. Acciones de control para la trayectoria triángulo de periodo 10 segundos con DR-ZOH.....	193
Figura 239. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 5 segundos y con DR-ZOH.....	193
Figura 240. Trayectoria efectuada por el Efecto final del Kuka KR5 sixx R650 con DR-ZOH y trayectoria triángulo con periodo 5 segundos.....	194
Figura 241. Acciones de control para la trayectoria triángulo de periodo 5 segundos con DR-ZOH.....	194
Figura 242. Comparativa entre las trayectorias efectuadas según el cambio de velocidad del objeto con trayectoria triángulo y interfaz DR-ZOH: en azul con periodo 30 segundos; en verde con periodo 10 segundos; en rojo con periodo 5 segundos.....	195
Figura 243. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 30 segundos y con DR-FOH.....	195
Figura 244. Trayectoria efectuada por el Efecto final del Kuka KR5 sixx R650 con DR-FOH y trayectoria triángulo con periodo 30 segundos.....	196
Figura 245. Acciones de control para la trayectoria triángulo de periodo 30 segundos con DR-FOH.....	196
Figura 246. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 10 segundos y con DR-FOH.....	197
Figura 247. Trayectoria efectuada por el Efecto final del Kuka KR5 sixx R650 con DR-FOH y trayectoria triángulo con periodo 10 segundos.....	197
Figura 248. Acciones de control para la trayectoria triángulo de periodo 10 segundos con DR-FOH.....	198
Figura 249. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 5 segundos y con DR-FOH.....	198
Figura 250. Trayectoria efectuada por el Efecto final del Kuka KR5 sixx R650 con DR-FOH y trayectoria triángulo con periodo 5 segundos.....	199
Figura 251. Acciones de control para la trayectoria triángulo de periodo 5 segundos con DR-FOH.....	199

Figura 252. Comparativa entre las trayectorias efectuadas según el cambio de velocidad del objeto con trayectoria triángulo y interfaz DR-ZOH: en azul con periodo 30 segundos; en verde con periodo 10 segundos; en rojo con periodo 5 segundos.	199
Figura 253. Comparación entre los dos interfaces DR-HOH utilizados en este experimento.	200
Figura 254. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-ZOH para la trayectoria cuadrado.	201
Figura 255. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-FOH para la trayectoria cuadrado.	201
Figura 256. Comparación entre los dos interfaces DR-HOH utilizados en este experimento.	202
Figura 257. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-ZOH para la trayectoria círculo.	203
Figura 258. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-FOH para la trayectoria círculo.	203
Figura 259. Comparación entre los dos interfaces DR-HOH utilizados en este experimento.	204
Figura 260. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-ZOH para la trayectoria infinito.	205
Figura 261. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-ZOH para la trayectoria infinito.	205
Figura 262. Comparación entre los dos interfaces DR-HOH utilizados en este experimento.	206
Figura 263. Comparativa entre los enfoques clásico y multifrecuencia (periodo 30 segundos).	207
Figura 264. Comparativa entre los enfoques clásico y multifrecuencia (periodo 10 segundos).	207
Figura 265. Comparativa entre los enfoques clásico y multifrecuencia (periodo 5 segundos).	208

CAPÍTULO 1

Introducción

A medida que ha avanzado la tecnología, los sensores de visión han adquirido mayor protagonismo en los procesos industriales. Es por ello que hoy en día en la industria hay una tendencia creciente en la utilización de este tipo de sensores para la realización automática de tareas. Debido a la gran cantidad de información que se puede extraer de los sensores de visión, no es de extrañar que se estén utilizando en aplicaciones tan diversas como los que se muestran en la Tabla 1.

ÁREA DE PRODUCCIÓN	APLICACIÓN	ÁREA DE PORDUCCIÓN	APLICACIÓN
CONTROL DE CALIDAD	Inspección de productos (papel, aluminio, acero, ...)	ASTRONOMÍA	Exploración del Espacio
	Identificación de piezas	RECONOCIMIENTO DE CARÁCTERES	Control de cheques, inspección de textos,...
	Etiquetado (fechas de caducidad, ...)	CONTROL DE TRÁFICO	Matrículas de coches
	Inspección de circuitos impresos		Tráfico viario
	Control de calidad de los alimentos (naranjas, carne, ...)	METEREOLOGÍA	Predicción del tiempo
ROBÓTICA	Control de soldaduras	AGRICULTURA	Interpretación de fotografías aéreas
	Guiado de robots (manipuladores, vehículos no tripulados)		Control de plantaciones
BIOMÉDICAS	Análisis de imágenes de microscopia (virus, células, proteínas, ...)	MILITARES	Seguimiento de objetivos
	Resonancias magnéticas, tomografías, genoma humano		Vigilancia

Tabla 1. Principales aplicaciones de la visión artificial.

Tal vez el mayor número de aplicaciones industriales que están funcionando a fecha de hoy con este tipo de sensores, sean las de control de calidad de productos (ver Figuras 1 y 2). Como se ha mencionado antes, la ventaja de los sensores de visión respecto a los demás tipos de sensores reside en la cantidad de información que captan y también el gran número de ámbitos que se pueden utilizar.

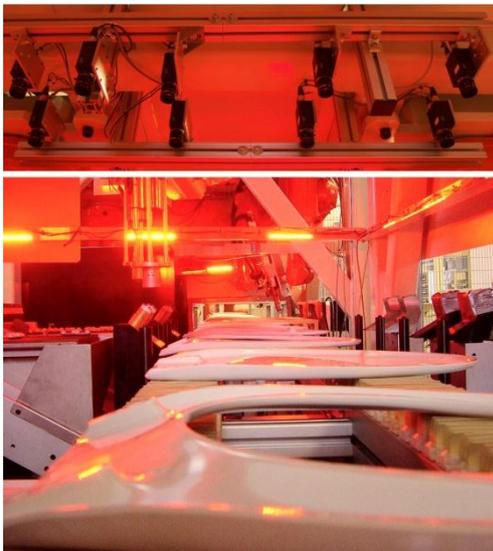


Figura 1. Sistema hardware de visión artificial para el control de calidad.

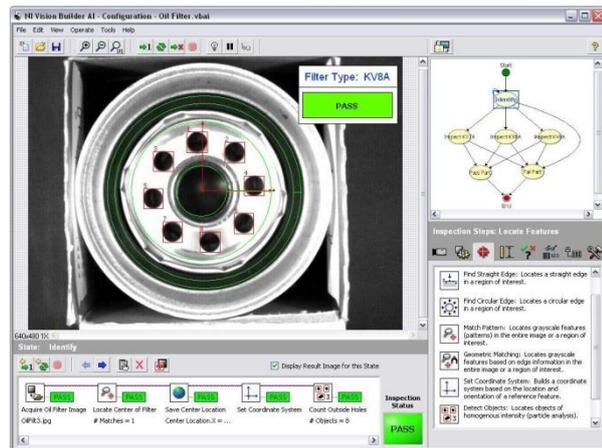


Figura 2. Software de visión para control de calidad en la industria del automóvil.

Desde hace ya más de tres décadas, se ha estado investigando el ámbito de guiar los sistemas robóticos, tales como robots manipuladores o robots autoguiados (AVG's). El término utilizado para definir este tipo de sistemas ha sido finalmente el de "Visual Servoing". En la actualidad, una de las industrias que con mayor éxito ha implantado este tipo de sistemas se trata de la alimentaria. En concreto se utilizan robots para la clasificación de productos tales como salchichas, croissants, etc. (ver Figuras 3 y 4).



Figura 3. Aplicaciones industriales del Visual Servoing ABB Flex-Pick de tortitas.



Figura 4. Aplicaciones industriales del Visual Servoing, ABB Flex-Pick de salchichas.

Hoy en día se está estudiando la posibilidad de implantar este tipo de sistemas para tareas más complejas, como coger objetos colocados de forma arbitraria amontonados, o la de realizar tareas tales como colocación de partes del coche mientras este está en movimiento.

Por otro lado, los sistemas que trabajan con visión tienen un problema asociado que es la gran latencia que generan en el sistema global. Esto significa lo siguiente: el periodo de ejecución de cada bucle de control va a venir determinado principalmente por el algoritmo de visión. Con los avances tecnológicos, en cuanto al hardware, se ha mejorado mucho el tiempo que se tarda en actualizar las características del/los objeto/s a detectar, pero aún así hay limitaciones hardware que impiden bajar más los tiempos del algoritmo de visión. Este problema se acentúa cuando se requieren seguimientos muy rápidos que, aunque la dinámica del robot en cuestión permitiría realizarlos, el conjunto sistema de visión + robot no lo permite.

Se puede apreciar en el párrafo anterior que sistema *Visual Servoing* es un sistema que tienen partes que trabajan a distintas frecuencias. Como se ha dicho, tradicionalmente los sistemas muestreados, los cuales están interconectados mediante muestreadores y retenedores, trabajan a la misma frecuencia, lo que implica directamente que se tiene que trabajar a la frecuencia del elemento más lento del sistema.

Sin embargo, desde hace ya unas cuatro décadas, se ha estado estudiando los sistemas muestreados que trabajan con retenedores y muestreadores a distintas frecuencias. A estos sistemas los denominamos "*Sistemas Muestreados Multifrecuenciales*".

En el ámbito de la *Ingeniería de Sistemas Automáticos*, los *Sistemas Muestreados Multifrecuenciales* tienen un gran interés en cuanto a la investigación debido al potencial que supone, ya que en la realidad todos los sistemas sensorizados son sistemas multifrecuencia. Así pues, la investigación de este tipo de sistemas tiene un claro interés en sistemas automatizados complejos, tales como:

- ***Sistemas con restricciones tecnológicas:*** son sistemas en las que las medidas deben muestrearse a una cierta frecuencia y las acciones de control a otra distinta, tales como sistemas robóticos [1], [2], [3], [4], [5], [6] y [7], procesos químicos [8], [9] y [10], etc.
- ***Sistemas de control distribuido:*** son los sistemas con redes industriales en los que el procesador tiene acceso a la misma [11], [12], [13], [14] y [15].
- ***Sistemas multivariables:*** realizando un control multifrecuencia se pueden obtener mejoras [16], [17], [18] y [19].

- **Sistemas de tiempo real:** en los que se implementan distintas tareas y deben ejecutarse según la planificación requerida [20].

En concreto, dentro del campo del *Visual Servoing* para manipuladores, los sistemas multifrecuencia deben ser estudiados debido a que este tipo de sistema tiene clara connotación multifrecuencial. Esto es, se tiene la parte de sensorización que trabaja a una frecuencia más alta que la parte de control. Por otro lado, el tema de las comunicaciones ha de tenerse en cuenta ya que puede influir, en nuestro caso de aplicación influye, pues obliga trabajar como mínimo a una frecuencia determinada. Además, el estudio de la planificación de tareas a realizar, se trata de una aplicación de tiempo real, impone también ciertas restricciones frecuenciales.

En nuestro caso de estudio se va a proponer el sistema denominado “*Eye-In-Hand*”, aunque hay otros tipos de enfoque que pueden utilizar más cámaras, colocadas tanto en el efector final del manipulador como fuera del robot, o también la incorporación de otros tipos de sensores diferentes, como es el caso del láser. En este sentido, el uso de técnicas *Multifrecuenciales* está plenamente justificado en un sistema *Visual Servoing*.

Como se ha venido diciendo, la gran mayoría de aplicaciones realizadas en el ámbito del *Visual Servoing* evitan toda la problemática de la multifrecuencia. En la mayoría de casos, se han limitado a simplificar algoritmos, o a mejorar de alguna manera el control para poder realizar la aplicación en los tiempos deseados. Por otro lado, existen investigaciones [21] y [22] que han intentado introducir la metodología de la multifrecuencia en el *Visual Servoing*, pero son muy escasos. Sin una sistemática de diseño bajo el punto de vista de la teoría de control de sistemas multifrecuencia, se limitan a dar soluciones de implementación.

Objetivos de la Tesina

El principal objetivo de la presente tesina es el estudio de las distintas metodologías del *Visual Servoing* para al caso de seguimiento de objetos dinámicos. A partir de este estudio se utilizarán técnicas de estimación y control bajo la perspectiva de la teoría de sistemas multifrecuencia.

En particular estas técnicas se han validado en ámbito del *Visual Servoing* para manipuladores industriales de *6 grados de libertad*. Con este fin, se ha tratado la problemática desde el punto de vista del muestreador y la estimación del estado.

En esta tesina se ha hecho hincapié en el estudio del estado del arte, tanto de la teoría de sistemas multifrecuencia como de sistemas *Visual Servoing*, y también en realizar la validación mediante simulaciones y experiencias reales, aplicándolo a robots de 6 grados de libertad.

Es por ello que todas las técnicas estudiadas y comprobadas en simulación se realizan y validan mediante experiencias reales, ya que de esta manera quedan resaltados todos los problemas y limitaciones que presentan las no linealidades en dichos sistemas.

En el Capítulo 2, se realiza un estudio del estado del arte tanto de los sistemas muestreados Multifrecuenciales como del *Visual Servoing* para robots manipuladores. Como se ha venido diciendo, una parte importante de la presente tesina es la comprensión de las distintas técnicas existentes en la actualidad en las dos temáticas principales, es por ello que en este capítulo se detallaran los aspectos más significativos de ambas.

En el Capítulo 3, se presenta el enfoque elegido para tratar la problemática de seguimiento de objetos dinámicos con robots manipuladores a partir de sensores de visión. Se detallará toda la teoría sobre este enfoque. Además se presentará el simulador realizado para el estudio teórico del planteamiento del *IBVS*. Por otra parte, se hará un listado del hardware utilizado para la experimentación real. También se explicará el *setup* del robot *Kuka KR5 sixx R650* que se ha realizado, ya que ha sido una de las partes que han consumido mayor tiempo en el transcurso de esta tesina. Finalmente, se presentan todos los resultados acerca del estudio del enfoque *IBVS* clásico, tanto en su versión simulada como en su versión con el proceso real.

El Capítulo 4, se centra en la explicación de cómo hay que proceder a la hora de implementar la multifrecuencia en un proceso no lineal, como es el caso del *Visual Servoing*. Para ello, se realizará el estudio detallado con un regulador PID y el sistema robotizado con 2 grados de libertad. Una vez estudiados los resultados y vistos los pros y contras de cada uno de los enfoques, se extrapolará todo al caso particular del enfoque *IBVS*. Se mostrará una comparativa de los resultados obtenidos en cada uno de los enfoques empleados.

Finalmente, en el Capítulo 5 se extraen algunas conclusiones al respecto de las aportaciones de la tesina y los trabajos futuros a realizar de cara a la tesis doctoral.

Además, se presentan como complemento a la memoria 3 anexos. Uno de los más interesantes es el Anexo 1, ya que en él se realiza un estudio muy detallado de un caso lineal de las técnicas que se han empleado para la realización del sistema *Visual Servoing Multifrecuencia*.

CAPÍTULO 2

Estado del arte

2.1.-Visual Servoing

La robótica guiada por sensores de visión es uno de los principales temas de investigación desde hace más de tres décadas. El reciente desarrollo tecnológico ha facilitado el avance en éste campo, el cual ha proporcionado algunos sistemas comerciales de mucho éxito (por ejemplo el clasificador de salchichas visto en la Figura 4). Como se ha comentado existen bastantes aplicaciones de sistemas guiados por visión, que van desde domótica hasta la industria. Sin embargo, en cuanto a robots manipuladores se refiere, el campo de investigación aún está abierto, ya que los algoritmos implementados y las técnicas utilizadas aún no proporcionan resultados totalmente satisfactorios en aplicaciones complejas. En esta sección se va a tratar de mostrar una visión general del estado actual en este campo, así como se definirá la notación a seguir durante todo el documento de la tesina¹.

2.1.1.-Visión general del Visual Servoing

Cuando se habla de controlar un robot con realimentación por visión se está refiriendo al término comúnmente conocido como “*Visual Servoing*” [23]. Las características visuales (basadas en la imagen) como los puntos, líneas y regiones pueden emplearse para, por ejemplo, permitir el alineamiento de un robot con una pinza en su efector final para coger un determinado objeto. De este modo, la visión proporciona la realimentación del estado actual del entorno al sistema de control, el cuál actúa en consecuencia. La temática del *Visual Servoing* ha sido estudiada de distintas formas desde hace ya unas tres décadas, empezando por tareas simples de pick-and-place, hasta tareas de manipulación de objetos en tiempo real. En este sentido (el de la manipulación de objetos), uno de los principales motivos de incorporar sensores de visión en el bucle de control fue la creciente demanda de *flexibilidad* en los sistemas robotizados.

Un problema aún abierto hoy en día en este campo es la necesidad de trabajar conjuntamente toda la comunidad dedicada al tema, ya que éste no es un tema sencillo de resolver y es necesario un intercambio de experiencias y aporte de nuevas ideas.

Un control en bucle cerrado de un sistema robot, donde la visión se utiliza como sensor de medida, normalmente consiste en dos procesos interconectados: tracking y control. El tracking proporciona una estimación continua y una actualización de las características a detectar durante el movimiento robot/objeto. Basándonos en las medidas proporcionadas por el sensor de visión, se genera una secuencia de control. Además, el sistema también requiere de una inicialización.

¹ Uno de los problemas que se tiene en este campo es la notación. Esta aún no está definida completamente y los autores de cada artículo revisado utilizan una notación distinta dependiendo del campo de aplicación al cual proceden.

Como se ha mencionado anteriormente, hace tres décadas la tecnología no permitía realizar aplicaciones de *Visual Servoing* complejas en tiempo real. En la actualidad, y debido al avance tecnológico, el *Visual Servoing* ha venido adquiriendo un considerable interés debido a que es posible realizar tareas complejas en tiempo real. Al mismo tiempo, se han realizado métodos robustos para escenarios del mundo real, permitiendo así abordar problemas reales cada vez más complejos. Un problema que ha empezado a aparecer con el incremento de publicaciones en este campo, es la gran variedad de terminología y taxonomía de los enfoques presentados.

Por otra parte, el *Visual Servoing* también se está utilizando en robótica móvil para el seguimiento de líneas, navegación terrestre, aérea y submarina [24] y [25]. En el campo de la robótica móvil aérea, como es el caso de helicópteros, actualmente han salido muchos artículos con distintos enfoques tales como [26], [27] y [28].

En este documento vamos a tratar el tema del *Visual Servoing* aplicado en robots manipuladores. En este campo, la aplicación más común es la de la manipulación de objetos, la cual requiere de una detección, segmentación, reconocimiento, seguimiento, alineamiento y al final coger el objeto. Aunque muchos enfoques no entran en todos estos aspectos, las tareas de manipulación proporcionan una visión global para considerar las distintas fases del *Visual Servoing*.

2.1.2.-Conceptos generales

En un sistema robot existen, normalmente, dos formas de utilizar la información visual recibida:

- **Control del robot en lazo abierto:** la extracción de las características del objeto por parte del sistema de visión y el control de robot son tareas separadas donde, en primer lugar, se procesa la información por el sistema de visión y en segundo lugar se realiza la generación de la secuencia de control (ver Figura 5, izquierda). Un caso típico es el reconocimiento del objeto a ser manipulado, calculando su posición y orientación (*pose*) relativa al sistema de coordenadas de la cámara (*robot*). Con esta información, el robot se mueve desde la posición actual a la deseada. Para estimar la pose del objeto se debe tener un modelo del mismo. Para mover el robot a partir de la información extraída respecto al sistema de coordenadas de la cámara, esta debe estar calibrada con respecto al robot. Además, se deben tener los modelos directo e inverso de la cinemática del robot en cuestión para poder pasar de coordenadas cartesianas a coordenadas del robot y viceversa. Así, el robot puede ejecutar la tarea pero de forma ciega ya que no va a saber si el objeto se ha movido del lugar mientras está ejecutando la misma (el llamado “*enfoque de lazo abierto*”).

- **Control del robot en lazo cerrado (*Visual Servoing*):** En 1979 [29] introdujeron el término *Visual Servoing* para distinguir su enfoque de los trabajos anteriores en este campo (ver Figura 5, derecha). En 1980 la siguiente taxonomía de sistemas *Visual Servoing* fue introducida en [30]:
 - **Sistemas dinámicos *look-and-move*.** Este tipo de sistemas mejoran el control del robot en dos fases: el sistema de visión proporciona información a la entrada del controlador, que utiliza información de las articulaciones para internamente estabilizar el robot.
 - **Sistemas *Visual Servoing Directo*.** Aquí, el controlador visual controla directamente las articulaciones del robot y el controlador del robot se elimina.

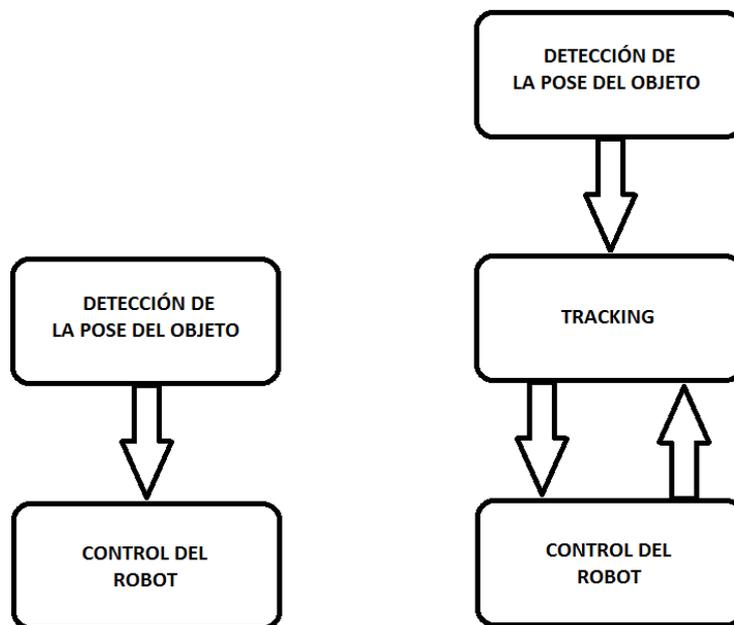


Figura 5. Izquierda: Control del robot en lazo abierto. La idea es detectar el objeto y a partir de la pose detectada calcular la secuencia de acciones de control que permitan al robot llegar a la posición deseada. Derecha: Control en lazo cerrado. La idea es, una vez detectado el objeto el control siempre va a tener en cuenta en cada iteración el estado del objeto con respecto al robot.

Una tarea típica del *Visual Servoing* incluye:

- “**Posicionamiento**”, tal como alinear la pinza del robot con el objetivo.
- “**Tracking**”, tener una pose constante entre el robot y el objeto en movimiento.

En ambos casos, la información extraída de la imagen se utiliza para medir el error entre la localización actual del robot y su referencia o localización deseada. La información de la imagen se utiliza para realizar la tarea, ya sea:

- Utilizando el plano de la imagen (*Visual Servoing* 2D o más conocido como ***Image-Based Visual Servoing***).
- *Visual Servoing* 3D (o más conocido como ***Positio-Based Visual Servoing***), donde se emplea un modelo de la cámara y del objeto para extraer la información de la pose del mismo con respecto al sistema de coordenadas cámara/mundo/robot.

Por lo tanto, utilizando lo anteriormente dicho, existen tres enfoques básicos conocidos por todo investigador de esta temática que son:

- ***Position-Based Visual Servoing (PBVS)***. El sistema extrae la información 3D de la escena, con el modelo de la cámara conocido (normalmente en conjunción con un modelo conocido del objeto) se utiliza para estimar la pose del objeto respecto del sistema de coordenadas de la cámara. La tarea de posicionamiento o tracking se define en el espacio 3D.
- ***Image-Based Visual Servoing (IBVS)***. Las medidas en el plano de imagen 2D se utilizan directamente para mover el robot de la posición actual a la deseada. Tareas típicas como tracking y posicionamiento se realizan reduciendo el error de las características en el plano de imagen.
- ***Hybrid-Based Visual Servoing (HBVS) o 2 ½ D Visual Servoing***. Se realiza una combinación entre los dos enfoques anteriores, donde el error a minimizar se especifica tanto en el plano de imagen como en el 3D.

Por lo tanto, la idea general del *Visual Servoing* es ver la relación existente entre el robot y el objeto, y calcular la velocidad asociada a las articulaciones del robot necesaria para minimizar el error especificado. Como se verá más adelante, lo que realmente se calcula es el conocido “*skew vector*”, el cuál es la velocidad asociada a cada una de las componentes del sistema de coordenadas asociado a la cámara. A partir de ahí, hay que hacer las transformaciones necesarias para asociarlo al sistema robot.

Como se ha mencionado anteriormente, el *Visual Servoing* utiliza conceptos asociados a distintas áreas de la ingeniería, muy diferentes entre sí, incluyendo modelado de robots (geometría, cinemática, dinámica), sistemas de tiempo real, teoría del control, integración de sistemas (sensores), visión por computador (procesamiento de imagen, *structure-from-motion*, calibración de cámaras). Como se señala en [31], hay diferentes formas para clasificar los resultados obtenidos: basándose en la configuración del sensor, según el número de cámaras utilizado, generación de movimiento (2D, 3D), interpretación de la escena, según el algoritmo de visión empleado, etc. Se va a emplear la Taxonomía mostrada en la Figura 6, ya que la emplean un gran número de publicaciones más actuales:

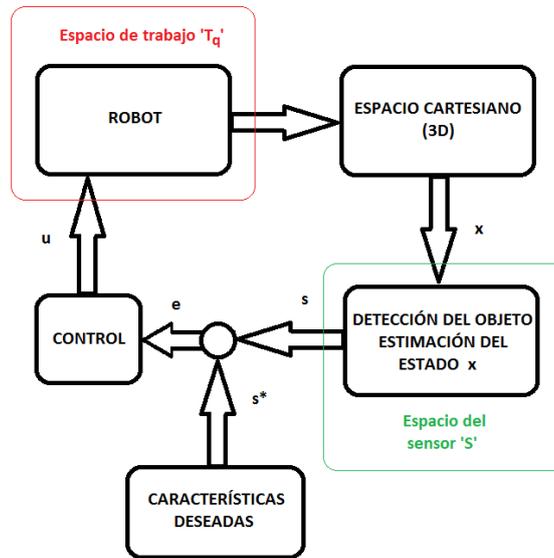


Figura 6. Procesos envueltos en un sistema *Visual Servoing*. El movimiento euclideo relativo del robot viene determinado por las acciones del controlador en el espacio de trabajo del robot T_q . “ x ” representa el vector de estado, s es el vector de características medidas y s^* es el vector de características deseadas que son medidas en el espacio del sensor S .

2.1.3.- Definiciones y notaciones

A partir de este punto se va a asumir, sin pérdida de generalidad, que se está trabajando en un sistema robot que trabaja en el espacio 3D. Como se ha mencionado, el movimiento euclideo relativo es debido por la entrada u en el espacio de trabajo T_q . Ya que, en este contexto, T_q representa el conjunto de todas las posibles configuraciones que puede alcanzar del efector final, este va a denotarse como $T_G \subseteq SE(3)$. Si el robot se controla utilizando 6 grados del libertad, el vector de acciones de control es $u = [V_x \ V_y \ V_z \ \omega_x \ \omega_y \ \omega_z]^T \in \mathcal{R}(6)$, que se denomina **screw vector** del robot. El vector de estado, x , representa la pose del objeto, $X = [R, t]$, normalmente representado por los parámetros de rotación y translación $q = [t_x \ t_y \ t_z \ \phi \ \psi \ \gamma]^T \in \mathcal{R}(6)$. Sin pérdida de generalidad se va a asumir que el vector de medidas s está formado por una serie de puntos que definen la pose del objeto, $s = \left[[x_1 \ y_1] \ \cdots \ [x_{k/2} \ y_{k/2}] \right]^T \in \mathcal{R}(k)$ mientras que s^* representa el vector de medidas deseado.

A las tareas de *Visual Servoing* también se les denomina como “*task function*” [32] o función de control del error. Representando como s^* el conjunto de

características deseadas y s , el conjunto de características actuales, el objetivo del *Visual Servoing* es hacer que este error sea 0. Cuando la tarea se completa, la ecuación (1) se cumple.

$$e(s^* - s) = 0 \quad (1)$$

La “*task function*” también es referenciada en algunos artículos como función del error cinemático o restricción cinemática virtual (en el caso de PBVS) y función de error en la imagen (en el caso de IBVS), [33]. En este sentido, básicamente las investigaciones están divididas hablan de:

- **Modelo de estimación Visual-Motor.** En [31] se hizo una distinción entre el control dinámico y el control cinemático visual, donde se estipula como un manipulador debe moverse como respuesta a la información visual percibida, mientras que las últimas investigaciones van encaminadas a tener en cuenta efectos de la dinámica que normalmente aparece en un sistema robótico. De acuerdo con esto (ver Figura 7), los sistemas se pueden clasificar como:
 - Sistemas donde el modelo Visual-Motor es conocido a priori.
 - El modelo es estimado

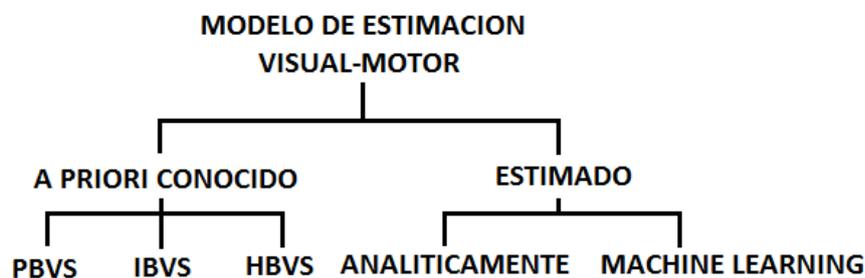


Figura 7. Clasificación según el Modelo de Estimación Visual-Motor. Los sistemas donde el modelo es conocido a priori utilizan el modelo cinemático del robot, los parámetros de la cámara con diferentes niveles de calibración entre la cámara y el robot para estimar la posición deseada del mismo. Por otro lado, hay sistemas que estiman el modelo Visual-Motor ya sea por técnicas de “*Machine Learning*” o analíticamente, permitiendo el control sin conocimiento de la geometría del robot.

- **Estimación del estado.** Los artículos relacionados con la extracción de información visual, clasifican el sistema según: la configuración de la cámara, el número de cámaras y según las técnicas de visión utilizadas. Las técnicas de extracción de características más utilizadas en este ámbito son tales como el flujo óptico, posición y orientación de un objeto o características como puntos o líneas. En la Figura 8 se muestran las distintas configuraciones sensor-robot que se utilizan dependiendo de la aplicación (*eye-in-hand*, sistema de cámara *stand alone* (cámara

fija) o una combinación de estas). La estimación en el caso de una cara puede realizarse en el espacio de la imagen (2D) y utilizarse conjuntamente con un modelo de cámara para obtener la información 3D. Para sistemas que utilizan directamente la información en el plano de imagen la “*task function*” está normalmente definida en el plano de imagen. Sin embargo, este no es el caso general para sistemas que calculan la pose 3D de objeto, en cuyo caso la “*task function*” viene expresada tanto en 2D como en 3D.



Figura 8. Configuraciones Cámara-Robot utilizadas en el *Visual Servoing* (de izquierda a derecha): Configuración 1 *monocular eye-in-hand*, Configuración 2 *stereo eye-in-hand*, Configuraciones 3 y 4 *monocular* y *estéreo stand-alone* y Configuración 5 sistema de cámaras redundantes.

- **Generación del Control.** Está muy relacionada con la de Estimación del modelo *Visual-Motor*. La exactitud del modelo afectará directamente al ratio de convergencia del sistema.

Por ser la más extendida, se va a comentar la clasificación proporcionada por el enfoque de estimación del modelo *Visual-Motor*. Dependiendo de la exactitud del modelo y la complejidad de la tarea los sistemas se clasifican en posición, imagen e híbridos (*PBVS*, *IBVS* y *HBVS*) que serán los que se comenten en la siguiente sección.

2.1.4.-Clasificación de los sistemas conocido el modelo

2.1.4.1.- Control basado en la posición (*PBVS*)

Usualmente el enfoque denominado *Position-Based Visual Servoing (PBVS)* se refiere como un control en el espacio 3D, ya que la información obtenida por la cámara es utilizada para determinar la posición del objeto respecto al sistema cámara u otro sistema perteneciente al mundo 3D. El error entre la posición deseada y la posición actual del objeto se define en el espacio de trabajo del robot, T_q . De este modo, el error es función de los parámetros de pose cartesianos, $e(X)$.

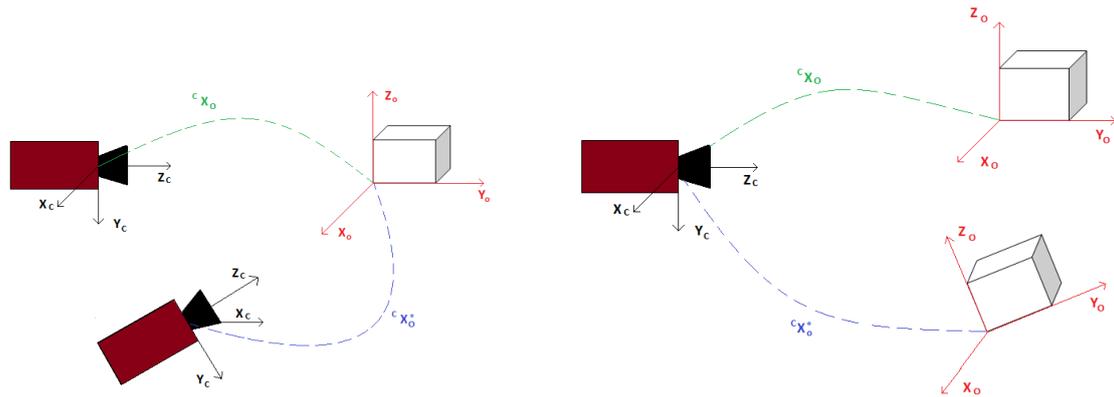


Figura 9. Dos ejemplos de PBVS: en la imagen de la izquierda, ejemplo de configuración de la cámara eye-in-hand donde la cámara-robot pasa de la configuración ${}^c X_o$ (posición actual) a la ${}^c X_o^*$ posición deseada; en la derecha, una cámara en configuración monocular stand-alone.

Dos ejemplos de *PBVS* se muestran en la Figura 9. La figura de la izquierda muestra un ejemplo donde la cámara está colocada en el efector final del robot (configuración *eye-in-hand*), dónde se controla desde la posición actual ${}^c X_o$ hasta alcanzar la posición deseada ${}^c X_o^*$. La figura de la derecha muestra un ejemplo de cámara estática y objeto en movimiento. Se asume que el objeto permite al manipulador llegar desde la posición actual a la deseada en todo su movimiento. Ya que la pose del objeto se estima en relación a la cámara, la transformación entre el robot y la cámara es conocida para generar el movimiento requerido del manipulador.

Estos ejemplos muestran los dos principales motivos del por qué el enfoque PBVS no se utiliza normalmente para tareas de Servoing:

- Requiere la estimación de la pose del objeto o algún modelo del mismo.
- Para estimar la velocidad deseada del robot y mejorar así el posicionamiento del mismo, se requiere una gran precisión en la calibración (cámara, cámara-robot).

La Figura 10 muestra el diagrama de bloques de este enfoque. En él, la diferencia entre la pose deseada y la actual representa un error que se utiliza para estimar el vector de velocidades que permite la minimización de dicho error, $\dot{q} = [V \quad \Omega]^T$.

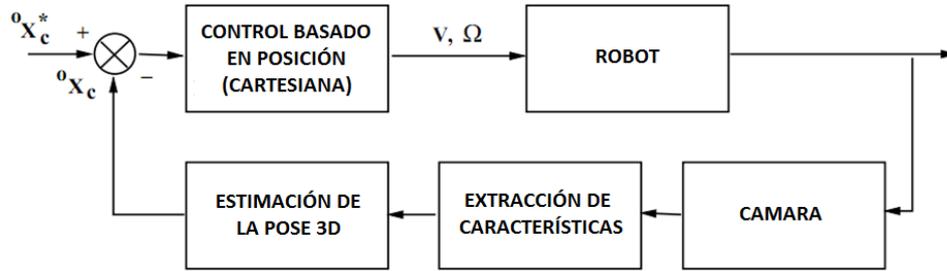


Figura 10. Diagrama de bloques del enfoque PBVS: la pose del objeto se estima, ${}^c X_o$, y se compara con la referencia (pose deseada), ${}^c X_o^*$. El error se utiliza luego para calcular el skew vector, $\dot{q} = [v \ \Omega]^T$ de esta forma el robot minimiza el error.

En general, la principal ventaja de este enfoque es que la trayectoria cámara-robot está dada directamente en coordenadas Cartesianas. Esto permite fácilmente realizar una planificación de la misma, por ejemplo para evitación de colisiones. Sin embargo, especialmente en el caso *eye-in-hand*, las características de la imagen usadas para la estimación de la pose del objeto pueden salirse del plano de imagen. La razón es que la ley de control no incorpora ninguna restricción en este aspecto. Si la cámara está mal calibrada (por ejemplo, los parámetros de la cámara se conocen aproximadamente), la posición actual y deseada de la cámara no se estimará exactamente lo cual hará un mal *Visual Servoing* o incluso puede fallar. Una de las soluciones a este problema es diseñar controladores de más bajo nivel para minimizar este tipo de fallos [33].

Existen ejemplos que utilizan tanto la configuración *eye-in-hand* como la configuración *stan-alone* para realizar tareas en PBVS. Los ejemplos van desde posicionamiento planar [34], a sistemas que utilizan modelos de objetos y determinación de la pose en tiempo real, [35], [36] y [37].

Una evaluación extensiva del enfoque *PBVS* con respecto a ventajas y desventajas de requerimiento de velocidad, exactitud y robustez viene dada en [38]. Por último los sistemas que utilizan este enfoque tienen que realizar una exacta extracción de la información visual porque son muy sensibles.

2.1.4.2.- Control basado en la imagen (IBVS)

El *Image Based Visual Servoing (IBVS)* realiza la estimación del *skew vector* del robot (es decir el vector de velocidades de las articulaciones del robot), \dot{q} , a partir de las características del objeto en el plano de imagen actual, s , a un conjunto de características en el plano de imagen deseadas, s^* , [39], [40] y [41]. El control *IBVS* se realiza por medio de la matriz Jacobiana de las características, llamada *matriz de interacción*, [33], [42] y [43]. La matriz de interacción representa la relación diferencial

entre el sistema de coordenadas de la escena y el sistema de coordenadas de la cámara (ambos sistemas habitualmente unidos al robot):

$$J(\mathbf{q}) = \left[\frac{\partial \mathbf{s}}{\partial \mathbf{q}} \right] = \begin{bmatrix} \frac{\partial s_1(\mathbf{q})}{\partial q_1} & \dots & \frac{\partial s_1(\mathbf{q})}{\partial q_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_k(\mathbf{q})}{\partial q_1} & \dots & \frac{\partial s_k(\mathbf{q})}{\partial q_m} \end{bmatrix} \quad (2)$$

Donde \mathbf{q} representa las coordenadas del efector final en cualquier parametrización del espacio de trabajo \mathbf{T}_q , $\mathbf{s} [s_1 \ s_2 \ \dots \ s_k]$ representa el vector de características de la imagen, m es la dimensión del espacio de trabajo \mathbf{T}_q y k es el número de características de la imagen. La relación entre el vector “velocity skew” asociado al manipulador y los parámetros de la imagen viene dado por:

$$\dot{\mathbf{s}} = \mathbf{J}\dot{\mathbf{q}} \quad (3)$$

Utilizando el clásico modelo de proyección llamando a la distancia focal f , y particularizando, sin pérdida de generalidad, para el caso de un punto, $\mathbf{p} = [x \ y \ z]^T$ se tiene,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{f}{Z} & 0 & -\frac{x}{Z} & xy & (1+x^2) & -y \\ 0 & \frac{f}{Z} & -\frac{y}{Z} & -(1+y^2) & xy & x \end{bmatrix} \dot{\mathbf{q}} \quad (4)$$

Donde Z representa la distancia en el mundo 3D del punto respecto a la cámara. Los sistemas IBVS expresan el control de la función error directamente en el espacio de la imagen. Si las posiciones de las características en el plano de imagen se utilizan como medidas, la función error se define simplemente como la diferencia entre la posición actual y la deseada de la característica en el plano de imagen:

$$\mathbf{e}(\mathbf{s}) = \mathbf{s}^* - \mathbf{s} \quad (5)$$

El enfoque más común para generar señales de control para los robots es utilizar un simple control proporcional [44] y [45]:

$$\mathbf{u} = \dot{\mathbf{q}} = \lambda \mathbf{J}^+ \mathbf{e}(\mathbf{s}) \quad (6)$$

Donde \mathbf{J}^+ es la pseudo-inversa de la matriz de interacción y λ es la ganancia.

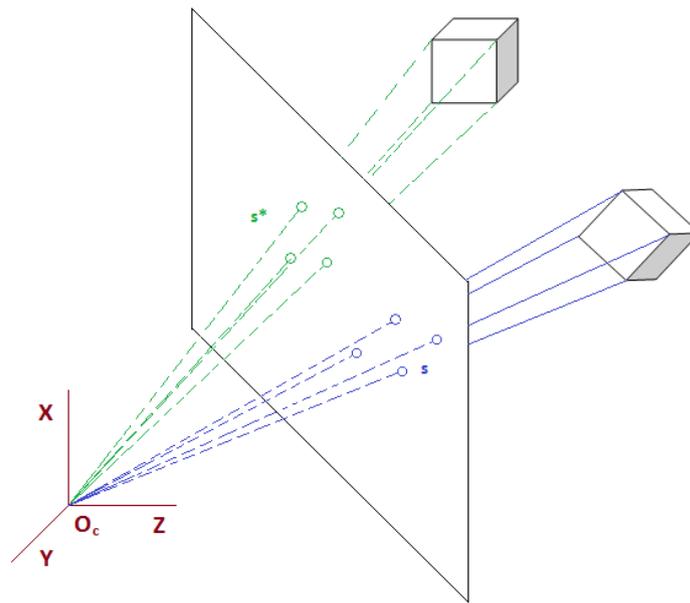


Figura 11. Ejemplo de IBVS. Se asume el caso de una cámara estática y un objeto móvil. Un número de características del objeto se utiliza para generar el vector s . El vector de características deseadas se denota como s^* . El error se define como la diferencia entre ambos vectores, $e(s) = s - s^*$. Esta función error se actualiza en cada bucle de control y se utiliza junto a la matriz de interacción para estimar la acción de control del robot.

La Figura 11 muestra un ejemplo del enfoque *IBVS*, donde se asume que la cámara es estática y lo que se mueve es el objeto. Se cogen un número de características del objeto que lo definen y con ello se genera el vector s . Asimismo el vector de la referencia deseada para esas características dentro del plano de imagen viene dado por el vector s^* . La función error se define como la distancia entre ambos vectores, de acuerdo con la Ecuación 5. Cada nueva imagen, esta función se actualiza y en conjunto con la matriz de interacción se estima las acciones de control para el robot de acuerdo con la Ecuación 6.

El vector de referencias, s^* , normalmente se genera utilizando la técnica llamada *“teach by showing”*, que consiste en mover en primer lugar el robot a la posición deseada y luego guardar las coordenadas de las características en el plano de imagen como referencia. Después de ser movido el robot a cualquier otra posición, que será la posición inicial del robot, se inicia el tracking. En [46], la posición deseada entre la pinza y el objeto se define mediante una representación proyectiva y la nueva imagen deseada se calcula cuando el objeto cambia en vez de realizarlo manualmente.

De acuerdo con la Ecuación 2, la estimación de la matriz de interacción requiere del conocimiento de los parámetros intrínsecos y extrínsecos de la cámara. Los parámetros extrínsecos representan un mapeo rígido entre la escena o alguna imagen de referencia y el sistema de la cámara. Si se utiliza una sola cámara en el

proceso de Servoing, la profundidad necesaria para la actualización de la Jacobiana se pierde. Por lo tanto, muchos sistemas lo que hacen es mantener constante esta matriz por ejemplo en la pose deseada de la cámara/efector final. Este es uno de los hándicaps de este enfoque, ya que la convergencia se garantiza solamente alrededor de la posición deseada. Este problema puede resolverse mediante la estimación adaptativa de dicha profundidad [44], determinando la profundidad desde una relación conocida a priori de las características o utilizando el *enfoque structure-from-motion* si el movimiento de la cámara puede medirse [47] y [48]. Sin embargo, utilizando la profundidad variable los resultados no son los adecuados llegando a caer en posibles mínimos locales y singularidades y llegando finalmente a inestabilizar el sistema [49]. Si se utiliza un sistema de una cámara *stand-alone* y si la calibración entre los sistemas de coordenadas robot y cámara son (parcialmente) conocidos, la profundidad requerida para la matriz de interacción puede conseguirse utilizando la cinemática directa del robot y los parámetros de calibración de la cámara. La matriz de interacción depende también del tipo de características utilizado y de la propia tarea de Servoing (posicionamiento punto a punto, posicionamiento punto a línea), ver los ejemplos en [50].

En general, se precisa de un mínimo de tres puntos para controlar la posición y orientación de la cámara en el espacio 3D (asumiendo que se está empleando una configuración *eye-in-hand*). Sin embargo, hay dos casos de configuraciones singulares [51]:

- Si los tres puntos están alineados
- Si el centro óptico recae en el cilindro que incluye los tres puntos y cuyo eje es perpendicular al plano que los contiene.

En [43] se demostró que si la matriz de interacción es de rango completo, en el caso de cuatro puntos, si tres de ellos no están alineados.

Comparado con la configuración *eye-in-hand*, donde el objeto a manipular y el efector final no aparecen simultáneamente en el campo de visión de la cámara (normalmente), en una configuración *stand-alone* se puede fácilmente observar simultáneamente tanto el objeto como el efector final del robot [50]. Si se utiliza un sistema estéreo, se tiene que tener en cuenta la siguiente propiedad:

“Disparidad cero entre un punto en el manipulador y un punto en el objeto en dos imágenes distintas significa que esos dos puntos son el mismo en el espacio 3D.”

La matriz de interacción se estima simplemente concatenando las dos matrices de interacción debidas a cada una de las cámaras. Si la geometría epipolar de la cámara es conocida, la estimación de la profundidad es trivial. Además, imponiendo una trayectoria lineal en las dos imágenes la resultante es una trayectoria lineal del

robot en el espacio 3D mientras que en el caso de una cámara ninguna curva plana se proyecta como una línea en la imagen.

El enfoque *IBVS* está considerado como el más robusto en relación a los errores de calibración de la cámara y del robot [23] y [52]. Una mala calibración de la cámara solamente afecta la ratio de convergencia del control, que tardará mayor tiempo en alcanzar la posición deseada.

2.1.4.3.- 2 ½ D Visual Servoing (HBVS)

En 1998, [40] presenta el enfoque *2 ½ Visual Servoing*. El método fue propuesto originalmente para una configuración *eye-in-hand*. Este enfoque es una “mezcla” entre los enfoques clásicos *PBVS* e *IBVS*. Con ello se evitan las desventajas de cada uno de ellos: contrariamente al *PBVS*, no se necesita ningún modelo geométrico 3D del objeto; con respecto al *IBVS*, se asegura convergencia de la ley de control en todo el espacio de trabajo T .

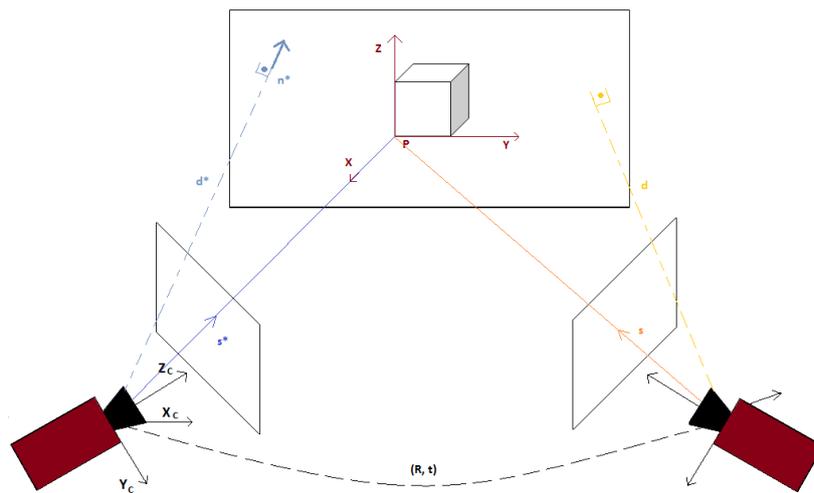


Figura 12. Una cámara observando un plano desde dos posiciones: hay una transformación lineal entre las coordenadas de un punto en dos imágenes de una cámara. Esta transformación es una matriz 3x3 de 8 grados de libertad, M , que debe estimarse utilizando cuatro o más correspondencias del punto. La homografía se puede escribir en términos de los parámetros internos de la cámara, A , el desplazamiento entre los puntos de vista, $X(R, t)$ y la ecuación del plano (n, d) dada por $f = Mf^* = A \left(R + tn^T / d \right) A^{-1} f^*$. Si los parámetros internos, A , son conocidos, la rotación R y el plano normal n pueden obtenerse a partir de la homografía. La translación t y la distancia al plano d pueden obtenerse solamente mediante un escalado [54].

El método está basado en la estimación del desplazamiento de la cámara (la rotación y la translación escalada de la cámara) entre los puntos de vista actual y deseado del objeto. En cada iteración, la rotación entre los dos puntos de vista se estima, lo que permite que el bucle de translación y rotación pueda desacoplarse. En [40] se propone la utilización de coordenadas extendidas de la imagen con lo que se

normaliza la componente z lo que hace que se pueda normalizar las coordenadas de la imagen [54]. Estas coordenadas se obtienen a partir de una reconstrucción parcial Euclídea.

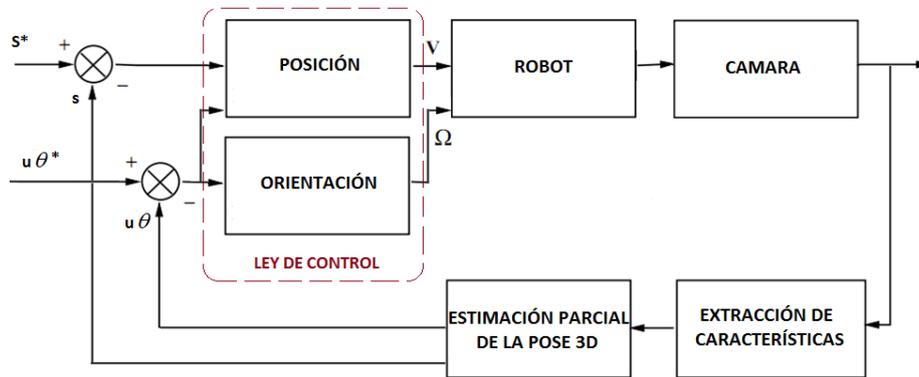


Figura 13. Diagrama de bloques del enfoque 2 ½ D Visual Servoing propuesto por [40]. La tarea de posicionamiento en este caso puede describirse de la siguiente forma: $e = \begin{bmatrix} s - s^* \\ u \\ \theta \end{bmatrix}^T$ donde u y θ representa el eje de rotación y el ángulo de rotación obtenidos de la matriz R .

En [40] se argumenta que la matriz de interacción asignada por los cambios diferenciales entre la velocidad del robot y las coordenadas extendidas, no tiene singularidades. Esto permite la convergencia de una tarea de posicionamiento en todo el espacio si los parámetros intrínsecos de la cámara son conocidos. En el caso de que estos parámetros no sean muy exactos, los autores lo proponen como condición necesaria y suficiente para una estabilidad local asintótica. La idea básica de este enfoque se muestra en la Figura 12 y el diagrama de bloques del mismo se puede ver en la Figura 13. Una cámara montada en el efector final del robot está observando un plano desde la referencia deseada y desde la referencia actual. Existe una transformación lineal que relaciona ambas coordenadas homogéneas s y s^* de un punto en dos imágenes de una cámara *pinhole* a una superficie planar. Esta transformación es una matriz 3x3 M que se estima utilizando cuatro o más correspondencias. La homografía puede escribirse en términos de parámetros internos de la cámara A , el desplazamiento de la cámara debido dos puntos de vista distintos, $X(R, t)$ y la ecuación del plano (n, d) siendo $s = Ms^* = A(R + tn^T/d)A^{-1}s^*$. Si los parámetros internos A son conocidos, la rotación R y el plano normal n pueden obtenerse a partir de la homografía. La translación t y la distancia al plano d se obtienen por escalado [54]. La función error que se tiene que minimiza toma la siguiente forma:

$$e = \begin{bmatrix} s - s^* \\ u \\ \theta \end{bmatrix}^T \quad (7)$$

Donde u y θ representan el eje de rotación y el ángulo de rotación obtenidos a partir de la matriz de rotación, R . Así, se puede diseñar un simple control proporcional para minimizar el error mostrado en la Ecuación 7. Para ver una completa derivación de la matriz de interacción ver [40].

2.2.-Sistemas Multifrecuencia

Los sistemas muestreados están compuestos principalmente por controladores, sensores y actuadores. Todas estas partes del conjunto están interconectadas entre sí por medio de muestreadores y retenedores. Al conjunto se le denomina *sistema muestreado* y puede ser tanto continuo como discreto.

Clásicamente, los *sistemas muestreados* trabajan a la misma frecuencia, aunque en los sistemas reales esto casi nunca se puede cumplir. Se pueden encontrar muchos ejemplos que demuestran la afirmación anterior: sensores (como los de visión) trabajan a una frecuencia más baja que los controladores y los actuadores del sistema; sensores y actuadores que normalmente trabajan a distinta frecuencia, etc. En teoría de control clásico, los controladores para estos sistemas se diseñan para una única frecuencia que debe ser la del elemento más lento del sistema. A estos sistemas se denominan como *sistemas monofrecuencia* trabajando a baja frecuencia.

Es importante señalar que, aunque los sistemas monofrecuencia que trabajan a baja frecuencia funcionan correctamente en procesos reales, se puede mejorar el comportamiento del sistema realizando un diseño de control multifrecuencia. Esto es, el sistema trabaja a diferente frecuencia dependiendo de cada componente del mismo.

En las últimas cuatro décadas, los sistemas multifrecuencia han mejorado muchos procesos que trabajan combinando componentes con distinta frecuencia. Se pueden encontrar ejemplos de aplicaciones industriales como:

- Sistemas que tienen limitaciones tecnológicas y/ o económicas [1], [56] y [57].
- Los denominados procesos o sistemas multiprocesador [11], [12] y [15].
- Los sistemas multivariable [16] y [58].
- Sistemas de tiempo real [59].

2.2.1.- Modelado de sistemas muestreados multifrecuencia

Hay diferentes modelos de sistemas muestreados multifrecuencia y por lo tanto en esta sección se va a describir la matemática que conllevan. Se puede realizar la siguiente clasificación de los modelos de sistemas muestreados multifrecuencia:

- Si se trabaja con relaciones entrada-salida, a estos modelos se les denominará “Modelos en representación Externa”.
- Si se trabaja con variables de estado, entonces a estos modelos se les denominará como “Modelos en representación interna”, [60].

2.2.1.1.- Modelos en representación externa

2.2.1.1.1.- Descomposición en frecuencia

Esta técnica es una de las primeras contribuciones al estudio de sistemas muestreados y fue introducida por [61]. Consiste en introducir muestreadores actuando cada uno de ellos a una frecuencia diferente, la cuál debe ser un múltiplo de la frecuencia base del sistema.

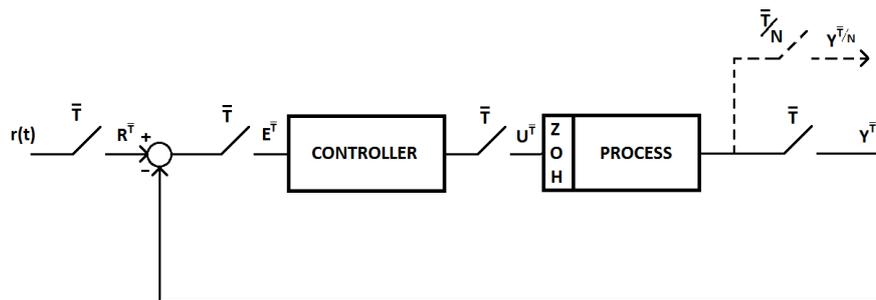


Figura 14. Descomposición frecuencial.

2.2.1.1.2.- Descomposición vectorial

Esta técnica consiste en modelar un muestreador rápido a partir de una combinación de muestreadores lentos mediante bloques de retraso (ver Figura 15), [62]. En general, un muestreador que trabaja a una frecuencia de T/N , siendo N entero, puede verse como N muestreadores trabajando a T (ver Figura 15). En este caso se define el meta-periodo como $\bar{T} = l.c.m.(T_1, T_2, \dots, T_n)$ y se define el periodo base como $T = g.c.d.(T_1, T_2, \dots, T_n)$. Por lo tanto se tiene que la relación entre meta-periodo y periodo base es:

$$\bar{T} = N \cdot T \tag{8}$$

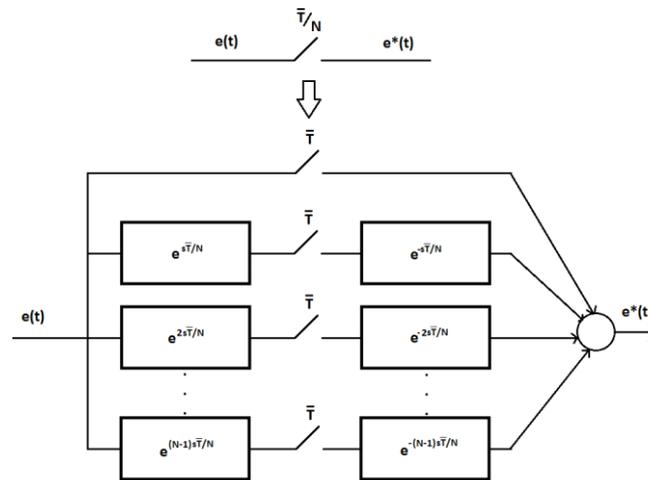


Figura 15. Descomposición Vectorial.

2.2.1.1.3.- Operadores Skip-Expand

Los operadores *Skip-Expand* fueron introducidos por [63] permitiendo modelar cualquier sistema SISO (Simple Input, Simple Output) multifrecuencia. El problema que tienen estos operadores es, que no hay una forma sistemática que permita modelar cualquier sistema, por lo que en la práctica se hace muy complicada su utilización.

El operador *Expand* permite convertir una señal de frecuencia baja a una señal de alta frecuencia ya que extiende o amplía la escala de tiempo (ver Figura 16).

$$y^{\bar{T}/N} = \left[\bar{y}^{\bar{T}}(z) \right]^{\bar{T}/N} = \bar{y}^{\bar{T}}(z^N) \quad N \in \mathbb{Z}^+ \tag{9}$$

En el dominio temporal el operador *Expand* viene definido como:

$$y(k) = \begin{cases} \bar{y}(k/N) & k = 0, N, 2N \\ 0 & \text{in other case} \end{cases} \tag{10}$$

El operador *Expand* cumple la siguiente propiedad:

$$\left[\bar{a}^{\bar{T}}(z) \cdot \bar{b}^{\bar{T}}(z) \right]^{\bar{T}/N} = \left[\bar{a}^{\bar{T}}(z) \right]^{\bar{T}/N} \cdot \left[\bar{b}^{\bar{T}}(z) \right]^{\bar{T}/N} \tag{11}$$

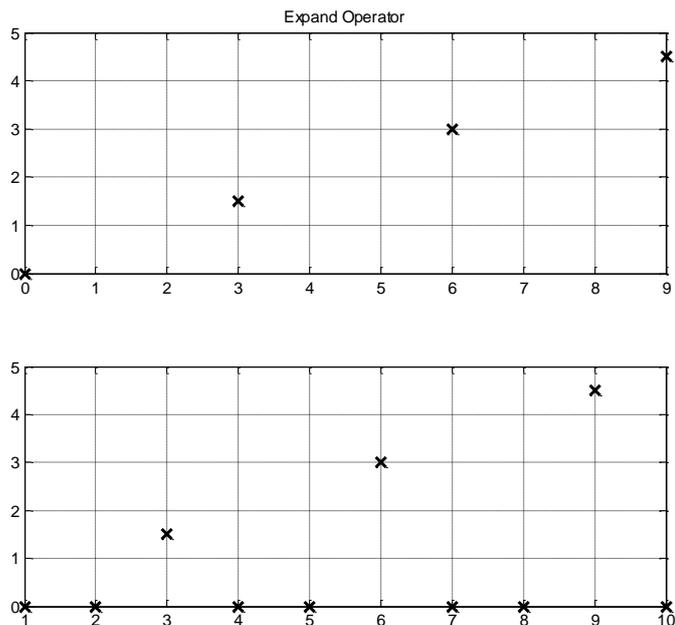


Figura 16. Operador Expand.

Por otra parte, se tiene el operador Skip que convierte una señal con un periodo rápido en otra de un periodo lento:

$$\bar{y}(\bar{z}) = \left[y^{\bar{T}/N}(z) \right]^{\bar{T}} = \frac{1}{N} \cdot \sum_{k=0}^{N-1} y^{\bar{T}/N} \left(e^{j2\pi k/N} \cdot z^{-1/N} \right) \quad (12)$$

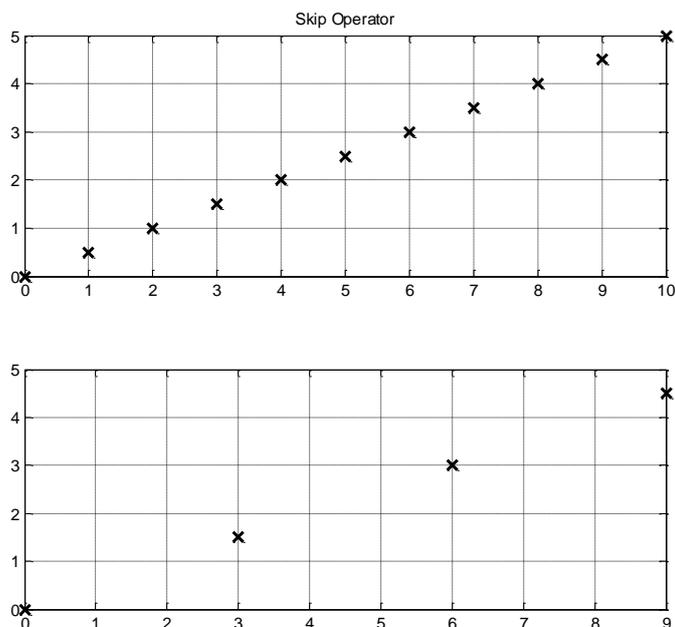


Figura 17. Operador Skip.

Pero el operador Skip no cumple la anterior propiedad (Ecuación 12):

$$\left[\bar{a}^{-T}(z) \cdot \bar{b}^{-T}(z) \right]^{\bar{T}/N} \neq \left[\bar{a}^{-T}(z) \right]^{\bar{T}/N} \cdot \left[\bar{b}^{-T}(z) \right]^{\bar{T}/N} \quad (13)$$

2.2.1.2.- Modelos en representación interna

2.2.1.2.1.- Lifting discreto

La técnica del *lifting* es una de las primeras contribuciones al estudio de sistemas muestreados y fue introducida por [61]. Consiste en introducir *muestreadores* actuando cada uno de ellos a distinta frecuencia, la cuál es múltiplo de una frecuencia base para todo el sistema.

El lifting fue introducido por [64] y en este trabajo los autores establecieron un isomorfismo entre un sistema periódico discreto y otro sistema lineal invariante en el tiempo (LTI). Esto significa que se mantiene las propiedades analíticas como la algebraica en ambos sistemas. Debido a esto, esta técnica es una de la más utilizada en sistemas multifrecuencia.

En primer lugar se utiliza el operador retardo definido como:

$$\Lambda[\alpha(z)] = z^{-1} \cdot \alpha(z) \quad (14)$$

Luego definimos el siguiente isomorfismo:

$$W : \Gamma \rightarrow \Gamma^N : \alpha(\alpha_0, \alpha_1, \dots) \rightarrow \begin{pmatrix} (\alpha_0, \alpha_N, \alpha_{2N}, \dots) \\ (\alpha_1, \alpha_{N+1}, \alpha_{2N+1}, \dots) \\ \vdots \\ (\alpha_{N-1}, \alpha_{2N-1}, \alpha_{3N}, \dots) \end{pmatrix} \quad (15)$$

Donde en el espacio de la transformada \mathbb{Z} , $\mathbf{W}(\alpha(z))$ se define como:

$$W[\alpha(z)] = \bar{\alpha}(z) = \left[\bar{\alpha}_0(z) \quad \bar{\alpha}_1(z) \quad \dots \quad \bar{\alpha}_{N-1}(z) \right]^T \quad (16)$$

Donde

$$\bar{\alpha}_i(z) = \sum_{j=0}^{\infty} \alpha_{N_{j+i}} \cdot z^{-j} \quad (17)$$

Además, el operador adjunto se define como:

$$W^{-1} \left(\begin{bmatrix} \overline{\alpha_0}(z) & \overline{\alpha_1}(z) & \dots & \overline{\alpha_{N-1}}(z) \end{bmatrix}^T \right) = \sum_{i=0}^{N-1} z^{-i} \cdot \overline{\alpha}(z^N) \quad (18)$$

El modelado establece que empezando de una aplicación Entrada-Salida definida como $G : \Gamma^{mN} \rightarrow \Gamma^{pN}$ tal y como se muestra en el siguiente diagrama:

$$\begin{array}{ccc} \Gamma^{mN} & \xrightarrow{\overline{G}} & \Gamma^{pN} \\ \uparrow W & & \uparrow W \\ \Gamma^m & \xrightarrow{G} & \Gamma^p \end{array} \quad (19)$$

Donde,

$$\overline{G} = W \cdot G \cdot W^{-1} \quad (20)$$

$$\overline{G} \cdot \Lambda = W \cdot G \cdot W^{-1} \cdot \Lambda = W \cdot G \cdot \Lambda^N \cdot W^{-1} = W \cdot \Lambda^N \cdot G \cdot W^{-1} = \Lambda \cdot W \cdot G \cdot W^{-1} = \Lambda \cdot \overline{G} \quad (21)$$

La demostración mostrada en la Ecuación 21 viene a decir que se tiene un sistema LTI. La relación entre ambas aplicaciones E/S puede verse comparando su acción en el dominio temporal:

$$\left\{ \begin{array}{l} G : \Gamma^m \rightarrow \Gamma^p : (u_0, u_1, \dots, u_N, u_{N+1}, \dots) \rightarrow (y_0, y_1, \dots, y_N, y_{N+1}, \dots) \\ \overline{G} : \Gamma^{mN} \rightarrow \Gamma^{pN} : \left[\begin{array}{c} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix}, \begin{pmatrix} u_N \\ u_{N+1} \\ \vdots \\ u_{2N-1} \end{pmatrix}, \dots \end{array} \right] \rightarrow \left[\begin{array}{c} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{pmatrix}, \begin{pmatrix} y_N \\ y_{N+1} \\ \vdots \\ y_{2N-1} \end{pmatrix}, \dots \end{array} \right] \end{array} \right. \quad (22)$$

Ambas formas preservan las propiedades analíticas y algebraicas. Las siguientes propiedades han de ser resaltadas:

- **Fórmulas de conexión:** Sean G_f y G_g dos sistemas N-periódicos lineales y discretos. La matriz de transferencia \overline{G}_{fg} del sistema LTI equivalente a la conexión serie entre ambos sistemas viene dada por:

$$\overline{G}_{fg}(\overline{z}) = \overline{G}_f(\overline{z}) \cdot \overline{G}_g(\overline{z}) \quad (23)$$

La matriz de transferencia del bucle cerrado, con G_f en la cadena directa y G_g en la realimentación:

$$\overline{G}_{cl}(\overline{z}) = \left[\mathbf{I} + \overline{G}_f(\overline{z}) \cdot \overline{G}_g(\overline{z}) \right]^{-1} \cdot \overline{G}_f(\overline{z}) \quad (24)$$

- **Estabilidad:** El sistema G_f es estable si y sólo si la matriz de transferencia \bar{G}_f es estable. Así mismo, el sistema con realimentación es estable si, y sólo si \bar{G}_g estabiliza internamente a \bar{G}_f .
- **Isometría:** Sea G un sistema N -periódico estable, lineal y discreto. Si \mathcal{H}_2 es el espacio de Hilbert del cuadrado de las secuencias de números reales sumables, G puede considerarse un operador lineal acotado cuya norma es:

$$\|G\| = \sup \{ \|Gu\|_2 : u \in \mathcal{H}_2, \|u\| = 1 \} \quad (25)$$

Dicha norma viene dada por la norma \mathcal{H}_∞ de su matriz función de transferencia $G(z)$, luego:

$$\|G\| = \|G\|_\infty = \sup \{ \bar{\sigma}(G(e^{j\omega})) : \omega \in [0, 2\pi] \} \quad (26)$$

Donde $\bar{\sigma}(M)$ denota el mayor valor singular de M .

Si restringimos el isomorfismo definido por el operador lifting W al espacio de Hilbert \mathcal{H}_2 , se comprueba que W es una isometría, esto es $\|Wu\|_2 = \|u\|_2$. Con ello se tiene:

$$\|\bar{G}(\bar{z})\| = \|WG(\bar{z})W^{-1}\| = \|G(z)\| = \|\bar{G}(\bar{z})\|_\infty \quad (27)$$

El potencial implícito en el isomorfismo que establece la aplicación vía lifting, justifica la notable difusión de la citada técnica, con una larga serie de variantes de la misma en función de qué se “estira” sólo los estados; sólo entradas y salidas; entradas, salidas y estados; etc.

2.2.1.2.2.- Modelado de Thompson

El modelado de Thompson [65], [66], está basado en la técnica de la descomposición vectorial de conexiones de [62]. Concebido para sistemas SISO, cuya única restricción impone que los periodos T_i tengan una relación entera con el meta-periodo y el periodo base.

Parte de la descomposición vectorial de conexiones de los muestreadores de la entrada y salida de un sistema. Se compacta mediante vectores de retardo E^- y adelanto E^+ :

$$\mathbf{E}^- = \begin{bmatrix} 1 & e^{-T_i s} & \dots & e^{-(N_i-1)s} \end{bmatrix}^T \quad (28)$$

$$\mathbf{E}^+ = \begin{bmatrix} 1 & e^{T_i s} & \dots & e^{(N_i-1)s} \end{bmatrix}^T \quad (29)$$

Con $N_i = \bar{T}/T_i$.

En la Figura 18 se puede ver el esquema de la técnica de descomposición vectorial de conexiones.

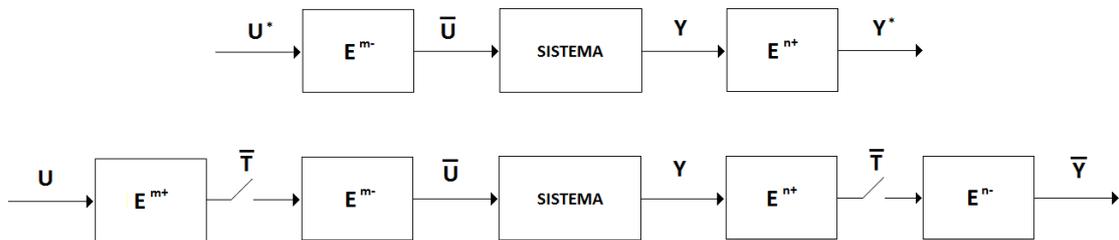


Figura 18. Notación compacta de la descomposición vectorial de conexiones.

Thompson establece el modelado en el espacio de estados del sistema que relaciona los vectores descompuestos \mathbf{U}^* e \mathbf{Y}^* , tal y como se muestra en la Figura 19, que denomina operador de *Kranc*. Dicho operador se define como un sistema digital que trabaja a meta-periodo \bar{T} , siendo:

$$\mathbf{U}^* = \begin{bmatrix} u^T(\bar{k} \cdot \bar{T}) & u^T\left(\bar{k} \cdot \bar{T} + \frac{N}{N_1}\right) & \dots & u^T\left(\bar{k} \cdot \bar{T} + \frac{(N_1-1) \cdot N}{N_1}\right) \end{bmatrix}^T \quad (30)$$

$$\mathbf{Y}^* = \begin{bmatrix} y^T(\bar{k} \cdot \bar{T}) & y^T\left(\bar{k} \cdot \bar{T} + \frac{N}{N_2}\right) & \dots & y^T\left(\bar{k} \cdot \bar{T} + \frac{(N_2-1) \cdot N}{N_2}\right) \end{bmatrix}^T \quad (31)$$

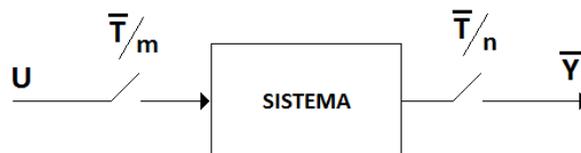


Figura 19. Operador de Kranc.

Dependiendo del tipo de sistema, Kranc define correctamente tres operadores básicos: Kranc Regular, ZOH Kranc y Kranc digital.

2.2.1.2.3.- Bloque multifrecuencia Input/Output (BMIO)

El modelo BMIO [59] establece una idea original entre la representación del sistema en espacio de estados y la obtenida a partir de su función de transferencia discreta. El modelo parte de un sistema LTI SISO continuo:

$$\left. \begin{aligned} x(t) &= \mathbf{A}_p \cdot x(t) + \mathbf{B}_p \cdot u(t) \\ y(t) &= \mathbf{C} \cdot x(t) \end{aligned} \right\} \quad (32)$$

Se consideran dos periodos, el meta-periodo y el periodo base, cuya relación es $\bar{T} = N \cdot T$.

La representación externa se describe a partir de la función de transferencia estrictamente propia a periodo base T .

$$\frac{y(z)}{u(z)} = \frac{\mathbf{B}(z^{-1})}{\mathbf{A}(z^{-1})} \Rightarrow \left\{ \begin{aligned} \mathbf{B}(z^{-1}) &= \sum_{i=0}^n b_i z^{-i}, b_i \neq 0 \\ \mathbf{A}(z^{-1}) &= 1 + \sum_{i=0}^n a_i z^{-i} \end{aligned} \right. \quad (33)$$

Se definen los bloques de entrada y salida expandidos considerando el meta-periodo:

$$\bar{Y}_{\bar{k}} = \begin{bmatrix} y(\bar{k} \cdot \bar{T} + T) \\ y(\bar{k} \cdot \bar{T} + 2T) \\ \vdots \\ y(\bar{k} \cdot \bar{T} + NT) \end{bmatrix} \quad \bar{U}_{\bar{k}} = \begin{bmatrix} u(\bar{k} \cdot \bar{T}) \\ u(\bar{k} \cdot \bar{T} + T) \\ \vdots \\ u(\bar{k} \cdot \bar{T} + (N-1)T) \end{bmatrix} \quad (34)$$

La idea consiste en relacionar las salidas y las entradas con un desfase temporal de un periodo base.

2.2.1.2.4.- Modelado Periódico

En un caso general, se considera un sistema multifrecuencia con m entradas y p salidas como el descrito en la Figura 20. Las entradas del sistema sólo pueden aplicarse en determinados instantes de muestreo dados por $t^{u_v} = \{t_{r_1^v}^{u_v}, t_{r_2^v}^{u_v}, \dots, t_{r_r^v}^{u_v}\}$, con $v=1,2,\dots,m$, dónde el subíndice r denota el número total de actualizaciones de la entrada v . Así mismo las salidas sólo pueden medirse en los instantes de muestreo

dados por $t^{y_\omega} = \{t_{s_1^\omega}^{y_\omega}, t_{s_2^\omega}^{y_\omega}, \dots, t_{s_s^\omega}^{y_\omega}\}$, con $\omega = 1, 2, \dots, p$, donde el subíndice s denota el número total de muestras de la salida ω .

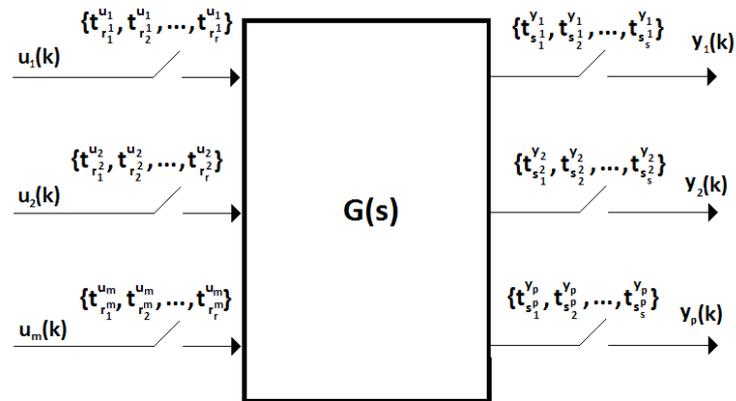


Figura 20. Esquema general del sistema multifrecuencia.

Basándose en las ideas de [67] y [68], se define una matriz conforme al patrón de muestreo del sistema tanto para la entrada como para la salida.

El modelado mediante esta técnica requiere de una ampliación del estado del sistema discretizado a periodo base, mediante dos nuevos vectores asociados con los mecanismos de retención de entrada y muestreo de salida, respectivamente. La idea principal es utilizar las matrices $\Delta_u(k)$ y $\Delta_y(k)$, para conmutar entre las entradas/salidas muestreadas o los vectores auxiliares de retención, que contienen información al respecto de entradas/salidas muestreadas en instantes anteriores.

La principal ventaja de esta técnica de modelado es que utiliza un modelo a periodo base y, por tanto, está orientada a sistemas de tiempo real, en los que se requiere la información procesada al periodo de muestreo rápido. Mediante el uso de esta técnica se pueden diseñar los diversos elementos de un bucle de control: controladores, retenedores, observadores multifrecuencia, etc.

2.2.2.-High-Order-Holds y High-Order-Samplers multifrecuencia.

Como se ha mencionado en la sección anterior, los sistemas muestreados reales están compuestos por sistemas continuos como es la planta, cuyas señales son muestreadas mediante convertidores D/A y A/D (retenedores y muestreadores), y sistemas discretos como son los controladores, estimadores, etc. Los retenedores y los muestreadores actúan como interfaces entre los dispositivos continuos y los discretos introduciendo el periodo de muestreo de cada sistema.

Los retenedores actúan como interfaz para generar una señal continua dado una secuencia discreta a periodo T :

$$u_h(t) = H_T \{u(k)\}, \quad t \in [kT, (k+1)T] \quad (35)$$

Por otra parte, los muestreadores convierten una señal continua en otra discreta:

$$y_s(t) = S_T y(t) \rightarrow y_s(k) \quad (36)$$

El retenedor más utilizado y por ello más conocido es el Retenedor de Orden Cero (ZOH), el cuál toma un valor de la señal de entrada y lo mantiene hasta el siguiente periodo. El retenedor ZOH implementa la función de transferencia de un convertidor D/A a periodo T :

$$G_{ZOH}(s) = \frac{1 - e^{-sT}}{s} \quad (37)$$

Del mismo modo, la captura ideal cada periodo de una señal de entrada en un tiempo infinitesimal genera una secuencia discreta:

$$\begin{cases} y_s(t) = \delta_T(t) y(t) \\ \delta_T(t) = \begin{cases} 1 & \text{if } \text{mod}(t, T) = 0 \\ 0 & \text{in other cases} \end{cases} \end{cases} \quad (38)$$

Hay diferentes modelos par los sistemas muestreados multifrecuencia, los cuales se van a describir a continuación. Estos modelos se clasifican de la siguiente forma:

- Si se trabaja en la relación Entrada/Salida se le denominan “*Modelos en Representación Externa*”.
- Si se trabaja con las variables de estado se le denomina “*Modelos en Representación Interna*” [60].

2.2.2.1.- Retenedor Bifrecuencia

En la Figura 21 se muestra el esquema de un retenedor Bi-frecuencia. La señal de entrada es medida a baja frecuencia (una vez cada meta-periodo), mientras que la señal de salida se aplica N veces más rápido (a periodo base T). Se va a emplear la siguiente notación para resaltar que se está trabajando a distintos periodos: $u(\bar{k}, i) = u(k)$. El primer índice, \bar{k} , es para los muestreos a baja frecuencia mientras que el segundo índice i , es para los muestreos a periodo rápido.

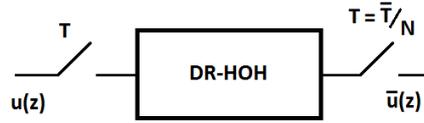


Figura 21. Retenedor de Orden Alto Bi-frecuencia (DR-HOH).

En general, un retenedor expande la señal de entrada:

$$\bar{u}(\bar{k}) = u(\bar{k}, 0) \rightarrow \begin{pmatrix} u_h(\bar{k}, 0) \\ u_h(\bar{k}, 1) \\ \vdots \\ u_h(\bar{k}, N-1) \end{pmatrix} = \bar{u}_h(\bar{k}) \quad (39)$$

Su función de transferencia es:

$$\begin{bmatrix} \bar{U}_0(\bar{z}) \\ \bar{U}_1(\bar{z}) \\ \vdots \\ \bar{U}_{N-1}(\bar{z}) \end{bmatrix} = \begin{bmatrix} \bar{G}_0(\bar{z}) \\ \bar{G}_1(\bar{z}) \\ \vdots \\ \bar{G}_{N-1}(\bar{z}) \end{bmatrix} \cdot U(\bar{z}) \quad (40)$$

Con:

$$\bar{U}_i(\bar{z}) = \sum_{k=0}^{\infty} u_h(\bar{k}, i) \cdot \bar{z}^{-k}, \quad \bar{z} = e^{s\bar{T}} \quad (41)$$

Por otra parte, es posible expresar un retenedor en espacio de estados, tanto para el modelado con el lifting como para el modelado periódico. Así, utilizando la técnica del lifting, se obtienen matrices con coeficientes constantes:

$$\bar{v}_u(\bar{k}+1) = \bar{A}_h \cdot \bar{v}_u(\bar{k}) + \bar{B}_h \cdot \bar{u}(\bar{k}) \quad (42)$$

$$\bar{u}_h(\bar{k}) = \bar{C}_h \cdot \bar{v}_u(\bar{k}) + \bar{D}_h \cdot \bar{u}(\bar{k}) \quad (43)$$

Mediante modelado periódico se obtienen las siguientes matrices variables con el tiempo:

$$v_u(k+1) = A_h \cdot v_u(k) + B_h \cdot u(k) \quad (44)$$

$$u_h(\bar{k}) = C_h \cdot v_u(k) + D_h \cdot u(k) \quad (45)$$

En [69], [70] y [71] se describen los DR-HOH a partir de los clásicos retenedores comunes monofrecuencia:

- Retenedor Impulso Bi-frecuencia (DR-IH): Este retenedor genera una señal de valor N veces superior a la muestreada durante el primer periodo base, mientras que el resto de periodos la salida vale 0.

$$u_h(\bar{k}, i) = u(\bar{k}, 0) \cdot N \cdot \delta_i \rightarrow \bar{G}_h(\bar{z}) = \begin{cases} N \cdot I & \text{sii } i = 0 \\ \mathbf{0} & \text{sii } i \neq 0 \end{cases} \quad (46)$$

Para $i = 0, 1, \dots, N - 1$.

- Retenedor de Orden Cero Bi-frecuencia (DR-ZOH): Este retenedor genera una secuencia de salida de igual valor a la entrada muestreada durante el primer instante.

$$u_h(\bar{k}, i) = u(\bar{k}, 0) \rightarrow \bar{G}_h(\bar{z}) = I \quad (47)$$

Para $i = 0, 1, \dots, N - 1$.

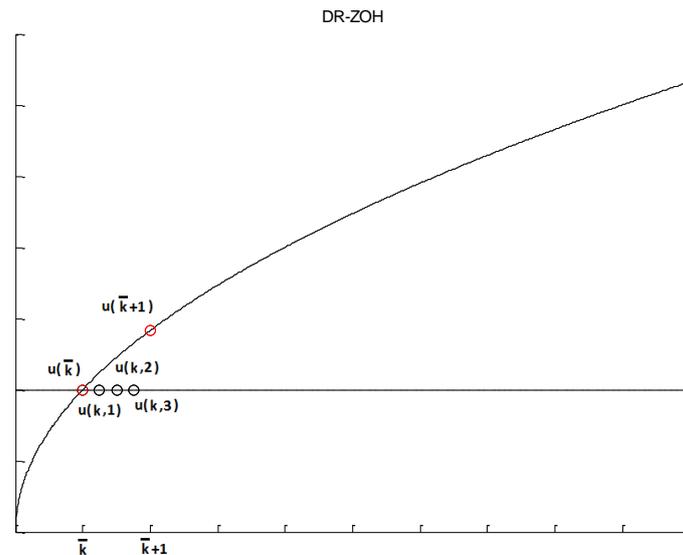


Figura 22. Retenedor bifrecuencia de orden cero (DR-ZOH).

- Retenedor de Orden Uno Bi-frecuencia (DR-FOH): Este retenedor genera una secuencia de salida con forma de rampa que depende de la muestra capturada a la entrada y de la anterior.

$$\left. \begin{aligned} u_h(\bar{k}, i) &= u(\bar{k}, 0) + \frac{i}{N} [u(\bar{k}, 0) - u(\bar{k} - 1, 0)] \\ \bar{G}_h(\bar{z}) &= \mathbf{I} + \frac{i}{N} [1 - \bar{z}^{-1}] \cdot \mathbf{I} = \frac{\frac{N+i}{N} \cdot \bar{z} \cdot \mathbf{I} - \frac{i}{N} \cdot \mathbf{I}}{\bar{z}} \end{aligned} \right\} \quad (48)$$

Para $i = 0, 1, \dots, N - 1$.

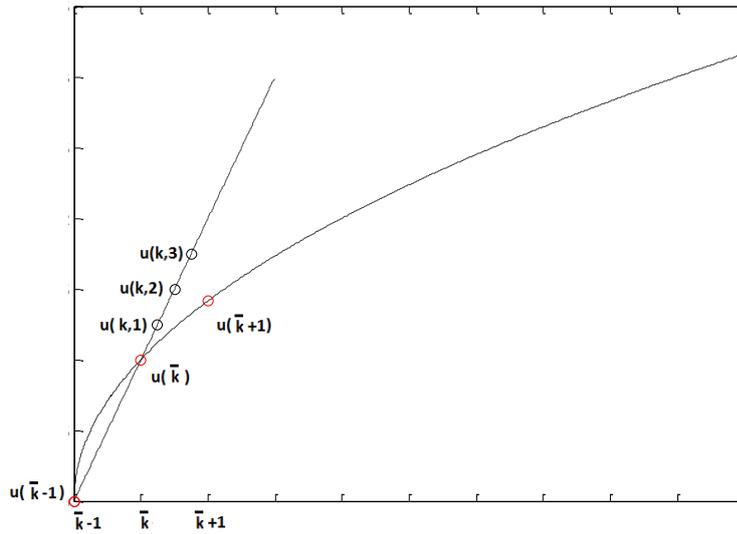


Figura 23. Retenedor bifrecuencia de primer orden (DR-FOH).

- Retenedor Triangular Bi-frecuencia (DR-TH): El retenedor emula el comportamiento de un retenedor triangular, generando una secuencia de salida con forma de rampa similar a la del DR-FOH.

$$\left. \begin{aligned} u_h(\bar{k}, i) &= u(\bar{k} - 1, 0) + \frac{i}{N} [u(\bar{k}, 0) - u(\bar{k} - 1, 0)] \\ \bar{G}_h(\bar{z}) &= z \cdot \mathbf{I} + \frac{i}{N} [1 - \bar{z}^{-1}] \cdot \mathbf{I} = \frac{\frac{i}{N} \cdot \bar{z} \cdot \mathbf{I} - \frac{N-i}{N} \cdot \mathbf{I}}{\bar{z}} \end{aligned} \right\} \quad (49)$$

Para $i = 0, 1, \dots, N - 1$.

- Retenedor de Segundo Orden Bi-frecuencia (DR-SOH): Este retenedor genera la secuencia de salidas conforme a una función polinómica de segundo orden.

$$\begin{aligned} u_h(\bar{k}, i) &= \left(1 + \frac{3 \cdot i}{2 \cdot N} + \frac{1}{2} \left(\frac{i}{N}\right)^2\right) u(\bar{k}) + \left(\frac{2 \cdot i}{N} + \left(\frac{i}{N}\right)^2\right) u(\bar{k} - 1, 0) \\ &\quad + \frac{1}{2} \cdot \left(\frac{i}{N} + \left(\frac{i}{N}\right)^2\right) u(\bar{k} - 2, 0) \end{aligned} \quad (50)$$

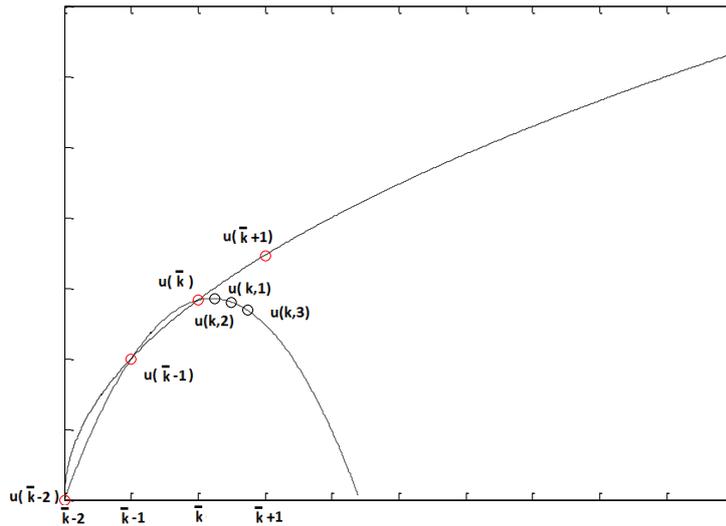


Figura 24. Retenedor bifrecuencia de primer orden (DR-SOH).

En este punto hay que hablar también de las distintas posibilidades de representación que tienen los DR-HOH. La Figura 25 muestra todas las representaciones que se han utilizado en el caso concreto del retenedor bifrecuencia de primer orden.

Especial importancia en este caso tienen las representaciones algorítmicas, que permiten el desarrollo en implementación real y la representación $G_h(z, z^N)$ que permite una rápida implementación en herramientas tales como Simulink [69]. Ambas formas se han utilizado en el desarrollo del estudio planteado en esta tesina y se puede ver su empleo en el Anexo 1.

$$\begin{array}{ccc}
 \begin{array}{l}
 \text{Discrete - Algorithm} \\
 u(kN + i) = u(kN) + \frac{i}{N} \cdot (u(kN) - u((k-1)N)) \\
 \uparrow \quad t = i \cdot T \\
 \quad kT \rightarrow k
 \end{array}
 &
 \begin{array}{c}
 \xrightarrow{Z_T} \\
 \xleftarrow{Z_T^{-1}}
 \end{array}
 &
 \begin{array}{l}
 G_{h,i}(z^N) = \frac{\left(\frac{N+i}{N}\right) \cdot z^N - \left(\frac{i}{N}\right)}{z^N} \\
 \uparrow \\
 \text{Lifting} \quad G_h(z, z^N) = \sum_{i=0}^{N-1} G_{h,i}(z^N) \cdot z^{-i} \\
 \downarrow \\
 \uparrow \quad \downarrow
 \end{array}
 \end{array}$$

$$\begin{array}{ccc}
 \begin{array}{l}
 u(kNT + t) = u(kNT) + \frac{t}{T} \cdot (u(kNT) - u((k-1)NT)) \\
 \uparrow \quad L^{-1} \\
 G(s) = (1 - e^{-NTs})^2 \cdot \left(\frac{1}{s} + \frac{1}{NT \cdot s^2}\right)
 \end{array}
 &
 \begin{array}{c}
 Z_T[G_{NT}^*(s)] \\
 \rightarrow
 \end{array}
 &
 \begin{array}{l}
 G_h(z, z^N) = \left(\frac{z^N - 1}{z - 1}\right)^2 \cdot \frac{z^2 - \left(\frac{N-1}{N}\right) \cdot z}{z^{2N}}
 \end{array}
 \end{array}$$

Figura 25. Representaciones de los DR-HOH (ejemplo con el DR-FOH).

La Tabla 2 muestra los retenedores anteriormente mencionados en sus representaciones más comunes:

	RETENEDOR MULTI-FRECUENCIA A T $G_{h,i}(z^N)$	ALGORITMO DE CONTROL A T	$G_h(s)$ A NT	RETENEDOR MULTI-FRECUENCIA A NT $G_h(z, z^N)$
IMPULSO	$G_{h,i}(z^N) = \begin{cases} N & \text{iff } i=0 \\ 0 & \text{otherwise} \end{cases}$	$e(k, i) = e(k, 0) \cdot \frac{1}{T} \cdot \delta_i$	1	1
ZOH	$G_{h,i}(z^N) = 1$	$e(k, i) = e(k, 0)$	$\frac{1 - e^{-NTs}}{s}$	$\left(\frac{z^N - 1}{z^N}\right) \cdot \left(\frac{z}{z-1}\right)$
FOH	$G_{h,i}(z^N) = \frac{\left(\frac{N+i}{N}\right)z^N - \left(\frac{i}{N}\right)}{z^N}$	$e(k, i) = e(k, 0) + \frac{i}{N} \cdot [e(k, 0) - e(k-1, 0)]$	$(1 - e^{-NTs})^2 \cdot \left(\frac{1}{s} + \frac{1}{NTs^2}\right)$	$\left(\frac{z^N - 1}{z^N}\right)^2 \cdot \frac{z^2 - \left(\frac{N-1}{N}\right) \cdot z}{(z-1)^2}$
TH	$G_{h,i}(z^N) = \frac{\left(\frac{i}{N}\right)z^N + \left(\frac{N-i}{N}\right)}{z^N}$	$e(k, i) = e(k-1, 0) + \frac{i}{N} \cdot [e(k, 0) - e(k-1, 0)]$	$(1 - e^{-NTs})^2 \cdot \left(\frac{1}{NTs^2}\right)$	$\left(\frac{z^N - 1}{z^N}\right)^2 \cdot \frac{z}{N \cdot (z-1)^2}$

Tabla 2. Representaciones de los retenedores de orden alto multifrecuencia.

2.2.2.2.- Muestreador Bifrecuencia

Los muestreadores Bifrecuencia (DR-HOS) [69] trabajan con señales muestreadas a periodo rápido en su entrada (periodo base) y proporcionan señales de salida muestreadas a periodo lento (meta-periodo).

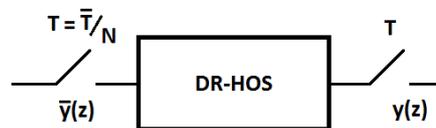


Figura 26. Muestreador de Orden Alto Bi-frecuencia (DR-HOS).

Los muestreadores compactan información de N entradas en una sola salida:

$$\bar{y}(\bar{k}) = \begin{pmatrix} y(\bar{k}, 0) \\ y(\bar{k}, 1) \\ \vdots \\ y(\bar{k}, N-1) \end{pmatrix} \rightarrow y_s(\bar{k}+1, 0) \quad (51)$$

Siendo la función de transferencia de los DR-HOS:

$$Y_0(\bar{z}) = \begin{bmatrix} \bar{G}_0(\bar{z}) & \bar{G}_1(\bar{z}) & \dots & \bar{G}_{N-1}(\bar{z}) \end{bmatrix} \cdot \begin{bmatrix} \bar{Y}_0(\bar{z}) \\ \bar{Y}_1(\bar{z}) \\ \vdots \\ \bar{Y}_{N-1}(\bar{z}) \end{bmatrix} \cdot U(\bar{z}) \quad (52)$$

Se pueden encontrar diferentes modelos de muestreadores, se va a describir muestreador de orden cero, por ser el más fácil y utilizado en la práctica.

$$y_s(\bar{k}+1, 0) = y(\bar{k}, 0) \quad (53)$$

$$\bar{G}_i(\bar{z}) = \begin{cases} I & \text{iff } i=0 \\ 0 & \text{in other cases} \end{cases} \quad (54)$$

Otro muestreador, un poco más complicado que el anterior, es el que realiza la media ponderada de las muestras:

$$y_s(\bar{k}+1, 0) = \sum_{i=0}^{N-1} y(\bar{k}, i) \cdot \left(\frac{\alpha_i}{\sum_{j=0}^{N-1} \alpha_j} \right) \quad (55)$$

$$\bar{G}_i(\bar{z}) = \left(\frac{\alpha_i}{\sum_{j=0}^{N-1} \alpha_j} \right) \cdot \bar{z}^{-1} \quad (56)$$

2.2.3.-Modelo de la planta Bifrecuencia

En la anterior sección se ha visto que los sistemas reales están compuestos por elementos continuos y elementos discretos interconectados por los retenedores y los muestreadores.

Los retenedores y muestreadores bi-frecuencia combinados con modelos mono-frecuencia de elementos a alta (periodo T) y baja frecuencia (periodo NT) generan modelos equivalentes bi-frecuencia de la planta, controlador, estimador, etc.

Desde el punto de vista bi-frecuencia, se pueden producir dos situaciones diferentes: *los N-delay Input* y *los N-delay Output*. En el caso del *N-delay input*, la salida se muestrea cada instante NT y la entrada se actualiza cada instante T (es decir, N veces más rápida que la salida). En el caso del *N-delay output* ocurre lo contrario, la entrada se actualiza cada instante NT mientras que la salida se actualiza cada instante T , es decir N veces más rápido.

El modelo bi-frecuencia discreto equivalente del modelo *N-delay input* se describe como:

$$Y(z^N) = \begin{bmatrix} G_0(z^N) & \dots & G_{N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} U_0(z^N) \\ \vdots \\ U_{N-1}(z^N) \end{bmatrix} \quad (57)$$

Donde se tiene que:

$$\begin{aligned} & \begin{bmatrix} G_0(z^N) & \dots & G_{N-1}(z^N) \end{bmatrix} = \\ & = \begin{bmatrix} G_{s,0}(z^N) & \dots & G_{s,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} G_{0,0}(z^N) & \dots & G_{0,N-1}(z^N) \\ \vdots & & \vdots \\ G_{N-1,0}(z^N) & \dots & G_{N-1,N-1}(z^N) \end{bmatrix} \end{aligned} \quad (58)$$

$$G_i(z^N) = \sum_{j=0}^{N-1} G_{s,j}(z^N) \cdot G_{j,i}(z^N) \quad (59)$$

Cuando se genera el modelo *N-delay input* de la planta, se es mucho más restrictivo en el uso de los muestreadores bi-frecuencia que cuando se discretizan controladores continuos. De hecho, el muestreador elegido debe reproducir el proceso de medida del convertidor A/D.

Además, asumiendo que se tiene un modelo continuo de la planta como el descrito por:

$$G_p(s) \quad \leftrightarrow \quad \begin{cases} \dot{x}_p(t) = A_p^c \cdot x_p(t) + B_p^c \cdot u(t) \\ y(t) = C_p \cdot x_p(t) + D_p \cdot u(t) \end{cases} \quad (60)$$

Por lo tanto, el modelo *N-delay input* es:

$$\left. \begin{aligned} G_{p,i}(z^N) &= C_p \cdot (z^N \cdot I - A_p^N)^{-1} \cdot A_p^{N-i-1} \cdot B_p + D_p \cdot \delta_i \\ \text{con } \delta_i &= 1 \text{ para } i = 0 \text{ y } \delta_i = 0 \text{ para } i = 1, 2, \dots, N-1 \end{aligned} \right\} \quad (61)$$

La función de transferencia anterior tiene unas ecuaciones de estado y de salida asociadas:

$$\left. \begin{aligned} x_p(k+1,0) &= A_p^N \cdot x_p(k,0) + \sum_{i=0}^{N-1} A_p^{N-i-1} \cdot B_p \cdot u(k,i) \\ y(k,0) &= C_p \cdot x_p(k,0) + D_p \cdot u(k,0) \\ \text{con } \sum_{i=i1}^{i2} &= 0 \quad \text{if } i2 < i1 \end{aligned} \right\} \quad (62)$$

Donde se definen A_p y B_p como:

$$A_p = e^{A_p^c T}, \quad B_p = \int_0^T e^{A_p^c(T-\tau)} B_p^c d\tau \quad (63)$$

Por otro lado, para el caso del N-delay output, el vector función de transferencia de la planta es:

$$\begin{bmatrix} Y_0(z^N) \\ \vdots \\ Y_{N-1}(z^N) \end{bmatrix} = \begin{bmatrix} G_0(z^N) \\ \vdots \\ G_{N-1}(z^N) \end{bmatrix} \cdot U(z^N) \quad (64)$$

Donde se tiene que:

$$\begin{bmatrix} G_0(z^N) \\ \vdots \\ G_{N-1}(z^N) \end{bmatrix} = \begin{bmatrix} G_{0,0}(z^N) & \cdots & G_{0,N-1}(z^N) \\ \vdots & & \vdots \\ G_{N-1,0}(z^N) & \cdots & G_{N-1,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} G_{h,0}(z^N) \\ \vdots \\ G_{h,N-1}(z^N) \end{bmatrix} \quad (65)$$

La expresión anterior puede representarse de forma más compacta como se muestra a continuación:

$$G_i(z^N) = \sum_{j=0}^{N-1} G_{i,j}(z^N) \cdot G_{h,j}(z^N) \quad (66)$$

De nuevo, para el modelo de la planta *N-delay output*, se tiene una mayor restricción en el uso de retenedores clásicos que actúan como convertidores D/A. Por

ejemplo, el modelo de la planta con un retenedor de orden cero bi-frecuencia (DR-ZOH) se convierte en:

$$G_{p,i}(z^N) = C_p \cdot A_p^i (z^N \cdot I - A_p^N)^{-1} \cdot \sum_{j=0}^{N-1} A_p^{N-j-1} \cdot B_p + C_p \cdot \sum_{j=0}^{i-1} A_p^{i-j-1} \cdot B_p + D_p \quad (67)$$

Las ecuaciones de estado asociadas son:

$$\begin{cases} x_p(k+1,0) = A_p^N \cdot x_p(k,0) + \sum_{j=0}^{N-1} A_p^{N-j-1} \cdot B_p \cdot u(k,0) \\ y(k,i) = C_p \cdot A_p^i \cdot x_p(k,0) + C_p \cdot \sum_{j=0}^{i-1} A_p^{i-j-1} \cdot B_p \cdot u(k,0) + D_p \cdot u(k,0) \end{cases} \quad (68)$$

La anterior expresión puede simplificarse, dando el siguiente resultado:

$$G_{p,i}(z^N) = C_p \cdot (z^N \cdot I - A_p^N)^{-1} \cdot \left\{ \sum_{j=i}^{N-1} A_p^j + z^N \cdot \sum_{j=0}^{i-1} A_p^j \right\} \cdot B_p + D_p \quad (69)$$

2.2.4.-Controladores equivalentes de orden alto Bifrecuencia

Los controladores bi-frecuencia se calculan a partir de la función de transferencia continua del controlador y uno de los retenedores de orden alto bi-frecuencia, tal y como se ha explicado en la sección anterior.

Se va a considerar el caso general, asumiendo que los controladores continuos pueden separarse en dos términos con diferente dinámica, por lo que así se seleccionan diferentes periodos de muestreo: el muestreo a frecuencia alta (periodo base T) y el muestreo a frecuencia baja (a meta-periodo $\bar{T} = NT$).

$$G_c(s) = G_{c,HF}(s) \cdot G_{c,LF}(s) \quad (70)$$

En este caso, el controlador bi-frecuencia equivalente se obtiene a partir de tres elementos: controlador monofrecuencia a alta frecuencia (periodo base T), un retenedor de orden alto bi-frecuencia y un controlador monofrecuencia a baja frecuencia (meta-periodo $\bar{T} = NT$). El retenedor de orden alto cumple principalmente con dos funciones:

- Sirve de interfaz entre las partes de alta y de baja frecuencia del controlador.
- Estimación de valores intermuestreo.

La expresión general de un controlador bi-frecuencia es:

$$\left. \begin{aligned}
 \begin{bmatrix} U_0(z^N) \\ \vdots \\ U_{N-1}(z^N) \end{bmatrix} &= \begin{bmatrix} G_{c,DR,0}(z^N) \\ \vdots \\ G_{c,DR,N-1}(z^N) \end{bmatrix} \cdot E(z^N) \\
 \text{where} \\
 \begin{bmatrix} G_{c,DR,0}(z^N) \\ \vdots \\ G_{c,DR,N-1}(z^N) \end{bmatrix} &= \\
 &= \begin{bmatrix} G_{c,HF,0,0}(z^N) & \cdots & G_{c,HF,0,N-1}(z^N) \\ \vdots & & \vdots \\ G_{c,HF,N-1,0}(z^N) & \cdots & G_{c,HF,N-1,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} G_{h,0}(z^N) \\ \vdots \\ G_{h,N-1}(z^N) \end{bmatrix} \cdot G_{c,LF}(z^N) \\
 \text{or} \\
 G_{c,DR}(z^N) &= G_{c,HF}(z^N) \cdot G_{h,DR}(z^N) \cdot G_{c,LF}(z^N)
 \end{aligned} \right\} (71)$$

Donde

- $G_{c,DR}(z^N)$ es la función de transferencia del controlador bi-frecuencia.
- $G_{c,HF}(z^N)$ es el modelo “estirado” de la función de transferencia de la parte rápida (alta frecuencia) del controlador.
- $G_{h,DR}(z^N)$ es la función de transferencia del retenedor de orden alto bi-frecuencia.
- $G_{c,LF}(z^N)$ es la función de transferencia de la parte lenta (baja frecuencia) del controlador.

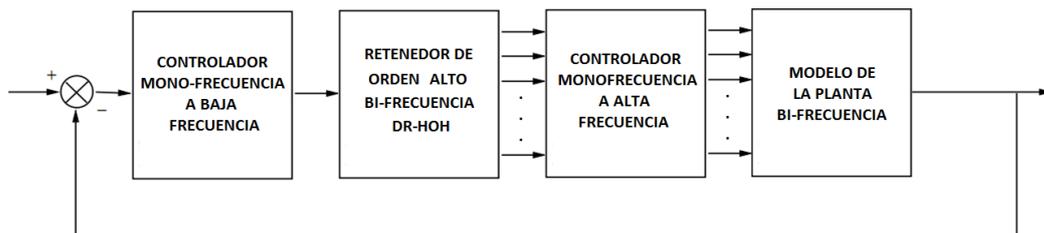


Figura 27. Controlador general bifrecuencia.

El esquema general de un controlador bi-frecuencia se muestra en la Figura 27. La función de transferencia monofrecuencia “estirada” del controlador a alta frecuencia se describe como:

$$G_{c,HFi,j}(z^N) = C_{c,HF} \cdot A_{c,HF}^i \cdot (z^N \cdot I - A_{c,HF}^N)^{-1} \cdot A_{c,HF}^{N-j-1} \cdot B_{c,HF} + D_{i,j}$$

$$\text{with } D_{i,j} = \begin{cases} C_{c,HF} \cdot A_{c,HF}^{i-j-1} \cdot B_{c,HF} & \text{iff } i > j \\ D_{c,HF} & \text{iff } i = j \\ 0 & \text{otherwise} \end{cases} \quad (72)$$

Donde $\begin{Bmatrix} A_{c,HF} & B_{c,HF} \\ C_{c,HF} & D_{c,HF} \end{Bmatrix}$ son las matrices del espacio de estados del controlador equivalente a alta frecuencia, $G_{c,HF}(z)$, obtenidas mediante técnicas de discretización.

A partir del caso general, la introducción de un estimador bi-frecuencia de orden alto provoca las siguientes dos situaciones extremas:

- El enfoque llamado “estimación de la señal de control”
- El enfoque llamado “estimación de la señal de error”

Hay que tener en cuenta que los DR-HOH siempre generan N señales de salida a partir de una señal de entrada. Cuando la señal de entrada es el error, se le llama estimación de la señal de error y cuando es la de control se le llama estimación de la señal de control.

2.2.4.1.- Estimación de la señal de control

El estimador de la señal de control evalúa las señales de control a altas frecuencias dada una señal proporcionada por un controlador a baja frecuencia. Esto es:

$$\begin{bmatrix} U_0(z^N) \\ \vdots \\ U_{N-1}(z^N) \end{bmatrix} = \begin{bmatrix} G_{c,0}(z^N) \\ \vdots \\ G_{c,N-1}(z^N) \end{bmatrix} \cdot E(z^N) \quad (73)$$

$$\begin{bmatrix} G_{c,0}(z^N) \\ \vdots \\ G_{c,N-1}(z^N) \end{bmatrix} = \begin{bmatrix} G_{h,0}(z^N) \\ \vdots \\ G_{h,N-1}(z^N) \end{bmatrix} \cdot G_{c,LF}(z^N) \quad (74)$$

De este modo se tiene que $G_{c,i}(z^N) = G_{h,i}(z^N) \cdot G_{c,LF}(z^N)$. Es interesante ver que en este caso, el retenedor bi-frecuencia de orden alto solamente afecta a la planta. Además, el sistema bi-frecuencia en bucle cerrado se puede analizar como un sistema discreto monofrecuencia a baja frecuencia. Esto es,

$$\left. \begin{aligned}
 Y(z^N) &= \begin{bmatrix} G_{p,0}(z^N) & \cdots & G_{p,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} G_{h,0}(z^N) \\ \vdots \\ G_{h,N-1}(z^N) \end{bmatrix} \cdot G_{c,LF}(z^N) \cdot E(z^N) = \\
 &= \bar{G}_p(z^N) \cdot G_{c,LF}(z^N) \cdot E(z^N) \\
 \text{where} \\
 \bar{G}_p(z^N) &= \sum_{i=0}^{N-1} G_{p,i}(z^N) \cdot G_{h,i}(z^N)
 \end{aligned} \right\} (75)$$

En este enfoque, el controlador puede calcularse discretizando el controlador continuo, previamente diseñado en el dominio continuo, o diseñado en el dominio discreto desde un modelo discreto monofrecuencia de la planta después de haber aplicado el retenedor de orden alto bi-frecuencia.

2.2.4.2.- Estimación de la señal de error

En el enfoque de la estimación de la señal de error, el estimador se incluye para evaluar la señal de error entre dos instantes de muestreo, luego se tiene:

$$\left. \begin{aligned}
 \begin{bmatrix} U_0(z^N) \\ \vdots \\ U_{N-1}(z^N) \end{bmatrix} &= \begin{bmatrix} G_{c,0}(z^N) \\ \vdots \\ G_{c,N-1}(z^N) \end{bmatrix} \cdot E(z^N) \\
 \text{where} \\
 \begin{bmatrix} G_{c,0}(z^N) \\ \vdots \\ G_{c,N-1}(z^N) \end{bmatrix} &= \begin{bmatrix} G_{c,HF,0,0}(z^N) & \cdots & G_{c,HF,0,N-1}(z^N) \\ \vdots & & \vdots \\ G_{c,HF,N-1,0}(z^N) & \cdots & G_{c,HF,N-1,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} G_{h,0}(z^N) \\ \vdots \\ G_{h,N-1}(z^N) \end{bmatrix}
 \end{aligned} \right\} (76)$$

En este caso, el controlador multifrecuencia se calcula combinando un retenedor de orden alto bi-frecuencia con el controlador discreto equivalente a alta frecuencia. Esto es:

$$\begin{aligned}
 G_{c,i}(z^N) &= \sum_{j=0}^{N-1} G_{c,HF,i,j}(z^N) \cdot G_{h,j}(z^N) = \\
 &= C_{c,HF} \cdot A_{c,HF}^i \cdot (z^N \cdot I - A_{c,HF}^N)^{-1} \cdot \sum_{j=0}^{N-1} A_{c,HF}^{N-j-1} \cdot B_{c,HF} \cdot G_{h,j}(z^N) + \\
 &\quad + C_{c,HF} \cdot \sum_{j=0}^{i-1} A_{c,HF}^{i-j-1} \cdot B_{c,HF} \cdot G_{h,j}(z^N) + D_{c,HF} \cdot G_{h,i}(z^N)
 \end{aligned} \quad (77)$$

Nota 1. La combinación de un modelo de la planta bi-frecuencia con un controlador bi-frecuencia genera un modelo monofrecuencia trabajando a baja frecuencia. La función de transferencia en bucle abierto es:

$$G_{open-loop,N}(z^N) = \sum_{i=0}^{N-1} G_{p,i}(z^N) \cdot G_{c,i}(z^N) \quad (78)$$

Nota 1. Es interesante analizar el efecto de los retenedores de orden alto bi-frecuencia en el límite, es decir cuándo $N \rightarrow \infty$. Esto es para periodo constante $\bar{T} = NT$, siendo baja frecuencia y así $\bar{z} = \exp(\bar{T}s)$. La Tabla 3 muestra las correspondientes situaciones límite tanto para el enfoque de la estimación de la señal de control como para el enfoque de la señal de error².

	RETENEDOR MULTI-FRECUENCIA $G_h(s)$ at \bar{T}	FUNCIÓN DE TRANSFERENCIA EN BUCLE ABIERTO CON ESTIMACIÓN DE LA SEÑAL DE CONTROL, $N \rightarrow \infty$	FUNCIÓN DE TRANSFERENCIA EN BUCLE ABIERTO CON ESTIMACIÓN DE LA SEÑAL DE LA SEÑAL DE ERROR, $N \rightarrow \infty$
IMPULSO	1	$Z_T[G_p(s)] \cdot G_c(z)$	$Z_{\bar{T}}[G_p(s) \cdot G_c(s)]$
ZOH	$\frac{1-e^{-\bar{T}s}}{s}$	$\left(\frac{\bar{z}-1}{\bar{z}}\right) \cdot Z_{\bar{T}}\left[\frac{G_p(s)}{s}\right] \cdot G_c(\bar{z})$	$\left(\frac{\bar{z}-1}{\bar{z}}\right) \cdot Z_{\bar{T}}\left[\frac{G_p(s) \cdot G_c(s)}{s}\right]$
FOH	$(1-e^{-\bar{T}s})^2 \cdot \left(\frac{1}{s} + \frac{1}{\bar{T}s^2}\right)$	$\left(\frac{\bar{z}-1}{\bar{z}}\right)^2 \cdot Z_{\bar{T}}\left[\frac{G(s)}{s} + \frac{G(s)}{\bar{T}s^2}\right] \cdot G_c(\bar{z})$	$\left(\frac{\bar{z}-1}{\bar{z}}\right)^2 \cdot Z_{\bar{T}}\left[\frac{G_p(s) \cdot G_c(s)}{s} + \frac{G_p(s) \cdot G_c(s)}{\bar{T}s^2}\right]$
TH	$(1-e^{-\bar{T}s})^2 \cdot \left(\frac{1}{\bar{T}s^2}\right)$	$\left(\frac{\bar{z}-1}{\bar{z}}\right)^2 \cdot Z_{\bar{T}}\left[\frac{G(s)}{\bar{T}s^2}\right] \cdot G_c(\bar{z})$	$\left(\frac{\bar{z}-1}{\bar{z}}\right)^2 \cdot Z_{\bar{T}}\left[\frac{G_p(s) \cdot G_c(s)}{\bar{T}s^2}\right]$

Tabla 3. Función de transferencia de los retenedores de orden alto para el caso límite $N \rightarrow \infty$.

² Cabe decir que para una mejor comprensión de los conceptos multifrecuencia mostrados en esta sección, se ha adjuntado un anexo con un ejemplo sencillo dónde el lector puede ver más detalladamente y paso por paso cada uno de ellos (ANEXO 1).

2.2.5.-Lugar de las raíces Multifrecuencia

En la anterior sección se ha hablado del efecto que producen sobre el sistema los retenedores de orden alto en los límites, cuando $N \rightarrow \infty$, y esto permite extender el concepto que se tiene en teoría clásica de control de sistemas lineales de *Lugar de las raíces*.

En teoría de control se define el concepto de lugar de las raíces como el lugar geométrico donde se sitúan los polos y los ceros de una función de transferencia. Este método permite el estudio de la dinámica de sistemas SISO en lazo cerrado, así como el diseño de controladores estudiando la estabilidad y dinámica que estos introducen en dicho sistema. Si se está trabajando en el dominio continuo, el lugar de las raíces presenta la forma mostrada en la Figura 28. En ella se puede diferenciar la zona de estabilidad (zona pintada de blanco) y la zona de inestabilidad (zona pintada de amarillo). Además, el sistema tiene una dinámica más rápida cuando tiene sus polos lo más alejados del eje 0 y tiene una dinámica más lenta en caso contrario.

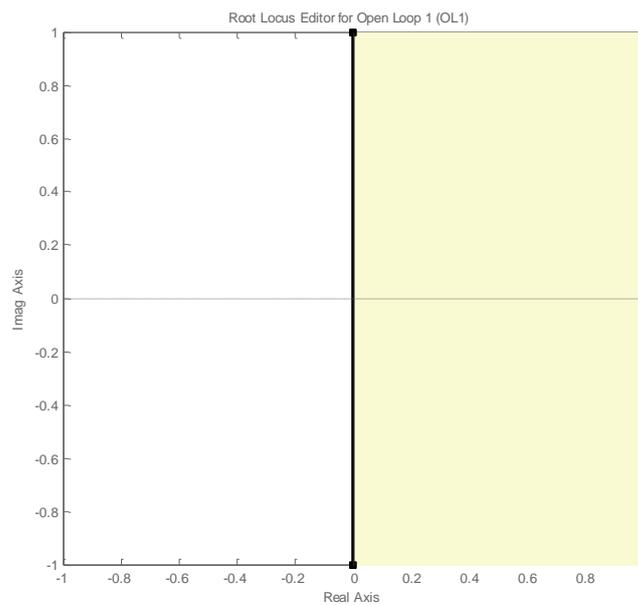


Figura 28. Lugar de las raíces continuo: pintada en blanco, la zona estable (izquierda del eje 0); en amarillo, la zona inestable (derecha del eje 0).

Del mismo modo se define el lugar de las raíces en el dominio discreto. En él, el lugar de las raíces presenta la forma mostrada en la Figura 29. En este caso, un sistema será estable si, y sólo si todos sus polos se encuentran dentro del círculo unidad.

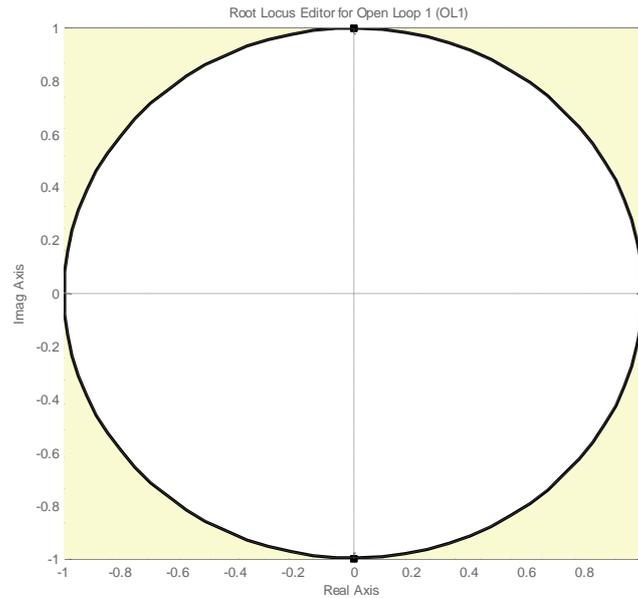


Figura 29. Lugar de las raíces en el dominio discreto: en blanco, zona estable (dentro del círculo unidad); en amarillo, zona inestable (fuera del círculo unidad).

En el diseño de controladores mediante el lugar de las raíces habitualmente se modifica la posición de los polos y los ceros para conseguir obtener la dinámica deseada del sistema en bucle cerrado. Además, se tiene otro grado de libertad: la ganancia del controlador, que permite establecer la magnitud de la señal de control.

Los sistemas realizados con un diseño multifrecuencial proporcionan otro grado de libertad a la hora de diseñar controladores. El efecto que produce la introducción de los DR-HOH en los sistemas se puede ver en las siguientes figuras:

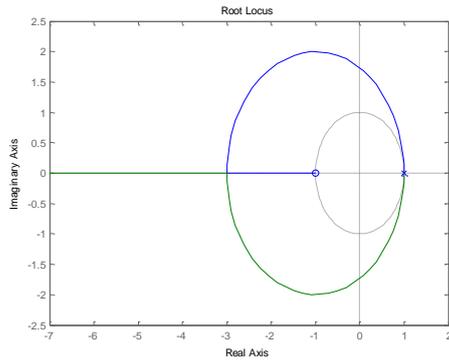


Figura 30. Lugar de las raíces multifrecuencia $N=1$ con FOH.

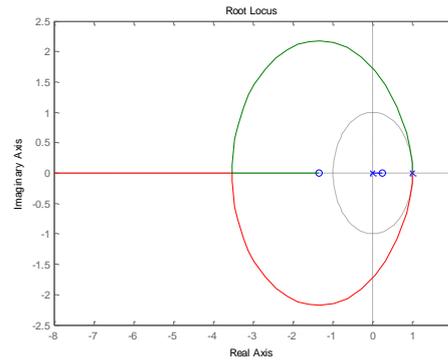


Figura 31. Lugar de las raíces multifrecuencia $N=2$ con FOH.

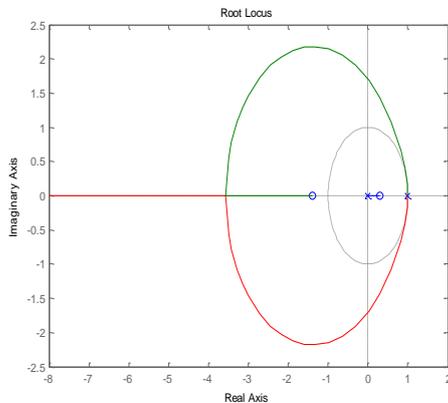


Figura 32. Lugar de las raíces multifrecuencia $N=3$ con FOH.

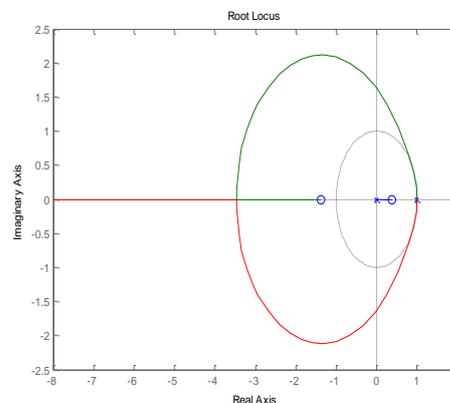


Figura 33. Lugar de las raíces multifrecuencia $N \rightarrow \infty$ con FOH.

Como se puede apreciar, según se ha modificado el orden de multiplicidad del retenedor de orden alto bi-frecuencia, la dinámica del sistema en bucle cerrado ha ido cambiando. Esto significa que se tiene otro grado de libertad a la hora de diseñar reguladores para sistemas SISO, lo que permite extender el clásico concepto de lugar de las raíces al del *Lugar de las Raíces Multifrecuencia*.

El efecto producido por los retenedores multifrecuencia se puede ver más claramente en la Figura 30. Como se puede observar a medida que crece el orden de multiplicidad del retenedor el sistema varía la dinámica. En este caso concreto se tiene que el sistema monofrecuencia a baja frecuencia es inestable en bucle cerrado mientras que cuando se introduce el retenedor a partir de un orden de multiplicidad N igual o superior a 2, el sistema se vuelve estable.

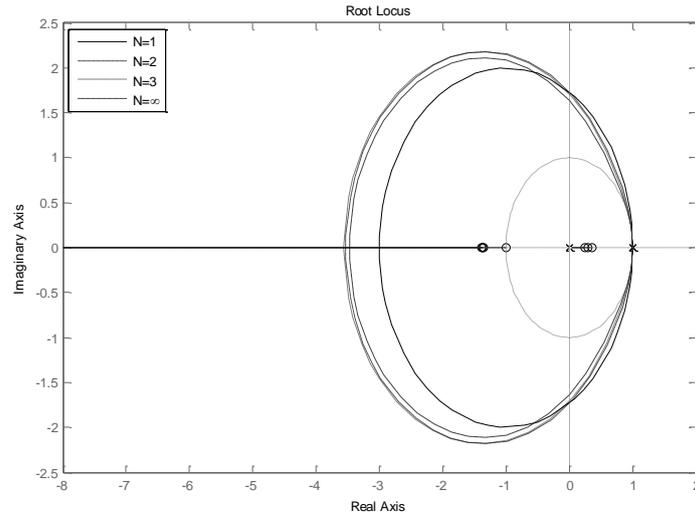


Figura 34. Variación del Lugar de las Raíces en función de N.

Otro fenómeno ligado al uso de retenedores multifrecuencia es que cada uno de ellos añade una dinámica distinta al sistema. La Figura 35 muestra este hecho con el uso de los retenedores más característicos utilizando el mismo orden de multiplicidad.

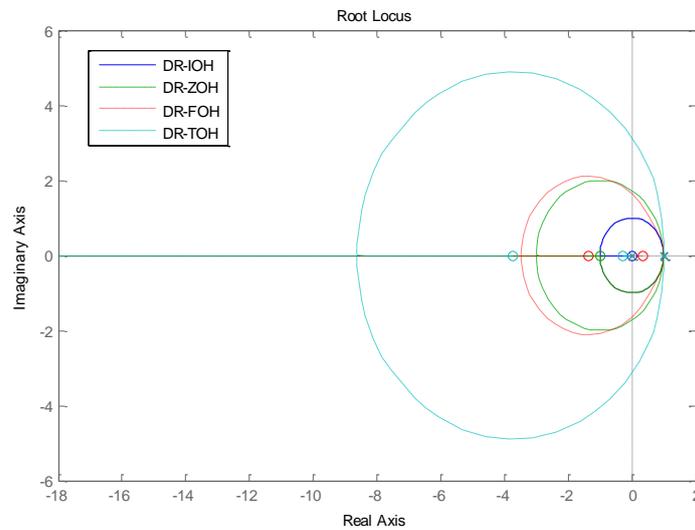


Figura 35. Lugar de las Raíces del sistema empleando distintos retenedores de orden alto bi-frecuencia.

CAPÍTULO 3

Image-Based Visual Servoing

3.1.-Conceptos y formulaciones

Los esquemas de control para el Image-Based Visual Servoing [52], [74], utilizan un conjunto de características cuyas coordenadas se sitúan en el plano de imagen para formar el vector s . Las unidades de las características son habitualmente los píxeles ya que las medidas se toman en el plano de imagen (aunque es posible utilizar cualquier otro sistema de medida).

En cuanto a los posibles esquemas de control que se utilizan dentro de este enfoque, una clasificación general de la misma es la siguiente:

- **Visual Servoing Directo:** en este caso el control basado en visión calcula directamente las acciones de control a partir de la dinámica del sistema. Esto hace que el sistema sea más rápido (generalmente la rapidez únicamente va a depender del algoritmo de visión, rondando los 40-50 milisegundos).

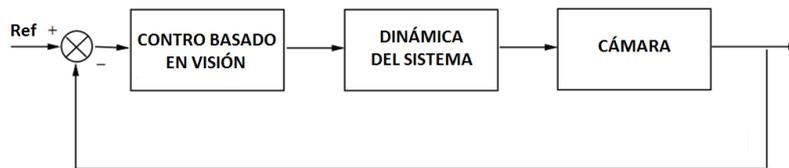


Figura 36. Esquema de control Directo del sistema Visual Servoing.

- **Visual Servoing Indirecto:** en este caso el sistema de control basado en visión calcula una referencia de control que es enviada a un controlador de más bajo nivel, y es éste último el que controla la dinámica del sistema. Muchas investigaciones actuales se centran en este esquema de control, también llamado "*Dynamic look-and-move*". En este caso, el controlador de bajo nivel ha de ser más rápido (generalmente menos de 10 milisegundos) que el de visión (unos 40~50 milisegundos).

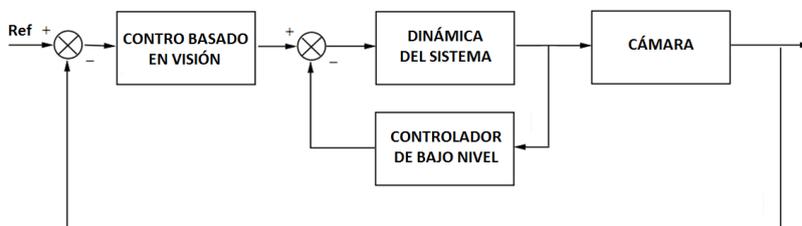


Figura 37. Esquema de control Indirecto del sistema Visual Servoing.

Teniendo en cuenta esta clasificación y debido a que en el enfoque del Image-Based Visual Servoing el error se obtiene a partir del espacio de la imagen 2D (por esta razón este enfoque es el denominado 2D Visual Servoing), pueden realizarse los esquemas de control dados por las Figuras 38 y 39.

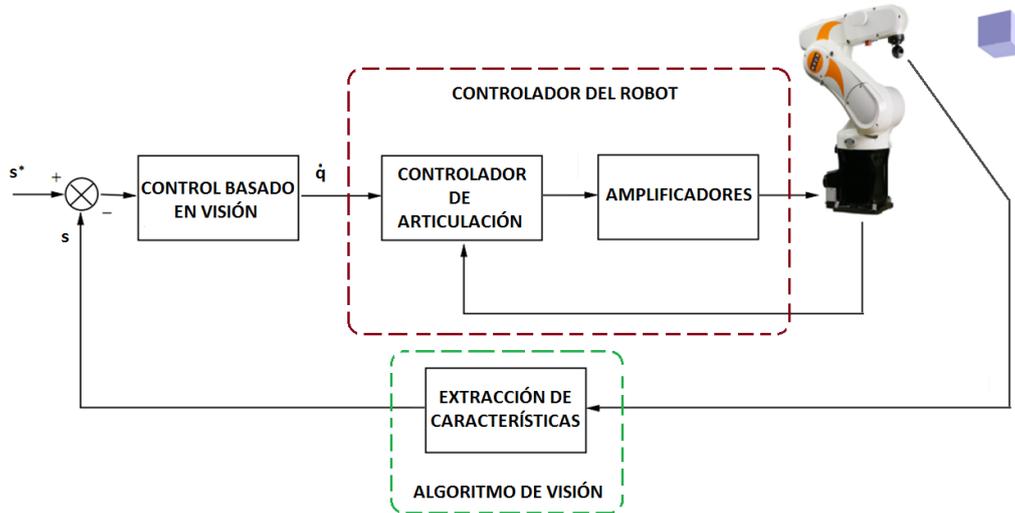


Figura 38. Esquema de control Indirecto para el Image-Based Visual Servoing.

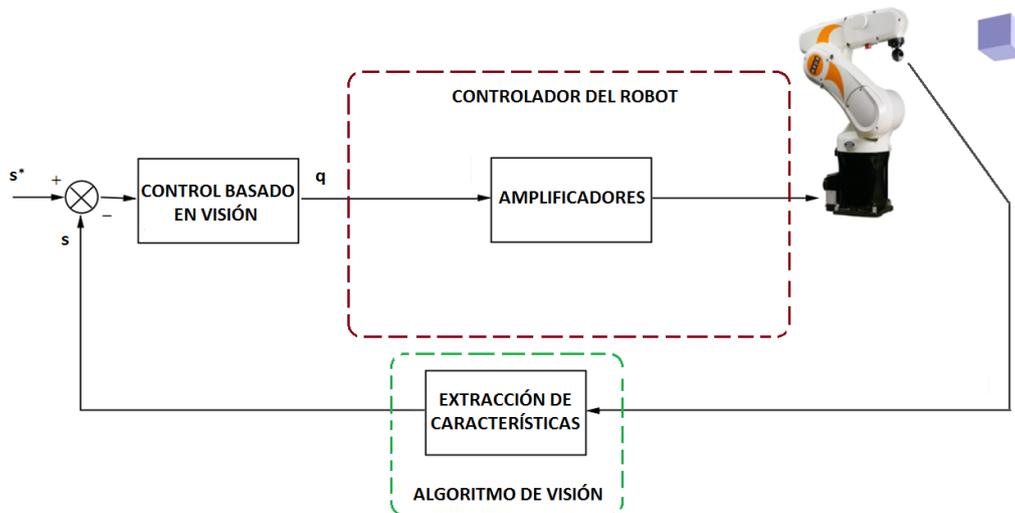


Figura 39. Esquema de control Directo para el Image-Based Visual Servoing.

Como se ha mencionado, la idea principal de todos los esquemas basados en visión es minimizar el error ($e(s)$) cometido entre la posición deseada y la posición actual del robot en relación al objeto de interés. Para el caso concreto del Image-Based Visual Servoing, este error viene definido a partir de las medidas tomadas directamente del plano de imagen de la cámara.

$$e(s) = s - s^* \quad (79)$$

Una vez se elige s (en este caso a partir de las medidas tomadas en el plano de imagen) el diseño de la ley de control es muy simple. Tal vez, el controlador más utilizado sea un controlador de velocidad. Para realizarlo, se requiere de la relación entre la variación de tiempo del vector de características s y la velocidad de la cámara. Se va a denotar como $V_c = (v_c, \omega_c)$ donde v_c es la velocidad instantánea lineal del origen de coordenadas de la cámara y ω_c es la velocidad instantánea angular del origen de coordenadas de la cámara. La relación entre s y V_c viene dada por la siguiente ecuación:

$$\dot{s} = L_s \cdot V_c \quad (80)$$

Siendo $L_s \in \mathbb{R}^{k \times n}$ la conocida *matriz de Interacción* [42]. El valor de k viene determinado por el número de características del objeto, mientras que la dimensión dada por n depende del número de articulaciones del robot. Para el caso de un robot de 6 grados de libertad (es el caso del robot que hay en el laboratorio, *Kuka KR5 sixx R650*) la dimensión de n sería 6.

Por otro lado en la literatura también es muy utilizado el *término Jacobiana de las características* introducido por [23] para definir L_s , aunque en muchos de los artículos que hay en la literatura este término se emplea para relacionar la variación de tiempo de las características con la velocidad de las articulaciones.

Relacionando las ecuaciones 79 y 80 se obtiene la siguiente relación entre la velocidad del error cometido y la velocidad de la cámara:

$$\dot{e} = L_s \cdot V_c \quad (81)$$

Si se considera que V_c es la velocidad que debe tener el sistema de referencia de la cámara en cada instante de tiempo y que se quiere tener un decrecimiento exponencial desacoplado del valor del error (como es $\dot{e} = -\lambda \cdot e$) se obtiene la siguiente ley de control:

$$V_c = -\lambda L_s^+ e(s) \quad (82)$$

La matriz $L_s^+ \in \mathbb{R}^{k \times n}$ se elige como la pseudo-inversa de Moore-Penrose de L_s , que es:

$$L_s^+ = (L_s^T \cdot L_s)^{-1} \cdot L_s^T \quad (83)$$

La elección hecha en la Ecuación 83 permite que $\|\dot{e} - \lambda \cdot L_s \cdot L_s^+ \cdot e\|$ y $\|V_c\|$ sean mínimos.

En sistemas Visual Servoing reales es imposible conocer perfectamente ni L_e ni L_e^+ . De esta forma es necesario realizar una aproximación o una estimación de estas matrices. Por este motivo, se va a denotar la aproximación de la pseudo-inversa de la matriz de Interacción mediante \hat{L}_s^+ . Utilizando esta notación la ley de control queda de la forma:

$$V_c = -\lambda \hat{L}_s^+ e(s) \quad (84)$$

Cerrando el bucle y asumiendo que el controlador del robot es perfectamente capaz de trabajar con velocidades, se tiene a partir de las ecuaciones 81 y 82:

$$\dot{e}(s) = -\lambda L_e \hat{L}_e^+ e(s) \quad (85)$$

La ecuación 80 caracteriza el comportamiento real del sistema en bucle cerrado, que será distinto en la medida en que $L_e \hat{L}_e^+ \neq I_6$.

3.1.1.-La matriz de Interacción

Para explicar los conceptos a partir de este punto, incluyendo simulaciones y pruebas reales, las características que se van a detectar serán puntos, siendo todo extrapolable a cualquier otro tipo de características como líneas, planos, etc.

Por lo tanto, para un punto 3D $P = (X \ Y \ Z)$ referenciado al sistema de coordenadas de la cámara, se proyecta en el plano de la imagen 2D $p = (x \ y)$, como:

$$\begin{cases} x = X/Z = (u - c_u) / f^\alpha \\ y = Y/Z = (v - c_v) / f^\alpha \end{cases} \quad (86)$$

Donde los parámetros u y v son las coordenadas del punto en el plano de imagen expresadas en píxeles, y los parámetros c_u , c_v , f y α son los parámetros intrínsecos de la cámara: c_u y c_v son las coordenadas del punto principal, f es la distancia focal y α es el ratio de la dimensión de los píxeles. El parámetro intrínseco β se asume normalmente nulo.

Siendo esto así, se toma como vector de características:

$$s = p = (x \quad y) \quad (87)$$

Todos los detalles de geometría y perspectiva de la proyección en el plano de imagen se pueden ver en [75] y [76] además de en cualquier libro de geometría en visión por computador.

Derivando la ecuación de proyección (85), se obtiene:

$$\begin{cases} \dot{x} = \dot{X}/Z - X\dot{Z}/Z^2 = (\dot{X} - x\dot{Z})/Z \\ \dot{y} = \dot{Y}/Z - Y\dot{Z}/Z^2 = (\dot{Y} - y\dot{Z})/Z \end{cases} \quad (88)$$

Relacionando la velocidad de un punto 3D con la velocidad espacial de la cámara utilizando la conocida expresión:

$$\dot{P} = -v_c - \omega_c \times P \Leftrightarrow \begin{cases} \dot{X} = -v_{cx} - \omega_{cy}Z + \omega_{cz}Y \\ \dot{Y} = -v_{cy} - \omega_{cz}X + \omega_{cx}Z \\ \dot{Z} = -v_{cz} - \omega_{cx}Y + \omega_{cy}X \end{cases} \quad (89)$$

Ahora uniendo las ecuaciones 83 y 84 y utilizando la expresión de la Ecuación 85, se tiene:

$$\begin{cases} \dot{x} = -v_{cx}/Z + xv_{cz}/Z + xy\omega_{cx} - (1+x^2)\omega_{cy} + y\omega_{cz} \\ \dot{y} = -v_{cy}/Z + yv_{cz}/Z + (1+y^2)\omega_{cy} - xy\omega_{cx} - x\omega_{cz} \end{cases} \quad (90)$$

La Ecuación 90 puede reescribirse del siguiente modo:

$$\dot{s} = L_s \cdot V_c \quad (91)$$

Por lo que se tiene que la expresión de la matriz de interacción cuando las características son puntos es:

$$L_s = \begin{pmatrix} -f/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -f/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \quad (92)$$

En la matriz de Interacción L_s , el parámetro Z es la profundidad que tiene el punto en relación al punto origen de coordenadas de la cámara en cada instante de tiempo. Así, cualquier esquema de control que utilice esta forma de matriz de Interacción debe estimar o aproximar el valor de Z . También hay que tener en cuenta que los parámetros intrínsecos de la cámara están involucrados en el cálculo de los

valores x e y . Así, L_s^+ no se puede utilizar directamente por lo que se debe realizar estimaciones o aproximaciones de esta matriz, esto es utilizar \hat{L}_s^+ .

3.1.2.-Aproximaciones de la matriz de Interacción

Hay varias formas de obtener la matriz de interacción estimada \hat{L}_s^+ para poder utilizarla en la ley de control. Uno de los esquemas más populares es elegir $\hat{L}_s^+ = L_s^+$ en el caso de que L_s^+ sea conocida, esto es si la profundidad Z en cada instante de tiempo, para cada punto, es conocida [23]. En la práctica, este parámetro debe estimarse en cada iteración del bucle de control. Los métodos básicos de Image-Based Visual Servoing utilizan métodos clásicos de *pose-estimation*.

Otra forma popularizada es la de elegir $\hat{L}_s^+ = L_{s^*}^+$ donde $L_{s^*}^+$ es el valor que tiene L_s^+ cuando la posición es la de referencia, es decir $s = s^* \rightarrow e = 0$, [42]. En este caso, $\hat{L}_{s^*}^+$ es constante, y solamente la profundidad Z de la posición deseada para cada punto se ha de configurar en el inicio de la tarea, lo que significa que no se estima la variación de los parámetros 3D durante la tarea de Visual Servoing.

Por último, recientemente se ha propuesto en [77] elegir $\hat{L}_{s^*}^+$ como $\hat{L}_s^+ = (L_s/2 + L_{s^*}/2)^+$. Ya que L_s está involucrada en este método, la profundidad Z de cada uno de los puntos en cada iteración del bucle de control ha de estimarse.

3.1.3.-Interpretación geométrica del IBVS.

Es muy interesante proporcionar una interpretación geométrica del comportamiento de los esquemas de control descritos anteriormente. En la Figura 36, se puede observar una rotación pura de cuatro puntos en el plano de imagen alrededor del eje óptico, desde una configuración inicial (puntos marrones) hasta una configuración deseada (puntos rojos).

Como se ha explicado en la sección anterior, utilizando L_s^+ en el esquema de control, se asegura un decrecimiento exponencial del error e . Esto significa que, cuando se mide el error a partir de los puntos en coordenadas de imagen x e y , la trayectoria de dichos puntos en el plano de imagen sigue una línea recta desde su posición inicial a su posición deseada, siempre y cuando sea posible. Esto se muestra en el vector pintado de verde en la imagen de la Figura 36. El movimiento de la cámara para poder realizar este movimiento se puede deducir fácilmente y, se hecho, está compuesto por un movimiento rotacional alrededor del eje óptico [78]. Este inesperado movimiento es debido a la elección de las características y al acoplamiento

que hay entre las columnas 3 a la 6 de la matriz de interacción. Si la rotación entre las configuraciones inicial y final es muy grande, este fenómeno se amplifica y provoca el caso particular de una rotación de π -radianes en la que no existe movimiento de rotación [49]. Por otra parte, cuando la rotación es muy pequeña, este fenómeno desaparece. Para finalizar, el comportamiento es localmente satisfactorio (por ejemplo cuando el error es pequeño), pero puede fallar cuando el error es grande.

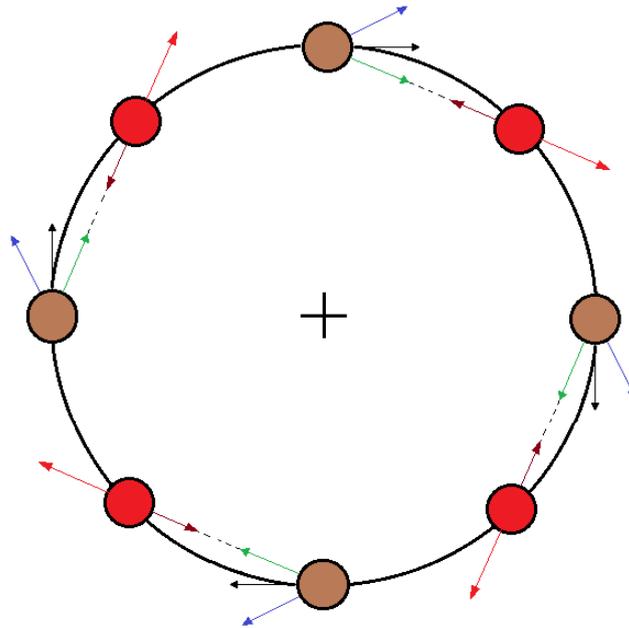


Figura 40. Interpretación geométrica del IBVS: el movimiento va de los puntos marrones a los puntos rojos. En verde, el movimiento de los puntos en el plano de imagen cuando se trabaja con L_s^+ ; en azul cuando se trabaja con $L_{s^*}^+$; en negro, el movimiento de los puntos cuando se trabaja con $(L_s/2 + L_{s^*}/2)^+$.

Si se utiliza como L_s^+ la aproximación dada de $\hat{L}_{s^*}^+$ para el esquema de control, el movimiento generado se puede ver representado por los vectores pintados de azul en la imagen de la Figura 36. De hecho, si se considera el mismo esquema de control como antes pero empezando desde s^* hasta s , se obtiene:

$$V_c = -\lambda \cdot L_{s^*}^+ (s^* - s) \quad (93)$$

Que como se ve también induce trayectorias rectas desde los puntos rojos a los puntos azules, causando el movimiento pintado por el vector marrón. Volviendo al problema inicial, en el cuál el esquema de control es exactamente opuesto al dado por la ecuación 88:

$$\mathbf{V}_c = -\lambda \cdot L_{s^*}^+ (\mathbf{s} - \mathbf{s}^*) \quad (94)$$

De este modo, se genera el movimiento descrito por los vectores pintados en rojo si partimos de los puntos rojos. Partiendo de los puntos azules, el movimiento de la cámara se describe por los vectores pintados en azul y vuelve a corresponderse con un movimiento de rotación alrededor del eje óptico de la cámara. El mismo análisis es equivalente para pequeños y grandes errores. Hay que añadir, que tan pronto como el error decrece significativamente, ambos esquemas de control se van haciendo más equivalentes y tienden a proporcionar los mismos valores de error y de matriz de interacción.

Por último, si se utiliza el esquema dado por $\hat{L}_s^+ = (L_s/2 + L_{s^*}/2)^+$, es intuitivo imaginar que generara una trayectoria intermedia entre las dos anteriores. De hecho esto se ha representado con los vectores pintados de negro en la imagen de la Figura 36, incluso si el error es grande. En todos los casos, menos en el de π -radianes, el movimiento de la cámara es una rotación pura alrededor del eje óptico de la cámara, sin que haya ninguna translación.

3.2.-Estudio del comportamiento del sistema mediante simulación

En esta sección se quiere analizar algunos de los anteriores resultados mediante el uso de herramientas de simulación. Hay que decir que en la actualidad, tanto para el estudio de la robótica como para el del Visual Servoing hay algunas aplicaciones muy interesantes para realizar simulaciones, como son el *The Machine Vision Toolbox* [79], *Java-Based Visual Servoing Simulator* [80] o el *Visual Servoing Platform* [81].

Cabe mencionar que, aunque cada versión que sale de cualquiera de las tres versiones del software mencionado es más completa, los simuladores realizados se basan en el sistema denominado de cámara libre. Esto es así porque cada robot tiene un comportamiento distinto que ha de ser modelado y que para el estudio de las distintas técnicas del Visual Servoing no es necesario la incorporación del robot en sí, entendiéndose que es más un problema de implementación.

Es por ello que se ha realizado la implementación de un simulador que reflejara el sistema Robot-Cámara, para estudiar mejor el comportamiento del sistema global y así entender las técnicas antes de su implementación en el sistema real.

Como se ha mencionado, se tenían distintas plataformas para la realización del simulador. Debido a las facilidades aportadas a la hora de realizar simuladores de Matlab/ Simulink se optó por ella. De este modo, se utilizaron para la elaboración del

simulador las herramientas *Matlab Toolbox Visual Servoing* [79] y *Robotics Toolbox* [82].

3.2.1.-Robot Kuka KR5 sixx R650

Para realizar varias investigaciones, entre ellas la que se está viendo en este documento, el Instituto de Diseño y Fabricación (IDF) de la Universidad Politécnica de Valencia tiene un robot Kuka KR5 sixx R650 (ver Figura 41). Se trata de un robot de los denominados pequeños con las siguientes características:

- Carga: 5 kg a máxima velocidad
- Zona de trabajo: máximo alcance 650 mm
- Número de ejes: 6 ejes (6 grados de libertad)
- Repetitividad: $<\pm 0.02$ mm
- Peso: 28 kg
- Posiciones de montaje: Suelo, Techo
- Unidades de control: KR C2sr
- Velocidad máxima: 8.2 m/s



Figura 41. Robot manipulador industrial Kuka KR5 sixx R650.

Como se puede ver es un robot pequeño idóneo para tareas como la del *Visual Servoing*. Las características geométricas se pueden ver en la Figura 42 y los valores de los distintos parámetros se muestran en la Tabla 4.

Con estas especificaciones ya se está en disposición de obtener los parámetros de la tabla *Denavit-Hartenberg* [83] que va a definir la geometría del robot a la hora de realizar su modelado.

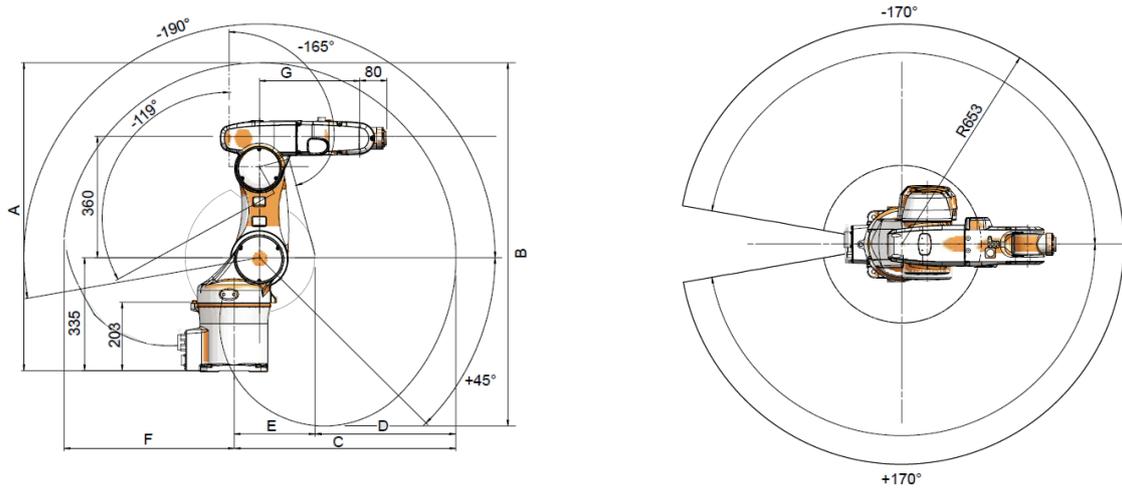


Figura 42. Características geométricas del Kuka KR5 sixx R650.

Dimensiones del Kuka KR5 sixx R650	
A	913 mm
B	1078 mm
C	855 mm
D	419 mm
E	234 mm
F	500 mm
G	295 mm

Tabla 4. Valores geométricos característicos del Kuka KR5 sixx R650.

En [83] se estableció un método matricial que permite establecer de manera sistemática un sistema de coordenadas (S_i) ligado a cada articulación i de una cadena dada, pudiéndose determinar a continuación las ecuaciones cinemáticas de la cadena completa.

Según la representación D-H, escogiendo adecuadamente los sistemas de coordenadas asociados para cada articulación, es posible pasar de uno al siguiente mediante cuatro transformaciones básicas que dependen exclusivamente de las

características geométricas de la articulación. Estas transformaciones básicas consisten en una sucesión de rotaciones y translaciones que permiten relacionar el sistema de referencia del elemento i con el sistema del elemento $i-1$. Los pasos a seguir son los siguientes:

- Elección de un sistema de referencia fijo $S_0 = [x_0 \ y_0 \ z_0]$ asociado a la base del robot.
- Localizar el eje de cada articulación z_i :
 - Si la articulación es rotativa, el eje tendrá la dirección del eje de giro.
 - Si la articulación es prismática, el eje tendrá la dirección de deslizamiento.
- Situar los ejes x_i en la línea normal común a z_{i-1} y z_i . Si estos son paralelos, se elige sobre la línea normal que corta a ambos ejes.
- El eje y_i debe completar el triedro dextrógiro.

Una vez establecidos todos los sistemas de coordenadas, se definen los parámetros D-H del siguiente modo:

- α_i : ángulo entre el eje z_{i-1} y z_i sobre el plano perpendicular a x_i . El signo lo da la regla de la mano derecha.
- a_i : distancia entre los ejes z_{i-1} y z_i a lo largo del eje x_i . El signo lo define el sentido de x_i .
- θ_i : ángulo que forman los ejes x_{i-1} y x_i sobre el plano perpendicular a z_{i-1} . El signo lo determina la regla de la mano derecha.
- d_i : distancia a lo largo del eje z_{i-1} desde el origen del sistema s_{i-1} hasta la intersección del eje z_{i-1} con el eje x_i . En el caso de articulaciones prismáticas será la variable de desplazamiento.

La Figura 39 muestra los sistemas de coordenadas según el convenio clásico y con todos los ángulos positivos. Este es un convenio que se utiliza en las lecciones sobre robótica industrial para que todos tengan los mismos resultados y que, por costumbre también se ha elegido para el modelado de este robot. Cabe destacar que se ha representado con \odot cuando el vector es perpendicular al plano y hacia afuera, y con \otimes cuando el vector es perpendicular al plano pero hacia adentro.

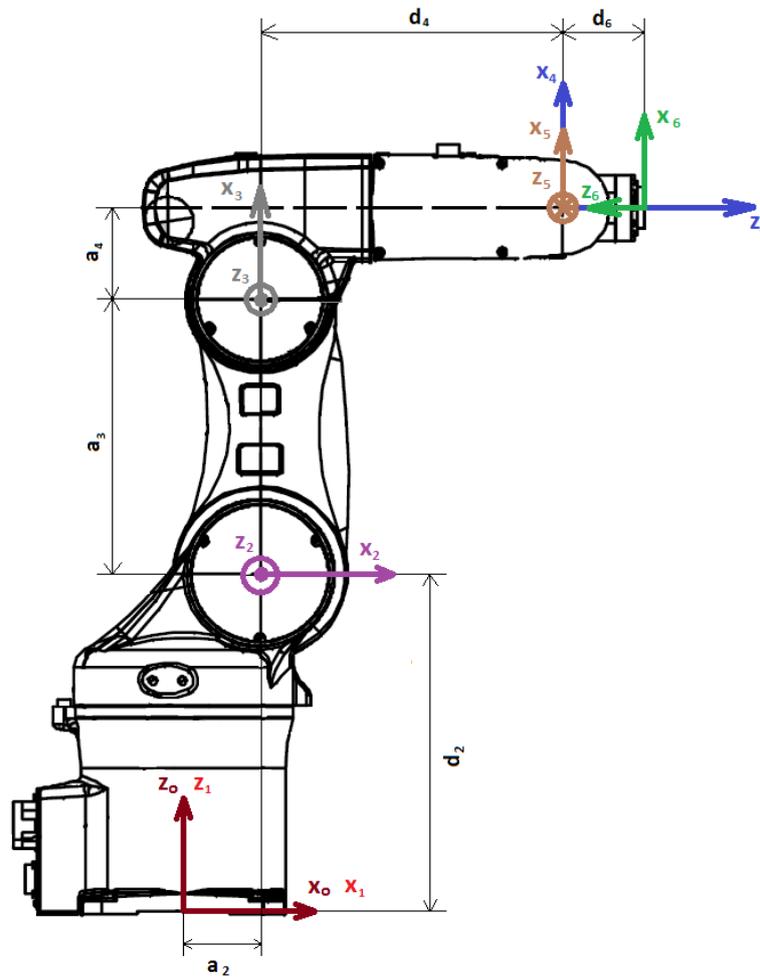


Figura 43. Sistemas de Referencia de cada articulación del Kuka KR5 sixx R650 con el convenio clásico D-H.

En la Tabla 5 se muestran los parámetros D-H según los criterios que se han adoptado. Estos parámetros dependen de los criterios elegidos a la hora de situar cada uno de los sistemas de coordenadas de cada articulación y de la posición inicial del robot.

Los parámetros que definen la geometría del robot toman los siguientes valores:

- $d_2 = 335mm$, $a_2 = 75mm$
- $a_3 = 270mm$
- $d_4 = 295mm$, $a_4 = 90mm$
- $d_6 = 80mm$

	θ_i	d_i	α_i	a_i
ARTICULACIÓN 1	θ_1	0	0	0
ARTICULACIÓN 2	θ_2	d_2	90	a_2
ARTICULACIÓN 3	$\theta_3 + 90$	0	0	a_3
ARTICULACIÓN 4	θ_4	d_4	90	a_4
ARTICULACIÓN 5	θ_5	0	90	0
ARTICULACIÓN 6	θ_6	$-d_6$	90	0

Tabla 5. Parámetros D-H para el robot Kuka KR5 sixx R650.

Una vez definidos los parámetros D-H y con la ayuda de la herramienta *Robotics Toolbox* para Matlab el robot modelado se puede ver en la Figura 44. El siguiente apartado muestra el proceso de obtención de los parámetros intrínsecos de la cámara empleada para realizar la tarea de Visual Servoing

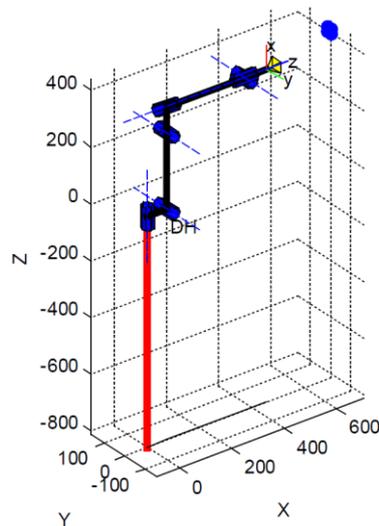


Figura 44. Simulador Kuka KR5 sixx R650 con Matlab/ Simulink.

3.2.2.-Calibración de la Cámara Logitech C300

La calibración de una cámara significa la obtención de sus parámetros intrínsecos, extrínsecos y de las distorsiones que tenga la cámara [84], [85], [86] y [87]. Es un paso necesario en cualquier proceso de visión por computador para poder realizar la extracción de la información de las medidas 2D en el plano de imagen, es decir las distancias en las imágenes.

Para calibrar la cámara se hizo uso de la herramienta *Camera Calibration Toolbox* de Matlab. La cámara utilizada es una Webcam Logitech C300 (ver Figura 45) que tiene las siguientes especificaciones:

- Sensor de 1.3-megapixel (1280x1024 pixeles)
- Captura de video: hasta 1280x1024 (aumento por software)
- Fotos: hasta 5-megapixel (aumento por software)
- Hasta 30 frames por segundo de video (con sistemas recomendados)
- Micrófono con tecnología Logitech RightSound™
- Hi-Speed USB 2.0
- Botón de captura de imágenes



Figura 45. Cámara Webcam Logitech C300.

3.2.2.1.- Procedimiento de calibración

En este apartado se va a detallar el procedimiento de calibración que se ha seguido para la obtención de los parámetros intrínsecos de la cámara necesarios para el modelado de la misma. Como se ha mencionado antes, se va a utilizar como herramienta de calibración el *Camera Calibration Toolbox* de Matlab. Se va a utilizar como plantilla la mostrada en la Figura 46. Además lo que se va a hacer es mantener la cámara fija y se irá moviendo la plantilla para ver y examinar los distintos planos.



Figura 46. Montaje de calibración. Webcam Logitech C300 y plantilla de calibración.

En primer lugar se ejecuta la herramienta de calibración y se cargan 20 imágenes tomadas por la cámara con distintas posiciones de la plantilla, como se ve en la Figura 47.

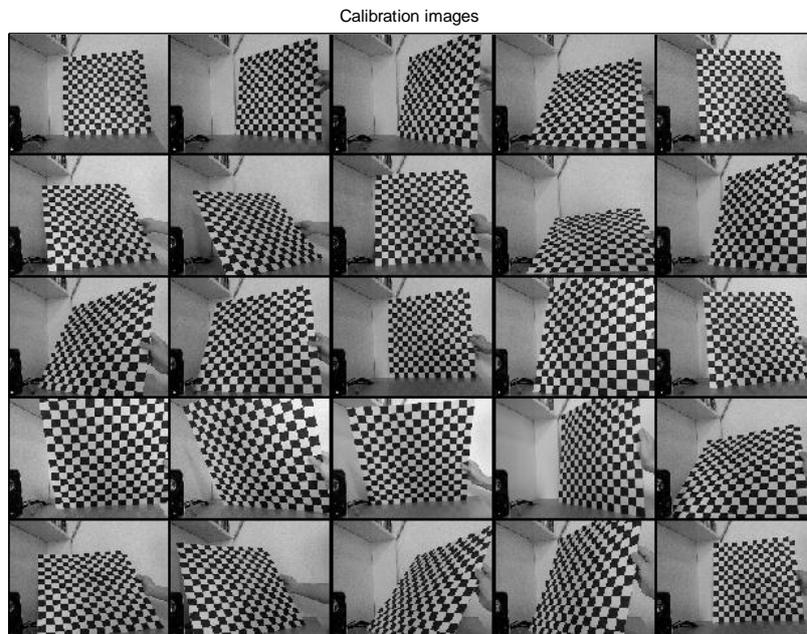


Figura 47. Imágenes tomadas para la calibración de la cámara.

El siguiente paso es realizar la extracción de las esquinas para cada una de las imágenes introducidas. Lo que se va a hacer es explicar el procedimiento de extracción en una de ellas, entendiendo que en las demás se realiza de la misma forma.

Por lo tanto cuando se presiona la opción de extracción de esquinas lo primero que aparece es la imagen para que se le introduzca el área dónde se va a realizar la detección de las esquinas (ver Figura 48). Decir que hay un método de extracción automática, aunque depende mucho de la calidad, posición distorsión etc., que hay en la imagen por lo que en este caso se ha preferido realizar la detección manualmente.

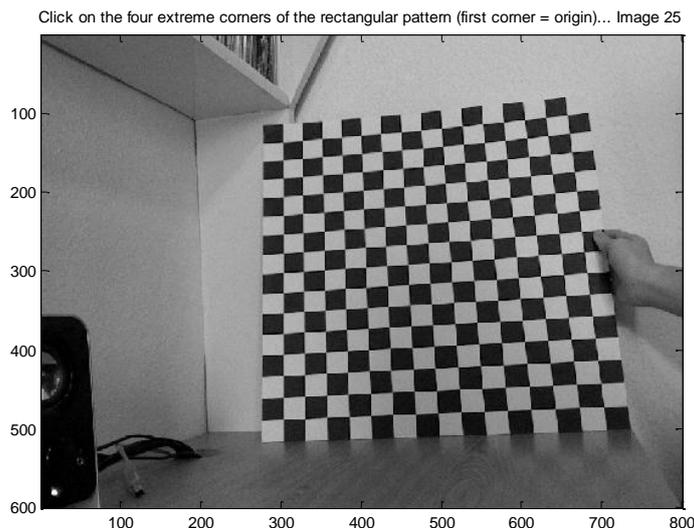


Figura 48. Imagen preparada para la detección del área donde se encuentra el patrón de calibración.

Dentro del patrón utilizado, se ha elegido, por comodidad una parte del mismo que constará de 10x10 cuadrados. De esta forma el tamaño del patrón será siempre el mismo, independientemente de si en alguna de las imágenes ha salido cortado (ver Figura 49).

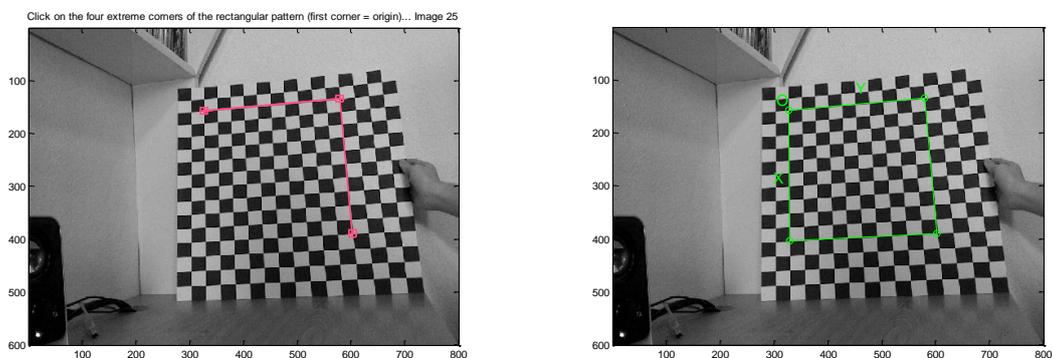


Figura 49. Selección manual del patrón a analizar.

Una vez elegido la zona donde está el patrón, el algoritmo debe asegurarse de si los cuadrados que ha detectado son los correctos, por lo que antes de realizar la extracción de las esquinas se pregunta si los datos detectados son correctos. Una vez realizado esto, se hace una primera estimación de dónde está las esquinas dentro del patrón (ver Figura 50). Si todo es correcto, se pasa a otra imagen. Es posible que realizando un zoom de la zona (a veces se ve a simple vista) se vea que hay un gran error en la estimación de las esquinas. Es por ello que la herramienta permite ajustar el parámetro de distorsión para solucionarlo.

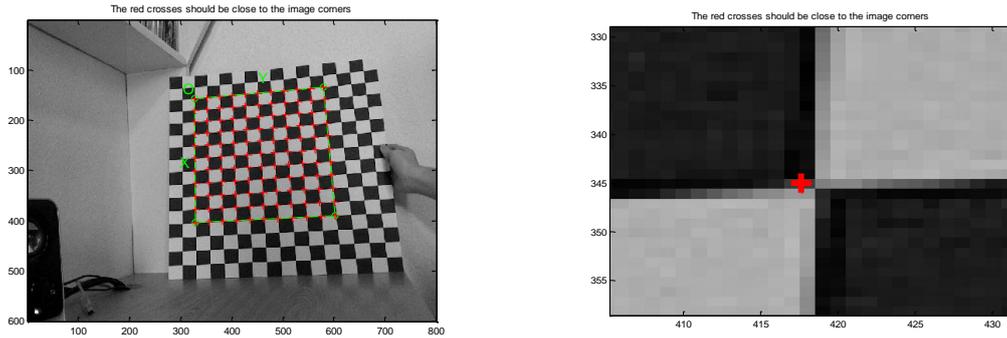


Figura 50. Extracción de las esquinas en el patrón elegido. A la izquierda la extracción realizada; a la de derecha el error cometido en la estimación.

Una vez conforme con los resultados mostrados, se le dice al algoritmo que el resultado de extracción es correcto y lo muestra por pantalla tal y como se muestra en la Figura 51. Seguidamente se carga la siguiente imagen y se repite el mismo procedimiento.

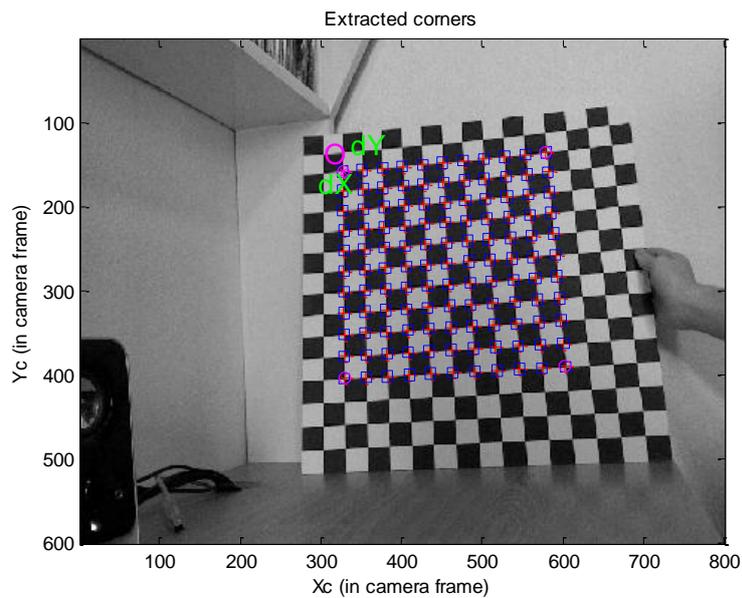


Figura 51. Resultado de la detección de las esquinas en el patrón.

Una vez establecidas todas las detecciones en todas las imágenes de calibración, el algoritmo está listo para realizar la estimación de los parámetros de la cámara. Es por ello que se le da la orden de realizar la calibración obteniendo los parámetros estimados de la cámara. A continuación se realizará una breve descripción de cada uno de ellos.

3.2.2.1.1.- Parámetros intrínsecos

El modelo de cámara que esta herramienta utiliza es muy similar al descrito por [85]. Los parámetros intrínsecos relacionan el sistema coordinado de la cámara con el sistema coordinado idealizado de la imagen. La lista de parámetros internos es la siguiente:

- Distancia focal (f)
- Punto principal ($C = [c_u \ c_v]$)
- Coeficiente de descentrado, "skew" (α)
- Distorsión (k)

Dependiendo del modelo utilizado de cámara puede haber más parámetros, aunque en modelo que se está utilizando estos son los más relevantes.

Para el caso concreto que nos ocupa, la herramienta ha estimado los siguientes valores para cada uno de los parámetros anteriores:

	VALORES DE LOS PARÁMETROS INTRÍNSECOS DESPUES DE LA OPTIMIZACIÓN
DISTANCIA FOCAL	$f_c = [892.80757 \ 885.85507] \pm [5.95110 \ 5.27838]$
PUNTO PRINCIPAL	$C_c = [308.66733 \ 30254208] \pm [8.90293 \ 6.86548]$
SKEW	$\alpha_c = [0.000] \pm [0.000]$
DISTORSIÓN	$K_c = [0.08881 \ -0.09835 \ 0.00185 \ -0.03917 \ 0.000] \pm [0.02845 \ 0.08284 \ 0.00287 \ 0.00471 \ 0.000]$

Tabla 6. Parámetros intrínsecos estimados después de la calibración.

3.2.2.1.2.- Parámetros extrínsecos

Por lo que respecta a los parámetros extrínsecos de una cámara, estos relacionan el sistema coordinado de la cámara a un sistema coordinado del mundo fijo y especifican su posición y orientación en el espacio.

Por lo tanto, estos vendrán dados por una matriz de orientación y una de translación que relacionará el sistema de coordenadas de la cámara con respecto a

cada una de las imágenes, ver Figura 52. En la Figura 53 se muestra el caso inverso, esto es que la cámara es la que se mueve mientras la escena permanece fija.

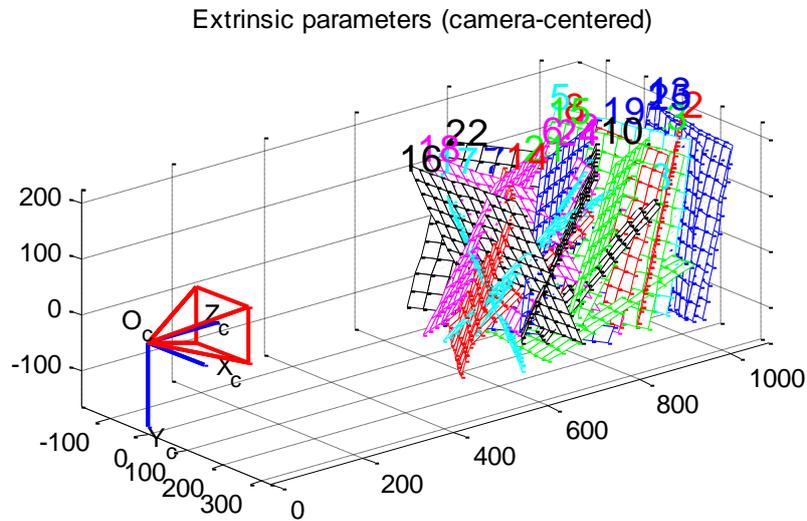


Figura 52. Parámetros extrínsecos de la cámara fija respecto a la escena.

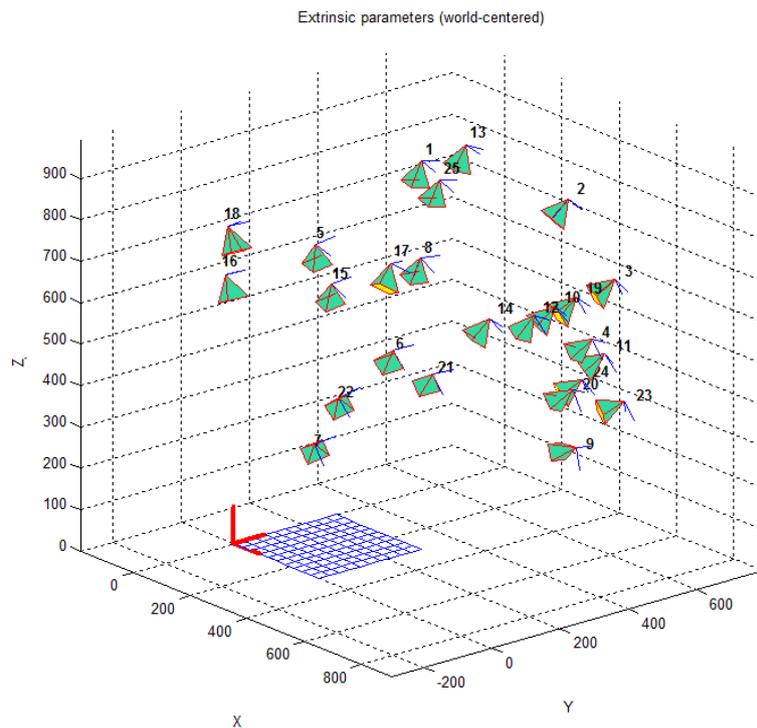


Figura 53. Parámetros extrínsecos manteniendo la escena fija y la cámara móvil.

En este caso, para la realización del simulador, los parámetros utilizados son los intrínsecos para modelar la cámara. Los parámetros extrínsecos no se han utilizado porque en simulación, en todo momento se conoce la posición de cada uno de los sistemas de coordenadas respecto al que se requiera. En el apartado de los experimentos realizados en la aplicación real, se explicará más detalladamente la

calibración *eye-in-hand* utilizada para obtener los parámetros extrínsecos y utilizarlos en el algoritmo de Visual Servoing.

3.2.3.-Simulador IBVS Kuka KR5 sixx R650

Con todo lo descrito anteriormente ya se tienen todas las herramientas para el modelado del sistema Visual Servoing. El diagrama Simulink del simulador se puede ver en la Figura 54. En él se diferencian cuatro partes:

- La simulación del objeto virtual
- El modelo de cámara
- El controlador
- El robot

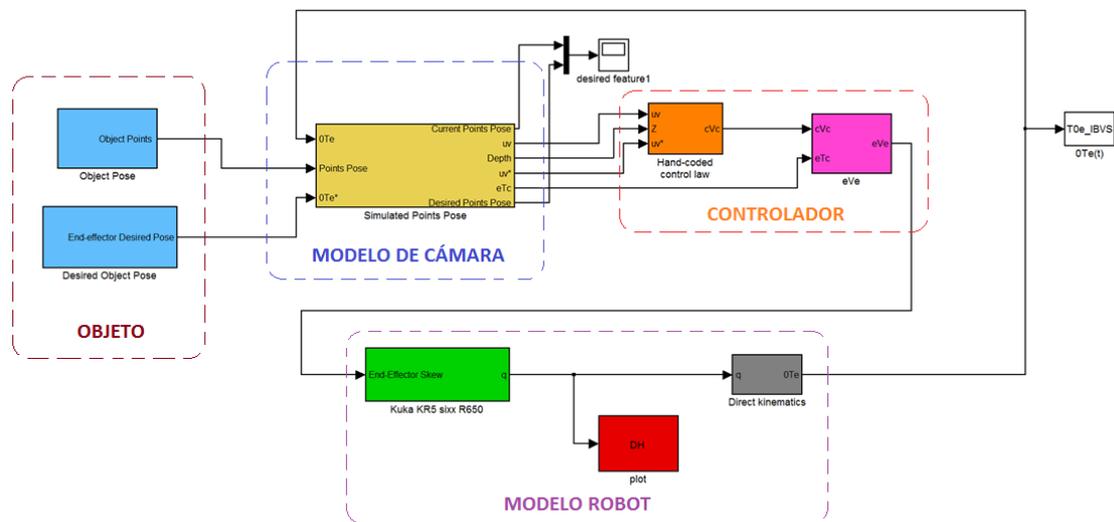


Figura 54. Diagrama Simulink del Simulador IBVS para el Kuka KR5 sixx R650.

Como se ha mencionado anteriormente, el simulador realizado implementa la técnica del Image-Based Visual Servoing (IBVS), con el esquema de control Directo (ver Figura 39). A modo de descripción se va a mostrar lo que contiene cada bloque en su interior:

- *Bloque Simulated Point Pose (Modelo de cámara):* es el bloque encargado de simular tanto la posición deseada en el plano de imagen de las características del objeto visto por la cámara situada en el efector final del robot, como de representar la situación del objeto en el plano de imagen en cada iteración del algoritmo. En la aplicación real, este bloque constituye la parte de adquisición y procesamiento del algoritmo de visión.

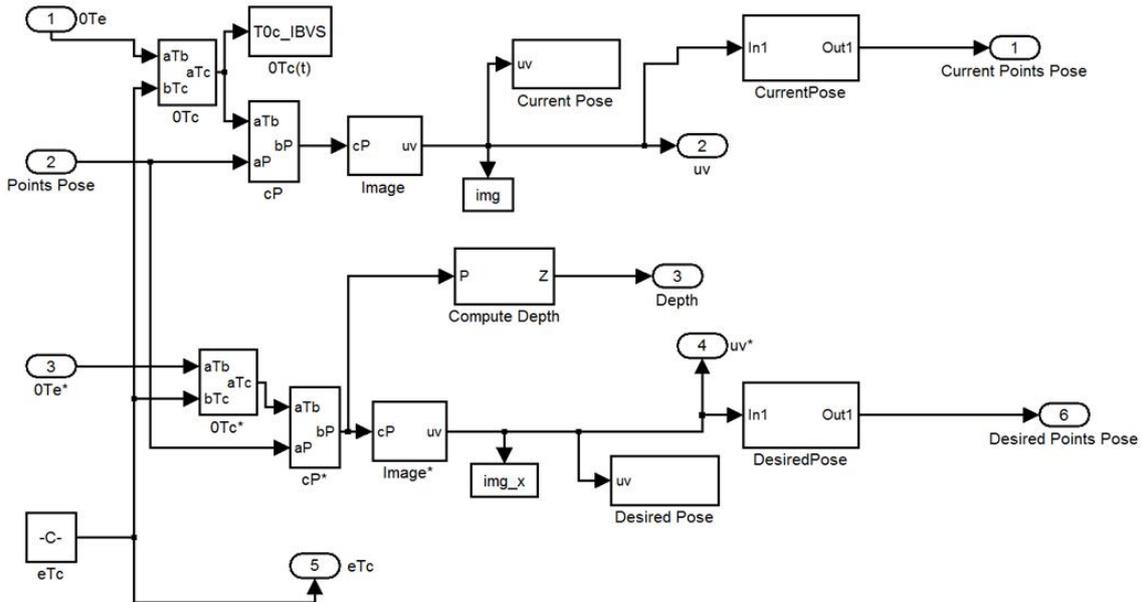


Figura 55. Diagrama Simulink del interior del bloque *Simulated point pose*.

- *Bloque Hand-Coded Control Law (Bloque de Control)*: es el bloque encargado de computar la ley de control IBVS. Como se puede ver en la Ecuación 82, para computar la ley de control hay que conocer la posición de los puntos en el plano de imagen, la estimación de la profundidad y los parámetros intrínsecos de la cámara. La salida es el *velocity skew vector* de debe tener la cámara para realizar el tracking del objeto. Es por ello que se realiza una transformación para conocer el *velocity skew vector* del efector final a partir del anterior (Bloque ${}_eV_e$).
- *Bloque Kuka KR5 sixx R650 (Bloque Robot)*: en él se calcula la posición final del robot modelado (con el Robotics Toolbox) en cada iteración y se muestra por pantalla.

En el siguiente apartado se van a mostrar los resultados de las simulaciones realizadas. Hay dos clases de experimentos bien diferenciados: uno muestra los distintos comportamientos obtenidos con un modelo estático del objeto, esto es una posición inicial y otra deseada fija; el otro experimento, que tiene más relación con el estudio que se realiza en el Capítulo 4, se trata del seguimiento de un objeto que se encuentra en continuo movimiento.

3.2.3.1.- Comportamiento del sistema

3.2.3.1.1.- Objeto Estático

En primer lugar se va a analizar el comportamiento que tiene el sistema dependiendo de la matriz de interacción que se utilice. Para ello, se dispone de un objeto que consta de cuatro puntos: imagínese un cuadrado en el que se detectan los cuatro vértices del mismo, ver Figura 56. Se ha elegido que el objeto tenga todos los elementos que se pueden tener para el tracking es decir, una translación, una rotación y un zoom.

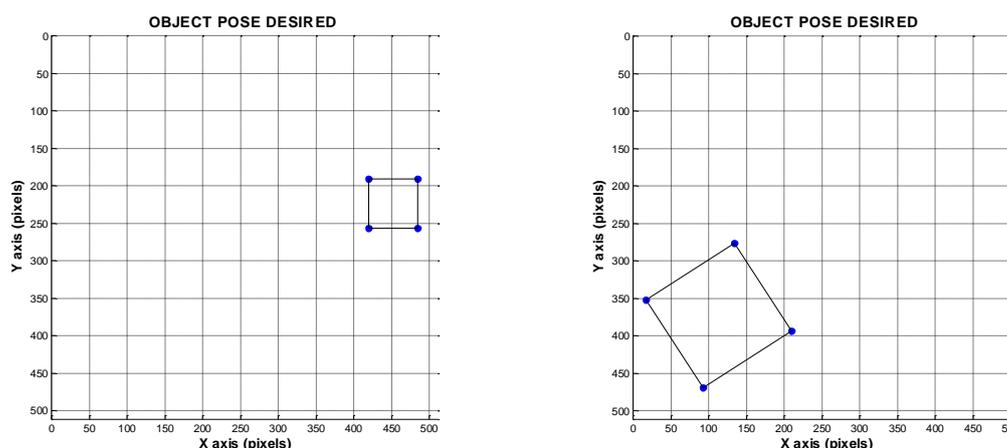


Figura 56. Representación en el plano de imagen del objeto utilizado para las simulaciones de objeto fijo: izquierda posición inicial del objeto; derecha posición deseada del objeto.

Como se ha dicho se van a analizar los tres métodos vistos en apartados anteriores para la estimación de la matriz de interacción:

- Simulaciones con $\hat{L}_s^+ = L_s^+$. Como se ha visto en el apartado correspondiente, esto significa que la distancia del sistema de coordenadas de la cámara respecto a cada uno de los puntos Z es conocida en cada iteración del algoritmo.

En primer lugar, la Figura 57 muestra la trayectoria en el plano de imagen seguida por los puntos desde la posición inicial a la final. Se puede ver que ésta intenta seguir una trayectoria de los puntos inicial hasta final en el plano de imagen. Cabe destacar que en cámara libre, esto es así. Este simulador tiene en cuenta la cinemática del robot por lo que no todas las posiciones son permitidas y pueden haber discrepancias entre el trabajar en cámara libre y con el robot, como se ve en la Figura 57.

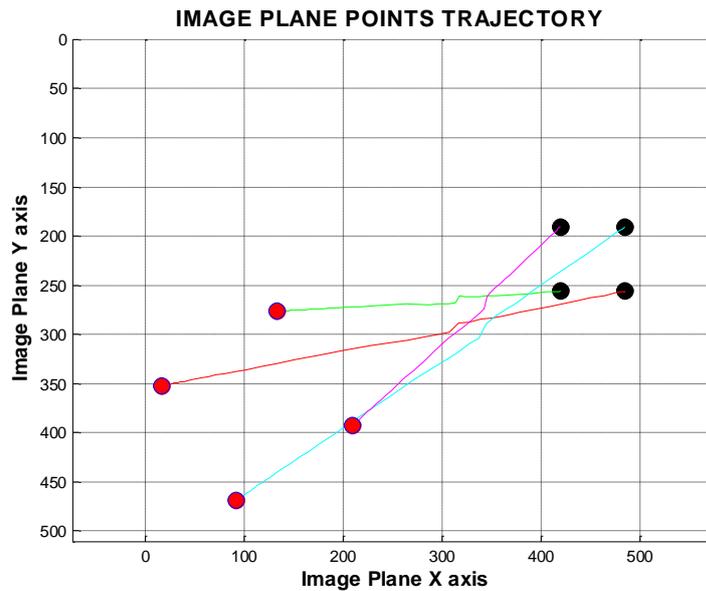


Figura 57. Trayectoria de los puntos desde la posición inicial a la final con el enfoque de $\hat{L}_s^+ = L_s^+$.

Por otra parte es interesante ver la trayectoria seguida por la cámara (que es la misma que la que ha seguido el efector final del robot) durante la ejecución. Es por ello que la Figura 58 muestra la trayectoria seguida por la cámara en el mundo 3D durante el periodo de ejecución del algoritmo. En la misma, se puede observar que en el mundo 3D el comportamiento no es ni mucho menos el de una trayectoria recta, sino que, para poder realizar un *matching* del objeto siguiendo una trayectoria recta en el plano de imagen, en el mundo cartesiano esto se convierte en una trayectoria curvada, como se comprueba en la imagen de la figura.

Por último, la Figura 59 muestra el error entre la posición de los puntos en cada instante de tiempo y la posición deseada de los mismos. Cabe destacar que los tiempos de convergencia dependen de muchos factores: la ganancia del controlador (λ), la dinámica del robot, la cantidad de características a detectar. Aunque cabe destacar que el *matching* cuando el sistema *Visual Servoing* tiene que controlar translación y orientación del efector final es bastante lento, sobre todo cuando más se acerca a la convergencia final.

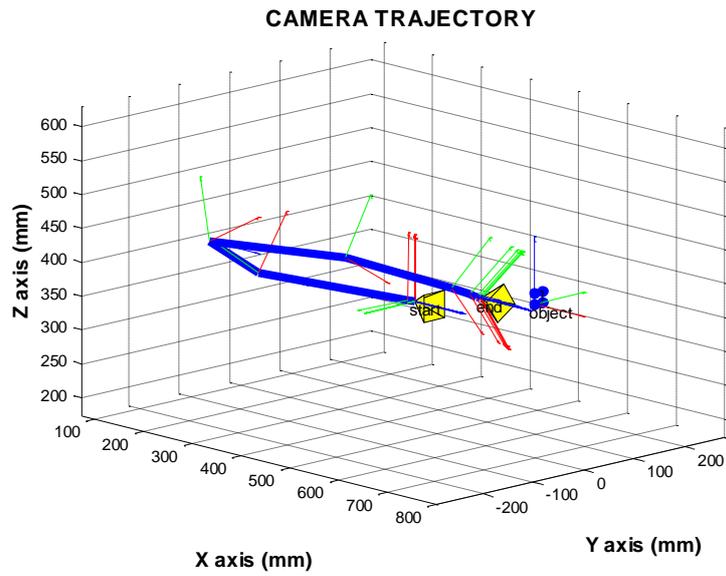


Figura 58. Trayectoria seguida por la cámara en el mundo cartesiano. Se comprueba que la realización de una trayectoria en línea recta en el plano de imagen se transforma en una trayectoria curva en el plano cartesiano.

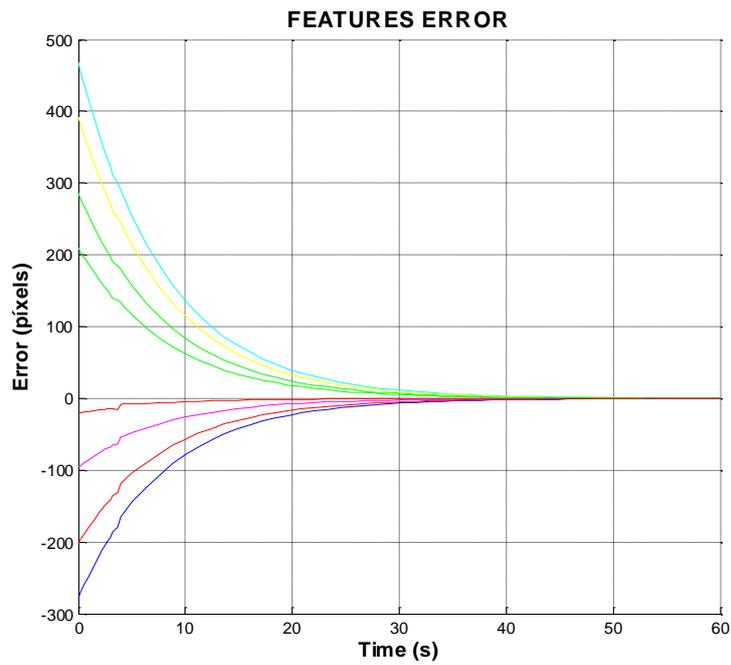


Figura 59. Error en función del tiempo de las características del objeto.

- Simulaciones con $\hat{L}_s^+ = L_{s^*}^+$: como ya se ha visto en el apartado de estimación de la matriz de interacción, este enfoque estipula que la distancia a los puntos sea la de la posición final, es decir $Z = Z^*$. Lo que se hace en este enfoque es mantener fija la matriz de interacción durante toda la terea de convergencia. La Figura 60 muestra el comportamiento del sistema bajo este enfoque. En ella se puede ver la trayectoria seguida por cada uno de los puntos durante la simulación desde la posición inicial a la posición final.

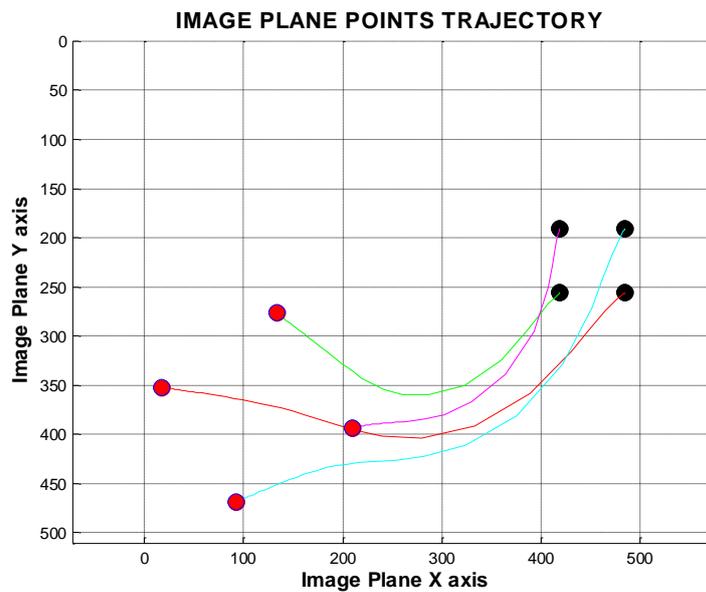


Figura 60. Trayectoria seguida en el plano de imagen desde la configuración inicial con $\hat{L}_s = L_s^-$: puntos en negro, hasta la configuración final, puntos en rojo.

Como era de esperar, la trayectoria ya no sigue una línea recta entre los puntos inicio y final. Aunque como se vio en la teoría, cuando más se acerca a la convergencia la trayectoria seguida en el plano de imagen más se asemeja a una línea recta (ver en la Figura 60 cuando la trayectoria se acerca a los puntos finales, puntos rojos). Esto es así y es lógico que lo sea por el siguiente motivo: en las inmediaciones de la convergencia, $L_s^+ \approx L_{s^*}^+$.

La Figura 61 muestra la trayectoria seguida por la cámara en el espacio cartesiano. Se puede ver que es diferente a la que se tenía para en el caso anterior.

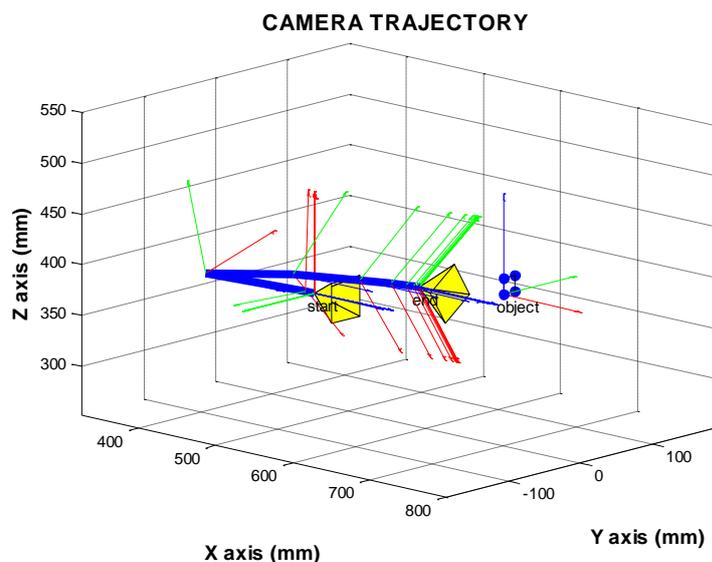


Figura 61. Trayectoria seguida por la cámara en el espacio cartesiano durante la tarea de *matching* con $\hat{L}_i = L_i^*$.

Por último se muestra el error de convergencia de las características del objeto en función del tiempo. Hay que darse cuenta que debido a que ahora la trayectoria no sigue una línea recta en el plano de imagen, se necesita de un mayor tiempo de convergencia del algoritmo. En este caso, bajo las mismas circunstancias se ve que es de alrededor de 10 segundos más lenta la convergencia con este enfoque.

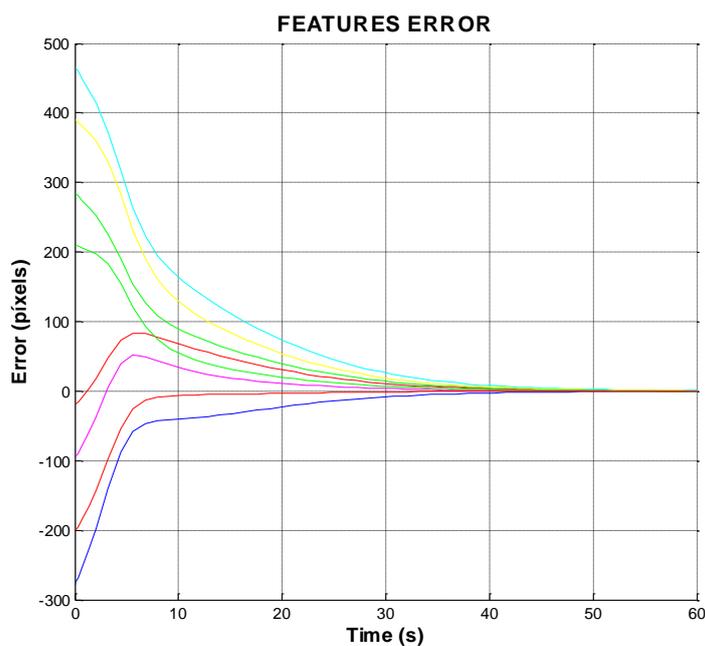


Figura 62. Error de convergencia de las características del objeto en función del tiempo con $\hat{L}_i = L_i^*$.

- Simulaciones con $\hat{L}_s^+ = \frac{1}{2}(L_s + L_{s^*})$: este enfoque es un paso intermedio entre los dos anteriores. Tiene en cuenta tanto la estimación de la distancia a los puntos Z en cada iteración y así obtener la matriz de interacción L_s . Además es necesario conocer la Z^* de la posición final deseada. Con esto, la Figura 63 muestra la trayectoria que siguen los puntos en el plano de imagen. Se ve claramente como su comportamiento está entre los dos anteriores y el tiempo de convergencia también lo debe estar.

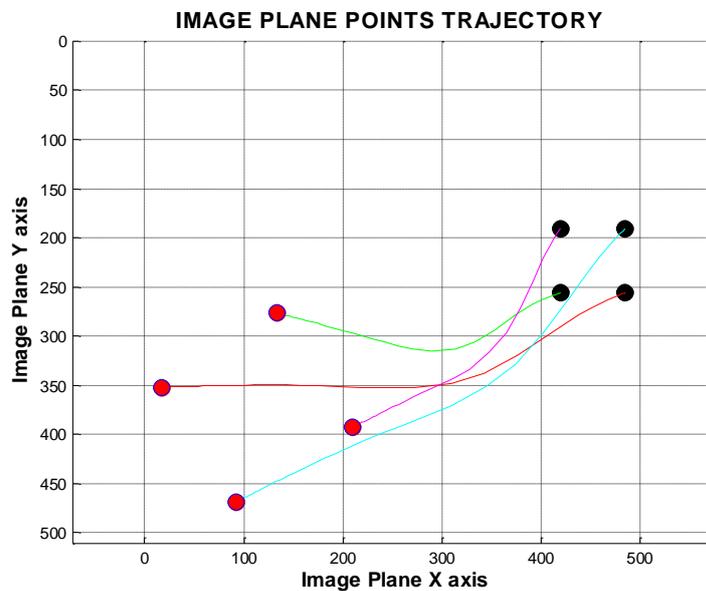


Figura 63. Trayectoria seguida en el plano de imagen con $\hat{L}_s^+ = \frac{1}{2}(L_s + L_{s^*})$.

Por otro lado, la trayectoria seguida por la cámara en el plano cartesiano se muestra en la Figura 65. Se ve también bastante claro, comparándola con la de los casos anteriores, que es una trayectoria intermedia entre ambos casos.

Finalmente la Figura 64 muestra el error de convergencia de las características del objeto en función de tiempo. En este caso el tiempo de convergencia es ligeramente superior al del caso en que $\hat{L}_s^+ = L_s^+$ pero ligeramente inferior al del caso en que $\hat{L}_s^+ = L_{s^*}^+$.

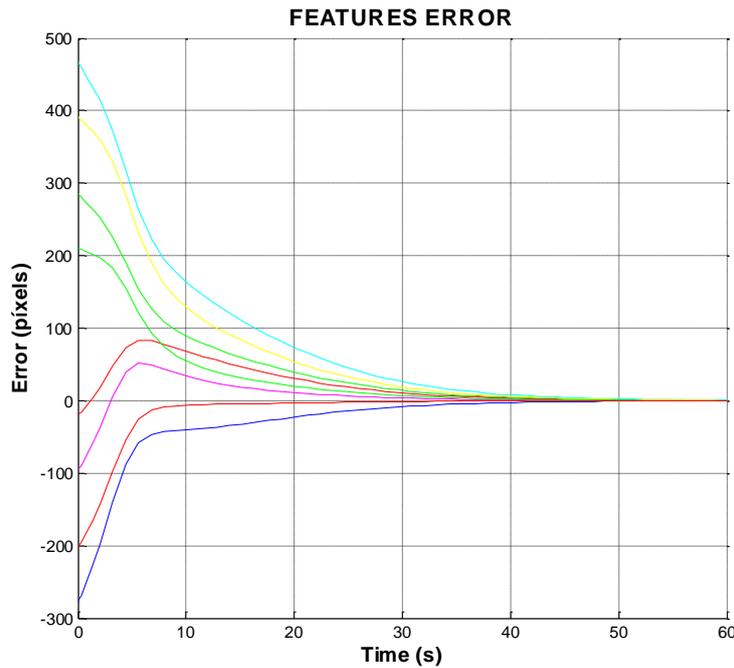


Figura 64. Error de convergencia en función del tiempo de las características del objeto par el caso

$$\hat{L}_s = \frac{1}{2}(L_s + L_{s'}).$$

Una vez estudiado el sistema IBVS con un objeto estático, es decir con una posición deseada fija y el objeto sin moverse se hace necesario, para la realización de posteriores estudios, de ver el comportamiento que tiene el sistema cuando se tiene que realizar el seguimiento de un objeto móvil.

3.2.3.1.2.- Objeto dinámico

Con el objeto de realizar un ejemplo más sencillo para entender perfectamente el comportamiento del sistema, se optó por realizar el seguimiento de una pelota. La ventaja de este tipo de experimentos es que al ser más sencillos se puede simplificar el problema para abordarlo de distintas formas realizando un mejor estudio del mismo y, en consecuencia, tener una mejor comprensión del sistema. La desventaja de es que, para controlar los seis grados de libertad que dispone el robot (tres de translación y tres de rotación) son necesarios, al menos, obtener 4 puntos del objeto. Esto no es importante para el estudio que se está realizando ya que, por ahora, lo que se intenta ver es cuánto más rápido es capaz de realizar un seguimiento el robot de un objeto en movimiento. Para futuras tareas el *grasping* debería realizado, lo que implica controlar tanto la translación como la orientación.

Por lo tanto, y con objeto de ver en primer lugar el comportamiento del sistema en simulación se tiene el siguiente problema a resolver: seguimiento de una pelota.

3.2.3.1.3.- Seguimiento de una pelota

La Figura 65 muestra el objeto al que se debe realizar el seguimiento. En este caso se muestra el centro de masas de la pelota de una posición inicial a otra deseada, por ejemplo centrar siempre la pelota en el centro de la imagen.

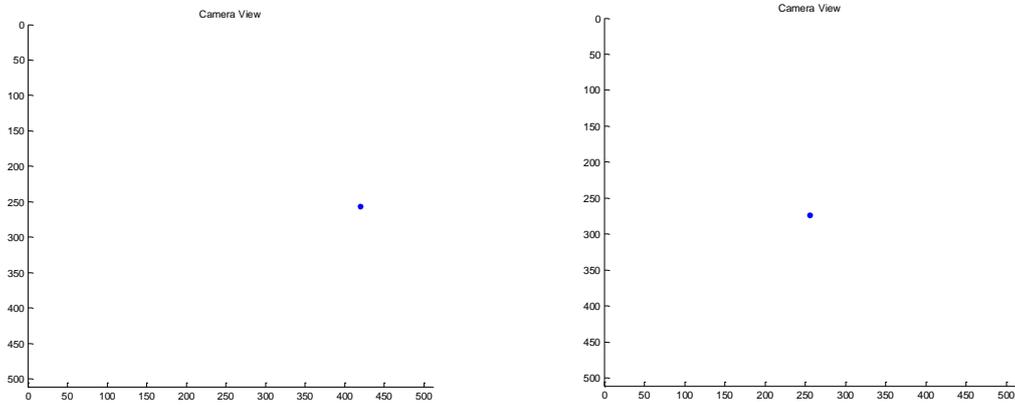


Figura 65. Posición inicial y deseada del centro de masas de la pelota: la imagen de la izquierda muestra la posición inicial y la imagen de la derecha muestra la posición deseada del centro de masas de la pelota.

Con esta configuración, si se realiza una simulación con la pelota quieta, se tienen los resultados que se describen a continuación. La Figura 66 muestra la trayectoria seguida en el plano de imagen. Se trata de una trayectoria recta desde la posición inicial a la deseada.

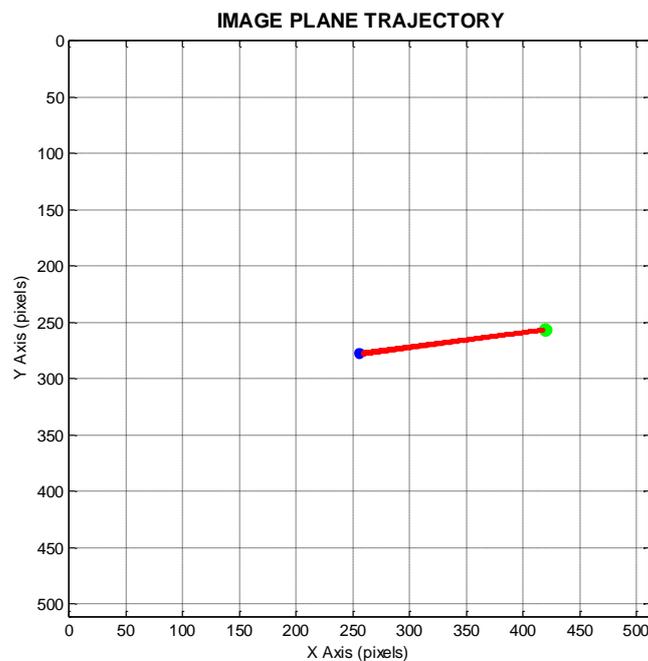


Figura 66. Trayectoria seguida desde la posición inicial a la deseada en el plano de imagen.

Del mismo modo y, debido a que este caso no necesita ninguna rotación la Figura 67 muestra que el comportamiento del sistema es también una trayectoria recta entre la posición inicial y la deseada. Esto no tiene por qué ser así en todos los casos. Depende también del movimiento que ha seguido el objeto, es decir si ha estado siempre moviéndose en el mismo plano (como es este caso) o bien se mueve en diferentes planos, cambiando el valor de la distancia a la cámara, etc.

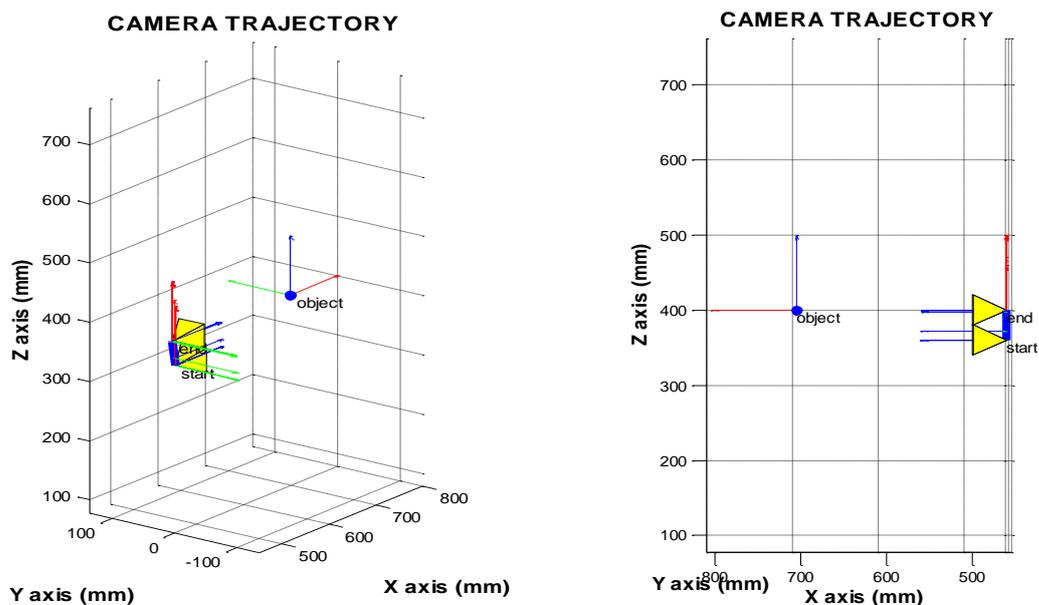


Figura 67. Trayectoria seguida por la cámara en el espacio cartesiano para el seguimiento de la pelota.

Una vez estudiado el comportamiento cuando la pelota está estática, se va a realizar el estudio cuando ésta está en movimiento.

3.2.3.1.4.- Estudio con la pelota en movimiento oscilatorio

Por lo tanto, como se ha dicho, en este apartado se va a ver el comportamiento en simulación del IBVS cuando se tiene una pelota que está en continuo movimiento. Este movimiento se ha decidido que sea oscilatorio. Para una mejor visualización de los resultados se va a dividir el problema en dos sub-problemas: cuando el objeto se mueve sólo horizontalmente, y cuando el objeto se mueve verticalmente, siempre desde el punto de vista del plano de imagen:

- Objeto moviéndose horizontalmente: la Figura 68 muestra una secuencia de imágenes que muestran el movimiento que está siguiendo el objeto a lo largo del experimento. Se puede ver que está realizando un movimiento oscilatorio (de vaivén) horizontalmente.

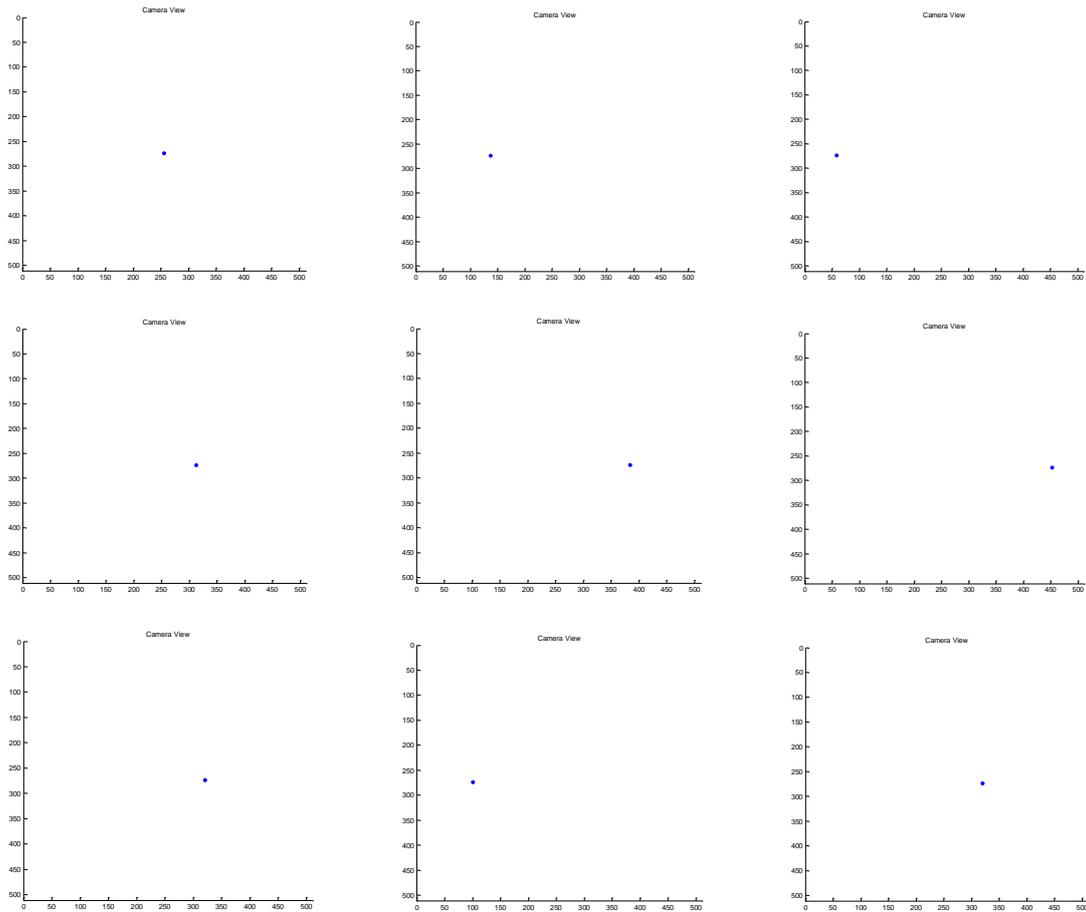


Figura 68. Pelota moviéndose horizontalmente.

Como en este caso de los dos parámetros $c_p = [x \ y]$, sólo se tiene el movimiento de uno de los dos parámetros (en este caso el de x , véase en la Figura 69), a partir de este punto se va a mostrar la trayectoria de la componente en función del tiempo de ejecución. Además, como se trata de realizar el seguimiento, se va a poner también la trayectoria real del objeto para poder estudiar mejor los resultados obtenidos.

Además se hará un estudio hasta ahora no realizado, y es el de ver cómo se están moviendo cada una de las articulaciones del robot así como una secuencia del movimiento del mismo. Esto permitirá observar mejor lo que realmente está pasando con el robot, para que quede aún más claro que no se está trabajando en *cámara libre*, sino que la cinemática del robot está influyendo en el sistema a la hora de realizar la tarea de seguimiento.

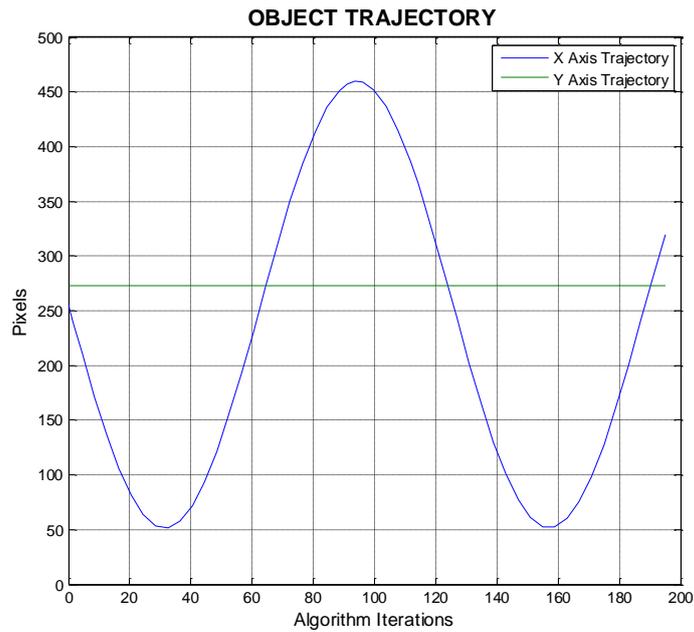


Figura 69. Trayectoria de la pelota en función del tiempo desde el punto de vista del plano de imagen.

Por lo tanto, una vez realizada las aclaraciones del cómo se van a mostrar los resultados, ya se está en disposición de verlos. Para este caso particular, la Figura 70 muestra la trayectoria seguida por la cámara en función de la trayectoria seguida por la pelota. Se puede ver que, aunque con un cierto *delay* producido claro está por el tiempo en que se tarda en adquirir, procesar y calcular la señal de control, se está realizando un buen seguimiento de la pelota.

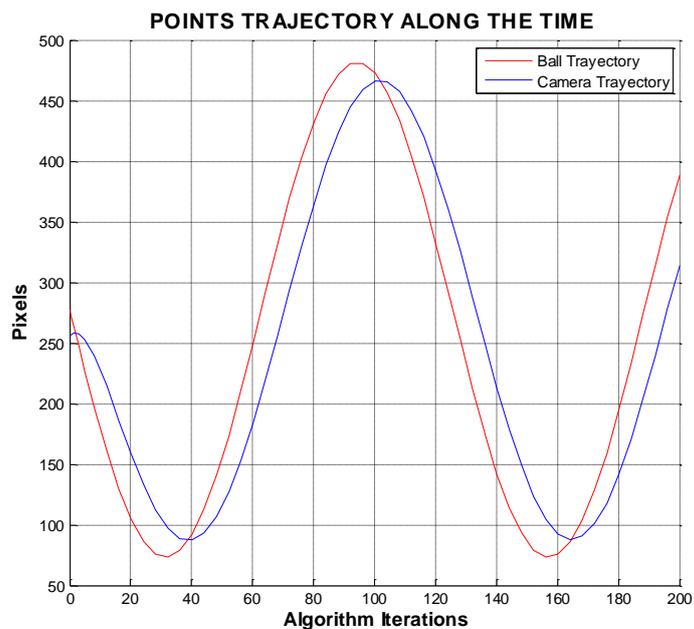


Figura 70. Seguimiento de la pelota en movimiento en función del tiempo.

Por otra parte, la trayectoria seguida por la cámara en el espacio cartesiano se puede ver en La Figura 71. En primer lugar realiza un posicionado para centrar el eje y en el plano de imagen y luego, al estar cambiando continuamente el eje x se hace el movimiento de vaivén para realiza la tarea de seguimiento.

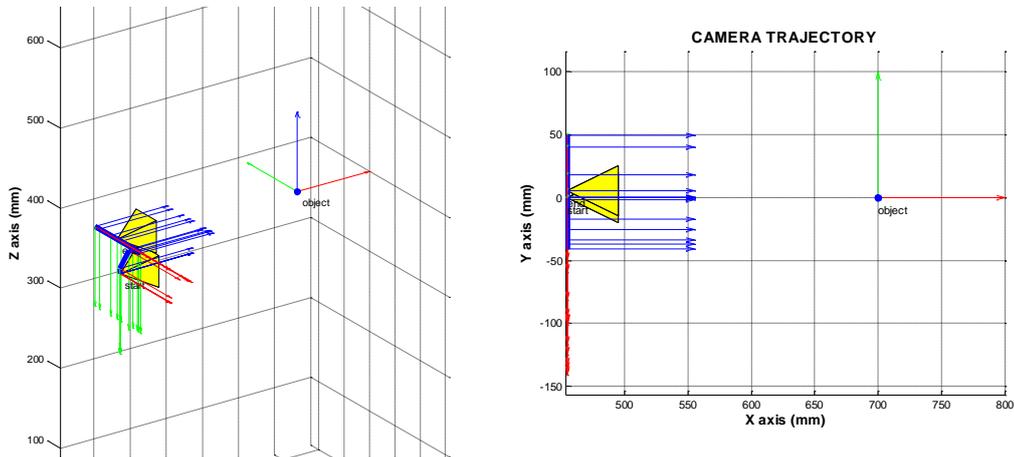


Figura 71. Trayectoria de la cámara en el plano cartesiano.

La Figura 72 muestra una secuencia de imágenes del seguimiento realizado por el robot.

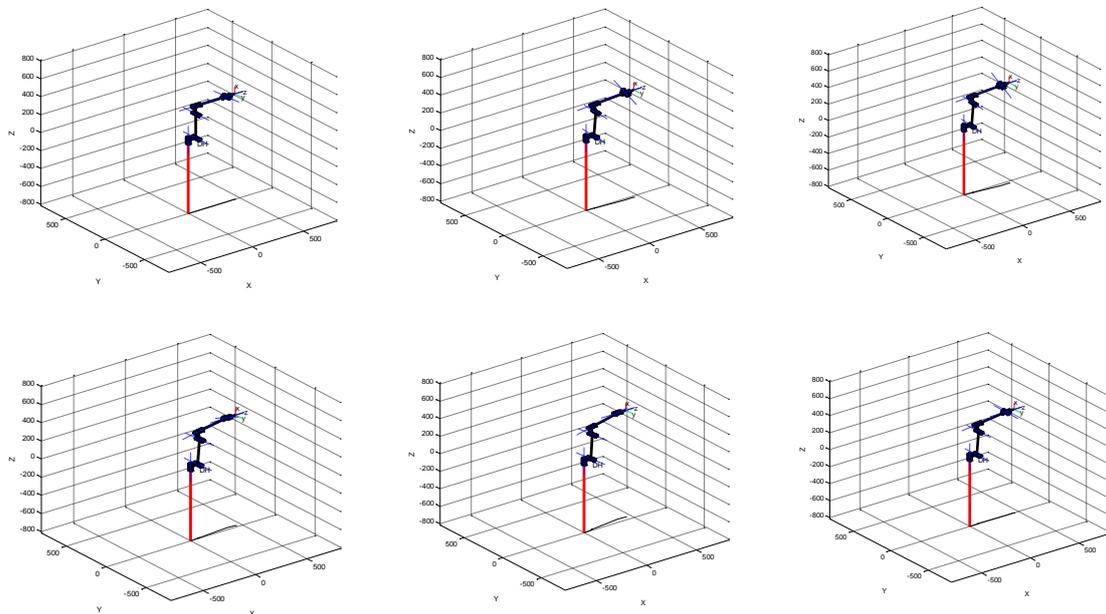


Figura 72. Secuencia de imágenes del robot siguiendo la pelota.

Más interesante es ver el comportamiento que está teniendo cada una de las articulaciones en función del tiempo, que es lo que se muestra en la Figura 73. Se puede ver claramente el movimiento de vaivén que están sufriendo cada una de las articulaciones. Significativo es el

movimiento seguido por las articulaciones 4 y 6. Ambas se mueven prácticamente con el mismo valor en cada instante de tiempo para poder mantener la cámara en el mismo plano en el que se encuentra mientras se está realizando la tarea de seguimiento.

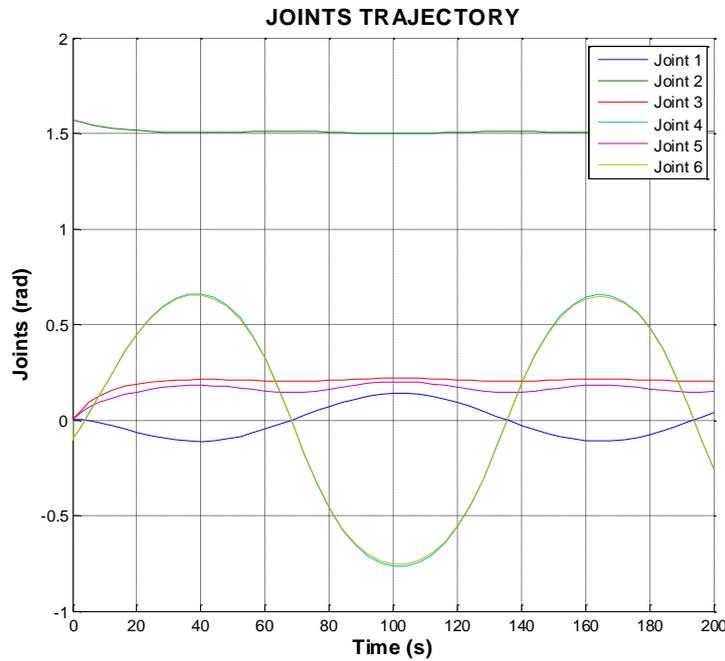


Figura 73. Trayectoria de las articulaciones del robot en función del tiempo.

- Objeto moviéndose verticalmente: en este caso el movimiento de la pelota se puede observar en la secuencia de imágenes mostradas en la Figura 74.

Por lo tanto se tiene un caso similar al anterior en todos los sentidos. La Figura 75 muestra como en este caso la pelota se va a mover a lo largo del eje y mientras que el eje x va a permanecer estático. Esto permitirá ver el comportamiento del robot cuando se está moviendo de arriba abajo, el comportamiento que tiene cada una de las articulaciones del mismo.

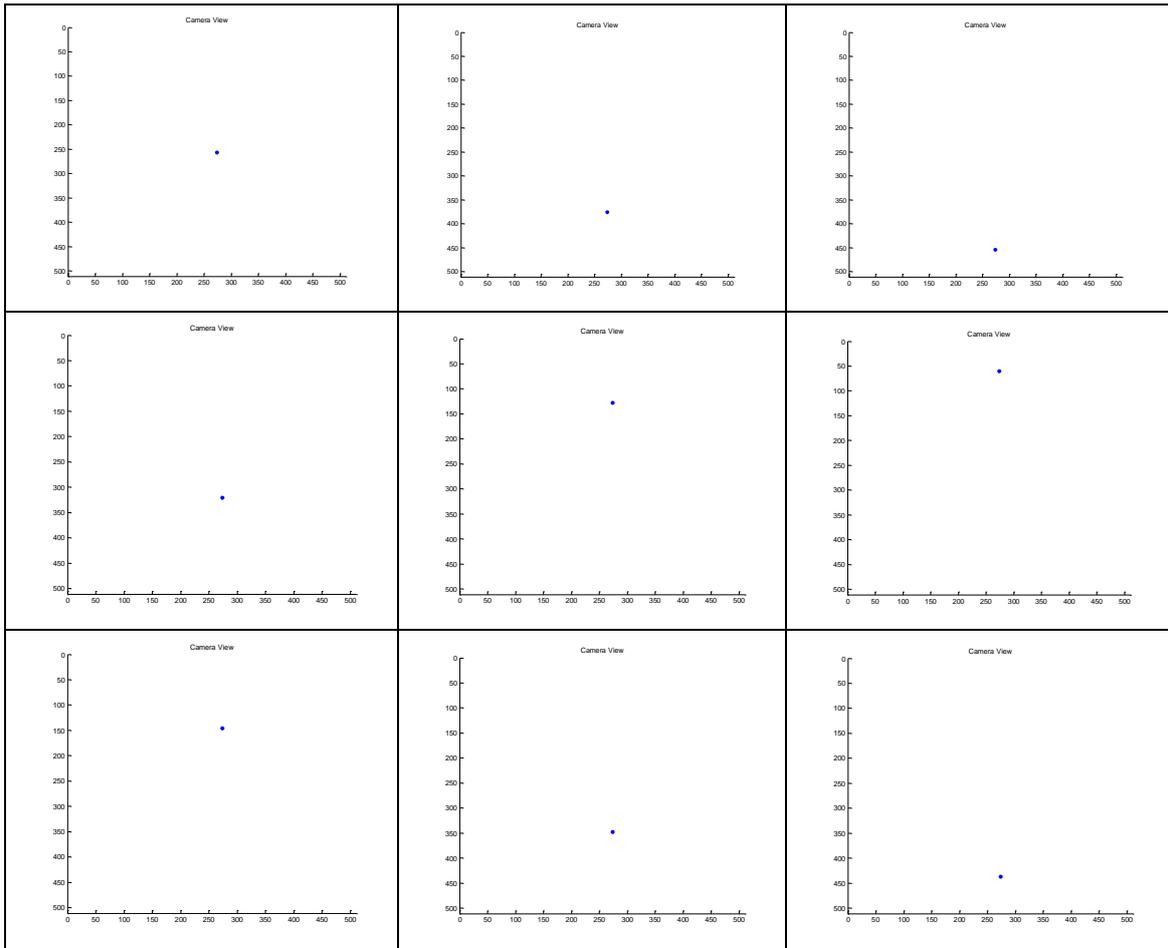


Figura 74. Pelota moviéndose verticalmente.

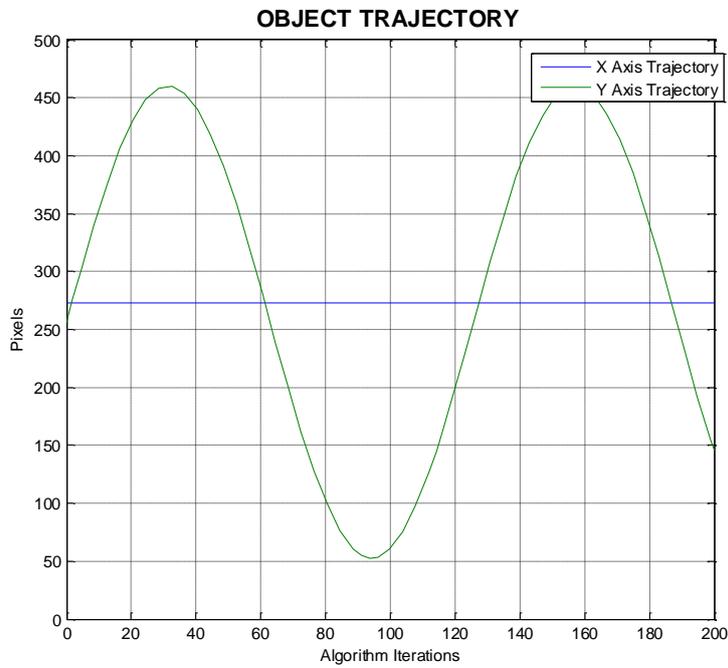


Figura 75. Trayectoria de cada una de las componentes del centro de masas de la pelota a lo largo de la simulación.

Por lo tanto ya se está listo para ver el comportamiento que tiene el sistema bajo estas condiciones de trabajo. Es por ello que la Figura 76 muestra el seguimiento realizado por el robot y la trayectoria de la cámara en cada instante de tiempo. Se puede ver como una vez la componente x alcanza la posición deseada, es el movimiento vertical (de la componente y) del que se está haciendo el seguimiento.

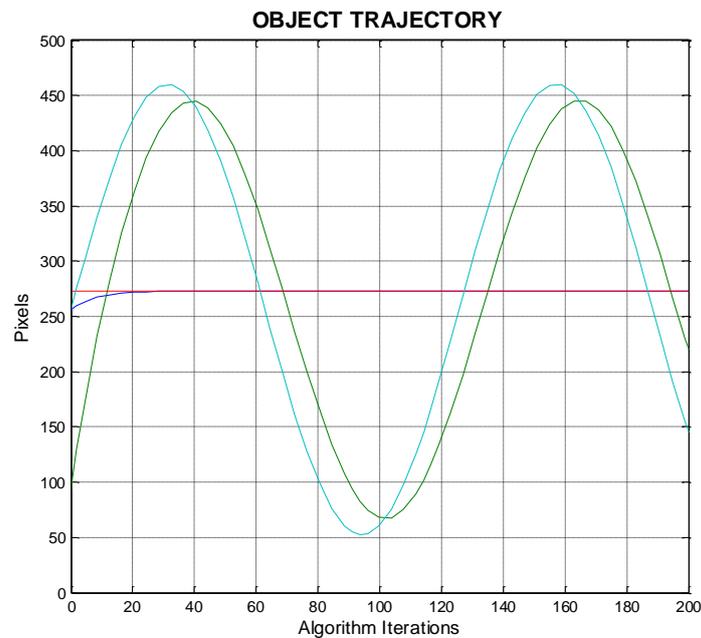


Figura 76. Seguimiento de la pelota moviéndose verticalmente.

Por otra parte, en la Figura 77 se muestra la trayectoria seguida por la cámara a lo largo del tiempo en el espacio cartesiano. En esta figura se puede ver el movimiento de vaivén vertical que está realizando el robot a lo largo de la simulación. Par observar mejor este proceso, en la Figura 78 se muestra una secuencia de imágenes en la que se ve el robot en dicho proceso de seguimiento vertical.

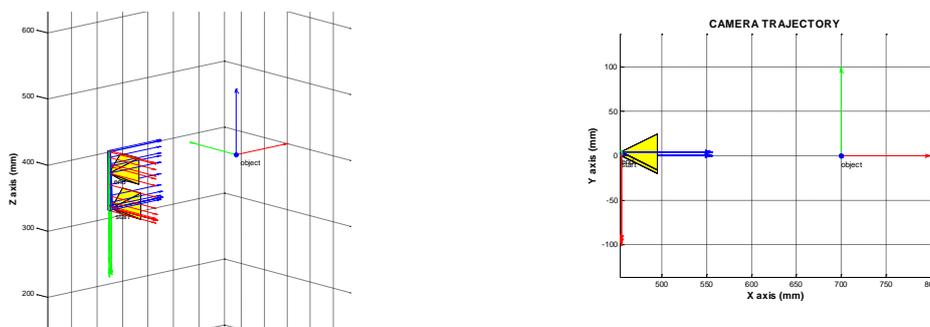


Figura 77. Trayectoria seguida por la cámara en el espacio cartesiano.

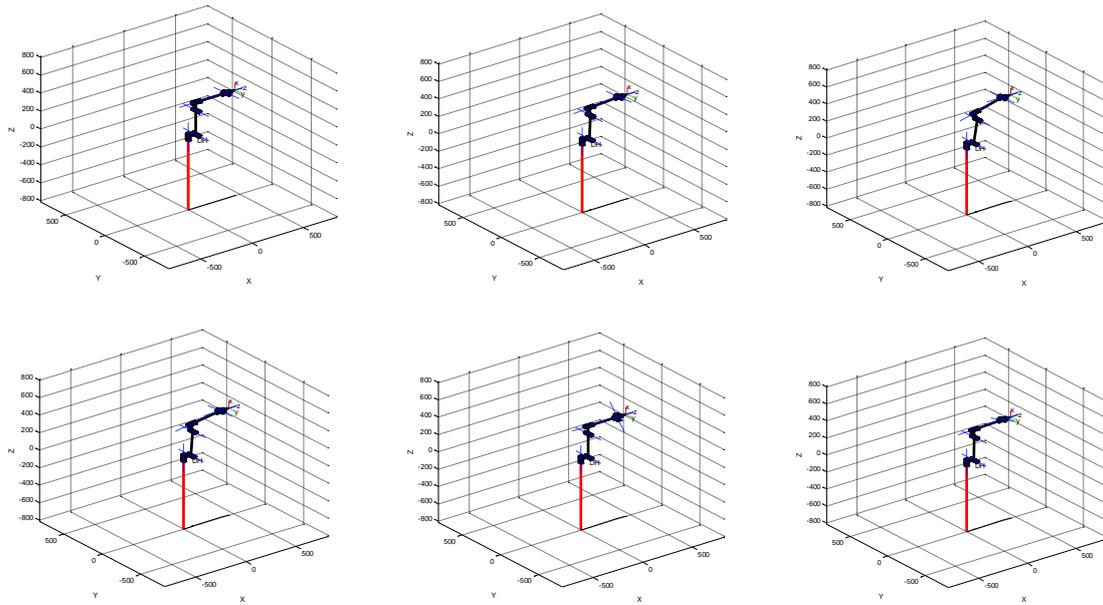


Figura 78. Secuencia de imágenes sobre el comportamiento del robot para el seguimiento de la pelota en el caso de movimiento vertical.

Para realizar un mejor estudio del comportamiento del Robot, en la Figura 79 se muestra el comportamiento de cada articulación a lo largo del experimento. Como era de esperar hay un comportamiento similar al de caso anterior.

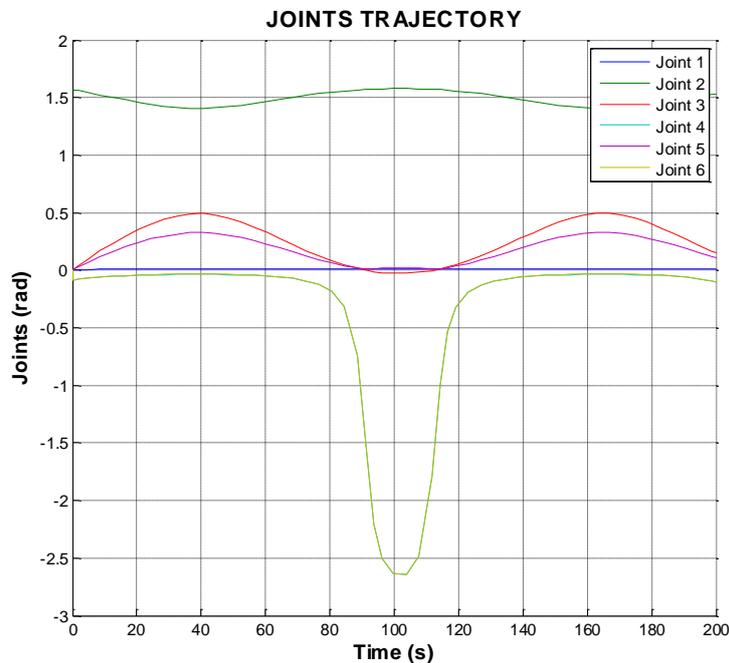


Figura 79. Trayectoria de cada articulación del robot a lo largo del experimento de seguimiento de una pelota con movimiento vertical.

Con estos experimentos se concluye el estudio de la técnica de Image-Based Visual Servoing clásica en simulación, viendo que los resultados obtenidos son los esperados desde el punto de vista teórico.

En el siguiente apartado se realiza la explicación del sistema real, poniendo en práctica los conceptos vistos en simulación. Hay que tener en cuenta de que lo que se pretende es tener un sistema parecido al del seguimiento de la pelota para realizar el estudio experimental de las técnicas multifrecuencia que se introducen y se explican en el Capítulo 4. En él se verá el motivo de querer tener un ejemplo sencillo y en movimiento.

3.3.-Estudio del comportamiento del sistema en implementación real.

Antes de pasar a ver los resultados de los experimentos con el sistema real se va a realizar una descripción del sistema de Visual Servoing que hay en el Instituto de Diseño y Fabricación de la Universidad Politécnica de Valencia.

En el capítulo 5 se explicará más en detalle el sistema y se mostrará la necesidad de introducir el concepto de diseño multifrecuencia para mejorar el comportamiento del mismo.

En este apartado se centrará en explicar el equipo que se dispone y en mostrar los resultados obtenidos con el enfoque monofrecuencia.

3.3.1.-Sistema Visual Servoing

Los elementos básicos que conforman cualquier sistema de Visual Servoing son:

- Cámara: como se ha mencionado anteriormente, aunque en el laboratorio se dispone de cámaras para aplicaciones industriales, se prefirió en su momento trabajar con una Logitech Webcam C300, tal y como se ha mostrado en el apartado anterior. Uno de los motivos es la facilidad con la que se puede trabajar, independientemente del entorno de desarrollo que se esté utilizando.
- Robot: como se ha venido diciendo, el robot que se está utilizando es el Kuka KR5 sixx R650, un robot pequeño montado en una plataforma en configuración techo, tal y como muestra la Figura 80. En la siguiente sección se dará una mayor descripción de los elementos hardware y software que permiten controlarlo en tiempo real.



Figura 80. Robot Kuka KR5 sixx R650 para aplicación de Visual Sevoing.

- Portatil DELL XPS: en definitiva es donde están implementados los distintos algoritmos de control que permiten realizar las tareas al robot. Se decidió trabajar en este portátil y no en una Workstation debido a la facilidad de transporte que se tiene y así poder trabajar con los algoritmos siempre desde un mismo PC (ver Figura 81).



Figura 81. Portátil DELL XPS para aplicación de Visual Servoing.

Además, se ha utilizado un proyector para la realización de los diferentes experimentos. La idea es poder realizar una comparativa teniendo tanto la referencia real del objeto a seguir como el tracking realizado por el sistema Visual Servoing (ver Figura 82). Para futuras tareas de grasping, el Instituto de Diseño y Fabricación dispondrá de herramientas como un disco circular o una cinta transportadora. Actualmente no están disponibles y por lo tanto se ha considerado buena idea la realización de los experimentos con este sistema.



Figura 82. Sistema Visual Servoing para experimentación: la mayor parte de los experimentos se realizarán utilizando un proyector para poder tener una referencia real del objeto y poder luego estudiar los resultados obtenidos.

3.3.1.1.- Configuración Hardware del sistema Visual Servoing

Como se ha mencionado anteriormente, en éste Capítulo de la memoria se va a tratar el sistema como si fuera un sistema muestreado monofrecuencia (como son tratados los sistemas en teoría de control clásica), dejando para el Capítulo 4 el motivo de que éste sea claramente un sistema multifrecuencia.

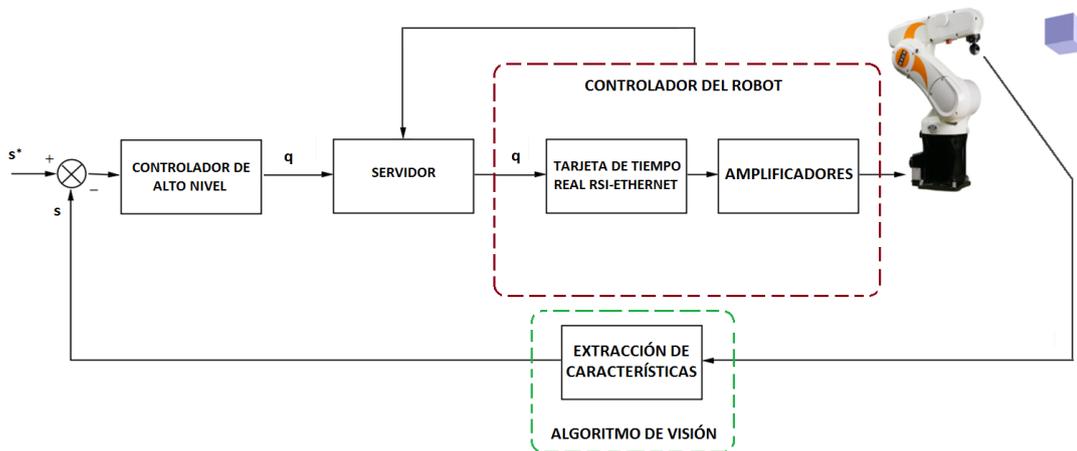


Figura 83. Configuración hardware del sistema Visual Servoing.

En la Figura 83 se muestra el diagrama de bloques relativo a la configuración hardware del sistema Visual Servoing. En ella se diferencian claramente los siguientes elementos:

- *Controlador de alto nivel:* éste hace referencia al PC que gestiona el control del sistema (donde se realizan los algoritmos de control y procesamiento de imagen).
- *Controlador de bajo nivel:* éste hace referencia la unidad de control del robot (ver Figura 84).



Figura 84. Unidad de control KC 2 del Kuka KR5 sixx R650.

- *Robot:* el robot utilizado para la experimentación real es el Kuka KR5 sixx R650.
- *Cámara WebCam Logitech C300.*
- *Servidor:* La comunicación en tiempo real entre el controlador del robot y el controlador de alto nivel se realiza mediante la tarjeta RSI-Ethernet. En el siguiente apartado se va a explicar dicha comunicación de forma más detallada.

3.3.1.1.1.- Comunicación mediante la tarjeta RSI-Ethernet

Para controlar el robot desde un sistema externo de tiempo real, se ha utilizado la tarjeta de Kuka RSI-XML Ethernet. El controlador del robot se comunica con un sistema externo vía un link de red *point-to-point* en tiempo real. El intercambio de datos son realizados mediante los protocolos Ethernet TCP/IP o UDP/IP como cadenas XML. La comunicación realizada mediante la tarjeta de tiempo real RSI-XML Ethernet de Kuka permite las siguientes funciones:

- Trasmisión de datos cíclicamente desde el controlador del robot a un sistema externo en una interpolación cada 12 milisegundos (ejemplo, datos de posicionamiento, ángulos de los ejes, modos de operación, etc.)
- Trasmisión de datos cíclicamente desde un sistema externo al controlador del robot cada 12 milisegundos (ejemplo, datos procedentes de un sensor)

- Controlando el robot cada ciclo de 12 milisegundos
- Intervención directa en el “*path planning*” del robot.

Algunas de las características más relevantes de este sistema de comunicación son:

- Recarga de los objetos RSI para la comunicación con un sistema externo, según KUKA.RobotSensorInterface(RSI)
- Módulo de comunicaciones con acceso al Ethernet estándar
- Libre definición de entradas y salidas
- Intercambio de datos con monitor de timeout.

3.3.1.1.1. Principio de funcionamiento

Si la señal de procesado se activa con el objeto de comunicación ST_COROB o ST_ETHERNET, el controlador del robot se conecta con el sistema externo como si fuera un cliente.

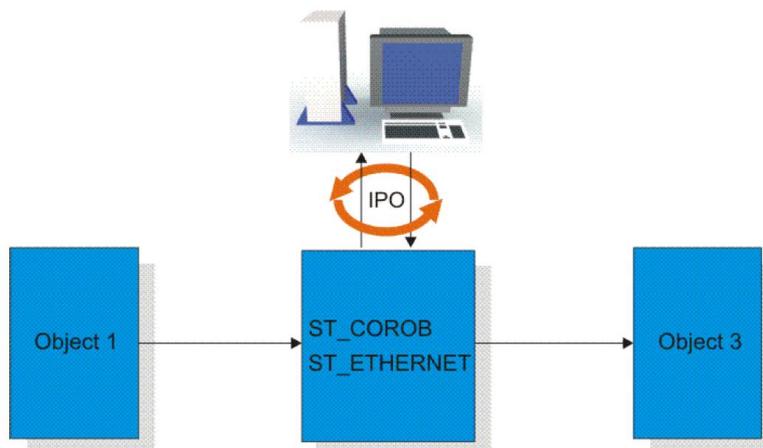


Figura 85. Principio de funcionamiento del intercambio de datos.

El controlador del robot inicia el intercambio de datos de forma cíclica con un paquete de datos KRC y transfiere dicho paquete al sistema externo cada ciclo de 12 milisegundos. El sistema externo debe de responder al paquete de datos KRC recibido con su propio paquete.

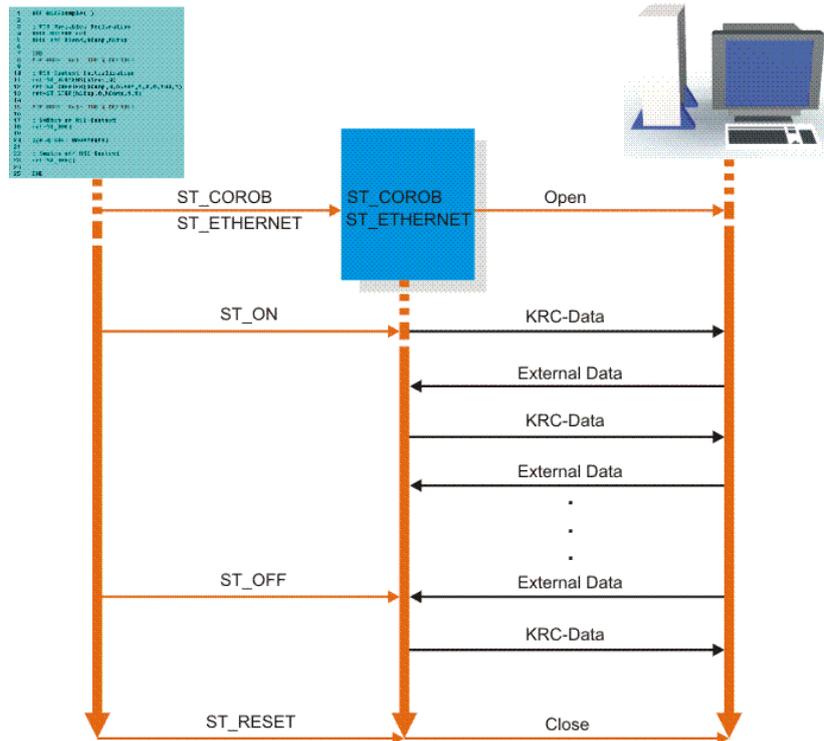


Figura 86. Secuencia de intercambio de datos.

Un paquete de datos recibido por el sistema externo se debe contestar en aproximadamente 10 milisegundos. Si el paquete de datos no es recibido por el controlador del robot en este periodo, la respuesta será clasificada como “too late”. Cuando se alcanza el máximo valor de paquetes perdidos (too late), el robot interpreta que hay algún problema y para su ejecución. Si el objeto de comunicación ST_COROB o ST_ETHERNET se elimina, la conexión entre el controlador del robot y el sistema externo se interrumpe. Ambos lados intercambian datos en forma de cadenas XML.

- ST_COROB o ST_ETHERNET:
 - Estructura de la cadena XML cuando se envía datos

```
- <Rob Type="KUKA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="KrcData.xsd">
- <Dat TaskType="b">
  <ComStatus>continuous</ComStatus>
  <RIst X="1620.0000" Y="1620.0000" Z="1620.0000" A="100.0000"
B="100.0000" C="100.0000" />
  <RSol X="1620.0000" Y="1620.0000" Z="1620.0000" A="100.0000"
B="100.0000" C="100.0000" />
  <AIPos A1="-180.0000" A2="-10.0000" A3="20.0000" A4="20.0000"
A5="20.0000" A6="20.0000" />
  <ASPos A1="-180.0000" A2="-10.0000" A3="20.0000" A4="20.0000"
A5="20.0000" A6="20.0000" />
  <EIPos E1="0.00000" E2="0.00000" E3="0.00000" E4="0.00000"
E5="0.00000" E6="0.00000" />
  <ESPos E1="0.00000" E2="0.00000" E3="0.00000" E4="0.00000"
E5="0.00000" E6="0.00000" />
  <MACur A1="1620.0000" A2="1620.0000" A3="1620.0000" A4="1620.0000"
A5="1620.0000" A6="1620.0000" />
  <MECur E1="1620.0000" E2="1620.0000" E3="1620.0000" E4="1620.0000"
E5="1620.0000" E6="1620.0000" />
  <IPOC>64</IPOC>
  <BMode>5</BMode>
  <IPOSstat>255</IPOSstat>
  <Tech x="1" p6="0" p7="0" p8="0" p6x1="0" p7x1="0"
p8x1="0" p6x2="0" p7x2="0" p8x2="0" p6x3="0" p7x3="0" p8x3="0" />
  <RGH X="234" Y="12332" Z="223" A="132" B="123" C="-123" T="10101"
/>
  <DiI>255</DiI>
  <Tick>0</Tick>
  <RWMode>C</RWMode>
</Dat>
</Rob>
```

Figura 87. Estructura XML cuando se envían datos con el objeto ST_COROB.

- Estructura de la cadena XML cuando se reciben datos

```
- <Sen Type="CoRob" xmlns:xsi="http://www.w3.org/2001/XMLSchema-in-
stance" xsi:noNamespaceSchemaLocation="ExternalData.xsd">
- <Dat TaskType="b">
  <Estr>Info: Nur zur Info</Estr>
  <RKorr X="0.1620 Y="0.1620 Z="0.1620 A="0.2000" B="0.2000" C="-
0.2000" />
  <AKorr A1="20.0000" A2="20.0000" A3="20.0000" A4="20.0000"
A5="20.0000" A6="20.0000" />
  <EKorr E1="120.0000" E2="120.0000" E3="120.0000" E4="120.0000"
E5="120.0000" E6="120.0000" />
  <Tech x="2" p3="0" p4="0" p5="0" p3x1="0" p4x1="0" p5x1="0"
p3x2="0"
p4x2="0" p5x2="0" p3x3="0" p4x3="0" p5x3="0" />
  <DiO>255</DiO>
  <IPOC>64</IPOC>
</Dat>
</Sen>
```

Figura 88. Estructura XML cuando se reciben datos con el objeto ST_COROB.

En el caso que se plantea en esta tesina, se ha realizado la programación de la comunicación mediante el objeto ST_ETHERNET. A continuación se hablará en detalle de cómo se ha de proceder para su correcta programación y posterior funcionamiento del sistema.

3.3.1.1.1.2. Objeto RSI ST_ETHERNET

La comunicación en tiempo real entre el controlador del robot y el sistema externo se realiza mediante el objeto ST_ETHERNET, que es una ampliación del ST_COROB. Este objeto debe crearse y ser vinculado en el programa KRL. Los elementos del objeto son:

- Parámetros de instanciado, que se asignan cuando se crea el objeto ST_ETHERNET para inicializar el objeto RSI.
- Parámetros objeto, para adaptar la función del ST_ETHERNET.
- Objetos de entrada, para cargar datos desde el contexto RSI y devolverlos al sistema externo.
- Objetos de salida, para redirigir los datos recibidos desde el sistema externo a los objetos RSI.

Con la señal de procesado activada, el objeto ST_ETHERNET siempre envía y recibe un conjunto de datos definidos por el usuario en cada ciclo. A diferencia del objeto ST_COROB donde el conjunto de datos es fijo, no es necesario especificar ningún conjunto de datos cuando se trabaja con el objeto ST_ETHERNET. El usuario debe configurar el conjunto de datos en un fichero XML.

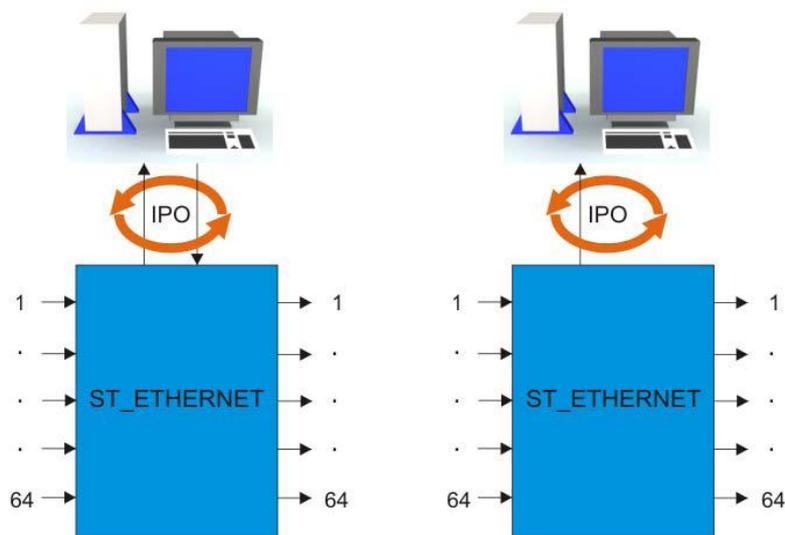


Figura 89. Objeto RSI ST_ETHERNET.

3.3.1.1.1.2.1. Creación de un objeto ST_ETHERNET

El objeto ST_ETHERNET se crea de acuerdo con **KUKA.RobotSensorInterface(RSI)** por medio de una línea de comando en el programa del KRL. Cuando dicho objeto es creado, se deben asignar los parámetros de instanciado para poder inicializar el objeto RSI.

En la creación del objeto, la conexión con el sistema externo se establece. Se pueden utilizar los siguientes protocolos de comunicación para establecerla:

- Protocolo TCP: Aparece un vínculo con el sistema externo.
- Protocolo UDP: No existe vínculo con el sistema externo. La comunicación se testea con un test de ciclo.

- Protocolo UDP y modo de transmisión “Only Send”: no es posible verificar cuando la comunicación con el sistema externo se establece.

La sintaxis es la siguiente:

<Return value>=ST_ETHERNET(Object_ID,Container_ID,Configuration_file[])

PARÁMETROS	Descripción
<Return value>	Contiene el código de error después de que el comando RSI se haya ejecutado.
Object_ID	ID del objeto para tener acceso al objeto RSI. El valor de la variable se asigna automáticamente por el sistema robot cuando se crea el objeto RSI.
Container_ID	Número del contenedor en el que se crea el objeto RSI.
Configuration_file[]	Carácter variable para el nombre del fichero de configuración. El nombre de este fichero puede ser también puesto directamente.

Tabla 7. Explicación de la sintaxis.

```

1  DEF Program( )
2  DECL RSIERR RET
3  INT hEthernet
4
5  ...
6
7  INI
8
9  ...
10
11 RET = ST_ETHERNET(hEthernet,0,"ERMconfig.xml")
12
13 ...
14
15 END
    
```

Figura 90. Ejemplo de código a realizar en el programa KRL.

3.3.1.1.1.2.2. Configuración del ST_ETHERNET

Los parámetros del objeto RSI ST_ETHERNET se pueden modificar para adaptar la funcionalidad de la secuencia del programa. Son un conjunto de parámetros que se modifican con la función ST_SETPARAM. Para más información ver el Anexo 3.

```

1  DEF Program( )
2  DECL RSIERR RET
3  INT hEthernet
4
5  ...
6
7  INI
8  ...
9  ...
10
11 RET=ST_ETHERNET(hEthernet,0,"ERXconfig.xml")
12
13 ...
14
15 RET=ST_SETPARAM(hEthernet,eERXmaxLatePackages,3)
16 RET=ST_SETPARAM(hEthernet,eERXmaxLateInPercent,8)
17 RET=ST_SETPARAM(hEthernet,eERXmaxFieldOfView,2345)
18 RET=ST_SETPARAM(hEthernet,eERXFastCycle,1)
19 RET=ST_SETPARAM(hEthernet,eERXErrorFlag,99)
20
21 ...
22
23 END

```

Figura 91. Ejemplo de modificación de los parámetros del objeto ST_ETHERNET.

3.3.1.1.1.2.3. Definición archivo de configuración

Para habilitar la comunicación entre el controlador del robot y el sistema externo, el usuario debe definir un XML en el directorio C:\KRC\ROBOTER\INIT. El fichero de configuración se especifica y carga cuando el objeto RSI ST_ETHERNET se crea.

```

<ROOT>
<CONFIG></CONFIG>
<SEND>
<ELEMENTS></ELEMENTS>
</SEND>
<RECEIVE>
<ELEMENTS></ELEMENTS>
</RECEIVE>
</ROOT>

```

Figura 92. Estructura del fichero XML.

3.3.1.1.1.2.4. Parámetros de comunicación

La siguiente tabla muestra los parámetros de comunicación que se pueden utilizar en la sección <CONFIG>...</CONFIG>.

PARAMETER	Descripción
<i>IP_NUMBER</i>	Dirección IP del sistema externo.
<i>PORT</i>	Número del puerto del sistema externo.
<i>PROTOCOL</i>	Protocolo: <ul style="list-style-type: none"> • TCP • UDP
<i>SENTYPE</i>	Identificador del sistema externo; seleccionable. El controlador del robot chequea este identificador en cada paquete de datos recibido.
<i>PROTCOLLENGTH</i>	Longitud del byte de transmisión del protocolo antes que el XML se envíe. Éste puede simplificar la programación del stream del socket. <ul style="list-style-type: none"> • ON: La longitud es enviada. • OFF: La longitud no se envía.
<i>ONLYSEND</i>	Dirección del intercambio de datos: <ul style="list-style-type: none"> • TRUE: el controlador del robot envía datos y no debe recibir ningún dato. • FALSE: el controlador del robot envía y recibe datos (por defectos).

Tabla 8. Parámetros de configuración de la comunicación con el sistema externo.

```

1 <CONFIG>
2 <IP_NUMBER>192.0.1.2</IP_NUMBER>
3 <PORT>6008</PORT>
4 <PROTOCOL>TCP</PROTOCOL>
5 <SENTYPE>ImFree</SENTYPE>
6 <PROTCOLLENGTH>off</PROTCOLLENGTH>
7 <ONLYSEND>FALSE</ONLYSEND>
8 </CONFIG>

```

Figura 93. Estructura del fichero XML con la configuración de los parámetros de comunicación.

3.3.1.1.1.2.5. Objetos de entrada del ST_ETHERNET

Hay que configurar la estructura XML para el envío de datos, que permite hasta 64 objetos de entrada. Para esto, las entradas son conectadas a los objetos RSI desde el contexto RSI. El formato XML de envío se genera automáticamente por medio del controlador del robot de acuerdo con la configuración. Estos datos se envían como un string XML al sistema externo.

En la sección <SEND>...</SEND> pueden definirse los siguientes parámetros de la señal de entrada al objeto RSI:

PARAMETER	Descripción
TAG	<p>Nombre del tag a ser generado. Son posibles las siguientes notaciones:</p> <ul style="list-style-type: none"> • TAG="Out". Se genera el siguiente tag: <Out>...</Out> • TAG="Out.o1". Se genera el siguiente tag: <Out o1="" />
TYPE	<p>Tipo de datos de la señal de entrada. Se permiten los siguientes tipos:</p> <ul style="list-style-type: none"> • BOOL • LONG • FLOAT • DOUBLE
INDX	Número del objeto de entrada. Dichos números han de ser consecutivos.
UNIT	Unidad de la señal de entrada.

Tabla 9. Parámetros de los objetos de entrada.

```

<SEND>
<ELEMENTS>
<ELEMENT TAG="Out.o1" TYPE="BOOL" INDX="1" UNIT="5467" />
<ELEMENT TAG="Out.o2" TYPE="BOOL" INDX="2" UNIT="5467" />
<ELEMENT TAG="Out.o3" TYPE="BOOL" INDX="3" UNIT="5467" />
<ELEMENT TAG="Out.o4" TYPE="BOOL" INDX="4" UNIT="5467" />
<ELEMENT TAG="Out.o5" TYPE="BOOL" INDX="5" UNIT="5467" />
<ELEMENT TAG="FTC.Fx" TYPE="FLOAT" INDX="6" UNIT="5467" />
<ELEMENT TAG="FTC.Fy" TYPE="FLOAT" INDX="7" UNIT="5467" />
<ELEMENT TAG="FTC.Fz" TYPE="FLOAT" INDX="8" UNIT="5467" />
<ELEMENT TAG="FTC.Mx" TYPE="FLOAT" INDX="9" UNIT="5467" />
<ELEMENT TAG="FTC.My" TYPE="FLOAT" INDX="10" UNIT="5467" />
<ELEMENT TAG="FTC.Mz" TYPE="FLOAT" INDX="11" UNIT="5467" />
<ELEMENT TAG="Override" TYPE="LONG" INDX="12" UNIT="5467" />
</ELEMENTS>
</SEND>
    
```

Figura 94. Estructura XML con la configuración de parámetros del objeto de entrada.

3.3.1.1.1.2.6. Objetos de salida del ST_ETHERNET

Para configurar el XML para la recepción de datos, se pueden definir hasta 64 objetos de salida. El controlador del robot espera un formato XML de acuerdo a la configuración establecida. Los datos de los objetos de salida se envían en formato XML al controlador del robot. Estos parámetros deben definirse en la sección <RECEIVE>...</RECEIVE> del fichero XML.

PARAMETER	Descripción
TAG	<p>Nombre del tag a ser generado. Son posibles las siguientes notaciones:</p> <ul style="list-style-type: none"> • TAG="Out". Se genera el siguiente tag: <Out>...</Out> • TAG="Out.o1". Se genera el siguiente tag: <Out o1="" />
TYPE	<p>Tipo de datos de la señal de entrada. Se permiten los siguientes tipos:</p> <ul style="list-style-type: none"> • BOOL • LONG • FLOAT • DOUBLE
INDX	Número del objeto de entrada. Dichos números han de ser consecutivos.
UNIT	Unidad de la señal de entrada.
HOLDON	<p>Comportamiento del objeto de salida con objeto de invalidar paquetes de datos que llegan tarde. Valores posibles:</p> <ul style="list-style-type: none"> • 0: La salida se resetea. • 1: se mantiene el valor más reciente de salida.

Tabla 10. Parámetros de los objetos de salida.

```

<RECEIVE>
<ELEMENTS>
<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" />
<ELEMENT TAG="AK.A1" TYPE="DOUBLE" INDX="7" UNIT="0" HOLDON="0" />
<ELEMENT TAG="AK.A2" TYPE="DOUBLE" INDX="8" UNIT="0" HOLDON="0" />
<ELEMENT TAG="AK.A3" TYPE="DOUBLE" INDX="9" UNIT="0" HOLDON="0" />
<ELEMENT TAG="AK.A4" TYPE="DOUBLE" INDX="10" UNIT="0" HOLDON="0" />
<ELEMENT TAG="AK.A5" TYPE="DOUBLE" INDX="11" UNIT="0" HOLDON="0" />
<ELEMENT TAG="AK.A6" TYPE="DOUBLE" INDX="12" UNIT="0" HOLDON="0" />
<ELEMENT TAG="EK.E1" TYPE="DOUBLE" INDX="13" UNIT="0" HOLDON="0" />
<ELEMENT TAG="EK.E2" TYPE="DOUBLE" INDX="14" UNIT="0" HOLDON="0" />
<ELEMENT TAG="EK.E3" TYPE="DOUBLE" INDX="15" UNIT="0" HOLDON="0" />
<ELEMENT TAG="EK.E4" TYPE="DOUBLE" INDX="16" UNIT="0" HOLDON="0" />
<ELEMENT TAG="EK.E5" TYPE="DOUBLE" INDX="17" UNIT="0" HOLDON="0" />
<ELEMENT TAG="EK.E6" TYPE="DOUBLE" INDX="18" UNIT="0" HOLDON="0" />
<ELEMENT TAG="D10" TYPE="LONG" INDX="19" UNIT="0" HOLDON="1" />
</ELEMENTS>
</RECEIVE>
    
```

Figura 95. Estructura XML con la configuración de parámetros del objeto de salida.

En el Anexo 4 se muestra el programa XML que se ha implementado para el caso particular que ocupa esta tesina. Por otra parte, en el Anexo 5 se puede ver el programa del controlador del robot efectuado para la adquisición y posterior manejo de los datos intercambiados entre el controlador de alto nivel y el de bajo nivel.

3.3.1.2.- Configuración Software del Sistema Visual Servoing

Todo el desarrollo software para la aplicación real del sistema Visual Servoing se realizó bajo el sistema operativo Linux versión Ubuntu 10.04, ver Figura 96. Este sistema operativo, aunque dista mucho de ser de tiempo real, si que permite trabajar en periodos de tiempo inferiores a 5 milisegundos. Como se verá en el Capítulo 4, este dato va a ser muy importante a la hora de realizar el estudio multifrecuencia.



Figura 96. Escritorio del sistema operativo Ubuntu 10.04

Por otra parte, se decidió implementar los algoritmos en el lenguaje de programación C++ bajo el compilador *NetBeans 6.8* (ver Figura 97). El motivo fue la comodidad a la hora de realizar los proyectos, de forma muy similar a como se hace en Microsoft Visual Studio o en Eclipse.

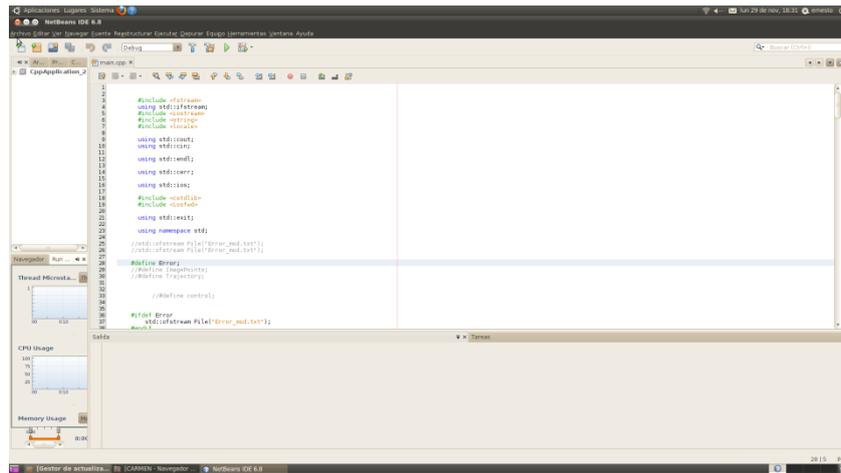


Figura 97. Entorno de trabajo Netbeans 6.8 para LINUX.

3.3.2.-Análisis del comportamiento del IBVS

Como se ha hecho en el caso de las simulaciones se realizarán los experimentos con un objeto que tiene una posición inicial y una deseada, primero con el objeto estático, y luego viendo qué ocurre cuando se encuentra en movimiento.

3.3.2.1.- Objeto estático

Este primer experimento consta de mantener el objeto en una posición fija distinta de la deseada para ver la dinámica que presenta el sistema. Como se ha dicho con anterioridad y con el objeto de realizar experimentos de forma sencilla, el objeto a detectar será un círculo cuya posición deseada será llevar su centro de masas al centro de la imagen, tal y como se muestra en la Figura 98.

Como se aprecia en dicha figura, se ha representado con una cruz verde el centro de masas del objeto detectado y a seguir. Además se ha representado con una cruz roja dentro de una diana azul el centro de la imagen. El algoritmo lo que hará será llevar siempre en centro de la imagen (diana azul) al centro de masas del objeto, minimizando de esta manera el error del posición en el plano de imagen.

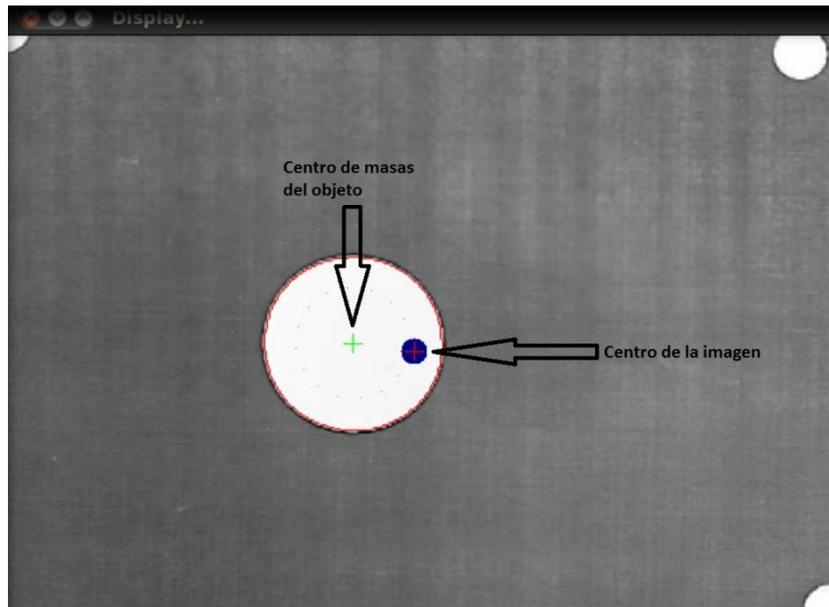


Figura 98. Detección del centro de masas del objeto en el plano de imagen.

La dinámica del sistema se puede ver claramente estudiando la gráfica del error de posición de las características detectadas. En este caso particular dichas características son la posición en x e y del centro de masas del objeto en el plano de imagen. Por lo tanto, las siguientes gráficas muestran dicho error:

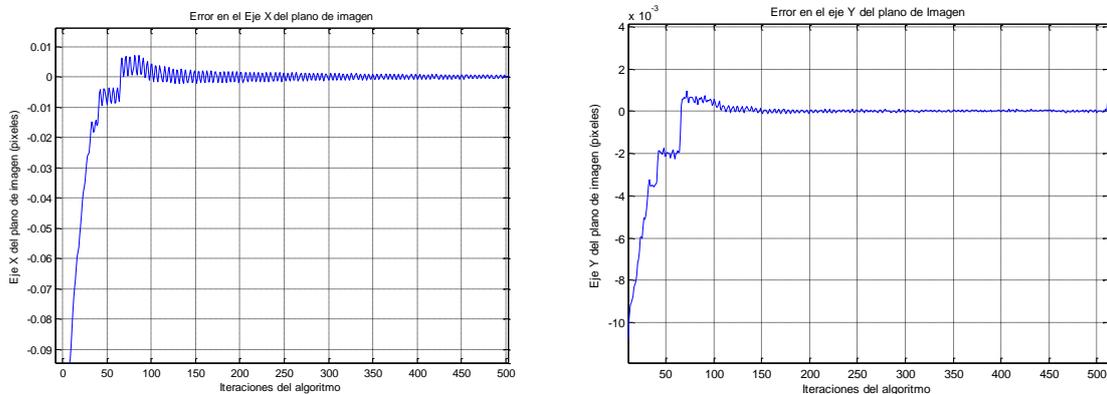


Figura 99. Error de las características detectadas a lo largo de la ejecución del algoritmo IBVS.

Hay varias conclusiones que se pueden realizar a partir de las gráficas mostradas en la Figura 99. La primera de ellas es relativa a la estructura donde el robot está montado. Como se puede observar en la Figura 100, el robot está montado en una estructura de aluminio la cuál en el eje X del robot, es decir cuando el robot avanza hacia la pantalla del proyector está muy bien reforzada, por lo que no hay vibraciones. Por el contrario, la parte que afecta al eje Y del robot no está reforzada, por lo que cada movimiento del mismo en el eje Y provoca una vibración.



Figura 100. Estructura de sujeción del Kuka KR5 sixx R650.

Esto se ve muy claro en la gráfica de la izquierda en la Figura 99. Las oscilaciones que se observan no son debidas a ningún problema del algoritmo de control, sino de la estructura que hace que el robot vibre y por consiguiente desplaza al objeto de su posición deseada.

La otra observación tiene que ver con la dinámica global del sistema. Como se observa en las gráficas el control IBVS no provoca sobreoscilación al sistema para alcanzar la posición deseada. Este resultado concuerda con el obtenido de forma simulada en el que el sistema alcanzaba la posición deseada sin presencia de sobreoscilación en el periodo de transición del estado inicial al estado final.

Por otro lado, las acciones de control proporcionadas por el control de alto nivel se pueden ver en la gráfica de la Figura 101. En ella se puede observar como todos los ejes son movidos por el control para poder alcanzar la pose deseada. Además, se observa que el control es de velocidad por lo que las acciones de control vienen en unidades de $^{\circ}/\text{seg}$.

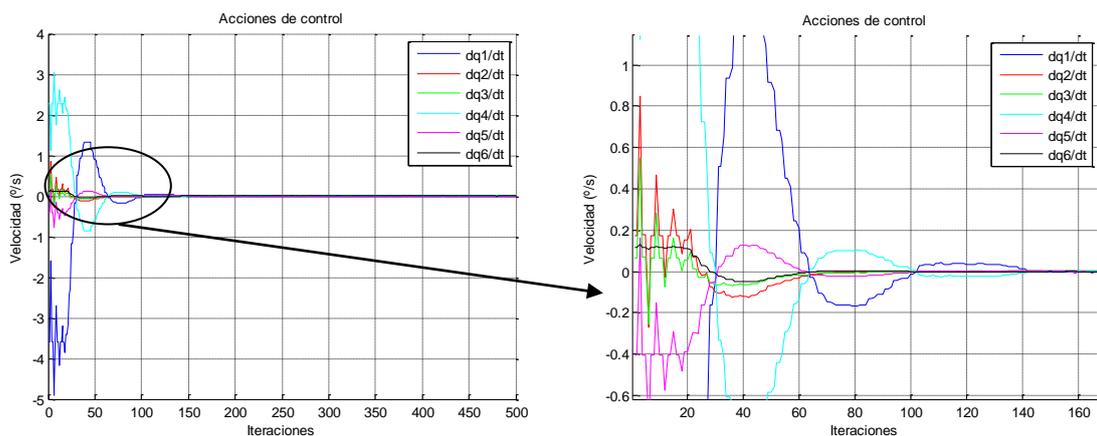


Figura 101. Acciones de control a lo largo del experimento para cada uno de los ejes.

En el siguiente apartado se va a realizar una comparativa más exhaustiva de lo que ocurre cuando el objeto está en constante movimiento con diferentes trayectorias.

3.3.2.2.- Objeto dinámico

En este apartado se va a estudiar el comportamiento del sistema frente a un objeto que está en continuo movimiento. Para ello, y con objeto de realizar un estudio más preciso se ha hecho uso de un proyector (para saber el movimiento del objeto) y se va a estudiar el comportamiento del sistema frente a distintas referencias, tal y como se muestra en la siguiente figura.

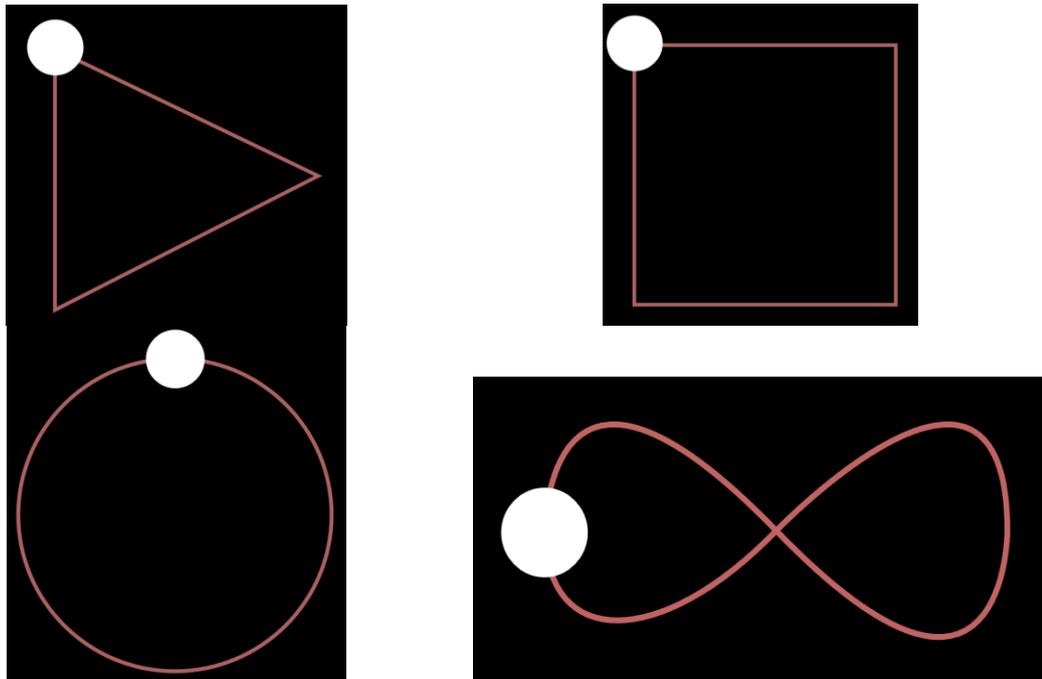


Figura 102. Referencias utilizadas para obtener los resultados reales.

A modo de ejemplo se van a exponer los resultados obtenidos para la trayectoria Triángulo, incluyendo gráficas de error y de control explicando detalladamente el motivo de las formas que presentan. Para no extender demasiado el estudio se realizará una comparativa de cada una de las formas viendo el error máximo que presentan.

Por lo tanto las gráficas de la Figura 103 muestran el error cometido cuando se está realizando una trayectoria triangular con una frecuencia de 30 segundos para finalizarla. Hay que tener en cuenta que estos tiempos son meramente indicativos ya que no se tiene el material preciso para establecer una correcta calibración del tiempo que le cuesta al objeto realizar el recorrido asignado. Es decir, depende de la longitud del recorrido. Es por ello que la velocidad del objeto con periodo 30 segundos cuando realiza la trayectoria triángulo será diferente que la velocidad que tiene cuando realiza la trayectoria infinito con periodo 30 segundos.

Pero lo que en el siguiente capítulo se hará es la comparativa con cada trayectoria en relación a cada uno de los enfoques, lo que sí que dará una idea clara de la mejora del comportamiento del sistema.

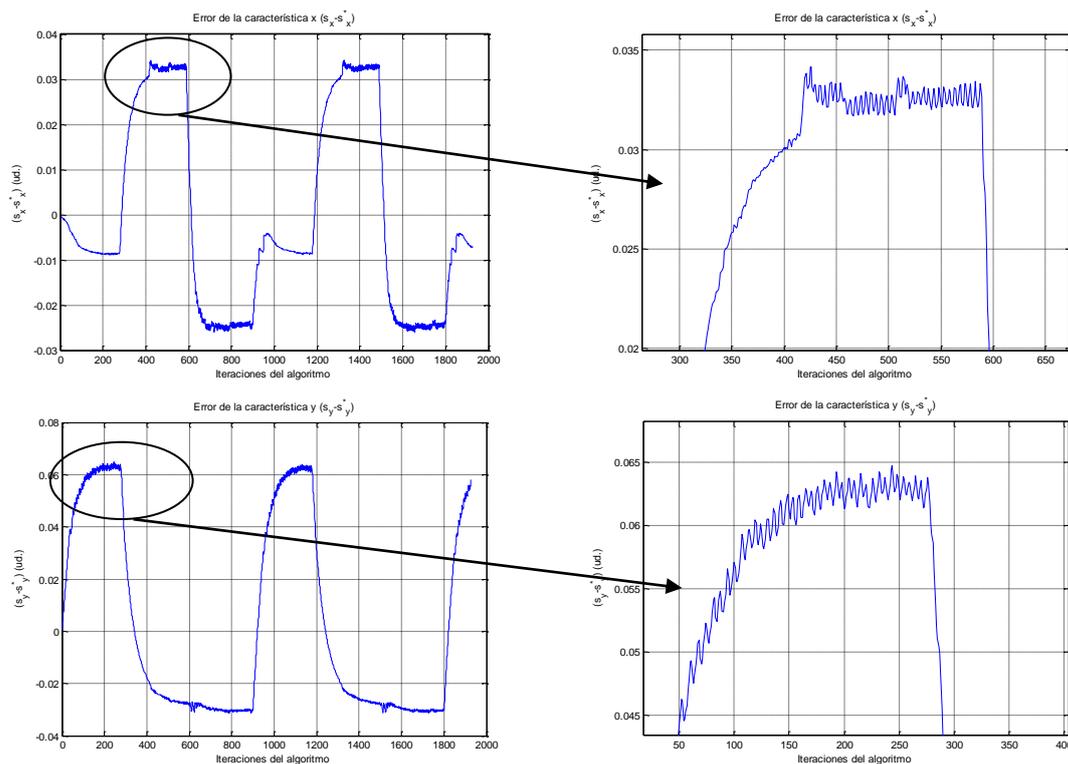


Figura 103. Errores de las características detectadas en el plano de imagen (trayectoria Triángulo con periodo 30 segundos).

En las gráficas de la Figura 103 se ha resaltado los momentos en los que el error se mantiene constante. Esto indica que el objeto está en una trayectoria sin cambios (trayectoria recta) y éste es el error de posición que siempre se va a tener mientras el objeto no cambie la velocidad o la dirección en su trayectoria.

Hay que tener en cuenta que el error que se está representando es el error de las características $(s - s^*)$ no directamente el error en píxeles. Por lo tanto si tiene sentido que los errores sean inferiores a un píxel aunque realmente esto no se podría observar en el plano de imagen.

Los resultados son que se tiene un error constante en el Eje X de 0.0342 ud. Y un error constante en el Eje Y de 0.0312 ud. Esto, que ahora no es representativo, se verá más adelante cuando se haga la comparativa con el comportamiento multifrecuencia como los valores se hacen mucho más pequeños.

Por otro lado, en las gráficas de la Figura 104 se representan las acciones de control de cada una de las articulaciones que componen el sistema robótico. Se puede ver en este caso, y como es natural, que el sistema no tiende a 0 ya que el objeto siempre se encuentra en movimiento por los que habrá articulaciones que necesiten una acción constante y mayor que otras.

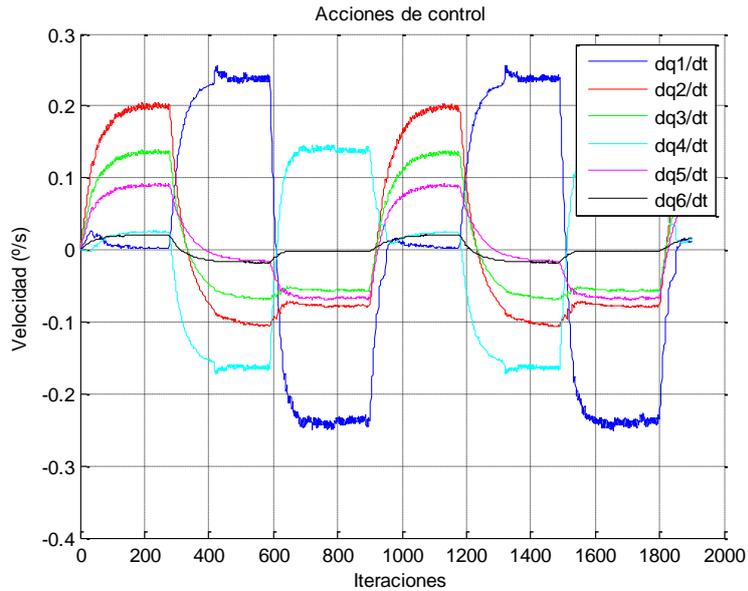


Figura 104. Acciones de control para cada una de las articulaciones (trayectoria Triángulo con periodo 30 segundos).

Ahora se va a ver qué ocurre con el error entre las características deseadas y las del objeto cuando se incrementa la velocidad del objeto. Esto es, el objeto va a tener menos tiempo para realizar la trayectoria Triángulo. En este caso el tiempo para realizarla será de 10 segundos, tres veces inferior al tiempo anterior.

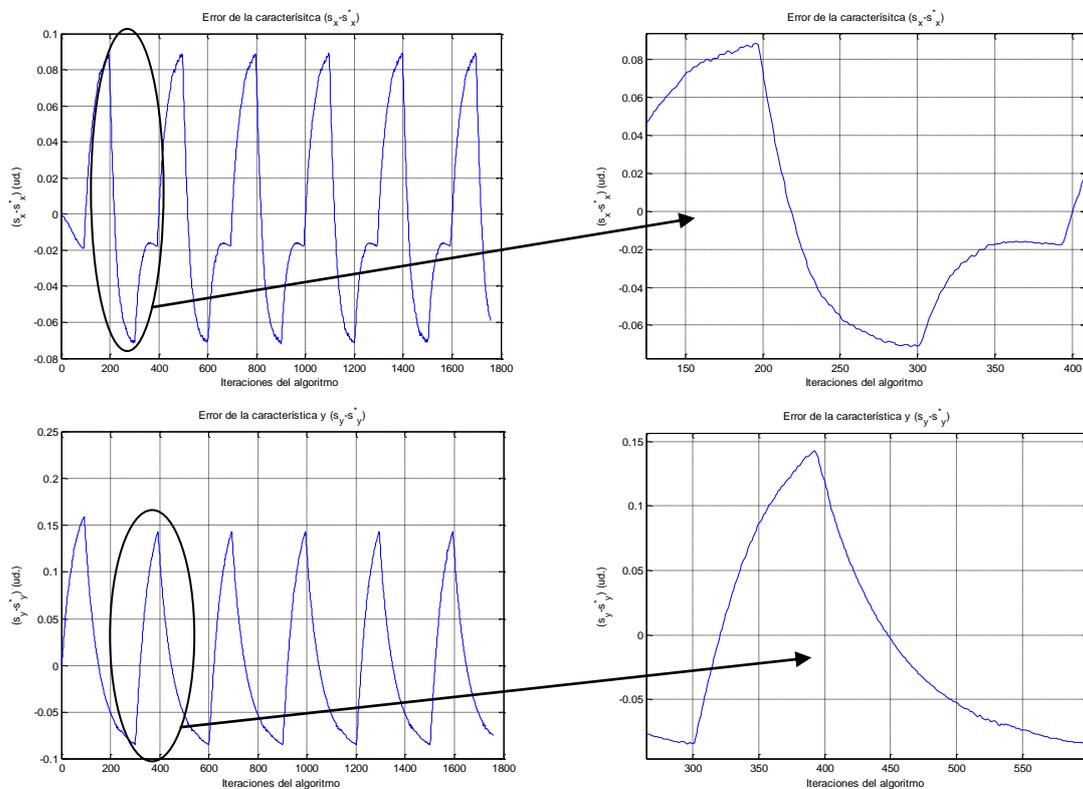


Figura 105. Errores de las características detectadas en el plano de imagen (Trayectoria triángulo con periodo 10 segundos).

Se puede observar como la forma de las gráficas son bastante distintas a las de la misma trayectoria pero con el objeto yendo más lentamente. Hay una explicación para ello. El sistema de control no consigue realizar la trayectoria correctamente, es decir, está yendo siempre dentro del triángulo haciendo una trayectoria casi circular. La siguiente Figura puede dar una visión más significativa entre ambas trayectorias.

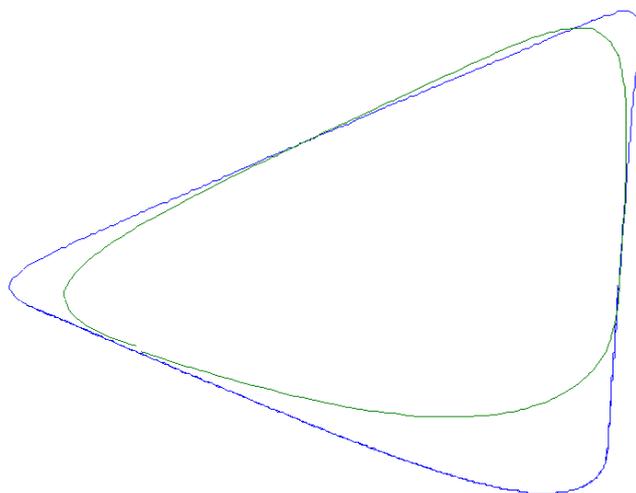


Figura 106. Trayectorias seguidas por el efector final del robot al seguir la trayectoria del objeto. En azul el objeto siguiendo una trayectoria triangular con 30 segundos de periodo. En verde el objeto siguiendo la misma trayectoria pero con el objeto moviéndose a 10 segundos.

Por este motivo, el error máximo ($\max(s - s^*)$) es ahora mayor que el que se tenía anteriormente: Eje Y, 0.14 ud. y Eje X 0.08 ud.

Por otro lado, en lo que respecta a las acciones de control, las gráficas de la Figura 107 muestra que las acciones de control también son más bruscas al intentar corregir la trayectoria, eso provoca también un mayor consumo de energía.

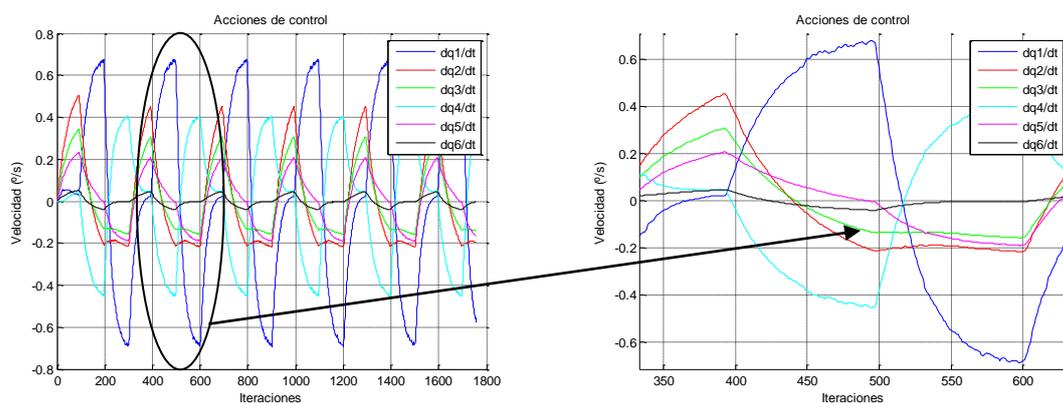


Figura 107. Acciones de control aplicadas para el seguimiento de la trayectoria Triangular con periodo de 10 segundos.

El último experimento fue con la misma trayectoria pero esta vez con un periodo de 5 segundos. El resultado fue que el algoritmo no fue capaz de seguir el objeto tal y como se demuestra en las siguientes gráficas.

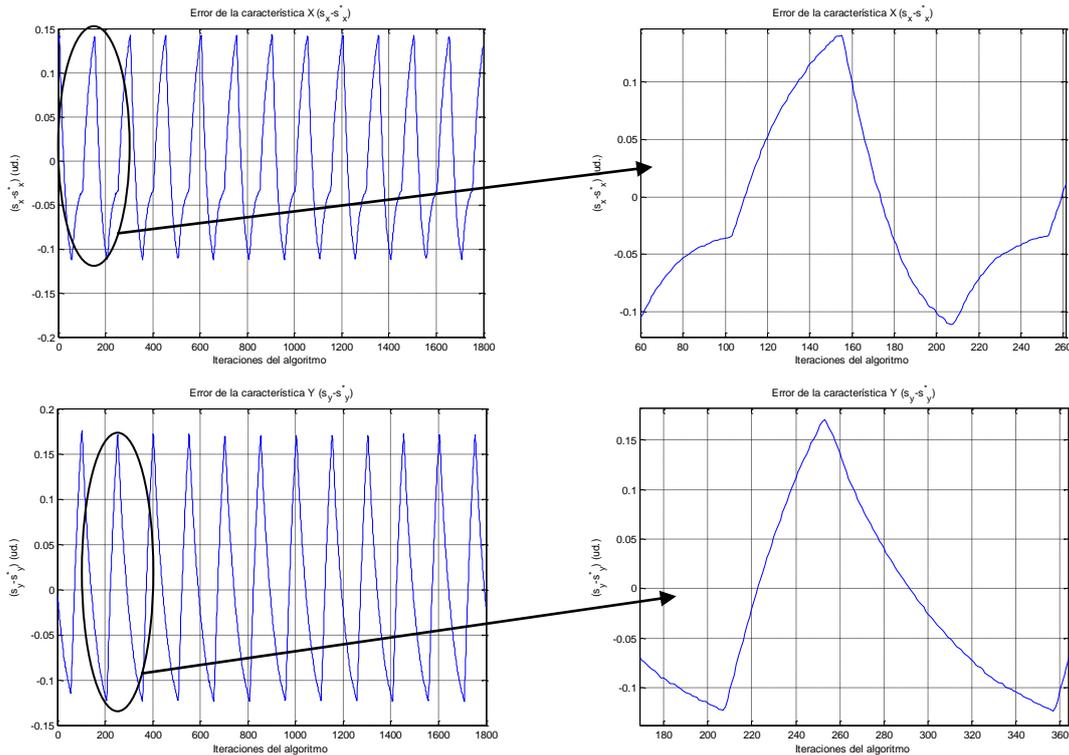


Figura 108. Errores de las características detectadas en el plano de imagen (trayectoria Triángulo con periodo 5 segundos).

Como ocurría en el caso anterior, en este tampoco se observa ninguna zona de transición en la que el efector final mantenga la distancia con el objeto. Esto es porque la trayectoria que se está siguiendo cada vez se parece más a la de un círculo. Esto también se puede observar porque el error ha aumentado respecto a los resultados anteriores. En este caso, el error máximo de la característica x es de 0.1430 unidades y en la de la característica y es de 0.1702 unidades.

Los resultados anteriores se pueden ver mejor en la Figura 109. Como se puede observar es una comparativa de la trayectoria que ha efectuado el efector final del robot en consecuencia de la trayectoria a seguir. En este caso se compara la trayectoria triangular en función de la velocidad a la que está yendo el objeto. Vemos que la trayectoria en rojo, que es la que se está estudiando en este apartado, está lejos de ser la trayectoria triangular seguida por el objeto, muy similar a la azul.

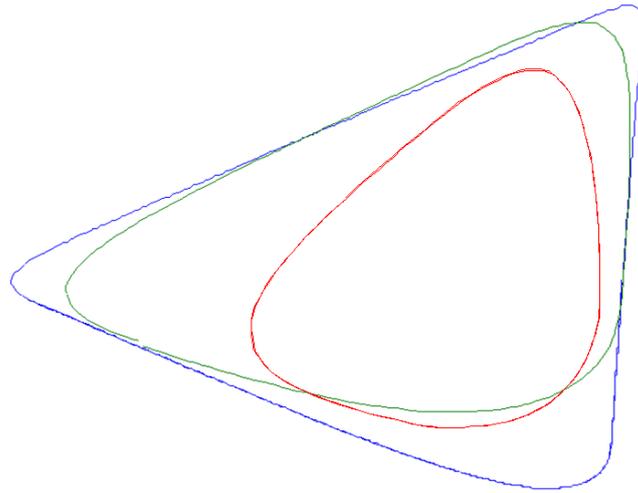


Figura 109. Trayectorias seguidas por el efector final en función de la velocidad del objeto: azul con periodo de 30 segundos, en verde con periodo de 10 segundos y en rojo con periodo de 5 segundos.

Como es de esperar, las acciones de control también son más agresivas por lo que el consumo de energía se incrementa, aunque no se puede seguir la trayectoria.

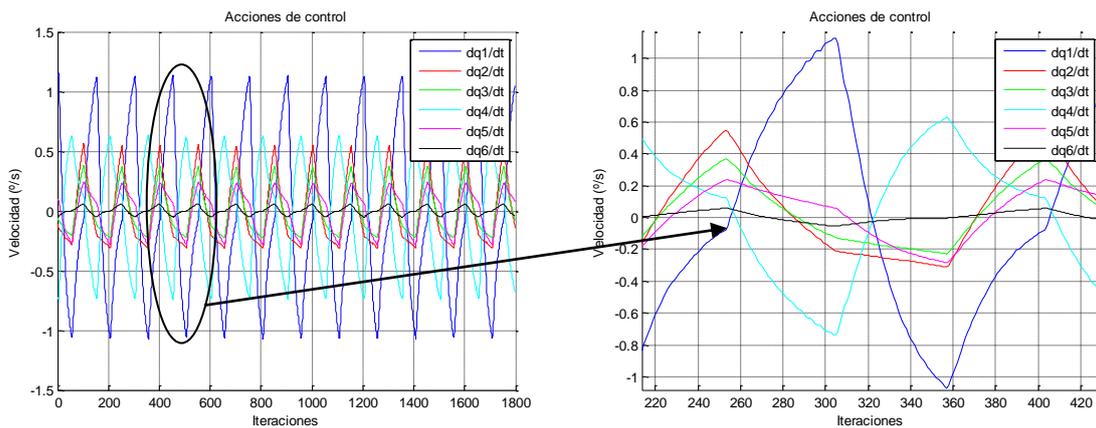


Figura 110. Acciones de control aplicadas para el seguimiento de la trayectoria Triangular con periodo de 5 segundos.

Este estudio se ha realizado con cada una de las trayectorias anteriormente mencionadas, con lo que los resultados son los que se muestran en el diagrama de barras de la Figura 111.

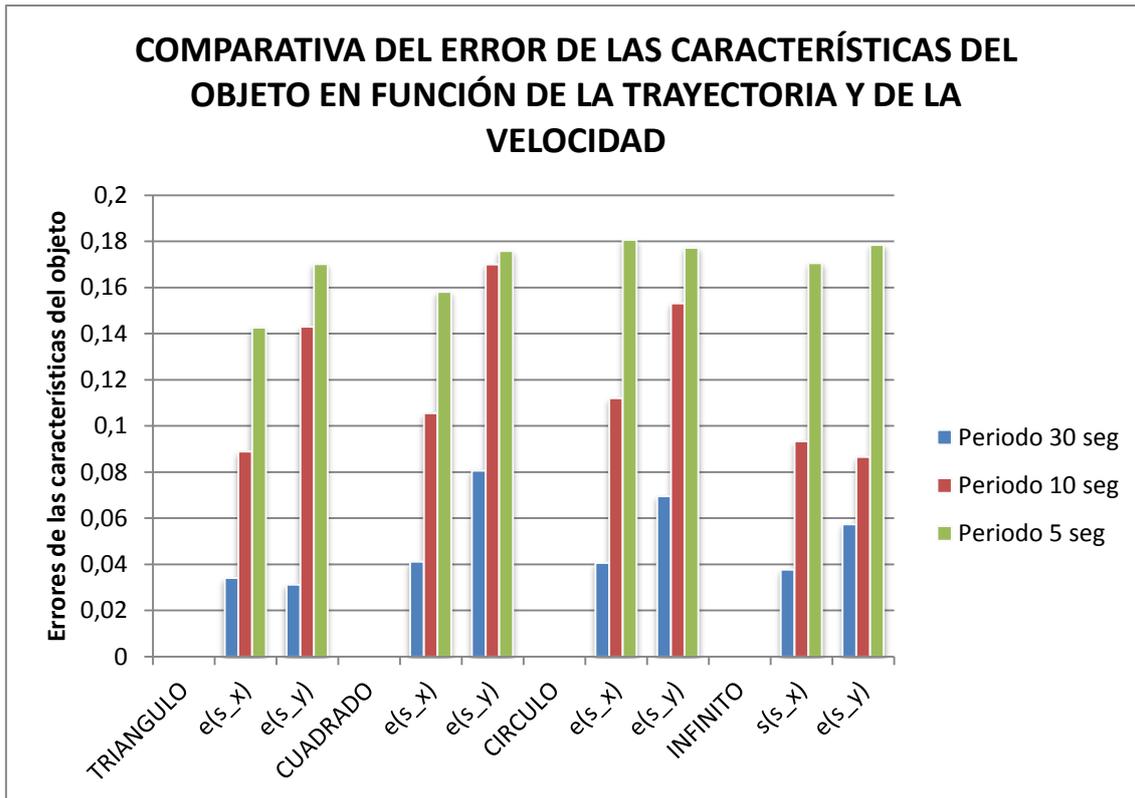


Figura 111. Comparativa del comportamiento del sistema frente a cambios de la trayectoria del objeto y de su velocidad.

Un resultado más interesante que el anterior es el que se observa a partir del error cuadrático de las características del objeto:

$$E = \left\| \sqrt{e^2 (s - s^*)} \right\| \quad (95)$$

Con esto se tiene una clara idea del error total cometido por el sistema a la hora de realizar el tracking del objeto para cada una de las trayectorias y sus variantes de velocidad. Se comprueba que la trayectoria que mayor error le produce al sistema es la trayectoria circular, ya que es la que mayor error produce. Esto es natural debido a las características del sistema. En los robots manipuladores, las trayectorias más complicadas de interpolar y, por lo tanto realizar, son los trazados circulares. Por lo tanto no es de extrañar que en el caso del visual Servoing aparezca también este fenómeno.

También hay que destacar que a velocidades pequeñas, el mayor error se comete con el cuadrado, aunque sea la trayectoria mejor seguida cuando se ve en tiempo real. Esto es así debido a los cambios de 90 grados que suponen un error grande de posición y es por ello que a medida que avanza la simulación produce mayor errores que el caso en que hay menos cambios bruscos (como es el caso del triángulo).

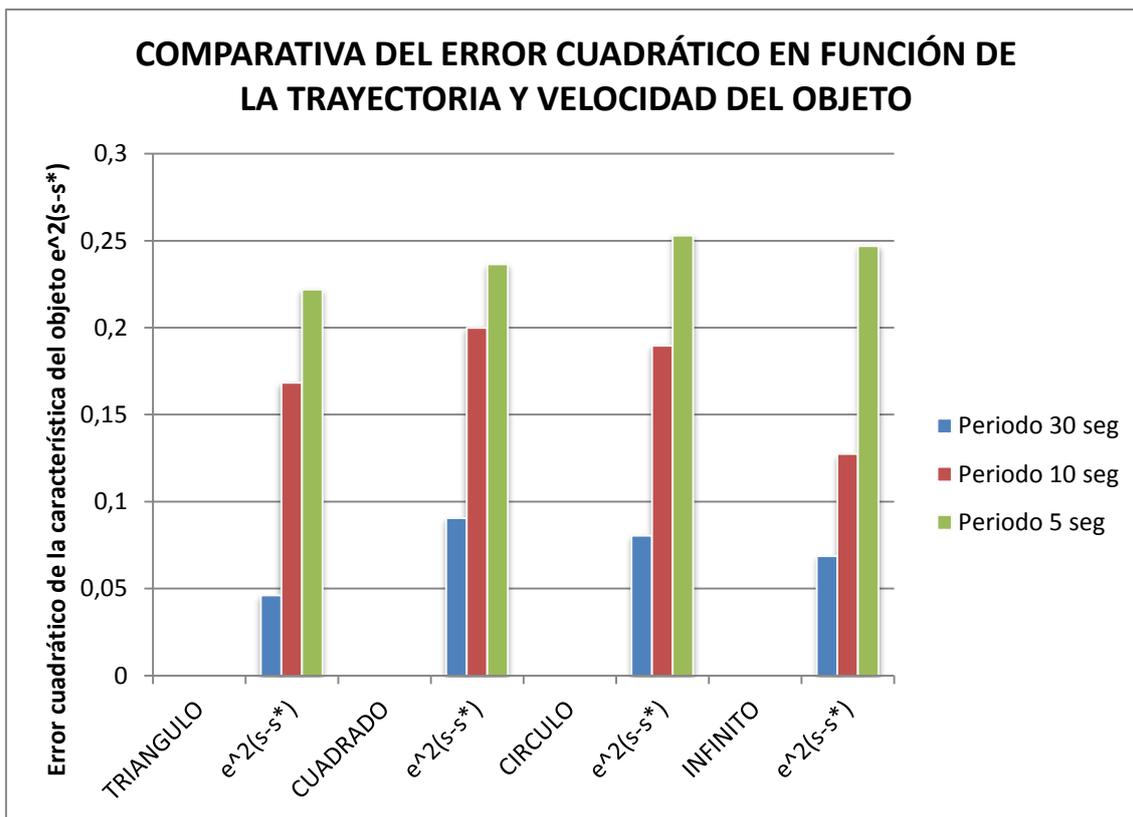


Figura 112. Comparativa del comportamiento del sistema frente a cambios de la trayectoria del objeto y de su velocidad (Error Cuadrático).

3.4.-Conclusiones

En este capítulo se ha visto todo el trabajo realizado para el estudio tanto teórico como real del enfoque clásico *Image-Based Visual Servoing (IBVS)*. En él se han vistos pros y contras en la elección de la matriz de interacción y se ha visto como define la trayectoria en el plano de imagen.

Además se ha expuesto el trabajo realizado para la implementación del sistema Visual Servoing en un robot real (*Kuka KR5 sixx R650*). Se puede ver todos los elementos del sistema y se ha resumido la forma de programarlos y las herramientas utilizadas para la implementación del IBVS en un robot de 6 grados de libertad.

Se ha realizado el análisis acerca del comportamiento tanto cuando se trabaja con el tracking de un objeto estático como en un objeto en movimiento. Además se ha realizado el análisis y posteriores conclusiones de dicho comportamiento dependiendo de la trayectoria y la velocidad del objeto.

CAPÍTULO 4

Visual Servoing Multifrecuencia

4.1.-Sistema muestreado multifrecuencia

En este capítulo se va a estudiar el comportamiento del sistema Visual Servoing visto en el caso anterior, aunque esta vez incorporando el punto de vista multifrecuencia. Para ello se hará uso de los interfaces DR_HOH vistos en el Capítulo 2 ya que, como se ha dicho en capítulos anteriores todo el modelado multifrecuencial está basado en modelos de plantas lineales. En este caso, el sistema que se tiene es no línea: sistema robotizado, adquisición de imágenes, comunicaciones, etc. Todos estos elementos incorporan no linealidades que hacen muy complejo el modelado del sistema.

Por otra parte se va a estudiar la diferencia de incorporar un enfoque asíncrono al sistema. La analogía de sistemas lineales y no lineales puede ayudar a comprender mejor este punto de vista: cuando se utilizan técnicas lineales de control para el diseño de controladores, se asumen muchos factores que bajo un punto de funcionamiento determinado, el sistema lineal puede que funcione de esa forma pero todo lo que pase en otro punto de funcionamiento es incierto. Es por ello que el comportamiento del sistema cuando se aplican metodologías de diseño no lineal mejora respecto del diseñado para un sistema lineal.

En la realidad, todos los sistemas son asíncronos. Esto quiere decir que el periodo base (T) y el metaperiodo (\bar{T}) de cada uno de los elementos varía en función de muchos condicionantes que rara vez se pueden determinar. Es por ello que se va a realizar una comparativa empleando ambos diseños, síncrono y asíncrono, demostrándose la afirmación anterior.

Por lo tanto, en primer lugar se va a ver cómo se debe actuar para realizar implementar el enfoque multifrecuencia en esta clase de sistemas, realizando una comparativa entre los distintos DR-HOH y el sistema trabajando a baja frecuencia. Luego se verán algunos resultados interesantes que se han realizado bajo simulación y por último se realizará el análisis con el sistema IBVS real.

4.2.-Control de dos grados de libertad con un PID y interfaces multifrecuencia

En este apartado se ha querido realizar el estudio detallado de la incorporación en el sistema robotizado el enfoque multifrecuencial a partir de las interfaces DR-HOH. Para facilitar la comprensión del desarrollo realizado se trabajará con un regulador muy conocido (PID) y se utilizarán sólo dos articulaciones del robot. Una vez entendido lo que se viene realizando, todo se extrapolará al control IBVS del sistema robotizado.

4.2.1.-Sistema muestreado multifrecuencia

Como se ha estado viniendo diciendo hasta ahora, todos los sistemas reales que incorporan sensorización y control discreto son sistemas muestreados. Además, debido a la naturaleza física de cada elemento que compone el sistema, los sistemas reales son sistemas muestreados multifrecuencia, ya que cada elemento aporta su propio periodo, el llamado metaperiodo.

El sistema robotizado, objeto de estudio de esta tesina presenta el diagrama de bloques proporcionado en la Figura 113. En ella se puede ver que el conjunto sensor de visión (adquisición + procesado de imagen) proporcionan un metaperiodo que desde ahora se llamará (\bar{T}_p) . Así mismo se tiene el tiempo que transcurre entre que se tiene una señal de error y se obtiene una señal de control. A este metaperiodo, que en algunos casos puede ser despreciable y en otros casos no, se llamará a partir de ahora (\bar{T}_c) . Por último se tiene el tiempo relacionado con las comunicaciones. Como se ha mencionado en el capítulo anterior, el control del robot se realiza a partir de la tarjeta RSI-XML Ethernet de Kuka. Una de las particularidades de esta tarjeta es que si hay un retardo de más de 12 ms en respuesta de alguna de las dos partes implicadas (controlador del robot y sistema externo), el controlador piensa que hay un problema y corta la ejecución del programa. Esto significa que, ya por tecnología se requiere un tratamiento multifrecuencia que, hasta ahora se realizaba ya que cada 12 ms se enviaba una acción, bien fuera nueva o bien fuera la anterior. A este metaperiodo se le denominará \bar{T}_s .

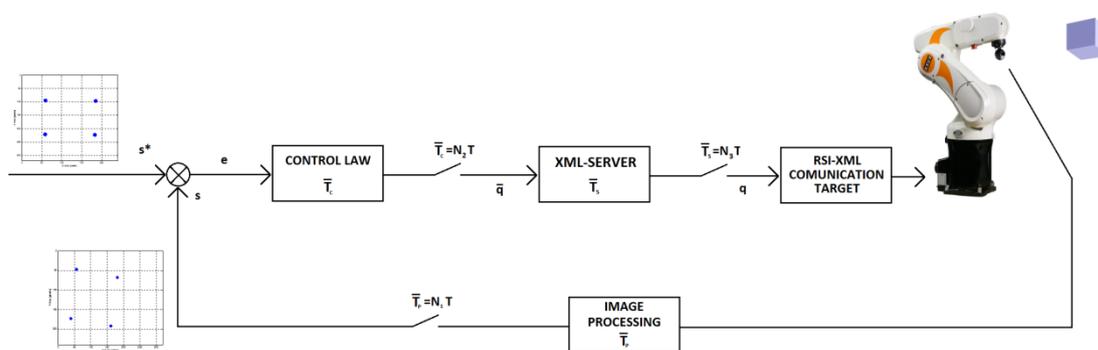


Figura 113. Diagrama de bloques multifrecuencia del sistema Visual Servoing.

Con el caso de control PID el anterior sistema se puede simplificar ya que (\bar{T}_c) es despreciable frente a los otros dos metaperiodos. Por lo tanto, el diagrama bloques de la Figura 114 muestra el sistema multifrecuencia que a partir de este punto será estudiado.

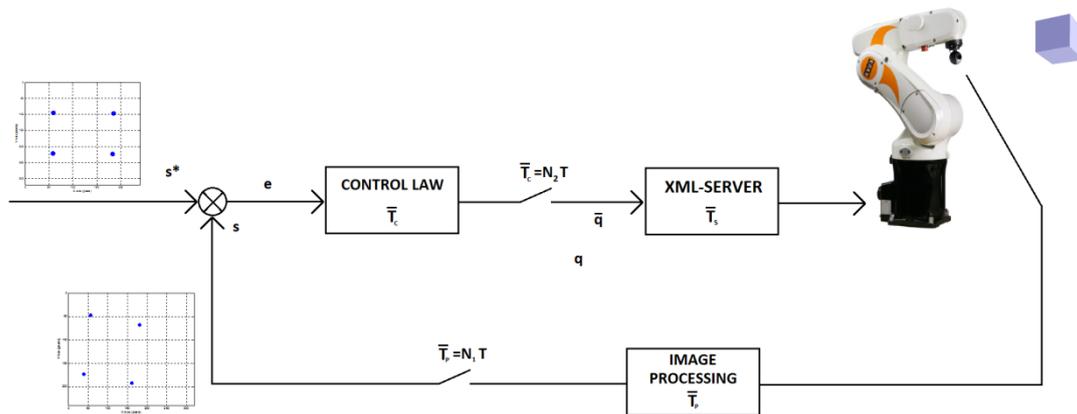


Figura 114. Diagrama de bloques multifrecuencia del sistema Visual Servoing simplificado.

Y para simplificar aún más el estudio lo que se va a hacer es a unificar el sistema controlador y servidor. De esta forma se tiene dos sistemas: el de adquisición y procesamiento trabajando a metaperiodo \bar{T}_p y el control de alto nivel mas comunicaciones trabajando a metaperiodo \bar{T}_{CS} .

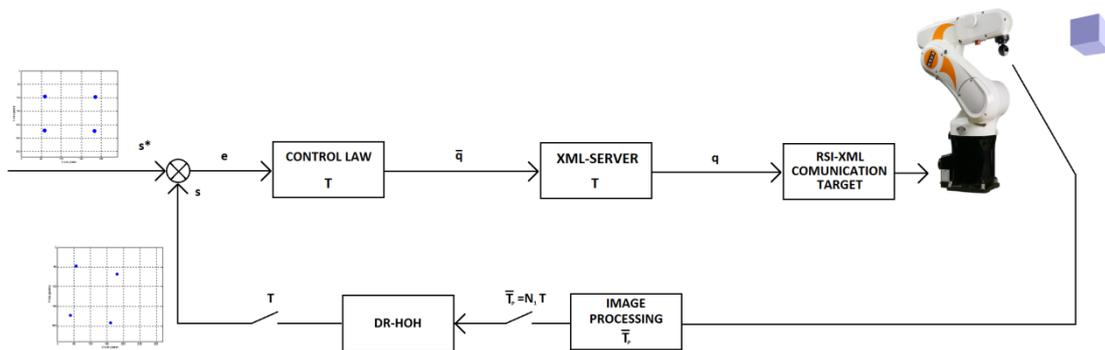


Figura 115. Máxima simplificación del sistema muestreado multifrecuencia.

Como se ha comprobado anteriormente cada 12 milisegundos el controlador de bajo nivel ha de recibir una respuesta, si no corta la comunicación. Con esto se quiere decir que el periodo base ha de ser $T = \bar{T}_{CS} < 12ms$. Debido a la optimización realizada en el procesado del algoritmo y a su sencillez³ el sistema de visión tiene una media de 33 milisegundos de periodo de muestreo, lo que implica que, al estar trabajando en 30 fps se está trabajando en tiempo real (ver Figura).

Es por esta razón que el periodo base, que es el mínimo común múltiplo de los metaperiodos del sistema, ha de ser inferior al tiempo de ciclo de las comunicaciones.

$$T = mcm(\bar{T}_1 \dots \bar{T}_n) \quad (96)$$

³ Ver el Anexo 2: pseudocódigo del algoritmo de Visión.

Siendo n el número total de elementos que conforman el sistema robotizado.

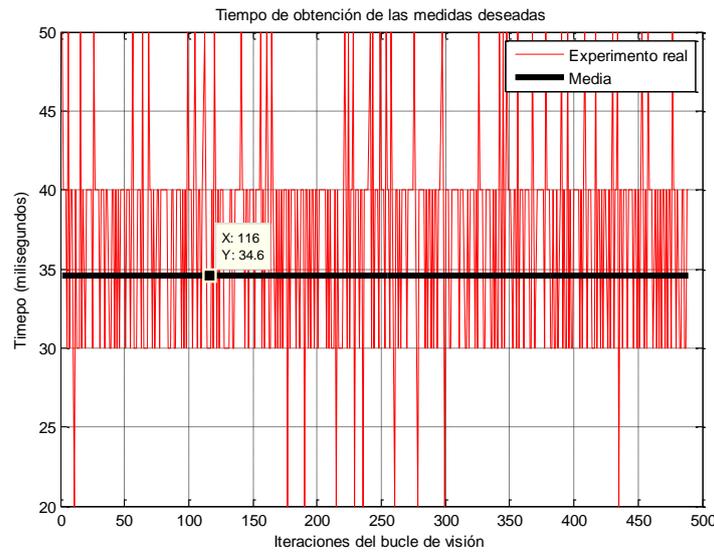


Figura 116. Metaperiodo del sistema de Visión (Adquisición más procesado de imagen).

Por otra parte y en lo referente al diseño del bucle de control, la interfaz puede colocarse o bien entre el procesado de la imagen y el bucle de control (ver Figura 117 izquierda), lo que significa diseñar un controlador rápido, o bien entre el controlador y el servidor (ver Figura 117 derecha) lo que implica que el controlador tiene que ser diseñado a baja frecuencia⁴.

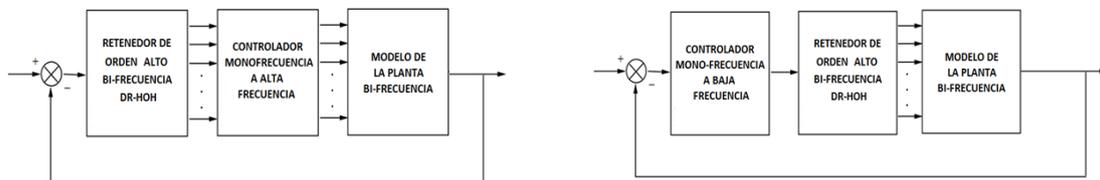


Figura 117. Diagramas de bloques de los enfoques de estimación de la señal de error (Derecha) y estimación de la señal de control (Izquierda).

Para la realización de los distintos experimentos se ha utilizado, como en el caso del capítulo 3, las trayectorias triángulo, cuadrado, círculo e infinito vistas representadas en la Figura 103. A l igual que se hiciera en el capítulo citado, se van a hacer las pruebas variando la velocidad del objeto.

El controlador discreto a alta frecuencia que se ha implementado es el que se muestra en la Ecuación 97.

$$G_R|_{HF}(z) = 0.005 \cdot \frac{z - 0.8}{z - 1} \quad (97)$$

⁴ Para más información acerca de los conceptos y procedimiento a seguir ver el ejemplo detallado del Anexo 1.

Cabe decir que se ha utilizado en ambas articulaciones el mismo regulador. Aunque es cierto que dinámicamente el sistema cambia mucho dependiendo qué articulación se quiere controlar, en este caso no se está interesado en realizar un control óptimo del sistema, sino en establecer correctamente las bases en las que se está trabajando.

Por otro lado hay que decidir a qué frecuencia se va a trabajar. En este caso y debido a que el sistema de control más comunicaciones es mucho más rápido que el de adquisición, el periodo base es escoge de 5 milisegundos y se establece que el metaperiodo del elemento de visión es de 50 milisegundos, se coge el peor de los casos para no tener pérdidas de periodos. Esto proporciona un orden de multiplicidad entre ambos sistemas de 7:

$$N = \frac{\bar{T}_p}{T} = 10 \quad (98)$$

Con esto se establece un enfoque claramente síncrono. Esto se puede observar en la Figura 118. Se da por hecho que la medida procesada siempre va a estar cada 35 milisegundos por lo que cada actualización de la variable se realizará en cada ciclo de este tiempo. Por otra parte, cada 5 milisegundos se estima la señal y el controlador calcula la nueva señal de control en función de esta estimación.

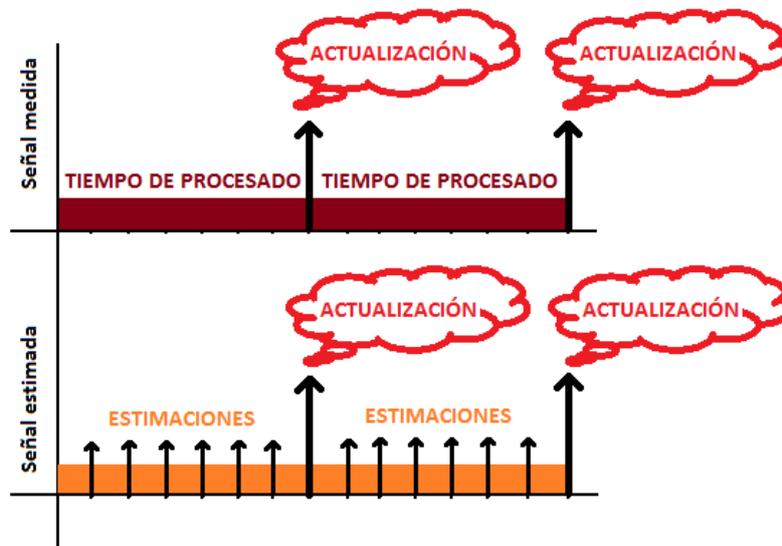


Figura 118. Diagrama de flujo de las interfaces DR-HOH.

4.2.1.1.- Pruebas realizadas con estimación de la señal de error de forma síncrona.

Como se puede ver en el diagrama de bloques de la Figura 119, el sistema con el enfoque de la estimación de la señal de error requiere del rediseño del controlador ya que debe trabajar a alta frecuencia.

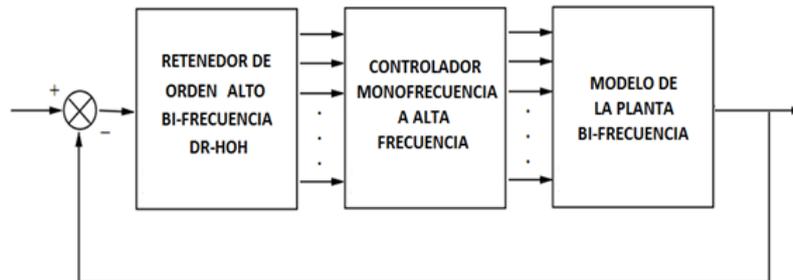


Figura 119. Estimación de la señal de error.

Lo más cómodo para obtener el controlador a baja frecuencia, ya que el regulador inicial a alta frecuencia es un regulador discreto, es aplicar la técnica de lifting al controlador a baja frecuencia, esto es:

$$\left. \begin{aligned}
 G_{i,j}(z^N) &= C \cdot A^i (z^N \cdot I - A^N)^{-1} \cdot A^{N-j-1} \cdot B + D_{i,j} \\
 D_{i,j} &= \begin{cases} C \cdot A^{i-j-1} \cdot B \rightarrow i > j \\ D \rightarrow i = j \\ 0 \rightarrow \text{other wise} \end{cases}
 \end{aligned} \right\} \quad (99)$$

Luego se utiliza la expresión dada por la Ecuación 100 para pasar de un sistema multivariable a su equivalente monovariable, obteniendo el controlador a baja frecuencia y así poder hacer la comparativa perfectamente.

$$G(z) = G_{0,0}(z^N) + \sum_{j=1}^{N-1} G_{0,j}(z^N) \cdot z^j \quad (100)$$

$$G(z) = G_{0,0}(z^N) + \sum_{i=1}^{N-1} G_{i,0}(z^N) \cdot z^{-i} \quad (101)$$

En este caso particular y para N=7 se tiene el siguiente regulador a baja frecuencia:

$$G_R|_{LF}(\bar{z}) = 0.005 \cdot \frac{\bar{z} + 1}{\bar{z} - 1} \quad (102)$$

En este caso, a diferencia del que se ha podido observar en el Capítulo 3, se está realizando un control por posición. Además el error de las características vendrá dado en unidades de píxeles. Se vuelve a recordar que este es un ejemplo del procedimiento que se debería hacer en cualquier sistema muestreado para la incorporación de interfaces Dual-Rate. No se realiza un control muy preciso del error de posición porque no es el objeto de este apartado, aunque se pone de manifiesto en los resultados la mejora del comportamiento del sistema cuando se incorpora el enfoque multifrecuencia dado por los DR-HOH.

Por lo tanto para cada una de las trayectorias se ha hecho una comparativa entre el sistema trabajando como un sistema monofrecuencia y el sistema trabajando como un sistema multifrecuencia con distintas interfaces.

- TRAYECTORIA TRIÁNGULO

En la Figura 120 se muestra un diagrama de barras que compara el error cometido en ambos ejes del plano de imagen en cada uno de los enfoques. Este resultado proporciona una idea clara de lo cerca o lejos que la posición deseada de la cámara está de la actual del objeto.

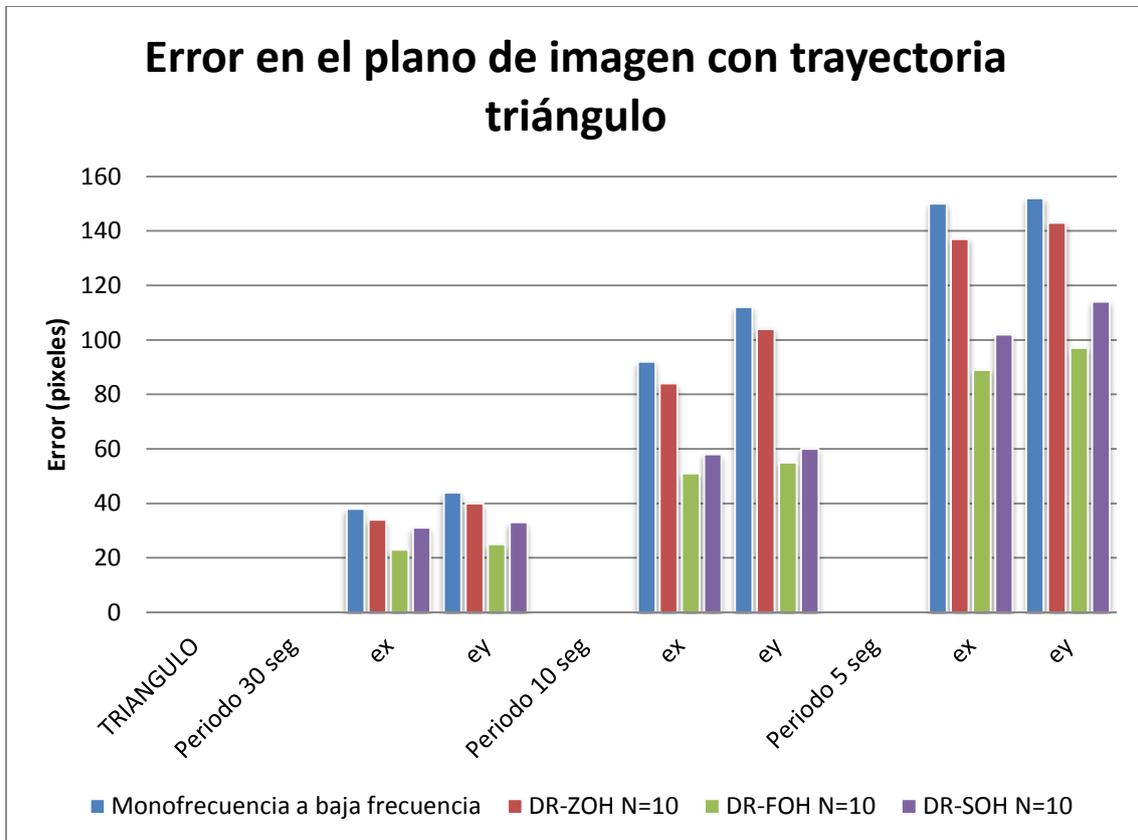


Figura 120. Comparativa para trayectoria triángulo.

Como se puede observar en el gráfico anterior, el sistema monofrecuencia y el multifrecuencia con un DR-ZOH tienen un comportamiento muy parecido. En el caso ideal este comportamiento debería ser el mismo.

Por otra parte, se ve que la introducción de una interfaz DR-FOH incorpora una mejora en el sistema de casi del 50% menos de error. Esto significa que el seguimiento del objeto es mejor ya que hay menor error entre la posición deseada y del objeto, es decir a lo largo del tiempo se está más cerca de alcanzarlo.

Otro punto de vista que hay que tener en cuenta es la trayectoria seguida por el objeto. Obviamente, cuando más cerca se está del objeto la trayectoria seguida por la cámara se parecerá más a la que está realizando el objeto en cuestión.

Además este gráfico también demuestra que la incorporación de DR-HOH de mayor orden ya no proporciona mejoras en este sistema concreto. Es decir, que en este caso y trabajando con las mismas condiciones el DR-HOH que mejores resultados proporciona es el de primer orden.

Por último, que se comenta menor error es directamente proporcional a decir que se puede realizar un tracking de objetos que tienen una velocidad mayor. Como se puede apreciar, el error cometido por el sistema monofrecuencia establece que el tracking no se está realizando.

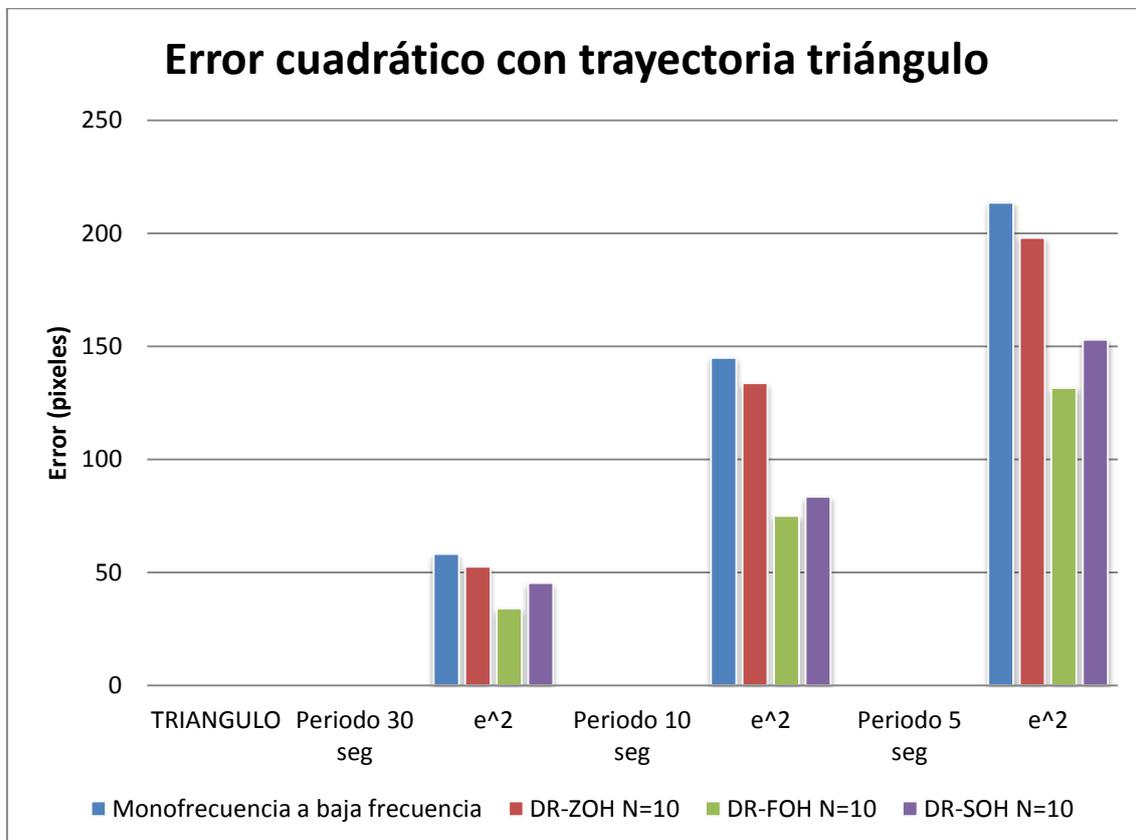


Figura 121. Error Cuadrático para trayectoria triángulo.

- TRAYECTORIA CUADRADO

Del mismo modo la Figura 122 muestra la comparativa realizada para la trayectoria cuadrado.

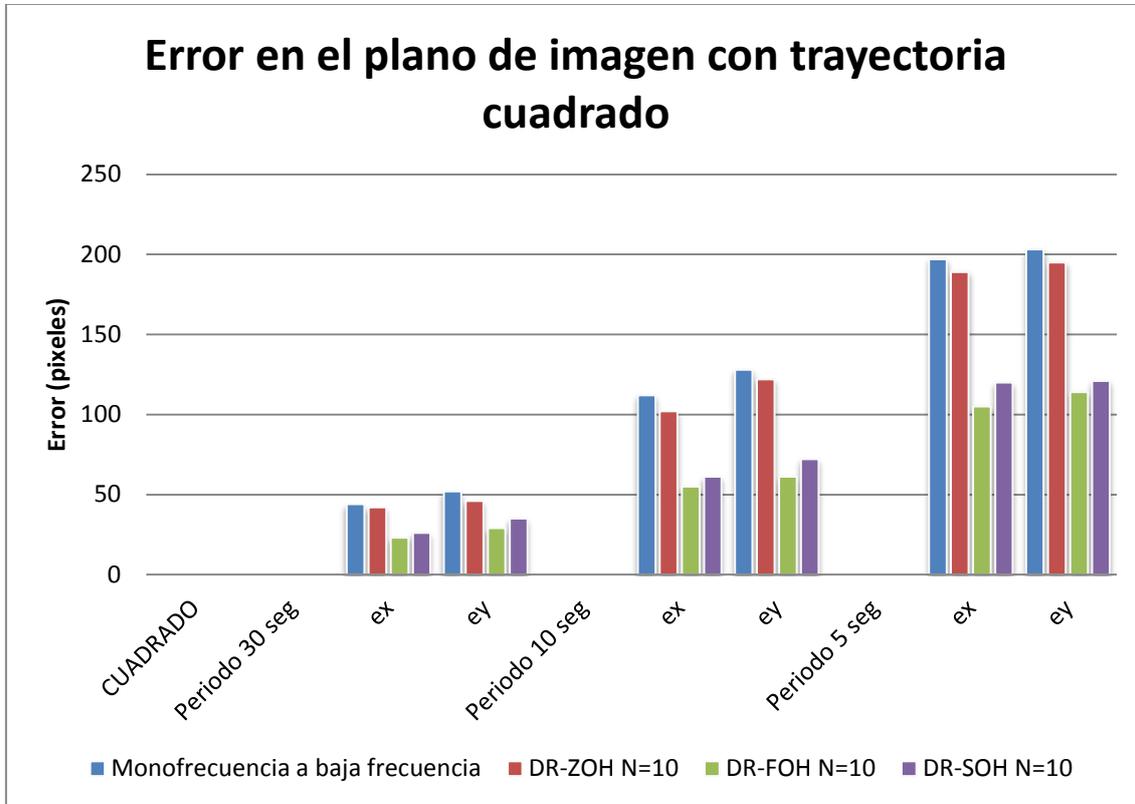


Figura 122. Comparativa para trayectoria triángulo.

En ella se aprecia como el enfoque multifrecuencia con la interfaz DR-FOH incorpora una notable mejoría en el comportamiento del mismo. La incorporación de interfaces de orden más elevado, aunque incorporan mejorías en el comportamiento del sistema respecto al monofrecuencia, no lo hacen respecto al sistema con el DR-FOH.

Hay que destacar que las predicciones de los DR-HOH son muy buenas cuando el objeto se mueve de forma predecible. Esto quiere decir que en tramos donde el objeto se está moviendo en línea recta, la aproximación será muy buena y por lo tanto el error será menor. En los cambios bruscos como son los cambios de sentido, la predicción empeorará y el error será mayor.

La Figura 123 muestra el error cuadrático de las características medidas. Este grafo aporta una visión más general del comportamiento del sistema frente a los distintos enfoques.

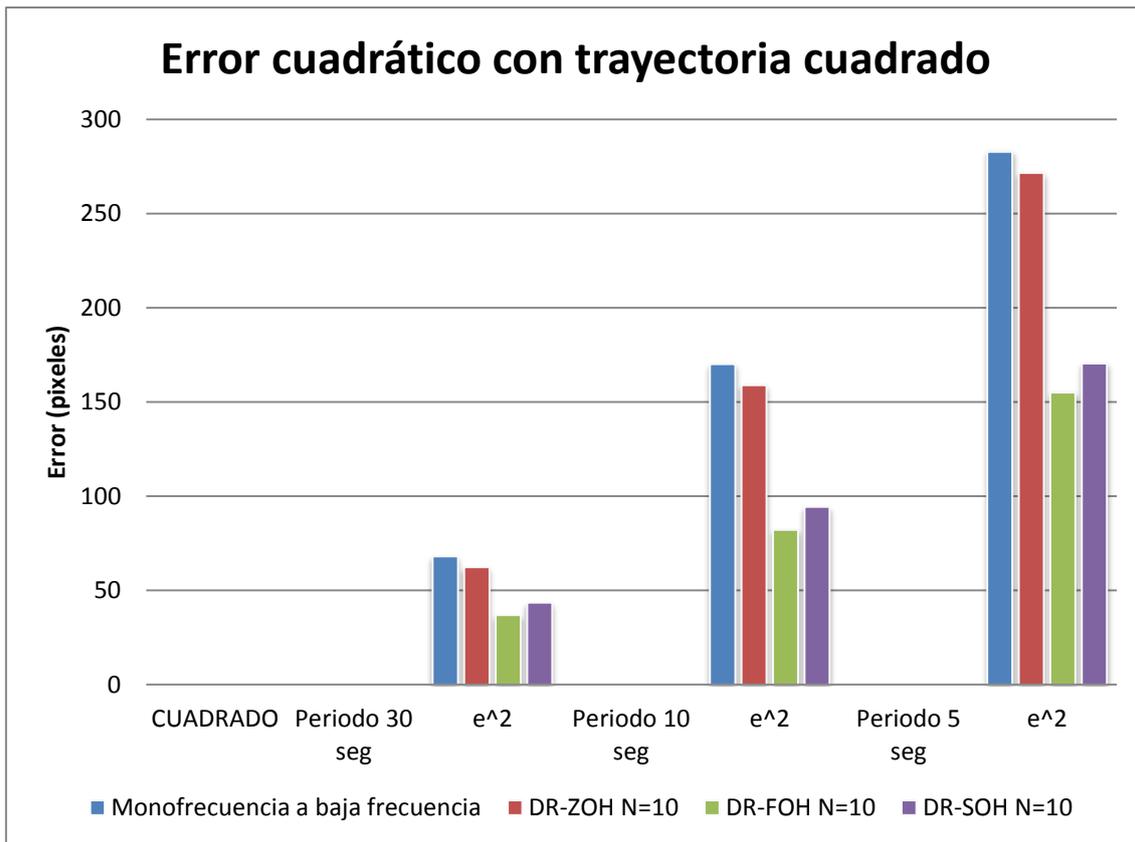


Figura 123. Error Cuadrático para trayectoria triángulo.

- TRAYECTORIA CIRCULO

Para la trayectoria circulo, se tienen los resultados que se muestran en la Figura 124. Se comprueba que, aunque esta trayectoria es mucho más difícil de estimar, el enfoque multifrecuencia con el Dual-Rate de primer orden proporciona mejores resultados que el sistema monofrecuencia o que el multifrecuencia con retenedores de orden cero.

En la Figura 125 se muestra el diagrama de barras que utiliza el error cuadrático para realizar la comparativa de los distintos métodos.

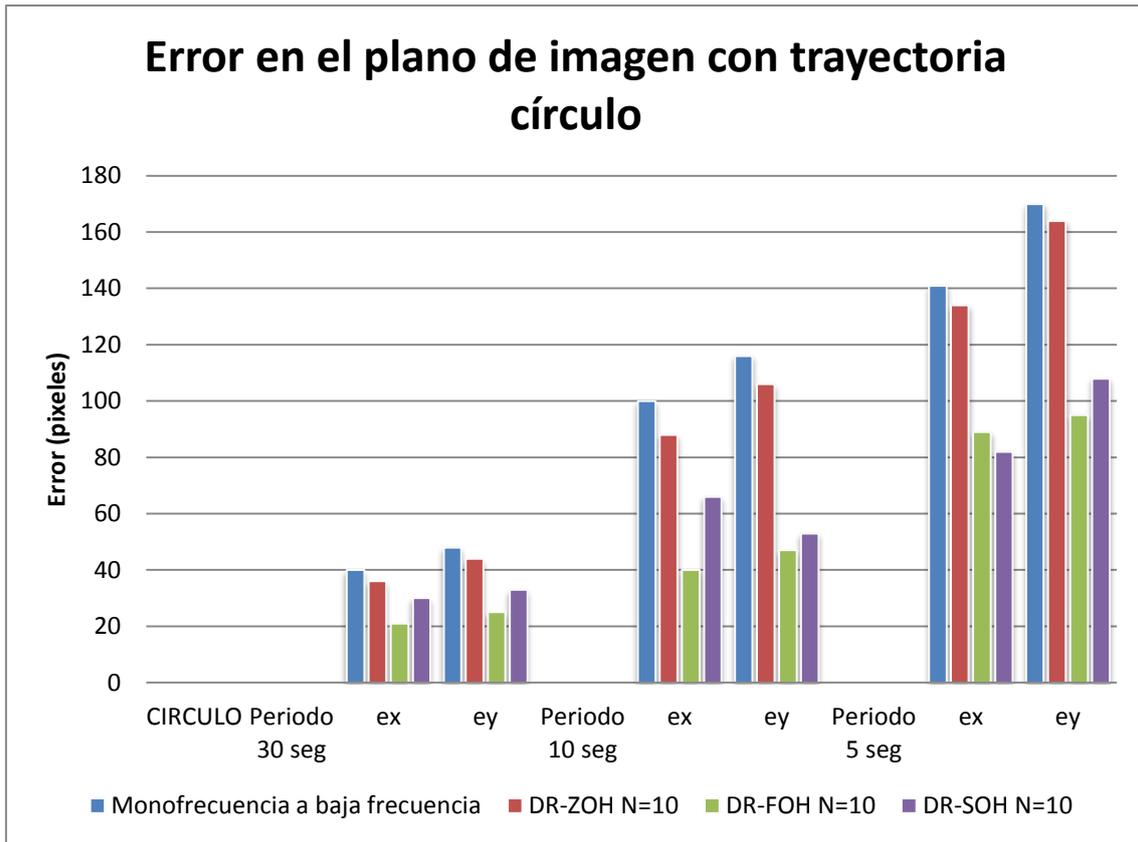


Figura 124. Comparativa para trayectoria círculo.

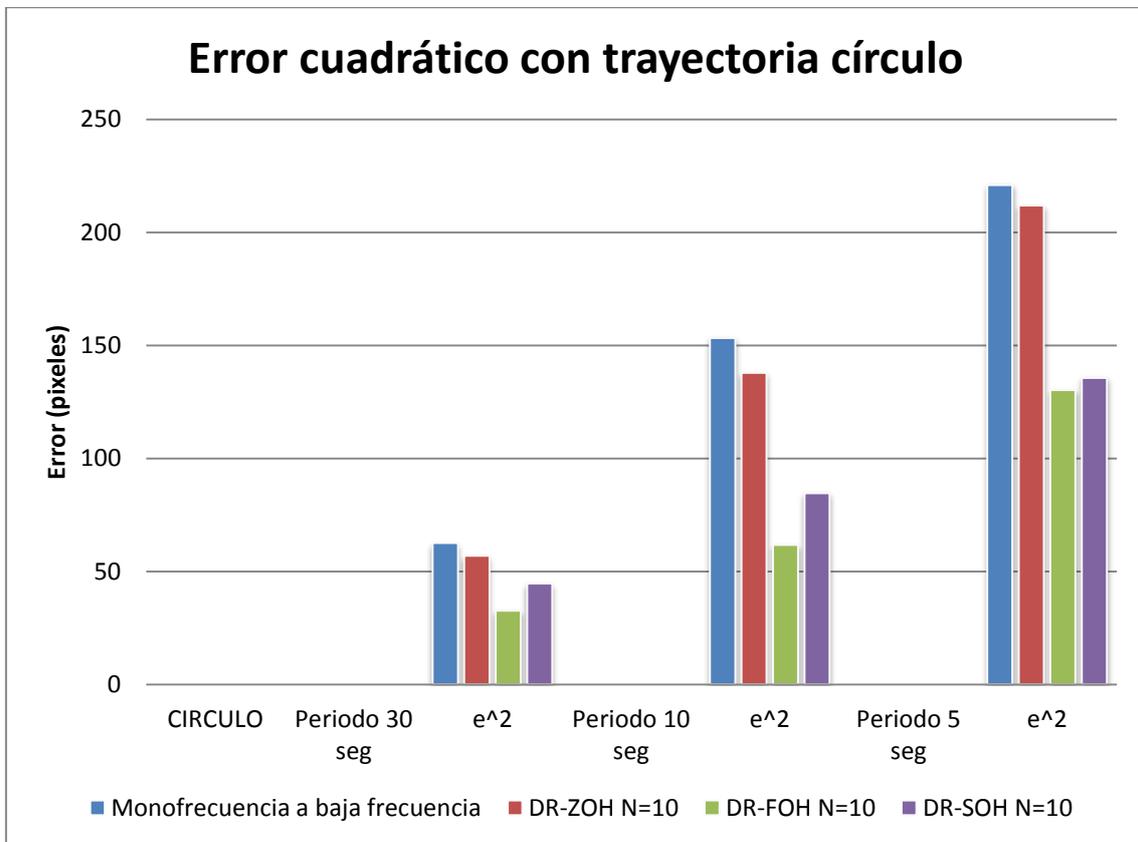


Figura 125. Error Cuadrático para trayectoria círculo.

- TRAYECTORIA INFINITO

Por último los resultados obtenidos para la trayectoria infinito se pueden observar en la Figura 126. También se demuestra que un enfoque multifrecuencia con las interfaces Dual Rate de orden alto proporcionan una mejor respuesta del sistema.

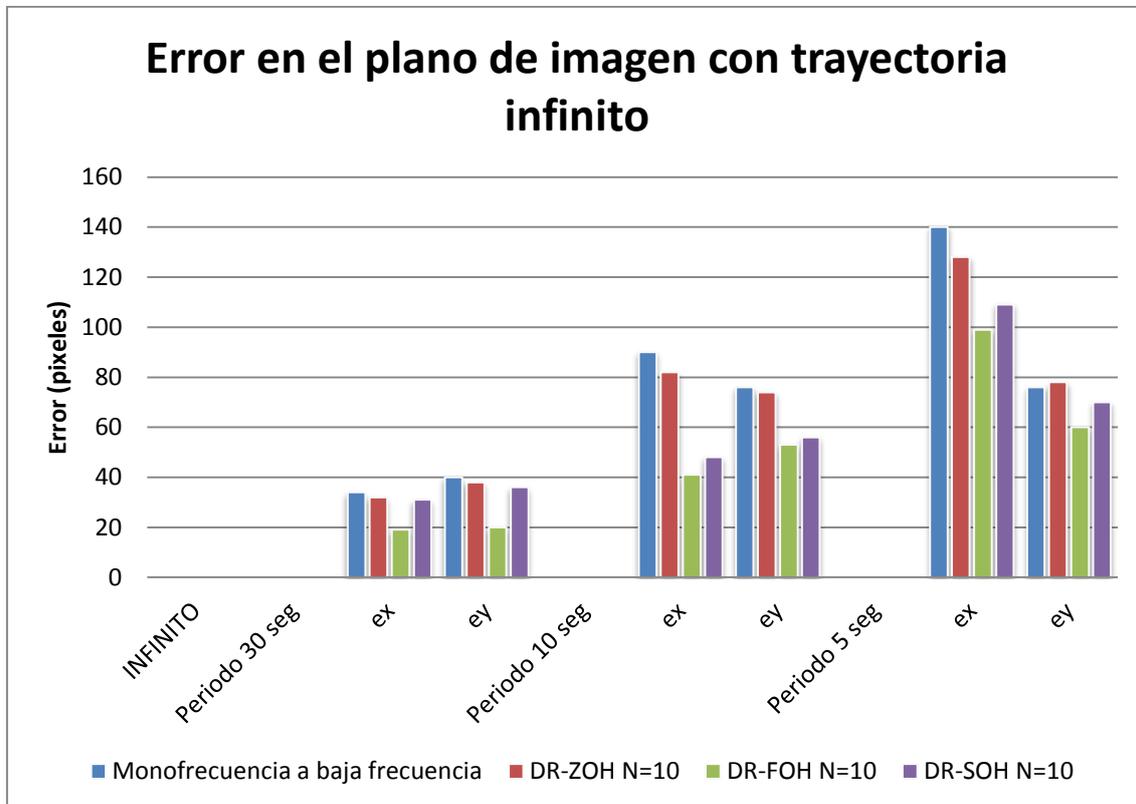


Figura 126. Comparativa para trayectoria círculo.

La Figura 127 muestra los resultados del error cuadrático con este experimento.

Por lo tanto, se ha visto que la incorporación del enfoque multifrecuencia utilizando interfaces bi-frecuencia ha proporcionado un mejor comportamiento del sistema. Todo ello con la premisa de que el sistema se está comportando idealmente, es decir que el sistema es síncrono.

Ahora hay que ver qué sucede si en vez de poner la interfaz entre la señal de error y el controlador, se pone entre la señal de control y la planta del sistema. Esto es se va a ver el comportamiento del sistema cuando se está trabajando con el enfoque de estimación de la señal de control.

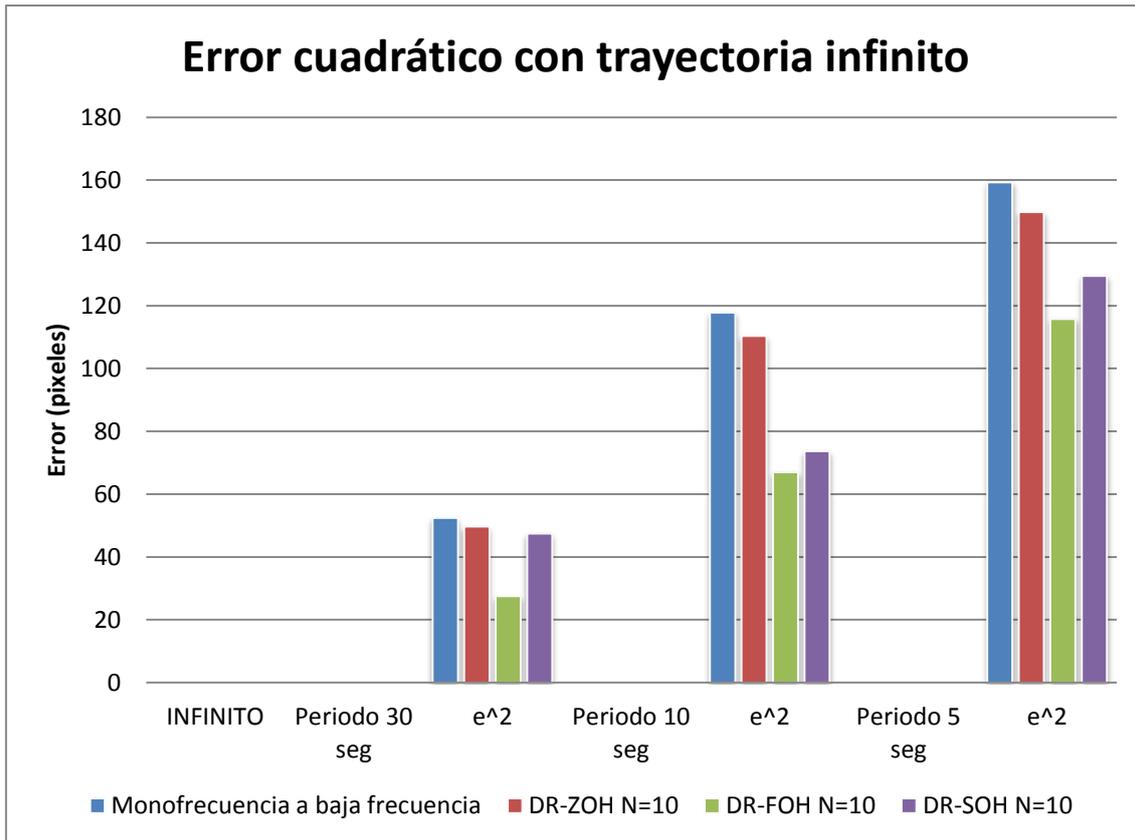


Figura 127. Comparativa para trayectoria círculo.

4.2.1.2.- Pruebas realizadas con estimación de la señal de control de forma síncrona.

El enfoque de estimación de la señal de control trata de a partir del regulador lento estimar las señales inter-muestreo para que el sistema trabaje a alta frecuencia. Esto se puede ver en el diagrama de bloques que presenta la Figura 128.

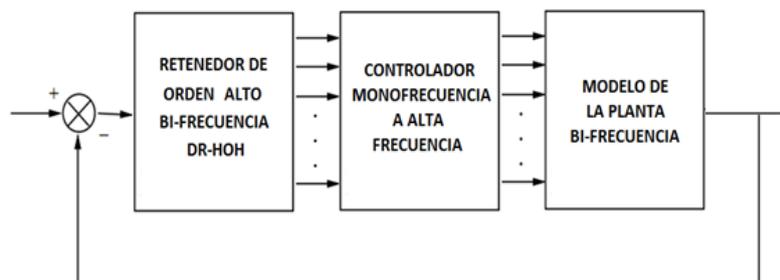


Figura 128. Diagrama de bloques del enfoque de estimación de la señal de control.

Por lo tanto, en este punto se va a utilizar el regulador lento y la interfaz se va a poner entre el controlador y el servidor. Con esto, en primer lugar se ha realizado el estudio y la comparativa entre el comportamiento del sistema trabajando a baja frecuencia y el comportamiento trabajando con el enfoque multifrecuencia.

- TRIÁNGULO

El comportamiento del sistema comparado con el de monofrecuencia utilizando varios DR-HOH se puede ver en el diagrama de barras que muestra la Figura 129. En él se comprueba que, al igual que pasaba cuando se utilizaba el enfoque de estimación de la señal de error, existe una mejora en el comportamiento, aunque más pronunciada en el caso de utilizar un DR-FOH.

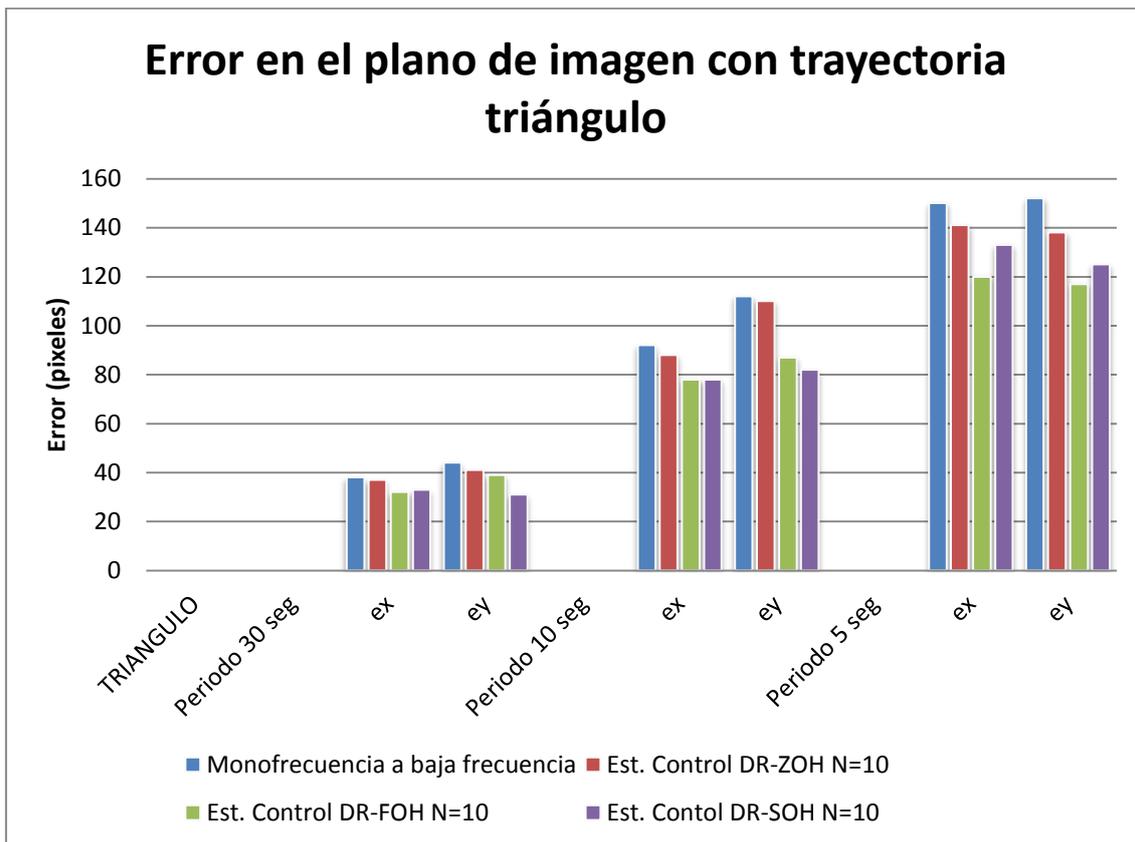


Figura 129. Comparativa del error utilizando el enfoque de estimación de la señal de control para la trayectoria triángulo.

Esta comparativa puede quedar más clara viendo el error cuadrático producido por cada uno de ellos. En este caso, se ve claramente que hay una mejor de alrededor un 10~20% respecto al tratamiento monofrecuencia o con la utilización de retenedores de orden cero.

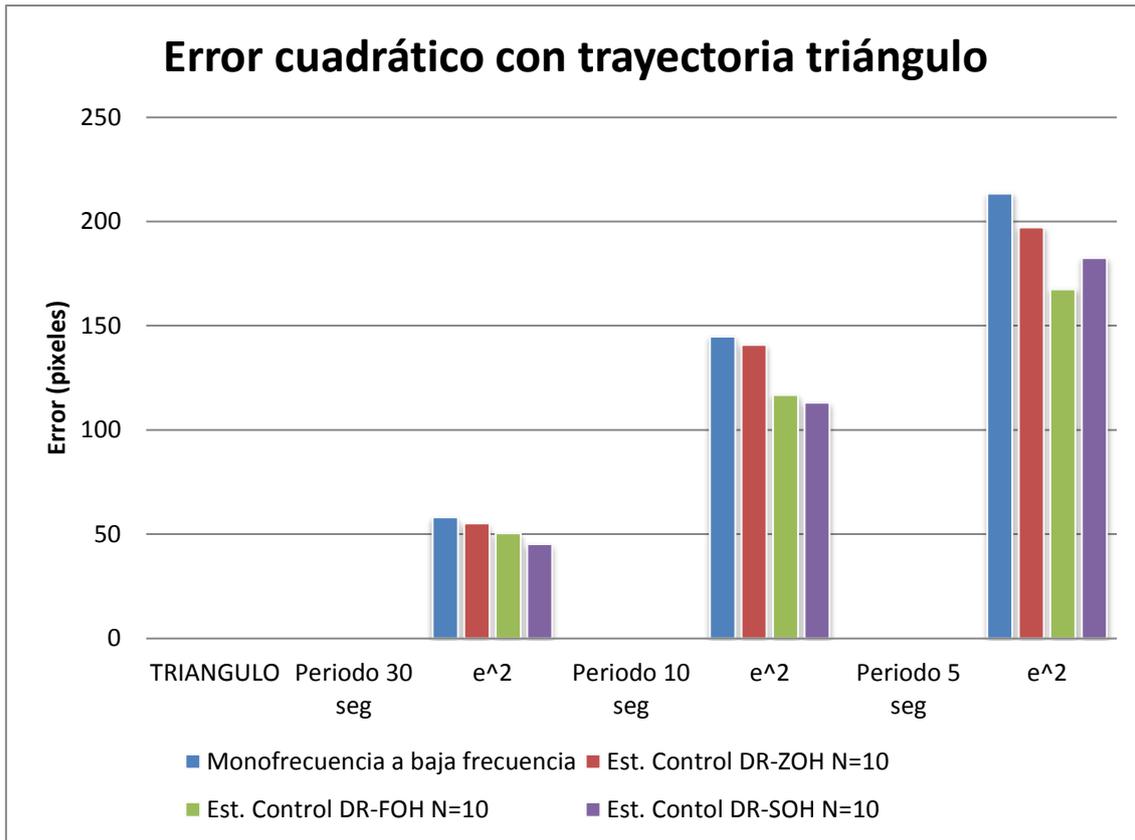


Figura 130. Comparativa del error cuadrático utilizando el enfoque de estimación de la señal de control para la trayectoria triángulo.

- CUADRADO

Del mismo modo, para la trayectoria cuadrado los resultados se pueden ver en las Figuras 131 y 132. En ellas se vuelve a demostrar que con este enfoque también hay una notable mejora del comportamiento del sistema.

Aún así y como se ha dicho en el apartado anterior, el uso de estos retenedores provoca que la estimación sea buena en los instantes en los que el objeto se mueve sin cambios bruscos en su dirección y empeoran con dichos cambios bruscos. Por lo tanto, cuantos menos cambios de dirección haga el objeto mejor será la estimación realizada y por lo tanto menor el error producido.

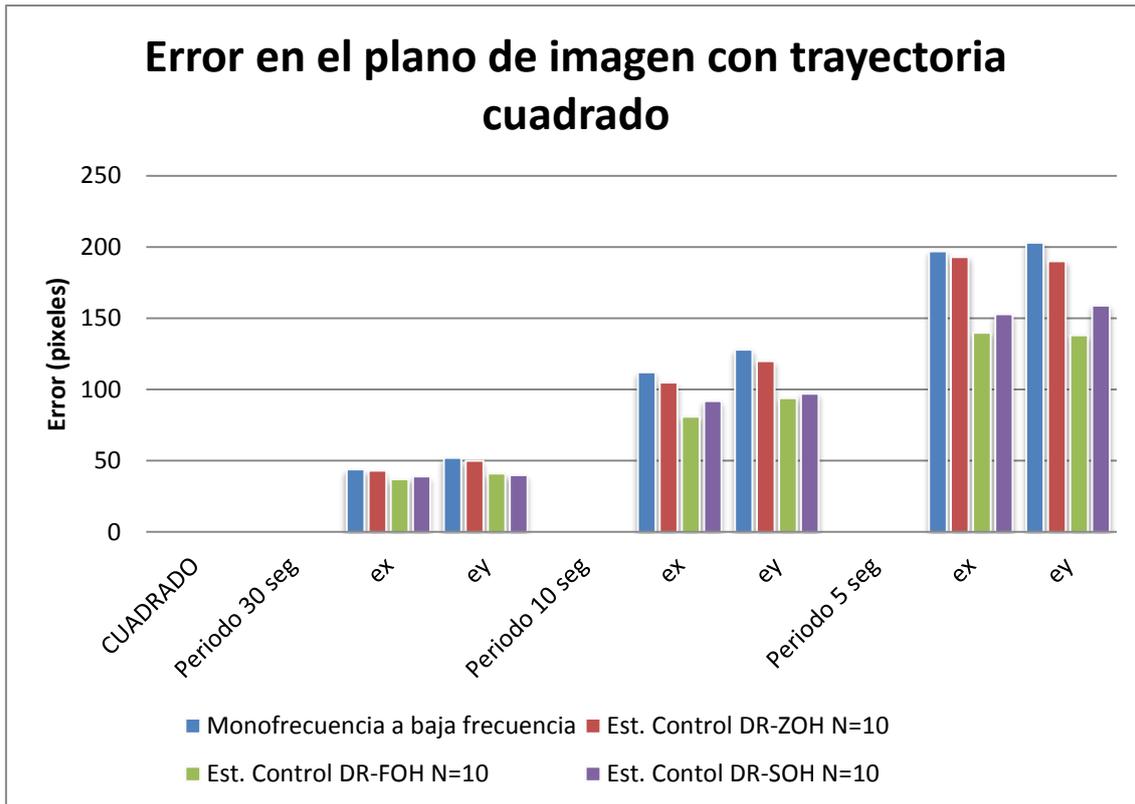


Figura 131. Comparativa del error utilizando el enfoque de estimación de la señal de control para la trayectoria cuadrado.

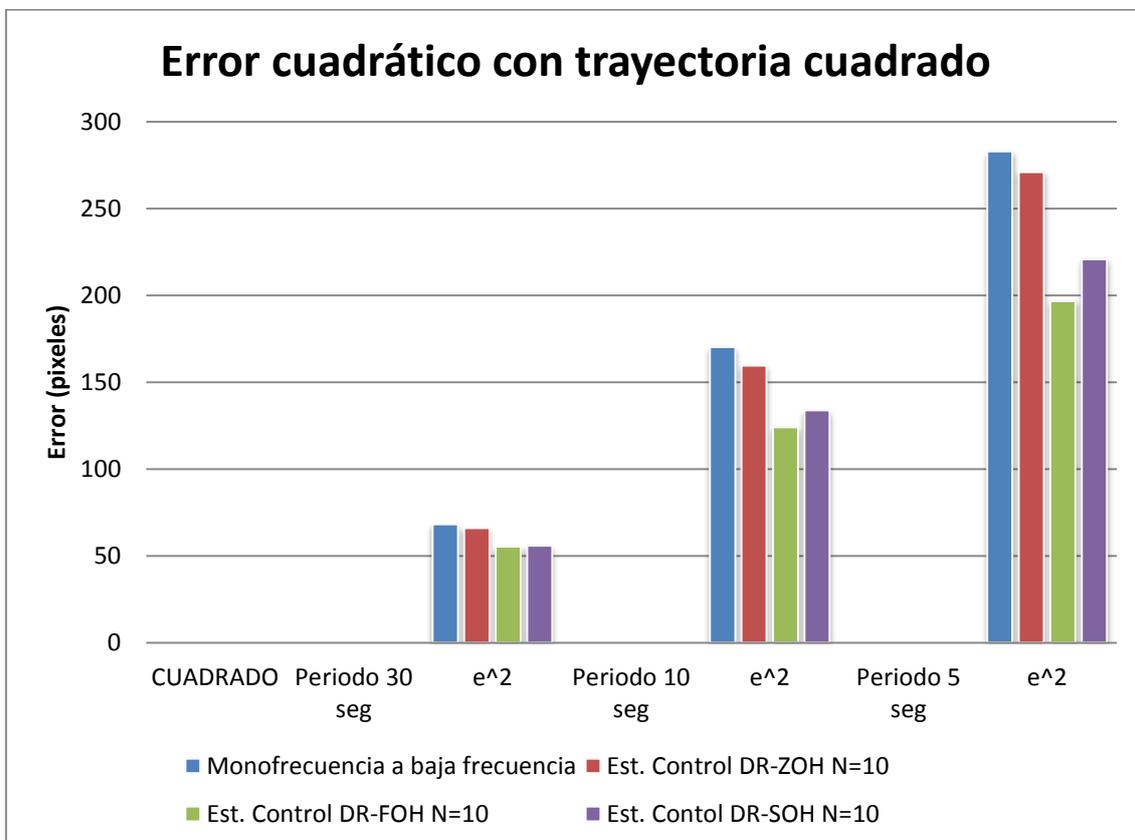


Figura 132. Comparativa del error cuadrático utilizando el enfoque de estimación de la señal de control para la trayectoria cuadrado.

- CÍRCULO

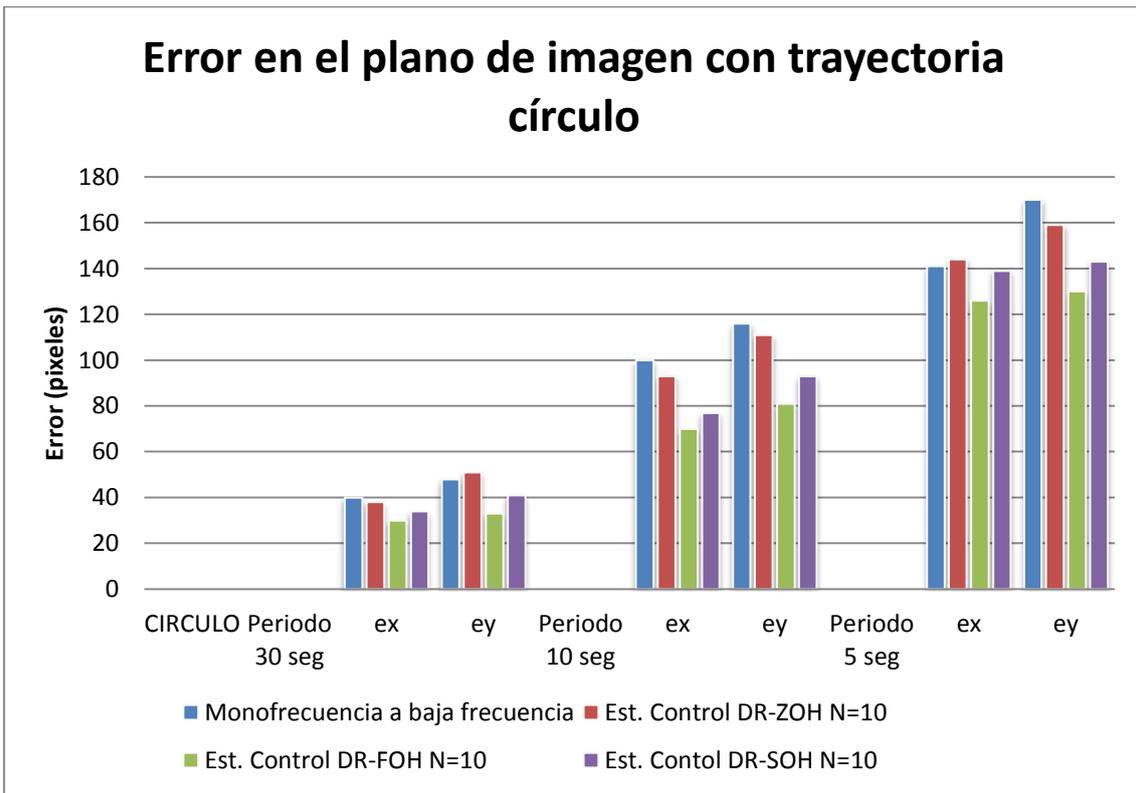


Figura 133. Comparativa del error utilizando el enfoque de estimación de la señal de control para la trayectoria círculo.

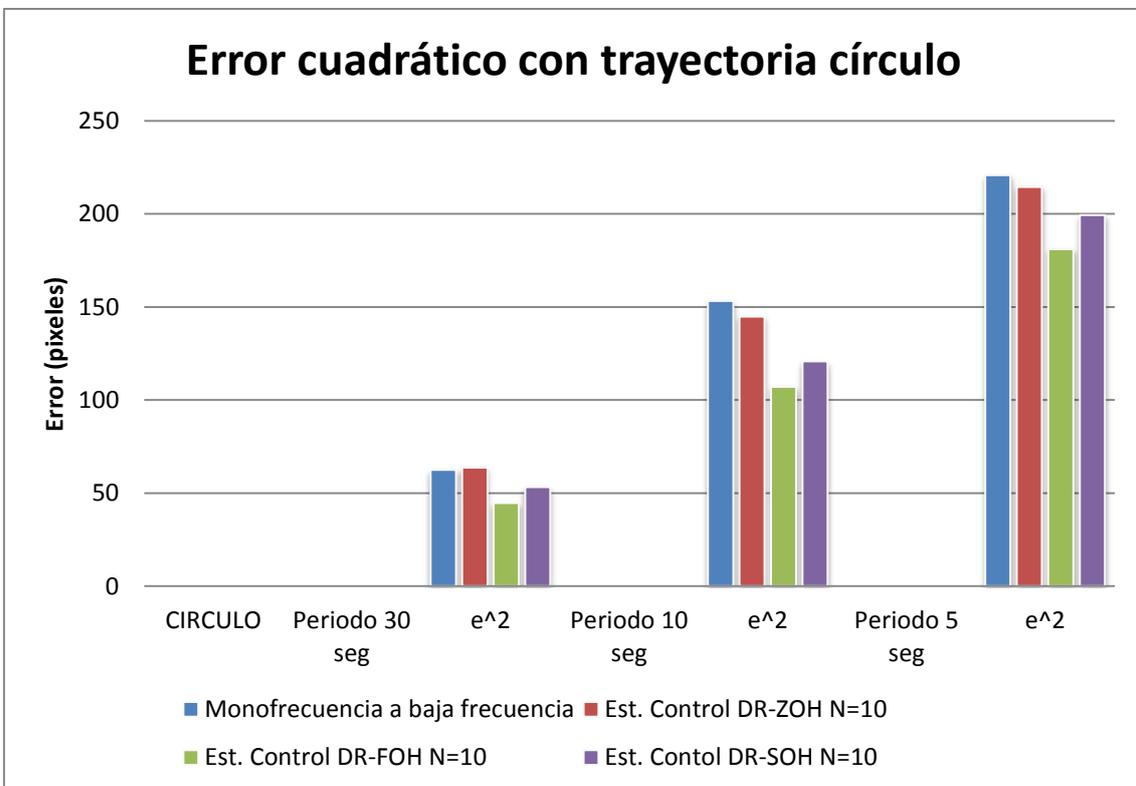


Figura 134. Comparativa del error cuadrático utilizando el enfoque de estimación de la señal de control para la trayectoria círculo.

- INFINITO

Por último los resultados obtenidos con la trayectoria infinito confirman que con este enfoque también se obtiene una mejora en el comportamiento del sistema respecto al caso monofrecuencia.

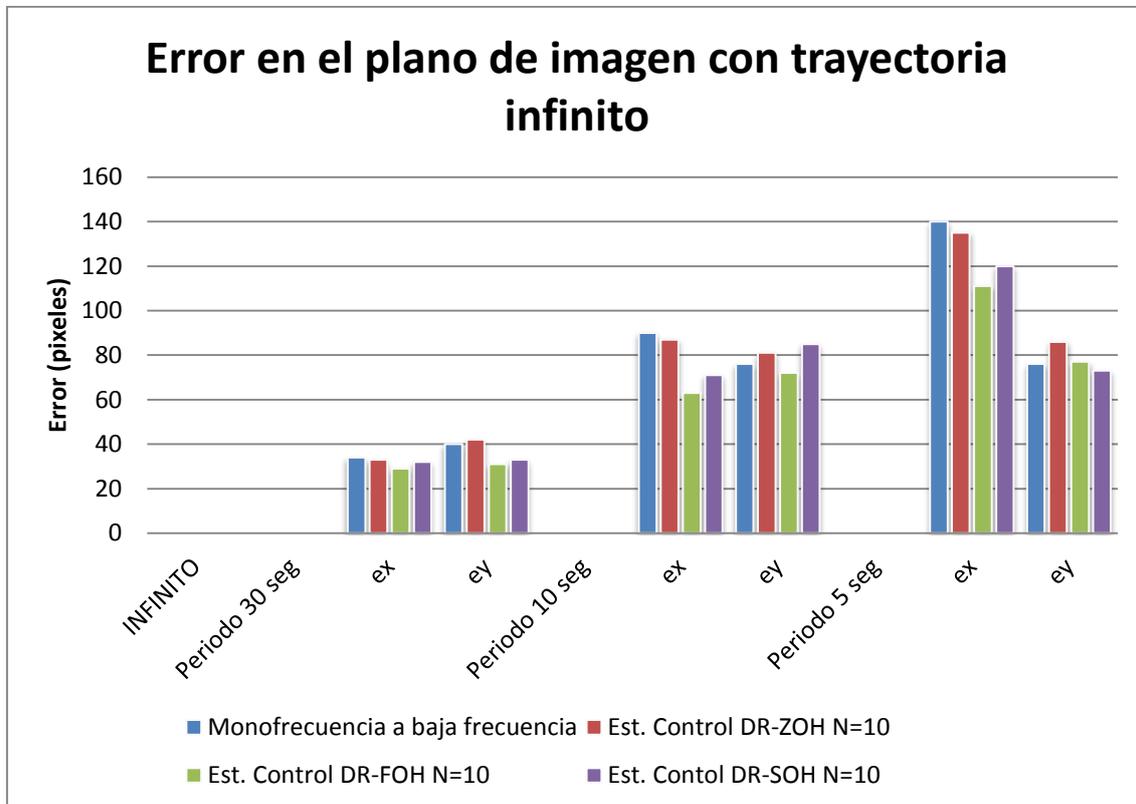


Figura 135. Comparativa del error utilizando el enfoque de estimación de la señal de control para la trayectoria infinito.

Como ha venido siendo habitual, los mejores resultados se han obtenido con el DR-FOH. Ahora la pregunta es clara, ¿cuál de los dos enfoques es el que se debería utilizar para este sistema particular? ¿Se podría realizar una mezcla de ambos enfoques?

La respuesta a la primera de las preguntas se responderá en la comparativa del siguiente apartado. La respuesta a la segunda viene dada por el diagrama de bloques de la Figura 137. En él se puede ver que si el controlador utilizado permite dividir su dinámica en dos partes, una parte lenta y una rápida, entonces se puede realizar primero una estimación de la señal de control lento, una estimación de dicha señal para la parte rápida del controlador y por consiguiente trabajar a alta frecuencia.

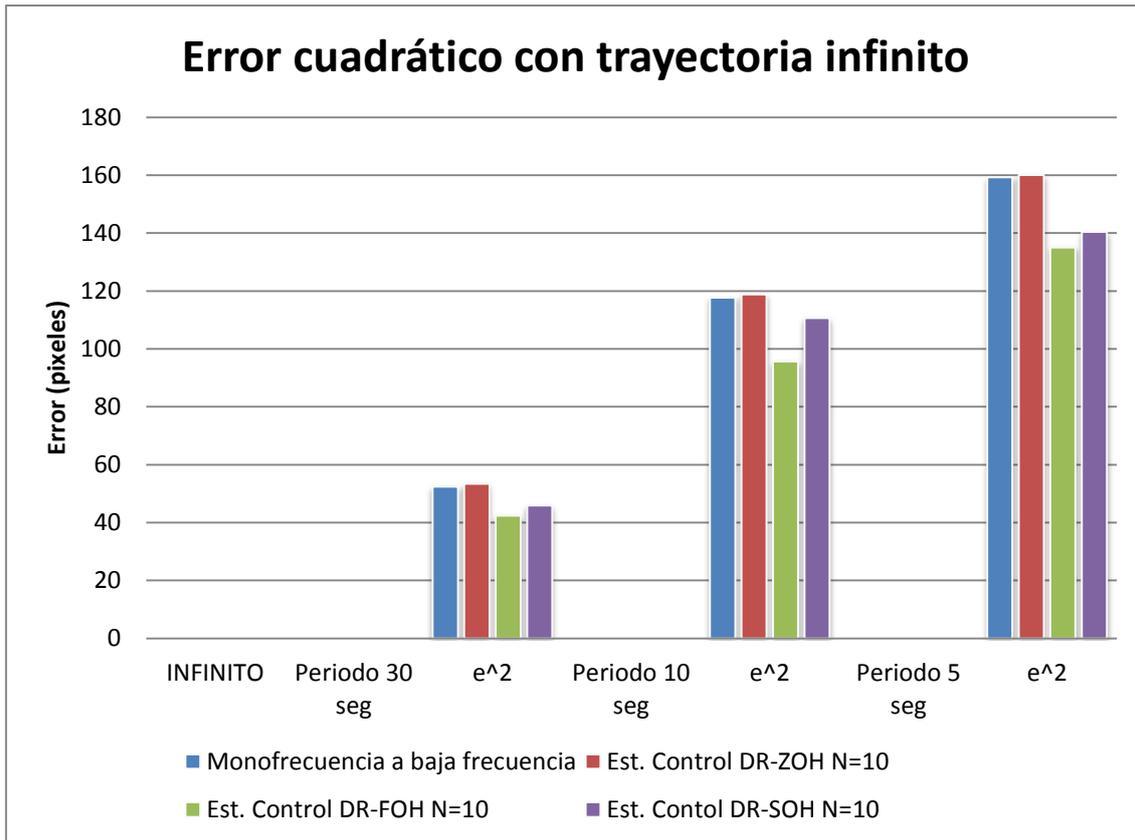


Figura 136. Comparativa del error cuadrático utilizando el enfoque de estimación de la señal de control para la trayectoria infinito.

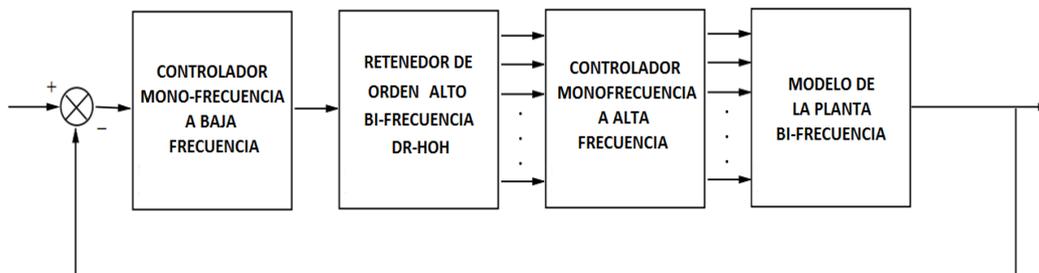


Figura 137. Diagrama de bloques genérico multifrecuencia.

4.2.1.3.- Comparativa entre ambos enfoques

En este apartado se va a comprobar, con los resultados experimentales obtenidos, cuál de los dos enfoques es el más adecuado en este sistema concreto. Con este resultado se dará una idea clara de cómo se deberá hacer la implementación del control multifrecuencia en el sistema Image-Based Visual Servoing Multifrecuencia.

A diferencia de lo que se ha venido haciendo en los apartados anteriores, con motivo de no extender la memoria, los resultados que se van a mostrar son los

relativos al error cuadrático, entendiendo que dan una idea bastante clara del comportamiento del sistema. Por lo tanto los resultados han sido los siguientes:

- TRIANGULO

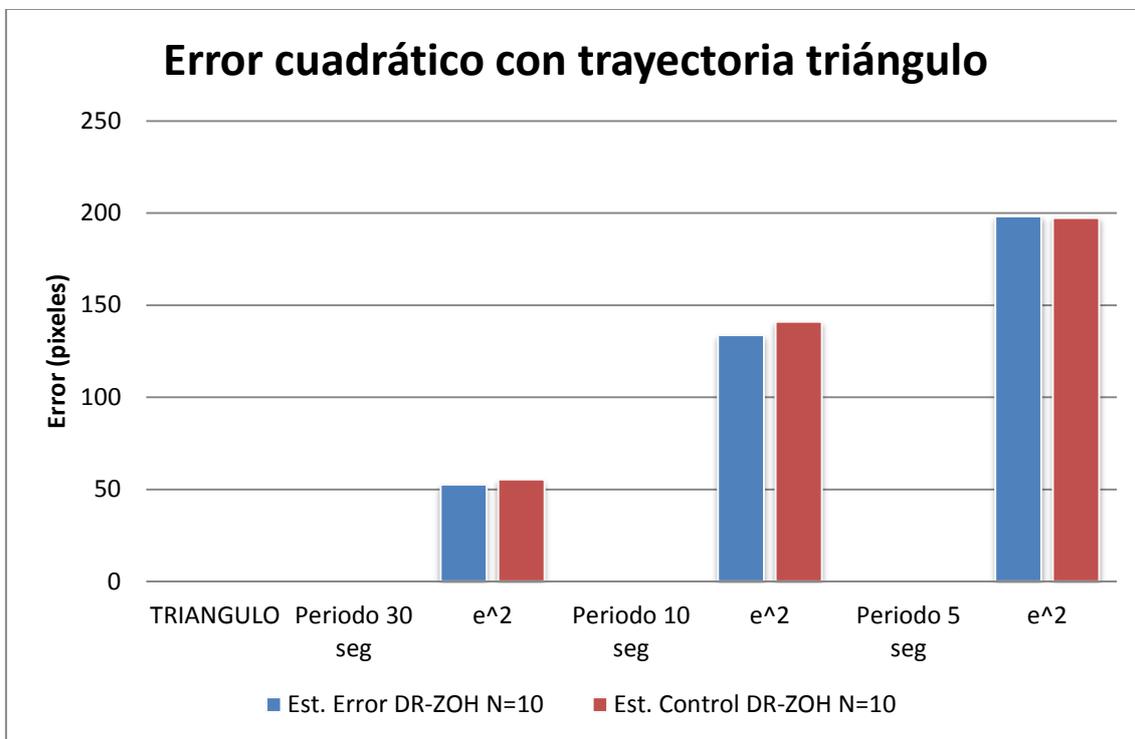


Figura 138. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria triángulo empleando como interfaz un DR-ZOH.

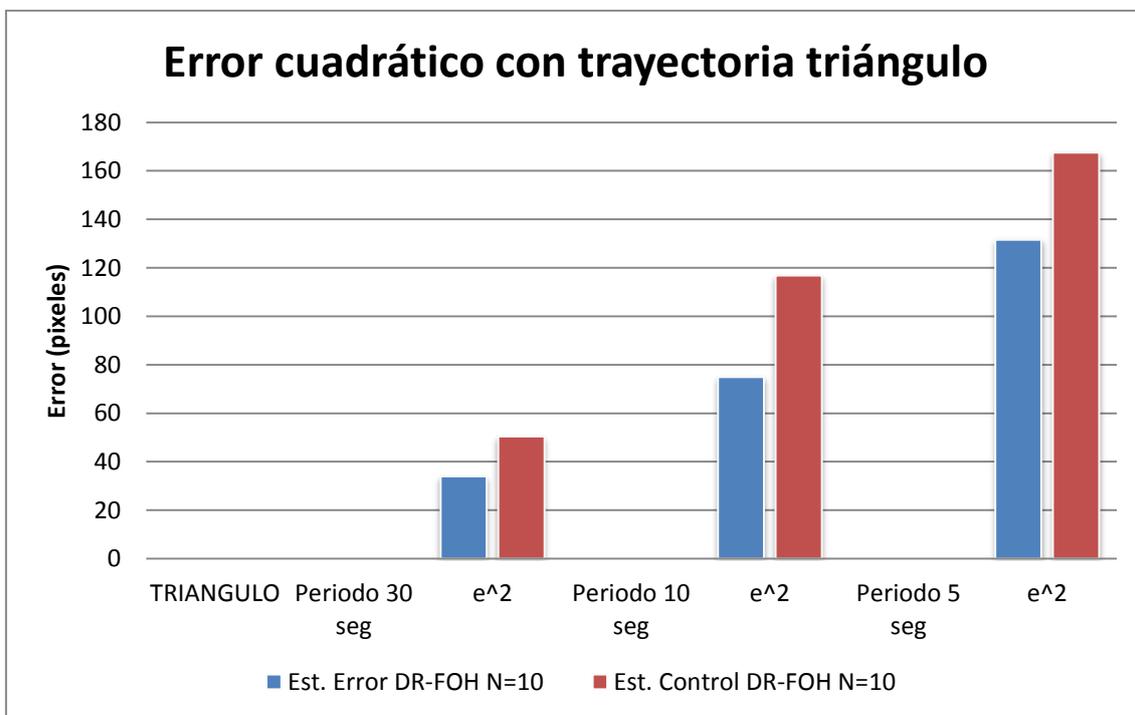


Figura 139. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria triángulo empleando como interfaz un DR-FOH.

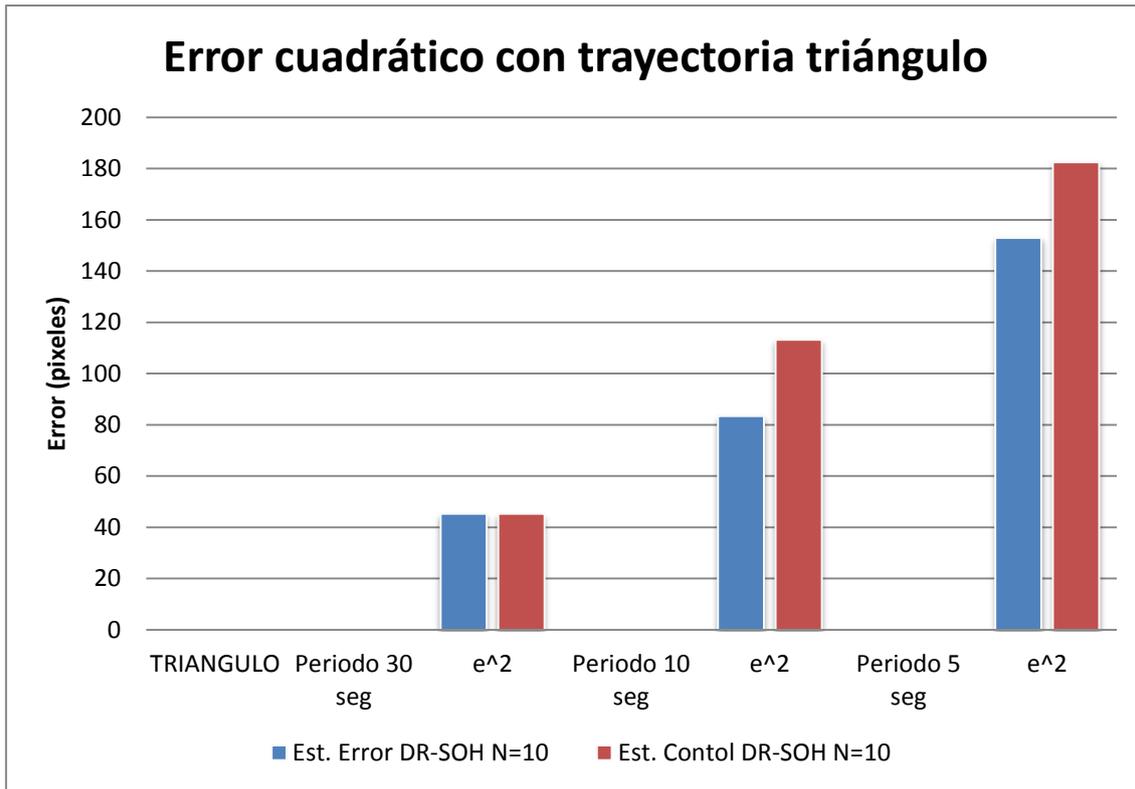


Figura 140. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria triángulo empleando como interfaz un DR-SOH.

- CUADRADO

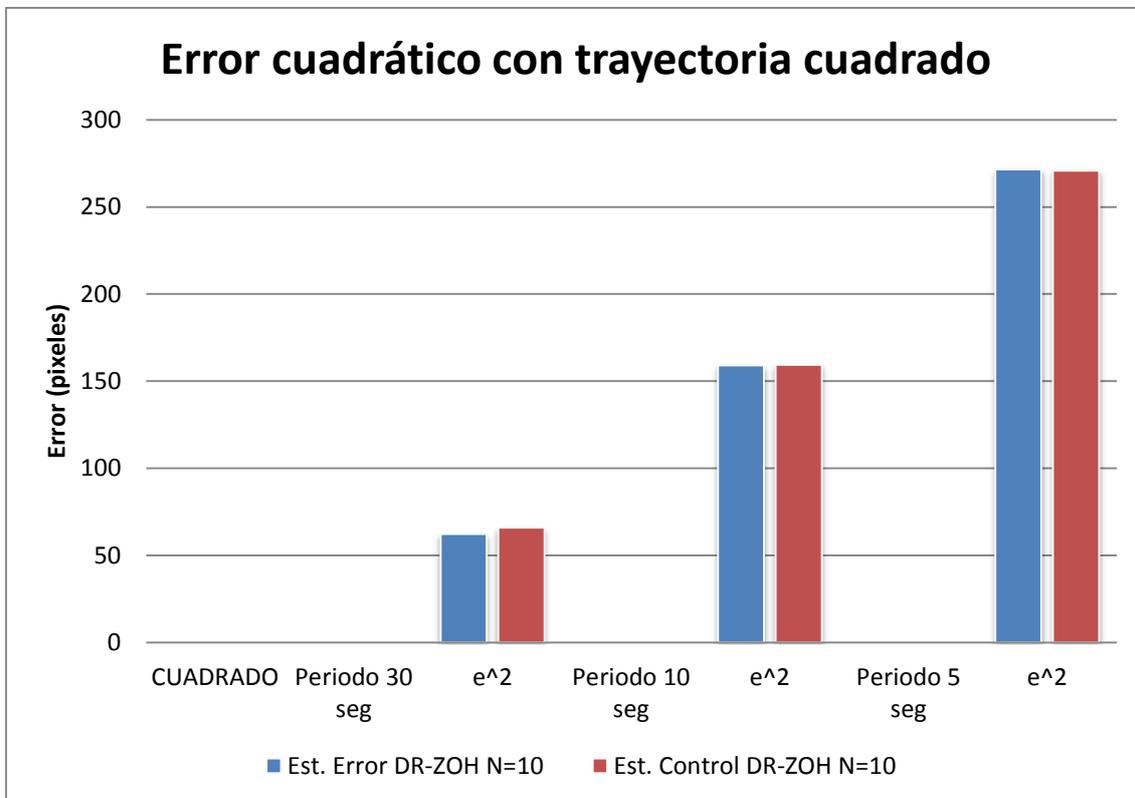


Figura 141. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria cuadrado empleando como interfaz un DR-ZOH.

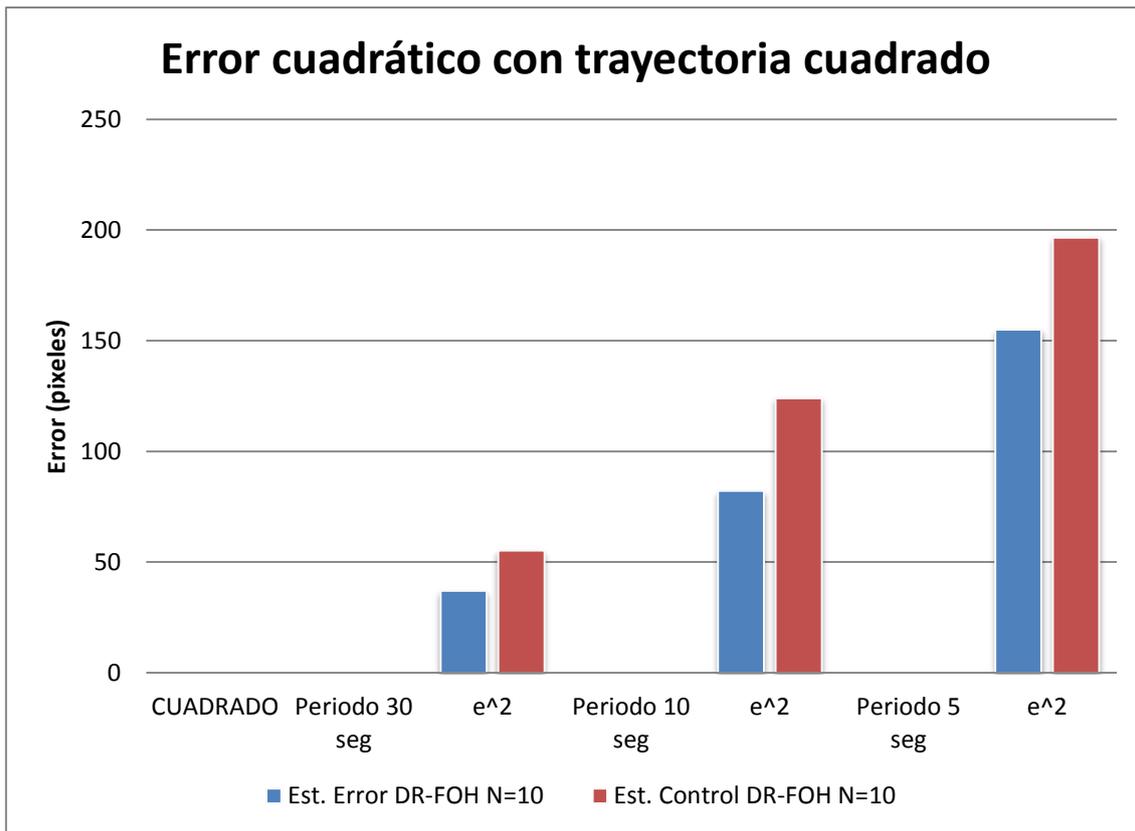


Figura 142. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria cuadrado empleando como interfaz un DR-FOH.

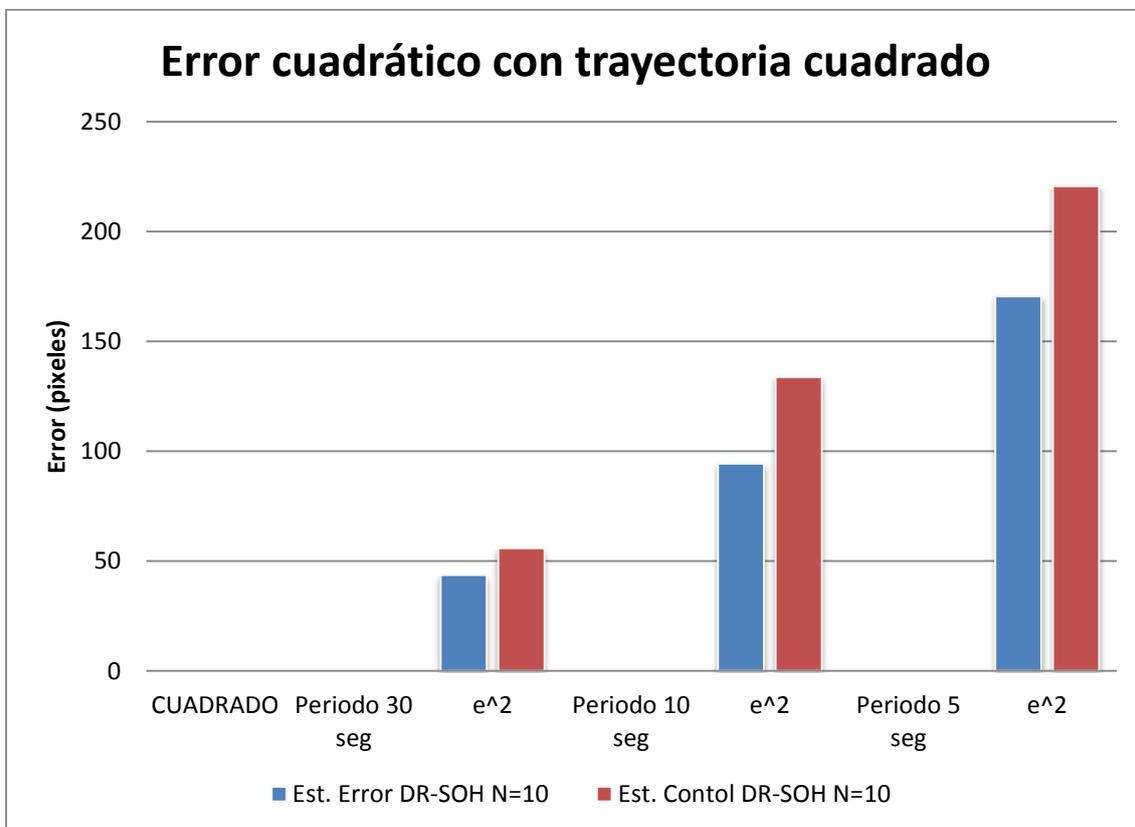


Figura 143. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria cuadrado empleando como interfaz un DR-SOH.

- CÍRCULO

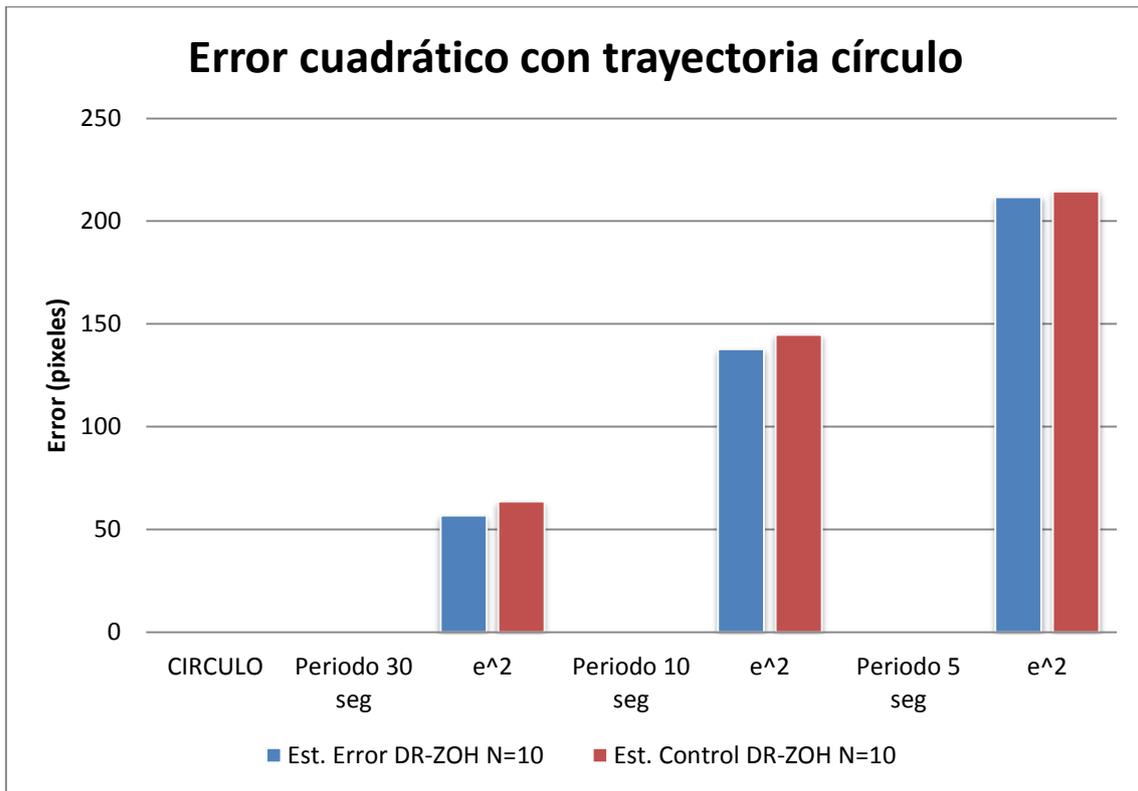


Figura 144. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria círculo empleando como interfaz un DR-ZOH.

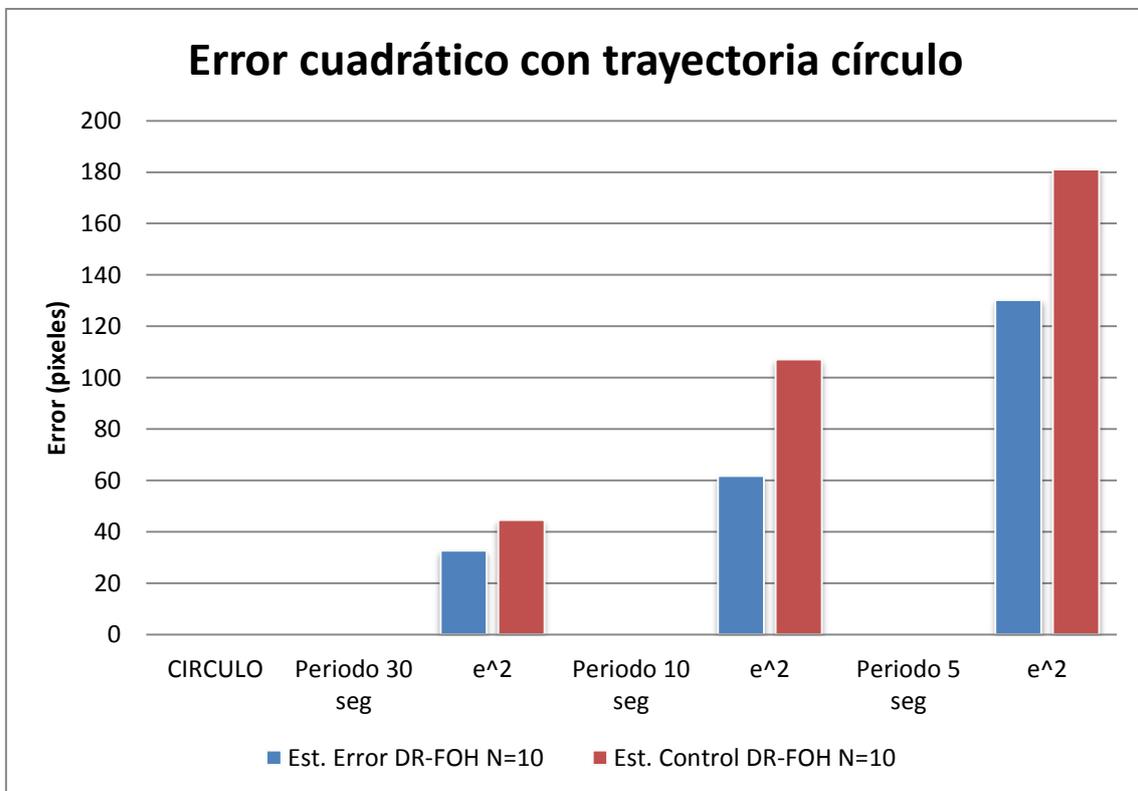


Figura 145. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria círculo empleando como interfaz un DR-FOH.

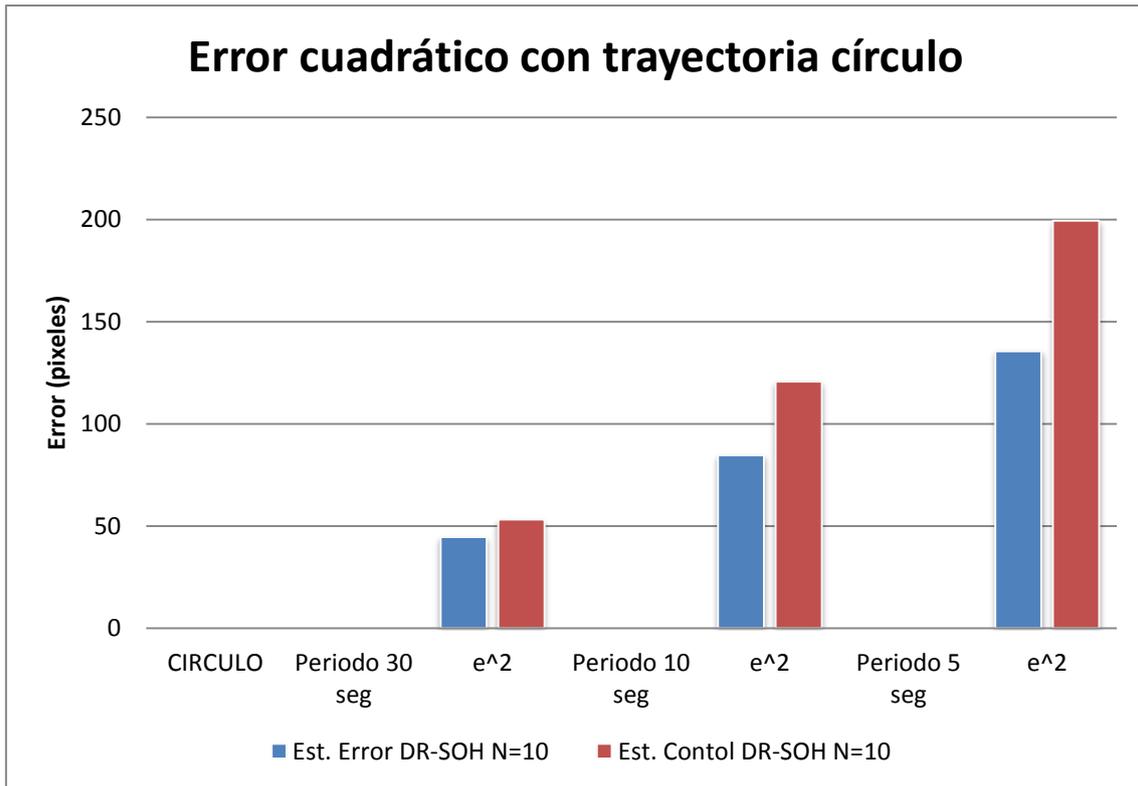


Figura 146. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria círculo empleando como interfaz un DR-SOH.

- INFINITO

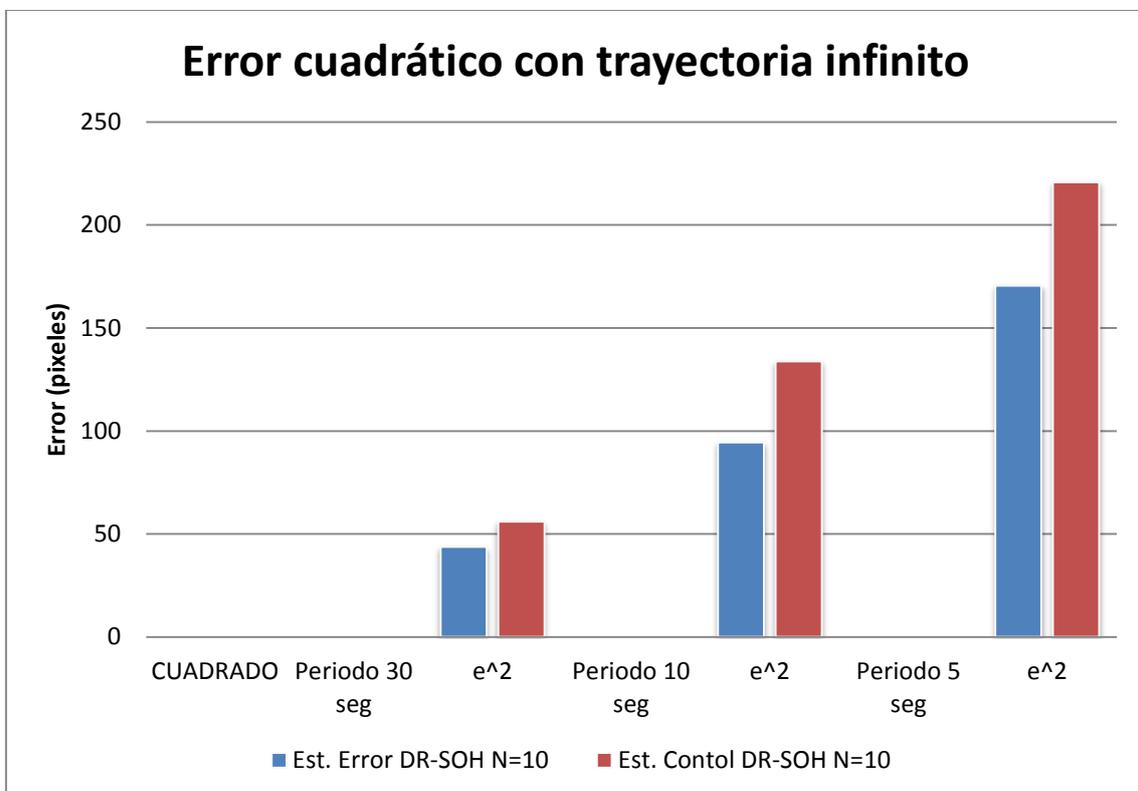


Figura 147. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria infinito empleando como interfaz un DR-ZOH.

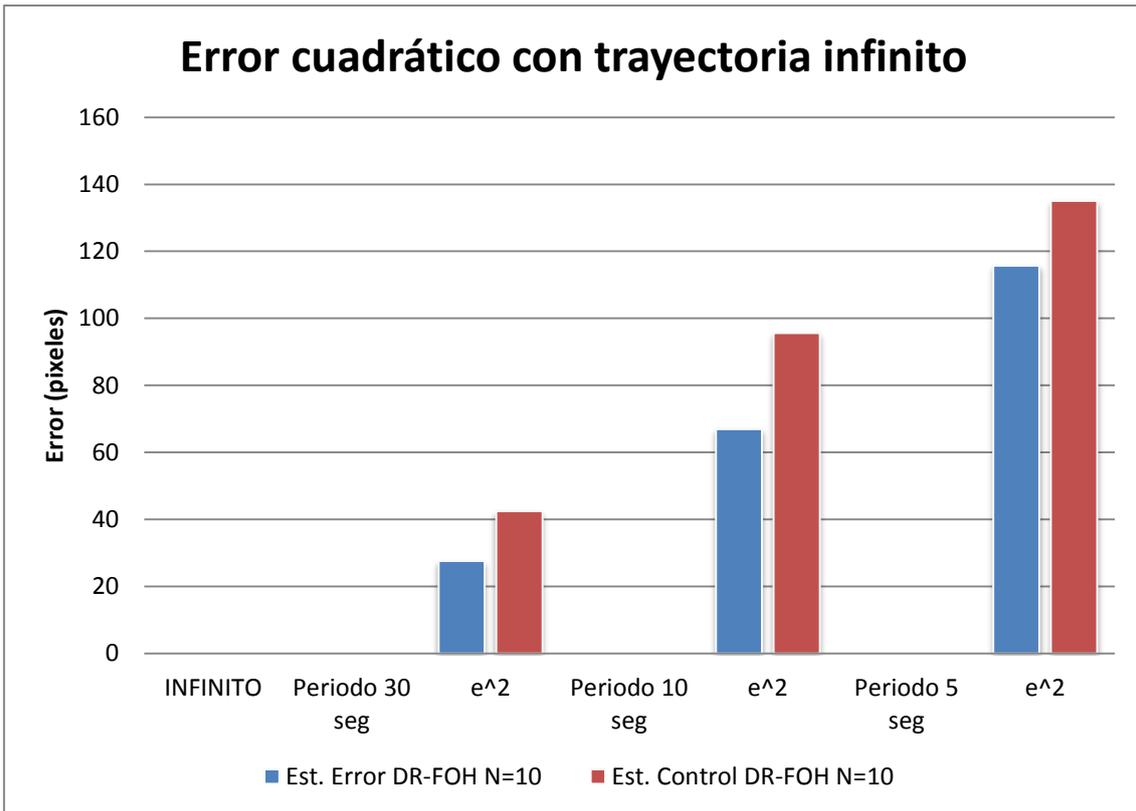


Figura 148. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria infinito empleando como interfaz un DR-FOH.

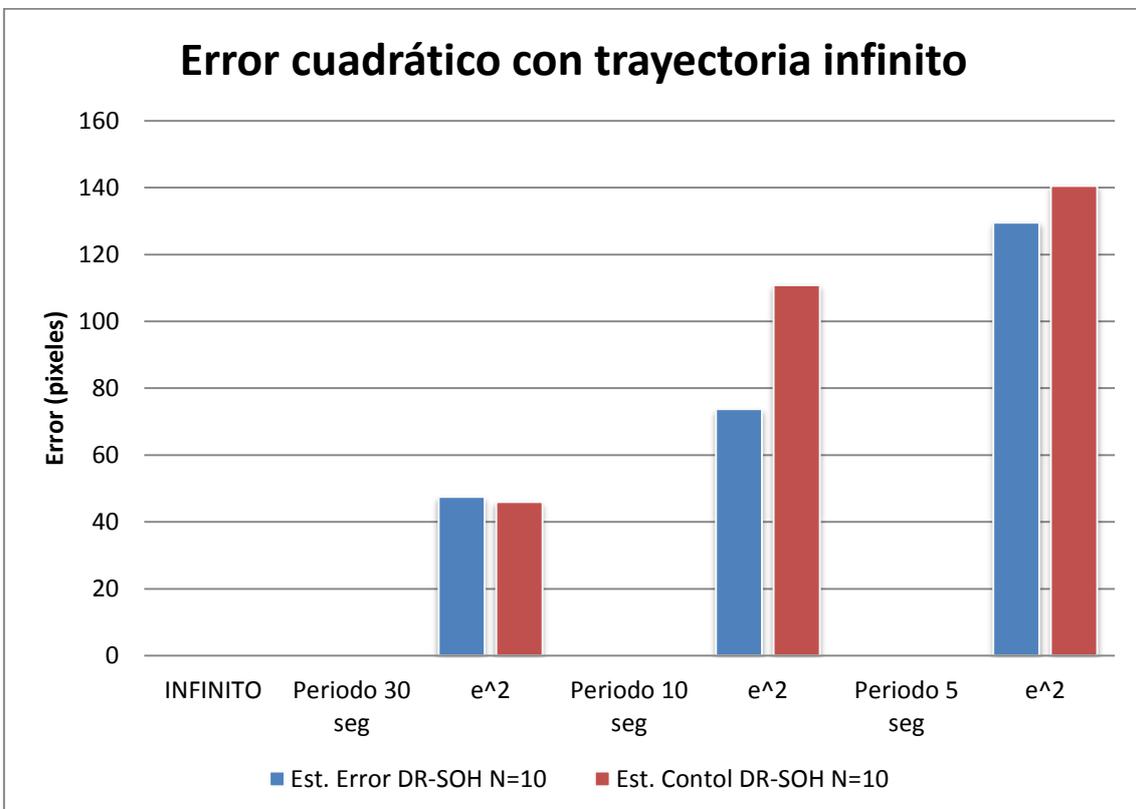


Figura 149. Comparativa del error cuadrático entre los enfoques de estimación de las señales de control y de error para la trayectoria infinito empleando como interfaz un DR-SOH.

Los resultados anteriores demuestran que los mejores resultados se obtienen cuando se emplean los estimadores multifrecuencia con el enfoque de estimación de la señal de error. Por lo tanto, en el sistema Image-Based Visual Servoing Multifrecuencia se va a emplear el enfoque de estimación de la señal de error.

En el siguiente apartado lo que se verá es la diferencia del comportamiento del sistema cuando se realiza un enfoque síncrono o cuando el enfoque utilizado es el asíncrono.

4.2.2.-Sistemas Multifrecuenciales asíncronos.

La idea básica de este enfoque es la siguiente: cuando la variable medida está lista para ser actualizada, se actualiza y se deja de estimar. En un sistema real lo que realmente sucede es lo que se muestra en la Figura 150. Como se ha dicho, para no tener pérdidas de tiempos de ejecución (planificación de las tareas), lo que se hecho es decir que el metaperiodo de la tarea de adquisición y procesado es mayor de lo que realmente. Es decir, se ha sobredimensionado. Eso implica que en el 95% de los casos, dicha tarea tendrá que esperar a que finalice la tarea de control para volver a ejecutarse.

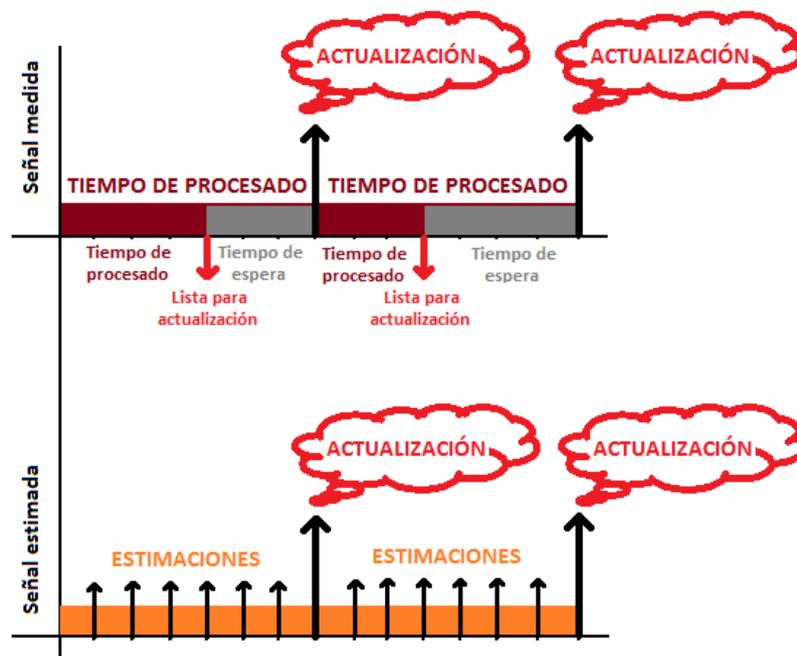


Figura 150. Ejecución de las tareas en un sistema real.

Además y como se muestra en la figura anterior, el tiempo de espera es incierto ya que depende del tiempo de adquisición de la cámara, del procesado de la imagen, etc.

Por lo tanto lo que se debe hacer es reformular la teoría clásica del enfoque multifrecuencia para incluir esta incertidumbre y así realizar un control asíncrono. La Figura 151 muestra cómo cada vez que se puede actualizar la variable medida, la tarea de control deja de realizar la estimación y utiliza la actualización para procesar la acción. Por otro lado, una vez actualizada la variable medida, la tarea de visión puede volver a adquirir y procesar una nueva imagen.



Figura 151. Enfoque multifrecuencia asíncrono.

En la Figura 151 se puede ver que el tiempo de obtención de la señal medida es incierto, por lo que unas veces puede ser mayor y otras puede ser menor. Es por ello que, en vez de coger el peor de los casos en el metaperiodo de dicho sistema, lo que se hace es cogerlo, a priori, pero en caso de que la variable medida este lista, se actualiza y se deja el proceso de estimación.

Con esta sencilla idea se evita el que cualquier de los dos procesos paralelos (*threads*) esté esperando a que el otro termine sin hacer nada.

El siguiente apartado está dedicado a mostrar la comparativa entre el comportamiento obtenido con el sistema multifrecuencia realizado mediante el enfoque clásico (síncrono) y el sistema multifrecuencia realizado con el enfoque asíncrono.

4.2.2.1.- Comparativa entre el enfoque multifrecuencia síncrono y el asíncrono.

Siguiendo el mismo procedimiento de los apartados anteriores, la comparativa se va a realizar con distintas trayectorias a distintas velocidades del objeto con cada una de las interfaces Dual Rate.

Para no extender demasiado la memoria y como se entiende que con el error cuadrático se ve con claridad lo que se está intentando demostrar, se va a utilizar el error cuadrático para el estudio de esta comparativa.

- TRIANGULO

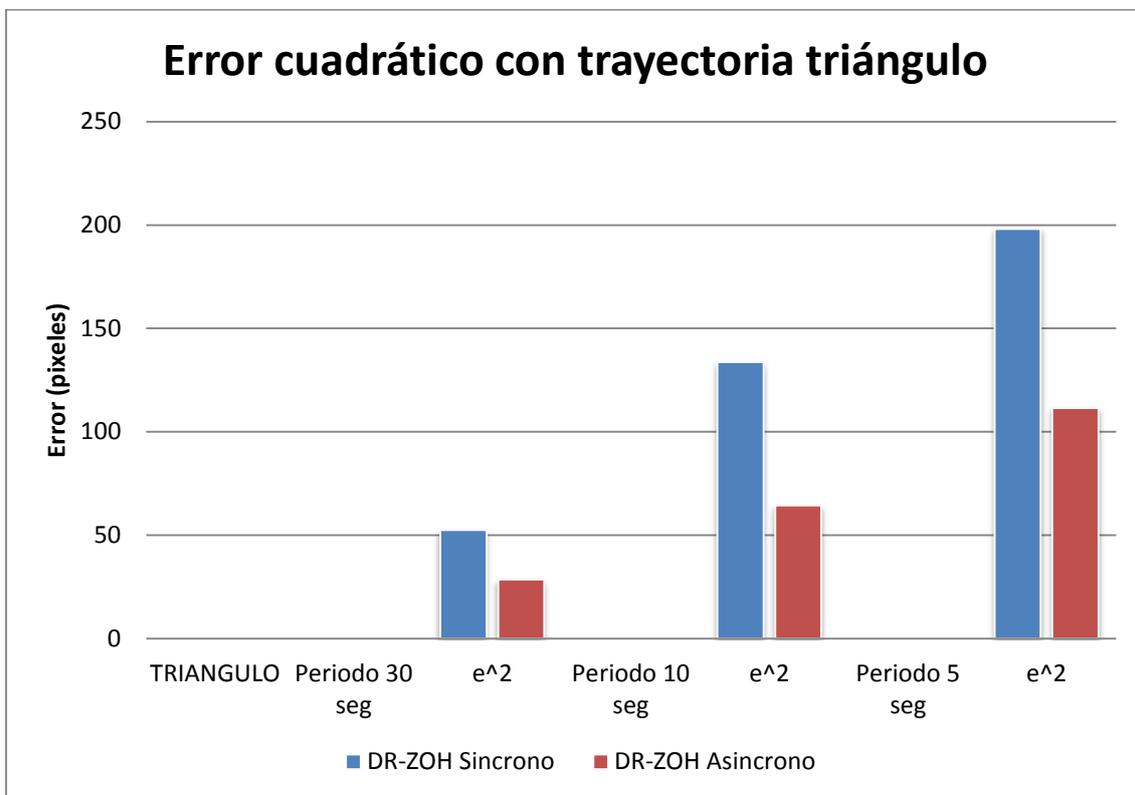


Figura 152. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error cuadrático para la trayectoria triángulo con DR-ZOH.

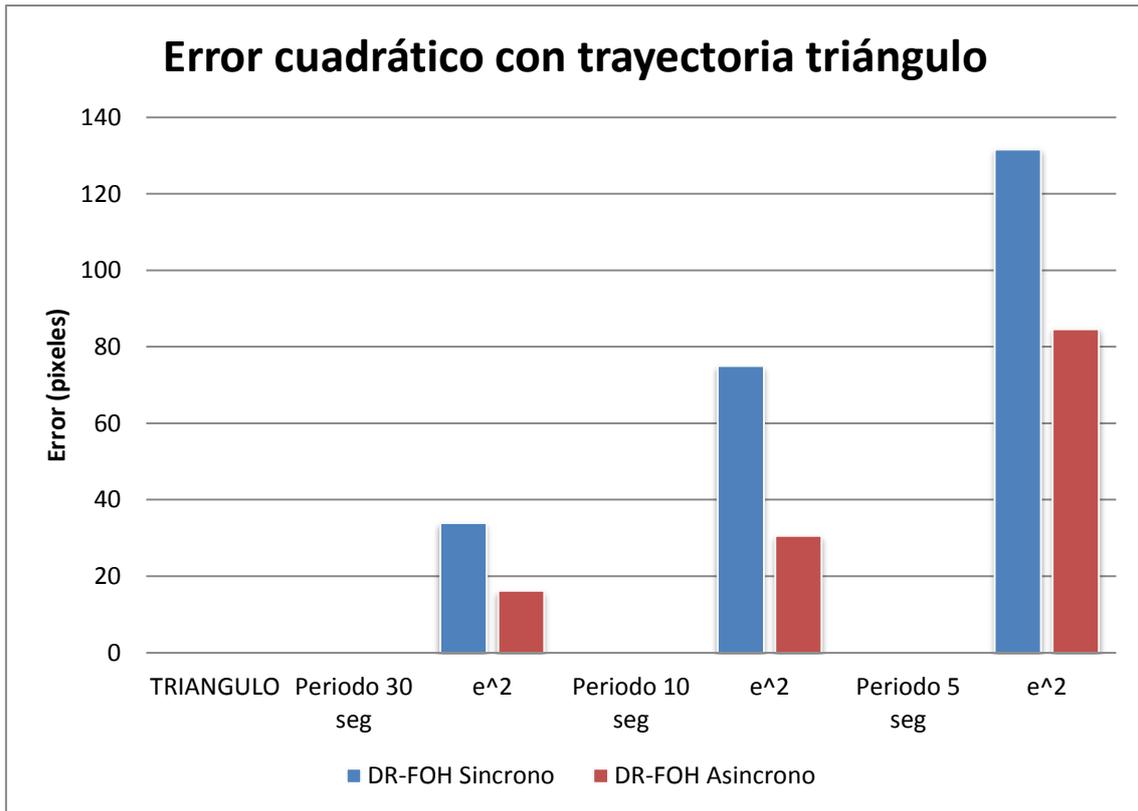


Figura 153. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error cuadrático para la trayectoria triángulo con DR-FOH.

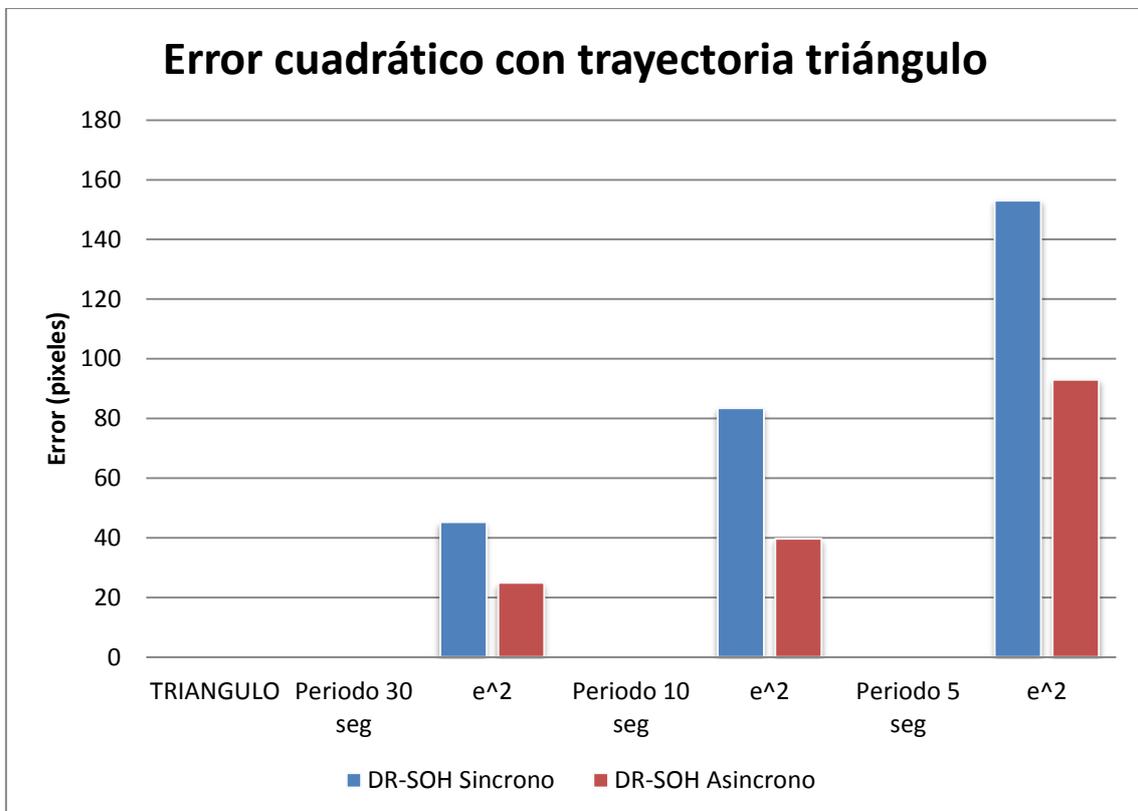


Figura 154. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error cuadrático para la trayectoria triángulo con DR-SOH.

- CUADRADO

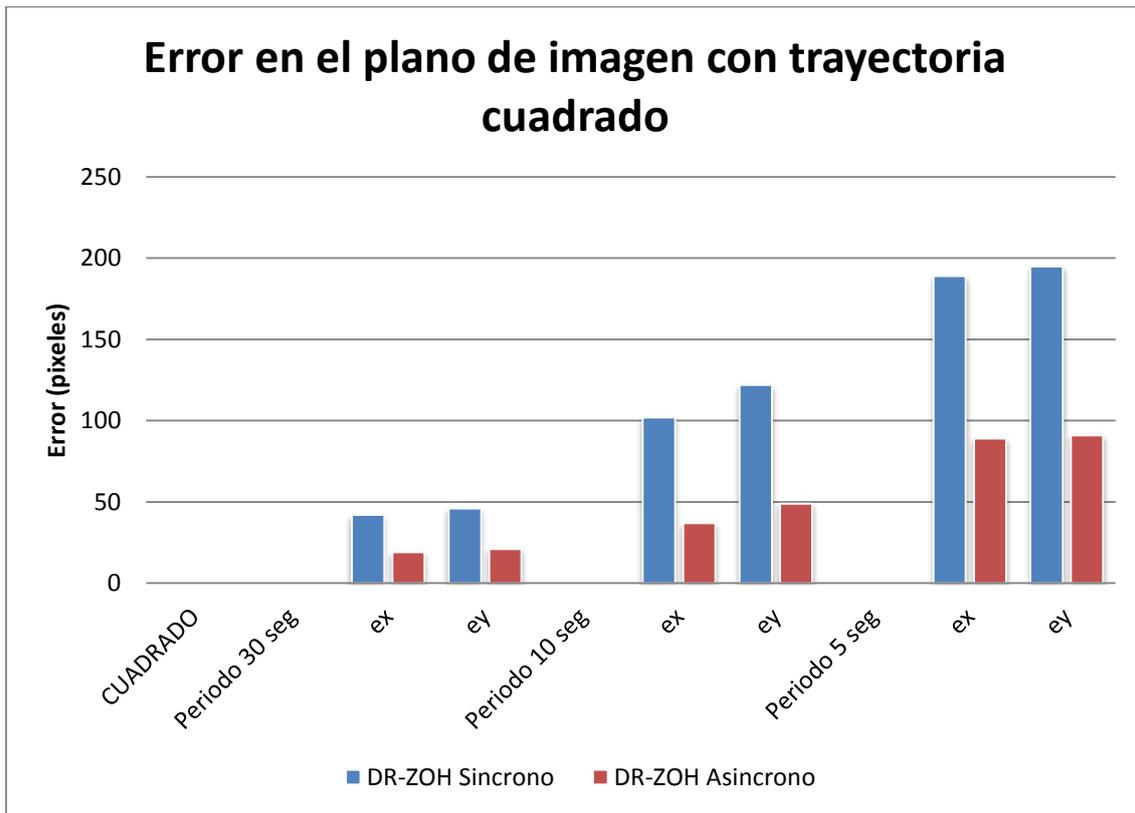


Figura 155. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria cuadrado con DR-ZOH.

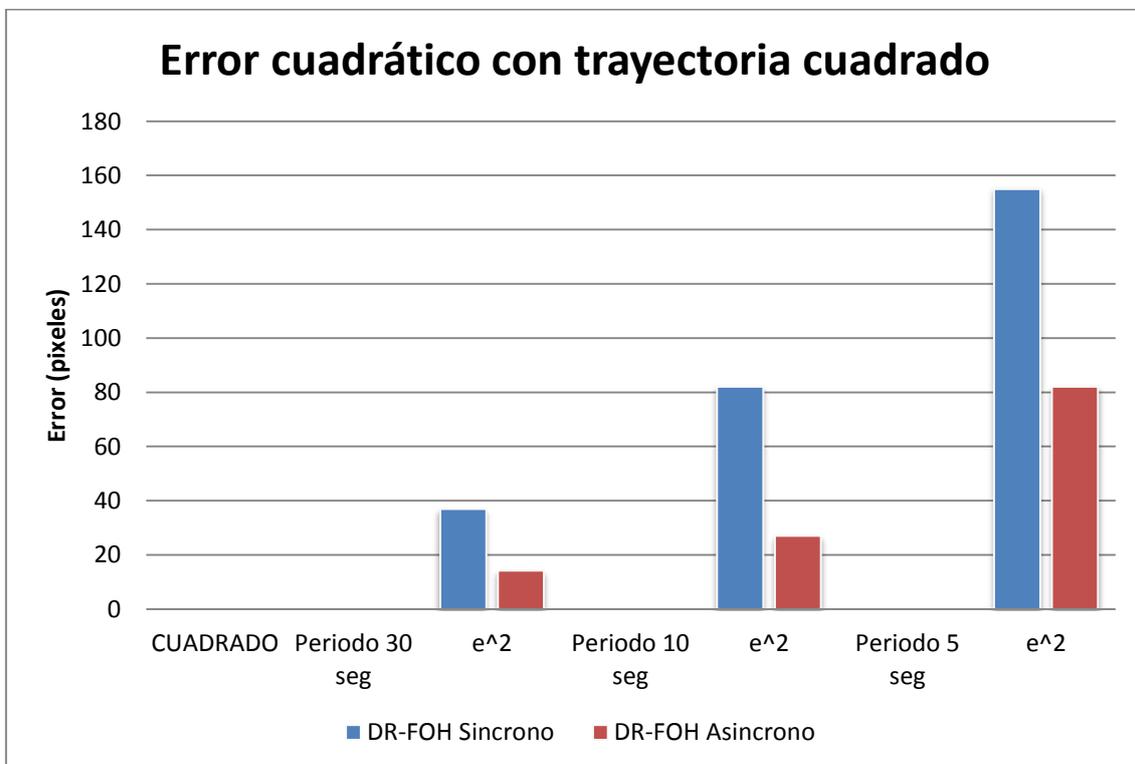


Figura 156. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria cuadrado con DR-FOH.

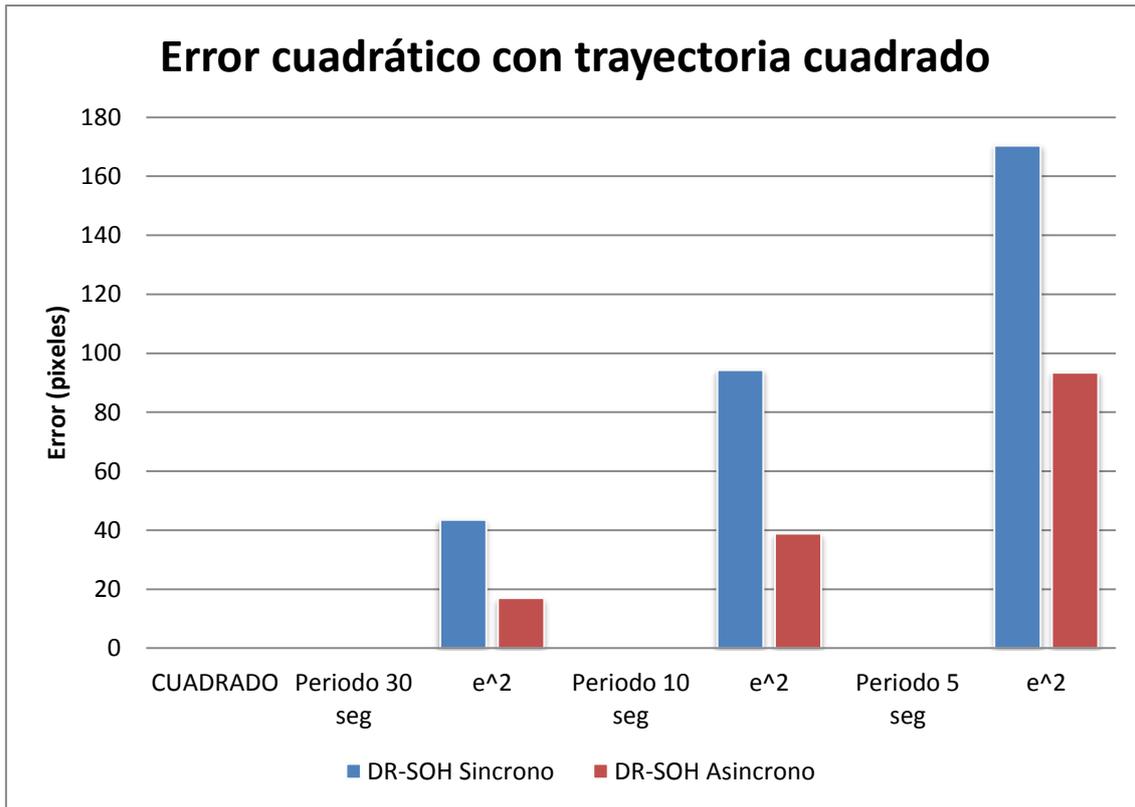


Figura 157. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria cuadrado con DR-SOH.

- CIRCULO

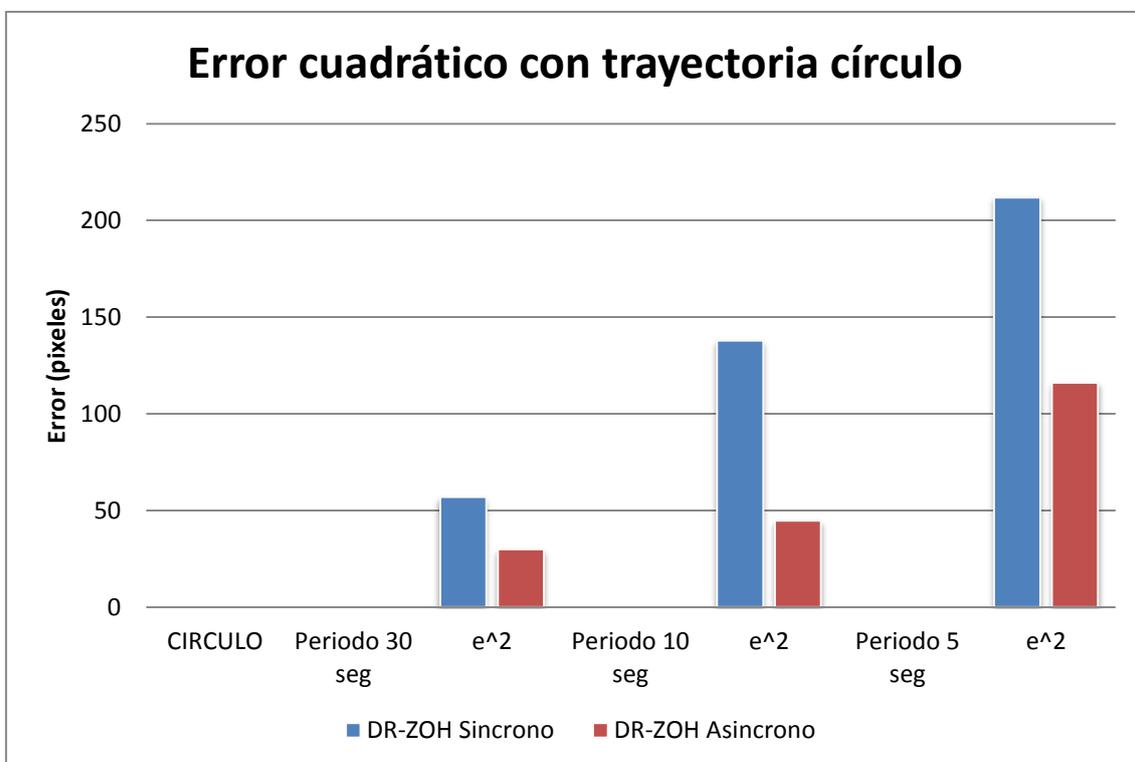


Figura 158. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria círculo con DR-ZOH.

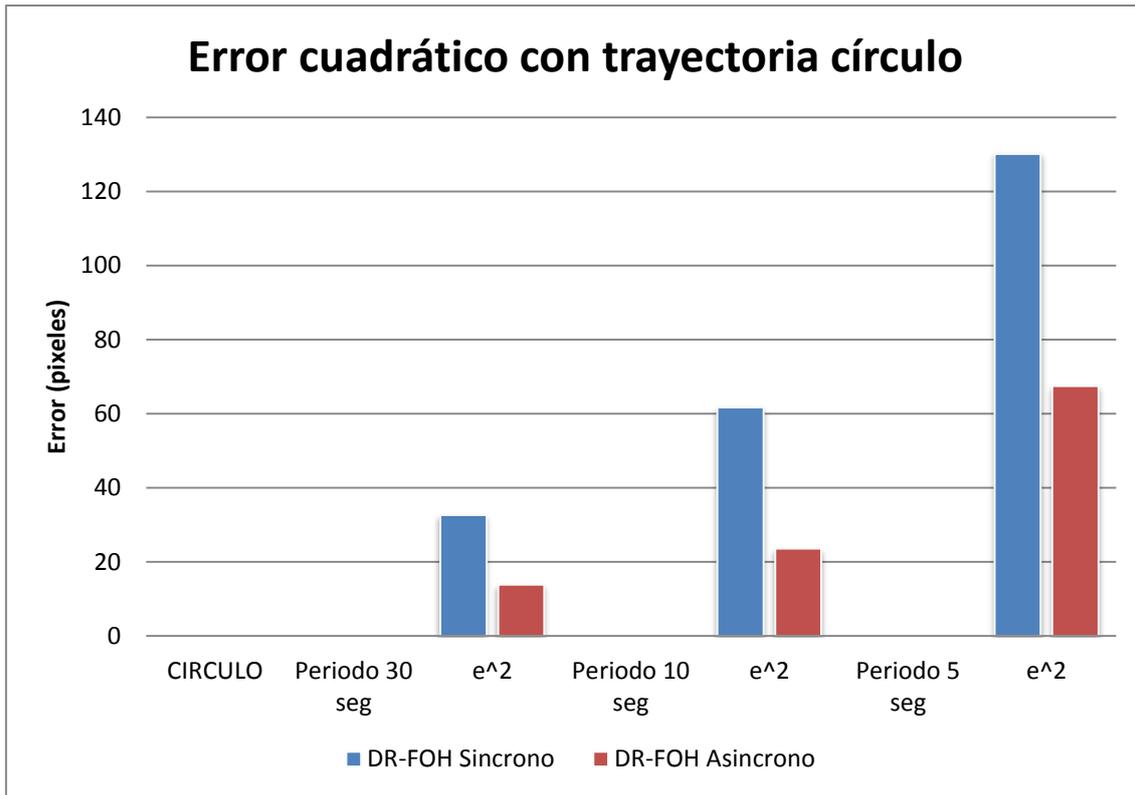


Figura 159. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria círculo con DR-FOH.

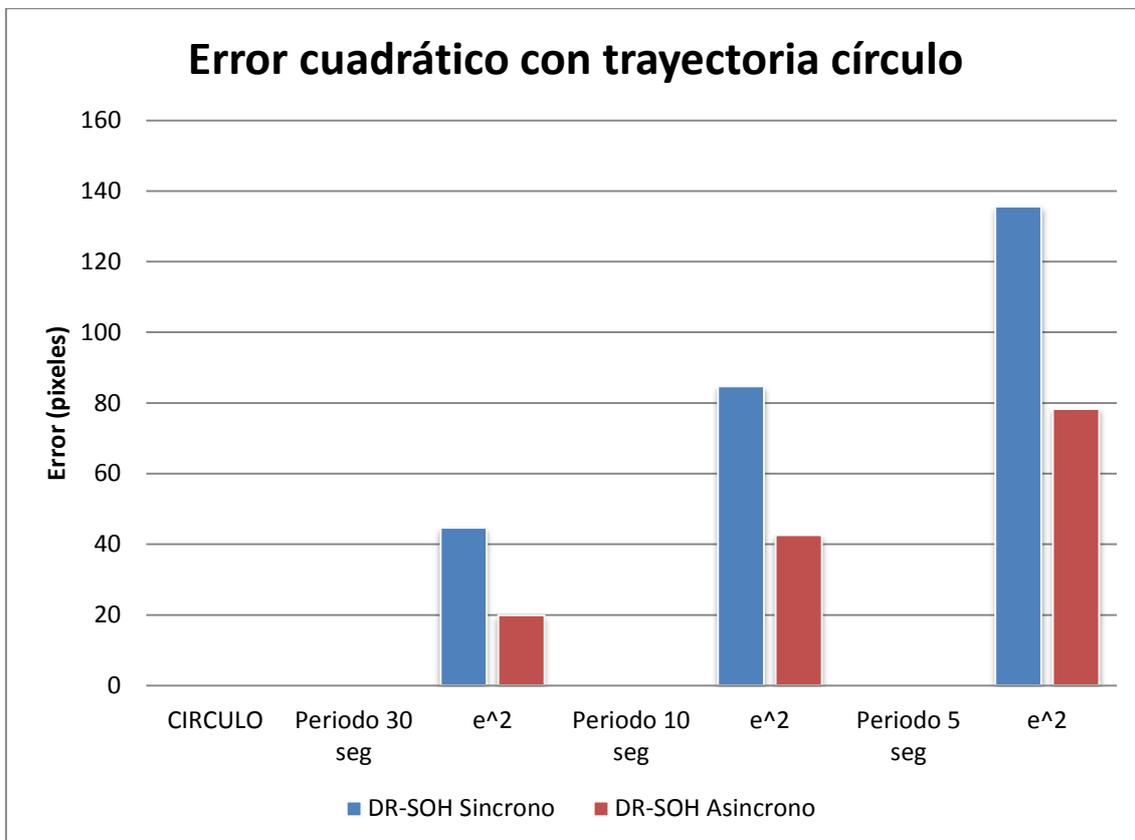


Figura 160. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria círculo con DR-SOH.

- INFINITO

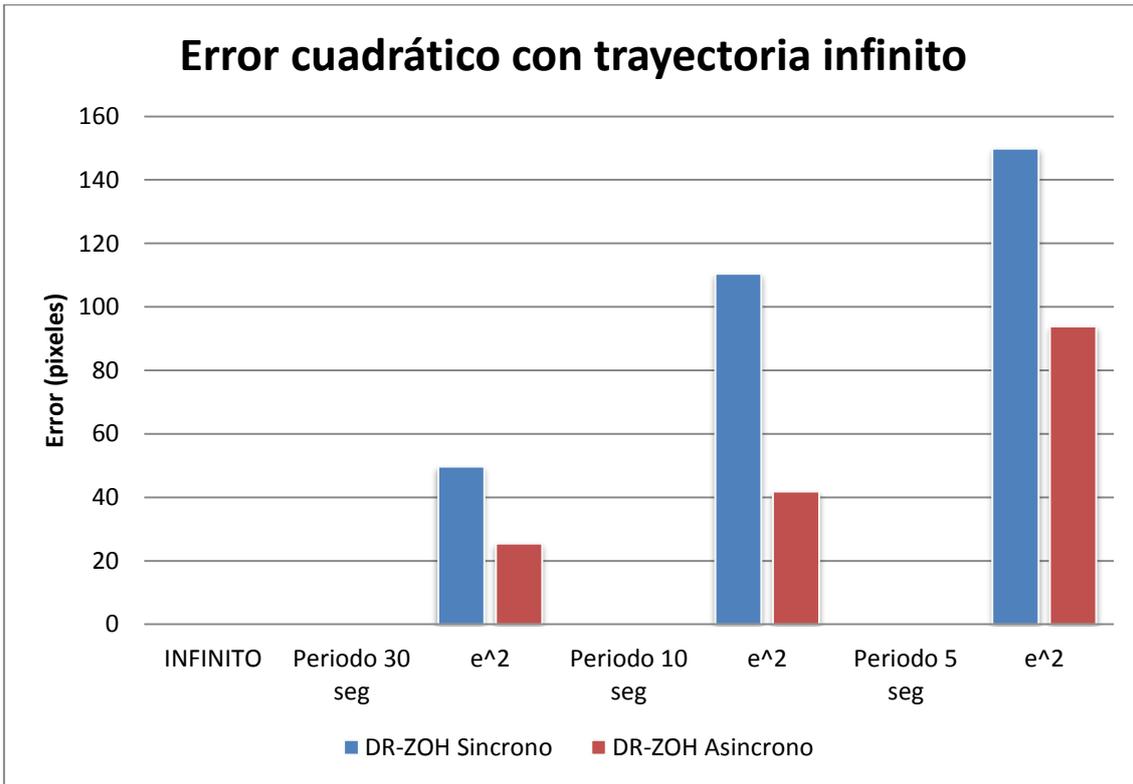


Figura 161. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria infinito con DR-ZOH.

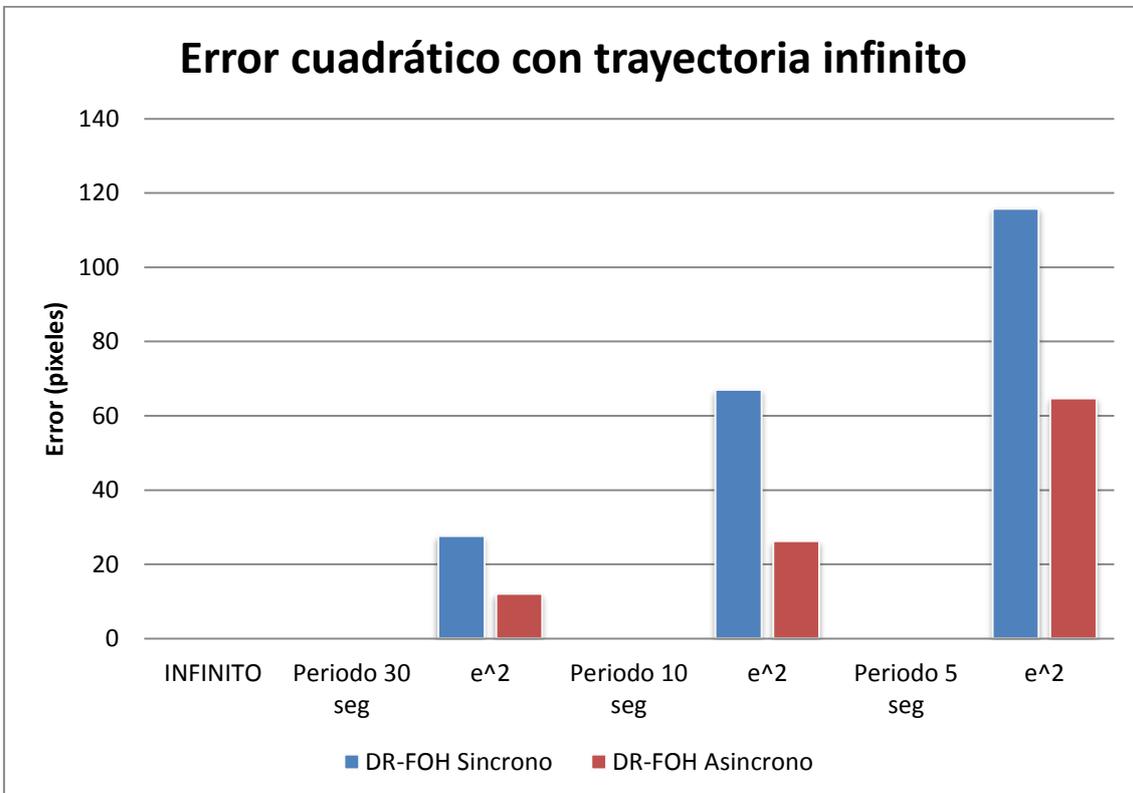


Figura 162. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria infinito con DR-FOH.

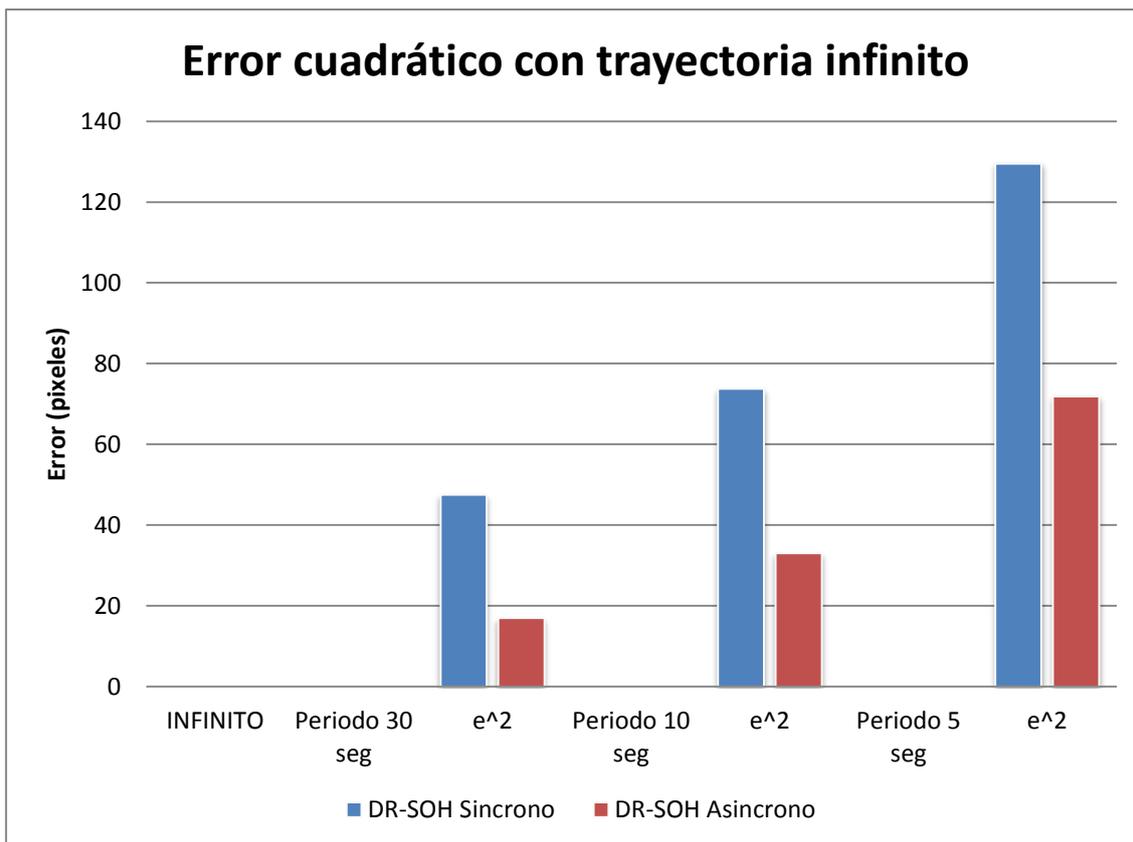


Figura 163. Comparativa entre el enfoque multifrecuencia síncrono y el enfoque asíncrono, error en los ejes del plano de imagen para trayectoria infinito con DR-SOH.

Los resultados obtenidos manifiestan algo que ya intuíamos y es que cuando se utilizan herramientas asíncronas para sistemas asíncronos el comportamiento del sistema es mucho mejor en todos los aspectos que cuando se emplean herramientas síncronas en sistemas asíncronos. La analogía con los sistemas no-lineales tiene el mismo resultado.

4.2.3.-Conclusiones

En este apartado se ha realizado el estudio de un sistema no lineal aplicando técnicas Multifrecuenciales a un controlador lineal y los resultados han sido los siguientes:

- La utilización de técnicas Multifrecuenciales para sistemas muestreados hacen que el comportamiento del sistema sea mucho mejor que el control clásico de estos sistemas
- Los mejores resultados se han obtenido mediante las interfaces DR-FOH
- Si además se realiza una implementación asíncrona del sistema de control multifrecuencia, el comportamiento del sistema mejora mucho más que con la multifrecuencia síncrona.

En el siguiente apartado se va a estudiar el comportamiento del sistema cuando extrapolamos lo visto en estos apartados al control Image-Based Visual Servoing.

4.3.-Sistema Image-Based Visual Servoing Multifrecuencia

En este apartado se va a estudiar el comportamiento del sistema cuando se le incorpora el enfoque multifrecuencia. En primer lugar se podrán ver los experimentos realizados en simulación y las conclusiones que se obtienen a partir de ellos. Seguidamente lo que se hará será mostrar los resultados reales, realizando la comparativa con los obtenidos con el enfoque clásico y permitiendo obtener distintas conclusiones acerca del estudio realizado en la presente Tesis de Máster.

4.3.1.-Análisis del comportamiento bajo la plataforma de simulación

Esta sección está dedicada al análisis teórico del comportamiento del sistema Visual Servoing Multifrecuencia. Para ello se ha utilizado el simulador visto en el Capítulo 3 incorporando las interfaces bi-frecuencia y realizando distintos experimentos para ver el comportamiento del mismo.

Se ha utilizado el objeto que se muestra en la Figura 164. Se trata de un cuadrado donde las características obtenidas son los cuatro vértices del mismo.

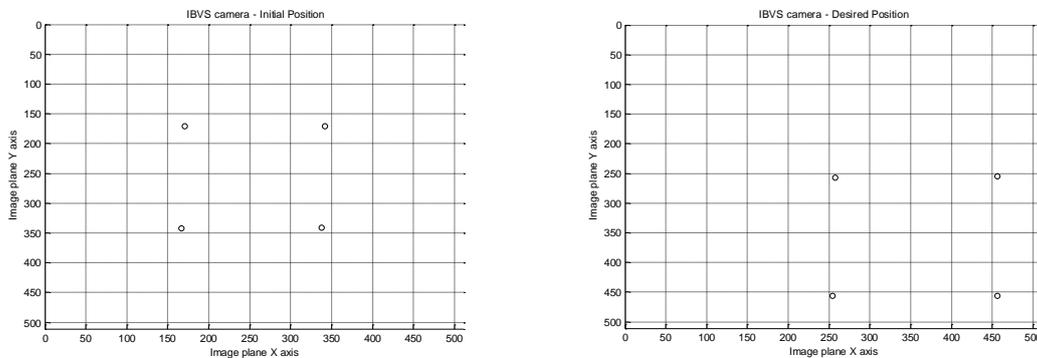


Figura 164. Objeto visto en el plano de imagen: pose actual del objeto (imagen de la izquierda), pose deseada del objeto (imagen de la derecha)

Como se está bajo la plataforma de simulación se va a proceder a realizar varios experimentos, algunos de ellos sin tener en cuenta las características físicas del sistema para ver el comportamiento alternando el orden de multiplicidad y colocando las interfaces multifrecuencia en distintos sitios.

En primer lugar se va a utilizar el esquema mostrado en la Figura 165 donde se tiene el sistema con los tres elementos Multifrecuenciales y por tanto harían falta tres interfaces para el tratamiento multifrecuencial del sistema.

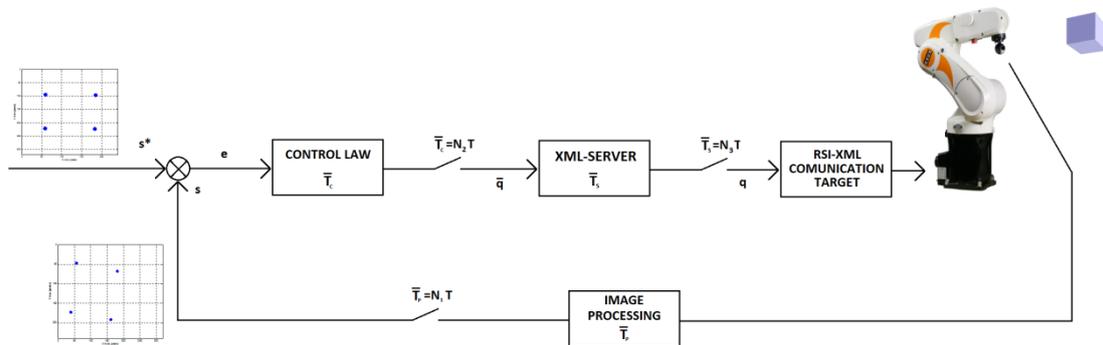


Figura 165. Esquema multifrecuencia del sistema Visual Servoing.

En primer lugar se va a comprobar que un tratamiento monofrecuencia de un sistema de estas características bajo unas condiciones de dinámica del objeto puede provocar comportamientos inesperados. El objeto se va a mover de forma periódica y con una frecuencia de 0.3 Hz. La Figura 166 muestra el comportamiento del sistema cuando es sometido a varias frecuencias. Es decir, cuando el tiempo de muestreo del sistema va incrementándose hasta que finalmente el sistema se vuelve inestable.

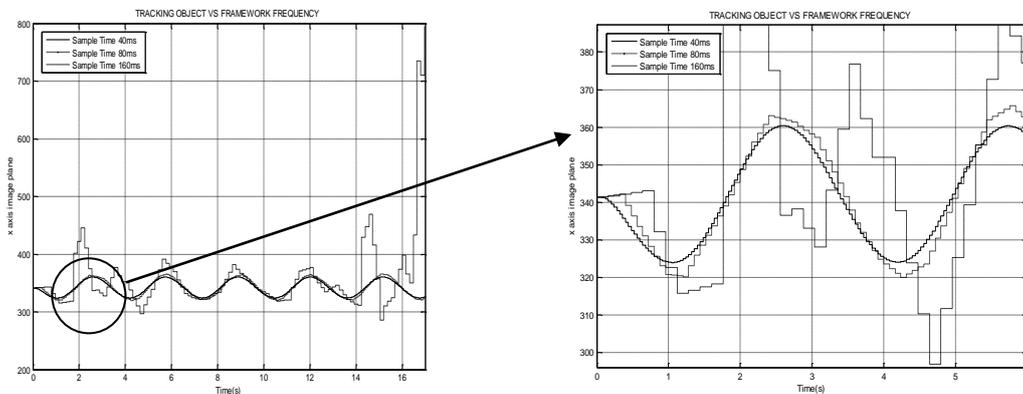


Figura 166. Comportamiento del sistema en función del metaperiodo del elemento más lento del sistema (la visión).

La Figura 166 demuestra que bajo determinadas condiciones de trabajo este sistema puede hacerse inestable. Y si esto ocurre en simulación, en la realidad será igual o peor.

Con esto se demuestra que es necesario el uso de un enfoque que contemple cada elemento como lo que es, un elemento que trabaja a una frecuencia determinada, es decir un enfoque multifrecuencia.

En el primer estudio que se va a realizar, se va a considerar como uno al sistema de control + servidor debido a que el tiempo de comunicación en simulación no existe por lo que el periodo base será de 10 milisegundos. Por otra parte tendremos el periodo del sistema de visión que se va a decir que es de 160 milisegundos. Para las pruebas de simulación se ha imaginado que se tiene un algoritmo muy robusto que provoca un tiempo de procesamiento muy grande que no permite al sistema trabajar a 33 fps. El esquema de control se puede observar en la Figura 167.

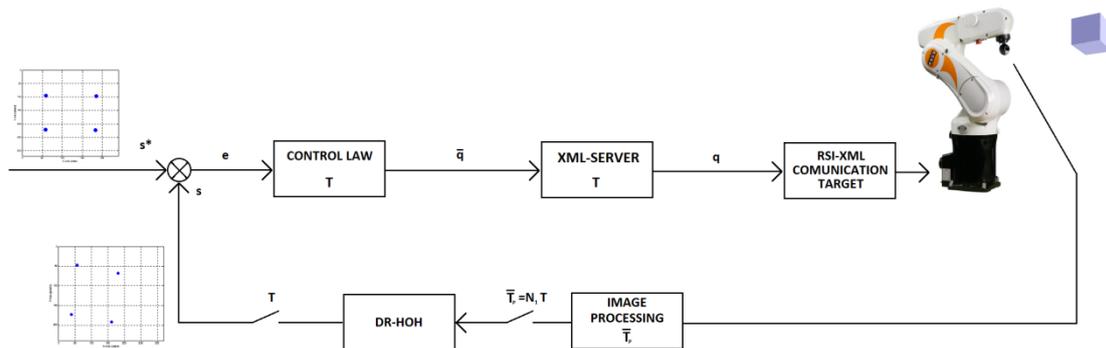


Figura 167. Esquema de control bi-frecuencia del sistema Visual Servoing.

Lo primero que se va a comprobar es el efecto que produce el cambio de la frecuencia de trabajo en este sistema. Los resultados obtenidos se pueden observar en la Figura 168.

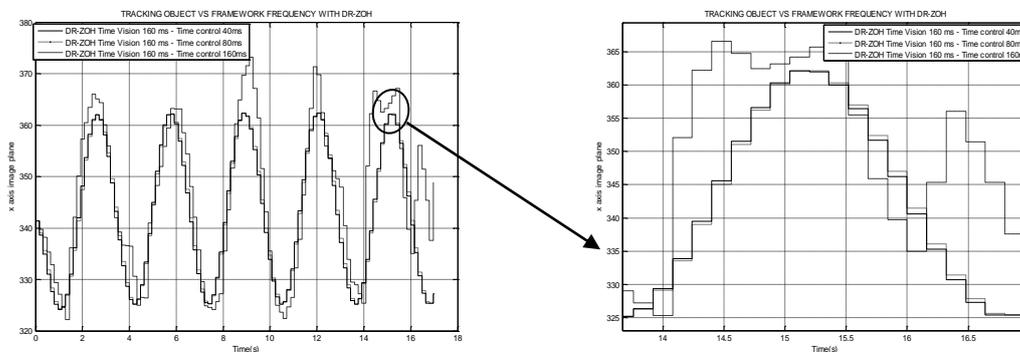


Figura 168. Seguimiento del objeto utilizando DR-ZOH. En las figuras se muestra el punto 3 en el plano de imagen.

El mismo estudio puede hacerse incorporando un DR-FOH. Como en el caso anterior se demuestra que a la frecuencia de 160 milisegundos el sistema se vuelve inestable, pero como resultado de la estimación a menores periodos base del sistema la estimación es mejor que en el caso del DR-ZOH.

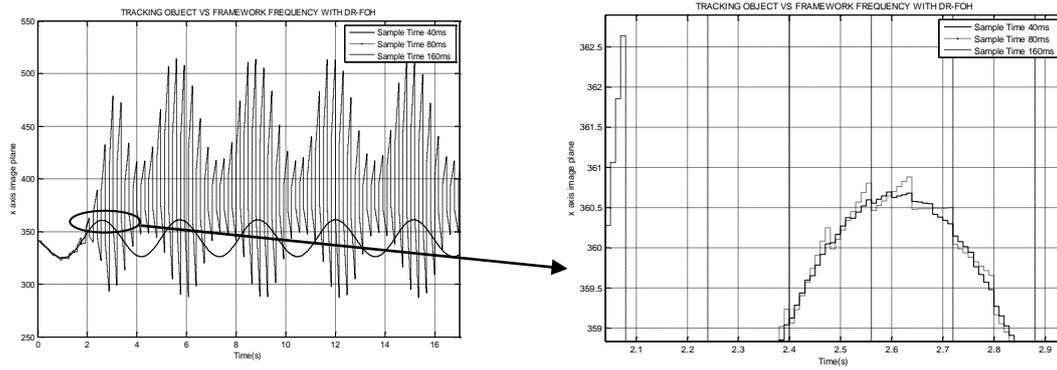


Figura 169. Comportamiento del sistema con un DR-FOH.

Con estos resultados se pone de manifiesto que un tratamiento multifrecuencia del sistema produce una mejoría en el comportamiento del sistema Visual Servoing.

El siguiente experimento a realizar es el de la introducción de dos interfaces bifrecuencia para que el sistema simulado se parezca más al sistema real. La primera interfaz será para estimar la señal de error y la segunda para estimar la señal de control. El esquema que muestra lo anterior se puede ver en la Figura 170.

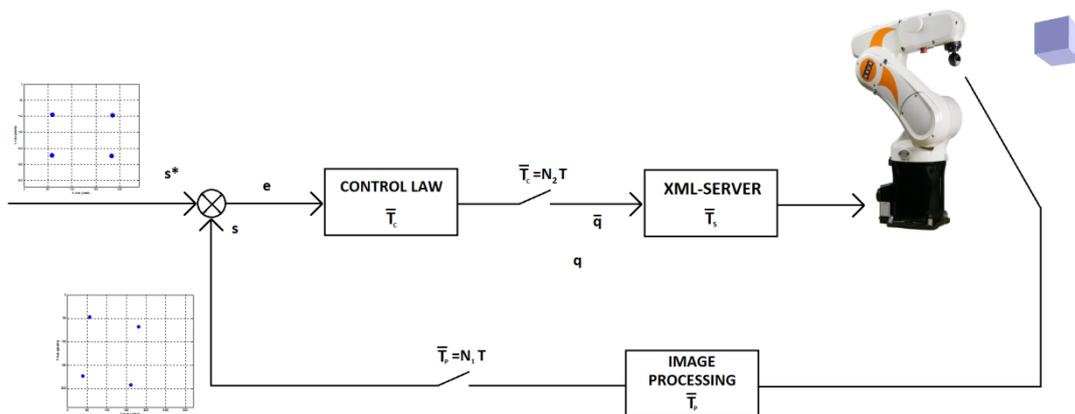


Figura 170. Esquema multifrecuencia del sistema Visual Servoing.

El experimento que se va a realizar es para observar cómo afecta, en simulación, los órdenes de multiplicidad N_1 y N_2 al comportamiento global del sistema. Para ello, se tienen que dejar de tener en cuenta las limitaciones físicas del sistema ya que por ejemplo no se pueden adquirir imágenes a periodos inferiores a 30 milisegundos. Hay también que tener en cuenta que la relación $N_1 \cdot N_2 \equiv cte$ ha de mantenerse constante en todos los experimentos realizados. En este caso, la constante va a tener un valor de 16. Este valor multiplicado por el periodo base va a ser igual al metaperiodo del elemento más lento del sistema (\bar{T}_p).

Este experimento trata de responder a preguntas del tipo: qué es mejor, ¿un controlador trabajando a baja frecuencia y una adquisición trabajando a frecuencia

alta o lo contrario, una adquisición trabajando a baja frecuencia y un sistema de control trabajando a alta frecuencia? Este tipo de cuestiones dependen de cada sistema en particular por lo que su respuesta no es trivial, aunque en un número elevado de casos, esto viene determinado por limitaciones físicas, por lo que la elección no existe.

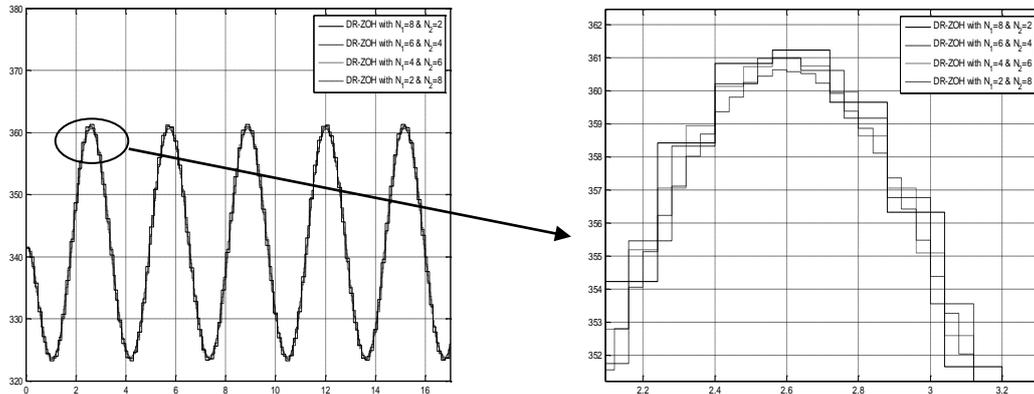


Figura 171. Seguimiento del un objeto cambiando los valores de N_1 and N_2 utilizando DR-ZOH. En las figuras se muestra el eje X del punto 3 en el plano de imagen.

La Figura 171 muestra la trayectoria de la componente x del punto 3 del objeto en el plano de imagen. Como se puede comprobar, no hay mucha diferencia en la trayectoria cuando se cambian los valores de N_1 y N_2 manteniendo constante la relación de ellos. Este resultado demuestra que se tiene el mismo comportamiento en ambos casos.

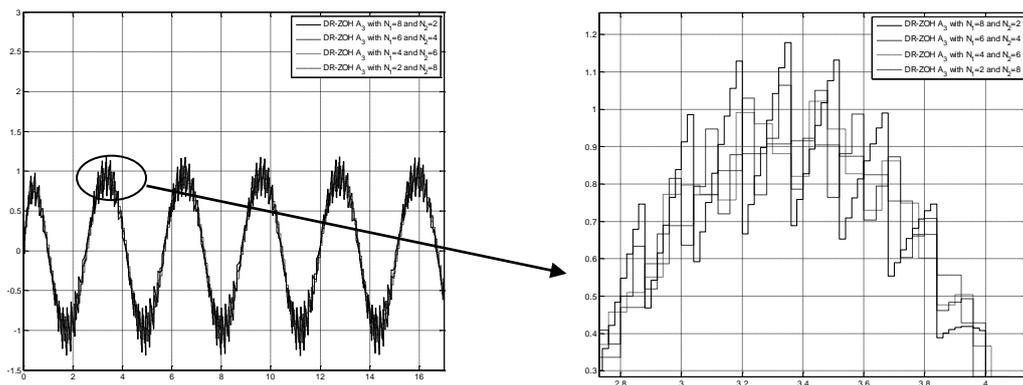


Figura 172. Acciones de control del eje A3 cambiando los valores de N_1 and N_2 utilizando DR-ZOH.

El mismo experimento se ha realizado pero esta vez utilizando DR-FOH. En este caso sí que se puede observar una mayor diferencia de las trayectorias del punto. Esto es debido sobre todo al comportamiento del DR-FOH cuando hay cambios bruscos, tal y como se muestra en la Figura 173. Mientras ese sector no tienen cambios la

estimación es mejor que en el caso anterior pero en los picos de la senoidal es donde más se refleja la diferencia.

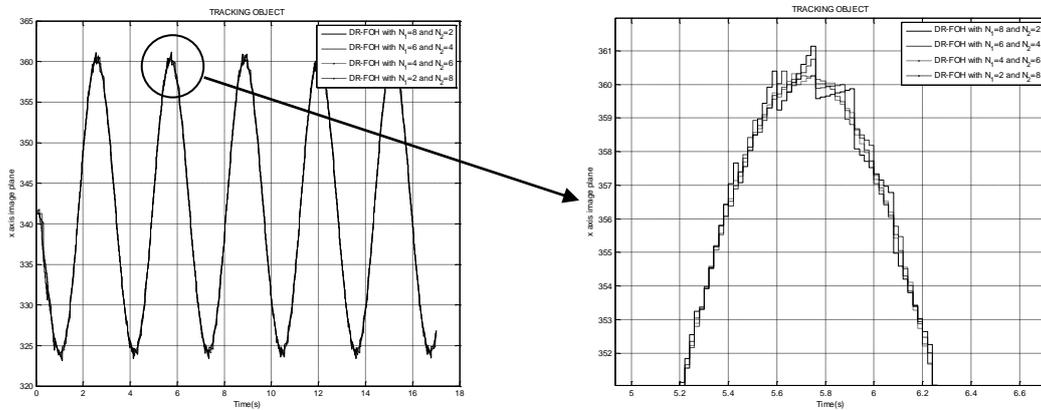


Figura 173. Seguimiento de un objeto cambiando los valores de N_1 and N_2 utilizando DR-FOH. En las figuras se muestra el eje X del punto 3 en el plano de imagen.

Aún así y observando las acciones del control de la Figura 174, se demuestra que son más suaves lo que conlleva a una mejora en el comportamiento global del sistema.

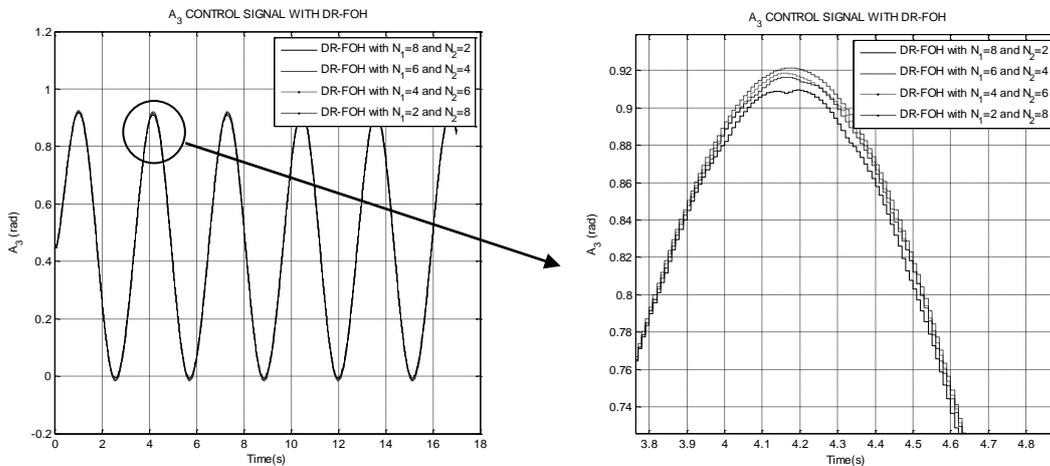


Figura 174. Acciones de control del eje A3 cambiando los valores de N_1 and N_2 utilizando DR-FOH.

Una vez comparados ambos enfoques (monofrecuencia y multifrecuencia) y vistos los beneficios y los no beneficios de cada uno de ellos, se concluye que el uso de la multifrecuencia para esta clase de sistemas provoca una clara mejoría en el comportamiento del mismo, por lo que es lo más adecuado.

Ahora, los siguientes experimentos van encaminados a determinar el verdadero comportamiento del sistema multifrecuencia teniendo en cuenta las limitaciones físicas del sistema. Además se va a estudiar su comportamiento en cada uno de los DR-HOH que se han visto anteriormente.

- Resultados con DR-ZOH

Las Figuras 175 y 176 muestran las trayectorias en el plano de imagen de las características del objeto.

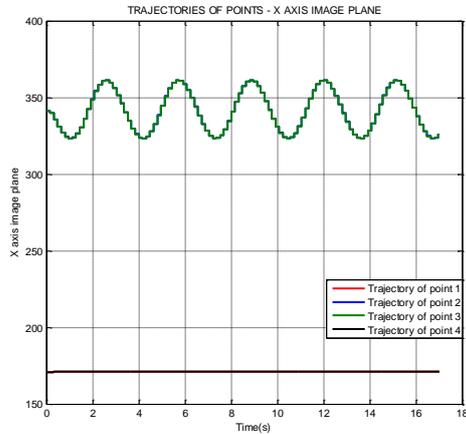


Figura 175. Trayectoria en el eje X de los puntos en el plano de imagen a lo largo del tiempo: P1 rojo, P2 azul, P3 verde y P4 negro.

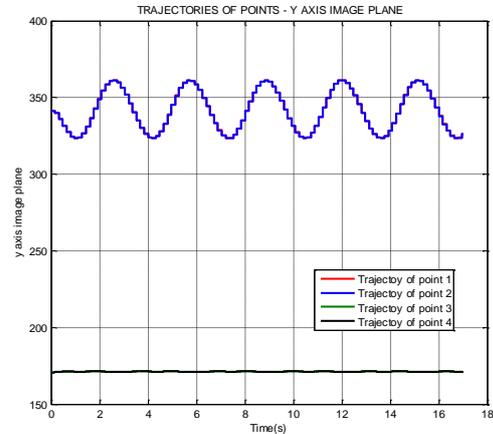


Figura 176. Trayectoria en el eje Y de los puntos en el plano de imagen a lo largo del tiempo: P1 rojo, P2 azul, P3 verde y P4 negro.

Para ver mejor el comportamiento de la interfaz DR-ZOH la siguiente figura muestra un zoom de una de las trayectorias. Como se puede ver, es este caso se actualiza cada 60 milisegundos, mientras que las estimaciones son cada 20 milisegundos.

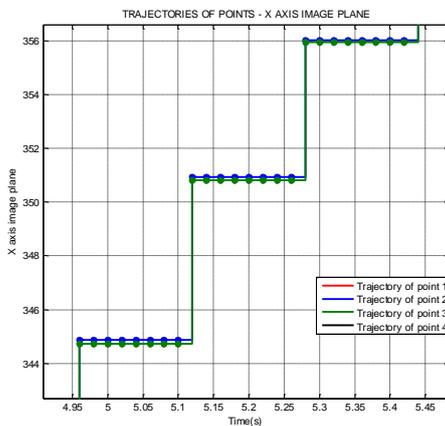


Figura 177. Zoom de las trayectorias en el eje X de los puntos 2 y 3.

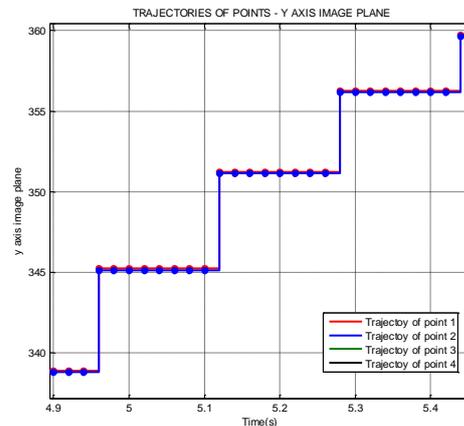


Figura 178. Zoom de las trayectorias en el eje Y de los puntos 2 y 3.

Las siguientes figuras muestran el comportamiento de las acciones de control en cada una de las articulaciones del robot. Se puede comprobar que cada 10 milisegundos se estima y envía una acción de control.

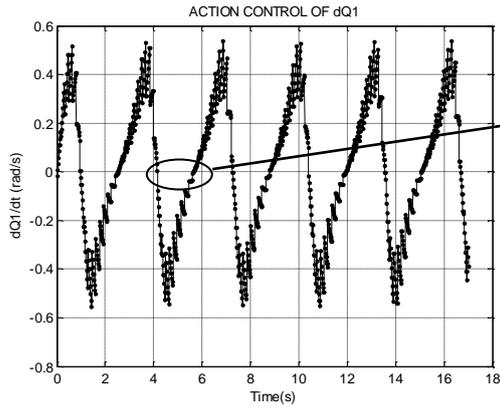


Figura 179. Acción de control para la articulación 1.

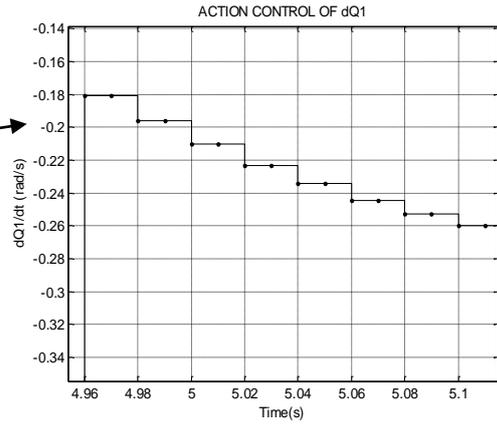


Figura 180. Zoom de la acción de control para la articulación 1.

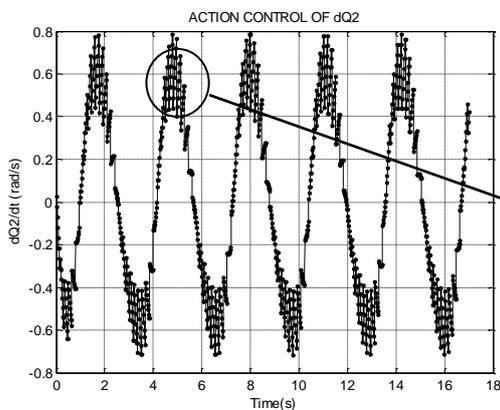


Figura 181. Acción de control para la articulación 2.

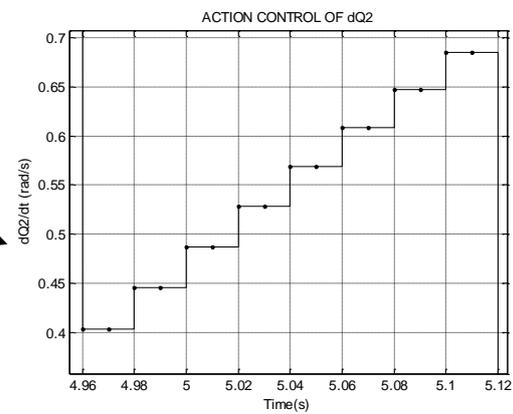


Figura 182. Zoom de la acción de control para la articulación 2.

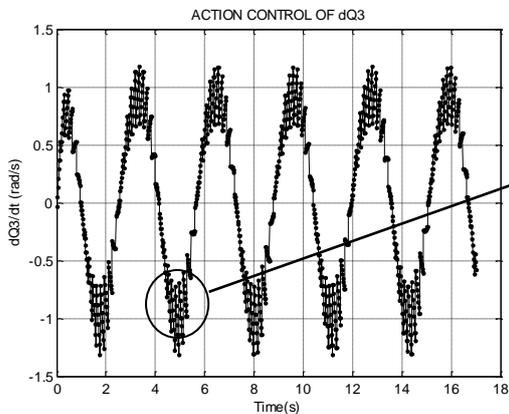


Figura 183. Acción de control para la articulación 3.

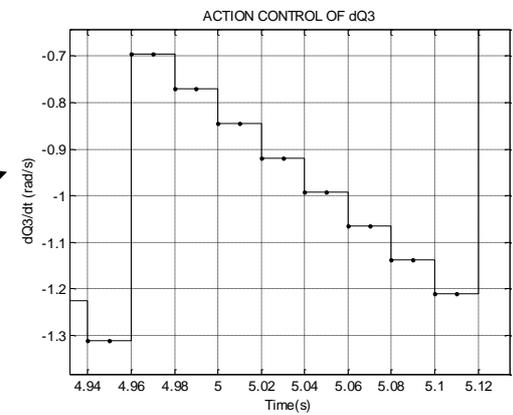


Figura 184. Zoom de la acción de control para la articulación 3.

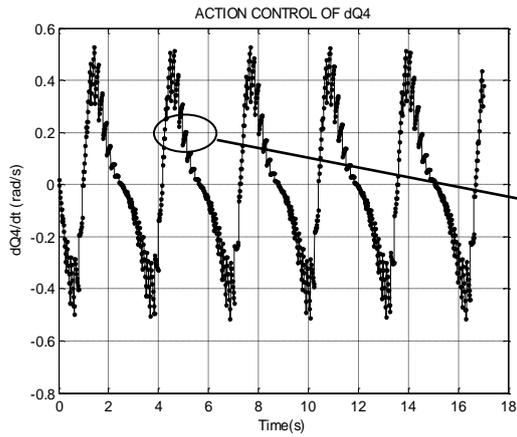


Figura 185. Acción de control para la articulación 4.

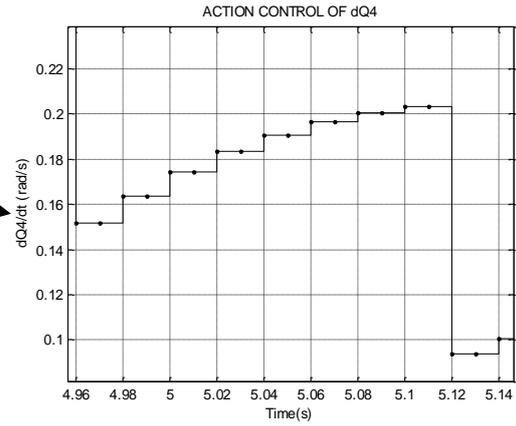


Figura 186. Zoom de la acción de control para la articulación 4.

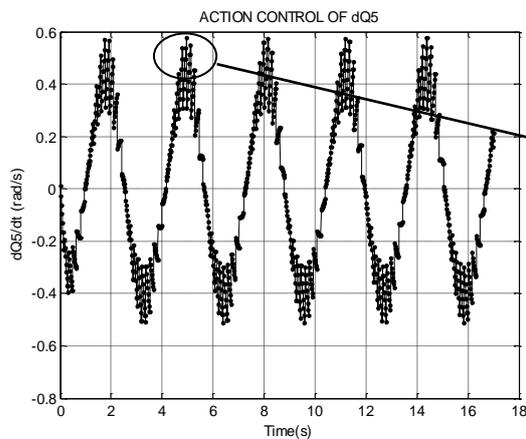


Figura 187. Acción de control para la articulación 5.

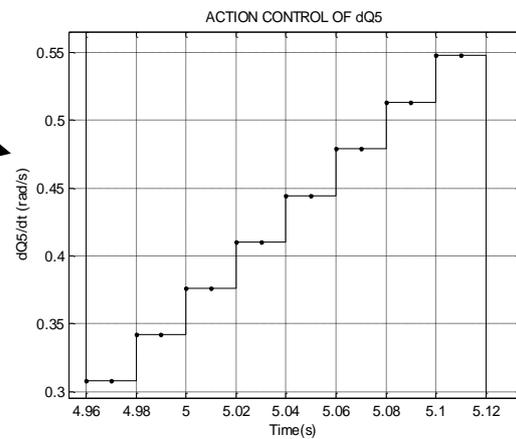


Figura 188. Zoom de la acción de control para la articulación 5.

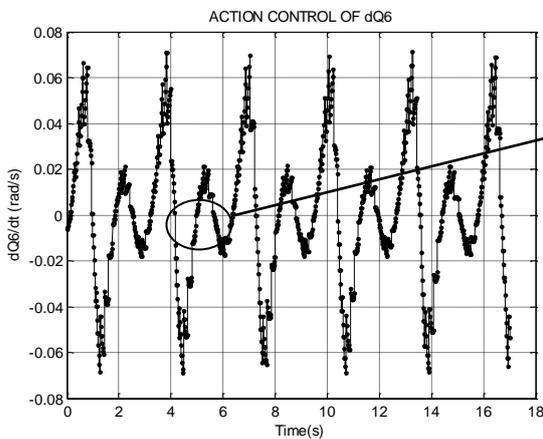


Figura 189. Acción de control para la articulación 6.

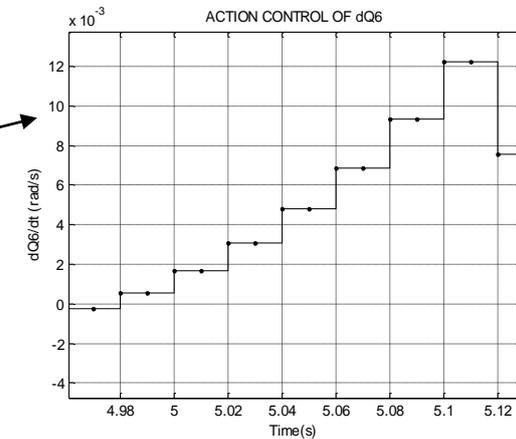


Figura 190. Zoom de la acción de control para la articulación 6.

- Resultados con DR-FOH

Del mismo modo se va a analizar el sistema pero ahora con la utilización de interfaces DR-FOH. Los resultados se pueden ver en las figuras siguientes:

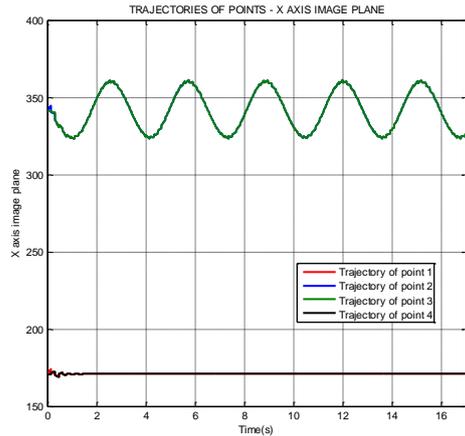


Figura 191. Trayectoria en el eje X de los puntos en el plano de imagen a lo largo del tiempo: P1 rojo, P2 azul, P3 verde y P4 negro.

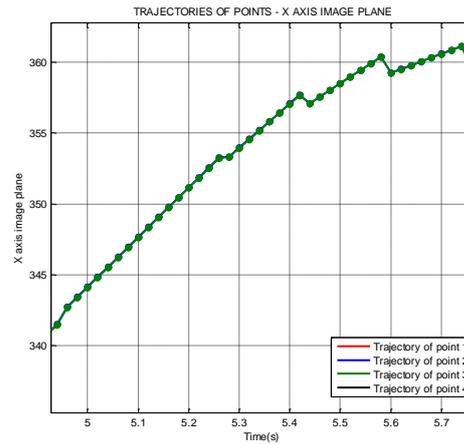


Figura 192. Zoom de la trayectoria en el eje X de los puntos en el plano de imagen a lo largo del tiempo: P1 rojo, P2 azul, P3 verde y P4 negro.

Por último, las siguientes figuras muestran las acciones de control a lo largo de la simulación.

Para finalizar decir que, tal y como se ha observado en el apartado anterior con el ejemplo de dos grados de libertad, en simulación también se demuestra que el comportamiento del sistema con un retenedor bi-frecuencia de primer orden es mejor que con uno de orden cero y mucho mejor que con el enfoque monofrecuencia clásico.

Con este propósito se va a realizar una comparativa entre los distintos enfoques tratados y así finalizaremos el estudio en simulación de la incorporación del enfoque multifrecuencia en el clásico Image-Based Visual Servoing.

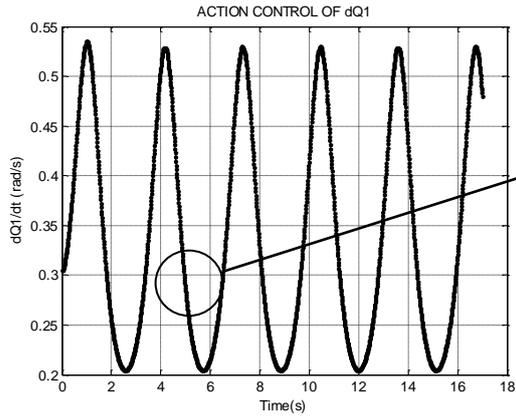


Figura 193. Acción de control para la articulación 1.

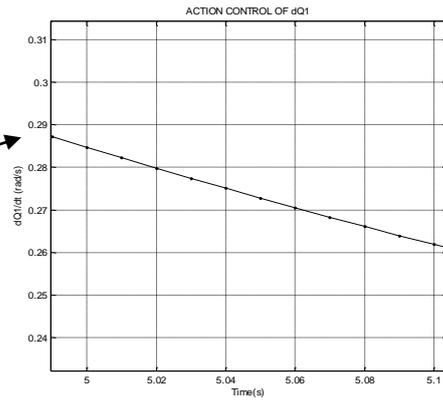


Figura 194. Zoom de la acción de control para la articulación 1.

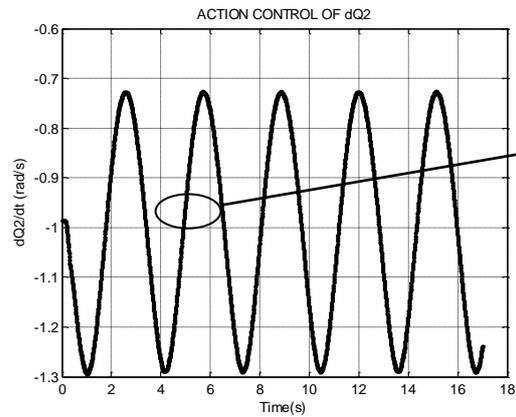


Figura 195. Acción de control para la articulación 2.

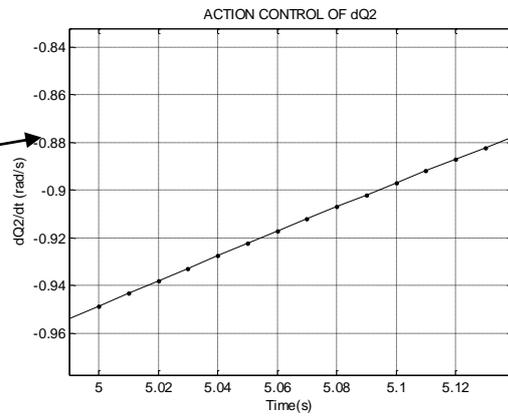


Figura 196. Zoom de la acción de control para la articulación 2.

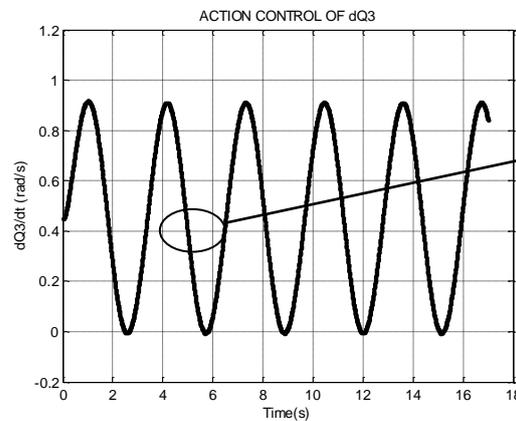


Figura 197. Acción de control para la articulación 3.

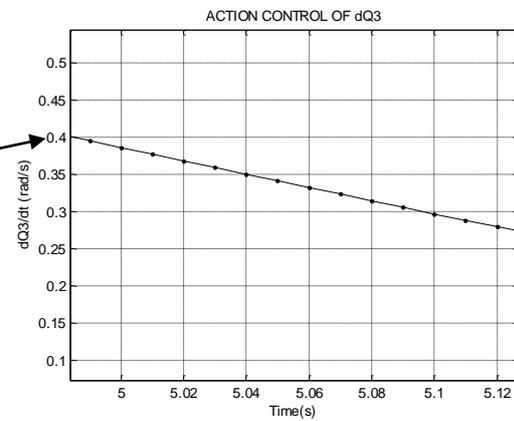


Figura 198. Zoom de la acción de control para la articulación 3.

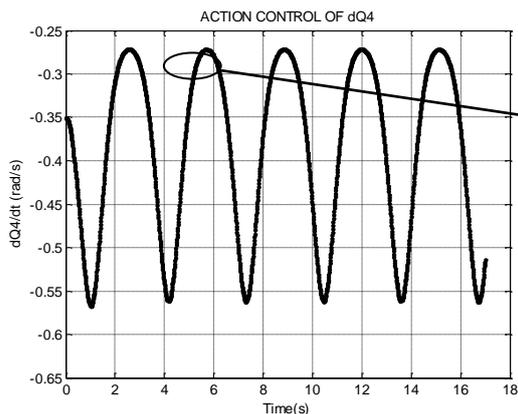


Figura 199. Acción de control para la articulación 4.

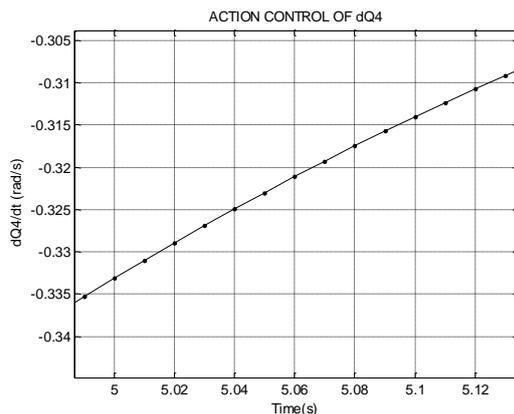


Figura 200. Zoom de la acción de control para la articulación 4.

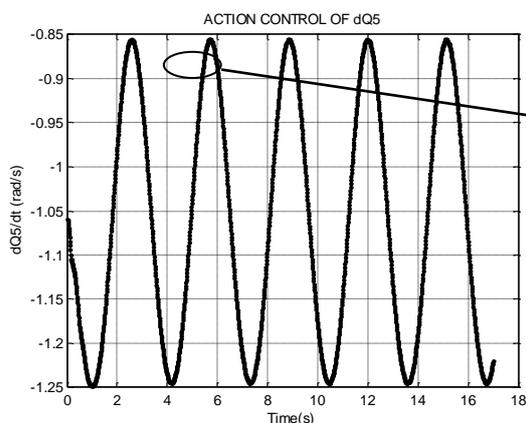


Figura 201. Acción de control para la articulación 5.

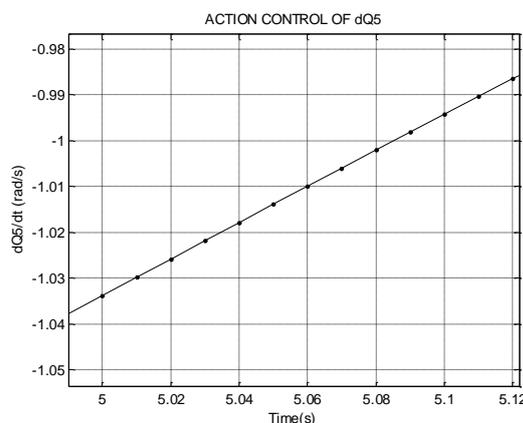


Figura 202. Zoom de la acción de control para la articulación 5.

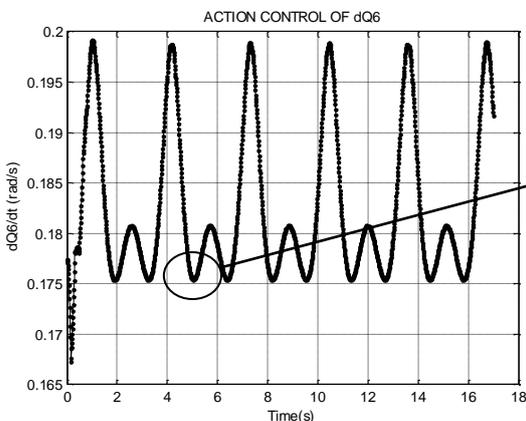


Figura 203. Acción de control para la articulación 6.

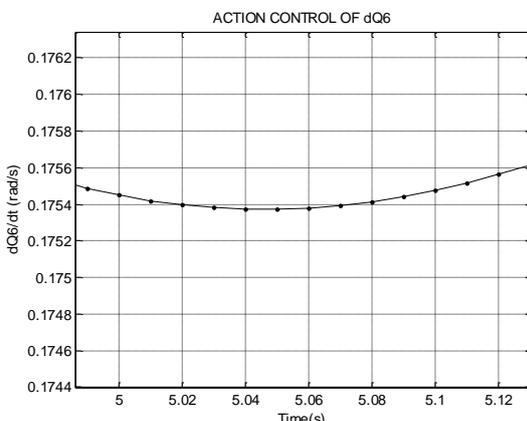


Figura 204. Zoom de la acción de control para la articulación 6.

4.3.1.1.- Comparativa entre los enfoques

En este apartado se va a mostrar la comparativa realizada con todos los enfoques utilizados. Esta comparativa permitirá una mejor visualización del comportamiento del sistema bajo las mismas condiciones en cada uno de los enfoques empleados. Se puede observar que el enfoque clásico monofrecuencia a baja frecuencia se hace inestable.

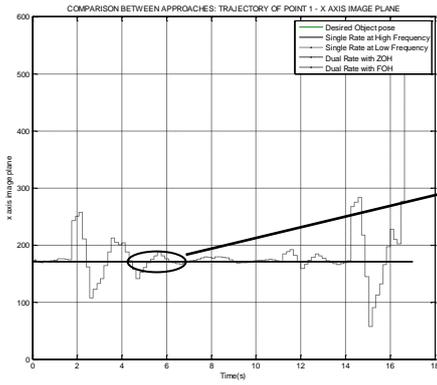


Figura 205. Trayectoria del punto 1 en el eje X del plano de imagen

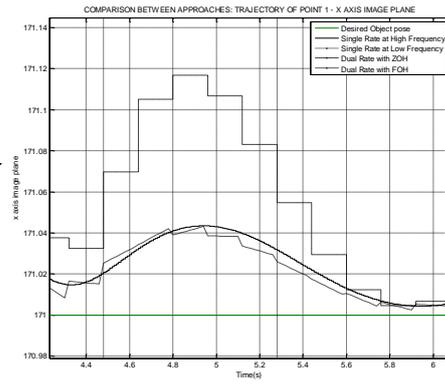


Figura 206. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen

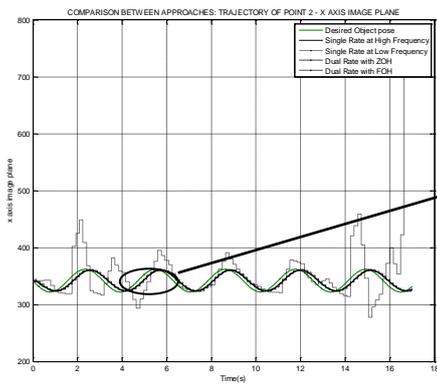


Figura 207. Trayectoria del punto 2 en el eje X del plano de imagen

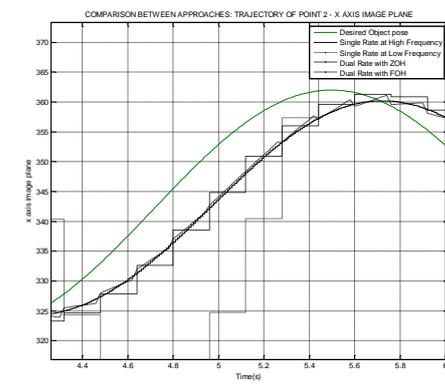


Figura 208. Zoom de la trayectoria del punto 2 en el eje X del plano de imagen

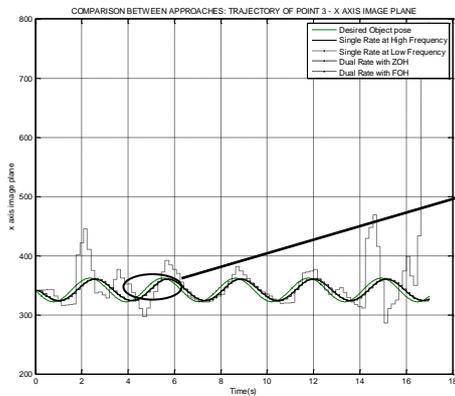


Figura 209. Trayectoria del punto 3 en el eje X del plano de imagen

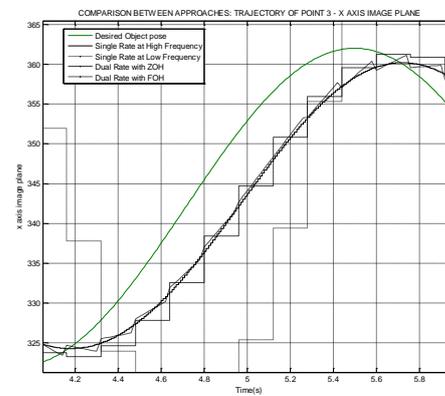


Figura 210. Zoom de la trayectoria del punto 3 en el eje X del plano de imagen

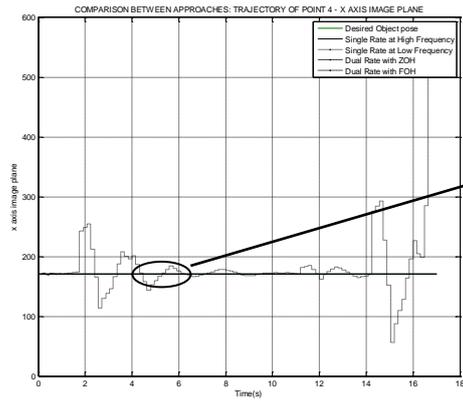


Figura 211. Trayectoria del punto 4 en el eje X del plano de imagen

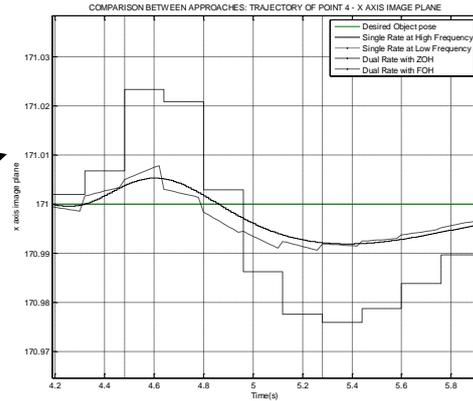


Figura 212. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen

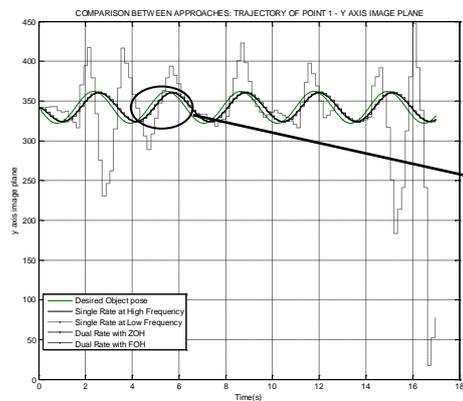


Figura 213. Trayectoria del punto 1 en el eje Y del plano de imagen

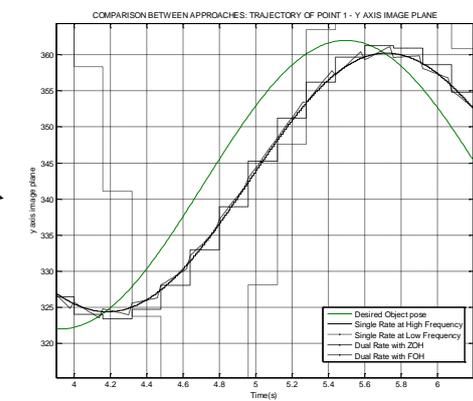


Figura 214. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen

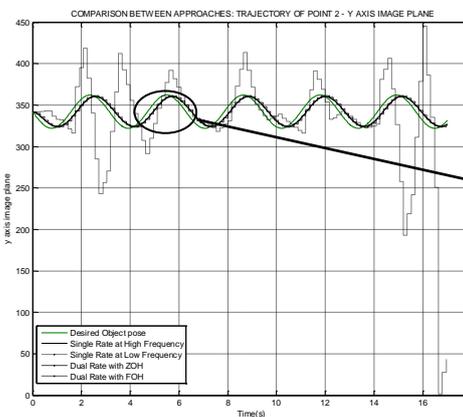


Figura 215. Trayectoria del punto 2 en el eje Y del plano de imagen

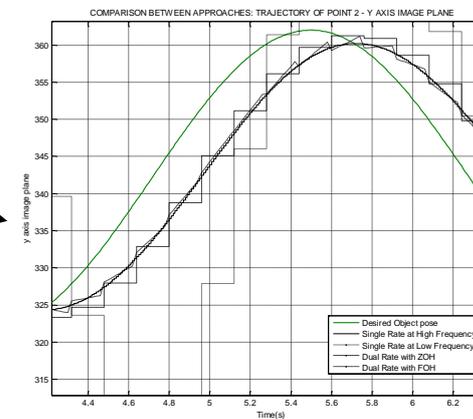


Figura 216. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen

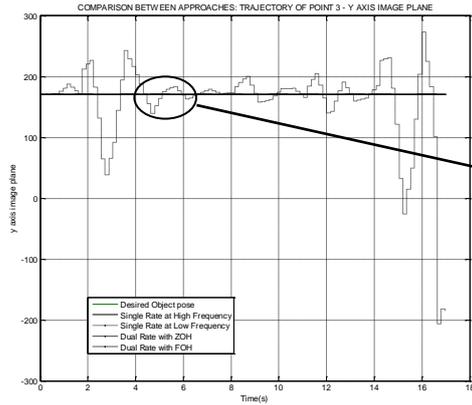


Figura 217. Trayectoria del punto 3 en el eje Y del plano de imagen

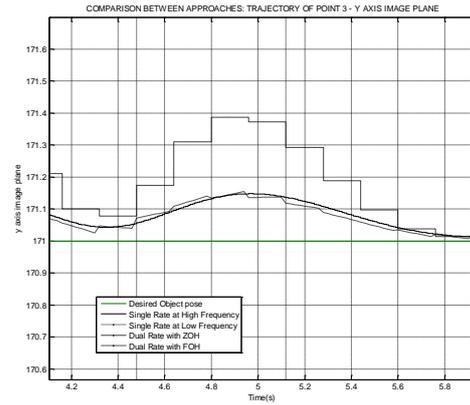


Figura 218. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen

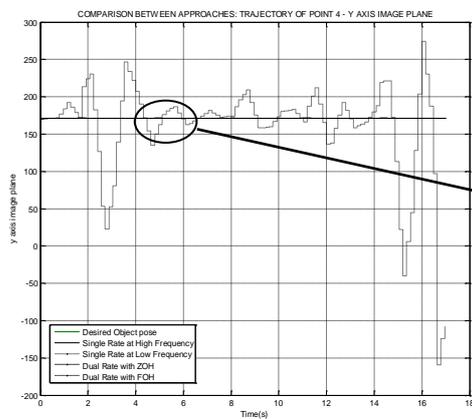


Figura 219. Trayectoria del punto 4 en el eje Y del plano de imagen

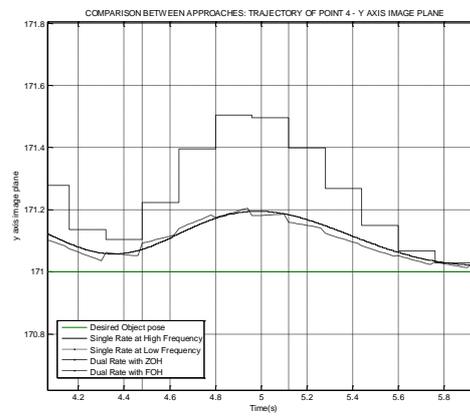


Figura 220. Zoom de la trayectoria del punto 1 en el eje X del plano de imagen

También se ve que ambos interfaces DR-HOH hacen que el sistema tenga un comportamiento estable y similar al que tendría si éste estuviera trabajando a alta frecuencia.

Por último las siguientes figuras muestran las acciones de control que se han obtenido en cada uno de los casos. Notar que las acciones que más parecido tienen con el sistema trabajando a alta frecuencia son cuando se utiliza el enfoque multifrecuencia con las interfaces DR-FOH. Este resultado corrobora la conclusión a la que se había llegado en el estudio del sistema con dos grados de libertad, y es que el sistema con un DR-FOH tiene el comportamiento más parecido al sistema monofrecuencia trabajando a frecuencia alta.

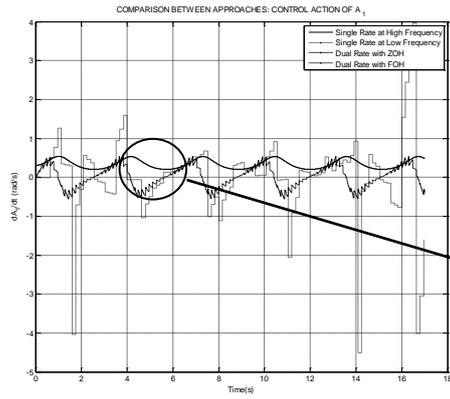


Figura 221. Acción de control para la articulación 1.

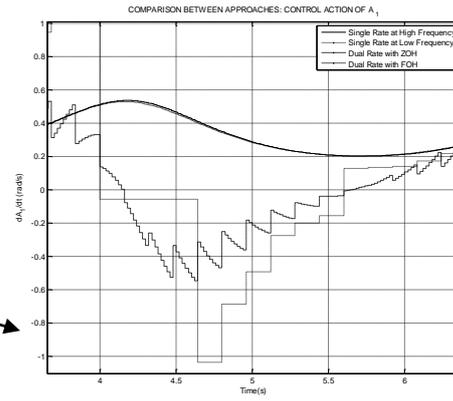


Figura 222. Zoom de la acción de control para la articulación 1.

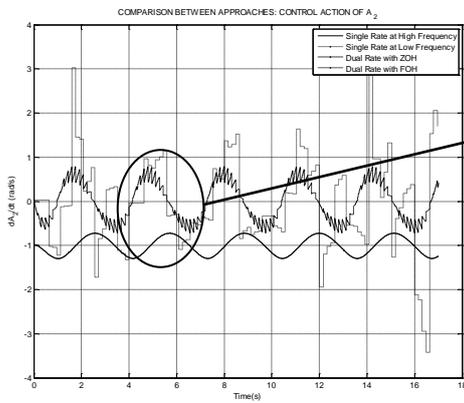


Figura 223. Acción de control para la articulación 2.

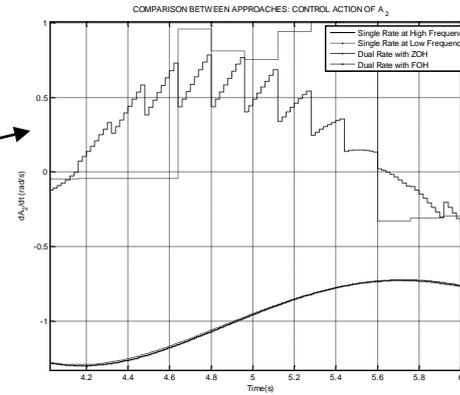


Figura 224. Zoom de la acción de control para la articulación 2.

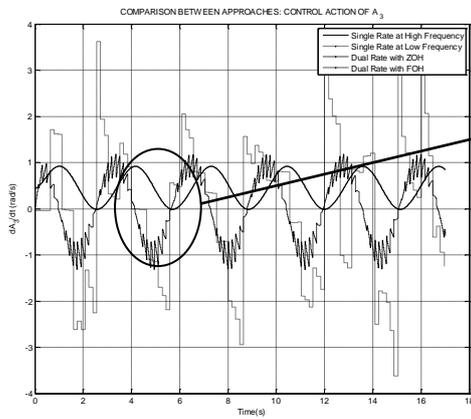


Figura 225. Acción de control para la articulación 3.

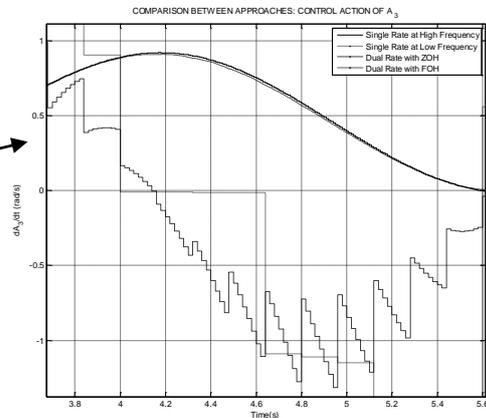


Figura 226. Zoom de la acción de control para la articulación 3.

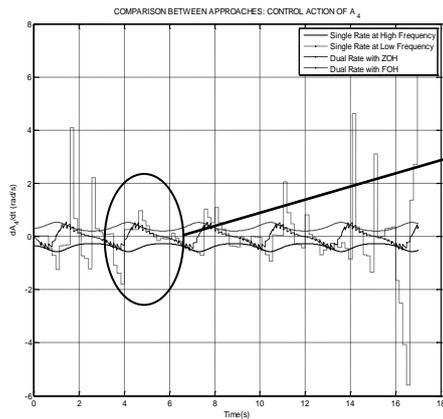


Figura 227. Acción de control para la articulación 4.

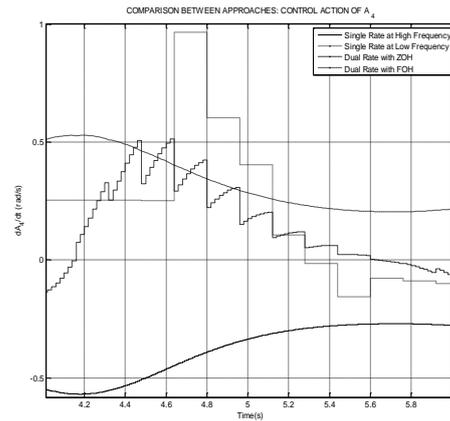


Figura 228. Zoom de la acción de control para la articulación 4.

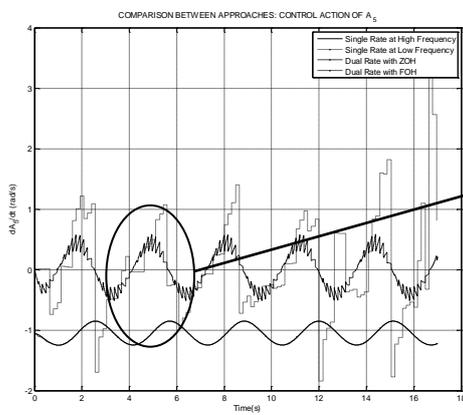


Figura 229. Acción de control para la articulación 5.

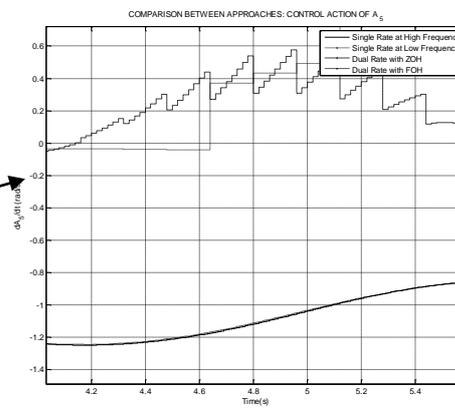


Figura 230. Zoom de la acción de control para la articulación 5.

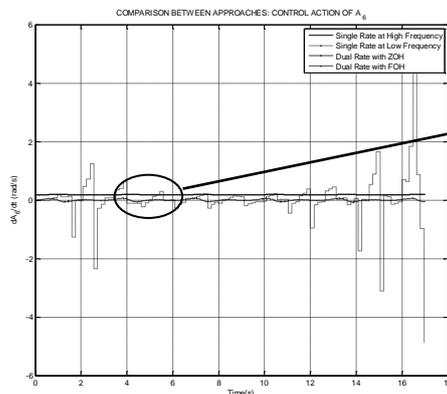


Figura 231. Acción de control para la articulación 6.

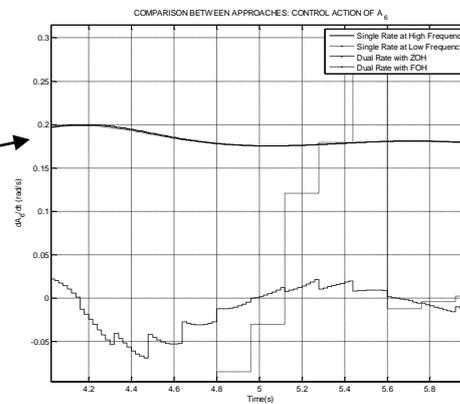


Figura 232. Zoom de la acción de control para la articulación 6.

En el siguiente apartado se va a mostrar los resultados obtenidos en el proceso real y se va a realizar una comparativa más exhaustiva para ver la mejora real que un enfoque multifrecuencia con las interfaces bi-frecuencia introduce al enfoque clásico del Image-Based Visual Servoing.

4.3.2.-Análisis del comportamiento bajo la plataforma real

Tal y como se hizo en el Capítulo 3, se va a realizar el estudio del sistema Visual Servoing Multifrecuencia utilizando las cuatro trayectorias (triángulo, cuadrado, círculo y intimito) modificando la velocidad del objeto en cada experimento. De esta forma, una vez obtenidos los resultados se procederá a realizar una comparativa con los resultados obtenidos en el enfoque IBVS clásico.

Del mismo modo que se hizo en el Capítulo 3, para no extender demasiado el presente documento, se va a mostrar los resultados de forma detallada para el caso de la trayectoria triángulo, por lo que con las demás trayectorias sólo se mostrarán los resultados obtenidos en forma de diagrama de barras.

- TRIÁNGULO

En las gráficas de la Figura 233 se puede observar las señales del error de las características cuando el objeto tiene 30 segundos para efectuar el recorrido de la trayectoria triángulo con la interfaz bifrecuencia DR-ZOH. Como se puede observar, la transición en los cambios de la trayectoria del objeto es muy rápida. Esto tiene el beneficio de que se puede seguir el objeto más cerca (es decir con menor error) aunque tiene la contrapartida de que se introduce una sobreoscilación en la dinámica del sistema.

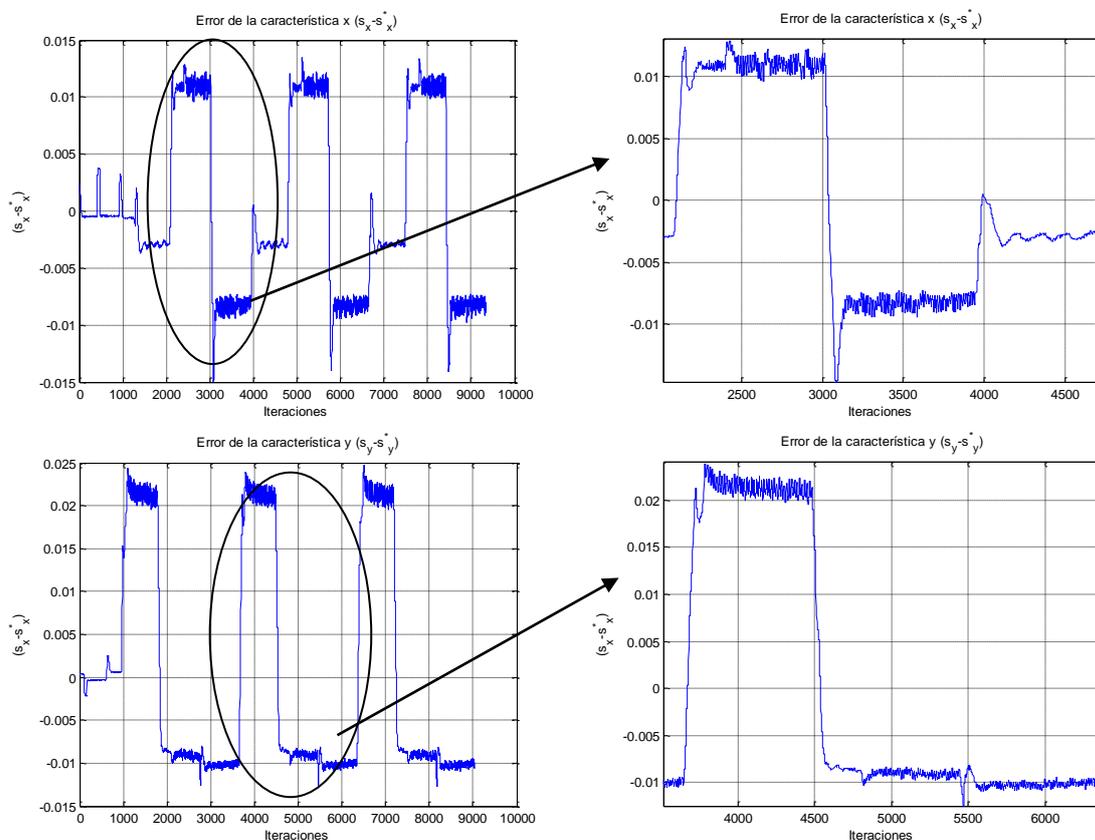


Figura 233. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 30 segundos y con DR-ZOH.

La trayectoria del sistema se puede ver más claramente en la gráfica de la Figura 234. Se ve como se está realizando una trayectoria triángulo. Además, se puede observar que en los cambios de sentido del objeto (en los vértices del triángulo) existe una pequeña variación debida a la oscilación introducida por el enfoque Multifrecuencia.

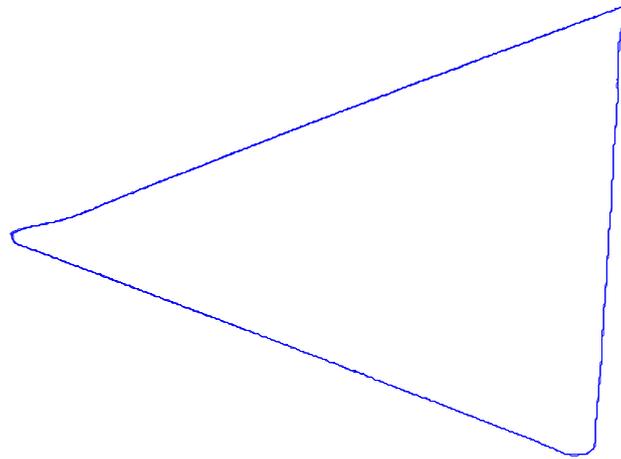


Figura 234. Trayectoria efectuada por el Efector final del Kuka KR5 sixx R650 con DR-ZOH y trayectoria triángulo con periodo 30 segundos.

Por último, las acciones de control se pueden observar en las gráficas de la Figura 235. En ellas se demuestra que se están utilizando todos los grados de libertad del robot.

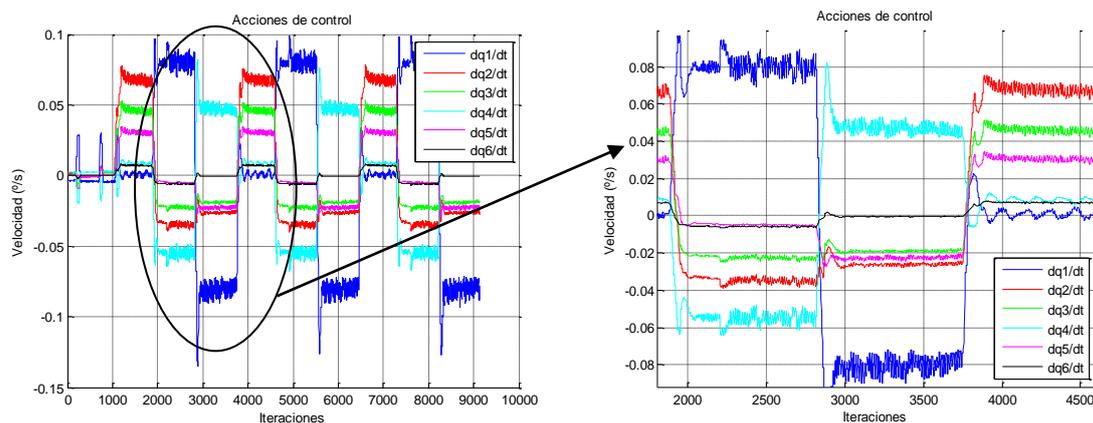


Figura 235. Acciones de control para la trayectoria triángulo de periodo 30 segundos con DR-ZOH.

Una vez vistos los resultados del experimento con periodo 30 segundos cuando la trayectoria es triangular y con DR-ZOH, las gráficas de la Figura 236 muestran los resultados obtenidos para cuando el periodo es de 10 segundos. En este caso, el objeto se mueve con mayor velocidad por lo que el error cometido ha aumentado.

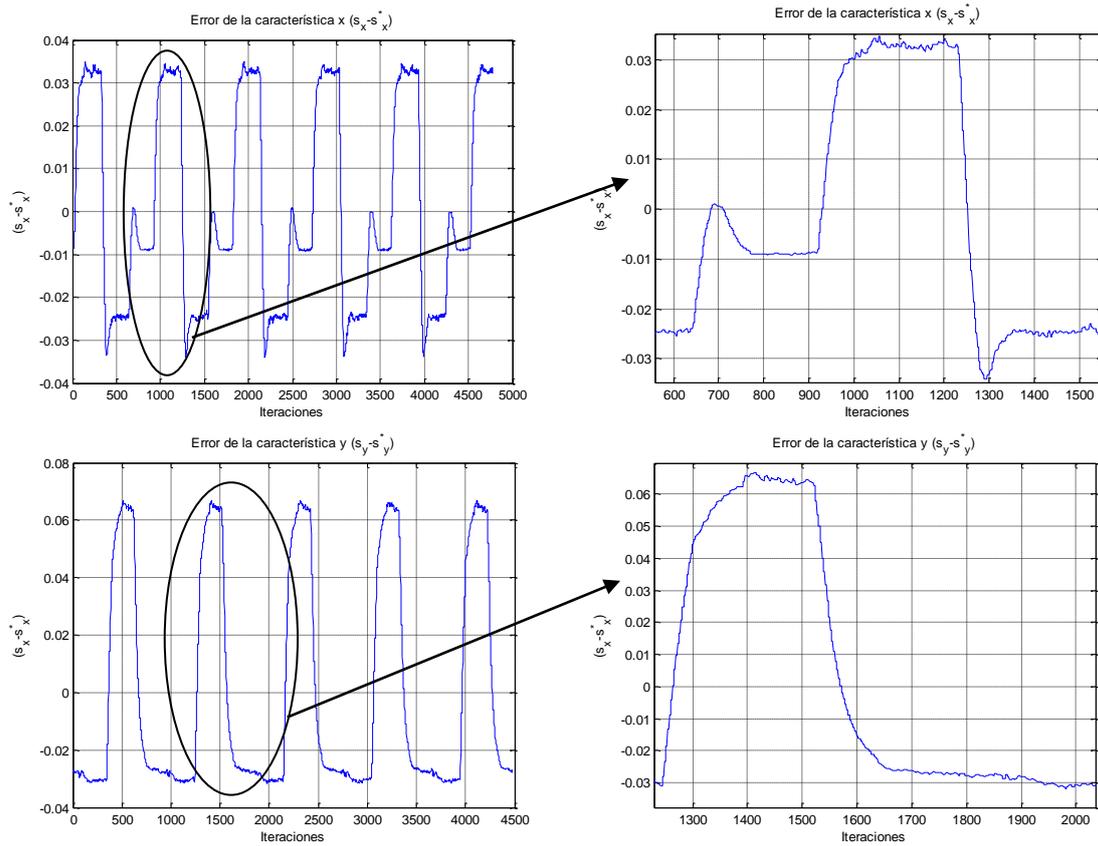


Figura 236. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 10 segundos y con DR-ZOH.

Para visualizar mejor lo que está pasando realmente, la trayectoria del efector final del robot se muestra en la Figura 237. Se puede comprobar que los vértices están un poco más redondeados que en el caso anterior y esto es porque el efector final no sigue tan de cerca al objeto.

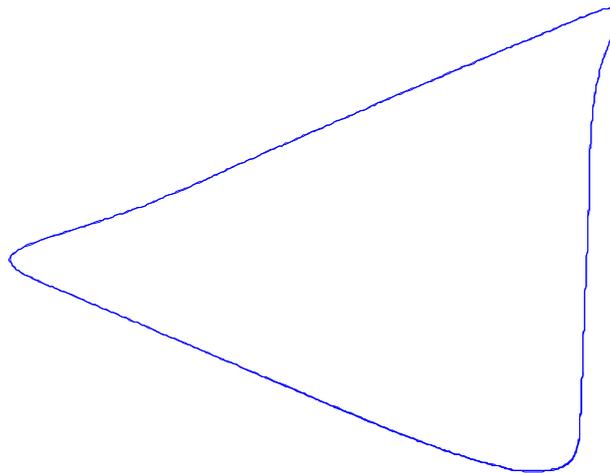


Figura 237. Trayectoria efectuada por el Efector final del Kuka KR5 sixx R650 con DR-ZOH y trayectoria triángulo con periodo 10 segundos.

Del mismo modo, las gráficas de la Figura 238 muestran las acciones de control para cada una de las articulaciones del brazo robot. Se puede comprobar que en este caso hay un mayor consumo que es el caso anterior.

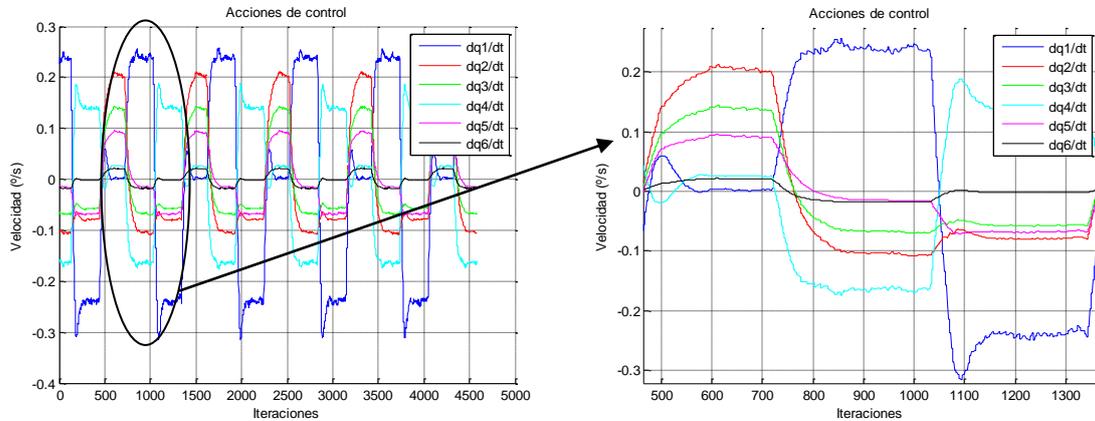


Figura 238. Acciones de control para la trayectoria triángulo de periodo 10 segundos con DR-ZOH.

Por último se va a incrementar aún más la velocidad del objeto con el experimento de periodo 5 segundos. Los resultados de la señal de error se muestran en las gráficas de la Figura 239.

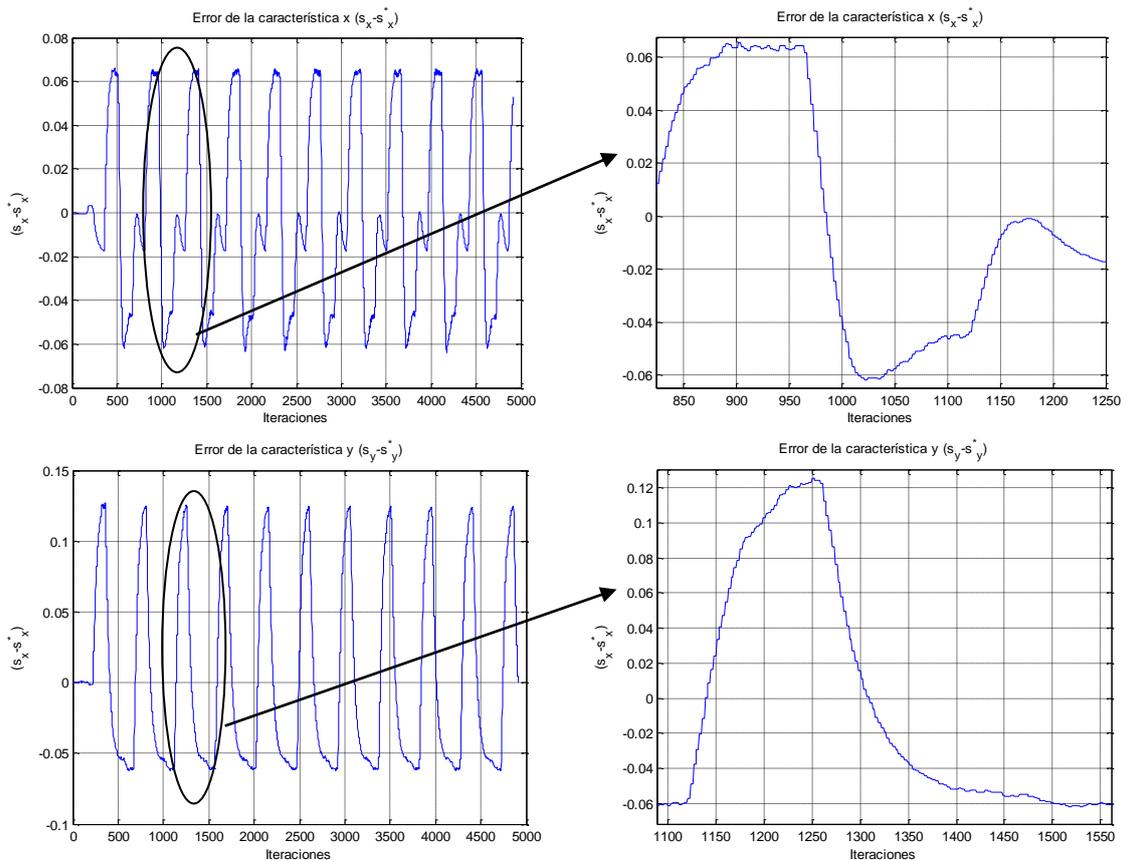


Figura 239. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 5 segundos y con DR-ZOH.

La trayectoria que ha seguido el efector final en este caso ha sido la que se muestra en la Figura 240. Se puede observar que, aunque la trayectoria se parece a un

triángulo, hay algunas zonas en las que los vértices están muy redondeados, síntoma de que el efector final comete más error con el objeto.

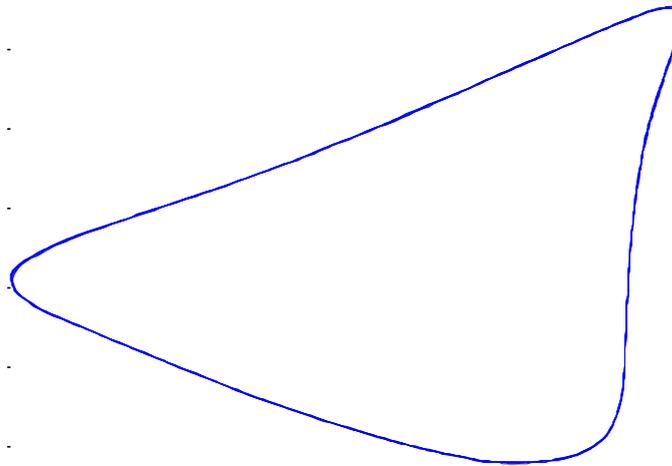


Figura 240. Trayectoria efectuada por el Efector final del Kuka KR5 sixx R650 con DR-ZOH y trayectoria triángulo con periodo 5 segundos.

Del mismo modo, la Figura 241 muestra las gráficas de las acciones de control en cada articulación del robot. En este caso se puede comprobar que las acciones son un poco más bruscas que en los anteriores casos, lo que significa que hay un mayor consumo de energía.

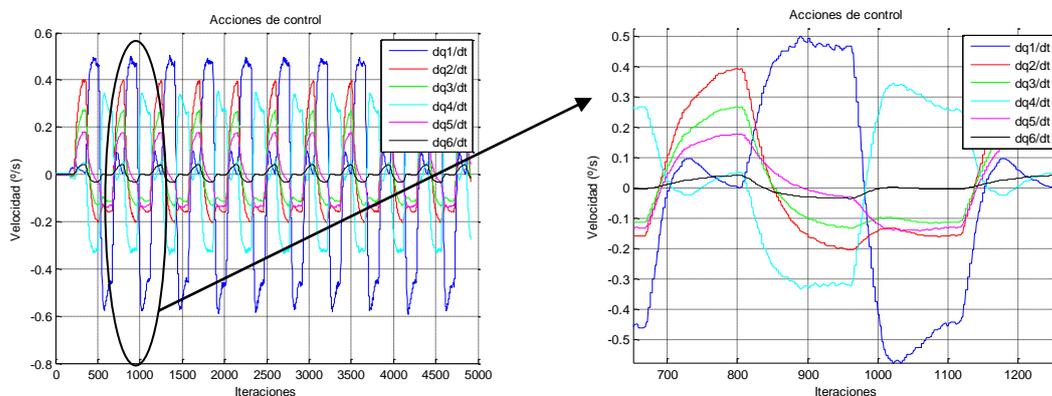


Figura 241. Acciones de control para la trayectoria triángulo de periodo 5 segundos con DR-ZOH.

La Figura 242 muestra la comparación entre las tres trayectorias obtenidas con el IBVS Multifrecuencia utilizando la interfaz DR-ZOH. En ella podemos ver que, como es natural, a medida que el objeto se mueve más rápido, la distancia entre la posición deseada y la del objeto es más grande en cada instante de tiempo, lo que provoca que la trayectoria que se realiza para intentar minimizar dicho error se vaya alejando de la trayectoria que realmente está efectuando el objeto, en este caso una trayectoria triangular.

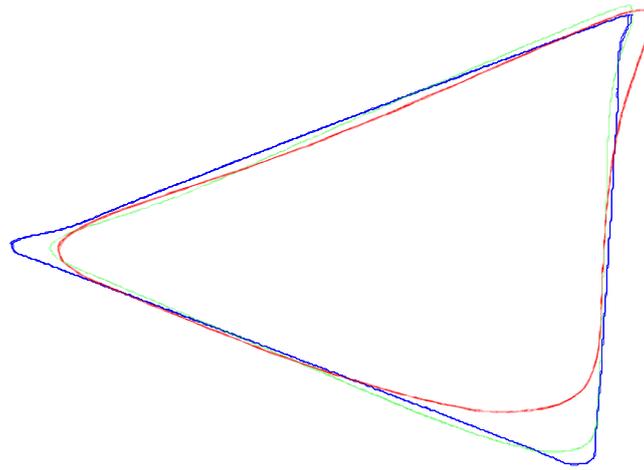


Figura 242. Comparativa entre las trayectorias efectuadas según el cambio de velocidad del objeto con trayectoria triángulo y interfaz DR-ZOH: en azul con periodo 30 segundos; en verde con periodo 10 segundos; en rojo con periodo 5 segundos.

Ahora se van a realizar los mismos experimentos pero utilizando la Interfaz DR-FOH. La gráfica de la Figura 243 muestra la señal de error obtenida del experimento con el enfoque IBVS Multifrecuencia con DR-FOH para un objeto que está siguiendo una trayectoria triángulo con un tiempo de 30 segundos para realizarla.

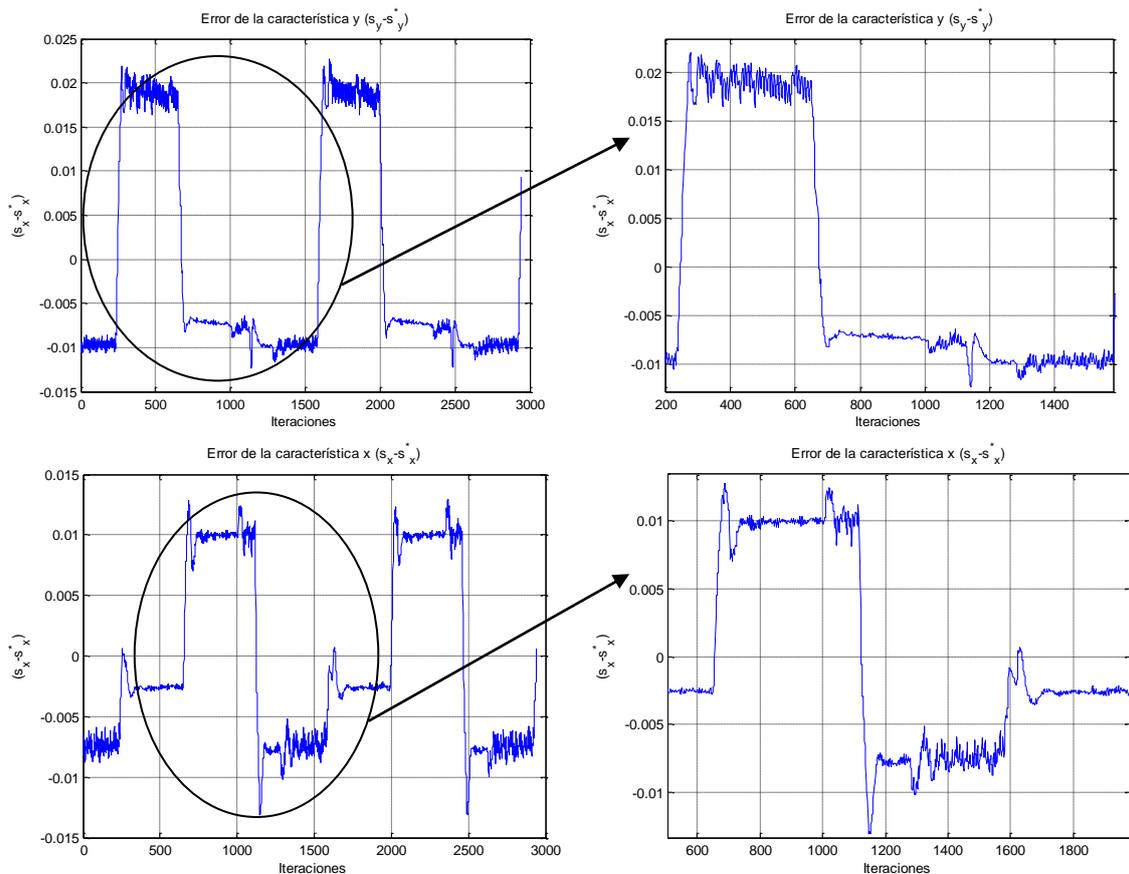


Figura 243. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 30 segundos y con DR-FOH.

Se puede comprobar que los resultados son muy similares a los vistos en el caso de la Interfaz DR-ZOH. Esto es así porque el objeto está yendo a una velocidad bastante lenta y ambos enfoques son capaces de seguirlo con un error muy pequeño. La Figura 244 muestra la trayectoria realizada por el efector final.

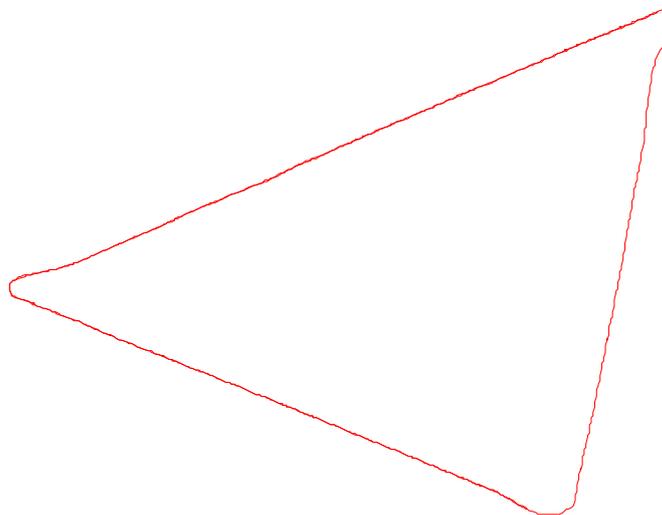


Figura 244. Trayectoria efectuada por el Efector final del Kuka KR5 sixx R650 con DR-FOH y trayectoria triángulo con periodo 30 segundos.

Las acciones de control efectuadas se pueden ver en las gráficas de la Figura 245. Como en el caso anterior aquí también se comprueban pequeñas oscilaciones cuando el objeto efectúa cambios bruscos en el sentido de su trayectoria.

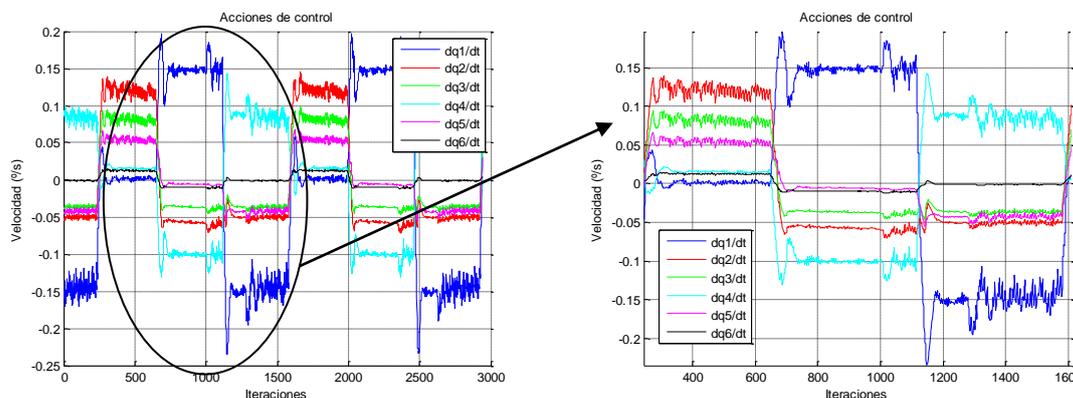


Figura 245. Acciones de control para la trayectoria triángulo de periodo 30 segundos con DR-FOH.

Las gráficas que se observan en la Figura 246 muestran la señal de error del sistema cuando el objeto tiene que recorrer la misma trayectoria pero en 10 segundos, es decir que el objeto se está moviendo más rápido.

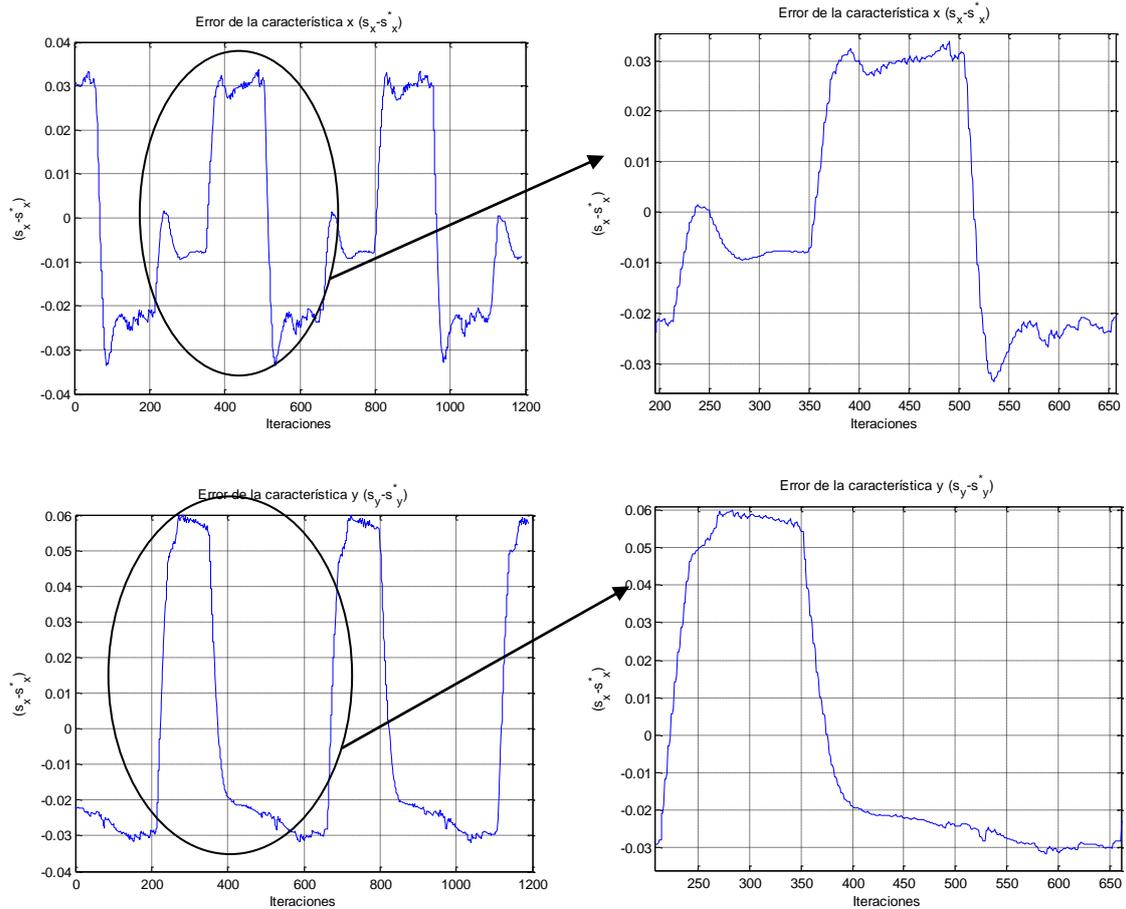


Figura 246. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 10 segundos y con DR-FOH.

Por otra parte la Figura 247 muestra la trayectoria realizada por el efector final, mientras que en las gráficas de la Figura 248 se pueden observar las acciones de control calculadas por el controlador de alto nivel del sistema.

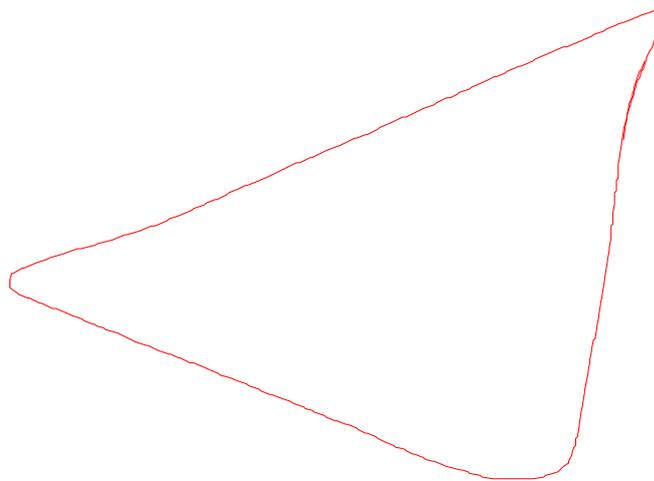


Figura 247. Trayectoria efectuada por el Efector final del Kuka KR5 sixx R650 con DR-FOH y trayectoria triángulo con periodo 10 segundos.

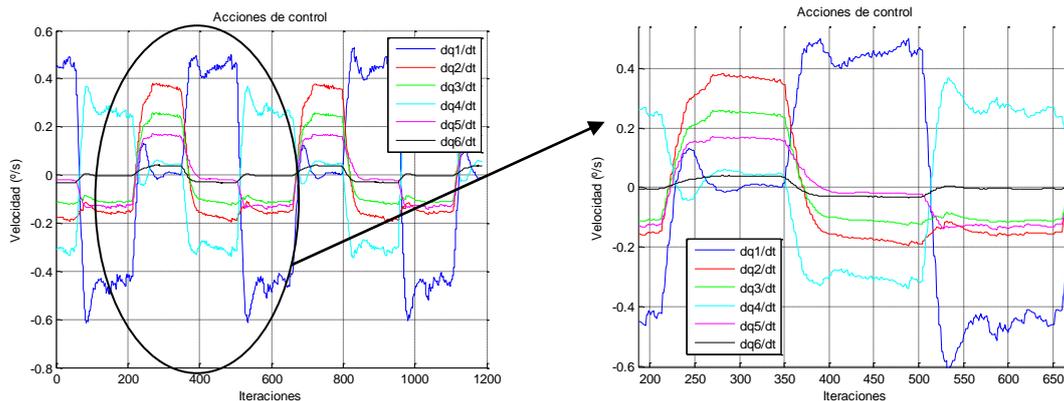


Figura 248. Acciones de control para la trayectoria triángulo de periodo 10 segundos con DR-FOH.

Por último, los resultados obtenidos con el objeto moviéndose con un periodo de 5 segundos se muestran a continuación. Las gráficas de la Figura 249 muestran la señal de error del sistema durante el tiempo que ha durado el experimento.

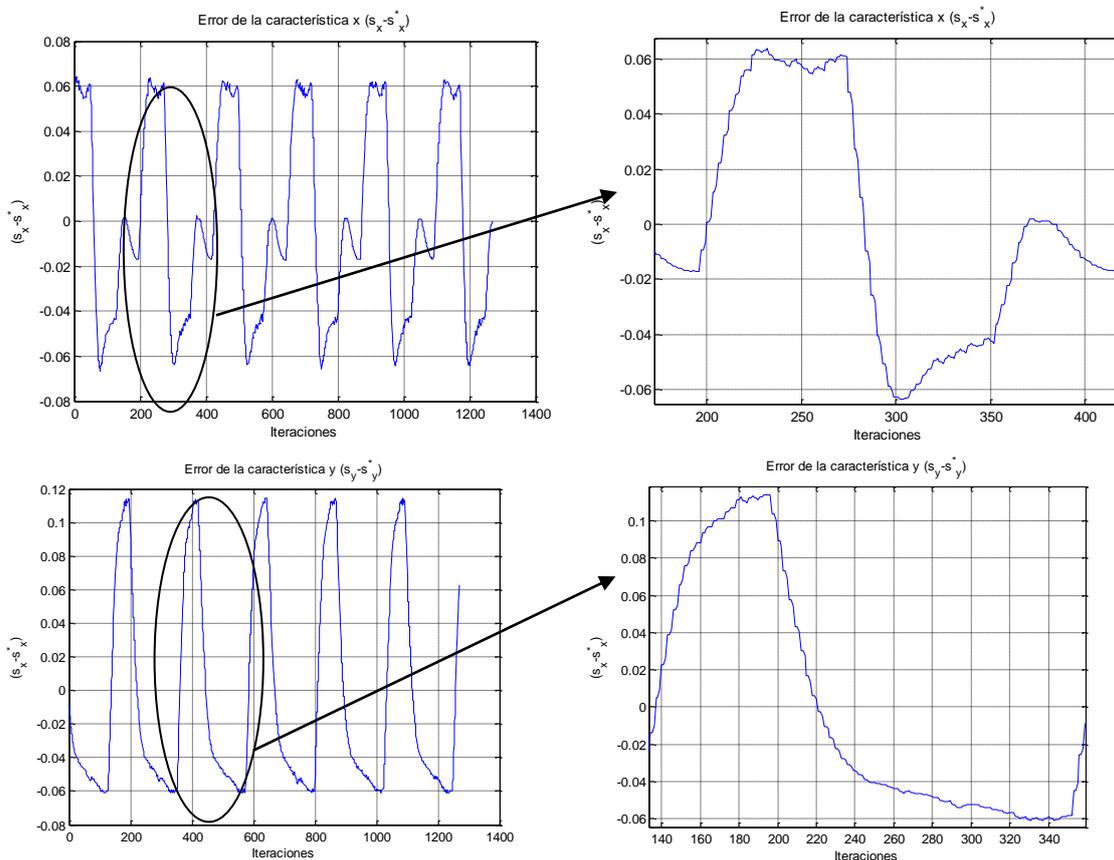


Figura 249. Señal de error con enfoque IBVS Multifrecuencia para trayectoria triángulo con Periodo 5 segundos y con DR-FOH.

La trayectoria efectuada por el efector final se puede observar en la gráfica de la Figura 250. Las acciones de control efectuadas en cada iteración per el algoritmo de control se pueden ver en las gráficas de la Figura 251.

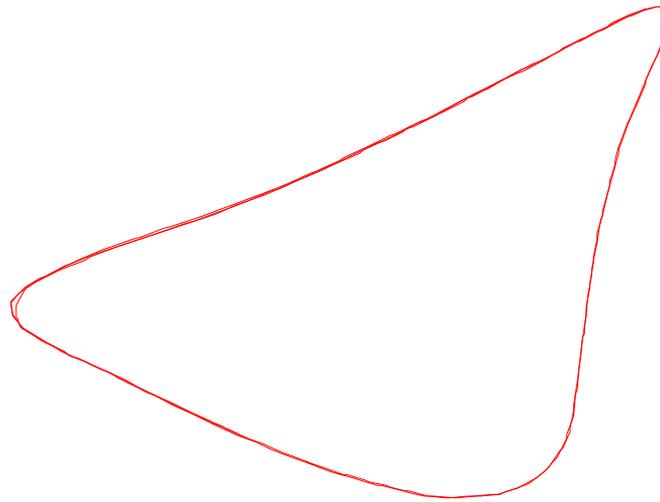


Figura 250. Trayectoria efectuada por el Efector final del Kuka KR5 sixx R650 con DR-FOH y trayectoria triángulo con periodo 5 segundos.

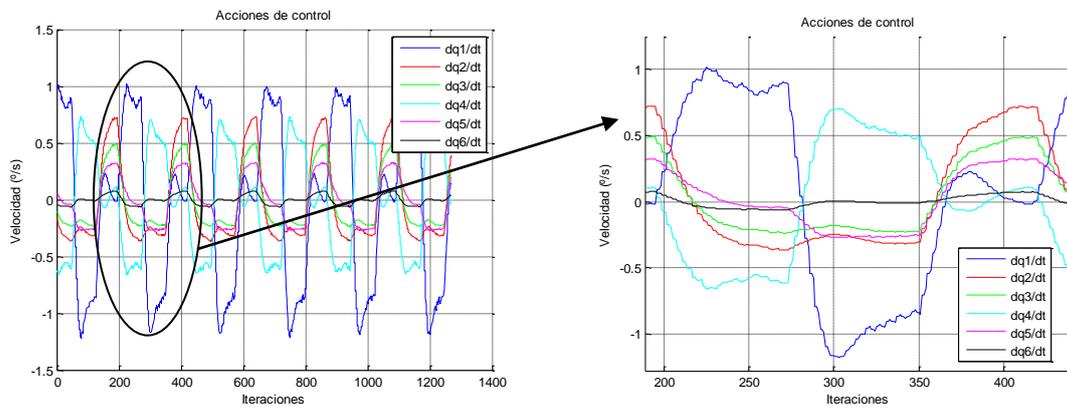


Figura 251. Acciones de control para la trayectoria triángulo de periodo 5 segundos con DR-FOH.

La gráfica de la Figura 252 muestra las trayectorias realizadas en función de la velocidad del objeto y cómo esta va deteriorándose en función de dicha velocidad.

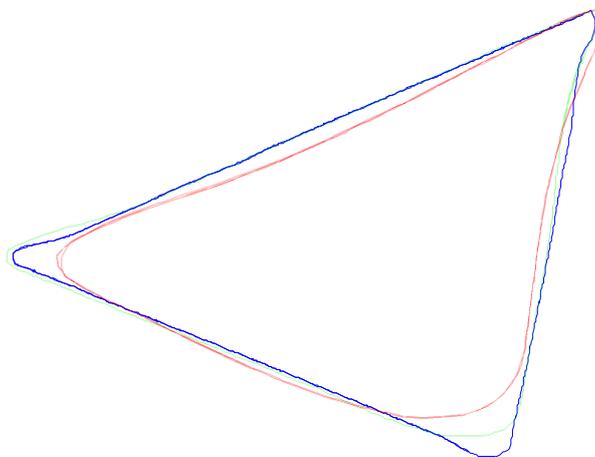


Figura 252. Comparativa entre las trayectorias efectuadas según el cambio de velocidad del objeto con trayectoria triángulo y interfaz DR-ZOH: en azul con periodo 30 segundos; en verde con periodo 10 segundos; en rojo con periodo 5 segundos.

Por último queda realizar una comparación entre cada uno de los dos enfoques. Las gráficas de la Figura 253 demuestran que ambos sistemas tienen un comportamiento similar aunque el error cometido por el enfoque realizado con el estimador DR-FOH sea ligeramente menor que con el DR-ZOH.

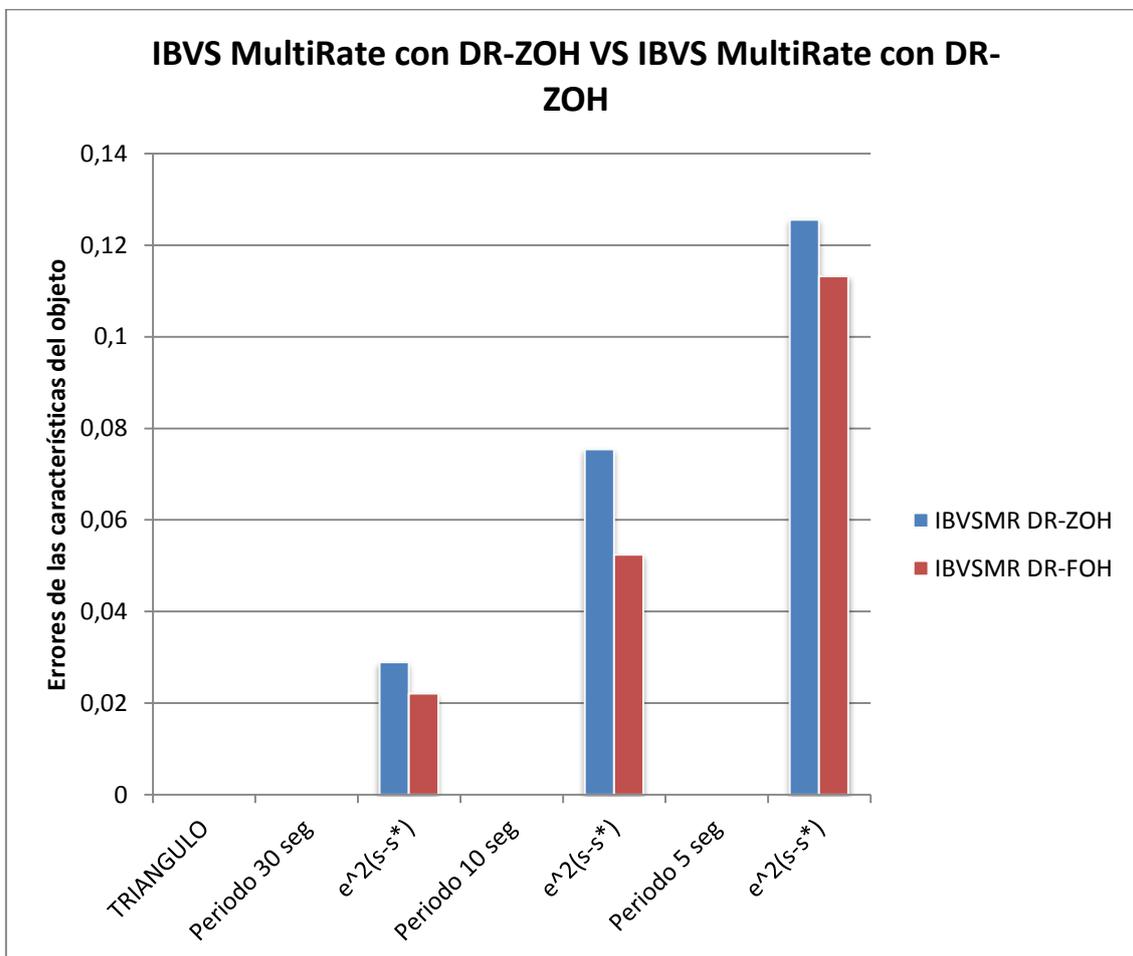


Figura 253. Comparación entre los dos interfaces DR-HOH utilizados en este experimento.

Las siguientes comparativas no se harán de forma tan exhaustiva, es decir, sólo se mostraran los diagramas de barras que indican los resultados obtenidos y la comparativa del comportamiento obtenido entre el IBVS Multifrecuencia con DR-ZOH y con DR-FOH.

- CUADRADO

Los resultados obtenidos con esta trayectoria cuando las interfaces utilizadas son las DR-ZOH se pueden ver en el gráfico mostrado en la Figura 254.

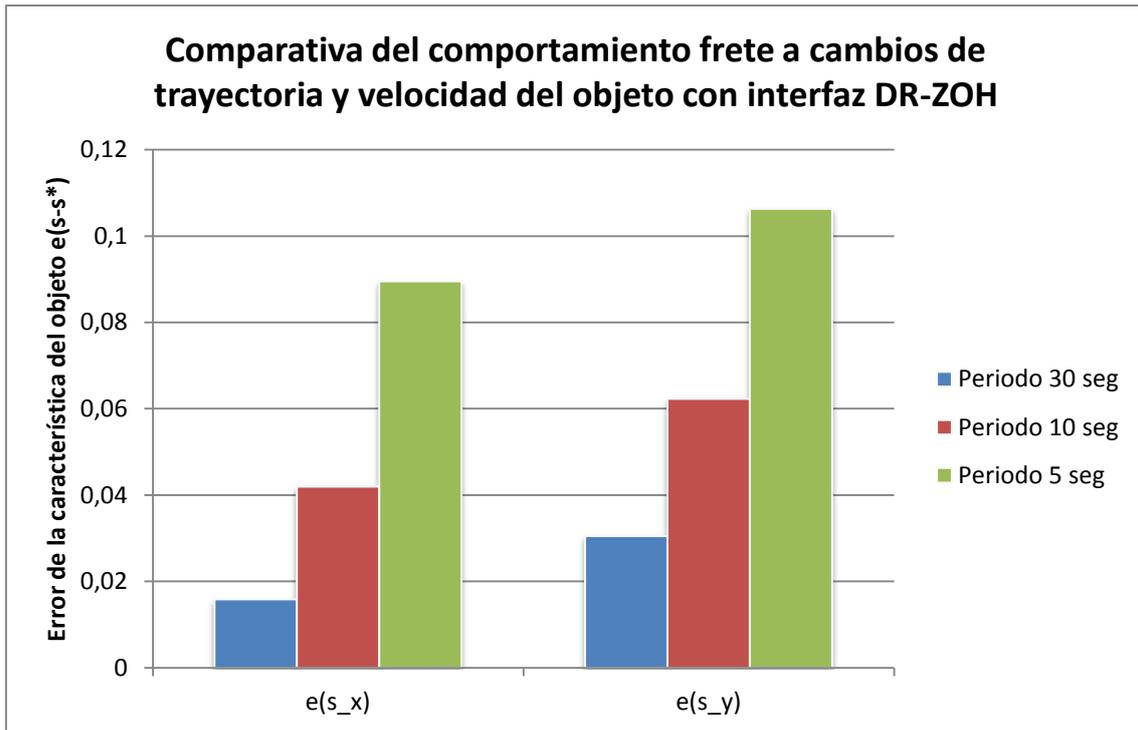


Figura 254. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-ZOH para la trayectoria cuadrado.

Del mismo modo, la Figura 255 muestra los resultados obtenidos en el caso de que la interfaz sea un DR-FOH.

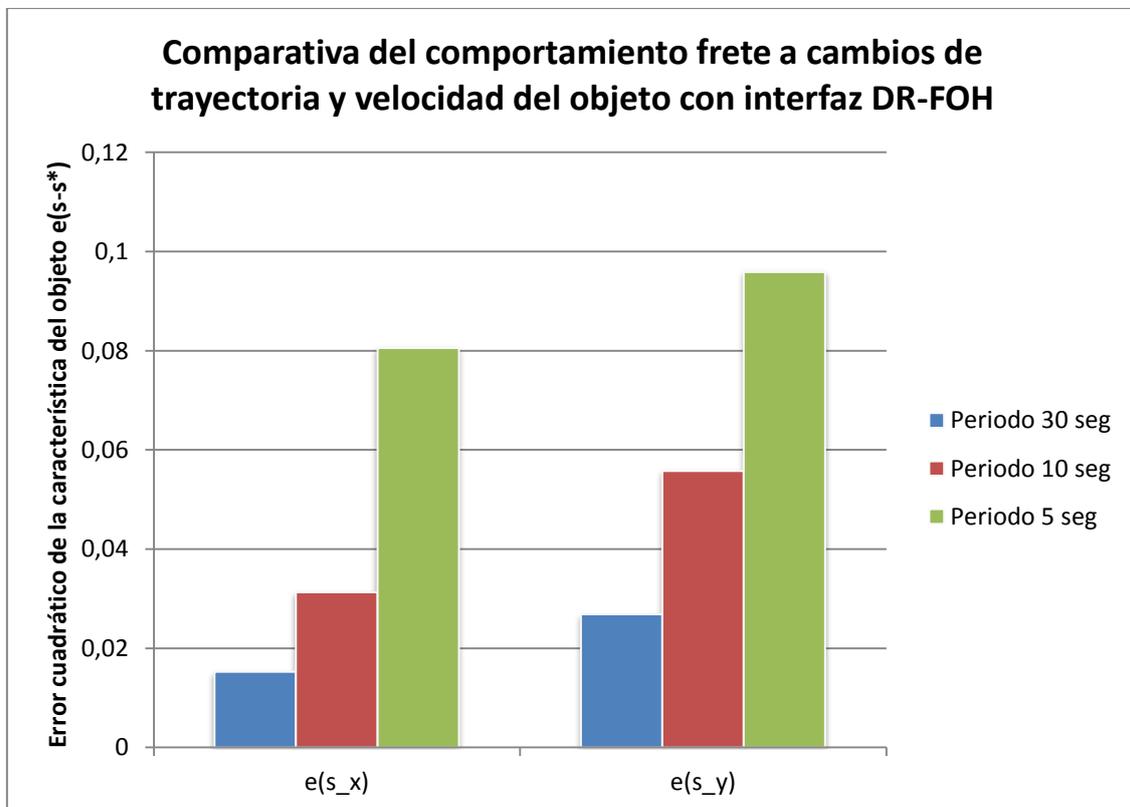


Figura 255. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-FOH para la trayectoria cuadrado.

Ahora, para comparar los resultados del comportamiento del sistema en ambos interfaces, la gráfica de la Figura 256 muestra el error cuadrático según la velocidad del objeto para cada uno de los enfoques anteriores. Como se puede ver, aunque el comportamiento es similar, el enfoque con un DR-FOH es ligeramente mejor que el del DR-ZOH.

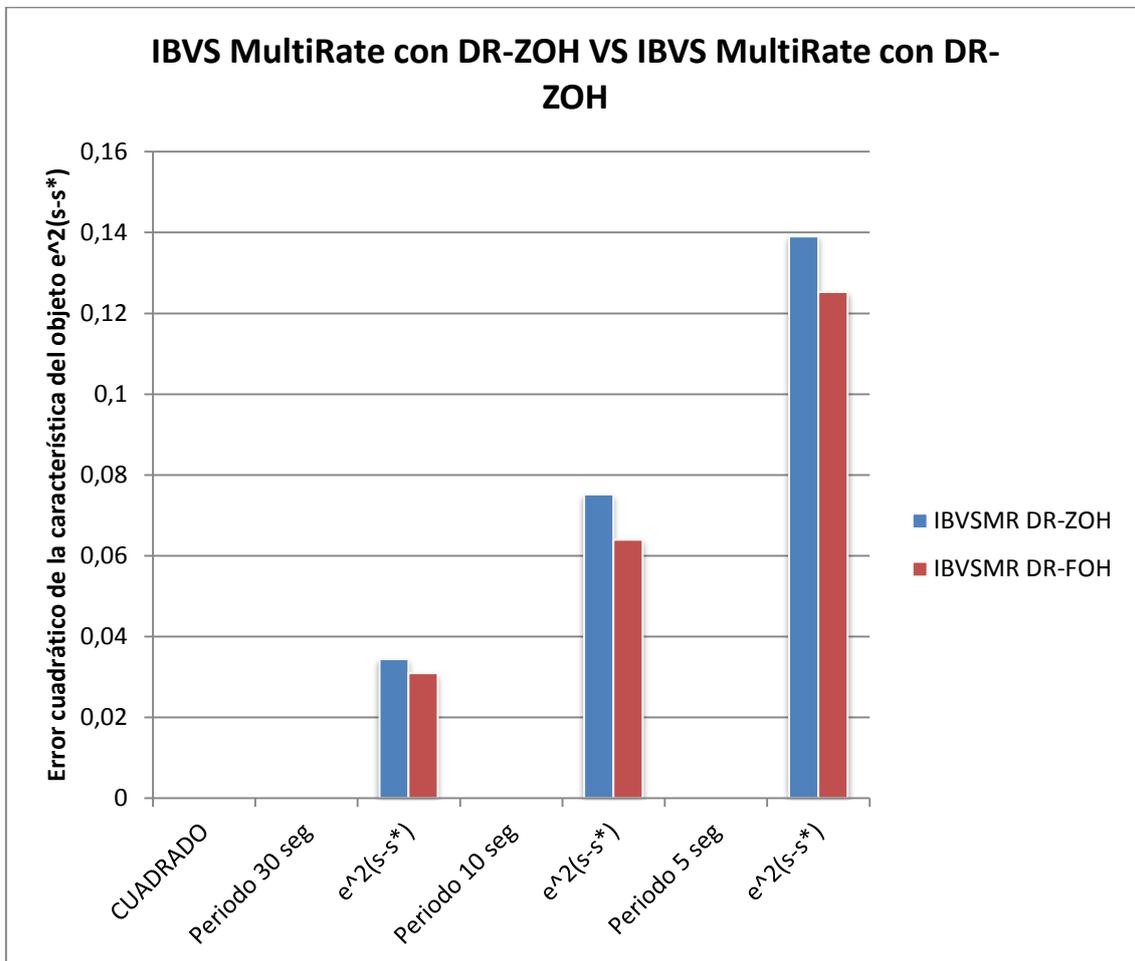


Figura 256. Comparación entre los dos interfaces DR-HOH utilizados en este experimento.

- CÍRCULO

Los resultados obtenidos con esta trayectoria en función de la velocidad del objeto utilizando interfaces DR-ZOH se muestran en el diagrama de barras de la Figura 257. Del mismo modo los resultados sobre el comportamiento del sistema cuando la interfaz utilizada es un DR-FOH se muestra en el gráfico de la Figura 258.

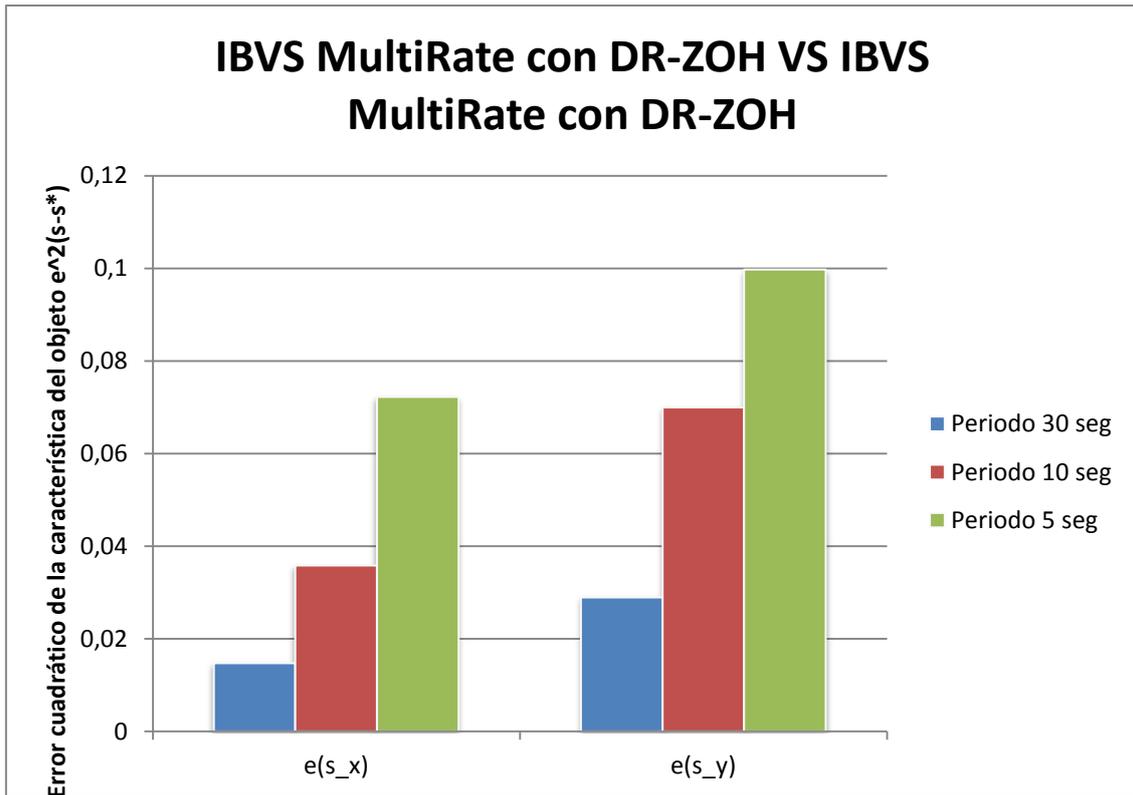


Figura 257. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-ZOH para la trayectoria círculo.

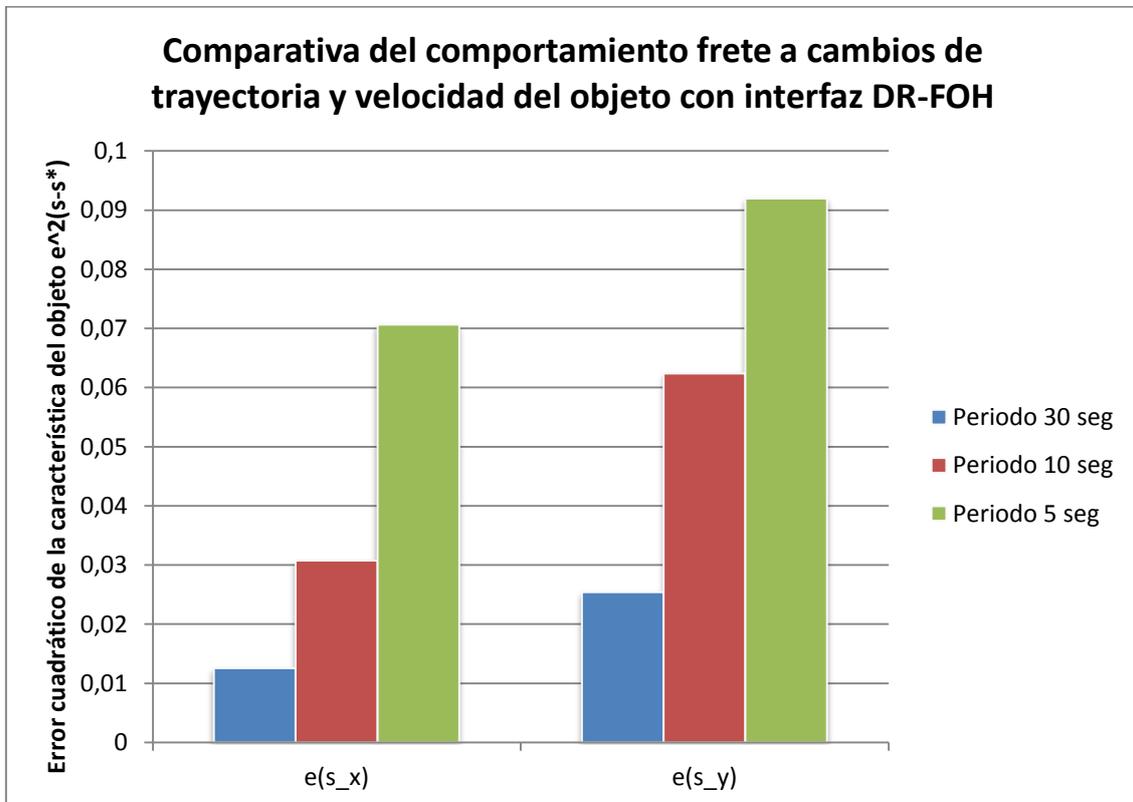


Figura 258. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-FOH para la trayectoria círculo.

La comparativa entre ambos DR-HOH se puede ver en el gráfico de la Figura 259. Se comprueba que, aunque el comportamiento es similar, el sistema con DR-FOH vuelve a presentar mejor comportamiento que empleando el DR-ZOH.

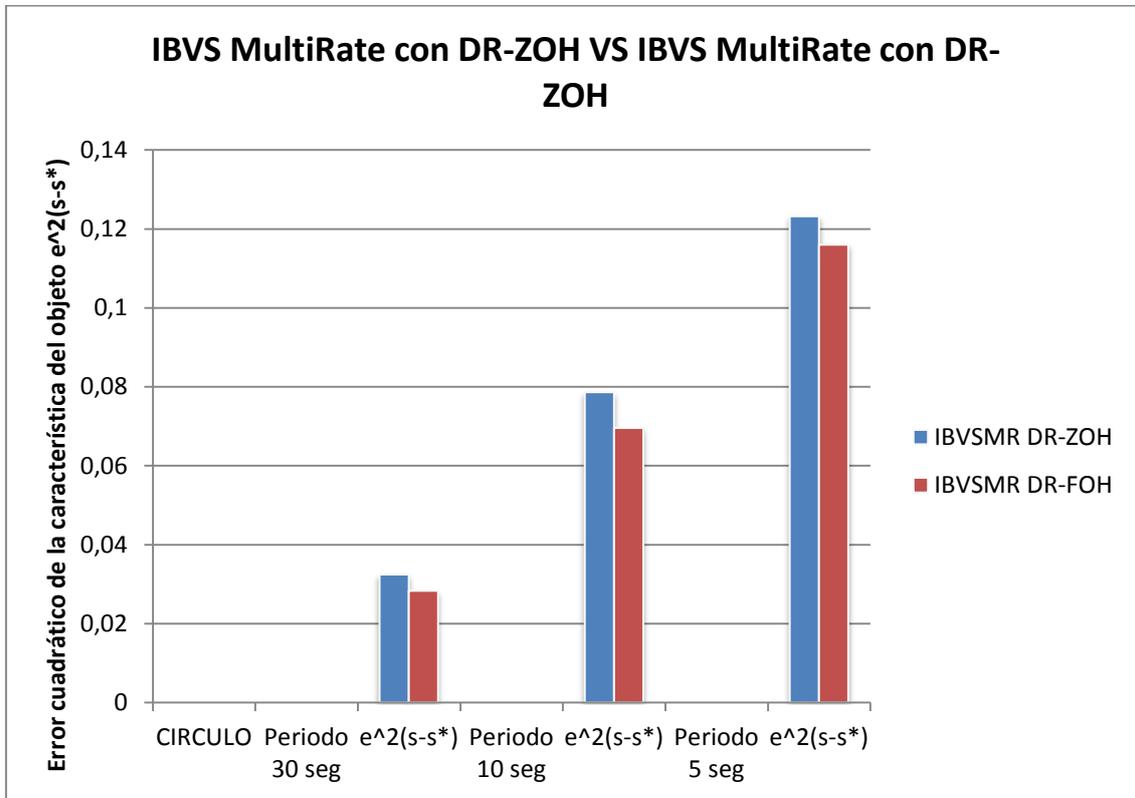


Figura 259. Comparación entre los dos interfaces DR-HOH utilizados en este experimento.

- INFINITO

Por último, los resultados obtenidos con esta trayectoria en función de la velocidad del objeto utilizando interfaces DR-ZOH se muestran en el diagrama de barras de la Figura 260. Del mismo modo los resultados sobre el comportamiento del sistema cuando la interfaz utilizada es un DR-FOH se muestra en el gráfico de la Figura 261.

Además, el gráfico de la Figura 262 muestra la comparativa entre las dos interfaces estudiadas. Se ha utilizado para ello el error cuadrático obtenido a partir de las componentes del error anteriormente vistas. Esta gráfica vuelve a confirmar que, aunque leve, existe una mejoría del comportamiento del sistema cuando se emplea la interfaz DR-FOH a cuando se emplea la interfaz DR-ZOH.

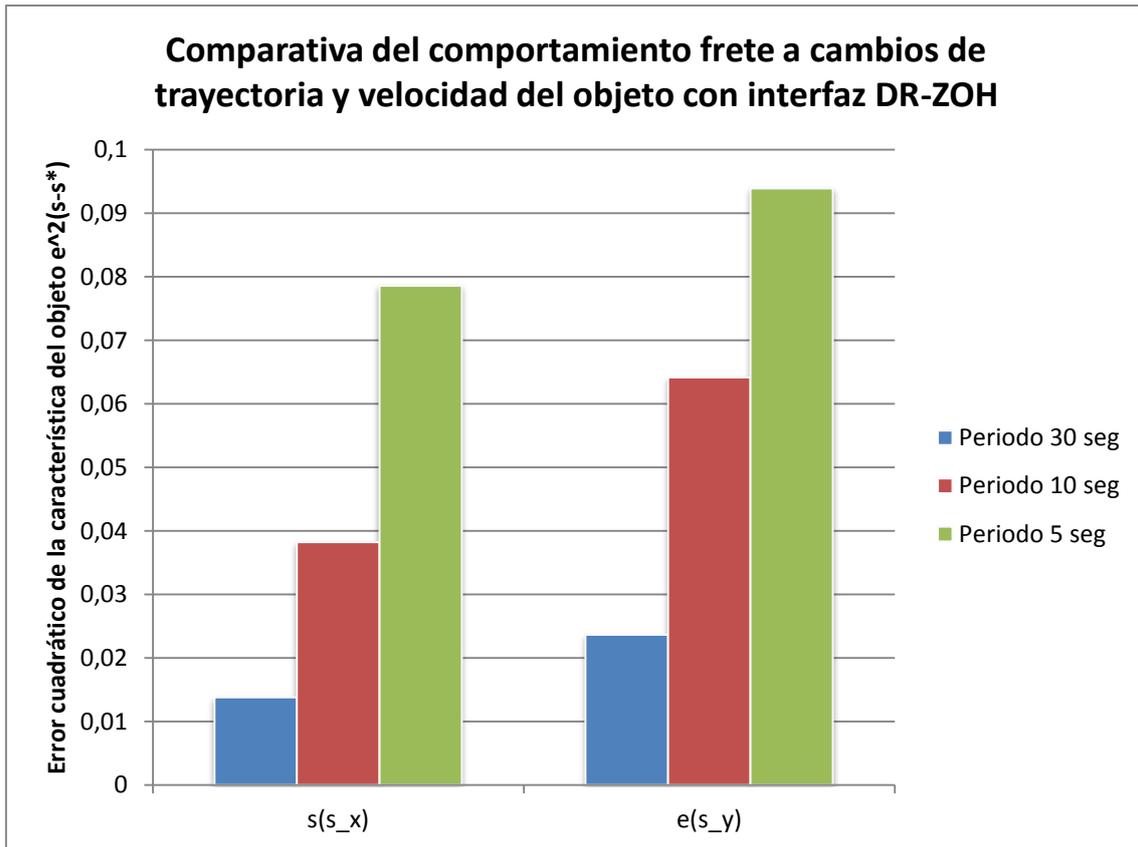


Figura 260. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-ZOH para la trayectoria infinito.

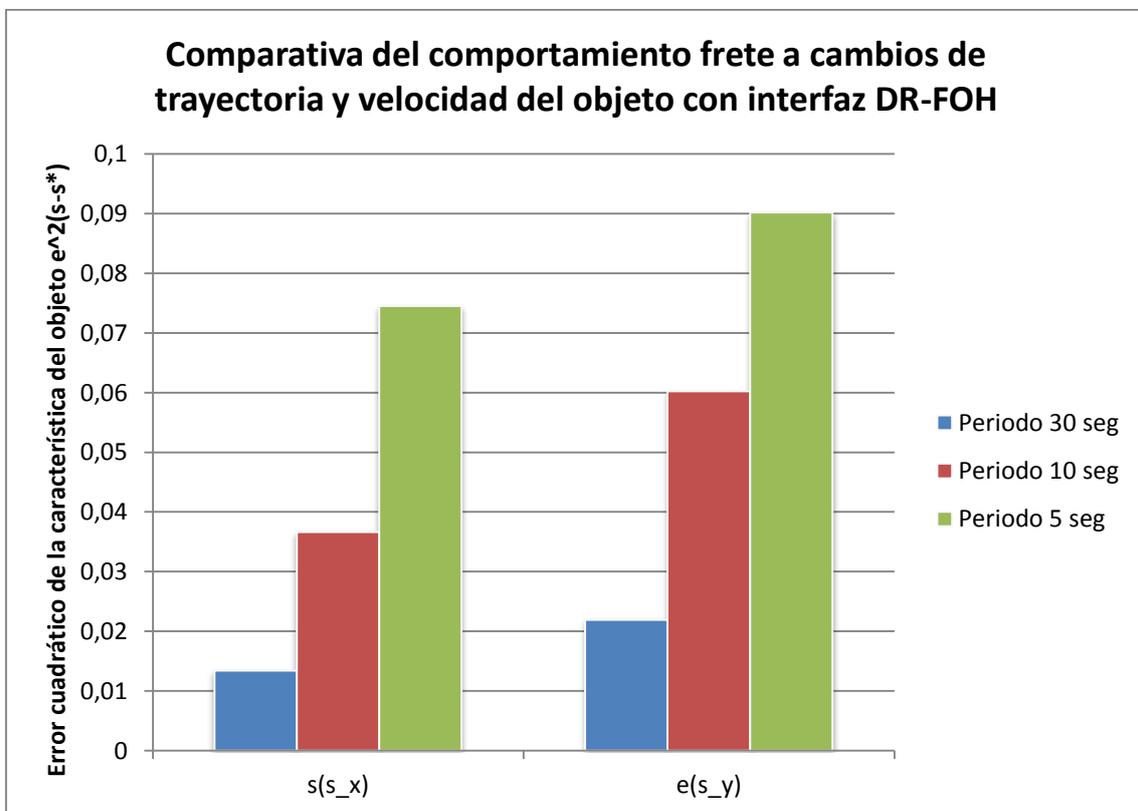


Figura 261. Resultados obtenidos con el enfoque IBVS Multifrecuencia con interfaz DR-ZOH para la trayectoria infinito.

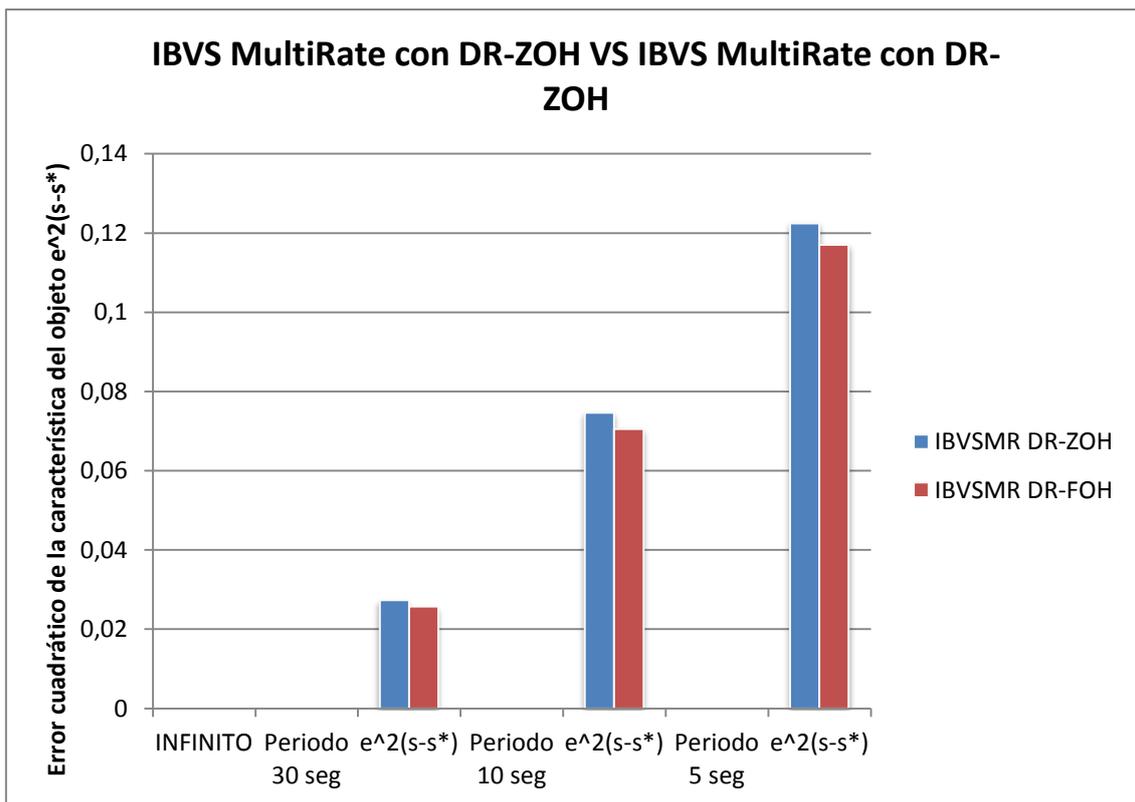


Figura 262. Comparación entre los dos interfaces DR-HOH utilizados en este experimento.

4.3.3.-Comparativa con IBVS clásico

Para finalizar con el estudio y del comportamiento del sistema IBVS Multifrecuencia, se hace necesario realizar una comparativa entre el enfoque clásico y el enfoque multifrecuencia para cuantificar los beneficios o los no beneficios de introducir este tipo de enfoques en este tipo de sistemas.

Es por ello que la comparativa se hará directamente observando los resultados obtenidos del error cuadrático, ya que se piensa que es el indicador que mejor visualiza el comportamiento obtenido en cada uno de los enfoques empleados.

Por consiguiente en la Figura 263 se puede observar la comparativa entre el IBVS clásico, el IBVS Multifrecuencia con DR-ZOH y el IBVS Multifrecuencia con DR-FOH cuando el objeto está moviéndose con periodo 30 segundos. Se observa que la mejora entre el enfoque clásico y el multifrecuencia ronda un 50% menos de error cuadrático. Es decir que se ha producido una mejora en la rapidez con la que se llega a la referencia del casi el 50% lo que permite el seguimiento de objetos que van a velocidades más grandes.

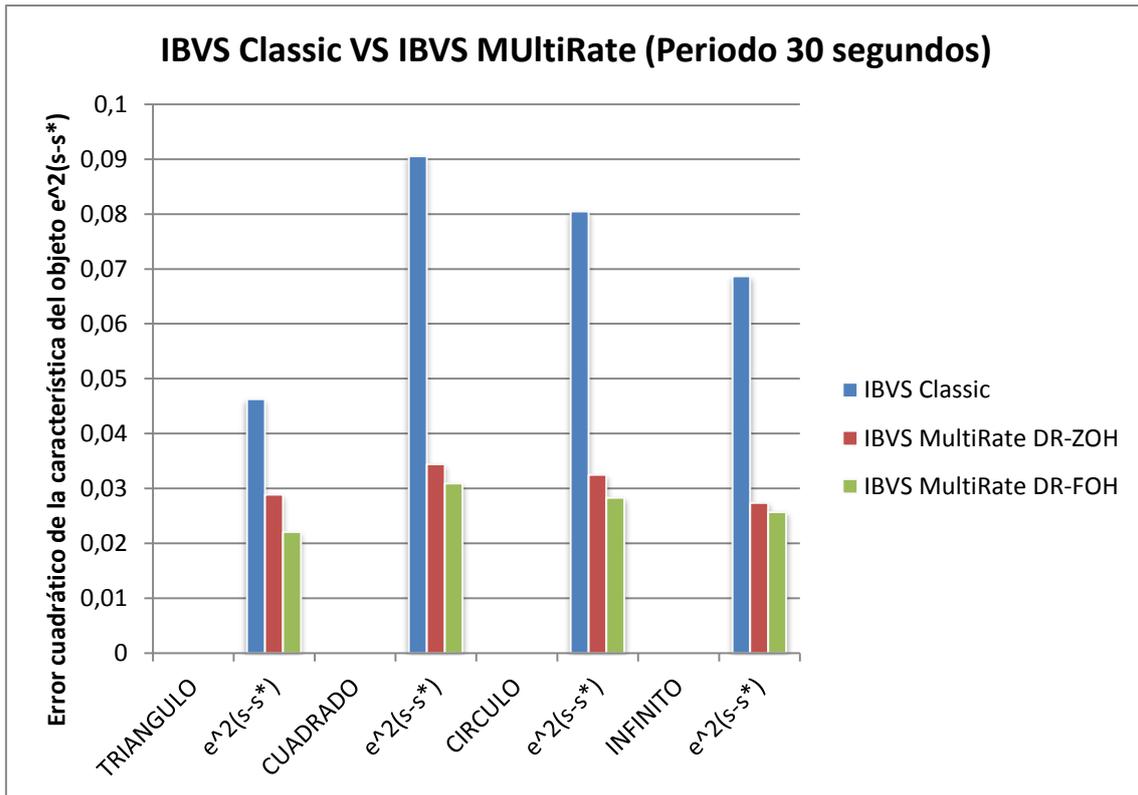


Figura 263. Comparativa entre los enfoques clásico y multifrecuencia (periodo 30 segundos).

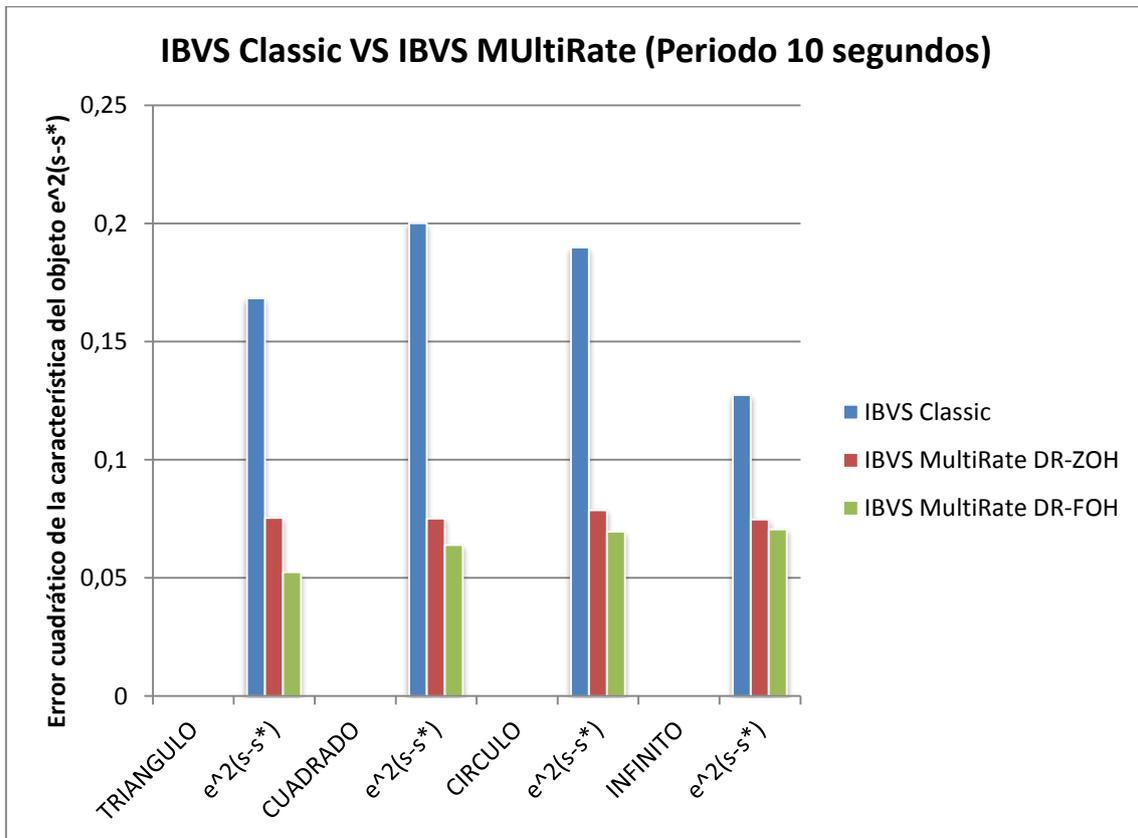


Figura 264. Comparativa entre los enfoques clásico y multifrecuencia (periodo 10 segundos).

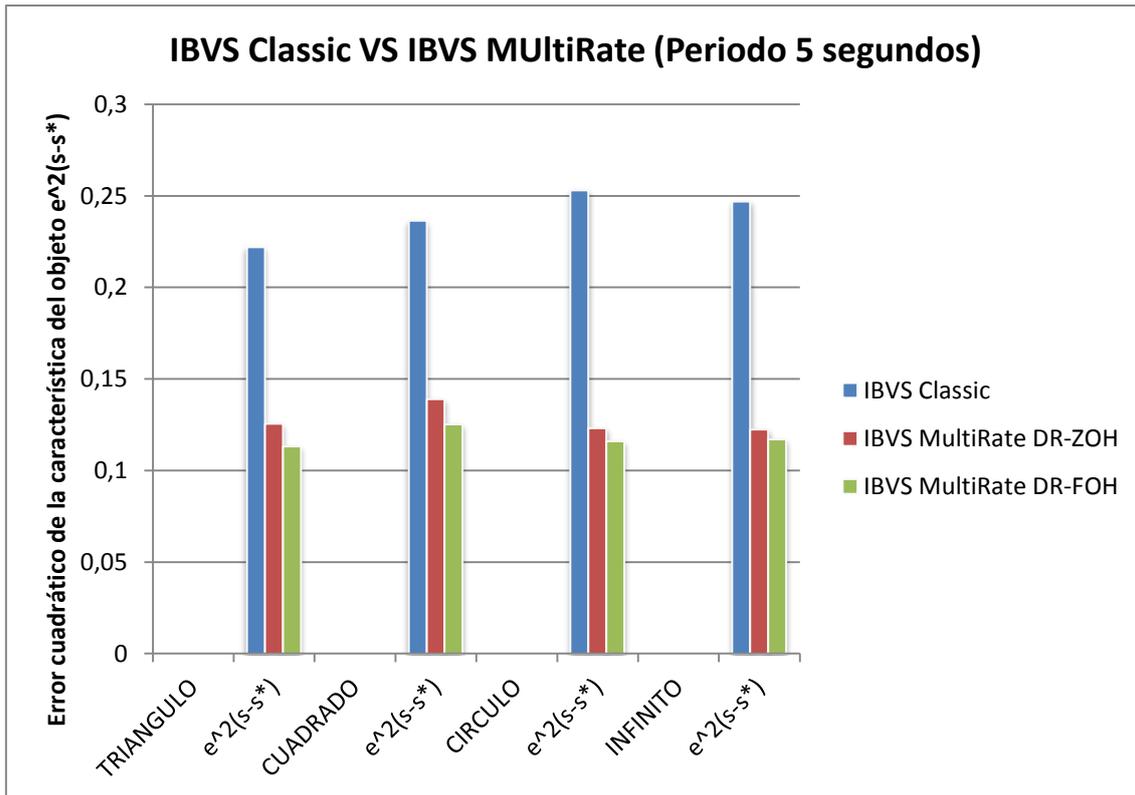


Figura 265. Comparativa entre los enfoques clásico y multifrecuencia (periodo 5 segundos).

La Figuras 264 y 265 muestran la misma comparativa pero con los periodos de 10 y 5 segundos, respectivamente. En ambos casos se comprueba que la respuesta con el enfoque multifrecuencia es mucho mejor que con la del enfoque clásico, siempre hablando que se tiene la misma ganancia en el regulador.

4.4.-Conclusiones

En este capítulo se ha podido ver la realización e implementación de un enfoque multifrecuencia como son las interfaces DR-HOH/ DR-HOS en un sistema altamente no lineal como es un sistema robotizado o un sistema de visión.

Además del estudio realizado con un regulador PID para un sistema Visual Servoing de 2 grados de libertad se han obtenido las siguientes conclusiones:

- El sistema con tratamiento Multifrecuencia mejora el comportamiento del sistema con respecto al que se tiene con el control clásico monofrecuencia.
- La interfaz que mejores resultados proporciona al sistema Visual Servoing, en este caso ha sido el DR-FOH.
- Si además, se hace un tratamiento asíncrono del sistema, su comportamiento es aún mejor.

Una vez obtenidas las anteriores conclusiones se ha realizado pruebas bajo el simulador realizado con Matlab/ Simulink para ver el efecto del orden de multiplicidad sobre el sistema, así como realizar comprobaciones acerca del comportamiento del mismo antes y después del enfoque multifrecuencia. Las conclusiones obtenidas han sido:

- En caso de no tener restricciones físicas, el comportamiento del sistema siempre que se cumpla la condición de que $N_1 \cdot N_2 \equiv cte$ es el mismo.
- Con restricciones físicas, el enfoque multifrecuencia mejora el comportamiento del sistema.
- Además, se vuelve a ratificar que el DR-FOH ofrece un comportamiento más parecido al que se tendría si se trabajara a alta frecuencia.

Con todos estos análisis claros, se ha realizado la implementación del sistema IBVS Multifrecuencia utilizando las dos interfaces con mejores resultados, la DR-ZOH y la DR-FOH. El resultado ha sido que, manteniendo constante la matriz de interacción, es decir, empleando $\hat{L}_s^+ = L_{s^*}^+$ el comportamiento del sistema cuando se trabaja con el enfoque multifrecuencia mejora en alrededor el 50% respecto al que se tiene cuando se trabaja con el enfoque clásico.

Por el contrario no hay tanta diferencia entre el comportamiento cuando se utiliza un DR-ZOH o cuando se emplea un DR-FOH. Esto es debido al regulador. Las simulaciones realizadas han sido empleando la matriz de interacción con estimación de Z, y es por ello que en cada iteración el controlador cambia y se discretiza online.

Por el contrario, cuando se emplea el enfoque $\hat{L}_s^+ = L_{s^*}^+$ el regulador no es más que una ganancia en cada instante de tiempo y por este motivo las diferencias no son tan manifiestas en el proceso real. Aun así, sí que se ha visto que se tiene una ligera mejoría en todos los experimentos realizados.

CAPÍTULO 5

Conclusiones y trabajos futuros

En esta tesina se ha estudiado la problemática de *Visual Servoing* para el tracking de objetos en movimiento a velocidad elevada. Para ello se ha combinado las siguientes técnicas:

- ***Image-Based Visual Servoing***
- ***Control Multifrecuencia basado en los DR-HOH***

En la parte de *Visual Servoing* se ha analizado e implementado los tres métodos tradicionales presentes en la literatura, con objeto de comparar las distintas técnicas, al objeto de seleccionar la más adecuada a la problemática planteada.

En concreto se ha elegido la técnica Image-Based Visual Servoing por ser la más empleada tanto en la literatura como en los procesos reales. El motivo es que al realizarse el control en el espacio de trabajo del plano de imagen (S), no existen problemas derivados a una mala calibración del sistema.

En el ámbito de control multifrecuencia se han estudiado las distintas técnicas, centrándonos en los retenedores multifrecuencia basados en funciones primitivas, habiéndose implementado los de tipo ZOH, FOH, TH y SOH.

La elección de esta técnica frente a otras basadas en modelos de los procesos, se justifica por las altas no linealidades asociadas a los procesos involucrados. Estas no linealidades derivan del uso de brazos robots con articulaciones de revolución, pero también de los modelos y algorítmica asociada a los procesos de adquisición y procesamiento de imágenes.

Las problemáticas a resolver mediante esta combinación de técnicas son muy variadas. No obstante, nos hemos centrado tanto para el estudio como para la implementación, en el tracking de objetos moviéndose a gran velocidad por medio de robots industriales. Vimos en esta temática numerosos procesos industriales, en que la resolución de este problema provocaría un aumento en los tiempos de producción, lo que en definitiva equivale a un aumento de los beneficios.

En este sentido, se ha puesto a punto (*setup*) un sistema Visual Servoing con un robot industrial, el *KUKA KR5 sixx R650*. Se trata de un brazo robot de la familia de los pequeños, que por medio de una tarjeta de tiempo real, RSI-XML Ethernet, es capaz de recibir información de sistemas externos.

En el sistema implementado, nos hemos encontrado con los siguientes problemas concretos:

- **El robot industrial utilizado no es un robot de investigación**, lo que implica mayor dificultad a la hora de realizar cualquier cambio o control. Estos robots están muy preparados para la realización de tareas triviales de *pick&place*, control *point-to-point*, etc., pero cuando se requiere

realizar, por ejemplo un control de velocidad por eje, el controlador no lo permite, ya que dicha velocidad la gestiona el controlador del KUKA, y la forma que lo hace es secreto de empresa. Como este ejemplo, muchos otros que han hecho más difícil el desarrollo de la aplicación.

- **La comunicación TCP/IP:** en este sentido el problema puede venir si hay un retardo en el envío de paquetes cuando el controlador está esperando la respuesta del sistema externo. Si el tiempo esperado es de más de 10 milisegundos, el controlador interpreta que ha habido un problema e interrumpe la comunicación, parando el programa en ejecución.
- **Sistema Eye-in-Hand:** en esta tesina se ha utilizado la configuración *Eye-in-Hand* con una sola cámara. Aunque en principio esto implica que la calibración del sistema es más simple, surge el problema de la estimación de la profundidad con solamente una cámara.

Los servidores proporcionados por la compañía KUKA, resuelven el problema de comunicación por medio de un retenedor de orden cero, aunque si el sistema externo tarda un cierto número de paquetes en comunicarse con el servidor, éste también interrumpe el envío y por tanto la comunicación. Esto, que a priori no parece tan grave, realmente puede serlo, ya que un retardo por culpa del sistema operativo del sistema externo, puede provocar un fallo de comunicación y que el robot pare. El motivo es que el sistema KUKA está más bien pensado para la inclusión de sensores y que el control lo lleve el controlador de bajo nivel, con las limitaciones que esto supone.

Durante el desarrollo de esta tesina, se ha estado trabajando en la implementación de un sistema de guiado del robot por medio de pantallas multitáctiles. La finalidad de este proyecto, es dar a los operarios de los robots mayor facilidad a la hora de realizar una teleoperación con este tipo de sistemas. Estos sistemas trabajan bajo el sistema operativo Windows 7 y utilizan la versión .NetFramework 4.0 para sacar todo el potencial a la pantalla multitáctil. Uno de los problemas que se tuvo en este proyecto fueron los cortes en la comunicación, debido a los retardos que el sistema provocaba.

Con el problema anterior ya quedaría justificado el uso del enfoque multifrecuencia. Si bien es cierto que la etapa de adquisición y procesamiento es más lenta que la dinámica del robot, lo que nos da lugar a la segunda justificación de la incorporación de un diseño multifrecuencia para proporcionar el comportamiento adecuado del mismo.

La combinación de ambas técnicas (la del Visual Servoing y la de la multifrecuencia) es muy novedosa, dado que la multifrecuencia está muy estudiada para sistemas lineales y no para sistemas no lineales. Mediante el uso de los

retenedores, obviamos el tratamiento de las no linealidades, lo que abre un campo nuevo de investigación: “*El Visual Servoing Multifrecuencia*”.

Es por ello que el estudio de los modelos de planta conocidos como *N-delay input/ output* ya se han analizado y estudiado en la presente tesina, cara a ampliar la multifrecuencia al resto de bucle de control.

Además de implementar en simulación las tres técnicas tradicionales (IBVS, PBVS y HBVS), se ha implementado en un sistema robotizado industrial, con los condicionamientos asociados, la técnica *Image-Based Visual Servoing*, mejorada mediante técnicas *Multifrecuenciales*. Tanto sobre simulaciones como en el sistema real, se ha realizado gran cantidad de experiencias con objeto de validar las propuestas de diseño realizadas a lo largo de la presente tesina.

El análisis de dichos experimentos ha dado como resultado que somos capaces de seguir objetos moviéndose mucho más rápido, con menor error de seguimiento. Esto implica que la trayectoria seguida por el robot es más parecida a la trayectoria que está efectuando el objeto. En concreto, se ha utilizado el índice cuadrático para evaluar esta mejora.

Para las experiencias se han utilizado trayectorias predefinidas y utilizadas en el ámbito industrial como serían, el movimiento punto a punto, trayectorias circulares y trapezoidales. Con ello se ha verificado tanto aspectos de precisión como de repetibilidad, ambos de vital importancia en ámbitos industriales.

De los resultados experimentales con el robot industrial, se deduce que la introducción de la multifrecuencia reduce los errores de seguimiento en alrededor de un 50% con respecto al Visual Servoing clásico. La mejora por la multifrecuencia resulta similar con el uso de distintos retenedores.

Es de destacar que con la utilización del PID multifrecuencia los resultados mejoran incluso más que con el controlador clásico IBVS, lo que nos va a hacer reflexionar de cara a implementaciones futuras. Estos resultados nos abren un nuevo campo de investigación en el que se combinará técnicas clásicas de Visual Servoing con técnicas de control, ambas en el ámbito multifrecuencia.

Se observa que en el caso de los controladores, los retenedores aportan distintos comportamientos. En concreto utilizando PID con FOH multifrecuencia, la mejora con respecto a las técnicas clásicas de Visual Servoing es de un 70%.

Como trabajos futuros, la multifrecuencia puede extenderse a partes de los procesos mediante procesos de linealización de dichas partes, dando lugar a un sinnúmero de posibilidades de investigación. De hecho, se desconocen aportaciones en esta línea en toda la literatura buscada a lo largo de la realización de esta tesina.

Un camino inmediato de actuación, en esta línea, nos lo da la utilidad de retenedores basada en funciones, como método para introducir la dinámica y de los procesos en la propia unidad multifrecuencia. También los muestreadores multifrecuencia basados en filtro de Kalman tienen ámbito de aplicación en esta línea. Otra de las líneas es extender estas técnicas para manipuladores móviles.

Bibliografía

- [1] J.J. Lee and Y. Xu. A multi-rate control scheme of robot manipulators. *Journal of System Eng.*, 1:22-33, 1993.
- [2] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization. *Mach. Learning Autonomous Robots*, pages 29-53, 1998.
- [3] T. Nemani, T. Tsao, and S. Hutchinson. Multi-rate analysis and design of visual feedback digital servo-control system. *Journal of Dynamic Systems, Measurement and Control*, 116(1): 45-55, 1994.
- [4] F. Sun, Z. Sun, R. Zhang, and F. Zhang. A multirate adaptive composite controller for flexible-link robots using neural networks. *IEEE International Conf. on Systems, Man and Cybernetics*, 2:1036-1041, 1996.
- [5] R. Piza. *Percepción, Modelado del Entorno y Localización de Robots Móviles Autónomos mediante Técnicas de Muestreo No Convencional*. PhD thesis, Dept. de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia, España, 2003.
- [6] A. Huster and S.M. Rock. Relative position sensing by fusing monocular vision and inertial rate sensors. In *Int. Conf. on Advanced Robotics*, pages 1562-1567, 2003.
- [7] L. Armesto, S. Chroust, M. Vincze, and J. Tornero. Multirate fusion with vision and inertial sensors. In *Int. Conf. on Robotics and Automation*, pages 193-199, 2004.
- [8] M.T. Tham, F. Vagi, A.J. Morris, and R.K. Wood. Multivariable and multirate selftuning control. A distillation column case study. In *IEE Proceedings*, volume 138 of *D*, pages 9-24, 1991.
- [9] J.H. Lee, M.S. Gelormino, and M. Morari. Model predictive control of multirate sampled-data systems: a state-space approach. *Internation Journal Control*, 55(1):153-191, 1992.
- [10] K.Y. Zhu and K.V. Ling. Multirate control for a gasification process. In *American Control Conference*, volume 3, pages 3268-3272, 1994.
- [11] S. Ritchev and G. Franklin. A stability criterion for asynchronous multirate linear systems. *IEEE Transactions on Automatic Control*, pages 529-535, 1989.

- [12] E. Hovestadt. Multirate control algorithms for time-variable sampling periods. In *IFAC Design Methods of Control Systems*, pages 105-110, 1991.
- [13] V. Casanova and J. Salt. First approximation on the design of a multirate control loop for an iccs. In *ISPE/IEEE International conference on CAD/CAM, robotics and factories of the future*, 1999.
- [14] V. Casanova and J. Salt. Multirate control for an iccs environment. Part i: The random access delays. In *IFAC Workshop on linear time delay systems*, 2000.
- [15] V. Casanova and J. Salt. Real-time implementation of multirate control techniques for an iccs. In *IFAC World Congress*, 2000.
- [16] R.V. Patel and N. Munro. *Multivariable System Theory and Design*. Pergamon Press, London, 1982.
- [17] K.J. Astrom and B. Wittenmark. *Sistemas controlados por Computador*. Paraninfo, Madrid, 1988.
- [18] P. Albertos and J.J. Salt. Receding horizon control of non-uniformly sampled-data systems. In *American Control Conference*, pages 4300-4304, 1999.
- [19] J.J. Salt and P. Albertos. Multirate controller design by rate decomposition. In *Conference on Decision and Control*, pages 4895-4900, 2000.
- [20] P. Albertos and A. Crespo. Real-time control of unconventionally sampled data systems. In *IFAC Workshop on Algorithms and Architectures for Real-time Control*, Vilamoura, Portugal, 1997.
- [21] H. Fujimoto. Visual Servoing of 6 DOF Manipulator by Multirate Control with Depth Identification. In *Proceedings of the 42nd IEEE Conference on Decision and Control Maui, Hawaii USA, December 2003*.
- [22] K. Sasajima and H. Fujimoto. 6 DOF Multirate Visual Servoing for Quick Moving Objects. In *American Control Conference ACC '07, 2007*.
- [23] S. Hutchinson, G. Hager and P. Corke. A tutorial on visual servo control. In *IEEE Transactions on Robotics and Automation* 12(5), 651-670, 1996.
- [24] G. Chesì, G.L. Mariottini, D. Prattichizzo and A. Vicino. Epipole-based visual servoing for mobile robots. In *Advanced Robotics, Vol. 20, Nº. 2, pp. 255-280, VSP and Robotics Society of Japan, 2006*.
- [25] H. Lang, Y. Wang, C.W. de Silva. Visual servoing with LQR control for mobile robots. In *Control and Automation (ICCA), Xiamen, Junio 2010*.

-
- [26] L. Mejias, P. Campoy, S. Saripalli and G.S. Sukhatme. A visual Servoing approach for tracking features in urban areas using an autonomous helicopter. In *Robotics and Automation (ICRA), Orlando, Mayo 2006*.
- [27] C. Fan, B. Song, X. Cai and Y. Liu. Dynamic visual servoing of a small scale autonomous helicopter in uncalibrated environments. In *Intelligent Robots and Systems (IROS) IEEE/RSJ International Conference, Octubre 2009*.
- [28] S. Salazar, H. Romero, J. Gomez, R. Lozano. Real-time stereo visual Servoing control of an UAV having eight-rotors. In *Electrical Engineering, Computing Science and Automatic Control (CCE), Toluca, Enero 2009*.
- [29] J. Hill and W. Park. Real time control of a robot with a mobile camera. In *'Proceedings of the 9th International Symposium on Industrial Robots', pp. 233–246, 1979*.
- [30] A. Sanderson and L. Weiss. Image-based visual servo control using relational graph error signals, *Proc. IEEE* pp. 1074–1077, 1980.
- [31] P. Corke and M. Good. Dynamic effects in visual closed-loop systems. In *IEEE Transactions on Robotics and Automation* **12**(5), 671–696, 1996.
- [32] F. Chaumette, P. Rives and B. Espiau. Positioning a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* **3**, 2248–2253, 1991.
- [33] S.A. Hutchinson, G.D. Hager and P. Corke. A tutorial on visual servo control. In *IEEE Trans. Robotics and Automation*, 12-5, 651-670, 1996.
- [34] P. Allen, A. Timcenko, B. Yoshimi and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system, *IEEE Transactions on Robotics and Automation* 9(2), 152–165, 1993.
- [35] W. Wilson, C.W. Hulls and F. Janabi-Sharifi. Robust image processing and position-based visual servoing, In *M. Vincze & G. Hager, eds, Robust Vision for Manipulation, Spie/IEEE Series*, pp. 163–220, 2000.
- [36] P. Wunsch, S. Winkler and G. Hirzinger. Real-Time pose estimation of 3D objects from camera images using neural networks, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Vol. 3, Albuquerque, New Mexico*, pp. 3232–3237, 1997.
- [37] T. Drummond and R. Cipolla. Visual tracking and control using Lie algebras, In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR), Vol. 2, Fort Collins, Colorado*, pp. 652–657, 1999b.

- [38] W. Wilson, C. W. Hulls, and F. Janabi-Sharifi. Robust image processing and position-based visual servoing, In *M. Vincze & G. Hager, eds, Robust Vision for Manipulation, Spie/IEEE Series*, pp. 163–220, 2000.
- [39] G. Hager, W. Chang and A. Morse. Robot hand-eye coordination based on stereo vision. In *IEEE Control Systems Magazine* **15**(1), 30–39, 1995.
- [40] E. Malis, F. Chaumette and S. Boudet. Positioning a coarse-calibrated camera with respect to an unknown object by 2D 1/2 visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 1, pp. 1352–1359, 1998.
- [41] F. Chaumette, P. Rives and B. Espiau. Positioning a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 3, 2248–2253, 1991.
- [42] B. Espiau, F. Chaumette and P. Rives. A new approach to visual servoing in robotics. In *IEEE Transactions on Robotics and Automation* **8**(3), 313–326, 1992.
- [43] K. Hashimoto and T. Noritsugu. Performance and sensitivity in visual servoing, In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 2, pp. 2321–2326, 1998.
- [44] N. Papanikolopoulos and P. Khosla. Adaptive robotic visual tracking: Theory and experiments. In *IEEE Transactions on Automatic Control* **38**(3), 429–445, 1993.
- [45] K. Hashimoto, T. Ebine and H. Kimura. Visual servoing with hand-eye manipulator-optimal control approach. In *IEEE Transactions on Robotics and Automation* **12**(5), 766–774, 1996.
- [46] R. Horaud, F. Dornaika and B. Espiau. Visually guided object grasping. In *IEEE Transactions on Robotics and Automation* **14**(4), 525–532, 1998.
- [47] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In *Nature* **293**, 133–135, 1981.
- [48] C. Jerian and R. Jain. Structure from motion - A critical analysis of methods. In *IEEE Transactions on Systems, Man and Cybernetics* **21**(3), 572–588, 1991.
- [49] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *Workshop on Vision and Control, Block Island, USA*, 1997.
- [50] G. Hager. A modular system for robust positioning using feedback from stereo vision. In *IEEE Transactions on Robotics and Automation* **13**(4), 582–595, 1997.

- [51] H. Michel and P. Rives. Singularities in the determination of the situation of a robot effector from the perspective view of 3 points. In *Technical report 1850, Institut National de Recherche en Informatique et en Automatique (INRIA), 1993.*
- [52] L. Weiss, A. Sanderson and C. Neuman. Dynamic visual servo control of robots: An adaptive image-based approach. In *IEEE Journal on Robotics and Automation* **3(5)**, 404–417, 1987.
- [54] O. Faugeras. Three-Dimensional Computer Vision : A Geometric Viewpoint. In *The MIT Press, 1993.*
- [55] M.T. Tham, F. Vagi, A.J. Morris, and R.K. Wood. Multivariable and multirate self-tuning control. a distillation column case study. In *IEE Proceedings, volume 138 of D, pages 9-24, 1991.*
- [56] T. Hara and M. Tomizuka. Multi-rate controller for hard-disk drive with redesign of state estimator. In *American Control Conference, pages 3033-3037, 1998.*
- [57] D.P. Glasson. A new technique for multirate digital control. In *AIAA Journal of Guidance and Control Dynamics*, **5:379-382, 1982.**
- [58] J. Salt. Control de Procesos Industriales con Muestreo No Convencional. *PhD thesis, Dept. Ing. Sistemas y Automática, Universidad Politécnica de Valencia., 1992.*
- [59] P. Albertos. Block Multirate Input-Output model for sampled-data control systems. In *IEEE Trans. on Automatic Control*, **AC-35(9):1085-1088, 1990.**
- [60] R.E. Kalman and J.E. Bertram. A unified approach to the theory of sampling systems. *Technical report, J. Franklin Institute, 1959.*
- [61] J. Sklansky and J.R. Ragazzini. Analysis of errors in sampled-data feedback systems. *AIEE Transactions*, **74(II): 65-71, 1955.**
- [62] G.M. Kranc. Input-output analysis of multirate feedback systems. In *IEEE Transactions on Automatic Control*, **AC-3:21-28, 1957.**
- [63] T.C. Co[®]ey and I.J. Williams. Stability analisis of multiloop, multirate sampled systems. In *AIAA Journal Guidance Control Dynamics*, **14(12):2178-2190, 1966.**
- [64] P.P. Khargonekar, K. Poolla, and A. Tannenbaum. Robust control of linear time-invariant plants using periodic compensation. In *IEEE Transactions on Automatic Control*, **AC-30:1088-1985, 1985.**

- [65] P.M. Thompson. Gain and phase margins of multirate sampled-data feedback systems. *International Journal Control*, 44(3):833-846, 1986.
- [66] P.M. Thompson. Program CC, Ver. 4: Reference Manual, Tutorial and User's Guide. *Systems Technology Inc., Hawthorne, CA., 1988.*
- [67] J. Tornero. Non-conventional sampled-data systems modelling. *Control System Center Report 640/1985, University of Manchester (UMIST), 1985.*
- [68] S. Longhi. Structural properties of multirate sampled systems. *IEEE Trans. on Automatic Control*, 39(3):692-696, 1994.
- [69] J. Tornero and M. Tomizuka. Dual-rate high order hold equivalent controllers. In *American Control Conference*, pages 175-179, 2000.
- [70] J. Tornero and M. Tomizuka. Modeling, analysis and design tools for dual-rate systems. In *American Control Conference*, pages 411-4121, 2002.
- [71] J. Tornero, M. Tomizuka, C. Camina, E. Ballester, and R. Piza. Design of dual-rate PID controllers. In *Proceedings of the IEEE International Conference on Control Applications*, pages 859-865. *IFAC Press, 2001.*
- [72] J. Tornero, M. Tomizuka. Analysis and Design of Dual-rate High Order Hold Equivalent Controllers.
- [73] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3D robotics Hand-Eye calibration. In *IEEE transactions on robotics and automation*, Vol. 5 Nº 3, Junio 1989.
- [74] J. Feddema and O. Mitchel. Vision-guided servoing with feature-based trajectory generation. In *IEEE transactions on robotics and automation*. 5, 691-700, 1989.
- [75] D. Forsyth, J. Ponce. Computer Vision: A modern approach. In *Prentice Hall, Upper Saddle River 2003.*
- [76] Y. Ma, S. Soatto, J. Kosecka and S.Sastry. An invitation to 3-D Vision: From Images to Geometric Models. In *Springer, New York, 2003.*
- [77] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *IEEE transactions on robotics and automation*, New York, 2004.
- [78] P. Corke and S. Hutchinson. A new partitioned approach to image-based servo control. In *IEEE transactions on robotics and automation*, 17, 507-515, 2001.

-
- [79] P. Corke. The Machine Vision Toolbox: a MATLAB toolbox for vision and vision-based control. In *IEEE Robotics & Automation Magazine*, Diciembre, 2005.
- [80] JaViSS - Java-based Visual Servo Simulator.
www.robot.uji.es/research/projects/javiss
- [81] E. Marchand, F. Spindler, F. Chaumette. ViSP for visual servoing: a generic software platform with a wide class of robot control skills. In *IEEE Robotics and Automation Magazine, Special Issue on "Software Packages for Vision-Based Control of Motion"*, P. Oh, D. Burschka (Eds.), 12(4):40-52, December 2005.
- [82] P. Corke. A computer tool for simulation and analysis: the Robotics Toolbox for MATLAB. In *Proceedings of the 1995 National Conference of the Australian Robot Association, Melbourne, Australia, pp 319-330, July, 1995.*
- [83] J. Denavit and R.S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. In *Trans ASME J. Appl. Mech*, 23:215-221,1955.
- [84] Z. Zhang. Flexible Camera Calibration by viewing a plane from unknown orientations. In *Microsoft Research, One Way, Remond, WA 98052-6399, USA, 1999.*
- [85] J. Heikkilä and O. Silvén. A Four-step Camera Calibration Procedure with Implicit Image Correction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Puerto Rico, 1997.*
- [86] R.Y. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses - In *IEEE J. Robotics Automat.* , pages 323-344, Vol. RA-3, No. 4, 1987.
- [87] P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. . In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, USA, 1999.*
- [88] Leopoldo Armesto. Técnicas de control y fusión sensorial Multifrecuenciales y su aplicación a la robótica móvil. *Tesis doctoral Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia, 2005.*

ANEXOS

ANEXO 1. Ejemplo Lineal

En esta sección se va a realizar un estudio en un ejemplo lineal sencillo de todos los conceptos del diseño multifrecuencia vistos en el Capítulo 2. Para ello se va a emplear el ejemplo de un doble integrador como planta del sistema.

$$G_p(s) = \frac{1}{s^2} \quad (1)$$

Por razones técnicas, el periodo máximo de muestreo es de $\bar{T} = 1.5 \text{ seg}$.

Una vez definido el problema, lo primero que se debe hacer es obtener el modelo en espacio de estados de la planta. Por lo tanto se tiene:

$$\left. \begin{aligned} x &= x_0 + v_0 \cdot t + a \cdot \frac{t^2}{2} \\ v &= v_0 + a \cdot t \end{aligned} \right\} \quad (2)$$

$$\left. \begin{aligned} \dot{x} &= \frac{dx}{dt} = v \\ \dot{v} &= \frac{dv}{dt} = a \end{aligned} \right\} \quad (3)$$

Combinando las ecuaciones 2 y 3 se obtiene:

$$\left. \begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a \\ x &= [1 \quad 0] \cdot \begin{bmatrix} x \\ y \end{bmatrix} + 0 \cdot a \end{aligned} \right\} \quad (4)$$

Por último, si discretiza el modelo a periodo T :

$$\left. \begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix}_{k+1} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} a_k \\ x_k &= [1 \quad 0] \cdot \begin{bmatrix} x \\ y \end{bmatrix}_k + 0 \cdot a_k \end{aligned} \right\} \quad (5)$$

Llegados a este punto, lo primero que se va a mostrar es cómo se obtienen los modelos N-delay input y N-delay output de la planta y se verá la su equivalencia. Una vez obtenidos los modelos se verá cómo trabajar con los retenedores de orden alto bifrecuencia y se hará un análisis de la estabilidad en función del orden de multiplicidad N . Por último se introducirá un control multifrecuencia viendo las mejoras que este enfoque conlleva en el comportamiento fina del sistema.

- Modelo N-delay output

En la Figura 1 se puede ver la idea fundamental de este enfoque. Lo que se pretende es, a partir de una entrada el sistema tiene el número de salidas proporcionado por el orden de multiplicidad que se le asigna.

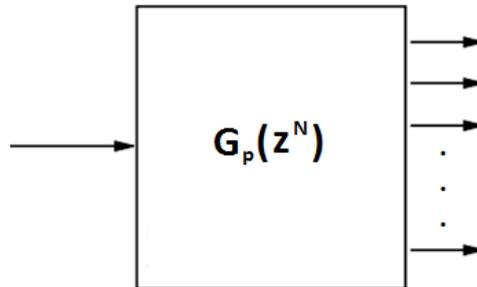


Figura 1. Modelo de la planta N-delay output.

Para este caso, se tiene que el vector función de transferencia de la planta es:

$$\begin{bmatrix} Y_0(z^N) \\ \vdots \\ Y_{N-1}(z^N) \end{bmatrix} = \begin{bmatrix} G_{p,0}(z^N) \\ \vdots \\ G_{p,N-1}(z^N) \end{bmatrix} \cdot U(z^N) \quad (6)$$

$$\begin{bmatrix} G_{p,0}(z^N) \\ \vdots \\ G_{p,N-1}(z^N) \end{bmatrix} = \begin{bmatrix} G_{0,0}(z^N) & \cdots & G_{0,N-1}(z^N) \\ \vdots & & \vdots \\ G_{N-1,0}(z^N) & \cdots & G_{N-1,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} G_{h,0}(z^N) \\ \vdots \\ G_{h,N-1}(z^N) \end{bmatrix} \quad (7)$$

Ahora se obtendrá el modelo “estirado” de la planta (ver ecuación 8). Para ello, lo primero que se va a hacer es obtener dicho modelo utilizando $N = 2$.

$$\left. \begin{aligned} G_{i,j}(z^N) &= C \cdot A^i (z^N \cdot I - A^N)^{-1} \cdot A^{N-j-1} \cdot B + D_{i,j} \\ D_{i,j} &= \begin{cases} C \cdot A^{i-j-1} \cdot B \rightarrow i > j \\ D \rightarrow i = j \\ 0 \rightarrow \text{other wise} \end{cases} \end{aligned} \right\} \quad (8)$$

Por lo tanto, aplicando la ecuación 8 al modelo en espacio de estados de la planta, se obtienen las siguientes funciones de transferencia:

$$\begin{aligned}
 \overline{G_{0,0}(z^2)} &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^{-1} \left(\begin{bmatrix} z^2 & 0 \\ 0 & z^2 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + 0 = \\
 &= [1 \ 0] \cdot \begin{bmatrix} \frac{1}{z^2-1} & \frac{2 \cdot T}{(z^2-1)^2} \\ 0 & \frac{1}{z^2-1} \end{bmatrix} \cdot \begin{bmatrix} 3 \cdot T^2/2 \\ T \end{bmatrix} = \\
 &= \frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2 - 1)^2}
 \end{aligned} \tag{9}$$

$$\begin{aligned}
 \overline{G_{0,1}(z^2)} &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^{-1} \left(\begin{bmatrix} z^2 & 0 \\ 0 & z^2 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + 0 = \\
 &= [1 \ 0] \cdot \begin{bmatrix} \frac{1}{z^2-1} & \frac{2 \cdot T}{(z^2-1)^2} \\ 0 & \frac{1}{z^2-1} \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = \\
 &= \frac{T^2 \cdot (z^2 + 3)}{(z^2 - 1)^2}
 \end{aligned} \tag{10}$$

$$\begin{aligned}
 \overline{G_{1,0}(z^2)} &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} z^2 & 0 \\ 0 & z^2 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + \\
 &\quad + [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = \\
 &= [1 \ T] \cdot \begin{bmatrix} \frac{1}{z^2-1} & \frac{2 \cdot T}{(z^2-1)^2} \\ 0 & \frac{1}{z^2-1} \end{bmatrix} \cdot \begin{bmatrix} 3 \cdot T^2/2 \\ T \end{bmatrix} + \frac{T^2}{2} = \\
 &= \frac{T^2 \cdot (z^4 + 3 \cdot z^2)}{(z^2 - 1)^2}
 \end{aligned} \tag{11}$$

$$\left. \begin{aligned}
 \overline{G_{1,1}}(z^2) &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} z^2 & 0 \\ 0 & z^2 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + 0 = \\
 &= [1 \ T] \cdot \begin{bmatrix} \frac{1}{z^2-1} & \frac{2 \cdot T}{(z^2-1)^2} \\ 0 & \frac{1}{z^2-1} \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = \\
 &= \frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2-1)^2}
 \end{aligned} \right\} \quad (13)$$

Las funciones de transferencia anteriores se pueden mostrar en forma matricial tal y como se ve en la Ecuación 14.

$$\left. \begin{aligned}
 \begin{bmatrix} G_{p,0}(z^2) \\ G_{p,1}(z^2) \end{bmatrix} &= \begin{bmatrix} \frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2-1)^2} & \frac{T^2 \cdot (z^2 + 3)}{(z^2-1)^2} \\ \frac{T^2 \cdot (z^4 + 3 \cdot z^2)}{(z^2-1)^2} & \frac{T^2 \cdot (3 \cdot z^2 + 1)^2}{(z^2-1)^2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{ZOH} = \\
 &= \begin{bmatrix} \frac{4 \cdot \frac{T^2}{2} \cdot (z^2 + 1)}{(z^2-1)^2} \\ \frac{T^2 \cdot (z^4 + 3 \cdot z^2 + 1)}{(z^2-1)^2} \end{bmatrix}
 \end{aligned} \right\} \quad (14)$$

Como muestra el modelo de la Ecuación 14, se tiene el modelo de la planta 2-delay output, que significa que a partir de una entrada se tienen dos salidas. Ahora se hará una sencilla demostración para determinar si este modelo es correcto, esto es, si el comportamiento del sistema, utilizando el monofrecuencia inicial y utilizando el 2-delay output es el mismo. Para ello, se puede demostrar de la siguiente forma:

$$G_{p,0}(z^2) = \frac{4 \cdot \frac{T^2}{2} \cdot (z^2 + 1)}{(z^2-1)^2} = \frac{4 \cdot \frac{T^2}{2} \cdot (z^{-2} + z^{-4})}{1 - 2 \cdot z^{-2} + z^{-4}} \quad (15)$$

$$y(2k+0) = 2 \cdot y(2(k-1)+0) - y(2(k-2)+0) + 4 \cdot \frac{T^2}{2} u(2(k-1)+0) + \left. \begin{aligned} &+ 4 \cdot \frac{T^2}{2} u(2(k-2)+0) \end{aligned} \right\} \quad (16)$$

$$G_{p,1}(z^2) = \frac{\frac{T^2}{2} \cdot (z^4 + 6 \cdot z^2 + 1)}{(z^2 - 1)^2} = \frac{\frac{T^2}{2} \cdot (1 + 6 \cdot z^{-2} + z^{-4})}{1 - 2 \cdot z^{-2} + z^{-4}} \quad (17)$$

$$y(2k+1) = 2 \cdot y(2(k-1)+1) - y(2(k-2)+1) + \frac{T^2}{2} u(2k+1) + \left. \begin{aligned} &+ 6 \cdot \frac{T^2}{2} u(2(k-1)+1) + \frac{T^2}{2} u(2(k-2)+1) \end{aligned} \right\} \quad (18)$$

Las ecuaciones 16 y 18 proporcionan las secuencias de salida en función de la entrada del modelo.

$$\left. \begin{aligned} y(2k+0) &= 2 \cdot y(2(k-1)+0) - y(2(k-2)+0) + 4 \cdot \frac{T^2}{2} u(2(k-1)+0) + \\ &+ 4 \cdot \frac{T^2}{2} u(2(k-2)+0) \\ y(2k+1) &= 2 \cdot y(2(k-1)+1) - y(2(k-2)+1) + \frac{T^2}{2} u(2k+1) + \\ &+ 6 \cdot \frac{T^2}{2} u(2(k-1)+1) + \frac{T^2}{2} u(2(k-2)+1) \end{aligned} \right\} \quad (19)$$

Ahora, empleando la siguiente señal de entrada al sistema:

$$u(k) = \{u(0), u(1), \dots, u(m)\} = \left\{ \underbrace{1, 1, \dots, 1}_{m+1} \right\} \quad (20)$$

Por lo tanto se tiene la siguiente secuencia de salida:

$$\left. \begin{aligned}
 k=0 & \left\{ \begin{aligned}
 y(0) &= 2 \cdot y(-2) - y(-4) + 4 \cdot \frac{T^2}{2} (u(-2) + u(-4)) = 0 \rightarrow t=0 \\
 y(1) &= 2 \cdot y(-1) - y(-3) + \frac{T^2}{2} (u(1) + 6 \cdot u(-1) + u(-3)) = \frac{T^2}{2} \rightarrow t=1
 \end{aligned} \right. \\
 k=1 & \left\{ \begin{aligned}
 y(2) &= 2 \cdot y(0) - y(-2) + 4 \cdot \frac{T^2}{2} (u(0) + u(-2)) = 4 \cdot \frac{T^2}{2} \rightarrow t=2 \\
 y(3) &= 2 \cdot y(1) - y(-1) + \frac{T^2}{2} (u(3) + 6 \cdot u(1) + u(-1)) = 9 \cdot \frac{T^2}{2} \rightarrow t=3
 \end{aligned} \right. \\
 k=2 & \left\{ \begin{aligned}
 y(4) &= 2 \cdot y(2) - y(0) + 4 \cdot \frac{T^2}{2} (u(2) + u(0)) = 16 \cdot \frac{T^2}{2} \rightarrow t=4 \\
 y(5) &= 2 \cdot y(3) - y(1) + \frac{T^2}{2} (u(5) + 6 \cdot u(3) + u(1)) = 25 \cdot \frac{T^2}{2} \rightarrow t=5
 \end{aligned} \right.
 \end{aligned} \right\} \quad (21)$$

De forma más compacta, se tiene:

$$y(t) = \{y(0), y(1), \dots, y(m)\} = \left\{ 0, \underbrace{\frac{T^2}{2}, 4 \cdot \frac{T^2}{2}, 9 \cdot \frac{T^2}{2}, 16 \cdot \frac{T^2}{2}, 25 \cdot \frac{T^2}{2}, \dots}_{m+1} \right\} \quad (22)$$

La Figura 2 muestra una comparativa entre las salidas del modelo SISO y el modelo N-delay output de la planta. Como se demuestra, ambos son equivalentes.

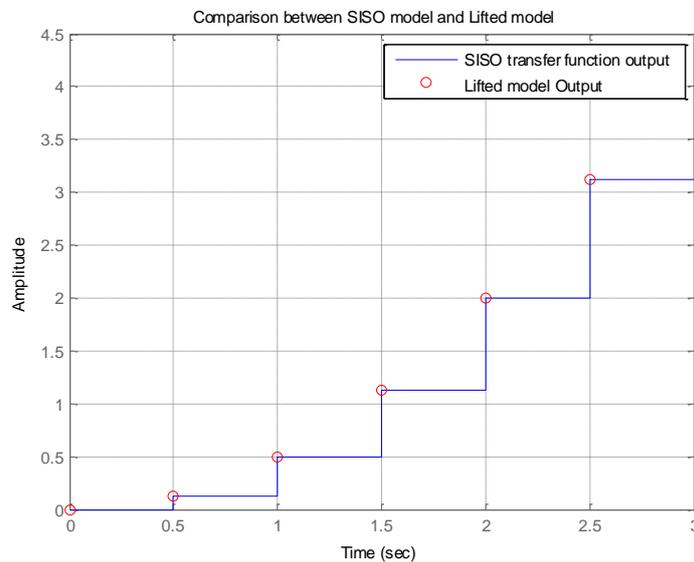


Figura 2. Comparación entre los modelos SISO y N-delay output de la planta con orden de multiplicidad 2.

- Modelo N-delay input

Siguiendo el mismo procedimiento, se puede obtener el modelo de la planta N-delay input. La Figura 3, muestra la idea de este modelo. Se trata de un sistema que recibe N entradas y proporciona una única salida.

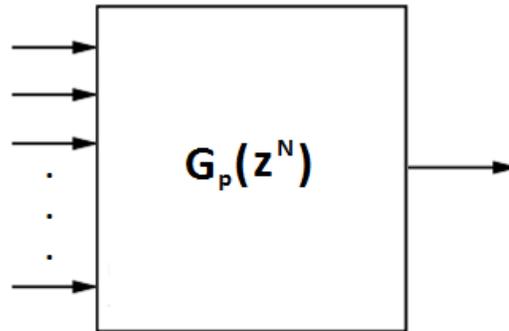


Figura 3. Modelo de la planta N-delay input.

En este caso, el vector función de transferencia de la planta es:

$$\begin{bmatrix} Y_0(z^N) \\ \vdots \\ Y_{N-1}(z^N) \end{bmatrix} = \begin{bmatrix} G_{p,0}(z^N) & \cdots & G_{p,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} U_0(z^N) \\ \vdots \\ U_{N-1}(z^N) \end{bmatrix} \quad (23)$$

$$\begin{bmatrix} G_{p,0}(z^N) \\ \vdots \\ G_{p,N-1}(z^N) \end{bmatrix}^T = \begin{bmatrix} G_{s,0}(z^N) & \cdots & G_{s,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} G_{0,0}(z^N) & \cdots & G_{0,N-1}(z^N) \\ \vdots & & \vdots \\ G_{N-1,0}(z^N) & \cdots & G_{N-1,N-1}(z^N) \end{bmatrix} \quad (24)$$

Ahora, utilizando el modelo “estirado” calculado en la sección anterior y incorporando un muestreador se obtienen los siguientes resultados:

$$\begin{bmatrix} G_{p,0}(z^2) \\ G_{p,1}(z^2) \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \end{bmatrix}_{SOH} \cdot \begin{bmatrix} \frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2 - 1)^2} & \frac{T^2 \cdot (z^2 + 3)}{(z^2 - 1)^2} \\ \frac{T^2 \cdot (z^4 + 3 \cdot z^2)}{(\bar{z}^2 - 1)^2} & \frac{T^2 \cdot (3 \cdot z^2 + 1)^2}{(\bar{z}^2 - 1)^2} \end{bmatrix} = \begin{bmatrix} \frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2 - 1)^2} \\ \frac{T^2 \cdot (\bar{z}^2 + 3)}{(\bar{z}^2 - 1)^2} \end{bmatrix}^T \quad (25)$$

Como se ha mencionado anteriormente y visto en el capítulo 2, para obtener la función de transferencia se puede aplicar cualquiera de los siguientes resultados:

$$G(z) = G_{0,0}(z^N) + \sum_{j=1}^{N-1} G_{0,j}(z^N) \cdot z^j \quad (26)$$

$$G(z) = G_{0,0}(z^N) + \sum_{i=1}^{N-1} G_{i,0}(z^N) \cdot z^{-i} \quad (27)$$

Por lo tanto, en este caso se tiene:

$$\left. \begin{aligned} \overline{G(z)} &= \frac{\frac{T^2}{2} \cdot (3 \cdot z^2 + 1)}{(z^2 - 1)^2} + \frac{\frac{T^2}{2} \cdot (z^2 + 3)}{(z^2 - 1)^2} \cdot z = \\ &= \frac{\frac{T^2}{2} \cdot (3 \cdot z^2 + 1) + \frac{T^2}{2} \cdot (z^3 + 3 \cdot z)}{(z^2 - 1)^2} = \frac{\frac{T^2}{2} \cdot (z^3 + 3 \cdot z^2 + 3 \cdot z + 1)}{(z^2 - 1)^2} \end{aligned} \right\} \quad (28)$$

La anterior función de transferencia se puede colocar en la forma de ecuación en diferencias tal y como se muestra a continuación:

$$y(2k) = 2 \times y(2(k-1)) - y(2(k-2)) + \frac{T^2}{2} \times \left[u(2(k-1)) + 3 \times u(2(k-2)) + \right. \\ \left. + 3 \times u(2(k-3)) + u(2(k-4)) \right] \quad (29)$$

Empleando la misma secuencia de control se tiene:

$$\begin{aligned} k=0 & \left\{ y(0) = 2 \cdot y(-2) - y(-4) + \frac{T^2}{2} (u(-1) + 3 \cdot u(-2) + 3 \cdot u(-3) + u(-4)) = 0 \rightarrow t=0 \right. \\ k=1 & \left\{ y(1) = 2 \cdot y(-1) - y(-3) + \frac{T^2}{2} (u(0) + 3 \cdot u(-1) + 3 \cdot u(-2) + u(-3)) = \frac{T^2}{2} \rightarrow t=1 \right. \\ k=2 & \left\{ y(2) = 2 \cdot y(0) - y(-2) + \frac{T^2}{2} (u(1) + 3 \cdot u(0) + 3 \cdot u(-1) + u(-2)) = 4 \cdot \frac{T^2}{2} \rightarrow t=2 \right. \\ k=3 & \left\{ y(3) = 2 \cdot y(1) - y(-1) + \frac{T^2}{2} (u(2) + 3 \cdot u(1) + 3 \cdot u(0) + u(-1)) = 9 \cdot \frac{T^2}{2} \rightarrow t=3 \right. \\ k=4 & \left\{ y(4) = 2 \cdot y(2) - y(0) + \frac{T^2}{2} (u(3) + 3 \cdot u(2) + 3 \cdot u(1) + u(0)) = 16 \cdot \frac{T^2}{2} \rightarrow t=4 \right. \\ k=5 & \left\{ y(5) = 2 \cdot y(3) - y(1) + \frac{T^2}{2} (u(4) + 3 \cdot u(3) + 3 \cdot u(2) + u(1)) = 25 \cdot \frac{T^2}{2} \rightarrow t=5 \right. \end{aligned}$$

Como se puede ver haciendo una comparativa con el caso anterior, la secuencia es equivalente y se confirma con la simulación que muestra la Figura 4.

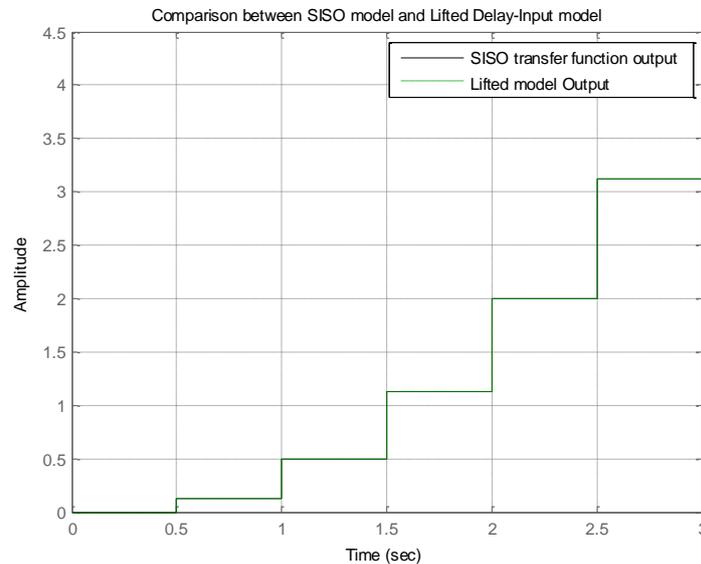


Figura 4. Comparativa entre los modelos SISO y N-delay input de la planta para N=2.

COMPORTAMIENTO DE LA DINÁMICA DEL SISTEMA AL INCLUIR DR-HOH.

Es interesante estudiar el comportamiento dinámico del sistema cuando se introducen los retenedores de orden alto bi-frecuencia. Esto significa que se tiene que estudiar dicho comportamiento a partir del modelo de la planta N-delay input, ya que, como se ha visto, los retenedores de orden alto bi-frecuencia a partir de una entrada proporcionan N salidas. Por lo tanto el modelo de la planta debe permitir la entrada de N señales y proporcionar una salida.

	VECTOR FUNCIÓN DE TRANSFERENCIA	ALGORITMO DE CONTROL A T
IMPULSO	$G_{h,i}(z^N) = \begin{cases} N & \text{iff } i=0 \\ 0 & \text{otherwise} \end{cases}$	$e(k, i) = e(k, 0) \cdot \frac{1}{T} \cdot \delta_i$
ZOH	$G_{h,i}(z^N) = 1$	$\left(\frac{\bar{z}-1}{\bar{z}}\right) \cdot Z_{\bar{T}} \left[\frac{G_p(s)}{s} \right] \cdot G_c(\bar{z})$
FOH	$G_{h,i}(z^N) = \frac{\left(\frac{N+i}{N}\right)z^N - \left(\frac{i}{N}\right)}{z^N}$	$\left(\frac{\bar{z}-1}{\bar{z}}\right)^2 \cdot Z_{\bar{T}} \left[\frac{G(s)}{s} + \frac{G(s)}{\bar{T}s^2} \right] \cdot G_c(\bar{z})$
TH	$G_{h,i}(z^N) = \frac{\left(\frac{i}{N}\right)z^N + \left(\frac{N-i}{N}\right)}{z^N}$	$\left(\frac{\bar{z}-1}{\bar{z}}\right)^2 \cdot Z_{\bar{T}} \left[\frac{G(s)}{\bar{T}s^2} \right] \cdot G_c(\bar{z})$

Tabla 1. Retenedores de orden alto bi-frecuencia (DR-HOH).

También es interesante analizar el efecto de los retenedores bi-frecuencia de orden alto en sistemas continuos cuando el valor de $N \rightarrow \infty$. Esto es dado un periodo constante $\bar{T} = N \cdot T$ correspondiente al de baja frecuencia siendo así $\bar{z} = e^{\bar{T}s}$. La Tabla 2 muestra las situaciones límite sin controlador para ver solamente el efecto dinámico de los retenedores sobre la planta.

	$G_h(s)$ at \bar{T}	FUNCIÓN DE TRANSFERENCIA EN BUCLE ABIERTO CON EL ENFOQUE DE ESTIMACIÓN DE LA SEÑAL DE CONTROL PARA $N \rightarrow \infty$
IMPULSO	1	$Z_{\bar{T}} [G_p(s)]$
ZOH	$\frac{1 - e^{-\bar{T}s}}{s}$	$\left(\frac{\bar{z}-1}{\bar{z}}\right) \cdot Z_{\bar{T}} \left[\frac{G_p(s)}{s} \right]$
FOH	$(1 - e^{-\bar{T}s})^2 \cdot \left(\frac{1}{s} + \frac{1}{\bar{T}s^2}\right)$	$\left(\frac{\bar{z}-1}{\bar{z}}\right)^2 \cdot Z_{\bar{T}} \left[\frac{G(s)}{s} + \frac{G(s)}{\bar{T}s^2} \right]$
TH	$(1 - e^{-\bar{T}s})^2 \cdot \left(\frac{1}{\bar{T}s^2}\right)$	$\left(\frac{\bar{z}-1}{\bar{z}}\right)^2 \cdot Z_{\bar{T}} \left[\frac{G(s)}{\bar{T}s^2} \right]$

Tabla 2. Función de transferencia en bucle cerrado sin tener en cuenta el controlador.

Ahora se está en disposición de ver cómo afectan cada uno de los DR-HOH en este problema concreto.

- Retenedor de impulso bi-frecuencia (DR-IH): Aplicando las ecuaciones proporcionadas en la tabla 4 se tiene que,

$$\left| \overline{G(\bar{z})} \right| = Z_{\bar{T}} \left(\frac{1}{s^2} \right) = \frac{\bar{T} \cdot \bar{z}^{-1}}{(1 - \bar{z}^{-1})^2} = \frac{\bar{T} \cdot \bar{z}}{(\bar{z} - 1)^2} \quad (30)$$

Y, particularizando cuando $\bar{T} = 1.5$ se tiene:

$$\left| \overline{G(\bar{z})} \right| = \frac{1.5 \cdot \bar{z}}{(\bar{z} - 1)^2} \quad (31)$$

Por lo tanto, el lugar de las raíces que se obtiene con este retenedor es:

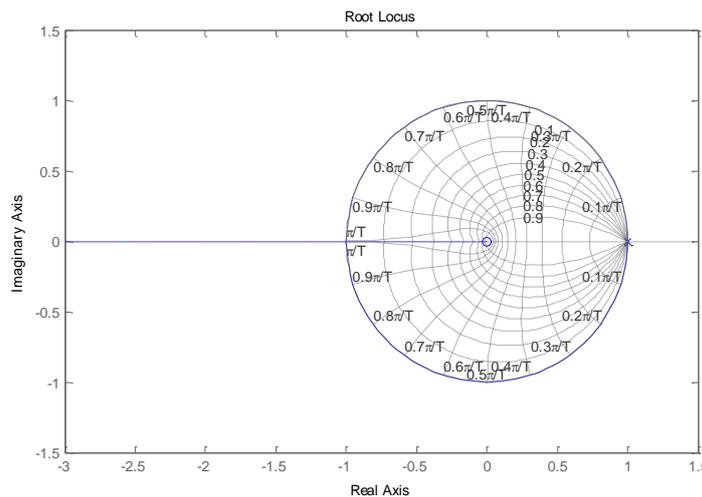


Figura 5. Lugar de las Raíces Multi-frecuencia de la planta con DR-IH.

- Retenedor de cero orden bi-frecuencia (DR-ZOH): En este caso, incluyendo el retenedor de orden cero bi-frecuencia,

$$\left| \overline{G(\bar{z})} \right| = \frac{\bar{z} - 1}{\bar{z}} \cdot Z_{\bar{T}} \left(\frac{1}{s^3} \right) = Z_{\bar{T}} \left(\frac{1}{2} \cdot \frac{2}{s^2} \right) = \frac{\bar{z} - 1}{\bar{z}} \cdot \frac{\bar{T}^2}{2} \cdot \frac{\bar{z}^{-1} \cdot (1 + \bar{z}^{-1})}{(1 - \bar{z}^{-1})^3} = \frac{\bar{T}^2}{2} \cdot \frac{(\bar{z} + 1)}{(\bar{z} - 1)^2} \quad (32)$$

Y para el caso concreto de $\bar{T} = 1.5$,

$$\left| \overline{G(\bar{z})} \right| = 1.125 \cdot \frac{(\bar{z} + 1)}{(\bar{z} - 1)^2} \quad (33)$$

Por lo tanto, el lugar de las raíces toma la siguiente forma:

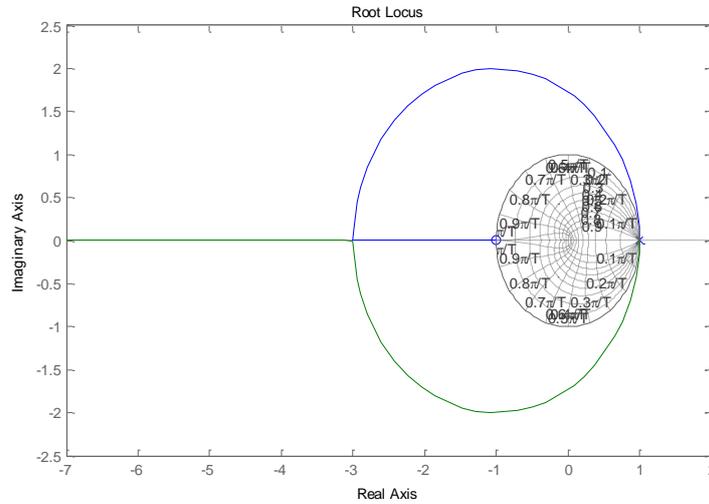


Figura 6. Lugar de las Raíces Multi-frecuencia de la planta con DR-ZOH.

- Retenedor de primer orden bi-frecuencia (DR-FOH): Del mismo modo, operando con un retenedor de primer orden bi-frecuencia se tiene,

$$\begin{aligned}
 \overline{G(\bar{z})} &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot Z_{\bar{T}} \left(\frac{1}{s^3} + \frac{1}{\bar{T} \cdot s^4} \right) = \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot Z_{\bar{T}} \left(\frac{1}{2} \cdot \frac{2}{s^3} + \frac{6}{6 \cdot \bar{T} \cdot s^4} \right) = \\
 &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot \frac{\bar{T}^2}{2} \cdot \left[\frac{\bar{z}^{-1} \cdot (1 + \bar{z}^{-1})}{(1 - \bar{z}^{-1})^3} + \frac{1}{3} \cdot \frac{\bar{z}^{-1} \cdot (1 + 4 \cdot \bar{z}^{-1} + \bar{z}^{-2})}{(1 - \bar{z}^{-1})^4} \right] = \\
 &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot \frac{\bar{T}^2}{2} \cdot z \cdot \left[\frac{2 \cdot \bar{z}^2 + 2 \cdot \bar{z} - 1}{(\bar{z} - 1)^4} \right] = \frac{6 \cdot \bar{T}^2}{2} \cdot \frac{2 \cdot \bar{z}^2 + 2 \cdot \bar{z} - 1}{\bar{z}(\bar{z} - 1)^2}
 \end{aligned} \tag{34}$$

Y para el caso de $\bar{T} = 1.5$:

$$\overline{G(\bar{z})} = 1.5 \cdot \frac{(\bar{z} - 0.366) \cdot (\bar{z} + 1.366)}{\bar{z}(\bar{z} - 1)^2} \tag{35}$$

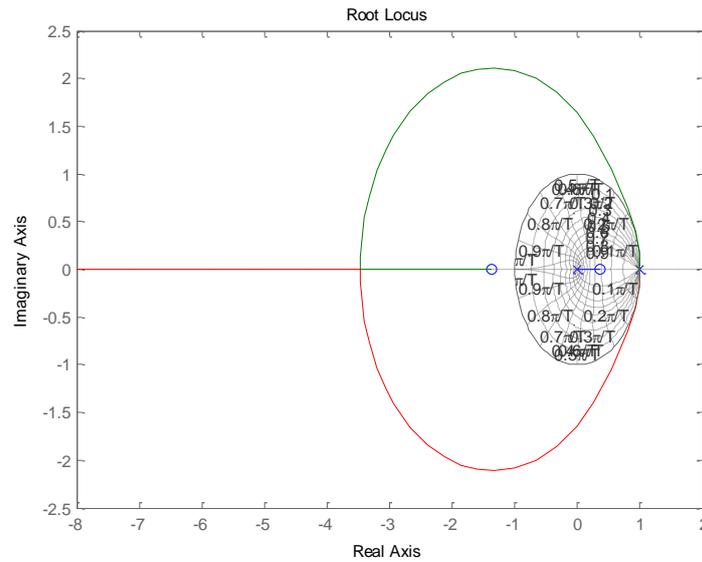


Figura 7. Lugar de las raíces de la planta con DR-FOH.

- Retenedor triangular bi-frecuencia (DR-TH):

$$\left. \begin{aligned} |G(\bar{z})| &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot Z_{\bar{T}} \left(\frac{1}{\bar{T} \cdot s^4} \right) = \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot Z_{\bar{T}} \left(\frac{6}{6 \cdot \bar{T} \cdot s^4} \right) = \\ &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot \frac{\bar{T}^2}{2} \cdot \left[\frac{1}{3} \cdot \frac{\bar{z}^{-1} \cdot (1 + 4 \cdot \bar{z}^{-1} + \bar{z}^{-2})}{(1 - \bar{z}^{-1})^4} \right] = \\ &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot \frac{\bar{T}^2}{6} \cdot z \cdot \left[\frac{\bar{z}^2 + 4 \cdot \bar{z} + 1}{(\bar{z}-1)^4} \right] = \frac{\bar{T}^2}{6} \cdot \frac{\bar{z}^2 + 4 \cdot \bar{z} + 1}{(\bar{z}-1)^2} \end{aligned} \right\} \quad (36)$$

Y para el caso particular de $\bar{T} = 1.5$,

$$|G(\bar{z})| = 0.375 \cdot \frac{\bar{z}^2 + 4 \cdot \bar{z} + 1}{\bar{z} \cdot (\bar{z}-1)^2} \quad (37)$$

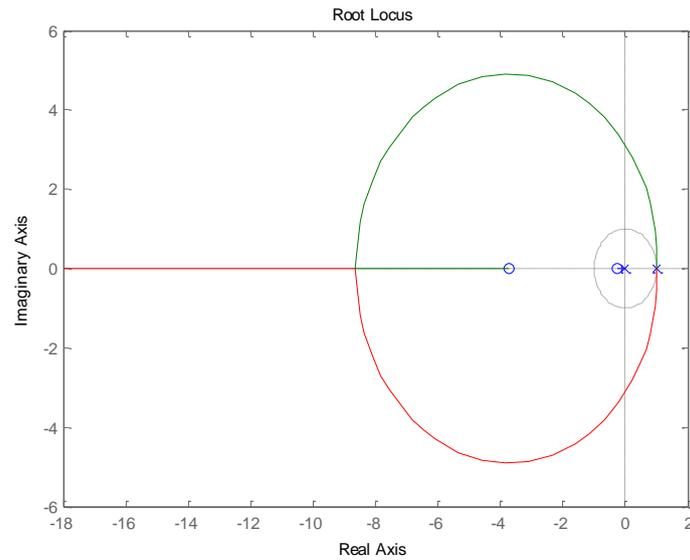


Figura 8. Lugar de las raíces de la planta con un DR-TH.

En la Tabla 5 muestra un resumen de los resultados obtenidos anteriormente con la inclusión en el sistema de los distintos DR-HOH. Además, para realizar una mejor comparativa la Figura 20 muestra el Lugar de las raíces perteneciente a cada uno de los resultados obtenidos.

	FUNCIÓN DE TRANSFERENCIA EN BUCLE ABIERTO CON EL ENFOQUE DE ESTIMACIÓN DE LA SEÑAL DE CONTROL PARA $N \rightarrow \infty$
IMPULSO	$\overline{G(\bar{z})} = \frac{1.5 \cdot \bar{z}}{(\bar{z} - 1)^2}$
ZOH	$\overline{G(\bar{z})} = 1.125 \cdot \frac{(\bar{z} + 1)}{(\bar{z} - 1)^2}$
FOH	$\overline{G(\bar{z})} = 1.5 \cdot \frac{(\bar{z} - 0.366) \cdot (\bar{z} + 1.366)}{\bar{z}(\bar{z} - 1)^2}$
TH	$\overline{G(\bar{z})} = 0.375 \cdot \frac{\bar{z}^2 + 4 \cdot \bar{z} + 1}{\bar{z} \cdot (\bar{z} - 1)^2}$

Tabla 3. Comparativa de la función de transferencia de la planta con $N \rightarrow \infty$ utilizando los distintos DR-HOH.

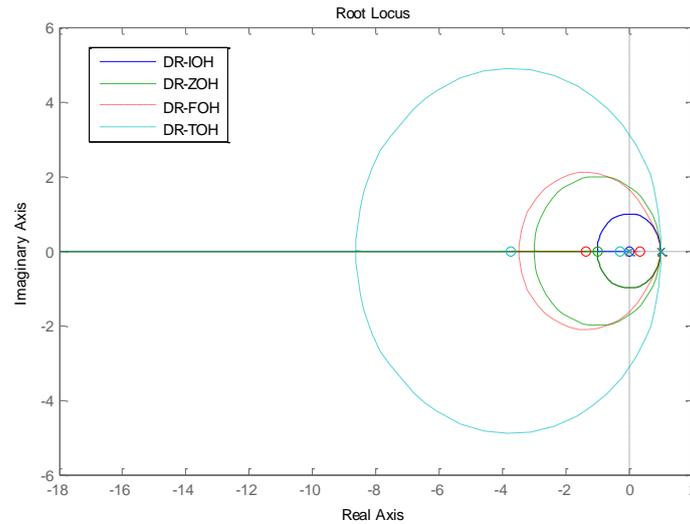


Figura 9. Lugar de las Raíces Multi-frecuencia comparando con los distintos DR-HOH con $N \rightarrow \infty$

Es interesante resaltar que en este enfoque, los retenedores bi-frecuencia Impulso y de cero orden no se utilizan habitualmente ya que no aportan estimaciones correctas. Respecto al primero, no aporta suficiente información y en el segundo caso el modelo de la planta no se modifica y es exactamente el mismo que se obtendría a baja frecuencia mediante discretización.

Para el caso particular del DR-FOH, la función de transferencia de la planta toma la siguiente estructura:

$$G_p(z^N) = \frac{K_p \cdot (z^N - z_1) \cdot (z^N - z_2)}{z^N \cdot (z^N - 1)^2} \quad (38)$$

Donde K_p, z_1, z_2 son parámetros que dependen del orden de multiplicidad N . Para demostrar esta afirmación se va a analizar el sistema para diferentes valores de N .

- $N = 1$
Lo primero que se ha de hacer es obtener el modelo “estirado” de la planta para $N = 1$ utilizando la formulación de la ecuación 8. Con ello se tiene:

$$\left. \begin{aligned} \overline{G_{0,0}}(\overline{z}) &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \left(\begin{bmatrix} \overline{z} & 0 \\ 0 & \overline{z} \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + 0 = \\ &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ \overline{z}-1 & (\overline{z}-1)^2 \\ 0 & \overline{z}-1 \end{bmatrix} \cdot \begin{bmatrix} 3 \cdot T^2/2 \\ T \end{bmatrix} = \frac{T^2 \cdot (\overline{z}+1)}{(\overline{z}-1)^2} \end{aligned} \right\} \quad (39)$$

Entonces, tomando el meta-periodo $\overline{T} = 1.5$, se obtiene:

$$\overline{G_{0,0}}(\overline{z}) = \frac{1.125 \cdot (\overline{z}+1)}{(\overline{z}-1)^2} \quad (40)$$

$$\overline{G_{open-loop}}(\overline{z}) = \frac{1.125 \cdot (\overline{z}+1)}{(\overline{z}-1)^2} \cdot 1 = \frac{1.125 \cdot (\overline{z}+1)}{(\overline{z}-1)^2} \quad (41)$$

En la Figura 10 se puede observar la forma que toma el lugar de las raíces cuando se está trabajando a frecuencia baja.

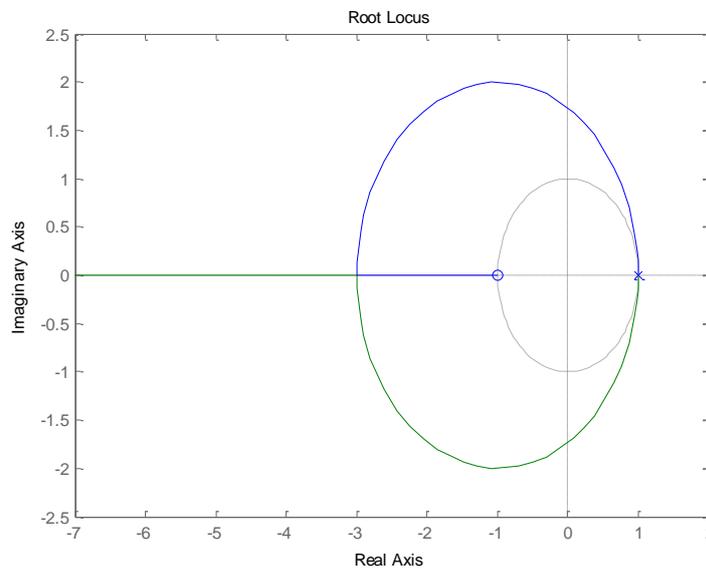


Figura 10. Lugar de las Raíces Multi-frecuencia con N=1 y DR-FOH.

A parte de la forma que adquiere el Lugar de la raíces otro análisis que se va a realizar es el análisis en frecuencia. Las Figuras 11 y 12 muestran el bode del sistema en bucle cerrado.

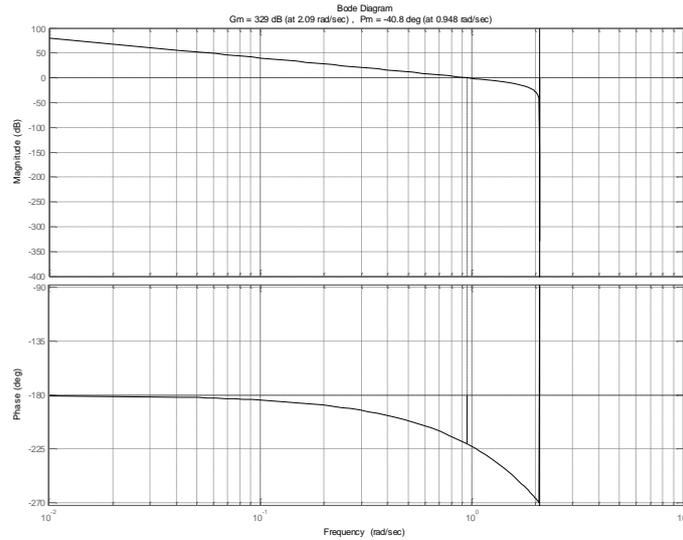


Figura 11. Análisis frecuencial del sistema con N=1.

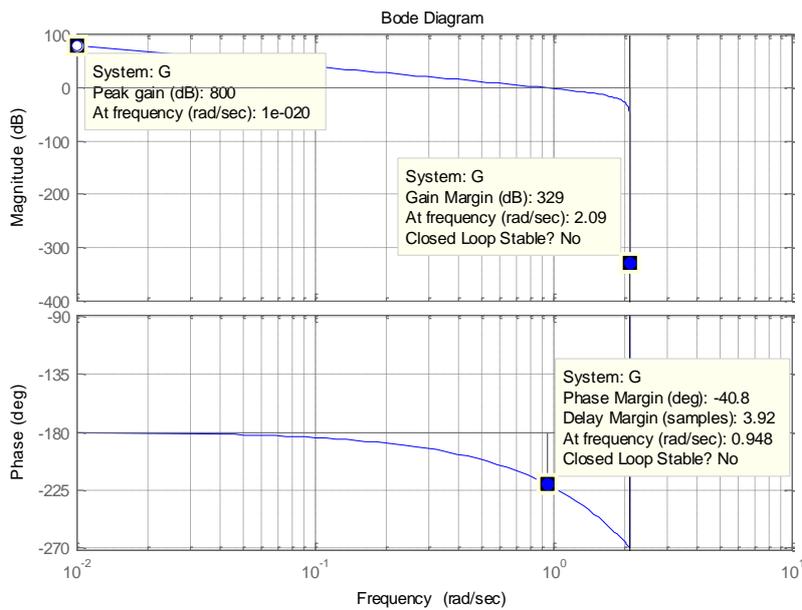


Figura 12. Márgenes de ganancia del sistema para N=1.

Lo que hay que destacar en este caso es que el análisis de frecuencia demuestra que el sistema es inestable en bucle cerrado.

- $N = 2$

Como se ha dicho antes, lo primero que cabe hacer es obtener el modelo “estirado” y, a continuación, el modelo 2-delay input.

$$\begin{bmatrix} G_{p,0}(z^2) \\ G_{p,1}(z^2) \end{bmatrix}^T = [1 \quad 0]_{SOH} \cdot \begin{bmatrix} \frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2 - 1)^2} & \frac{T^2 \cdot (z^2 + 3)}{(z^2 - 1)^2} \\ \frac{T^2 \cdot (z^4 + 3 \cdot z^2)}{(z^2 - 1)^2} & \frac{T^2 \cdot (3 \cdot z^2 + 1)^2}{(z^2 - 1)^2} \end{bmatrix} = \begin{bmatrix} \frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2 - 1)^2} \\ \frac{T^2 \cdot (z^2 + 3)}{(z^2 - 1)^2} \end{bmatrix}^T \quad (42)$$

Como se ha mencionado antes, para obtener la dinámica introducida por un DR-HOH dependiendo del orden de multiplicidad N, hay que aplicar:

$$Y(z^N) = \begin{bmatrix} G_{p,0}(z^N) & \cdots & G_{p,N-1}(z^N) \end{bmatrix} \cdot \begin{bmatrix} G_{h,0}(z^N) \\ \vdots \\ G_{h,N-1}(z^N) \end{bmatrix} \cdot E(z^N) \quad (43)$$

Para el caso de un DR-HOH con N=2, se tiene:

$$\begin{bmatrix} G_{h,0}(z^N) \\ G_{h,1}(z^N) \end{bmatrix} = \begin{bmatrix} \frac{\left(\frac{2+0}{2}\right)z^2 - \left(\frac{0}{2}\right)}{z^2} \\ \frac{\left(\frac{2+1}{2}\right)z^2 - \left(\frac{1}{2}\right)}{z^2} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2} \cdot \frac{3 \cdot z^2 - 1}{z^2} \end{bmatrix} \quad (44)$$

Con ello, se tiene el siguiente modelo del conjunto planta+DR-FOH:

$$\overline{G_{open-loop}}(z) = \left[\frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2 - 1)^2} \quad \frac{T^2 \cdot (z^2 + 3)}{(z^2 - 1)^2} \right] \cdot \begin{bmatrix} 1 \\ \frac{1}{2} \cdot \frac{3 \cdot z^2 - 1}{z^2} \end{bmatrix} = \left. \begin{aligned} &= \frac{T^2 \cdot (3 \cdot z^2 + 1)}{(z^2 - 1)^2} + \frac{T^2 \cdot (z^2 + 3)}{(z^2 - 1)^2} \cdot \frac{1}{2} \cdot \frac{3 \cdot z^2 - 1}{z^2} = \\ &\frac{T^2 \cdot 9 \cdot z^4 + 10 \cdot z^2 - 3}{4 \cdot z^2 \cdot (z^2 - 1)} = \frac{9 \cdot T^2 \cdot (z^2 + 1.3568) \cdot (z^2 - 0.2457)}{4 \cdot z^2 \cdot (z^2 - 1)} \end{aligned} \right\} \quad (45)$$

Y para $T = 0.75$ se tiene:

$$\overline{G_{open-loop}(z^2)} = 1.2656 \cdot \frac{(z^2 + 1.3568) \cdot (z^2 - 0.2457)}{z^2 \cdot (z^2 - 1)^2} \quad (46)$$

Por lo que el Lugar de las Raíces del bucle cerrado $G_{open-loop}(z^2)$ se muestra en la siguiente figura:

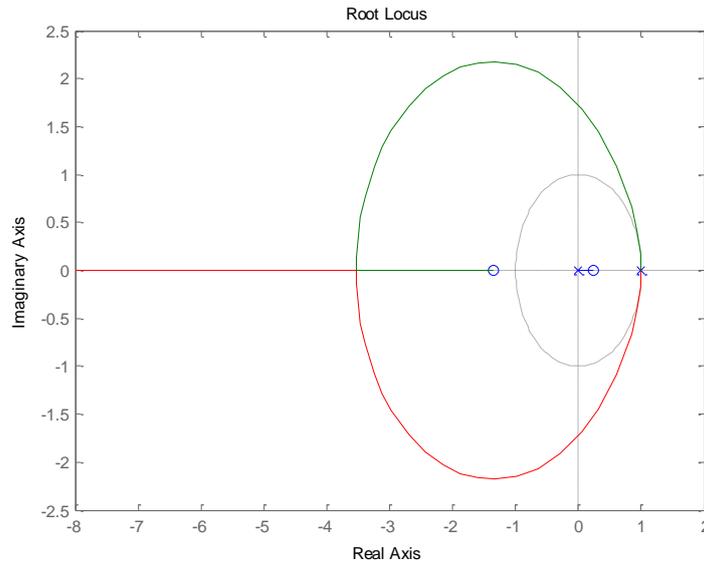


Figura 13. Lugar de las Raíces Multi-frecuencia con N=2 y DR-FOH.

Comparando el resultado anterior con este resultado se puede observar que la dinámica a cambiado ligeramente (ver Figura 14).

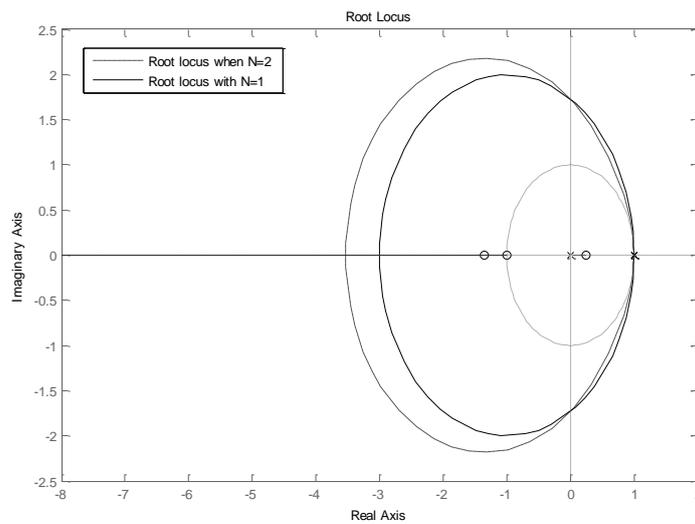


Figura 14. Comparación del Lugar de las Raíces Multifrecuencia con N=1 u N=2 y un DR-FOH.

Como se ha hecho para el caso anterior, las Figuras 13 y 14 muestran el análisis en frecuencia del sistema cuando $N=2$.

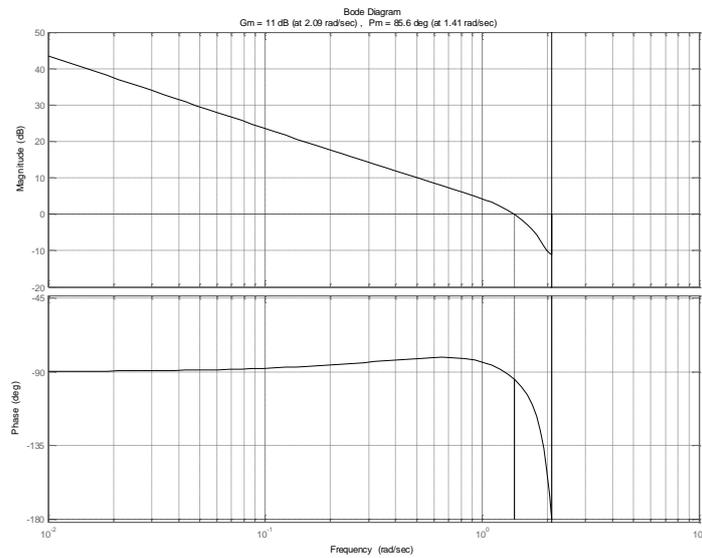


Figura 15. Análisis frecuencial en bucle cerrado para $N=2$.

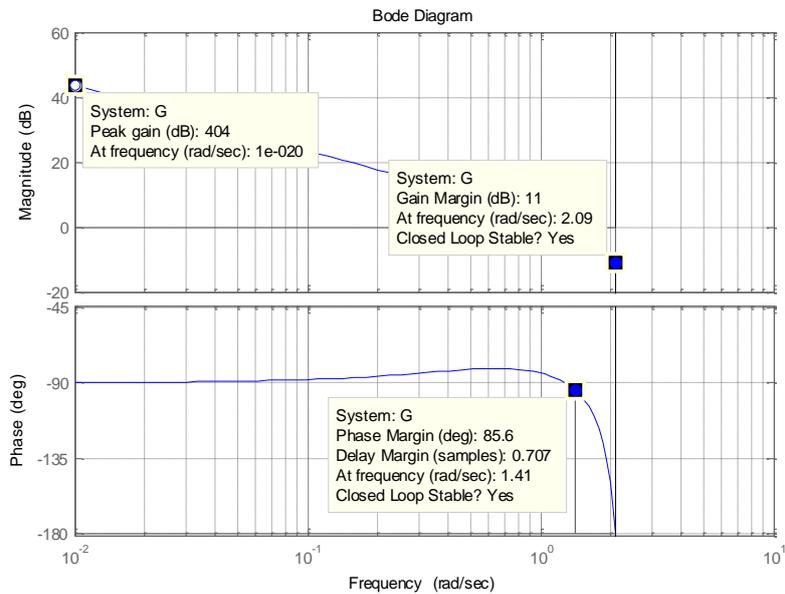


Figura 16. Márgenes de ganancia y de fase en el Bode del sistema con $N=2$.

Lo primero que impacta es que se ha pasado de tener un sistema inestable en bucle cerrado a que el sistema sea estable en bucle cerrado.

Por otro lado, el margen de ganancia es de 2 dB y el margen de fase es de 85.5° .

- $N = 3$

Del mismo modo, que lo realizado en los casos anteriores, lo primero que se tiene que hacer es obtener el modelo “estirado” de la planta para el caso en que $N=3$. Esto es:

$$\left. \begin{aligned} \overline{G_{0,0}}(z^3) &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + 0 = \\ &= [1 \ 0] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} 5 \cdot T^2/2 \\ T \end{bmatrix} = \\ &= \frac{5 \cdot T^2}{2} \cdot \frac{(z^3 + 1/5)}{(z^3 - 1)^2} \end{aligned} \right\} \quad (47)$$

$$\left. \begin{aligned} \overline{G_{0,1}}(z^3) &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + 0 = \\ &= [1 \ 0] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} 3 \cdot T^2/2 \\ T \end{bmatrix} = \\ &= \frac{3 \cdot T^2}{2} \cdot \frac{(z^3 + 1)}{(z^3 - 1)^2} \end{aligned} \right\} \quad (48)$$

$$\left. \begin{aligned} \overline{G_{0,2}}(z^3) &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + 0 = \\ &= [1 \ 0] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = \\ &= \frac{T^2}{2} \cdot \frac{(z^3 + 5)}{(z^3 - 1)^2} \end{aligned} \right\} \quad (49)$$

$$\left. \begin{aligned}
 \overline{G_{1,0}(z^3)} &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + \\
 &+ [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = [1 \ T] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} 5 \cdot T^2/2 \\ T \end{bmatrix} + T^2/2 = \\
 &= \frac{T^2}{2} \cdot \frac{(z^6 + 4 \cdot z^3 + 1)}{(z^3 - 1)^2}
 \end{aligned} \right\} (50)$$

$$\left. \begin{aligned}
 \overline{G_{1,1}(z^3)} &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \\
 &= [1 \ T] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} 3 \cdot T^2/2 \\ T \end{bmatrix} = \frac{5 \cdot T^2}{2} \cdot \frac{(z^3 + 1/5)}{(z^3 - 1)^2}
 \end{aligned} \right\} (51)$$

$$\left. \begin{aligned}
 \overline{G_{1,2}(z^3)} &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + \\
 &= [1 \ T] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = \frac{3 \cdot T^2}{2} \cdot \frac{(z^3 + 1)}{(z^3 - 1)^2}
 \end{aligned} \right\} (52)$$

$$\left. \begin{aligned}
 \overline{G_{2,0}(z^3)} &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + \\
 &+ [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = [1 \ 2 \cdot T] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} 5 \cdot T^2/2 \\ T \end{bmatrix} + 3 \cdot T^2/2 = \\
 &= \frac{3 \cdot T^2}{2} \cdot \frac{(z^6 + 5/3 \cdot z^3)}{(z^3 - 1)^2}
 \end{aligned} \right\} (53)$$

$$\left. \begin{aligned} \overline{G_{2,1}}(z^3) &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + \\ &+ [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = [1 \ 2 \cdot T] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} 3 \cdot T^2/2 \\ T \end{bmatrix} + T^2/2 = \\ &= \frac{T^2 \cdot (z^6 + 5 \cdot z^3 - 4)}{2 \cdot (z^3 - 1)^2} \end{aligned} \right\} \quad (54)$$

$$\left. \begin{aligned} \overline{G_{2,2}}(z^3) &= [1 \ 0] \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^2 \left(\begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^3 \right)^{-1} \cdot \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^0 \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} + 0 = \\ &= [1 \ 2 \cdot T] \cdot \begin{bmatrix} \frac{1}{z^3-1} & \frac{3 \cdot T}{(z^3-1)^2} \\ 0 & \frac{1}{z^3-1} \end{bmatrix} \cdot \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = \frac{5 \cdot T^2}{2} \cdot \frac{(z^3 + 1/5)}{(z^3 - 1)^2} \end{aligned} \right\} \quad (55)$$

Por lo tanto, el modelo estirado de la planta presenta la siguiente forma:

$$G(z^3) = \begin{bmatrix} \frac{5 \cdot T^2}{2} \cdot \frac{(z^3 + 1/5)}{(z^3 - 1)^2} & \frac{3 \cdot T^2}{2} \cdot \frac{(z^3 + 1)}{(z^3 - 1)^2} & \frac{T^2}{2} \cdot \frac{(z^3 + 5)}{(z^3 - 1)^2} \\ \frac{T^2}{2} \cdot \frac{(z^6 + 4 \cdot z^3 + 1)}{(z^3 - 1)^2} & \frac{5 \cdot T^2}{2} \cdot \frac{(z^3 + 1/5)}{(z^3 - 1)^2} & \frac{3 \cdot T^2}{2} \cdot \frac{(z^3 + 1)}{(z^3 - 1)^2} \\ \frac{3 \cdot T^2}{2} \cdot \frac{(z^6 + 5/3 \cdot z^3)}{(z^3 - 1)^2} & \frac{T^2}{2} \cdot \frac{(z^6 + 5 \cdot z^3 - 4)}{(z^3 - 1)^2} & \frac{5 \cdot T^2}{2} \cdot \frac{(z^3 + 1/5)}{(z^3 - 1)^2} \end{bmatrix} \quad (56)$$

Por lo tanto el modelo 3-delay input de la plata es:

$$\begin{aligned}
 \begin{bmatrix} G_{p,0}(z^3) \\ G_{p,1}(z^3) \\ G_{p,2}(z^3) \end{bmatrix}^T &= [1 \ 0 \ 0] \cdot \begin{bmatrix} \frac{5 \cdot T^2 \cdot (z^3 + 1/5)}{(z^3 - 1)^2} & \frac{3 \cdot T^2 \cdot (z^3 + 1)}{(z^3 - 1)^2} & \frac{T^2 \cdot (z^3 + 5)}{(z^3 - 1)^2} \\ \frac{T^2 \cdot (z^6 + 4 \cdot z^3 + 1)}{(z^3 - 1)^2} & \frac{5 \cdot T^2 \cdot (z^3 + 1/5)}{(z^3 - 1)^2} & \frac{3 \cdot T^2 \cdot (z^3 + 1)}{(z^3 - 1)^2} \\ \frac{3 \cdot T^2 \cdot (z^6 + 5/3 \cdot z^3)}{(z^3 - 1)^2} & \frac{T^2 \cdot (z^6 + 5 \cdot z^3 - 4)}{(z^3 - 1)^2} & \frac{5 \cdot T^2 \cdot (z^3 + 1/5)}{(z^3 - 1)^2} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{5 \cdot T^2 \cdot (z^3 + 1/5)}{(z^3 - 1)^2} & \frac{3 \cdot T^2 \cdot (z^3 + 1)}{(z^3 - 1)^2} & \frac{T^2 \cdot (z^3 + 5)}{(z^3 - 1)^2} \end{bmatrix}
 \end{aligned} \tag{57}$$

Para estar convencidos de que el modelo es correcto, se va a realizar la misma prueba que en el caso de N=2.

$$\begin{aligned}
 \overline{G(z)} &= \begin{bmatrix} \frac{5 \cdot T^2 \cdot (z^3 + 1/5)}{(z^3 - 1)^2} & \frac{3 \cdot T^2 \cdot (z^3 + 1)}{(z^3 - 1)^2} & \frac{T^2 \cdot (z^3 + 5)}{(z^3 - 1)^2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ z \\ z^2 \end{bmatrix} = \\
 &= \frac{5 \cdot T^2 \cdot (z^3 + 1/5)}{(z^3 - 1)^2} + \frac{3 \cdot T^2 \cdot (z^3 + 1)}{(z^3 - 1)^2} \cdot z + \frac{T^2 \cdot (z^3 + 5)}{(z^3 - 1)^2} \cdot z^2 = \\
 &= \frac{T^2 \cdot (z^5 + 3 \cdot z^4 + 5 \cdot z^3 + 5 \cdot z^2 + 3 \cdot z + 1)}{(z^2 - 1)^2}
 \end{aligned} \tag{58}$$

La siguiente figura muestra la salida del modelo N-delay input de la planta para N=3, y demuestra que el modelo es correcto ya que se

obtiene la misma señal de salida que para el caso monofrecuencia a baja frecuencia.

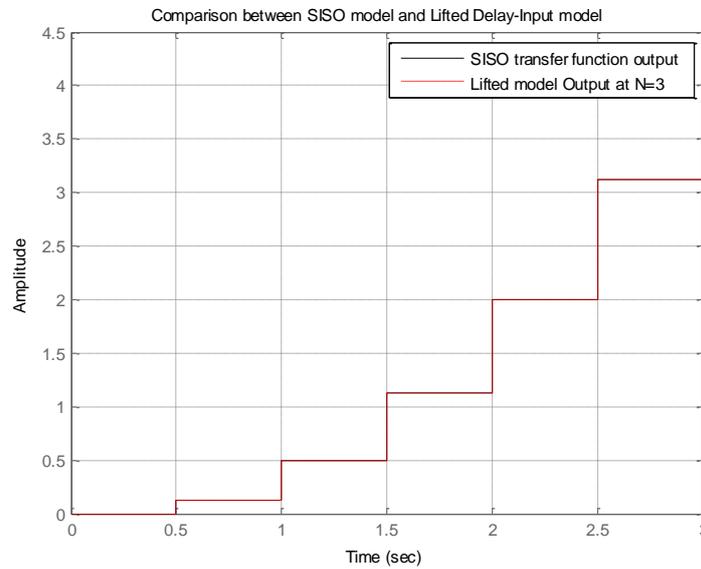


Figura 17. Comparación entre el sistema SISO del modelo de la planta y el sistema 3-delay input.

Llegados a este punto, ya se está listo para analizar el efecto en la dinámica del sistema que provoca la incorporación del DR-FOH. Con tal motivo, utilizando la formulación de la Tabla 2 se tiene la siguiente matriz de funciones de transferencia para el retenedor:

$$\begin{bmatrix} G_{h,0}(z^3) \\ G_{h,1}(z^3) \\ G_{h,2}(z^3) \end{bmatrix} = \begin{bmatrix} \frac{\left(\frac{3+0}{3}\right)z^3 - \left(\frac{0}{3}\right)}{z^3} \\ \frac{\left(\frac{3+1}{3}\right)z^3 - \left(\frac{1}{3}\right)}{z^3} \\ \frac{\left(\frac{3+2}{3}\right)z^3 - \left(\frac{2}{3}\right)}{z^3} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{3} \cdot \frac{4 \cdot z^3 - 1}{z^3} \\ \frac{1}{3} \cdot \frac{5 \cdot z^3 - 2}{z^3} \end{bmatrix} \quad (59)$$

Con ello, la dinámica del sistema DR-FOH + modelo 3-delay input de la planta es:

$$\begin{aligned}
 \overline{G_{open-loop}}(z^3) &= \left[\begin{array}{ccc} \frac{5 \cdot T^2 \cdot (z^3 + 1/5)}{(z^3 - 1)^2} & \frac{3 \cdot T^2 \cdot (z^3 + 1)}{(z^3 - 1)^2} & \frac{T^2 \cdot (z^3 + 5)}{(z^3 - 1)^2} \end{array} \right] \cdot \left[\begin{array}{c} 1 \\ \frac{1}{3} \cdot \frac{4 \cdot z^3 - 1}{z^3} \\ \frac{1}{3} \cdot \frac{5 \cdot z^3 - 2}{z^3} \end{array} \right] = \\
 &= \frac{5 \cdot T^2 \cdot (z^3 + 1/5)}{(z^3 - 1)^2} + \frac{3 \cdot T^2 \cdot (z^3 + 1)}{(z^3 - 1)^2} \cdot \frac{1}{3} \cdot \frac{4 \cdot z^3 - 1}{z^3} + \frac{T^2 \cdot (z^3 + 5)}{(z^3 - 1)^2} \cdot \frac{1}{3} \cdot \frac{5 \cdot z^3 - 2}{z^3} = \\
 &= \frac{T^2 \cdot (32 \cdot z^6 + 35 \cdot z^3 - 13)}{6 \cdot z^2 \cdot (z^2 - 1)} = \frac{32 \cdot T^2 \cdot (z^3 + 1.3867) \cdot (z^3 - 0.2930)}{6 \cdot z^3 \cdot (z^3 - 1)}
 \end{aligned} \tag{60}$$

La siguiente figura muestra el comportamiento del sistema trabajando en bucle cerrado cuando N=3 por medio del lugar de las raíces:

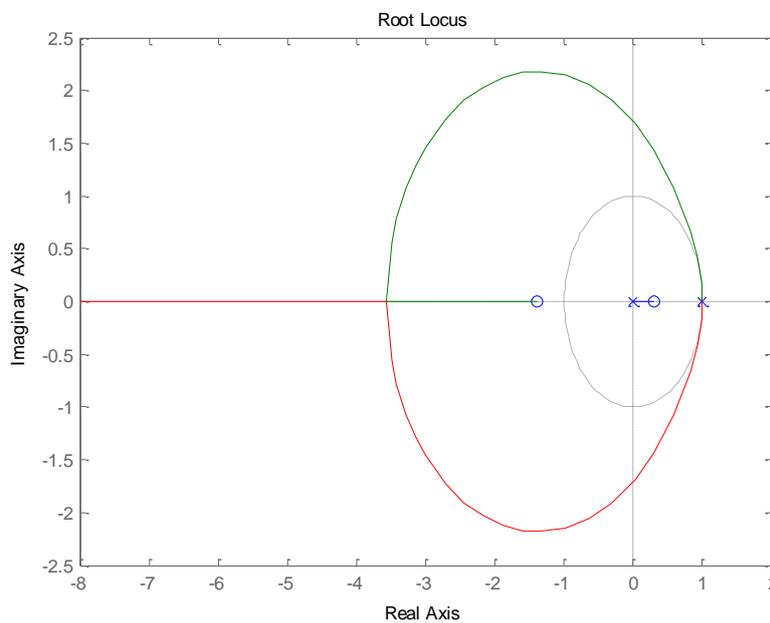


Figura 18. Lugar de las Raíces del sistema Planta + DR-FOH con N=3.

Por lo que respecta al análisis en frecuencia del sistema, los resultados se pueden observar en las Figuras 19 y 20.

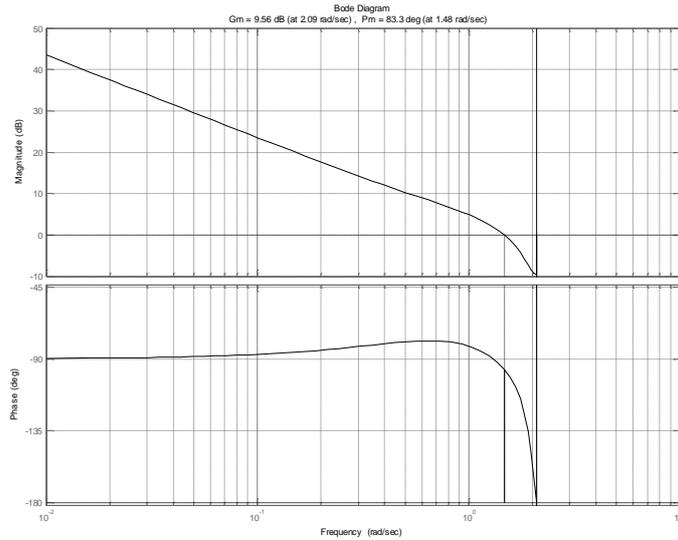


Figura 19. Diagrama de bode del sistema para N=3.

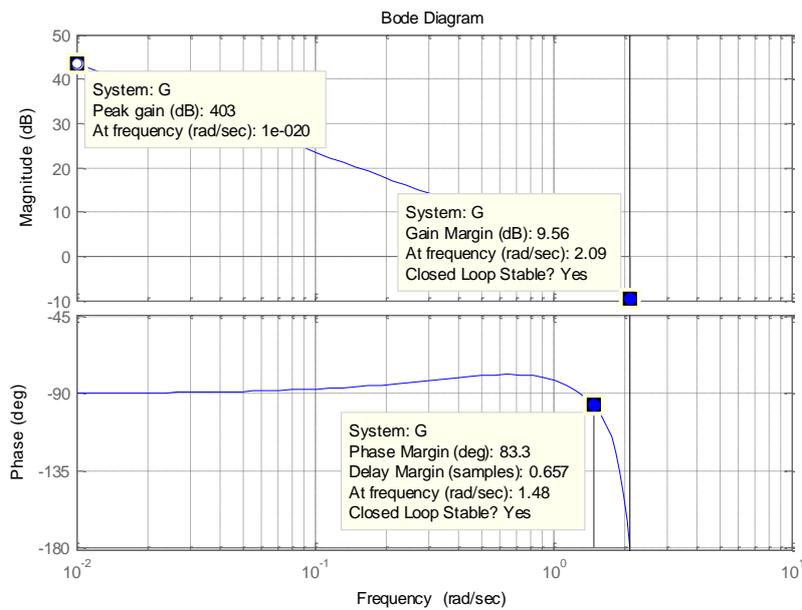


Figura 20. Márgenes de ganancia y de fase del sistema en bucle cerrado para N=3.

Se comprueba que el sistema sigue siendo estable en bucle cerrado. Los márgenes de ganancia y de fase toman los valores de 9.56 dB y de 83.3°.

- $N \rightarrow \infty$
Por último realizamos el análisis en el caso límite, como si se estuviera trabajando en continuo ($N \rightarrow \infty$). Con ello, se tiene:

$$\left. \begin{aligned}
 \overline{G(\bar{z})} &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot Z_{\bar{T}} \left(\frac{1}{s^3} + \frac{1}{\bar{T} \cdot s^4} \right) = \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot Z_{\bar{T}} \left(\frac{1}{2} \cdot \frac{2}{s^3} + \frac{6}{6 \cdot \bar{T} \cdot s^4} \right) = \\
 &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot \frac{\bar{T}^2}{2} \cdot \left[\frac{\bar{z}^{-1} \cdot (1 + \bar{z}^{-1})}{(1 - \bar{z}^{-1})^3} + \frac{1}{3} \cdot \frac{\bar{z}^{-1} \cdot (1 + 4 \cdot \bar{z}^{-1} + \bar{z}^{-2})}{(1 - \bar{z}^{-1})^4} \right] = \\
 &= \left(\frac{\bar{z}-1}{\bar{z}} \right)^2 \cdot \frac{\bar{T}^2}{2} \cdot z \cdot \left[\frac{2 \cdot \bar{z}^2 + 2 \cdot \bar{z} - 1}{(\bar{z} - 1)^4} \right] = \frac{6 \cdot \bar{T}^2}{2} \cdot \frac{2 \cdot \bar{z}^2 + 2 \cdot \bar{z} - 1}{\bar{z}(\bar{z} - 1)^2}
 \end{aligned} \right\} \quad (61)$$

Y para el caso particular de $\bar{T} = 1.5$ se tiene:

$$\overline{G(\bar{z})} = 1.5 \cdot \frac{(\bar{z} - 0.366) \cdot (\bar{z} + 1.366)}{\bar{z}(\bar{z} - 1)^2} \quad (62)$$

Por lo tanto, el lugar de las raíces del sistema global es el que se muestra en la siguiente figura:

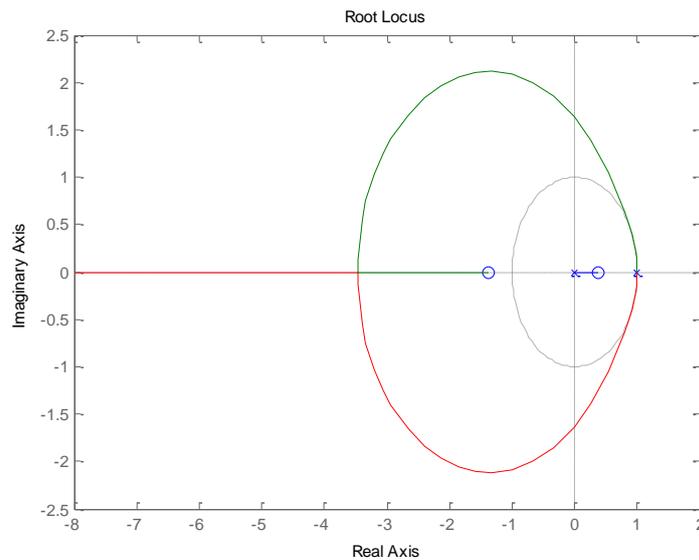


Figura 21. Lugar de las Raíces Multi-frecuencia con $N \rightarrow \infty$.

El análisis en frecuencia del sistema se puede observar en los diagramas de Bode de las figuras 22 y 23.

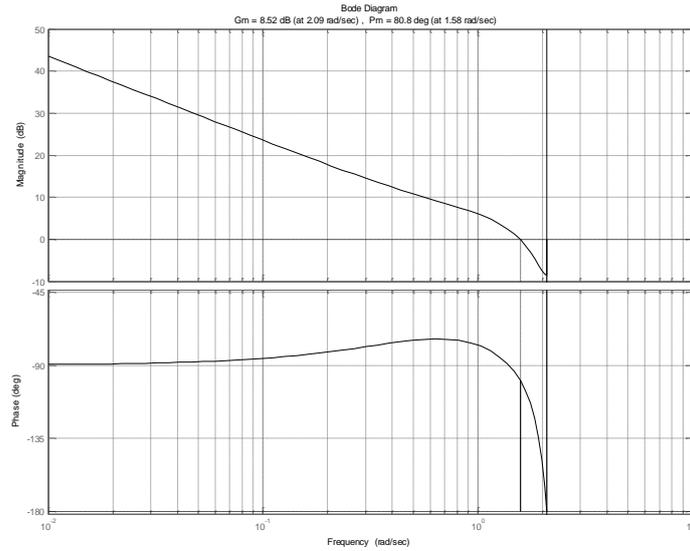


Figura 22. Análisis en frecuencia del sistema en bucle cerrado para $N \rightarrow \infty$.

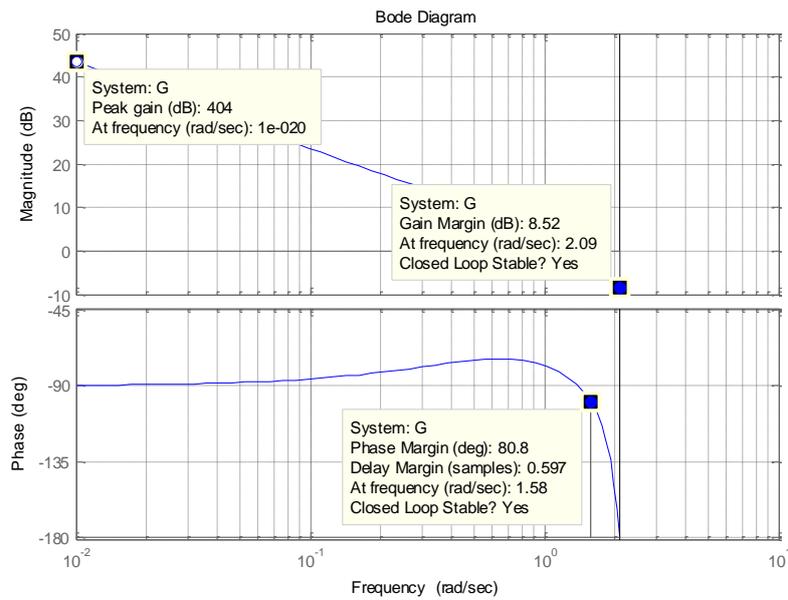


Figura 23. Márgenes de ganancia y de fase del sistema en bucle cerrado con $N \rightarrow \infty$.

Como era de esperar, el sistema es estable, y los márgenes de ganancia toman los valores de 8.52 dB y 80.8° respectivamente.

La Tabla 4 muestra la evolución que sigue cada uno de los parámetros del sistema en bucle abierto. Se puede ver que existe una convergencia al valor continuo ($N \rightarrow \infty$).

	z_1	z_2	k_p
IMPULSO	0.0	-1.0	1.125
ZOH	0.2457	-1.3568	1.2656
FOH	0.2930	-1.3867	1.33
TH	0.36	-1.36	1.5

Tabla 4. Comparación de los valores que han ido tomando los distintos parámetros del sistema en bucle abierto.

Finalmente, la Figura 24 muestra la comparativa realizada en el dominio del Lugar de las Raíces Multifrecuencia. Se observa como las aspas van achatándose entrando ligeramente dentro del círculo unidad convirtiendo un sistema inestable en bucle cerrado en un sistema estable en bucle cerrado, tal y como ha demostrado el estudio frecuencial y del lugar de las raíces.

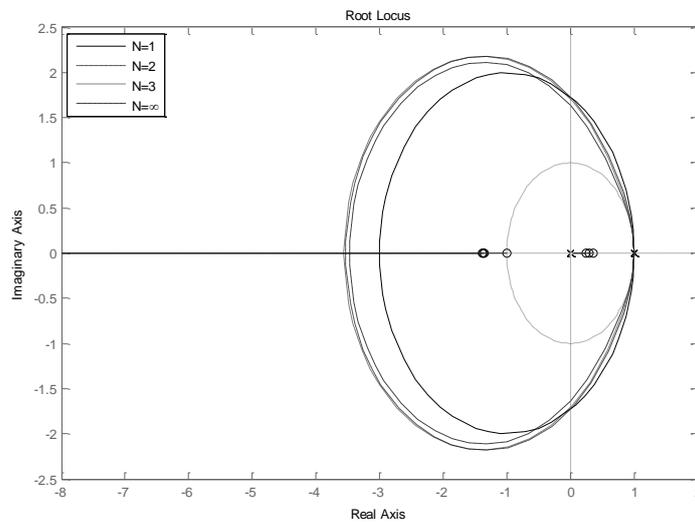


Figura 24. Variación del lugar de las raíces del sistema solamente variando el parámetro N.

También es posible combinar el efecto de los retenedores de orden alto con una función de transferencia general. Además, en la Tabla 5 se muestra las funciones de transferencia discretas equivalentes de los DR-HOH para los cuatro retenedores estudiados.

	$G(z, z^N) = Z_T \left[L^{-1} \{ G(s) \cdot G_h(s)_{NT} \} \right]$
IMPULSO	$G(z, z^N) = Z_T [G(s)]$
ZOH	$G(z, z^N) = \left(\frac{z^N - 1}{z^N} \right) \cdot Z_T \left[\frac{G(s)}{s} \right]$
FOH	$G(z, z^N) = \left(\frac{z^N - 1}{z^N} \right)^2 \cdot Z_T \left[\frac{G(s)}{s} + \frac{G(s)}{NTs^2} \right]$
TH	$G(z, z^N) = \left(\frac{z^N - 1}{z^N} \right)^2 \cdot Z_T \left[\frac{G(s)}{NTs^2} \right]$

Tabla 5. Modelos equivalentes discretos.

Es posible la función de transferencia en tiempo discreto bi-frecuencia del controlador continuo. Además, una forma fácil de calcular los elementos del vector función de transferencia bi-frecuencia para el análisis es realizarla mediante la técnica de lifting.

Muchas técnicas de discretización se pueden adaptar y aplicar para generar modelos discretos bi-frecuencia. En particular, si la entrada es conocida, la siguiente expresión se puede usar para generar la función de transferencia bi-frecuencia $G(z, z^N)$:

$$\left. \begin{aligned} G(s), U(s) \rightarrow Z_T [G_h(s) \cdot U(s)] &= G(z, z^N) \cdot Z_{NT} [U(s)] \\ G(z, z^N) &= Z_T [G_h(s) \cdot U(s)] \cdot Z_{NT} [U(s)]^{-1} \end{aligned} \right\} \quad (63)$$

A modo de ejemplo, pongamos el caso que tenemos el siguiente regulador continuo para nuestra planta:

$$G_c(s) = \frac{s - 0.5}{s + 3.0} \quad (64)$$

La primera cosa que hay que hacer es ver el comportamiento del sistema en bucle cerrado cuando se discretiza a baja frecuencia. Por lo tanto se discretiza el controlador continuo para $\bar{T} = 1.5$ segundos, obteniendo la siguiente función de transferencia:

$$\bar{G}(\bar{z}) = \frac{\bar{z} - 0.8352}{z - 0.0111}, \quad \bar{T} = 1.5 \quad (65)$$

La respuesta del sistema con este controlador en bucle cerrado se muestra en la Figura 25. Como referencia se ha puesto un seno. Se puede ver que el sistema se hace inestable.

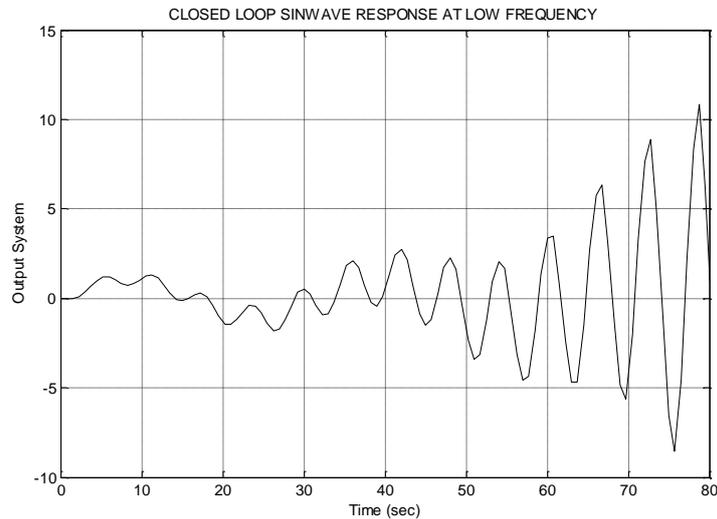


Figura 25. Respuesta del sistema mono-frecuencia en bucle cerrado trabajando a baja frecuencia.

Como se ha estado haciendo durante este documento y, teniendo en cuenta de que se va a trabajar en el enfoque de Estimación de la señal de Error, se obtiene el modelo discreto del controlador trabajando a alta frecuencia, esto es:

$$G_c(s) \xrightarrow{T=0.75, ZOH} G_{c_HF}(z) = \frac{z - 0.8509}{z - 0.1054} \quad (66)$$

Ahora, se obtiene su representación en espacio de estados:

$$G_c(z) \leftrightarrow \begin{cases} x_c(k+1) = A_c \cdot x_c(k) + B_c \cdot u(k) \\ y(k) = C_c \cdot x_c(k) + D_c \cdot u(k) \end{cases} \quad (67)$$

$$A_c = 0.1054, \quad B_c = 1, \quad C_c = -0.7455, \quad D_c = 1 \quad (68)$$

Ahora se aplica la técnica del lifting obteniendo:

$$\begin{aligned} \begin{bmatrix} G_{c,0}(z^2) \\ G_{c,1}(z^2) \end{bmatrix} &= \begin{bmatrix} \frac{z^2 - 0.0897}{z^2 - 0.0111} & \frac{-0.7455}{z^2 - 0.0111} \\ \frac{-0.7455}{z^2 - 0.0111} & \frac{z^2 - 0.0897}{z^2 - 0.0111} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{z^2 - 0.8352}{z^2 - 0.0111} \\ \frac{0.2545 \cdot z^2 - 0.0897}{z^2 - 0.0111} \end{bmatrix} \end{aligned} \quad (69)$$

Utilizando este regulador se obtienen los siguientes resultados:

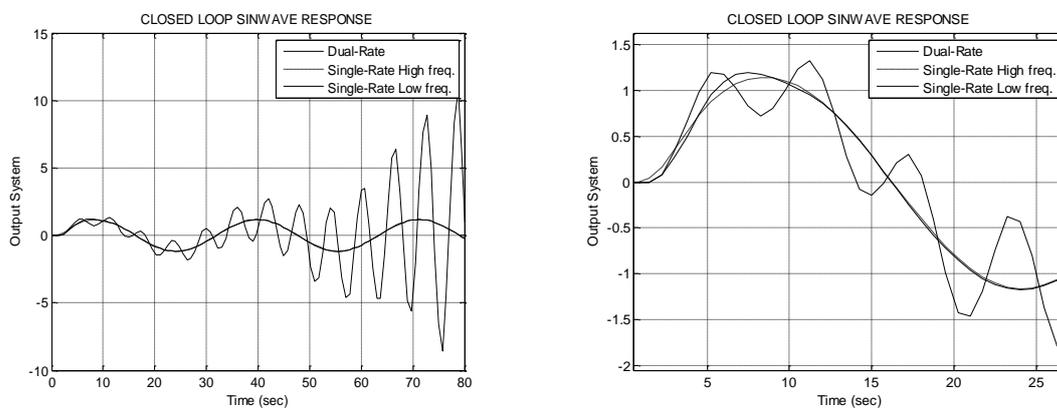


Figura 26. Resultados del enfoque multi-frecuencia.

Como se ve en la Figura 26, lo primero es que se consigue que el sistema siga la referencia y sea estable. Por otra parte, se puede ver que la trayectoria que se consigue es muy similar a la del sistema trabajando a alta frecuencia, lo cual significa que se ha pasado de un sistema monofrecuencia que trabaja a baja frecuencia y es inestable a un sistema bi-frecuencia que, aunque globalmente trabaje a baja frecuencia, por medio de estimaciones se hace que trabaje a alta frecuencia y tenga un comportamiento similar al que tendría el mismo sistema monofrecuencia trabajando a alta frecuencia.

Las figuras que se muestran a continuación son una comparativa entre las distintas señales de control (monofrecuencia a alta frecuencia, a baja frecuencia y con el enfoque multifrecuencia).

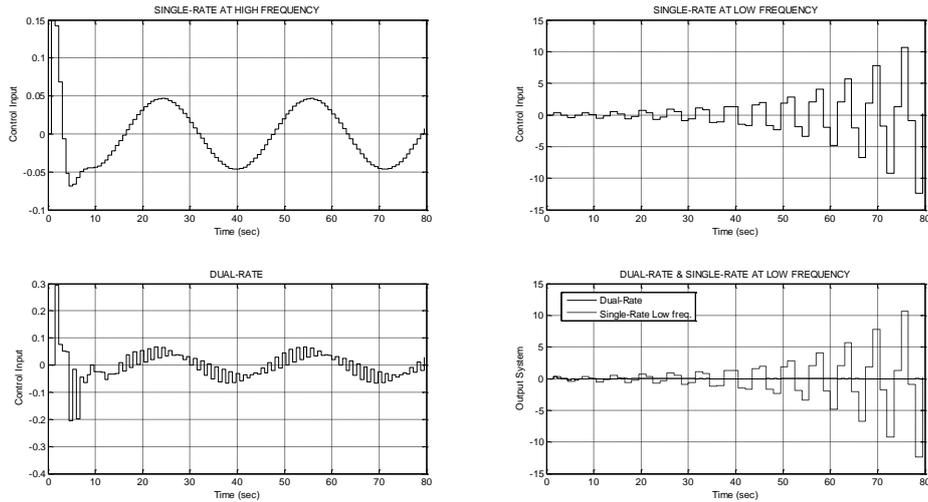


Figura 27. Comparativa de las señales de control.

En este punto y de acuerdo con la Tabla 5, la función de transferencia discreta equivalente con retenedor ZOH bi-frecuencia del controlador descrito en esta sección es:

$$G_c(z, z^2) = \left(\frac{z^2 - 1}{z^2} \right) \cdot Z_T \left[\frac{1}{s} \cdot \frac{s + 0.5}{(s + 3)} \right] = \left. \begin{aligned} &= \frac{z^2 + 0.1491 \cdot z - 0.8509}{z^2 - 0.1054 \cdot z} \end{aligned} \right\} \quad (70)$$

Ahora se va a demostrar que se cumple la equivalencia entre las representaciones. Para ello, se toma la siguiente secuencia de señal de error para el modelo estirado del controlador:

$$E(z^2) = \left\{ \begin{aligned} &1_{k=0, i=0}, 2_{k=1, i=0}, 3_{k=2, i=0}, \dots \\ &1_{k=0, i=1}, 2_{k=1, i=1}, 3_{k=2, i=1}, \dots \end{aligned} \right\} \quad (71)$$

Con ella se tiene la siguiente salida:

$$U(z^2) = \left\{ \begin{aligned} &0.167_{k=0, i=0}, 0.7356_{k=1, i=0}, 0.8675_{k=2, i=0}, \dots \\ &0.167_{k=0, i=1}, 0.1749_{k=1, i=1}, 0.1211_{k=2, i=1}, \dots \end{aligned} \right\} \rightarrow U(z) = \{0.333, 0.9105, 0.9886, \dots\} \quad (72)$$

Comparándola con la secuencia de controlador equivalente utilizando la misma secuencia de error:

$$E(z) = \{1_{k=0}, 2_{k=1}, 3_{k=2}, \dots\} \quad (73)$$

$$U(z) = \{0.333, 0.909, 0.9899, \dots\} \quad (74)$$

Se comprueba que las dos secuencias proporcionan la misma señal de control (las diferencias obtenidas son despreciables).

De esta forma, para una función de transferencia general se tienen las siguientes equivalencias (estas están realizadas para el caso de un DR-FOH):

$$\begin{array}{ccc}
 \begin{array}{l}
 \text{Discrete - Algorithm} \\
 u(kN+i) = u(kN) + \frac{i}{N} \cdot (u(kN) - u((k-1)N)) \\
 \uparrow \quad \begin{array}{l} t = i \cdot T \\ kT \rightarrow k \end{array} \\
 u(kNT+t) = u(kNT) + \frac{t}{T} \cdot (u(kNT) - u((k-1)NT)) \\
 \uparrow \quad L^{-1} \\
 G(s) = (1 - e^{-NTs})^2 \cdot \left(\frac{1}{s} + \frac{1}{NT \cdot s^2} \right)
 \end{array}
 &
 \begin{array}{c}
 \xrightarrow{z_T} \\
 \xleftarrow{z_T^{-1}} \\
 \\
 \\
 \\
 \xrightarrow{Z_T[G_{NT}^*(s)]}
 \end{array}
 &
 \begin{array}{l}
 G_{h,i}(z^N) = \frac{\left(\frac{N+i}{N}\right) \cdot z^N - \left(\frac{i}{N}\right)}{z^N} \\
 \uparrow \qquad \qquad \downarrow \\
 \text{Lifting} \quad G_h(z, z^N) = \sum_{i=0}^{N-1} G_{h,i}(z^N) \cdot z^{-i} \\
 \uparrow \qquad \qquad \downarrow \\
 G_h(z, z^N) = \left(\frac{z^N - 1}{z - 1}\right)^2 \cdot \frac{z^2 - \left(\frac{N-1}{N}\right) \cdot z}{z^{2N}}
 \end{array}
 \end{array}$$

Figura 28. Representaciones del modelo DR-FOH.

ANEXO 2. LENGUAJE RSI-XML

ST_ON

Description:

Starts the RSI execution.

After starting the RSI execution all RSI objects are executed in calling order every sensor cycle. Corrections with ST_PATHCORR() are applied in BASE in integration mode. In operation mode T1 and T2 an acknowledge message !!!Attention-RSI sensor mode active!!! appears.

Declaration:

GLOBAL DEFFCT RSIERR ST_ON()

Parameters:

No

Return Values:

RSIOK: Successful

ST_ON1

Description:

Starts the RSI execution.

After starting the RSI execution all RSI objects are executed in calling order every sensor cycle. The behavior of the correction processing with ST_PATHCORR is defined with parameters TSYS and INTEG. In operation mode T1 and T2 an acknowledge message !!!Attention-RSI sensor mode active!!! appears.

Declaration:

GLOBAL DEFFCT RSIERR ST_ON1(TSYS:IN,INTEG:IN)

Parameters:

IN TSYS (TECHSYS): Correction coordinate system

-> #BASE

-> #TCP

-> #TTS

-> #WORLD

IN INTEG (INT): Integration mode

-> 0: Absolut

-> 1: Relativ

Return Values:

RSIOK: Successful

ST_OFF

Description:

Stops the RSI execution.

Declaration:

GLOBAL DEFFCT RSIERR ST_OFF()

Parameters:

No

Return Values:

RSIOK: Successful

ST_PERSIST

Description:

Toggle for normal and persistent mode. In persistent mode RSI objects are not destroyed when the KRL program is resetted or closed.

Declaration:

GLOBAL DEFFCT RSIERR ST_PERSIST (PFLAG:IN)

Parameters:

IN PFLAG (BOOL): Persistence mode

FALSE: Standard (Default)

TRUE: Persistent
Return Values:
RSIOK: Successful

ST_UPDATE

Description:

Updates the calling order.

The calling order is defined by the linkage of RSI objects. That is the order the RSI objects are executed every sensor cycle. Updating the calling order is normally done automatically by the RSI framework.

Declaration:

GLOBAL DEFFCT RSIERR ST_UPDATE()

Parameters:

No

Return Values:

RSIOK: Successful

ST_RESET

Description:

Deletes all RSI objects.

Declaration:

GLOBAL DEFFCT RSIERR ST_RESET()

Parameters:

No

Return Values:

RSIOK: Successful

ST_TESTMODULE

Description:

Looks if a module specified by a filename is loaded. If not loaded a GUI message is displayed. This command works too if the RSI framework is not loaded.

Declaration:

GLOBAL DEFFCT BOOL ST_TESTMODULE(MODNAME[:OUT],SFLAG:IN)

Parameters:

IN MODNAME (CHAR[]): Module filename

IN SFLAG (BOOL): Stop flag prevents program to continue

Return Values:

TRUE: Module loaded

FALSE: Module not loaded

ST_TESTRSILIB

Description:

Looks if RSI base framework is loaded.

If not loaded a GUI message is displayed.

Declaration:

GLOBAL DEFFCT BOOL ST_TESTRSILIB(SFLAG:IN)

Parameters:

IN SFLAG (BOOL): Stop flag prevents program to continue

Return Values:

TRUE: RSI module loaded

FALSE RSI module not loaded

ST_NEWLINK

Description:

Links a signal from an output of a RSI object to an optional input of another RSI object. This command is only usable to link optional inputs. All mandatory inputs are linked with the creation of the RSI Object.

Declaration:

GLOBAL DEFFCT RSIERR ST_NEWLINK(IDO:IN,IDXO:IN,IDI:IN,IDXI:IN)

Parameters:

IN IDO (INT): ID of source RSI object
 IN IDXO (INT): Output index of source RSI object
 IN IDI (INT): ID of target RSI object
 IN IDXI (INT): Input index of target RSI object

Return Values:

RSIOK: Successful
 RSILNKOUTOBJ: Invalid source ID
 RSILNKINOBJ: Invalid target ID
 RSILNKOUTID: Invalid source index
 RSILNKINID: Invalid target index
 RSILNKSIGNAL: Signal check error

ST_DELLINK

Description:

Deletes an optional signal link. This command is only usable to delete links to optional inputs. All mandatory links can not be deleted.

Declaration:

GLOBAL DEFFCT RSIERR ST_DELLINK(IDO:IN,IDXO:IN,IDI:IN,IDXI:IN)

Parameters:

IN IDO (INT): ID of source RSI object IN IDXO (INT): Output index of source RSI object
 IN IDI (INT): ID of target RSI object IN IDXI (INT): Input index of target RSI object

Return Values:

RSIOK: Successful
 RSIINVINPID: Invalid target ID
 RSIINVOUTID: Invalid source ID
 RSIINVINPIND: Invalid target index
 RSIINVOUTIND: Invalid target index
 RSIINVLNK: Invalid linkage

ST_CHANGELINK

Description:

Changes an existing signal link to a new source defined with NEWIDO and NEWIDXO. This command is only usable if the sensor execution is switched off or for optional signal links.

Declaration:

GLOBAL DEFFCT RSIERR ST_CHANGELINK
 (IDO:IN,IDXO:IN,IDI:IN,IDXI:IN,NEWIDO:IN,NEWIDXO:IN)

Parameters:

IN IDO (INT): Old ID of source RSI object IN IDXO (INT): Old Output index of source RSI object
 IN IDI (INT): ID of target RSI object IN IDXI (INT): Input index of target RSI object IN NEWIDO
 (INT): New ID of source RSI object IN NEWIDX (INT): New output index of source RSI object

Return Values:

RSIOK: Successful
 RSIINVINPID: Invalid target ID
 RSIINVOUTID: Invalid source ID
 RSIINVINPIND: Invalid target index
 RSIINVOUTIND: Invalid target index
 RSIINVLNK: Invalid linka

ST_DIGIN

Description:

Creates a RSI object to access the digital inputs \$IN[].

Declaration:

GLOBAL DEFFCT RSIERR ST_DIGIN(OBJID:OUT,CONTID:IN,INIDX:IN,LEN:IN,UNIT:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN INIDX (INT): Offset of digitals input to read

-> in Bit for LEN 0 (Bit) >=1

-> in Byte for LEN 1,2,3 (Byte, Word) >=1

IN LEN (INT): Range of bits to read

-> 0: Bit

-> 1: Byte

-> 2: Word unsigned

-> 3: Word signed

IN UNIT (INT): Unit of input signal (unused for Bits)

Object Parameters:

1 INIDX (INT): s. parameter INIDX

Object Inputs:

No

Object Outputs:

1 (INT): Digital input value at \$IN[INIDX]

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter INIDX

ST_DIGOUT

Description:

Creates a RSI object to access the digital outputs \$OUT[].

Declaration:

GLOBAL DEFFCT RSIERR ST_DIGOUT(OBJID:OUT,CONTID:IN,INIDX:IN,LEN:IN,UNIT:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN INIDX (INT): Offset of digitals output to read

-> in Bit for LEN 0 (Bit) >=1

-> in Byte for LEN 1,2,3 (Byte, Word) >=1

IN LEN (INT): Range of bits to read

-> 0: Bit

-> 1: Byte

-> 2: Word unsigned

-> 3: Word signed

IN UNIT (INT): Unit of input signal (unused for Bits)

Object Parameters:

1 INIDX (INT): s. parameter INIDX

Object Inputs:

No

Object Outputs:

1 (INT): Digital output value at \$OUT[INIDX]

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter INIDX

ST_ANAIN

Description:

Creates a RSI object to access the analog input \$ANIN[INIDX].

Declaration:

GLOBAL DEFFCT RSIERR ST_ANAIN(OBJID:OUT,CONTID:IN,INIDX:IN,UNIT:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN INIDX (INT): Offset of analog input to read >=1

IN UNIT (INT): Unit of output signal

Object Parameters:

1 INIDX (INT): s. parameter INIDX

Object Inputs:

No

Object Outputs:

1 (REAL): Analog input value at \$ANIN[INIDX]

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter INIDX

ST_ANAOUT

Description:

Creates a RSI object to access the analog output \$ANOUT[OUTIDX].

Declaration:

GLOBAL DEFFCT RSIERR ST_ANAOUT(OBJID:OUT,CONTID:IN,OUTIDX:IN,UNIT:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN OUTIDX (INT): Offset of analog output to read >=1

IN UNIT (INT): Unit of output signal

Object Parameters:

1 INIDX (INT): s. parameter OUTIDX

Object Inputs:

No

Object Outputs:

1 (REAL): Analog output value at \$ANOUT[OUTIDX]

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter OUTIDX

ST_SEN_PINT

Description:

Creates a RSI object to access the sensor variables \$SEN_PINT[VARIDX].

Declaration:

GLOBAL DEFFCT RSIERR ST_SEN_PINT(OBJID:OUT,CONTID:IN,VARIDX:IN,UNIT:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN VARIDX (INT): Index in \$SEN_PINT array >=1

IN UNIT (INT): Unit of output signal

Object Parameters:

1 VARIDX (INT): s. parameter VARIDX

Object Inputs:

No

Object Outputs:

1 (REAL): Signal value of \$SEN_PINT[VARIDX]

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter VARIDX

ST_SEN_PREA

Description:

Creates a RSI object to access the sensor variables \$SEN_PREA[VARIDX].

Declaration:

GLOBAL DEFFCT RSIERR ST_SEN_PREA(OBJID:OUT,CONTID:IN,VARIDX:IN,UNIT:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN VARIDX (INT): Index in \$SEN_PREA array >=1

IN UNIT (INT): Unit of output signal

Object Parameters:

1 INIDX (INT): s. parameter VARIDX

Object Inputs:

No

Object Outputs:

1 (REAL): Signal value of \$SEN_PREA[VARIDX]

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter VARIDX

ST_ACTPOS

Description:

Creates a RSI object to access the robots actual cartesian position. The delivered position relates to the actual settings of \$BASE and \$TOOL.

Declaration:

GLOBAL DEFFCT RSIERR ST_ACTPOS(OBJID:OUT,CONTID:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

Object Parameters:

1 TYPE (INT): Type of position (Default: 0)

-> 0: measured position

-> 1: set position before filter

Object Inputs:

No

Object Outputs:

1 (REAL): X Position [mm]

2 (REAL): Y Position [mm]

3 (REAL): Z Position [mm]

4 (REAL): A Rotation [°]

5 (REAL): B Rotation [°]

6 (REAL): C Rotation [°]

Return Values:

RSIOK : Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

ST_ACTAXIS

Description:

Creates a RSI object to acces the actual axis positions. The unit depends on the axis type:

Linear: mm

Rotatory: °

Declaration:

GLOBAL DEFFCT RSIERR ST_ACTAXIS(OBJID:OUT,CONTID:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

Object Parameters:

1 TYPE (INT): Type of position (Default: 0)

-> 0: measured position

-> 1: set position before filter

Object Inputs:

No

Object Outputs:

1 (REAL): A1 Position axis 1

2 (REAL): A2 Position axis 2 3 (REAL): A3 Position axis 3

4 (REAL): A4 Position axis 4

5 (REAL): A5 Position axis 5

6 (REAL): A6 Position axis 6

7 (REAL): E1 Position external axis 1

8 (REAL): E2 Position external axis 2

9 (REAL): E3 Position external axis 3

10 (REAL): E4 Position external axis 4

11 (REAL): E5 Position external axis 5

12 (REAL): E6 Position external axis 6

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

Return Values:

RSIOK : Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

ST_SOURCE

Description:

Creates a RSI object that can generate standard signals:

-> constant signal

-> sine /cosine signal

-> square signal

-> saw tooth signal

Declaration:

GLOBAL DEFFCT RSIERR ST_SOURCE(OBJID:OUT,CONTID:IN,UNIT:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN UNIT (INT): Unit of output signal

Object Parameters:

1 TYPE (INT): Type of signal (Default: 0)

-> 0: constant signal

-> 1: sine signal

-> 2: cosine signal

-> 3: square signal

-> 4: sawtooth signal

2 OFFSET (REAL): Amplitude offset (Default: 0)

- 3 AMPL (REAL): Amplitude (Default: 1)
- 4 PERIOD (REAL): Periode in [sec] (Default: 5)
- 5 PARAM1 (REAL): Depends on signal type (Default: 0.25)
- 6 PARAM2 (REAL): Depends on signal type (Default: 0.25)
- 7 PARAM3 (REAL): Depends on signal type (Default: 0.25)
- 8 PARAM4 (REAL): Depends on signal type (Default: 0.25)
- 9 PARAM5 (REAL): Depends on signal type (Default: 0.25)

Object Inputs:

No

Object Outputs:

1 (REAL): Desired signal

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

ST_GEARTORQUE

Description:

Creates a RSI object to access the current motor torques.

Declaration:

GLOBAL DEFFCT RSIERR ST_GEARTORQUE(OBJID:OUT,CONTID:IN,TYPE:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN TYPE (INT): Type of torques to access

-> 0: Desired gear torques

-> 1: Desired model gear torques

Object Parameters:

1 TYPE (INT): s. parameter TYPE

Object Inputs:

No

Object Outputs:

1 (DOUBLE): Current for axis 1

...

6 (DOUBLE): Current for axis 6

7 (DOUBLE): Current for external axis 1

...

12 (DOUBLE): Current for external axis 6

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter TYPE

ST_MOTORCURRENT

Description:

Creates a RSI object to access the motor currents.

Declaration:

GLOBAL DEFFCT RSIERR ST_MOTORCURRENT(OBJID:OUT,CONTID:IN,TYPE:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of the container for the new object

IN TYPE (INT): Type of motor currents to access

-> 0 : Desired motor currents (DSE interface)

Object Parameters:

1 TYPE (INT): s. parameter TYPE

Object Inputs:

No

Object Outputs:

1 (DOUBLE): Current for axis 1

...

6 (DOUBLE): Current for axis 6

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter TYPE

ST_GENTRANS

Description:

Creates a RSI object to transform a Input signal vector from frame a FrI to a frame FrO. The object parameters 1 - 6 define the translation and rotation of FrI in FrO With parameter TYPE set to 1 or 3 only the rotation is taken into account. With TYPE set to 0, 2 or 4 the transformation takes the translation and the rotation of the frames into account. TYPE 0 and 1 are used for the 3 dimensional vector of input 1 - 3. In this case the parameters for the linkage of inputs 4 - 6 have to be set to 0. For the 6 dimensional vector transformed with TYPE 2, 3 and 4 the vector follows the X,Y,Z,A,B,C convention. With TYPE 4 a 6 dimensional force torque vector is transformed from FrI to FrO with respect to the dependency of force and torque.

Declaration:

GLOBAL DEFFCT RSIERR ST_GENTRANS

(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,IDX2:IN,IDX3:IN,IDX4:IN,IDX5:IN,IDX6:IN,TYPE:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN ID2 (INT): ID of the object to link to input 2

IN IDX2 (INT): Output index of ID1 to link to input 2

IN ID3 (INT): ID of the object to link to input 3

IN IDX3 (INT): Output index of ID1 to link to input 3

IN ID4 (INT): ID of the object to link to input 4

IN IDX4 (INT): Output index of ID1 to link to input 4

IN ID5 (INT): ID of the object to link to input 5

IN IDX5 (INT): Output index of ID1 to link to input 5

IN ID6 (INT): ID of the object to link to input 6

IN IDX6 (INT): Output index of ID1 to link to input 6

IN TYPE (INT): Type of calculation

-> 0: 3 dim. vector - rotation and translation

-> 1: 3 dim. vector - only rotation

-> 2: 6 dim. vector - rot. and tran. X,Y,Z,A,B,C input

-> 3: 6 dim. vector - only rot. X,Y,Z,A,B,C input

-> 4: 6 dim. vector - force torque Fx,Fy,Fz,Mz,My,Mx

Object Parameters:

1 TransX (REAL): Translation X of FrO in FrI

2 TransY (REAL): Translation Y of FrO in FrI

3 TransZ (REAL): Translation Z of FrO in FrI

4 RotZ (REAL): Rotation A of FrO in FrI

5 RotY (REAL): Rotation B of FrO in FrI

6 RotX (REAL): Rotation C of FrO in FrI

7 TYPE (INT): Type of calculation (s. parameter TYPE)

Object Inputs:

1 (REAL): Input signal 1

2 (REAL): Input signal 2

3 (REAL): Input signal 3
4 (REAL): Input signal 4 (optional s. TYPE)
5 (REAL): Input signal 5 (optional s. TYPE)
6 (REAL): Input signal 6 (optional s. TYPE)

Object Outputs:

1 (REAL): Transformed output signal 1
2 (REAL): Transformed output signal 2
3 (REAL): Transformed output signal 3
4 (REAL): Transformed output signal 4 (optional s. TYPE)
5 (REAL): Transformed output signal 5 (optional s. TYPE)
6 (REAL): Transformed output signal 6 (optional s. TYPE)

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSIINVPARAM: Invalid parameter TYPE
RSILNKOUTOBJ: Invalid ID for linkage
RSILNKOUTINDEX: Invalid index for linkage
RSILNKOUTDISABLED:
RSILNKMAXDEP:
RSILNK SIGNAL: Input output signal mismatch
RSILNKCYCLE: Cyclic linkage
RSINOLNKOBJ: Object not linkable

ST_SN2TOOL

Description:

Creates a RSI object to transform a Input signal vector from a sensor frame (FrSen) attached to the robot to the actual TOOL frame (FrTool).

The object parameters 1 - 6 define the translation and rotation of FrSen in the robot flange frame (FrFlange). With parameter TYPE set to 1 or 3 only the rotation is taken into account. With TYPE set to 0, 2 or 4 the transformation takes the translation and the rotation of the frames into account. TYPE 0 and 1 are used for the 3 dimensional vector of input 1 - 3. In this case the parameters for the linkage of inputs 4 - 6 have to be set to 0. For the 6 dimensional vector transformed with TYPE 2, 3 and 4 the vector follows the X,Y,Z,A,B,C convention. With TYPE 4 a 6 dimensional force torque vector is transformed from Fr1 to Fr0 with respect to the dependency of force and torque.

Declaration:

GLOBAL DEFFCT RSIERR ST_SN2TOOL
(OBJID:OUT,CONTID:IN, ID1:IN,IDX1:IN, ID2:IN,IDX2:IN, ID3:IN,IDX3:IN, ID4:IN,IDX4:IN, ID5:IN, IDX5:IN, ID6:IN,IDX6:IN, TYPE:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN ID2 (INT): ID of the object to link to input 2
IN IDX2 (INT): Output index of ID1 to link to input 2
IN ID3 (INT): ID of the object to link to input 3
IN IDX3 (INT): Output index of ID1 to link to input 3
IN ID4 (INT): ID of the object to link to input 4
IN IDX4 (INT): Output index of ID1 to link to input 4
IN ID5 (INT): ID of the object to link to input 5
IN IDX5 (INT): Output index of ID1 to link to input 5
IN ID6 (INT): ID of the object to link to input 6
IN IDX6 (INT): Output index of ID1 to link to input 6
IN TYPE (INT): Type of calculation
-> 0: 3 dim. vector - rotation and translation

- > 1: 3 dim. vector - only rotation
- > 2: 6 dim. vector - rot. and tran. X,Y,Z,A,B,C input
- > 3: 6 dim. vector - only rot. X,Y,Z,A,B,C input
- > 4: 6 dim. vector - force torque Fx,Fy,Fz,Mz,My,Mx

Object Parameters:

- 1 TransX (REAL): Translation X of FrSen in FrFlange
- 2 TransY (REAL): Translation Y of FrSen in FrFlange
- 3 TransZ (REAL): Translation Z of FrSen in FrFlange
- 4 RotZ (REAL): Rotation A of FrSen in FrFlange
- 5 RotY (REAL): Rotation B of FrSen in FrFlange
- 6 RotX (REAL): Rotation C of FrSen in FrFlange
- 7 TYPE (INT): Type of calculation (s. parameter TYPE)

Object Inputs:

- 1 (REAL): Input signal 1
- 2 (REAL): Input signal 2
- 3 (REAL): Input signal 3
- 4 (REAL): Input signal 4 (optional s. TYPE)
- 5 (REAL): Input signal 5 (optional s. TYPE)
- 6 (REAL): Input signal 6 (optional s. TYPE)

Object Outputs:

- 1 (REAL): Transformed output signal 1
- 2 (REAL): Transformed output signal 2
- 3 (REAL): Transformed output signal 3
- 4 (REAL): Transformed output signal 4 (optional s. TYPE)
- 5 (REAL): Transformed output signal 5 (optional s. TYPE)
- 6 (REAL): Transformed output signal 6 (optional s. TYPE)

Return Values:

- RSIOK: Successful
- RSIINVCONT: Invalid container
- RSICONTOVERFLOW: Container overflow
- RSIINVPARAM: Invalid parameter TYPE
- RSILNKOUTOBJ: Invalid ID for linkage
- RSILNKOUTINDEX: Invalid index for linkage
- RSILNK SIGNAL: Input output signal mismatch
- RSILNKCYCLE: Cyclic linkage
- RSINOLNKOBJ: Object not linkable

ST_SN2BASE

Description:

Creates a RSI object to transform a Input signal vector from a sensor frame (FrSen) attached to the robot to the actual BASE frame (FrBase).

The object parameters 1 - 6 define the translation and rotation of FrSen in the robot flange frame (FrFlange). With parameter TYPE set to 1 or 3

only the rotation is taken into account. With TYPE set to 0, 2 or 4 the transformation takes the translation and the rotation of the frames into

account. TYPE 0 and 1 are used for the 3 dimensional vector of input 1 - 3. In this case the parameters for the linkage of inputs 4 - 6 have to be

set to 0. For the 6 dimensional vector transformed with TYPE 2, 3 and 4 the vector follows the X,Y,Z,A,B,C convention. With TYPE 4 a 6

dimensional force torque vector is transformed from FrI to Fr0 with respect to the dependency of force and torque.

Declaration:

```
GLOBAL DEFFCT RSIERR ST_SN2BASE
(OBJID:OUT,CONTID:IN,IDX1:IN,IDX2:IN,IDX3:IN,IDX4:IN,IDX5:IN,
IDX6:IN,TYPE:IN)
```

Parameters:

- OUT OBJID (INT): ID of the created RSI object
- IN CONTID (INT): ID of container for the new object
- IN IDX1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1
IN ID2 (INT): ID of the object to link to input 2
IN IDX2 (INT): Output index of ID1 to link to input 2
IN ID3 (INT): ID of the object to link to input 3
IN IDX3 (INT): Output index of ID1 to link to input 3
IN ID4 (INT): ID of the object to link to input 4
IN IDX4 (INT): Output index of ID1 to link to input 4
IN ID5 (INT): ID of the object to link to input 5
IN IDX5 (INT): Output index of ID1 to link to input 5
IN ID6 (INT): ID of the object to link to input 6
IN IDX6 (INT): Output index of ID1 to link to input 6
IN TYPE (INT): Type of calculation
-> 0: 3 dim. vector - rotation and translation
-> 1: 3 dim. vector - only rotation
-> 2: 6 dim. vector - rot. and tran. X,Y,Z,A,B,C input
-> 3: 6 dim. vector - only rot. X,Y,Z,A,B,C input
-> 4: 6 dim. vector - force torque Fx,Fy,Fz,Mz,My,Mx

Object Parameters:

1 TransX (REAL): Translation X of FrSen in FrFlange
2 TransY (REAL): Translation Y of FrSen in FrFlange
3 TransZ (REAL): Translation Z of FrSen in FrFlange
4 RotZ (REAL): Rotation A of FrSen in FrFlange
5 RotY (REAL): Rotation B of FrSen in FrFlange
6 RotX (REAL): Rotation C of FrSen in FrFlange
7 TYPE (INT): Type of calculation (s. parameter TYPE)

Object Inputs:

1 (REAL): Input signal 1
2 (REAL): Input signal 2
3 (REAL): Input signal 3
4 (REAL): Input signal 4 (optional s. TYPE)
5 (REAL): Input signal 5 (optional s. TYPE)
6 (REAL): Input signal 6 (optional s. TYPE)

Object Outputs:

1 (REAL): Transformed output signal 1
2 (REAL): Transformed output signal 2
3 (REAL): Transformed output signal 3
4 (REAL): Transformed output signal 4 (optional s. TYPE)
5 (REAL): Transformed output signal 5 (optional s. TYPE)
6 (REAL): Transformed output signal 6 (optional s. TYPE)

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSIINVPARAM: Invalid parameter TYPE
RSILNKOUTOBJ: Invalid ID for linkage
RSILNKOUTINDEX: Invalid index for linkage
RSILNKSIGNAL: Input output signal mismatch
RSILNKCYCLE: Cyclic linkage
RSINOLNKOBJ: Object not linkable
RSIINVPARAM: Invalid parameter TYPE

ST_SN2WORLD

Description:

Creates a RSI object to transform a Input signal vector from a sensor frame (FrSen) attached to the robot to the actual WORLD frame (FrWorld). The object parameters 1 - 6 define the translation and rotation of FrSen in the robot flange frame (FrFlange). With parameter TYPE set to 1 or 3 only the rotation is taken into account. With TYPE set to 0, 2 or 4 the transformation takes the translation and the rotation of the

frames into account. TYPE 0 and 1 are used for the 3 dimensional vector of input 1 - 3. In this case the parameters for the linkage of inputs 4 - 6 have to be set to 0. For the 6 dimensional vector transformed with TYPE 2, 3 and 4 the vector follows the X,Y,Z,A,B,C convention. With TYPE 4 a 6 dimensional force torque vector is transformed from FrI to Fr0 with respect to the dependency of force and torque.

Declaration:

GLOBAL DEFFCT RSIERR ST_SN2WORLD
(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,IDX2:IN,IDX3:IN,IDX4:IN,IDX5:IN,IDX6:IN,TYPE:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN ID2 (INT): ID of the object to link to input 2
IN IDX2 (INT): Output index of ID1 to link to input 2
IN ID3 (INT): ID of the object to link to input 3
IN IDX3 (INT): Output index of ID1 to link to input 3
IN ID4 (INT): ID of the object to link to input 4
IN IDX4 (INT): Output index of ID1 to link to input 4
IN ID5 (INT): ID of the object to link to input 5
IN IDX5 (INT): Output index of ID1 to link to input 5
IN ID6 (INT): ID of the object to link to input 6
IN IDX6 (INT): Output index of ID1 to link to input 6
IN TYPE (INT): Type of calculation
-> 0: 3 dim. vector - rotation and translation
-> 1: 3 dim. vector - only rotation
-> 2: 6 dim. vector - rot. and tran. X,Y,Z,A,B,C input
-> 3: 6 dim. vector - only rot. X,Y,Z,A,B,C input
-> 4: 6 dim. vector - force torque Fx,Fy,Fz,Mz,My,Mx

Object Parameters:

1 TransX (REAL): Translation X of FrSen in FrFlange
2 TransY (REAL): Translation Y of FrSen in FrFlange
3 TransZ (REAL): Translation Z of FrSen in FrFlange
4 RotZ (REAL): Rotation A of FrSen in FrFlange
5 RotY (REAL): Rotation B of FrSen in FrFlange
6 RotX (REAL): Rotation C of FrSen in FrFlange
7 TYPE (INT): Type of calculation (s. parameter TYPE)

Object Inputs:

1 (REAL): Input signal 1
2 (REAL): Input signal 2
3 (REAL): Input signal 3
4 (REAL): Input signal 4 (optional s. TYPE)
5 (REAL): Input signal 5 (optional s. TYPE)
6 (REAL): Input signal 6 (optional s. TYPE)

Object Outputs:

1 (REAL): Transformed output signal 1
2 (REAL): Transformed output signal 2
3 (REAL): Transformed output signal 3
4 (REAL): Transformed output signal 4 (optional s. TYPE)
5 (REAL): Transformed output signal 5 (optional s. TYPE)
6 (REAL): Transformed output signal 6 (optional s. TYPE)

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSIINVPARAM: Invalid parameter TYPE
RSILNKOUTOBJ: Invalid ID for linkage
RSILNKOUTINDEX: Invalid index for linkage

RSILNKSIGNAL: Input output signal mismatch

RSILNKCYCLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_SN2TTS

Description:

Creates a RSI object to transform a Input signal vector from a sensor frame (FrSen) attached to the robot to the actual TTS frame (FrTTS). The

object parameters 1 - 6 define the translation and rotation of FrSen in the robot flange frame (FrFlange).

With parameter TYPE set to 1 or 3 only

the rotation is taken into account. With TYPE set to 0, 2 or 4 the transformation takes the translation and the rotation of the frames into account.

TYPE 0 and 1 are used for the 3 dimensional vector of input 1 - 3. In this case the parameters for the linkage of inputs 4 - 6 have to be set to 0.

For the 6 dimensional vector transformed with TYPE 2, 3 and 4 the vector follows the X,Y,Z,A,B,C convention. With TYPE 4 a 6 dimensional

force torque vector is transformed from FrI to Fr0 with respect to the dependency of force and torque.

Declaration:

GLOBAL DEFFCT RSIERR ST_SN2TTS

(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,IDX2:IN,IDX3:IN,IDX4:IN,IDX5:IN,IDX6:IN,TYPE:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN ID2 (INT): ID of the object to link to input 2

IN IDX2 (INT): Output index of ID1 to link to input 2

IN ID3 (INT): ID of the object to link to input 3

IN IDX3 (INT): Output index of ID1 to link to input 3

IN ID4 (INT): ID of the object to link to input 4

IN IDX4 (INT): Output index of ID1 to link to input 4

IN ID5 (INT): ID of the object to link to input 5

IN IDX5 (INT): Output index of ID1 to link to input 5

IN ID6 (INT): ID of the object to link to input 6

IN IDX6 (INT): Output index of ID1 to link to input 6

IN TYPE (INT): Type of calculation

-> 0: 3 dim. vector - rotation and translation

-> 1: 3 dim. vector - only rotation

-> 2: 6 dim. vector - rot. and tran. X,Y,Z,A,B,C input

-> 3: 6 dim. vector - only rot. X,Y,Z,A,B,C input

-> 4: 6 dim. vector - force torque Fx,Fy,Fz,Mz,My,Mx

Object Parameters:

1 TransX (REAL): Translation X of FrSen in FrFlange

2 TransY (REAL): Translation Y of FrSen in FrFlange

3 TransZ (REAL): Translation Z of FrSen in FrFlange

4 RotZ (REAL): Rotation A of FrSen in FrFlange

5 RotY (REAL): Rotation B of FrSen in FrFlange

6 RotX (REAL): Rotation C of FrSen in FrFlange

7 TYPE (INT): Type of calculation (s. parameter TYPE)

Object Inputs:

1 (REAL): Input signal 1

2 (REAL): Input signal 2

3 (REAL): Input signal 3

4 (REAL): Input signal 4 (optional s. TYPE)

5 (REAL): Input signal 5 (optional s. TYPE)

6 (REAL): Input signal 6 (optional s. TYPE)

Object Outputs:

1 (REAL): Transformed output signal 1

2 (REAL): Transformed output signal 2
 3 (REAL): Transformed output signal 3
 4 (REAL): Transformed output signal 4 (optional s. TYPE)
 5 (REAL): Transformed output signal 5 (optional s. TYPE)
 6 (REAL): Transformed output signal 6 (optional s. TYPE)

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow
 RSIINVPARAM: Invalid parameter TYPE
 RSILNKOUTOBJ: Invalid ID for linkage
 RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYCLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_AND

Description:

Creates a RSI object to do a binary AND operation. The inputs 1 and 2 have to be linked at creation time. All further inputs 3 - 10 can be established with ST_NEWLINK().

Declaration:

GLOBAL DEFFCT RSIERR ST_AND(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN ID2 (INT): ID of the object to link to input 2
 IN IDX2 (INT): Output index of ID2 to link to input 2

Object Parameters:

No

Object Inputs:

1 (BOOL): Input signal 1
 2 (BOOL): Input signal 2
 3 (BOOL): Input signal 3 (optional)
 ...
 10 (BOOL): Input signal 10 (optional)

Object Outputs:

1 (BOOL): AND operation result

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow
 RSILNKOUTOBJ: Invalid ID for linkage
 RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYCLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_OR

Description:

Creates a RSI object to do a binary OR operation. The inputs 1 and 2 have to be linked at creation time. All further inputs 3 - 10 can be established with ST_NEWLINK().

Declaration:

GLOBAL DEFFCT RSIERR ST_OR(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN ID2 (INT): ID of the object to link to input 2
IN IDX2 (INT): Output index of ID2 to link to input 2

Object Parameters:

No

Object Inputs:

1 (BOOL): Input signal 1 (mandatory)
2 (BOOL): Input signal 2 (mandatory)
3 (BOOL): Input signal 3 (optional)

...

10 (BOOL): Input signal 10 (optional)

Object Outputs:

1 (BOOL): OR operation result

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSILNKOUTOBJ: Invalid ID for linkage
RSILNKOUTINDEX: Invalid index for linkage
RSILNKSIGNAL: Input output signal mismatch
RSILNKCYCLE: Cyclic linkage
RSINOLNKOBJ: Object not linkable

ST_XOR

Description:

Creates a RSI object to do a binary XOR operation.

Declaration:

GLOBAL DEFFCT RSIERR ST_XOR(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN ID2 (INT): ID of the object to link to input 2
IN IDX2 (INT): Output index of ID2 to link to input 2

Object Parameters:

No

Object Inputs:

1 (BOOL): Input signal 1
2 (BOOL): Input signal 2

Object Outputs:

1 (BOOL): XOR operation result

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSILNKOUTOBJ: Invalid ID for linkage
RSILNKOUTINDEX: Invalid index for linkage
RSILNKSIGNAL: Input output signal mismatch
RSILNKCYCLE: Cyclic linkage
RSINOLNKOBJ: Object not linkable

ST_NOT

Description:

Creates a RSI object to do a binary NOT operation.

Declaration:

GLOBAL DEFFCT RSIERR ST_NOT(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input

IN IDX1 (INT): Output index of ID to link to input

Object Parameters:

No

Object Inputs:

1 (BOOL): Input signal 1

Object Outputs:

1 (BOOL): NOT operation result

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage

RSILNKOUTINDEX: Invalid index for linkage

RSILNKSIGNAL: Input output signal mismatch

RSILNKCYLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_BAND

Description:

Creates a RSI object to do a bitwise AND operation. If the second input is unlinked the operation is done with the first signal and the constant value of parameter DEFVAL. In this case ID2 and IDX2 have to be set to 0.

Declaration:

GLOBAL DEFFCT RSIERR

ST_BAND(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN,DEFVAL:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN ID2 (INT): ID of the object to link to input 2

IN IDX2 (INT): Output index of ID2 to link to input 2

IN DEFVAL (INT): Default value if input 2 is unlinked

Object Parameters:

1 DEFVAL (INT): Default value if input 2 is unlinked

Object Inputs:

1 (INT): Input signal 1

2 (INT): Input signal 2 (optional)

Object Outputs:

1 (INT): Bitwise AND operation result

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage

RSILNKOUTINDEX: Invalid index for linkage

RSILNKSIGNAL: Input output signal mismatch

RSILNKCYLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_BOR

Description:

Creates a RSI object to do a bitwise OR operation. If the second input is unlinked the operation is done with the first signal and the constant value of parameter DEFVAL. In this case ID2 and IDX2 have to be set to 0.

Declaration:

GLOBAL DEFFCT RSIERR

ST_BOR(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN,DEFVAL:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN ID2 (INT): ID of the object to link to input 2
IN IDX2 (INT): Output index of ID1 to link to input 2
IN DEFVAL (INT): Default value if input 2 is unlinked

Object Parameters:

1 DEFVAL (INT): Default value if input 2 is unlinked

Object Inputs:

1 (INT): Input signal 1
2 (INT): Input signal 2 (optional)

Object Outputs:

1 (INT): Bitwise OR operation result

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSILNKOUTOBJ: Invalid ID for linkage
RSILNKOUTINDEX: Invalid index for linkage
RSILNKSIGNAL: Input output signal mismatch
RSILNKCYCLE: Cyclic linkage
RSINOLNKOBJ: Object not linkable

ST_BCOMPL

Description:

Creates a RSI object to calculate the bitwise complement.

Declaration:

GLOBAL DEFFCT RSIERR ST_BCOMPL(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1

Object Parameters:

No

Object Inputs:

1 (INT): Input signal 1

Object Outputs:

1 (INT): Calculated complement

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSILNKOUTOBJ: Invalid ID for linkage
RSILNKOUTINDEX: Invalid index for linkage
RSILNKSIGNAL: Input output signal mismatch
RSILNKCYCLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_EQUAL

Description:

Creates a RSI object to compares its input signals. The result is TRUE if the first and the second input are equal. If the second input is unlinked the first signal is compared with the constant value of parameter DEFVAL. In this case ID2 and IDX2 have to be set to 0.

Declaration:

GLOBAL DEFFCT RSIERR

ST_EQUAL(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN,DEFVAL:IN,DEV:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN ID2 (INT): ID of the object to link to input 2
 IN IDX2 (INT): Output index of ID2 to link to input 2
 IN DEFVAL (REAL): Default value if input 2 is unlinked
 IN DEV (REAL): Deviation for comparison

Object Parameters:

1 DEFVAL (REAL): s. parameter DEFVAL
 2 DEV (REAL): s. parameter DEV

Object Inputs:

1 (REAL): Input signal 1
 2 (REAL): Input signal 2 (optional)

Object Outputs:

1 (BOOL): Comparison operation result

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow
 RSILNKOUTOBJ: Invalid ID for linkage
 RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYCLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_LESS

Description:

Creates a RSI object to compare its input signals. The result is TRUE if the first input is less than the second one. If the second input is unlinked the first signal is compared with the constant value of parameter DEFVAL. In this case ID2 and IDX2 have to be set to 0.

Declaration:

GLOBAL DEFFCT RSIERR

ST_LESS(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN,DEFVAL:IN,HIST:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN ID2 (INT): ID of the object to link to input 2
 IN IDX2 (INT): Output index of ID2 to link to input 2
 IN DEFVAL (REAL): Default value if input 2 is unlinked
 IN HIST (REAL): Histeresis for comparison

Object Parameters:

1 DEFVAL (REAL): s. parameter DEFVAL

2 HIST (REAL): s. parameter HIST

Object Inputs:

1 (REAL): Input signal 1

2 (REAL): Input signal 2 (optional)

Object Outputs:

1 (BOOL): Comparison operation result

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage

RSILNKOUTINDEX: Invalid index for linkage

RSILNK SIGNAL: Input output signal mismatch

RSILNKCYCLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_GREATER

Description:

Creates a RSI object to compare its input signals. The result is TRUE if the first input is greater than the second one. If the second input is unlinked the first signal is compared with the constant value of parameter DEFVAL. In this case ID2 and IDX2 have to be set to 0.

Declaration:

GLOBAL DEFFCT RSIERR

ST_GREATER(OBJID:OUT,CONTID:IN,IDX1:IN,IDX2:IN,DEFVAL:IN,HIST:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN ID2 (INT): ID of the object to link to input 2

IN IDX2 (INT): Output index of ID2 to link to input 2

IN DEFVAL (REAL): Default value if input 2 is unlinked

IN HIST (REAL): Histeresis for comparison

Object Parameters:

1 DEFVAL (REAL): s. parameter DEFVAL

2 HIST (REAL): s. parameter HIST

Object Inputs:

1 (REAL): Input signal 1

2 (REAL): Input signal 2 (optional)

Object Outputs:

1 (BOOL): Comparison operation result

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage

RSILNKOUTINDEX: Invalid index for linkage

RSILNK SIGNAL: Input output signal mismatch

RSILNKCYCLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_P

Description:

Creates a RSI object with P control functionality. With KR set to 1 the object can be used to change the signal unit to a desired value.

$$y(k) = KR * x(k)$$

Declaration:

GLOBAL DEFFCT RSIERR ST_P(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,KR:IN,UNIT:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN KR (REAL): Control parameter KR
 IN UNIT (INT): Unit of output signal

Object Parameters:

1: KR (REAL): s. parameter KR

Object Inputs:

1 (REAL): Input signal 1

Object Outputs:

1 (REAL): Output signal 1

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow
 RSILNKOUTOBJ: Invalid ID for linkage
 RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_PD

Description:

Creates a RSI object with PD control functionality.

$$y(k) = B0 * x(k) + B1 * x(k-1)$$

with:

$$B0 = KR * (1 + (TV / 0.012))$$

$$B1 = -KR * (TV / 0.012)$$

Declaration:

GLOBAL DEFFCT RSIERR ST_PD(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,KR:IN,TV:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN KR (REAL): Control parameter KR
 IN TV (REAL): Control parameter TV in [sec]

Object Parameters:

1: KR (REAL): s. parameter KR

2: TV (REAL): s. parameter TV

Object Inputs:

1 (REAL): Input signal 1

Object Outputs:

1 (REAL): Output signal 1

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow
 RSILNKOUTOBJ: Invalid ID for linkage
 RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_I

Description:

Creates a RSI object with I control functionality with the trapeze algorithm. With parameter IMODE set to 1 the

integration is only done when a CP movement (LIN / CIRC) is active.

$$y(k) = B0 * (x(k) + x(k-1)) + y(k-1)$$

with:

$$B0 = TI * 0.012 / 2$$

Declaration:

GLOBAL DEFFCT RSIERR ST_I(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,TI:IN,IMODE:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN TI (REAL): Control parameter TI in [sec]

IN IMODE (INT): Mode of integration

-> 0: Always

-> 1: Only on CP movements

Object Parameters:

1: RESET (BOOL): Set to TRUE resets the integrator

2: TI (REAL): s. parameter TI

3: IMODE (INT): s. parameter IMODE

Object Inputs:

1 (REAL): Input signal 1

Object Outputs:

1 (REAL): Output signal 1

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage

RSILNKOUTINDEX: Invalid index for linkage

RSILNKSIGNAL: Input output signal mismatch

RSILNKCYCLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_PID

Description:

Creates a RSI object with PID control functionality.

$$y(k) = y(k-1) + B0 * x(k) + B1 * x(k-1) + B2 * x(k-2)$$

with:

$$B0 = KR * (1 + TV / 0.012)$$

$$B1 = - KR * (1 - 0.012 / TN + 2 * TV / 0.012)$$

$$B2 = KR * TV / 0.012$$

Declaration:

GLOBAL DEFFCT RSIERR ST_PID(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,KR:IN,TV:IN,TN:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN KR (REAL): Control parameter KR

IN TV (REAL): Control parameter TV in [sec]

IN TN (REAL): Control parameter TN in [sec]

Object Parameters:

1: KR (REAL): s. parameter KR

2: TV (REAL): s. parameter TV

3: TN (REAL): s. parameter TN

Object Inputs:

1 (REAL): Input signal 1

Object Outputs:

1 (REAL): Output signal 1

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage

RSILNKOUTINDEX: Invalid index for linkage

RSILNKSIGNAL: Input output signal mismatch

RSILNKCYCLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_PT1

Description:

Creates a RSI object with PT1 control functionality.

$$y(k) = -A0 * y(k-1) + B0 * x(k)$$

with:

$$A0 = -\exp(-0.012 / T1)$$

$$B0 = KR * (1 - \exp(-0.012 / T1))$$

Declaration:

GLOBAL DEFFCT RSIERR ST_PT1(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,KP:IN,T1:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN KP (REAL): Control parameter KR

IN T1 (REAL): Control parameter T1 in [sec]

Object Parameters:

1: KR (REAL): s. parameter KR

2: T1 (REAL): s. parameter T1

Object Inputs:

1 (REAL): Input signal 1

Object Outputs:

1 (REAL): Output signal 1

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage

RSILNKOUTINDEX: Invalid index for linkage

RSILNKSIGNAL: Input output signal mismatch

RSILNKCYCLE: Cyclic linkage

RSINOLNKOBJ: Object not linkable

ST_PT2

Description:

Creates a RSI object with PT2 control functionality.

$$y(k) = -A0 * y(k-1) - A1 * y(k-2) + B0 * x(k) + B1 * x(k-1)$$

with:

Case 1: T1 != T2

$$Z1 = \exp(-0.012 / T1)$$

$$Z2 = \exp(-0.012 / T2)$$

$A0 = -Z1 - Z2$
 $A1 = Z1 * Z2$
 $B0 = (KP / (T1 - T2)) * (T1 * (1 - Z1) - T2 * (1 - Z2))$
 $B1 = (KP / (T1 - T2)) * (T2 * Z1 * (1 - Z2) - T1 * Z2 * (1 - Z1))$
 Case 2: $T1 == T2$
 $Z0 = \exp(-0.012 / T1)$
 $A0 = -2 * Z0$
 $A1 = \text{pow}(Z0,2)$
 $B0 = KP * (1 - Z0 * ((0.012 / T1) + 1))$
 $B1 = KP * Z0 * (Z0 + (0.012 / T1) - 1)$

Declaration:

GLOBAL DEFFCT RSIERR ST_PT2(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,KP:IN,T1:IN,T2:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN KP (REAL): Control parameter KR
 IN T1 (REAL): Control parameter T1 in [sec]
 IN T2 (REAL): Control parameter T2 in [sec]

Object Parameters:

1: KR (REAL): s. parameter KR
 2: T1 (REAL): s. parameter T1
 3: T2 (REAL): s. parameter T2

Object Inputs:

1 (REAL): Input signal 1

Object Outputs:

1 (REAL): Output signal 1

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow
 RSILNKOUTOBJ: Invalid ID for linkage
 RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYCLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_GENCTRL

Description:

Creates a RSI object with generic transfer functionality.

$y(z) = B0 * u(z) + B1 * u(z-1) + B2 * u(z-2) + \dots + B8 * u(z-8)$
 $- A1 * y(z-1) - A2 * y(z-2) - \dots - A8 * y(z-8)$

Declaration:

GLOBAL DEFFCT RSIERR ST_GENCTRL(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1

Object Parameters:

1: RESET (BOOL): Reset the internal values
 2: A1 (REAL): A1 coefficient
 3: A2 (REAL): A2 coefficient
 4: A3 (REAL): A3 coefficient
 5: A4 (REAL): A4 coefficient
 6: A5 (REAL): A5 coefficient

7: A6 (REAL): A6 coefficient
 8: A7 (REAL): A7 coefficient
 9: A8 (REAL): A8 coefficient
 10: B0 (REAL): B0 coefficient
 11: B1 (REAL): B1 coefficient
 12: B2 (REAL): B2 coefficient
 13: B3 (REAL): B3 coefficient
 14: B4 (REAL): B4 coefficient
 15: B5 (REAL): B5 coefficient
 16: B6 (REAL): B6 coefficient
 17: B7 (REAL): B7 coefficient
 18: B8 (REAL): B8 coefficient

Object Inputs:

1 (REAL): input signal 1

Object Outputs:

1 (REAL): output signal 1

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow
 RSILNKOUTOBJ: Invalid ID for linkage
 RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYCLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_IIRFILTER

Description:

Creates a RSI object with IIR filter functionality.

Declaration:

GLOBAL DEFFCT RSIERR ST_IIRFILTER
 (OBJID:OUT,CONTID:IN,IDX1:IN,TYPE:IN,NAME:IN,ORDER:IN,CUTOFFREQ:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN IDX1 (INT): ID of the object to link to input 1
 IN ORDER (INT): Output index of ID1 to link to input 1
 IN TYPE (INT): Type of filter
 -> 0: Lowpass
 -> 1: Highpass
 IN NAME (INT):
 -> 0: Bessel
 -> 1: Butterworth
 -> 2: Tschebyscheff with ripple 0.5 db
 -> 3: Tschebyscheff with ripple 1 db
 -> 4: Tschebyscheff with ripple 2 db
 -> 5: Tschebyscheff with ripple 3 db
 IN ORDER (INT) 2,4,6,8,10
 IN CUTFREQ (REAL):Cutoff frequency in [Hz]

Object Parameters:

1: RESET (BOOL): Reset the internal values

Object Inputs:

1 (REAL): input signal 1

Object Outputs:

1 (REAL): output signal 1

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage
RSILNKOUTINDEX: Invalid index for linkage
RSILNK SIGNAL: Input output signal mismatch
RSILNKCYCLE: Cyclic linkage
RSINOLNKOBJ: Object not linkable

ST_CONTAINER

Description:

Creates a container object. A container is used to group RSI objects. The commands ST_DELOBJ, ST_ENABLE and ST_DISABLE are usable on a containers. RSI objects are assigned to a container at creation time with the second parameter CONTID. A first level container with ID 0 is predefined. Objects created on a second level container are by default disabled.

Declaration:

GLOBAL DEFFCT RSIERR ST_CONTAINER(OBJID:OUT)

Parameters:

OUT OBJID (INT): ID of the created RSI object

Object Parameters:

No

Object Inputs:

No - A container has no signal functionality

Object Outputs:

No - A container has no signal functionality

Return Values:

RSIOK: Successful

RSICONTOVERFLOW: Container overflow

ST_SUM

Description:

Creates a RSI object to sum up its input signals. The object supports up to 5 inputs - at least one input has to be linked. The constant value of CVAL is added to the input sum. If the second input should be unlinked ID2 and IDX2 have to be set to 0.

Declaration:

GLOBAL DEFFCT RSIERR

ST_SUM(OBJID:OUT,CONTID:IN, ID1:IN,IDX1:IN, ID2:IN,IDX2:IN,CVAL:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN ID2 (INT): ID of the object to link to input 2

IN IDX2 (INT): Output index of ID2 to link to input 2

IN CVAL (REAL): Constant value to add

Object Parameters:

1: CVAL (REAL): s. parameter CVAL

Object Inputs:

1 (REAL): Input signal 1

...

5 (REAL): Input signal 5

Object Outputs:

1 (REAL): Addition operation result

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSILNKOUTOBJ: Invalid ID for linkage

RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYCLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_MULTI

Description:

Creates a RSI object to multiply its two input signals.

Declaration:

GLOBAL DEFFCT RSIERR ST_MULTI(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN ID2 (INT): ID of the object to link to input 2
 IN IDX2 (INT): Output index of ID2 to link to input 2

Object Parameters:

No

Object Inputs:

1 (REAL): Input signal 1
 2 (REAL): Input signal 2

Object Outputs:

1 (REAL): Multiplication operation result

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow
 RSILNKOUTOBJ: Invalid ID for linkage
 RSILNKOUTINDEX: Invalid index for linkage
 RSILNKSIGNAL: Input output signal mismatch
 RSILNKCYCLE: Cyclic linkage
 RSINOLNKOBJ: Object not linkable

ST_TIMER

Description:

Creates a RSI object with a bool output flipping to TRUE after a defineable period of time. The timer is running if the Input signal 1 is TRUE.

Declaration:

GLOBAL DEFFCT RSIERR ST_TIMER(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,TIME:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN TIME (REAL): Timer length in [sec]

Object Parameters:

1: RESET (BOOL): Set to TRUE resets the timer
 2: TIME (REAL): s. parameter TIME

Object Inputs:

1 (BOOL): Input signal 1 to activate running

Object Outputs:

1 (BOOL): Timer output - TRUE when timer expired

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter TIME

ST_LIMIT

Description:

Bounds the range of input signal to upper and lower saturation values.

Declaration:

GLOBAL DEFFCT RSIERR

ST_LIMIT(OBJID:OUT,CONTID:IN,IDX1:IN,LOWLIM:IN,UPPLIM:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN LOWLIM (REAL): Lower limit to bound

IN UPPLIM (REAL): Upper limit to bound

Object Parameters:

1: LOWLIM (REAL): s. parameter LOWLIM

2: UPPLIM (REAL): s. parameter UPPLIM

Object Inputs:

1 (REAL): Input signal 1

Object Outputs:

1 (REAL): Bounded output signal 1

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

ST_MINMAX

Description:

Creates a RSI object that finds out the minimal and the maximal signal value of all inputs. The minimal value is written to output 1 and the maximal value is written to the output 2. The inputs 1 and 2 have to be linked at creation time. All further inputs 3 - 10 can be established with ST_NEWLINK().

Declaration:

GLOBAL DEFFCT RSIERR

ST_MINMAX(OBJID:OUT,CONTID:IN,IDX1:IN,IDX2:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN ID2 (INT): ID of the object to link to input 2

IN IDX2 (INT): Output index of ID2 to link to input 2

Object Parameters:

No

Object Inputs:

1 (REAL): Input signal 1

2 (REAL): Input signal 2

3 (REAL): Input signal 3 (optional)

...

10 (REAL): Input signal 10(optional)

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

ST_DELAY

Declaration:

GLOBAL DEFFCT RSIERR ST_DELAY(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,DELAY:IN)

Description:

Creates a RSI object to delay the input signal for a defineable period of time.

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1
 IN DELAY (REAL): Delay time in [sec]

Object Parameters:

No

Object Inputs:

1 (REAL): Input signal 1

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow

ST_ABS

Description:

Creates a RSI object to make a RSI signal positiv.

Declaration:

GLOBAL DEFFCT RSIERR ST_ABS(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1

Object Parameters:

No

Object Inputs:

1 (REAL): Input signal 1

Object Outputs:

1 (REAL): Positiv output signal 1

Return Values:

RSIOK: Successful
 RSIINVCONT: Invalid container
 RSICONTOVERFLOW: Container overflow

ST_SIGNALSWITCH

Description:

Creates a RSI object to switch between two signals in dependance of a control input. If the input 1 is TRUE then the input signal 2 is mapped to the output 1 and the input signal 3 is mapped to the output 2. If input 1 is FALSE then input signal 2 is mapped to the output 2 and the input signal 3 is mapped to the output 1.

Declaration:

GLOBAL DEFFCT RSIERR
 ST_SIGNALSWITCH(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,ID2:IN,IDX2:IN,ID3:IN,IDX3:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
 IN CONTID (INT): ID of container for the new object
 IN ID1 (INT): ID of the object to link to input 1
 IN IDX1 (INT): Output index of ID1 to link to input 1

IN ID2 (INT): ID of the object to link to input 2
IN IDX2 (INT): Output index of ID1 to link to input 2
IN ID3 (INT): ID of the object to link to input 3
IN IDX3 (INT): Output index of ID1 to link to input 3

Object Parameters:

No

Object Inputs:

1 (BOOL): Control input 1
2 (REAL): Input signal 2
3 (REAL): Input signal 3

Object Outputs:

1 (REAL): Output signal 1
2 (REAL): Output signal 2

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow

ST_VEL

Description:

Creates a RSI object to modify the program override \$OV_PRO.

Declaration:

GLOBAL DEFFCT RSIERR ST_VEL(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1

Object Parameters:

No

Object Inputs:

1 (INT): Input signal 1 for value \$OV_PRO (Range 0 - 100)

Object Outputs:

No

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow

ST_STOP

Declaration:

GLOBAL DEFFCT RSIERR ST_STOP(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,MODE:IN)

Description:

Creates a RSI object to stop a movement with a message in the GUI if the input becomes TRUE.

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN MODE (INT): Stop MODE

0 : No stop - only notification

1 : Normal stop

2 : Velocity stop

3 : Fast stop

Object Parameters:

1 MODE (INT): s. parameter MODE

Object Inputs:

1 (BOOL): Input signal 1

Object Outputs:

No

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter MODE

ST_MAP2SEN_PINT

Declaration:

GLOBAL DEFFCT RSIERR

ST_MAP2SEN_PINT(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,VARIDX:IN)

Description:

Creates a RSI object to map the input signal to the KRL variable \$SEN_PINT[VARIDX].

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN VARIDX (INT): Index in \$SEN_PINT array ($\geq 1 < 20$)

Object Parameters:

1 VARIDX (INT): s. parameter VARIDX

Object Inputs:

1 (INT): Input signal 1 to map to \$SEN_PINT[VARIDX]

Object Outputs:

No

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter VARIDX

RSILNKOUTOBJ: invalid output object ID to link (see logging)

RSILNKOUTINDEX: invalid index at output object (see logging)

RSILNK SIGNAL: input and output signal don't match together (see logging)

ST_MAP2SEN_PREA

Declaration:

GLOBAL DEFFCT RSIERR

ST_MAP2SEN_PREA(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,VARIDX:IN)

Description:

Creates a RSI object to map the input signal to the KRL variable \$SEN_PREA[VARIDX].

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN VARIDX (INT): Index in \$SEN_PREA array ($\geq 1 < 20$)

Object Parameters:

1 VARIDX (INT): s. parameter VARIDX

Object Inputs:

1 (REAL): Input signal 1 to map to \$SEN_PREA[VARIDX]

Object Outputs:

No

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSIINVPARAM: Invalid parameter VARIDX
RSILNKOUTOBJ: invalid output object ID to link (see logging)
RSILNKOUTINDEX: invalid index at output object (see logging)
RSILNKSIGNAL: input and output signal don't match together (see logging)
RSIINVPARAM: invalid object parameter Index

ST_MAP2DIGOUT

Description:

Creates a RSI object to map the input signal to the digital outputs \$OUT[].

Declaration:

GLOBAL DEFFCT RSIERR

ST_MAP2DIGOUT(OBJID:OUT,CONTID:IN,IDX1:IN,OUTIDX:IN,LEN:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN OUTIDX (INT): Offset of digital outputs to write
-> in Bit for LEN 0 (Bit) >=1
-> in Byte for LEN 1,2,3 (Byte, Word) >=1
IN LEN (INT): Range of bits to write
-> 0: Bit
-> 1: Byte
-> 2: Word

Object Parameters:

1 OUTIDX (INT): s. parameter OUTIDX
1 LEN (INT): s. parameter LEN

Object Inputs:

1 (INT): Input signal 1 to map to \$OUT[]

Object Outputs:

No

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSIINVPARAM: Invalid parameter
RSILNKOUTOBJ: invalid output object ID to link (see logging)
RSILNKOUTINDEX: invalid index at output object (see logging)
RSILNKSIGNAL: input and output signal don't match together (see logging)
RSIINVPARAM: invalid object parameter (see logging)

ST_MAP2ANAOUT

Description:

Creates a RSI object to map the Input signal to the analog output \$ANOUT[OUTIDX].

Declaration:

GLOBAL DEFFCT RSIERR

ST_MAP2ANAOUT(OBJID:OUT,CONTID:IN,IDX1:IN,OUTIDX:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN OUTIDX (INT): Offset of analog outputs to write >=1

Object Parameters:

1 OUTIDX (INT): s. parameter OUTIDX

Object Inputs:

1 (REAL): Input signal 1 to map to \$ANOUT[OUTIDX].

Object Outputs:

No

Return Type:

RSIERR

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter OUTIDX

RSILNKOUTOBJ: invalid output object ID to link (see logging)

RSILNKOUTINDEX: invalid index at output object (see logging)

RSILNKSIGNAL: input and output signal don't match together (see logging)

RSIINVPARAM: invalid object parameter (see logging)

ST_SETDIGOUT

Description:

Creates a RSI object to set a digital output \$OUT[OUTIDX] to TRUE if the signal input becomes TRUE.

In difference to ST_MAP2DIGOUT

the output stays TRUE independent of the further signal flow.

Declaration:

GLOBAL DEFFCT RSIERR

ST_SETDIGOUT(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,OUTIDX:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1

IN IDX1 (INT): Output index of ID1 to link to input 1

IN OUTIDX (INT): Offset of digital output to write >=1

Object Parameters:

1 OUTIDX (INT): s. parameter OUTIDX

Object Inputs:

1 (BOOL): Input signal 1 to set to \$OUT[OUTIDX]

Object Outputs:

No

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter OUIDX

RSILNKOUTOBJ: invalid output object ID to link (see logging)

RSILNKOUTINDEX: invalid index at output object (see logging)

RSILNKSIGNAL: input and output signal don't match together (see logging)

RSIINVPARAM: invalid object parameter Index

ST_RESETDIGOUT

Description:

Creates a RSI object to set a digital output \$OUT[OUTIDX] to FALSE if the signal input becomes

TRUE. In difference to ST_MAP2DIGOUT

the output stays FALSE independent of the further signal flow.

Declaration:

GLOBAL DEFFCT RSIERR

ST_RESETDIGOUT(OBJID:OUT,CONTID:IN,ID1:IN,IDX1:IN,OUTIDX:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

IN ID1 (INT): ID of the object to link to input 1
IN IDX1 (INT): Output index of ID1 to link to input 1
IN OUTIDX (INT): Offset of digital output to write ≥ 1

Object Parameters:

1 OUTIDX (INT): s. parameter OUTIDX

Object Inputs:

1 (BOOL): Input signal 1 to reset \$OUT[OUTIDX]

Object Outputs:

No

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

RSIINVPARAM: Invalid parameter OUTIDX

RSILNKOUTOBJ: invalid output object ID to link (see logging)

RSILNKOUTINDEX: invalid index at output object (see logging)

RSILNK SIGNAL: input and output signal don't match together (see logging)

RSIINVPARAM: invalid object parameter Index

ST_PATHCORR

Description:

Creates a RSI object to do a cartesian path correction. The object allows to set correction values for the robot position every sensor cycle. The

linkage to this object has to be done with ST_NEWLINK().

The reference coordinate system and

integration mode depends command ST_ON()

or ST_ON1()

Declaration:

GLOBAL DEFFCT RSIERR ST_PATHCORR(OBJID:OUT,CONTID:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

Object Parameters:

1 (REAL): Lower bound of correction in X, Default: -5

2 (REAL): Lower bound of correction in Y, Default: -5

3 (REAL): Lower bound of correction in Z, Default: -5

4 (REAL): Lower bound of correction in A, Default: -5

5 (REAL): Lower bound of correction in B, Default: -5

6 (REAL): Lower bound of correction in C, Default: -5

7 (REAL): Upper bound of correction in X, Default: 5

8 (REAL): Upper bound of correction in Y, Default: 5

9 (REAL): Upper bound of correction in Z, Default: 5

10 (REAL): Upper bound of correction in A, Default: 5

11 (REAL): Upper bound of correction in B, Default: 5

12 (REAL): Upper bound of correction in C, Default: 5

Object Inputs:

1 (REAL): X translation correction value [mm]

2 (REAL): Y translation correction value [mm]

3 (REAL): Z translation correction value [mm]

4 (REAL): A rotation correction value [°]

5 (REAL): B rotation correction value [°]

6 (REAL): C rotation correction value [°]

Object Outputs:

1 (INT): Correction active signal

-> 0: Interpolator not active

-> 1: Correction active

-> 2: Correction out of range

Return Values:

RSIOK: Successful

RSIINVCONT: Invalid container

RSICONTOVERFLOW: Container overflow

ST_AXISCORR

Description:

Creates a RSI object to do a axis correction. The object allows to set correction values for the robot position every sensor cycle. The linkage to this object has to be done with ST_NEWLINK().

The unit depends on the axis type:

Linear: mm

Rotatory: °

Declaration:

GLOBAL DEFFCT RSIERR ST_AXISCORR(OBJID:OUT,CONTID:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object

IN CONTID (INT): ID of container for the new object

Object Parameters:

1 INTEG (INT): Integration mode

-> 0: Absolut

-> 1: Relativ (Default)

2 (REAL): Lower bound of correction A1, Default: -5

3 (REAL): Lower bound of correction A2, Default: -5

4 (REAL): Lower bound of correction A3, Default: -5

5 (REAL): Lower bound of correction A4, Default: -5

6 (REAL): Lower bound of correction A5, Default: -5

7 (REAL): Lower bound of correction A6, Default: -5

8 (REAL): Lower bound of correction E1, Default: -5

9 (REAL): Lower bound of correction E2, Default: -5

10 (REAL): Lower bound of correction E3, Default: -5

11 (REAL): Lower bound of correction E4, Default: -5

12 (REAL): Lower bound of correction E5, Default: -5

13 (REAL): Lower bound of correction E6, Default: -5

14 (REAL): Upper bound of correction A1, Default: 5

15 (REAL): Upper bound of correction A2, Default: 5

16 (REAL): Upper bound of correction A3, Default: 5

17 (REAL): Upper bound of correction A4, Default: 5

18 (REAL): Upper bound of correction A5, Default: 5

19 (REAL): Upper bound of correction A6, Default: 5

20 (REAL): Upper bound of correction E1, Default: 5

21 (REAL): Upper bound of correction E2, Default: 5

22 (REAL): Upper bound of correction E3, Default: 5

23 (REAL): Upper bound of correction E4, Default: 5

24 (REAL): Upper bound of correction E5, Default: 5

25 (REAL): Upper bound of correction E6, Default: 5

Object Inputs:

1 (REAL): A1 correction

2 (REAL): A2 correction

3 (REAL): A3 correction

4 (REAL): A4 correction

5 (REAL): A5 correction

6 (REAL): A6 correction

7 (REAL): E1 correction

...

12(REAL): E12 angle correction

Object Outputs:

1 (INT): Correction active signal

-> 0: Interpolator not active

-> 1: Correction active

-> 2: Correction out of range

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow

ST_MONITOR

Description:

Creates a RSI object to send RSI signal data via TCP/IP the corresponding server application RSIMonitor.exe. The linkage to this object has to be done with ST_NEWLINK()

Declaration:

GLOBAL DEFFCT RSIERR
ST_MONITOR(OBJID:OUT,CONTID:IN,IP[:OUT,PORT:IN,REFRESH:IN)

Parameters:

OUT OBJID (INT): ID of the created RSI object
IN CONTID (INT): ID of container for the new object
IN IP[] (CHAR[]): Target IP address
IN PORT (INT): Target port
IN REFRESH (INT): Send cycle in multiples of 12ms

Object Parameters:

1 MOD (INT): Mode of communication
-> 0: Not sending (Default)
-> 1: Sending

Object Inputs:

1 (BOOL): Control input
-> Linked: TRUE sending / FALSE not sending
-> Unlinked: Always sending
2 (REAL): Data channel 1

...

25 (REAL): Data channel 24

Object Outputs:

No

Return Values:

RSIOK: Successful
RSIINVCONT: Invalid container
RSICONTOVERFLOW: Container overflow
RSIINVPARAM: Invalid parameter

ANEXO 3. Archivo XML para comunicación con ST_ETHERNET


```

<ROOT>
  <CONFIG>
    <IP_NUMBER>172.16.18.210</IP_NUMBER>
    <!-- IP Number of the socket ! -->
    <PORT>6008</PORT>
    <!-- Port Number of the socket ! -->
    <PROTOCOL>TCP</PROTOCOL>
    <!-- TCP or UDP, Protocol of the socket!-->
    <SENSTYPE>ImFree</SENSTYPE>
    <!-- The name of your system send in <Sen Type="" > ! -->
    <PROTCOLLENGTH>Off</PROTCOLLENGTH>
    <!-- On or Off, Send the length of data in bytes before XML data begins! -->
    <ONLYSEND>FALSE</ONLYSEND>
    <!-- TRUE means the client don't expect a answer. Do not send anything to robot! -->
  </CONFIG>
  <!-- RSI Data: TYPE= "BOOL", "STRING", "LONG", "FLOAT", "DOUBLE" ! -->
  <!-- INDX= "INTERNAL" switch on internal read values. Needed by DEF_...! -->
  <!-- INDX= "nmb" Input/Output index of RSI-Object / Maximum of RSI Channels: 64 ! -->
  <!-- UNIT: The same like RSI Units, insert a decimal value. ! -->
  <!-- HOLDON="1", set this output index of RSI Object to the last value ! -->
  <!-- DEF_Delay count the late packages and send it back to server ! -->
  <!-- DEF_Tech: .C = advance .T = main run / .C1 advance set function generator 1 ! -->
  <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" />

<ELEMENT TAG="DEF_EIPos" TYPE="DOUBLE" INDX="INTERNAL"
UNIT="0" />

<ELEMENT TAG="DEF_ESPos" TYPE="DOUBLE"
INDX="INTERNAL" UNIT="0" />

<ELEMENT TAG="DEF_MACur" TYPE="DOUBLE"
INDX="INTERNAL" UNIT="0" />

<ELEMENT TAG="DEF_MECur" TYPE="DOUBLE"
INDX="INTERNAL" UNIT="0" />

<ELEMENT TAG="DEF_Delay" TYPE="LONG" INDX="INTERNAL"
UNIT="0" />

<ELEMENT TAG="DEF_Tech.C1" TYPE="FLOAT"
INDX="INTERNAL" UNIT="0" />

<ELEMENT TAG="DiL" TYPE="LONG" INDX="1" UNIT="0" />

<ELEMENT TAG="Digout.o1" TYPE="BOOL" INDX="2" UNIT="0" />

<ELEMENT TAG="Digout.o2" TYPE="BOOL" INDX="3" UNIT="0" />

<ELEMENT TAG="Digout.o3" TYPE="BOOL" INDX="4" UNIT="0" />

<ELEMENT TAG="ST_Source" TYPE="FLOAT" INDX="5"
UNIT="3601" />

</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" />

<ELEMENT TAG="AKorr.A1" TYPE="DOUBLE" INDX="7" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="AKorr.A2" TYPE="DOUBLE" INDX="8" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="AKorr.A3" TYPE="DOUBLE" INDX="9" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="AKorr.A4" TYPE="DOUBLE" INDX="10" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="AKorr.A5" TYPE="DOUBLE" INDX="11" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="AKorr.A6" TYPE="DOUBLE" INDX="12" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="EKorr.E1" TYPE="DOUBLE" INDX="13" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="EKorr.E2" TYPE="DOUBLE" INDX="14" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="EKorr.E3" TYPE="DOUBLE" INDX="15" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="EKorr.E4" TYPE="DOUBLE" INDX="16" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="EKorr.E5" TYPE="DOUBLE" INDX="17" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="EKorr.E6" TYPE="DOUBLE" INDX="18" UNIT="0"
HOLDON="0" />

<ELEMENT TAG="DEF_Tech.T2" TYPE="FLOAT"
INDX="INTERNAL" UNIT="0" />

<ELEMENT TAG="DiO" TYPE="LONG" INDX="19" UNIT="0"
HOLDON="1" />

</ELEMENTS>

</RECEIVE></ROOT>
```


ANEXO 4. PROGRAMA DEL CONTROLADOR DEL ROBOT


```
&ACCESS RVP

&REL 64

&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe

&PARAM EDITMASK = *

DEF erxdemo_ernesto( )

; =====
;
; EXAMPLE OF: ST_ETHERNET

; Type: RSI Object

;
; =====

;FOLD Overview notes

; =====

; NAME

; KUKA.Ethernet.RSIXML

; COPYRIGHT

; KUKA Robter GmbH

;
; =====

;ENDFOLD

;FOLD INI

;FOLD BASISTECH INI

GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )

INTERRUPT ON 3

BAS (#INITMOV,0 )

;ENDFOLD (BASISTECH INI)
```

```

;FOLD SPOTTECH INI
USERSPOT(#INIT)
;ENDFOLD (SPOTTECH INI)
;FOLD GRIPPERTECH INI
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPERTECH INI)
;FOLD USER INI
;Make your modifications here
;$TOOL = {X 0,Y -130,Z 0,A 0,B 0,C 20}

$TOOL = {X 0,Y 0,Z 0,A 0,B 0,C 0}

;ENDFOLD (USER INI)
;ENDFOLD (INI)

HALT

; The current position will be used as SAK movement!
;PTP $POS_ACT
; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
;HOME = {AXIS: A1 0,A2 -90,A3 90,A4 0,A5 -90,A6 0}
HOME = {AXIS: A1 0,A2 -120,A3 85,A4 0,A5 35,A6 0}
;HOME = {AXIS: A1 0,A2 -90,A3 90,A4 0,A5 0,A6 0}
PTP HOME
; Create RSI Object ST_Ethernet, read object configuration .../INIT/ERXConfig.xml
err = ST_ETHERNET(hEthernet,0,"ERXconfig.xml")
IF (err <> #RSIOK) THEN
HALT

```

```
ENDIF

; err = ST_SETPARAM(hEthernet,eERXmaxLatePackages,500) ; after "value" to late
  packages the robot stopps

; err = ST_SETPARAM(hEthernet,eERXmaxLateInPercent,10) ; RSIWARNING if the
  limit reached

; err = ST_SETPARAM(hEthernet,eERXmaxFieldOfView,1000) ;reset every 'value'
  statistics.

err = ST_SETPARAM(hEthernet, eERXFastCycle, 0) ; FALSE: Time to answer 11ms /
  TRUE: Fast cycle: answer <2ms necessary!

; err = ST_SETPARAM(hEthernet, eERXerrorFlag, 1) ; $FLAG[1] will be set in case off
  errors

;FOLD RSI-Objects to link in ST_Ethernet

; read $IN[1-16]

err = ST_DIGIN(hDin,0,1,2,0)

IF (err <> #RSIOK) THEN

  HALT

ENDIF

err = ST_NEWLINK_IN(hDin,1,hEthernet,1,"DiL")

IF (err <> #RSIOK) THEN

  HALT

ENDIF

; read $OUT[1-3]

err = ST_DIGOUT(hDout1,0,1,0,0)

IF (err <> #RSIOK) THEN

  HALT

ENDIF

err = ST_DIGOUT(hDout2,0,2,0,0)

IF (err <> #RSIOK) THEN

  HALT
```

```
ENDIF
err = ST_DIGOUT(hDout3,0,3,0,0)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_IN(hDout1,1,hEthernet,2,"Digout.o1")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_IN(hDout2,1,hEthernet,3,"Digout.o2")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_IN(hDout3,1,hEthernet,4,"Digout.o3")
IF (err <> #RSIOK) THEN
    HALT
ENDIF

; make sine signal
UNIT_RSI = 3601
err = ST_SOURCE(hsource,0,UNIT_RSI)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_SETPARAM(hsource,1,1)
IF (err <> #RSIOK) THEN
    HALT
```

```
ENDIF
err = ST_SETPARAM(hsource,3,50)
IF (err <> #RSIOK) THEN
  HALT
ENDIF
err = ST_NEWLINK_IN(hsource,1,hEthernet,5,"ST_Source")
IF (err <> #RSIOK) THEN
  HALT
ENDIF
;ENDFOLD

;FOLD RSI-Objects to link out of ST_Ethernet
; link RKorr to correction on path
err = ST_PATHCORR(hPath,0)
IF (err <> #RSIOK) THEN
  HALT
ENDIF

; link AKorr to movement of axis
err = ST_AXISCORR(hAxis,0)
IF (err <> #RSIOK) THEN
  HALT
ENDIF

err = ST_SETPARAM(hPath,1,-1000);350)
err = ST_SETPARAM(hPath,2,-1000);220)
err = ST_SETPARAM(hPath,3,-1000);60)
```

```
err = ST_SETPARAM(hPath,4,-500);30)
err = ST_SETPARAM(hPath,5,-500);30)
err = ST_SETPARAM(hPath,6,-500);30)
err = ST_SETPARAM(hPath,7,1000);40)
err = ST_SETPARAM(hPath,8,1000);220)
err = ST_SETPARAM(hPath,9,500);150)
err = ST_SETPARAM(hPath,10,500);30)
err = ST_SETPARAM(hPath,11,500);10)
err = ST_SETPARAM(hPath,12,500);30)

err = ST_SETPARAM(hAxis,1,0)
err = ST_SETPARAM(hAxis,2,-168)
err = ST_SETPARAM(hAxis,14,168)
err = ST_SETPARAM(hAxis,3,-188)
err = ST_SETPARAM(hAxis,15,43)
err = ST_SETPARAM(hAxis,4,-117)
err = ST_SETPARAM(hAxis,16,150)
err = ST_SETPARAM(hAxis,5,-188)
err = ST_SETPARAM(hAxis,17,188)
err = ST_SETPARAM(hAxis,6,-94)
err = ST_SETPARAM(hAxis,18,94)
err = ST_SETPARAM(hAxis,7,-356)
err = ST_SETPARAM(hAxis,19,356)
err = ST_NEWLINK_OUT(hEthernet,1,hPath,1,"RKorr.X")
IF (err <> #RSIOK) THEN
  HALT
ENDIF
```

```
err = ST_NEWLINK_OUT(hEthernet,2,hPath,2,"RKorr.Y")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,3,hPath,3,"RKorr.Z")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,4,hPath,4,"RKorr.A")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,5,hPath,5,"RKorr.B")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,6,hPath,6,"RKorr.C")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,7,hAxis,1,"AKorr.A1")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,8,hAxis,2,"AKorr.A2")
IF (err <> #RSIOK) THEN
    HALT
```

```
ENDIF
err = ST_NEWLINK_OUT(hEthernet,9,hAxis,3,"AKorr.A3")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,10,hAxis,4,"AKorr.A4")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,11,hAxis,5,"AKorr.A5")
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_NEWLINK_OUT(hEthernet,12,hAxis,6,"AKorr.A6")
IF (err <> #RSIOK) THEN
    HALT
ENDIF

; DiO map to $OUT[8-24]
err = ST_MAP2DIGOUT(hMapDio,0,hEthernet,19,2,2)
IF (err <> #RSIOK) THEN
    HALT
ENDIF

; show RKorr on $SEN_PREA[1-6]
err = ST_MAP2SEN_PREA(hmap,0,hEthernet,1,1)
IF (err <> #RSIOK) THEN
    HALT
```

```
ENDIF
err = ST_MAP2SEN_PREA(hmap,0,hEthernet,2,2)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_MAP2SEN_PREA(hmap,0,hEthernet,3,3)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_MAP2SEN_PREA(hmap,0,hEthernet,4,4)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_MAP2SEN_PREA(hmap,0,hEthernet,5,5)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_MAP2SEN_PREA(hmap,0,hEthernet,6,6)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_MAP2SEN_PREA(hmap,0,hEthernet,7,7)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
err = ST_MAP2SEN_PREA(hmap,0,hEthernet,8,8)
IF (err <> #RSIOK) THEN
```

```
    HALT
ENDIF

err = ST_MAP2SEN_PREA(hmap,0,hEthernet,9,9)
IF (err <> #RSIOK) THEN
    HALT
ENDIF

err = ST_MAP2SEN_PREA(hmap,0,hEthernet,10,10)
IF (err <> #RSIOK) THEN
    HALT
ENDIF

err = ST_MAP2SEN_PREA(hmap,0,hEthernet,11,11)
IF (err <> #RSIOK) THEN
    HALT
ENDIF

err = ST_MAP2SEN_PREA(hmap,0,hEthernet,12,12)
IF (err <> #RSIOK) THEN
    HALT
ENDIF

; show DiO on $SEN_PINT[1]
err = ST_MAP2SEN_PINT(hmap,0,hEthernet,19,1)
IF (err <> #RSIOK) THEN
    HALT
ENDIF
;ENDFOLD

err = ST_ON1(#BASE,0)
```

```

;err = ST_ON1(#TCP,0)

;err = ST_ON1(#WORLD,0)
IF (err <> #RSIOK) THEN
  HALT
ENDIF

;*****
ST_SKIPSENSQ ;Hold on - until RSI-Break reason occur

;*****

PTP $POS_ACT

END

```