



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Núcleo de control y diseño de controladores modulares en entornos distribuidos

Tesis doctoral

Valencia, Julio 2011

Autor:

Raúl Simarro Fernández

Directores:

Dr. José Luis Navarro Herrero

Dr. José Enrique Simó Ten

Instituto de Automática e Informática Industrial

Universidad Politécnica de Valencia

A mi padre,
al que echo de menos

Agradecimientos

En primer lugar quiero agradecer a mis directores de tesis José Simó y José Luis Navarro por su apoyo y la paciencia que han tenido conmigo. Gracias a ellos trabajo como profesor e investigador, y me dedico a algo que me encanta: la docencia y “el cacharreo”. Espero que la colaboración entre nosotros pueda durar mucho tiempo más. Gracias.

A Pedro Albertos, por el interés mostrado en mi trabajo y por sus consejos y ánimos.

A mis compañeros del Departamento de Ingeniería de Sistemas y Automática (DISA), por hacerme partícipe de la impartición de tantas asignaturas y de temáticas tan diferentes durante estos años. Voy a echar de menos tanta carga docente ;)

A mis amigos del Departamento de Ingeniería de Sistemas y Computadores (DISCA): Patricia, Silvia, Paco, Salva, Vicent, Jopolu... he compartido muchos momentos buenos y espero seguir haciéndolo.

A mis padres Juan y Adela, por su cariño y confianza.

Y por último, y no menos importantes, a Pili y Marcos por aguantar mis agobios,...algunos producidos por la tesis. Muchísimas gracias.

Valencia, Julio 2011

Resumen

Los sistemas empotrados tienen un amplio rango de aplicabilidad en muchos sectores y su importancia crece continuamente. Uno de los campos de aplicación incluye la realización de tareas de control.

La heterogeneidad de los sistemas actuales, formados por múltiples componentes de diferentes características conectados en red, sugiere el desarrollo de sistemas de control distribuido en los que las distintas funciones del control se implementen a diferentes niveles.

La consideración de sistemas distribuidos con presencia de redes de comunicación, unido a potencia de cálculo limitada, implica la necesidad de considerar la realización del control en condiciones no convencionales, tales como un control local que garantice la seguridad, controles alternativos en función de la disponibilidad de recursos, activación de distintos modos de funcionamiento que garanticen una degradación admisible de prestaciones ante la presencia de retardos, pérdidas de medidas o tiempo excesivo de cálculo.

El núcleo de control, asimilable al núcleo de un sistema operativo, se define como el código mínimo que debe ejecutarse en una aplicación de control para que el funcionamiento sea seguro, aunque pueda presentar una fuerte degradación de prestaciones o incluso evolucionar hacia un estado seguro de desconexión. El núcleo de control permite la modularidad y adaptación del sistema, así como la capacidad de desarrollo rápido de aplicaciones de control mediante servicios de soporte (*middleware*), necesarios para ofrecer a los algoritmos de control soporte para sistemas distribuidos, computación ubicua, movilidad de código y restricciones de tiempo real.

El objetivo de la tesis es la creación de una estrategia de control distribuida, con elementos empotrados, utilizando el núcleo de control, en el que se utilicen controladores digitales de altas prestaciones en sistemas con capacidad de cómputo limitada. Además, el sistema de control debe hacer frente a los problemas mencionados anteriormente de pérdida de datos, retardos o tiempo excesivo de cálculo, consiguiendo una degradación admisible de prestaciones.

En la tesis se propone una metodología, tanto para la obtención de una métrica que permita la comparación de los distintos modos de funcionamiento ante pérdidas de datos, como para la elección de los parámetros de los controladores a implementar en cada nodo del sistema distribuido de control. Esta metodología ha sido probada mediante una herramienta de simulación y análisis, creada a tal efecto, en procesos con distinta dinámica. Además, también se ha evaluado el sistema distribuido de control sobre varios procesos reales, lo que ha permitido la validación de la herramienta de simulación así como las prestaciones conseguidas ante pérdida de datos en las comunicaciones.

Resum

Els sistemes encastats tenen un ampli rang d'aplicabilitat en molts sectors i la seua importància creix contínuament. Un dels camps d'aplicació inclou la realització de tasques de control.

L'heterogeneïtat dels sistemes actuals, formats per múltiples components de diferents característiques connectats en xarxa, suggereix el desenrotllament de sistemes de control distribuït en els que les distintes funcions del control s'implementen a diferents nivells.

La consideració de sistemes distribuïts amb presència de xarxes de comunicació, unit a potència de càlcul limitada, implica la necessitat considerar la realització del control en condicions no convencionals, com ara un control local que garantisca la seguretat, controls alternatius en funció de la disponibilitat de recursos, activació de distints modes de funcionament que garantisquen una degradació admissible de prestacions davant de la presència de reptardos, pèrdues de mesures o temps excessiu de càlcul.

El nucli de control, assimilable al nucli d'un sistema operatiu, es defineix com el codi mínim que ha d'executar-se en una aplicació de control perquè el funcionament siga segur, encara que pugua presentar una forta degradació de prestacions o inclús evolucionar cap a un estat segur de desconnexió. El nucli de control permet la modularidad i adaptació del sistema, així com la capacitat de desenrotllament ràpid d'aplicacions de control per mitjà de servicis de suport (*middleware*), necessaris per a oferir als algorismes de control suport per a sistemes distribuïts, computació ubiqa, mobilitat de codi i restriccions de temps real.

L'objectiu de la tesi és la creació d'una estratègia de control distribuïda, amb elements encastats, utilitzant el nucli de control, en el que s'utilitzen controladors digitals d'altres prestacions en sistemes amb capacitat de còmput limitada. A més, el sistema de control ha de fer front als problemes mencionats anteriorment de pèrdua de dades, reptardos o temps excessiu de càlcul, aconseguint una degradació admissible de prestacions.

En la tesi es proposa una metodologia, tant per a l'obtenció d'una mètrica que permeta la comparació dels distints modes de funcionament davant de pèrdues de dades, com per a l'elecció dels paràmetres dels controladors a implementar en cada node del sistema distribuït de control. Esta metodologia ha sigut provada per mitjà d'una ferramenta de simulació i anàlisi, creada a este efecte, en processos amb distinta dinàmica. A més, també s'ha avaluat el sistema distribuït de control sobre diversos processos reals, la qual cosa ha permés la validació de la ferramenta de simulació així com les prestacions aconseguides davant de pèrdua de dades en les comunicacions.

Abstract

Embedded systems have a wide range of applications in many sectors and their importance is continually growing. One field of application includes control tasks. The heterogeneity of current systems consist of multiple components from different networked features, suggests the development of distributed control systems in which the various control functions are implemented at different levels.

Considering the presence of distributed systems with communication networks, coupled with limited computing power, implies the need to consider the performance of control in unconventional conditions, such as a local control to ensure safety, alternative controls to the availability resources, activation of different modes of operation to ensure acceptable performance degradation in the presence of delays, losses, actions or excessive computation time.

The control kernel, comparable to the kernel of an operating system, is defined as the minimum code to be executed in a control application for safe operation, although it may produce a strong performance degradation or even evolve into a state insurance disconnection. Control kernel allows modularity and adaptation of the system and the capacity for rapid development of control applications with support services (middleware) necessary to provide support control algorithms for distributed systems, ubiquitous computing, code mobility and real-time constraints.

The objective of this thesis is the creation of a distributed control strategy, with embedded items, using the control kernel, which uses high-performance digital controllers in systems with limited computing power. Furthermore, the control system must address the aforementioned problems of data loss, delays or excessive computation time, achieving acceptable performance degradation.

The thesis proposes a methodology, both to obtain a metric that allows comparison of different modes of operation for any loss of data, and for the choice of parameters of the controllers to be implemented in each node of the distributed system control. This methodology has been tested by simulation and analysis tool, created for this purpose, in processes with different dynamics. In addition, we evaluate the distributed control system on several real processes, which has permitted the validation of the simulation tool as well as supplies collected to data loss in communications.

Contenido

Parte I. Introducción	1
1 INTRODUCCIÓN.....	3
1.1 Objetivos.....	3
1.2 Motivación	4
1.3 Contribuciones de la tesis.....	8
1.4 Organización de la tesis	8
Parte II. Estado del Arte	11
2 SISTEMAS EMPOTRADOS DE CONTROL.....	13
2.1 Motivación y objetivos del capítulo.....	13
2.2 Sistemas empotrados de control. Introducción	13
2.3 Características de los sistemas empotrados de control.....	15
2.4 Sistemas ciber-físicos	18
2.5 Sistemas empotrados y sensores/actuadores inteligentes.....	20
2.5.1 Redes inalámbricas de sensores (WSN) y Redes inalámbricas de sensores y actuadores (WSAN)	21
2.5.2 Sistemas de inteligencia ambiental.....	22
2.6 Núcleo de control.....	23
2.7 Resumen y conclusiones.....	24
3 SISTEMA DISTRIBUIDO DE CONTROL	27
3.1 Motivación y objetivos del capítulo.....	27
3.2 Sistemas distribuidos de control. Introducción.....	27
3.3 Control empotrado distribuido	33
3.4 Control inteligente distribuido	34

3.4.1	Control inteligente distribuido basado en agentes	34
3.4.2	Características de tiempo real de los procesos continuos.....	35
3.4.3	Representación del conocimiento en los sistemas de control inteligente distribuido.....	35
3.5	Comunicaciones en el sistema distribuido de control	36
3.6	Middleware de control	37
3.6.1	Principales middlewares dedicados al control	40
3.6.2	Requisitos a cumplir en el middleware de control. Consideraciones de la arquitectura.....	42
3.7	Resumen y conclusiones del capítulo	43
Parte III. Propuesta de Diseño de Reguladores Modulares en un Entorno Distribuido		
.....		45
4	NÚCLEO DE CONTROL DISTRIBUIDO	47
4.1	Motivación y objetivos del capítulo.....	47
4.2	Núcleo de control.....	48
4.2.1	Definición y principales características del núcleo de control	48
4.2.2	Elementos y arquitectura del núcleo de control.....	49
4.2.3	Middleware del núcleo de control.....	50
4.3	Núcleo de control distribuido	53
4.3.1	Middleware completo del núcleo de control	54
4.3.2	Middleware reducido del núcleo de control.....	57
4.3.3	Estructura del núcleo de control distribuido	59
4.4	Escenario del problema	62
4.4.1	Parámetros del modelo.....	62
4.4.2	Características de las comunicaciones	65
4.5	Estrategia de control utilizando el núcleo de control distribuido.....	68

4.5.1	Tratamiento de distintos periodos entre los nodos.....	68
4.5.2	Sincronización de los nodos del sistema distribuido	76
4.5.3	Estructura de control del nodo supervisor	76
4.5.4	Estructura de control del nodo local	77
4.5.5	Tratamiento de los errores de modelado y el ruido. Compensación de la acción de control	82
4.5.6	Modos de funcionamiento	85
4.5.7	Tratamiento de los retardos y desfases en la comunicación.....	96
4.6	Metodología de diseño para la elección de los parámetros de control en el sistema distribuido.....	99
4.6.1	Elección de los parámetros del control MBPC en el nodo supervisor	100
4.6.2	Elección de la constante K de compensación de la acción de control en el nodo local.....	102
4.7	Resumen y conclusiones del capítulo	104
5	PRUEBAS EN SIMULACIÓN.....	107
5.1	Métodos de estimación de la calidad de la respuesta de control	107
5.1.1	Parámetros descriptivos de la evolución temporal de la variable controlada	107
5.1.2	Medidas de la desviación de la variable controlada respecto de la referencia	108
5.1.3	Método propuesto para estimar la calidad de la respuesta de control	110
5.2	Ejemplos de implementación	112
5.2.1	Pruebas realizadas en el proceso de segundo orden	113
5.2.2	Comparación de las distintas pruebas para los procesos en estudio....	129
5.3	Influencia de las condiciones del entorno en el funcionamiento del sistema distribuido de control.....	133
5.3.1	Discrepancia del modelo respecto al proceso real	133
5.3.2	Ancho de banda consumido.....	137

5.3.3	Tiempo de cómputo del controlador	145
5.3.4	Perturbaciones a la salida del sistema.....	150
5.4	Resumen y conclusiones del capítulo	155
6	APLICACIONES.....	157
6.1	Introducción.....	157
6.2	Aplicación de la conmutación de controladores en un sistema robótico ..	157
6.3	Sistema distribuido de control utilizando el núcleo de control y los distintos modos de funcionamiento.....	160
6.3.1	Proceso de segundo orden subamortiguado	160
6.3.2	Control de posición de un motor de corriente continua	172
6.4	Resumen y conclusiones del capítulo	182
Parte IV. Conclusiones		183
7	CONCLUSIONES Y TRABAJO FUTURO	185
7.1	Conclusiones	185
7.2	Trabajo futuro	186
Parte V. Bibliografía		189
Bibliografía.....		191
Parte VI. Anexos.....		199
Anexo A. Herramienta de cálculo y simulación del sistema distribuido de control		201
A.1	Características de la herramienta	202
A.1.1	Programa principal	203
A.1.2	Cálculo del control predictivo	204
A.1.3	Simulación.....	204
A.2	Modelo en simulación	204
A.2.1	Sistema distribuido con nodos fuertemente interconectados	205

A.2.2	Sistema distribuido con canal de comunicaciones compartido.....	206
A.3	Limitaciones de la herramienta de simulación.....	208
Anexo B. Pruebas realizadas con el simulador.....		209
B.1	Proceso de primer orden	209
B.1.1	Elección de los parámetros del sistema distribuido de control	209
B.1.2	Comportamiento del sistema distribuido de control	215
B.2	Proceso de fase no mínima	221
B.2.1	Elección de los parámetros del sistema distribuido de control	221
B.2.2	Comportamiento del sistema distribuido de control	226
B.3	Proceso con retardo múltiplo del periodo de muestreo.....	233
B.3.1	Elección de los parámetros del sistema distribuido de control	233
B.3.2	Comportamiento del sistema distribuido de control	238
B.4	Proceso con efecto integral	244
B.4.1	Elección de los parámetros del sistema distribuido de control	245
B.4.2	Comportamiento del sistema distribuido de control	249
Anexo C. Utilización de controladores predictivos en el nodo supervisor		257
C.1	Introducción.....	257
C.2	Características del control predictivo basado en modelo	258
C.3	Estrategia del MBPC	259
C.4	Controladores predictivos GPC y CRHPC	261
C.5	Controladores predictivos con restricciones	267
Anexo D. Método de interpolación basado en spline cúbica natural		271

Figuras

Figura 1. Sistema de control basado en red sin control local	4
Figura 2. Esquema Truetime del sistema de control basado en red sin control local.....	5
Figura 3. Comportamiento del sistema sin fallos en la comunicación.....	6
Figura 4. Comportamiento del sistema con fallos en la comunicación	6
Figura 5. Estructura de un sistema empotrado de control. Fuente: (P. Albertos, y otros 2005).....	16
Figura 6. Arquitectura física de las WSAAN (Akyildiz y Kasimoglu, <i>Wireless Sensor and Actor Networks: Research Challenges 2004</i>).....	22
Figura 7. Evolución histórica de los sistemas de control distribuidos	29
Figura 8. Control supervisado.....	29
Figura 9. Control directo.....	30
Figura 10. Control redundante (dual)	31
Figura 11. Control jerárquico.....	31
Figura 12. Control distribuido.....	32
Figura 13. Software basado en componentes para un control digital. Fuente: adaptación de (Baliga y Kumar 2005).	38
Figura 14. Sistema de control en red. Fuente: adaptación de (Baliga y Kumar 2005).....	39
Figura 15. Arquitectura del núcleo de control	50
Figura 16. Componentes del middleware completo del núcleo de control.....	55
Figura 17. Interfaces de los componentes del middleware completo del núcleo de control (parte 1)	57
Figura 18. Interfaces de los componentes del middleware completo del núcleo de control (parte 2)	57
Figura 19. Componentes del middleware reducido del núcleo de control.....	58
Figura 20. Interfaces de los componentes del middleware reducido del núcleo de control	59
Figura 21. Arquitectura para el modelo de control en un sistema distribuido	60
Figura 22. Interpolación de los datos de salida proporcionados por el nodo supervisor.....	70
Figura 23. Adaptación de los datos cuando $T_{CS} > T_{CL}$, con T_{CL} divisor de T_{CS}	72
Figura 24. Ajuste de los datos de la acción de control proporcionados por el nodo supervisor	74
Figura 25. Adaptación de los datos cuando $T_{CS} < T_{CL}$, con T_{CL} múltiplo de T_{CS}	75
Figura 26. Salida y acción de control postuladas en el instante $k=1$	80

Figura 27. Salida y acción de control postuladas en el instante $k=2$	81
Figura 28. Salida y acción de control postuladas en el instante $k=3$	81
Figura 29. Nodo local con compensación de la acción de control	84
Figura 30. Modo 1 de funcionamiento	86
Figura 31. Modo 2 de funcionamiento	87
Figura 32. Modo 3 de funcionamiento	90
Figura 33. Modo 4 de funcionamiento	94
Figura 34. Actualización de los vectores de datos ante retardos y desfases en la comunicación	97
Figura 35. Metodología para la elección de los parámetros del sistema de control	100
Figura 36. Valor mínimo de IAE para cada N y el Nu que hay que aplicar para conseguirlo.....	102
Figura 37. Indicadores de error para distintos valores de K aplicados a un proceso de 2º orden subamortiguado	103
Figura 38. Indicadores de error para distintos valores de K aplicados a un proceso con retardo múltiplo del periodo.....	104
Figura 39. Régimen transitorio y permanente.....	108
Figura 40. Curvas de errores IAE máximos y medios cometidos por cada porcentaje de pérdidas.....	112
Figura 41. Circuito electrónico del proceso de segundo orden subamortiguado	113
Figura 42. Diagrama de bloques del proceso utilizado en las pruebas.....	113
Figura 43. Proceso de segundo orden con el controlador GPC.....	115
Figura 44. Estudio de Nu en función de N con el que se obtiene el mínimo error	116
Figura 45. Errores cometidos en función de Nu para $N=12$	116
Figura 46. Errores cometidos en función de K para $N=12$ y $Nu=8$	117
Figura 47. Proceso de segundo orden con el controlador CRHPC.....	118
Figura 48. Estudio del valor de Nu en función de $N+m$ con el que se obtiene el mínimo error	119
Figura 49. Errores cometidos en función de Nu para $N+m=10$, con $m=2$	120
Figura 50. Errores cometidos en función de K para $N=8$, $m=2$ y $Nu=8$	120
Figura 51. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....	122
Figura 52. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC	122

<i>Figura 53. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>123</i>
<i>Figura 54. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>124</i>
<i>Figura 55. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>124</i>
<i>Figura 56. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>125</i>
<i>Figura 57. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>126</i>
<i>Figura 58. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>126</i>
<i>Figura 59. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>127</i>
<i>Figura 60. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>128</i>
<i>Figura 61. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>128</i>
<i>Figura 62. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>129</i>
<i>Figura 63. IAE máximos para cada porcentaje de pérdida de datos cometidos por los distintos procesos en el modo de funcionamiento 2</i>	<i>135</i>
<i>Figura 64. IAE máximos para cada porcentaje de pérdida de datos cometidos por los distintos procesos en el modo de funcionamiento 3</i>	<i>136</i>
<i>Figura 65. IAE máximos para cada porcentaje de pérdida de datos cometidos por los distintos procesos en el modo de funcionamiento 4</i>	<i>136</i>
<i>Figura 66. IAE máximos para cada porcentaje de pérdida de datos cometidos por los distintos procesos en el modo de funcionamiento 5</i>	<i>137</i>
<i>Figura 67. Errores cometidos en función del ancho de banda dedicada a las tareas de control para $N=8$, $m=2$ y $Nu=8$</i>	<i>138</i>
<i>Figura 68. Activación de las tareas dedicadas al control en cada nodo para un ancho de banda del 90%</i>	<i>139</i>
<i>Figura 69. Comportamiento de la salida del proceso y la acción de control aplicada para un ancho de banda del 90%.....</i>	<i>139</i>
<i>Figura 70. Activación de las tareas dedicadas al control en cada nodo para un ancho de banda del 10%</i>	<i>140</i>
<i>Figura 71. Comportamiento de la salida del proceso y la acción de control aplicada para un ancho de banda del 10%.....</i>	<i>140</i>

<i>Figura 72. Influencia del ancho de banda utilizando el modo 2. IAE máximo</i>	141
<i>Figura 73. Influencia del ancho de banda utilizando el modo 2. IAE medio</i>	142
<i>Figura 74. Influencia del ancho de banda utilizando el modo 3. IAE máximo</i>	142
<i>Figura 75. Influencia del ancho de banda utilizando el modo 3. IAE medio</i>	143
<i>Figura 76. Influencia del ancho de banda utilizando el modo 4. IAE máximo</i>	143
<i>Figura 77. Influencia del ancho de banda utilizando el modo 4. IAE medio</i>	144
<i>Figura 78. Influencia del ancho de banda utilizando el modo 5. IAE máximo</i>	144
<i>Figura 79. Influencia del ancho de banda utilizando el modo 5. IAE medio</i>	145
<i>Figura 80. Influencia del tiempo de cómputo utilizando el modo 2. IAE máximo</i>	146
<i>Figura 81. Influencia del tiempo de cómputo utilizando el modo 2. IAE medio</i>	147
<i>Figura 82. Influencia del tiempo de cómputo utilizando el modo 3. IAE máximo</i>	147
<i>Figura 83. Influencia del tiempo de cómputo utilizando el modo 3. IAE medio</i>	148
<i>Figura 84. Influencia del tiempo de cómputo utilizando el modo 4. IAE máximo</i>	148
<i>Figura 85. Influencia del tiempo de cómputo utilizando el modo 4. IAE medio</i>	149
<i>Figura 86. Influencia del tiempo de cómputo utilizando el modo 5. IAE máximo</i>	149
<i>Figura 87. Influencia del tiempo de cómputo utilizando el modo 5. IAE medio</i>	150
<i>Figura 88. Influencia de perturbaciones utilizando el modo 2. IAE máximo</i>	151
<i>Figura 89. Influencia de perturbaciones utilizando el modo 2. IAE medio</i>	151
<i>Figura 90. Influencia de perturbaciones utilizando el modo 3. IAE máximo</i>	152
<i>Figura 91. Influencia de perturbaciones utilizando el modo 3. IAE medio</i>	152
<i>Figura 92. Influencia de perturbaciones utilizando el modo 4. IAE máximo</i>	153
<i>Figura 93. Influencia de perturbaciones utilizando el modo 4. IAE medio</i>	153
<i>Figura 94. Influencia de perturbaciones utilizando el modo 5. IAE máximo</i>	154
<i>Figura 95. Influencia de perturbaciones utilizando el modo 5. IAE medio</i>	154
<i>Figura 96. Estructura distribuida basada en comportamientos</i>	158
<i>Figura 97. (a) Seguimiento de una trayectoria con dos obstáculos circulares no planificados. (b) Alcanzar un punto objetivo en un entorno no estructurado</i>	159
<i>Figura 98. Proceso electrónico de 2º orden subamortiguado</i>	161
<i>Figura 99. XScale utilizado como nodo supervisor</i>	162

<i>Figura 100. dsPIC de Michochip con la placa de desarrollo que forma el nodo local</i>	162
<i>Figura 101. Estructura de control distribuida para la realización de la prueba</i>	163
<i>Figura 102. Salida del proceso real con el sistema distribuido de control sin pérdida de datos</i>	164
<i>Figura 103. IAE de los distintos modos de funcionamiento en la prueba real</i>	164
<i>Figura 104. Comparación de los IAE máximos para cada modo de funcionamiento en la prueba real</i> ...	165
<i>Figura 105. Comparación de los IAE medios para cada modo de funcionamiento en la prueba real</i>	165
<i>Figura 106. Código de colores utilizado en las representaciones de las pruebas</i>	166
<i>Figura 107. Comportamiento ante pérdida de datos utilizando el modo 2</i>	166
<i>Figura 108. Comportamiento ante pérdida de datos utilizando el modo 3</i>	167
<i>Figura 109. Comportamiento ante pérdida de datos utilizando el modo 4</i>	167
<i>Figura 110. Comportamiento ante pérdida de datos utilizando el modo 5</i>	168
<i>Figura 111. Respuesta real del proceso ante pérdidas de datos utilizando los distintos modos de funcionamiento</i>	168
<i>Figura 112. Comportamiento ante pérdida de datos utilizando el modo 2</i>	169
<i>Figura 113. Comportamiento ante pérdida de datos utilizando el modo 3</i>	170
<i>Figura 114. Comportamiento ante pérdida de datos utilizando el modo 4</i>	170
<i>Figura 115. Comportamiento ante pérdida de datos utilizando el modo 5</i>	171
<i>Figura 116. Respuesta real del proceso ante pérdidas de datos utilizando los distintos modos de funcionamiento</i>	171
<i>Figura 117. Estructura de control distribuida utilizada en la prueba</i>	172
<i>Figura 118. Motor de corriente continua utilizado para el control de posición</i>	172
<i>Figura 119. Comportamiento del proceso real con el sistema distribuido de control sin pérdida de datos</i>	173
<i>Figura 120. IAE de los distintos modos de funcionamiento en la prueba real</i>	174
<i>Figura 121. IAE máximos para cada modo de funcionamiento en la prueba real</i>	174
<i>Figura 122. IAE medios para cada modo de funcionamiento en la prueba real</i>	175
<i>Figura 123. Comportamiento ante pérdida de datos utilizando el modo 2</i>	176
<i>Figura 124. Comportamiento ante pérdida de datos utilizando el modo 3</i>	176
<i>Figura 125. Comportamiento ante pérdida de datos utilizando el modo 4</i>	177
<i>Figura 126. Comportamiento ante pérdida de datos utilizando el modo 5</i>	177

<i>Figura 127. Respuesta real del proceso ante pérdidas de datos utilizando los distintos modos de funcionamiento</i>	<i>178</i>
<i>Figura 128. Comportamiento ante pérdida de datos utilizando el modo 2.....</i>	<i>179</i>
<i>Figura 129. Comportamiento ante pérdida de datos utilizando el modo 3.....</i>	<i>179</i>
<i>Figura 130. Comportamiento ante pérdida de datos utilizando el modo 4.....</i>	<i>180</i>
<i>Figura 131. Comportamiento ante pérdida de datos utilizando el modo 5.....</i>	<i>180</i>
<i>Figura 132. Respuesta real del proceso ante pérdidas de datos utilizando los distintos modos de funcionamiento</i>	<i>181</i>
<i>Figura 133. Bloques funcionales de la herramienta</i>	<i>202</i>
<i>Figura 134. Modelo del sistema utilizado en simulación con nodos fuertemente interconectados.....</i>	<i>205</i>
<i>Figura 135. Modelo del sistema utilizado en simulación con canal de comunicaciones compartido.....</i>	<i>207</i>
<i>Figura 136. Proceso de primer orden con el controlador GPC.....</i>	<i>210</i>
<i>Figura 137. Valor de Nu en función de N con el que se obtiene el mínimo error para el proceso de primer orden</i>	<i>211</i>
<i>Figura 138. Errores cometidos en función de Nu para N=8</i>	<i>211</i>
<i>Figura 139. Errores cometidos en función de K para N=8 y Nu=6</i>	<i>212</i>
<i>Figura 140. Proceso de primer orden con el controlador CRHPC.....</i>	<i>213</i>
<i>Figura 141. Estudio del valor de Nu en función de N+m con el que se obtiene el mínimo error</i>	<i>213</i>
<i>Figura 142. Errores cometidos en función de Nu para N+m=8 instantes, con m=2.....</i>	<i>214</i>
<i>Figura 143. Errores cometidos en función de K para N=6, m=2 y Nu=4</i>	<i>214</i>
<i>Figura 144. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>215</i>
<i>Figura 145. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>216</i>
<i>Figura 146. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>216</i>
<i>Figura 147. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>217</i>
<i>Figura 148. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>217</i>
<i>Figura 149. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>218</i>
<i>Figura 150. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>218</i>

<i>Figura 151. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>219</i>
<i>Figura 152. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>219</i>
<i>Figura 153. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>220</i>
<i>Figura 154. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>220</i>
<i>Figura 155. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>221</i>
<i>Figura 156. Proceso de fase no mínima con el controlador GPC</i>	<i>222</i>
<i>Figura 157. Estudio del valor de Nu en función de N con el que se obtiene el mínimo error.....</i>	<i>222</i>
<i>Figura 158. Errores cometidos en función de Nu para N=14</i>	<i>223</i>
<i>Figura 159. Errores cometidos en función de K para N=14 y Nu=6</i>	<i>223</i>
<i>Figura 160. Proceso de fase no mínima con el controlador CRHPC.....</i>	<i>224</i>
<i>Figura 161. Estudio del valor de Nu en función de N+m con el que se obtiene el mínimo error</i>	<i>225</i>
<i>Figura 162. Errores cometidos en función de Nu para N+m=14, con m=2</i>	<i>225</i>
<i>Figura 163. Errores cometidos en función de K para N=12, m=2 y Nu=10</i>	<i>226</i>
<i>Figura 164. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>227</i>
<i>Figura 165. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>227</i>
<i>Figura 166. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>228</i>
<i>Figura 167. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>228</i>
<i>Figura 168. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>229</i>
<i>Figura 169. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>229</i>
<i>Figura 170. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>230</i>
<i>Figura 171. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>230</i>
<i>Figura 172. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>231</i>

<i>Figura 173. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>231</i>
<i>Figura 174. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>232</i>
<i>Figura 175. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>232</i>
<i>Figura 176. Proceso con retardo con el controlador GPC.....</i>	<i>233</i>
<i>Figura 177. Estudio del valor de Nu en función de N con el que se obtiene el mínimo error.....</i>	<i>234</i>
<i>Figura 178. Errores cometidos en función de Nu para N=15</i>	<i>234</i>
<i>Figura 179. Errores cometidos en función de K para N=15 y Nu=6</i>	<i>235</i>
<i>Figura 180. Proceso con retardo con el controlador CRHPC.....</i>	<i>236</i>
<i>Figura 181. Estudio del valor de Nu en función de N+m con el que se obtiene el mínimo error</i>	<i>236</i>
<i>Figura 182. Errores cometidos en función de Nu para N+m=15, con m=2</i>	<i>237</i>
<i>Figura 183. Errores cometidos en función de K para N=13, m=2 y Nu=5</i>	<i>237</i>
<i>Figura 184. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>238</i>
<i>Figura 185. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>239</i>
<i>Figura 186. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>239</i>
<i>Figura 187. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>240</i>
<i>Figura 188. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>240</i>
<i>Figura 189. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>241</i>
<i>Figura 190. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>241</i>
<i>Figura 191. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>242</i>
<i>Figura 192. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>242</i>
<i>Figura 193. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>243</i>
<i>Figura 194. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>243</i>

<i>Figura 195. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>244</i>
<i>Figura 196. Proceso con efecto integral con el controlador GPC.....</i>	<i>245</i>
<i>Figura 197. Estudio del valor de Nu en función de N con el que se obtiene el mínimo error.....</i>	<i>246</i>
<i>Figura 198. Errores cometidos en función de Nu para N=12</i>	<i>246</i>
<i>Figura 199. Errores cometidos en función de K para N=12 y Nu=3</i>	<i>247</i>
<i>Figura 200. Proceso con efecto integral con el controlador CRHPC</i>	<i>247</i>
<i>Figura 201. Estudio del valor de Nu en función de N+m con el que se obtiene el mínimo error</i>	<i>248</i>
<i>Figura 202. Errores cometidos en función de Nu para N+m=8, con m=2</i>	<i>248</i>
<i>Figura 203. Errores cometidos en función de K para N=6, m=2 y Nu=6</i>	<i>249</i>
<i>Figura 204. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>250</i>
<i>Figura 205. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>250</i>
<i>Figura 206. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>251</i>
<i>Figura 207. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>251</i>
<i>Figura 208. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>252</i>
<i>Figura 209. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>252</i>
<i>Figura 210. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC.....</i>	<i>253</i>
<i>Figura 211. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>253</i>
<i>Figura 212. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC</i>	<i>254</i>
<i>Figura 213. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC.....</i>	<i>254</i>
<i>Figura 214. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>255</i>
<i>Figura 215. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC</i>	<i>255</i>
<i>Figura 216. Estrategia del MBPC.....</i>	<i>259</i>

<i>Figura 217. Estructura básica de un MPC.....</i>	<i>261</i>
<i>Figura 218. Constrained Receding-Horizon Predictive Control.....</i>	<i>264</i>

Tablas

<i>Tabla 1. Vector con la trayectoria de las acciones de control que almacena el nodo local en el modo 3 de funcionamiento</i>	<i>88</i>
<i>Tabla 2. Actualización del vector con la trayectoria de las acciones de control que almacena el nodo local cuando se pierde un dato</i>	<i>90</i>
<i>Tabla 3. Vectores con las trayectorias de las acciones de control y las salidas que almacena el nodo local en el modo 4 de funcionamiento</i>	<i>92</i>
<i>Tabla 4. Actualización de los vectores con las trayectorias de las acciones de control y las salidas que almacena el nodo local cuando se pierde un dato</i>	<i>93</i>
<i>Tabla 5. Parámetros del control supervisor y del control local.....</i>	<i>130</i>
<i>Tabla 6. Índices de error IAE máximos obtenidos por porcentaje de pérdidas con canal de comunicaciones de uso exclusivo.....</i>	<i>131</i>
<i>Tabla 7. Índices de error IAE máximos obtenidos por porcentaje de pérdidas con canal de comunicaciones compartido</i>	<i>132</i>
<i>Tabla 8. Opciones disponibles del sistema distribuido de control</i>	<i>203</i>
<i>Tabla 9. Algoritmos de resolución de MPC lineales</i>	<i>268</i>

Parte I. Introducción

1 INTRODUCCIÓN

1.1 Objetivos

El objetivo de la presente tesis es definir una metodología para el diseño y la implementación de reguladores en un sistema empotrado en un entorno distribuido. Por lo tanto, se debe estudiar la problemática de los sistemas empotrados de control en un entorno distribuido, desde la comunicación de los distintos elementos de la red hasta las características que se deben implementar en el sistema empotrado para asegurar un funcionamiento adecuado en situaciones de pérdida de datos, retardos en la comunicación, conmutación de reguladores, computación excesiva de un nodo, delegación de código, etc.

Los controladores desarrollados deben permitir la adaptación de los mismos a las condiciones de trabajo, utilizando los recursos adecuados en cada momento, para no tener pérdidas significativas de prestaciones en el logro de los objetivos a conseguir.

Se considerará un diseño modular de los elementos que intervienen en el control, de forma que se maximicen las prestaciones, se asegure la estabilidad de funcionamiento y se adapte a la disponibilidad de recursos cambiante del entorno, proporcionando un funcionamiento autónomo seguro.

Los temas que se tratan a lo largo de la tesis se pueden resumir en los siguientes:

- Estudio de la problemática de la aplicación de reguladores en un entorno distribuido: estabilidad, elección de parámetros de control, cambio del control on-line, estudio de los transitorios, comunicaciones con otros módulos, etc.
- Aplicación a un sistema empotrado de control, en el que las restricciones temporales y de cómputo son críticas.
- Como consecuencia del comportamiento dinámico de la carga y las restricciones de utilización de recursos, deberá establecerse un sistema supervisor del propio sistema empotrado para que éste pueda adaptarse,

modificando las capacidades que ofrece atendiendo a los requisitos del sistema de control.

- Implementación eficiente del modelo, software y protocolo de comunicaciones, ya que la calidad de un sistema depende fuertemente del control y los algoritmos de decisión implementados en el sistema.
- Actuación ante pérdidas de datos para seguir manteniendo una salida adecuada en el sistema a controlar. Adaptación del control local a ésta nueva situación.

1.2 Motivación

Para ilustrar la problemática que se quiere abordar en la tesis, y a la que se pretende dar solución, se presenta un ejemplo de un sistema distribuido de control, en el que se producen problemas debidos a fallos en las comunicaciones entre los nodos.

Ejemplo Sea un sistema distribuido de control formado por una parte local, implementada por un sistema empotrado, con recursos limitados de computación y conectado físicamente al proceso a controlar, y una parte remota sin ninguna limitación computacional. Una forma de implementar un algoritmo de control de altas prestaciones en el sistema es que la parte remota se encargue de realizar los cálculos en cada iteración de la acción de control, y se la entregue a la parte local, que es la que en última instancia aplica la acción de control proporcionada por la parte remota sobre el proceso, por lo tanto el nodo local opera como un actuador con capacidad limitada de cómputo.

El esquema del sistema distribuido de control corresponde a un sistema de control basado en red sin control local, tal y como se define en (Casanova 2005). En la Figura 1 se representa este tipo de control basado en red.

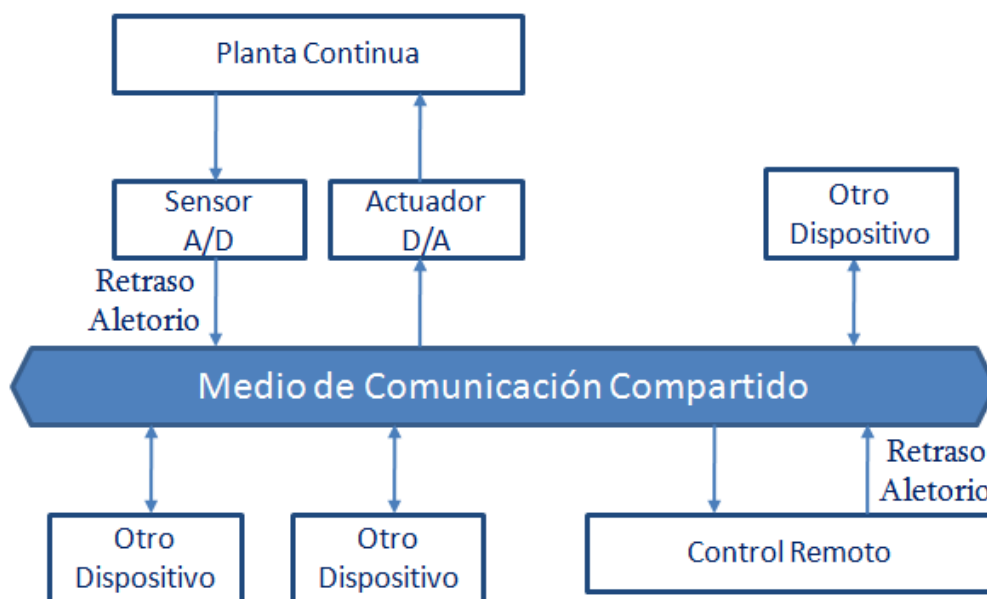


Figura 1. Sistema de control basado en red sin control local

Para simular el comportamiento del sistema distribuido se ha utilizado la herramienta de simulación Truetime, que proporciona una Toolbox basada en Matlab/Simulink y permite la simulación del canal de comunicaciones distribuido así como los tiempos de cómputo de las distintas tareas de control (Ohlin, Henriksson y Cervin 2007). En la Figura 2 se muestra el esquema Simulink utilizado mediante la herramienta Truetime.

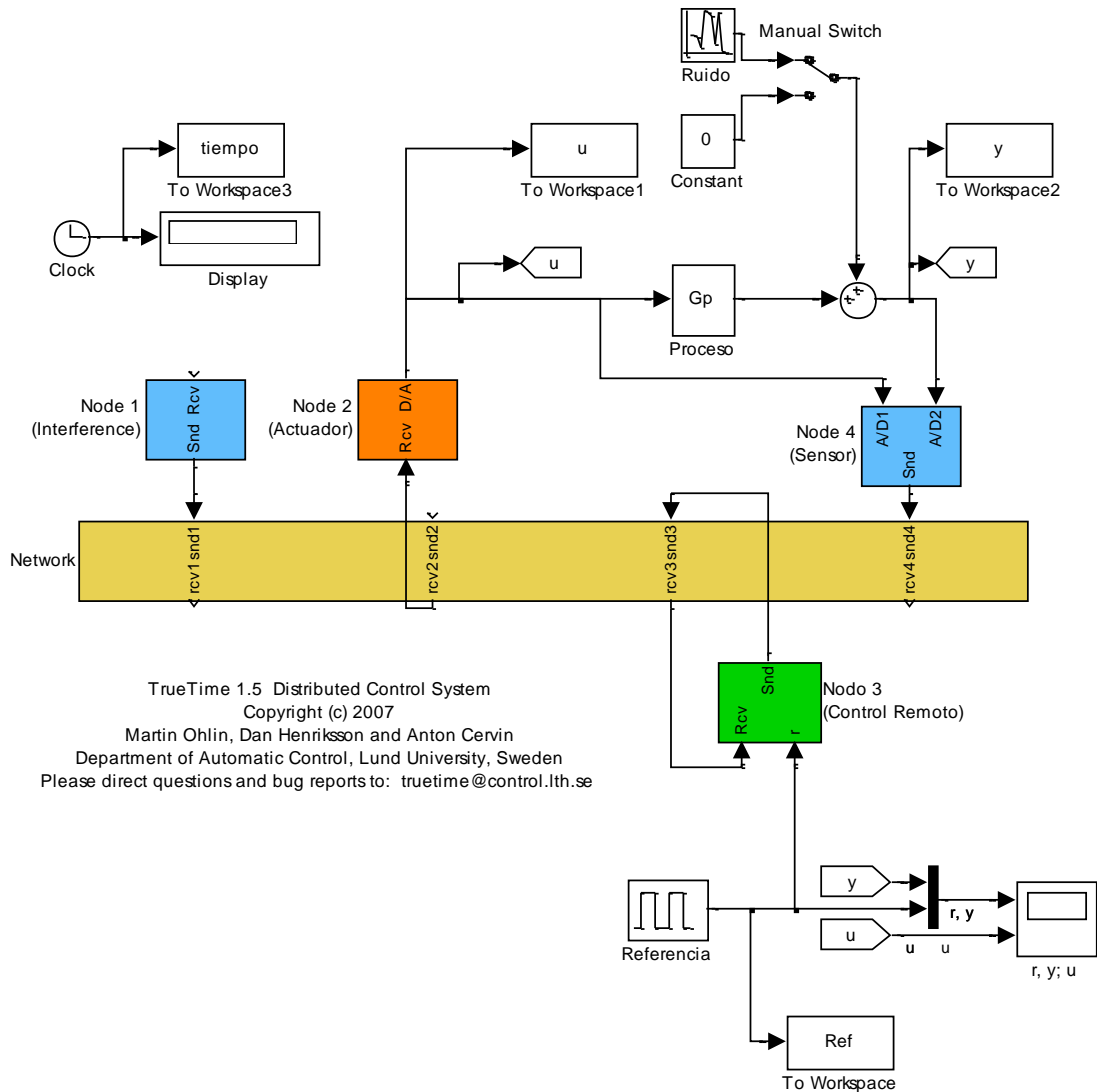


Figura 2. Esquema Truetime del sistema de control basado en red sin control local

Cuando no hay pérdida de datos el comportamiento que tiene la salida del proceso y la acción de control entregada por el nodo local se muestra en la Figura 3.

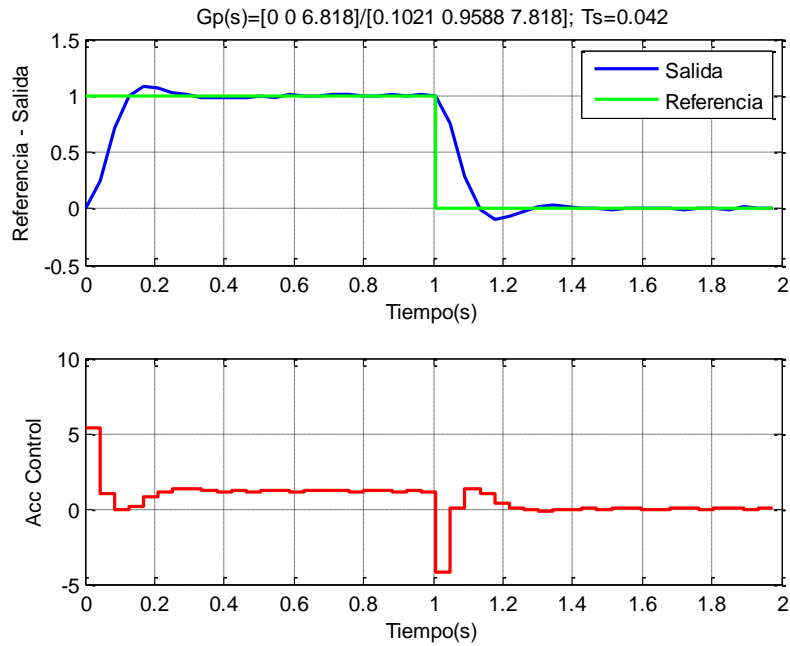


Figura 3. Comportamiento del sistema sin fallos en la comunicación

El problema de este esquema es que, al utilizar un canal de comunicaciones de uso no exclusivo, se pueden producir pérdidas de datos en el envío de las señales. En caso de pérdida de datos entre la parte remota y la parte local ésta podría mantener la última acción de control, a la espera de una acción de control más actualizada. A continuación se muestra la respuesta del proceso, junto con la acción de control aplicada por el controlador local, ante una pérdida aleatoria del 50% de los datos enviados:

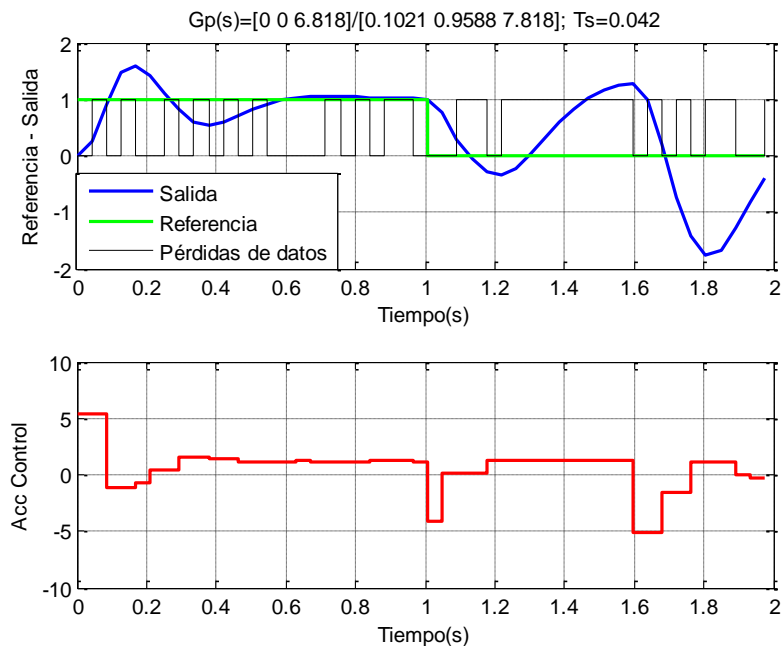


Figura 4. Comportamiento del sistema con fallos en la comunicación

En la figura anterior se puede ver cuál es el efecto de la pérdida de datos en un sistema distribuido cuando en la parte local, en caso de fallos en las comunicaciones, se mantiene la acción de control anterior. La señal binaria de color negro es la que informa con un valor de 1 cuando se producen esos fallos.

Como se muestra en la Figura 4, la salida del sistema tiene un comportamiento no deseado, consecuencia de la pérdida de datos en el transitorio, que en algunos casos puede hacer que se desestabilice el sistema.

■

La principal motivación de la tesis es la realización de una estrategia de control, en un entorno distribuido, que permita adaptar el comportamiento a las limitaciones existentes, para conseguir que en todo momento se le entregue al proceso una acción de control apropiada. Para ello se deberá diseñar la estructura, los modos de funcionamiento y los algoritmos de control a implementar en los elementos del sistema distribuido.

Algunas propuestas de este trabajo se han desarrollado dentro del proyecto DPI2005-09327-C02-01/02 titulado “Núcleo de Control en los Sistemas Empotrados Fuertemente Interconectados (KERTROL)”. El principal objetivo del proyecto es dotar a los sistemas empotrados de control de un alto nivel de inteligencia mediante la interacción con el entorno y la interconexión con el resto de sistemas que componen una solución basada en sistemas empotrados. Este aumento en el nivel de inteligencia deberá compatibilizarse con el cumplimiento de restricciones de tiempo real que requieran las tareas de control.

El proyecto consta de dos subproyectos relativos al área de control de procesos por una parte y a la de sistemas operativos de tiempo real y sensorización inteligente por otra.

Otro proyecto de investigación en el que se ha desarrollado parte del trabajo de la presente tesis es el DPI2008-06737-C02-01/DPI que lleva por título “Sistemas distribuidos con recursos limitados. Núcleo de Control y Coordinación (SIDIRELI)” El objetivo global y a largo plazo del proyecto es el análisis, diseño, desarrollo e implementación de sistemas ciber-físicos en los que las restricciones de recursos y los requisitos de control son fuertes. A corto plazo, se pretende desarrollar sistemas de control distribuidos, conectados en red y con fuertes restricciones de recursos.

En este proyecto aparecen campos de actividad separados y complementarios. Por una parte, las cuestiones relativas al control requieren una experiencia en teoría de control digital y aplicaciones en procesos industriales. Por otra parte, los aspectos de planificación y computación en tiempo real, así como la influencia de las comunicaciones entre subprocesos.

El trabajo desarrollado en la tesis contribuye al cumplimiento de los objetivos mencionados en ambos proyectos de investigación, aunque se centra principalmente en el desarrollo e implementación de soluciones de control en el entorno distribuido.

El autor de la tesis ha participado como investigador en ambos proyectos.

1.3 Contribuciones de la tesis

Las aportaciones de la tesis se resumen en:

- Generación de una estrategia de control utilizando el núcleo de control en un entorno distribuido, en el que se definen los nodos supervisores y locales. Se han especificado las características de cada nodo, los modos de funcionamiento y los datos que se intercambian.
- Especificación de una estrategia de control en la que se tienen en cuenta los problemas derivados del sistema de comunicaciones: falta de datos, desfase en la recepción de los mensajes, etc.
- Propuesta del método de compensación de la acción de control al utilizar las acciones de control postuladas por el nodo supervisor.
- Especificación de una metodología para la obtención de los parámetros de los controles a implementar, en el nodo supervisor y local, del sistema distribuido de control.
- Definición de una metodología para la evaluación de las prestaciones que se consiguen con cada modo de funcionamiento, en presencia de pérdida de datos y problemas en la comunicación de los nodos.
- Creación de una herramienta de asistencia al diseño y simulación del sistema distribuido de control, que permite evaluar las prestaciones conseguidas por cada modo de funcionamiento ante diversas situaciones: pérdida de datos, ruidos en la medida, ancho de banda dedicado al control en el sistema de comunicaciones, etc. Esta herramienta se ha utilizado para obtener los parámetros más adecuados de los controladores a implementar, en el nodo supervisor y local, para procesos con distinta dinámica, tanto de forma simulada como real.

1.4 Organización de la tesis

La presente tesis está organizada en 7 capítulos. El capítulo 1 expone los objetivos y las motivaciones para la realización de la tesis. En los capítulos 2 y 3 se expone un breve estado del arte sobre las dos áreas de investigación sobre las que trata este trabajo: los sistemas empotrados de control y los sistemas distribuidos de control con elementos basados en sistemas empotrados. El capítulo 4 presenta las aportaciones de la tesis, dando solución a la implementación de controladores en entornos empotrados y en sistemas distribuidos con elementos empotrados, basándose en el núcleo de control. El capítulo 5 se muestra, mediante pruebas simuladas, la implementación de los algoritmos de control presentados en el capítulo 4 sobre

diferentes tipos de procesos. El capítulo 6 presenta la aplicación de los controladores desarrollados sobre sistemas reales. Por último, en el capítulo 7 se exponen las conclusiones y líneas futuras de trabajo.

A continuación se detalla el contenido de cada uno de los capítulos de la tesis:

Capítulo 2. Sistemas empotrados de control. El capítulo presenta las características de los sistemas de control cuando se implementan con elementos empotrados, desde las características de los controles hasta los sensores utilizados. También se hace una breve descripción de las características y aplicaciones de los sistemas de inteligencia ambiental. Así mismo, se realiza una descripción de las redes de sensores y actuadores así como de los sistemas ciber-físicos, que son los que enmarcan el entorno de trabajo utilizado.

Capítulo 3. Sistema distribuido de control. Este capítulo amplía las características de la implementación de controladores en elementos empotrados cuando el sistema está distribuido. Se analizan las características de los sistemas distribuidos de control y de los controles implementados en elementos empotrados, prestando especial atención a los controles inteligentes distribuidos. También se hace un análisis de las comunicaciones en el sistema distribuido y de las características de los servicios middleware dedicados al control.

Capítulo 4. Núcleo de control. Se presenta una primera aproximación a una estructura que permita la implementación de controladores y que respete las características descritas en los capítulos previos, enmarcando la solución en los sistemas ciber-físicos. En el capítulo se describen las características que definen al núcleo de control, desde los elementos involucrados, la arquitectura utilizada y los servicios middleware implementados. Posteriormente, el capítulo propone una adaptación del núcleo de control en el caso de un sistema distribuido, desde los componentes del núcleo de control a la distribución de los servicios middleware que debe implementar cada nodo, centrándose especialmente en las características de los controles que se implementarán en cada nodo, los modos de funcionamiento y sobre todo a las soluciones para asegurar el cumplimiento de prestaciones mínimas en el caso de pérdida de datos y retardos en las señales. Se presenta como solución la compensación del error de salida.

Capítulo 5. Pruebas en simulación. En este capítulo se evalúan las estructuras de control planteadas en el capítulo anterior mediante herramientas de simulación. Para ello primero se propone un método para obtener analíticamente el valor del error que represente las prestaciones conseguidas con la estrategia de control elegida. Se muestran distintos ejemplos de sistemas que implementan los controladores, utilizando el núcleo de control, tanto en las situaciones de funcionamiento normal como en situaciones de mal funcionamiento. Además, se muestra la metodología para la elección de los parámetros de control en los distintos nodos del sistema distribuido.

Capítulo 6. Aplicaciones. En este capítulo se presentan la implementación de la solución propuesta para los sistemas de control sobre varios sistemas reales, estudiando las mejoras que se producen sobre la respuesta del sistema y la calidad del control proporcionado.

Capítulo 7. Conclusiones y trabajo futuro. En el capítulo que cierra la tesis se resumirán las conclusiones que se han obtenido en el desarrollo del trabajo, resaltando las diferentes aportaciones realizadas. El capítulo concluye con un apartado de cuestiones abiertas, donde se concretan una serie de líneas de trabajo que sugiere la línea de investigación de la tesis.

Anexo A. Herramienta de cálculo y simulación del sistema distribuido de control. En este anexo se presenta la herramienta, creada en esta tesis, para el diseño y cálculo del sistema distribuido de control, en el que se aplican los algoritmos presentados utilizando el núcleo de control. La herramienta permite tanto la simulación como la aplicación directa de la solución sobre el sistema real.

Anexo B. Pruebas realizadas con el simulador. En este anexo se presentan las pruebas de distintos procesos realizadas con el simulador, en el que se eligen los parámetros de cada controlador, que forma parte del sistema distribuido, y se ponen en marcha sobre diversas condiciones de entorno y modos de funcionamiento.

Anexo C. Utilización de controladores predictivos en el nodo supervisor. Los controles predictivos son los que se han utilizado en el trabajo para diseñar las trayectorias de la salida y de la acción de control. En este anexo se presentan sus principales características, así como el proceso de cálculo de las trayectorias futuras cuando no se utiliza la estrategia del horizonte deslizante, para el caso de los reguladores predictivos GPC y CRHPC.

Anexo D. Método de interpolación basado en spline cúbica natural. En este anexo se desarrolla el algoritmo de interpolación utilizado, cuando las frecuencias de actualización de los distintos nodos del sistema distribuido no son iguales.

Parte II. Estado del Arte

2 SISTEMAS EMPOTRADOS DE CONTROL

2.1 Motivación y objetivos del capítulo

El objetivo de este capítulo es estudiar las características de los sistemas de control cuando se emplean computadores digitales empotrados, ya que ésta será una de las características a desarrollar en el presente trabajo.

Así pues es necesario caracterizar los sistemas empotrados dedicados al control, poniendo las referencias de los trabajos más sobresalientes en este campo así como los grupos de investigación dedicados al estudio de los mismos.

Por otro lado, para controlar un proceso el sistema de control necesita interactuar con el mismo mediante la utilización de sensores y actuadores. En este capítulo se realizará un estudio de los sistemas de sensores y actuadores utilizados en los sistemas empotrados de control, así como de los sistemas ciber-físicos, en los que se va a ubicar el trabajo desarrollado en la tesis.

Por último se presenta el núcleo de control, desarrollado en el proyecto de investigación KERTROL, en el que se proporciona una solución al control de sistemas con elementos empotrados. Éste será el punto de partida para desarrollar los algoritmos de control en el sistema distribuido empotrado que será el objetivo de la tesis.

2.2 Sistemas empotrados de control. Introducción

Los sistemas empotrados son utilizados ampliamente en variedad de sectores industriales, en la electrónica de consumo, control de automóviles, equipamiento médico, etc. (Chou, Ortega y Borriello 1995), (ARTIST2 GCECS 2008).

La definición que se da en la *IEEE eLearning Library* (IEEE eLearning 2010) de un sistema empotrado es: “Combinación de hardware y software, y en algunos casos partes mecánicas, diseñado para realizar una función específica”. Por lo tanto podría

decirse que es cualquier dispositivo que incluye un computador programable pero en sí mismo no es un computador de propósito general. Trabaja en un ambiente reactivo y con restricciones temporales.

En el marco del presente trabajo se entiende por un sistema empotrado aquel que se diseña para tener un comportamiento como el de un componente, satisfaciendo unos requisitos de comunicación e interacción con otros elementos de un sistema global y que presenta un elevado nivel de autonomía, con gran fiabilidad. Usualmente está implícito el supuesto de miniaturización o, al menos, volumen reducido (Vallés 2004).

El control de un sistema empotrado está basado en un microprocesador (o microcontrolador) de propósito general, y dedicado a realizar una tarea o un conjunto de tareas específicas (Ayala, Lomeña y López-Vallejo 2002).

En la actualidad los sistemas empotrados ofrecen una gran variedad de servicios gracias a la integración electrónica, añadiendo más funcionalidades, y no se limita sólo a una tarea específica. La aparición de sistemas operativos empotrados como Linux (con varias distribuciones), Windows CE, VxWorks, Mimix, QNX, OS-9, etc., presenta la opción de desarrollar aplicaciones complejas para estos sistemas, como si se tratara de ordenadores personales, lo que significa un nuevo desafío en la programación (Marwedel 2006).

Los sistemas empotrados se encuentran en una amplia variedad de productos de mercado, como por ejemplo (ARTIST2 GCECS 2008):

- Equipos de redes como cortafuegos, routers, switches, etc.
- Electrónica de consumo como reproductores mp3, teléfonos móviles, PDAs, cámaras digitales, cámaras de video, etc.
- Electrodomésticos como microondas, lavavajillas, etc.

La alta difusión de los productos anteriores impone restricciones respecto de su precio final. Además, dada la naturaleza propia de los sistemas en los que se hace uso de ellos, se añaden limitaciones en el tamaño físico y el consumo de energía. Estas limitaciones del producto provocan un aumento en las restricciones de recursos a nivel de la plataforma de computación, por ejemplo, en la velocidad de cálculo (CPU), tamaño de memoria, ancho de banda de comunicación, así como en la disponibilidad de energía, ya que muchos de estos sistemas funcionan alimentados por batería. Esto es así a pesar de la rápida evolución del hardware, debido a las exigencias económicas que es necesario que cumplan los sistemas empotrados, no justificándose la utilización de procesadores de mayor capacidad y favoreciendo el uso de componentes de computación de propósito general sobre soluciones hardware diseñadas de forma específica.

El diseño de este tipo de sistemas ha estado marcado por la necesidad de reducir tamaño, coste y, en general, recursos aumentando la fiabilidad y sin reducir prestaciones. Aunque hay muchos trabajos relativos a sistemas empotrados, (Berger 2002), (Catsoulis 2005), particularmente en el campo de la computación, (Pont 2002), (Grehan, Cyliax y Moote 1998), (Li y Yao 2003), es preciso hacer una reformulación del propio sistema de control si el objetivo final es implementarlo en un sistema empotrado.

2.3 Características de los sistemas empotrados de control

En (Balarin, y otros 1997) se presentan las características más sobresalientes que se exigen en el diseño de un sistema empotrado, y que son una tónica general en el diseño de altas prestaciones de estos dispositivos. Estas características son:

- Concurrencia en la ejecución de componentes
- Fiabilidad y seguridad en el manejo de errores
- Reactividad y tiempo real
- Interacción con dispositivos físicos
- Robustez frente a las condiciones del entorno
- Bajo consumo
- Precio reducido
- Pequeñas dimensiones.

Estas características siguen vigentes en la actualidad, como se pone de manifiesto en los numerosos trabajos publicados en los últimos años relativos al diseño de sistemas empotrados de control (Martí, y otros 2010), herramientas para la implementación de los controladores (Törngren, y otros 2006), sobre las características del software en los sistemas empotrados (Liggesmeyer y Trapp 2009), planificación en los sistemas empotrados de control (Crespo, y otros 2006), controladores empotrados de bajo coste (Peng, Ma y Xia 2008) y sistemas empotrados en entornos distribuidos (Verhoef 2008).

Las características mencionadas anteriormente están relacionadas entre ellas, peor aún, compiten una con otra, de modo que mejorar una implica la degradación de otras. Como ejemplo claro tenemos que, cuando disminuimos el tamaño del sistema empotrado, se puede estar mejorando en tamaño y peso, pero afectando de manera negativa la funcionalidad al no contar con algún soporte hardware de control de errores. A fin de optimizar estas métricas, el diseñador debe estar en conocimiento de una variedad de tecnologías de implementación hardware y software, a fin de encontrar la mejor implementación para una aplicación y restricciones dadas. De este modo, el diseñador debe ser un experto tanto en el diseño software como en el diseño hardware, donde la validación de las metodologías mediante la selección apropiada de diversos casos de estudio se hace indispensable.

Las características anteriores condicionan ampliamente los sistemas empotrados para su utilización en aplicaciones de control, donde muchos bucles de control están compitiendo por los recursos y surgen una serie de problemas relacionados con la temporización y la planificación de tareas (P. Albertos, y otros 2005). La estructura de un sistema de control empotrado se muestra en Figura 5, donde se muestran los diferentes sensores, actuadores y su interacción con el entorno.

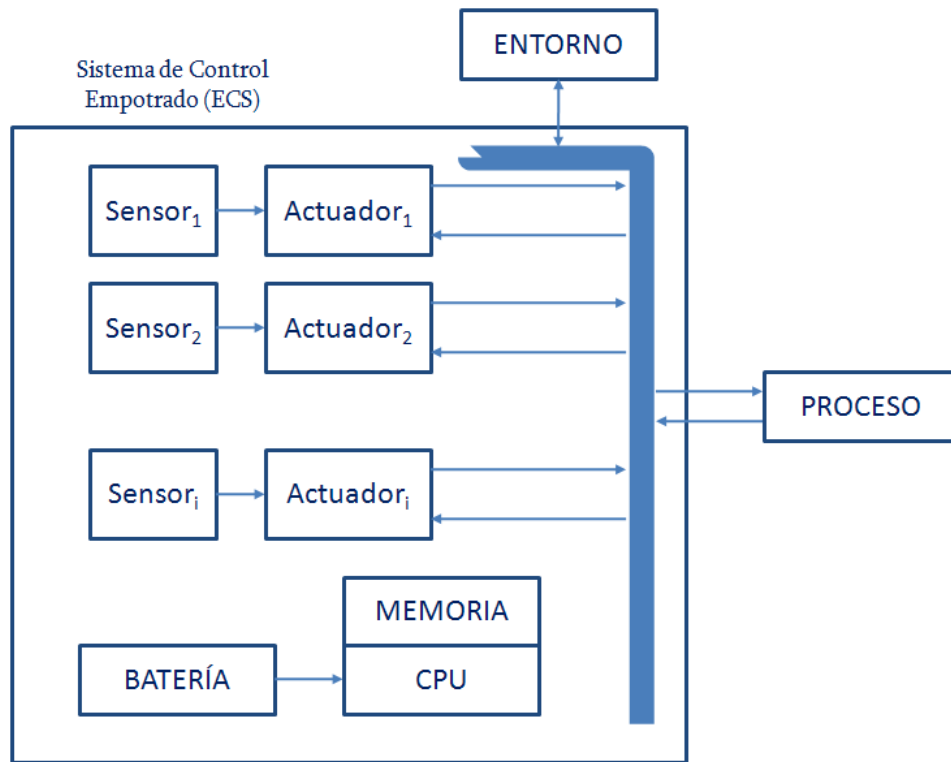


Figura 5. Estructura de un sistema empotrado de control. Fuente: (P. Albertos, y otros 2005)

En (Bouyssounouse y Sifakis 2005) se exponen los temas más relevantes en el estado y las tendencias de los sistemas empotrados. En (ARTIST 2008) se describen la situación y el futuro sobre una serie de áreas. Las características básicas de los sistemas empotrados de control, que se exponen en las referencias anteriores, pueden resumirse en:

- Autonomía, para trabajar independientemente de la unidad de interfaz con el usuario.
- Tolerancia a fallos, para ser capaz de trabajar en un modo degradado o con fallo.
- Operación con datos escasos, para hacer frente a situaciones de pérdida de información.
- Reconfigurable, para adaptarse a nuevas condiciones de operación. Ello requiere distintas opciones de control y mecanismos de transferencia entre modos de operación.

- Algoritmos de control híbrido, combinando la toma de decisiones y la elaboración de señales de control.
- Adecuación para la ejecución en entornos multitarea, compartiendo recursos con otras aplicaciones.
- Restricciones de tiempo real.
- Retardos variables de computación y de intercambio de datos.
- Optimización de recursos (CPU, batería, memoria, ...).
- Seguridad, en entornos con interacción con humanos, así como seguridad del funcionamiento para evitar fallos. Ello requiere la certificación de las soluciones implementadas.

Estas características requieren la consideración conjunta de las cuestiones de diseño de control y de su implementación en entornos con recursos limitados. Hay un gran número de grupos de investigación que trabajan en estas líneas, tales como:

- Department of Automatic Control. Lund Institute of Technology. Sweden
- Embedded Control Systems Research Group. School of Industrial Engineering and Management. Department of Machine Design. KTH Royal Institute of Technology. Stockholm, Sweden
- RETIS Laboratory on real-time systems. Scuola Superiore Sant'Anna of Pisa, Italy
- Interdisciplinary Studies in Intelligent Systems. Department of Electrical Engineering University of Notre Dame. Indiana, USA
- Center for Control, Dynamical-systems, and Computation. Electrical & Computer Engineering Dept. University of California, Santa Barbara. USA

En este sentido, hay que tener en cuenta la relevancia de cada señal de control y, consecuentemente, de cada tarea de control (ello determina los retardos máximos aceptables), modificar adecuadamente las prioridades de cada actividad para distribuir el tiempo de cálculo, reducir los posibles retardos en la entrega de la señal de control o, en última instancia, equilibrar estos retrasos para conseguir la menos degradación de prestaciones de todo el sistema de control obligado a operar con la limitación de recursos que la autonomía impone.

En (Sanz y Arzen 2003) se hace una propuesta de cuáles deberían ser las características del software de control para hacer frente al incremento en la complejidad requerido por los sistemas empotrados de control para aplicaciones críticas.

Estas **necesidades del software de control** se resumen en:

- **Tiempo crítico:** Se requieren altas prestaciones y/o estar sometido a restricciones de tiempo real.
- **Empotrado:** Deben ejecutarse en plataformas con recursos limitados e interactuar con el entorno.

- **Tolerante a fallos:** Deben mantener (algún nivel de) prestaciones bajo condiciones de fallo.
- **Distribuido:** Los componentes software a menudo están distribuidos en varios computadores interconectados.
- **Inteligente:** Se puede requerir algún nivel de inteligencia y comportamiento autónomo.
- **Amplio:** Puede contener millones de líneas de código.
- **Integrado:** Una aplicación requiere la integración de varios subsistemas en una sencilla, unidad cohesiva.
- **Abierto:** Debería ser software no propietario y capaz de soportar aplicaciones de terceros.
- **Heterogéneos:** Debería ser capaz de ejecutar en una variedad y mezcla de plataformas de cómputo.

El software es crítico en los sistemas de control. Cada vez se necesitan unos requisitos más difíciles: tamaño más pequeño, bajo coste, menos tiempo, más funcionalidades, más desarrollos, y más dependencias. Para alcanzar estos requisitos se necesita de las siguientes áreas: tecnología de agentes, diseño basado en arquitectura, inteligencia artificial, ingeniería concurrente, composición, diseño de modelos, sistemas empotrados distribuidos, ingeniería de dominios, sistemas empotrados, marcos, integración, ciclos de vida, ingeniería del software basado en modelos, sistemas modulares, programación orientada a objetos, ontología, ingeniería de línea de productos, sistemas distribuidos de tiempo real, reutilización, componentes software, y procesos software.

En (P. Albertos, y otros 2005) se exponen las características que se deben cumplir en un sistema empotrado de control desde el punto de vista de la implementación, computacional, del algoritmo de control y del sistema operativo empotrado.

2.4 Sistemas ciber-físicos

En (Rajkumar, y otros 2010) se define a un sistema ciber-físico como un sistema que integra computación, capacidad de almacenamiento, seguimiento y control de entidades en el mundo físico, que operen de forma independiente, segura, fiable, eficiente, flexible y robusta. Es decir, se trata de sistemas de control con restricciones de tiempo real y espaciales actuando sobre un entorno físico, debiendo integrar los conceptos generales de control, computación y comunicación, con las características dinámicas de los sistemas físicos y de ingeniería.

Un sistema ciber-físico es un conjunto de sistemas de cómputo y comunicación que interactúan con el mundo físico en tiempo real. Este mundo físico, distribuido por naturaleza, es cambiante e impone restricciones de acceso y manipulación.

En (Stankovic, y otros 2005) se definen las características que se deben cumplir en los sistemas físicos de computación, que son el antecedente de los sistemas ciber-físicos.

En la tesis (S. Graham 2004) se realiza un trabajo sobre la ubicación virtual de componentes software, que trata también la robustez ante retardos de red, transitorios y fallos. Igualmente, introduce el concepto de autonomía local temporal, o la posibilidad de que un componente pueda sustituir temporalmente a otro que haya fallado.

La fiabilidad de operación es una característica crucial tanto para grandes sistemas distribuidos (con repercusión de fallos entre componentes) como para sistemas que interaccionan con el mundo físico. La fiabilidad es no sólo a nivel de cada componente sino a nivel de los inevitables posibles errores inducidos por fallos de otros componentes. Entre las propiedades a considerar en los sistemas ciber-físicos, y que forman parte de las conclusiones de los *workshops* (NSF 2006) y (ARTIST WCES 2007), están:

- **Autonomía local temporal:** Un componente del sistema, sensor, unidad de proceso o accionador, debe ser capaz de operar correctamente en ausencia de comunicación, al menos durante un tiempo preestablecido. Si el componente está directamente conectado a un sistema físico, debe poder proporcionar un funcionamiento seguro en presencia de cualquier fallo, propio o inducido. Por ejemplo, un controlador, en vez de proporcionar simplemente la acción de control actual debe ser capaz de proporcionar una secuencia de acciones de control, pre-computadas o calculadas localmente, en espera de una recuperación del fallo. Los sistemas ciber-físicos deben permitir una suave degradación de prestaciones, lo que aconseja un desarrollo incremental, permitiendo su evolución a sistemas cada vez más complejos.
- **Estabilidad:** No sólo referida a la estabilidad del sistema de control, propiedad básica del mismo, sino también ante cambios en el sistema informático o de comunicaciones (reiniciaciones, p.ej.), o la reutilización de componentes no específicamente diseñados para la aplicación.
- **Confiabilidad:** Funcionamiento independiente entre componentes. El fallo de un componente no debe abortar otro, además debe haber una coherencia temporal entre los mismos y no existir un punto muerto por interferencias. El sistema debe ser **recuperable**, por ejemplo, mediante una transferencia de tareas entre componentes ante fallos o funcionamientos incorrectos.
- **Conocimiento explícito de tiempos y retardos:** Si se conocen se pueden incluir en el diseño o en el cómputo. Los distintos relojes no tienen por qué estar sincronizados, pero se debe conocer el tiempo transcurrido desde que los datos recibidos fueron creados.

- **Situación virtual:** Los componentes se diseñan ubicados en una situación hipotética virtual, y las condiciones concretas en cada momento determinan las restricciones de tiempo y recursos a utilizar en el cómputo. Ello conduce a un direccionamiento semántico: cada “cliente” pide un servicio que le proporciona el servidor adecuado. Solamente entidades especialmente diseñadas para este propósito se encargan de la distribución de roles y tareas.
- **Migración para auto-optimización y fiabilidad:** Sensores y accionadores específicos requieren posiciones fijas, pero el resto de componentes admiten una ubicación virtual, pudiendo existir y ejecutarse en cualquier enclave posible. La situación óptima se determinará en función de factores tales como restricciones temporales, carga computacional o retardos de comunicación esperados. El sistema debe dar el soporte adecuado para permitir la comunicación desde el nuevo emplazamiento, calcular los recursos disponibles antes y después de la migración, retardos y pérdidas de información en la nueva configuración. Para ello, se debe desarrollar el middleware apropiado.
- **Sistemas espacio-temporales:** Capacidad de razonamiento sobre el tiempo y el espacio.

2.5 Sistemas empotrados y sensores/actuadores inteligentes

La sensorización es indispensable en los sistemas de control. Cada proceso a controlar requiere determinados tipos de sensores, de modo que se obtenga una percepción lo más precisa posible de las variables observables del sistema a controlar. Los sensores deben ofrecer información precisa y sin retraso excesivo al sistema de control, de modo que el bucle de control del que forman parte permita la respuesta en tiempo real del sistema a controlar.

En los sistemas de control complejos tales como robots autónomos, robots bípedos, etc., el sistema sensorial es disperso, de modo que los sensores se encuentran físicamente separados, e interconectados por algún tipo de conexión digital (cableada ó inalámbrica) al módulo o módulos de control. Conforme aumenta la complejidad del sistema a controlar, aumenta el número de sensores y actuadores, así como el flujo de información en el bus de interconexión.

Por otro lado, las tendencias actuales hacia la miniaturización de los sensores (MEMS¹) y la disponibilidad de pequeños procesadores específicos integrados que incluyen la circuitería de comunicaciones, permiten la obtención de información

¹ MEMS: del inglés, *MicroElectromechanical Systems*

precisa y altamente fiable en sensores de muy reducido tamaño y consumo (Boucher y Lensch 2010).

A continuación se presenta la tendencia de los sistemas de sensores y actuadores en los sistemas distribuidos. En el apartado 2.5.1 se presentan las redes inalámbricas de sensores y actuadores, y en 2.5.2 los sistemas de inteligencia ambiental. En ambos casos el objetivo que se persigue, en un sistema distribuido de control, es el de obtener la información del sistema y actuar sobre el mismo mediante dispositivos empotrados y de pequeño tamaño.

2.5.1 Redes inalámbricas de sensores (WSN) y Redes inalámbricas de sensores y actuadores (WSAN)

Las Redes Inalámbricas de Sensores (WSN²) (Akyildiz, Su, y otros 2002) se han convertido en un campo de estudio que se encuentra en continuo crecimiento. Consisten en un conjunto de nodos de pequeño tamaño, también llamados motas, de muy bajo consumo, que se comunican entre sí inalámbricamente formando una red y que a su vez se conectan con un sistema central siguiendo diferentes tipologías. Este sistema es el encargado de recopilar la información recogida por cada uno de los sensores. El ámbito de aplicación de este tipo de sistemas es muy amplio: monitorización de entornos naturales (Mainwaring, y otros 2002), aplicaciones para la defensa (Marcy, y otros 1999), aplicaciones médicas (Schwierbert, Gupta y Weinmann 2001), etc.

En las Redes Inalámbricas de Sensores y Actuadores (WSAN³) (Akyildiz y Kasimoglu 2004) además de nodos sensores existen nodos actuadores. Los sensores van reuniendo información sobre el medio físico, mientras que los actuadores toman decisiones y ejecutan las acciones apropiadas sobre el entorno. Este tipo de redes tienen algunas características que no son propias de las WSN (Buendía, y otros 2007):

- Mientras que los nodos sensores son dispositivos pequeños, baratos, con capacidad de comunicación y procesamiento limitados, los nodos actuadores consumen más energía, son más caros, tienen mayor capacidad de procesamiento y comunicación.
- La capacidad de reacción de los actuadores ante un evento en tiempo real es una cuestión importante. Es necesario introducir mecanismos de coordinación entre los sensores y los actuadores.
- El número de nodos sensores suele ser muy superior al número de actuadores.

² WSN: del inglés, *Wireless Sensor Networks*

³ WSAN: del inglés, *Wireless Sensor and Actor Networks*

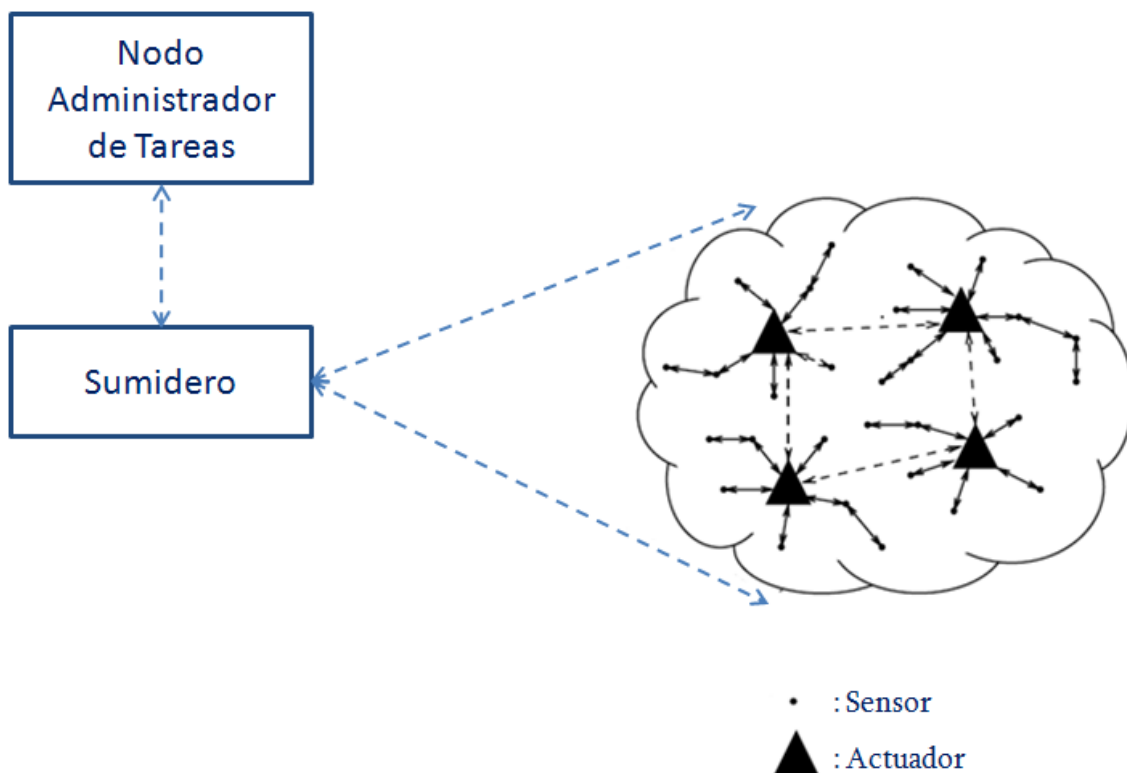


Figura 6. Arquitectura física de las WSN (Akyildiz y Kasimoglu, *Wireless Sensor and Actor Networks: Research Challenges 2004*)

2.5.2 Sistemas de inteligencia ambiental

La inteligencia ambiental es un concepto planteado por la *Information Society Technologies Advisory Group* (ISTAG) (Ducatel, y otros 2001) como un entorno en el que todo el mundo tendrá a su alrededor sistemas informatizados, empotrados en los objetos cotidianos. En lugar de ser las personas quienes interactúen con los objetos, serán estos quienes de una forma intuitiva e inteligente crearán un ambiente capaz de reconocer y responder a las necesidades de los usuarios (O'Hare, y otros 2004).

La inteligencia ambiental ubica al usuario como objetivo de todo el desarrollo tecnológico. Esto implica un cambio en la visión que se tiene de los sistemas, en el que la tecnología debe ser desarrollada para las personas, en lugar de que las personas se adapten a la tecnología (Friedewald y Da Costa 2003).

Los sistemas de inteligencia ambiental usan tecnologías desarrolladas en otras áreas de entre las que destaca la de las redes inalámbricas sensoriales y de actuadores (WSAN). Estas tecnologías se agrupan en tres campos sobre los que se sustenta los fundamentos de la inteligencia ambiental: la computación ubicua, la comunicación ubicua y las interfaces inteligentes.

El término "computación ubicua" fue acuñado desde el Centro de Investigación de Xerox en Palo Alto en 1991 por Mark Weiser (Weiser 1991). Partiendo de los

computadores como dispositivos complejos, se debe ocultar dicha complejidad a los usuarios. Esto se traduce en la integración de microprocesadores en todo tipo de objetos de forma que se puedan actuar sobre ellos o se pueda realizar un tratamiento informático de sus características.

La comunicación ubicua permite a los objetos comunicarse entre ellos y los usuarios por medio de redes (fundamentalmente redes inalámbricas) para poder interactuar entre ellos. Existe una gran variedad de redes y protocolos para el control y las comunicaciones de los sensores y actuadores en un entorno ubicuo. Para redes de propósito general destacan EIBus (1998) y LONWork (1999), orientadas a la automatización de viviendas, X10 (1984), CEBus (1993) y SCP de Microsoft, para el manejo de información multimedia HAVi (1998), HomePNA (1998) y especificaciones para soportar redes inalámbricas como CSMA/CA bajo IEEE802.11, SWAP y la popular BlueTooth (2001).

Finalmente, los interfaces de usuario inteligentes permiten el control y la interacción con el sistema inteligente formado por los anteriores componentes. Este control no sólo se realiza por medio de los dispositivos habituales sino que también incluye interfaces naturales (voz, gestos) como personalización del sistema (preferencias, contexto) con el principal objetivo de la adaptación. Un sistema adaptativo (Benyon 1993) es “aquel que, basado en el conocimiento, altera automáticamente aspectos de funcionalidad e interacción para lograr acomodar las distintas preferencias y requerimientos de sus distintos usuarios.”

A partir de estos ingredientes, la inteligencia ambiental supone un paso más en la evolución de la tecnología hacia todos los aspectos de la vida diaria. Desde la adaptación del hogar al estado de ánimo de los habitantes, a los sistemas inteligentes de control evolutivos que aprenden de la interacción con el medio para mejorar su rendimiento a las circunstancias en las que trabajen en cada momento. Esto hace que se apliquen en domótica, en automatización industrial, robótica autónoma, etc.

2.6 Núcleo de control

El núcleo de control fue definido en el proyecto KERTROL como el código mínimo a implementar en la tarea de control de un sistema empotrado para garantizar el funcionamiento seguro del proceso que se está regulando, cumpliendo las especificaciones deseadas.

El núcleo de control se ocupa de las actividades esenciales que garanticen un comportamiento seguro al sistema. En (Crespo, y otros 2006) se indican las funcionalidades que se incluyen en el núcleo de control:

- **Asegurar la entrega de las acciones de control (CA):** la acción de control debe ser entregada a tiempo aunque no se haya podido calcular la acción para el periodo actual. En ese caso, una acción de control debe entregarse una acción

de control precalculada (acción de control de reserva) o bien una acción segura.

- **Adquisición de datos de las señales importantes:** Los datos se adquieren basándose en un periodo adecuado, el retraso en la adquisición de los datos no puede producir un retraso en el sistema. Desde este punto de vista, se considera mejor la utilización de datos pasados y no calcular la acción de control (el cómputo de la acción de control, en este caso, sigue siendo válido).
- **Transferencia de una nueva estructura de control:** El sistema de control proporciona varios reguladores que se pueden utilizar en diferentes situaciones.
- **Servicios de comunicación:** El núcleo de control debe soportar las comunicaciones con dispositivos externos.

El núcleo de control dota a los sistemas empotrados de control de un alto nivel de inteligencia mediante la interacción con el entorno y la interconexión con el resto de sistemas que componen una solución basada en sistemas empotrados. Este aumento en el nivel de inteligencia deberá compatibilizarse con el cumplimiento de las restricciones de tiempo real que requieran las tareas de control.

En el apartado 4.2 se exponen las principales características del núcleo de control así como los elementos que lo componen.

2.7 Resumen y conclusiones

En este capítulo se han expuesto los temas más relevantes de los sistemas empotrados de control relacionados con los contenidos que aborda la tesis. Así, se ha empezado revisando las características de los sistemas empotrados de control, para a continuación definir y exponer las características de los sistemas ciber-físicos. Finalmente se han presentado los sensores y actuadores inteligentes.

A continuación se pasa a detallar las conclusiones más importantes relacionadas con cada uno de los aspectos tratados.

Sobre los sistemas empotrados

En la actualidad el número de computadores empotrados sobrepasa considerablemente el número de computadores personales. Ello es en gran parte debido a la incorporación de los sistemas de control digital como componentes en muchas aplicaciones.

En la etapa de diseño del controlador habría que realizar ciertas consideraciones si se desea implementar en un sistema empotrado, dado que la actividad de control comprende diversas tareas, de distinta complejidad. Desde las habituales en todo

sistema de control, como son la toma de datos, el cálculo y la entrega de las acciones de control, hasta la toma de decisiones, actualización de las variables y parámetros o la repercusión de información almacenada previamente. Si estas tareas han de ejecutarse concurrentemente en un solo computador, surge el problema de la planificación y cumplimiento de posibles restricciones temporales debidas a los retardos que la interacción de las tareas puede provocar.

Por otro lado, el hecho de trabajar con plataformas empotradas con el objeto de implementar el sistema de control provoca que se realice el esfuerzo por conseguir que el uso de los recursos de computación sea lo mejor posible, siendo necesario el uso de algoritmos de planificación que lo permitan.

Para desarrollar sistemas empotrados inteligentes, interoperables y conectados es necesario el desarrollo de capacidades y soluciones industriales en dominios críticos como el diseño de sistemas, arquitecturas distribuidas, plataformas de cómputo, seguridad, middleware y herramientas. Para ello se han revisado las características que deben cumplir el software de control, la implementación, las características computacionales, el algoritmo de control y el sistema operativo empotrado, citando los trabajos más importantes en cada área.

Sobre los sistemas ciber-físicos

En el apartado 2.4 se han definido los sistemas ciber-físicos, se han expuesto sus principales características y propiedades. Estas propiedades son muy importantes en el desarrollo de la tesis, ya que el objetivo de la misma es el diseño de una arquitectura de control que permita cumplir con estas propiedades en un sistema de control distribuido con elementos empotrados.

Sobre los sistemas empotrados y sensores inteligentes

La sensorización y la actuación son indispensables en los sistemas de control. En el apartado 2.5 se han revisado las características de las redes de sensores y actuadores. Se han expuesto sus principales características y se han definido los conceptos de inteligencia ambiental y comunicación ubicua. Estos conceptos están muy relacionados con los sistemas ciber-físicos.

3 SISTEMA DISTRIBUIDO DE CONTROL

3.1 Motivación y objetivos del capítulo

El objetivo de este capítulo es estudiar la problemática asociada a los sistemas distribuidos de control, en especial aquellos que integran elementos implementados sobre sistemas de escasos recursos de computación como sistemas empotrados.

En el apartado 3.2 se presentan las características de los sistemas distribuidos de control y se revisan las principales arquitecturas utilizadas, para pasar a describir en 3.3 las actividades a tener en cuenta en un sistema de control empotrado distribuido.

En el apartado 3.4 se definen las características de los sistemas de control inteligente, estos sistemas permiten representar y mantener la información en el sistema distribuido.

En el apartado 3.5 se describen los temas relativos a comunicaciones en los sistemas distribuidos de control, y se presentan los problemas asociados. Estos problemas son los que la tesis tratará de solucionar mediante la implementación de un algoritmo de control apropiado.

Finalmente, en el apartado 3.6 se describe la capa denominada middleware como una solución al funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas, exponiendo sus principales características y propiedades. Así mismo, en este apartado se realiza una revisión de los principales middlewares dedicados al control.

3.2 Sistemas distribuidos de control. Introducción

La heterogeneidad de los sistemas actuales, formados por múltiples componentes de diferentes características conectados en red, sugiere el desarrollo de sistemas de control distribuido en los que las distintas funciones de control se implementen a diferentes niveles. Este tipo de sistemas se denominan sistemas distribuidos de

control (DCS⁴) y están formados por varios elementos de procesamiento y por un sistema de comunicaciones. De esta forma, las tareas de control se pueden ejecutar en distintos sistemas de computación, pero comunicados. Se deberá tener en cuenta como se distribuyen las tareas entre los elementos de procesado y analizar los posibles retardos que introduciría el sistema de comunicaciones (Vallés 2004).

Cuando en esta configuración se incluyen elementos dotados de movilidad (robots, vehículos,...), la capacidad de obtener y procesar información, así como la de generar acciones de control, es variable con el tiempo. Es más, los recursos disponibles en cada momento por cada elemento para realizar las actividades que se le asignen, también variables, son limitados. Por ello, es preciso definir una estructura de control jerarquizado de forma que se asegure que las acciones de máxima prioridad se ejecuten a nivel local y que el resto (opcionales, refinamientos, de coordinación) se lleven a cabo en niveles superiores, con menor prioridad.

Habría que considerar, por tanto, al sistema distribuido de control como un sistema con restricciones de tiempo real. Un sistema distribuido de tiempo real es un sistema integrado compuesto de un grupo de hardware dedicado interconectado a través de un canal de comunicaciones en el que los elementos intercambian datos y mensajes. Generalmente, este hardware está compuesto de computadores empotrados que pueden funcionar en un sistema de control en bucle cerrado: muestrean la información de los sensores, calculan las respuestas apropiadas y las envían a los actuadores. Por lo tanto, para conseguir un funcionamiento correcto de los bucles de control y una apropiada integración de la información, espacial y temporal, es necesario un funcionamiento en tiempo real. Y por lo tanto, los recursos de computación de cada nodo empotrado así como las características temporales del BUS de campo para comunicaciones deben considerarse en el diseño (Heath 2003).

La realización de un sistema distribuido ha sido utilizada tradicionalmente para la realización del control de plantas en el que se abarcan espacios geográficos amplios y en el que se controlan gran número y variedad de dispositivos.

La evolución que se ha seguido en el diseño de controladores hasta llegar al control distribuido se muestra en la Figura 7.

⁴ DCS: *Distributed Control Systems*

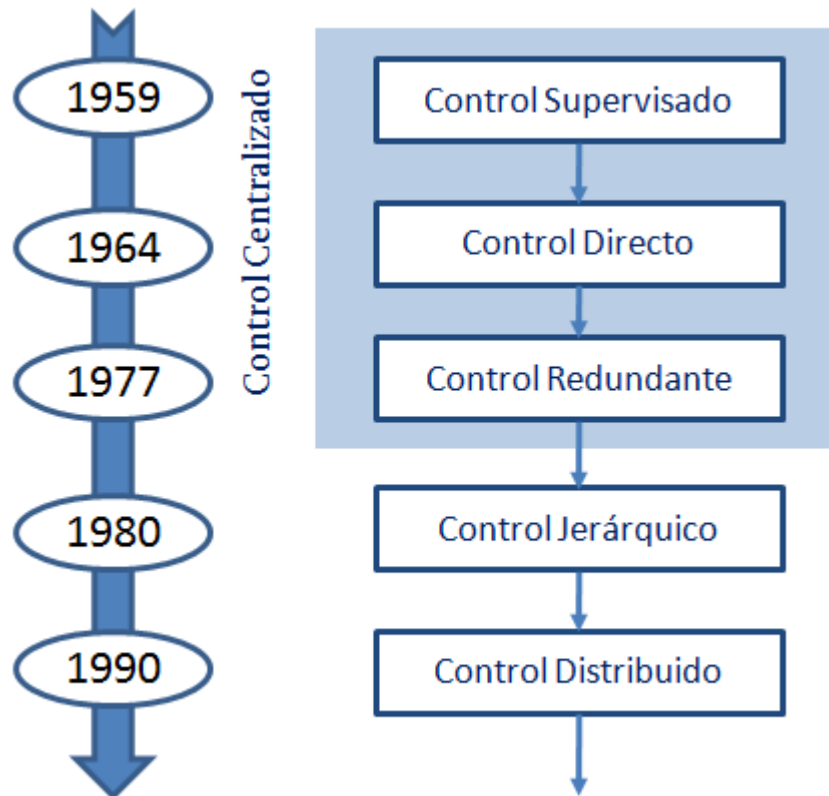


Figura 7. Evolución histórica de los sistemas de control distribuidos

Las principales características de cada uno de estos controles son las siguientes:

Control supervisado. Un único microprocesador controla los parámetros de todos los dispositivos, con posible intervención humana.

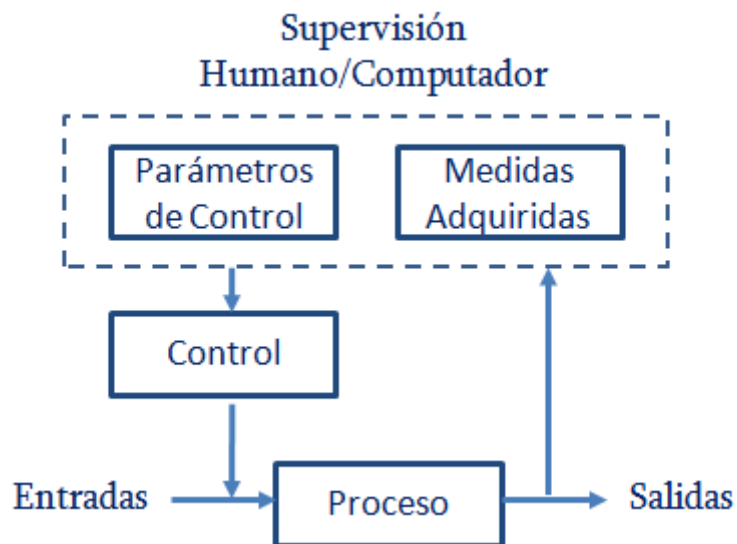


Figura 8. Control supervisado

Ventajas:

- Reducción de costes
- Reducción de dispositivos para la visualización

Inconvenientes:

- Diferentes requisitos temporales lleva asociado una mayor complejidad
- Sensibilidad a fallos

Control directo. Un microprocesador implementa directamente la ley de control. Un conmutador activa un controlador por defecto en caso de fallo. El control por defecto puede ser un control humano o automático.

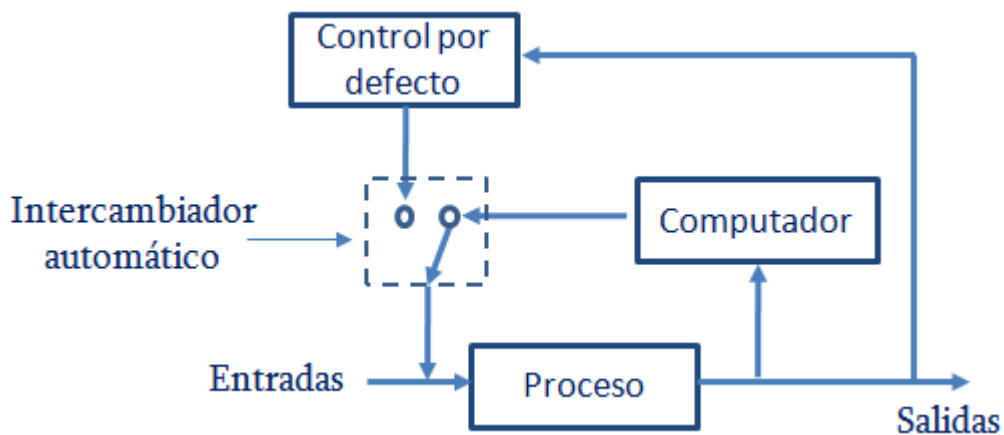


Figura 9. Control directo

Ventajas:

- Implementación de control muy sofisticado (control adaptativo, óptimo, multivariable, etc.)

Inconvenientes:

- Necesidad de un conmutador (intercambiador) para fallos
- Limitado a control de pocos procesos

Control redundante (dual). Dos microprocesadores implementan la ley de control. Un conmutador activa uno u otro en caso de fallo

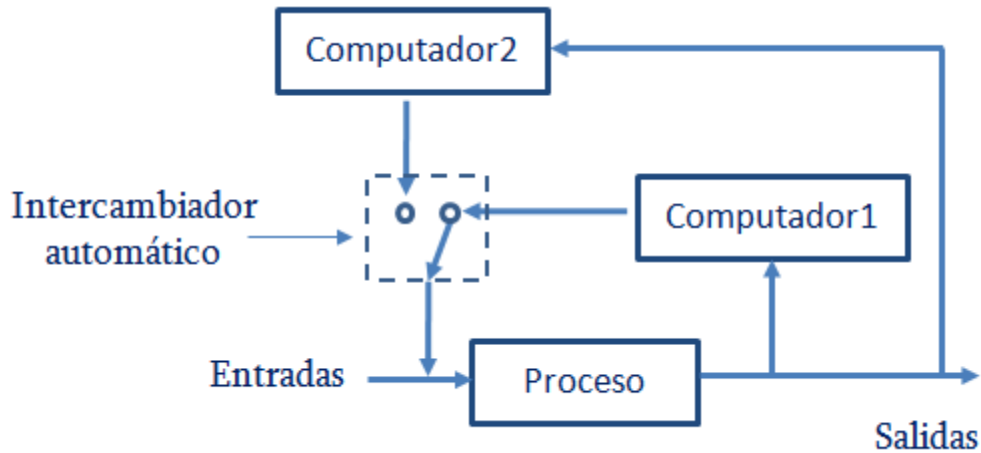


Figura 10. Control redundante (dual)

Ventajas:

- Mayor robustez frente a fallos

Inconvenientes:

- Cableado duplicado
- Software duplicado
- El conmutador (intercambiador) es crítico
- Incremento de la complejidad del diseño

Control jerárquico. Varios microprocesadores se distribuyen las tareas de control en diversas capas organizadas según los requisitos temporales de los procesos

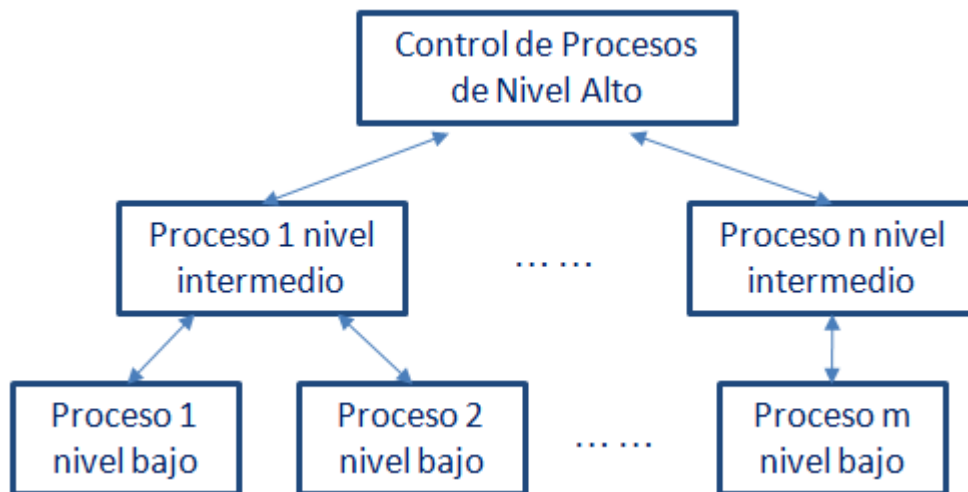


Figura 11. Control jerárquico

Ventajas:

- El diseño se facilita al dividir distintas tareas por sus requisitos temporales

Inconvenientes:

- Rigidez en la asignación de tareas a los microprocesadores
- La mayor diversidad en los dispositivos de control lleva asociado un mayor coste

Control distribuido. Microprocesadores similares entre los que se dividen las tareas de control. Pueden tener características jerárquicas (modelos híbridos).

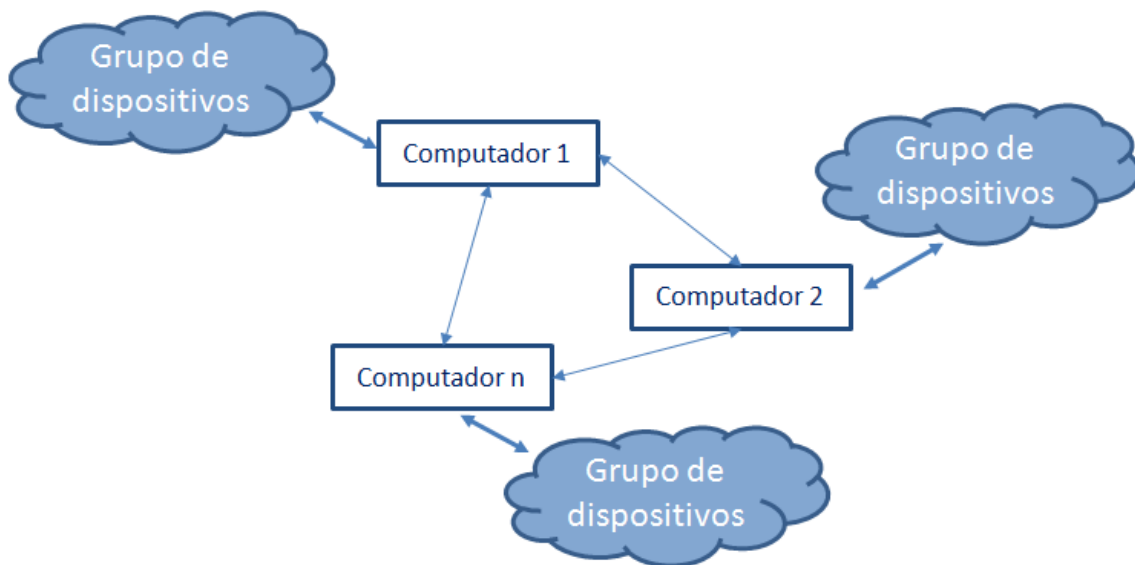


Figura 12. Control distribuido

Ventajas:

- Mayor robustez a fallos de algún microprocesador
- Mayor eficacia debida al proceso paralelo
- Menor rigidez en los dispositivos
- Mayor escalabilidad
- Cobertura de plantas de mayor tamaño

Inconvenientes:

- Necesidad de alto flujo de datos en caso de distribución automática de tareas

Los sistemas distribuidos necesitan del acceso y comunicación tanto con los sistemas sensoriales como a los nodos de cómputo, y por lo tanto se deben tratar los siguientes temas:

- **Sistema distribuido de información sensorial.** Habrá que estudiar el acceso a la información de este tipo de sistema y la forma de fusionar la información proveniente de distintos sensores.
- **Sistema distribuido de cómputo.** Con lo que habrá que estudiar como estructurar el sistema y como medir el tiempo de cómputo. Se pueden proponer formas de balancear la carga computacional para conseguir un tiempo de cómputo adecuado que consiga cumplir las especificaciones del sistema.

Los sistemas distribuidos de control (DCS) se utilizan en la industria y en la ingeniería civil para la realización de aplicaciones de monitorización y control distribuido, permitiendo la intervención remota de un operador humano. Además, estos sistemas utilizan una red para interconectar sensores, controles, terminales de operador y actuadores.

Mientras la capacidad de computación de los sistemas empotrados se incrementa cada vez más, la tendencia en la tecnología de las redes de comunicaciones es cambiar desde el control específico de redes hacia Ethernet, basado en infraestructuras y comunicaciones inalámbricas. En este escenario, la estrategia del control debería ser tolerante a las variaciones en el retardo de las señales intercambiadas entre los elementos, la robustez frente a la pérdida de señales así como a la disponibilidad del ancho de banda entre las aplicaciones distribuidas. Algunos trabajos se han realizado en protocolos de comunicación con caracterización de tráfico, reparto de ancho de banda y sincronización de relojes (J. Coronel, y otros 2005). Pero este tipo de interacción física-informática motiva una gran cantidad de innovaciones en muchos campos relacionados con la tecnología de la información incluyendo las arquitecturas distribuidas y el diseño de controladores (Lee 2006).

3.3 Control empotrado distribuido

La seguridad es fundamental en los sistemas de control empotrados. Independientemente del número de variables a controlar por el mismo procesador, los sistemas con requisitos de tiempo real crítico⁵ deben asegurar el envío y la entrega de las acciones de control a todos los actuadores. La calidad de la señal enviada depende del nivel de procesamiento: datos usados, algoritmos de computación, disponibilidad de recursos,..., pero siempre debe asegurar un funcionamiento adecuado del sistema (Albertos, Crespo y Simó 2006).

⁵ *Hard Real-Time Systems*

Además del mal funcionamiento de los componentes, en los sistemas de control distribuidos complejos, la seguridad puede estar afectada por la variación de la dinámica del sistema controlado que requiere conmutación de controladores, pérdida de plazos de entrega en la ejecución de las tareas, pérdida de mensajes y variación en los retardos de las comunicaciones. En este contexto, para la ejecución de aplicaciones de control en un modo seguro, se deben tener en cuenta las siguientes actividades:

- Activación de la conexión de comunicaciones con otras actividades.
- Algunos datos deben ser guardados, representados, almacenados en una base de datos y actualizados.
- Debe existir como mínimo un controlador que calcule la acción de control basado en los datos disponibles en cada instante de tiempo y usando algoritmos predefinidos.
- De acuerdo con el comportamiento del sistema, debe anticipar acciones como: desconectar y conmutar controladores. Los controladores son partes de código que se ejecutan diseminados en el entorno distribuido.
- Si la acción de control no ha sido enviada por el regulador actual a tiempo, una acción de control segura debería ser enviada al proceso. Esta señal puede ser el resultado de un cálculo sencillo (pero suficientemente segura) con una desconexión de emergencia o simplemente una acción de *back-up* que puede ser mantener sin cambios a la acción de control. Esta operación puede interpretarse como una conmutación entre distintos controladores.

3.4 Control inteligente distribuido

A continuación se presentan unas cuantas ideas relativas al control inteligente distribuido. Este tipo de sistemas, además de la realización del control, permiten representar y mantener la información en el sistema distribuido.

3.4.1 Control inteligente distribuido basado en agentes

En (Stothert y MacLeod 1997) se define a un agente como la ejecución de un programa autónomo en un procesador que forma parte de una red de procesadores distribuida.

El control inteligente distribuido tiene como objetivo utilizar múltiples agentes semi-autónomos para el control de un proceso y de esta forma cumplir sus especificaciones operacionales. Cada agente normalmente implementará un paradigma (posiblemente “inteligente”), y a través de la comunicación entre ellos los agentes podrán interactuar y coordinarse para implementar una estrategia de control

completa. El uso de múltiples agentes, cada uno de ellos abordando una parte del problema de control, hace al regulador inteligente distribuido más flexible y potente que un agente individual.

Los sistemas de control inteligente distribuido basado en agentes son utilizados en campos como el control de múltiples robots (Parker 2008) (Bullo, Cortés y Martínez 2009) y los procesos de fabricación distribuida (Leitao 2009).

La idea de utilizar agentes semi-autónomos para el paso de información se puede traspasar a los diferentes nodos del sistema distribuido de control, de manera que en cada instante cada uno de los nodos del sistema distribuido pueda mantener la información adecuada para su desempeño mediante la comunicación con los demás nodos del sistema.

3.4.2 Características de tiempo real de los procesos continuos.

El control en tiempo real de procesos de estados continuos requiere que las acciones de control sean las correctas y que se produzcan en el tiempo adecuado. Esto significa que el regulador en tiempo real debe tener la capacidad de detectar y responder a cambios en las características del proceso lo suficientemente rápido como para poder cambiar estas características (I. M. MacLeod 1984). Estos plazos (*deadlines*) en los sistemas de tiempo real estricto implican que un regulador inteligente debe proporcionar una acción de control adecuada antes de cada plazo. Por lo tanto, los paradigmas puestos en ejecución deben ser capaces de proporcionar una solución completa, o si esto no es posible, una solución parcial correcta o cercana antes del plazo. La elección de los paradigmas de los agentes no debe centrarse solamente en su capacidad de solucionar el problema requerido, sino en proporcionar soluciones incompletas en el caso de que no pueda cumplirse un plazo.

Por lo tanto, desde el punto de vista de un sistema de control en tiempo real se requiere la aplicación de la acción de control correcta en el momento adecuado. Este es uno de los objetivos del presente trabajo, ya que lo que se pretende es la aplicación de una acción de control en el sistema distribuido aunque se produzcan situaciones de pérdida de datos o retardos excesivos, con lo que es importante tener alternativas que nos permitan obtener una acción de control válida con los datos disponibles.

3.4.3 Representación del conocimiento en los sistemas de control inteligente distribuido.

Las técnicas de inteligencia artificial pueden ser caracterizadas en términos de la representación del conocimiento que se ha utilizado. La representación determina cómo se manipula el conocimiento (Reeke y Edelman 1990) y consecuentemente cómo afecta a la “inteligencia” de la técnica. En inteligencia artificial distribuida, no sólo es importante la representación de conocimiento sino los métodos de distribuir/compartir el conocimiento entre agentes. Para procesos dinámicos de

estados continuos es conveniente distinguir entre dos tipos de conocimiento, concretamente, el conocimiento previo (a priori) y conocimiento operacional.

El conocimiento previo define qué se sabe sobre la planta. Se utiliza durante el diseño del regulador para determinar la estructura del sistema distribuido, cómo se eligen los agentes y el propósito de cada uno. El conocimiento previo se utiliza para construir un marco que facilita una solución para el problema de control que se está tratando. El conocimiento operacional es el conocimiento disponible durante la operación del proceso, se comunica entre los agentes que lo usan en el sistema distribuido para determinar las acciones de control futuras. Mientras que el conocimiento previo determina un marco para el regulador distribuido, el conocimiento operacional se procesa por el marco de control. Debe tenerse en cuenta que el conocimiento operacional es dinámico ya que describe el estado del proceso controlado en un instante de tiempo en concreto. Para los procesos dinámicos de estados continuos esto significa que el conocimiento operacional es un conocimiento típicamente numérico. Los procesos de estado continuo requieren a menudo paradigmas “inteligentes” que puedan procesar conocimientos numéricos o extraer conocimientos de los datos numéricos continuos (Stoother y MacLeod 1994).

Los conocimientos previo y operacional son complementarios uno del otro, como se ha mencionado anteriormente, el conocimiento previo define el marco del controlador distribuido mientras que el conocimiento operacional es el conocimiento usado por el marco. Si el conocimiento previo no puede definir totalmente el marco del regulador, el conocimiento operacional debe llegar a ser más complejo para distribuir exactamente las operaciones entre los agentes y por tanto, el protocolo de comunicación entre agentes se hace más sofisticado. Un ejemplo es donde se sabe que agente se requiere y su propósito, como resultado del conocimiento previo, pero no se sabe cómo se encuentran interconectados. Aquí el conocimiento operacional debe permitir la comunicación (conexiones y contenido) entre los agentes a cambiar durante la operacional. Una metáfora de negociación (MacLeod y Lun 1991), (Davis y R.G. 1983) protocolo de comunicaciones debería usarse para comunicar el conocimiento operacional a través del sistema distribuido.

Este tipo de situaciones será las que tenga que tener en cuenta el nodo supervisor cuando se trabaje con varios nodos locales. Cada uno de los nodos locales puede interactuar con el proceso a controlar y es necesaria la coordinación y la comunicación entre los mismos para distribuir el control entre dichos nodos.

3.5 Comunicaciones en el sistema distribuido de control

En (Cauffriez, y otros 2004) se estudian las comunicaciones en los sistemas distribuidos de control. La utilización de buses de campo y la aparición de sensores y actuadores inteligentes están abriendo nuevas posibilidades en los sistemas de control distribuidos, pero también están introduciendo restricciones adicionales en cuanto a la fiabilidad de los objetivos alcanzados. El tipo de ambiente productivo

determinará ampliamente el criterio predominante para el sistema de control automático, es decir, la confiabilidad, disponibilidad, la capacidad de mantenimiento, seguridad, etc. Por otra parte, la elección del bus de campo dependerá de factores tales como el tamaño de la aplicación, rendimiento del procesado de datos, y la integración con las restricciones temporales. Otros criterios importantes incluyen el coste, confidencialidad, y compatibilidad con el equipamiento existente. Por lo tanto, parece esencial que los diseñadores tengan en cuenta los medios para calcular la fiabilidad en cada etapa de diseño realimentado integrado desde la experiencia. La determinación de la fiabilidad a menudo se limita a una evaluación al final del proceso de diseño, que en la mayoría de los casos implica reelección las opciones anteriores.

En definitiva, un sistema distribuido está constituido por dispositivos que contienen muchas funciones e intercambian y comparten datos.

En un sistema distribuido que dispone de un canal de comunicaciones compartido hay que considerar en el algoritmo de control las características propias de este sistema. En el *Workshop* (WIDEoNCS 2009) se expusieron las características de los sistemas de control en red y se analizaron los principales problemas derivados de las comunicaciones en el sistema distribuido, que se pueden resumir en:

- Pérdida de datos entre los nodos del sistema distribuido
- Tiempo de espera excesivo (equivalente a una pérdida de datos)
- Periodos de actualización de las señales distintos en cada nodo de la red
- Retrasos en el envío y la recepción de las señales
- Desfases
- Ruido en la transmisión

3.6 Middleware de control

La aplicación de técnicas de control a un sistema software requiere de ciertos servicios genéricos que son comunes a la mayoría de las aplicaciones. A menudo, estos tienen que ver con el sensor y el actuador del bucle de control (ARTIST 2008).

Un middleware es una infraestructura de software que se ha utilizado con éxito para integrar y administrar sistemas distribuidos complejos. La mayoría del middleware se ocupa de un dominio en particular, como los servicios web, y define una arquitectura simple y uniforme para el desarrollo de aplicaciones en ese dominio. Los mecanismos estándar para definir interfaces software y funcionalidades fomentan el desarrollo de software bien definido y reutilizable. Por ejemplo, los dispositivos hardware como sensores y actuadores pueden potencialmente incluirse con software complejo como unidades modulares. Un middleware apropiado permite luego que esas unidades puedan integrarse fácilmente en el algoritmo de control. Además, los servicios del middleware proporcionan funcionalidades estándar como soporte para robustez y tolerancia a fallos, que pueden ser utilizados fácilmente en la mayoría de

las aplicaciones. Por consiguiente, el middleware proporciona un marco bastante efectivo y probado para el desarrollo, integración y gestión de elementos heterogéneos software de los sistemas de control empotrados distribuidos (Baliga y Kumar 2005).

El middleware ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red). El middleware nos abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware.

La tecnología middleware se utiliza habitualmente en los sistemas distribuidos para proporcionar servicios de comunicación. Algunos ejemplos son Java-RMI, COM de Microsoft, y CORBA. Los esquemas middleware también están disponibles en tiempo real y sistemas empotrados, por ejemplo, RT-CORBA y Embedded CORBA. Hay también una gran cantidad de investigación relacionada con las redes invasivas para aplicaciones de sistemas empotrados, por ejemplo, los sistemas móviles y sistemas de sensores. Algunos ejemplos de estos son GAIA (Romn, y otros 2002), WSAMI (Issarny, y otros 2005), y AURA (Garlan, y otros 2002).

En los sistemas de control distribuidos con elementos empotrados, que es el entorno en el que se va a desarrollar esta tesis, en donde se tienen nodos con diversas cantidades de recursos hardware, resulta conveniente establecer diversas clases de servicios middleware a ser implementados en función de la restricción de los recursos de cada sistema de cómputo en el que se instale.

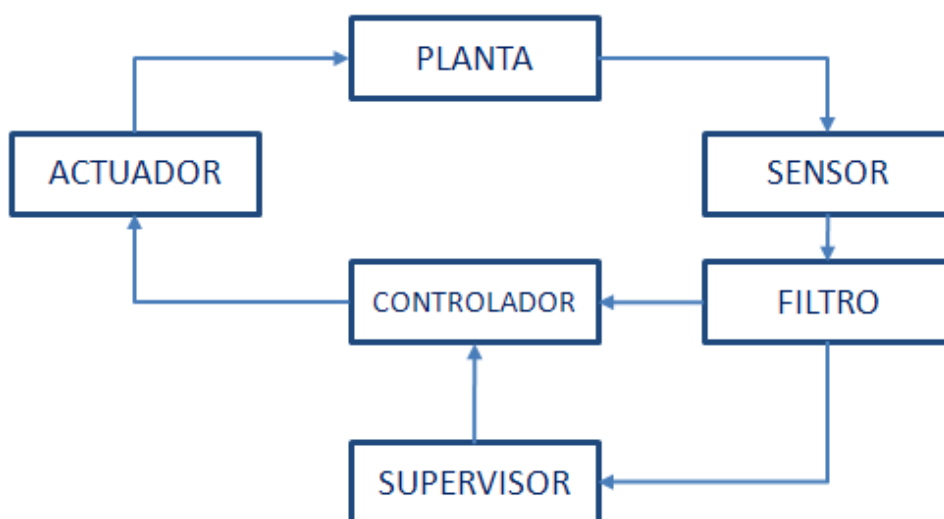


Figura 13. Software basado en componentes para un control digital. Fuente: adaptación de (Baliga y Kumar 2005).

Una arquitectura software basada en componentes tiene varias ventajas (Baliga y Kumar 2005). Dado que los componentes están bien definidos pueden ser reemplazados sin afectar al resto del sistema. Una arquitectura de componentes también permite que cada uno de ellos pueda desarrollarse por separado y que sea fácilmente integrado más tarde, que es muy importante para el desarrollo de grandes sistemas. Además, esa arquitectura promueve la reutilización de software, ya que un componente bien diseñado para un algoritmo de control, probado para un sistema, puede ser trasladado a otro sistema similar.

En los sistemas de control en red se tendrán la ejecución de componentes software en múltiples nodos de computación conectados en red. Los bucles de control pueden ser distribuidos, tal como se muestra en la Figura 14, y los componentes software tendrán que comunicarse a través de la red. En estas condiciones, es prácticamente imposible garantizar los periodos de comunicación en los plazos de un sistema de tiempo real duro, tales como redes inalámbricas y redes basadas en IP. Por lo tanto, un objetivo muy importante es proporcionar abstracciones del software apropiadas que permitan usar la teoría del control digital en estos sistemas.

La ubicación virtual (*virtual collocation*) es la abstracción de que todos los componentes software se ejecutan en un único equipo. Esto permite ocultar los detalles acerca de la ubicación de redes, topologías y protocolos de comunicación, y así el software de control no distingue entre los componentes locales y remotos.

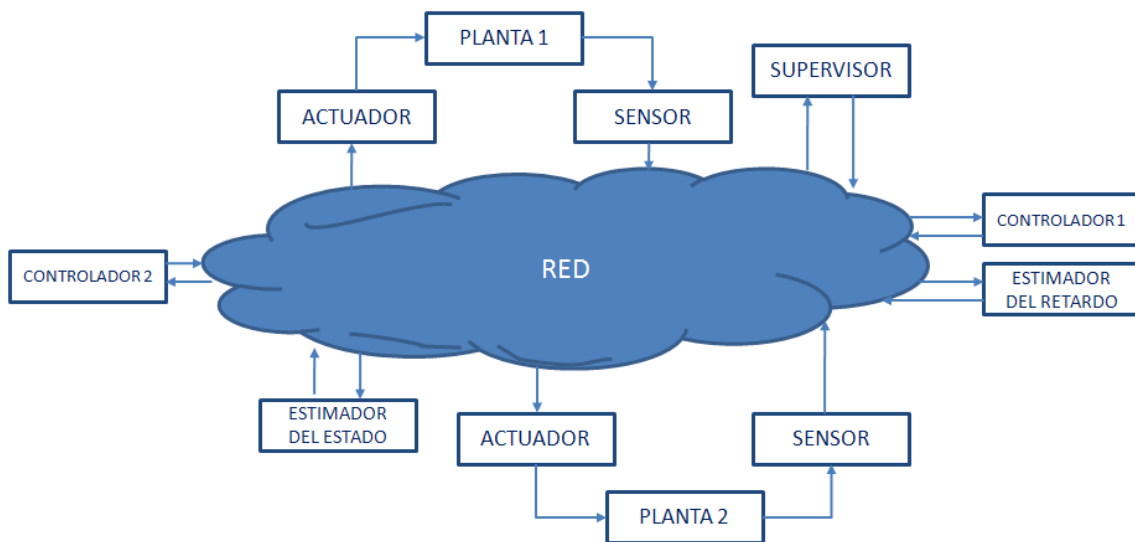


Figura 14. Sistema de control en red. Fuente: adaptación de (Baliga y Kumar 2005)

Esta abstracción tiene varios beneficios adicionales. Dado que no se hacen suposiciones acerca de la localización de la red, los mismos componentes pueden ser reutilizados en diferentes sistemas y sobre diferentes redes. De hecho, los componentes pueden migrar dinámicamente para optimizar el procesamiento y el uso de la red. En consecuencia, la ubicación virtual no sólo es una abstracción para

simplificar el control, sino también una herramienta muy útil para mejorar la integración de sistemas, gestión y reutilización.

La autonomía temporal local (S. Graham 2004) es un método para mejorar la robustez y fiabilidad. Consiste en asegurarse que un componente puede ejecutarse una cantidad de tiempo pequeña, incluso después de que otro componente conectado haya fallado. Por ejemplo, usar un estimador del estado puede ayudar al regulador a soportar retardos o pérdidas en un sensor. Además, en caso de fallo en el software, si el sensor se reinicia lo suficientemente rápido, el regulador puede ejecutarse sin darse cuenta del fallo. Éste y otros mecanismos similares permiten tener un sistema robusto ante retardos en la red, transitorios, y tolerancia a fallos, y ayudan a realzar la abstracción de la colocación virtual.

3.6.1 Principales middlewares dedicados al control

Tan sólo unas pocas soluciones middleware se han desarrollado específicamente para sistemas de control. Algunos de los middlewares dedicados al control son:

- **ControlWare** (Zhang, y otros 2002) es un middleware con una arquitectura de control con calidad de servicio (QoS⁶), originalmente diseñado para ayudar a los programadores a aplicar la teoría de control en el desarrollo de software de control. Permite al usuario expresar especificaciones QoS *off-line*, trazando estas especificaciones apropiadamente en bucles realimentados, sintonizando los controladores para garantizar la ejecución de varias especificaciones, y conectar adecuadamente los bucles a los sensores y actuadores en el código de la aplicación de tal manera que se logre la QoS deseada (Abdelzaher, y otros 2003). El objetivo de ControlWare es aislar la programación de la aplicación software de cuestiones teóricas de control. Al mismo tiempo, aísla al ingeniero de control de relacionar el regulador al sistema software de control y el diseño software de ejecución de sensores y actuadores.
- **Etherware** (Baliga y Kumar 2005) es un middleware para control con una arquitectura software basada en componentes, donde un componente es un módulo autónomo software con una funcionalidad bien definida. Por ejemplo, sensor, actuador, controlador, filtro y supervisor son los posibles componentes para un control digital como el de la Figura 13.
- **CAMRIT** (*Control-based Adaptive Middleware for Real-time Image Transmission*) (Wang, y otros 2004) propone una arquitectura middleware para la transmisión de imágenes en tiempo real. Su objetivo es la transmisión

⁶ QoS: *Quality of Service*

de la imagen desde el nodo servidor al cliente dentro de un plazo determinado maximizando dinámicamente la calidad de la imagen a transmitir. CAMRIT está diseñado como un middleware para RT-CORBA, siendo todas sus tareas administradas de acuerdo al algoritmo de planificación Rate Monotonic.

- **TAO** (*The ACE ORB*) (Schantz, y otros 2006) es una arquitectura middleware para el desarrollo de software basado en patrones de programación de red descritos en (Schmidt, y otros 2000) y construidos sobre el marco ACE⁷ (D. Schmidt 1993). La incorporación de APIs⁸ implementando el estándar RT-CORBA permite cumplir los requisitos de tiempo real de aplicaciones con prioridad fija. El núcleo ORB⁹ gestiona las conexiones, reparte las peticiones de los clientes en un adaptador de objetos y responde a los clientes retornando en su caso los datos necesarios. También es responsable del manejo del modelo de concurrencia utilizado por los componentes de la aplicación. Uno de los propósitos de TAO es proporcionar calidad de servicio (QoS) extremo a extremo.
- **Middlestar** (JFE Technical Report 2007) es un middleware que permite la renovación de los sistemas de control de procesos a un bajo coste. Este es un paquete middleware diseñado para facilitar el desarrollo de sistemas de control en computadores de propósito general con la funcionalidad y las prestaciones requeridas para cumplir las necesidades de un control de procesos de alto nivel. Middlestar soporta dos tipos de programa en base al modo de funcionamiento: dirigido por eventos con prioridad y dirigido por mensajes. Estos dos modos de funcionamiento pueden ser combinados para obtener un software adecuado que cumpla los requisitos del sistema.
- **MiRPA** (*Middleware for Robotic and Process Control Applications*) (Technical University of Braunschweig 2009) es un middleware de tiempo real basado en computadores de propósito general que soporta modelos de comunicación cliente/servidor y publicador/subscriptor. El middleware está dirigido por mensajes, y ha sido diseñado para cumplir con demandas espaciales en aplicaciones de robótica y automatización. Permite la integración de nuevos sensores y actuadores fácilmente sin necesidad de cambiar el núcleo del sistema de control.

⁷ ACE: *Adaptive Communication Environment*

⁸ API: *Application Program Interface*

⁹ ORB: *Object Request Broker*

- **Inter Component Exchange Manager** es un middleware creado dentro del proyecto EEA (Embedded Electronic Architecture) en el que se especificaron nuevas soluciones para el desarrollo de sistemas empotrados en un vehículo (Simonot-Lion y Trinquet 2003). Este middleware permite asegurar la independencia de los componentes de las aplicaciones software mediante una especificación común de servicios. Además, se especificó un lenguaje de descripción de arquitectura con una interface de la aplicación común denominado AIL_Transport.

3.6.2 Requisitos a cumplir en el middleware de control. Consideraciones de la arquitectura

El soporte a la colocación virtual y promover el desarrollo de software reutilizable impone muchos requisitos en el *middleware*. A continuación se muestra una lista de los requisitos que debe cumplir el middleware de control (Baliga y Kumar 2005):

- **Modularidad:** La mayoría de los sistemas se diseñan de forma modular, y el software se forma con componentes bien definidos. Esto promueve la reutilización de software y permite que los componentes individuales sean substituidos sin afectar al resto del sistema. Sin embargo, hay que asegurar que los efectos colaterales se reduzcan al mínimo durante tales cambios, los componentes y sus substitutos deben ser compatibles. Por ejemplo, un regulador puede ser reemplazado solamente por otro que funcione en una planta similar y que interactúe con componentes similares con protocolos compatibles. El middleware debe promover el desarrollo de aplicaciones con tales componentes.
- **Localización independiente:** La asignación de computación a los procesadores es una tarea de bajo nivel que requiere información detallada sobre las configuraciones del sistema. La proliferación de las redes de sistemas de control facilita la reducción de tiempo en el diseño del ciclo. Puesto que la mayoría de tales tareas pueden ser automatizadas, el middleware es el que debe asignar y optimizar esas tareas, ya que tiene acceso a los detalles de la configuración. Sin embargo, para permitir esa independencia de la localización, el middleware debe soportar arquitecturas de aplicación de forma que el mismo componente pueda potencialmente ser localizado apropiadamente para diferentes computadores en diferentes escenarios.
- **Colocación virtual:** En un sistema en red, los componentes software se ejecutan en diversos computadores con los relojes potencialmente no sincronizados. Por lo tanto, la localización de los componentes en la red debe ser rastreada, y las marcas temporales (*time-stamp*) deben ser trasladadas. Sin embargo, puesto que esta información se abstrae lejos de la colocación virtual, los mecanismos apropiados deben ser implementados en el middleware. Por otra parte, estos mecanismos deben ser transparentes a los componentes

software de forma que esas aplicaciones no tienen que seguir los cambios de configuración.

- **Robustez:** Los sistemas de control interactúan con el mundo real y por lo tanto deben ser robustos a los fallos. Como se ha visto anteriormente, varias técnicas se pueden utilizar para promover la autonomía temporal local de componentes para poder tolerar tales fallos. Sin embargo, los mecanismos clave, como reiniciar de forma eficiente los componentes, son necesarios para complementar tales técnicas y no afectar a otros componentes conectados durante esos cambios.
- **Flexibilidad:** Los componentes necesitan ser iniciados correctamente tras el arranque. En particular, instancias reutilizadas de un componente pueden tener distintos roles en diferentes aplicaciones. El middleware debe soportar adecuadamente la inicialización y gestión de tales componentes.
- **Evolución del sistema:** El soporte para la evolución del sistema es necesaria para promover la longevidad de los sistemas de control en red. Por ejemplo, la actualización del proceso puede requerir de una actualización del regulador.
- **Migración de componentes:** La migración de componentes pueden optimizar la configuración del sistema reduciendo o balanceando las cargas de computación y de comunicaciones. Esas operaciones pueden y deberían hacerse mientras el sistema está funcionando, es decir, en tiempo real. En particular, el diseñador no debería preocuparse de los detalles de la asignación y optimización de recursos.

3.7 Resumen y conclusiones del capítulo

En este capítulo se han descrito las principales características de los sistemas distribuidos de control, así como de los controladores implementados en elementos empotrados, prestando especial atención a los controladores inteligentes distribuidos. Además, se hace un análisis de las comunicaciones en el sistema distribuido y de las características de los servicios middleware dedicados al control, haciendo una breve descripción de aquellos dedicados especialmente a tareas de control.

Las conclusiones correspondientes a este capítulo son las siguientes:

- En los sistemas formados por múltiples componentes de diferentes características se sugiere un sistema distribuido de control (DCS).
- Los sistemas de control se ejecutarán en distintos sistemas de computación comunicados.
- Los recursos de computación de cada nodo empotrado, así como las características temporales del BUS de campo para comunicaciones, deben considerarse en el diseño.

- La estrategia de control debe ser tolerante a las variaciones en el retardo de las señales intercambiadas entre los elementos, la robustez frente a la pérdida de señales así como a la disponibilidad del ancho de banda entre las aplicaciones distribuidas.
- Los sistemas de control deben asegurar el envío y la entrega de las acciones de control a todos los actuadores.
- Un middleware ofrece una adecuada separación entre el sistema operativo (si lo hubiese) de cada nodo y la aplicación de control, brindando un diseño rápido de aplicaciones y desarrollo transparente.
- El middleware facilita la implementación del sistema distribuido de control ofreciendo servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

Parte III. Propuesta de Diseo de Reguladores Modulares en un Entorno Distribuido

4 NÚCLEO DE CONTROL DISTRIBUIDO

4.1 Motivación y objetivos del capítulo

El objetivo de este capítulo es adaptar el núcleo de control para ser utilizado en un entorno distribuido como parte del sistema de control. Una de las características a tener en cuenta en este sistema es que alguno de los nodos que lo forman serán elementos empotrados con escasos recursos de computación. Además, la comunicación entre algunos elementos se realizará a través de un canal de comunicaciones compartido, por lo que el algoritmo de control debe hacer frente a los posibles problemas derivados de esta configuración: pérdida de datos, falta de sincronización de los nodos, desfases, retardos...

La implementación del núcleo de control lleva aparejada su relación con el núcleo del sistema operativo en el que se ejecuta la aplicación. La revolución que actualmente ofrece la miniaturización, comunicación ubicua y la convergencia digital está impulsando a los sistemas empotrados a distribuirse y ejecutar selectivamente distintas actividades, convirtiéndose en sistemas fuertemente interconectados. Esto proporciona la perspectiva de nuevos niveles de complejidad en el control que, consecuentemente conducirán a nuevos niveles de seguridad, confort y productividad en todas las áreas cubriendo desde los entornos industriales hasta los individuos. Estas características requieren la consideración conjunta de las cuestiones de diseño del control y de su implementación en entornos con limitación de recursos. Hay que tener en cuenta la relevancia de cada señal de control y, consecuentemente (ello determina los retardos máximos aceptables), para cada tarea de control, modificar adecuadamente las prioridades de cada actividad para distribuir el tiempo de cálculo, reducir los posibles retardos en la entrega de cada señal de control o, en última instancia, equilibrar estos retrasos para conseguir la menor degradación de prestaciones de todo el sistema de control obligado a operar con la limitación de recursos que la autonomía impone.

La modularidad y adaptación del sistema así como la capacidad de desarrollo rápido de aplicaciones de control se sustentarán en el desarrollo de servicios de soporte, llamados *middleware*, necesarios para ofrecer a los algoritmos de control soporte para sistemas distribuidos, computación ubicua, movilidad de código y restricciones de tiempo real.

Una vez definido el núcleo de control se realizará la adaptación del mismo para utilizarlo en el marco de un sistema distribuido, en el que se darán solución a los problemas de control planteados anteriormente y que suponen el objetivo del presente trabajo, aplicando y analizando la estrategia de control a los sistemas ciberfísicos, presentados en el apartado 2.4.

La aportación principal de este capítulo es la adaptación del núcleo de control en un entorno distribuido, para que utilice controladores que aseguren un comportamiento adecuado en caso de pérdida temporal de información, tiempo excesivo de espera, consumo excesivo de recursos computacionales o ancho de banda en las comunicaciones. En los siguientes apartados de este capítulo se mostrará como la utilización de la estrategia de control desarrollada mejora la respuesta de control del sistema.

4.2 Núcleo de control

La definición del núcleo de control es asimilable al núcleo de un sistema operativo, y fue desarrollado en el proyecto de investigación KERTROL. El núcleo de control determina cuál es el código mínimo que debe ejecutarse en una aplicación de control para que el funcionamiento sea seguro, aunque pueda presentar una fuerte degradación de prestaciones o incluso evolucionar hacia un estado de desconexión segura.

4.2.1 Definición y principales características del núcleo de control

La misión principal del núcleo de control es garantizar un funcionamiento seguro del proceso que se está regulando para el cumplimiento de las especificaciones deseadas. Así, el núcleo de control tiene que tomar las decisiones, en última instancia, que permitan llevar a cabo esta tarea de la forma más eficaz posible. Además, tal y como se ha definido anteriormente, debido a su implementación en sistemas empotrados, el código empleado en este núcleo debe ser mínimo.

El núcleo de control debe tener un carácter modular, de tal forma que, en función de la disponibilidad de recursos y de las tareas encomendadas a un sistema empotrado conectado en red, se estructure en niveles de operación que garanticen la operación segura del sistema.

La implementación del núcleo de control se realizará en un entorno de núcleo del Sistema Operativo (SO), aunque no tiene por qué formar parte de él como se verá en

el apartado 4.2.2, ofreciendo opciones compatibles con los distintos niveles de ejecución del SO.

Teniendo en cuenta las funcionalidades principales que debe incluir el núcleo de control, expuestas en 2.6, a continuación se proponen los elementos necesarios y la arquitectura a implementar.

4.2.2 Elementos y arquitectura del núcleo de control

En el capítulo 2 se estudiaron las características que debía de presentar un sistema empotrado de control desde el punto de vista de la implementación, computacional, el algoritmo de control y el sistema operativo empotrado. Estos conceptos se han trasladado al núcleo de control.

El núcleo de control puede ser implementado de dos maneras, como una parte anexa al sistema operativo o como un middleware que se sitúa entre la aplicación de control y el sistema operativo, si lo hubiese en su caso. En (Crespo, y otros 2006) se indican las características de cada una de estas implementaciones:

Como parte del sistema operativo. Además de los servicios que proporciona el sistema operativo, se deben ejecutar las nuevas funcionalidades requeridas por el núcleo de control. En este caso, el núcleo de control será parte del sistema operativo y el interfaz de programación de aplicaciones (*Application Programming Interface*, API) debe incluir nuevas llamadas para definir los parámetros de control y conectarlos cuando sea necesario.

La elección de una implementación del núcleo de control como parte del sistema operativo supondría la modificación del mismo, con lo que se debe realizar un esfuerzo importante en determinar qué modificaciones son las más adecuadas, esto tiene desventajas importantes, por lo menos en las fases iniciales de puesta en marcha del núcleo de control. Por otra parte, la inclusión de este componente puede introducir más complejidades de las necesarias, además, reduce la portabilidad del sistema operativo a otros sistemas empotrados.

Como un middleware específico. En este caso, el sistema operativo mantiene las mismas funcionalidades, y el núcleo de control se implementa como una biblioteca que es enlazada con la aplicación y una serie de servicios que se ejecutan sobre el sistema operativo.

Por lo tanto, un middleware ofrece una adecuada separación entre el sistema operativo, si lo hubiese, y la aplicación de control, brindando un diseño rápido de aplicaciones y un desarrollo transparente.

4.2.3 Middleware del núcleo de control

Debido a las ventajas que proporciona, en cuanto a diseño rápido de aplicaciones y desarrollo transparente, así como al incremento de la portabilidad del mismo a otros sistemas empotrados, es lógico pensar que la implementación mediante un middleware es la más adecuada.

En este apartado se definen las características que debe tener el middleware para conseguir transparencia en el paso de información entre los distintos elementos del núcleo de control, así como mejorar la conectividad de los mismos.

En la Figura 15 se muestra, de forma muy general, los elementos que forman parte de la implementación del middleware de control, tanto en el sistema operativo como en el entorno de aplicación. Dicho esquema hace referencia al núcleo de control que fue desarrollado en el proyecto KERTROL.

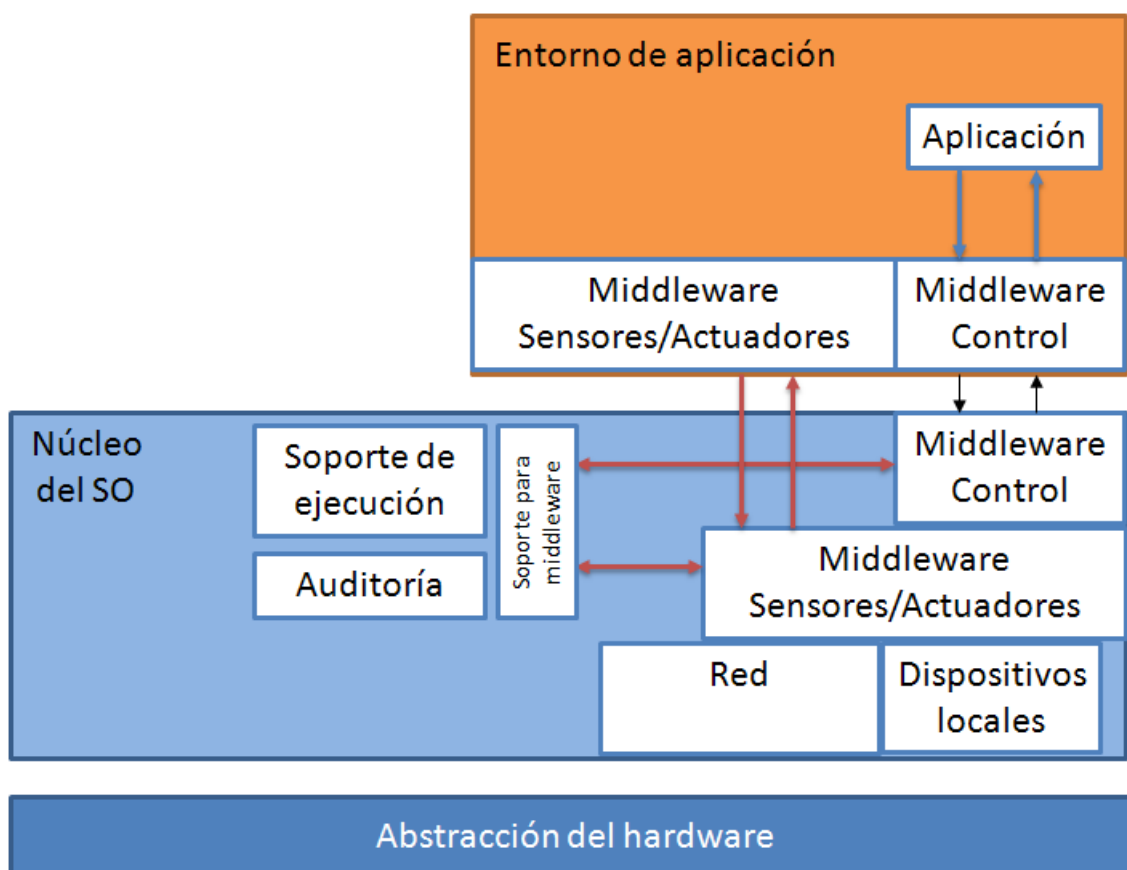


Figura 15. Arquitectura del núcleo de control

A partir de las definiciones y funcionalidades de los servicios middleware dedicados al control, que se han estudiado en el capítulo anterior, parece claro que el middleware debe estar basado en componentes, con localización independiente y colocación virtual. En este apartado se detallan las características y estructura del middleware implementado en el núcleo de control.

El middleware de control debe proveer, entre otros, transparencia en el diseño y desarrollo de aplicaciones de control, permitir la delegación de nuevos controladores, proporcionar mecanismos para que dicha delegación se produzca sin necesidad de detener el control del proceso, gestionar fallos que puedan producirse en el control y generar alarmas, dotar de mecanismos de tolerancia a fallos, ofrecer la posibilidad de realizar las tareas de sensorización, control o actuación en otros nodos de la red distribuida o definir una acción de control segura en caso de fallo del controlador.

Estas características confieren un gran atractivo para los ingenieros de control cuando éstos desarrollan aplicaciones, ya que con el núcleo de control se abstrae los recursos hardware del sistema pudiendo centrarse en los conceptos de control, obviando incluso la problemática de enviar siempre una acción de control segura en cada periodo de actuación o despreocupándose de las gestiones de delegación de código de nuevos controladores, por ejemplo.

Un aspecto especialmente importante para la industria es que el middleware brinda la posibilidad de delegación de código de nuevos controladores en caliente, esto es, la posibilidad de cambiar el algoritmo de control sin necesidad de detener el control del proceso. Esto permite ahorrar tiempo en el proceso de producción y por tanto, dinero.

Otro aspecto importante que ofrece el núcleo de control es la posibilidad de realizar un control distribuido, es decir, que mientras un nodo realiza las tareas de sensorización y actuación, otro nodo distinto puede realizar los cálculos del controlador. Ello permite poder realizar cálculos más complicados en nodos de la red de mayores prestaciones.

En la implementación de una estrategia de control existen distintos aspectos a tener en cuenta. Uno de ellos es la necesidad de recursos que se requiere, la disponibilidad de los mismos en un determinado momento, y que pueden corresponder a dispositivos con características distintas y con capacidades de comunicación también distintas. Esta última consideración es la que hace necesario el desarrollo de un *middleware* de control.

La interconexión del *middleware* con los componentes de control, exige el desarrollo de una serie de interfaces comunes a los componentes del sistema, independientemente de la tarea de control que desarrollen, de la ubicación en la que se encuentren dentro del sistema o de la tecnología por medio de la cual se han implementado.

La interfaz a desarrollar debe cumplir algunos requisitos, entre los cuales se encuentran los siguientes:

- Compatibilidad completa con las interfaces DCPS (*Data Centric Publish Subscribe*) del estándar DDS (*Data Distribution Service*) de OMG (*The Object Management Group*).
- Abstracción que oculte los detalles del *middleware* y se centre en los requerimientos de los componentes de control.
- Adaptabilidad y minimización en cuanto a que los componentes empotrados, que no siempre emplean toda la funcionalidad ofrecida por el *middleware*, por lo que el interfaz debe adaptarse a las necesidades de los componentes de control.

En (J. Coronel, y otros 2008) se detallan los requisitos mínimos que hay que considerar en la etapa de diseño e implementación de un *middleware* dedicado al control, que se resumen en:

- Debe asegurar una operación segura y fiable de todo el sistema de control de tiempo real. Para ello debe incorporar mecanismos de detección y aislamiento de fallos.
- La actualización y distribución de la información dentro del *middleware* de control debe estar minimizada.
- Los retardos en las tareas de control dentro del *middleware* de control deberán ser tenidos en cuenta en los bucles completos de control, los cuales serán debidos a que algunos de los recursos deberán ser compartidos dinámicamente entre diferentes actividades de cómputo.
- La utilización de la CPU deberá ser monitorizada y regulada mediante acciones que permitan su optimización.
- Cuando el *middleware* de control se ejecute en entornos autónomos, el sistema deberá adaptar sus actividades a la energía disponible.
- Para permitir un comportamiento autónomo se deberá considerar la inclusión de mecanismos que ofrezcan capacidades de distribución, reconfiguración y actualización de componentes software dentro del *middleware* de control, para lo cual se apoyará en el paradigma de delegación de código.
- Es necesario considerar que en entornos dinámicos la prioridad, las señales disponibles, la cantidad de memoria y el tiempo de asignación de recursos para las actividades de control podrán cambiar dependiendo de las condiciones de trabajo.
- Los cambios en el entorno y la disponibilidad de recursos deberán ser controlados por un nivel de supervisión y resolución que involucre decisiones

tales como la selección de las tareas de control más apropiadas en función del estado y el modo de operación actual del sistema.

- Aunque la mayoría de los algoritmos de control son manejados por tiempo mediante acciones separadas por intervalos constantes de tiempo, en entornos con cierta incertidumbre muchas veces es necesario disponer también de mecanismos para controlar eventos discretos.

4.3 Núcleo de control distribuido

En este apartado se desarrolla una metodología de diseño modular del algoritmo de control, de forma que se maximicen las prestaciones, se asegure la estabilidad de funcionamiento y se adapte a la disponibilidad de recursos cambiante en un entorno interactivo, proporcionando un funcionamiento autónomo seguro. La implementación de los controles requiere el desarrollo de un núcleo de control formado por una parte residente y otra distribuida.

Se plantea un paso adelante del desarrollo previo, ampliando el núcleo de control como soporte de sistemas de sensores y actuadores en red operando bajo restricciones de tiempo real. Esto representa la consideración del núcleo de control como una capa de middleware que ofrece servicios de control, comunicación y compartición de la información a todos los subsistemas conectados. El sistema a estudiar es heterogéneo, consistiendo en una serie de sistemas empotrados específicos interactuando entre ellos a través de los servicios proporcionados por los componentes middleware de control. En este escenario, las líneas de investigación activas son diversas y recientemente se han visto agrupadas en lo que se ha venido a llamar estudio de los sistemas ciber-físicos, descritos anteriormente en el apartado 2.4. Particularmente se deben tener en cuenta los siguientes aspectos:

- Modelos de interacción en sistemas ciber-físicos
- Co-diseño hardware-software bajo restricciones
- Middleware de soporte para la sensorización y el control
- Desarrollo de nodos con capacidad sensorial específica
- Esquemas de planificación del uso del procesador orientados a la ejecución de tareas de control con carga variable

Debido a la existencia de gran variedad de sistemas empotrados con diversas características como capacidades implementadas, tiempos de cómputo, etc., es necesario establecer diversos niveles de servicios middleware a ser implementados en función de la restricción de recursos prevista por cada sistema. En (J. Coronel, y otros 2008) se proponen dos modelos de middleware del núcleo de control

compuestos por diversos grupos de servicios: un middleware completo (*Full-Middleware*) y un middleware reducido (*Tiny-Middleware*).

La versión completa implementa todas las funcionalidades del middleware de control ofreciendo interfaces y mecanismos para la gestión de bucles complejos de control. Gestiona, organiza e interconecta toda la red distribuida de control. El nivel operativo está compuesto por un núcleo operativo de tiempo real.

Este modelo de middleware se utiliza en nodos con procesadores empotrados potentes con sistemas operativos de tiempo real y con altas capacidades de E/S y red.

La versión reducida del middleware de control se comunica con la versión completa mediante interfaces básicas para la gestión de sensores, actuadores y código de controladores. Además, incluye interfaces de red para la interconexión a sensores y actuadores, y para la descarga de código de controladores desde nodos con la versión completa. El nivel operativo está formado por un conjunto de API's de control hardware con planificación cíclica.

Este modelo está dirigido a nodos con procesadores con capacidad de cómputo y red limitada, pero con características completas de E/S. Ejemplo: PIC (controlador de interfaz periférico), DSP, etc.

En el trabajo de tesis de J. Coronel, no presentada en el momento de escribir esta parte (Mayo 2011), se describen los principales servicios y componentes tanto del middleware completo como del reducido del núcleo de control. A continuación se indican las principales características de esos dos middlewares que van a servir como base a la implementación de la estructura de control propuesta en esta tesis.

4.3.1 Middleware completo del núcleo de control

La versión completa del middleware del núcleo de control (CKM¹⁰) está conformada por un conjunto de componentes tal y como se muestra en la Figura 16.

¹⁰ CKM: *Control Kernel Middleware*



Figura 16. Componentes del middleware completo del núcleo de control

Cada uno de estos componentes tiene las siguientes funcionalidades:

Gestor de la aplicación: Define a una entidad software que ofrece una interfaz inmediata a las aplicaciones de control para acceder a los servicios ofrecidos por el middleware del núcleo de control. A cada aplicación de control se le asignará un componente de este tipo. Este componente tendrá el rol principal de gestionar el acceso de la aplicación al entorno middleware.

Gestor de QoS: La calidad de servicio (QoS) determina el nivel de satisfacción de un servicio, garantizando un cierto grado de rendimiento en la ejecución del servicio. Este componente se encargará de negociar y gestionar que la ejecución de las tareas propias del middleware y las aplicaciones del sistema se realice dentro de determinados niveles de rendimiento.

Gestor de delegación: Este componente se encargará de coordinar la delegación de código entre los distintos nodos de cómputo del sistema distribuido. Las decisiones de este componente dependerán del componente de aplicación, del gestor de recursos y del gestor QoS, y más específicamente, de las peticiones propias de las aplicaciones, de la disponibilidad de recursos locales para el cumplimiento de las restricciones temporales de las aplicaciones (factores de comodidad de ejecución), de la distribución física de sensores y/o actuadores, de sobrecargas en redes de comunicaciones, entre otras.

Gestor de datos: Este componente se basará en la especificación de OMG DDS (*Data Distribution Service* – Servicio de Distribución de Datos) (Object Management Group 2008) para la distribución de datos de forma predecible dentro del sistema distribuido de tiempo real. Este componente utilizará un modelo de intercambio céntrico de datos y más específicamente el modelo DCPS (*Data Centric*

Publish/Subscribe – Datos Centrales Publicador/Subscriber), el cual se construye sobre el concepto de espacio de datos global que es accesible por todas las aplicaciones interesadas.

Data: Hace referencia al objeto que es generado y consumido respectivamente por aplicaciones publicadoras y subscriptoras dentro del sistema middleware en lo que se conoce como “espacio global de datos”. Cada Data tiene asociado diversos subscriptores y es actualizado por una aplicación publicadora.

Gestor de recursos: Este componente en coordinación con el gestor de QoS, se encarga de administrar el uso de los recursos físicos de los sistemas de cómputo tales como memoria, CPU, ancho de banda, consumo de energía, entre otros.

Gestor de red: Este componente se encarga de administrar el hardware de red, accediendo a los controladores ofrecidos por el sistema operativo. Gestiona el transporte de mensajes en el sistema distribuido ofreciéndoles al middleware servicios de transmisión y recepción de datos a través de diversos buses de comunicaciones.

Gestor de eventos: Este componente informa al nivel de aplicación acerca de los eventos que ocurren dentro del middleware de control. La notificación de estos eventos al nivel de aplicación se hace mediante interfaces preestablecidas por el gestor de aplicación. Las aplicaciones recibirán por defecto notificaciones de eventos propios del middleware tales como pérdida de plazos de ejecución, errores de ejecución, entre otros, y además podrán instalar notificadores de eventos específicos a través de mecanismos de escucha (*listener*).

Servicio de Actuación: Es el componente que se encuentra físicamente conectado al proceso a controlar y cuya misión es la entrega de las señales de control en cada periodo de actuación, por lo tanto es el encargado en última instancia de hacer que se cumplan los requisitos temporales y hacer llegar los datos de manera adecuada.

Servicio Sensorial: Mediante este componente se realiza el intercambio de información entre el proceso a controlar y los controles involucrados. Con la información recibida en este componente el controlador debe elaborar las señales de control adecuadas en cada periodo de muestreo.

Los componentes están configurados con las interfaces mostradas en la Figura 17 y la Figura 18.

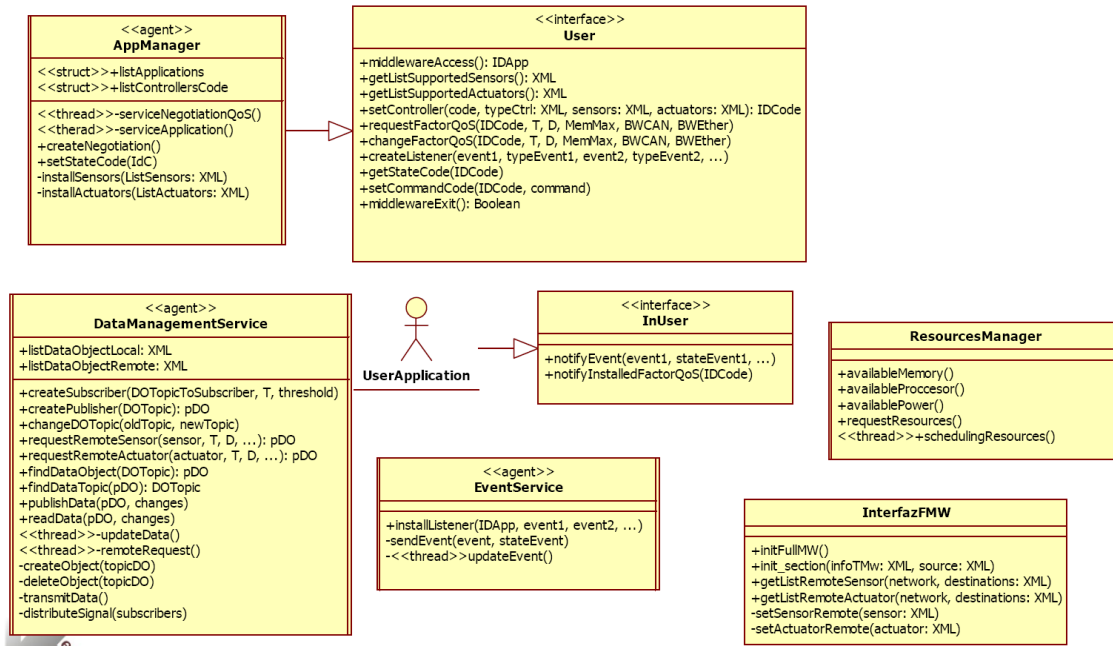


Figura 17. Interfaces de los componentes del middleware completo del núcleo de control (parte 1)

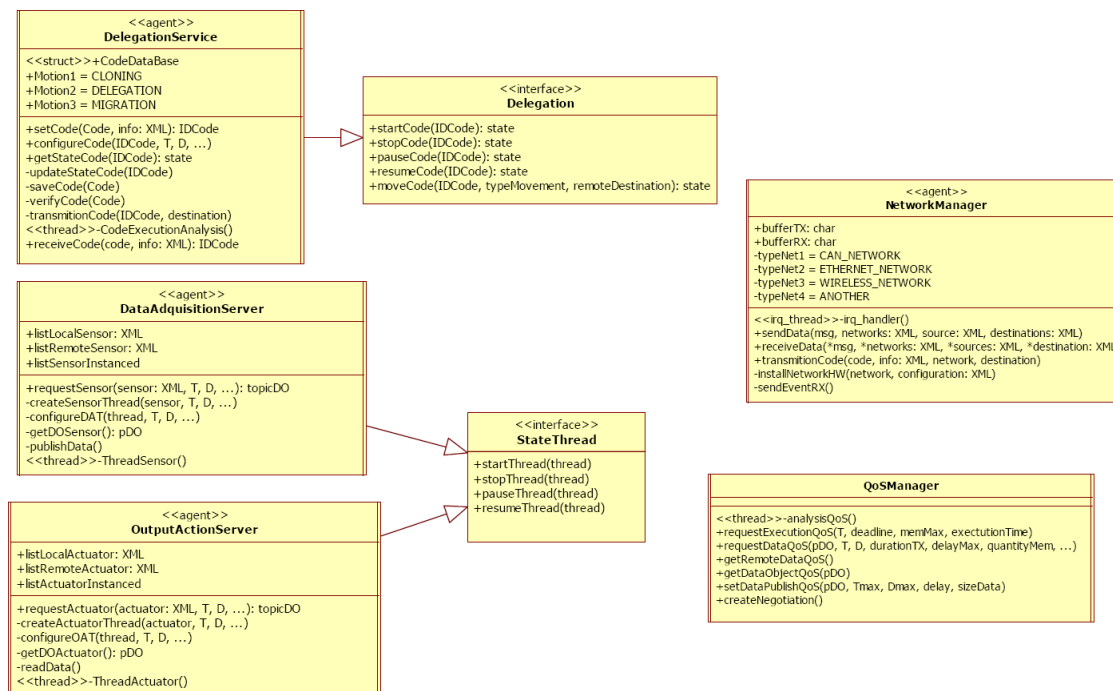


Figura 18. Interfaces de los componentes del middleware completo del núcleo de control (parte 2)

4.3.2 Middleware reducido del núcleo de control

El middleware reducido del núcleo de control (CKM *Runtime*) dispone de los servicios básicos necesarios para proporcionar la gestión, comunicación, sensorización, actuación e implementación de reguladores en aquellos nodos con

restricciones de cálculo. Los servicios necesarios para el control de procesos con restricciones temporales y la incorporación del movimiento de código en sistemas middleware reducidos para sistemas empotrados con recursos limitados se muestra en la Figura 19.



Figura 19. Componentes del middleware reducido del núcleo de control

Desde el punto de vista de implementación estos componentes son una versión reducida de los componentes definidos en el middleware completo del núcleo de control, aunque desde el punto de vista conceptual y lógico tienen las mismas características. Un servicio propio de este componente es el siguiente:

Servicio de eventos y monitorización: Es el encargado por una parte de monitorizar el control efectuado por el nodo de recursos limitados, y por otro lado tiene encomendadas tareas de administración del middleware, tales como por ejemplo el modo de funcionamiento del núcleo de control o el estado del nodo.

Para monitorizar el control realizado por el nodo se establecen unos parámetros de calidad que se van periódicamente evaluando. Cuando se decide que dicho parámetro no llega al límite de calidad establecido, entonces se genera la alarma correspondiente que es enviada a través de la red a los nodos supervisores, esto es, aquellos que implementan un middleware completo del núcleo de control.

La labor de administración tiene que ver con la gestión del estado en el que se encuentra el nodo y la gestión de cambio de modo de funcionamiento, informando adecuadamente a los nodos supervisores y realizando las acciones oportunas.

Los interfaces de los componentes del middleware reducido del núcleo de control se muestran en la Figura 20.

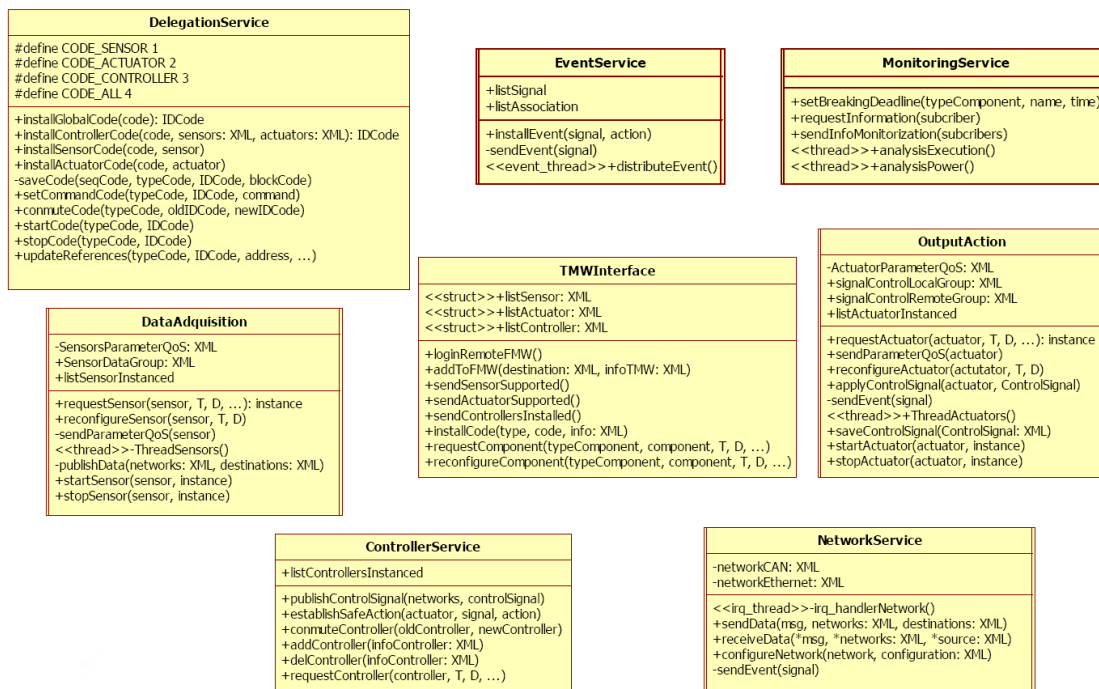


Figura 20. Interfaces de los componentes del middleware reducido del núcleo de control

4.3.3 Estructura del núcleo de control distribuido

Teniendo en cuenta los componentes definidos en el apartado anterior, el núcleo de control en un sistema distribuido está formado mediante dos tipos de nodos: Nodos Supervisores, también llamados nodos de servicio, y Nodos Locales, también llamados nodos ligeros (Figura 21). Los nodos supervisores son potentes computadores empotrados ejecutando un sistema operativo de tiempo real (RTOS¹¹) y con completas capacidades de red y entrada/salida. Los nodos locales son procesadores SoC¹² pequeños y de bajo consumo con capacidades limitadas de redes y computación, pero características completas de entrada/salida (Simarro, y otros 2008).

¹¹ RTOS: *Real-Time Operating System*, es un sistema operativo multitarea pensado para aplicaciones de tiempo real

¹² SoC: *System on Chip*, se basa en integrar todos los componentes de un computador u otro sistema electrónico en un simple circuito integrado (chip)

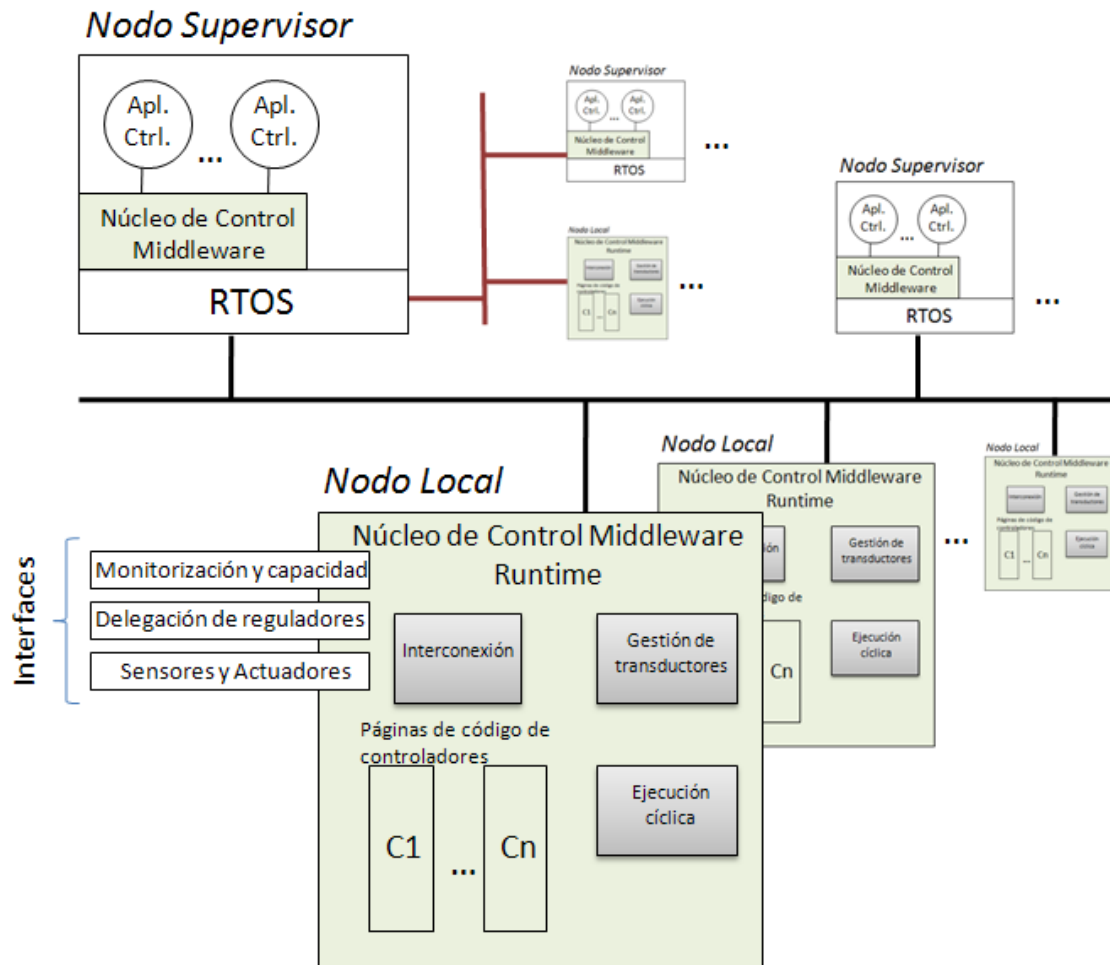


Figura 21. Arquitectura para el modelo de control en un sistema distribuido

4.3.3.1 Características principales de los nodos supervisores

Las aplicaciones de control de alto nivel se ejecutan en los nodos supervisores, que implementan un middleware completo del núcleo de control (CKM). Este middleware ofrece abstracciones y funcionalidades relacionadas con la ejecución de tareas de control en tiempo real, acceso a los sensores y actuadores, y gestión de comunicaciones, tal y como se han definido en el apartado 4.3.1. El modelo de programación del CKM sigue el concepto de delegación de código. En este sentido, una aplicación de control delega la ejecución de algún código de control al *middleware* que proporciona los recursos de computación para poder ejecutarlo. Una tarea de control, una vez dentro del CKM, puede ser ejecutada en cualquier nodo supervisor del sistema distribuido que tiene acceso al espacio de comunicaciones de las tareas.

Los nodos supervisores pueden incluir la supervisión y optimización de actividades de control de forma que se mejore el comportamiento global del sistema bajo control.

A continuación se describen las tareas realizadas por los servicios del middleware completo en este nodo supervisor para el tratamiento de los datos recibidos y enviados:

- Los datos a este nodo llegarán a través del gestor de red, provenientes de un nodo local. El gestor de datos se encargará de publicarlos mediante el modelo DCPS.
- La información proporcionada por la red, a través de la publicación del gestor de datos, es utilizada por el gestor de aplicación para elaborar la acción de control.
- Dependiendo del modo de funcionamiento que se esté ejecutando, y que se detalla en el apartado 4.5.6, en lugar del envío de la acción de control para ese instante de tiempo se puede enviar los vectores con las trayectorias de las salidas y de las acciones de control, calculadas mediante un regulador predictivo, u otro regulador que permita la obtención de dichas trayectorias.
- El funcionamiento de este sistema se debe hacer coincidiendo con los múltiplos del periodo, sin embargo, puede que en algunos casos no sea necesario el envío de ninguna información. En el caso de que el nodo local utilice la información previamente calculada por el nodo supervisor, éste puede que no tenga que realizar ningún cálculo, y por lo tanto en esos instantes puede utilizar el cómputo extra para controlar otros procesos o simplemente realizar una labor de monitorización y coordinación de actividades con otros nodos.

4.3.3.2 Características principales de los nodos locales

Los nodos locales son una solución económica de disponer de capacidad de computación cerca de cada actuador. Esta característica es obligatoria para reducir el indeterminismo en el tiempo de envío de las acciones de control al proceso. Los nodos locales ejecutan un middleware reducido del núcleo de control. Este middleware reducido se comunica con el middleware completo, que se encuentra en los nodos supervisores, ofreciendo interfaces para la gestión, sensorización, actuación y descarga de páginas de código de reguladores, tal y como se ha descrito en el apartado 4.3.2. Un nodo local puede ser usado como un simple componente esclavo para interactuar en el entorno distribuido de control o puede ejecutar reguladores de forma cíclica.

El objetivo fundamental de los nodos locales es asegurar que siempre exista una acción de control para mandar al proceso. Esta señal de control puede ser una acción de control segura (desconectar, abrir, cerrar, mantenerse sin cambios, etc.), el resultado de un cálculo sencillo mediante un PID localmente implementado en el

nodo, que garantice como mínimo la estabilidad del sistema, o puede ser la señal calculada y enviada desde un nodo supervisor.

Como se ha visto anteriormente, entre las funcionalidades que se incluyen en el núcleo de control cabe destacar el asegurar la entrega de las señales de control y la transferencia de nuevas estructuras de control, proporcionando un conjunto de reguladores que pueden utilizarse en diferentes situaciones. En definitiva, estas funcionalidades tienen como objetivo que al proceso le sigan llegando señales de control en cada periodo de muestreo, y que su salida siga cumpliendo con las especificaciones deseadas, dentro de unos márgenes de degradación tolerables.

A continuación se indican los servicios utilizados en el middleware reducido, implementado en el nodo local, para la consecución del control en el sistema distribuido:

- Mediante el servicio sensorial se realiza la lectura de la salida del sistema y mediante el servicio de actuación se realiza el envío de la acción de control al proceso.
- La acción de control puede ser cualquiera de las opciones que se estudiarán en el siguiente apartado. El componente de gestión de datos y controladores es el encargado de elaborar la acción de control que en última instancia se mandará hacia el proceso en función de la información disponible y el modo de funcionamiento elegido.

4.4 Escenario del problema

La Figura 21 muestra la arquitectura para el sistema de control distribuido. En esta estructura aparecen distintos elementos que utilizan un canal de comunicaciones compartido para el envío/recepción de las distintas señales involucradas. La estrategia de control utilizada en este sistema debe ser capaz de hacer frente a las distintas situaciones que se pueden dar en cada uno de estos elementos, así como adaptar su funcionamiento a los recursos disponibles, para asegurar el cumplimiento de los objetivos de control dentro de un margen de degradación adecuado.

Por este motivo, el objetivo de este apartado consiste en el estudio de las distintas situaciones y modos de funcionamiento que se puedan producir, considerando las particularidades de los elementos involucrados en cada una de ellas.

4.4.1 Parámetros del modelo

Se relacionan a continuación los parámetros que caracterizan los distintos elementos del escenario del problema. En función del valor de estos parámetros y la relación entre ellos se estudiarán las estrategias de control más apropiadas.

4.4.1.1 Señales involucradas en el proceso

Las señales vinculadas a un proceso se pueden dividir en entradas y salidas. El control del proceso consistirá en actuar sobre las entradas manipulables del mismo para que las señales de salida tengan el comportamiento deseado.

A las señales de salida también se las conoce como controladas, ya que describen el comportamiento del proceso y son las que se tratan de controlar (cumplir una serie de restricciones) para satisfacer los objetivos impuestos.

Las entradas manipulables son aquellas sobre las que se puede actuar para influir sobre las variables controladas. Las perturbaciones son entradas que influyen sobre las variables controladas y sobre las que no se puede actuar. Las perturbaciones son una de las causas de la utilización del control en bucle cerrado y de estrategias que permitan disminuir su efecto sobre la salida del proceso.

En la estrategia de control desarrollada se considerarán procesos con una entrada y una salida, es decir, procesos SISO. Y se tendrá en cuenta el efecto de perturbaciones. En cualquier caso, el método es perfectamente adaptable a sistemas MIMO.

4.4.1.2 Periodos de muestreo en los nodos del sistema distribuido

Los algoritmos de control se pueden considerar como sistemas con fuertes requisitos temporales, en el cálculo de las acciones de control no sólo importa el valor de las mismas sino también el instante de tiempo en el que están disponibles.

Cuando los nodos del sistema distribuido se implementan sobre un mismo procesador, o bien se utilizan elementos físicamente separados comunicados mediante un canal de uso exclusivo, la sincronización entre los nodos está determinada y no es un problema que cada uno disponga de un periodo distinto. En los sistemas de control distribuidos en los que los elementos se encuentran físicamente separados, aunque comunicados mediante un enlace compartido, hay que considerar mecanismos de sincronización entre las bases temporales de cada nodo, así como la posibilidad de que los periodos utilizados en cada uno de ellos puedan ser distintos.

Sea T_{CS} el periodo implementado en el control supervisor y T_{CL} el periodo en el control local. Se considerará que los periodos de muestreo son siempre conmensurables entre sí, es decir, que hay una relación de multiplicidad entera entre dichos periodos del nodo supervisor y local. Esto simplificará los métodos de ajuste como se verá posteriormente. Los casos estudiados son los siguientes:

- a) $T_{CS} = T_{CL}$
- b) $T_{CS} > T_{CL}$, con T_{CL} divisor de T_{CS}
- c) $T_{CS} < T_{CL}$, con T_{CL} múltiplo de T_{CS}

Cuando los periodos T_{CS} y T_{CL} no son iguales es necesario realizar una estimación de los valores de cada vector proporcionado por el control supervisor al control local con los periodos de muestro utilizados. Así mismo, cuando el control local proporciona la información de los valores al proceso real el control supervisor también debe estimar los vectores con el periodo de muestro con el que trabaja.

Para realizar la estimación de los valores proporcionados por cada vector al periodo utilizado se emplearán métodos matemáticos para realizar interpolaciones entre los valores proporcionados.

Debido a las restricciones computacionales del nodo local se utilizará el mismo periodo para el control y para la adquisición de datos, por lo tanto, el periodo básico con el que se debe realizar el control del proceso coincide con el periodo del control local T_{CL} .

4.4.1.3 Errores de modelado

El modelo teórico o empírico de un sistema siempre es una aproximación matemática simple del sistema real, y por lo tanto los errores son inherentes al mismo. Los errores de modelado se deben tanto a las deficiencias en la estimación de parámetros, como a la simplicidad del modelo considerado.

Una de las cualidades más importantes que se buscan en el controlador es su robustez. El controlador se diseña basado en el modelo del proceso, y sería deseable que tenga un funcionamiento adecuado cuando se aplique sobre el proceso real, a pesar de incertidumbres paramétricas y dinámica no modelada. Las incertidumbres paramétricas se refieren a las variaciones de los parámetros del modelo del proceso, que ocurren cuando las condiciones de operación cambian, y que son consecuencia de asumir un modelo lineal. La dinámica no modelada es debida generalmente a la dinámica de los actuadores de control.

La estructura del sistema distribuido de control debe hacer frente a los errores de modelado y se deben desarrollar estrategias que permitan minimizarlos.

4.4.1.4 Tiempo de cómputo del controlador

El tiempo de cómputo del controlador es el tiempo de CPU necesario para completar la ejecución de la tarea de control en cada una de las activaciones. Este tiempo depende de la complejidad del algoritmo de control y la velocidad del procesador. El tiempo de cómputo puede variar atendiendo a distintos aspectos: código con ejecución condicional que puede consumir distinto tiempo y el subsistema de ejecución. Esto hace que la ejecución de una activación cualquiera de la tarea de control pueda variar entre dos límites, dependiendo de si se ejecuta la parte condicional. Con el fin de considerar la peor situación posible, en el análisis de la tarea de control se considerará siempre el tiempo máximo, y se supondrá que el tiempo de cómputo de la tarea de control corresponde al peor caso.

El sistema distribuido de control debe hacer frente al consumo excesivo del tiempo de cómputo mediante estrategias que permitan adaptarse a estas situaciones: cambio de controladores, cambio del modo de funcionamiento, delegación de actividades a otros nodos,...

En el trabajo desarrollado en la tesis se supondrá que existen dos tipos de nodos: el nodo local con pocos recursos de computación, y por lo tanto limitado en las operaciones que puede realizar, y el nodo supervisor, que se supondrá sin problemas de tiempo de cómputo. En cualquier caso se estudiará la influencia del tiempo de cómputo dedicado al control en las prestaciones conseguidas por cada nodo.

El objetivo de la tesis será la obtención del tiempo de cómputo mínimo, dedicado a la tarea de control en cada nodo del sistema distribuido, para asegurar el cumplimiento de las especificaciones con un error determinado.

4.4.2 Características de las comunicaciones

4.4.2.1 Retrasos

El medio de comunicación, y especialmente el retardo que pueda producir en la misma, puede influir de manera importante en la eficiencia del sistema desarrollado. Además de los problemas de pérdida de operatividad del sistema derivados del desfase entre acciones del control supervisor y control local, realización de las mismas en el lugar remoto y recepción del resultado por el control supervisor, los retardos en las comunicaciones afectan a la estabilidad del sistema de control ya que sus bucles se cierran a través del canal de comunicaciones (Aracil 2002).

Este problema es grave cuando los retardos superan ampliamente el periodo de muestreo, y cuando se utiliza un canal de comunicaciones con retardos variables.

En situaciones razonables de retardos uniformes y relativamente pequeños este problema se puede solucionar incluyéndolos en el modelo del sistema, y diseñando el control contando con ellos.

En los sistemas con retardos elevados en los que no se pueden obviar, el problema se aborda cerrando el bucle de control en el lugar remoto y en el control supervisor.

Los retrasos que se pueden considerar en el sistema, asociados al medio de comunicación compartido, son debidos a la comunicación entre el nodo local y el nodo supervisor y viceversa. No se consideran los retrasos debidos a los tiempos de cómputo de los controladores así como los debidos a las conversiones A/D, D/A y D/D. Así pues, se definen los siguientes retrasos (Casanova 2005):

- **Retraso en la comunicación entre el nodo local y el nodo supervisor (δ_{ls}).** Incluye el retraso de acceso al medio correspondiente a las peticiones del sensor que captura la información para la realimentación remota y el retraso

de propagación de la información transmitida. En definitiva, determina el tiempo que transcurre desde que el sensor solicita la utilización del enlace compartido hasta que la información transmitida llega al receptor que, en este caso, es el controlador supervisor. Debido a la naturaleza aleatoria del tráfico de información en el enlace, este retraso se considerará variable de forma aleatoria, aunque acotado superiormente. La cota superior de este retraso, que se supone constante y conocida, es la que determina el periodo de muestro T_{CL} y viene impuesta, fundamentalmente, por la política de arbitraje del enlace.

- **Retraso en la comunicación entre el nodo supervisor y local (δ_{SL}).** Igual que el anterior pero para la comunicación desde el controlador supervisor hasta la planta. Incluye el retraso de acceso correspondiente a las peticiones del controlador remoto y el retraso de propagación de la información transmitida por éste. De la misma forma que el anterior, se considera aleatorio y acotado superiormente por el periodo de muestreo T_{CS} .

Para evitar la pérdida de información en el acceso al medio compartido se impone la condición de que el retraso de acceso en el peor de los casos será siempre menor que el periodo de muestreo de la señal discreta generada por el emisor de información:

- $\delta_{LS} < T_{CL}$: Esta condición garantiza que no se perderán muestras capturadas por el sensor para realimentación remota por estar esperando el acceso al enlace compartido. Incluso en el peor de los casos la concesión del enlace llegará antes de que se capture la siguiente muestra a transmitir.
- $\delta_{SL} < T_{CS}$: De la misma forma que el anterior, con esta condición se garantiza que no hay pérdida de acciones generadas por el controlador supervisor mientras se espera la concesión del acceso al enlace compartido. Todas las acciones generadas en el controlador supervisor llegarán, tarde o temprano, al extremo local del enlace compartido.

El retraso de acceso en ambos nodos de comunicación es de carácter aleatorio pero cíclico.

4.4.2.2 Desfases

Debido a que los nodos en los que se estructura el sistema distribuido, supervisor y local, no tienen una comunicación permanente es difícil garantizar una perfecta sincronización entre las bases de tiempo que controlan las tareas que se llevan a cabo en ambos extremos del enlace compartido. Esto significa que los instantes correspondientes a dichas bases de tiempo no tienen necesariamente que suceder a la vez.

En el nodo local están presentes dos bases de tiempo iguales en frecuencia, determinada por el periodo de muestreo básico. La primera de ellas determina la

ocurrencia de los instantes de muestreo en los cuales se captura la información que es realimentada hacia el control supervisor. La segunda está asociada al dispositivo actuador y al resto de elementos de la estructura de control local, en caso de que exista. La única base de tiempos presente en la parte del nodo supervisor determina la generación de acciones en el control supervisor, a partir de la información recibida a través del enlace compartido. El desfase será nulo en el caso ideal de perfecta sincronización. Cuando no se pueda garantizar dicha sincronización entre las bases de tiempo a ambos lados del enlace compartido se modelará considerando un desfase. En principio, dicho desfase será considerado constante. La posibilidad de desfase variable puede modelarse como el muestreo irregular que aparece en el modo de operación por eventos de los dispositivos sensor y actuador. Los dos parámetros que caracterizan el desfase son los siguientes:

- Δ_{LS} : Desfase entre la señal de reloj del sensor para realimentación entre el nodo local y el controlador supervisor. Representa el mínimo tiempo que transcurre entre un instante de muestreo y un instante de generación de la acción de control.
- Δ_{SL} : Desfase entre la señal de reloj del controlador supervisor y la del actuador. Corresponde al mínimo tiempo entre un instante de generación de acciones remotas y un instante de actuación.

En el caso de que exista una estructura de control local a la planta donde todos los elementos están comunicados mediante enlaces exclusivos, la base de tiempos es común y, por lo tanto, no existe desfase. En este caso los instantes de actuación coinciden con los instantes de muestreo en el sensor para la realimentación local y con los instantes de generación de acciones en el controlador local.

En cualquier caso, el desfase nunca será mayor que el periodo de muestreo básico del sistema T . Este periodo es el que se utiliza para el diseño y aplicación del control, y como se ha comentado en el apartado 4.4.1.2 coincide con T_{CL} . Por lo tanto, las condiciones a imponer a estos parámetros, que caracterizan el desfase, serán las siguientes:

- $\Delta_{LS} < T$
- $\Delta_{SL} < T$

4.4.2.3 Ancho de banda de las comunicaciones

Se define el ancho de banda como la medida de datos y recursos de comunicación disponible o consumida expresados en bits/s o sus múltiplos (Kbits/s, Mbits/s, entre otros).

El ancho de banda puede referirse a la capacidad de ancho de banda, o ancho de banda disponible en bits/s, lo cual típicamente significa el rango neto de bits disponible en un sistema de comunicación digital.

El ancho de banda también puede referirse al ancho de banda consumido, que corresponde a la cantidad de información o de datos utilizados en el canal de comunicaciones en un periodo de tiempo dado por parte de las tareas involucradas en el sistema distribuido.

En un sistema distribuido con un canal de comunicaciones compartido es importante evaluar el ancho de banda dedicado a las tareas de control, y su influencia en las prestaciones conseguidas sobre el proceso a controlar.

El objetivo es obtener cuál debe ser el ancho de banda mínimo dedicado a las tareas de control para asegurar el cumplimiento de las especificaciones del control con un error determinado.

4.5 Estrategia de control utilizando el núcleo de control distribuido

El objetivo de este apartado es establecer la estrategia de control a implementar en los nodos del sistema distribuido para dar solución a los problemas expuestos en el apartado anterior.

El núcleo de control deberá contar con mecanismos para hacer frente a las perturbaciones, periodos de los distintos nodos, tiempo de cómputo, retrasos, desfases, ancho de banda dedicado al control, etc., y adaptarse a las distintas situaciones mediante la activación de modos de funcionamiento adecuados, para mejorar de esta manera las prestaciones del sistema.

4.5.1 Tratamiento de distintos periodos entre los nodos

En el apartado 4.4.1.2 se exponen las distintas situaciones que se pueden dar cuando los periodos entre el nodo supervisor y el nodo local no son iguales, aunque conmensurables entre sí. A continuación se va a proceder a detallar la metodología a seguir para el tratamiento de estos casos.

Caso a) $T_{cs} = T_{cl}$

Los periodos de actuación del nodo local y del nodo supervisor son iguales, lo único que se ha de asegurar es la sincronización entre las dos bases de tiempo, o al menos el cálculo del desfase que se produce entre las bases de tiempo.

En un sistema distribuido, las bases de tiempo se encuentran en elementos físicamente separados y comunicados a través de un canal compartido, lo que dificulta garantizar la sincronización.

La sincronización entre los datos es una característica que se debe cumplir en todos los casos, ya que la adaptación de los datos se basa en que las bases de tiempo de los nodos involucrados estén sincronizadas. En el apartado 4.5.2 se trata el tema de la

sincronización, en el que el middleware del núcleo de control es el que se encarga de las tareas de sincronización entre los distintos nodos del sistema distribuido.

Caso b) $T_{CS} > T_{CL}$, con T_{CL} divisor de T_{CS}

En este caso se produce una reducción de la base de tiempos del control local con respecto a la base de tiempos del control supervisor. Esta reducción es un múltiplo entero que se puede expresar como:

$$T_{CS} = m \cdot T_{CL} \quad m \in \mathbb{N} \quad (4.1)$$

Dependiendo del modo de funcionamiento utilizado en la estrategia de control distribuida, que se analizará en el apartado 4.5.6, las operaciones de adaptación de los datos a realizar en cada nodo serán distintas.

En el caso en el que el nodo supervisor envía la acción de control y el nodo local la aplica, éste tan solo debe mantener la acción de control constante hasta la llegada de un nuevo dato. Además, en el momento de la recepción de los datos actualizados es cuando el nodo local envía la salida hacia el nodo supervisor para que éste calcule la nueva acción de control. En este modo de funcionamiento no es necesario ningún tipo de adaptación de datos.

En el caso en el que el nodo supervisor envíe las trayectorias de las acciones de control y de las salidas hacia el nodo local, éste necesita realizar una interpolación de ciertos datos. En las trayectorias futuras de la salida recibidas por el nodo local éste necesita estimar, mediante interpolación, aquellos puntos en los que no se tenga información. En concreto, se cumple que entre dos instantes proporcionados por el control supervisor el control local deberá interpolar $m-1$ muestras. No obstante, en el caso de la acción de control se supondrá que no se ha producido ningún cambio. En la Figura 22 se muestra el efecto de dichas estimaciones para que la trayectoria de salida del controlador local sea la misma que la propuesta por el supervisor.

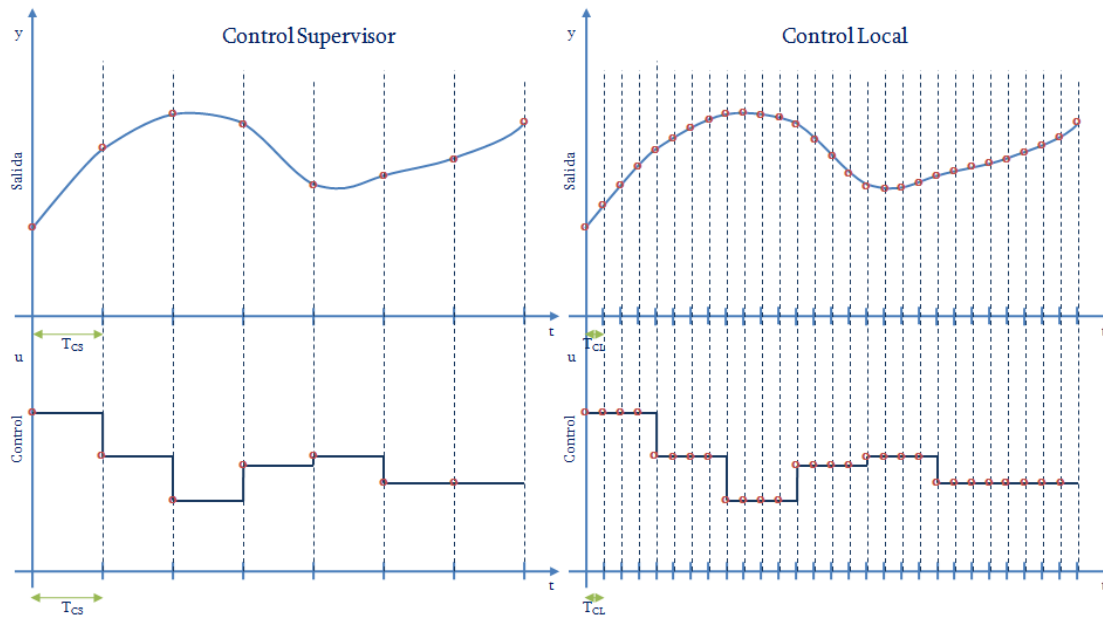


Figura 22. Interpolación de los datos de salida proporcionados por el nodo supervisor

El método utilizado para la estimación de los valores de la salida es el del spline cúbico natural. En el anexo D se ofrece una visión general de dicho método.

Sean (4.2) y (4.3) las secuencias generadas por el control supervisor y que son recibidas por el nodo local:

$$U_{CS} = \{u_{cs}(0), u_{cs}(1), \dots, u_{cs}(\mu_u - 1)\} \quad (4.2)$$

$$Y_{CS} = \{y_{cs}(0), y_{cs}(1), \dots, y_{cs}(\mu_y)\} \quad (4.3)$$

Donde:

U_{CS} : Es la secuencia con la trayectoria de la acción de control generada por el control supervisor con un número de elementos igual a μ_u

Y_{CS} : Es la secuencia con la trayectoria de la salida generada por el control supervisor con un número de elementos igual a $\mu_y + 1$

El nodo local recibirá las secuencias anteriores y deberá aplicar el método expuesto anteriormente, interpolando los datos de la secuencia de salida y manteniendo constante las acciones de control. Como se ha mencionado anteriormente entre cada dos instantes de la salida el control local tendrá que interpolar $m-1$ muestras, así pues, el número de muestras interpoladas, considerando que el número de elementos de Y_{CS} es de $\mu_y + 1$, será igual a $\mu_y \cdot (m - 1)$.

Sean (4.4) y (4.5) las secuencias que adaptará el nodo local a partir de las secuencias recibidas por el nodo supervisor:

$$U_{CL} = \{u_{cl}(0), u_{cl}(1), \dots, u_{cl}(m \cdot \mu_u - 1)\} \quad (4.4)$$

$$Y_{CL} = \{y_{cl}(0), y_{cl}(1), \dots, y_{cl}(m \cdot \mu_y)\} \quad (4.5)$$

Donde:

U_{CL} : Es la secuencia con la trayectoria de la acción de control adaptada por el control local con un número de elementos igual a $m \cdot \mu_u$

Y_{CL} : Es la secuencia con la trayectoria de la salida adaptada por el control local con un número de elementos igual a $m \cdot \mu_y + 1$

El algoritmo para evaluar la señal de salida en el nodo local, con periodo de actuación T_{CL} , con los datos provenientes del nodo supervisor, con periodo de actuación $T_{CS}=m \cdot T_{CL}$, será el siguiente:

Para $i=0$ hasta μ_y

Si i es distinto a μ_y

$$a_i = y_{cs}(i)$$

$$b_i = 0$$

$$c_i = 3 \cdot (y_{cs}(i + 1) - y_{cs}(i))$$

$$d_i = 2 \cdot (y_{cs}(i) - y_{cs}(i + 1))$$

Para $j=0$ hasta $m-1$

$$t = m \cdot j$$

$$y_{cl}(j) = a_i + b_i \cdot t + c_i \cdot t^2 + d_i \cdot t^3$$

fin_para

fin_si

Si i es igual a μ_y

$$y_{cl}(i) = y_{cs}(i)$$

fin_si

fin_para

La acción de control se mantendrá sin cambios cuando el control local no disponga de datos, y por lo tanto no hace falta aplicar ningún tipo de interpolación. En este caso el algoritmo a aplicar en las acciones de control será el siguiente:

Para $i=0$ hasta $m \cdot \mu_u$

$$u_{cl}(i) = u_{cs}(\text{floor}(\frac{i}{m}))$$

fin_para

Para el cálculo de las trayectorias de las acciones de control y de la salida, el controlador supervisor necesitará de la información actualizada de las salidas y de las acciones de control aplicadas, por lo tanto el control local deberá informar al nodo supervisor de estos datos. El control local deberá enviar la información tan sólo en los múltiplos de $m \cdot T_{CL}$, y dando los suficientes datos anteriores para que el nodo supervisor tenga toda la información que le hace falta. El operador que realiza esta función se conoce como *skip*. El operador *skip* realiza una operación de diezmo, dejando pasar sólo una de cada grupo de m muestras consecutivas almacenadas hasta el orden que tenga el proceso, que será la información anterior que necesita el control supervisor para el cálculo de las acciones de control actuales.

Una vez que los datos han sido adaptados el nodo local actualizará sus vectores internos, y para ello se seguirá la metodología expuesta en el apartado 4.5.7.

Ejemplo El siguiente ejemplo muestra la aplicación de la metodología anterior, realizada mediante la herramienta de simulación del sistema distribuido de control desarrollada para este trabajo y que se explica en el anexo A. Se ha tomado un periodo de $T_{CS}=0.2$ segundos en el nodo supervisor y $T_{CL}=0.05$ segundos en el nodo local de. La Figura 23 muestra la adaptación de los datos realizada por el nodo local, en el que se interpolan los datos de la salida y el mantiene la acción de control.

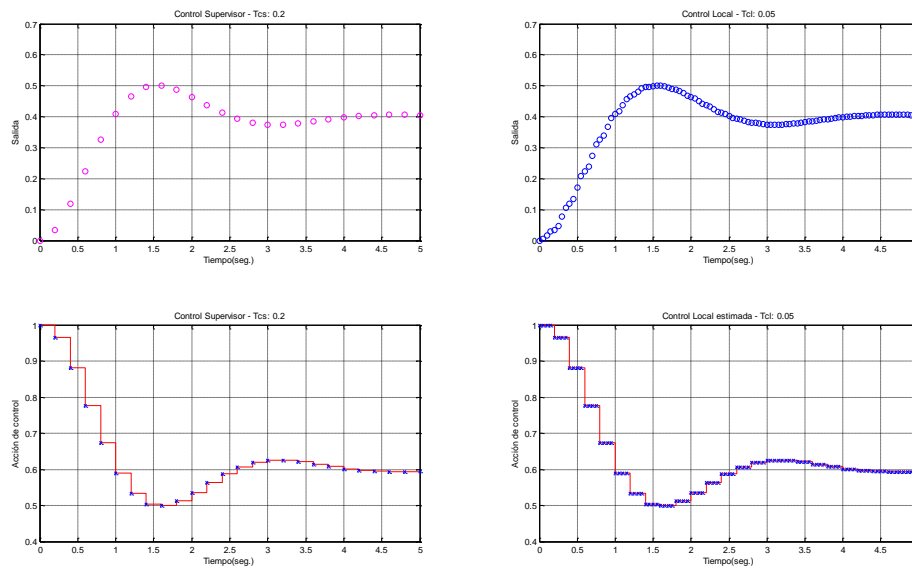


Figura 23. Adaptación de los datos cuando $T_{CS} > T_{CL}$, con T_{CL} divisor de T_{CS}

Caso c) $T_{CS} < T_{CL}$, con T_{CL} múltiplo de T_{CS}

En este caso el periodo de actuación del nodo local es mayor que el periodo del control supervisor. La relación entre ambos periodos es un múltiplo entero que se puede expresar como:

$$T_{CL} = m \cdot T_{CS} \quad m \in \mathbb{N} \quad (4.6)$$

Al igual que ocurre en el caso anterior, dependiendo del modo de funcionamiento utilizado en la estrategia de control distribuida, que se analizará en el apartado 4.5.6, las operaciones de adaptación de los datos a realizar en cada nodo serán distintas.

En el caso en el que el nodo supervisor envía la acción de control y el nodo local la aplica, el nodo supervisor debería adaptar el envío de la acción de control a los instantes múltiplos de m del control supervisor, y por lo tanto debería diseñar el controlador con el periodo de actuación del control local, que debe considerarse como el periodo de muestreo básico y que debe ser compatible con el teorema del muestreo. De esta manera el control supervisor actúa como si tuviese un periodo igual que el local.

En el caso en el que el nodo supervisor envía las trayectorias de las acciones de control y de las salidas hacia el nodo local, éste necesita realizar una interpolación de ciertos datos. En las trayectorias futuras de la salida que recibe el control local, éste tomará tan solo la información de los instantes múltiplos de m que coinciden con su periodo de muestreo, es decir, se produce una operación de diezmado de toda la trayectoria proporcionada por el nodo supervisor. En el caso de la acción de control se debe cumplir que el área de la acción de control proporcionada por el nodo supervisor cada $m \cdot T_{CS}$ segundos sea la misma que realice el control local cada T_{CL} segundos.

El objetivo es que el nodo local entregue al proceso una acción de control que consiga llevar la salida del sistema a la previsión que ha realizado el nodo supervisor a los m instantes. Por lo tanto, la energía entregada por el nodo local deberá ser la misma que la prevista en esos instantes para conseguir el mismo efecto, sin embargo en lugar de entregarla de forma escalonada la entregará en el instante en el que pueda actuar.

La operación de ajuste de la acción de control se puede realizar con una simple media entre las acciones de control proporcionadas por el nodo supervisor dentro del periodo utilizado por el control local.

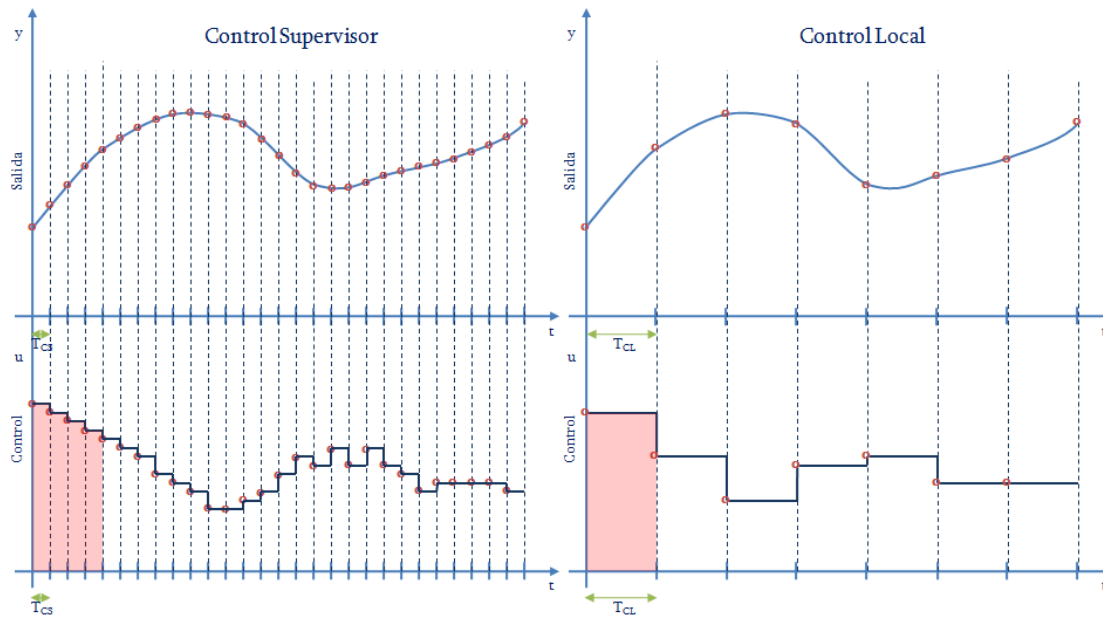


Figura 24. Ajuste de los datos de la acción de control proporcionados por el nodo supervisor

Sean (4.2) y (4.3) las secuencias generadas por el control supervisor y que son recibidas por el nodo local.

El nodo local recibirá las secuencias anteriores y aplica el método expuesto anteriormente, tomando los datos de las salidas cada m instantes, y ajustando la acción de control para que el área de la acción de control proporcionada por el nodo supervisor cada $m \cdot T_{CS}$ segundos sea la misma que realice el control local cada T_{CL} segundos.

Sean (4.7) y (4.8) las secuencias que adaptará el nodo local a partir de las secuencias recibidas por el nodo supervisor:

$$U_{CL} = \{u_{cl}(0), u_{cl}(1), \dots, u_{cl}(\mu_u/m - 1)\} \quad (4.7)$$

$$Y_{CL} = \{y_{cl}(0), y_{cl}(1), \dots, y_{cl}(\mu_y/m)\} \quad (4.8)$$

Donde:

U_{CL} : Es la secuencia con la trayectoria de la acción de control adaptada por el control local con un número de elementos igual a $\frac{\mu_u}{m}$

Y_{CL} : Es la secuencia con la trayectoria de la salida adaptada por el control local con un número de elementos igual a $\frac{\mu_y}{m} + 1$

El algoritmo para evaluar las señales de salida y de control en el nodo local, con periodo de actuación T_{CL} , con los datos provenientes del nodo supervisor, con periodo de actuación T_{CS} , en los que se cumple $T_{CL}=m \cdot T_{CS}$ será el siguiente:

```

Para  $i=0$  hasta  $\mu_y/m$ 
     $y_{cl}(i) = y_{CS}(i \cdot m)$ 
fin_para

Para  $i=0$  hasta  $(\mu_u/m - 1)$ 
     $u_{cl}(i) = \frac{\sum_{k=i \cdot m}^{m \cdot (i+1) - 1} u_{CS}(k)}{m}$ 
fin_para
    
```

Una vez que los datos han sido adaptados el nodo local actualizará los vectores internos con la información más actualizada, y para ello se seguirá la metodología expuesta en el apartado 4.5.7.

Ejemplo El siguiente ejemplo muestra la aplicación de la metodología anterior, realizada mediante la herramienta de simulación del sistema distribuido de control desarrollada que se explica en el anexo A. Para este ejemplo se ha tomado un periodo en el nodo supervisor de $T_{CS}=0.05$ segundos y un periodo en el nodo local de $T_{CL}=0.2$ segundos. La Figura 25 muestra la adaptación de los datos que realiza el nodo local, en el que se ha aplicado una interpolación en los datos de la salida y el mantenimiento de la acción de control.

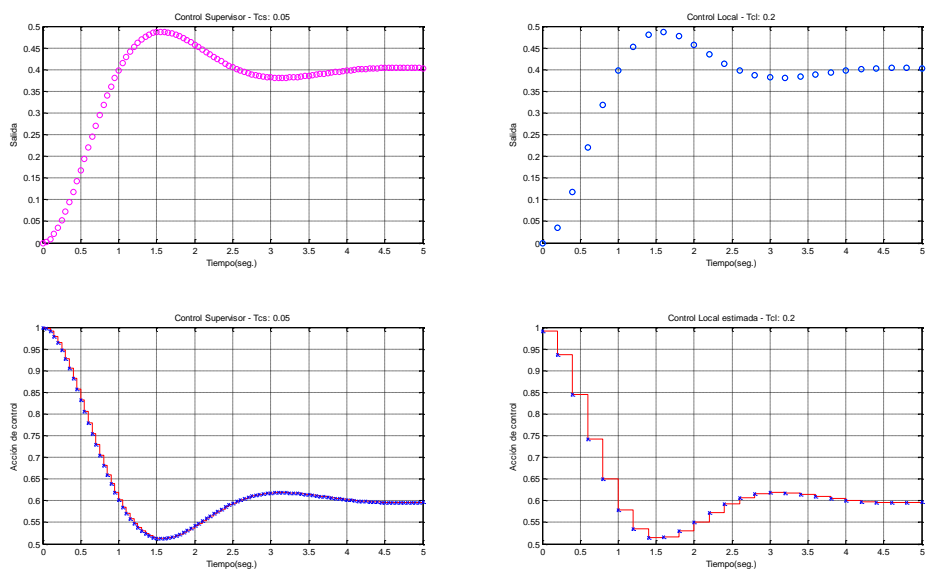


Figura 25. Adaptación de los datos cuando $T_{CS} < T_{CL}$, con T_{CL} múltiplo de T_{CS}



4.5.2 Sincronización de los nodos del sistema distribuido

Cuando se dispone de un sistema distribuido en el que los nodos se encuentran separados físicamente, y en el que las bases de tiempo pueden ser distintas, tal y como se ha estudiado en el apartado 4.5.1, es necesario un mecanismo que permita la sincronización de dichas bases de tiempo.

El núcleo de control dispone de mecanismos para sincronizar los relojes de los nodos dentro del sistema distribuido. Estos mecanismos permiten determinar el tiempo que ha pasado desde la generación de un mensaje hasta que éste llega al destino.

De esta manera es el núcleo de control, mediante los servicios middleware, el que se encarga de la sincronización de los nodos, y determina en cualquier caso si los mensajes recibidos por cada nodo son válidos temporalmente. En caso de retraso en la recepción o pérdidas de datos, el middleware informa a la aplicación de control para que ésta realice las tareas que se consideren oportunas. El objetivo de la tesis será la realización de la estrategia de control distribuida para hacer frente a estos problemas, tal y como se mostrará en el apartado 4.5.6.

4.5.3 Estructura de control del nodo supervisor

El nodo supervisor es el encargado de hacer llegar la información necesaria a los nodos locales para que se cumplan las tareas de control del sistema distribuido. Además, es el que a partir de la información suministrada por cada nodo decide la estrategia a seguir, y de esta manera coordina el comportamiento de los distintos nodos del sistema distribuido para que el comportamiento global del sistema sea el deseado.

A continuación se analizan las distintas estructuras de control que se pueden implementar en el nodo supervisor.

4.5.3.1 Sin control supervisor

El nodo supervisor en este caso no ejecuta ningún algoritmo de control sino que simplemente se encarga de suministrar las referencias a los nodos locales, que son en los que delega la tarea de control para hacer cumplir las especificaciones.

En cualquier caso este nodo supervisor debería recibir la salida del proceso, para verificar el buen funcionamiento de sistema y para poder coordinar el envío de las próximas referencias. Este apartado es importante en el caso de que el proceso deba sincronizarse con otros, y por lo tanto es el supervisor el que realiza esta tarea de coordinación.

4.5.3.2 Con control supervisor

En este caso en el nodo supervisor se ejecuta un algoritmo de control que calcula la acción de control a enviar a cada nodo local. El nodo local por lo tanto tendría un funcionamiento de un actuador.

En caso de considerar posibles pérdidas de datos en el envío de las acciones de control hacia el nodo local se debería ejecutar en un algoritmo de control sencillo o una acción de control segura, tal y como se explica en el apartado 4.5.4.2

4.5.3.3 Con control supervisor con cálculo de trayectorias postuladas de la acción de control y de la salida

Con esta alternativa el control supervisor calcularía la acción de control a aplicar al proceso junto con la trayectoria prevista, dentro de un horizonte de control, por ejemplo mediante un control predictivo. Esta trayectoria se enviaría al nodo local, de forma que en caso de pérdida de datos el nodo local podría utilizar las acciones de control postuladas que fueron recibidas en algún instante anterior.

Si el nodo local utiliza la acción de control postulada directamente se estaría utilizando una estrategia de control en bucle abierto, por lo que una mejora de esta estrategia de control sería la de enviar también la trayectoria de la salida prevista hacia el nodo local, de forma que esta trayectoria sería utilizada como la trayectoria nominal que debería seguir la salida del sistema si se utilizara las acciones de control postuladas. En caso de no cumplirse dichas salidas siempre se pueden hacer ajustes en la acción de control para intentar compensarla.

Este método de compensación de la acción de control por parte del nodo local se desarrollará en el apartado 4.5.5, siendo una de las contribuciones del presente trabajo.

4.5.4 Estructura de control del nodo local

Dependiendo del modo de funcionamiento utilizado en el sistema distribuido, tal y como se estudiará en el apartado 4.5.6, las actividades asignadas a las tareas de control en el nodo local pueden cambiar. Teniendo en cuenta que se considera a los nodos locales como sistemas empotrados con recursos limitados en computación, aunque con amplias capacidades en las comunicaciones e interacción con el resto del sistema distribuido, estas tareas de control deben ser compatibles con dichas características, y por lo tanto no pueden llevarse a cabo ni la implementación de controladores de alto nivel, ni cálculos elaborados en la obtención de la acción de control a aplicar en cada instante. Esto hace que sea necesaria la utilización de un control supervisor que complete las tareas y se encargue de la sincronización de los distintos nodos del sistema distribuido.

A continuación se presentan las distintas estructuras de control que pueden ser implementadas en el nodo local, que dependen de la estrategia utilizada en el sistema distribuido.

4.5.4.1 Sin control local

En este caso el nodo local se comporta como un simple accionador, cuya misión principal es la aplicación de la acción de control proveniente del nodo supervisor sobre el proceso.

Otra funcionalidad que puede ser implementada en este nodo local sin control es, en caso de no recibir la acción de control desde el nodo supervisor, la aplicación de una acción de control segura. La acción de control segura puede cambiar dependiendo del proceso que se esté controlando. En algunos casos puede ser desconectar, abrir, cerrar, mantener la acción de control sin cambios, etc. En ningún caso el nodo local realiza ningún tipo de cálculo de la acción de control, si no que utiliza la información previamente almacenada según las características del proceso bajo control.

4.5.4.2 Con control local

Otro método para conseguir mantener las prestaciones del control local en caso de pérdida de datos, o de carecer de un nodo supervisor que calcule un control de alto nivel, es la utilización de un control que utilice pocos recursos computacionales y de memoria, y que pueda ser implementado con las características disponibles en el nodo local. Los reguladores PID son los más adecuados para cumplir con las prestaciones dinámicas y estáticas en el seguimiento de la referencia. Además, como se están utilizando controles digitales siempre se puede implementar un regulador calculado mediante un método de diseño algebraico: reguladores discretos de asignación de polos o tiempo finito.

Debido a que este control local se implementa para mejorar las prestaciones cuando se tienen problemas de recepción de la acción de control proporcionada por el nodo supervisor, se debe permitir la conmutación entre un control proporcionado por el supervisor, con lo que se estaría actuando como en el caso anterior, o bien el control directo mediante el regulador sencillo implementado localmente.

En este caso el control local también puede proporcionar una acción de control segura, e igual que ocurre en el caso anterior dependiendo del proceso que se esté controlando puede cambiar, aunque también puede ser un caso particular del PID anterior. Se utilizará cuando se desee llevar a la salida del proceso a una situación segura estable, por lo tanto, en este caso el control local es el que propone la referencia y el controlador PID a utilizar para conseguir el objetivo.

4.5.4.3 Con control local con compensación de la acción de control proveniente del nodo supervisor

Una manera mediante la cual el nodo local aplique una acción de control de alto nivel, consecuencia de una minimización de una función de coste, un cálculo iterativo, programación lineal, etc., sería que el control supervisor se encargue de realizar los cálculos y los envíe hacia el control local para que éste los aplique. Con este sistema la estructura del control local sería similar a la explicada en el apartado 4.5.4.1. El problema está cuando se producen pérdidas en la recepción de estas acciones de control consecuencia del enlace compartido entre el nodo supervisor y el nodo local. Ya se ha comentado que una posible solución sería aplicar una acción de control segura.

Otra solución a la posible falta de datos sería que el nodo local dispusiera de información postulada de la acción de control, calculada por el nodo supervisor en algún instante previo, de manera que en caso necesario el nodo local podría utilizar dichas acciones sobre el proceso.

Sea Y el vector con las salidas futuras, obtenidas como consecuencia de aplicar al modelo del proceso las acciones de control postuladas U , calculadas en el nodo supervisor mediante un control de alto nivel, por ejemplo un predictivo:

$$U = \{u(k|k), u(k+1|k), \dots, u(k+N_u-1|k)\} \quad (4.9)$$

$$Y = \{y(k), \hat{y}(k+1|k), \hat{y}(k+2|k), \dots, \hat{y}(k+N|k)\} \quad (4.10)$$

El nodo supervisor implementa un controlador de alto nivel que gestiona el cálculo de las secuencias anteriores, que son enviadas en cada iteración al nodo local. El control de alto nivel puede ser un controlador predictivo basado en modelo (MBPC) o bien algún tipo de regulador que proponga las acciones de control futuras que deben aplicarse a un proceso para que la salida del sistema cumpla con la especificación deseada.

Por lo tanto U , definido mediante la secuencia (4.9), es un vector con la acción de control que hay que aplicar en el instante actual junto con las acciones de control, dentro del horizonte de predicción, que habría que aplicar para que la salida del sistema fuera Y , definida mediante la secuencia (4.10).

El funcionamiento del control del nodo local consiste en aplicar la acción de control $u(k|k)$, siempre que no haya pérdida de datos, ya que es la que el nodo supervisor ha calculado con la información actualizada de acciones de control y salidas anteriores. Para la siguiente iteración, el control supervisor dispone de nueva información, y por lo tanto, aplicará la técnica del horizonte deslizante, es decir, se desechan las acciones de control futuras y se vuelve a calcular el vector con la información

actualizada de la salida real que se ha producido en el proceso, enviando las nuevas secuencias hacia el nodo local.

En caso de pérdida de datos desde el control supervisor hacia el control local, éste dispone de las acciones de control postuladas que fueron calculadas en algún instante anterior. Esta secuencia de acciones de control es óptima, ya que en el caso de los controladores MBPC se obtiene como consecuencia de la minimización de una función de coste. En el caso de un sistema con escasos recursos de computación, como un sistema empotrado, esta información es muy valiosa, ya que permite al control local la utilización de las acciones de control postuladas para llevar a la salida del sistema a las especificaciones de diseño.

La Figura 26, Figura 27 y Figura 28 muestran la salida y la acción de control postulada en cada instante (azul claro) comparada con la salida que tendría el sistema si se aplicara la estrategia del horizonte móvil (magenta) y la acción de control (rojo).

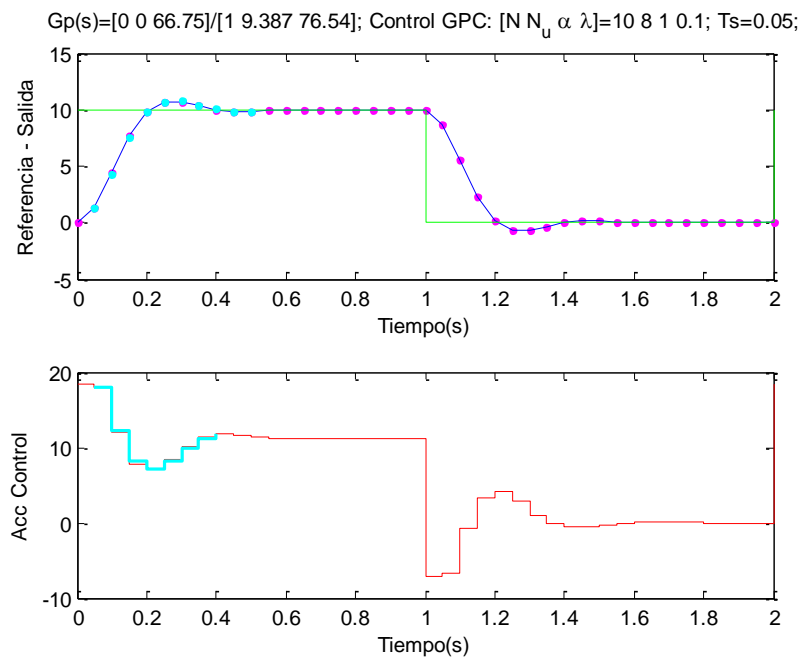


Figura 26. Salida y acción de control postuladas en el instante $k=1$

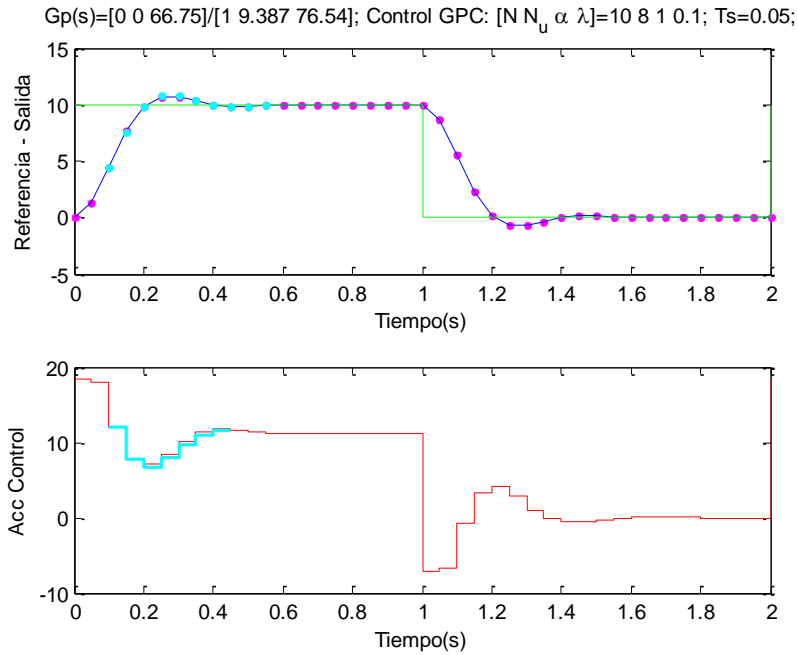


Figura 27. Salida y acción de control postuladas en el instante $k=2$

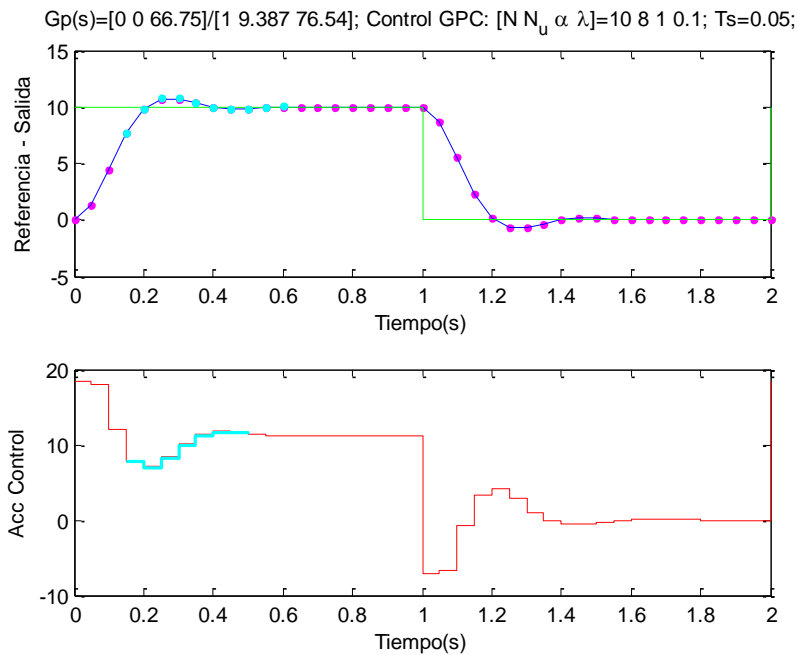


Figura 28. Salida y acción de control postuladas en el instante $k=3$

Si se aplicara directamente las acciones de control postuladas el control local estaría actuando como un sistema en bucle abierto, situación que no es deseable, ya que el control local dispone de la información de la salida del proceso para verificar si el comportamiento que sigue es el esperado. Por lo tanto, lo que se hace es una compensación de la acción de control postulada mediante el error que se comete entre la salida que debería seguir el proceso, a partir de la información proporcionada por el control supervisor, y la salida que realmente está siguiendo,

mediante la medición de la salida real. Esta técnica, que se explica en el apartado 4.5.5, permite el ajuste de la acción de control postulada para tratar de minimizar el ruido del proceso real.

Al recuperar las comunicaciones entre el control supervisor y el control local éste debe informar de las acciones de control aplicadas y de las salidas reales producidas en los instantes anteriores, para que de esta forma el control supervisor pueda calcular de manera eficiente los nuevos vectores de salidas y acciones de control futuras. El número de elementos necesarios a incluir en las acciones de control y salidas anteriores coincide con el orden del modelo del proceso (n), ya que son los que utiliza el regulador MPC que maneja el control supervisor.

$$\{U_{anterior}\} = \{u(k-1), \dots, u(k-n)\} \quad (4.11)$$

$$\{Y_{anterior}\} = \{y(k-1), \dots, y(k-n)\} \quad (4.12)$$

Este método permite en caso de pérdida masiva de datos entre el control supervisor y el control local utilizar la última acción de control como una acción constante, ya que los horizontes de predicción se calculan de forma que engloben a toda la dinámica del proceso. En el apartado 4.6.1 se detalla la forma de calcular el horizonte de predicción y el de control para asegurar que la última salida postulada se encuentra en la referencia.

4.5.5 Tratamiento de los errores de modelado y el ruido. Compensación de la acción de control

Como se ha visto anteriormente, la utilización directamente de las acciones de control postuladas por parte del nodo local en caso de pérdida de datos, tiene el problema de utilizar una estrategia en bucle abierto, con los consabidos problemas en cuanto a estabilidad debidos a los posibles errores cometidos en el modelado y a la existencia de los inevitables ruidos en la medida.

En caso de que no existan ruidos ni perturbaciones en el sistema y ningún error de modelado, se podría utilizar la estrategia en bucle abierto sin ningún problema, ya que no habría ninguna discrepancia en las salidas calculadas por el modelo y las salidas reales.

Para minimizar estos problemas se propone que el control local, en caso de utilizar las acciones de control postuladas por el control predictivo, realice modificaciones en la acción de control propuesta, teniendo en cuenta la discrepancia que existe entre la trayectoria de la salida precalculada y la que verdaderamente se aplica.

El error en la predicción de la salida se calcula como:

$$ey = \hat{y}(k + i|k) - y_{real}(k + i), \quad i = 1, \dots, N \quad (4.13)$$

La estrategia utilizada en el control local a partir del error en la salida y de la información disponible será:

<p><i>If $i < N_u - 1$, then</i></p> $u = u(k + i k) + K_{gain} \cdot ey$ <p><i>If $(i \geq N_u - 1) \& (i < N)$, then</i></p> $u = u(k + N_u - 1 k) + K_{gain} \cdot ey$ <p><i>If $(i \geq N)$, then</i></p> $u = u(k + N_u - 1 k) + K_{gain} \cdot (\hat{y}(k + N k) - y_{real}(k + i))$	(4.14)
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------

donde:

- $\hat{y}(k + i|k)$ es la salida postulada por el control predictivo para el instante de tiempo $k+i$, calculada con la información hasta el instante k
- $y_{real}(k + i)$ es la salida real medida en el proceso real en el instante $k+i$
- K_{gain} es la ganancia del compensador local que se debe determinar de forma experimental, estudiando la influencia de los errores cometidos entre la trayectoria y la salida en la acción de control final

En el último caso, en el que se ha utilizado toda la información proveniente del nodo supervisor consumiendo todos los datos se puede aplicar una acción de control segura:

$$u = u_{segura} \quad (4.15)$$

En esta acción de control segura, dependiendo del proceso a controlar, podría ser una de estas alternativas:

- Mantenimiento de la última acción de control con comportamiento de la salida estable. Una variación de esta alternativa, en que se utiliza la compensación de la acción de control, es la que se ha indicado en (4.14)
- Aplicación de un regulador local sencillo
- Secuencia de apagado

A continuación se muestra la estrategia propuesta:

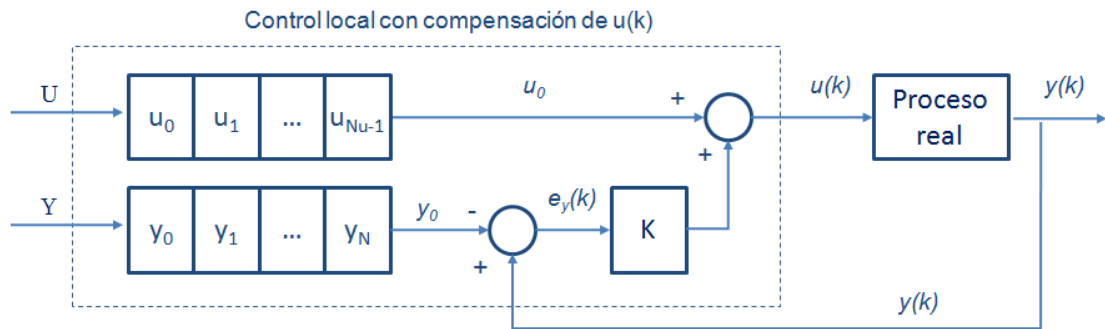


Figura 29. Nodo local con compensación de la acción de control

El cómputo dedicado por cada control local es despreciable, ya que lo único que debe hacer es aplicar la acción de control que le llega a través del canal de comunicaciones, y en caso de pérdida de datos aplicar la acción de control postulada por el control predictivo MBPC compensándola mediante el error que se producen entre la trayectoria prevista para ese instante y la salida real.

La ventaja de este tipo de sistemas es que el control supervisor puede realizar la supervisión de varios controles locales, y por consiguiente, si estos controles forman parte del mismo sistema, este control supervisor actuaría como un control global en el que tiene en cuenta lo que le está ocurriendo a todos los subsistemas dependientes.

La estrategia de la compensación de la acción de control es la más sencilla que se puede aplicar, pero también se podría utilizar en el nodo local un regulador PID o bien un control de mínima varianza. En ambos casos los reguladores son muy dependientes del proceso a controlar, y de la magnitud de la perturbación, por lo que habría que calcular los parámetros de dichos reguladores para cada situación. La ventaja de la compensación de la acción de control utilizando una ganancia es que sólo hay que cambiar un parámetro.

4.5.5.1 Propiedades de la compensación de la acción de control

Propiedad 1. La única información que necesita el nodo local cuando aplica la técnica de la compensación de la acción de control es la ganancia, ya que los demás parámetros son independientes del proceso a controlar.

Propiedad 2. La compensación de la acción de control se realiza para anular el efecto del ruido de la salida sobre el proceso y reducir el efecto de las perturbaciones y errores de modelado. La ganancia a aplicar debe ser relativamente pequeña, ya que lo que se busca es ajustar la acción de control predicha a las condiciones reales.

Propiedad 3. La compensación de la acción de control puede utilizarse tanto en los casos esporádicos de pérdida de datos o tiempo excesivo de espera, como en la utilización del envío de las trayectorias de la acción de control y de la salida sólo en

los cambios de referencia. En este último caso la compensación de la acción de control se utilizará en el nodo local hasta el envío de las siguientes trayectorias por parte del nodo supervisor.

Propiedad 4. En procesos no estables la compensación de la acción de control actúa como un lazo de control en bucle cerrado cuando se pierde la conexión con el nodo supervisor. Las pruebas, tanto simuladas como reales, sugieren que, en los sistemas inestables, se puede ganar tiempo antes de llegar a una región de recuperación imposible, por lo que al utilizar esta técnica se pueden asumir retrasos o pérdidas de datos mayores que si no se hiciese así.

4.5.6 Modos de funcionamiento

El objetivo del nodo local, como se ha indicado anteriormente, es reducir el indeterminismo en el tiempo de envío de las acciones de control al proceso, realizando la entrega en el instante adecuado. Debido a las limitaciones en el tiempo de cómputo inherentes a este nodo se deben realizar los mínimos cálculos posibles para que se asegure el cumplimiento de este objetivo.

De esta forma, se propone la implementación de varios modos de funcionamiento, que permitan al sistema adaptarse a las características del mismo y cumplan las especificaciones del control con una degradación en las prestaciones del sistema adecuada. Además, cuando el canal de comunicaciones utilizado entre los elementos del sistema distribuido no es de uso exclusivo se producen una serie de problemas que el algoritmo de control deberá tener en cuenta, como pueden ser: la sincronización entre los nodos, el tratamiento de los retardos y desfases, y el tratamiento de distintos periodos de actuación entre los nodos.

La estrategia de control desarrollada es válida para procesos con una entrada y una salida (procesos SISO), y en los que exista un nodo supervisor y un nodo local, aunque las estrategias de control presentadas se pueden ampliar a un sistema con un nodo supervisor y varios nodos locales con relativa facilidad.

A continuación se detallan en qué consisten cada uno de estos modos de funcionamiento, en el que se hace un estudio desde los más sencillos (modos 1 y 2), hasta los más sofisticados (modos 3, 4 y 5) en los que se utiliza el núcleo de control distribuido, que es una aportación del presente trabajo.

4.5.6.1 Modo 1. Control jerarquizado

El nodo supervisor envía las referencias y el nodo local las sigue mediante un regulador sencillo. Este modo de funcionamiento es similar a un control jerarquizado con un canal de comunicaciones compartido.

En este caso la labor del nodo supervisor es la de coordinación del comportamiento de la salida del proceso, indicando al nodo local el momento en el que debe cambiar

las consignas. Este tipo de modo de funcionamiento es ampliamente utilizado en los sistemas formados por múltiples subprocesos que necesitan una coordinación de sus salidas para que cambien en el instante adecuado, como es el caso de un robot humanoide, formado por múltiples servomecanismos, en los que hay que coordinar la posición de los mismos para que la postura tomada por el robot sea la adecuada en cada momento.

En este modo de funcionamiento el nodo local tendría un control local sencillo, mientras que el nodo supervisor no tendría ningún tipo de control. En el nodo local se realizarían las tareas de lectura de la variable bajo control y el envío hacia el control supervisor, ya que, tal y como se ha explicado anteriormente, el nodo supervisor debe ser informado de la situación de la variable bajo control.

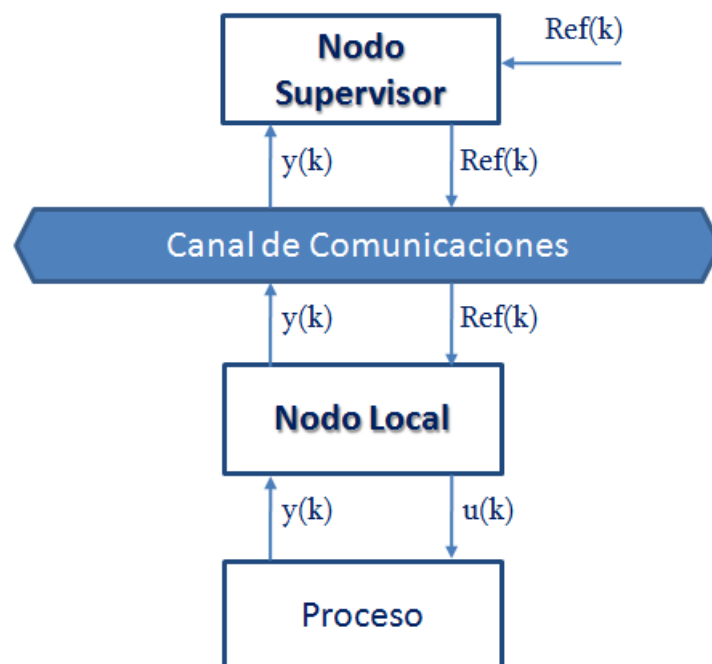


Figura 30. Modo 1 de funcionamiento

- **Ventajas:** En caso de pérdida de datos el control local siempre puede utilizar la última referencia para mantenerse en ella, o proponer la referencia que le lleve a una situación segura.
- **Inconvenientes:** Debido a la utilización de un nodo local con pocos recursos computacionales el regulador local debe ser muy sencillo. Por lo tanto no se pueden realizar cálculos de controladores de alto nivel. El control del nodo local se debe diseñar específicamente para el proceso a controlar.

4.5.6.2 Modo 2. Control basado en red

El nodo supervisor envía las acciones de control al nodo local y éste las aplica, trabajando como un actuador. Este modo de funcionamiento es similar al control

digital directo con un canal de comunicaciones compartido, es lo que se denomina un sistema de control basado en red. En la tesis de (Casanova 2005) se encuentra información acerca de este tipo sistemas en los que se utiliza un canal de comunicaciones compartido.

En este modo de funcionamiento el nodo local carece de un sistema de control, ya que es el nodo supervisor el que lo implementa. En el nodo local se realizarían las tareas de lectura de la variable bajo control y el envío hacia el control supervisor.

El nodo local debe enviar, junto con la salida actual, las salidas y acciones de control aplicadas anteriores hasta un instante n , para que, en caso de problemas de comunicación entre el nodo local y el supervisor, éste tenga la información necesaria para el cálculo de la nueva acción de control al recuperar las comunicaciones. Las secuencias a enviar por el nodo local serán (4.11) y (4.12).

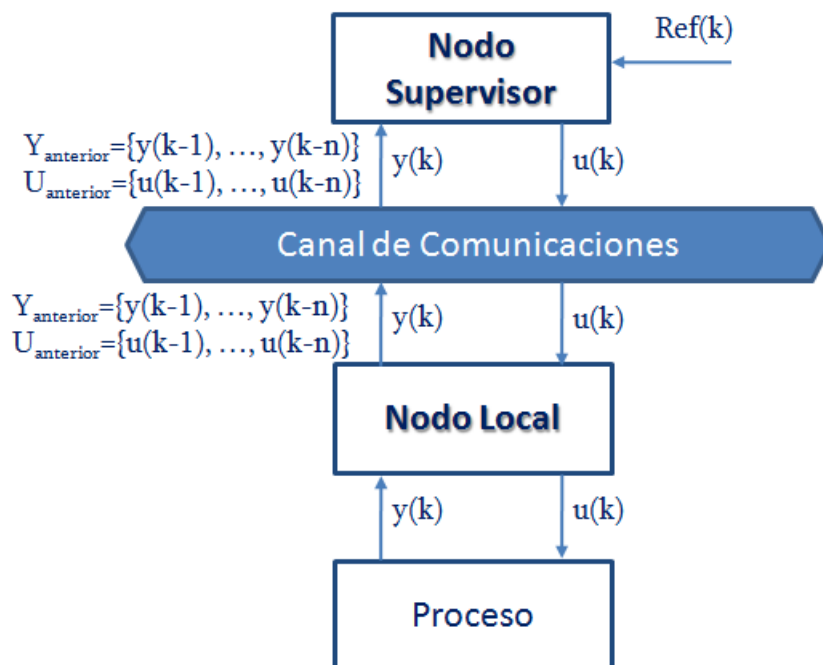


Figura 31. Modo 2 de funcionamiento

- **Ventajas:** El nodo local no necesita realizar ningún tipo de cálculo adicional, tan sólo tiene que aplicar la acción de control y comunicar su estado.
- **Inconvenientes:** En caso de pérdida de datos o de problemas en el envío de la acción de control por parte del nodo supervisor, el nodo local no recibiría la información actualizada, y por lo tanto debería aplicar la acción de control anterior (o incluso realizar una secuencia de seguridad con paro).

Para mejorar las prestaciones se puede añadir un control local para que en caso de pérdida en la comunicación entre los nodos el proceso siga estando controlado.

4.5.6.3 Modo 3. Envío de las acciones de control postuladas

En el nodo supervisor se ejecuta un control de alto nivel, como por ejemplo un regulador predictivo, capaz de calcular las trayectorias de las acciones de control postuladas. El nodo supervisor envía las trayectorias de las acciones de control hacia el control local en cada instante. En este modo, en caso de pérdida de datos el nodo local podría utilizar las acciones de control postuladas, aunque se estaría realizando un control en bucle abierto.

En el nodo local se debe implementar un algoritmo de actualización de la información recibida, para enviar hacia el proceso la acción de control apropiada en cada situación.

En detalle, este algoritmo de control consistiría en realizar las siguientes tareas:

- El nodo supervisor calcula mediante un controlador predictivo la trayectoria de las acciones de control, dentro de un horizonte de control N_u , que se deben aplicar a un proceso para que siga una referencia, esta trayectoria es (4.9).

El primer elemento de (4.9) es la acción de control que se debe aplicar en el instante k con la información hasta ese instante k , mientras que los demás elementos son las acciones de control postuladas dentro del horizonte de control.

- El nodo local recibe esta trayectoria, y la almacena internamente en un vector con tantos elementos como el horizonte de control calculado N_u . En la
- Tabla 1 se muestra la información que se almacena en cada elemento del vector.

Etiqueta	u_0	u_1	...	u_{N_u-2}	u_{N_u-1}
Elemento almacenado	$u(k k)$	$u(k+1 k)$...	$u(k+N_u-2 k)$	$u(k+N_u-1 k)$

Tabla 1. Vector con la trayectoria de las acciones de control que almacena el nodo local en el modo 3 de funcionamiento

La acción que se aplica sería la correspondiente al elemento u_0 , que coincide con la acción de control calculada con los datos hasta ese instante $u(k|k)$

- En el siguiente instante el nodo supervisor volvería a calcular la nueva trayectoria, con la información actualizada de la salida, y se la enviaría hacia el nodo local nuevamente, desechando las acciones de control postuladas, esto es lo que se conoce como la estrategia del horizonte móvil.

- En caso de que no haya pérdida de datos el nodo local actualizaría el vector interno de la trayectoria de las acciones de control con los nuevos datos, aplicando la acción de control correspondiente al primer elemento u_0 .
- Sin embargo, en caso de que haya pérdida de datos y el nodo local no recibiese la trayectoria actualizada de la acción de control, lo que haría es utilizar la información postulada almacenada internamente en un vector. La información de dicho vector se actualizaría de la siguiente manera:

$$\begin{aligned}
 u_0 &= u_1 \\
 u_1 &= u_2 \\
 &\dots \\
 u_{Nu-2} &= u_{Nu-1}
 \end{aligned}
 \tag{4.16}$$

Para aquellas operaciones en las que se utilicen las acciones de control postuladas es interesante saber la antigüedad del vector de datos que tiene almacenado el nodo local, para ello se utiliza el entero δ_v que indica los instantes de tiempo que se encuentra el vector sin ser actualizado. Este entero es necesario, como se verá en el apartado 4.5.7, cuando tratamos con sistemas con retardos y desfases para comprobar si los datos que llegan son más actuales que los que se encuentran almacenados en el vector.

El valor de δ_v se pondrá a 0 cada vez que se actualice completamente el vector de datos interno, y se incrementará en una unidad cada vez que se van utilizando los valores postulados. En la primera iteración, cuando se generan los vectores internos, el entero δ_v se debe inicializar a 1, indicando que la información que contienen los vectores todavía no está actualizada.

Por lo tanto a las operaciones descritas en la ecuación (4.16) habría que añadir la actualización del entero que memoriza la antigüedad del vector:

$$\delta_v = \delta_v + 1 \quad \delta_v \in \mathbb{N}
 \tag{4.17}$$

El resultado de la actualización del vector interno del nodo local para la pérdida de un dato se puede ver en la Tabla 2. Nuevamente la acción que se aplica sería la correspondiente al elemento u_0 , que en este caso coincidiría con la acción de control postulada en algún instante anterior. En este caso el valor de δ_v sería igual a 1.

Etiqueta	u_0	u_1	...	u_{N_u-2}	u_{N_u-1}
Elemento almacenado	$u(k+1 k)$	$u(k+2 k)$...	$u(k+N_u-1 k)$	$u(k+N_u-1 k)$

Etiqueta	u_0	u_1	...	u_{N_u-2}	u_{N_u-1}
Elemento almacenado	$u(k+1 k)$	$u(k+2 k)$...	$u(k+N_u-1 k)$	$u(k+N_u-1 k)$

Tabla 2. Actualización del vector con la trayectoria de las acciones de control que almacena el nodo local cuando se pierde un dato

- Al recuperar las comunicaciones se volvería a desechar nuevamente la información almacenada, actualizándola con la nueva trayectoria recibida, que corresponde a la trayectoria de la acción de control calculada con la información actualizada.
- Este modo, igual que el anterior, supone que el nodo local debe enviar la acción de control aplicada, junto con la salida leída hacia el nodo supervisor para que éste calcule la acción de control con toda la información actualizada proveniente del proceso, es decir, debería enviar las secuencias con las salidas y las acciones de control anteriores (4.11) y (4.12).

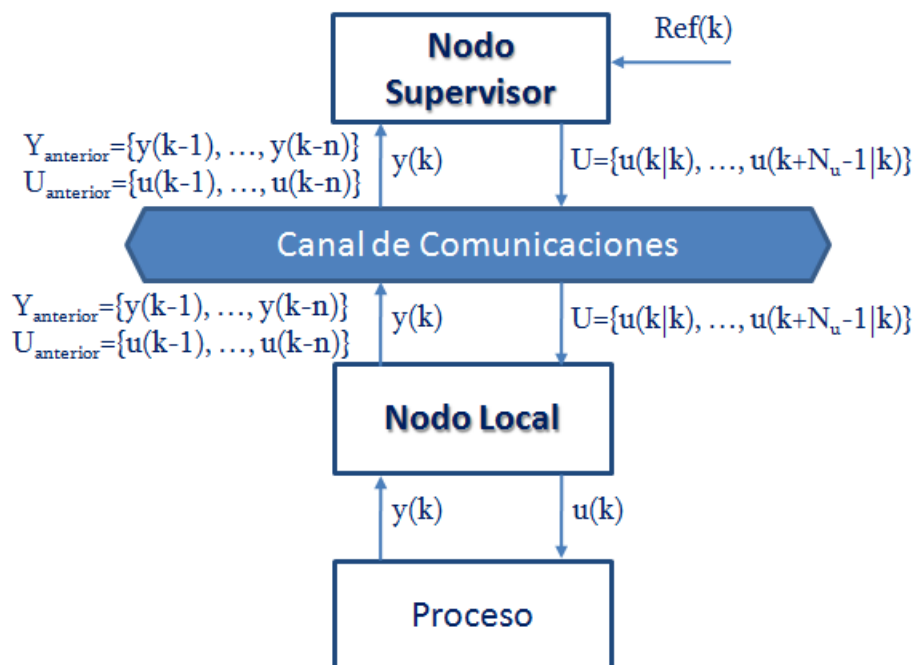


Figura 32. Modo 3 de funcionamiento

- Ventajas: En caso de pérdida de datos el nodo local puede utilizar las acciones de control postuladas.
- Inconvenientes: La utilización de las acciones de control postuladas hace que el nodo local esté realizando un control en bucle abierto, con todos los inconvenientes que ello implica. Además, al mandar más datos por el canal de comunicaciones se pueden producir más pérdidas y retrasos en los envíos y recepción de los mismos. Siempre que la información enviada utilice el tamaño mínimo de la trama esto no será un problema.

Este modo de funcionamiento es similar al control de red basado en paquetes, en el que se aprovecha el campo de datos de los paquetes para el envío de información, sin incurrir en costes extras de sobrecarga de las comunicaciones. Se basa en el envío de paquetes con las acciones de control futuras (*packed-based – PBNCS- o packetized networked control systems –PNCS-*). Esta aproximación se apoya en técnicas e ideas del ámbito del control predictivo (Dormido, Sánchez y Kofman 2008).

4.5.6.4 Modo 4. Envío de las acciones de control postuladas y de la trayectoria de salida nominal

En este caso el nodo supervisor envía las trayectorias de las acciones de control junto con las salidas postuladas. Este modo de funcionamiento es una mejora del anterior, ya que en caso de pérdida de datos el control local puede verificar si la salida real se corresponde con la salida postulada por el control supervisor y, en caso de discrepancia, realizar la compensación de la acción de control mediante el error cometido en la salida, tal y como se ha mostrado en el punto 4.5.5. Este modo de funcionamiento con la compensación de la acción de control es una de las aportaciones del presente trabajo.

Al utilizar las trayectorias postuladas y compensar la acción de control el nodo local debe informar al supervisor de los cambios realizados, para que al calcular las nuevas trayectorias se utilice la información actualizada.

El algoritmo de control consistiría en realizar las siguientes tareas:

- El nodo supervisor calcula mediante un controlador predictivo la trayectoria de las acciones de control, dentro de un horizonte de control N_u , que se deben aplicar a un proceso para que siga una referencia, junto con las salidas futuras, obtenidas como consecuencia de aplicar al modelo del proceso las acciones de control postuladas, dentro de un horizonte de predicción N . Estas trayectorias son (4.9) y (4.10).

El primer elemento de U es la acción de control que se debe aplicar en el instante k con la información hasta ese instante k , mientras que los demás

elementos son las acciones de control postuladas dentro del horizonte de control.

El primer elemento de Y es la salida leída actualmente en el sistema, y los demás elementos son las salidas postuladas, dentro del horizonte de predicción N , que se producen si se aplicaran las acciones de control U .

- El nodo local recibe estas trayectorias, y almacena internamente esta información en dos vectores. El vector de la trayectoria de las acciones de control dispone de N_u elementos, y el vector de la trayectoria de las salidas dispone de $N+1$ elementos. En Tabla 3 se muestra la información que se almacena en cada elemento de los vectores.

Etiqueta	u_0	u_1	...	u_{N_u-2}	u_{N_u-1}
Elemento almacenado	$u(k k)$	$u(k+1 k)$...	$u(k+N_u-2 k)$	$u(k+N_u-1 k)$

Etiqueta	y_0	y_2	...	y_{N-1}	y_N
Elemento almacenado	$y(k)$	$\hat{y}(k+1 k)$...	$\hat{y}(k+N-1 k)$	$\hat{y}(k+N k)$

Tabla 3. Vectores con las trayectorias de las acciones de control y las salidas que almacena el nodo local en el modo 4 de funcionamiento

La acción que se aplica sería la correspondiente al elemento u_0 , que coincide con la acción de control calculada con los datos hasta ese instante $u(k|k)$

- En el siguiente instante el nodo supervisor emplearía la estrategia del horizonte móvil, desechando los valores anteriores y volviendo a calcular las nuevas trayectorias con la información actualizada de la salida, y se las enviaría hacia el nodo local nuevamente.
- En caso de que no haya pérdida de datos el nodo local actualizaría los vectores internos de la trayectoria de las acciones de control y de la salida con los nuevos datos, aplicando la acción de control correspondiente al primer elemento u_0 .
- Sin embargo, en caso de que haya pérdida de datos y el nodo local no recibiese las trayectorias actualizadas lo que haría es utilizar la información postulada almacenada internamente en los vectores de la acción de control y de la salida.

La información de dichos vectores se actualizaría de la siguiente manera:

$$\begin{aligned}
 u_0 &= u_1 & y_0 &= y_1 \\
 u_1 &= u_2 & y_1 &= y_2 \\
 &\dots & &\dots \\
 u_{N_u-2} &= u_{N_u-1} & y_{N-1} &= y_N
 \end{aligned}
 \tag{4.18}$$

En este modo de funcionamiento también se utiliza un entero que memoriza la antigüedad de los vectores de datos δ_v , e igual que ocurría en el caso anterior, hay que actualizar este entero tal y como aparece en la ecuación (4.17).

El resultado de la actualización de los vectores internos del nodo local para la pérdida de un dato se puede ver en la Tabla 4.

Etiqueta	u_0	u_1	...	u_{N_u-2}	u_{N_u-1}
Elemento almacenado	$u(k+1 k)$	$u(k+2 k)$...	$u(k+N_u-1 k)$	$u(k+N_u-1 k)$

Etiqueta	y_0	y_2	...	y_{N-1}	y_N
Elemento almacenado	$\hat{y}(k+1 k)$	$\hat{y}(k+2 k)$...	$\hat{y}(k+N k)$	$\hat{y}(k+N k)$

Tabla 4. Actualización de los vectores con las trayectorias de las acciones de control y las salidas que almacena el nodo local cuando se pierde un dato

Nuevamente la acción de control que se aplica sería la correspondiente al elemento u_0 , que en este caso coincidirá con la acción de control postulada en algún instante anterior, pero en este caso se tiene la información de la salida que se debería haber producido, por lo que se puede compensar la acción de control aplicada para intentar seguir dicha salida comparándola con la que realmente existe. El método de la compensación de la acción de control se ha explicado en el apartado 4.5.4.3, como una de las posibles estructuras de control implementadas en el nodo local.

- Al recuperar las comunicaciones se volvería a desechar nuevamente la información almacenada, actualizándola con las nuevas trayectorias recibidas, correspondientes a la acción de control y la salida calculadas por el nodo supervisor.
- Este método supone que el nodo local debe enviar la acción de control aplicada, junto con la salida leída hacia el nodo supervisor para que éste calcule la acción

de control con toda la información actualizada proveniente del proceso. Las secuencias correspondientes a las acciones de control y salidas anteriores a enviar son (4.11) y (4.12).

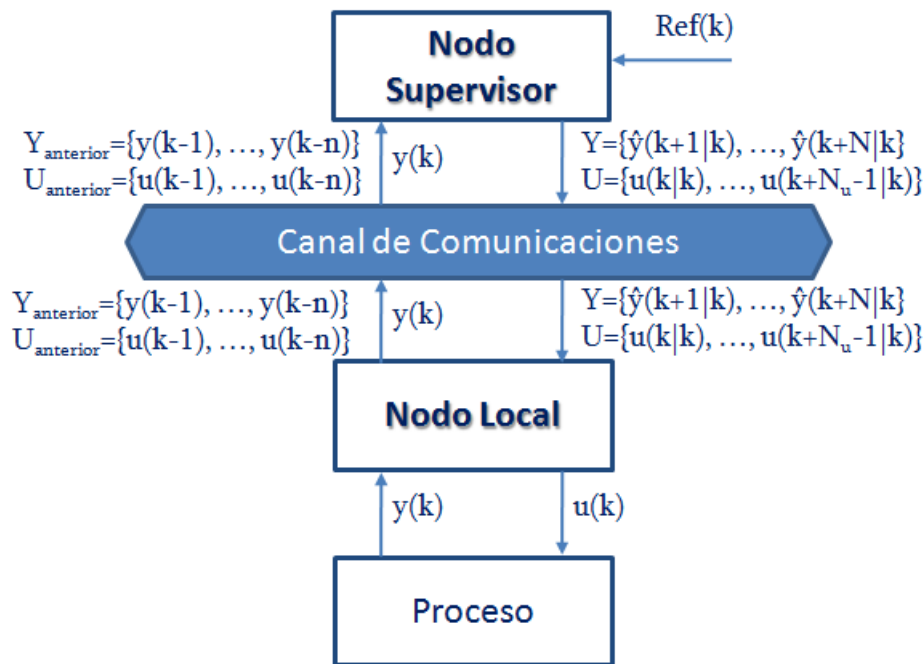


Figura 33. Modo 4 de funcionamiento

- **Ventajas:** En caso de pérdida de datos el nodo local puede utilizar las acciones de control postuladas y compensar las mismas con el error que se comete entre la salida real, que mide el nodo local, y la postulada por el control predictivo.
- **Inconvenientes:** El nodo supervisor calcula y envía las trayectorias de las acciones de control y de las salidas en cada periodo, con el coste computacional y de comunicaciones que ello conlleva, ya que se está enviando por el canal de comunicación mucha más información que la necesaria para realizar el control como prevención ante la pérdida de datos posterior.

Tal y como ocurre con el modo 3 de funcionamiento, dependiendo del tamaño de la trama, y del canal de comunicaciones utilizado, puede que el envío de las trayectorias de la salida y de la acción de control no suponga retrasos adicionales, siempre que entren todos los datos en una trama de tamaño mínimo, con lo que se estaría aprovechando el ancho de banda dedicado a las comunicaciones. Por ejemplo, la longitud del campo de datos de un paquete Ethernet puede oscilar de un mínimo de 46 a un máximo de 1500 bytes sin introducir una sobrecarga adicional en las comunicaciones (Lian, Moyne y Tilbury 2001).

4.5.6.5 *Modo 5. Envío de las acciones de control postuladas y de la trayectoria de salida nominal en los cambios de referencia*

En este modo de funcionamiento el nodo supervisor envía las trayectorias de las acciones de control y de las salidas postuladas hacia el nodo local. En caso de que no se produzca un cambio en la trayectoria previamente enviada, el nodo supervisor no realizará ningún cálculo ni envío mientras el nodo local disponga de información suficiente, por lo que el nodo local debe utilizar estas trayectorias, aplicando la compensación de la acción de control para seguir con la trayectoria nominal de la salida hasta que reciba nuevos datos, consecuencia de una actualización de la trayectoria por parte del nodo supervisor. Este modo de funcionamiento es similar a un sistema de control basado en eventos, en este caso el evento sería el cambio de referencia que da lugar a una actualización de los datos de las trayectorias de la salida y de la acción de control, que son las que utiliza el control local.

El cálculo de trayectoria nominal realizada por el control supervisor debería llevar en última instancia al sistema a una situación estable, para que en caso de perder los datos entre el nodo supervisor y el nodo local éste pueda llevar a la salida del sistema a una situación estable realizando pequeños cambios en la acción de control para contrarrestar el efecto del ruido.

Los aspectos relativos a la información que almacena el nodo local y como actúa cuando no tiene datos, es decir, hasta que el nodo supervisor le envíe las nuevas trayectorias, son similares a las presentadas en el modo de funcionamiento anterior.

Con esta estrategia de control el nodo supervisor debe enviar los datos dentro de un horizonte de predicción que englobe a toda la dinámica del sistema, para que se asegure que la última salida postulada se encuentre sobre la referencia en una situación estable. Los aspectos relativos a la elección de los parámetros del control así como a la elección del horizonte de predicción se estudiarán en el apartado 4.6.

Al igual que ocurría anteriormente, al utilizar las trayectorias postuladas y compensar la acción de control, el nodo local debe informar al supervisor de los cambios realizados, para que al calcular las nuevas trayectorias se utilice la información actualizada, para ello se utilizarán las ecuaciones (4.11) y (4.12).

- Ventajas: Disminución significativa del ancho de banda dedicado a la comunicación entre el nodo supervisor y el nodo local, así como los cálculos realizados en el nodo supervisor. Además, el nodo local mantiene la metodología explicada en el modo anterior.
- Inconvenientes: Se amplían las señales involucradas y se realiza un diseño modular más complicado, aunque las ventajas justifican este incremento en el diseño del control. Además en este caso la respuesta transitoria podría empeorar al utilizar en todo momento las trayectorias postuladas.

En este modo de funcionamiento el nodo supervisor recibe en cada momento la salida real del proceso por lo que, en caso de error significativo entre la salida nominal y la salida actual del proceso, éste puede recalcular nuevas trayectorias y enviárselas al nodo local.

En el modo 5 el nodo local está constantemente enviando las acciones de control y las salidas anteriores, de esta forma el nodo supervisor puede realizar el cálculo de unas nuevas trayectorias, con la información actualizada del comportamiento del proceso, cuando considere oportuno. Por lo tanto una mejora, en cuanto al paso de información en el canal de comunicaciones, sería enviar los datos entre el local y el supervisor cuando éste los necesite. En ese caso el protocolo de comunicaciones y las señales involucradas se complican.

4.5.7 Tratamiento de los retardos y desfases en la comunicación

Uno de los problemas al utilizar un sistema distribuido, en el que el canal de comunicaciones no es de uso exclusivo, son los debidos a los retardos y desfases en la comunicación, así como las pérdidas de datos. En este último caso ya se han mostrado algunos modos de funcionamiento en la estrategia de control que permiten dar solución a esta eventualidad. En el caso de tratarse de un retardo o de un desfase en las comunicaciones los datos que llegan al receptor no están actualizados y por lo tanto requieren de un tratamiento especial.

En este apartado es importante la sincronización de los nodos, ya que la única forma de saber cuál es el retraso en la información recibida es mediante las marcas temporales que llevan asociados los datos, y por lo tanto son necesarios mecanismos de sincronización de nodos, tal y como se ha explicado en el apartado 4.5.2.

Cuando al nodo local le llegue directamente la acción de control, o bien la referencia, tal y como se ha estudiado en el apartado 4.5.6 en los modos de funcionamiento 1 y 2, se debe comprobar la antigüedad del dato recibido, para ello el nodo supervisor añadirá junto con el dato una marca temporal del instante de tiempo para el que ha sido generado. Sea τ_{SL} la diferencia temporal computada en el nodo local entre el tiempo actual y el que se recibe junto con el dato correspondiente. En el caso de que la diferencia temporal sea menor que el periodo del nodo local (T_{CL}) el dato es válido y se utilizará directamente, en caso contrario el dato no es válido y por lo tanto se desecha.

En aquellos modos de funcionamiento en el que el nodo supervisor envíe las trayectorias de las acciones de control y las salidas, tal y como se describe en las ecuaciones (4.9) y (4.10), el nodo supervisor igual que en el caso anterior, añadirá junto a los datos una marca temporal del instante de tiempo para el que han sido generados, de forma que el nodo local debería comprobar con esa marca la antigüedad de los datos recibidos. Sea τ_{SL} la diferencia temporal computada en el

nodo local entre el tiempo actual y el que se recibe junto con los datos de las trayectorias.

Considerando que los periodos de actuación de ambos nodos son iguales, el tratamiento de los datos recibidos por parte del nodo local será el siguiente:

Los datos serán válidos siempre que sean más actuales que los que se encuentran almacenados, para ello se debe cumplir lo siguiente:

$$\tau_{SL} < \delta_v \cdot T_{CL} \quad \delta_v \in \mathbb{N} \tag{4.19}$$

siendo δ_v el entero que informa de la antigüedad del vector con los datos de las acciones de control y de las trayectorias, tal y como se ha mostrado en el apartado 4.5.6 en los modos de funcionamiento 3, 4 y 5, y T_{CL} el periodo de actuación del control local. La siguiente figura muestra como se generan, envían y reciben los datos en el sistema distribuido:

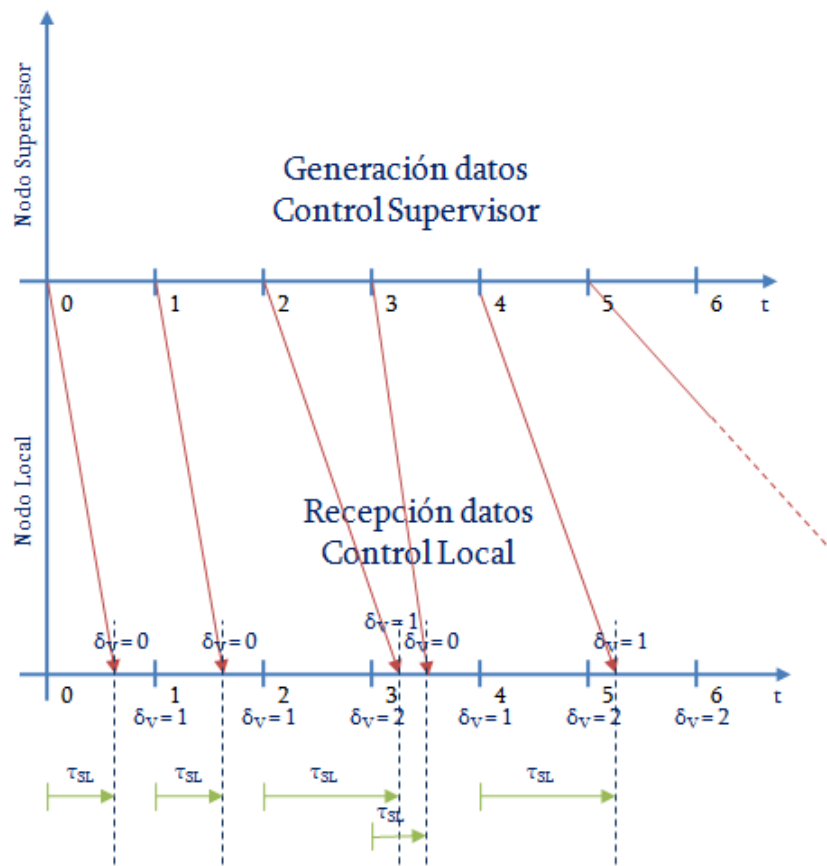


Figura 34. Actualización de los vectores de datos ante retardos y desfases en la comunicación

Cuando se cumpla (4.19) los datos serán válidos, y el nodo local procederá a actualizar los vectores internos de las trayectorias de la siguiente manera:

Caso 1. $\delta_v = 1$

En este caso τ_{SL} será menor que el periodo de actuación del nodo local T_{CL} , y se considerará que no se ha producido ningún retraso en la recepción de la información, por lo tanto se actualizarán los vectores internos del control local directamente, utilizando la acción de control propuesta por el primer elemento del vector y poniendo el valor del entero con la antigüedad del vector de datos $\delta_v = 0$.

Caso 2. $\delta_v > 1$

En este caso los datos son más actuales que los que se encuentran almacenados, pero τ_{SL} es mayor que el periodo de actuación del nodo local, por lo tanto hay elementos en las trayectorias recibidas que no válidos, ya que hacen referencia a instantes de tiempo que ya se han producido.

Sea ψ el entero con el número de elementos no válidos de las trayectorias recibidas procedentes del control supervisor. Este entero se puede calcular de la siguiente manera:

$$\psi = \text{floor}\left(\frac{\tau_{SL}}{T_{CL}}\right) \quad (4.20)$$

Por lo tanto ψ corresponde al entero inmediatamente inferior del resultado de la división entre el retardo de los datos recibidos y el periodo de muestreo del control local.

Se copiarán las trayectorias recibidas en los valores internos y se deberá proceder a realizar operaciones de diezmado de los elementos no válidos, actualizando los vectores internos de las trayectorias del control y la salida de forma adecuada. La información de dichos vectores se actualizaría de la siguiente manera:

$$\begin{aligned} u_0 &= u_{\min(\psi, Nu-1)} & y_0 &= y_{\min(\psi, N)} \\ u_1 &= u_{\min(\psi+1, Nu-1)} & y_1 &= y_{\min(\psi+1, N)} \\ & \dots & & \dots \\ u_{Nu-1} &= u_{\min(\psi+Nu-1, Nu-1)} & y_N &= y_{\min(\psi+N, N)} \end{aligned} \quad (4.21)$$

Si el índice de la actualización supera el tamaño del vector se tomará el último dato para actualizar esos valores.

Una vez actualizados los vectores internos de las acciones de control y salida se aplicará la estrategia de control correspondiente al modo de funcionamiento en el

que se encuentre configurado el sistema, tal y como se ha explicado en el apartado 4.5.6.

Así mismo, en el caso de que se consideren retrasos en la recepción de los datos suministrados desde el nodo local hacia el nodo supervisor, se debería actuar de forma similar. El nodo local añadirá junto con los datos una marca temporal del instante de tiempo para el que han sido generados, de forma que el nodo supervisor debería comprobar con esa marca la antigüedad de los datos recibidos. Sea τ_{LS} la diferencia temporal computada por el nodo supervisor entre el tiempo actual y el que se recibe junto con los datos. En este caso si la diferencia es menor que el periodo no hay ningún problema. Sin embargo, en caso de que sea mayor se deberían desechar, ya que para generar la acción de control al nodo supervisor solo le interesan los valores actualizados.

4.6 Metodología de diseño para la elección de los parámetros de control en el sistema distribuido

En este apartado se presenta una metodología que permite la elección de los parámetros del control distribuido para el caso de un nodo supervisor que implemente un control predictivo, y un nodo local que utilice la compensación de la acción de control, es decir, cuando en el sistema distribuido de control se utilice los modos de funcionamiento 4 ó 5, tal y como se ha explicado en el apartado 4.5.6. También es aplicable al modo de funcionamiento 3, pero en este caso no se utilizaría la compensación de la acción de control en el nodo local.

La utilización de los controles predictivos basados en modelo permite la obtención de las trayectorias postuladas de las acciones de control y de las salidas a partir de la minimización de una función de coste, y por lo tanto son aplicables en la utilización del control distribuido presentado en el apartado 4.5, aunque se puede utilizar cualquier otro control que permita obtener dichas trayectorias postuladas.

La Figura 35 muestra el método para elegir los parámetros en el sistema distribuido de control. Los tres primeros pasos corresponden a la elección de los parámetros del control predictivo que implementará el nodo supervisor, mientras que el cuarto paso es el que permite seleccionar la ganancia que implementará el control local para realizar la compensación de la acción de control a partir del error en la salida.

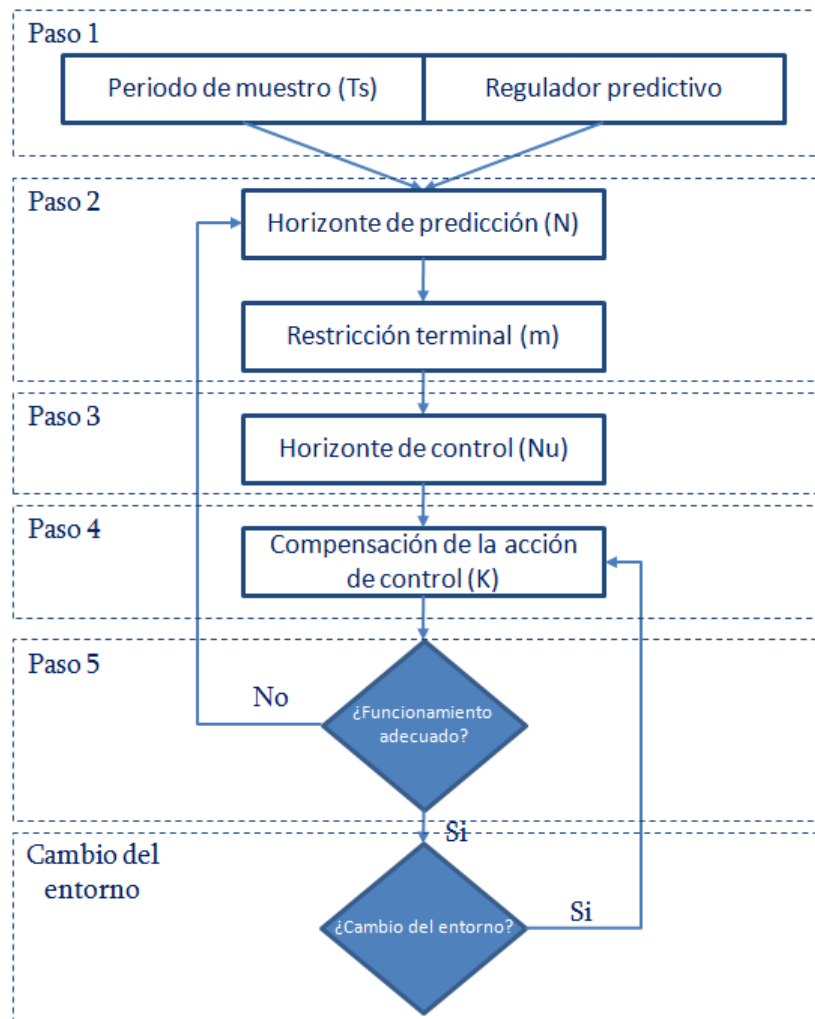


Figura 35. Metodología para la elección de los parámetros del sistema de control

En los siguientes apartados se desarrollan cada uno de estos pasos a realizar para la selección de los parámetros de cada nodo.

4.6.1 Elección de los parámetros del control MBPC en el nodo supervisor

El objetivo del nodo local es, con el menor coste computacional posible, conseguir que la salida real siga a la trayectoria nominal proporcionada por el nodo supervisor, incluso en presencia de perturbaciones y pérdida de datos. Para ello es necesario que el nodo local disponga de las trayectorias postuladas suficientes, tanto de la salida como de las acciones de control, para que en caso de una pérdida significativa de datos se consiga llevar a la salida a la última referencia dada con la dinámica prevista.

El método de elección de los parámetros de un control MBPC más adecuados para cumplir con el objetivo anterior, se corresponde con los pasos del 1 al 3 de la Figura 35, y consistiría en lo siguiente:

Partiendo del modelo del proceso se elige un periodo y un control predictivo, por ejemplo GPC ó CRHPC, que se encargue del cálculo de las trayectorias postuladas de la salida y de la acción de control.

Realizando una simulación con el control elegido sin pérdida de datos, se prueba el comportamiento del sistema, para ello se debe tomar un horizonte de predicción y de control iniciales. La idea es que el horizonte de predicción (N) contenga el transitorio de la salida del sistema que se quiere controlar. Un horizonte de predicción alrededor de diez instantes suele ser un buen comienzo. El objetivo es que el control predictivo proporcione las acciones de control futuras y las salidas futuras a partir del instante actual, por lo tanto el horizonte de predicción debe cumplir que $N_1=1$ y $N_2=N$. En el caso de utilizar un control predictivo CRHPC que necesita de la restricción terminal (m) se debe definir este número de instantes.

Por otro lado, en el cálculo del horizonte de control N_u hay que tener en cuenta que se pueden utilizar los valores postulados, y por lo tanto lo que interesa es que la acción de control no cambie muy bruscamente, además de que la salida postulada se parezca lo máximo posible a la salida que tendría el sistema si se aplicara la estrategia del horizonte deslizante. Cuanto mayor es el horizonte de control se permite que las variaciones en la acción de control se realicen durante más instantes. A medida que aumenta el horizonte de predicción N_u también aumenta de forma significativa el tiempo de cómputo invertido por el nodo supervisor en el cálculo de los vectores con las trayectorias postuladas. Sin embargo esto no es problema, ya que el nodo supervisor no tiene limitaciones en el tiempo de cómputo, aunque siempre es deseable que sea el menor posible.

Por lo tanto, se debe estudiar, para el caso concreto del proceso a controlar, así como el regulador implementado, cual es el mínimo valor de N_u que consigue una discrepancia en la salida postulada aceptable cuando se compara con el comportamiento del sistema cuando no hay problemas. Esto puede hacerse mediante un método heurístico de optimización numérica, es decir, aplicando distintos valores del horizonte de control N_u y estudiando a partir de qué valor se consigue disminuir el error de forma adecuada.

Además, en función del valor del horizonte de predicción el valor de N_u que consigue un error mínimo entre la salida postulada y la real cambia, por lo tanto se plantea la realización de distintas pruebas para evaluar que combinación de horizonte de predicción y horizonte de control consigue el error más pequeño. Para cada horizonte de predicción N se evaluación todos los posibles horizontes de control (desde $N_u=2$ hasta $N_u=N$), y se calcula el error cometido entre la predicción y la salida real cuando se pierden todos los datos, quedándose con el N_u que consiga el mínimo error. Al final se elegirá la combinación de N y N_u que consiga el mínimo error. En la Figura 36 se muestra un ejemplo de cómo varía el error cometido en la salida, entre el valor predicho y el real cuando se utiliza las acciones postuladas, para

los distintos horizontes de predicción (N) y cuál es el valor de N_u que consigue ese error mínimo. De todo ese conjunto de valores se puede obtener aquella combinación que consigue el mínimo error de todos. El índice utilizado para el cálculo del error ha sido el absoluto (IAE), en el apartado 5.1 se presentan los índices de error utilizados para la estimación de la calidad de la respuesta de control.

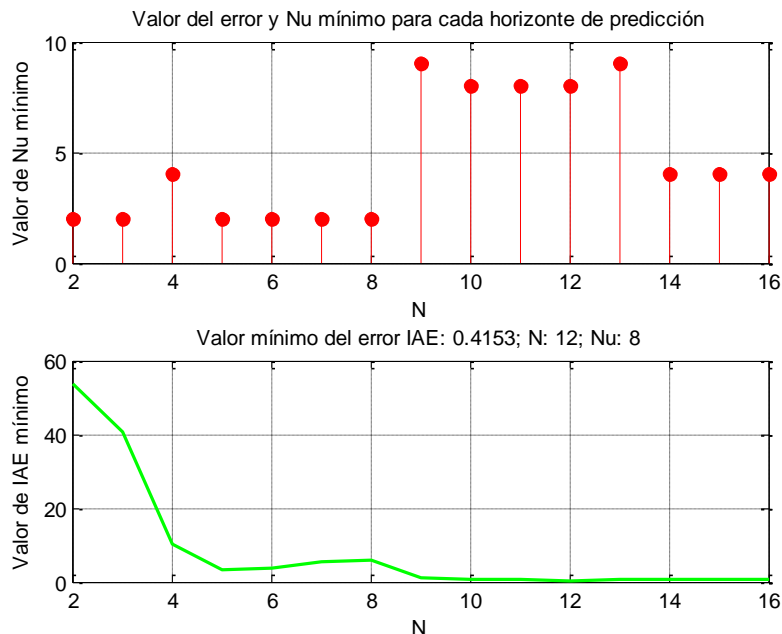


Figura 36. Valor mínimo de IAE para cada N y el N_u que hay que aplicar para conseguirlo

El horizonte de predicción debe ser mayor de 1 para que al aplicar la estrategia del horizonte móvil la dinámica de la salida sea la misma, independientemente del número de instantes del horizonte de control y predicción.

En el siguiente capítulo se muestra, mediante varios procesos distintos, la aplicación de la metodología explicada.

4.6.2 Elección de la constante K de compensación de la acción de control en el nodo local

La elección de la constante de compensación corresponde al paso 4 del método descrito en la Figura 35. La aplicación de la compensación de la acción de control sirve para mejorar la respuesta del sistema cuando se utilizan las acciones de control postuladas, de forma que no se realice un control en bucle abierto, tal y como se ha visto en el apartado 4.5.4.3. La elección de la constante que multiplica al error en la salida es muy importante, ya que dependiendo de su valor puede hacer que el comportamiento mejore o empeore comparado con la aplicación directa de las acciones postuladas.

Para calcular el rango de valores de la ganancia de la compensación adecuados a las condiciones del sistema es importante estudiar las condiciones concretas de aplicación, es decir, el proceso que se desea controlar y la magnitud del error en la salida.

Un método para obtener el rango de valores de la ganancia de la compensación de la acción de control sería a partir de un método heurístico de optimización numérica, aplicando distintos valores de K en simulación, de forma que se seleccionan aquellos que hagan mínimo el error en la salida.

Ejemplo A continuación se muestran los errores cometidos por la salida real con relación a la trayectoria nominal para distintas K cuando se utiliza la predicción con la compensación de la acción de control, para el caso de un proceso de 2º orden subamortiguado (Figura 37) y para un proceso con retardo múltiple del periodo (Figura 38). En el capítulo 5 y el anexo B se detallarán los procesos utilizados así como la pruebas.

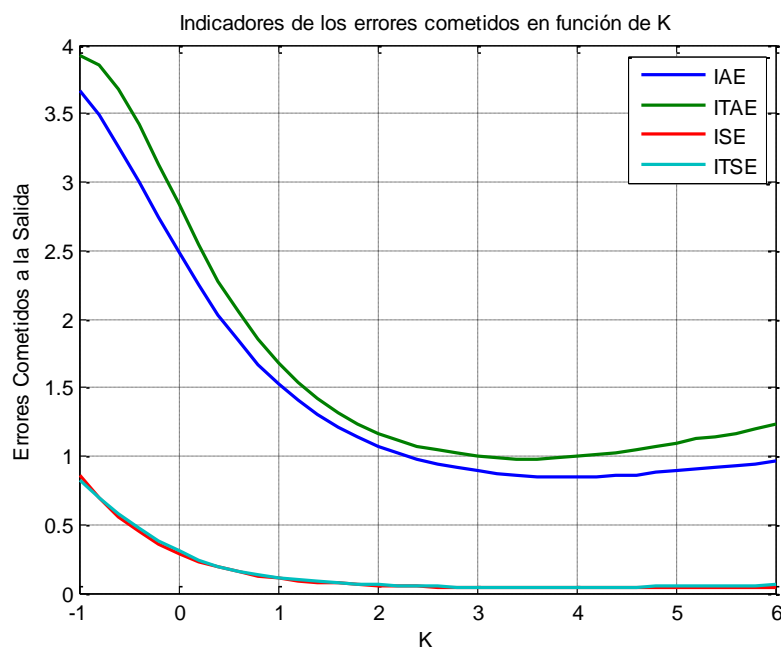


Figura 37. Indicadores de error para distintos valores de K aplicados a un proceso de 2º orden subamortiguado

Cualquiera de los indicadores de error, tanto los absoluto (IAE, ITAE) como los cuadráticos (ISE, ITSE) muestran la misma tendencia.

En el primer caso (Figura 37) se muestra como los valores adecuados de K para el proceso y las condiciones del entorno tiene un mínimo error alrededor de $K=4$.

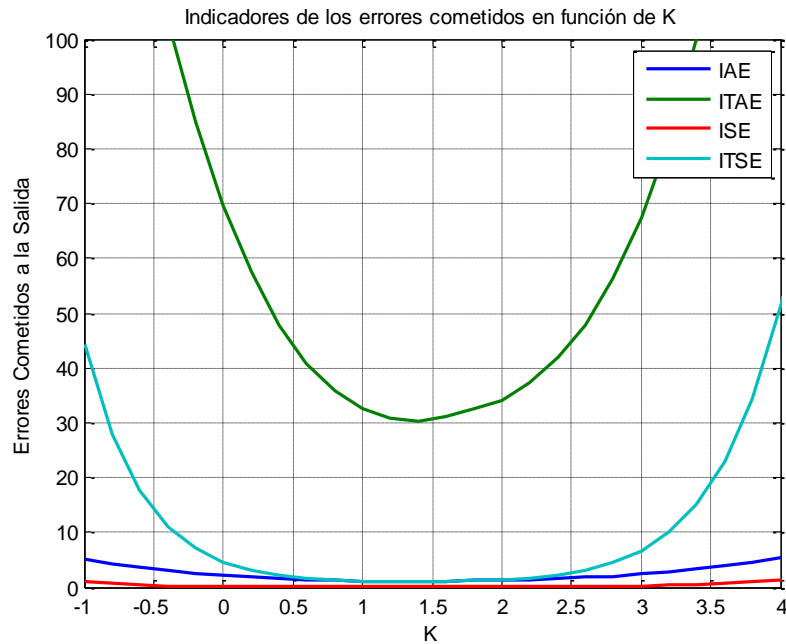


Figura 38. Indicadores de error para distintos valores de K aplicados a un proceso con retardo múltiplo del periodo

Por otro lado, en el segundo caso (Figura 38) para el proceso con un retardo múltiplo del periodo el valor de K que consigue un error mínimo es $K=1.2$. ■

4.7 Resumen y conclusiones del capítulo

En este capítulo se ha definido el núcleo de control y su adaptación a un sistema distribuido. Para poder realizar esa adaptación se ha definido el middleware del núcleo de control, en el que se han generado dos versiones: la completa y la reducida.

Se ha propuesto una estructura distribuida formada por nodos supervisores, sin problemas computacionales, y nodos locales que son los que se encuentran cerca de los procesos a controlar pero con limitaciones de cómputo. Los nodos supervisores implementan el middleware completo, mientras que los nodos locales implementan la versión reducida.

Se ha definido el escenario del problema en el que se va a aplicar este sistema distribuido de control. Los procesos considerados son SISO, en los que se tienen en cuenta perturbaciones a la salida, además de otro tipo de problemas como pueden ser los errores de modelado, distintos periodos en los nodos del sistema distribuido, tiempo de cómputo y aquellos relacionados con el canal de comunicaciones: pérdida de datos, retrasos, desfases y ancho de banda dedicado a las tareas de control.

La estructura distribuida y la estrategia de control han dado solución a los distintos problemas del entorno que se han mencionado y se han generado distintos modos de

funcionamiento, que permiten variar los tiempos de cómputo de los nodos y que el control local aplique una acción de control incluso ante pérdida de datos con el control supervisor.

Una aportación de la tesis ha sido la utilización de las acciones de control predichas por el control supervisor por parte del control local en caso de pérdida de datos, utilizando la compensación de la acción de control con respecto al error cometido a la salida.

Además se propone una metodología para la elección de los parámetros de los controladores a implementar en el nodo local y supervisor, en caso de utilizar una compensación con una ganancia y un control predictivo basado en modelo GPC o CRHPC, respectivamente.

La herramienta de simulación del sistema distribuido desarrollada en esta tesis permite la evaluación de las distintas estructuras así como el diseño de los controladores que se implementaran en el control local y supervisor. En el capítulo 5 y el anexo B se realiza la prueba de esta metodología sobre distintos procesos utilizando esta herramienta.

5 PRUEBAS EN SIMULACIÓN

El objetivo de este capítulo es la evaluación de la estructura distribuida y la estrategia de control propuesta en el capítulo anterior mediante el uso de herramientas de simulación. Para ello se ha empleado Matlab junto con Simulink para la generación de las estructuras de control y el modelado de los sistemas, dando como resultado una herramienta de simulación del sistema distribuido de control. En el anexo A se encuentra más información al respecto.

5.1 Métodos de estimación de la calidad de la respuesta de control

Antes de entrar a presentar las pruebas realizadas es necesario un método que estime la calidad de la respuesta de control mediante un indicador. Este indicador permite plasmar mediante una formulación matemática una medida cuantitativa del funcionamiento de un sistema en un instante dado, con lo que se puede evaluar y elegir los parámetros a implementar en el sistema distribuido de control que mejor se ajustan a las distintas situaciones.

En teoría de control de procesos aparecen distintos indicadores que tratan de describir la evolución dinámica de un proceso, y se pueden englobar en dos grandes grupos:

- Parámetros descriptivos de la evolución temporal de la variable controlada
- Medidas de la desviación de la variable controlada respecto de la referencia

5.1.1 Parámetros descriptivos de la evolución temporal de la variable controlada

La respuesta que ofrecerá un determinado sistema de control, siempre que ésta se mantenga estable, la podemos dividir en la respuesta en el régimen transitorio y la correspondiente al régimen permanente, tal como se puede ver en la figura.

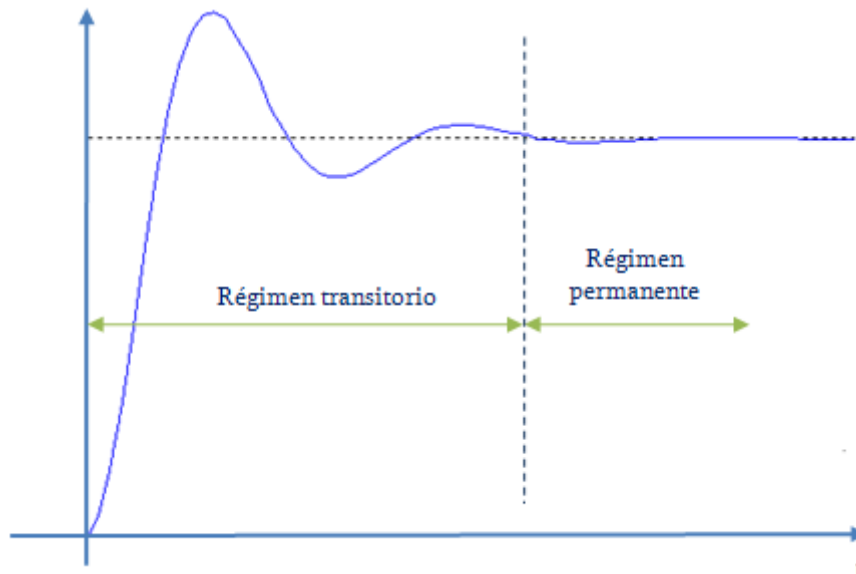


Figura 39. Régimen transitorio y permanente

En la teoría de control es conocido que el comportamiento en el régimen transitorio se caracteriza por el porcentaje de sobreoscilación máxima, el tiempo de establecimiento, la frecuencia de las oscilaciones, tiempo de subida, etc. considerándose que la respuesta será mejor cuanto más pequeños sean los valores del tiempo de establecimiento y la sobreoscilación, lo cual dependerá de las características del sistema y del controlador en ausencia de perturbaciones. En el caso de que éstas aparecieran, dicho comportamiento empeoraría.

En el caso que nos ocupa se considerarían perturbaciones tanto habituales en control (ruido, dinámicas no modeladas,...) como la interacción de las tareas de control al ser implementadas, con elementos empotrados con escasos recursos computacionales, en un sistema distribuido con un canal de comunicaciones compartido.

En cuanto al comportamiento en el régimen permanente se puede caracterizar por el error en régimen permanente y el valor final.

5.1.2 Medidas de la desviación de la variable controlada respecto de la referencia

Cuando se quiere evaluar la calidad de la respuesta de control, generalmente se usan criterios basados en la medida del error en bucle cerrado, obteniendo una medida cuantitativa de esta calidad (Graham y Lathrop 1953). Esto es, cuanto menor es el valor obtenido, menor es también el error. Los índices que utilizan el error absoluto se conocen como IAE e ITAE (Dorf y Bishop 1995).

IAE: integral del valor absoluto del error

$$IAE = \int_{t_0}^{t_f} |y_{des}(t) - y_{act}(t)| dt \quad (5.1)$$

ITAE: integral del valor absoluto del error ponderado según el tiempo

$$ITAE = \int_{t_0}^{t_f} t \cdot |y_{des}(t) - y_{act}(t)| dt \quad (5.2)$$

donde:

- y_{des} = respuesta deseada del sistema
- y_{act} = respuesta real del sistema
- t_0, t_f = instantes inicial y final del periodo de evaluación

ITAE le da un peso mayor a los últimos errores mientras que IAE pondera todos los errores de manera idéntica.

En lugar del valor absoluto se puede utilizar el error cuadrático, en ese caso se definen los índices ISE e ITSE (Marlin 2000):

ISE: integral del error al cuadrado

$$ISE = \int_{t_0}^{t_f} (y_{des}(t) - y_{act}(t))^2 dt \quad (5.3)$$

ITSE: integral del error al cuadrado ponderado según el tiempo

$$ITSE = \int_{t_0}^{t_f} t \cdot (y_{des}(t) - y_{act}(t))^2 dt \quad (5.4)$$

Como el cálculo de la integral hay que hacerlo en tiempo de ejecución y es costoso, se puede utilizar una aproximación al cálculo integral o se puede calcular otro parámetro de calidad como es la varianza de la respuesta:

$$var(x) = \frac{1}{n-1} \cdot \sum_{i=1}^n (x_i - \bar{x})^2 \quad (5.5)$$

La varianza se va calculando en línea utilizando una ventana temporal de N muestras.

La varianza es otro indicativo del buen o mal funcionamiento del sistema, esto es, aunque el error en un momento dado no sea excesivamente grande, podría darse el caso de que estuviese oscilando alrededor del valor de referencia, cosa que denotaría también poca calidad en la respuesta de control.

5.1.3 Método propuesto para estimar la calidad de la respuesta de control

El objetivo de la estrategia de control desarrollada es que la salida del proceso real coincida con la trayectoria propuesta por el control supervisor. Esta trayectoria es tomada por el nodo local como una trayectoria nominal, por lo tanto se ha de considerar como la salida deseada.

En un sistema real los errores existentes se producen debido a discrepancias entre el modelo y el sistema real, así como a los ruidos y perturbaciones en las señales involucradas. Además, con este mismo método también se pueden evaluar los errores cuando se utilizan las acciones de control postuladas en lugar del método del horizonte deslizando.

En el caso de un sistema discreto se propone la siguiente aproximación para el cálculo de los índices anteriores:

$$\begin{aligned} IAE &= \sum_{i=0}^n |y_{des}(i) - y_{act}(i)| \\ ITAE &= \sum_{i=0}^n i \cdot T \cdot |y_{des}(i) - y_{act}(i)| \\ ISE &= \sum_{i=0}^n (y_{des}(i) - y_{act}(i))^2 \\ ITSE &= \sum_{i=0}^n i \cdot T \cdot (y_{des}(i) - y_{act}(i))^2 \end{aligned} \quad (5.6)$$

Siendo:

- n : número de muestras consideradas para el cálculo del error
- T : periodo de muestreo de la señal de salida

Por lo tanto, el error cometido en el instante k será:

$$Error(k) = y_{des}(k) - y_{act}(k) \quad (5.7)$$

La reducción en las prestaciones que sufre un sistema distribuido cuando se produce la pérdida de datos depende del instante en el que se produce la pérdida, y esto puede variar de un proceso a otro. Además, esta reducción de prestaciones es muy dependiente de la pérdida de datos producidas anteriormente, y por lo tanto de la situación en la que se encuentre la salida del proceso.

Para evaluar cómo afectan las distintas pérdidas de datos a las prestaciones del sistema se propone el siguiente método:

- Someter al sistema a varias pruebas con el mismo porcentaje de pérdidas de datos, pero distribuidas las pérdidas de forma aleatoria a lo largo de la simulación.
- Se tomará como indicador de la pérdida de prestaciones el error cometido en la salida del proceso en cada instante respecto de la salida del proceso sin pérdida de datos.
- El indicador que determina las prestaciones para cada porcentaje de pérdidas será tomar el mayor índice obtenido para el conjunto de pruebas con ese porcentaje de pérdidas. Este índice expresa el peor de los casos por cada porcentaje de pérdidas. En las pruebas realizadas también se representa el valor medio del error cometido para cada porcentaje de pérdidas, lo que permite mostrar si el peor caso es un valor puntual o hay una tendencia.
- Repetir las pruebas para evaluar la pérdida de prestaciones en cada porcentaje de pérdidas.

El índice utilizado para evaluar la pérdida de prestaciones puede ser cualquiera de los presentados en (5.6). En los ejemplos que se presentan en el apartado 5.2 y en el anexo B se ha utilizado el índice IAE.

En la Figura 40 se muestran las curvas de los errores IAE máximos y medios cometidos por cada porcentaje de pérdidas de datos, en el que se han realizado 100 pruebas con cada porcentaje de pérdida de datos, distribuidas de forma aleatoria en el tiempo de la prueba.

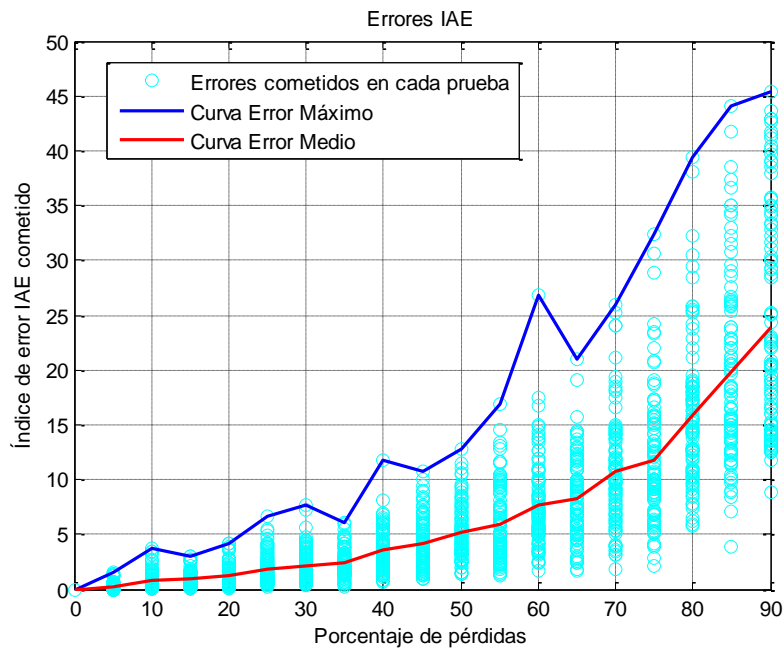


Figura 40. Curvas de errores IAE máximos y medios cometidos por cada porcentaje de pérdidas

Con esta metodología se pueden evaluar la influencia de los distintos modos de funcionamiento ante las mismas condiciones del entorno. Este estudio se muestra en el siguiente apartado aplicado a distintos procesos.

5.2 Ejemplos de implementación

En este apartado se muestra, con ejemplos, la implementación de la metodología de elección de los parámetros del sistema de control distribuido y la evaluación de su funcionamiento. Se estudiará el comportamiento del sistema distribuido, tanto en su funcionamiento normal como en situaciones de pérdida de datos. Estas pruebas permitirán también conocer las características de la herramienta de simulación del sistema distribuido con la que se facilita la elección de los parámetros del control y la simulación de los sistemas ante diversas condiciones de entorno. Las características de la herramienta se encuentran desarrolladas en el anexo A.

Los distintos procesos SISO utilizados en las pruebas proporcionan dinámicas diferentes. La elección de los distintos procesos se ha realizado teniendo en cuenta que pertenecen a procesos reales, y que se disponen de ellos en el laboratorio o que se han documentado en algún trabajo previo. Las características de los procesos elegidos son similares a los modelos para la evaluación en el Benchmark 2010-2011 del Grupo Temático de Ingeniería de Control de CEA para la Evaluación de algoritmos de Auto-Ajuste de Controladores PID (Benchmark CEA 2010-2011).

Los procesos elegidos son los siguientes:

- Proceso de primer orden
- Proceso de segundo orden
- Proceso de fase no mínima
- Proceso con retardo múltiplo del periodo de muestreo
- Proceso con efecto integral

A continuación se detallan la elección de parámetros así como las pruebas realizadas para el proceso de segundo orden. En estas pruebas se utilizarán distintos controles supervisores y estructuras distribuidas, así como los distintos modos de funcionamiento que se han detallado en el apartado 4.5.6. Las pruebas realizadas con el resto de procesos se encuentran detalladas en el anexo B. En el apartado 5.2.2 se encuentra un resumen con los datos obtenidos en todas las pruebas realizadas.

5.2.1 Pruebas realizadas en el proceso de segundo orden

El proceso elegido en este caso para realizar las pruebas es un circuito electrónico con una dinámica de segundo orden con respuesta ante escalón subamortiguada. En la Figura 41 se muestra el circuito electrónico del proceso.

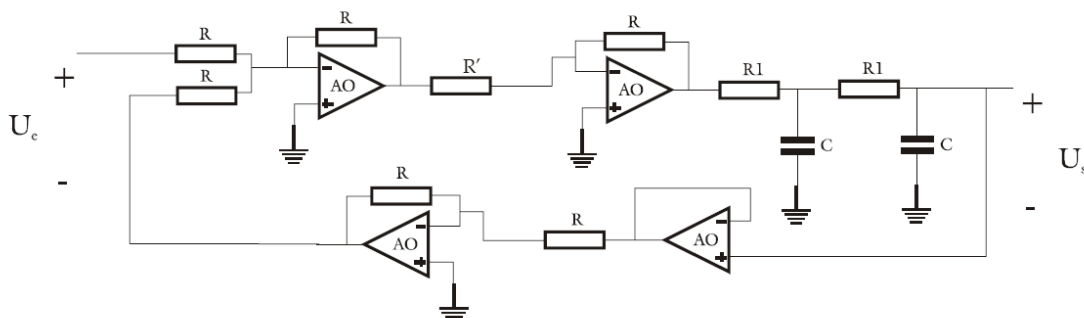


Figura 41. Circuito electrónico del proceso de segundo orden subamortiguado

Desde el punto de vista del control, el proceso anterior presenta el siguiente modelo en forma de diagrama de bloques:

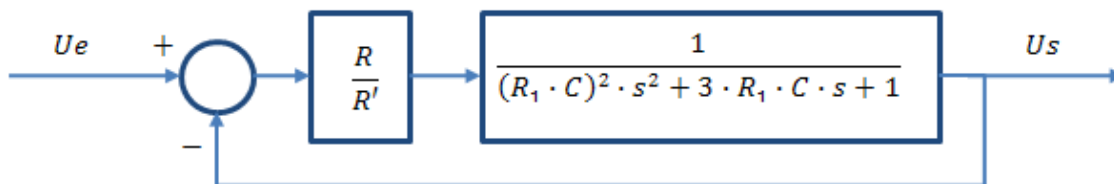


Figura 42. Diagrama de bloques del proceso utilizado en las pruebas

Los valores que toman los diferentes componentes en el circuito electrónico son los siguientes:

- $R = 150 \text{ K}\Omega$
- $R' = 22 \text{ K}\Omega$
- $R1 = 470 \text{ K}\Omega$
- $C = 0.68 \text{ }\mu\text{F}$

Para las características concretas del circuito electrónico se obtiene la siguiente función de transferencia:

$$G_p(s) = \frac{U_s}{U_e} = \frac{6.818}{0.1021 \cdot s^2 + 0.9588 \cdot s + 7.818}$$

Este proceso dispone de dos polos complejos estables, por lo que consigue una respuesta subamortiguada ante escalón.

5.2.1.1 Elección de los parámetros del sistema distribuido de control

Mediante este ejemplo se propone estudiar el comportamiento de un sistema distribuido de control formado por un nodo supervisor, en el que se ejecuta un control predictivo, y un nodo local conectado directamente al proceso de segundo orden que se desea controlar. En este caso, los parámetros necesarios en el sistema distribuido de control son los siguientes:

- Periodo de muestreo (T_s)
- Regulador predictivo (GPC ó CRHPC)
- Horizonte de predicción (N)
- Restricción terminal (m)
- Horizonte de control (N_u)
- Compensación de la acción de control (K)

Para esta prueba se ha utilizado el mismo periodo de muestreo T_s de 50 milisegundos tanto en el nodo supervisor como en el local. En la elección de este periodo se ha tenido en cuenta la respuesta dinámica del sistema en bucle cerrado, de forma que se obtengan alrededor de 10 muestras en la respuesta dinámica del proceso a controlar.

Además de los parámetros anteriores en los regulares predictivos GPC y CRHPC también están estos:

- Factor de ponderación de las salidas (α)
- Factor de ponderación del esfuerzo de control (λ)
- Polinomio del control predictivo ($T(z^{-1})$)

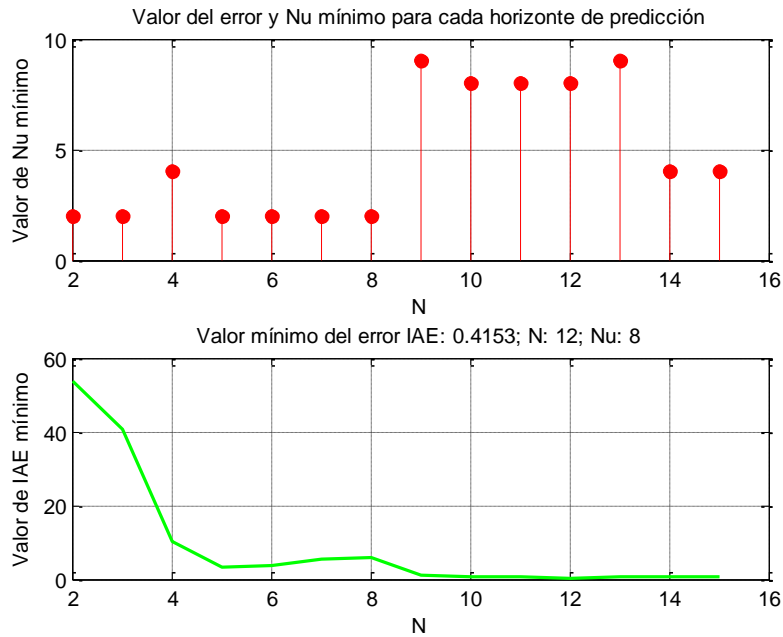


Figura 44. Estudio de Nu en función de N con el que se obtiene el mínimo error

En la figura anterior se muestra el valor de IAE mínimo para cada N y el horizonte de control con el que se ha obtenido. La combinación que consigue el mínimo error es $N=12$ para un $Nu=8$. La herramienta muestra en la cabecera de la figura el valor numérico de esa combinación con la que obtiene el IAE más pequeño.

A continuación se muestra la variación de los errores IAE a medida que se varía Nu para el caso concreto de $N=12$, en la que para $Nu=8$ se consigue ese valor mínimo.

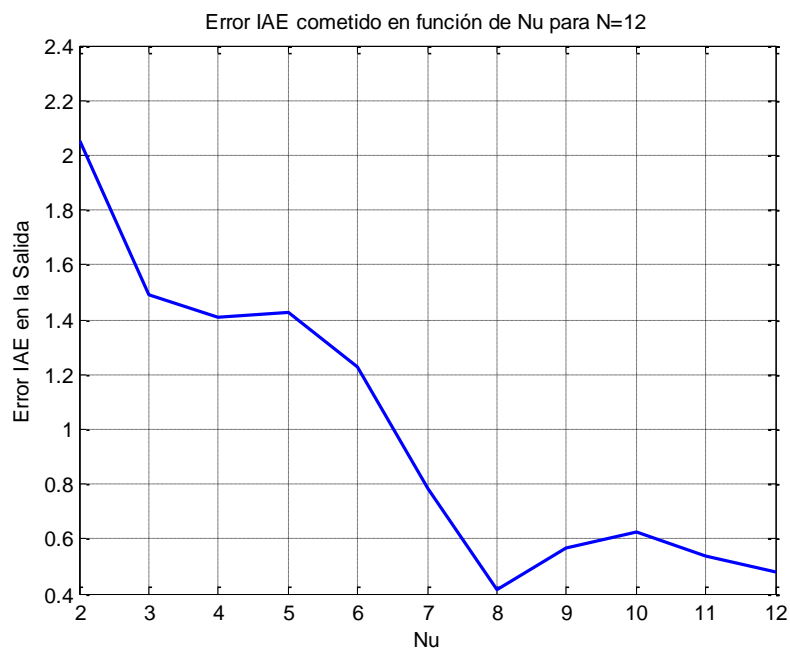


Figura 45. Errores cometidos en función de Nu para $N=12$

En el nodo local se puede utilizar la estrategia de la compensación de la acción de control, mediante la discrepancia entre la salida postulada y la salida real, y por lo tanto hace falta obtener el valor de K . Para ello lo que se hace es, a partir de los parámetros del control supervisor obtenidos anteriormente, simular el comportamiento ante una pérdida masiva de datos a partir del instante 2 con distintos valores de la ganancia en el compensador, y elegir aquel valor que consiga un mínimo error entre la salida obtenida con esta técnica y la salida que se obtendría si no hubiesen pérdidas de datos. Los valores de la ganancia K deben ser pequeños, ya que la discrepancia que se trata de minimizar es debida al ruido del sistema.

Hay que tener en cuenta que la técnica de la compensación de la acción de control está pensada para la corrección del error a corto plazo. Por lo tanto una pérdida significativa de muchas muestras podría suponer problemas de inestabilidad al sistema. En cualquier caso, esta situación será el peor caso y por lo tanto es la que se ha utilizado para evaluar el valor de la ganancia más adecuado.

A continuación se muestra la evolución del índice de error para distintos valores de K cuando se utiliza un control predictivo GPC con $N=12$ y $Nu=8$:

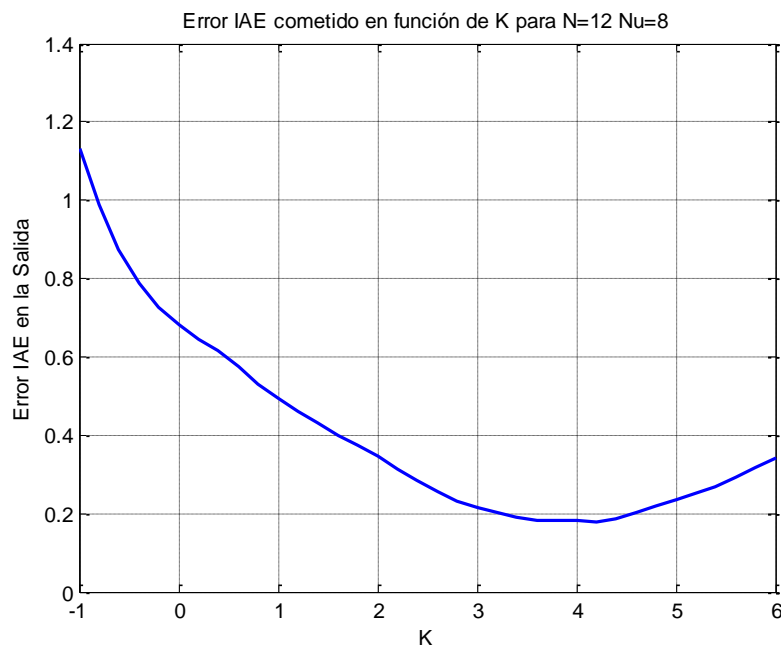


Figura 46. Errores cometidos en función de K para $N=12$ y $Nu=8$

La herramienta de simulación calcula analíticamente el valor de K que produce el mínimo índice de error. En este caso se consigue un IAE mínimo de 0.18 para el valor de $K=4.2$, aunque como se muestra en la figura anterior se puede tomar un valor de K entre 2.5 y 5.5, ya que el error en este rango está cercano al IAE mínimo.

Regulador predictivo CRHPC

Este regulador predictivo se calcula de manera que se obliga a que la salida del proceso coincida con la referencia durante un cierto intervalo (m) al final del horizonte de predicción. Con la estrategia de control utilizada, en la que se pueden utilizar los valores postulados para alcanzar la referencia, este regulador puede mejorar la respuesta de un GPC.

La simulación del sistema con el periodo de 50 ms y el regulador CRHPC se muestra en la siguiente figura:

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 10 \ 1 \ 0.1$; $T_s=0.05$; $K=0$

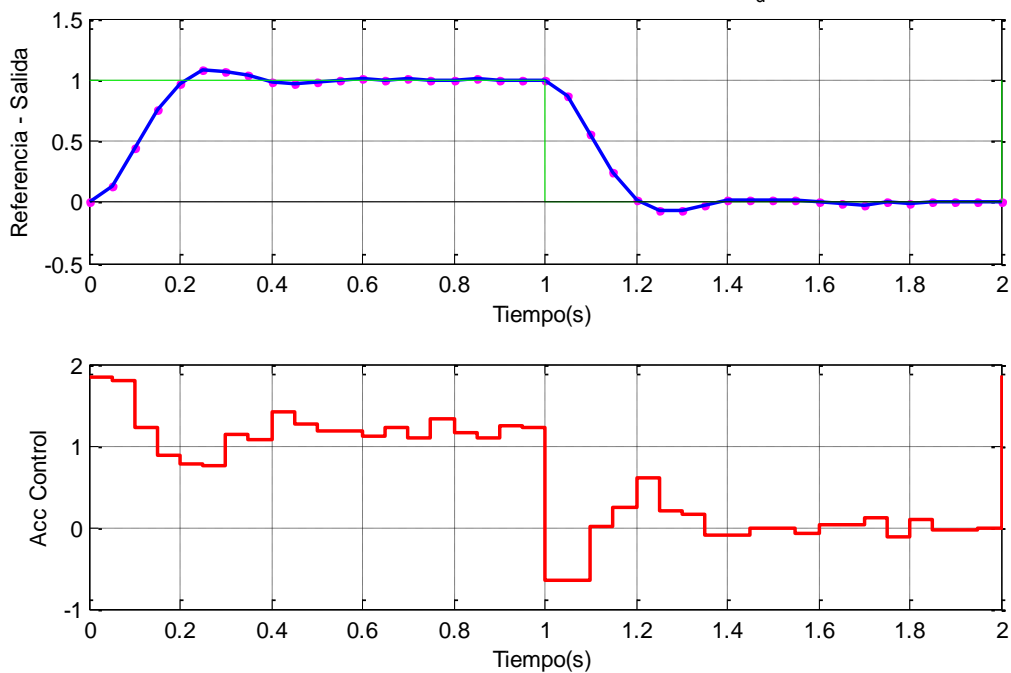


Figura 47. Proceso de segundo orden con el controlador CRHPC

A continuación se simula el comportamiento del sistema para cada horizonte de predicción $N+m$ con todas las combinaciones del horizonte de control N_u entre 2 y $N+m$. La Figura 48 muestra para cada $N+m$ el valor del horizonte de control N_u con el que se ha obtenido el índice IAE más bajo, así como el valor de éste.

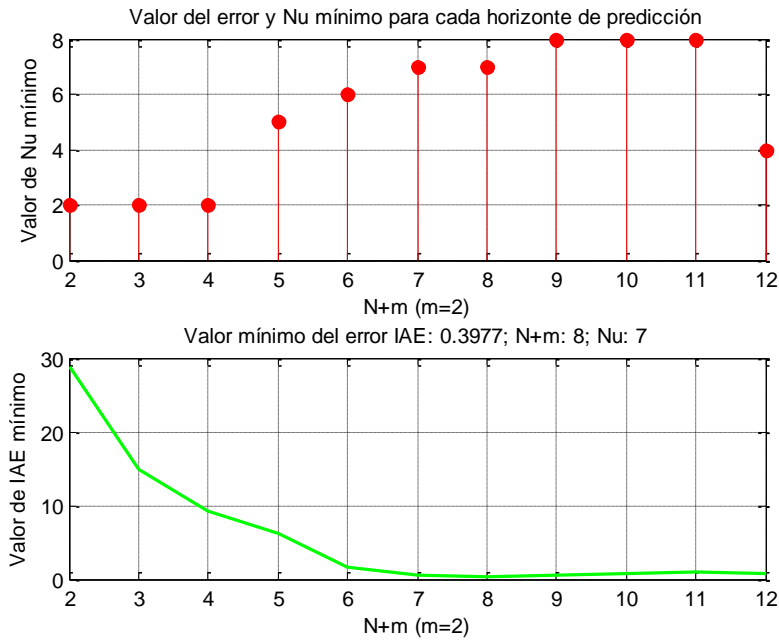


Figura 48. Estudio del valor de Nu en función de N+m con el que se obtiene el mínimo error

Como se ve en la cabecera de la figura anterior el horizonte de predicción que consigue el mínimo IAE es $N+m=8$ (con $m=2$), pero con ese horizonte no se llega a englobar a toda la respuesta dinámica, por lo que se toma $N+m=10$ instantes, con $m=2$, ya que de esta forma se engloba a la respuesta dinámica y el índice de error obtenido es similar al de la combinación mínima.

En este caso el horizonte de control que consigue el valor mínimo del error con $N+m=10$ es $Nu=8$. A continuación se muestra cómo afecta la elección del horizonte de control en el índice de error para el caso de $N+m=10$, para comprobar que efectivamente se obtiene el valor de $Nu=8$.

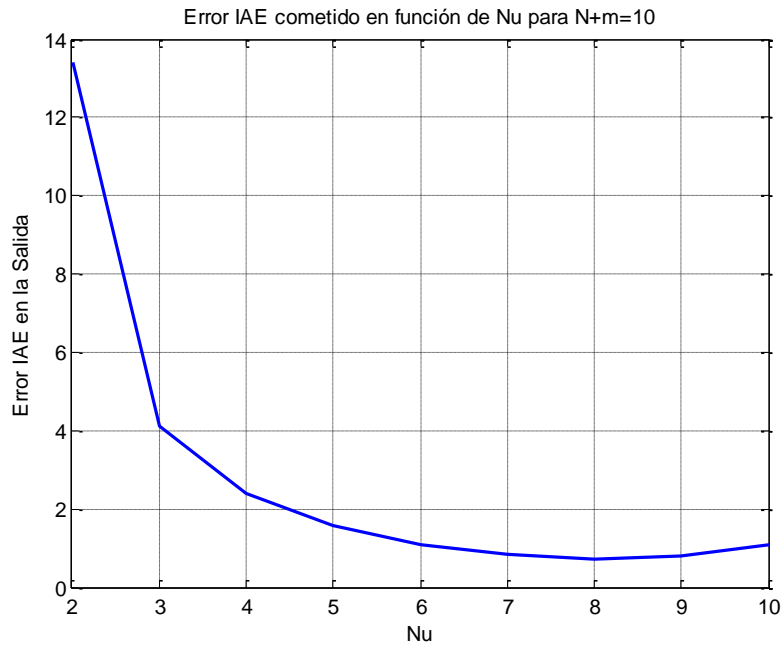


Figura 49. Errores cometidos en función de Nu para $N+m=10$, con $m=2$

Finalmente se simula el sistema con los parámetros anteriores ante una pérdida masiva de datos a partir del instante 2 para obtener el valor de la ganancia de la compensación de la acción de control.

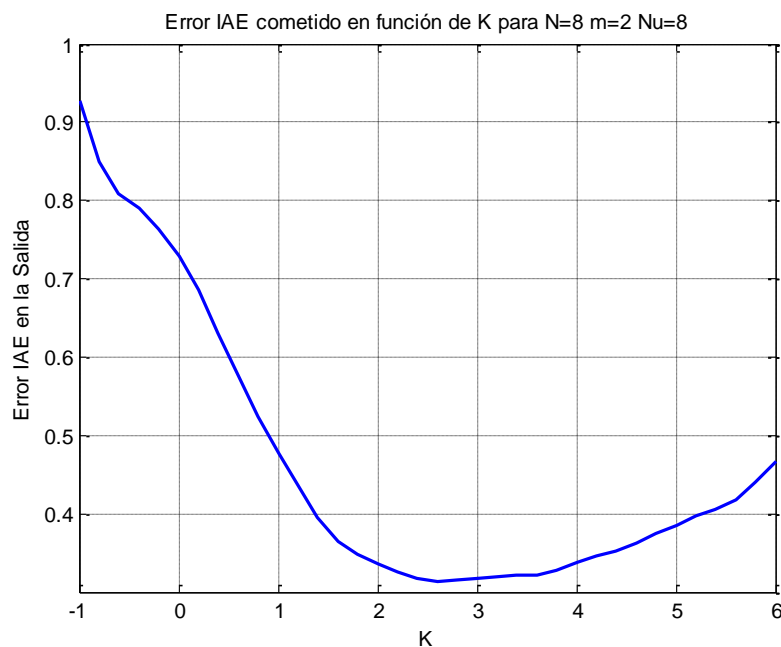


Figura 50. Errores cometidos en función de K para $N=8$, $m=2$ y $Nu=8$

Con $K=2.6$ se obtiene el mínimo valor del error $IAE=0.3137$. Como se muestra en la figura anterior se puede tomar un valor de K entre 1.5 y 4.5, ya que el error en este rango está cercano al IAE mínimo.

5.2.1.2 Comportamiento del sistema distribuido de control

Una vez elegidos los parámetros de los controladores del sistema distribuido, tanto del nodo supervisor como del nodo local, a continuación se muestra su funcionamiento ante distintas situaciones. Se compararán los modos de funcionamiento desde el 2 al 5, que son aquellos en los que las acciones de control son calculadas por el nodo supervisor mediante controladores de alto nivel. En el modo 1 se dispone de un controlador local sencillo, en lugar de recibir las acciones de control del nodo supervisor, y por lo tanto no es comparable al resto de modos. La descripción de los modos de funcionamiento se encuentra en el punto 4.5.6.

Las pruebas se realizarán utilizando un canal de comunicaciones exclusivo y un canal de comunicaciones compartido.

Caso 1. Sistema distribuido con canal de comunicaciones exclusivo

En este caso el nodo supervisor y el nodo local utilizan un canal de comunicaciones exclusivo, y por lo tanto no hay problemas en cuanto al ancho de banda dedicado a la transmisión de datos entre los nodos, sin embargo, se pueden producir pérdidas de datos debido a ruidos en el sistema, retardos o bien una mala interpretación de los mensajes recibidos.

La simulación del ruido que se introduce en la salida del proceso se ha tomado de media nula y desviación típica 5×10^{-5} , que es la medida registrada en las pruebas reales.

A continuación se muestran los resultados de los distintos índices de error para los distintos modos de funcionamiento (del 2 al 5):

GPC: N=12, Nu=8, K=4.2

La Figura 51 muestra los índices de error IAE cometidos en la salida del proceso para distintos porcentajes de pérdidas de datos, cuando se comparan con la salida en ausencia de pérdidas, para cada modo de funcionamiento. Se ha seguido el método propuesto en el apartado 5.1.3.

Por cada porcentaje de pérdida de datos se realizan 100 simulaciones distribuidas de forma aleatoria a lo largo de todo el tiempo de la prueba, ya que los errores cometidos dependen del instante en el que se produce la pérdida de datos. Esto supone la realización de 1900 simulaciones por cada modo de funcionamiento. Por lo tanto, se han realizado 7600 simulaciones para comparar los cuatro modos de funcionamiento que se muestran en la siguiente figura.

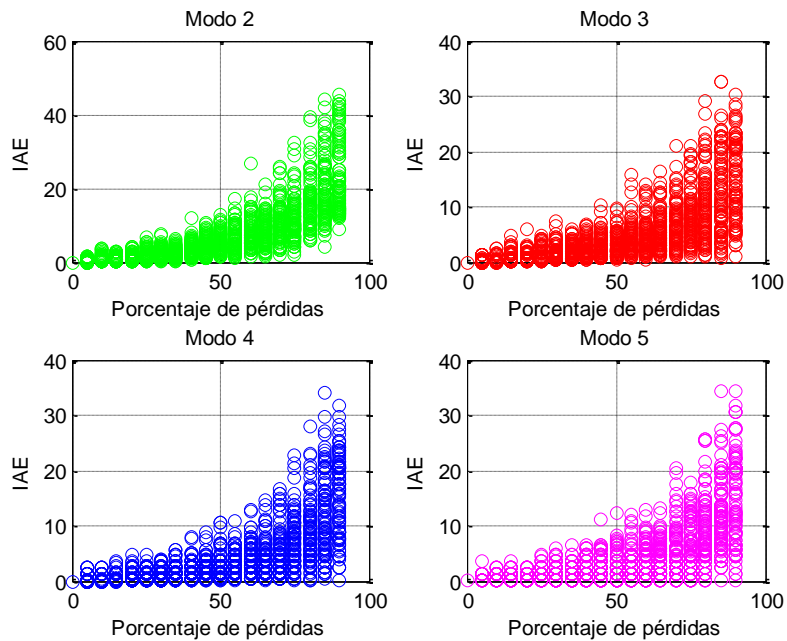


Figura 51. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

La figura anterior muestra como a medida que se producen más pérdidas los indicadores de error aumentan. A continuación se comparan los distintos modos de funcionamiento sobre la misma gráfica, tomando el índice de error más alto conseguido con cada porcentaje de pérdida de datos. Como se ha explicado anteriormente este criterio permite considerar el peor caso.

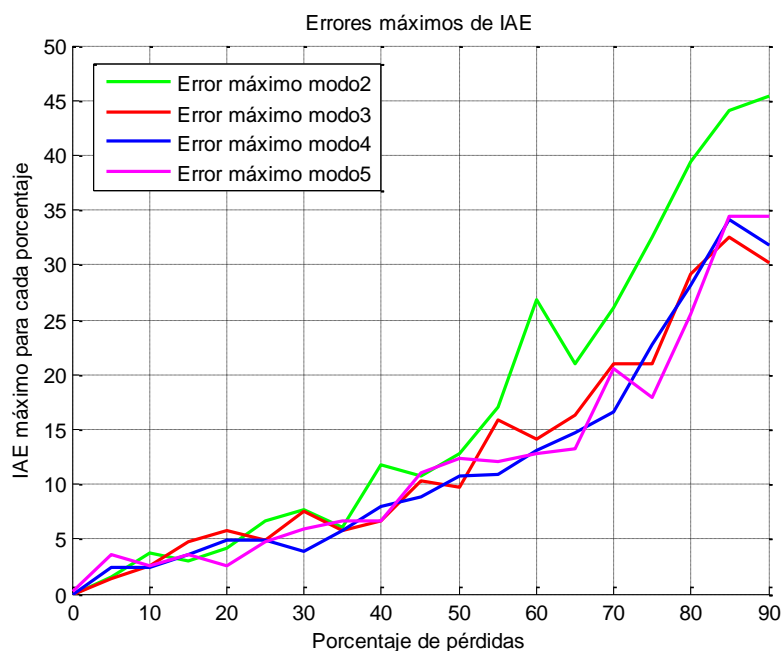


Figura 52. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

En la siguiente figura se muestran los errores medios con cada porcentaje de pérdidas. Los errores medios permiten estudiar la tendencia del error conforme aumenta el porcentaje de pérdidas.

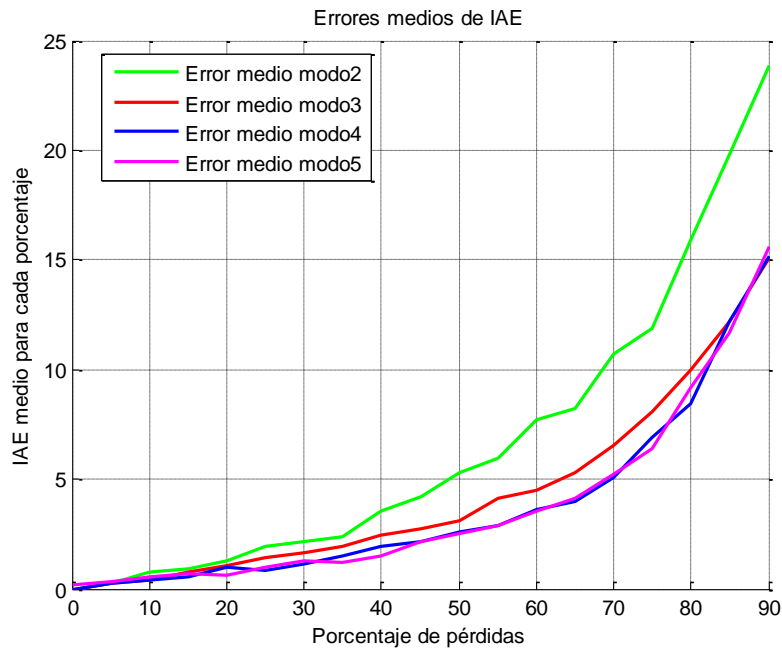


Figura 53. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

Las figuras anteriores muestran como los modos de funcionamiento que utilizan las salidas postuladas, son los que obtienen unos índices de error más bajos en cada porcentaje de pérdidas, sobre todo cuando se pierden más del 40% de los datos. Además, los modos de funcionamiento 4 y 5, que son los que aplican la compensación de la acción de control con el error cometido a la salida, son lo que obtienen unos índices de error más bajos.

El modo de funcionamiento 2, que es el que mantiene la acción de control aplicada en el instante anterior cuando se pierden los datos, es que el que obtiene el mayor índice de error, sobre todo para porcentajes de pérdidas grandes.

CRHPC: N=8, m=2, Nu=8, K=2.6

Para el caso del regulador predictivo CRHPC se han realizado la misma cantidad de simulaciones comentada anteriormente, siguiendo la metodología expuesta en 5.1.3. La siguiente figura muestra los índices de error obtenidos, con cada una de las pruebas realizadas, con los cuatro modos de funcionamiento estudiados.

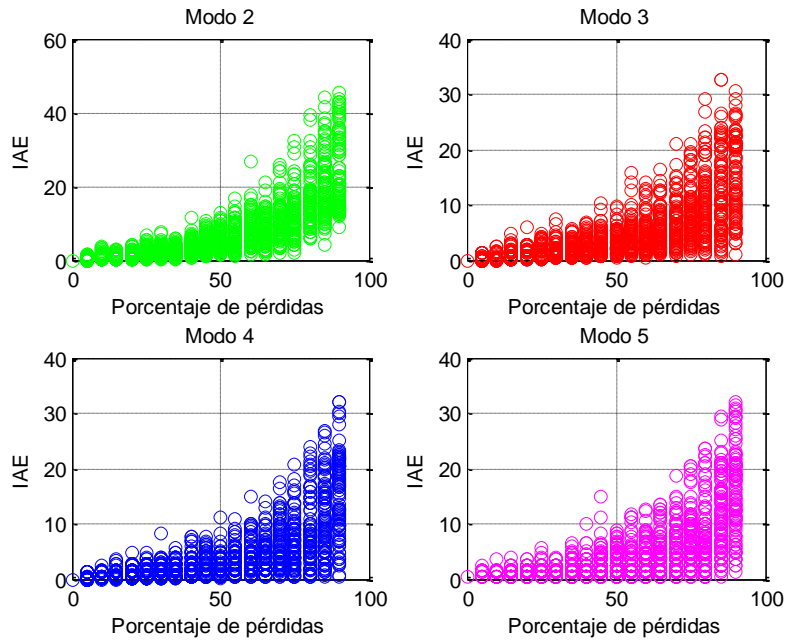


Figura 54. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

A continuación se comparan los 4 modos de funcionamiento tomando el mayor error cometido en cada porcentaje de pérdida de datos así como el error medio.

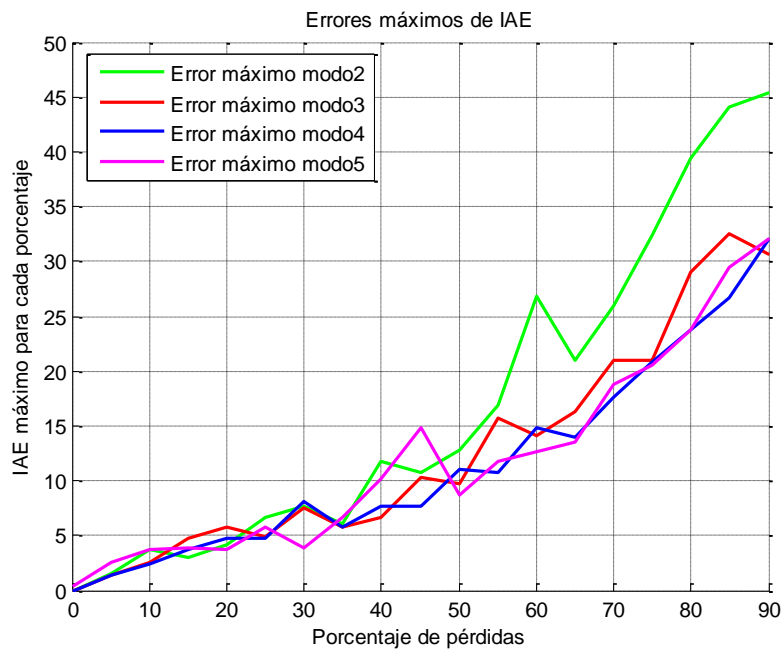


Figura 55. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

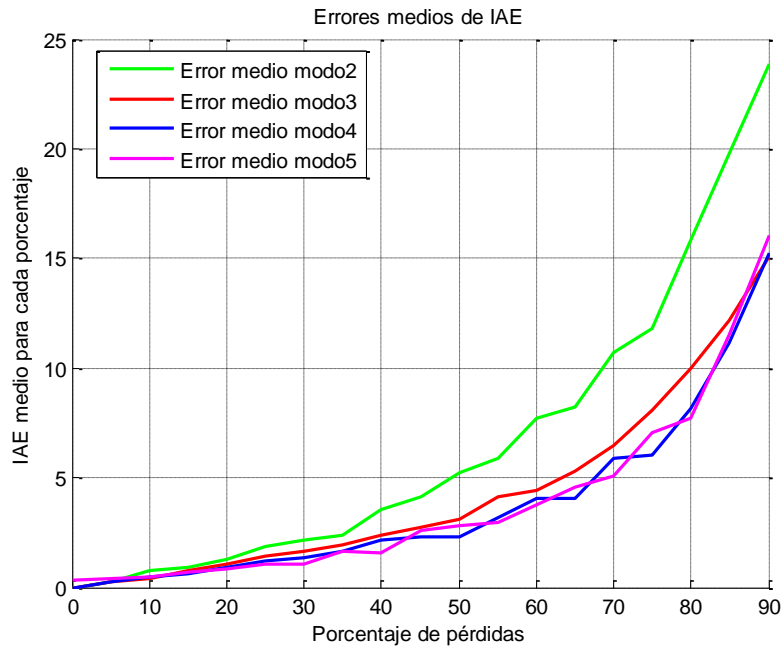


Figura 56. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

En este caso el comportamiento de los 4 modos de funcionamiento es similar al expuesto para el caso del GPC. Los que consiguen un índice de error más pequeño son los modos de funcionamiento 4 y 5, y el que mayor error presenta es el modo 2.

Comparativamente, este regulador CRHPC con el GPC expuesto anteriormente no consigue una mejora significativa en los índices de error obtenidos.

Caso 2. Sistema distribuido con canal de comunicaciones compartido

En este caso el canal de comunicaciones no es de uso exclusivo, y por lo tanto el ancho de banda dedicado al envío de los mensajes es un factor determinante en la calidad de la respuesta del proceso bajo control.

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} , es decir, la misma que en la prueba anterior. Además se ha introducido una interferencia en la red que consume un 10% del ancho de banda y en este caso se tiene en cuenta un coste computacional del regulador predictivo de $500 \mu\text{s}$.

En el apartado 5.3.2 y 5.3.3 se estudiará la influencia del ancho de banda y del tiempo de cómputo, respectivamente, en la respuesta obtenida por el sistema.

GPC: N=12, Nu=8, K=4.2

A continuación se muestran los índices de error cometidos en cada una de las 1900 pruebas realizadas por cada modo de funcionamiento, en el que para cada porcentaje

de pérdida de datos se realizan 100 simulaciones con una distribución aleatoria del instante en el que se producirá la pérdida.

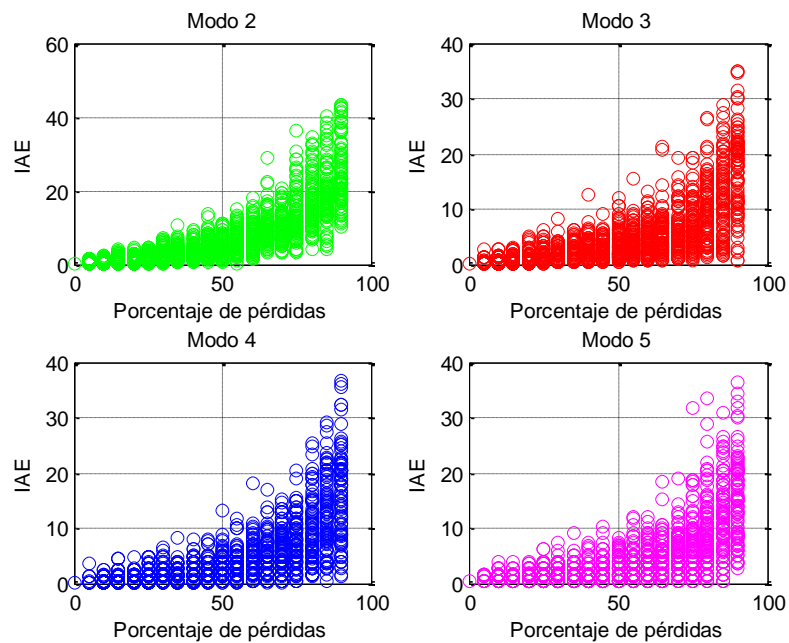


Figura 57. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

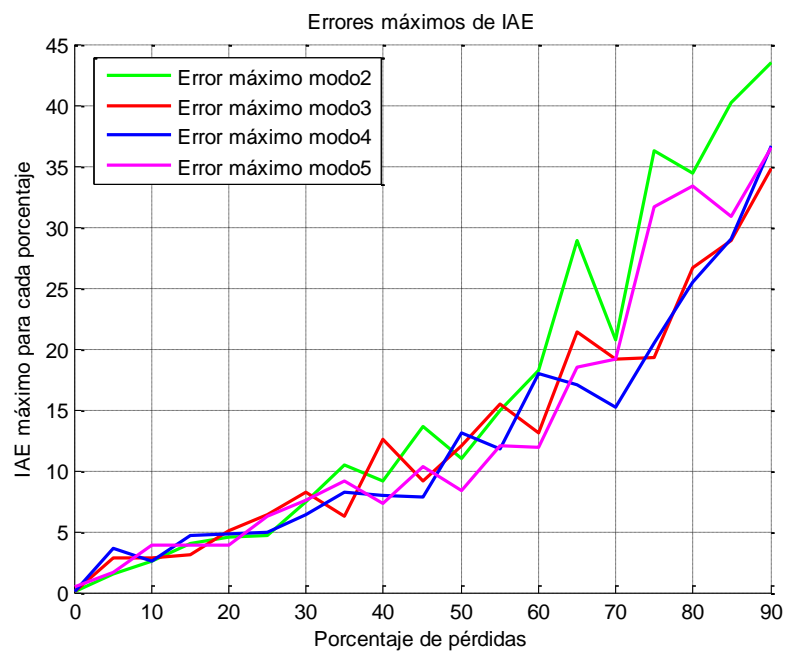


Figura 58. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

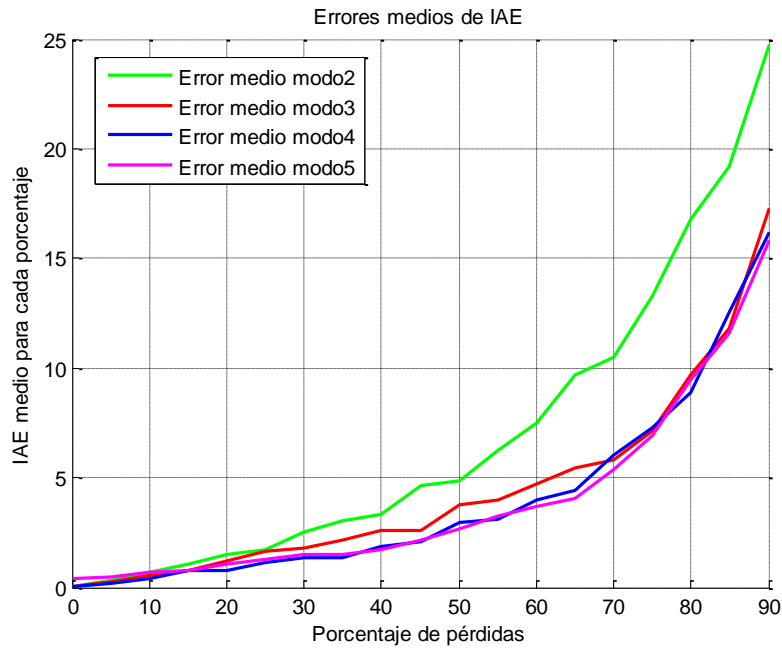


Figura 59. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

El comportamiento de los distintos modos de funcionamiento es similar al expuesto en el caso del canal de comunicaciones exclusivo, y además los índices obtenidos en cada modo de funcionamiento por cada porcentaje de pérdida de datos también son similares. Por lo tanto, una utilización del 90% del ancho de banda en las comunicaciones como el que se ha tomado para esta simulación no empeora los resultados obtenidos. En el apartado 5.3.2 se estudiará la influencia del ancho de banda y cuál debe ser el ancho de banda mínimo dedicado a las comunicaciones entre los nodos para no degradar la respuesta del proceso.

CRHPC: N=8, m=2, Nu=8, K=2.6

En las siguientes figuras se muestran los índices de error cometidos en el caso de utilizar el regulador CRHPC en el nodo supervisor.

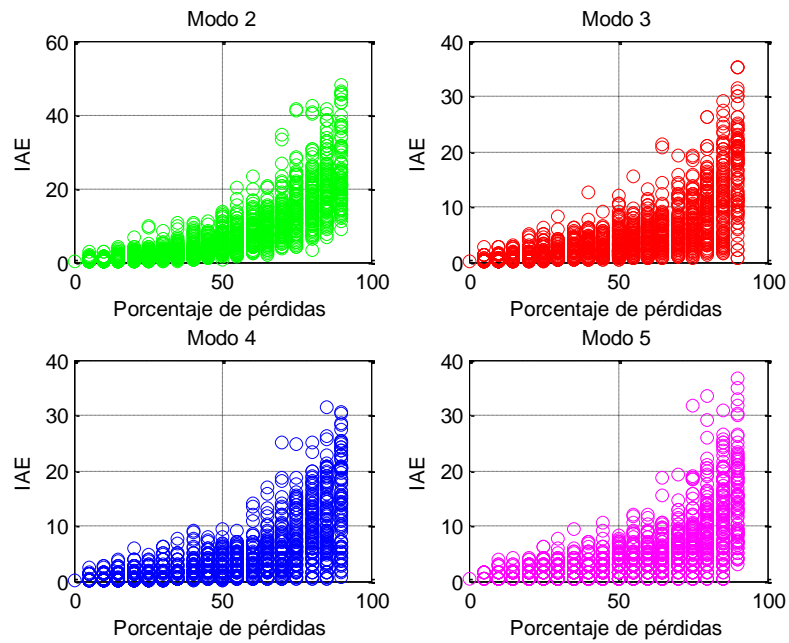


Figura 60. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

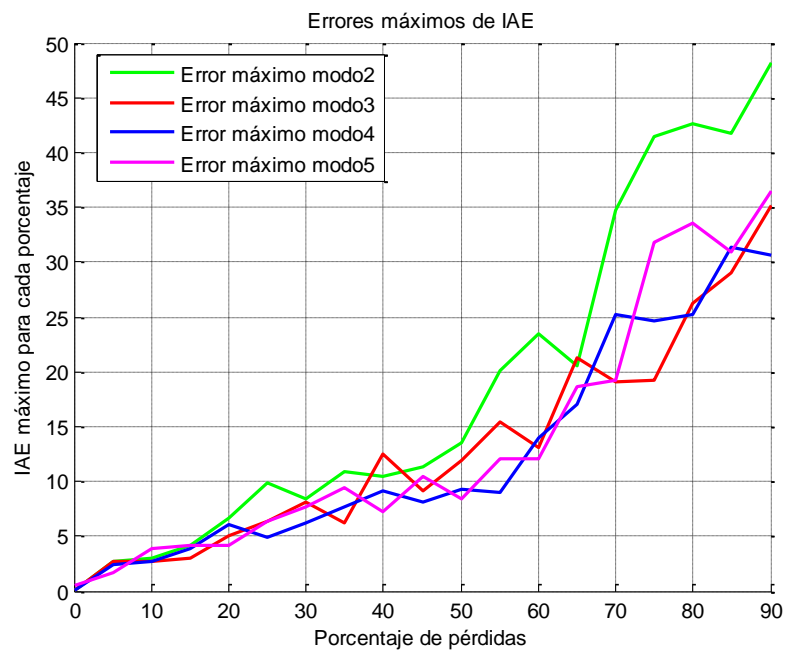


Figura 61. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

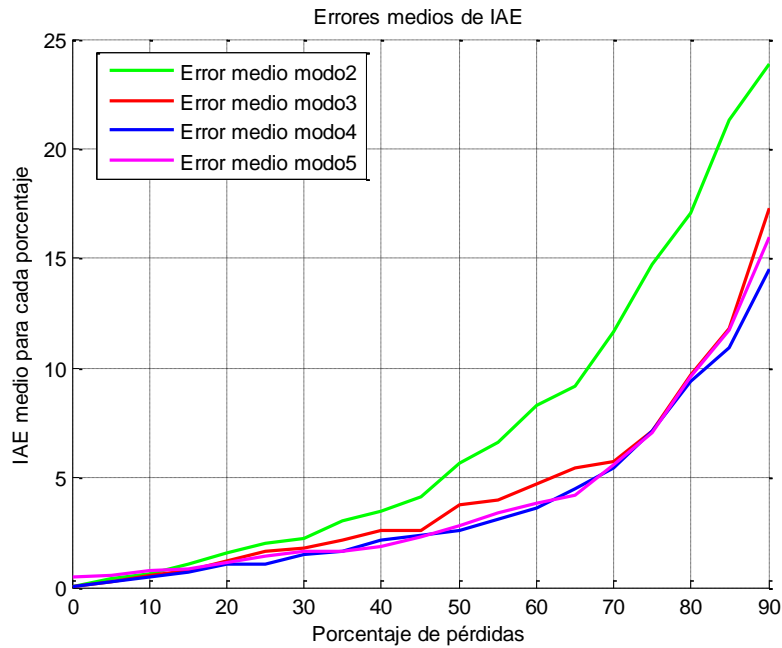


Figura 62. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

Igual que ocurría en el caso del canal de comunicaciones exclusivo, no hay una gran diferencia en los errores cometidos por el regulador predictivo CRHPC con respecto al regulador predictivo GPC estudiado anteriormente.

5.2.2 Comparación de las distintas pruebas para los procesos en estudio

A continuación se realizará la comparación de la implementación de los distintos modos de funcionamiento, con los reguladores predictivos GPC y CRHPC, para los distintos procesos SISO con dinámicas diferentes que se han presentado en la introducción del apartado 5.2. Las pruebas realizadas para el caso del proceso de primer orden, de fase no mínima, con retardo y con efecto integral se encuentran en el anexo B.

En la Tabla 5 se resumen los parámetros obtenidos para los procesos sometidos a prueba, tanto con un control supervisor GPC como CRHPC, utilizando la metodología explicada en el apartado 4.6.

	Ts (ms)	GPC	CRHPC
Proceso de 1er orden	100	N=8, Nu=6 K _{min} =2.2 (IAE=0.2246) (K entre 0.5 y 3)	N+m=8 (m=2), Nu=4 K _{min} =2.2 (IAE=0.2798) (K entre 0.5 y 3)
Proceso de 2º orden	50	N=12, Nu=8 K _{min} =4.2 (IAE=0.18) (K entre 2.5 y 5.5)	N+m=10 (m=2), Nu=8 K _{min} =2.6 (IAE=0.3137) (K entre 1.5 y 4.5)
Proceso de fase no mínima	80	N=14, Nu=6 K _{min} =-0.6 (IAE=3.9) (K entre -1 y 0.5)	N+m=14 (m=2), Nu=10 K _{min} =-0.6 (IAE=3.204) (K entre -1 y 0.5)
Proceso con retardo	1000	N=15, Nu=6 K _{min} =1.2 (IAE=0.8273) (K entre 0 y 2.5)	N+m=15 (m=2), Nu=5 K _{min} =1.2 (IAE=0.9727) (K entre 0 y 2.5)
Proceso con efecto integral	100	N=12, Nu=3 K _{min} =6 (IAE=1.183) (K entre 0.5 y 6)	N+m=8 (m=2), Nu=6 K _{min} =4 (IAE=0.6576) (K entre 0.5 y 6)

Tabla 5. Parámetros del control supervisor y del control local

En la tabla se indica el valor de la ganancia de compensación con el que se obtiene el mínimo índice de error, pero también se indica un rango de valores de K con el que el índice de error es bajo.

Para cada uno de los controles supervisores GPC y CRHPC se han comparado las prestaciones obtenidas en cada modo de funcionamiento, y con dos canales de comunicaciones, uno de uso exclusivo, entre el control local y el control supervisor, y otro compartido.

En las siguientes tablas se muestra un resumen de los valores obtenidos en el índice del error para distintos porcentajes de pérdidas. El valor que se muestra en cada porcentaje es el índice de error más alto de las simulaciones realizadas con el mismo porcentaje de pérdidas de datos, con una distribución aleatoria del instante en el que se producirá la pérdida.

	GPC					CRHPC				
	10%	20%	40%	60%	80%	10%	20%	40%	60%	80%
Proceso de 1er orden	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	1.32051	2.4050	4.0516	34.7893	673.691	0.7567	4.9553	14.5289	100.696	2822.26
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	2.1790	2.3361	5.8252	5.0646	8.0240	2.2621	1.1722	2.4861	9.1112	20.4511
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
2.1181	2.1165	4.2203	6.2463	12.1967	1.0868	1.0681	4.1820	13.2250	19.0933	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:
3.2027	1.2587	7.1930	9.2860	15.3606	0.2798	2.2908	6.3383	5.3626	21.3713	
Proceso de 2º orden	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	3.7975	4.2387	11.7748	26.8537	39.4260	3.7809	4.2238	11.7108	26.8278	39.4132
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	2.5873	5.8324	6.7173	14.1371	29.2111	2.5771	5.8200	6.7134	14.1096	29.0894
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
2.4476	4.8625	7.9196	13.0383	28.1112	2.4241	4.7624	7.7083	14.9069	23.8118	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:
2.4945	2.4945	6.6596	12.8597	25.5790	3.7486	3.7339	10.0884	12.6438	23.6955	
Proceso de fase no mínima	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	4.5118	7.6147	16.7518	28.0472	171.766	2.9855	8.6118	12.4827	21.1970	138.254
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	3.8023	6.1146	9.7010	12.3566	21.7901	5.1179	5.2668	8.9633	16.0934	27.6544
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
7.5155	5.1141	8.4705	12.4393	27.3564	2.3839	6.3545	6.2995	16.3900	25.0004	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:
5.5442	6.9904	6.9530	10.8259	25.4028	5.2584	9.4151	8.0600	8.9148	24.3361	
Proceso con retardo	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	1.6848	1.8970	5.2431	12.5561	37.7650	1.6766	2.8794	9.8395	13.9388	25.8087
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	1.6029	2.673	4.4770	12.4697	18.8778	2.4391	2.4879	3.7323	5.1176	12.8018
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
2.5758	2.6747	6.1334	9.2041	18.5181	2.5340	1.5263	3.8069	5.8872	10.7670	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:
0.8273	3.0406	5.3267	10.3703	26.8645	1.8837	4.0050	4.0206	7.8805	11.6397	
Proceso con efecto integral	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	3.1884	3.3308	8.3614	29.3162	968.6107	1.5091	2.8606	9.6319	47.9945	432.448
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	1.5505	4.4506	9.0442	9.3955	20.7900	1.6189	4.7006	4.0700	10.6794	30.6203
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
1.1480	1.3110	3.5392	7.6877	12.7212	1.2183	2.5593	4.9514	10.5989	11.8256	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:
2.2460	3.1308	6.1636	8.5358	15.5822	1.6194	2.6125	5.8623	6.2026	21.4877	

Tabla 6. Índices de error IAE máximos obtenidos por porcentaje de pérdidas con canal de comunicaciones de uso exclusivo

Para las pruebas realizadas con los procesos de distinta dinámica se obtienen resultados similares a los expuestos para el caso del proceso de 2º orden:

- Los modos de funcionamiento 3, 4 y 5, que son los que utilizan las trayectorias postuladas en caso de pérdida de datos, obtienen unos índices de error más pequeños que el modo 2.
- La compensación de la acción de control mejora la respuesta del sistema, sobre todo cuando hay perturbaciones y ruidos no considerados en el cálculo de las trayectorias.
- Donde se consigue una mejora significativa de la utilización de las trayectorias postuladas con la compensación de la acción de control es en el

control de aquellos procesos que tienen una respuesta inestable en bucle abierto, como es el caso del proceso con efecto integral.

- El modo 5, que es el que utiliza las trayectorias postuladas hasta el cambio de referencia, consigue un buen resultado. Esto puede ser debido a que los demás modos conmutan entre dos tipos de control cuando pierden datos, mientras que el modo 5 utiliza solo las trayectorias postuladas y la compensación de la acción de control. Esta falta de conmutación hace que la salida obtenida sea más progresiva y consiga un índice de error bajo. Además, la pérdida de datos en el modo 5 afectará tan sólo a los instantes en los que el control supervisor debe enviar la actualización de las trayectorias al control local, y no en cada instante, como ocurre con el resto de modos de funcionamiento.

	GPC					CRHPC				
	10%	20%	40%	60%	80%	10%	20%	40%	60%	80%
Proceso de 1er orden	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	3.5131	7.0944	17.5627	250.223	438.351	6.0930	16.2968	24.7992	367.135	530.567
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	4.0025	4.5577	8.4032	16.6806	19.0903	3.9006	5.1437	8.1198	10.8846	37.4410
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
3.7622	3.0852	7.6096	11.7430	15.2796	4.0329	4.9367	5.4507	9.5943	30.3661	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:
2.5345	2.4222	5.2860	5.9520	14.4714	2.1832	4.5420	6.8714	6.6192	23.1448	
Proceso de 2º orden	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	2.5532	4.5747	9.1672	18.2387	34.4470	2.9938	6.6522	10.4711	23.4603	42.5580
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	2.7822	5.0566	12.5424	13.0852	26.6151	2.7696	5.0363	12.5200	13.0853	26.3137
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
2.6000	4.7775	7.9582	18.0133	25.4386	2.7533	6.0555	9.1182	13.9655	25.1775	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	
3.8292	3.8307	7.2797	11.9690	33.3887	3.8639	4.1935	7.2099	12.0173	33.5049	
Proceso de fase no mínima	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	4.7919	11.9659	11.2756	52.5931	114.6517	8.7227	8.4954	12.3596	39.6297	117.026
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	4.1224	4.6901	13.4696	13.1424	22.9053	7.6408	4.9346	13.8568	17.9670	22.2401
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
4.5024	7.1182	10.6584	13.8512	27.0747	4.5887	8.3881	12.0423	14.5629	22.9254	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	
5.6665	9.0258	9.5050	16.8646	27.8724	4.5371	7.8102	12.4131	13.3485	22.9709	
Proceso con retardo	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	1.4324	8.0133	4.9766	13.0983	39.2120	2.2020	2.3440	4.3987	18.4505	35.0491
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	1.1937	2.6444	6.9810	8.1117	24.6367	1.3332	4.4635	2.5612	11.2738	23.3560
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
1.3322	1.3650	4.6985	11.6530	11.5522	1.1947	2.3882	4.6600	8.1389	28.3845	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	
1.5855	2.0823	4.6890	10.7617	12.6358	3.067	2.3298	4.3687	5.3897	12.8731	
Proceso con efecto integral	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:	Modo2:
	1.4163	2.4508	5.9055	55.2927	128.0911	2.9954	6.2001	23.0150	16.1623	112.543
	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:	Modo3:
	1.2140	3.1148	2.9844	9.8174	31.6780	0.8315	2.8518	8.7560	8.1212	15.4689
	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:	Modo4:
1.2031	2.3342	6.3351	5.6541	20.3107	1.1575	4.5927	6.7497	9.9027	18.6083	
Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	Modo5:	
1.2906	3.2490	2.3904	14.9902	27.8590	1.8758	2.8467	6.3181	12.7774	15.5132	

Tabla 7. Índices de error IAE máximos obtenidos por porcentaje de pérdidas con canal de comunicaciones compartido

La tabla anterior muestra los resultados obtenidos cuando el canal de comunicaciones es compartido, las conclusiones de las pruebas son las mismas que el caso del canal de comunicaciones exclusivo y, tal y como se ha expuesto en el caso del sistema de segundo orden, los índices de error obtenidos son similares en ambos casos. Por lo tanto, el valor del ancho de banda que se ha tomado dedicado a las comunicaciones de los nodos del sistema distribuido no es crítico. En el apartado 5.3.2 se estudiará la influencia de ese ancho de banda dedicado a las comunicaciones de control y cuál es el valor mínimo que debe tomarse para no tener problemas.

Una conclusión importante, que también muestran estas pruebas, es que no hay una diferencia significativa en la utilización del control supervisor GPC y el CRHPC, ambos muestran un comportamiento similar.

5.3 Influencia de las condiciones del entorno en el funcionamiento del sistema distribuido de control

En las pruebas anteriores se ha mostrado el método para elegir los parámetros del control en el sistema distribuido. En el funcionamiento del sistema influyen las condiciones del entorno y los parámetros del control en las prestaciones conseguidas. En este apartado se muestra la influencia, tanto del entorno como los parámetros del control, en el funcionamiento del sistema.

Las condiciones que se van a evaluar para mostrar su influencia en las prestaciones del sistema son:

- Discrepancia del modelo respecto al proceso real
- Ancho de banda consumido
- Tiempo de cómputo del controlador
- Ruido en la salida del sistema

Para mostrar la influencia de todos los factores anteriores las pruebas se van a realizar sobre el proceso de segundo orden subamortiguado que ha sido presentado en el apartado 5.2.1, en el que la comunicación entre el nodo supervisor, que implementa un control predictivo CRHPC, y el nodo local se hace a través de un canal compartido.

La Figura 135, correspondiente al anexo A, muestra la estructura del sistema distribuido con el canal de comunicación compartido que ha sido utilizado por la herramienta simulación para la realización de las pruebas.

5.3.1 Discrepancia del modelo respecto al proceso real

Tal y como se ha comentado en el apartado 4.4.1.3 el modelo del proceso utilizado para el diseño de la estrategia de control es una aproximación matemática del sistema real, y por lo tanto siempre existirán errores de modelado debido a la simplicidad del modelo considerado y a las deficiencias en la estimación de parámetros.

Por ello, en todas las simulaciones realizadas en el apartado 5.2 el modelo del proceso utilizado para el diseño del regulador y el cálculo de los parámetros es un poco distinto del utilizado en la simulación. De esta manera en las simulaciones se tiene en cuenta la discrepancia que siempre existe entre el modelo y el proceso real. En concreto, el modelo del proceso de segundo orden utilizado en el diseño del sistema distribuido de control, mostrado en 5.2.1.2, se ha realizado a partir de la ecuaciones básicas de la física, obtenido la función de transferencia $G_{p0}(s)$ que aparece en (5.8), mientras que para la simulación se ha tomado el modelo del proceso realizado a partir de los datos experimentales con un proceso real, obteniendo la función de transferencia $G_{p1}(s)$ que aparece en (5.9).

En este apartado se va a mostrar la influencia de la discrepancia entre el modelo del proceso tomado para la realización del sistema de control, con respecto al proceso real al que finalmente se aplica, y de esta forma mostrar la robustez de la estrategia de control.

Para ello, se va a simular el sistema distribuido poniendo como proceso real modificaciones del modelo utilizado para la realización del diseño de control. Se ha tomado el caso del sistema de comunicaciones compartido cuando se utiliza el regulador predictivo CRHPC en el nodo supervisor, para el proceso de segundo orden.

El modelo del proceso utilizado para el diseño de la estrategia de control es:

$$G_{p0}(s) = \frac{U_s}{U_e} = \frac{6.818}{0.1021 \cdot s^2 + 0.9588 \cdot s + 7.818} \quad (5.8)$$

A continuación se presentan los modelos de los procesos utilizados para realizar las simulaciones:

$$G_{p1}(s) = \frac{U_s}{U_e} = \frac{6.75}{0.1 \cdot s^2 + 0.9 \cdot s + 8} \quad (5.9)$$

$$G_{p2}(s) = \frac{U_s}{U_e} = \frac{7}{0.11 \cdot s^2 + 0.9 \cdot s + 7.5} \quad (5.10)$$

$$G_{p3}(s) = \frac{U_s}{U_e} = \frac{6.75}{0.09 \cdot s^2 + 0.96 \cdot s + 7.5} \quad (5.11)$$

$$G_{p4}(s) = \frac{U_s}{U_e} = \frac{7.2}{0.1 \cdot s^2 + 0.96 \cdot s + 8.5} \quad (5.12)$$

$$G_{p5}(s) = \frac{U_s}{U_e} = \frac{6.9}{0.1 \cdot s^2 + s + 8} \quad (5.13)$$

Las siguientes figuras muestran los errores máximos cometidos cuando se utilizan los modelos anteriores en la simulación del sistema, para cada uno de los modos de funcionamiento del sistema distribuido de control. Para ello se ha aplicado la metodología expuesta en 5.1.3, en la que por cada porcentaje de pérdida de datos se realizan 10 simulaciones distribuidas de forma aleatoria a lo largo de todo el tiempo de la prueba, y se representa el valor máximo. Esto supone la realización de 1140 simulaciones por cada modo de funcionamiento.

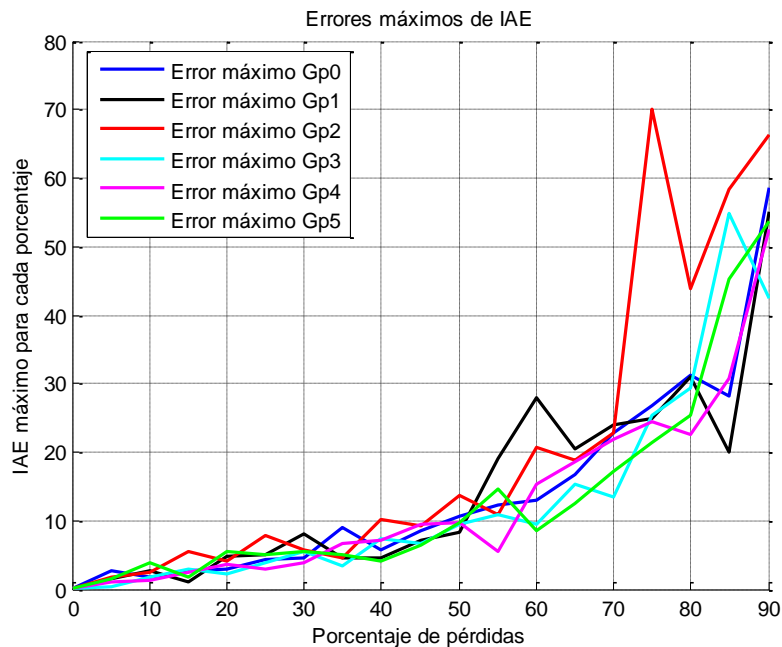


Figura 63. IAE máximos para cada porcentaje de pérdida de datos cometidos por los distintos procesos en el modo de funcionamiento 2

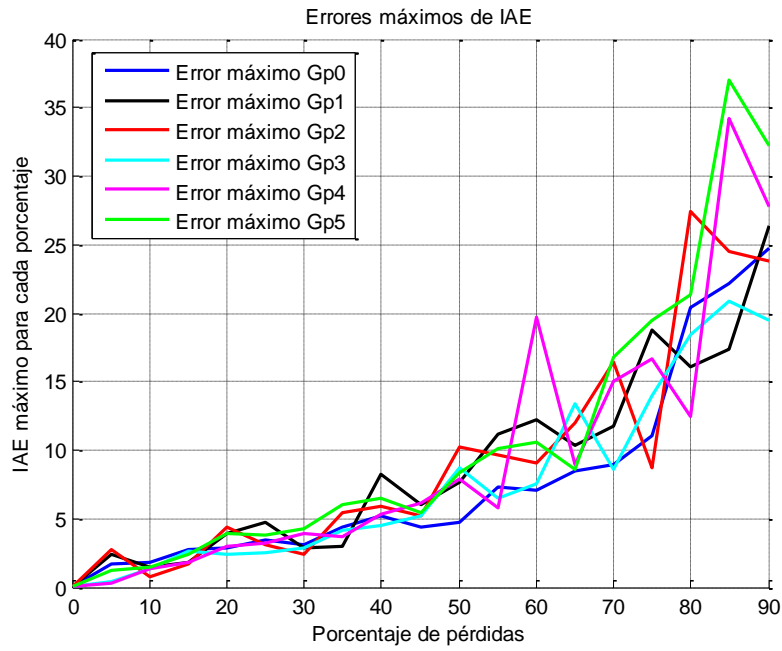


Figura 64. IAE máximos para cada porcentaje de pérdida de datos cometidos por los distintos procesos en el modo de funcionamiento 3

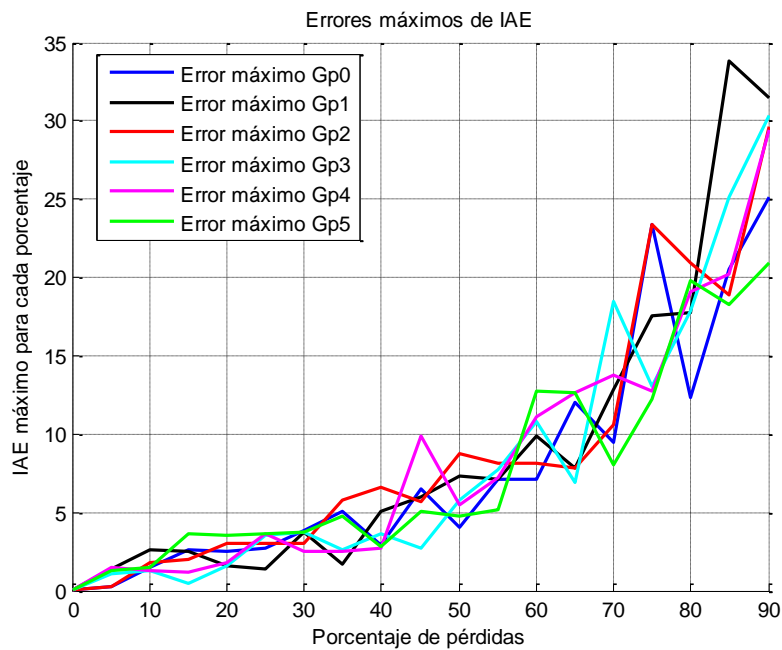


Figura 65. IAE máximos para cada porcentaje de pérdida de datos cometidos por los distintos procesos en el modo de funcionamiento 4

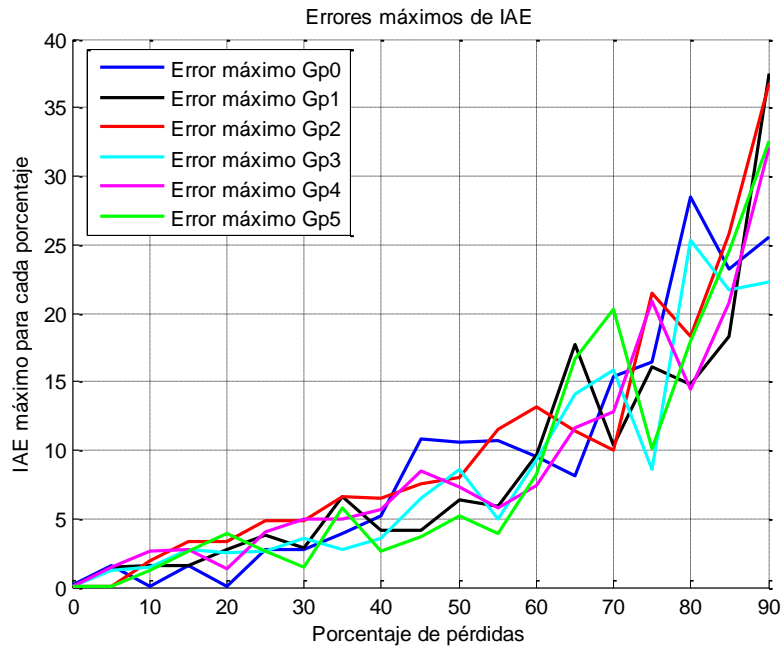


Figura 66. IAE máximos para cada porcentaje de pérdida de datos cometidos por los distintos procesos en el modo de funcionamiento 5

Las figuras anteriores muestran como a medida que aumenta el porcentaje de pérdida de datos, la dispersión de los errores máximos cometidos con cada modelo va creciendo. Sin embargo, no hay una gran diferencia entre los errores cometidos con cada proceso, para los distintos modos, para pérdidas de datos por debajo del 50%.

Además, se mantiene la tendencia de que los modos de funcionamiento que utilizan las trayectorias predichas por el control supervisor son los que consiguen los errores más bajos, siendo el modo 4 ligeramente el que mejores resultados consigue.

5.3.2 Ancho de banda consumido

En los sistemas distribuidos en los que el canal de comunicaciones está compartido la influencia del ancho de banda dedicado a las tareas de control respecto de la calidad de la respuesta es muy importante. El porcentaje de datos perdidos, y sobre todo los retrasos en el envío y la recepción de los mismos, se acentúan a medida que se pierde ancho de banda dedicado a las tareas de control.

Mediante la herramienta de simulación se puede estudiar la sensibilidad del sistema de control distribuido respecto al ancho de banda. La experiencia consiste en ir variando el ancho de banda dedicado a las tareas de control, y estudiar cual será el comportamiento del sistema así como la evolución de las tareas en la planificación del canal de comunicaciones. En este caso no se ha utilizado la compensación de la acción de control ni el tratamiento de retrasos que han sido expuestos en los apartados 4.5.5 y 4.5.7, respectivamente, para de esta forma no influir sobre la prueba.

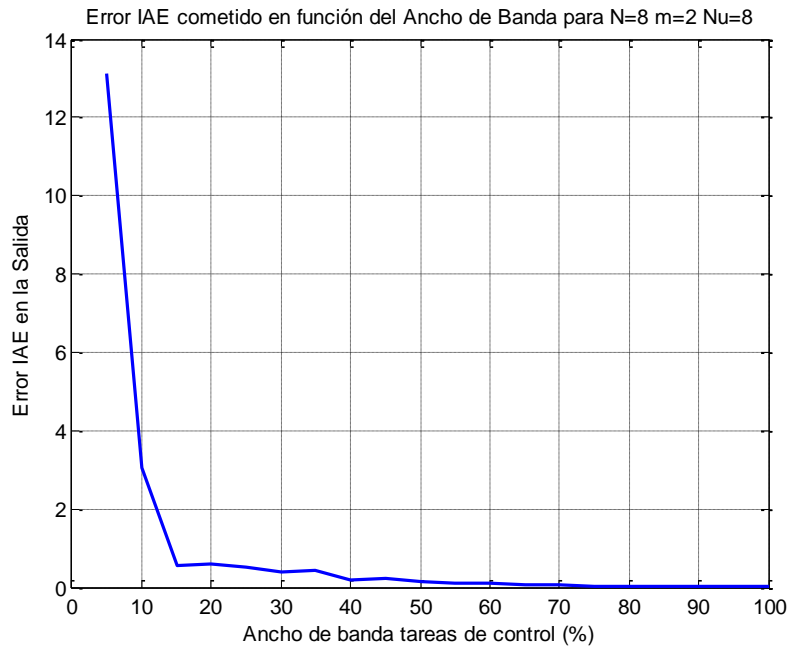


Figura 67. Errores cometidos en función del ancho de banda dedicada a las tareas de control para $N=8$, $m=2$ y $Nu=8$

La figura anterior muestra la influencia del ancho de banda dedicado a las tareas de control respecto al error cometido. Se puede ver como a medida que se le dedica menos ancho de banda a las tareas de control los errores cometidos aumentan, llegando a hacer al sistema inestable a partir de un cierto ancho de banda. Este ancho de banda mínimo nos permite especificar a priori cual debe ser el porcentaje dedicado a las tareas de control en las comunicaciones para no tener problemas. Sin embargo, en caso de disminuir este porcentaje, la estrategia de control puede hacer frente a estas anomalías, tal y como se ha descrito en el apartado 4.5.

En las siguientes figuras se muestra la planificación de las tareas dedicadas al control, por parte del canal de comunicaciones compartido, así como la evolución de la salida del proceso y de la acción de control cuando el ancho de banda dedicado a las tareas de control es de un 90%:

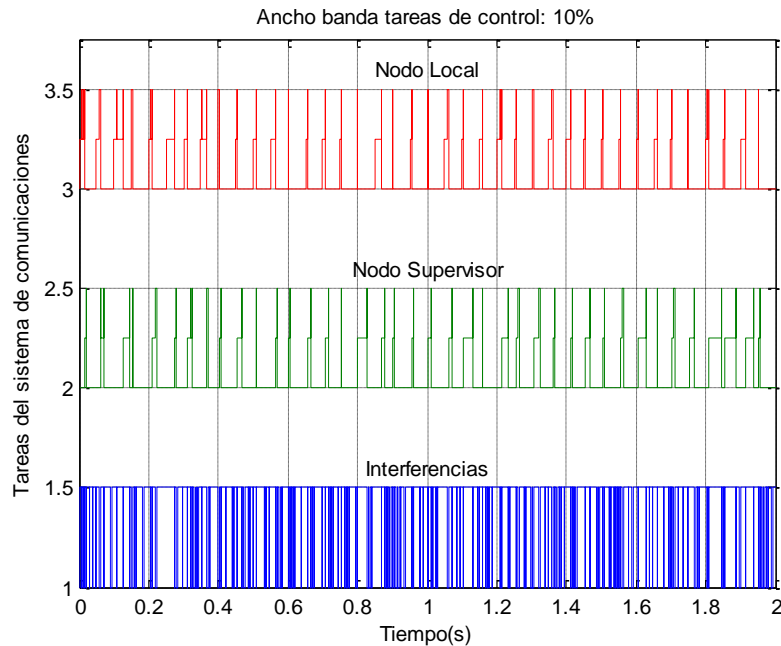


Figura 70. Activación de las tareas dedicadas al control en cada nodo para un ancho de banda del 10%

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 7 \ 1 \ 0.1$; $T_s=0.05$; $K=4.2$

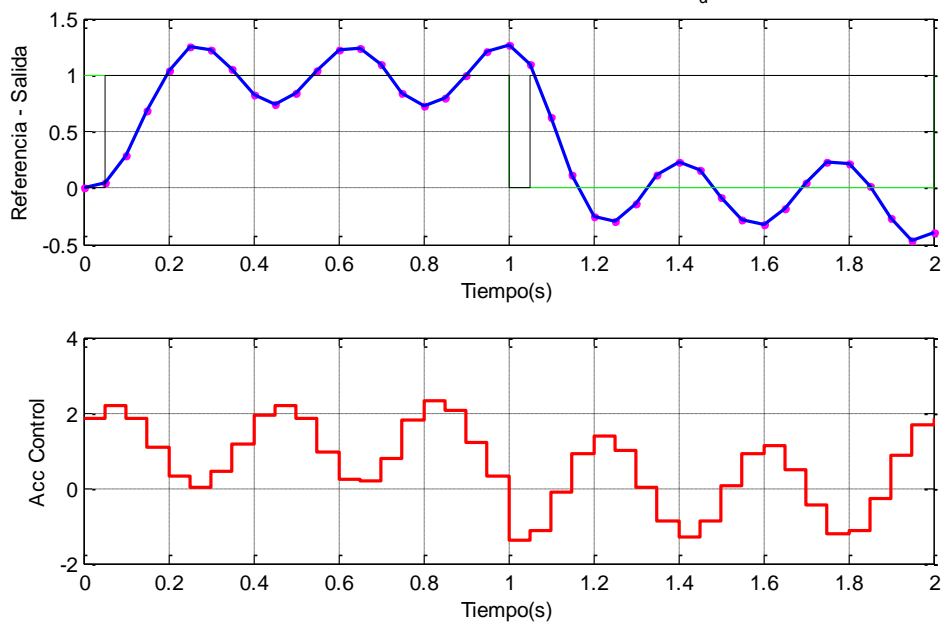


Figura 71. Comportamiento de la salida del proceso y la acción de control aplicada para un ancho de banda del 10%

Se puede observar como a medida que disminuye el ancho de banda dedicado a las tareas de control el comportamiento de la salida del sistema se degrada, incluso llegando a hacerse inestable. En el planificador se muestra como se produce un aumento de los tiempos dedicados a las tareas de control del nodo supervisor y local, lo que se traduce en desfases y retrasos en los datos enviados.

En este caso la aplicación del modo de funcionamiento 5, que utiliza una estrategia basada en eventos, en la que se consigue disminuir el ancho de banda de las comunicaciones dedicadas al control, sería lo más apropiado para evitar este comportamiento, además de utilizar la metodología explicada en el caso de retrasos y desfases en las comunicaciones.

En las siguientes figuras se muestra la influencia del ancho de banda para cada porcentaje de pérdidas. Se realizan 10 simulaciones por cada porcentaje de pérdida de datos distribuidas de forma aleatoria a lo largo de todo el tiempo, representado tanto el valor del error máximo como del medio de cada una de las simulaciones por cada porcentaje de pérdidas. Además, el ancho de banda de las interrupciones se va variando desde el 0 a 99%. Por lo tanto, para representar cada una de las figuras para cada modo de funcionamiento se necesitan 2280 simulaciones.

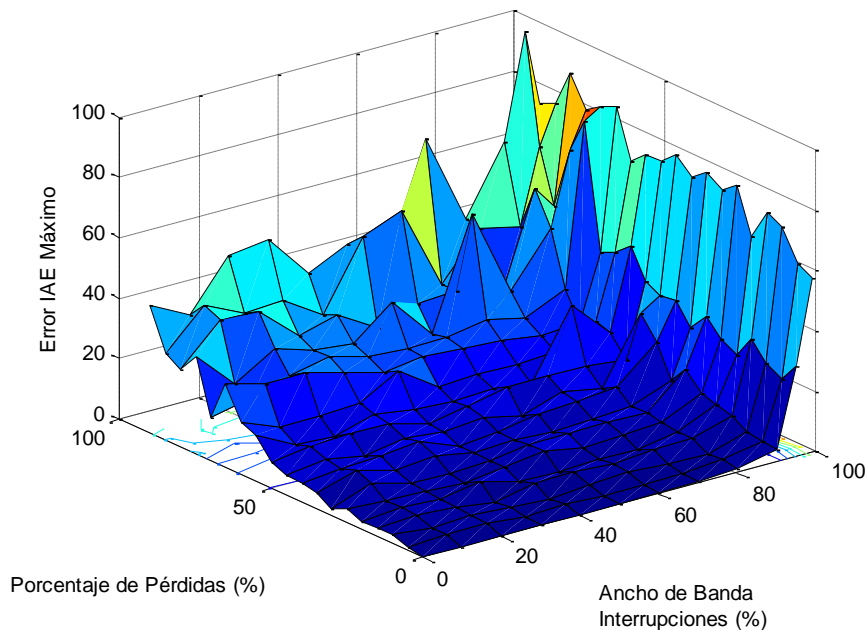


Figura 72. Influencia del ancho de banda utilizando el modo 2. IAE máximo

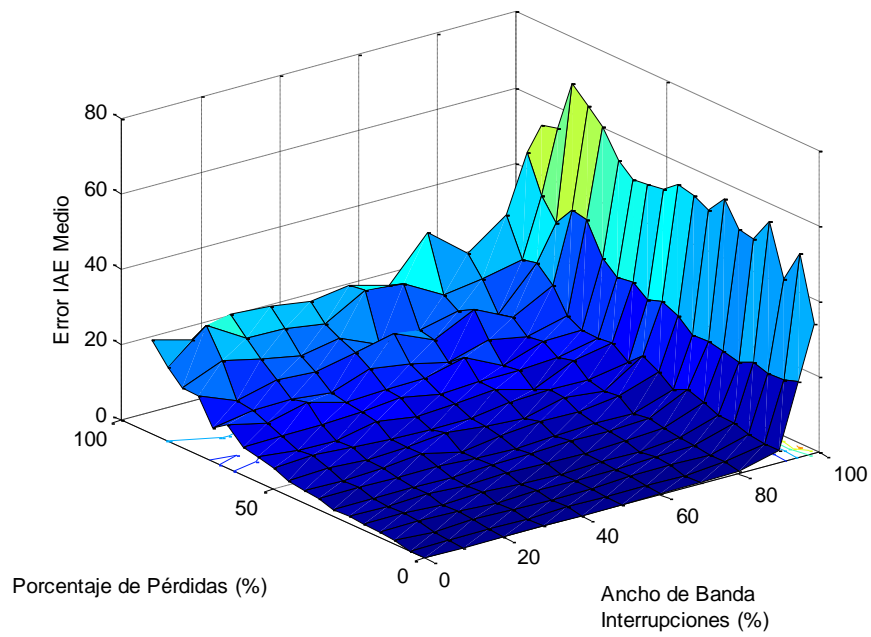


Figura 73. Influencia del ancho de banda utilizando el modo 2. IAE medio

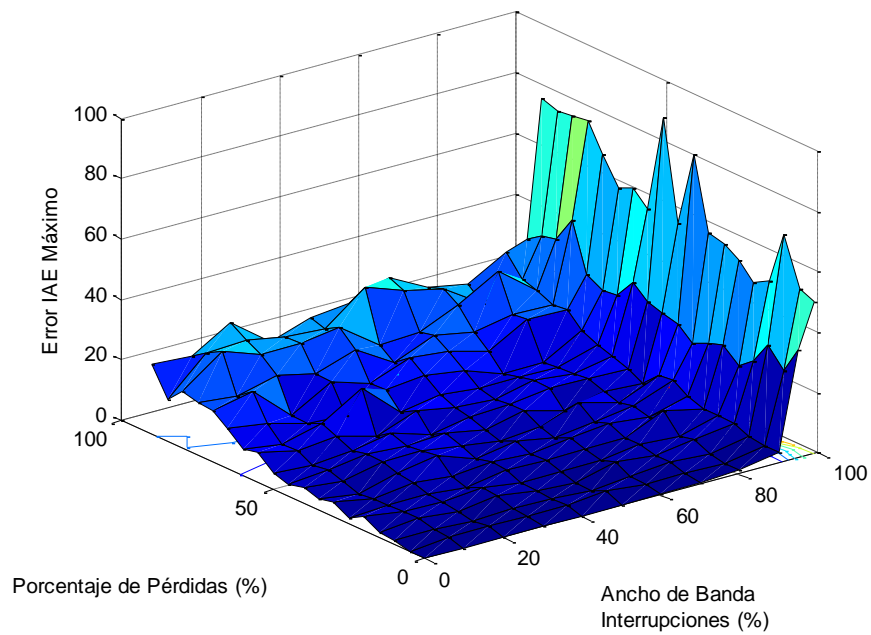


Figura 74. Influencia del ancho de banda utilizando el modo 3. IAE máximo

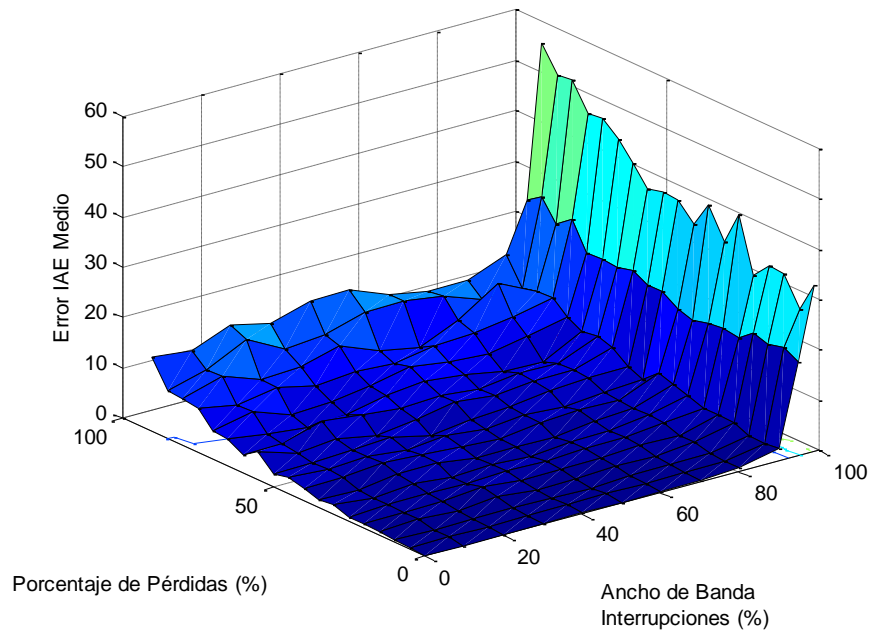


Figura 75. Influencia del ancho de banda utilizando el modo 3. IAE medio

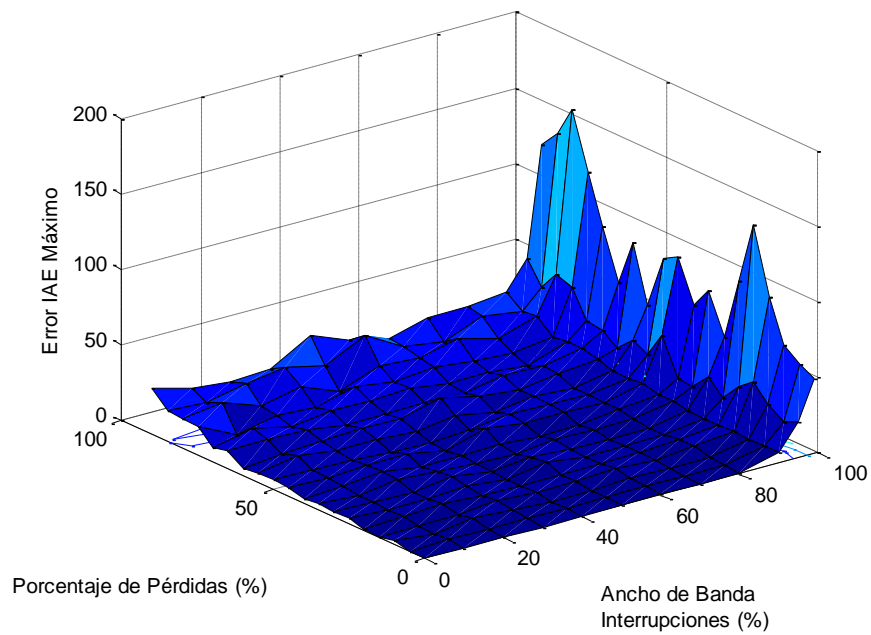


Figura 76. Influencia del ancho de banda utilizando el modo 4. IAE máximo

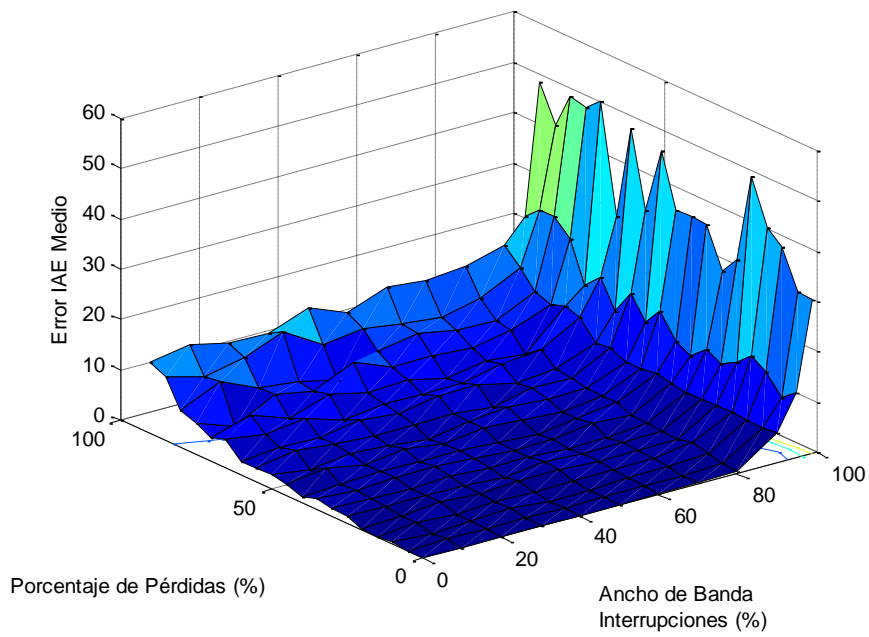


Figura 77. Influencia del ancho de banda utilizando el modo 4. IAE medio

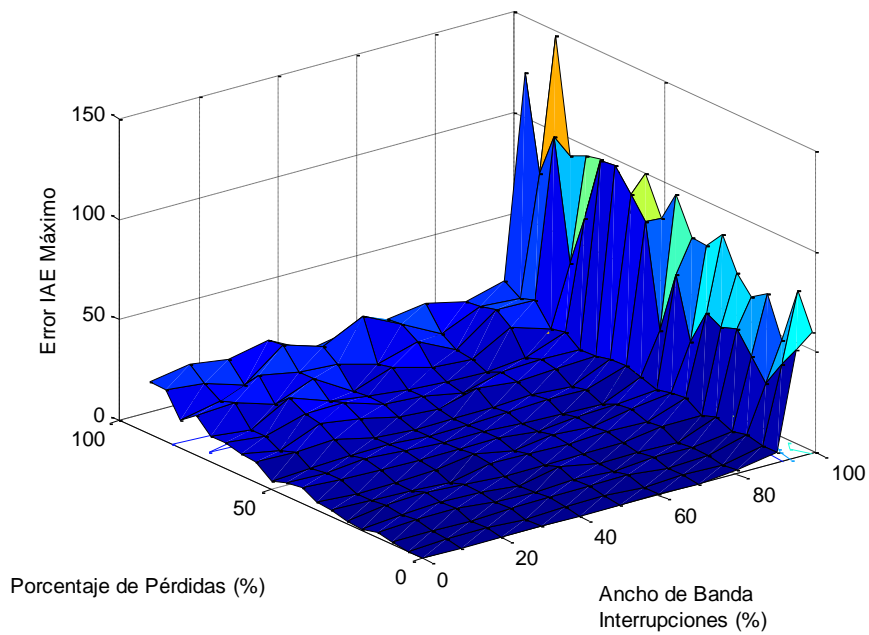


Figura 78. Influencia del ancho de banda utilizando el modo 5. IAE máximo

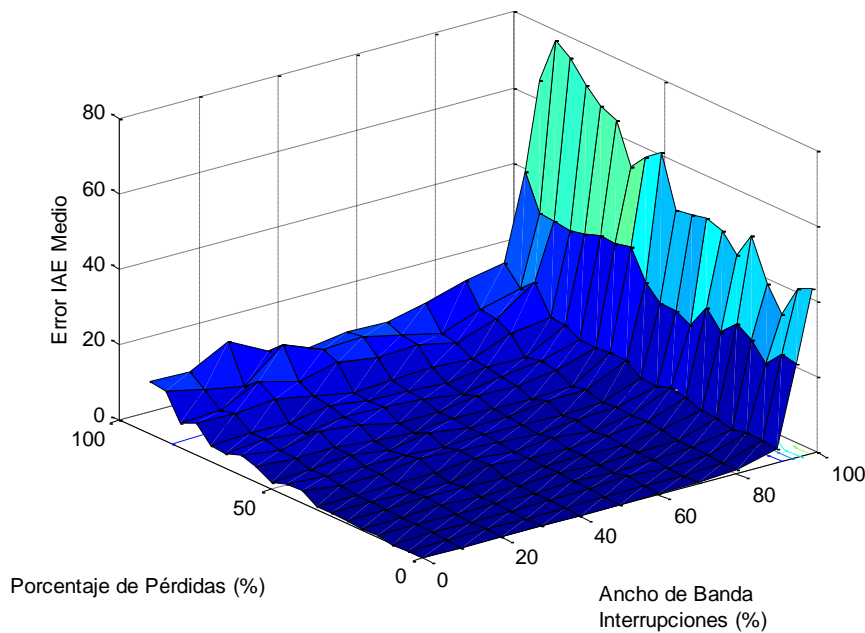


Figura 79. Influencia del ancho de banda utilizando el modo 5. IAE medio

Las figuras anteriores indican el mismo comportamiento mostrado en el caso inicial (Figura 67) en el que no se considera el porcentaje de pérdida de datos. Es decir, se debe asegurar un mínimo de ancho de banda dedicado a las comunicaciones de las tareas de control para este ancho de banda no suponga problemas en las prestaciones del sistema.

Las simulaciones anteriores permiten conocer a priori cual debe ser ese ancho de banda mínimo, que es dependiente tanto del proceso, el control aplicado y el tipo de comunicaciones utilizado. Para el caso en concreto del proceso de segundo orden con el regulador supervisor CRHPC se debe asegurar un ancho de banda del 15-20% para no tener problemas.

5.3.3 Tiempo de cómputo del controlador

El tiempo de cómputo del controlador es el tiempo de CPU necesario para completar la ejecución de la tarea de control. Este tiempo depende de la complejidad del algoritmo de control y de la velocidad del procesador.

En la estructura del sistema distribuido de control se han considerado dos tipos de nodos: el nodo local con pocos recursos de computación, y por lo tanto limitado en las operaciones que puede realizar, y el nodo supervisor, que se supondrá sin problemas de tiempo de cómputo. En cualquier caso, en este apartado se estudiará la influencia del tiempo de cómputo dedicado al cálculo del control supervisor en cuanto a las prestaciones conseguidas en el sistema.

Las pruebas se han realizado utilizando el proceso de 2º orden subamortiguado con el regulador CRHPC en el nodo supervisor, con los siguientes parámetros: $N=8$, $m=2$, $Nu=8$, $K=2.6$. Se utiliza un canal de comunicaciones compartido con un ancho de banda de las tareas dedicadas al control del 90%.

En las siguientes figuras se muestra la influencia del tiempo de cómputo para cada porcentaje de pérdidas. El tiempo de cómputo de la tarea de control va desde el 1 al 90% del periodo de muestreo y el porcentaje de pérdidas desde el 0 al 90%. Igual que en las pruebas anteriores se representan el error máximo y el medio conseguido en cada una de las simulaciones para cada porcentaje de pérdidas. Para representar cada una de las figuras para cada modo de funcionamiento se necesitan 1140 simulaciones.

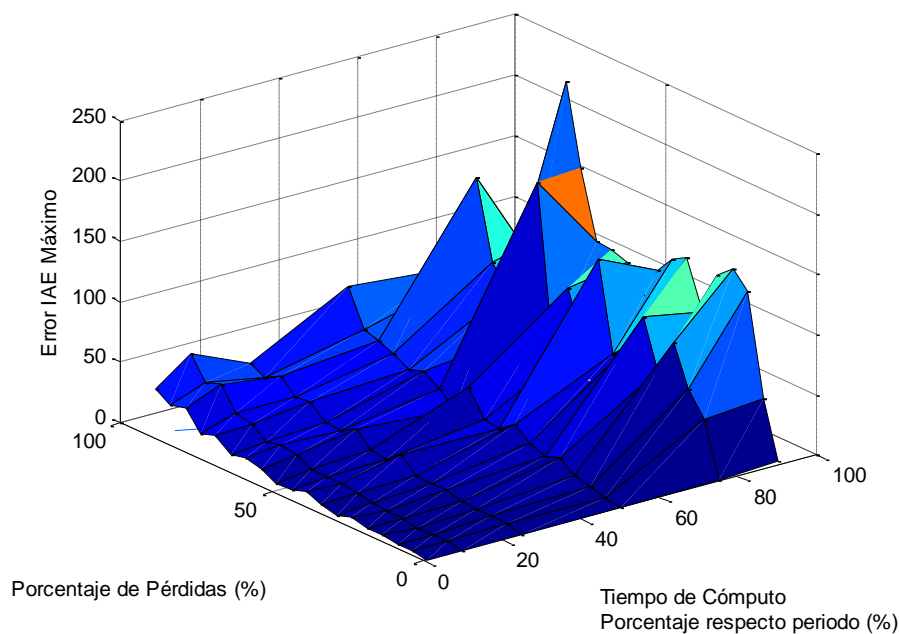


Figura 80. Influencia del tiempo de cómputo utilizando el modo 2. IAE máximo

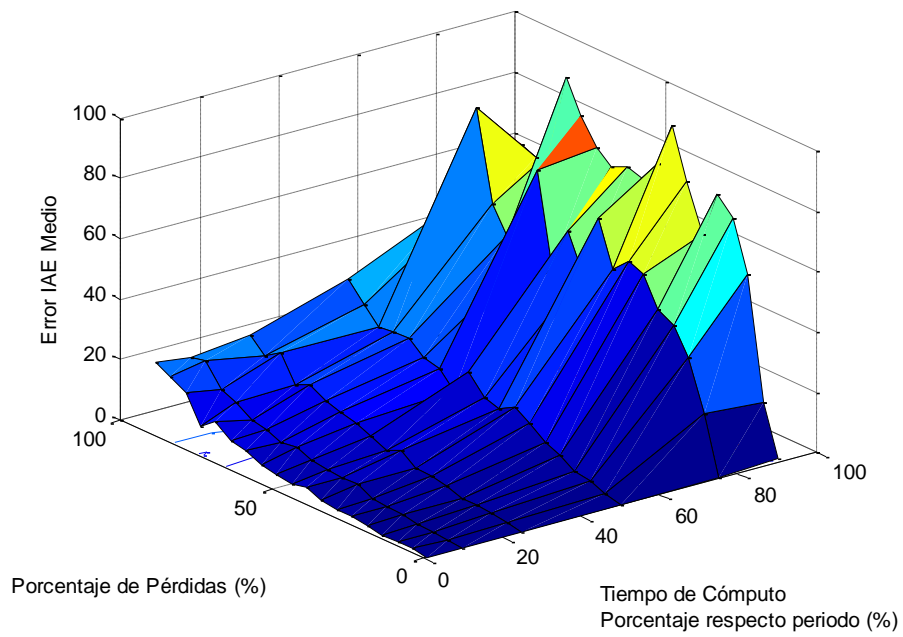


Figura 81. Influencia del tiempo de cómputo utilizando el modo 2. IAE medio

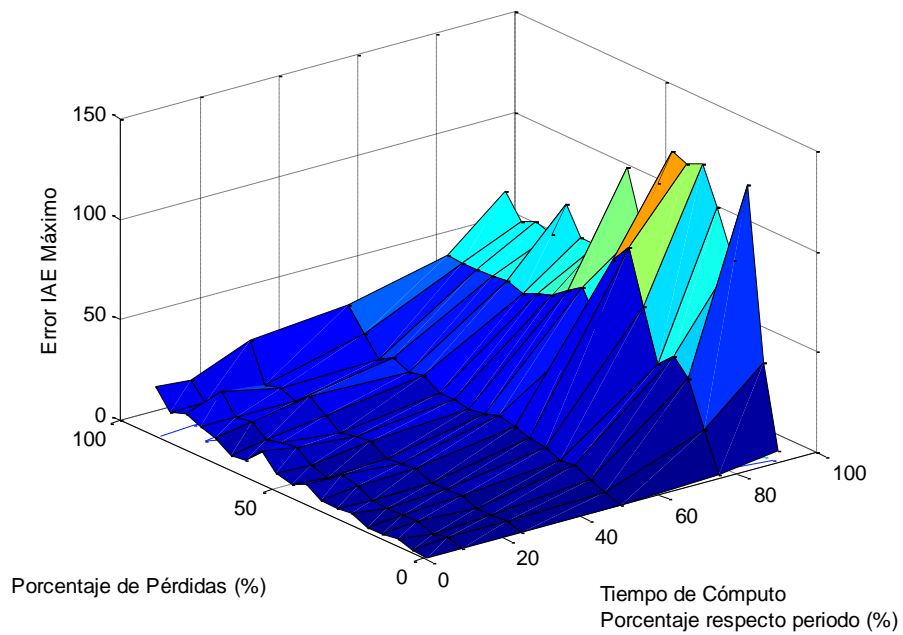


Figura 82. Influencia del tiempo de cómputo utilizando el modo 3. IAE máximo

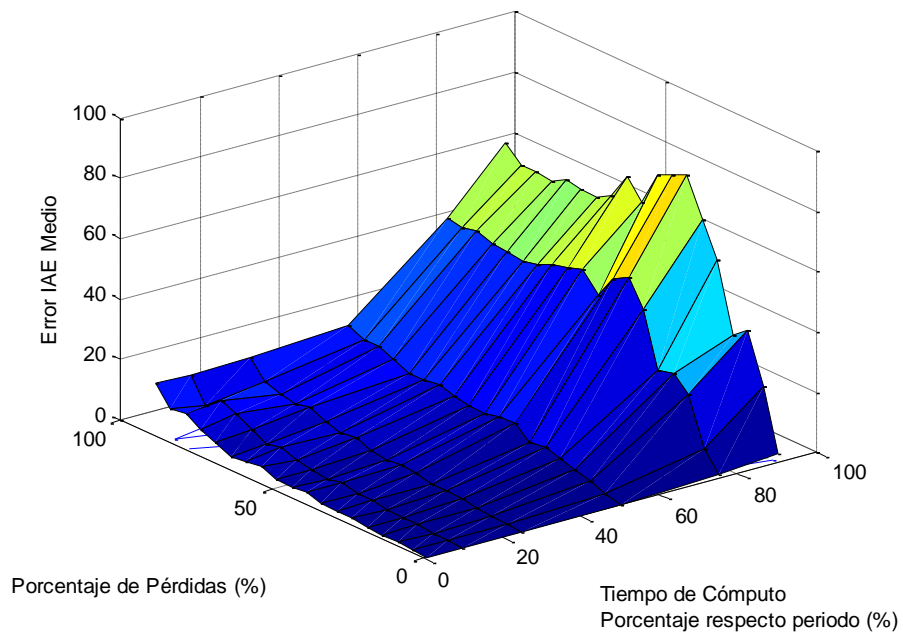


Figura 83. Influencia del tiempo de cómputo utilizando el modo 3. IAE medio

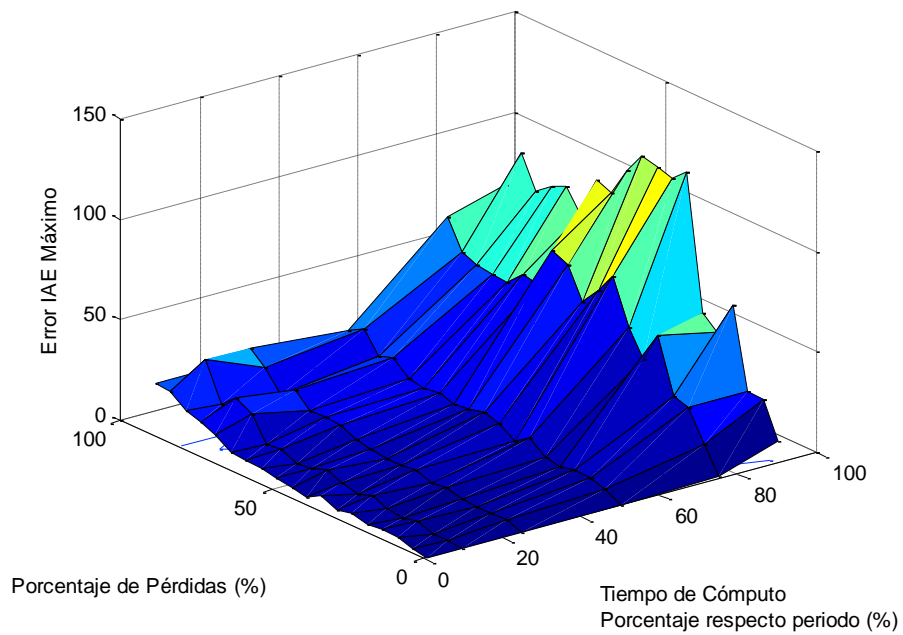


Figura 84. Influencia del tiempo de cómputo utilizando el modo 4. IAE máximo

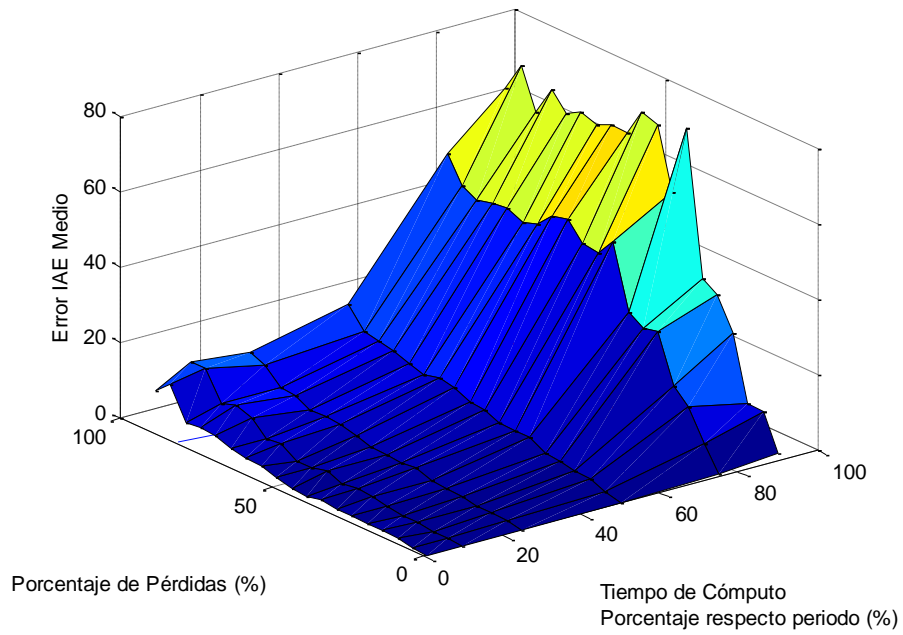


Figura 85. Influencia del tiempo de cómputo utilizando el modo 4. IAE medio

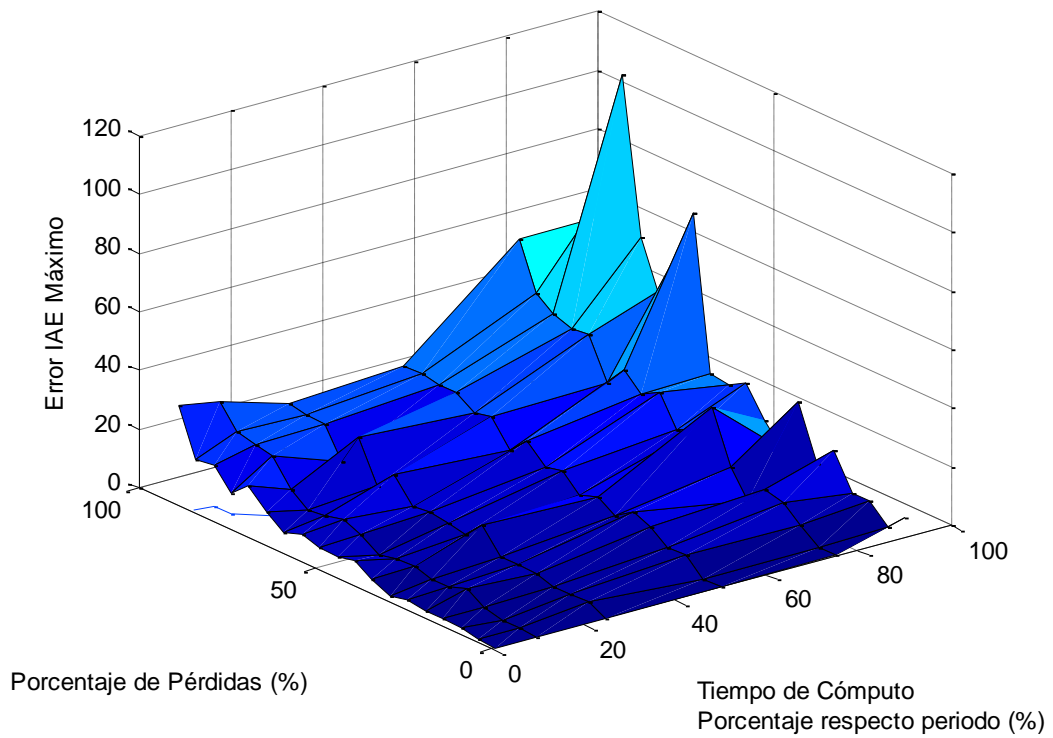


Figura 86. Influencia del tiempo de cómputo utilizando el modo 5. IAE máximo

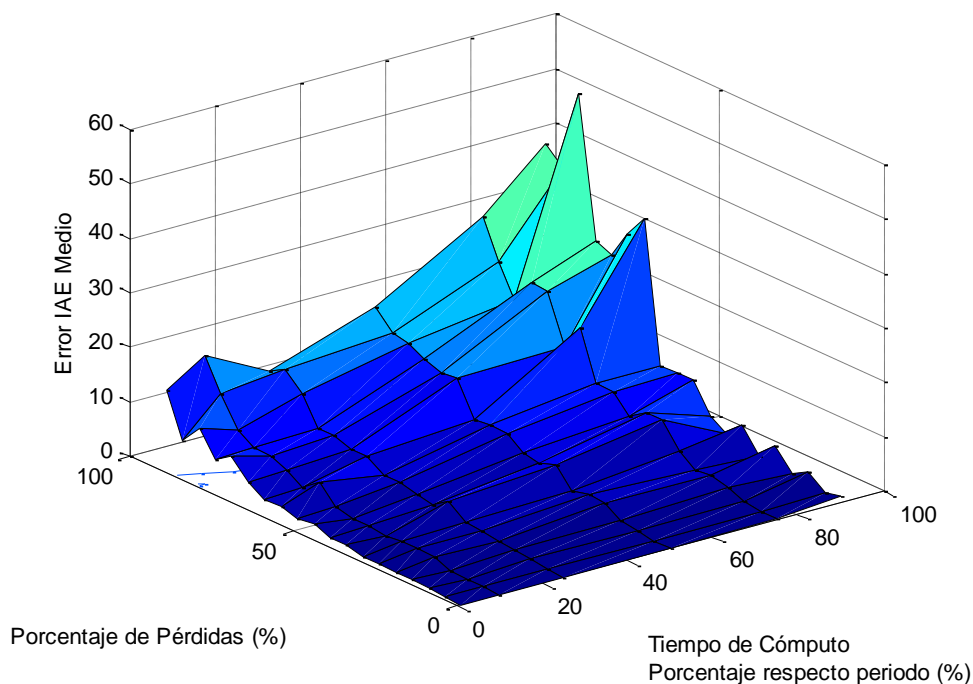


Figura 87. Influencia del tiempo de cómputo utilizando el modo 5. IAE medio

Las figuras anteriores muestran la influencia del tiempo de cómputo respecto de las prestaciones conseguidas. Al igual que en el caso del ancho de banda, se puede ver como hay un valor mínimo del tiempo del cómputo a partir del cual se degrada la respuesta del sistema. En este caso, el modo 5 es el que consigue un mejor comportamiento, ya que sólo utiliza el tiempo de cómputo en los cambios de referencia.

5.3.4 Perturbaciones a la salida del sistema

Las perturbaciones son entradas que influyen sobre las variables controladas y sobre las que no se puede actuar. En este caso se ha considerado una perturbación que se introduce en la salida del sistema, tal y como se muestra en la Figura 134 y la Figura 135.

Las perturbaciones son una de las causas de la utilización del control en bucle cerrado y de estrategias que permitan disminuir su efecto sobre la salida del proceso. En este caso, se dispone de la compensación de la acción de control, para seguir llevando a la salida del sistema a su trayectoria prevista en caso de pérdida de datos.

Las pruebas se han realizado, igual que los casos anteriores, utilizando el proceso de 2º orden subamortiguado con el regulador CRHPC en el nodo supervisor, con los siguientes parámetros: $N=8$, $m=2$, $Nu=8$, $K=2.6$. Se utiliza un canal de comunicaciones compartido con un ancho de banda de las tareas dedicadas al control del 90%.

En las siguientes figuras se muestra la influencia del ruido para cada porcentaje de pérdidas. El ruido a la salida del proceso va desde un 0.0 a 0.1 de varianza con una media de 0, y el porcentaje de pérdidas desde el 0 al 90%. A continuación se muestran el error máximo y el medio conseguido en cada una de las simulaciones para cada porcentaje de pérdidas. Para representar cada una de las figuras para cada modo de funcionamiento se necesitan 1140 simulaciones.

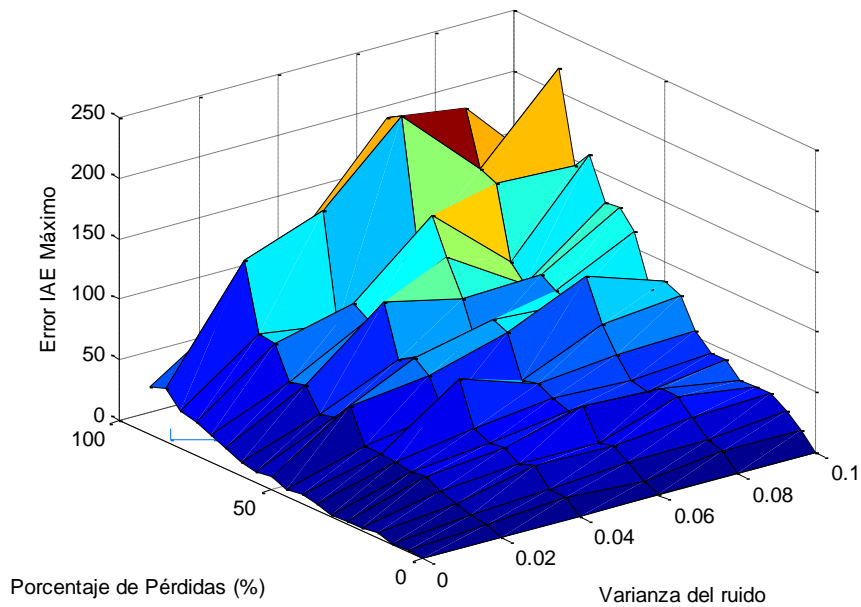


Figura 88. Influencia de perturbaciones utilizando el modo 2. IAE máximo

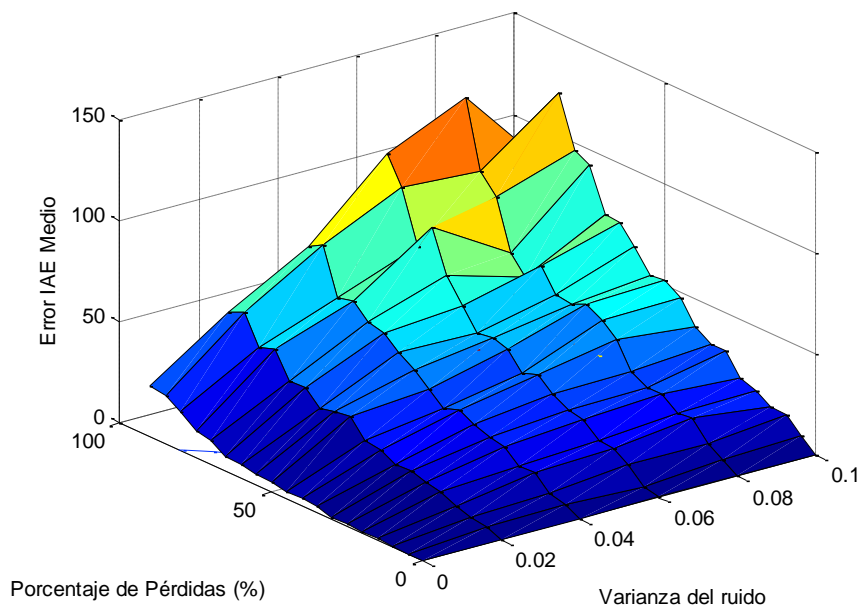


Figura 89. Influencia de perturbaciones utilizando el modo 2. IAE medio

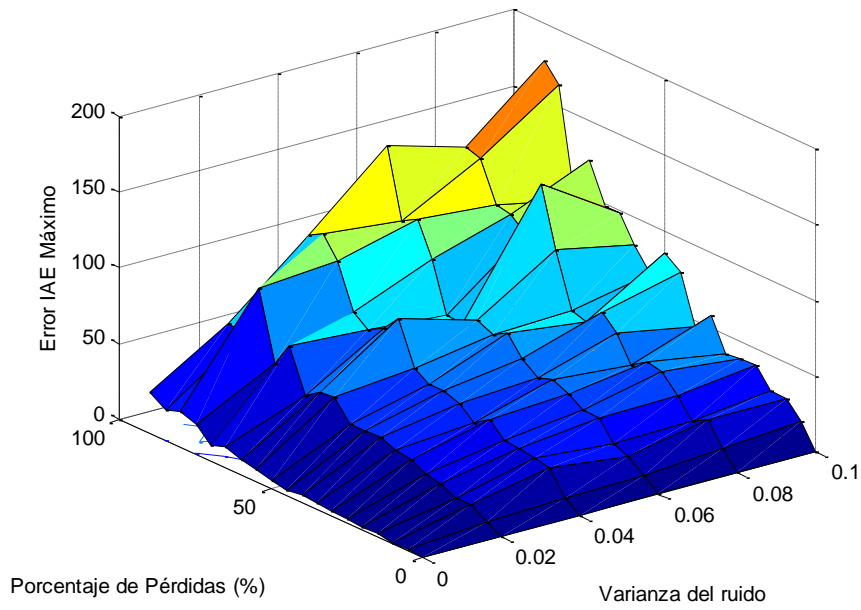


Figura 90. Influencia de perturbaciones utilizando el modo 3. IAE máximo

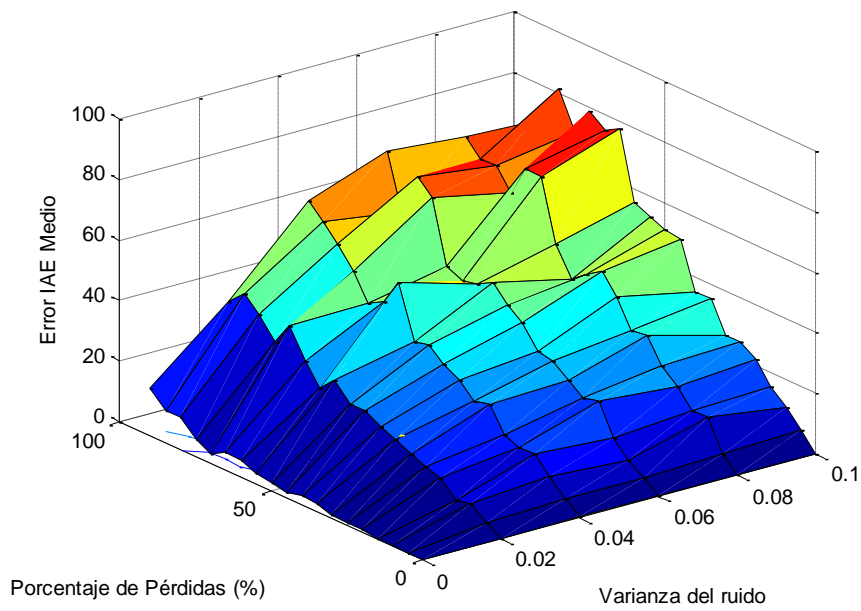


Figura 91. Influencia de perturbaciones utilizando el modo 3. IAE medio

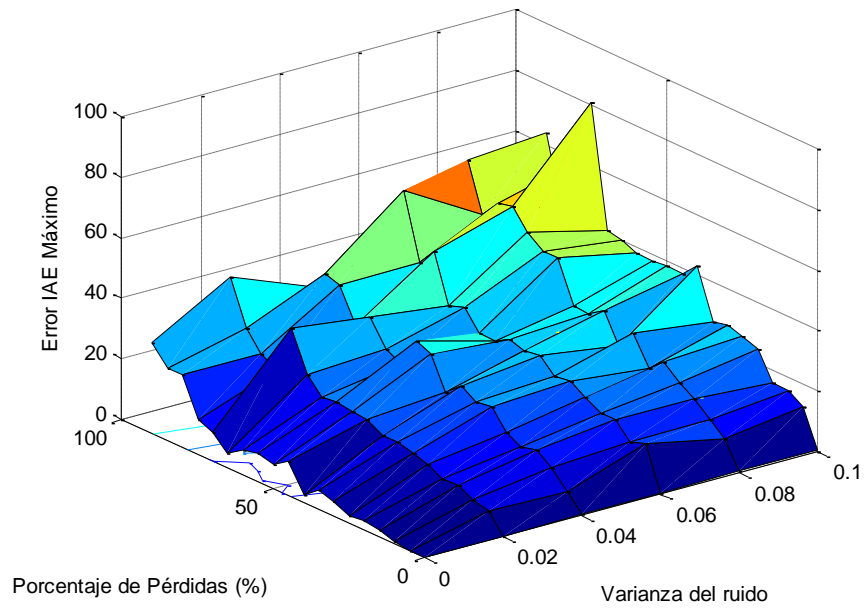


Figura 92. Influencia de perturbaciones utilizando el modo 4. IAE máximo

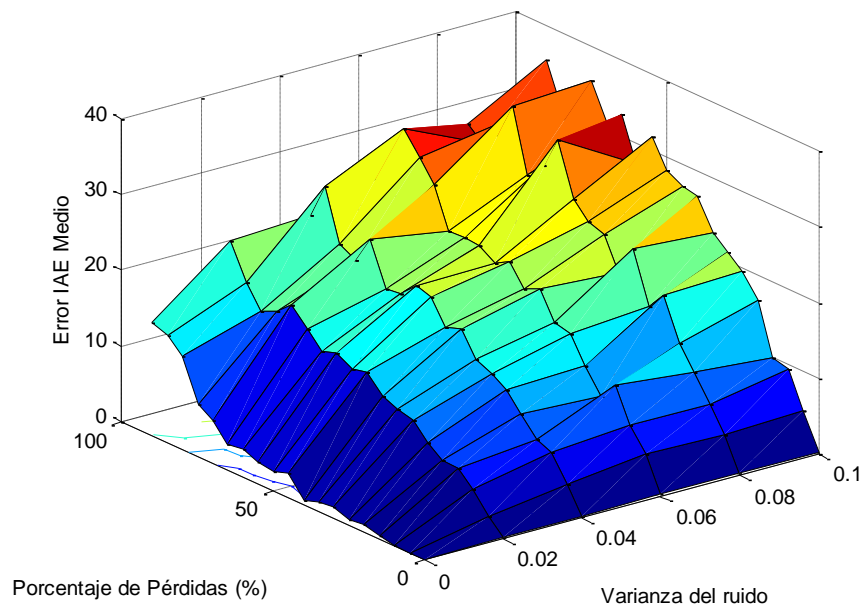


Figura 93. Influencia de perturbaciones utilizando el modo 4. IAE medio

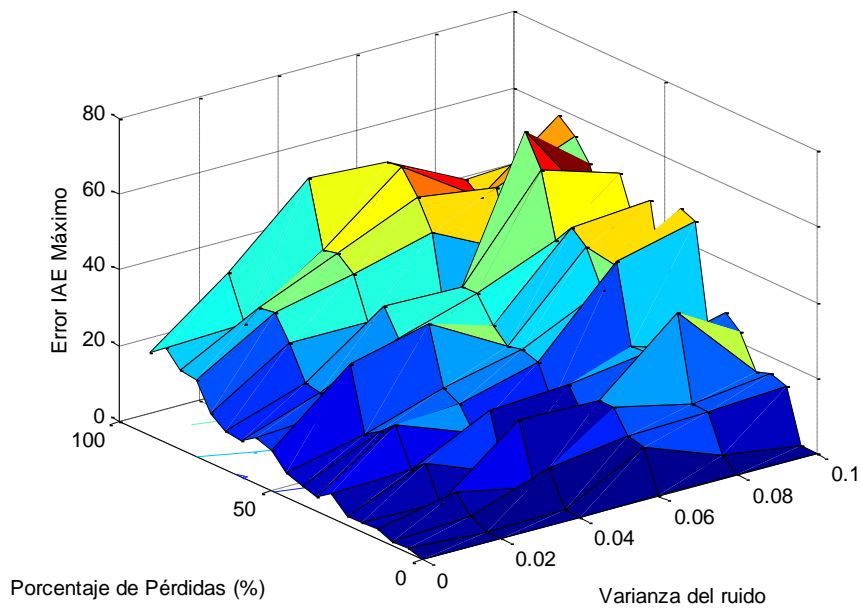


Figura 94. Influencia de perturbaciones utilizando el modo 5. IAE máximo

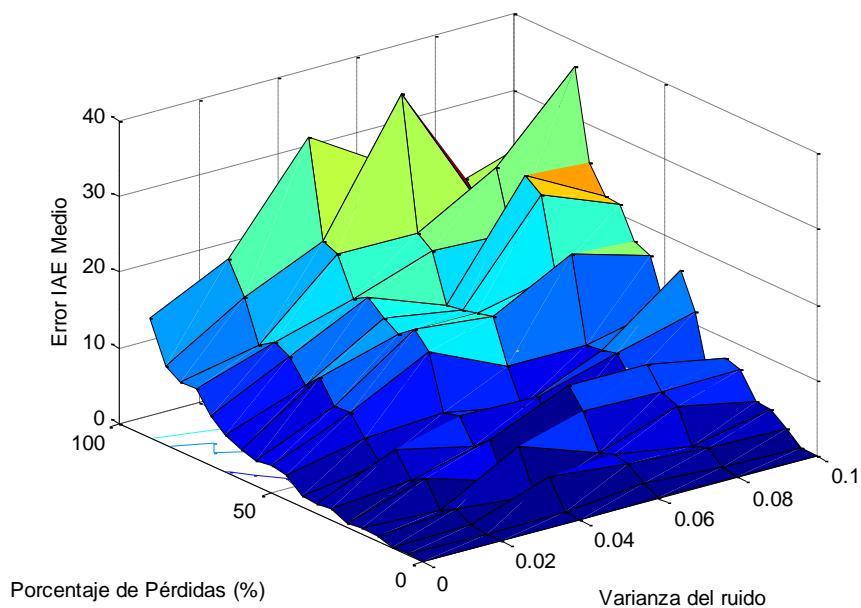


Figura 95. Influencia de perturbaciones utilizando el modo 5. IAE medio

En las figuras anteriores se muestra como el error aumenta con el porcentaje de pérdidas y con la varianza del ruido. Los modos de funcionamiento que consiguen unos errores más bajos son el modo 4 y 5, que son aquellos que utilizan la compensación de la acción de control. Por lo tanto, la compensación de la acción de control se muestra como un buen método para conseguir disminuir los errores que se cometen al perder datos y utilizar las acciones de control postuladas. Como se expuso

en el apartado 4.5.5.1 uno de los objetivos de la compensación de la acción de control era hacer frente a las perturbaciones y los errores de modelado.

5.4 Resumen y conclusiones del capítulo

En el presente capítulo se han definido los índices que han permitido evaluar el desempeño de los distintos modos de funcionamiento del sistema distribuido de control ante diversas situaciones.

Se ha realizado una herramienta de simulación en Matlab, utilizando la toolbox Truetime, que permite simular, evaluar y asistir en el diseño de los parámetros de los controladores a implementar en cada nodo.

Para evaluar las prestaciones conseguidas por los distintos modos de funcionamiento en el sistema distribuido de control cuando se pierden datos, se han realizado pruebas simuladas sobre procesos de naturaleza distinta, utilizando la herramienta de simulación creada en este trabajo de tesis. En cada ejemplo primero se han definido los parámetros de cada controlador, aplicando la metodología explicada en el apartado 4.6. Con estos parámetros se han comparado el funcionamiento del sistema distribuido con los distintos modos de funcionamiento ante pérdidas de datos con un canal de comunicaciones exclusivo y compartido.

Para el caso concreto del proceso de segundo orden subamortiguado se ha estudiado la influencia de la discrepancia del modelo respecto del proceso real, el ancho de banda consumido, el tiempo de cómputo del controlador y la magnitud de las perturbaciones a la salida. Con estas pruebas pueden establecerse las características del entorno que influyen sobre el sistema y elegir apropiadamente las condiciones de utilización: ancho de banda del control, tiempo de cómputo y ruido máximo admisible.

Las pruebas realizadas a los distintos procesos, mediante la herramienta de asistencia y simulación del sistema distribuido de control, permiten obtener las siguientes conclusiones:

- Mediante el estudio de los valores de K , en la compensación de la acción de control, se puede estudiar el margen de valores a aplicar que mantendrán al sistema dentro de unos márgenes de error establecidos en caso de pérdida de datos.
- En general para bajas pérdidas de datos (entre 1 y 15%) los índices de error cometidos con cada modo de funcionamiento son similares.
- A medida que aumenta el porcentaje de pérdidas, el error cometido por los modos de funcionamiento que utilizan las trayectorias postuladas es menor que el que utiliza la acción de control anterior.

- En general los modos de funcionamiento 4 y 5, que son los que compensan la acción de control con el error cometido en la salida, son los que obtienen un índice de error más bajo.
- Mediante la herramienta de simulación se puede evaluar la influencia de la discrepancia del modelo respecto del proceso real, el ancho de banda dedicado a las comunicaciones, el tiempo de cómputo dedicado al cálculo del controlador y las perturbaciones a la salida con respecto al porcentaje de pérdida de datos para obtener el índice de error en cada caso.
- Los modos de funcionamiento que utilizan la compensación de la acción de control permiten obtener mejores resultados en presencia de perturbaciones y errores de modelado.

6 APLICACIONES

6.1 Introducción

En este capítulo se muestran tres aplicaciones en las que se utiliza la estructura de control distribuida, usando el núcleo de control, así como la metodología para la elección de los parámetros de los controladores que se implementan en cada nodo, presentada y analizada a lo largo de la tesis.

La primera de las aplicaciones presenta un método para balancear la carga computacional del sistema de control de un robot móvil con tracción diferencial, consistente en la conmutación del regulador que calcula las acciones de control. La orden de conmutación del regulador puede ser aplicada por el mismo control local en situaciones de sobrecarga o venir impuesta por el control supervisor, para que cada nodo de control se adecue a las condiciones globales del sistema. Además, en caso de pérdida en las comunicaciones el sistema robótico seguirá la trayectoria estimada por el control supervisor.

Las otras dos pruebas muestran la aplicación de la estructura de control distribuida, formada por nodos supervisores y locales, en las que se utilizan los distintos modos de funcionamiento y su actuación frente a las pérdidas de datos. Los procesos utilizados han sido estudiados en las pruebas simuladas: un proceso electrónico de segundo orden con respuesta ante escalón subamortiguada y el control de posición de un motor de corriente continua.

6.2 Aplicación de la conmutación de controladores en un sistema robótico

En el siguiente ejemplo se muestra una estructura distribuida basada en comportamientos. Para esta experiencia se utiliza un robot móvil Khepera. El Khepera es un pequeño robot móvil con capacidad limitada de cómputo. Por lo tanto, un computador remoto se encarga de ejecutar las tareas que requieran de una alta capacidad de cómputo, comunicándose con el robot mediante una torreta de comunicaciones inalámbrica. El objetivo principal de esta prueba es que el robot siga

una trayectoria planificada y que evite los obstáculos móviles con los que se va encontrando. Mediante la información de los encoders de las ruedas y del modelo cinemático del robot, se calcula su posición en el entorno. A partir de la trayectoria deseada, y mediante el control cinemático, se calcula el vector de velocidad que debe seguir el robot y que finalmente proporciona las velocidades a aplicar a cada rueda, para cumplir con el vector de velocidad calculado. Estas dos tareas se ejecutan en un ordenador remoto, debido a la imposibilidad de realizar los cálculos trigonométricos necesarios por parte del robot. Las comunicaciones entre el robot y el ordenador remoto son de tipo bidireccional, ya que el robot proporciona la lectura de los encoders y recibe la velocidad que debe aplicar a cada uno de los motores de las ruedas. La Figura 96 muestra el esquema distribuido implementado en este ejemplo.

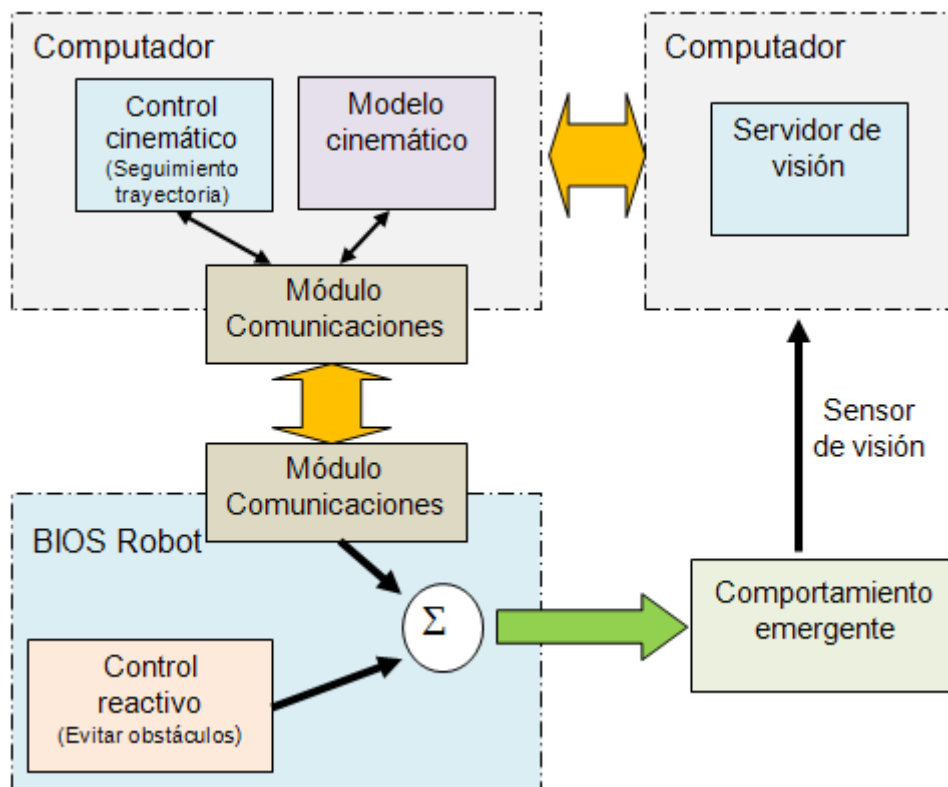


Figura 96. Estructura distribuida basada en comportamientos

Para evitar los obstáculos no contemplados en la planificación de la trayectoria se utiliza un control reactivo basado en un vehículo de Braitenberg (Braitenberg 1984), e implementado dentro de la BIOS del robot. Esta tarea es de una prioridad mayor con respecto al seguimiento de la trayectoria, y por lo tanto, ante la presencia de un obstáculo el robot reacciona evitándolo y tratando de seguir, en la medida de lo posible, la trayectoria planificada inicialmente, ya que se realiza una composición de comportamientos (Murphy 2000).

En caso de pérdida de datos o retardo en el suministro de las acciones de control que debe aplicar el robot, el control local puede tomar la información postulada para la acción de control aplicando la compensación con el error cometido en la salida, y

generar así una nueva señal de control. Esta señal de control será muy sencilla, debido a la limitación de cómputo del robot, pero permite mantener al robot en la trayectoria en caso de pérdidas ocasionales de datos.

Otra prueba realizada, utilizando el mismo esquema, es una conmutación entre dos controladores, uno implementado en el computador remoto y otro más sencillo implementado en la BIOS del robot. Dependiendo de la disponibilidad de cómputo en cada instante y del número de tareas se opta por la ejecución de un controlador u otro. El objetivo es asegurar siempre el envío de la acción de control a los motores de las ruedas para conseguir el objetivo deseado, o en última instancia llevar al robot a una situación segura.

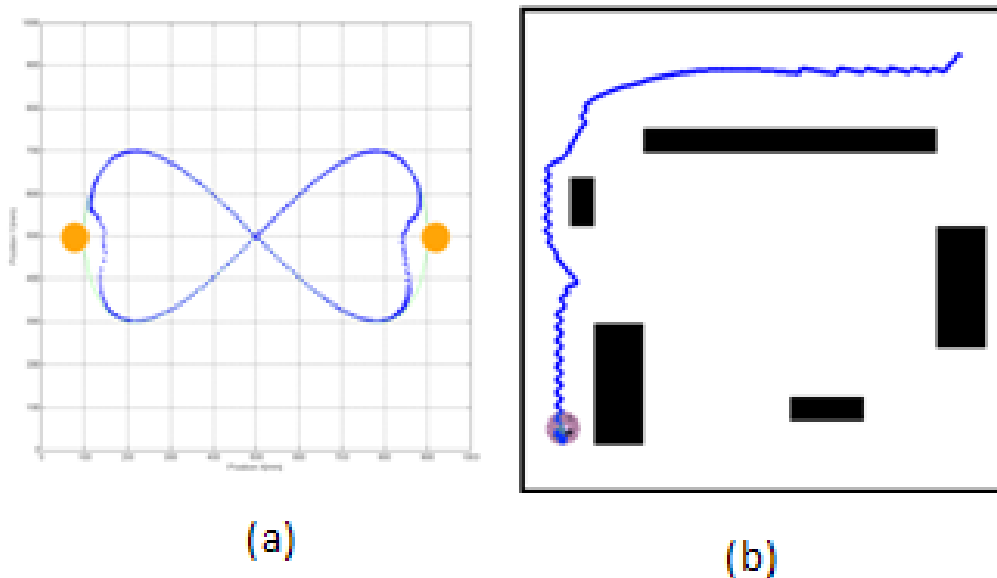


Figura 97. (a) Seguimiento de una trayectoria con dos obstáculos circulares no planificados. (b) Alcanzar un punto objetivo en un entorno no estructurado

La implementación de las tareas en la BIOS del robot se realiza mediante un compilador C, con licencia GNU, que genera el código que se ejecutará directamente en el procesador del robot. El compilador cruzado proporciona las bibliotecas de todas las funciones de bajo nivel disponibles en la BIOS de K-Team incluyendo el control de motores, lectura de sensores, comunicaciones, y gestión multitarea. El compilador cruzado también facilita un entorno completo de programación C.

La utilización del supervisor para el control de robots móviles se encuentra detallado en (Benitez, y otros 2007). El trabajo de la conmutación de distintos controles y la utilización de la trayectoria prevista fue presentado en el IFAC World Congress 2008 (Valera, y otros 2008).

Las pruebas realizadas sobre el robot móvil fueron el punto de partida para la elaboración de la estrategia de control del sistema distribuido, en el que al perder

datos se utilizan las acciones de control y las trayectorias postuladas que se ha expuesto en el apartado 4.5.5.

6.3 Sistema distribuido de control utilizando el núcleo de control y los distintos modos de funcionamiento

En este apartado se propone estudiar el comportamiento de un sistema distribuido de control formado por un nodo supervisor, en el que se ejecuta un control predictivo, y un nodo local conectado directamente al proceso que se desea controlar, aplicado a procesos reales. En las pruebas se utilizarán los modos de funcionamiento 2, 3, 4 y 5, que son aquellos en los que las acciones de control son calculadas por el control supervisor, tal y como se presentó en el apartado 4.5.6.

Los procesos reales que se van a probar consisten en un proceso de segundo orden con respuesta subamortiguada ante escalón y el control de posición de un motor de corriente continua. La estimación de los parámetros del control distribuido y la simulación del comportamiento de estos sistemas se ha presentado en los apartados 5.2.1 y B.4.

6.3.1 Proceso de segundo orden subamortiguado

Se propone como ejemplo la aplicación real del sistema de segundo orden con respuesta subamortiguada ante escalón que se ha estudiado en el apartado 5.2.1, y de esta forma probar la viabilidad de implementar la estrategia de control propuesta usando unidades de control electrónicas (ECUs) comerciales. Para ello el nodo supervisor, en el que se ejecuta un control predictivo, será implementado por un microprocesador XScale, mientras que el sistema local será implementado en un sistema empotrado que está formado por un dsPIC de Microchip (Microchip 2008) y un sistema de comunicaciones CAN, que es el que permitirá el paso de información entre el nodo supervisor y el nodo local.

El proceso real a controlar estará formado por un circuito electrónico. Se trata de un sistema de 2º orden con polos complejos, tal como se ha descrito en el apartado 5.2.1. En la Figura 98 se muestra una imagen del proceso real.

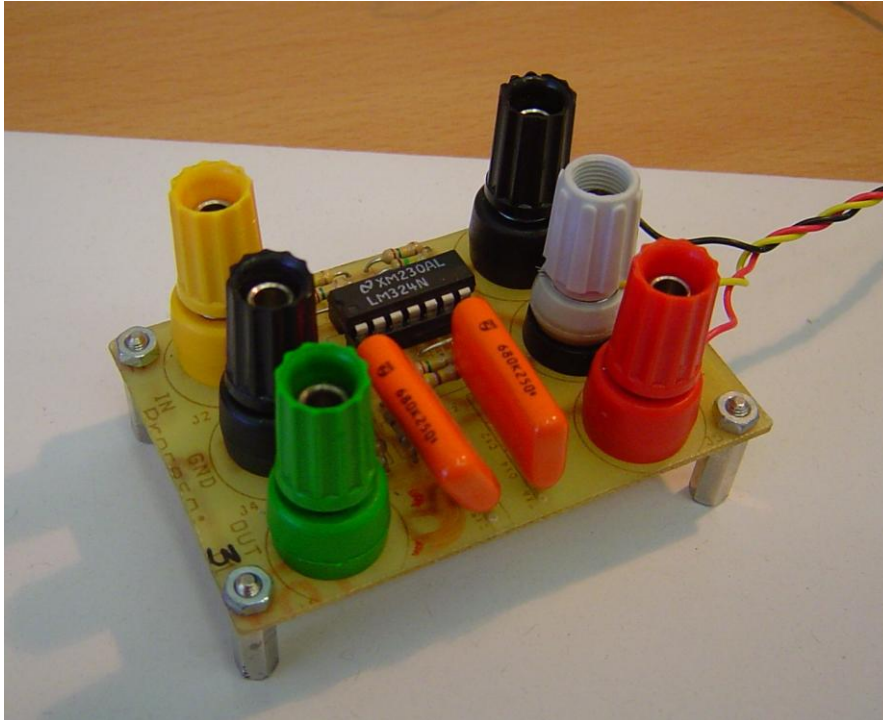


Figura 98. Proceso electrónico de 2º orden subamortiguado

El dsPIC de Microchip combina la gran velocidad de cálculo de un procesador digital de señal (DSP), con un poderoso microcontrolador (MCU) de 16-bits, para producir en un único chip con un solo flujo de instrucciones, una solución para el diseño de sistemas empotrados. Este dispositivo dsPIC alcanza velocidades de hasta 30 MIPS, es eficiente para la programación en C y dispone de memoria Flash programable, datos EEPROM, datos SRAM, periféricos y una gran variedad de bibliotecas software.

La Figura 99 muestra el computador empotrado XScale utilizado para la implementación del nodo supervisor, y en la Figura 100 se muestra una imagen de nodo local formado por el dsPIC y la placa de desarrollo. Este sistema se ha utilizado en el proyecto de investigación KERTROL para la prueba de los controladores locales.



Figura 99. XScale utilizado como nodo supervisor

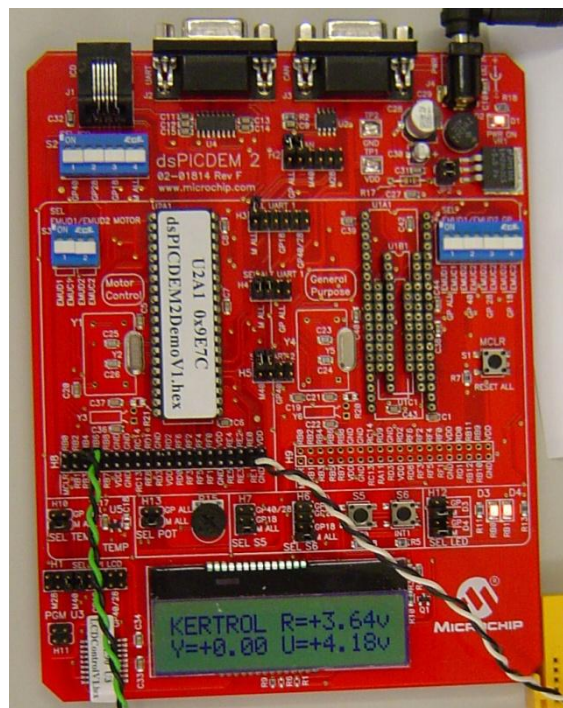


Figura 100. dsPIC de Michochip con la placa de desarrollo que forma el nodo local

Por otra parte, la arquitectura empotrada XScale ha sido elegida como plataforma de desarrollo para el nodo supervisor, que implementa el control supervisor, debido a su bajo consumo de energía, su alto rendimiento y su bajo costo. Todas las generaciones de XScale son de 32 bits ARM v5TE, procesadores fabricados con tecnología de 0,18 micras. Este procesador permite cambios de frecuencias entre 100 MHz y 400 MHz

para la optimización del consumo de energía con un alto poder de computación de 700 MIPS.

Además, este nodo supervisor utiliza el sistema operativo de tiempo real RTLinux (Yodaiken y Barabanov 1997), que ofrece características de sistema de tiempo real estricto en un núcleo de tiempo real con multi-hilos, y que se puede utilizar como sistema operativo empotrado. Para ello es necesario hacer una portabilidad para adaptar el código fuente original de RTLinux a la arquitectura XScale. Se ha desarrollado una versión RTLinux con soporte a la arquitectura XScale, disponible en (J. Coronel 2007).

La Figura 101 muestra la conexión utilizada entre los distintos elementos que forman parte del sistema distribuido en la prueba real. Para la comunicación entre el nodo supervisor y local se utiliza CAN. El ordenador es utilizado para monitorizar los datos y guardar la información de las pruebas, además, es el encargado de provocar las distintas situaciones de fallo.

En (Simarro, y otros 2008) se presentó una versión inicial del sistema distribuido de control aplicado a un proceso subamortiguado de segundo orden, utilizando el nodo supervisor y local descritos anteriormente.

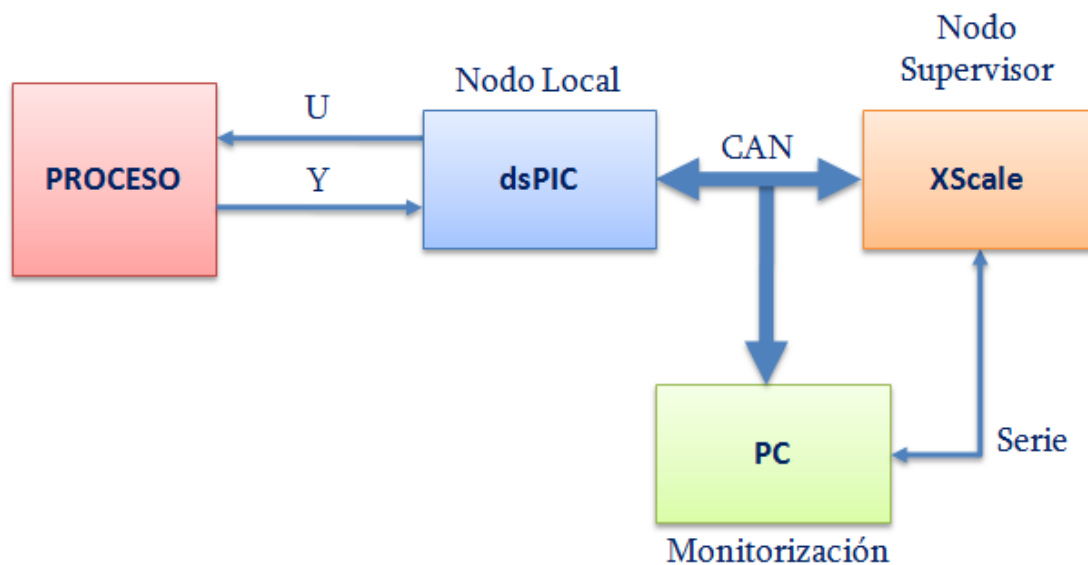


Figura 101. Estructura de control distribuida para la realización de la prueba

Para las pruebas reales se ha utilizado en el nodo supervisor el control predictivo CRHPC con los parámetros obtenidos en el apartado 5.2.1.1. Estos parámetros son: $N=8$, $m=2$, $N_u=8$, además en el nodo local se han implementado un controlador de compensación para los modos de funcionamiento 4 y 5 con una $K=2.6$.

En la siguiente figura se muestra el comportamiento de la salida del sistema cuando se aplica el sistema distribuido de control al proceso real sin pérdida de datos.

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

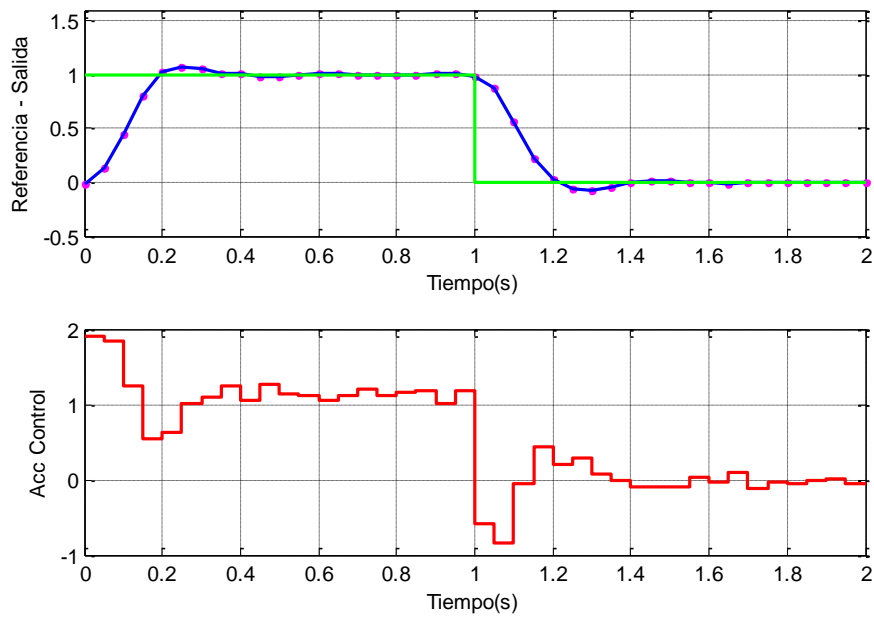


Figura 102. Salida del proceso real con el sistema distribuido de control sin pérdida de datos

A continuación se muestran las pruebas realizadas sobre el proceso real del comportamiento de los distintos modos de funcionamiento cuando se producen pérdidas de datos. Para ello en el control supervisor se ha implementado un algoritmo que provoca las pérdidas de datos de forma aleatoria a lo largo de todo el tiempo de la simulación con distintos porcentajes, es decir, de una manera similar a la que se puso en práctica en simulación.

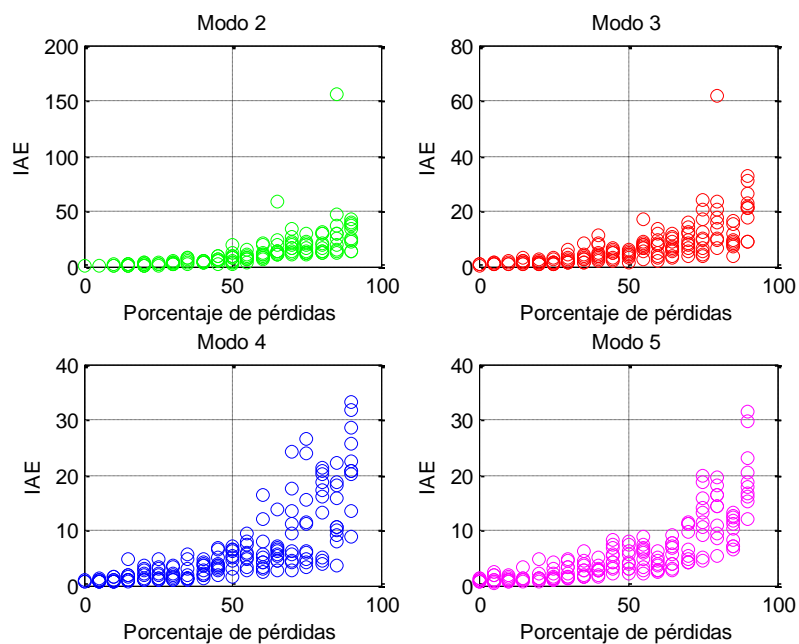


Figura 103. IAE de los distintos modos de funcionamiento en la prueba real

En las siguientes figuras se muestra la comparación de los valores máximos y medios del IAE en cada porcentaje de pérdida de datos con los distintos modos de funcionamiento.

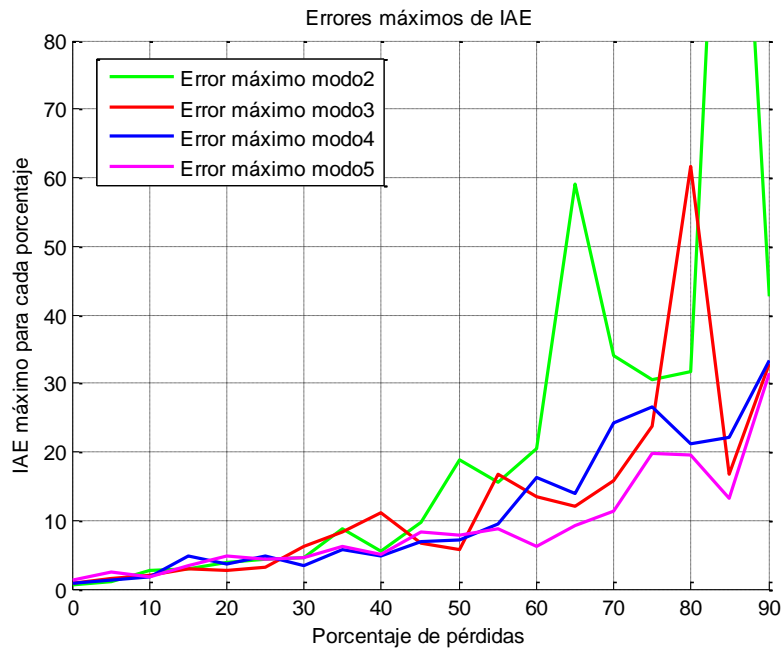


Figura 104. Comparación de los IAE máximos para cada modo de funcionamiento en la prueba real

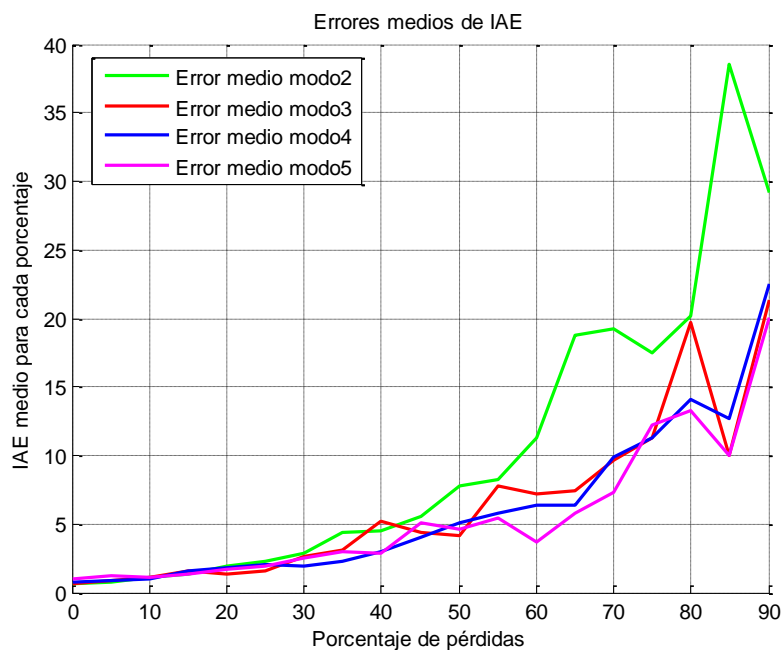


Figura 105. Comparación de los IAE medios para cada modo de funcionamiento en la prueba real

Las pruebas reales muestran un funcionamiento similar, tanto en los errores cometidos como en el comportamiento con cada modo de funcionamiento, a las presentadas mediante la herramienta de simulación en el apartado 5.2.1.2 para el caso del control supervisor CRHPC. El modo de funcionamiento 2, que es el que mantiene la acción de control sin cambios cuando se pierden datos, es el que obtiene el mayor error, sobre todo para porcentajes de pérdidas grandes. El resto de modos, que son los que utilizan las acciones de control postuladas, son los que consiguen los errores más pequeños, siendo los modos 4 y 5, que son los que utilizan la compensación de la acción de control, los que mejores resultados obtienen.

A continuación se muestra el comportamiento de la salida del sistema ante el mismo patrón de pérdidas cuando se utilizan los distintos modos de funcionamiento. El porcentaje de pérdidas que provoca este patrón es de un 40%.

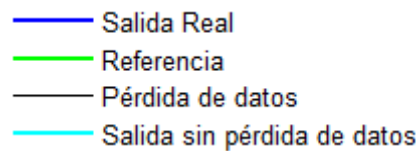


Figura 106. Código de colores utilizado en las representaciones de las pruebas

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

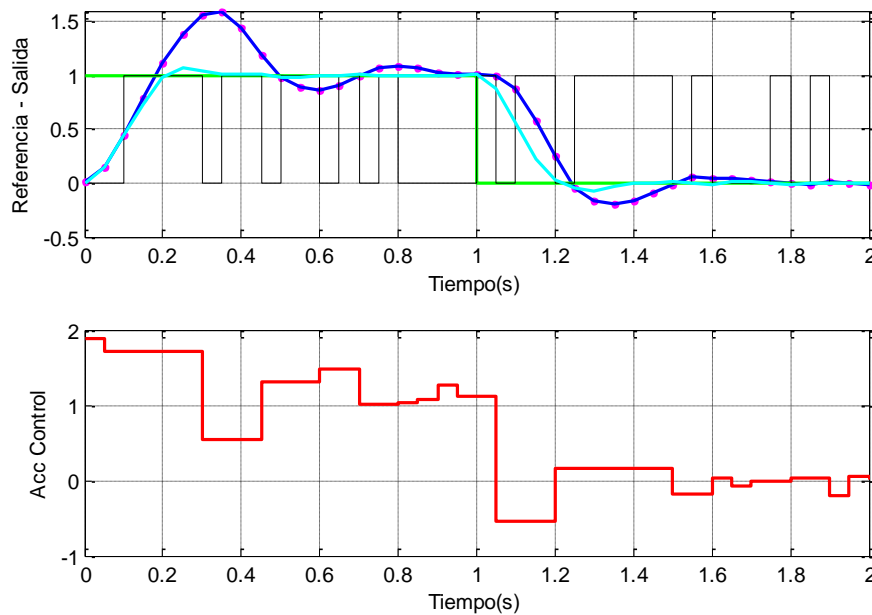


Figura 107. Comportamiento ante pérdida de datos utilizando el modo 2

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

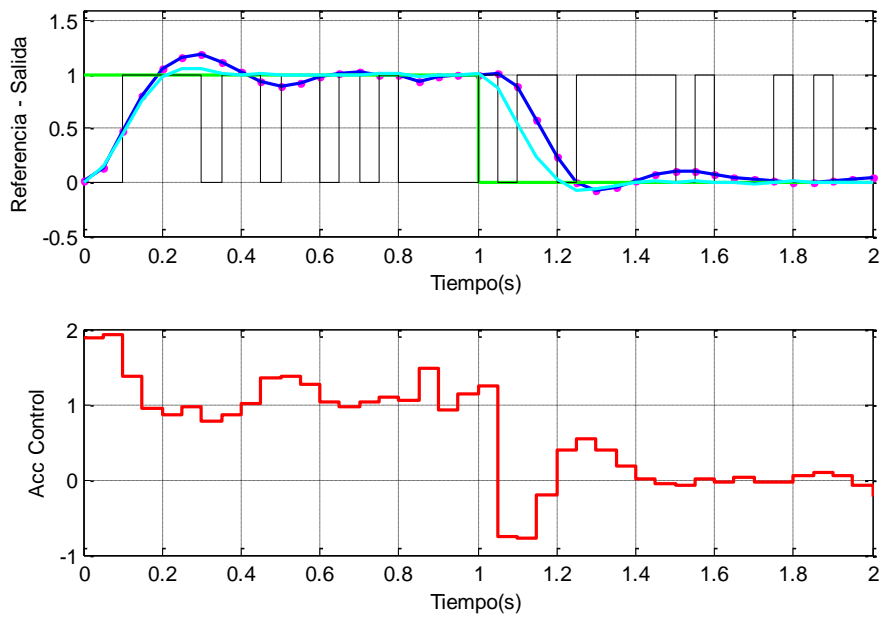


Figura 108. Comportamiento ante pérdida de datos utilizando el modo 3

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

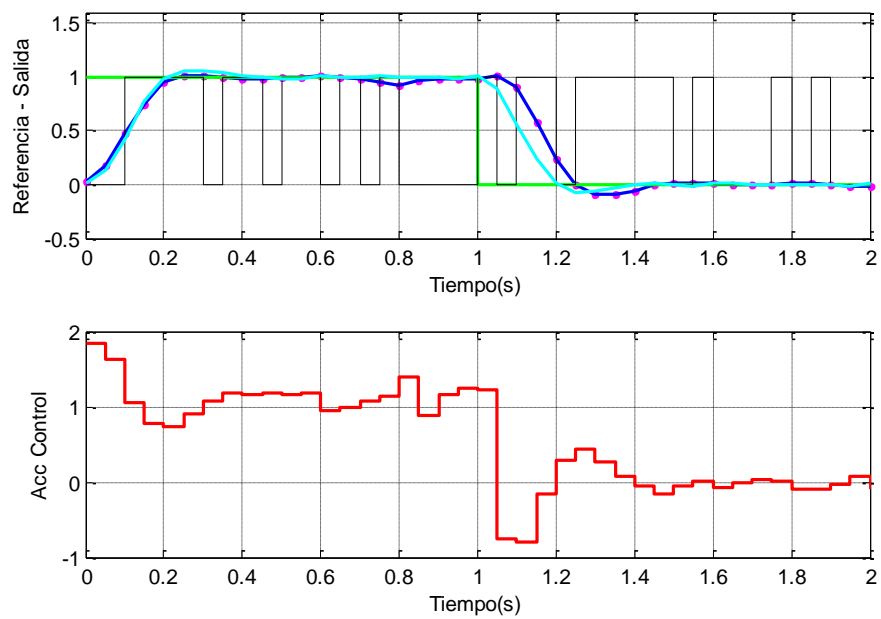


Figura 109. Comportamiento ante pérdida de datos utilizando el modo 4

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

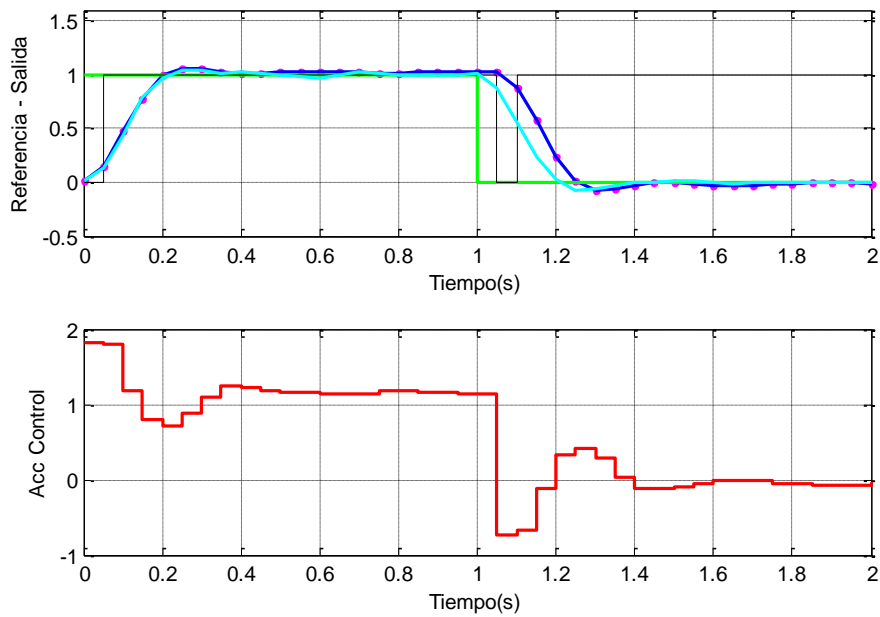


Figura 110. Comportamiento ante pérdida de datos utilizando el modo 5

En la siguiente figura se comparan las respuestas de la ejecución de los distintos modos de funcionamiento ante el mismo patrón de pérdidas de datos. En cada figura se muestran los valores numéricos de los errores cometidos en la respuesta real comparada con la respuesta sin pérdida de datos.

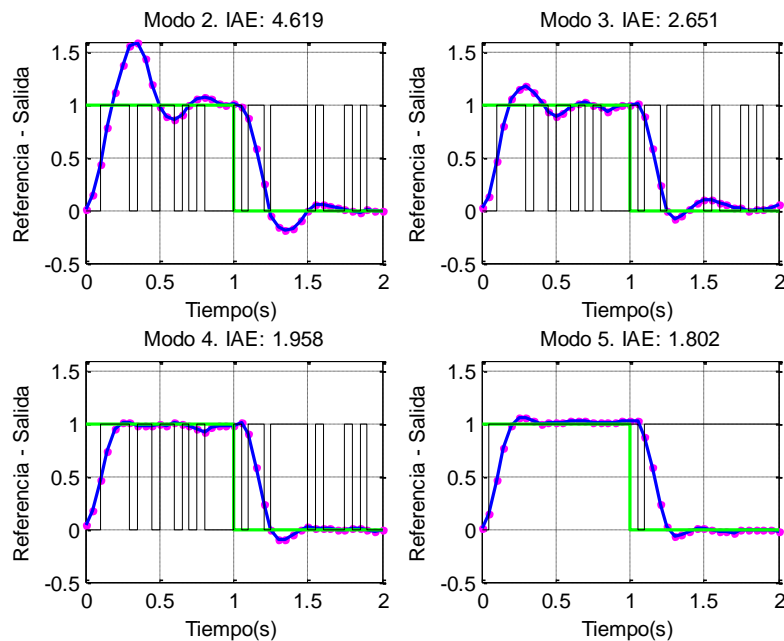


Figura 111. Respuesta real del proceso ante pérdidas de datos utilizando los distintos modos de funcionamiento

La respuesta del proceso real, ante la misma pérdida de datos para los distintos modos de funcionamiento, obtiene datos que son similares a los obtenidos en el estudio de las pérdidas porcentuales utilizando el peor caso. Los modos de funcionamiento que mejores resultados obtienen son aquellos que utilizan las acciones de control postuladas, y sobre todo los que además usan la técnica de la compensación de la acción de control, las salidas de los procesos en estos casos son muy parecidas a las obtenidas en ausencia de pérdidas.

En el caso del modo de funcionamiento 5 la pérdida de datos afecta en aquellos instantes en que se actualizan las trayectorias nominales que envía el control supervisor al control local. Como se muestra en la Figura 110, se produce un retraso en el envío de las trayectorias al coincidir el cambio de referencia con la pérdida de un dato.

A continuación se muestra el comportamiento del sistema cuando la salida tiene una perturbación senoidal de 50 rad/s de frecuencia y amplitud 0.025. El código de colores para la representación de las señales es el mismo que en el caso anterior (Figura 106).

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

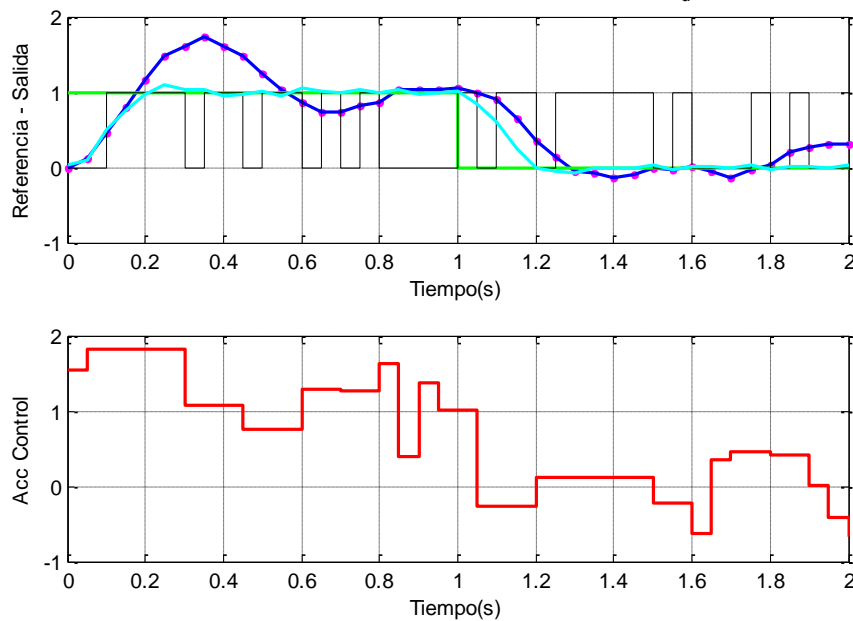


Figura 112. Comportamiento ante pérdida de datos utilizando el modo 2

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

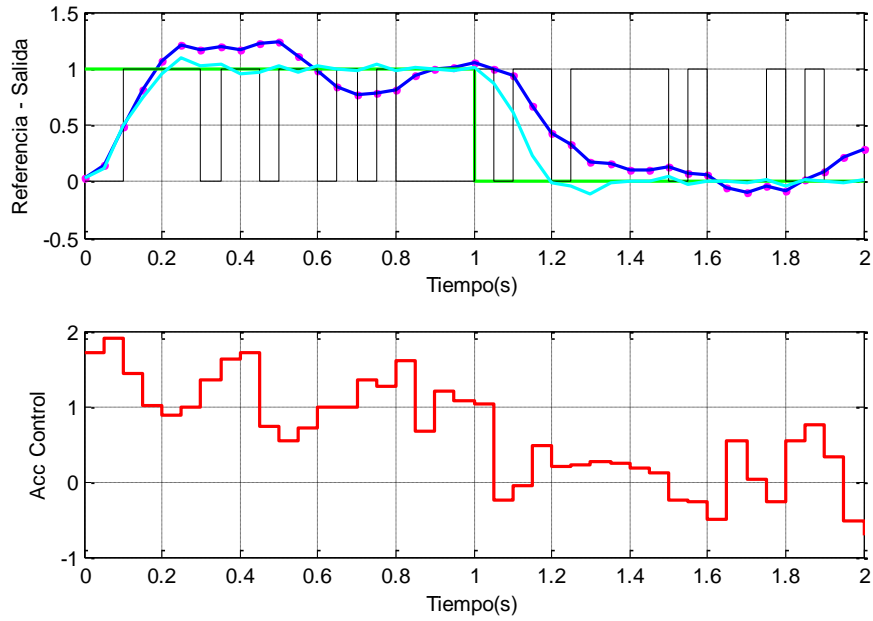


Figura 113. Comportamiento ante pérdida de datos utilizando el modo 3

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

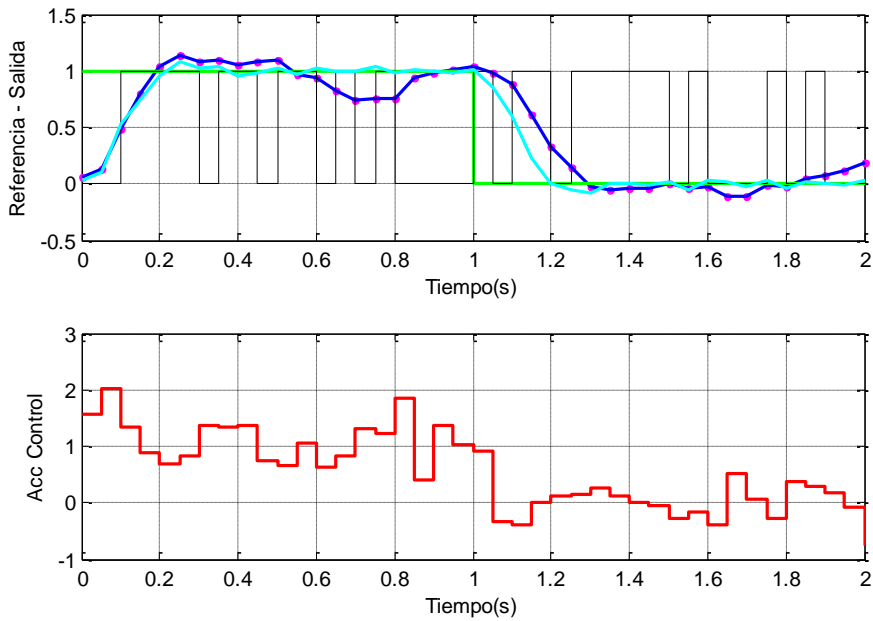


Figura 114. Comportamiento ante pérdida de datos utilizando el modo 4

$G_p(s)=[0 \ 0 \ 6.818]/[0.1021 \ 0.9588 \ 7.818]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=8 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.05$;

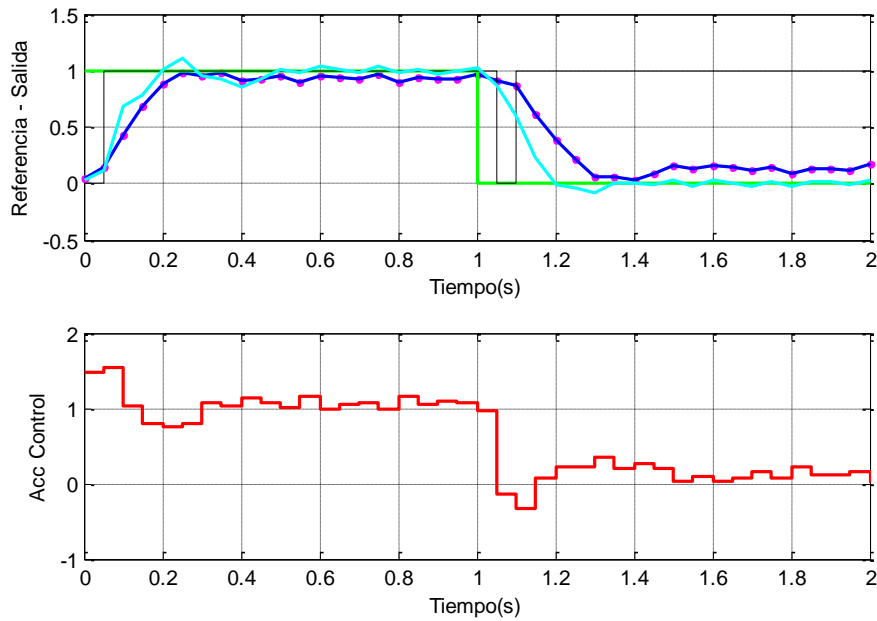


Figura 115. Comportamiento ante pérdida de datos utilizando el modo 5

La siguiente figura muestra las respuestas del proceso cuando se aplican los distintos modos de funcionamiento. Se han puesto los valores numéricos de los errores cometidos en la respuesta real comparada con la respuesta sin pérdida de datos.

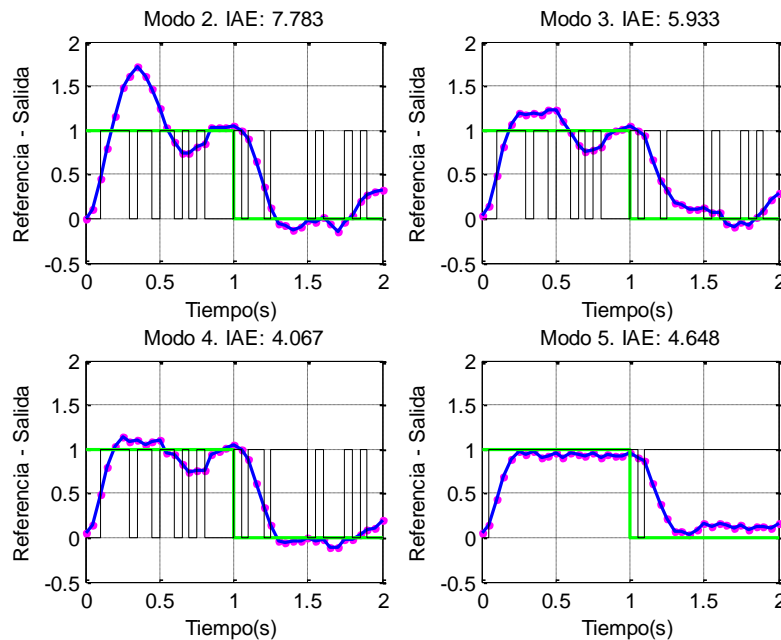


Figura 116. Respuesta real del proceso ante pérdidas de datos utilizando los distintos modos de funcionamiento

Los valores de los índices de error son mayores a los obtenidos sin la perturbación. Tal y como ocurría anteriormente los modos de funcionamiento que utilizan las

trayectorias postuladas son los que obtienen un mejor resultado. En esta ocasión el modo de funcionamiento 4 es el que obtiene una respuesta más parecida al caso en el que no se tienen pérdidas de datos.

6.3.2 Control de posición de un motor de corriente continua

En esta prueba se han utilizado un PC como nodo supervisor que implementa el middleware completo, y para el nodo local se utiliza un dsPIC que implementa el middleware reducido del núcleo de control, siendo el sistema de comunicaciones entre los nodos implementado mediante CAN. En la siguiente figura se muestra el esquema utilizado.

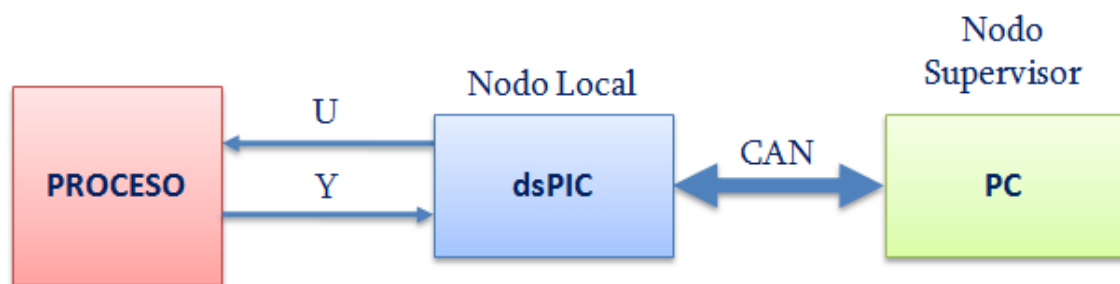


Figura 117. Estructura de control distribuida utilizada en la prueba

El proceso a controlar es la posición del eje de un motor de corriente continua del fabricante Feedback. La Figura 118 muestra el motor real utilizado en esta prueba.

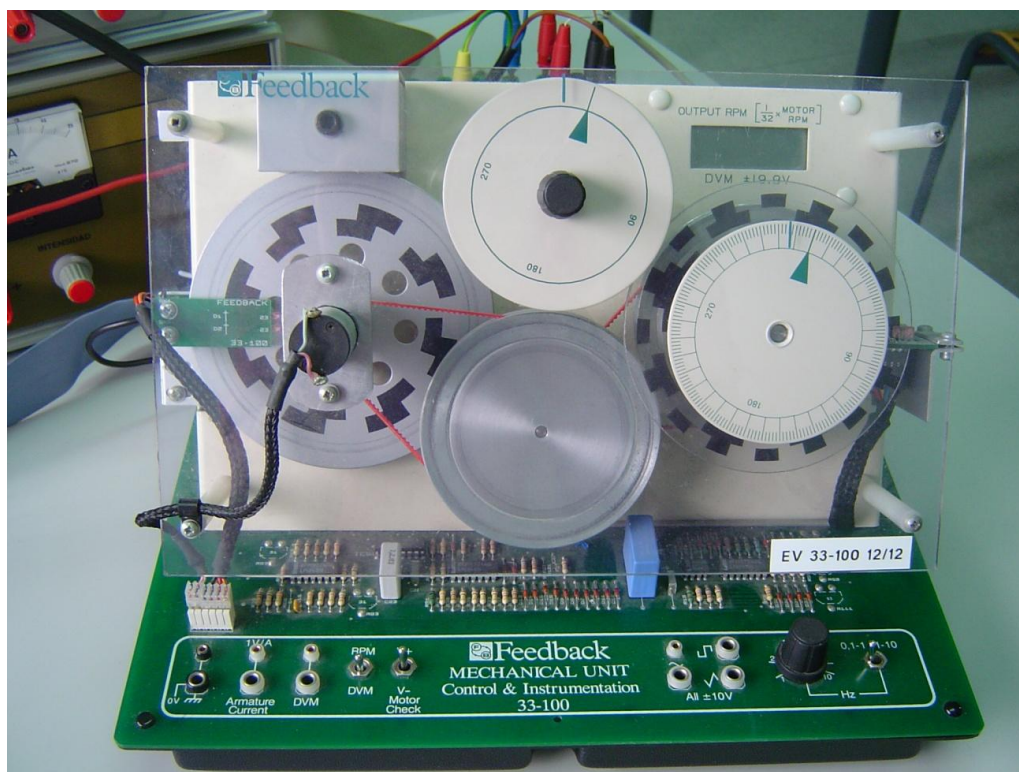


Figura 118. Motor de corriente continua utilizado para el control de posición

La función de transferencia del modelo de posición del motor, así como los parámetros a implementar en los controladores de cada nodo, se encuentran descritos en el anexo B, apartado B.4. Para esta prueba se ha utilizado un control supervisor CRHPC con los siguientes parámetros: $N=6$, $m=2$, $N_u=6$, $K=4$.

A continuación se muestra el comportamiento del proceso utilizando el sistema distribuido de control sin pérdida de datos. Se han puesto unas referencias de tensión entre 1.8984 y 0.1794 voltios, lo que corresponde con una referencia entre 30° y 0° .

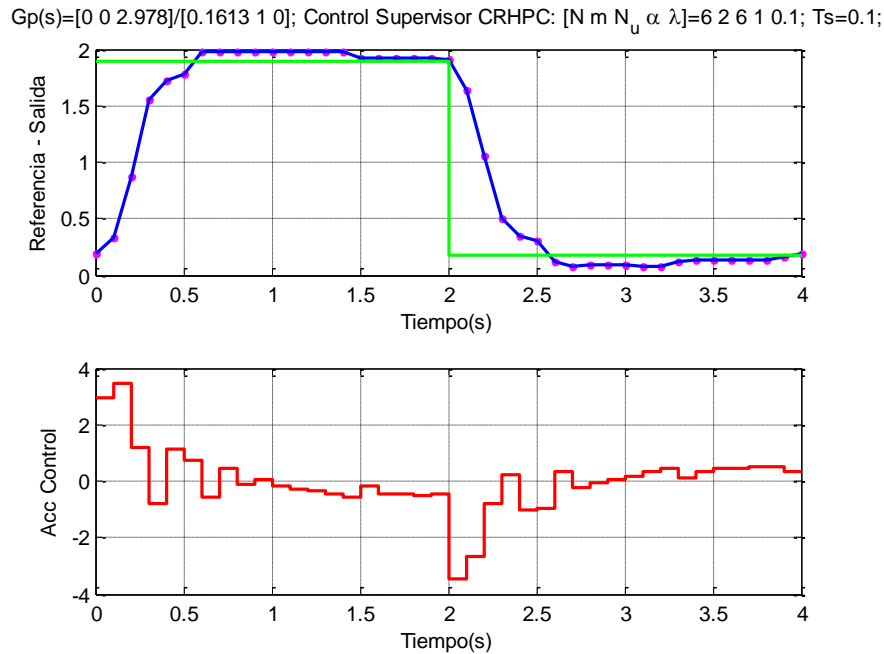


Figura 119. Comportamiento del proceso real con el sistema distribuido de control sin pérdida de datos

A continuación se muestran las pruebas realizadas sobre el proceso real cuando se producen pérdidas de datos con los distintos modos de funcionamiento.

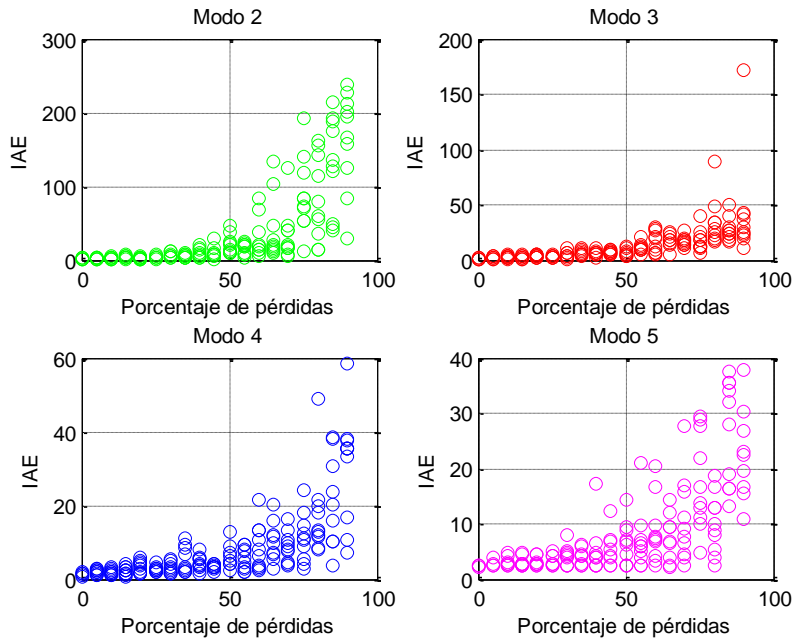


Figura 120. IAE de los distintos modos de funcionamiento en la prueba real

En las siguientes figuras se muestra la comparación de los valores máximos y medios, en cada porcentaje de pérdida de datos, con los distintos modos de funcionamiento.

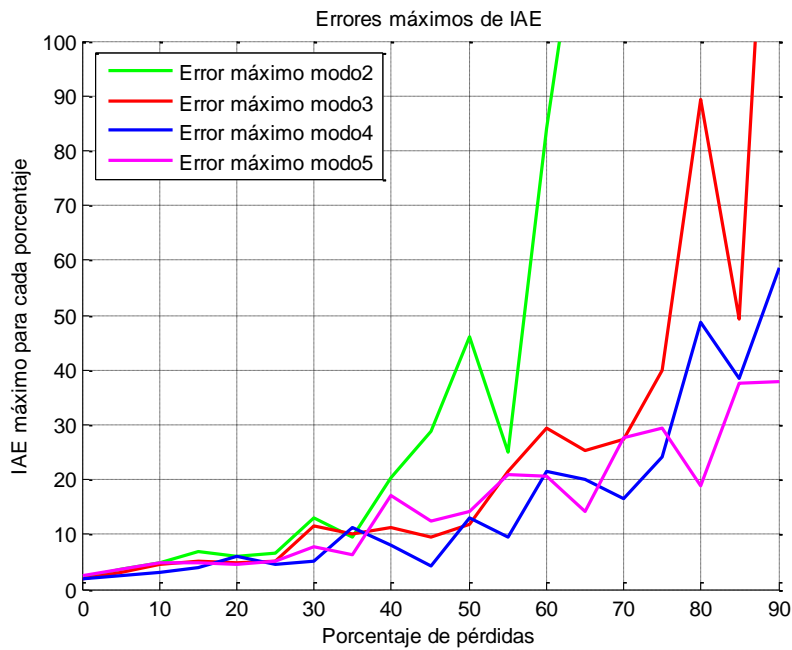


Figura 121. IAE máximos para cada modo de funcionamiento en la prueba real

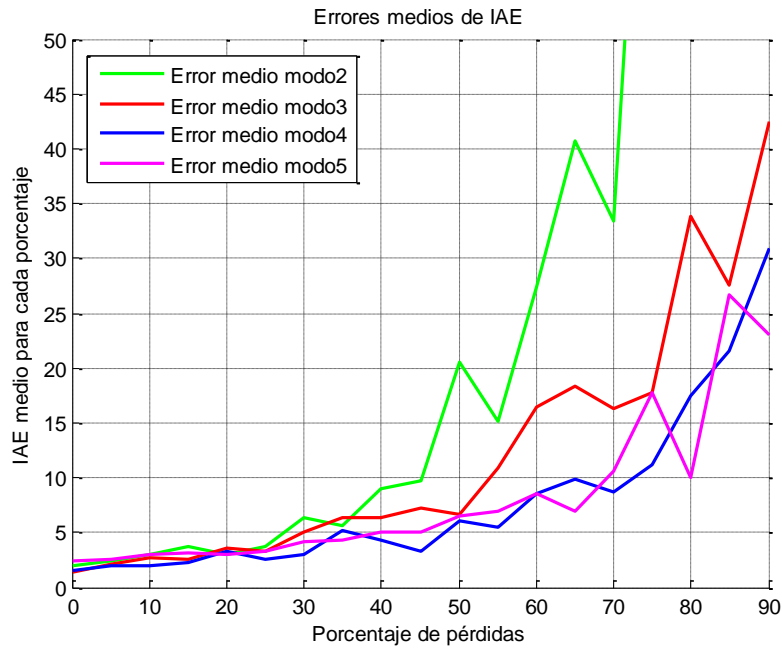


Figura 122. IAE medios para cada modo de funcionamiento en la prueba real

El comportamiento del sistema real es similar al obtenido, con las pruebas simuladas, por el control supervisor CRHPC en el caso del canal de comunicaciones compartido, que se muestra en el anexo B Figura 214 y Figura 215. Los modos de funcionamiento que utilizan las trayectorias postuladas por el control supervisor, son los que obtienen unos índices de error más pequeños, sobre todo para grandes pérdidas de datos, siendo el modo 4 el que consigue un índice de error más pequeño.

A continuación se muestra el comportamiento de la salida del sistema ante el mismo patrón de pérdidas cuando se utilizan los distintos modos de funcionamiento. El porcentaje de pérdidas que provoca este patrón es de un 40%. El código de colores para la representación de las señales se muestra en la Figura 106.

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[6 \ 2 \ 6 \ 1 \ 0.1]$; $T_s=0.1$;

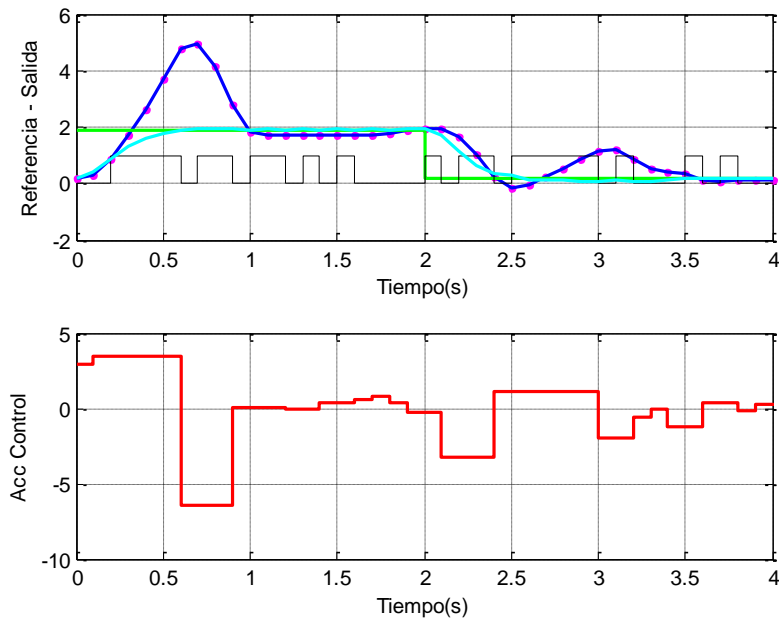


Figura 123. Comportamiento ante pérdida de datos utilizando el modo 2

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[6 \ 2 \ 6 \ 1 \ 0.1]$; $T_s=0.1$;

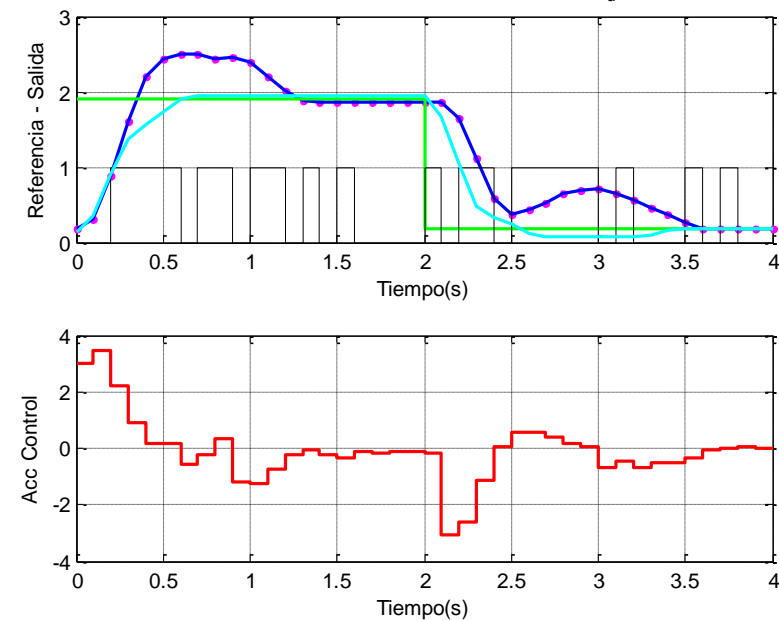


Figura 124. Comportamiento ante pérdida de datos utilizando el modo 3

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[6 \ 2 \ 6 \ 1 \ 0.1]$; $T_s=0.1$;

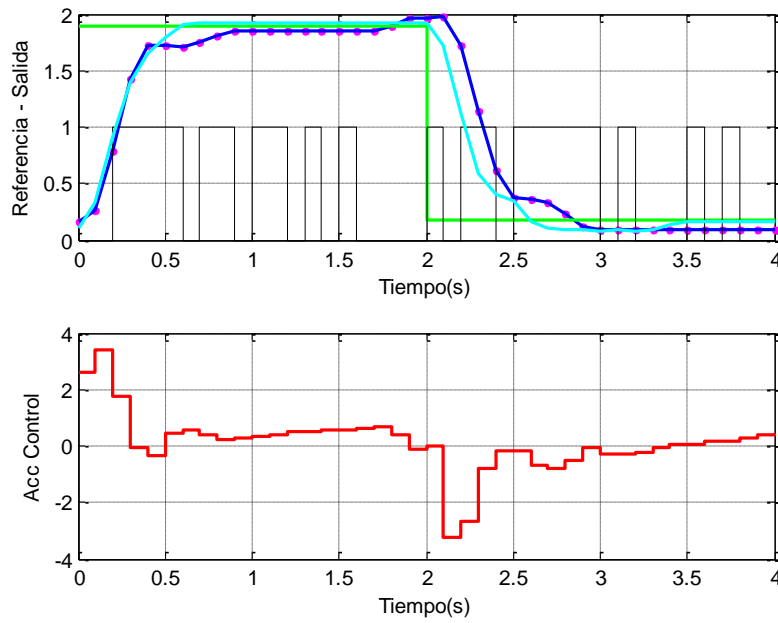


Figura 125. Comportamiento ante pérdida de datos utilizando el modo 4

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[6 \ 2 \ 6 \ 1 \ 0.1]$; $T_s=0.1$;

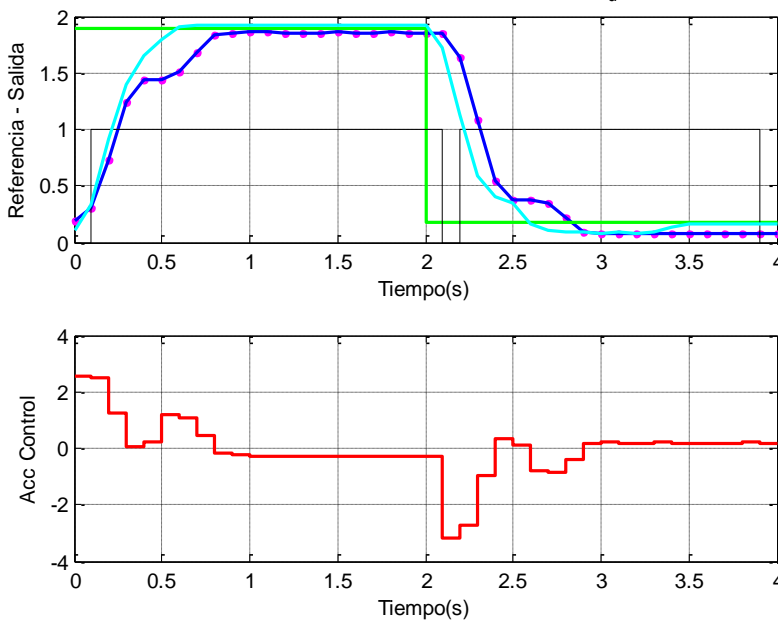


Figura 126. Comportamiento ante pérdida de datos utilizando el modo 5

La siguiente figura muestra la comparación de la salida del proceso cuando se utilizan los distintos modos de funcionamiento. En cada figura se incluyen los valores numéricos de los errores cometidos en la respuesta real del proceso comparada con la respuesta del sistema sin pérdida de datos.

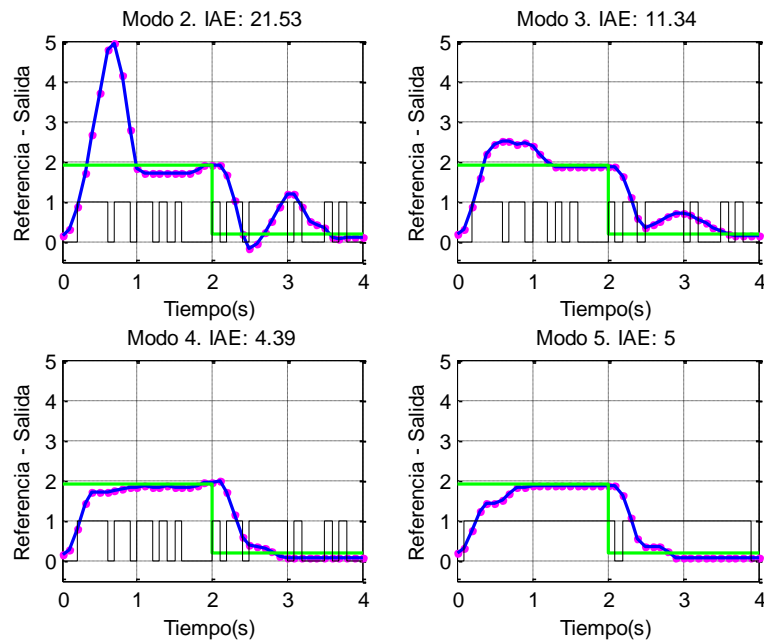


Figura 127. Respuesta real del proceso ante pérdidas de datos utilizando los distintos modos de funcionamiento

En este caso el comportamiento ante ese patrón de pérdidas del 40% muestra como los modos de funcionamiento que utilizan la compensación de la acción de control consiguen un índice de error más pequeño que el modo de funcionamiento 2, en cualquier caso, todos los modos de funcionamiento consiguen estabilizar la respuesta del proceso. Tal y como ocurría con el ejemplo anterior, el modo de funcionamiento 4, que es el que utiliza la compensación de la acción de control, es el que consigue un índice de error más pequeño.

A continuación se muestran las pruebas realizadas, cuando se pone el freno en el motor, con el mismo patrón anterior. Cuando se aplica el freno en el motor la dinámica del sistema cambia, por lo tanto, esta prueba permite comprobar la robustez del sistema distribuido de control ante cambios en el modelo.

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[6 \ 2 \ 6 \ 1 \ 0.1]$; $T_s=0.1$;

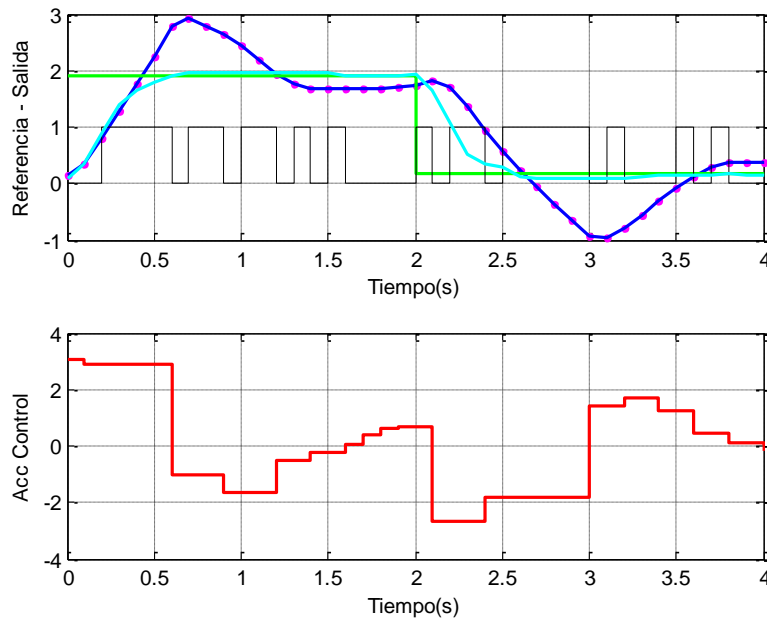


Figura 128. Comportamiento ante pérdida de datos utilizando el modo 2

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[6 \ 2 \ 6 \ 1 \ 0.1]$; $T_s=0.1$;

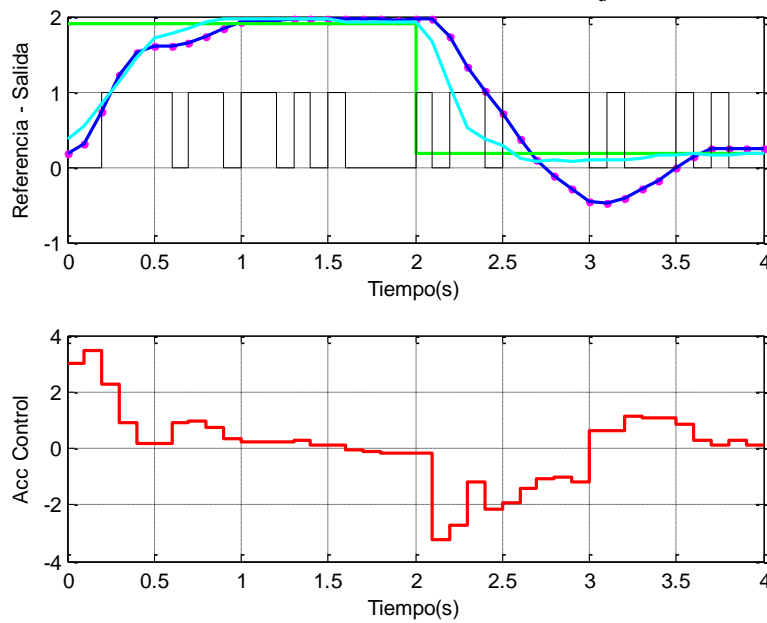


Figura 129. Comportamiento ante pérdida de datos utilizando el modo 3

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[6 \ 2 \ 6 \ 1 \ 0.1]$; $T_s=0.1$;

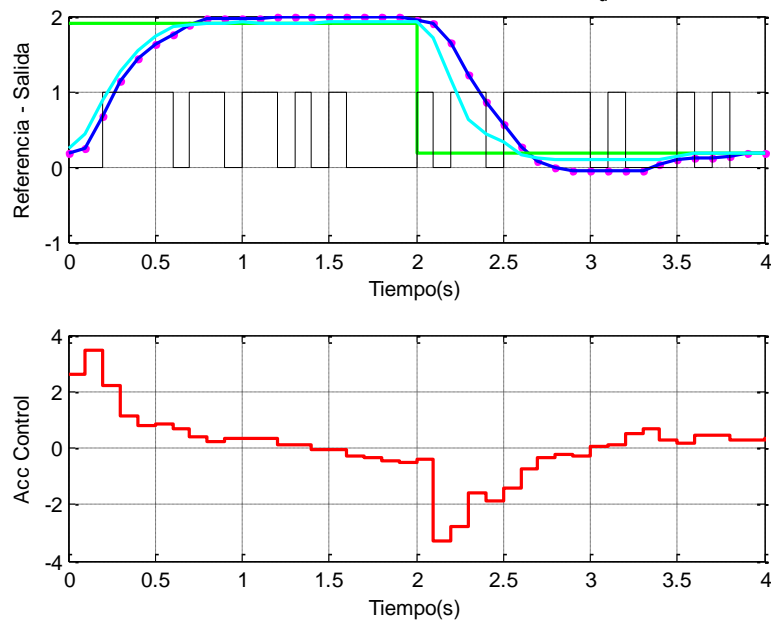


Figura 130. Comportamiento ante pérdida de datos utilizando el modo 4

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[6 \ 2 \ 6 \ 1 \ 0.1]$; $T_s=0.1$;

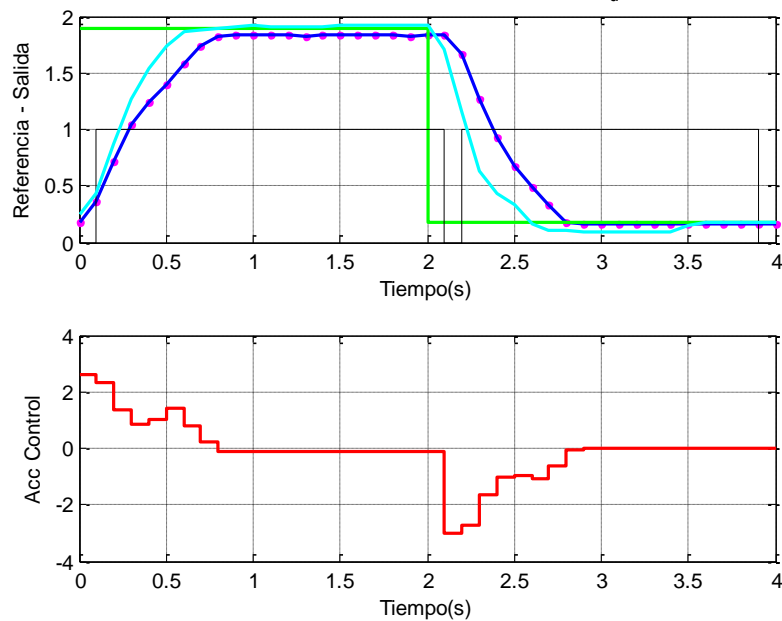


Figura 131. Comportamiento ante pérdida de datos utilizando el modo 5

La siguiente figura muestra la comparación de la salida del proceso cuando se utilizan los distintos modos de funcionamiento. En cada figura se incluyen los valores numéricos de los errores cometidos en la respuesta real del proceso comparada con la respuesta del sistema sin pérdida de datos.

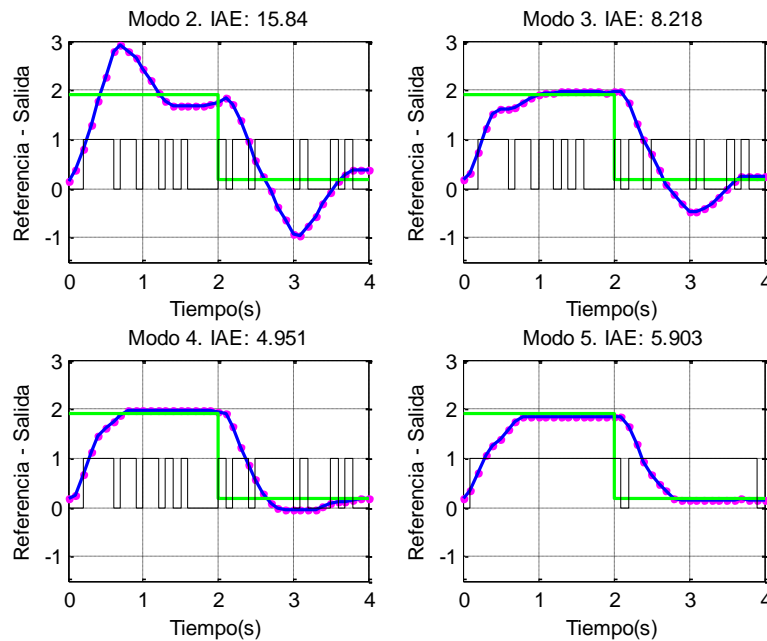


Figura 132. Respuesta real del proceso ante pérdidas de datos utilizando los distintos modos de funcionamiento

El comportamiento del proceso con el freno ante las pérdidas de datos del 40% muestra un comportamiento similar al proceso sin frenar presentado anteriormente. En esta ocasión la dinámica del sistema se hace más lenta y los picos al utilizar el modo 2 no son tan acentuados, en cualquier caso, todos los modos de funcionamiento consiguen estabilizar la respuesta del proceso, siendo nuevamente el modo de funcionamiento 4 el que consigue unos resultados mejores.

En las pruebas realizadas con la herramienta de simulación con el motor de corriente continua, el comportamiento de la salida difiere ligeramente del presentado en las pruebas reales. Estas discrepancias son debidas a las características del sistema real que no se han tenido en cuenta en el modelo del proceso:

- Comportamiento no lineal del motor debido a la zona muerta
- El sensor con el que se obtiene la medida de la posición no tiene un comportamiento lineal, teniendo un punto de ruptura en 180° .
- Limitación de la acción de control

Todas estas características pueden ser tenidas en cuenta en el modelo, sin embargo, tal y como se ha mostrado en las pruebas anteriores, la utilización de un modelo lineal sencillo permite obtener unos resultados muy aceptables.

6.4 Resumen y conclusiones del capítulo

En este capítulo se ha mostrado la aplicación del sistema distribuido de control sobre diversos tipos de procesos.

En el primero de ellos se aplica un sistema de compensación de la acción de control sobre un robot móvil, utilizando las trayectorias previamente calculadas por el control supervisor en caso de falta de datos.

En las siguientes pruebas se realiza la aplicación del sistema distribuido de control sobre dos procesos que se han estudiado en simulación. Los dos procesos reales son el proceso de segundo orden con respuesta subamortiguada ante escalón y el control de posición de un motor de corriente continua.

En las pruebas realizadas se ha puesto de manifiesto el buen funcionamiento del sistema distribuido de control, con el que se puede hacer frente a las pérdidas de datos mediante el uso de los distintos modos de funcionamiento.

La utilización de las acciones de control postuladas muestran un comportamiento muy adecuado en caso de pérdida de datos, sobre todo cuando se utiliza la compensación de la acción de control, ya que permiten al sistema retrasar su paso a una situación inestable y obtener unos índices de error bajos, incluso con grandes pérdidas de datos.

Parte IV. Conclusiones

7 CONCLUSIONES Y TRABAJO FUTURO

7.1 Conclusiones

El objetivo general de esta tesis ha sido desarrollar una estrategia de control que permita, en un sistema distribuido con elementos empotrados, aplicar controladores de alto nivel sobre el proceso. Para ello, tras revisar los trabajos relativos al control empotrado en sistemas distribuidos, se ha propuesto la utilización de controladores locales, con escasos recursos de computación, y controles supervisores que implementan controladores de alto nivel.

La estrategia de control ha de hacer frente a los problemas derivados de un sistema distribuido en el que el canal de comunicaciones es compartido y en el que algunos nodos tienen escasos recursos de computación, por lo tanto, se han propuesto soluciones a los retardos, falta de datos, etc.. Para ello se ha utilizado el núcleo de control, desarrollado en los proyectos de investigación KERTROL y SIDIRELI, y sus versiones del middleware completo y reducido que serán implementados en los nodos supervisores y locales, respectivamente.

La estrategia de control, que es aplicada por el control local sobre el proceso, se basa en la utilización de las acciones de control predichas por un control de alto nivel implementado en el nodo supervisor. Además, para evitar los problemas derivados de la utilización de una estrategia de control en bucle abierto, el control supervisor también envía las trayectorias de las salidas estimadas. Por lo tanto, el control local puede utilizar estas salidas como una trayectoria nominal a seguir, de forma que en caso de fallo en la recepción de las trayectorias podría utilizar esta trayectoria para compensar la acción de control postulada y evitar errores.

La compensación de la acción de control se ha mostrado como un buen método para hacer frente a la pérdida de datos en comparación con otras posibles soluciones.

Además, se han definido diferentes modos de funcionamiento que permiten al sistema distribuido adecuarse a distintas situaciones. Con cada modo de funcionamiento se utilizan diferentes mensajes, y se puede ahorrar tiempo de cómputo en las comunicaciones. Un método que es especialmente sencillo es aquél que utiliza la información postulada hasta recibir las nuevas directrices. En este caso, el consumo por el canal de comunicaciones se simplifica enormemente. Este modo de funcionamiento es similar a un control basado en eventos, en el que el evento sería el envío de las nuevas trayectorias a seguir por el control local debido a un cambio en la trayectoria nominal.

La herramienta de simulación del sistema distribuido permite obtener los parámetros más adecuados para la implementación de los controles supervisor y local. Además, con esta herramienta se puede conocer el ancho de banda mínimo dedicado al control, el tiempo de cómputo y la magnitud de las perturbaciones, para no tener problemas en las comunicaciones entre el nodo supervisor y el nodo local.

En cualquier caso la estrategia de control está preparada para hacer frente a todos los problemas derivados de las comunicaciones.

Para evaluar las prestaciones del sistema distribuido de control con los distintos modos de funcionamiento ante pérdidas de datos, se han utilizado procesos con distinta dinámica, realizando pruebas tanto simuladas como reales. Ambas pruebas presentan un comportamiento similar, ello ha permitido validar la herramienta de simulación y al mismo tiempo comprobar las prestaciones del sistema ante la presencia de pérdida de datos.

7.2 Trabajo futuro

Una vez desarrollado el trabajo para un sistema distribuido formado por un control supervisor y un control local, se plantea la ampliación de los resultados a otro tipo de sistemas:

- Ampliación de la metodología propuesta y la estrategia de control para un sistema distribuido en el que los procesos sean MIMO.
- Sistemas formados por varios nodos locales, en el que se pueden plantear el balance de carga computacional para poder hacer frente a los requisitos del sistema. En este caso, el supervisor sería el encargado de cambiar las estrategias de control de los nodos locales, para repartir la carga computaciones entre los nodos del sistema.
- Sistemas formados por varios nodos locales, en el que el nodo supervisor, además de hacer llegar los vectores correspondientes a las trayectorias a seguir, debe coordinar el comportamiento de los distintos nodos del sistema. Una aplicación muy interesante en este tipo de sistemas puede ser el campo

de los robots humanoides, formados por múltiples servos, en el que el control global del sistema debe coordinar perfectamente las operaciones de cada servo para que el robot humanoide realice las funciones deseadas. Una operación como la de andar supone la coordinación de todos los servos de robot, por lo que en caso de una perturbación, como por ejemplo un desequilibrio del robot, se deberían tener mecanismos para hacer llegar las nuevas trayectorias, y que el robot pudiera hacer frente a las mismas.

Una tarea a desarrollar es la realización de un análisis teórico del método de la compensación de la acción de control, combinado con la recepción irregular de los datos. Además, se puede plantear la inclusión de otro tipo de ajuste de los datos en el control local, como por ejemplo reguladores PID, de mínima varianza,...que mejoren la respuesta de la compensación de la acción de control.

Por último, la mejora de la herramienta desarrollada para la asistencia al diseño de reguladores siguiendo el modelo propuesto, en el que se añadan las características descritas anteriormente, así como una interfaz gráfica que mejore la experiencia del usuario.

Parte V. Bibliografía

Bibliografía

Abdelzaher, T.F., J.A. Stankovic, C. Lu, R. Zhang, y Y. Lu. «Feedback Performance Control in Software Services.» *IEEE Control Systems Magazine* 23, nº 3 (2003): 74-90.

Akyildiz, I.F., W. Su, Y. Sankarasubramaniam, y E. Cayirci. «Wireless Sensor Networks: A Survey.» *Computer Networks* 38, nº 4 (2002): 393-422.

Akyildiz, I.F., y I.H. Kasimoglu. «Wireless Sensor and Actor Networks: Research Challenges.» *Ad Hoc Networks Journal* (Elsevier), October 2004: 351-367.

Akyildiz, Ian F., y Ismail H. Kasimoglu. «Wireless Sensor and Actor Networks: Research Challenges.» *Ad Hoc Networks* 2, 2004: 351-367.

Albertos, P, A. Crespo, y J. Simó. «Control Kernel: A key concept in embedded control systems.» *4th IFAC Symposium on Mechatronic Systems*. 2006.

Albertos, P., A. Crespo, M. Vallés, y I. Ripoll. «Embedded Control Systems: Some Issues and Solutions.» *16th IFAC World Congress*. Prague, Czech Republic: Elsevier, 2005.

Aracil, R. «Teleoperación.» *Enseñanza vía Internet / Web de la Ingeniería de Sistemas y Automática. EIWISA '02*. Alicante, 2002.

ARTIST. «ARTIST. Advanced Real-Time Systems.» *Roadmaps for Embedded Systems Design*. 2008. <http://www.artist-embedded.org/artist/-Roadmaps-.html>.

ARTIST WCES. 2007. 2nd International ARTIST Workshop on Control for Embedded Systems. <http://www.artist-embedded.org/artist/Overview,917.html>.

ARTIST2 GCECS. Mayo de 2008. Graduate Course on Embedded Control Systems. Stockholm, Sweden. <http://www.artist-embedded.org/artist/ARTIST2-Graduate-Course-on,1323.html>.

Ayala, J.L., A.G. Lomeña, y M. López-Vallejo. «Aplicación de metodologías de co-diseño HW/SW de sistemas empotrados para el diseño e implementación de un control de accesos distribuido.» *TELEC 02*. Santiago de Cuba (Cuba), 2002.

Balarin, F., y otros. *Hardware-Software Co-Design of Embedded Systems: The Polis Approach*. Kluwer Academic Press, 1997.

Baliga, G., y P.R. Kumar. «A middleware for control over networks.» *Proceedings of the IEEE Conference on Decision & Control, including the Symposium on Adaptive Processes*. 2005.

Bemporad, A., F. Borelli, y M. Morari. «Model Predictive Control Based on Linear Programming - The Explicit Solution.» *IEEE Transactions on Automatic Control* 47, nº 12 (December 2002): 1974-1985.

Bemporad, Alberto, Manfred Morari, Vivek Dua, y Efstratios N. Pistikopoulos. «The Explicit Linear Quadratic Regulator for Constrained Systems.» *Automatica* (Pergamon), nº 38 (2002): 3-20.

Benchmark CEA. 2010-2011. Grupo Temático de Ingeniería de Control de CEA. Evaluación de algoritmos de auto-ajuste de controladores PID. http://www.cea-ifac.es/w3grupos/ingcontrol/repositorio/ingcontrol/benchmark_autoajuste_pid.pdf.

Benitez, Ignacio, José Luis Díez, Raúl Simarro, y Pedro Albertos. «Diseño de agentes supervisores para el control de robots móviles.» *6th International Workshop on Practical Applications of Agents and Multiagents Systems. IWPAAMS*. Universidad de Salamanca, 2007.

Benyon, D. «Accommodating Individual Differences through an Adaptive User Interface.» Editado por M., Kühme, T. and Malinowski, U. (eds.) Schneider-Hufschmidt. *Adaptive User Interfaces - Results and Prospects* (Elsevier Science Publications), 1993.

Berger, A.S. *Embedded Systems Design: And Introduction to Processes, Tools and Techniques*. CMP Books, 2002.

Bordóns, C. *Control predictivo generalizado de procesos industriales: formulaciones aproximadas*. PhD thesis, Escuela Técnica Superior de Ingenieros Industriales. Universidad de Sevilla, 1994.

Boucher, Charles, y Robert Lensch. *MEMS Tchnology. Nearing the Tipping Point (2nd Edition)*. Report, Boucher-Lensch Associates LLC, 2010.

Bouyssounouse, Bruno, y Joseph Sifakis. *Embedded Systems Design. The ARTIST Roadmap for Research and Development*. Springer, 2005.

Braitenberg, V. *Vehicles*. MIT Press, 1984.

Buendía, M.J., J.A. Vera Repullo, F. Losilla López, y P.J. Meseguer. «Redes de Sensores y Actuadores (WSAN) en domótica.» *Telecoforum*, 2007.

Bullo, Francesco, Jorge Cortés, y Sonia Martínez. *Distributed Control of Robotic Networks. A Mathematical Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.

Camacho, E.F., y C. Bordóns. *Model Predictive Control*. Springer, 2004.

Casanova, Vicente. *Sistemas de Control Basados en Red. Modelado y Diseño de Estructuras de Control*. PhD thesis, Departamento de Ingeniería de Sistemas y Automática. Universidad Politécnica de Valencia, 2005.

Catsoulis, J. *Designing Embedded Hardware, Second Edition*. O'Reilly, 2005.

Cauffriez, L., J. Ciccotelli, B. Conrard, y M. Bayart. «Design of Intelligent Distributed Control Systems: A Dependability Point of View.» *Reliability Engineering and System Safety* (Elsevier), nº 84 (2004): 19 - 32.

Chou, P.H., R.B. Ortega, y G. Borriello. «The Chinook Hardware/Software Co-Synthesis System.» *8th International Symposium on Systems Synthesis*. Cannes, France, 1995. 22-27.

Clarke, D.W, y R. Scattolini. «Constrained Receding-Horizon Predictive Control.» *IEE Proceedings* 138, nº 4 (July 1991): 347-354.

Clarke, D.W., C. Mohtadi, y P.S. Tuffs. «Generalized predictive control, parts I and II.» *Automatica*, 1987: vol.23, nº2, pp 137-160.

Coronel, J., F. Blanes, G. Benet, P. Pérez, y J.E. Simó. «CAN-Based Distributed Control Architecture using the SCoCAN Communication Protocol.» *Proc. IEEE International Conference on Emerging Technologies and Factory Automation*. 2005.

Coronel, J.O, F. Blanes, J. Simó, y V. Nicolau. «Middleware de Kernel de Control Para el Desarrollo de Aplicaciones en Sistemas Empotrados de Tiempo Real.» *XXIX Jornadas de Automática*. Tarragona, 2008.

Coronel, Javier. «Porting of RTLinux 3.2rc1 to Xscale processors.» 2007. <http://rtportal.upv.es/apps/xscale/> (último acceso: 2010).

Crespo, A., P. Albertos, P. Balbastre, M. Vallés, M. Lluesma, y J.E. Simó. «Schedulability issues in complex embedded control systems.» *Proceedings of the 2006 IEEE Conference on Computer Aided Control Systems Design*. Munich, Germany, 2006.

Davis, R., y Smith R.G. «Negotiation as a Metaphor for Distributed Problem Solving.» *Artificial Intelligent* 20 (1983): 63-109.

Dorf, R., y R. H. Bishop. *Modern Control Systems, Seventh Edition*. Addison-Wesley, 1995.

Dormido, S., J. Sánchez, y E. Kofman. «Muestreo, control y comunicación basados en eventos.» *Revista Iberoamericana de Automática e Informática Industrial (RIAI)* 5, nº 1 (2008): 5-26.

Ducatel, K, M. Bogdanowicz, F. Scapolo, J. Leijten, y J.C. Burgelma. «Istag: Scenarios for Ambient Intelligent in 2010.» *ISTAG 2001 Final Report*. 2001.

Friedewald, M., y O. Da Costa. «Science and Technology Roadmapping: Ambient Intelligent in Everyday Life.» *JRC/IPTS - ESTO Study*. 2003.

Garlan, D., D.P. Siewiorek, A. Smailagic, y P. Steenkiste. «AURA: Toward Distraction-Free Pervasive Computing.» *IEEE Pervasive Computing* 1, nº 2 (2002): 22-31.

Graham, D., y R.C. Lathrop. «The Synthesis of optimum transient response: Criteria and standard forms.» *AIEE Transactions Part II: Applications and Industry*, nº 72 (1953): 273-288.

Graham, S.R. *Issues in the Convergence of Control with Communication and Computation*. Champaign, IL, USA: PhD thesis, University of Illinois at Urbana-Champaign, 2004.

Grehan, R., I Cyliax, y R. Moote. *Real-Time Programming: A Guide to 32-Bit Embedded Development*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

Heath, S. *Embedded Systems Design. Second Edition*. Newness. Elsevier Science, 2003.

IEEE eLearning. 2010. Technical Glossary. <http://iee-elearning.org/mod/glossary/view.php?id=434>.

Issarny, V., y otros. «Developing Ambient Intelligence Systems: A Solution Based on Web Services.» *Journal of Automated Software Engineering* 12 (2005).

JFE Technical Report. «Middlestar Middleware for Process Control Computer System.» 2007, No. 10 Dec. pp 56-58.

Lee, E.A. «Cyber-Physical Systems - Are Computing Foundations Adequate?» *Position paper for NSF Workshop on Cyber-Physical Systems*. Austin, Texas, 2006.

Leitao, Paulo. «Agent-Based Distributed Manufacturing Control: A State-of-the-Art Survey.» *Engineering Applications of Artificial Intelligence*, nº 22 (2009): 979-991.

Li, Q., y C. Yao. *Real-Time Concepts for Embedded Systems*. CMP Books, 2003.

Lian, F.L., J.R. Moyne, y D.M. Tilbury. «Performance evaluation of control networks.» *IEEE Control Systems Magazine* 21, nº 1 (2001): 66-83.

Liggesmeyer, Peter, y Mario Trapp. «Trends in Embedded Software Engineering.» *IEEE Software* 26, nº 3 (2009): 19-25.

MacLeod, I M. «Using Real-Time to Achieve Coordination in Distributed Computer Control Systems.» *Proc. IFAC World Congress*. Budapest, Hungary, 1984.

MacLeod, I.M., y V. Lun. «An Assessment of the Negotiation Metaphor for Flexible Manufacturing Control.» *Engineering Applications of Artificial Intelligence* (Elsevier Science Ltd.) 4 (1991): 167 - 175.

Mainwaring, Alan, Joseph Polastre, Robert Szewczyk, David Culler, y John Anderson. «Wireless Sensor Networks for Habitat Monitoring.» *Proc. Wireless Sensor Networks and Applications (WSNA '02)*. Atlanta, Ga, USA, 2002. 88-97.

Marcy, H.O., J.R. Agre, C. Chien, L.P. Clare, N. Romanov, y A. Twarowski. «Wireless Sensor Networks for Area Monitoring and Integrated Vehicle Health Management Applications.» *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*. Portland, Ore, USA: Collection of Technical Papers, 1999.

Marlin, T. *Process Control: Designing Processes and Control Systems for Dynamic Performance*. New York: McGraw Hill, 2000.

Martí, Pau, Manel Velasco, Josep M. Fuertes, Antonio Camacho, y Giorgio Buttazzo. «Design of an Embedded Control System Laboratory Experiment.» *IEEE Transactions on Industrial Electronics* 57, nº 10 (October 2010): 3297-3307.

Marwedel, Peter. *Embedded System Design*. Springer, 2006.

Microchip. «Digital Signal Controller. dsPIC30F Family Reference Manual.» *Microchip Technology Inc*. 2008. www.microchip.com (último acceso: 2010).

Murphy, R.R. *Introduction to AI Robotics*. A Bradford Book. The MIT Press, 2000.

NSF. 2006. NSF Workshop on Cyber-Physical Systems. <http://varma.ece.cmu.edu/CPS/>.

Object Management Group. «UML Profile for DDS Specification.» *OMG Technical Meeting*. Ottawa, Canada, 2008.

O'Hare, G.M.P., M.J. O'Grady, S. Keegan, D. O'Kane, R. Tynan, y D. Marsh. «Intelligent Agile Agents: Active Enablers for Ambient Intelligent.» *In Proceedings of the Ambient Intelligent for Science Discovery Conference*. 2004.

Ohlin, M., D. Henriksson, y A. Cervin. *TrueTime 1.5 - Reference Manual*. Lund University: Department of Automatic Control, 2007.

Parker, Lynne E. «Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems.» *Journal of Physical Agents* 2, nº 1 (2008).

Peng, Zhe, Longhua Ma, y Feng Xia. «A Low-Cost Embedded Controller for Complex Control Systems.» *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. Shanghai, 2008. 23-29.

Pont, M.J. *Embedded C*. Addison-Wesley, 2002.

Rajkumar, R., I. Lee, L. Sha, y J. Stankovic. «Cyber-Physical Systems: The Next Computing Revolution.» *Design Automation Conference (DAC), 2010 47th ACM/IEEE*. Anaheim, CA, 2010. 731-736.

Reeke, G.N., y G.M. Edelman. «Real Brains and Artificial Intelligence.» Editado por Ed S.R. Graubard. *The Artificial Intelligence Debate: False Starts, Real Foundations*, 1990: MIT Press.

Romn, M., C. Hess, R. Cergueira, R. Campbell, y K. Nahrstedt. «GAIA: A Middleware Infrastructure to Enable Active Spaces.» *IEEE Pervasive Computing*, 2002: 74-83.

Sanchís, J., M. Martínez, X. Blasco, y J. Senent. «Experimentación en control predictivo basado en modelos. Práctica para la enseñanza de controladores predictivos generalizados (GPCs) en la ETSII.» *XVII Jornadas de Automática*. Girona, Spain, 1997.

Sanz, R., y K.E. Arzen. «Trends in software and control.» *IEEE Control Systems Magazine*, 2003: vol. 23, nº 3, pp 12-13.

Schantz, R.E., D.C. Schmidt, Loyall J.P., y C. Rodrigues. «Controlling Quality-of-Service in Distributed Real-Time and Embedded Systems Via Adaptive Middleware: Experiences with Auto-adaptive and Reconfigurable Systems.» *Software—Practice & Experience* (John Wiley & Sons, Inc.) 36 (2006): 1189 - 1208.

Schmidt, D.C. «The ADAPTIVE Communication Environment: Object-oriented Network Programming Components for Developing Client/Server Applications.» 1993.

Schmidt, D.C., M. Stal, H. Rohnert, y F. Buschmann. *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. Wiley & Sons, 2000.

Schwierbert, L., S.K. Gupta, y J. Weinmann. «Research Challenges in Wireless Networks of Biomedical Sensors.» *Proc. 7th ACM International Conference on Mobile Computing Networking (MobiCOM'01)*. Rome, Italy, 2001. 151-165.

Scokaert, P.O.M., y D.W. Clarke. *Stability Questions in Predictive Control*. Technical Report OUEL 1970/93. Department of Engineering Science. University of Oxford., 1993.

Simarro, R., J. Coronel, J.E. Simó, y J.F. Blanes. «Hierarchical and Distributed Embedded Control Kernel.» *IFAC 2008 World Congress*. Seoul, Korea, 2008.

Simonot-Lion, Francois, y Yvon Trinquet. «New Solutions for In-Vehicle Embedded System Development.» *European Research Consortium for Informatics and Mathematics (ERCIM NEWS)*. *Special Theme: Embedded Systems*, nº 52 (2003): 18-19.

Stankovic, J.A., I. Lee, A. Mok, y R. Rajkumar. «Opportunities and Obligations for Physical Computing Systems.» *Computer (IEEE)* 38, nº 11 (November 2005): 23-31.

Stothert, A, y I.M. MacLeod. «Feature Extraction for Real-Time Expert Systems.» *Eng. Appl. Artificial Intelligence* 7 (1994): 361-366.

Stothert, A., y I.M. MacLeod. «Distributed Intelligent Control System for a Continuous-State Plant.» *IEEE Transactions on Systems, Man, and Cybernetics -Part B: Cybernetics* 27, nº 3 (1997): 395 - 401.

Technical University of Braunschweig. «MiRPA - Middleware for Robotic and Process Control Applications.» 2009. <http://www.rob.cs.tu-bs.de/en/research/projects/mirpa>.

Törngren, Martin, Dan Henriksson, Karl-Erik Arzen, Anton Cervin, y Zdenek Hanzalek. «Tool Supporting the Co-Design of Control Systems and Their Real-Time Implementation: Current Status and Future Directions.» *Conference on Computer Aided Control Systems Design*. Munich, Germany: Proceedings of the 2006 IEEE, 2006.

Valera, A., M. Vallés, P. Albertos, R. Simarro, I. Benítez, y C. Llácer. «Embedded Implementation of Mobile Robots Control.» *IFAC 2008 World Congress*. Seoul, Korea, 2008. 6821-6826.

Vallés, M. *Implementación de sistemas de control híbrido con recursos limitados de computación*. PhD thesis, Departamento de Ingeniería de Sistemas y Automática. Universidad Politécnica de Valencia, 2004.

Verhoef, Marcel. *Modeling and Validating Distributed Embedded Real-Time Control Systems*. PhD thesis, Rabboud University Nijmegen, 2008.

Wang, W., H.M. Huang, V. Subramonian, C. Lu, y C. Gill. «CAMRIT: Control-Based Adaptive Middleware for Real-Time Image Transmission.» *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. Toronto, Canada, 2004.

Weiser, M. «The Computer for the 21st Century.» *Scientific American* 265, nº 3 (1991): 66 - 75.

WIDEoNCS. Julio de 2009. 3rd WIDE PhD School on Networked Control Systems. Siena, Italy. <http://ist-wide.dii.unisi.it/school09/>.

Yodaiken, V., y M. Barabanov. «Introducing Real-Time Linux.» *Linux Journal*, 1997.

Yoon, T.W, y D.W. Clarke. «A reformulation of receding-horizon predictive control.» *Int J System Science* 26(7), 1995: 1383-1400.

Zhang, R., C. Lu, T.F. Abdelzaher, y A. Stankovic. «ControlWare: A Middleware Architecture for Feedback Control of Software Performance.» *International Conference on Distributed Computing Systems*. Vienna, Austria, 2002.

Parte VI. Anexos

Anexo A. Herramienta de cálculo y simulación del sistema distribuido de control

La simulación de los algoritmos de control del sistema distribuido, así como los métodos de elección de los parámetros de los controladores, ha necesitado el desarrollo de una herramienta que permitiera, la parametrización de dichos controladores y comprobar las distintas estrategias de control y modos de funcionamiento.

La herramienta está desarrollada en Matlab, lo que permite la utilización de funciones matemáticas, la simulación de estructuras de control con Simulink, la implementación de los algoritmos de control mediante funciones y las representaciones gráficas de una manera sencilla. Para poder simular la ejecución en los nodos del sistema distribuido, los tiempos de cómputo, el canal de comunicaciones distribuido, el ancho de banda, etc. se ha utilizado Truetime, que es una toolbox basada en Matlab/Simulink que permite simular el comportamiento de un núcleo de ejecución de tiempo real así como el medio de comunicación utilizado para la transmisión de señales. Los detalles correspondientes al uso de Truetime se pueden consultar en (Ohlin, Henriksson y Cervin 2007).

En este caso la herramienta permite la utilización de controladores predictivos en el nodo supervisor, para de esta forma obtener las trayectorias de las acciones de control y salidas postuladas. Las funciones existentes para el cálculo de las acciones de control de un controlador predictivo no contemplan la posibilidad de utilizar las acciones de control postuladas, ya que utilizan la estrategia del horizonte deslizante, y por lo tanto se han tenido que desarrollar nuevas funciones que permitan el cálculo de dichas trayectorias.

A continuación se exponen las características y opciones que permite la herramienta, la metodología utilizada para la realización de las simulaciones así como las

estructuras de control implementadas, que permiten tener en cuenta el tipo de comunicaciones existente entre los nodos del sistema distribuido.

A.1 Características de la herramienta

Una de las principales dificultades a la hora de probar la influencia de los distintos parámetros del control en el sistema distribuido es que el diseño de los controladores en el nodo supervisor es muy laborioso. En las pruebas realizadas se ha optado por la implementación de controladores predictivos, en los que hay que calcular las matrices que permitan el cálculo de las trayectorias de control y la salida, tal y como se explica en el anexo C. Dichas matrices son muy dependientes de los parámetros del control predictivo que se quiera diseñar y sobre todo del modelo del proceso que se quiera controlar, y por lo tanto, cualquier modificación en los parámetros del regulador o un cambio en el proceso hacen necesario recalcularse dichas matrices.

Un problema añadido es que al implementar un regulador predictivo se utiliza la estrategia del horizonte deslizante, y por lo tanto se utiliza tan sólo la señal de la acción de control calculada para el instante actual, desechando las acciones de control postuladas. Esto hace que las funciones disponibles para el diseño de reguladores predictivos hagan simplificaciones para llevar a cabo esta estrategia de una forma más eficaz. Por ello, es necesario implementar las matrices y realizar nuevas funciones de cálculo de las trayectorias con toda la información disponible.

El controlador predictivo forma parte del nodo supervisor, el cual enviará las trayectorias de las acciones de control y la salida postulada hacia el nodo local a través del canal de comunicaciones. Además, el nodo local debe informar al nodo supervisor de la acción de control aplicada al proceso además de proporcionar la lectura de la salida del proceso. Con esa información el nodo supervisor vuelve a componer las trayectorias a aplicar en la siguiente iteración.

En la Figura 133 se muestran los bloques principales de la herramienta y como interactúan entre ellos para llevar a cabo la simulación del sistema.

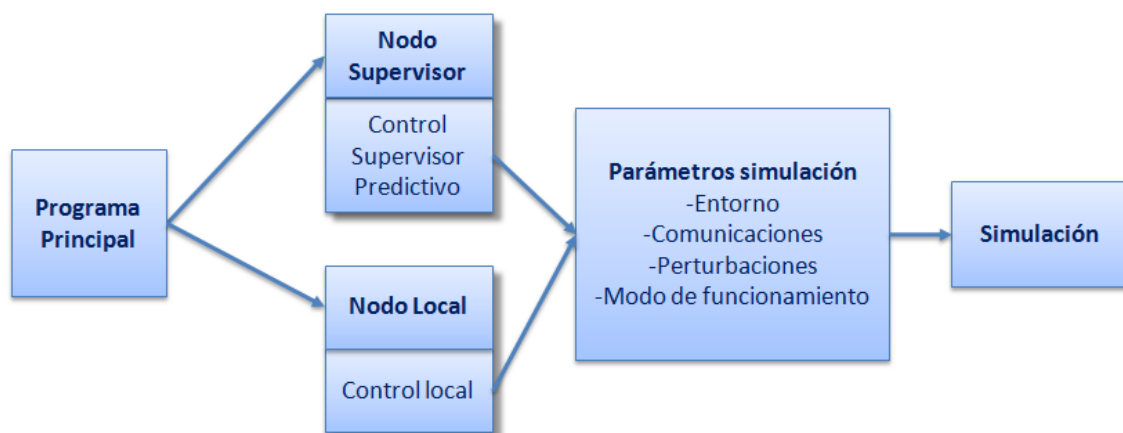


Figura 133. Bloques funcionales de la herramienta

A continuación se detallan las principales características de cada bloque y las opciones de configuración que permiten cada uno de ellos.

A.1.1 Programa principal

El programa principal es el encargado de recoger las características del sistema distribuido que se desea simular. Para ello la herramienta permite configurar diversos parámetros de los controladores. En la siguiente tabla se detallan las opciones disponibles en cada apartado:

Control Predictivo	<ul style="list-style-type: none"> • GPC • CRHPC
Parámetros del Control	<ul style="list-style-type: none"> • N • m (En caso de seleccionar un CRHPC) • Nu • K
Proceso (SISO)	<ul style="list-style-type: none"> • 1er orden • 2º orden • Fase no mínima • Retardo • Efecto integral • Introducción de un proceso
Entorno	<ul style="list-style-type: none"> • % Ancho de banda comunicaciones • Tiempo de cómputo
Modo funcionamiento	<ul style="list-style-type: none"> • Modo1. Envío de la referencia • Modo2. Envío de la acción de control • Modo3. Envío de las trayectorias postuladas sin compensación • Modo4. Envío de las trayectorias postuladas con compensación • Modo5. Envío de las trayectorias postuladas en los cambios de referencia con compensación
Tipos de fallos a simular	<ul style="list-style-type: none"> • Sin pérdida de datos • Pérdida de datos cada dos instantes • Pérdida de un dato en un instante • Pérdida de datos a partir de un instante • Pérdida aleatoria de datos (selección del porcentaje de pérdida)

Tabla 8. Opciones disponibles del sistema distribuido de control

Una vez seleccionadas las características, tanto del sistema distribuido de control como del entorno, es necesario calcular las matrices que permiten el cálculo del control predictivo.

A.1.2 Cálculo del control predictivo

A partir de los parámetros del control predictivo seleccionados anteriormente se obtienen las matrices necesarias para el cálculo de las trayectorias de la acción de control y de la salida que serán utilizadas por el nodo supervisor. En el anexo C se detalla la manera de obtener dichas trayectorias a partir de las matrices.

Como se ha indicado anteriormente, al no utilizar la estrategia del horizonte deslizante y necesitar la información de las acciones de control postuladas y de las salidas futuras, que actuará como una trayectoria nominal, se han tenido que crear las funciones de cálculo de las matrices del controlador predictivo que servirán para el cálculo de las mismas.

A.1.3 Simulación

Una vez seleccionadas las características del control distribuido y del entorno, y calculadas las matrices del controlador predictivo, el programa principal ejecuta el simulador, que se encarga de llevar a cabo la comprobación de las características del mismo y proporcionar las gráficas que permitirán evaluar el algoritmo propuesto.

Dentro de la simulación se encuentran el nodo local y el nodo supervisor. El código implementado en cada uno de estos nodos es similar al que se debería implementar en los nodos reales, y por lo tanto las pruebas llevadas a cabo con el simulador permiten comprobar el funcionamiento de los mismos ante las diversas situaciones que se pueden producir.

Al finalizar la simulación la aplicación permite visualizar la representación gráfica de la evolución de las señales más importantes: la referencia, la salida y la acción de control aplicada al proceso, así como los instantes en los que se han producido fallos en la recepción de los datos. Además, todos estos datos se pueden almacenar en un fichero para comparar cada una de las pruebas y obtener la combinación de parámetros que resulta más adecuada.

En el siguiente apartado se desarrolla la implementación del simulador, con las opciones disponibles así como las distintas estructuras realizadas en función del tipo de comunicación del sistema distribuido, que se ha realizado con ayuda de Simulink y con Truetime.

A.2 Modelo en simulación

La herramienta permite la posibilidad de simular el comportamiento de un sistema en el que los nodos dispongan de un canal exclusivo para las comunicaciones o de

uno en el que el canal de comunicaciones esté compartido. La distribución de los nodos, así como los algoritmos de control implementados en cada uno de ellos y los modos de funcionamiento son similares, sin embargo, en el caso de tener un canal de comunicaciones compartido la problemática de las pérdidas de datos y retrasos en el envío y la recepción de los mismos se acentúa. Por lo tanto, se hace necesario disponer de distintas estructuras que permitan evaluar la bondad del sistema distribuido de control en ambos casos.

A.2.1 Sistema distribuido con nodos fuertemente interconectados

En este caso los nodos del sistema distribuido utilizan un canal de comunicaciones exclusivo, es lo que ocurre cuando los nodos están implementados en un sistema único.

La Figura 134 muestra el modelo en simulación empleado, que servirá para el estudio de la afección que sobre la variable controlada ejercen las distintas situaciones previstas: pérdida de datos, desfase en las señales, etc. en función de la estructura aplicada.

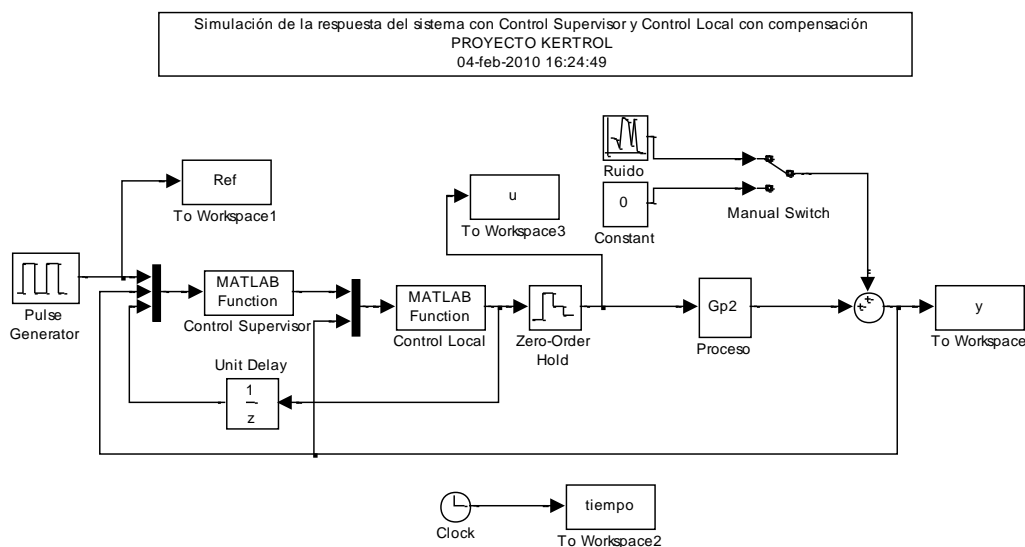


Figura 134. Modelo del sistema utilizado en simulación con nodos fuertemente interconectados

El modelo del sistema simulado se ha generado mediante bloques Simulink y funciones desarrolladas en Matlab. Los elementos que forman parte de este modelo son los siguientes:

- Proceso: Función de transferencia continua que modela el comportamiento dinámico del proceso controlado. Para el estudio de la robustez del sistema

frente a errores en el modelado esta función de transferencia es distinta a la utilizada para el diseño de los reguladores.

- Control Local: El control local es el encargado en última instancia de asegurar el envío de las acciones de control al proceso en los periodos de muestreo adecuados. Para ello este bloque utilizará las acciones de control propuestas por el control supervisor. En caso de pérdida de datos o tiempo excesivo de espera este control utilizará la información de los datos futuros de las acciones de control junto con las trayectorias para aplicar una acción de control utilizando la estrategia de compensación del error en la salida.
- Control Supervisor: Se trata del control supervisor que estará formado por un controlador predictivo que enviará al control local la trayectoria de las acciones y salidas futuras.
- Ruido: Para modelar el comportamiento de las variables de entrada que no se pueden manipular se incluye una señal de ruido configurable.
- Referencia: Para el estudio del comportamiento de la salida del sistema ante cambios en la consigna se incluye una señal de referencia cambiante con el tiempo.

Las funciones implementadas en el simulador, tanto en el nodo local como en el nodo supervisor, se han escrito de forma que utilicen la misma nomenclatura que el middleware del núcleo de control, y por lo tanto pueden ser prácticamente transcritas al lenguaje de programación de los nodos reales. Esto facilita el trabajo de implementación sobre el sistema real, a partir de las funciones probadas previamente con la herramienta de simulación del sistema distribuido de control.

A.2.2 Sistema distribuido con canal de comunicaciones compartido

Para estudiar la influencia que sobre el proceso tiene la distribución del control, el medio utilizado para las comunicaciones, el ancho de banda utilizado por las señales y el tiempo de cómputo dedicado a la realización del control, se ha utilizado, tal y como se ha comentado anteriormente, Truetime (Ohlin, Henriksson y Cervin 2007).

La Figura 135 muestra la estructura utilizada para el sistema distribuido de control cuando se dispone de un canal de comunicaciones compartido, en el que se utiliza Truetime para simular el canal de comunicaciones compartido así como cada uno de los nodos del sistema distribuido.

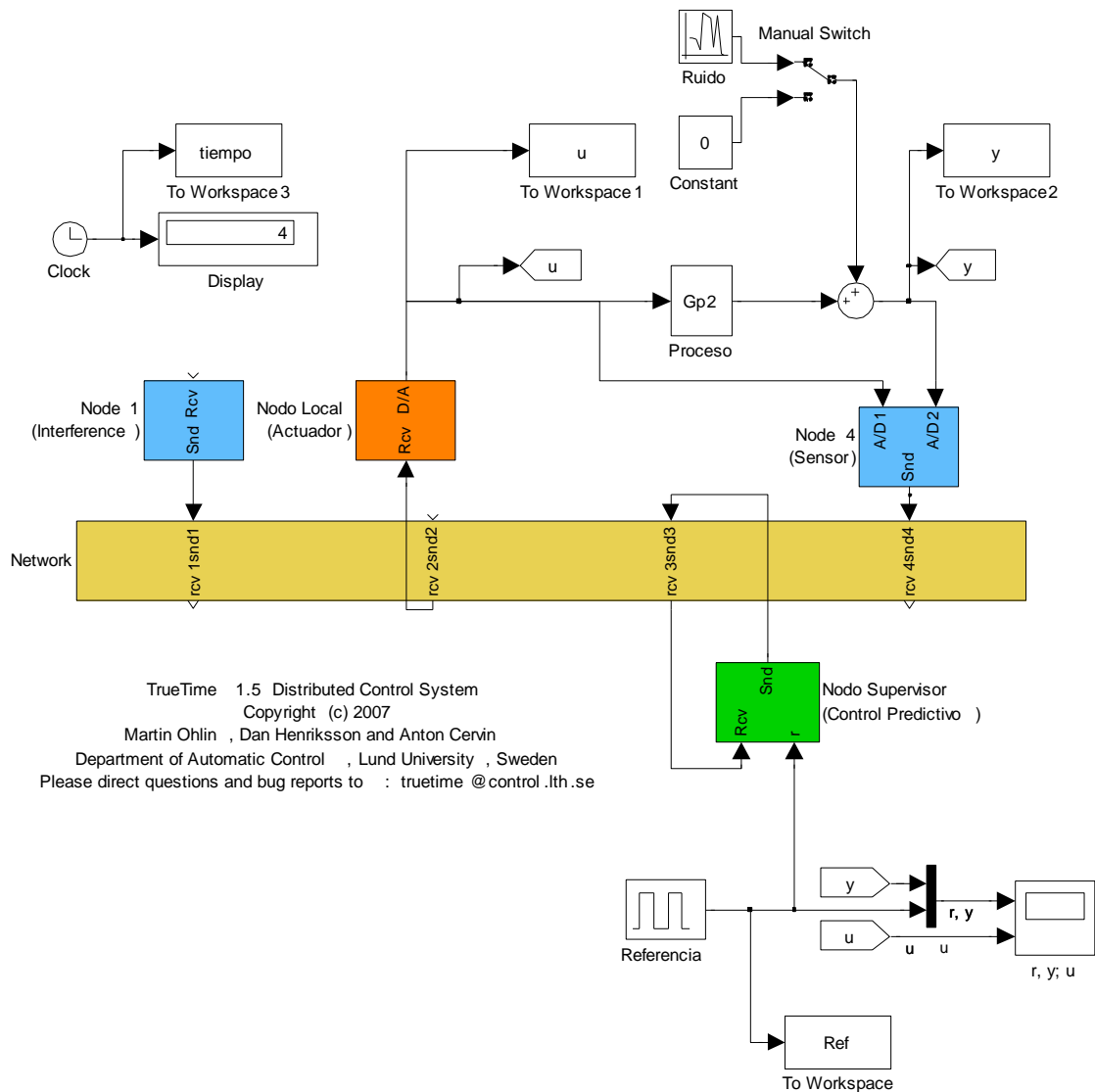


Figura 135. Modelo del sistema utilizado en simulación con canal de comunicaciones compartido

En este caso además de los nodos supervisor y local, el proceso y el ruido existen los siguientes elementos:

- **Canal de comunicaciones:** Este elemento es el encargado de simular las características propias de un sistema de comunicaciones compartido. Las características de las comunicaciones son completamente configurables, pudiendo definir el tipo de red (CAN, Ethernet, Round Robin, FDMA, TDMA, Switched Ethernet), el número de nodos, velocidad de los datos (bits/s), probabilidad de pérdidas,...
- **Interferencia:** Este elemento simula la utilización del canal de comunicaciones por otra tarea que no es la de control, definida mediante el consumo del ancho de banda en las comunicaciones que utiliza.

- Convertidores A/D y D/A: El nodo local es el que se encuentra más cerca del proceso a controlar, y por lo tanto es el que realiza las tareas de muestreo de la señal de salida y envío de la acción de control. Los convertidores son los encargados de realizar esta tarea, y mediante Truetime se pueden simular.

A.3 Limitaciones de la herramienta de simulación

La herramienta de simulación es muy dependiente de la estructura del sistema distribuido de control. Tal y como se ha mostrado en la Figura 134 y Figura 135 la estructura utilizada es válida para sistemas SISO y con un sistema distribuido formado por un nodo supervisor y un nodo local.

Para cualquier otro tipo de estructuras es necesario modificar los modelos creados con Simulink/Truetime, sin embargo, el cálculo y diseño de los reguladores se puede utilizar la misma herramienta con pequeñas modificaciones en el código.

Por otra parte las comunicaciones compartidas se limitan a las proporcionadas por Truetime, sin embargo es fácil cambiar los parámetros de dichas comunicaciones accediendo al elemento del canal de comunicaciones.

Anexo B. Pruebas realizadas con el simulador

A continuación se detallan la elección de parámetros así como las pruebas realizadas para los procesos de primer orden, de fase no mínima, con retardo múltiplo del periodo y con efecto integral. Las pruebas realizadas se analizan en el apartado 5.2 en el que se muestra la validez de la herramienta de simulación así como de la metodología para la elección de los parámetros del sistema distribuido de control.

B.1 Proceso de primer orden

Para este caso se ha tomado como proceso la salida de velocidad de un motor de corriente continua de la marca Feedback, cuyo modelo en forma de función de transferencia se ha identificado como:

$$G_p(s) = \frac{\omega(\text{rpm})}{U(\text{voltios})} = \frac{9.985}{1 + 0.165 \cdot s}$$

Además, la relación entre la velocidad en rpm y los voltios que suministra el tacómetro es:

$$\text{Velocidad (rpm)} = 11.9248 \times \text{Tensión_tacómetro (voltios)}$$

B.1.1 Elección de los parámetros del sistema distribuido de control

Se propone estudiar el comportamiento de un sistema distribuido de control formado por un nodo supervisor, en el que se ejecuta un control predictivo, y un nodo local conectado directamente al proceso de primer orden que se desea controlar. Inicialmente se deben obtener los parámetros del sistema distribuido de control, T_s , N , m , N_u y K , utilizando el método mostrado en 5.2. En estas pruebas se han dejado en los siguientes valores fijos del regulador predictivo: $\alpha=1$, $\lambda=0.1$ y $T(z^{-1})=1$. En algunos casos será necesario cambiar dichos parámetros para que el controlador predictivo funcione mejor, aunque el ajuste es empírico y por lo tanto no es sencillo.

Teniendo en cuenta la dinámica del sistema en bucle cerrado se ha elegido un periodo de 10 milisegundos.

Regulador predictivo GPC

A continuación se muestra la simulación del sistema con el regulador GPC para un periodo de 10 ms:

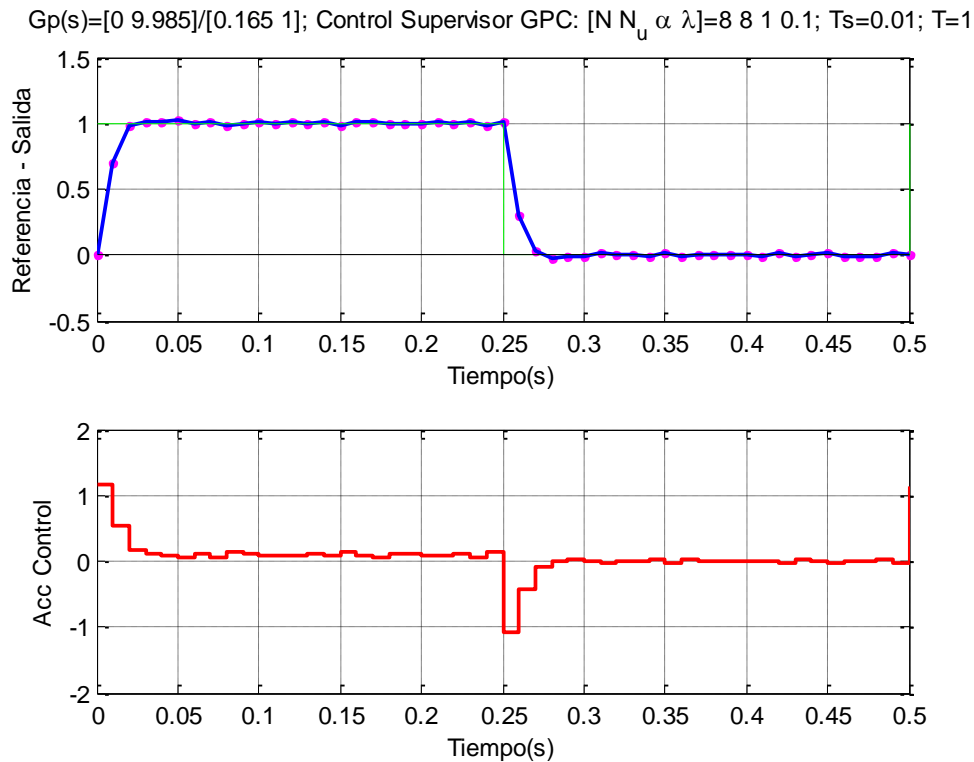


Figura 136. Proceso de primer orden con el controlador GPC

En este caso, con el periodo de muestreo seleccionado, un horizonte de predicción a partir de 6 ó 7 instantes es lo más adecuado para poder englobar a toda la respuesta dinámica.

El siguiente paso es estudiar la mejor combinación entre el horizonte de predicción (N) y el horizonte de control (N_u) que consiga un índice de error más pequeño cuando se utiliza la acción de control postulada. En la siguiente figura se muestra el estudio realizado para el proceso de primer orden:

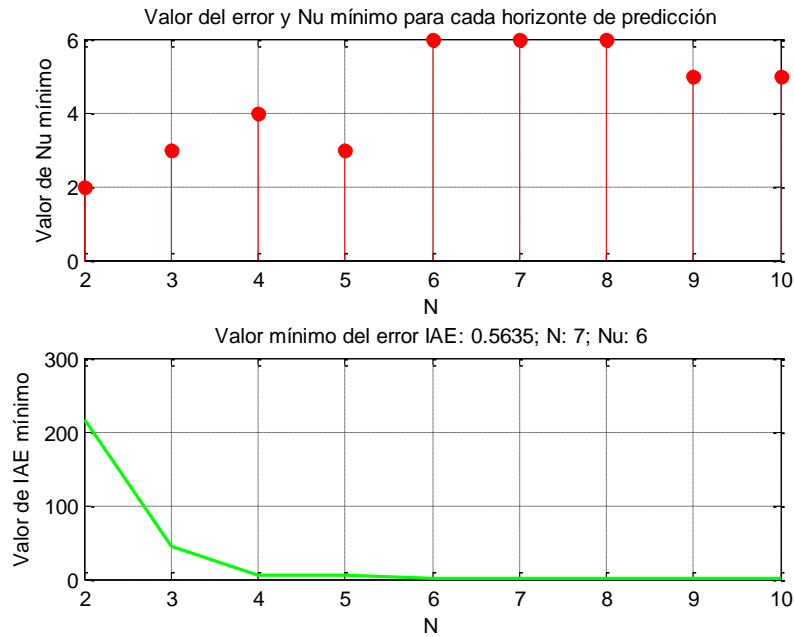


Figura 137. Valor de Nu en función de N con el que se obtiene el mínimo error para el proceso de primer orden

La figura anterior muestra el valor de IAE mínimo para cada N y el horizonte de control con el que se ha obtenido. La combinación que consigue el mínimo error es $N=7$ para un $Nu=6$. La diferencia del error en la salida entre tomar $N=7$ ó $N=8$ es muy pequeña, y puesto que es mejor ampliar el horizonte de predicción se ha optado por tomar la $N=8$. La Figura 138 muestra la variación del error al cambiar Nu para $N=8$. Tal y como se mostraba en la gráfica anterior el error mínimo se consigue para $Nu=6$.

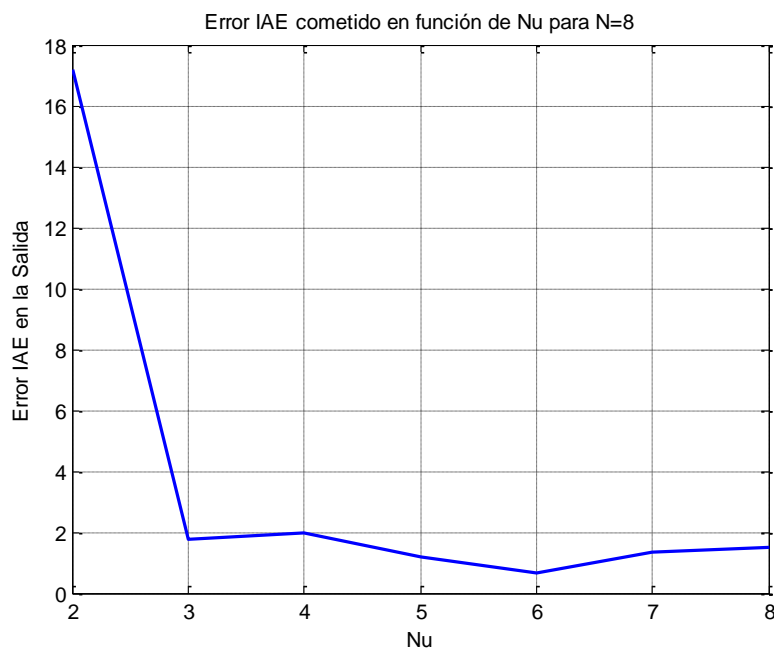


Figura 138. Errores cometidos en función de Nu para $N=8$

A continuación se muestra el estudio de los valores de K que consiguen el mínimo error cuando se utiliza la compensación de la acción de control ante una pérdida masiva de datos a partir del instante 2:

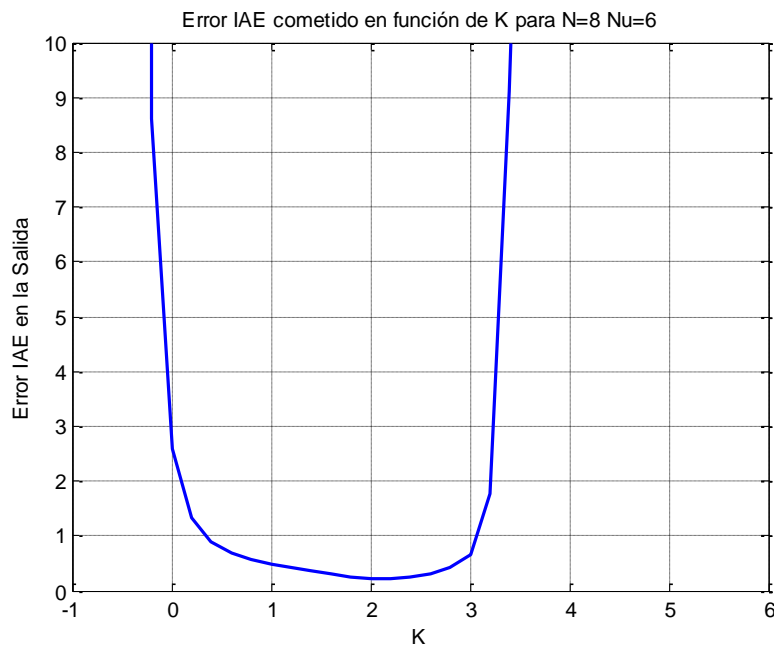


Figura 139. Errores cometidos en función de K para $N=8$ y $Nu=6$

La herramienta de simulación calcula analíticamente el valor de K que produce el mínimo índice de error. En este caso se consigue un IAE mínimo de 0.2246 para el valor de $K=2.2$, aunque como se muestra en la figura se puede tomar un valor de K entre 0.5 y 3.0, ya que el error en este rango está cercano al IAE mínimo.

Regulador predictivo CRHPC

Siguiendo la metodología descrita en 5.2.1 para el caso del regulador CRHPC, a continuación se muestra la simulación del sistema con el periodo de 10 ms y el regulador CRHPC:

$G_p(s)=[0 \ 9.985]/[0.165 \ 1]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=6 \ 2 \ 8 \ 1 \ 0.1$; $T_s=0.01$; $K=0$

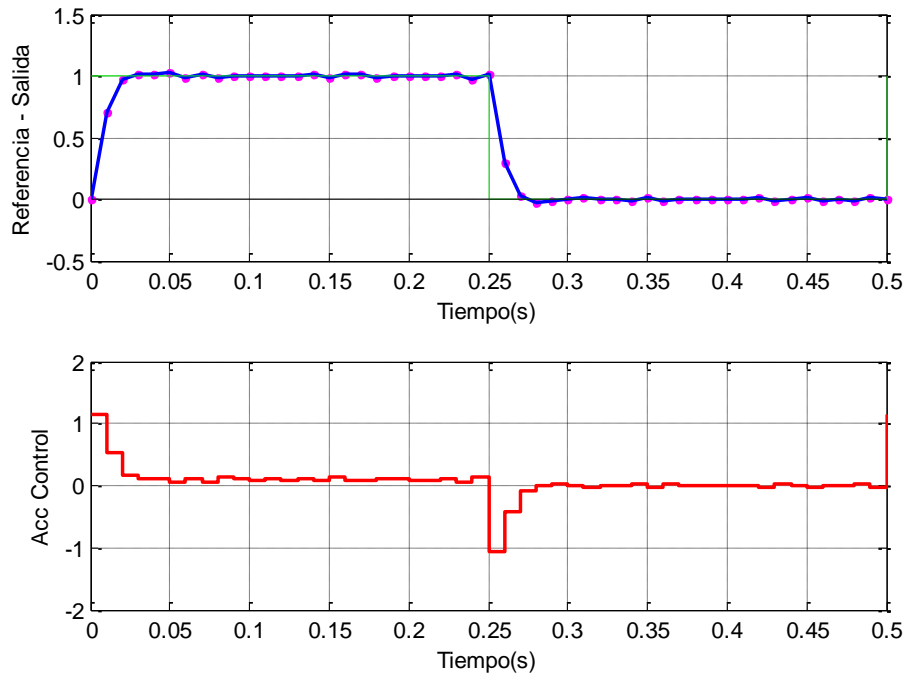


Figura 140. Proceso de primer orden con el controlador CRHPC

En este caso se ha tomado un horizonte de predicción de 8 instantes ($N+m$), en el que los dos últimos pertenecen a la restricción terminal ($m=2$). A continuación se simula el comportamiento para distintas combinaciones del horizonte de control y de predicción que permita obtener el menor error.

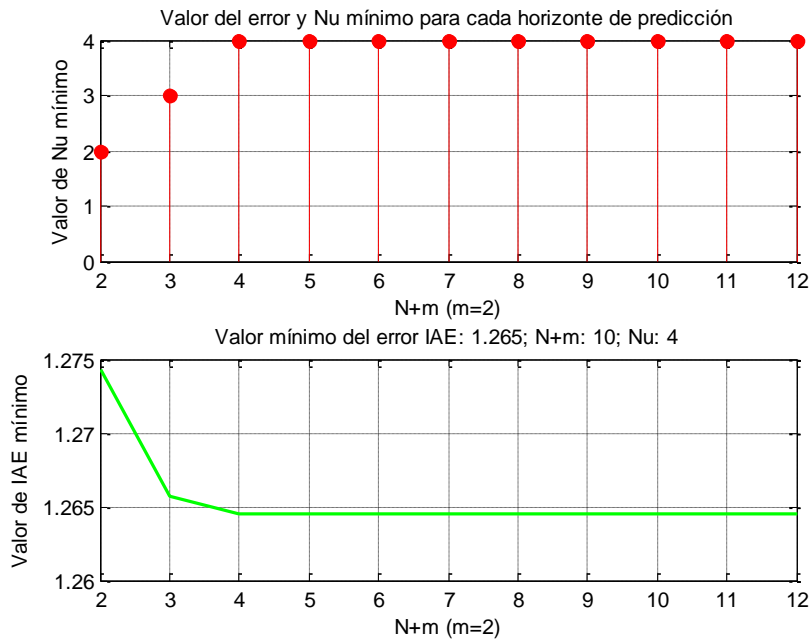


Figura 141. Estudio del valor de N_u en función de $N+m$ con el que se obtiene el mínimo error

Se ha elegido un horizonte de predicción de $N+m=8$ instantes, con $m=2$, para englobar a toda la respuesta dinámica del proceso. En este caso el horizonte de control que consigue el menor error es $Nu=4$. A continuación se muestra cómo afecta la elección del horizonte de control en el índice de error.

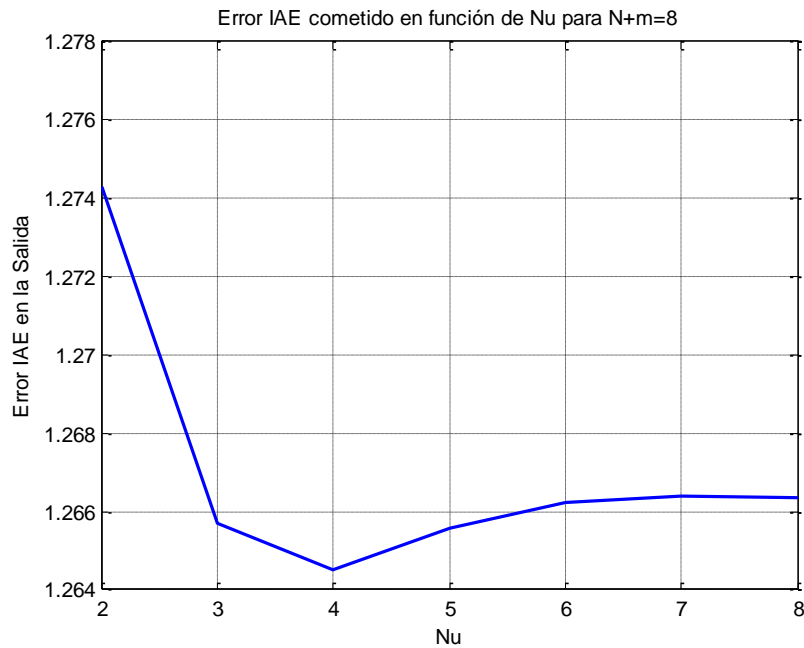


Figura 142. Errores cometidos en función de Nu para $N+m=8$ instantes, con $m=2$

Finalmente se simula el sistema con los parámetros anteriores ante una pérdida masiva de datos a partir del instante 2 para obtener el valor de la ganancia de la compensación de la acción de control.

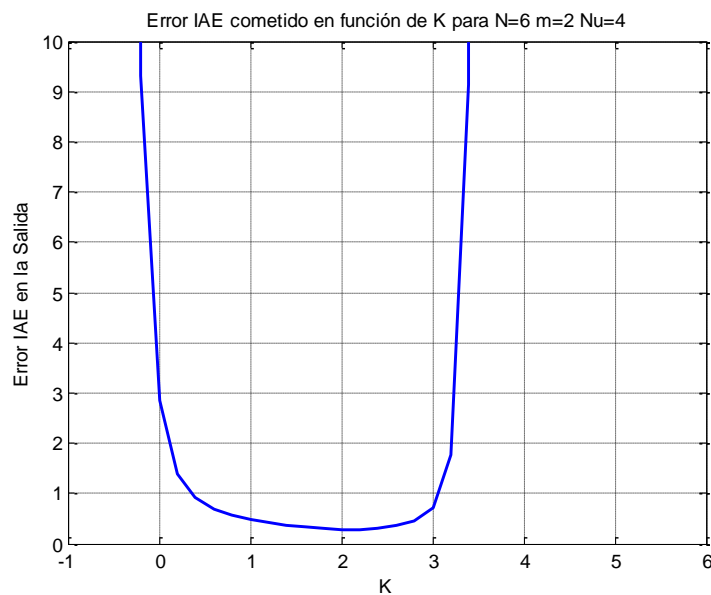


Figura 143. Errores cometidos en función de K para $N=6$, $m=2$ y $Nu=4$

Con $K=2.2$ se obtiene el mínimo valor del índice del error $IAE=0.2798$, aunque se puede tomar un valor de K entre 0.5 y 3.0, ya que el error en este rango está cercano al IAE mínimo.

B.1.2 Comportamiento del sistema distribuido de control

A continuación se muestra el comportamiento del sistema distribuido de control mediante dos casos, en función del tipo de comunicaciones entre el nodo local y el nodo supervisor, para los modos de funcionamiento 2 al 5.

Caso 1. Sistema distribuido con canal de comunicaciones exclusivo

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} . Por cada porcentaje de pérdida de datos se realizan 10 simulaciones distribuidas de forma aleatoria a lo largo de todo el tiempo de la prueba. Esto supone la realización de 190 simulaciones por cada modo de funcionamiento.

A continuación se muestran los resultados de los distintos índices de error para los distintos modos de funcionamiento (del 2 al 5):

GPC: $N=8$, $Nu=6$, $K=2.2$

En la siguiente figura se comparan los errores IAE en la salida para cada modo de funcionamiento:

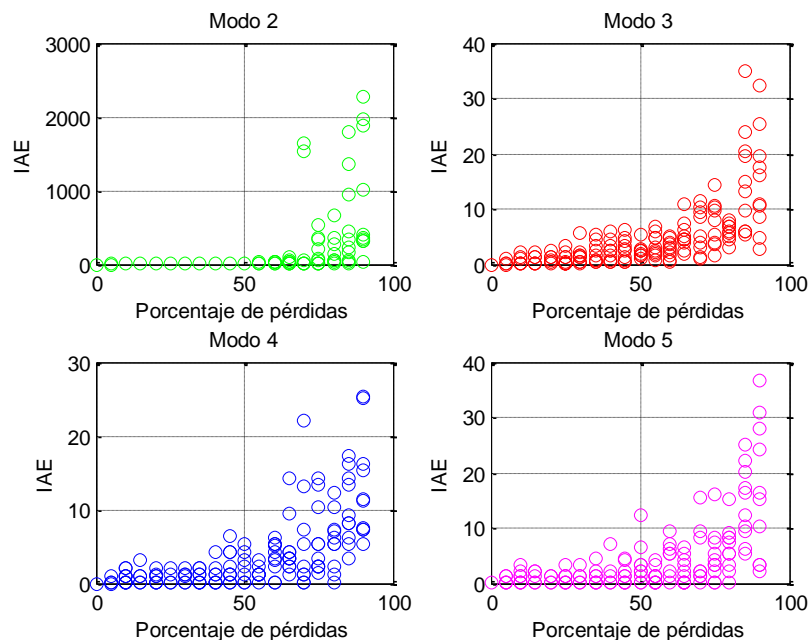


Figura 144. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

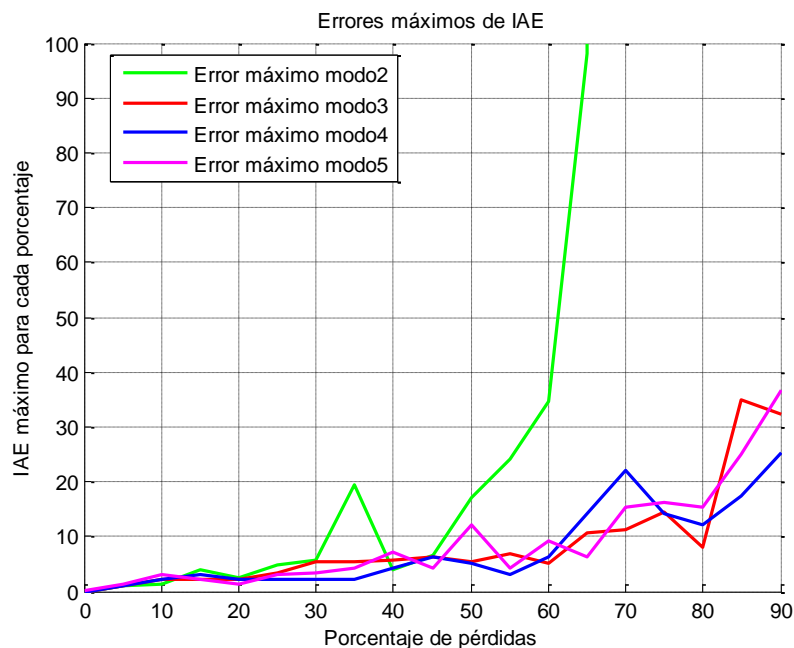


Figura 145. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

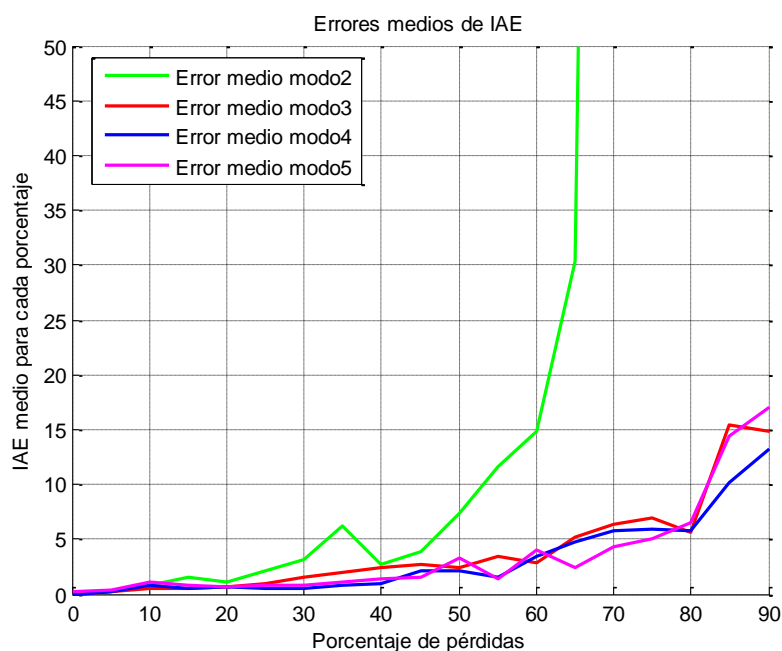


Figura 146. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

En la figura anterior se muestra el índice del error absoluto para cada uno de los porcentajes de pérdidas en el caso más desfavorable. Los modos de funcionamiento que utilizan las trayectorias con la compensación de la acción de control son los que consiguen el error máximo más pequeño para todos los porcentajes de pérdidas.

CRHPC: $N=6$, $m=2$, $Nu=4$, $K=2.2$

Comparación de los errores IAE en cada modo de funcionamiento:

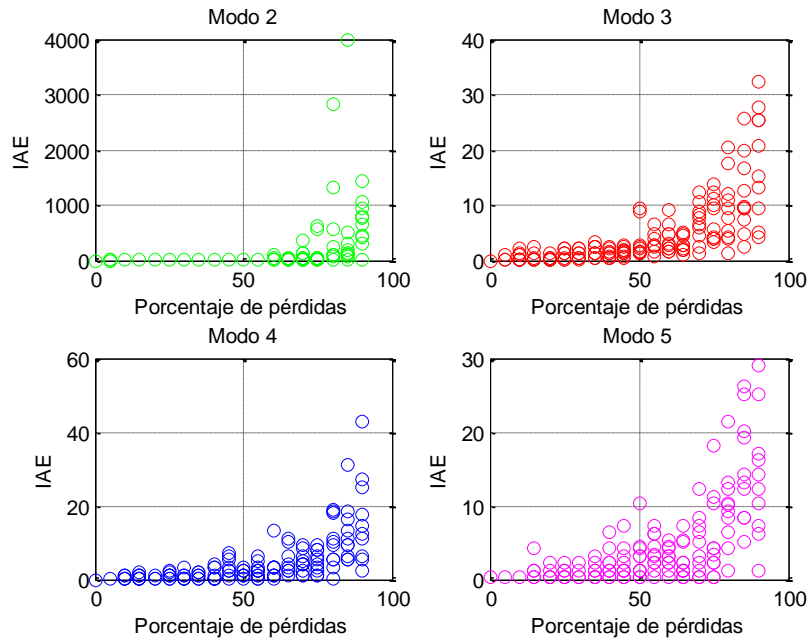


Figura 147. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

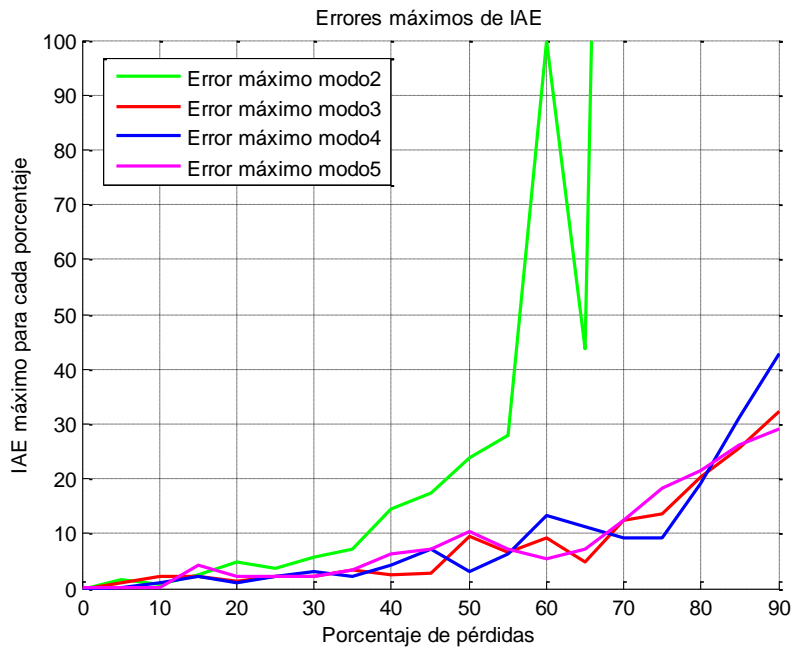


Figura 148. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

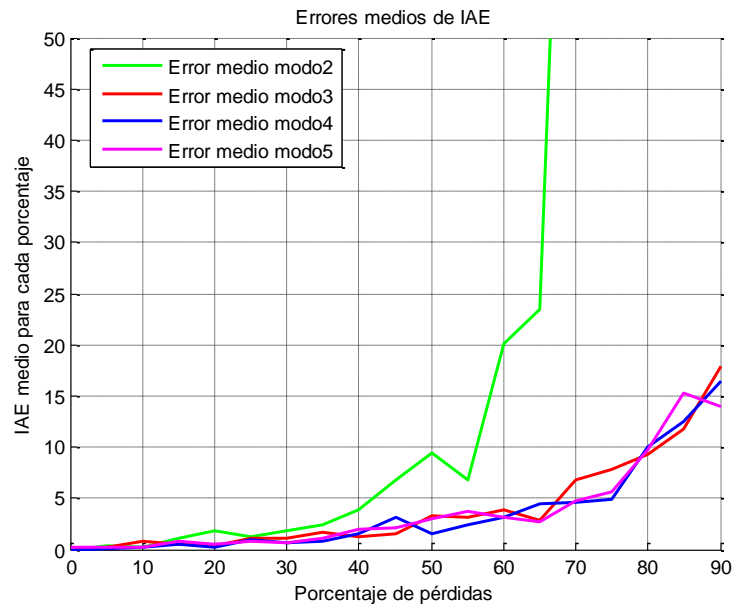


Figura 149. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

Caso 2. Sistema distribuido con canal de comunicaciones compartido

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} , es decir, la misma que en la prueba anterior. Además se ha introducido una interferencia en la red que consume un 10% del ancho de banda y en este caso se tiene en cuenta el coste computacional del regulador predictivo de 100 μ s.

GPC: $N=8$, $Nu=6$, $K=2.2$

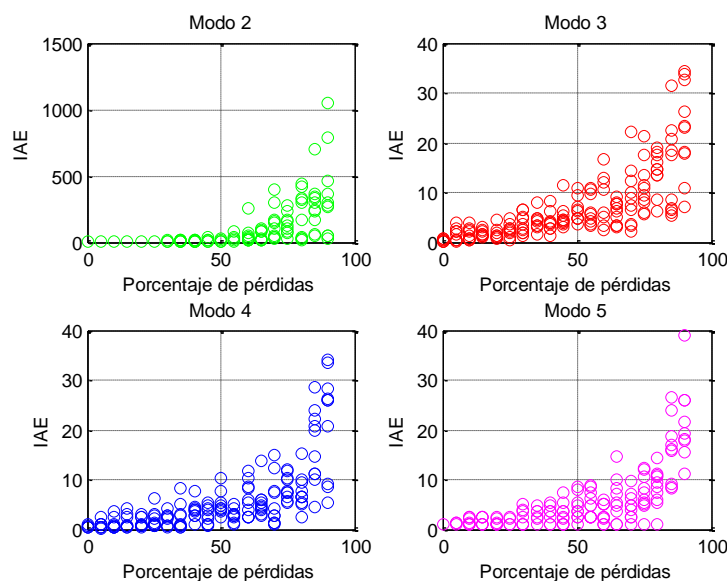


Figura 150. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

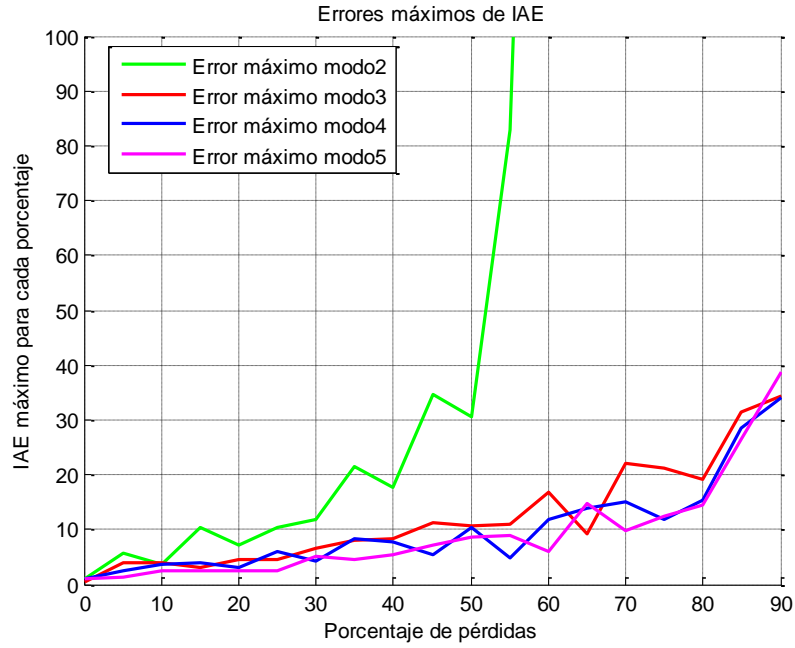


Figura 151. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

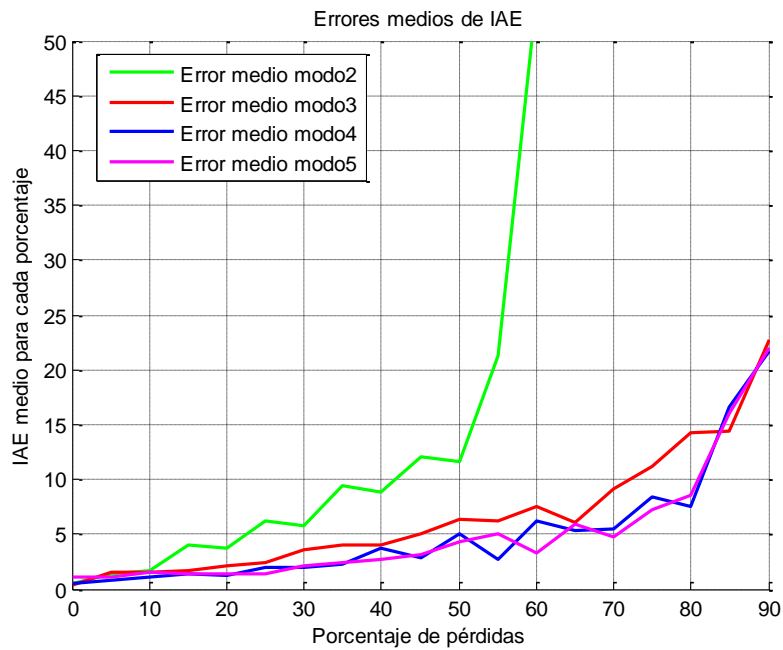


Figura 152. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

CRHPC: $N=6$, $m=2$, $N_u=4$, $K=2.2$

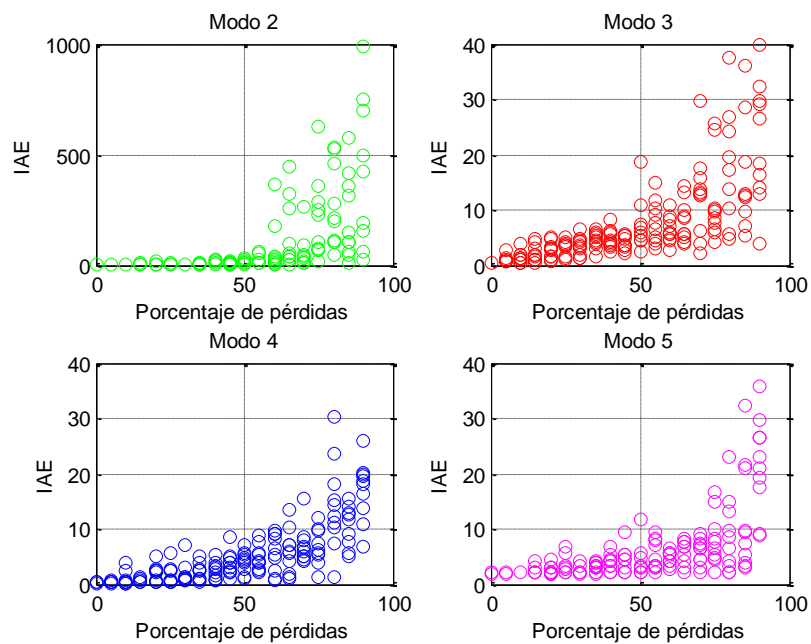


Figura 153. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

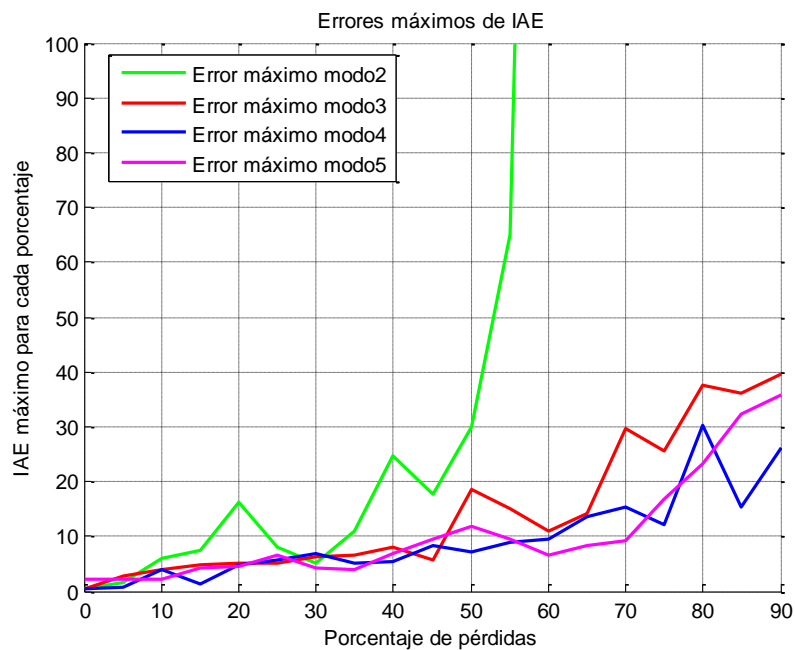


Figura 154. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

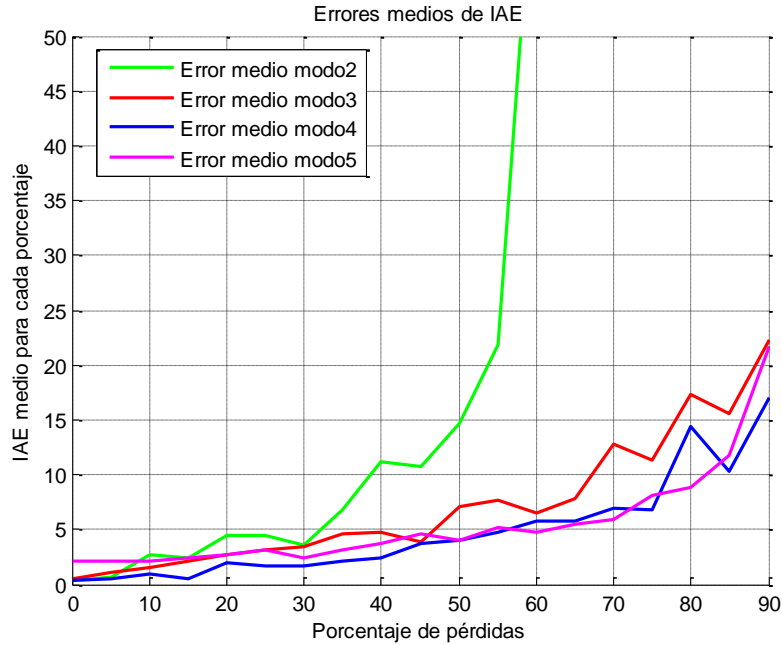


Figura 155. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

B.2 Proceso de fase no mínima

Proceso con el que se han realizado pruebas:

$$G_p(s) = \frac{U_s}{U_e} = \frac{8 \cdot s - 32}{s^2 + 7 \cdot s + 10} = \frac{8 \cdot (s - 4)}{(s + 5) \cdot (s + 2)}$$

B.2.1 Elección de los parámetros del sistema distribuido de control

A continuación se aplica el mismo método mostrado en las pruebas anteriores para el sistema distribuido de control formado por un nodo supervisor, en el que se ejecuta un control predictivo, y un nodo local conectado directamente al proceso que se desea controlar. Se deben obtener los parámetros del sistema distribuido de control: T_s , N , m , N_u y K , dejando los siguientes valores fijos: $\alpha=1$, $\lambda=0.1$ y $T(z^{-1})=1$.

Teniendo en cuenta la dinámica del sistema en bucle cerrado el periodo elegido es de 80 milisegundos.

Regulador predictivo GPC

A continuación se muestra la simulación del sistema con el regulador GPC para un periodo de 80 ms:

$G_p(s)=[8 -32]/[1 7 10]$; Control Supervisor GPC: $[N N_u \alpha \lambda]=16 16 1 0.1$; $T_s=0.08$; $K=0$

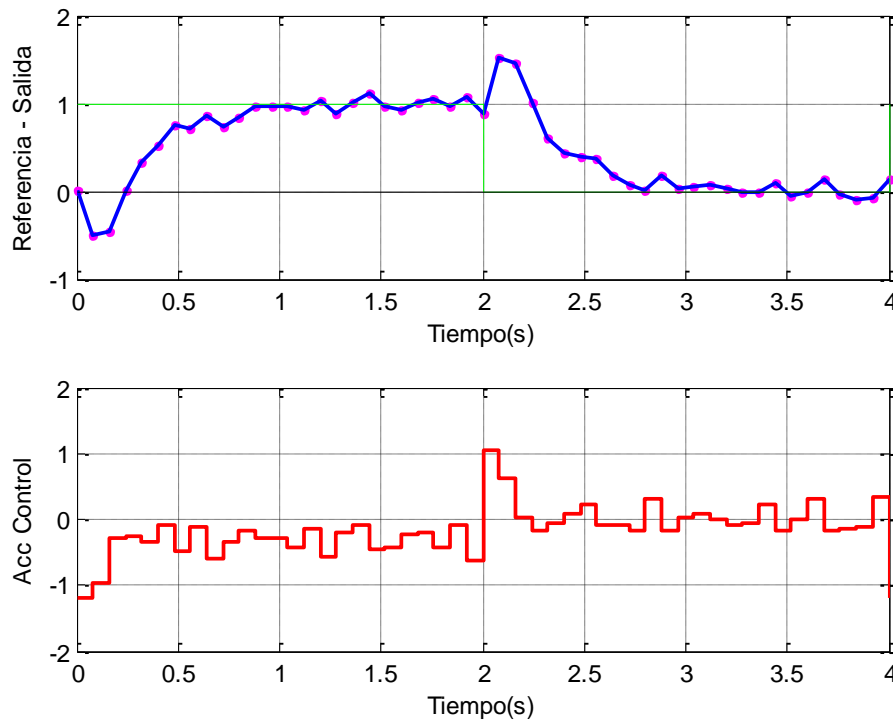


Figura 156. Proceso de fase no mínima con el controlador GPC

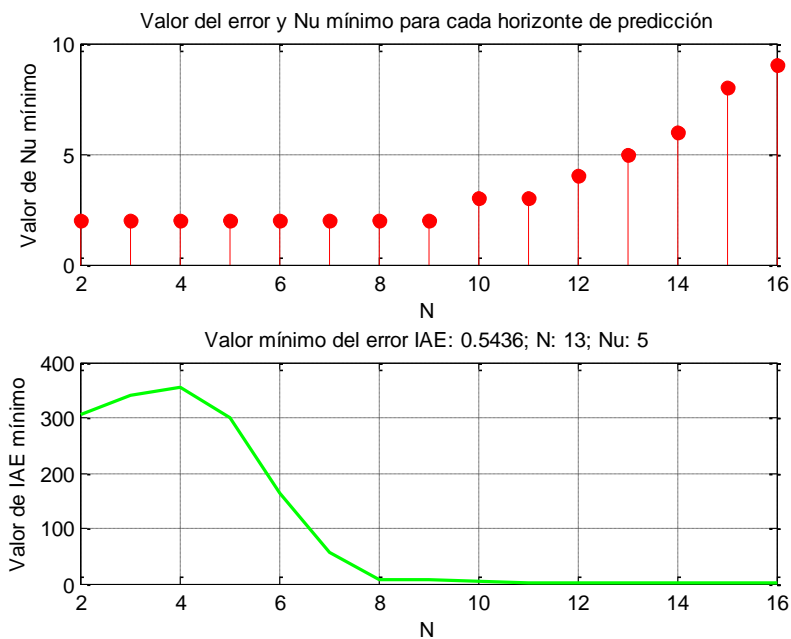


Figura 157. Estudio del valor de Nu en función de N con el que se obtiene el mínimo error

Se elige un horizonte de predicción de $N=14$ instantes que permite englobar a toda la dinámica del sistema. En este caso el valor mínimo para ese horizonte de predicción se consigue con $N_u=6$.

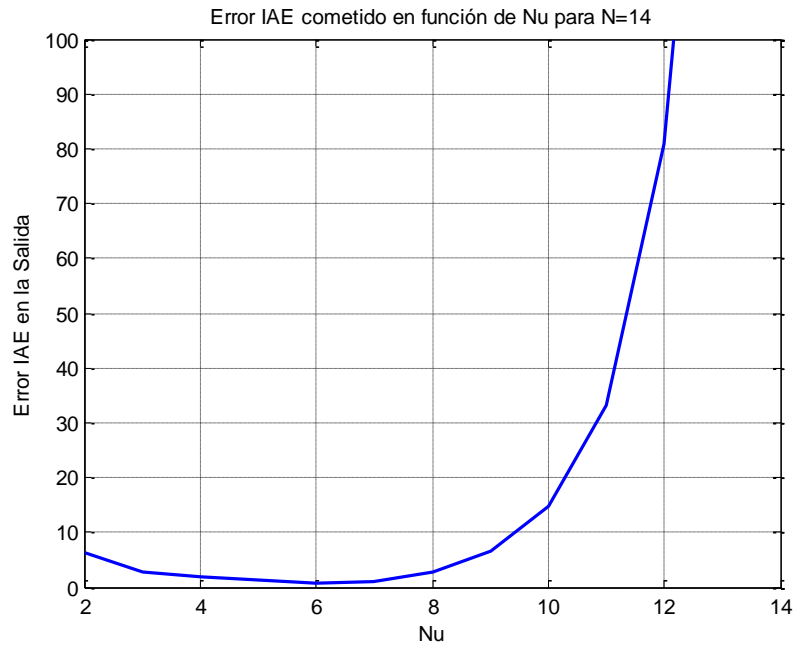


Figura 158. Errores cometidos en función de Nu para N=14

La simulación del sistema, con los parámetros del control elegidos anteriormente, ante una pérdida masiva de datos a partir de instante 2, se muestra en la siguiente figura, con la que se obtiene el valor de la ganancia de la compensación de la acción de control.

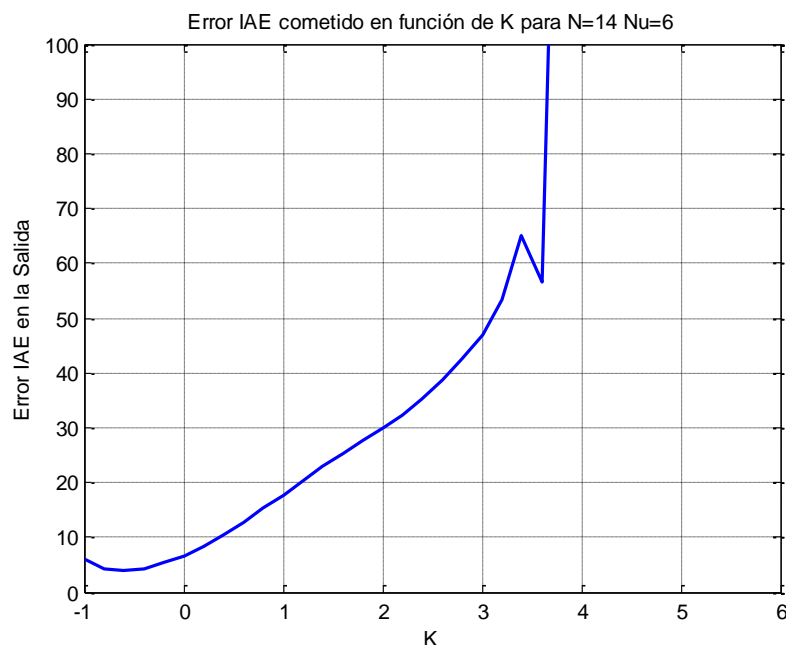


Figura 159. Errores cometidos en función de K para N=14 y Nu=6

El valor mínimo de IAE=3.9 se obtiene para una K=-0.6, es decir, con ganancia negativa, consecuencia de tener un proceso de fase no mínima.

En este caso el valor obtenido de la ganancia al ser negativo funciona porque el número de muestras es finito. En caso contrario se podría tener un sistema inestable, de todas maneras con una K entre -1 y 0.5 el error cometido está muy cerca del mínimo.

Regulador predictivo CRHPC

Siguiendo la metodología anterior:

$G_p(s)=[8 \ -32]/[1 \ 7 \ 10]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=12 \ 4 \ 16 \ 1 \ 0.1$; $T_s=0.08$; $K=0$

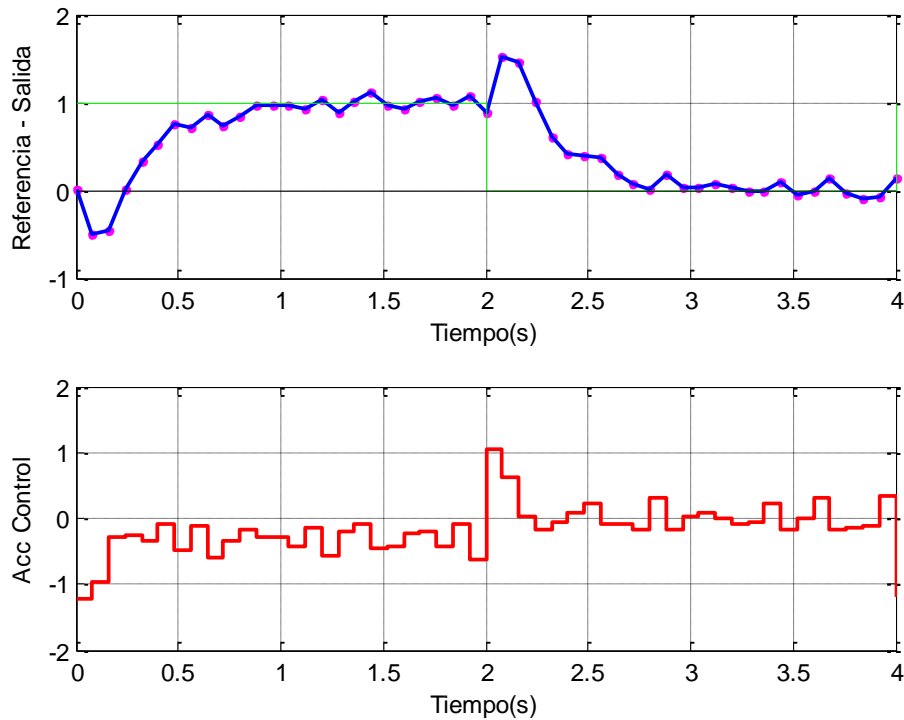


Figura 160. Proceso de fase no mínima con el controlador CRHPC

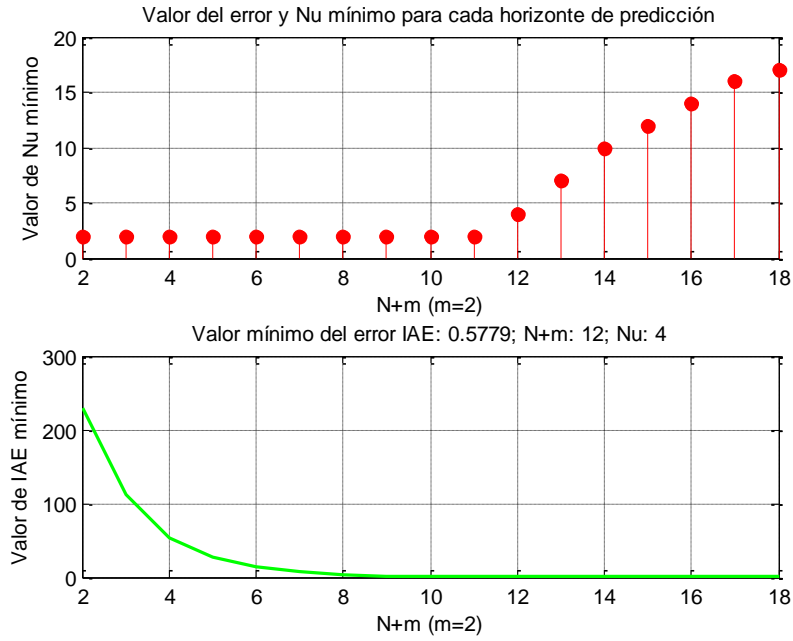


Figura 161. Estudio del valor de Nu en función de N+m con el que se obtiene el mínimo error

Se elige un horizonte de predicción de 14 instantes (N+m), en el que los dos últimos pertenecen a la restricción terminal (m=2), para englobar a toda la respuesta dinámica del proceso. Para este horizonte de predicción el valor de Nu que consigue el mínimo error es de 10 instantes.

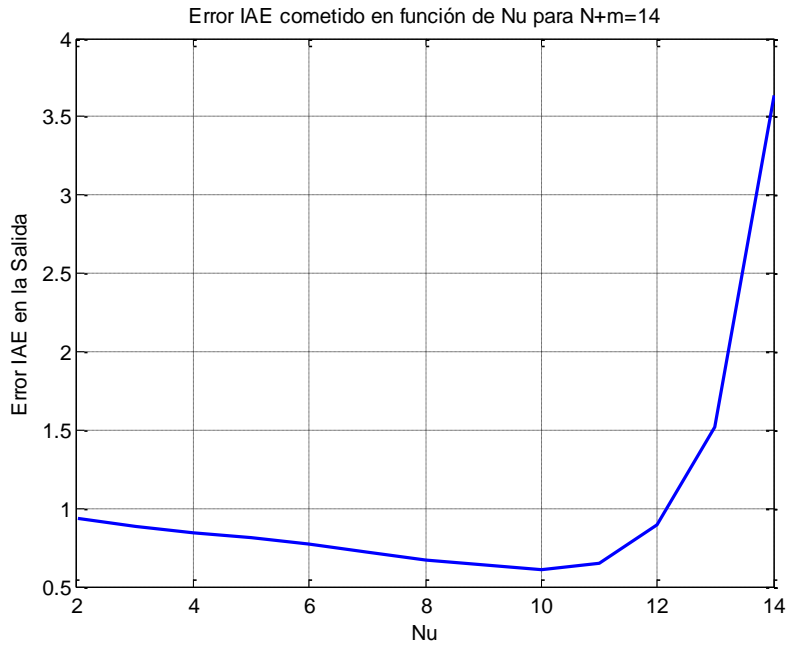


Figura 162. Errores cometidos en función de Nu para N+m=14, con m=2

Finalmente se simula el sistema con los parámetros anteriores ante una pérdida masiva de datos a partir del instante 2 para obtener el valor de la ganancia de la compensación de la acción de control.

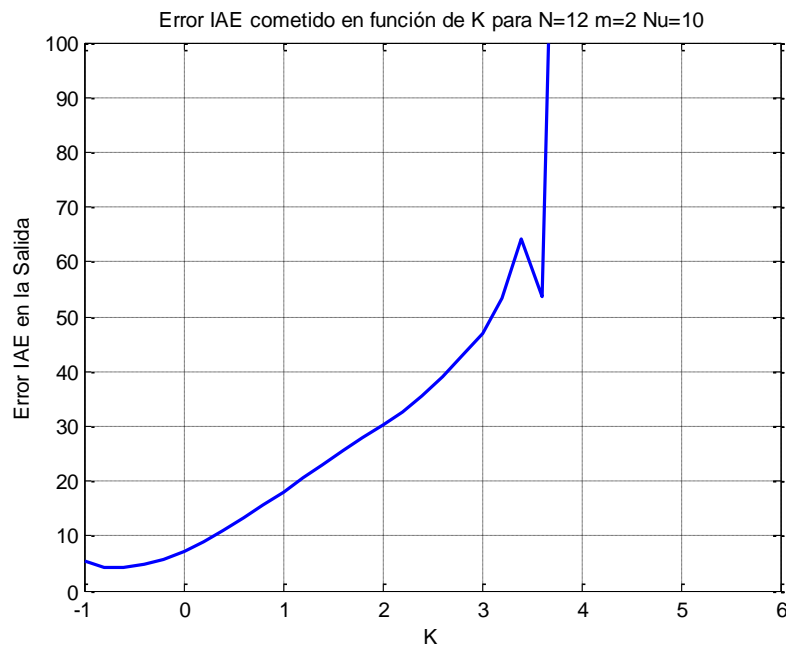


Figura 163. Errores cometidos en función de K para $N=12$, $m=2$ y $Nu=10$

Con $K=-0.6$ se obtiene el mínimo valor de $IAE=4.128$.

Al igual que ocurría anteriormente, el valor obtenido de la ganancia es negativo, y funciona porque el número de muestras es finito. En caso contrario se podría tener un sistema inestable. Valores de K entre -1 y 0.5 siguen teniendo un error muy cercano al mínimo.

B.2.2 Comportamiento del sistema distribuido de control

A continuación se muestra el comportamiento del sistema distribuido de control mediante dos casos, en función del tipo de comunicaciones entre el nodo local y el nodo supervisor, utilizando los parámetros obtenidos en el apartado anterior.

Caso 1. Sistema distribuido con canal de comunicaciones exclusivo

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} , realizando, por cada porcentaje de pérdida de datos, 10 simulaciones distribuidas de forma aleatoria a lo largo de todo el tiempo de la prueba. Esto supone la realización de 190 simulaciones por cada modo de funcionamiento.

A continuación se muestran los resultados de los distintos índices de error para los distintos modos de funcionamiento:

GPC: N=14, Nu=6, K=-0.6

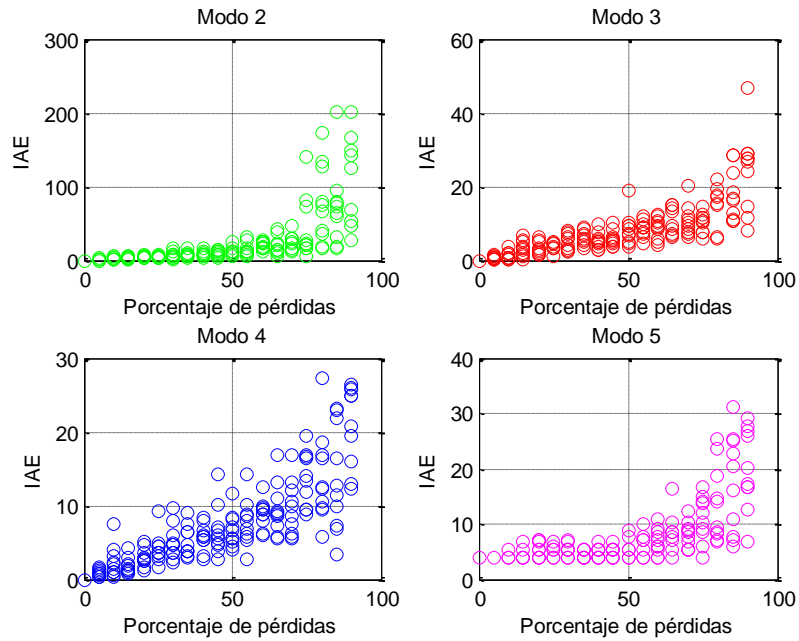


Figura 164. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

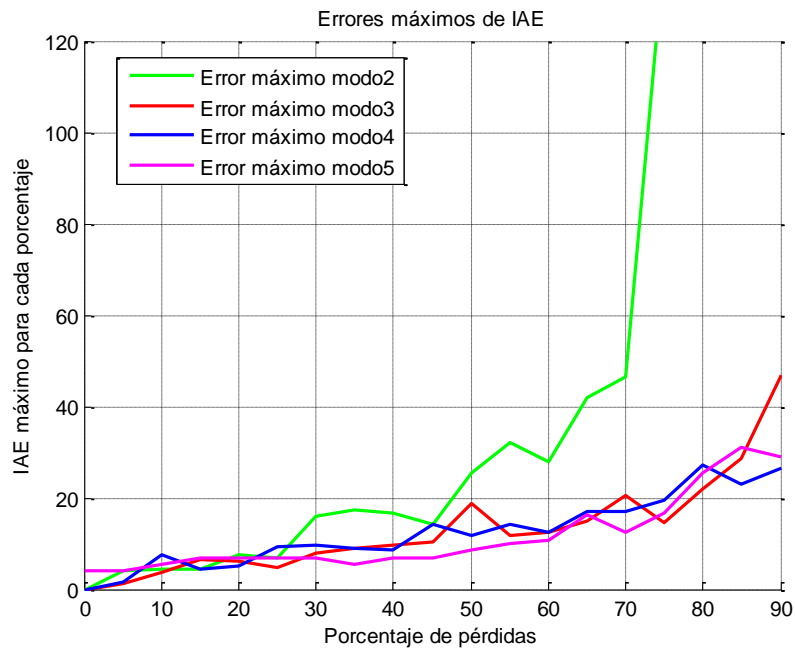


Figura 165. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

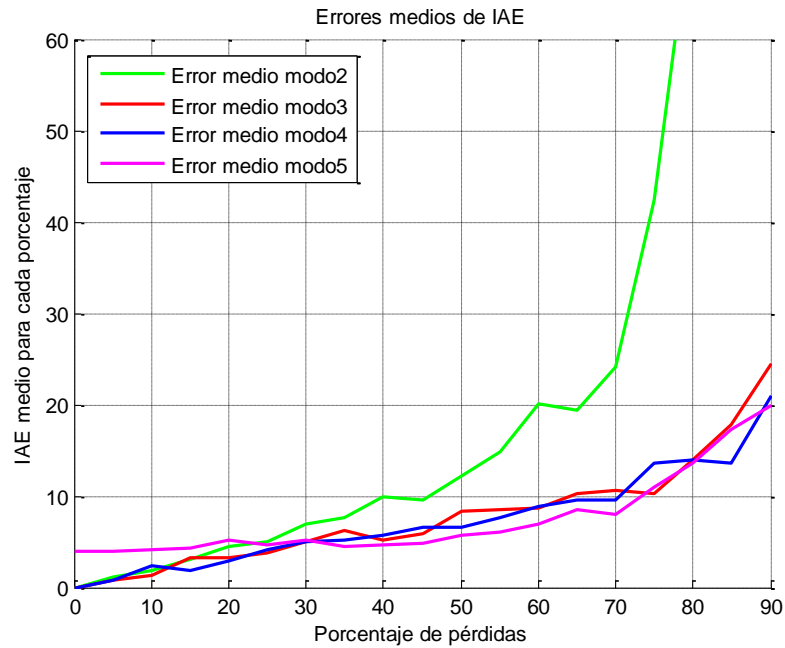


Figura 166. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

CRHPC: $N=12$, $m=2$, $Nu=10$, $K=-0.6$

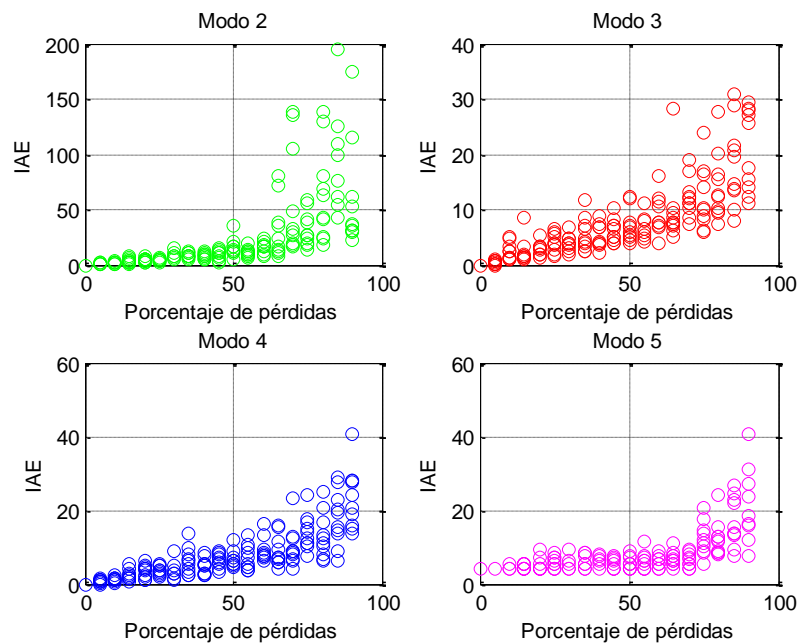


Figura 167. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

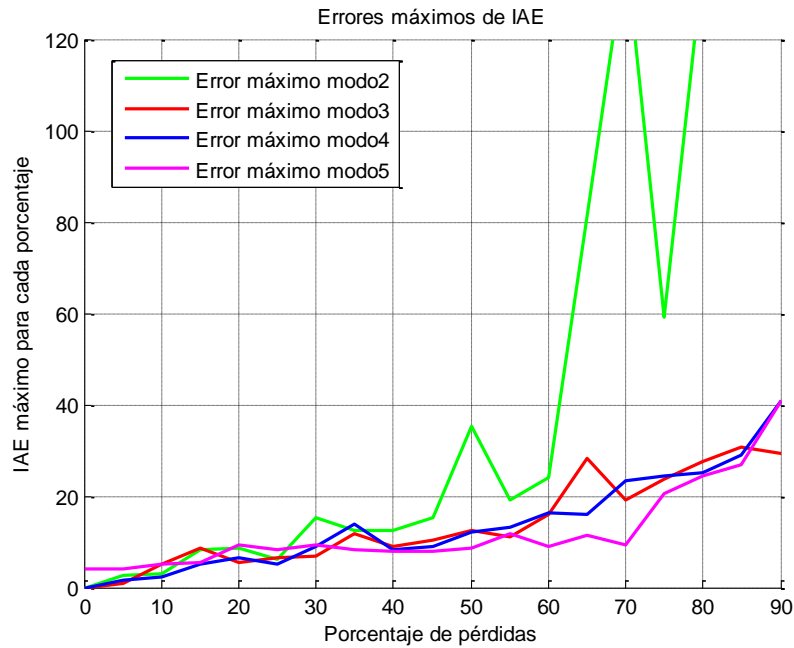


Figura 168. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

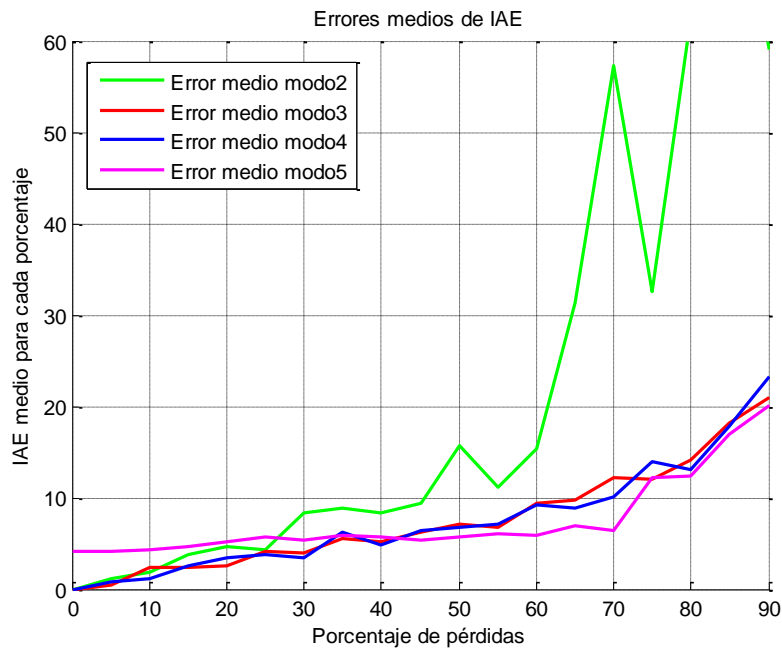


Figura 169. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

Caso 2. Sistema distribuido con canal de comunicaciones compartido

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} , es decir, la misma que en la prueba anterior. Además se ha introducido una

interferencia en la red que consume un 10% del ancho de banda y en este caso se tiene en cuenta el coste computacional de regulador predictivo 800 μ s.

GPC: N=14, Nu=6, K=-0.6

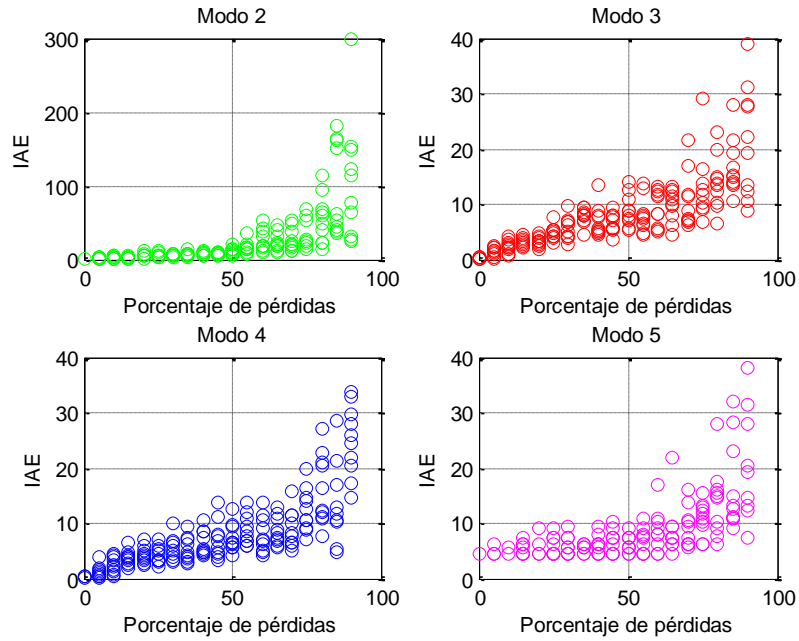


Figura 170. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

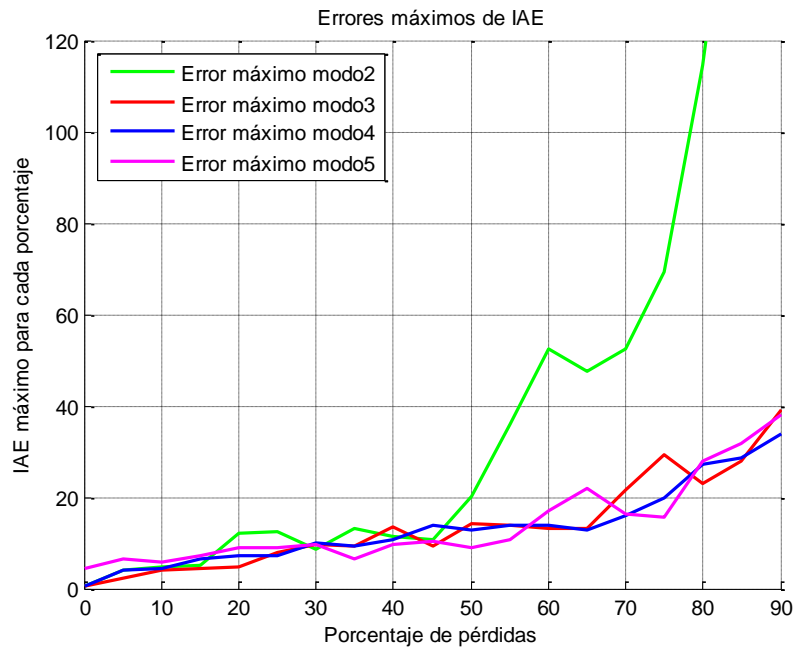


Figura 171. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

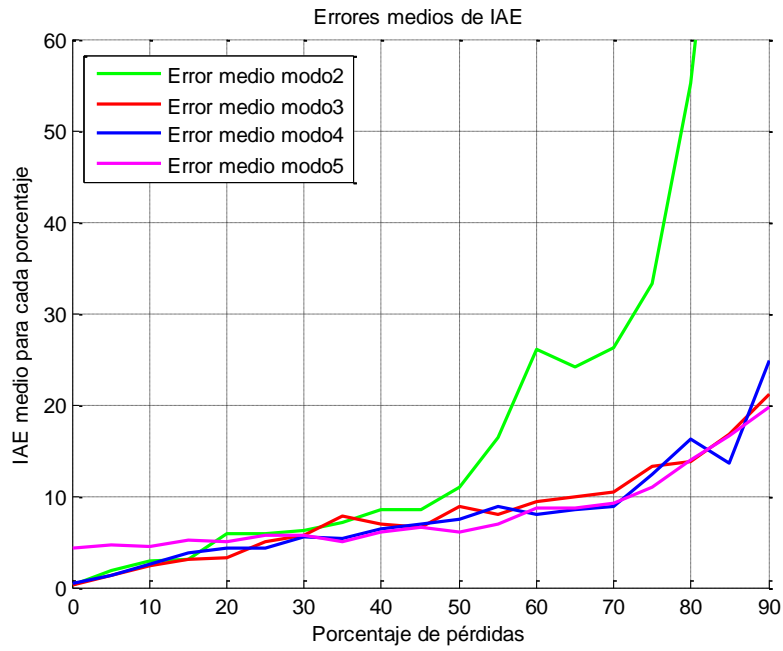


Figura 172. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

CRHPC: $N=12$, $m=2$, $Nu=10$, $K=-0.6$

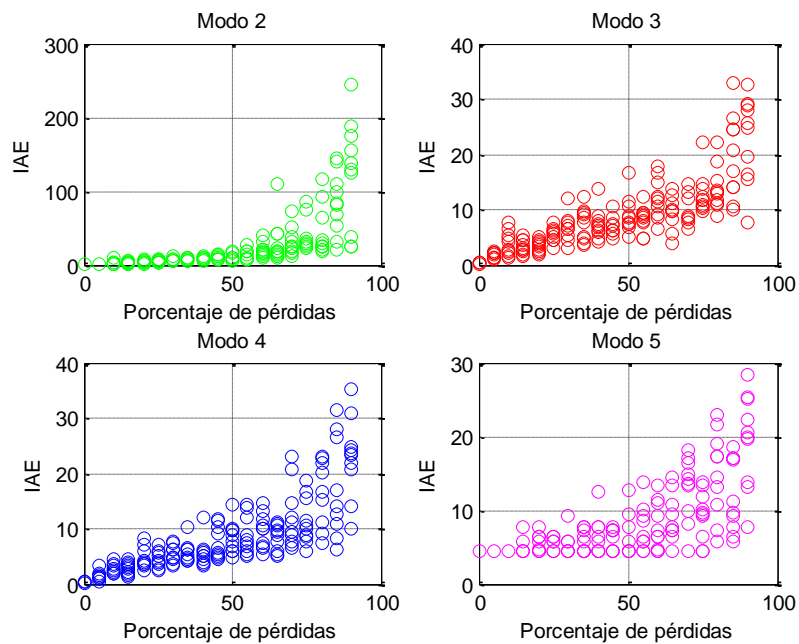


Figura 173. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

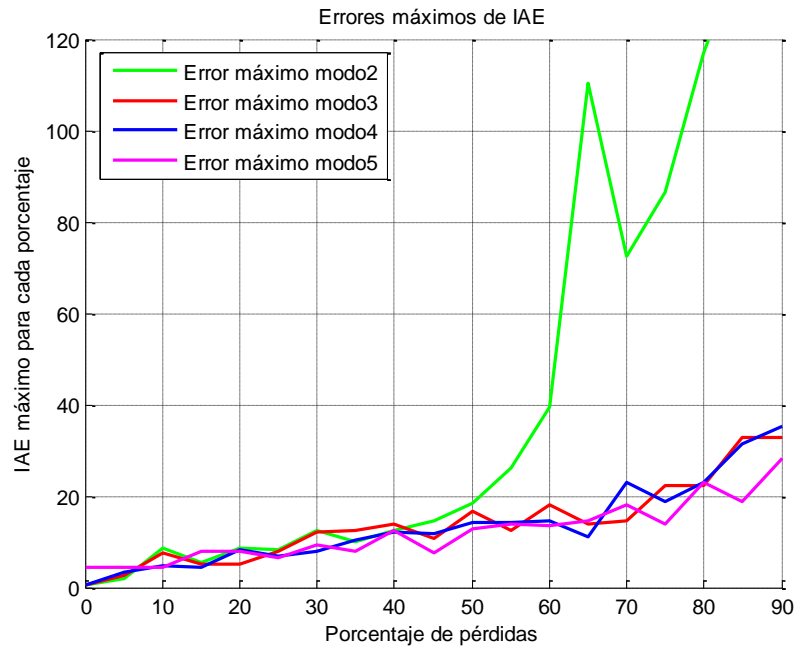


Figura 174. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

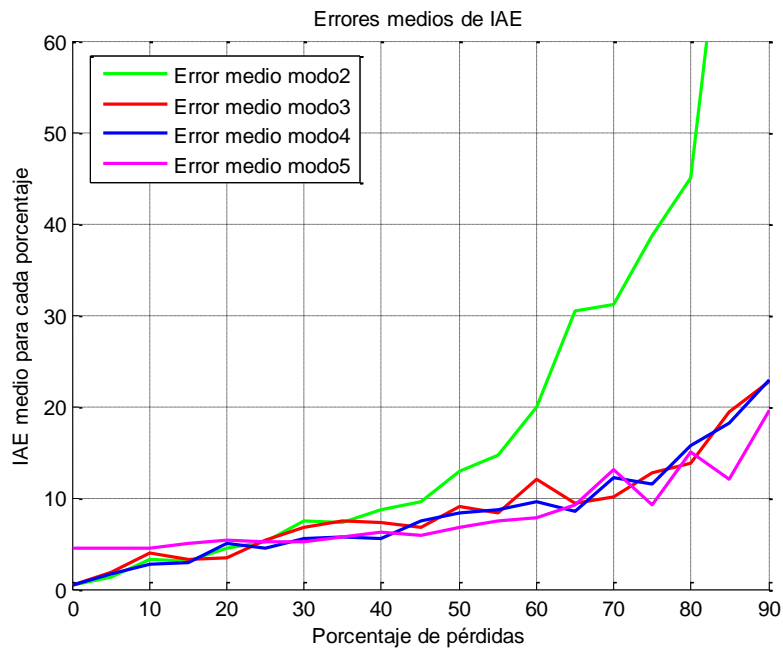


Figura 175. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

B.3 Proceso con retardo múltiple del periodo de muestreo

El proceso con retardo elegido aparece referenciado en la tesis Carlos Bordons (Bordóns 1994) (pág. 59) y en el libro (Camacho y Bordóns 2004) (pág. 46 y 95), y consiste en un calentador de agua, cuyo modelo continuo es el siguiente:

$$G_p(s) = \frac{U_s}{U_e} = \frac{0.2965}{s - 0.1802} \cdot e^{-4 \cdot s}$$

El periodo elegido para discretizar el proceso en las referencias anteriores es de $T_s=1$ segundo, obteniendo el siguiente proceso discreto:

$$G_p(z^{-1}) = \frac{U_s}{U_e} = \frac{0.2713 \cdot z^{-1}}{1 - 0.8351 \cdot z^{-1}} \cdot z^{-4}$$

B.3.1 Elección de los parámetros del sistema distribuido de control

Para el sistema distribuido de control formado por un nodo supervisor, en el que se ejecuta un control predictivo, y un nodo local conectado directamente al proceso que se desea controlar, se deben obtener los parámetros del sistema distribuido de control: T_s , N , m , N_u y K , dejando los siguientes valores fijos: $\alpha=1$, $\lambda=0.1$ y $T(z^{-1})=1$.

Regulador predictivo GPC

A continuación se muestra la simulación del sistema con el regulador GPC para un periodo de 1 segundo:

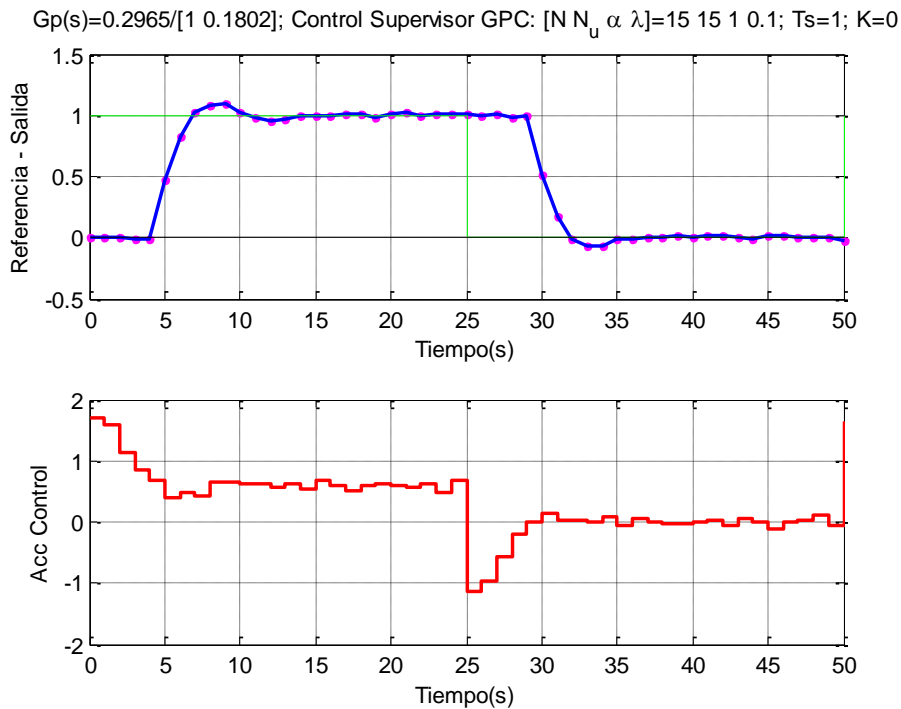


Figura 176. Proceso con retardo con el controlador GPC

Se elige un horizonte de predicción de $N=15$ instantes que permite englobar a toda la dinámica del sistema. Con este horizonte de predicción el error mínimo se consigue para el horizonte de control de $Nu=6$.

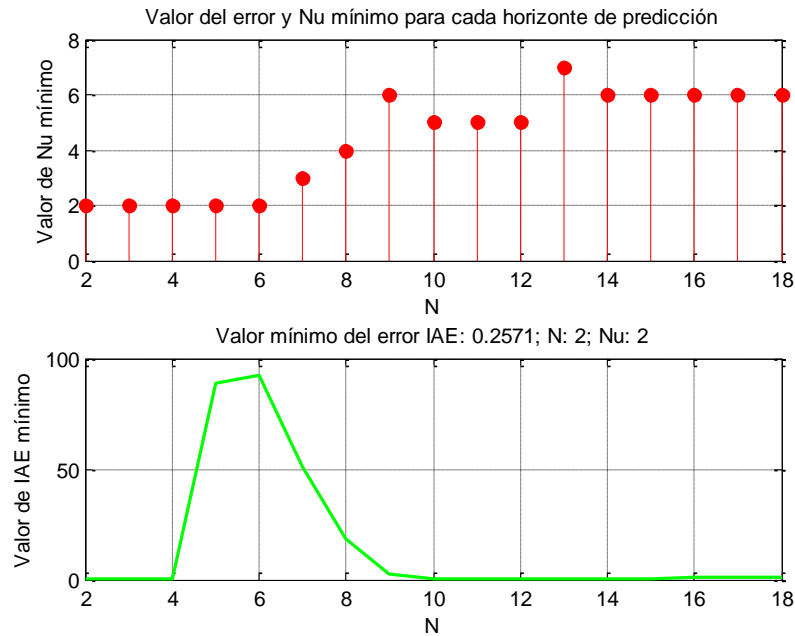


Figura 177. Estudio del valor de Nu en función de N con el que se obtiene el mínimo error

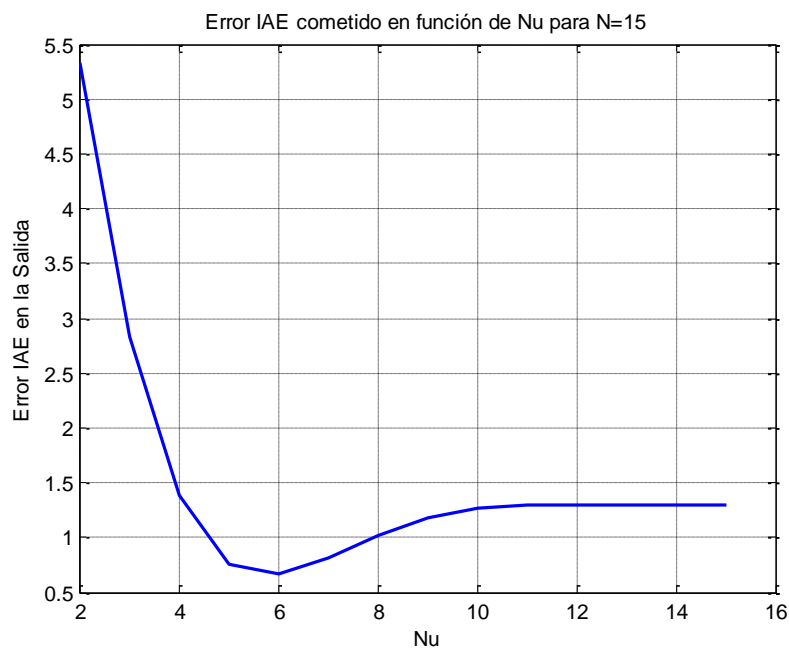


Figura 178. Errores cometidos en función de Nu para $N=15$

La simulación del sistema, con los parámetros del control elegidos anteriormente, ante una pérdida masiva de datos a partir del instante 2 se muestra en la siguiente

figura, con la que se obtiene el valor de la ganancia de la compensación de la acción de control.

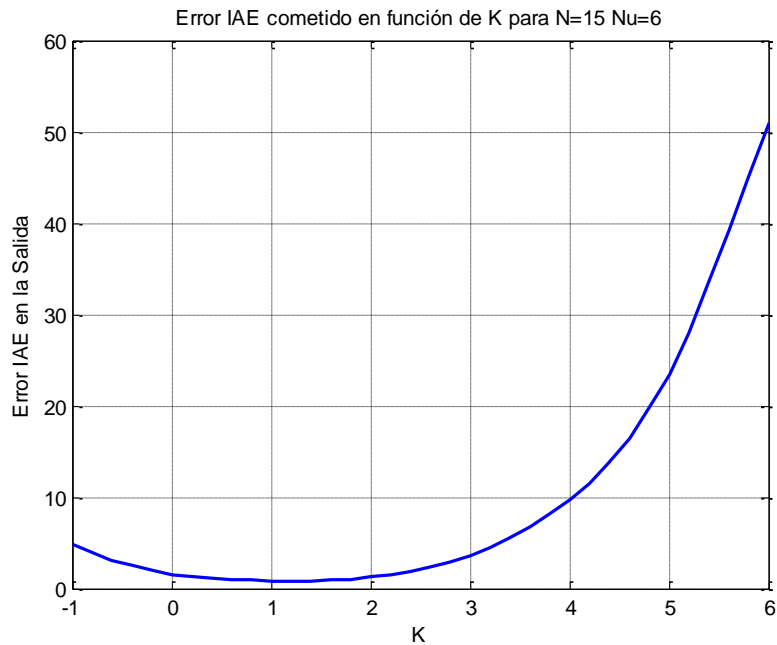


Figura 179. Errores cometidos en función de K para N=15 y Nu=6

El valor mínimo de IAE=0.9466 se obtiene para una K=1.2, pudiendo utilizar valores de K entre 0 y 2.5, ya que están muy cerca de ese valor mínimo.

Siguiendo con esta metodología, a continuación se presenta la selección de los parámetros del control distribuido para el caso del controlador CRHPC:

Regulador predictivo CRHPC

$G_p(s)=0.2965/[1 \ 0.1802]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=13 \ 2 \ 15 \ 1 \ 0.1$; $T_s=1$; $K=0$

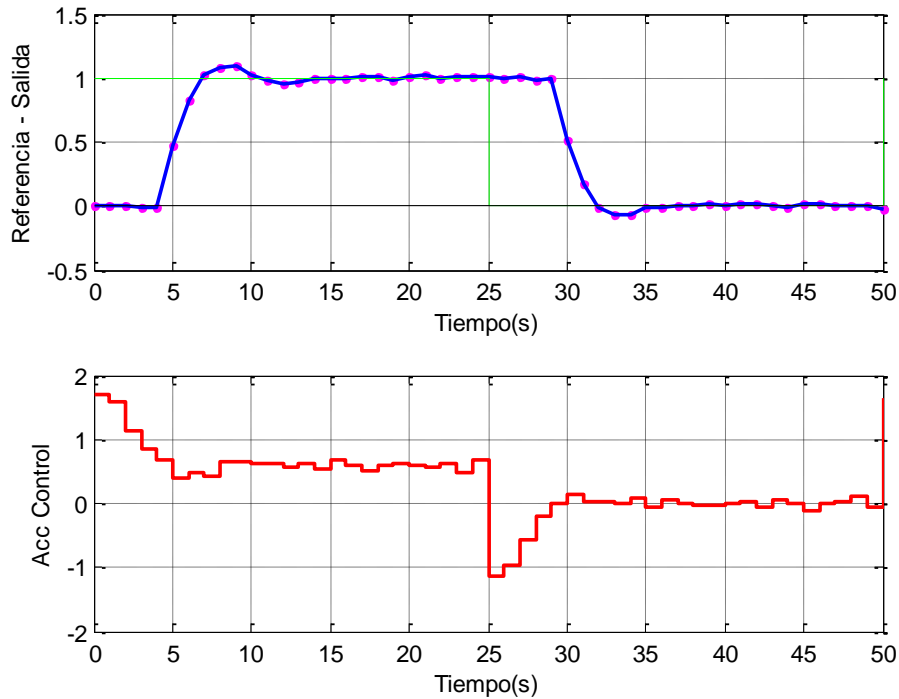


Figura 180. Proceso con retardo con el controlador CRHPC

Se elige un horizonte de predicción de 15 instantes ($N+m$), en el que los dos últimos pertenecen a la restricción terminal ($m=2$).

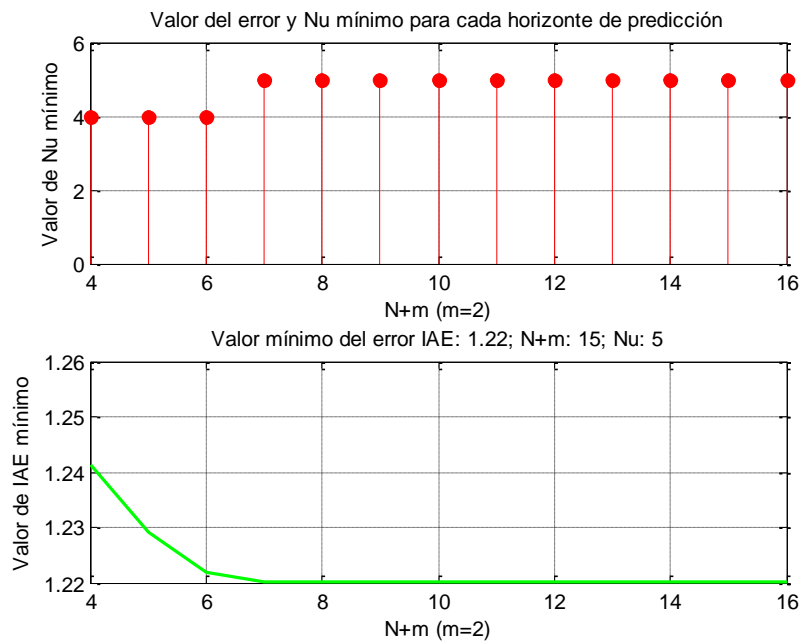


Figura 181. Estudio del valor de N_u en función de $N+m$ con el que se obtiene el mínimo error

Se toma un horizonte de predicción de $N+m=15$ instantes que permite englobar a toda la dinámica del sistema, con la restricción terminal $m=2$. Con este horizonte de predicción el error mínimo se consigue para el horizonte de control de $N_u=5$.

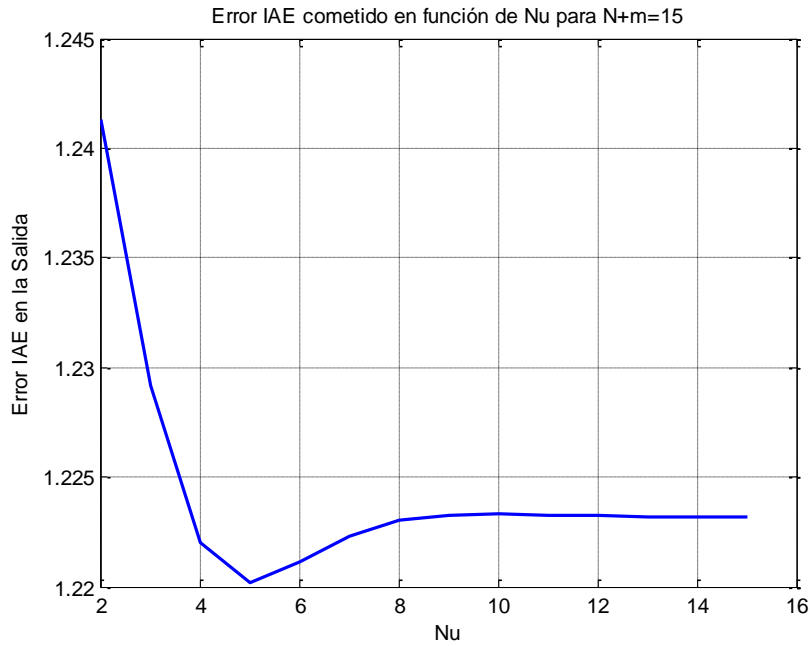


Figura 182. Errores cometidos en función de N_u para $N+m=15$, con $m=2$

Finalmente se simula el sistema con los parámetros anteriores, ante una pérdida masiva de datos a partir de instante 2, para obtener el valor de la ganancia de la compensación de la acción de control.

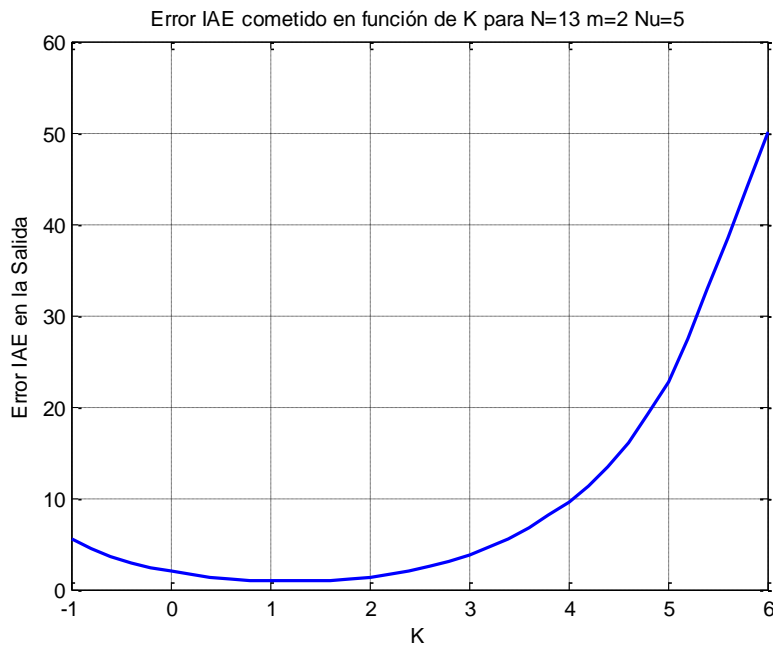


Figura 183. Errores cometidos en función de K para $N=13$, $m=2$ y $N_u=5$

Con $K=1.2$ se obtiene el mínimo valor de $IAE=1.034$. Al igual que ocurría en el caso del GPC para valores de K entre 0 y 2.5 el error cometido está muy cerca del mínimo.

B.3.2 Comportamiento del sistema distribuido de control

A continuación se muestra el comportamiento del sistema distribuido de control mediante dos casos, en función del tipo de comunicaciones entre el nodo local y el nodo supervisor, utilizando los parámetros obtenidos en el apartado anterior.

Caso 1. Sistema distribuido con canal de comunicaciones exclusivo

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} . En esta prueba, igual que en las presentadas anteriormente, por cada porcentaje de pérdida de datos, se han realizado 10 simulaciones distribuidas de forma aleatoria a lo largo de todo el tiempo de la prueba. Esto supone la realización de 190 simulaciones por cada modo de funcionamiento.

A continuación se muestran los resultados de los distintos índices de error para los distintos modos de funcionamiento:

GPC: $N=15$, $Nu=6$, $K=1.2$

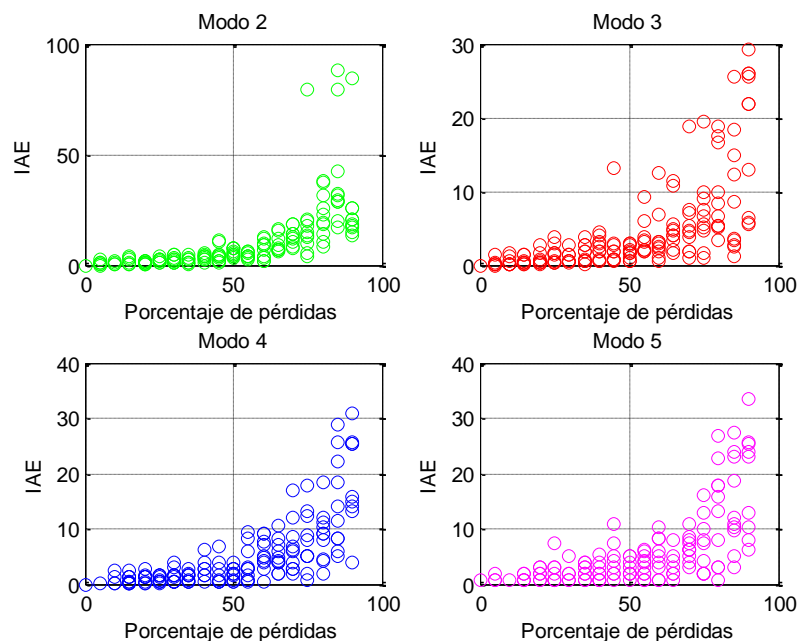


Figura 184. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

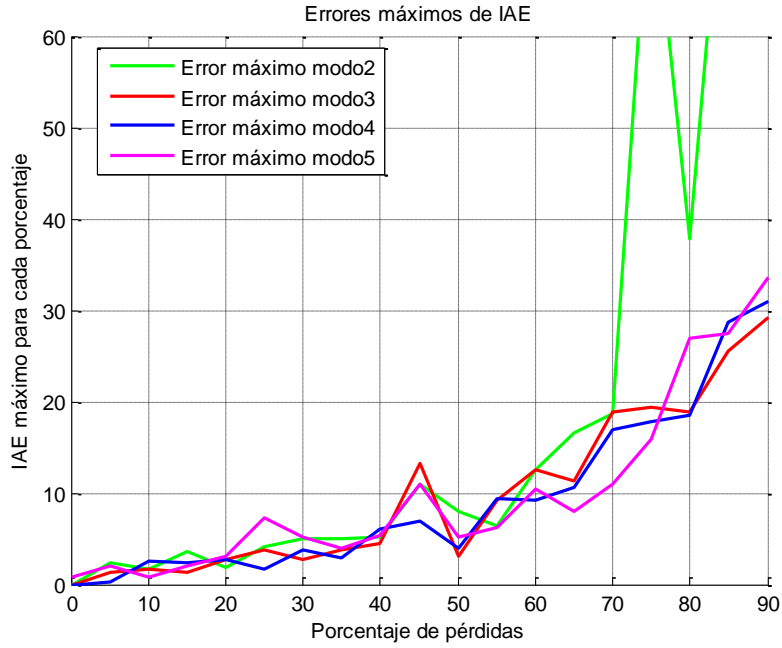


Figura 185. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

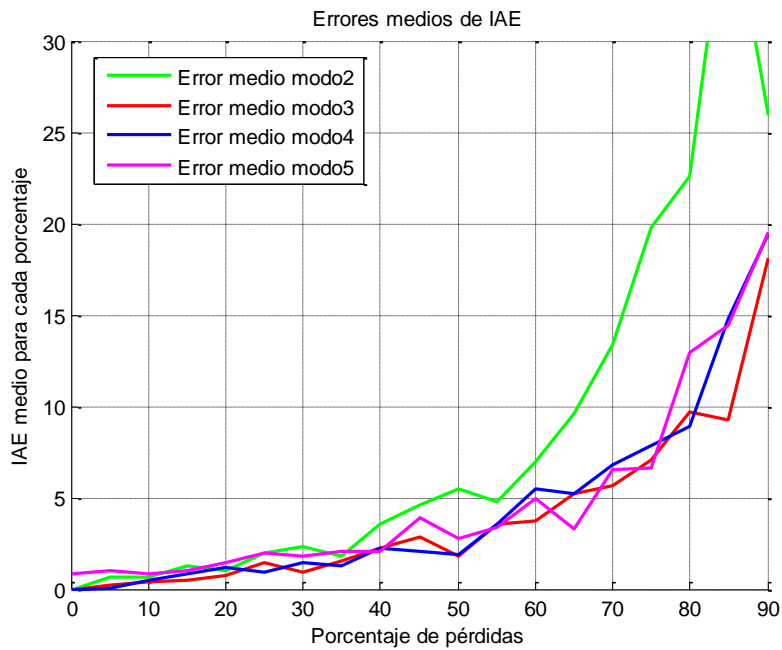


Figura 186. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

CRHPC: $N=13$, $m=2$, $N_u=5$, $K=1.2$

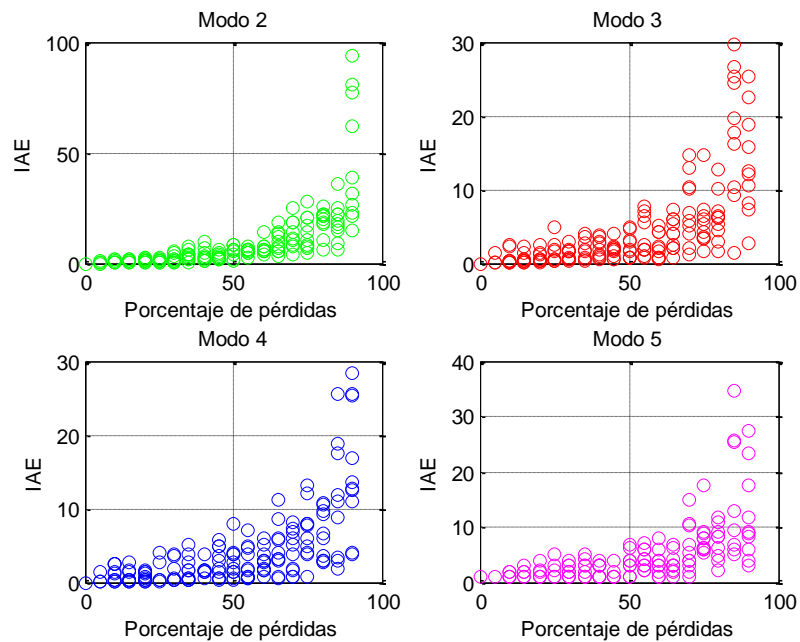


Figura 187. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

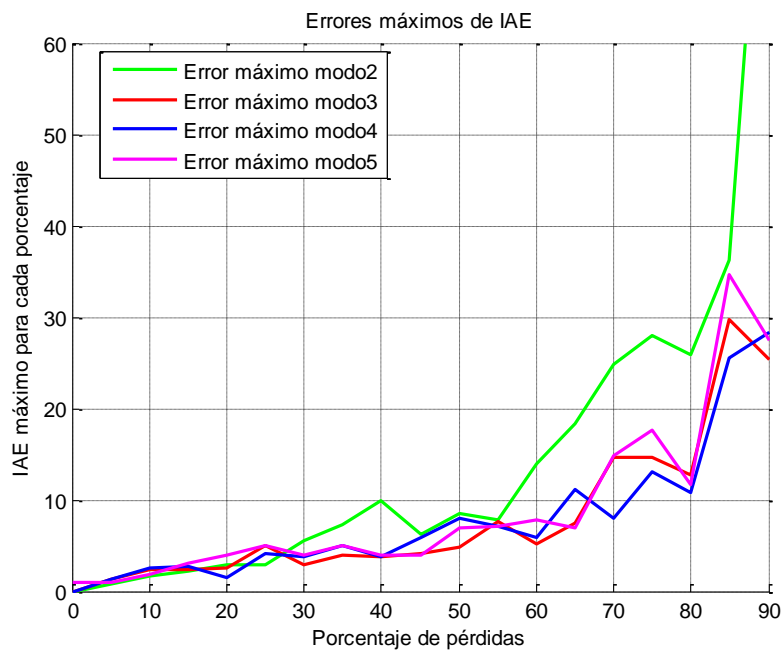


Figura 188. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

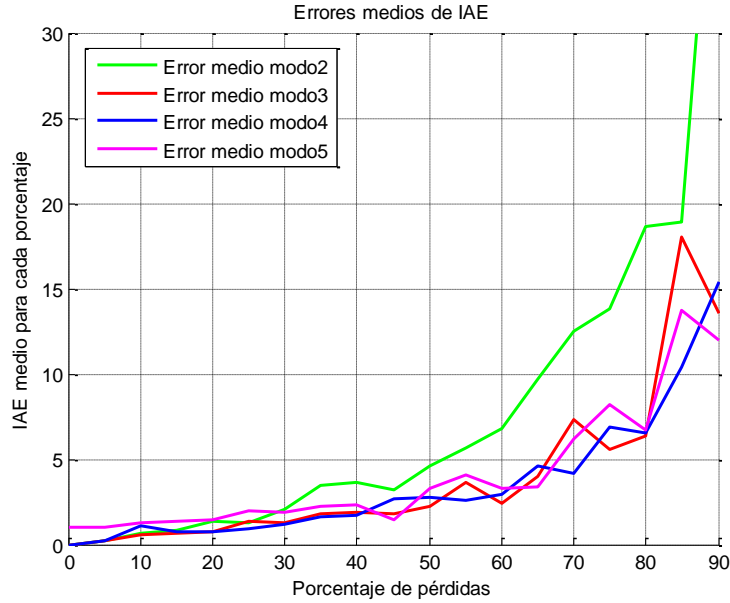


Figura 189. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

Caso 2. Sistema distribuido con canal de comunicaciones compartido

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} . Además se ha introducido una interferencia que consume un 10% del ancho de banda y un coste computacional del cálculo del regulador predictivo de 10 ms.

GPC: $N=15$, $Nu=6$, $K=1.2$

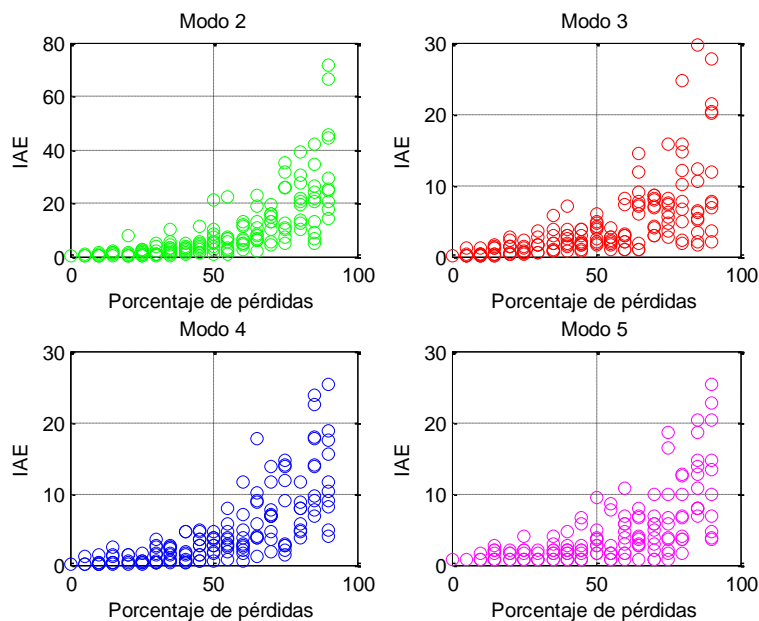


Figura 190. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

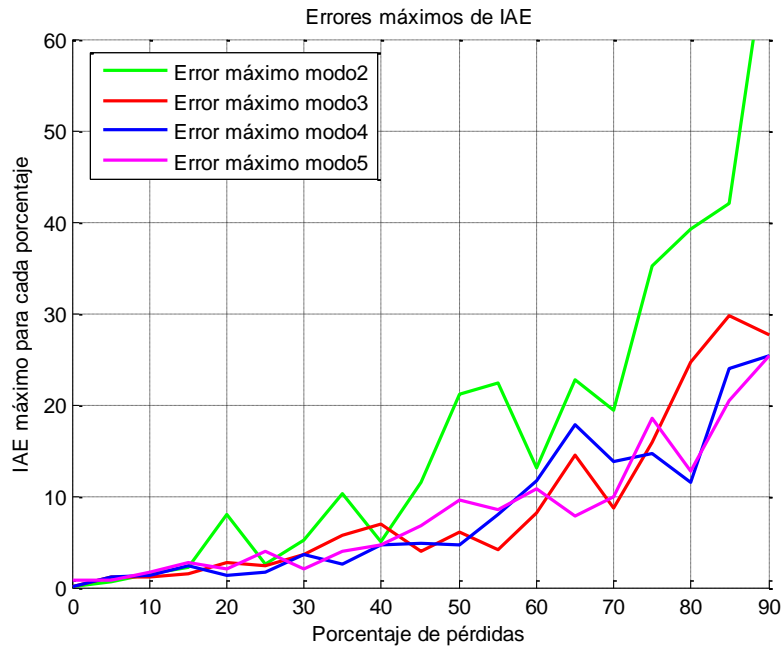


Figura 191. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

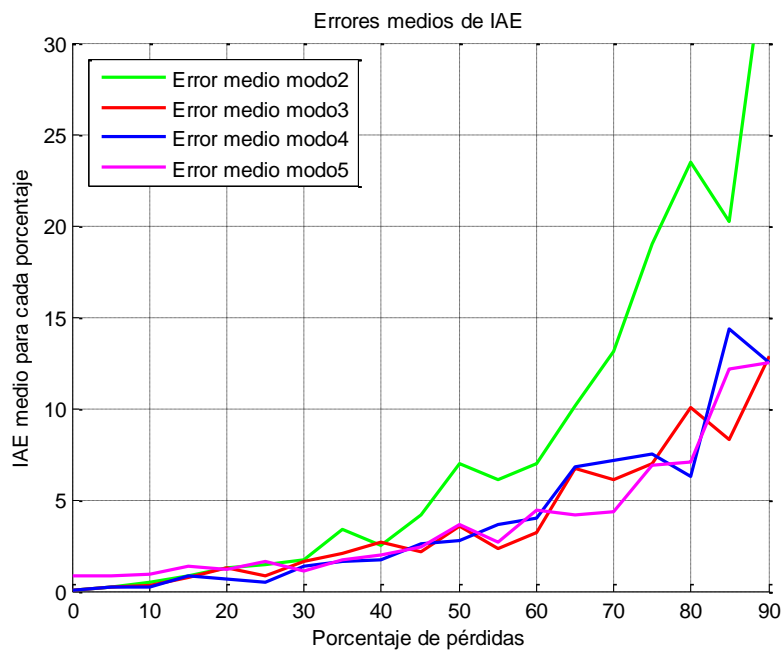


Figura 192. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

CRHPC: $N=13$, $m=2$, $Nu=5$, $K=1.2$

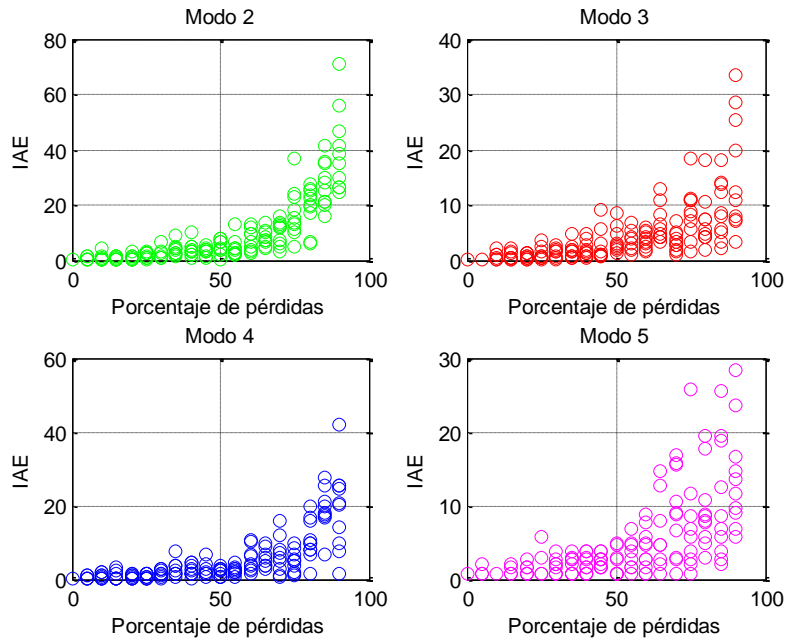


Figura 193. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

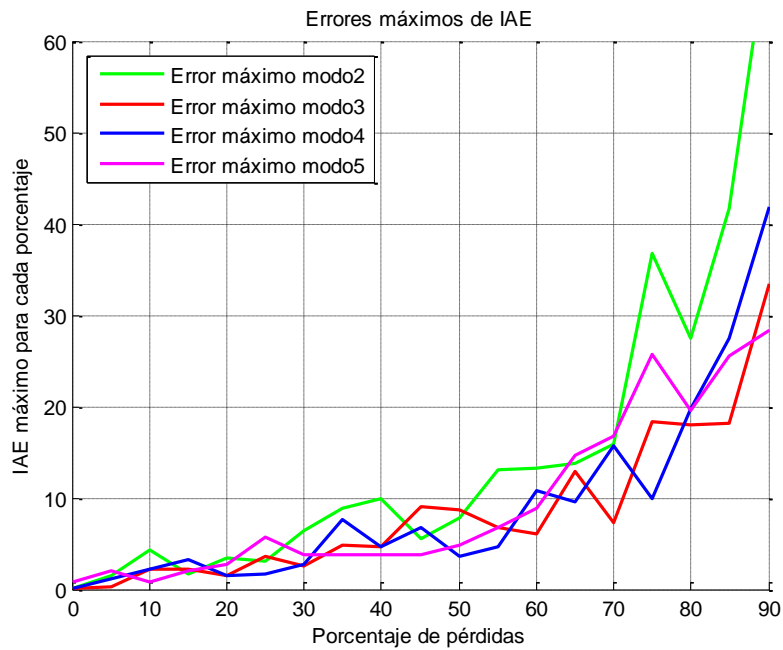


Figura 194. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

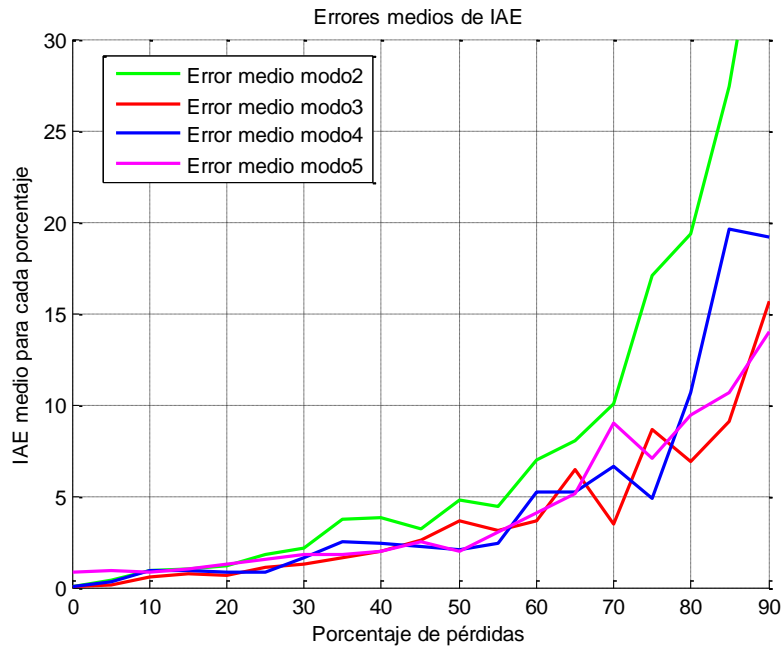


Figura 195. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

B.4 Proceso con efecto integral

El proceso con efecto integral escogido es el control de posición de un motor de corriente continua. Este proceso también es utilizado en la tesis de Carlos Bordons (Bordóns 1994) (pág. 74) y en el libro de Camacho-Bordons (Camacho y Bordóns 2004) (pág. 109) para la realización de pruebas con controladores predictivos.

La posición del motor de corriente continua se ajusta al siguiente modelo:

$$G_p(s) = \frac{\theta(s)}{U(s)} = \frac{K}{s \cdot (1 + \tau \cdot s)}$$

Donde los parámetros constructivos para el motor considerado son $K=2.9776$, $\tau=0.1613$ y retardo nulo, que pertenecen a un motor de la marca Feedback del que se dispone en el laboratorio y con el que se pueden hacer pruebas reales. El modelo queda de la siguiente forma:

$$G_p(s) = \frac{\theta(\text{voltios})}{U(\text{voltios})} = \frac{2.9776}{s \cdot (1 + 0.1613 \cdot s)} = \frac{18.46}{s \cdot (s + 6.2)}$$

La relación entre la posición en grados y los voltios que suministra el sensor de posición es:

$$\text{Posición (grados)} = 17.4597 \times \text{Tensión_potenciómetro (voltios)} + 3.1325$$

B.4.1 Elección de los parámetros del sistema distribuido de control

Para el sistema distribuido de control formado por un nodo supervisor, en el que se ejecuta un control predictivo, y un nodo local conectado directamente al proceso que se desea controlar, se deben obtener los parámetros del sistema distribuido de control: T_s , N , m , N_u y K , dejando los siguientes valores fijos: $\alpha=1$, $\lambda=0.1$ y $T(z^{-1})=1$.

Teniendo en cuenta la dinámica del sistema en bucle cerrado se ha tomado un periodo de 100 milisegundos.

Regulador predictivo GPC

A continuación se muestra la simulación del sistema con el regulador GPC para un periodo de 100 milisegundos:

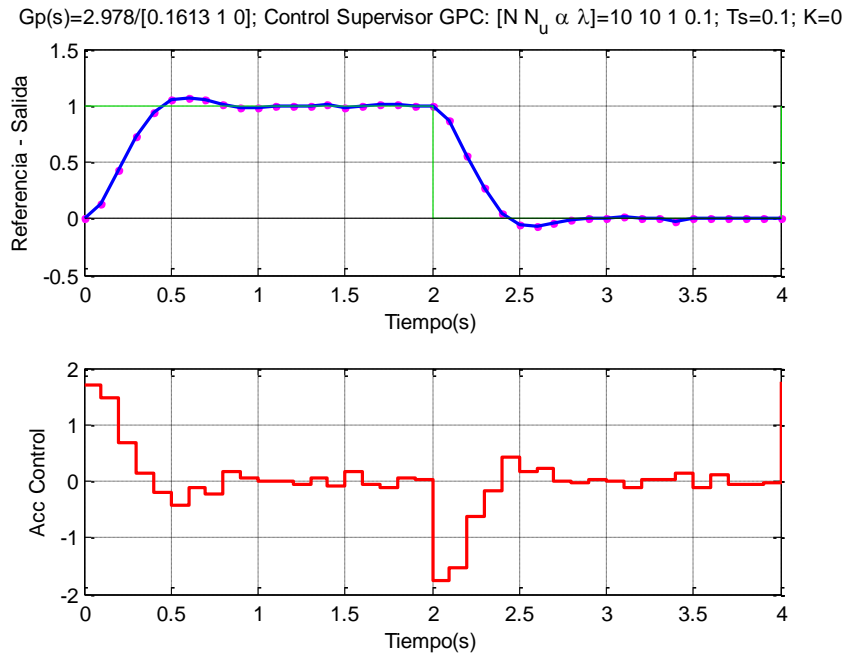


Figura 196. Proceso con efecto integral con el controlador GPC

Se elige un horizonte de predicción de $N=12$ instantes que permite englobar a toda la dinámica del sistema. Con este horizonte de predicción el error mínimo se consigue para el horizonte de control de $N_u=3$.

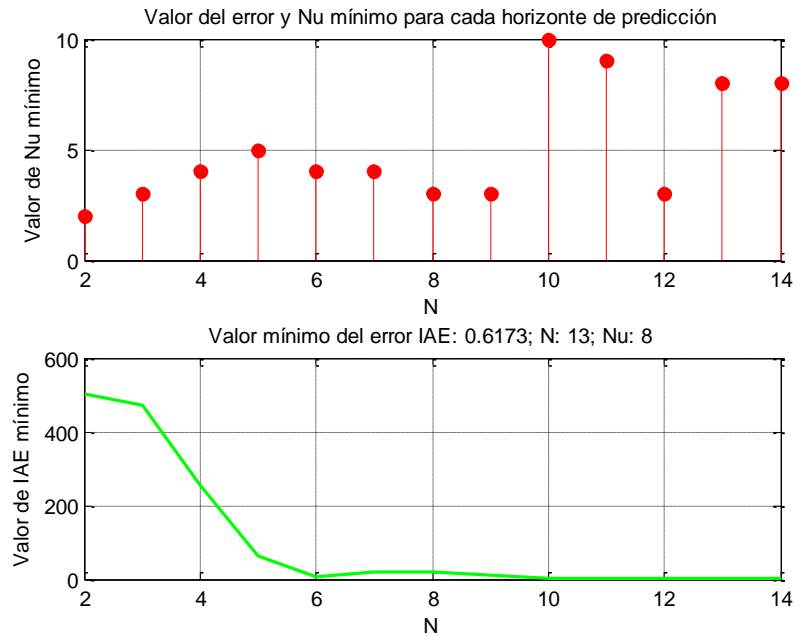


Figura 197. Estudio del valor de Nu en función de N con el que se obtiene el mínimo error

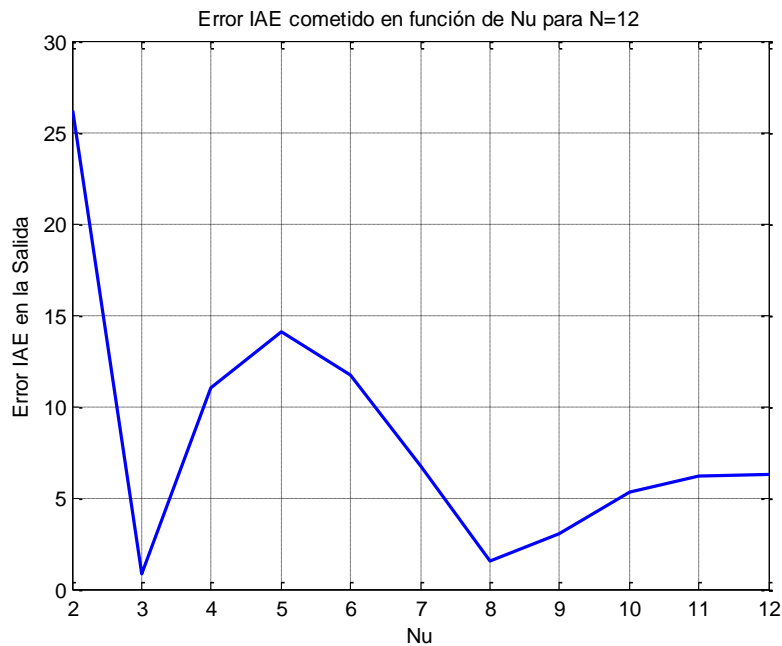


Figura 198. Errores cometidos en función de Nu para N=12

La simulación del sistema, con los parámetros del control elegidos anteriormente, ante una pérdida masiva de datos a partir del instante 2, se muestra en la siguiente figura, con la que se obtiene el valor de la ganancia de la compensación de la acción de control.

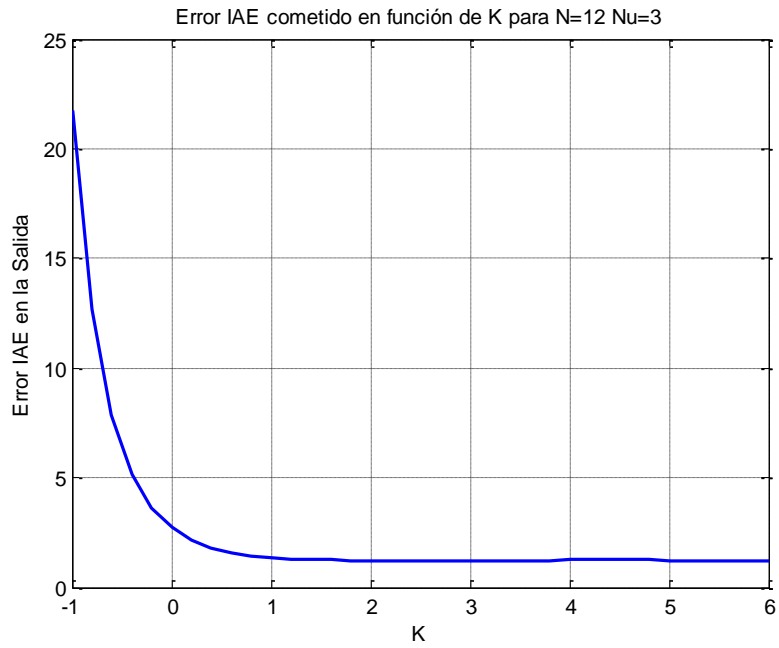


Figura 199. Errores cometidos en función de K para N=12 y Nu=3

El valor mínimo de IAE=1.183 se obtiene para una K=6. En este caso el rango de valores en los que el error está muy cerca del mínimo está comprendido entre 0.5 y 6.

Regulador predictivo CRHPC

Siguiendo la metodología anterior:

$G_p(s)=[0 \ 0 \ 2.978]/[0.1613 \ 1 \ 0]$; Control Supervisor CRHPC: $[N \ m \ N_u \ \alpha \ \lambda]=[8 \ 2 \ 10 \ 1 \ 0.1]$; $T_s=0.1$; $K=0$

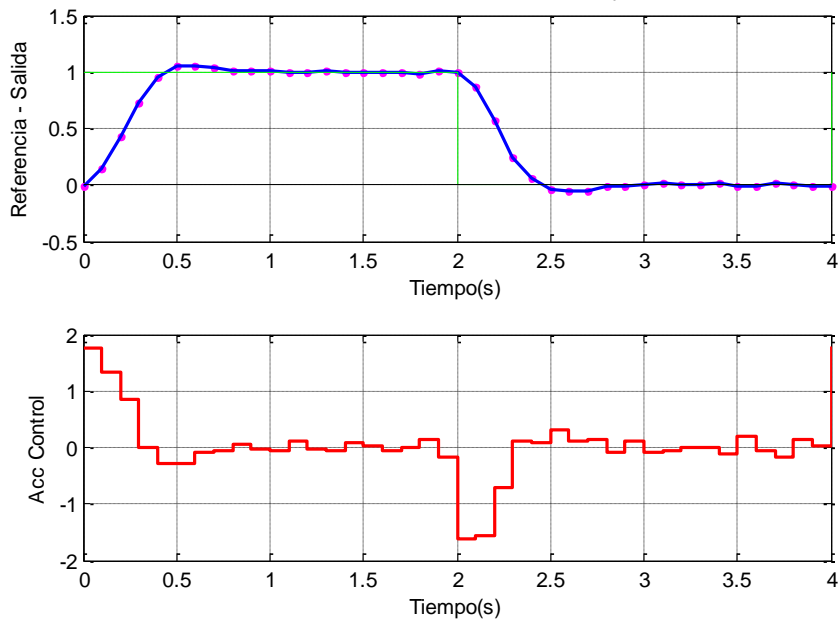


Figura 200. Proceso con efecto integral con el controlador CRHPC

Se elige un horizonte de predicción de 8 instantes ($N+m$), en el que los dos últimos pertenecen a la restricción terminal ($m=2$), para englobar a toda la respuesta dinámica del proceso. Para este horizonte de predicción el valor de N_u que consigue el mínimo error es de 6 instantes. En la Figura 202 se muestran los errores cometidos en función de N_u para el horizonte de predicción elegido.

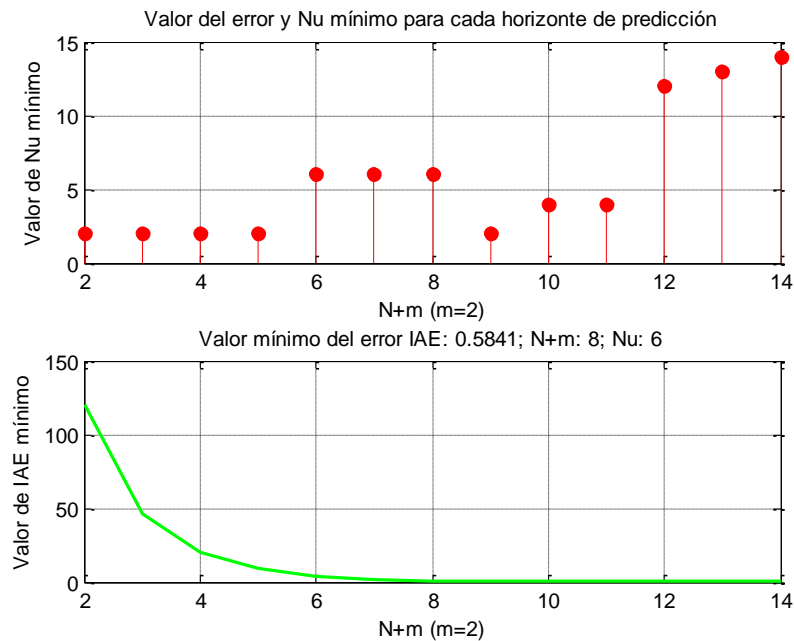


Figura 201. Estudio del valor de N_u en función de $N+m$ con el que se obtiene el mínimo error

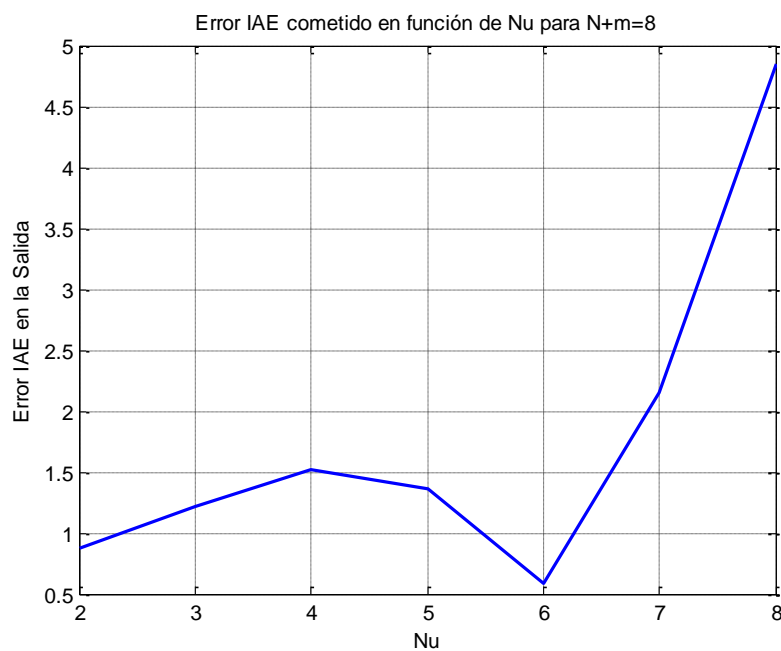


Figura 202. Errores cometidos en función de N_u para $N+m=8$, con $m=2$

Finalmente se simula el sistema con los parámetros anteriores ante una pérdida masiva de datos a partir del instante 2 para obtener el valor de la ganancia de la compensación de la acción de control.

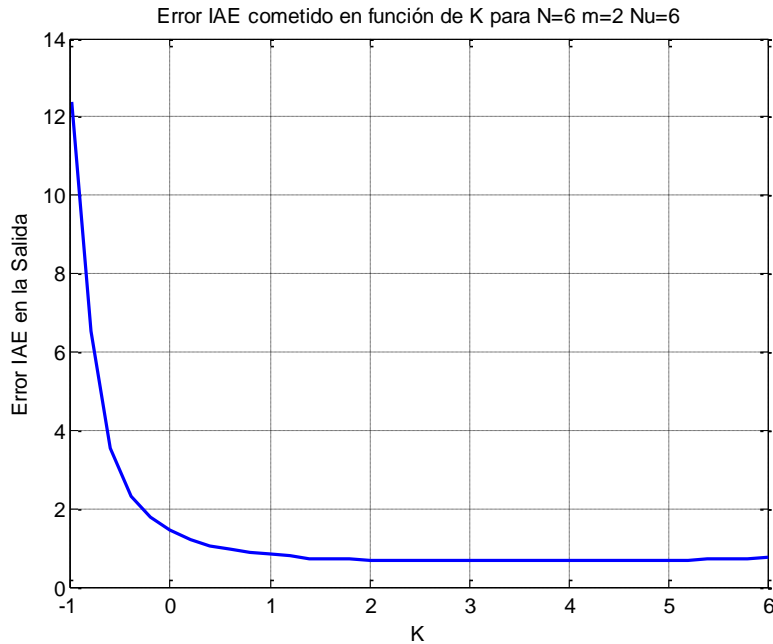


Figura 203. Errores cometidos en función de K para $N=6$, $m=2$ y $Nu=6$

Con $K=4$ se obtiene el mínimo valor de $IAE=0.6576$. En este caso se puede tomar un valor de K entre 0.5 y 6, ya que el error cometido es similar al valor mínimo.

B.4.2 Comportamiento del sistema distribuido de control

A continuación se muestra el comportamiento del sistema distribuido de control mediante dos casos, en función del tipo de comunicaciones entre el nodo local y el nodo supervisor, utilizando los parámetros obtenidos en el apartado anterior.

Caso 1. Sistema distribuido con canal de comunicaciones exclusivo

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} . Por cada porcentaje de pérdida de datos se realizan 10 simulaciones distribuidas de forma aleatoria a lo largo de todo el tiempo de la prueba. Esto supone la realización de 190 simulaciones por cada modo de funcionamiento.

A continuación se muestran los resultados de los distintos índices de error para los distintos modos de funcionamiento:

GPC: $N=12$, $Nu=3$, $K=6$

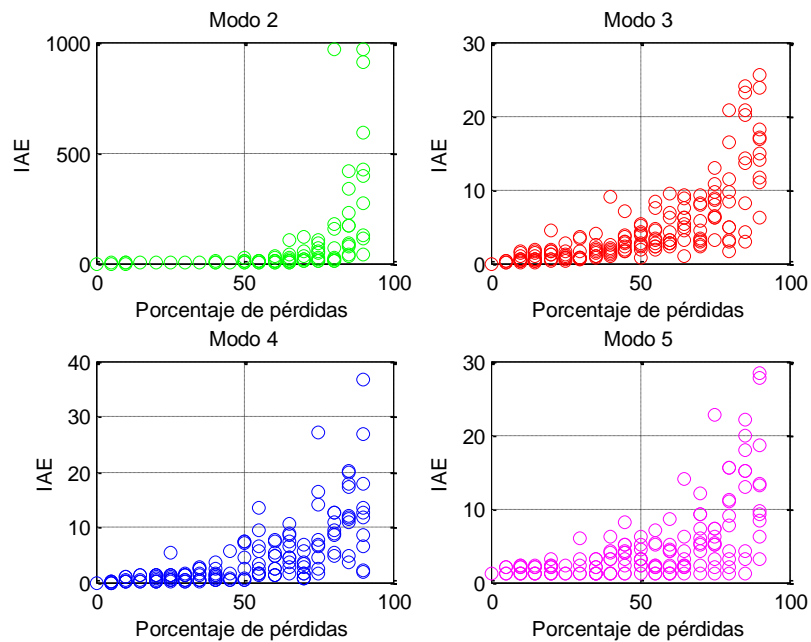


Figura 204. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

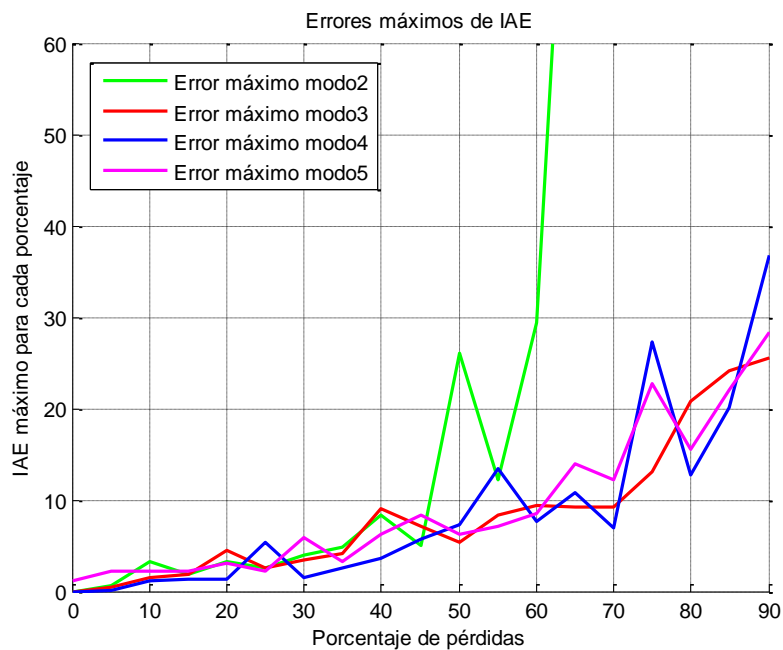


Figura 205. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

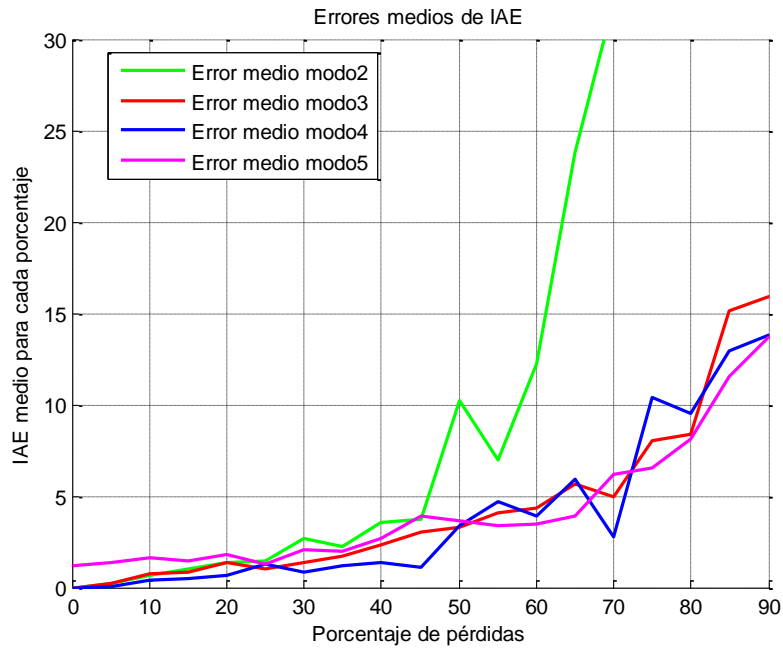


Figura 206. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

CRHPC: $N=6$, $m=2$, $Nu=6$, $K=4$

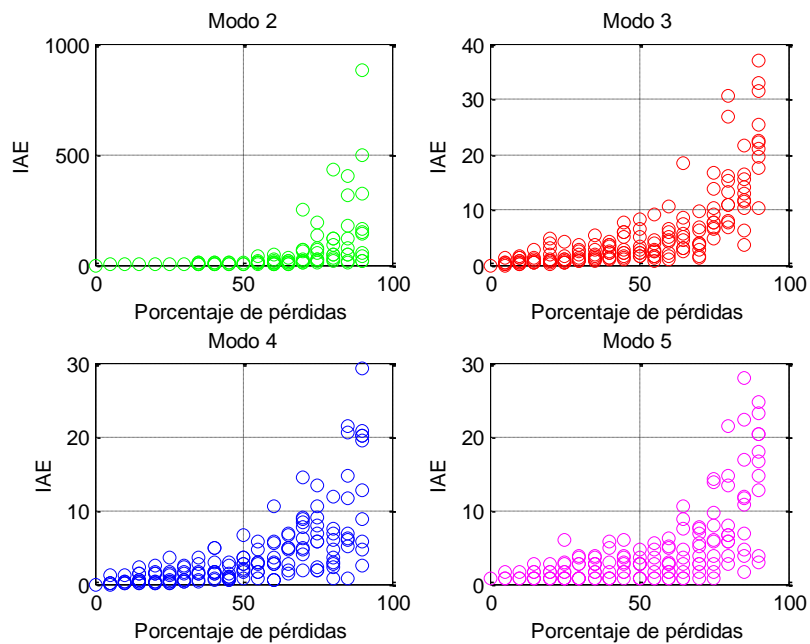


Figura 207. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

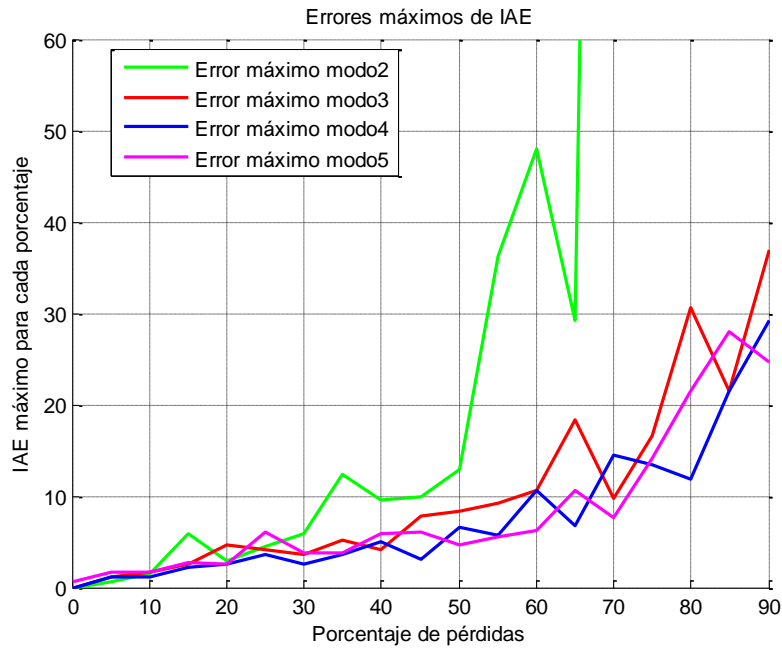


Figura 208. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

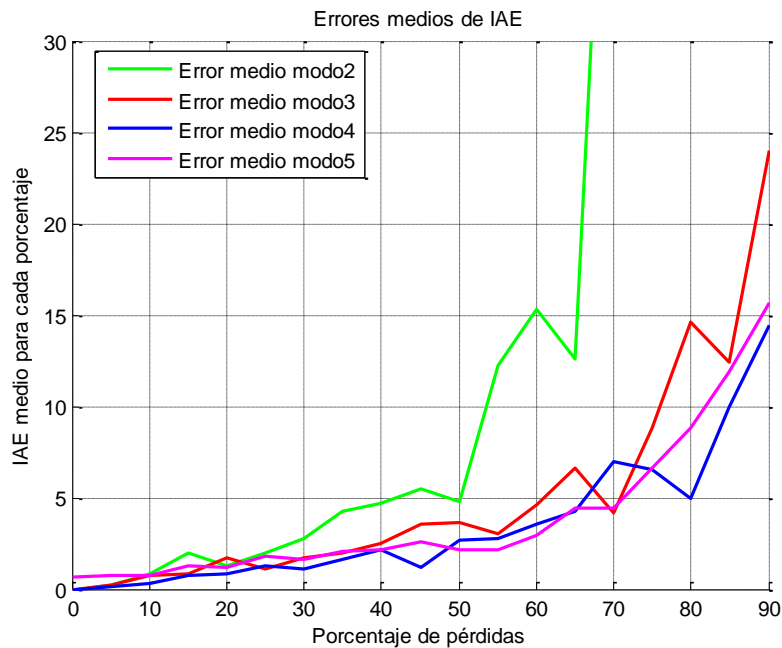


Figura 209. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

Caso 2. Sistema distribuido con canal de comunicaciones compartido

La simulación del ruido se ha tomado de media nula y desviación típica 5×10^{-5} , es decir, la misma que en la prueba anterior. Además se ha introducido una

interferencia en la red que consume un 10% del ancho de banda y en este caso se tiene en cuenta el coste computacional de regulador predictivo 1 ms.

GPC: $N=12$, $Nu=3$, $K=6$

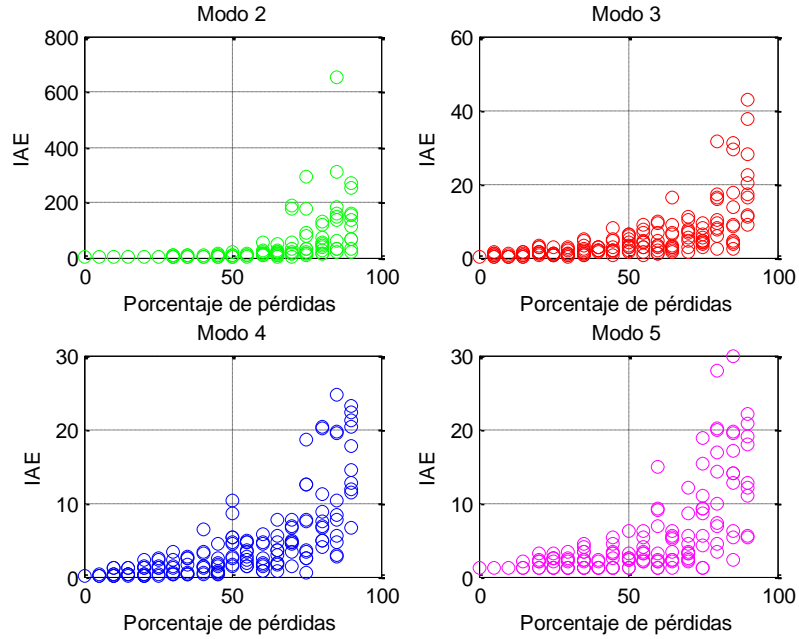


Figura 210. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor GPC

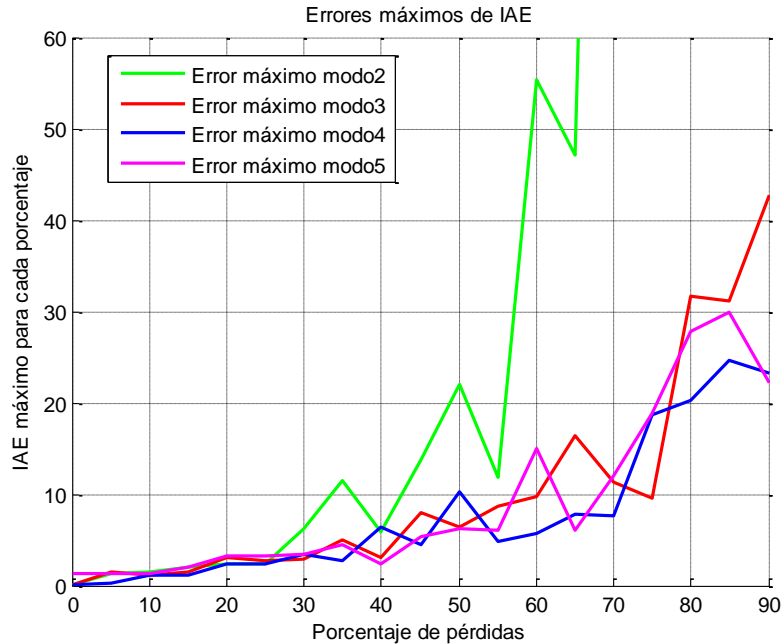


Figura 211. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

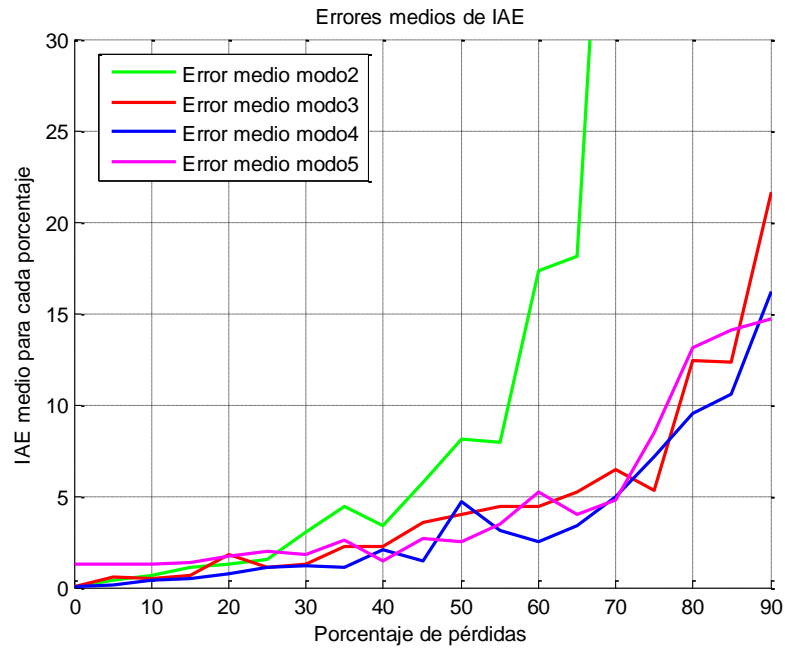


Figura 212. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor GPC

CRHPC: $N=6$, $m=2$, $Nu=6$, $K=4$

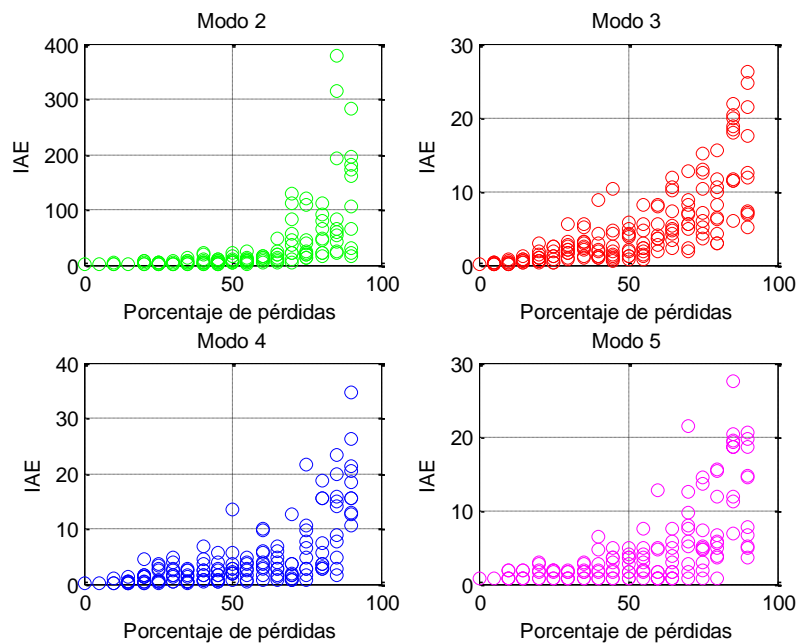


Figura 213. IAE para distintos porcentajes de pérdida de datos en los distintos modos de funcionamiento utilizando un control supervisor CRHPC

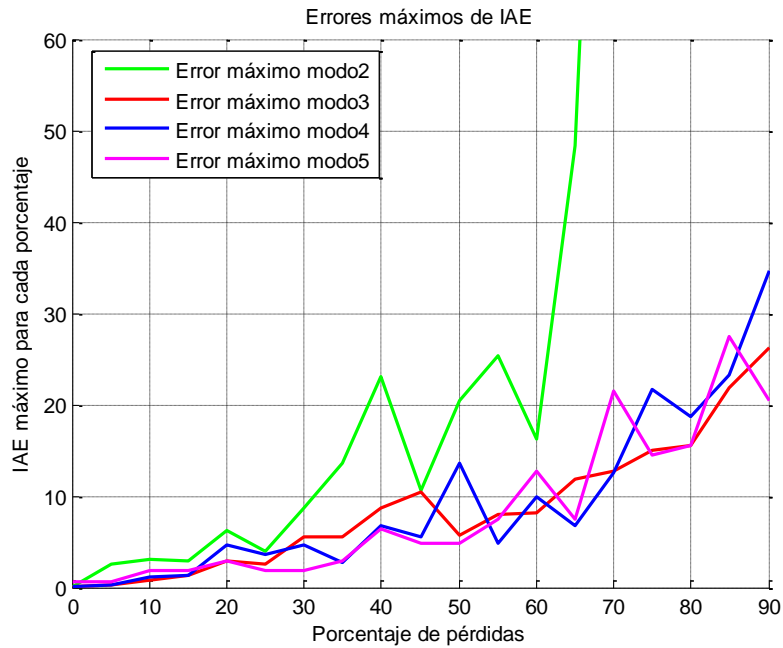


Figura 214. Comparación de los IAE máximos para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

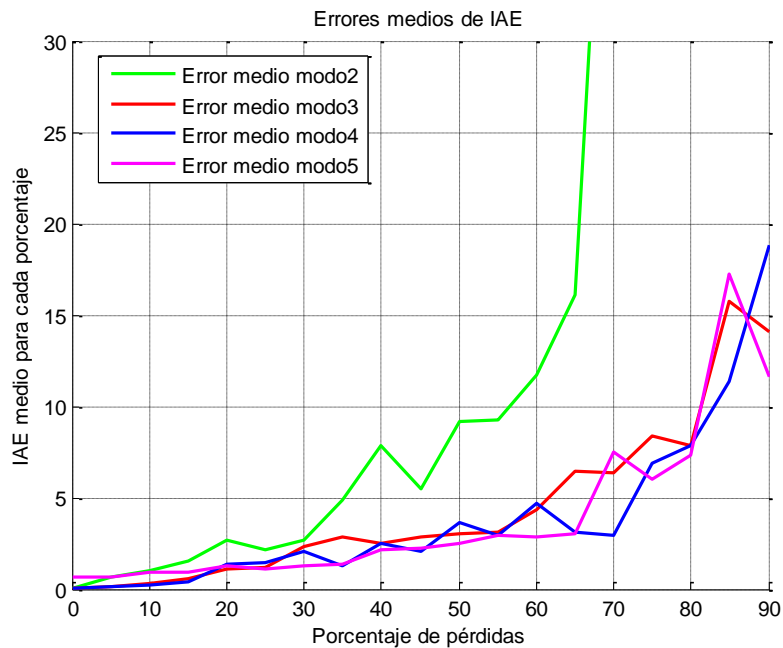


Figura 215. Comparación de los IAE medios para cada porcentaje de pérdida de datos en los distintos modos de funcionamiento para un control supervisor CRHPC

Anexo C. Utilización de controladores predictivos en el nodo supervisor

Los controladores predictivos basados en modelo permiten la obtención de las acciones de control y de las salidas mediante la minimización de una función de coste, y por lo tanto es un método muy adecuado para la obtención de las trayectorias nominales que serán calculadas por el control supervisor y enviadas hacia el control local.

En este apartado se realiza una visión de los controladores predictivos y se hace un estudio de los dos utilizados en este trabajo y que han sido implementados en la herramienta de simulación.

C.1 Introducción

En el nodo supervisor se necesita de un regulador de alto nivel que proponga las acciones de control futuras que deben aplicarse a un proceso para que la salida del sistema cumpla con la especificación deseada. Un controlador que cumple con estas características puede ser un controlador predictivo basado en modelo (MPC).

Cuando se utilizan controladores que minimizan una función de coste se obtiene una secuencia de control óptima para el índice de coste propuesto. La estrategia utilizada es la del horizonte móvil, en la que en cada iteración se vuelve a calcular la secuencia de control con la nueva información disponible, desechando las acciones de control postuladas. En un sistema con recursos escasos de computación la información de las acciones de control futuras es muy valiosa, puesto que podría ser utilizada, dentro del horizonte de predicción, para aplicarlas en caso de pérdidas de datos o de utilización excesiva del tiempo de cómputo para ese periodo.

La implementación de reguladores predictivos en el nodo supervisor proporciona un método muy adecuado de suministrar las trayectorias de la acción de control y de la salida, sin embargo, debido a que el control local podría utilizar las acciones postuladas se hace necesario una adecuada elección de los parámetros del control.

En este apartado se realiza un estudio de los principales controladores predictivos y como utilizar la información disponible de acciones de control futuras en caso de necesidad.

C.2 Características del control predictivo basado en modelo

El control predictivo basado en modelo (MBPC¹³ ó MPC¹⁴) son un conjunto de técnicas que formulan el criterio de control en un instante determinado mediante una función explícita de las salidas futuras de la planta y en algunas ocasiones de las entradas de la misma (Clarke, Mohtadi y Tuffs 1987). No es una estrategia de control específica, sino que se trata más bien de un campo muy amplio de métodos de control desarrollados en torno a ciertas ideas comunes. Estos métodos de diseño conducen a controladores lineales que poseen prácticamente la misma estructura y presentan suficientes grados de libertad. Sus principales características son:

- Se utiliza de forma explícita un modelo del proceso
- A partir de la minimización de una cierta función de coste, se obtienen en cada instante, no una señal de control sino una secuencia de señales de control óptimas, aunque no tiene por qué ser la solución óptima.
- Se aplica el concepto denominado horizonte móvil, o también llamada estrategia deslizante. Es decir, aunque se obtenga una secuencia de control óptima, sólo se utiliza la primera acción de control de todas ellas, despreciándose las demás. En el instante siguiente, y utilizando la nueva información, se repiten de nuevo los cálculos.

Los distintos algoritmos de MBPC difieren entre sí casi exclusivamente en el modelo usado para representar el proceso y los ruidos y la función a minimizar. Aunque las diferencias puedan parecer pequeñas a priori, pueden provocar distintos comportamientos en bucle cerrado, siendo críticas para el éxito de un determinado algoritmo en una determinada aplicación (Bordóns 1994).

Algunas de las ventajas que presenta el MBPC son:

- La simplicidad de la formulación, que no requiere de complejos desarrollos matemáticos.
- La modificación de la función objetivo a partir de la cual se obtiene la ley de control permite la utilización de estrategias tendentes a conseguir objetivos distintos.
- La función objetivo dispone de parámetros que se pueden sintonizar para ponderar los distintos elementos de la misma de una forma intuitiva.

¹³ MBPC: *Model Based Predictive Control*

¹⁴ MPC: *Model Predictive Control* (también llamado Long-Range Predictive Control o LRPC)

- La minimización del índice asegura que las acciones de control que se obtienen con esta estrategia son óptimas.
- Se puede extender al caso multivariable.
- Se pueden utilizar las referencias futuras para anticipar los cambios.
- Posee intrínsecamente compensación del retardo y de perturbaciones medibles.
- La posibilidad de considerar las restricciones, tanto de las entradas como de las salidas, lo que no sucede con un gran número de estrategias de control.
- Es una metodología completamente abierta basada en algunos principios básicos que permite futuras extensiones.

Entre los inconvenientes se pueden citar los siguientes:

- La sensibilidad de los resultados obtenidos en cuanto a prestaciones, en función de la bondad del modelo utilizado en las predicciones. Las prestaciones obtenidas por el regulador dependerán en gran medida de las discrepancias existentes entre el proceso real y el modelo usado.
- El coste computacional elevado, en especial en aplicaciones en las que se utilizan modelos no lineales, lo que obliga, en ocasiones, a la utilización de técnicas iterativas.

C.3 Estrategia del MBPC

Los controladores MBPC se caracterizan por utilizar la siguiente estrategia, representada en la Figura 216 (Bordóns 1994):

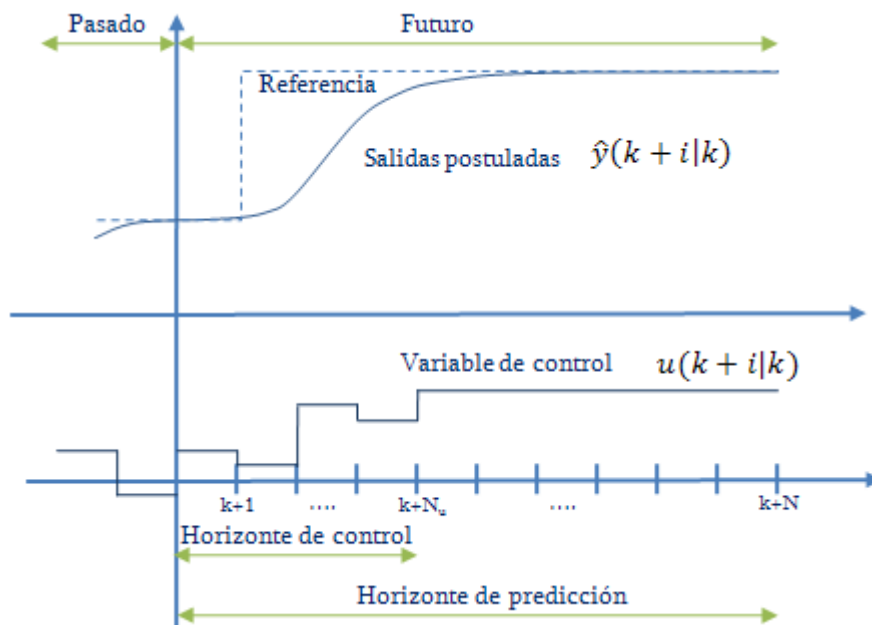


Figura 216. Estrategia del MBPC

1. En cada instante k y haciendo uso del modelo del proceso se predicen las futuras salidas para un determinado horizonte N , llamado horizonte de predicción. Estas salidas postuladas (predichas), $\hat{y}(k+i|k)$ ¹⁵ para $k=1\dots N$ dependen de los valores conocidos hasta el instante k (entradas y salidas pasadas) y de las señales de control futuras $u(k+i|k)$, $k=0\dots N_u-1$ que se pretenden mandar al sistema y que son las que se quieren calcular.
2. El conjunto de señales de control futuras se calcula optimizando un determinado criterio, en el que se pretende mantener el proceso lo más próximo posible a la trayectoria de referencia $\omega(k+i)$, que puede ser directamente la referencia o una suave aproximación de ésta. Este criterio suele tomar la forma de una función cuadrática de los errores entre la salida postulada y la trayectoria de referencia, también postulada, incluyendo en muchos casos el esfuerzo de control. Si el criterio es cuadrático, el modelo lineal y no existen restricciones se puede obtener una solución analítica, en otro caso se debe usar un método iterativo de optimización. Adicionalmente se hace alguna suposición sobre la estructura de la ley de control futura, como por ejemplo que va a ser constante a partir de cierto instante.
3. La señal de control $u(k|k)$ es enviada al proceso mientras que las siguientes señales de control calculadas son desechadas, puesto que en el siguiente instante de muestreo se conoce la salida real $y(k+1)$ y se repite el paso 1 con este nuevo valor y todas las secuencias son actualizadas. Se calcula por tanto el $u(k+1|k+1)$ que en principio será diferente al $u(k+1|k)$ al disponer de nueva información. Esta estrategia se denomina horizonte móvil.

La estructura básica de un MPC se muestra en la Figura 217, en ella se usa un modelo para predecir las salidas futuras del sistema en base a las entradas y salidas pasadas, y a las futuras acciones de control, que se obtienen durante el proceso de optimización del algoritmo, teniendo en cuenta las restricciones y la función de coste.

¹⁵ $\hat{y}(k+i|k)$: Expresa la predicción de la salida en el instante $k+i$ realizada con la información disponible en el instante k

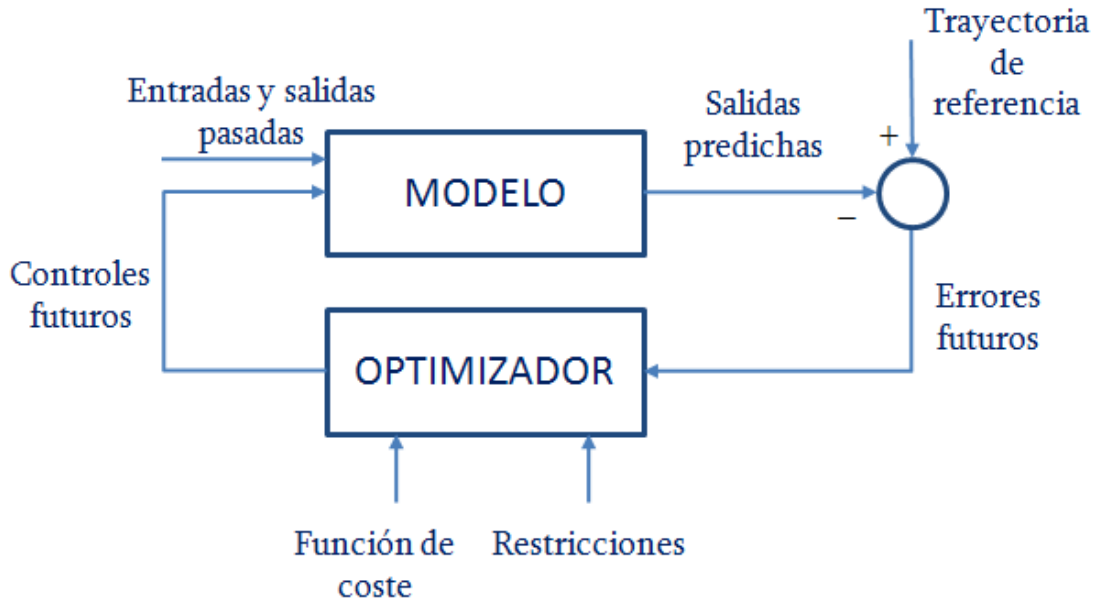


Figura 217. Estructura básica de un MPC

El optimizador es la una función que, con la información disponible hasta el instante k de las entradas y salidas: $\varphi = (u(k), u(k-1), \dots, y(k), y(k-1), \dots)$ y un modelo del proceso $G_p(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})}$, obtiene el valor de la respuesta en instantes posteriores $\hat{y}(k+i|k)$ ante una entrada determinada $u = u(k+i|k)$

$$\hat{y}(k+i|k) = f(u, \varphi, G_p(z^{-1})) \quad (\text{C.1})$$

Por lo tanto, todos los algoritmos MBPC poseen los siguientes elementos comunes:

- Modelo de predicción
- Función objetivo
- Obtención de la ley de control

C.4 Controladores predictivos GPC y CRHPC

El Control Predictivo Generalizado (GPC)¹⁶ (Clarke, Mohtadi y Tuffs 1987) es un tipo de estrategia de control predictivo basada en modelos (MBPC).

Los GPC's pueden ser aplicados a una gran variedad de procesos, desde plantas con una dinámica muy sencilla, hasta otras más complejas, incluyendo sistemas con retardo, inestables, de fase no mínima, etc. (Camacho y Bordóns 2004). Este es uno

¹⁶ GPC: *Generalized Predictive Control*

de los motivos que han llevado enseñarlo y aplicarlo en el mundo académico (Sanchís, y otros 1997).

La idea básica del GPC es calcular una secuencia de futuras acciones de control, de tal forma que minimice una función de coste multipaso. El índice a minimizar es la esperanza matemática de una función cuadrática, que evalúa por un lado la discrepancia entre la salida postulada del sistema, y una cierta trayectoria de referencia hasta el horizonte de predicción, y por otro el esfuerzo de control necesario para obtener dicha salida.

En el diseño del GPC se considera un índice de coste cuadrático, expresado habitualmente en la forma:

$$J(N_1, N_2, N_u) = E \left\{ \sum_{j=N_1}^{N_2} \psi_j \cdot [\hat{y}(k+j|k) - w(k+j)]^2 + \sum_{j=1}^{N_u} \lambda_j \cdot [\Delta u(k+j-1)]^2 \right\} \quad (C.2)$$

El primer término tiene en cuenta el error que se produce entre las salidas y las referencias en el horizonte de predicción, es decir, los errores futuros, y el segundo término penaliza el esfuerzo de control requerido. La ponderación entre estos dos objetivos se hace mediante una secuencia de pesos con el factor ψ_j , en el caso de los errores en la salida y el factor λ_j para las acciones de control.

Reescribiendo en forma matricial,

$$J = (Y - W)^T \cdot \psi \cdot (Y - W) + \lambda \cdot \Delta U^T \cdot \Delta U \quad (C.3)$$

Y minimizando analíticamente este índice respecto a ΔU , obtenemos la ley de control:

$$\Delta U = [G^T \cdot \psi \cdot G + \lambda \cdot I]^{-1} \cdot G^T \cdot \psi \cdot [W - \Gamma \cdot \Delta U^f - F \cdot Y^f] \quad (C.4)$$

y el modelo de predicción:

$$Y = G \cdot \Delta U + \Gamma \cdot \Delta U^f + F \cdot Y^f \quad (C.5)$$

donde:

- $\Delta U = [\Delta u(k), \dots, \Delta u(k + N_u - 1)]$: Son los incrementos en las acciones de control postuladas
- $W = [w(k + 1), \dots, w(k + N)]^T$: Son las referencias futuras para el horizonte de predicción
- $\Delta U^f = [\Delta u^f(k - 1), \dots, \Delta u^f(k - n_r)]^T$: Son los incrementos en las acciones de control conocidas, anteriores al instante k hasta n_r , siendo n_r el número de columnas de la matriz Γ
- $Y^f = [y^f(k), \dots, y^f(k - n_a)]^T$: Son las salidas del proceso conocidas, anteriores al instante k hasta n_a , siendo n_a el orden del proceso.
- G, F, Γ : Son matrices cuyas filas contienen los coeficientes de los polinomios calculados de forma recursiva

ψ, λ : Factores de ponderación. Son matrices diagonales que tienen los pesos ψ_j y λ_j en cada fila. Estas matrices permiten dar diferentes pesos a cada uno de los factores del índice, lo que se traduce en una mayor importancia relativa de unos sobre otros. El ajuste es empírico y por lo tanto no es sencillo.

El control predictivo restringido de horizonte deslizante (CRHPC)¹⁷ (Clarke y Scattolini 1991) permite obtener resultados sobre estabilidad y robustez para horizontes pequeños. La idea consiste básicamente en obligar a que la salida coincida con la referencia durante un cierto intervalo al final del horizonte de predicción “m”, tal y como se muestra en la Figura 218. Se emplean algunos de los grados de libertad de las señales de control futuras para forzar la salida mientras que el resto queda disponible para minimizar la función de coste en un cierto intervalo. El hecho de que la salida coincida con la referencia en un cierto intervalo garantiza la monotonicidad de la ecuación de Riccati asociada y por tanto la estabilidad.

¹⁷ CRHPC: *Constrained Receding Horizon Predictive Control*

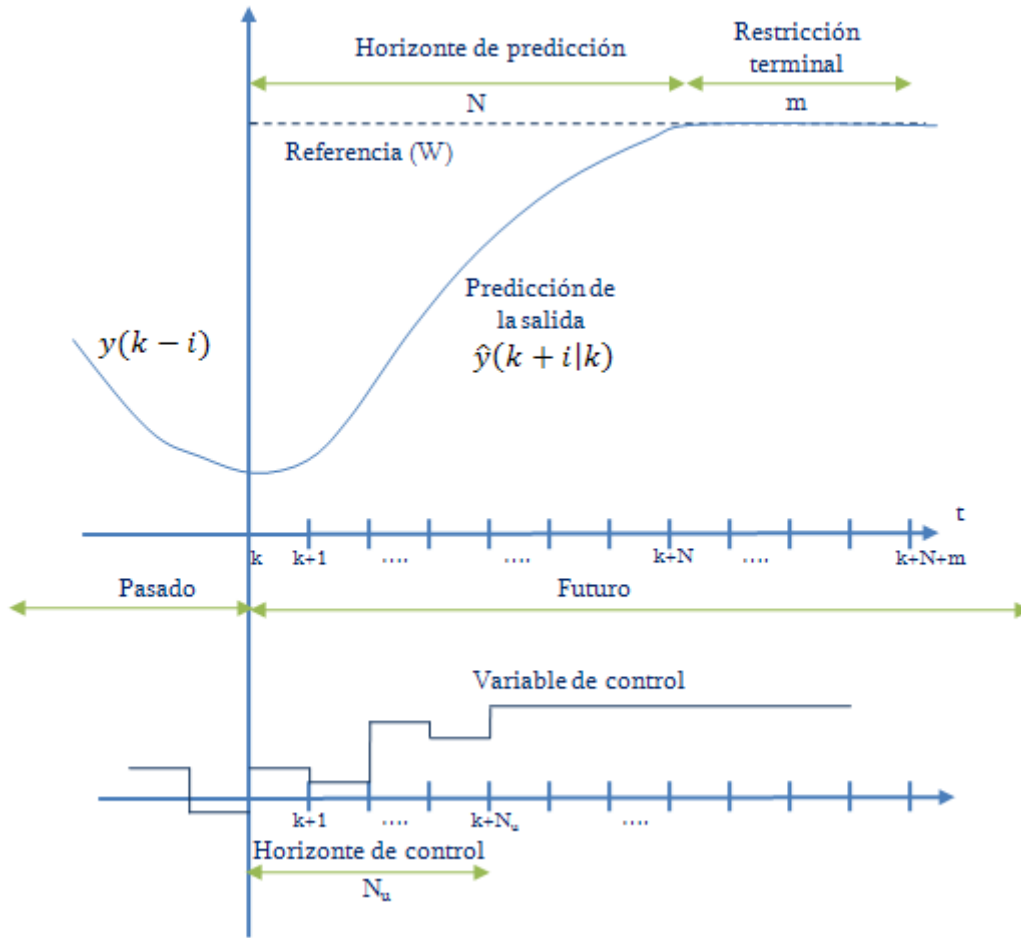


Figura 218. Constrained Receding-Horizon Predictive Control

El problema de control predictivo se plantea del mismo modo que en el GPC estándar, utilizando el mismo índice cuadrático:

$$J(N_1, N_2, N_u) = E \left\{ \sum_{j=N_1}^{N_2} \psi_j \cdot [\hat{y}(k+j|k) - w(k+j)]^2 + \sum_{j=1}^{N_u} \lambda_j \cdot [\Delta u(k+j-1)]^2 \right\} \quad (C.6)$$

pero en este caso el problema de minimización está sujeto a nuevas restricciones planteadas del siguiente modo¹⁸:

$$\hat{y}(k+N+i) = w(k+N), \quad i \in [1 \dots m] \quad (C.7)$$

¹⁸ Se asume un horizonte de predicción igual a N, por lo que N₁=1 y N₂=N

$$\Delta u(k + N_u + i) = 0, \quad i \geq 0 \quad (\text{C.8})$$

Utilizando el modelo de predicción que se usa en el GPC

$$\hat{y}(k + i|k) = G_i(z^{-1})\Delta u(k + i - 1) + f(k + i|k) \quad (\text{C.9})$$

$$f(k + i|k) = \frac{F_i(z^{-1})}{T(z^{-1})}y(k) + \frac{\Gamma_i(z^{-1})}{T(z^{-1})}\Delta u(k - 1) \quad (\text{C.10})$$

y con los polinomios G_i , Γ_i y F_i calculados recursivamente para $i=1\dots N$, se predice la salida para todo el horizonte de predicción,

$$\begin{bmatrix} \hat{y}(k + 1|k) \\ \hat{y}(k + 2|k) \\ \dots \\ \hat{y}(k + N|k) \end{bmatrix} = G \cdot \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k + 1|k) \\ \dots \\ \Delta u(k + N_u - 1|k) \end{bmatrix} + \begin{bmatrix} f(k + 1|k) \\ f(k + 2|k) \\ \dots \\ f(k + N|k) \end{bmatrix} \quad (\text{C.11})$$

o bien

$$Y_{Nx1} = G_{NxN_u} \cdot \Delta u_{N_u \times 1} + f_{Nx1} \quad (\text{C.12})$$

donde

$$G_{NxN_u} = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ g_{N-1} & g_{N-2} & \dots & g_{N-N_u} \end{bmatrix} \quad (\text{C.13})$$

y se extiende la predicción m instantes más allá de este horizonte,

$$\begin{bmatrix} \hat{y}(k + N + 1|k) \\ \hat{y}(k + N + 2|k) \\ \dots \\ \hat{y}(k + N + m|k) \end{bmatrix} = \tilde{G} \cdot \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k + 1|k) \\ \dots \\ \Delta u(k + N_u - 1|k) \end{bmatrix} + \begin{bmatrix} f(k + N + 1|k) \\ f(k + N + 2|k) \\ \dots \\ f(k + N + m|k) \end{bmatrix} \quad (\text{C.14})$$

o bien

$$\tilde{Y}_{mx1} = \tilde{G}_{mxN_u} \cdot \Delta u_{N_u \times 1} + \tilde{f}_{mx1} \quad (\text{C.15})$$

donde

$$G_{m \times N_u} = \begin{bmatrix} g_N & g_{N-1} & \cdots & g_{N+1-N_u} \\ g_{N+1} & g_N & \cdots & g_{N+2-N_u} \\ \cdots & \cdots & \cdots & \cdots \\ g_{N+m-1} & g_{N+m-2} & \cdots & g_{N+m-1-N_u} \end{bmatrix} \quad (C.16)$$

De esta forma, el problema de control predictivo se reduce a

$$\min_{\Delta u} J(\Delta u) = (G\Delta u + f - w)^T \cdot \psi \cdot (G\Delta u + f - w) + \Delta u^T \cdot \lambda \cdot \Delta u \quad (C.17)$$

$$\text{Sujeto a: } \tilde{G}\Delta u + \tilde{f} = \tilde{w} \quad (C.18)$$

donde

$$\tilde{w} = [w(k+N|k) \quad w(k+N|k) \quad \dots \quad w(k+N|k)] \quad (C.19)$$

Una solución analítica a este problema de minimización puede ser calculada utilizando multiplicadores de Lagrange, produciendo el siguiente resultado:

$$\Delta U = \bar{G} \cdot [I - \tilde{G}^T \cdot M \cdot \tilde{G} \cdot \bar{G}] \cdot G^T \cdot \psi \cdot (W - f) + \bar{G} \cdot \tilde{G}^T \cdot M \cdot (\tilde{W} - \tilde{f}) \quad (C.20)$$

donde:

$$f = \Gamma \cdot \begin{bmatrix} \Delta u(k-1) \\ \dots \\ \Delta u(k-n_t) \end{bmatrix} + F \cdot \begin{bmatrix} y(k) \\ \dots \\ y(k-n_a) \end{bmatrix} \quad (C.21)$$

$$\tilde{f} = \tilde{\Gamma} \cdot \begin{bmatrix} \Delta u(k-1) \\ \dots \\ \Delta u(k-n_t) \end{bmatrix} + \tilde{F} \cdot \begin{bmatrix} y(k) \\ \dots \\ y(k-n_a) \end{bmatrix} \quad (C.22)$$

$$M_{m \times m} = (\tilde{G} \cdot \bar{G} \cdot \tilde{G}^T)^{-1} \quad (C.23)$$

$$\bar{G}_{N_u \times N_u} = (G^T \cdot \psi \cdot G + \lambda)^{-1} \quad (C.24)$$

Al ser una estrategia deslizante, sólo el primer elemento del vector $\Delta U(k)$ es utilizado, desechándose los restantes y procediendo a un nuevo cálculo en el siguiente instante de muestreo. Este método proporciona una solución analítica que, como se observa en la ecuación anterior, resulta más compleja que la proporcionada por el GPC. La carga de cálculo puede resultar considerable pues se realizan diversas operaciones con matrices incluyendo inversión, aunque los cálculos se pueden optimizar sabiendo que G es triangular y las matrices a invertir son simétricas. Este factor puede ser determinante en el caso adaptativo donde todos los vectores y matrices pueden cambiar constantemente.

El método es capaz de estabilizar cualquier tipo de proceso como sistemas inestables o de fase no mínima. Al igual que el GPC, el uso de entradas y salidas filtradas puede resultar en un control de asignación de polos.

Dichas propiedades se basan en la idea de que, si existe una solución y el horizonte de predicción es suficientemente largo como para cubrir el régimen transitorio de la salida, la función de coste decrece monótonamente (siempre que no existan perturbaciones y el modelo sea igual al proceso) y puede ser interpretada como una función de Lyapunov para garantizar estabilidad.

En concreto, se puede demostrar que el bucle cerrado es asintóticamente estable si (Scolaert y Clarke 1993), (Yoon y Clarke 1995):

1. $\psi \geq 0, \lambda \geq 0$
2. $m = n_a$
3. $N_u \geq 0$
4. $N \geq N_u + \max(n_b, n_a) - n_a$

donde $n_a = \deg(A(z^{-1})\Delta)$ y $n_b = \deg(B(z^{-1}))$, y asumiendo que $(A(z^{-1})\Delta)$ y $(B(z^{-1}))$ son primos entre sí, es decir, no poseen factores comunes.

C.5 Controladores predictivos con restricciones

Una formulación más general del problema es cuando se introducen restricciones en el control predictivo. El MPC es la única metodología capaz de incorporar las restricciones de forma sistemática en la fase de diseño del controlador.

En la práctica todos los procesos están sujetos a restricciones, ya que los actuadores tienen un campo limitado de actuación (límites físicos), existen límites de seguridad (por ejemplo para presiones y temperaturas máximas), requerimientos tecnológicos que hay que mantener en las variables y limitaciones de calidad del producto o normativa medioambiental.

En la práctica es habitual usar el GPC estándar para calcular la señal $u(k)$. Si ésta viola las restricciones entonces se satura a sus límites. El caso de que las señales de control futuras sobrepasen las restricciones (bien en amplitud o de velocidad de cambio) no es casi nunca considerado, pues en la mayoría de casos ni siquiera se calculan. Lógicamente esta forma de operar no garantiza el óptimo, ya que se puede trabajar en un punto de operación con mayor valor de la función objetivo y por tanto se puede reducir las prestaciones. En cualquier caso, las variables manipuladas siempre pueden mantenerse dentro de ciertos límites por medio del programa de control o por los actuadores.

No ocurre así con las variables controladas, cuya violación de las restricciones puede ocasionar daños físicos en los equipos y pérdidas en la producción. Estas violaciones no pueden resolverse saturando las acciones de control, sino que deben ser tratadas aprovechando la capacidad de predicción del MPC. El sistema de control debe anticiparse al incumplimiento de las restricciones y corregirlas de forma adecuada.

Aparte de las restricciones en la amplitud y velocidad de cambio de las señales de control y de amplitud en las salidas, normalmente impuestas por razones de seguridad, también se puede forzar a que la respuesta del proceso posea ciertas características como mantener la salida dentro de una banda, evitar sobreoscilaciones o comportamiento de fase no mínima, obligar a una respuesta monótona, tener en cuenta las no linealidades de los actuadores o imponer restricciones de igualdad al estado final (usado en el CRHPC).

En este caso, la inclusión de restricciones lleva a una solución que no puede ser descrita mediante una ecuación diferencial de forma explícita, si no que se tiene un problema en cuya resolución debe utilizarse programación cuadrática para el cálculo de la acción de control, y además, este cálculo debe hacerse en cada instante de muestreo.

Modelo	Función de Coste	Restricciones	Solución
Lineal	Cuadrática	Ninguna	Explícita
Lineal	Cuadrática	Lineales	Programación Cuadrática (QP)
Lineal	Norma-1	Lineales	Programación Lineal (LP)

Tabla 9. Algoritmos de resolución de MPC lineales

Por lo tanto, en el caso que nos ocupa, nos encontramos con un problema de minimización de una función de coste cuadrática con restricciones lineales, es decir, un problema de programación cuadrática (QP). Se trata de un algoritmo de optimización muy estudiado, hay que utilizar métodos numéricos mediante un proceso de iteración para alcanzar el mínimo.

Esta característica hace que sea necesaria la utilización de procesadores con una amplia capacidad de cálculo para poder llegar a la solución en cada periodo de muestreo, y por lo tanto la estructura de un sistema distribuido formado por un nodo supervisor (de amplia capacidad computacional) junto con el control local es la más adecuada. Además, al no utilizar la estrategia del horizonte deslizante si no que se utiliza toda la información de las secuencias de las acciones de control y salidas futuras el cálculo se multiplica.

En algunos casos se ha conseguido realizar la implementación de controles predictivos con restricciones mediante el cálculo explícito del MPC, cuando hay que utilizar programación cuadrática en un sistema empotrado, mediante una partición del espacio de estados (Bemporad, Borelli y Morari 2002), (Bemporad, y otros 2002).

En cualquier caso, independientemente del uso de restricciones en el cálculo de las salidas y acciones de control futuras, la consecuencia siempre es el envío por parte del control supervisor de las secuencias futuras correspondientes a los controles locales.

Anexo D. Método de interpolación basado en spline cúbica natural

Una curva denominada Spline viene determinada por el siguiente polinomio cúbico:

$$q_d(t) = a + bt + ct^2 + dt^3 \quad (\text{D.1})$$

donde t está normalizada a $0 \leq t \leq 1$.

Con el objeto de realizar una interpolación haciendo uso de esta técnica, se partirá del conocimiento de los puntos a partir de los cuales se quiere realizar la interpolación así como de sus derivadas (o bien, se podría realizar su cálculo a partir de la media ponderada de cada punto, el anterior y el siguiente). Para obtener los puntos intermedios buscados se deben hallar los valores de los coeficientes a_i , b_i , c_i , d_i , coeficientes de los polinomios cúbicos, para cada una de las componentes que definen los puntos, y en cada uno de los subintervalos (entre 2 puntos consecutivos) así, por ejemplo, en 2D, si se quiere interpolar 3 puntos, se deberán hallar los coeficientes de 4 polinomios cúbicos.

Así, de manera algorítmica, para cada una de las componentes, se deberán realizar las siguientes operaciones (suponiendo que se trata de la componente y del punto):

La coordenada y del punto inicial del segmento, es decir, para $t=0$, cumplirá:

$$Y_i(0) = y_i = a_i$$

La coordenada del punto final, es decir, para $t=1$, cumplirá:

$$Y_i(1) = y_{i+1} = a_i + b_i + c_i + d_i$$

Y las derivadas de $Y(t)$ en los puntos inicial y final, se consiguieron las dos ecuaciones que faltan:

$$Y_i'(0) = D_i = b_i$$

$$Y_i' = D_{i+1} = b_i + 2c_i + 3d_i$$

que, resolviéndolo, obtendríamos:

$$a_i = y_i$$

$$b_i = D_i$$

$$c_i = 3(y_{i+1} - y_i) - 2D_i - D_{i+1}$$

$$d_i = 2(y_i - y_{i+1}) + D_i + D_{i+1}$$

Si se elige como valor de las derivadas D_i y D_{i+1} , $D_i = D_{i+1} = 0$, la expresión anterior quedará como:

$$a_i = y_i$$

$$b_i = 0$$

$$c_i = 3(y_{i+1} - y_i)$$

$$d_i = 2(y_i - y_{i+1})$$

siendo estas expresiones fáciles de implementar en un computador.