



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

**FACULTAD DE INGENIERÍA**

**“Controlador Virtual Determinístico  
por Modelo Interno”**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE:**

**INGENIERO ELÉCTRICO Y ELECTRÓNICO**

**P R E S E N T A N:**

**ROGELIO GABRIEL GUADARRAMA MENDOZA**

**MIGUEL ARMANDO MONDRAGÓN DE LA PEÑA**



**DIRECTOR DE TESIS:  
M. I. RICARDO GARIBAY JIMÉNEZ**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*a mamá y papá*

*RGGM*

*Agradezco a Dios a mi familia y amigos por estar siempre conmigo.*

*A mis padres y a mi hermana por el gran apoyo que siempre me han dado a lo largo de mi carrera además de su gran cariño y amor.*

*Al M.I. Ricardo Garibay Jiménez, por su paciencia, apoyo y enseñanzas durante el desarrollo de esta tesis.*

*Gracias a la Universidad Nacional Autónoma de México por los conocimientos que he adquirido y por los buenos momentos que he vivido por estudiar en sus instalaciones.*

*Miguel Armando Mondragón de la Peña*

# Índice general

<b>Introducción</b>	<b>XIII</b>
<b>1. Control por Modelo Interno de Un Grado de Libertad</b>	<b>1</b>
1.1. Propiedades del Control por Modelo Interno de Un Grado de Libertad . . . . .	2
1.1.1. Descripción del Esquema de Control . . . . .	2
1.1.2. Comportamiento en el Estado Estable . . . . .	3
1.1.3. Propiedad de Control Ideal . . . . .	4
1.1.4. El Controlador CMI $q_r(s)$ . . . . .	5
1.2. Diseño del Controlador CMI $q_r(s)$ para Procesos con Modelo Exacto . . . . .	6
1.2.1. Diseño del Controlador CMI para Procesos de Primer Orden con un Tiempo de Retardo . . . . .	6
1.2.2. Diseño del Controlador CMI para Procesos sin Ceros Próximos al Eje Imaginario o Sobre el Semiplano Derecho del Plano $S$ . . . . .	8
1.2.3. Diseño del Controlador CMI para Procesos con Ceros Cerca del Eje Imaginario . . . . .	8
1.2.4. Diseño del Controlador CMI para Procesos con Ceros en el Semiplano Derecho . . . . .	10
1.3. Descripción de Incertidumbre en el Proceso: Incertidumbre Paramétrica . . . . .	12
1.4. Ajuste $M_P$ para Procesos con Incertidumbre . . . . .	13
1.4.1. Relación entre las Características en el Tiempo y en la Frecuencia de un Sistema de Segundo Orden . . . . .	13
1.4.2. Ajuste $M_P$ . . . . .	15
1.4.3. Existencia de la Solución en el Ajuste $M_P$ . . . . .	19
1.5. Síntesis $M_P$ para Procesos con Incertidumbre . . . . .	21
<b>2. Control por Modelo Interno de Dos Grados de Libertad</b>	<b>27</b>
2.1. Justificación del Control por Modelo Interno de Dos Grados de Libertad . . . . .	27
2.2. Esquema de Control por Modelo Interno de Dos Grados de Libertad . . . . .	29
2.2.1. Diseño de los Controladores $q_r(s)$ y $qq_d(s)$ para Procesos con Modelo Exacto . . . . .	30

2.3.	Ajuste del Control por Modelo Interno de Dos Grados de Libertad	34
2.3.1.	Ajuste $M_P$ para Procesos Estables y/o Sobreamortiguados con Incertidumbre . . . . .	35
2.3.2.	Ajuste $M_P$ para Procesos Estables y Subamortiguados con Incertidumbre . . . . .	42
2.4.	Síntesis del Control por Modelo Interno de Dos Grados de Libertad	46
2.4.1.	Síntesis $M_P$ para Sistemas Estables y/o Sobreamortiguados con Incertidumbre . . . . .	46
2.4.2.	Síntesis $M_P$ para Sistemas Estables y Subamortiguados con Incertidumbre . . . . .	49
<b>3.</b>	<b>Implantación de Sistemas de Control en Tiempo Real con LabVIEW Real-Time 7.1 y la tarjeta PCI-7041/6040E</b>	<b>53</b>
3.1.	Sistemas de Tiempo Real . . . . .	53
3.1.1.	Clasificación de Sistemas de Tiempo Real . . . . .	54
3.1.2.	Características de los Sistemas de Tiempo Real . . . . .	55
3.2.	¿Cómo Implantar un Sistema de Tiempo Real? . . . . .	55
3.2.1.	Los Sistemas de Tiempo Real Basados en Computadora . . . . .	56
3.2.2.	Descripción de un Sistema de Control de Lazo Cerrado en Tiempo Real Basado en Computadora . . . . .	57
3.3.	El Lenguaje Gráfico de Programación LabVIEW . . . . .	63
3.4.	Descripción de la Plataforma de Desarrollo LabVIEW Real-Time 7.1 y PCI 7041/6040E . . . . .	65
3.5.	Construcción Típica de un Sistema de Control en Tiempo Real con LabVIEW Real-Time 7.1 y PCI-7041/6040E . . . . .	68
3.5.1.	Conexión de Señales Analógicas a la Tarjeta PCI-7041/6040E	68
3.5.2.	Buenas Técnicas de Programación de una Aplicación en Tiempo Real . . . . .	72
3.6.	Herramientas de Tiempo Real del LabVIEW Real-Time Module 7.1 . . . . .	74
3.6.1.	Estructura <i>Timed Loop</i> . . . . .	74
3.6.2.	Comunicación entre Tareas con Real-Time FIFOs . . . . .	76
3.6.3.	Comunicación entre la Tarjeta PCI y la Máquina Anfitrión con Memoria Compartida . . . . .	77
3.7.	LabVIEW Execution Trace Toolkit 7.1 . . . . .	78
<b>4.</b>	<b>Diseño e Implantación del Controlador Virtual Determinístico por Modelo Interno</b>	<b>81</b>
4.1.	Simulador de Procesos PCS327 . . . . .	81
4.2.	Ajuste del Controlador Virtual Determinístico Por Modelo Interno	82
4.2.1.	Modelo e Incertidumbre del Proceso a Controlar . . . . .	83
4.2.2.	Ajuste de Un Grado de Libertad . . . . .	83
4.2.3.	Ajuste de Dos Grados de Libertad . . . . .	90
4.3.	Implantación del Controlador Virtual Determinístico por Modelo Interno . . . . .	96
4.3.1.	Programación de la Tarjeta de Tiempo Real . . . . .	98

4.3.2. Programación de la Máquina Anfitrión . . . . .	108
4.4. Operación del Controlador Virtual Determinístico por Modelo Interno . . . . .	113
4.4.1. Desempeño en el Tiempo del Controlador Virtual Deter- minístico por Modelo Interno . . . . .	113
4.5. Conclusiones . . . . .	117
<b>A. Bloque de Conexiones BNC-2120</b>	<b>121</b>
<b>Bibliografía</b>	<b>125</b>

# Índice de figuras

1.1. Esquema de Control por Modelo Interno de Un Grado de Libertad (CMI 1GL). . . . .	2
1.2. Respuesta a una entrada tipo escalón en la referencia del sistema CMI 1GL para el SPOTR. . . . .	7
1.3. Curva de ganancia de $q_r(s, \varepsilon_r) = \frac{(s+2)^4}{(s^2+0.004s+16)(\varepsilon s+1)^2}$ con $\varepsilon_r = 0.22$ . . . . .	9
1.4. Respuesta a escalón del sistema en lazo cerrado del ejemplo 1, $p(s)q(s) = \frac{s^2+0.004s+16}{(s^2+(2\zeta*4)s+16)(0.22s+1)^2}$ con $\zeta = 0.35$ . . . . .	10
1.5. Respuesta del sistema con un cero no invertible del ejemplo 2 a una entrada escalón unitario en la referencia. . . . .	12
1.6. Respuesta a una entrada escalón del sistema $1/(s^2 + s + 1)$ . . . . .	14
1.7. Traza de magnitud de $1/(1 - \omega^2 + j\omega)$ . . . . .	15
1.8. Sobrepasso y factor de amortiguamiento contra $M_P$ para $1/(s^2 + 2\zeta s + 1)$ . . . . .	16
1.9. Curvas de ganancia máxima y ganancia mínima de la función complementaria de sensibilidad $CS$ . . . . .	18
1.10. Respuestas más rápida, más lenta, y con mayor sobrepasso de la variable de salida $y(t)$ a una entrada tipo escalón en la referencia $r(t)$ , para el ejemplo 4. . . . .	19
1.11. Esquema de Controlabilidad Integral. . . . .	20
1.12. Comparación de las curvas de máxima y mínima magnitud de la función de sensibilidad complementaria, $CS$ , para los sistemas con controladores CMI descritos por las ecuaciones (1.39) y (1.40). . . . .	23
1.13. Respuestas del sistema CMI más rápidas ( $K = 20, T = 10$ ) y más lentas ( $K = 10, \tau = 10$ ) a una entrada escalón, para los controladores descritos por las ecuaciones (1.39) y (1.40). . . . .	23
1.14. Comparación de las curvas de máxima y mínima magnitud de la función de sensibilidad complementaria, $CS$ , para los sistemas con modelos y controladores CMI descritos por las ecuaciones 1.38 y 1.40, y 1.41 y 1.42. . . . .	24
1.15. Comparación de las respuestas para el proceso con $K = 15$ y $T = 15$ , de los sistemas descritos por $\tilde{p}(s) = \frac{15}{(20s+1)}e^{-15s}$ y $q_r(s) = \frac{(20s+1)}{15(35.98s+1)}$ , y las expresiones (1.43) y (1.44). . . . .	26



2.1. Respuesta del sistema CMI 1GL a una entrada tipo escalón en la perturbación, $d(t)$ . . . . .	28
2.2. Esquema de Control por Modelo Interno de Dos Grados de Libertad (CMI 2GL). . . . .	29
2.3. Comparación de las respuestas a la perturbación para los sistemas CMI 1GL y CMI 2GL. . . . .	33
2.4. Respuestas del sistema CMI 1GL, y de los sistemas CMI 2GL descritos por las ecuaciones (2.17) y (2.19). . . . .	34
2.5. Respuestas de los sistemas CMI 1GL y CMI 2GL, para el proceso $p_d(s) = p(s) = \frac{e^{-s}}{s^2+0.2s+1}$ . . . . .	35
2.6. Curva de magnitud máxima de la función de sensibilidad parcial cuando $qq_d(s) = \frac{(5s+1)(4.70s+1)}{5(3s+1)^2}$ , ejemplo 6. . . . .	38
2.7. Respuestas del sistema de control a una entrada escalón en la perturbación, cuando $qq_d(s) = \frac{(5s+1)(4.70s+1)}{5(3s+1)^2}$ , ejemplo 6. . . . .	38
2.8. Respuesta a la referencia, con $\varepsilon_r = 5.17$ , ejemplo 6. . . . .	39
2.9. Magnitud máxima de la función de sensibilidad parcial, cuando $p_d(s) = 1$ , ejemplo 6. . . . .	39
2.10. Curva de máxima magnitud de la función de sensibilidad parcial, cuando $qq_d(s) = \frac{(5s+1)(9.58s+1)}{5(3.64s+1)^2}$ , ejemplo 7. . . . .	40
2.11. Respuesta del sistema a una entrada tipo escalón en la perturbación, cuando $qq_d(s) = \frac{(5s+1)(9.58s+1)}{5(3.64s+1)^2}$ , ejemplo 7. . . . .	41
2.12. Respuesta a la referencia, con $\varepsilon = 3.64$ y $\varepsilon_r = 5.17$ , ejemplo 7. . . . .	41
2.13. Magnitud máxima de la curva de sensibilidad parcial, cuando $p_d(s) = 1$ y $\varepsilon = 3.64$ , ejemplo 7. . . . .	42
2.14. Magnitud máxima de la función de sensibilidad para el sistema descrito por $\varepsilon = 0.6$ , ejemplo 8. . . . .	43
2.15. Respuesta del sistema descrito por $\varepsilon = 0.6$ a una entrada escalón en la perturbación, ejemplo 8. . . . .	44
2.16. Magnitud máxima de la función de sensibilidad para el sistema descrito por $\varepsilon = 0.7$ , ejemplo 8. . . . .	44
2.17. Respuesta del sistema descrito por $\varepsilon = 0.7$ a una entrada escalón en la perturbación, ejemplo 8. . . . .	45
2.18. Magnitud máxima de la función de sensibilidad para el sistema descrito por $\varepsilon = 0.55$ , ejemplo 8. . . . .	45
2.19. Respuesta del sistema descrito por $\varepsilon = 0.55$ a una entrada escalón en la perturbación, ejemplo 8. . . . .	46
2.20. Respuestas más lentas (con $K, T = 10$ ) de los sistemas de control de la tabla 2.1, cuando se aplica una entrada escalón en la perturbación. . . . .	48
2.21. Curva mínima de la función de sensibilidad parcial de los mejores diseños de 1GL y 2GL (primer y último sistema de la tabla 2.1). . . . .	48
2.22. Respuestas más lentas a entrada escalón en la referencia de algunos controladores del ejemplo 9 ( $K, T = 10$ ). . . . .	49

2.23. Comparación de la curva máxima de la función de sensibilidad del sistema con ganancia media ( $K = 1$ ) y del sistema óptimo ( $K = 1.3$ ). . . . .	50
2.24. Comparación de las respuestas de lazo cerrado en el tiempo a entradas escalón en la perturbación, para una ganancia del proceso igual a 0.4, para los modelos de rango medio y óptimo. . . . .	51
2.25. Comparación de las respuestas de lazo cerrado en el tiempo a entradas escalón en la perturbación, para una ganancia del proceso igual a 1.6, para los modelos de rango medio y óptimo. . . . .	51
3.1. Diagrama de un sistema de control en tiempo real basado en una computadora. . . . .	57
3.2. Etapas de acondicionamiento de señal para un sistema de computadora. . . . .	59
3.3. Estrategia Round Robin. . . . .	61
3.4. Estrategia por prioridades. El número indica la prioridad del proceso, con 1 como la prioridad máxima. . . . .	61
3.5. Ejemplo de un convertidor de grados Celsius a grados Fahrenheit, programado en LabVIEW 7.1. . . . .	64
3.6. Arquitectura de la plataforma de desarrollo de sistemas en tiempo real con LabVIEW. . . . .	66
3.7. Tarjeta de tiempo real PCI-7041/6040E. . . . .	66
3.8. Fuentes de voltaje a) flotante y b) referida a tierra. . . . .	68
3.9. Sistema de medición diferencial de 8 entradas. . . . .	69
3.10. Sistema de medición de hasta 16 canales, en modo común referido. . . . .	70
3.11. Sistema de medición de hasta 16 canales en modo común no referido. . . . .	71
3.12. Resumen de conexiones de entrada analógica. . . . .	71
3.13. Conexiones para salida analógica. . . . .	72
3.14. Esquema de la programación de un sistema de control en tiempo real en LabVIEW Real-Time. . . . .	73
3.15. Estructura <i>Timed Loop</i> . . . . .	75
3.16. Ventana de diálogo del nodo de entrada del <i>Timed Loop</i> . . . . .	76
3.17. Ejemplo de uso del Real-Time FIFO. . . . .	77
3.18. Algunos VIs que operan en la memoria compartida, en el modo de lectura y escritura simple. . . . .	78
3.19. LabVIEW Execution Trace Tool. . . . .	79
3.20. Utilización típica de los bloques <i>TraceTool Start Trace VI</i> y <i>TraceTool Stop and Send VI</i> , del <i>Execution Trace Tool VIs</i> . . . . .	79
3.21. Aspecto de una sesión con el LabVIEW Execution Trace Toolkit 7.1, de una aplicación en tiempo real. . . . .	80
4.1. Simulador de procesos PCS327. . . . .	82
4.2. Magnitud máxima y mínima de la función de sensibilidad complementaria, para el primer diseño de Un Grado de Libertad. . . . .	84
4.3. Respuesta a una entrada en la referencia, del sistema correspondiente al primer diseño de Un Grado de Libertad. . . . .	85

4.4. Respuesta a una entrada en la perturbación, del sistema correspondiente al primer diseño de Un Grado de Libertad. . . . .	85
4.5. Máxima y mínima magnitud de $CS$ , para los diseños primero y segundo de Un Grado de Libertad. . . . .	87
4.6. Respuesta a una entrada en la referencia, del sistema correspondiente al segundo diseño de Un Grado de Libertad. . . . .	88
4.7. Respuesta a una entrada en la perturbación, del sistema correspondiente al segundo diseño de Un Grado de Libertad. . . . .	88
4.8. Máxima y mínima magnitud de $CS$ , para los diseños segundo y tercero de Un Grado de Libertad. . . . .	89
4.9. Respuesta a una entrada en la referencia, del sistema correspondiente al tercer diseño de Un Grado de Libertad. . . . .	90
4.10. Respuesta a una entrada en la perturbación, del sistema correspondiente al segundo diseño de Un Grado de Libertad. . . . .	91
4.11. Mínima magnitud de $PS$ de los sistemas de Un Grado de Libertad, expresado por las ecuaciones (4.8) y (4.10), y de Dos Grados de Libertad, representado por las ecuaciones (4.11) y (4.12). . . . .	92
4.12. Respuesta a la perturbación de los sistemas de Un Grado de Libertad, expresado por las ecuaciones (4.8) y (4.10), y de Dos Grados de Libertad, representado por las ecuaciones (4.11) y (4.12). . . . .	93
4.13. Respuesta a la referencia de los sistemas de Un Grado de Libertad, expresado por las ecuaciones (4.8) y (4.10), y de Dos Grados de Libertad, representado por las ecuaciones (4.11), (4.12) y (4.13). . . . .	93
4.14. Mínima magnitud de $PS$ de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones (4.11) y (4.12), (4.14) y (4.15). . . . .	94
4.15. Respuesta a la perturbación de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones 4.11 y 4.12, y 4.14 y 4.15. . . . .	95
4.16. Respuesta a la referencia de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones (4.11), (4.12) y (4.13), y (4.14), (4.15) y (4.16). . . . .	95
4.17. Mínima magnitud de $PS$ de los sistemas de Dos Grado de Libertad, expresados por las ecuaciones (4.11) y (4.12), y (4.17) y (4.18). . . . .	97
4.18. Respuesta a la perturbación de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones (4.11) y (4.12), y (4.17) y (4.18). . . . .	97
4.19. Respuesta a la referencia de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones (4.11), (4.12) y (4.13), y (4.17), (4.18) y (4.19). . . . .	98
4.20. Configuración del canal para: (1) el muestreo de la variable del proceso, y (2) la escritura del esfuerzo de control. . . . .	100
4.21. Diagrama de tiempo del Sistema Controlador Virtual Determinístico por Modelo Interno. . . . .	101
4.22. Conexión de bloques, para asegurar un comportamiento determinístico del lazo de control, de acuerdo a la frecuencia de muestreo de la variable del proceso. . . . .	102

4.23. Implantación en el lazo de control de esquema CMI 1GL. . . . . 103

4.24. Implantación en el lazo de control de esquema CMI 2GL. . . . . 103

4.25. Programación del algoritmo CMI 1GL. . . . . 105

4.26. Programación del algoritmo CMI 2GL. . . . . 106

4.27. Paro del proceso. . . . . 107

4.28. Ciclo de comunicación. . . . . 108

4.29. Comunicación vía FIFOs entre los ciclos de comunicación y control. 109

4.30. Panel de la interfaz de usuario. . . . . 111

4.31. Parte del diagrama de bloques de la interfaz de usuario, encargada de la lectura y escritura en memoria compartida. . . . . 112

4.32. Almacenamiento de datos en un archivo. . . . . 112

4.33. Detalle del diagrama de bloques de la interfaz, encargada de la generación de reportes. . . . . 112

4.34. Subrutina *Genera Reporte VI*. . . . . 114

4.35. Respuesta del CVDMI a un cambio escalón en la referencia, bajo la estrategia CMI 1GL, para el proceso definido por  $K = 2$  y  $T = 1[seg]$ . . . . . 115

4.36. Respuesta del CVDMI a una perturbación tipo escalón, actuante a la salida del proceso,  $p(s)$ , bajo la estrategia CMI 1GL, para los valores de parámetros  $K = 2$  y  $T = 1[seg]$ . . . . . 115

4.37. Respuesta del CVDMI a un cambio escalón en la referencia, bajo la estrategia CMI 2GL, para el proceso definido por  $K = 0.5$  y  $T = 0.01[seg]$ . . . . . 116

4.38. Respuesta del CVDMI a una perturbación tipo escalón, actuante a la entrada del proceso,  $p(s)$ , bajo la estrategia CMI 2GL, para los valores de parámetros  $K = 0.5$  y  $T = 0.01[seg]$ . . . . . 116

4.39. Adición de los bloques (1) *TraceTool Start Trace VI* y (2) *TraceTool Stop Trace and Send VI*, al diagrama de bloques de los lazos de control y comunicación. . . . . 117

4.40. LabVIEW Execution Trace Tool con la sesión correspondiente al sistema de Un Grado de Libertad. . . . . 118

4.41. LabVIEW Execution Trace Tool con la sesión correspondiente al sistema de Dos Grados de Libertad. . . . . 118

A.1. Bloque conector BNC-2120. . . . . 122

A.2. Resumen de opciones de configuración del BNC-2120. . . . . 123



# Introducción

En la vida diaria es necesario que una tarea específica, o la operación de una máquina, o un determinado proceso, se desarrolle de acuerdo con un cierto desempeño deseado. Pensemos, por ejemplo, en un automóvil. Dicho vehículo está diseñado para reaccionar a los comandos que el conductor decida realizar: ya sea a través del volante, para definir la dirección y sentido a tomar; o a través de los pedales de aceleración y de freno, para decidir la rapidez en el movimiento del mismo. Igualmente deberá proveer una operación segura y estable, a fin de reducir el riesgo de ocurrencia de un accidente y proteger así al conductor, a otros conductores y al peatón. En suma, el conductor debe poseer el control absoluto en la operación del vehículo, estableciendo siempre la velocidad<sup>1</sup> por él deseada, con el claro propósito de llegar a su destino sin haber sufrido percance alguno. Por lo tanto el control, como en éste y otros tantos casos, juega un papel primordial.

Controlar es, en términos generales, lograr que cierta actividad se realice como uno quiere. *Un sistema de control es la conjunción de métodos, técnicas, herramientas y equipo, orientados en el cumplimiento de este objetivo.* Los sistemas de control son responsables en gran medida de nuestro desarrollo tecnológico y grado de bienestar actual, y seguramente lo seguirán siendo en el futuro. Prácticamente en cualquier sistema, equipo, o máquina en la que podamos pensar está involucrado un sistema de control: desde el sistema de aire acondicionado que regula la temperatura de un cuarto, hasta un cohete espacial, en donde es fundamental controlar a cada instante su trayectoria, posición y velocidad.

Anteriormente mencionamos que un sistema de control no está solamente compuesto por equipo (de tipo mecánico, eléctrico, hidráulico o neumático, entre otros) e instrumentos, como el necesario en la medición de variables (*sensores*), sean éstas sujetas a control o no, o en la ejecución de las acciones de control (*actuadores*), entre otros. También se requieren métodos, técnicas, y algoritmos de control, en los cuales se diga cómo habrán de calcularse las acciones de control pertinentes, para llevar a un valor deseado las variables bajo control del proceso.

---

<sup>1</sup>Conocida también, en el control de procesos, como la variable bajo control.

Con ello, un *microprocesador* o cualquier otro circuito electrónico analógico y/o digital, por citar sólo una posibilidad, podrá realizar el cálculo de dichas acciones de control: a esta parte del sistema se le conoce popularmente como *controlador* o *compensador*.

En general, podemos diferenciar estos métodos o estrategias de control en dos categorías:

- *métodos de control de lazo abierto*, y
- *métodos de control de lazo cerrado*.

La diferencia básica entre los métodos de ambas categorías es la necesidad de los segundos de conocer el estado actual del proceso bajo control, a fin de poder calcular la acción de control a tomar; o dicho de otra manera, *en el control de lazo cerrado debemos realimentar el estado presente del proceso*. Para la clasificación de los sistemas de control suele usarse también este criterio, definiéndose así los *sistemas de control de lazo abierto*, y los *sistemas de control de lazo cerrado*.

Dentro del control realimentado clásico, es muy popular la estrategia de control proporcional-integral-derivativa, o simplemente PID. En ella se compara la salida actual del proceso con una *referencia* dada, que a su vez representa el estado de operación deseado del mismo. De la sustracción de ambas señales, se obtendrá una *señal de error*, y la *acción de control* se define después de modificar esta señal de error al multiplicarla una ganancia proporcional, integrarla y derivarla. Dependiendo del peso relativo dado a cada una de estas tres acciones, se logrará un cierto desempeño en el tiempo del proceso bajo control, cuando el sistema de control actúa sobre él. *En general, siempre será posible diseñar un sistema de control bajo la estrategia PID, si se cuenta con un modelo matemático que aproxime relativamente bien al proceso controlado*. En la mayoría de los casos, como modelos se utilizan expresiones matemáticas lineales, que resultan una buena aproximación de las relaciones entre una o más variables de salida<sup>2</sup> del proceso controlado, con una o más variables de entrada<sup>3</sup>. Dichas expresiones pueden tomar la forma de ecuaciones diferenciales, funciones de transferencia o representaciones en variables de estado.

Sin embargo, el diseño clásico de controladores PID (por el lugar geométrico de las raíces o el diseño en frecuencia) no considera la posible existencia de incertidumbre en el modelo del proceso bajo control<sup>4</sup>. Por lo tanto, el sistema PID no logrará cumplir con cierto desempeño deseable del proceso controlado.

<sup>2</sup>Algunas de las cuales serán las variables a controlar, y que pueden representar cantidades físicas reales como son la temperatura, la presión, la velocidad, etcétera.

<sup>3</sup>Algunas de las cuales corresponderán a las acciones de control.

<sup>4</sup>La incertidumbre en el modelo puede tener varias causas. Por ejemplo, todos los procesos reales son no lineales; sin embargo se prefieren los modelos lineales para representarlos. También diferentes condiciones de operación del proceso pueden repercutir en cambios de los valores de los parámetros del modelo lineal, y por ello no pueden conocerse dichos valores con exactitud.

*El diseño de un sistema de control de desempeño robusto buscará que el proceso controlado cumpla con cierto criterio de comportamiento, cuando el sistema es excitado en una de sus entradas, y a pesar de la presencia de incertidumbre en el proceso.* Es decir, en el diseño de un sistema de control que trabaje satisfactoriamente en la realidad deben de especificarse:

- el modelo del proceso,
- los límites de incertidumbre del modelo,
- el tipo de entradas, y
- los objetivos de desempeño.

Recordemos que el diseño del controlador se basa en un modelo del proceso, de una determinada complejidad de acuerdo a la naturaleza del mismo. La no consideración de incertidumbre en el modelo conduce a la obtención de sistemas de control demasiado justos, y que se comportarán indeseablemente en el entorno real. Además, el diseñador debe decidir qué tipo de entradas se aplicarán al sistema y son entonces más importantes, ya que es imposible diseñar un sistema que responda satisfactoriamente a todo tipo de entradas. Finalmente, la medición del desempeño del sistema dependerá de los objetivos de control definidos por el diseñador.

El Control por Modelo Interno se presenta como una posibilidad en el diseño de un sistema de control realimentado, el cual pueda producir un comportamiento deseable cuando actúa sobre un proceso dado, a pesar de la presencia de incertidumbre en el modelo y en el proceso mismo. *Es decir, este sistema presenta la posibilidad de alcanzar un desempeño robusto.* Esta tesis se concentrará, en una parte, en explicar los métodos de Control por Modelo Interno de Uno y Dos Grados de Libertad (abreviados por CMI 1GL y CMI 2GL, respectivamente), y su diseño para procesos SISO (Single Input Single Output, o de una sola entrada y una sola salida) que sean estables pero inciertos en su representación.

Un *sistema de tiempo real* es aquél donde su correcta operación depende de la obtención de los resultados deseados, y también de que estos resultados se presenten en un tiempo oportuno. *A dicha característica en la operación de los sistemas de tiempo real se le conoce como determinismo.* Regresemos, ahora, al ejemplo del automóvil. Aquí, el sistema de frenado del vehículo puede ser visto como un sistema de tiempo real; ya que el tiempo transcurrido entre el accionar del pedal de freno por parte del conductor, y la efectiva reducción de la velocidad del automóvil, no debe de rebasar cierto límite considerado dentro del diseño del sistema, porque de lo contrario podría suceder un accidente al no realizarse la reducción en la velocidad del automóvil de manera oportuna.

Algunos sistemas de control realimentado pueden requerir de una limitante en el tiempo de ejecución del lazo de control; lo anterior es especialmente cierto



cuando la operación segura del proceso bajo control requiere de cumplir con la revisión periódica del estado actual. De lo contrario, el proceso podría alcanzar una condición de riesgo y fallar, lo que provocaría quizá su destrucción parcial o total, o también podría amenazar la salud y seguridad de cualquier persona que se encuentre en el entorno inmediato del proceso al momento de su falla. De aquí, podemos identificar una nueva clase de sistema de control: *el sistema de control en tiempo real*. Para la implantación de sistemas de tiempo real, y en específico de sistemas de control en tiempo real, las computadoras cuentan en el presente con una gran preferencia por parte de ingenieros y técnicos desarrolladores de estos sistemas, gracias a sus relativos bajos costos y versatilidad.

*El objetivo de este trabajo de tesis fue diseñar e implantar un sistema de control en tiempo real, que opere bajo los métodos de Control por Modelo Interno de Uno y Dos Grados de Libertad, para regular el comportamiento de un proceso real, del cual es posible obtener una representación matemática lineal, a la que se le agrega además una descripción con incertidumbre.* En el capítulo 1 se explicará el diseño del Control por Modelo Interno de Un Grado de Libertad, para procesos con modelo exacto, y procesos que contemplen una descripción con incertidumbre. En el capítulo 2 se explicarán las mismas ideas, pero para el esquema de Control por Modelo Interno de Dos Grados de Libertad, estrategia particularmente ventajosa por su mejor rechazo a las perturbaciones actuantes sobre el sistema. En el capítulo 3 se dará una breve introducción a los sistemas de tiempo real, de sus ideas y características fundamentales. Igualmente se abordará el uso de una tecnología basada en computadora y enfocada a la implantación de sistemas en tiempo real, y utilizada por nosotros en la construcción de nuestro sistema de control: la tarjeta de tiempo real PCI-7041/6040E y el software LabVIEW<sup>MR</sup> Real-Time 7.1, ambos desarrollados por National Instruments. Finalmente, en el capítulo 4 se describirá el diseño y construcción de nuestro sistema: el Controlador Virtual Determinístico por Modelo Interno. Ahí mismo se presentarán y comentarán los resultados obtenidos, además de compartir algunas reflexiones derivadas a partir del desarrollo del presente trabajo de tesis.

# Capítulo 1

## Control por Modelo Interno de Un Grado de Libertad

A pesar de que la mayoría de los procedimientos sugeridos en el diseño y ajuste de controladores proponen la aplicación de sistemas lineales para controlar procesos lineales, la realidad nos dice que todo proceso real es no lineal. De igual forma, si hipotéticamente se contara con el modelo de un proceso lineal, nada nos aseguraría que dicho modelo sea exacto en los valores de sus parámetros. *Por lo tanto, los procesos reales son tanto no lineales como inciertos.* Sin embargo, los métodos lineales de control se han logrado adaptar de forma satisfactoria para la resolución de problemas típicos de control en la industria, en la presencia o no presencia de incertidumbre.

Es posible lidiar con la no linealidad de los procesos reales, si se considera que tal proceso opera en un estado donde se puede suponer un comportamiento esencialmente lineal. Por otro lado, la incertidumbre en el proceso puede ser representada por un modelo lineal (llámese función de transferencia), en donde los valores de sus parámetros pueden irse modificando. Entre los factores causantes de esta variación en los parámetros pueden mencionarse las condiciones ambientales existentes en el entorno del proceso (humedad, temperatura, presión, etcétera) o el envejecimiento de los componentes del mismo, por ejemplo. Este enfoque para la modelación de procesos inciertos se denomina *incertidumbre paramétrica*. El enfoque ignora también el comportamiento del proceso durante el cambio de un punto de operación a otro (definido cada uno por un conjunto distinto de valores de los parámetros), y se concentra en el comportamiento del proceso cuando se encuentra en alguno de los puntos de operación posibles. Tal aproximación es razonable cuando el proceso pasa la mayor parte del tiempo operando en un cierto punto de operación, en vez de estar cambiando continuamente de un punto de operación a otro.

Un *sistema de control con desempeño robusto* buscará no sólo lograr la es-

tabilidad de lazo cerrado del sistema de control que opere sobre un proceso incierto, sino igualmente buscará alcanzar el cumplimiento de un determinado criterio de desempeño en la respuesta del sistema al aplicarse a éste alguna señal de entrada, tomando para ello en cuenta también a la incertidumbre del proceso. El esquema de Control por Modelo Interno de Un Grado de Libertad trata la obtención de estos dos objetivos.

## 1.1. Propiedades del Control por Modelo Interno de Un Grado de Libertad

### 1.1.1. Descripción del Esquema de Control

La estructura de Control por Modelo Interno (CMI) propone una estrategia alternativa al control PID. De acuerdo a lo mostrado en la Figura 1.1, el esquema CMI consiste de un controlador,  $q_r(s, \varepsilon_r)$ , y de un modelo del proceso,  $\tilde{p}(s)$ . *Al contemplar el esquema de control el diseño de un solo controlador, se reconoce un método de control de un grado de libertad.* El objetivo del esquema de Control por Modelo Interno de Un Grado de Libertad es lograr para un proceso estable el seguimiento de la señal de referencia,  $r(t)$ , por parte de la variable de salida,  $y(t)$  (o variable del proceso), a pesar de la incertidumbre inherente del proceso,  $p(s)$ ; y al mismo tiempo rechazar los efectos indeseables producidos por la acción de perturbaciones externas,  $d(t)$ , que alteran el comportamiento de la variable del proceso.

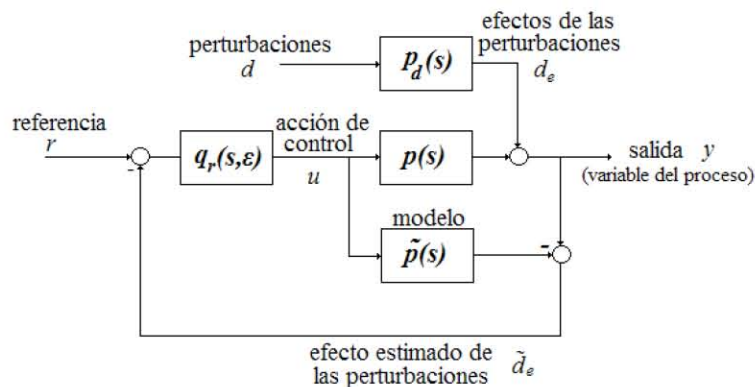


Figura 1.1: Esquema de Control por Modelo Interno de Un Grado de Libertad (CMI 1GL).

El diseño del controlador CMI,  $q_r(s, \varepsilon_r)$  o sólo  $q_r(s)$ , requiere del conocimiento previo de un modelo matemático del proceso,  $\tilde{p}(s)$ , que relacione el comportamiento entre la variable de salida,  $y(s)$ , y cualquier señal de entrada o acción de control,  $u(s)$ . *En específico, se buscará contar con la función de transferencia que relacione a dichas variables.* Igualmente con el ajuste de  $q_r(s)$  se alcanzará un determinado comportamiento deseado del sistema de control, para toda la descripción con incertidumbre del proceso a controlar.

### 1.1.2. Comportamiento en el Estado Estable

Las funciones de transferencia de lazo cerrado que relacionan a la variable de salida,  $y(s)$ , con la referencia,  $r(s)$ , y la perturbación,  $d(s)$ , son:

$$\frac{y(s)}{r(s)} = \frac{p(s)q_r(s)}{1 + (p(s) - \tilde{p}(s))q_r(s)}, \quad (1.1)$$

y

$$\frac{y(s)}{d(s)} = \frac{(1 - \tilde{p}(s)q_r(s))p_d(s)}{1 + (p(s) - \tilde{p}(s))q_r(s)}. \quad (1.2)$$

conocidas también como *función de sensibilidad complementaria* y *función de sensibilidad*, respectivamente.

El comportamiento del sistema CMI 1GL en el estado estable puede analizarse a través de la aplicación del teorema del valor final. En este caso asumiremos que:

- las entradas son de tipo escalón<sup>1</sup>,
- el proceso, su modelo y el controlador son estables, por lo que las ganancias en el estado estable son respectivamente  $p(0)$ ,  $\tilde{p}(0)$  y  $q_r(0)$ , y
- el sistema de control (en lazo cerrado) es también estable.

Bajo estas condiciones obtenemos que

$$\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sy(s) = s \frac{R}{s} \frac{p(0)q_r(0)}{1 + (p(0) - \tilde{p}(0))q_r(0)} = R \quad (1.3)$$

y

$$\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sy(s) = s \frac{D}{s} \frac{(1 - \tilde{p}(0)q_r(0))p_d(0)}{1 + (p(0) - \tilde{p}(0))q_r(0)} = 0 \quad (1.4)$$

si y sólo si  $q_r(0) = [\tilde{p}(0)]^{-1}$ . *Por lo tanto será posible alcanzar en el estado estable el seguimiento a la referencia (expresión (1.3)) y el rechazo a la perturbación (expresión(1.4)), si la ganancia en el estado estable del controlador es igual al inverso de la ganancia en el estado estable del modelo.*

---

<sup>1</sup>En el tiempo  $r(t) = \begin{cases} R; t \geq 0 \\ 0; t < 0 \end{cases}$  y  $d(t) = \begin{cases} D; t \geq 0 \\ 0; t < 0 \end{cases}$ , y en la frecuencia  $r(s) = \frac{R}{s}$  y  $d(s) = \frac{D}{s}$ .

### 1.1.3. Propiedad de Control Ideal

Un sistema de control ideal forzará a la variable del proceso a seguir instantáneamente a la referencia indicada,

$$y(s) = r(s),$$

a la vez de suprimir el efecto de la perturbación existente que afecte a dicha variable de proceso,

$$y(s)/d(s) = 0.$$

De las funciones de transferencia (1.1) y (1.2) se nota que lo anterior es posible para el esquema CMI 1GL cuando

$$\tilde{p}(s) = p(s) \tag{1.5}$$

y

$$p(s)q_r(s) = 1. \tag{1.6}$$

Es decir, *un condición de control perfecto o ideal se alcanzará en el control CMI 1GL si se cuenta con un modelo exacto del proceso* (se trata de un proceso sin incertidumbre, ecuación (1.5)), *y si se logra invertir la ganancia del mismo modelo,  $\tilde{p}(s)$ , para el diseño del controlador,  $q_r(s)$*  (ecuación (1.6)).

En la realidad, las dos condiciones anteriores jamás podrán alcanzarse, por lo que nunca se podrá construir un sistema que presente las propiedades del control perfecto. Por un lado, ya se mencionó la imposibilidad de contar con un modelo exacto de un proceso físico real. Por el otro lado, si se contara con un modelo exacto del proceso, seguramente esta expresión tendría elementos no posibles de invertir, dada la imposibilidad física de construirlos, como son:

- *Tiempos de Retardo:* El término  $e^{Ts}$  representa una predicción irrealizable de salidas futuras. Recordemos que la transformada inversa de  $f(s)e^{Ts}$  es  $f(t+T)$ . De la Figura 1.1, se observa que el controlador,  $q_r(s)$ , opera sobre la diferencia entre la referencia  $r(t)$  y la estimación de la perturbación  $\tilde{d}_e(s)$ . Por lo tanto, si el controlador tiene un término de la forma  $e^{Ts}$ , el controlador requerirá de la predicción de la perturbación en  $T$  unidades de tiempo en el futuro, y una predicción exacta es imposible a menos que tengamos un conocimiento previo de la perturbación, la cual generalmente no es conocida.
- *Ceros no invertibles:* Algunos modelos del proceso,  $\tilde{p}(s)$ , tienen elementos dinámicos en el numerador, denominados también ceros. No importa si alguno de estos elementos está ubicado en el semiplano derecho del plano  $S$ , ya que no afectarán la estabilidad del proceso; sin embargo, si tales ceros se invierten, provocarán una expresión del controlador,  $q_r(s)$ , que será inestable, la cual no es conveniente implantar.

#### 1.1.4. El Controlador CMI $q_r(s)$

Dado que la obtención de la expresión inversa del modelo del proceso es imposible, en el Control por Modelo Interno se buscará aproximar la inversión de dicha expresión al segregarse y eliminarse los elementos de  $\tilde{p}(s)$  que no son posibles de invertir. Entonces, es necesario factorizar el modelo en dos partes: la invertible y la no invertible.

$$\tilde{p}(s) = \tilde{p}_+(s)\tilde{p}_-(s) \quad (1.7)$$

con  $\tilde{p}_+(s)$  como la parte no invertible, la cual contiene los términos que implican predicciones futuras irrealizables y polos inestables, teniendo también ganancia en el estado estable unitaria ( $\tilde{p}_+(0) = 1$ ); y con  $\tilde{p}_-(s)$  conteniendo los elementos si invertibles, y de ganancia en el estado estable igual a la ganancia del proceso ( $\tilde{p}_-(0) = \tilde{p}(0)$ ).

El controlador CMI,  $q_r(s)$ , no toma en cuenta a los elementos en el modelo del proceso,  $\tilde{p}(s)$ , que provocarían un controlador irrealizable, al invertir sólo los elementos del factor invertible del modelo,  $p_-(s)$ .

$$q_r(s) = [\tilde{p}_-(s)]^{-1}. \quad (1.8)$$

Sin embargo, los procesos reales son modelados en su mayoría por funciones de transferencia expresadas como un cociente de polinomios en  $s$ , donde el grado del polinomio del denominador es mayor al grado del polinomio del numerador. Esto originaría una expresión del controlador,  $q_r(s)$ , de acuerdo con la ecuación (1.8), que contendría elementos de diferenciación pura, también imposibles de realizar físicamente. Para solventar este problema, se agrega un filtro pasobajas de la forma,

$$f(s) = \frac{1}{(\varepsilon_r + 1)^\rho}, \quad (1.9)$$

donde  $\rho$  es el grado relativo entre el numerador y el denominador de  $\tilde{p}_-(s)$ <sup>2</sup>. Por lo tanto, se modifica la ecuación (1.8) y finalmente queda el controlador CMI como

$$q_r(s) = \frac{1}{(\varepsilon_r + 1)^\rho} [\tilde{p}_-(s)]^{-1}. \quad (1.10)$$

El valor de la constante de tiempo del filtro,  $\varepsilon_r$ , debe ajustarse para obtener el comportamiento deseado del sistema de control a entradas en la referencia o en la perturbación. Posteriormente se verá que este valor también tendrá que ajustarse, a fin de lidiar con la incertidumbre del proceso, expresada también en su modelo.

---

<sup>2</sup>Para funciones de transferencia que se expresan como el cociente de dos polinomios, el grado relativo  $\rho$  está definido como el orden del polinomio del denominador menos el orden del polinomio del numerador.

## 1.2. Diseño del Controlador CMI $q_r(s)$ para Procesos con Modelo Exacto

En las secciones siguientes se resumirán la ideas de diseño y ajuste de un sistema de control que opere bajo la estrategia CMI 1GL, según lo propuesto por Brosilow y Joshep. En esta primera etapa de diseño, se asume que contamos con un modelo exacto del proceso ( $\tilde{p}(s) = p(s)$ ), el cual además representa a un proceso estable. Igualmente, se considera para todos los casos por analizar que la perturbación  $d(t)$  actúa directamente sobre la variable de salida del proceso  $y(t)$  ( $p_d(s) = 1$ ).

### 1.2.1. Diseño del Controlador CMI para Procesos de Primer Orden con un Tiempo de Retardo

Se considera una planta que se puede modelar como un Sistema de Primer Orden con un Tiempo de Retardo (SPOTR), es decir, la función de transferencia entre la variable de salida y la variable de entrada es de la siguiente forma,

$$p(s) = \frac{K}{\tau s + 1} e^{-Ts}, \quad (1.11)$$

donde  $\tau$  es la constante de tiempo del sistema de primer orden,  $K$  es la ganancia del proceso y  $T$  es el tiempo de retardo.

La expresión que invertiría en su totalidad a dicho modelo y que supondría la realización de un control perfecto, sería

$$p^{-1}(s) = \frac{\tau s + 1}{K} e^{Ts}. \quad (1.12)$$

Como se explicó anteriormente, en esta expresión observamos un único elemento realizable, el cual es el inverso de la ganancia ( $1/K$ ). Por lo tanto, la mejor aproximación posible para la inversión del modelo es de la forma

$$q_r(s) = \frac{\tau s + 1}{K(\varepsilon_r s + 1)}, \quad (1.13)$$

donde  $\varepsilon_r$  es la constante de tiempo del filtro pasobajas de primer orden, añadido al controlador para hacer al mismo realizable y para evitar excesiva amplificación de ruido a frecuencias altas. Considerando las ecuaciones (1.11) y (1.13) junto con el hecho de que  $p(s) = \tilde{p}(s)$  y  $p_d(s) = 1$ , las funciones de transferencia descritas por las ecuaciones (1.1) y (1.2) quedan.

$$y(s) = \frac{e^{-Ts}}{\varepsilon_r s + 1} r(s) + \left(1 - \frac{e^{-Ts}}{\varepsilon_r s + 1}\right) d(s). \quad (1.14)$$

La Figura 1.2 muestra la respuesta en el tiempo de lazo cerrado cuando la constante de tiempo del filtro,  $\varepsilon_r$ , tiene valores de 1 y 0.05. Para ambos casos, se supone una no presencia de perturbación,  $d(s)$ , que altere a la salida del proceso ( $d(s) = 0$ ). Como es posible observar en la Figura 1.2, generalmente un valor mayor en la constante del filtro  $\varepsilon_r$  ocasionará un tiempo de asentamiento mayor de la respuesta. En ambos casos  $T = 1$ .

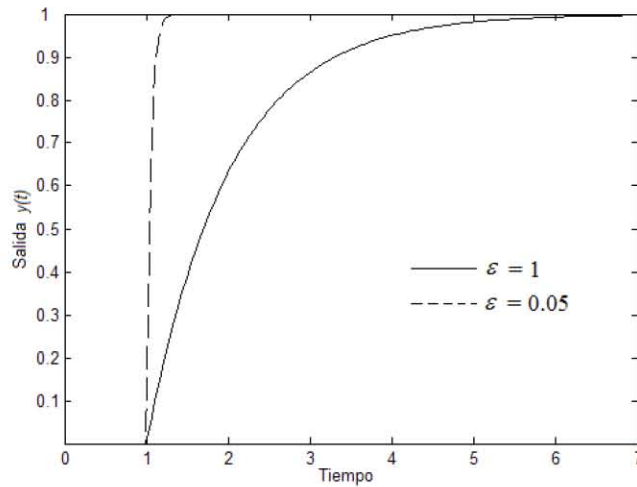


Figura 1.2: Respuesta a una entrada tipo escalón en la referencia del sistema CMI 1GL para el SPOTR.

La elección del valor de  $\varepsilon_r$  no es arbitraria. En primera instancia, dicho valor dependerá del valor máximo permisible para la amplificación del ruido a altas frecuencias por parte del controlador CMI,  $q_r(s)$ . Con el objeto de evitar una amplificación excesiva de ruido a altas frecuencias, generalmente se considera un valor de  $\varepsilon_r$  que no permita una amplificación del ruido en el rango de altas frecuencias mayor a 20 veces, respecto de la ganancia en el rango de bajas frecuencias; o bien

$$|q_r(\infty)|/|q_r(0)| \leq 20. \quad (1.15)$$

Este criterio es aplicado en la mayoría de los procesos industriales; aunque otros criterios suelen ser más estrictos, limitando la amplificación de ruido a factores no mayores de 5 o de 10.

La idea de diseño del controlador CMI para un SPOTR se extiende directamente a procesos cuya función de transferencia es de la forma



$$p(s) = \frac{N(s)}{D(s)} e^{-Ts}, \quad (1.16)$$

donde  $N(s)$  y  $D(s)$  son polinomios en  $s$ . Como se expondrá en la siguiente sección, *el tipo y la ubicación de los ceros de  $p(s)$  (o las raíces de  $N(s)$ ) guiará el diseño del controlador CMI,  $q_r(s)$ .*

### 1.2.2. Diseño del Controlador CMI para Procesos sin Ceros Próximos al Eje Imaginario o Sobre el Semiplano Derecho del Plano $S$

Cuando  $N(s)$  no contiene ceros sobre el semiplano derecho, o que cuando están sobre el semiplano izquierdo no están próximos al eje imaginario, el inverso del modelo es estable y no excesivamente oscilatorio. En este caso, el diseño del controlador,  $q_r(s)$ , para un proceso descrito por la ecuación (1.16) es simple y de la forma

$$q_r(s) = \frac{D(s)}{N(s)(\varepsilon_r s + 1)^\rho}, \quad (1.17)$$

donde  $\rho$  es el grado relativo entre  $N(s)$  y  $D(s)$ . El valor de la constante de tiempo,  $\varepsilon_r$ , debe satisfacer

$$\varepsilon_r \geq \left( \lim_{s \rightarrow \infty} \frac{D(s)N(0)}{20s^\rho N(s)D(0)} \right)^{1/\rho}, \quad (1.18)$$

a fin de no obtener alta ganancia (mayor a 20) en altas frecuencias respecto a la ganancia del controlador en bajas frecuencias.

### 1.2.3. Diseño del Controlador CMI para Procesos con Ceros Cerca del Eje Imaginario

Si el polinomio  $N(s)$  contiene ceros complejos, que al ser invertidos producen polos con factores de amortiguamiento,  $\zeta$ , considerablemente bajos ( $0 \leq \zeta \leq 0.4$ ), la elección de un controlador cuyo diseño siga la idea expuesta por las ecuaciones (1.17) y (1.18) *puede originar un controlador que amplifique notablemente el ruido a frecuencias intermedias*. Por ello, en este caso se imponen dos lógicas para el diseño del controlador  $q_r(s)$ :

- La primera consiste en aumentar el valor de la constante de tiempo del filtro del controlador,  $\varepsilon_r$ , lo suficiente como para reducir la ganancia al valor deseado en el rango de frecuencia medias. Sin embargo, esta idea ocasionaría un incremento en el tiempo de asentamiento de la respuesta de lazo cerrado, por lo que no es conveniente seguirla.

- La segunda y muy recomendable opción es la de no invertir los términos con factor de amortiguamiento bajo, y modificarlos por otros cuyo factor de amortiguamiento sea lo suficientemente grande para evitar la amplificación de ruido a frecuencias medias.

**Ejemplo 1** Tomemos como ejemplo al proceso descrito por la función de transferencia

$$p(s) = \frac{s^2 + 0.004s + 16}{(s + 2)^4}. \quad (1.19)$$

Siguiendo la idea de diseño del controlador CMI,  $q_r(s)$ , expuesta por las ecuaciones (1.17) y (1.18), tendríamos que

$$q_r(s) = \frac{(s + 2)^4}{(s^2 + 0.004s + 16)(\varepsilon_r s + 1)^2}, \quad (1.20)$$

con  $\varepsilon_r = 1/\sqrt{20} = 0.22$ . La respuesta en frecuencia de  $q_r(s)$  se muestra en la Figura 1.3, donde se aprecia un pico aproximado de 14,100 en la curva de magnitud, a una frecuencia de 4 [rad/unidad tiempo], muy por encima de la especificación deseada de ruido, expresada por (1.18).

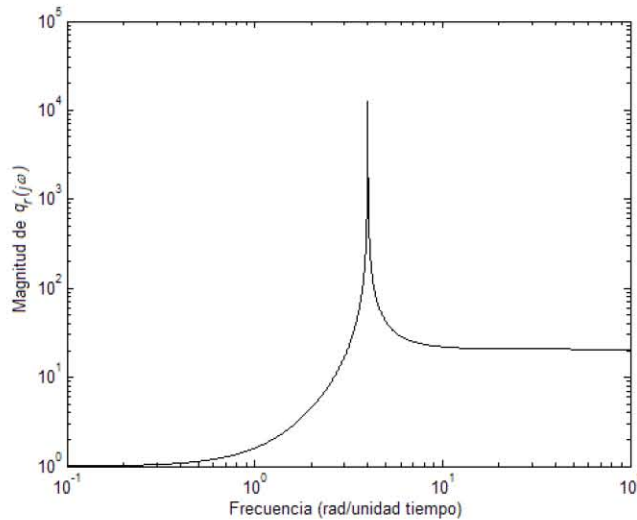


Figura 1.3: Curva de ganancia de  $q_r(s, \varepsilon_r) = \frac{(s+2)^4}{(s^2+0.004s+16)(\varepsilon_r s+1)^2}$  con  $\varepsilon_r = 0.22$ .

Aumentando el valor de  $\varepsilon_r$  a 8.85 se puede disminuir el valor de este pico a la especificación de ganancia de 20, pero esto aumenta el tiempo de asentamiento en la respuesta a escalón de 1.75 [unidad tiempo] a 45 [unidad tiempo]. Por lo tanto, se propone un controlador  $q_r(s, \varepsilon_r)$  de la forma

$$q_r(s) = \frac{(s + 2)^4}{(s^2 + (2\zeta * 4)s + 16)(0.22s + 1)^2}, \quad (1.21)$$

que requiere de un valor del factor de amortiguamiento,  $\zeta$ , de 0.35 para reducir la ganancia pico a 20, manteniendo  $\varepsilon_r = 0.22$ . Como puede verse en la respuesta de seguimiento a escalón de lazo cerrado ( $p(s)q(s) = \frac{s^2 + 0.004s + 16}{(s^2 + (2\zeta * 4)s + 16)(0.22s + 1)^2}$ ) mostrada en la Figura 1.4, el tiempo de asentamiento no se altera significativamente, a pesar de que se produce un poco de oscilación en la respuesta transitoria.

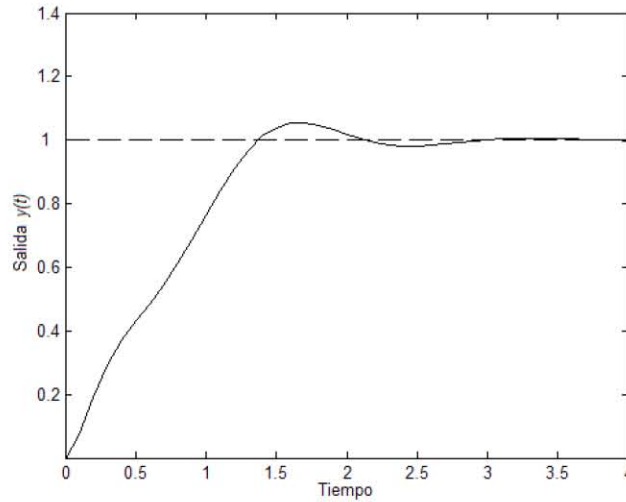


Figura 1.4: Respuesta a escalón del sistema en lazo cerrado del ejemplo 1,  $p(s)q(s) = \frac{s^2 + 0.004s + 16}{(s^2 + (2\zeta * 4)s + 16)(0.22s + 1)^2}$  con  $\zeta = 0.35$ .

#### 1.2.4. Diseño del Controlador CMI para Procesos con Ceros en el Semiplano Derecho

Si  $N(s)$  contiene ceros en el semiplano derecho, su inverso genera un controlador que es inestable, y por lo tanto inconveniente de implantar. La mejor solución en estos casos es la de invertir los ceros existentes en el semiplano izquierdo, y añadir polos que sean la imagen en el espejo de los ceros inestables. Este diseño asegura un comportamiento óptimo en cuanto a un mínimo del error integral cuadrático<sup>3</sup>. Entonces, para un proceso del tipo

<sup>3</sup>  $EIC = \int_0^{\infty} (y(t) - r(t))^2 dt$ .

$$p(s) = \frac{N_+(s)N_-(s)}{D(s)}e^{-Ts}, \quad (1.22)$$

donde  $N_+(s)$  contiene los ceros no invertibles sobre el semiplano derecho, expresados en la forma

$$N_+(s) = \prod_{i,j} (-\tau_i s + 1)(\tau_j^2 s^2 - 2\tau_j \zeta_j s + 1), \quad (1.23)$$

donde  $\tau_i, \tau_j > 0$ ,  $0 < \zeta_j < 1$ , y  $N_-(s)$  contiene los ceros incluidos en el semiplano izquierdo<sup>4</sup>, el controlador,  $q_r(s)$ , se expresa como

$$q_r(s) = \frac{D(s)}{N_-(s)N_+(-s)(\varepsilon_r s + 1)^\rho}, \quad (1.24)$$

donde  $\rho$  es nuevamente el grado relativo entre  $N(s)$  y  $D(s)$ . Aquí  $N_+(-s)$  contiene polos que son la imagen espejo de aquellos ceros no invertibles. En general, la elección de un controlador descrito por la ecuación (1.24) resultará en una respuesta de lazo descrita por

$$\frac{y(s)}{r(s)} = p(s)q(s) = \prod_{i,j} \left( \frac{-\tau_i + 1}{\tau_i + 1} \right) \left( \frac{\tau_j^2 s^2 - 2\tau_j \zeta_j s + 1}{\tau_j^2 s^2 + 2\tau_j \zeta_j s + 1} \right) \frac{e^{-Ts}}{(\varepsilon_r s + 1)^\rho}, \quad (1.25)$$

donde  $\tau_i, \tau_j > 0$ ,  $0 < \zeta_j < 1$ .

**Ejemplo 2** Sea el proceso descrito por la función de transferencia

$$p(s) = \frac{100(s-10)}{27(s+\frac{10}{3})^3} = \frac{-1(-s+10)}{(\frac{3}{10}s+1)^3}. \quad (1.26)$$

El controlador CMI,  $q_r(s)$ , es

$$q_r(s) = \frac{-1(\frac{3}{10}s+1)^3}{(s+10)(\varepsilon_r s+1)^2}, \quad (1.27)$$

de tal suerte que la respuesta de lazo,  $y(s)/r(s)$ , queda

$$p(s)q_r(s) = \frac{(-s+10)}{(s+10)(\varepsilon_r s+1)^2}. \quad (1.28)$$

En la Figura 1.5 se muestra la respuesta EIC óptima con  $\varepsilon_r = 0$  contra otros dos desempeños no óptimos, incrementando el valor de la constante de tiempo del controlador. Todas las respuestas muestran un respuesta inicial negativa, contraria en signo a la ganancia en el estado estable. Como se ve, la respuesta óptima es el mejor compromiso entre una respuesta rápida y una menor respuesta inicial negativa.

<sup>4</sup>Y sin factores de amortiguamiento,  $\zeta$ , pequeños ( $0 \leq \zeta \leq 0.4$ ).

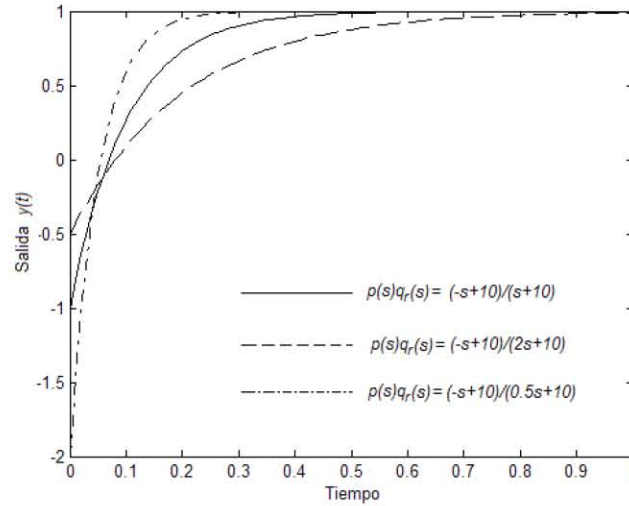


Figura 1.5: Respuesta del sistema con un cero no invertible del ejemplo 2 a una entrada escalón unitario en la referencia.

### 1.3. Descripción de Incertidumbre en el Proceso: Incertidumbre Paramétrica

Un modo de describir matemáticamente la incertidumbre de un proceso real es señalando rangos en los valores de los parámetros de la función de transferencia, de tal forma que para el proceso se define una familia de funciones de transferencia, donde cada miembro de la familia es distinto de otro de acuerdo con los valores que tengan los parámetros en cada caso. La fijación de los límites de incertidumbre del proceso no es tarea sencilla; ya que generalmente la experiencia y el conocimiento del proceso son más importantes para este propósito que cualquier regla teórica. De cualquier forma, *la elección de los límites de incertidumbre desempeñará siempre un papel importante en el ajuste del controlador, así como en el mejoramiento del comportamiento de todo el sistema de control.*

**Definición 1** Definiremos a  $\Pi$  como el conjunto de funciones de transferencia con parámetros inciertos,  $p(s, \beta(\alpha))$ , con el vector de parámetros  $\alpha$  contenido en el conjunto  $S_\alpha$ , donde  $S_\alpha$  es el conjunto de todos los vectores de parámetros  $\alpha$  que cumplen con la restricción  $\underline{\alpha}_i \leq \alpha_i \leq \bar{\alpha}_i$ , en donde  $\bar{\alpha}_i$  es el límite superior del parámetro  $\alpha_i$ , y  $\underline{\alpha}_i$  es el límite inferior del mismo parámetro.  $\beta(\alpha)$  es el vector de funciones continuas de parámetros del vector  $\alpha$ .

**Ejemplo 3** Como ejemplo tomemos un proceso incierto de tipo SPOTR. Si todos los parámetros son inciertos, entonces podemos escribir que el conjunto  $\Pi$  es

$$\Pi = \left\{ p(s) = \frac{Ke^{-T_s}}{\tau s + 1} \mid K \leq K \leq \bar{K}, T_s \leq T_s \leq \bar{T}_s, \tau \leq \tau \leq \bar{\tau} \right\},$$

donde cada uno de los parámetros toma un valor dentro de los límites definidos.

## 1.4. Ajuste $M_P$ para Procesos con Incertidumbre

El objetivo en el ajuste de parámetros de cualquier método de control lineal es cumplir con cierto comportamiento de la respuesta en el tiempo de la variable bajo control, cuando se aplica una determinada señal de entrada (como podría ser una entrada escalón en la referencia). A fin de evaluar el comportamiento en el tiempo de un sistema de control, se definen distintos criterios, los cuales caracterizan a la respuesta tanto en el estado transitorio como en el estado estable<sup>5</sup>. El Ajuste  $M_P$  del CMI 1GL consiste en encontrar el valor mínimo en la constante de tiempo del filtro,  $\varepsilon_r$ , del controlador CMI, de tal suerte que se asegure el comportamiento deseado en el tiempo de la variable bajo control,  $y(t)$ , cuando se aplica una entrada tipo escalón en la referencia,  $r(t)$ , a pesar de la incertidumbre en el proceso,  $p(s)$ , y/o de la acción de una posible perturbación,  $d(t)$ .

### 1.4.1. Relación entre las Características en el Tiempo y en la Frecuencia de un Sistema de Segundo Orden

Para un sistema de segundo orden de la forma  $y(s)/r(s) = 1/(\tau^2 s^2 + 2\zeta\tau s + 1)$ , existen expresiones matemáticas que relacionan las características transitorias en el tiempo de una entrada escalón con los parámetros  $\tau$  y  $\zeta$ . También es posible encontrar relaciones entre dichas características del estado transitorio, y especificaciones en el dominio de la frecuencia, obtenidas a partir de las trazas de magnitud y de fase de la función de transferencia  $y(s)/r(s)$  cuando  $s = j\omega$ . Para un sistema de segundo orden como el mencionado, el sobrepaso máximo  $SP$  (una característica en el tiempo) a una entrada tipo escalón está relacionado con el máximo  $M_P$  en la traza de magnitud contra la frecuencia a través de la relación

<sup>5</sup>Para una entrada tipo escalón, en el estado transitorio: el tiempo de asentamiento  $t_a$ , el tiempo de levantamiento  $t_l$ , el tiempo de retardo  $t_r$ , y el sobrepaso máximo  $SP$ . Además, en el estado estable, el error de estado estable.

$$SP = e^{-\pi M_P(1-(1-1/M_P^2)^{0.5})}, M_P > 1. \quad (1.29)$$

A su vez, el factor de amortiguamiento,  $\zeta$ , está relacionado con  $M_P$  por medio de la expresión

$$\zeta = ((1 - (1 - 1/M_P^2)^{0.5})/2)^{0.5}. \quad (1.30)$$

En las Figuras 1.6 y 1.7 se presentan tanto la respuesta en el tiempo a una entrada escalón unitario como la respuesta en frecuencia del sistema de segundo orden  $1/(s^2 + s + 1)$ . Obsérvese, en este caso, la relación existente entre el sobrepaso  $SP = 0.16$  y el máximo  $M_P = 1.15$ .

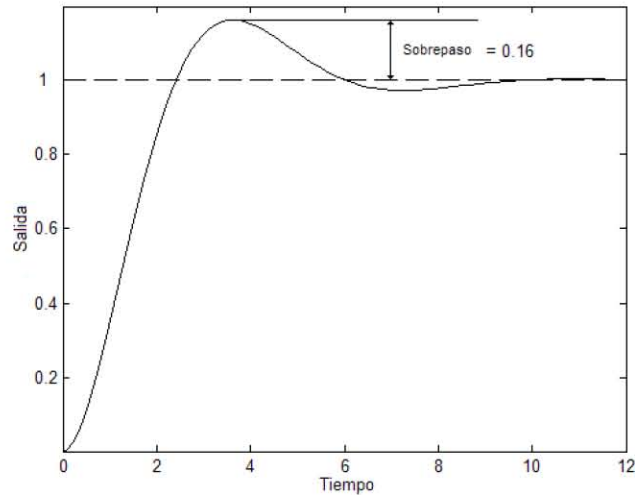


Figura 1.6: Respuesta a una entrada escalón del sistema  $1/(s^2 + s + 1)$ .

Como se puede ver también, la traza de magnitud de un sistema de segundo orden se aproxima en altas y bajas frecuencias a una recta asíntota. La intersección de estas dos rectas define a la frecuencia de corte  $\omega_c$ , la cual es inversamente proporcional a la constante de tiempo  $\tau$  ( $\tau = 1/\omega_c$ ). Como es claro, el valor de este parámetro determina la rapidez en la respuesta en el tiempo del sistema.

En algunos casos es posible aproximar la función de transferencia de lazo cerrado de un sistema de control con la de un sistema de segundo orden, como el presentado anteriormente. *Ello implica que se puede aproximar el comportamiento del sistema de control con el de un sistema de segundo orden equivalente, con lo que seguiría siendo válida la relación entre  $M_P$  y  $SP$ .* En el ajuste

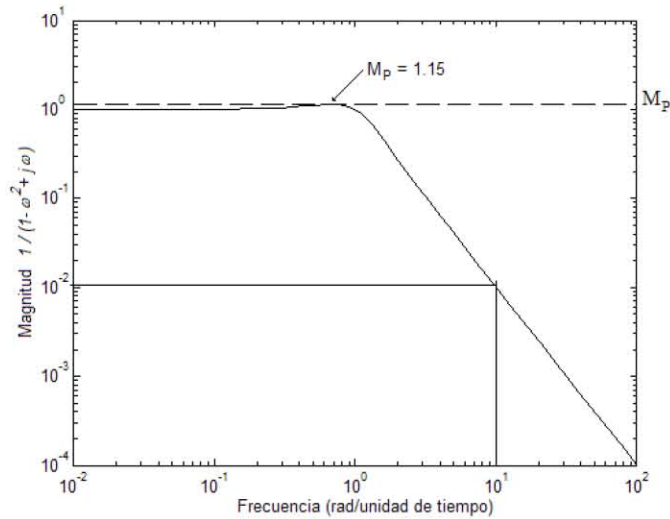


Figura 1.7: Traza de magnitud de  $1/(1 - \omega^2 + j\omega)$ .

del CMI 1GL se tomará este enfoque: buscando limitar la magnitud de  $M_P$  en la traza de magnitud de la función de transferencia de lazo cerrado,  $y(s)/r(s)$  (la función de sensibilidad complementaria  $CS$ ), de acuerdo a un cierto proceso,  $p(s)$ , dentro del conjunto de incertidumbre  $\Pi$ , se definirá un determinado sobrepaso,  $SP$ , en la respuesta en el tiempo de la variable de salida,  $y(t)$ , a una entrada escalón en la referencia  $r(t)$ .

Como regla general de diseño, una ganancia máxima  $M_P$  de 1.05 corresponde a un 10% de sobrepaso, y a un factor de amortiguamiento,  $\zeta$ , igual a 0.6. Valores mayores de  $M_P$  provocarán mayores valores en el sobrepaso y menores factores de amortiguamiento, como se muestra en la Figura 1.8.

#### 1.4.2. Ajuste $M_P$

El problema en el Ajuste  $M_P$  es encontrar el valor más pequeño de la constante de tiempo del filtro,  $\varepsilon_r$ , del controlador CMI,  $q_r(s)$ , que asegura que ninguna traza de magnitud de lazo cerrado contra la frecuencia, entre la variable del proceso,  $y(s)$ , y la referencia,  $r(s)$ , sea mayor a un cierto valor máximo especificado de  $M_P$ . Enunciado formalmente el problema de Ajuste  $M_P$  se debe:

encontrar la constante de tiempo del filtro  $\varepsilon_r$  del controlador CMI,  $q_r(s)$ , de tal manera que la magnitud de la función de sensibilidad



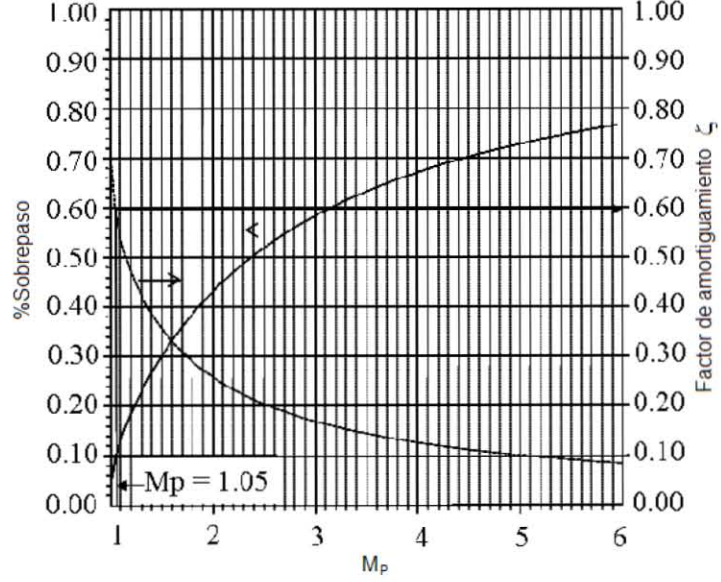


Figura 1.8: Sobrepaso y factor de amortiguamiento contra  $M_P$  para  $1/(s^2 + 2\zeta s + 1)$ .

complementaria,  $CS(i\omega)$ , sea igual o menor a una  $M_P$  especificada para todos los procesos,  $p(s)$ , pertenecientes a un conjunto de incertidumbre predefinido  $\Pi$ . Para al menos un proceso en  $\Pi$ , la magnitud de  $CS(i\omega)$  debe igualar a la  $M_P$  especificada en una o más frecuencias. Es decir

$$|CS(i\omega, \varepsilon_r)| \leq M_P \quad \forall p(s) \in \Pi \text{ y } \forall \omega \quad (1.31)$$

y también

$$|CS(i\omega^*, \varepsilon_r)| = M_P \quad (1.32)$$

para al menos un proceso  $p(s) \in \Pi$  y en ciertas frecuencias  $\omega_1^*, \omega_2^*, \dots$ . Se debe recordar que, en el esquema de control CMI 1GL,  $CS(s)$  viene dada por

$$CS(s, \varepsilon_r) = \frac{p(s)q_r(s)}{1 + (p(s) - \tilde{p}(s))q_r(s)} \quad (1.33)$$

donde  $p(s)$  es cualquier proceso dentro de  $\Pi$  y  $\tilde{p}(s)$  es un modelo nominal empleado en el diseño del controlador CMI,  $q_r(s)$ .

Así, distintos procesos (cada uno diferente de acuerdo a los valores que tomen en cada caso los parámetros) producirán distintas curvas de magnitud de *CS*. Por lo tanto se puede construir una curva de máxima magnitud o ganancia de *CS*, consistente en el valor máximo de dicha función de transferencia para cada frecuencia. La misma idea se puede aplicar en la obtención de la curva de mínima ganancia de *CS*.

**Ejemplo 4 Ajuste  $M_P$  para un Proceso de Primer Orden con Tiempo de Retardo.** Como ejemplo se considera al sistema con función de transferencia

$$p(s) = \frac{Ke^{-Ts}}{\tau s + 1}, \quad (1.34)$$

con parámetros inciertos  $0.6 \leq K \leq 1.4$ ,  $6 \leq \tau \leq 12$ ,  $0.4 \leq T \leq 1$ . Escogiendo un modelo,  $\tilde{p}(s)$ , donde los valores de los parámetros se encuentren a la mitad de cada uno de los rangos de incertidumbre respectivos, se tiene

$$\tilde{p}(s) = \frac{1e^{-0.7s}}{9s + 1}, \quad (1.35)$$

y el controlador,  $q_r(s)$ , es de la forma

$$q_r(s) = \frac{9s + 1}{1(\varepsilon_r s + 1)}. \quad (1.36)$$

El valor para limitar la amplificación de ruido en altas frecuencias por parte del controlador es  $\varepsilon_r = 0.4$ . El valor de  $\varepsilon_r$  para lograr el ajuste de una  $M_P$  máxima de 1.0521 a una frecuencia única de 0.9577[rad/unidad tiempo] es  $\varepsilon_r = 2.3930$ . En la Figura 1.9 se muestran las gráficas de las curvas máxima y mínima ganancia de la función de sensibilidad complementaria, *CS*, para el rango de frecuencias de interés.

Las curvas mostradas en la Figura 1.9 son similares a las de un sistema de 2do. orden, y pueden ayudarnos a estimar las constantes de tiempo de la respuesta más rápida y más lenta, a partir del recíproco de la frecuencia de corte de las curvas de máxima ganancia y mínima ganancia, respectivamente. Los valores aproximados de las frecuencias de corte son 0.9114 [rad/unidad tiempo] para la curva de máxima ganancia y 0.1860 [rad/unidad tiempo] para la de ganancia mínima; ello corresponde entonces a constantes de tiempo de  $1/0.9114 = 1.0972$  [unidad tiempo], y  $1/0.1860 = 5.3763$  [unidad tiempo].

Como ya se mencionó, las curvas mencionadas anteriormente no corresponden a un solo proceso. En cualquier frecuencia, las curvas representan el máximo y el mínimo de la función de sensibilidad complementaria, *CS*, sobre los parámetros del proceso  $K$ ,  $\tau$  y  $T$ ; y los valores de estos parámetros pueden ser cualquiera dentro de la descripción de incertidumbre, con lo cual cada punto

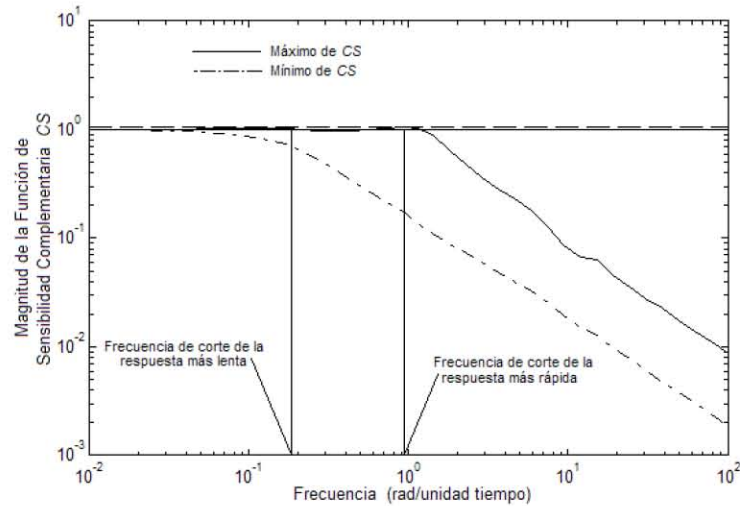


Figura 1.9: Curvas de ganancia máxima y ganancia mínima de la función complementaria de sensibilidad  $CS$ .

dentro de la curva puede representar un proceso distinto. La ganancia máxima de 1.0522 para una frecuencia de 0.9578 [rad/unidad tiempo] corresponde al proceso con parámetros  $K = 1,4$ ,  $\tau = 6$  y  $T = 1$ . Para dicho proceso, se muestra en la Figura 1.10 la respuesta de la variable de salida,  $y(t)$ , a una entrada escalón en la referencia,  $r(t)$ . Ésta es, en específico, la respuesta que mayor sobrepaso presenta (aproximadamente del 10%, de acuerdo con nuestro diseño). Igualmente, se presentan las respuestas más rápida y más lenta del control (correspondientes a los procesos con parámetros  $K = 1.4$ ,  $\tau = 6$ ,  $T = 0.4$  y  $K = 0.6$ ,  $\tau = 12$ ,  $T = 0.5$ , respectivamente).

Hasta ahora se ha supuesto que las curvas de máxima y mínima ganancia de la  $CS$  corresponden en forma a la curva de respuesta en frecuencia característica de un sistema de 2do. orden. Sin embargo, estas curvas de respuesta en frecuencia de  $CS$  presentan un comportamiento distinto, mostrando máximos (picos) y mínimos (valles) relativos a lo largo de la curva de ganancia. Por ejemplo en la curva de ganancia máxima, además de tener el pico absoluto de 1.0521 a una frecuencia de 0.9577[rad/unidad tiempo], se presenta un máximo relativo de 1.0296 a la frecuencia de 0.1203[rad/unidad tiempo]. Igualmente, existen mínimos relativos de 0.9605 en 0.3712[rad/unidad tiempo], y de 0.9998 en 0.0201[rad/unidad tiempo], entre otros.

Existe la necesidad de limitar la diferencia entre picos y valles adyacentes en la respuesta en frecuencia, a fin de evitar oscilaciones indeseables en la respuesta en el tiempo. En general, se impondrá un límite de no más de 0.1 en la

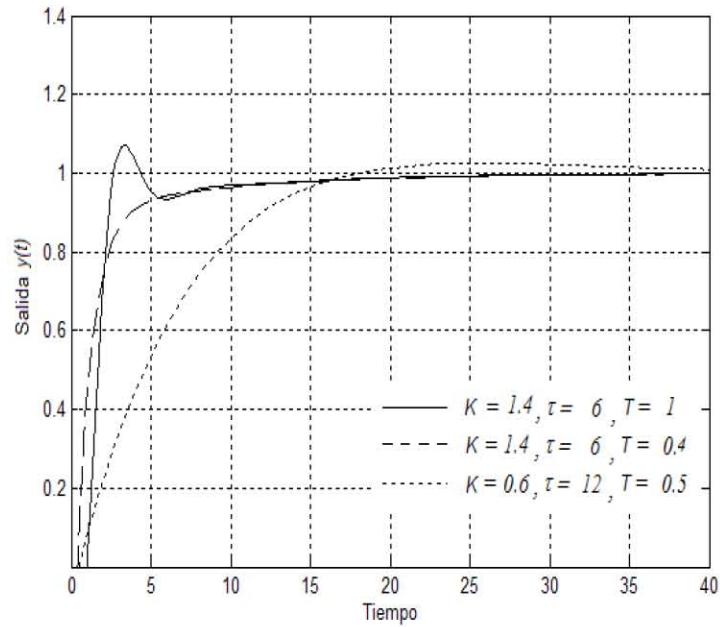


Figura 1.10: Respuestas más rápida, más lenta, y con mayor sobrepaso de la variable de salida  $y(t)$  a una entrada tipo escalón en la referencia  $r(t)$ , para el ejemplo 4.

*diferencia entre la ganancia de un pico relativo y la ganancia del valle adyacente; de lo contrario, será necesario aumentar la ganancia de  $\varepsilon_r$  para limitar dicha diferencia. En este ejemplo,  $1.0521 - 0.9605 = 0.0916 < 0.1$ , por lo que no es necesario aumentar el valor de  $\varepsilon_r$ .*

### 1.4.3. Existencia de la Solución en el Ajuste $M_P$

El Ajuste  $M_P$  asume implícitamente que:

- si todas las trazas de la función de sensibilidad complementaria,  $CS$ , tienen magnitud finita, entonces el sistema de control es estable para todos los procesos dentro de la descripción de incertidumbre  $\Pi$ , y
- existe siempre una constante de tiempo del filtro,  $\varepsilon_r$ , tal que causará que la especificación  $M_P$  de la función de sensibilidad complementaria se ajuste a cualquier límite próximo y mayor a 1 para todos los procesos dentro de la descripción de incertidumbre.

### Teorema de la Controlabilidad Integral

La pregunta ahora es: para un proceso incierto, ¿qué límites existen, de así serlo, en los rangos de los parámetros inciertos del proceso, tal que el sistema de control integral sea estable sobre todos los valores posibles de los parámetros? Para responder esta pregunta, primero se define el concepto de controlabilidad integral.

**Definición 2 Controlabilidad integral.** *El sistema incierto y estable de lazo abierto  $h(s)$  en la Figura 1.11 es llamado integralmente controlable si existe una  $k^* > 0$  tal que el sistema de control es estable para toda  $k$  en el rango  $0 < k < k^*$ . Un sistema de este tipo es uno para el cual siempre es posible diseñar un sistema de control estable y sin error en el estado estable para toda la descripción de incertidumbre.*

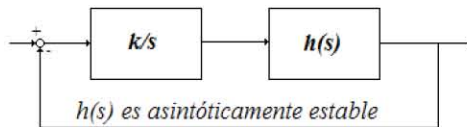


Figura 1.11: Esquema de Controlabilidad Integral.

La condición bajo la cual un sistema es integralmente controlable se enuncia en el siguiente teorema.

**Teorema 1 Teorema de la controlabilidad integral.** *Un sistema incierto  $h(s)$  es integralmente controlable si y sólo si las ganancias de  $h(s)$  ( $h(0)$ ) son positivas para todas las plantas dentro de la descripción de incertidumbre.*

La condición de que todas las ganancias de  $h(s)$  sean positivas *requiere que todas las ganancias de los procesos  $p(0)$  dentro de la descripción de incertidumbre tengan el mismo signo, y nunca sean igual a cero.* Si las ganancias de los procesos son todas negativas, entonces la ganancia del controlador también debe de serlo. En la Figura 1.11 la ganancia negativa del controlador, junto con su dinámica, están incluidas en  $h(s)$ ; tal que  $h(0)$  deba de ser positiva para todos los procesos posibles.

Para un sistema CMI, una definición equivalente de controlabilidad integral es que un sistema estable y de lazo abierto es integralmente controlable, si y sólo

si existe una constante de tiempo del filtro  $\varepsilon^* > 0$ , tal que el sistema CMI de lazo cerrado es estable para todos los valores  $\varepsilon$  mayores o iguales a  $\varepsilon^*$ . Nuevamente la ganancia del proceso no debe de cambiar de signo.

### Condiciones Necesarias y Suficientes para la Existencia de una Solución en el Problema de Ajuste $M_P$ para Cualquier Especificación de $M_P$ Mayor a 1

Existe también una limitante entre la selección de la ganancia del modelo del proceso,  $\tilde{p}(s)$ , y el rango de incertidumbre de la ganancia del proceso,  $p(s)$ , necesaria para poder cumplir con cualquier especificación de  $M_P$  mayor a 1. En la siguiente tabla se señalan los límites máximos del cociente entre la ganancia del proceso y la ganancia del modelo para las cuales es posible lograr especificaciones de  $M_P$  tan cercanas a 1 como se quiera.

**Tabla 1.1. Límites superiores del cociente entre las ganancias del proceso y del modelo para lograr cumplir con especificaciones de  $M_P$  tan cercanas a uno como se quiera**

orden de filtro, n	límite máximo del cociente de ganancia $p(0)/\tilde{p}(0)$
1	$\infty$
2	2
3	1.5
4	1.34
5	1.25

## 1.5. Síntesis $M_P$ para Procesos con Incertidumbre

La síntesis  $M_P$  atiende a la pregunta de qué combinación de modelos del proceso,  $\tilde{p}(s)$ , y del controlador CMI,  $q_r(s)$ , produce respuestas más rápidas para el control, sin dejar de cumplir con la especificación de sobrepaso máximo  $M_P$ . Recordando que la frecuencia de corte de la curva mínima de magnitud de  $CS$  proporciona una buena estimación de la constante de tiempo de la respuesta más lenta, lo anterior equivale a encontrar la combinación de modelos  $\tilde{p}(s)$  y  $q_r(s)$ , tal que dicha curva se desplace hacia la derecha, cumpliendo aún con el límite máximo de  $M_P$ .

Se toma un proceso incierto, estable y sobreamortiguado descrito por

$$p(s) = \frac{K}{(12s + 1)(8s + 1)} e^{-Ts}, \text{ con } 10 \leq K, T < 20; \quad (1.37)$$

y sea el modelo del proceso

$$\tilde{p}(s) = \frac{15}{(12s+1)(8s+1)}e^{-15s}, \quad (1.38)$$

con los parámetros inciertos ( $K$  y  $T$ ) ubicados a la mitad del rango de incertidumbre. Ahora sean los controladores CMI

$$q_r(s) = \frac{(12s+1)(8s+1)}{15(23.51s+1)^2} \quad (1.39)$$

y

$$q_r(s) = \frac{(20s+1)}{15(33.17s+1)}. \quad (1.40)$$

Ambos controladores han sido ajustados para cumplir con una especificación máxima de  $M_P$  de 1.05. En el caso del controlador descrito en la ecuación (1.40), éste se obtuvo al elegir una expresión de primer orden con su constante de tiempo igual a la suma de las constantes de tiempo del sistema de segundo orden original (ecuación (1.38)). En la Figura 1.12 se muestran las trazas de magnitud máxima y mínima para cada uno de los sistemas de control CMI, correspondientes a cada controlador. Como es posible observar, las curvas correspondientes al controlador descrito por (1.40) están más hacia la derecha de lo que están las curvas correspondientes al controlador de la ecuación (1.39). Por lo tanto, las respuestas más lenta y más rápida dadas por el segundo controlador serán más rápidas, sin dejar de cumplir por ello con la especificación máxima de  $M_P$ .

También se muestran en la Figura 1.13 las gráficas de las respuestas más rápida y más lenta a una entrada escalón unitario, de cada uno de los dos sistemas CMI. Aquí es posible apreciar cómo ambas respuestas de lazo cerrado con el controlador de la ecuación (1.40) son más rápidas que aquellas obtenidas con el controlador de la ecuación (1.39).

Si se utilizara un modelo del proceso de primer orden (ie,  $\tilde{p} = \frac{15}{(20s+1)}e^{-15s}$ ); el controlador adoptaría la forma  $q_r(s) = \frac{(20s+1)}{15(35.98s+1)}$ . Sin embargo, las curvas de máxima y mínima magnitud de la función de sensibilidad complementaria,  $CS$ , son prácticamente iguales a las obtenidas por el sistema descrito por las ecuaciones (1.38) y (1.40).

Hasta el momento se ha elegido un modelo del proceso,  $\tilde{p}(s)$ , donde los parámetros  $K$  y  $T$  son iguales al valor ubicado a la mitad del rango de incertidumbre. Si en vez de dichos valores se toman aquellos correspondientes a los límites superiores de  $K$  y  $T$ , en general se puede obtener un sistema que responda más rápidamente a entradas en la referencia,  $r(t)$ . Sea el modelo

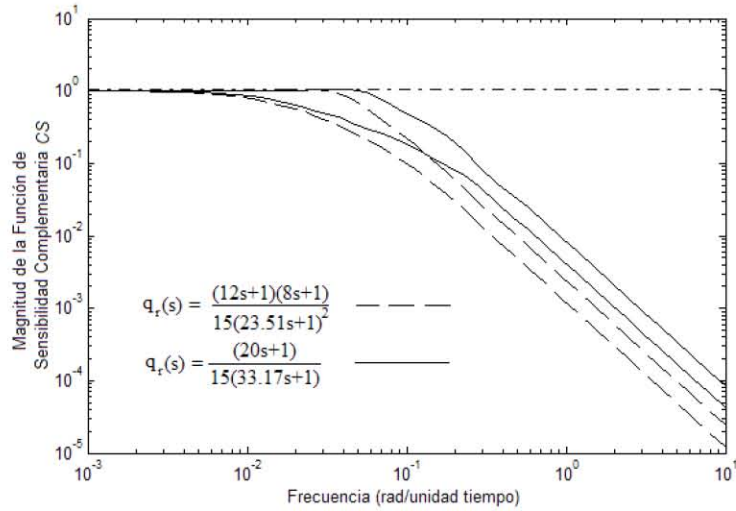


Figura 1.12: Comparación de las curvas de máxima y mínima magnitud de la función de sensibilidad complementaria,  $CS$ , para los sistemas con controladores CMI descritos por las ecuaciones (1.39) y (1.40).

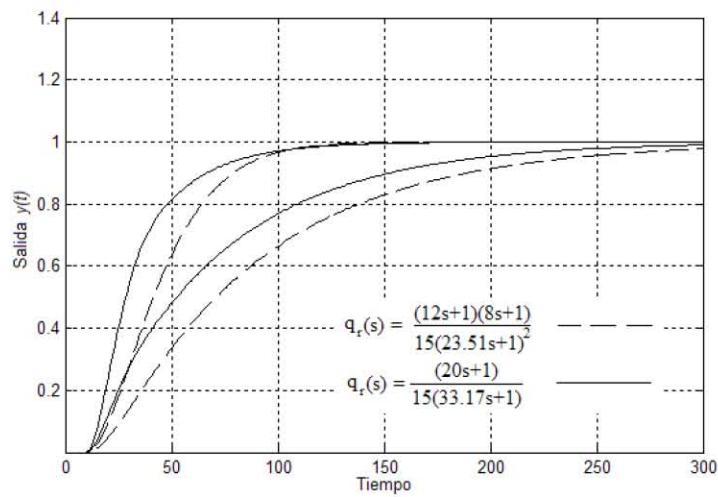


Figura 1.13: Respuestas del sistema CMI más rápidas ( $K = 20, T = 10$ ) y más lentas ( $K = 10, \tau = 10$ ) a una entrada escalón, para los controladores descritos por las ecuaciones (1.39) y (1.40).



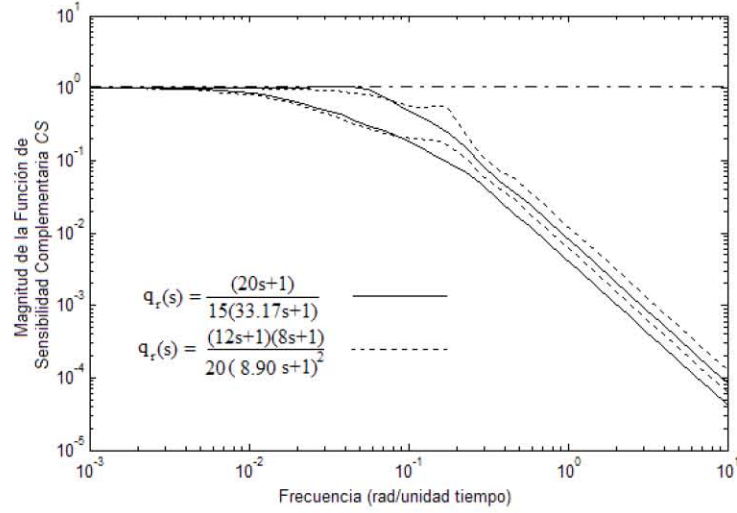


Figura 1.14: Comparación de las curvas de máxima y mínima magnitud de la función de sensibilidad complementaria,  $CS$ , para los sistemas con modelos y controladores CMI descritos por las ecuaciones 1.38 y 1.40, y 1.41 y 1.42.

$$\tilde{p}(s) = \frac{20}{(12s+1)(8s+1)} e^{-20s}, \quad (1.41)$$

entonces la expresión del controlador CMI queda

$$q_r(s) = \frac{(12s+1)(8s+1)}{20(8.90s+1)^2}. \quad (1.42)$$

En la Figura 1.14 se comparan las curvas de máxima y mínima ganancia de  $CS$  para los sistemas descritos conjuntamente por (1.38) y (1.40), y por (1.41) y (1.42). Nótese como para el segundo sistema ambas curvas están más hacia la derecha; por lo tanto producirá respuestas ligeramente más rápidas.

En el ajuste de la constante de tiempo,  $\varepsilon_r$ , de la expresión (1.42) se necesitó limitar la diferencia entre máximos y mínimos relativos de la función de sensibilidad complementaria; ya que originalmente con un valor de  $\varepsilon_r = 6.60$  existía una diferencia entre el máximo  $M_P$  de 1.0514 (a una frecuencia de 0.1873 [rad/unidad tiempo]) y el más próximo mínimo relativo de 0.4856 (a una frecuencia de 0.1010 [rad/unidad tiempo]) igual a 0.5658.

Eligiendo un modelo del proceso,  $\tilde{p}(s)$ , de acuerdo con la expresión

$$\tilde{p}(s) = \frac{20}{20s + 1} e^{-20s} \quad (1.43)$$

se obtiene una expresión del controlador  $q_r(s)$

$$q_r(s) = \frac{20s + 1}{20(14.72s + 1)} \quad (1.44)$$

Nuevamente las curvas de máxima y mínima magnitud de la función de sensibilidad complementaria,  $CS$ , descritas por este último par de funciones de transferencia están ligeramente más hacia la derecha de las obtenidas por el sistema descrito por (1.41) y (1.42). Además, generalmente se preferirán expresiones más sencillas para  $\tilde{p}(s)$  y  $q_r(s)$ , debido a que ello conlleva una implantación más sencilla del esquema de control.

Además, si se compara la respuesta del sistema descrito por (1.43) y (1.44) para el proceso donde  $K = 15$  y  $T = 15$ ; con la del sistema que cuenta con expresiones para el modelo y el controlador iguales a  $\tilde{p}(s) = \frac{15}{(20s+1)} e^{-15s}$  y  $q_r(s) = \frac{(20s+1)}{15(35.98s+1)}$  respectivamente, donde los valores de dichos parámetros para un proceso con modelo exacto son los anteriormente mencionados, se verá que ambas respuestas son muy similares. Dicho de otra manera: a pesar de que el punto de operación del proceso,  $p(s)$ , corresponda la mayor parte del tiempo al descrito por los valores de los parámetros intermedios en el rango de incertidumbre, el sistema que toma para el diseño los valores máximos de  $K$  y  $T$  no responde de manera inapropiada a este caso particular.

Por lo tanto, el ejemplo anterior demuestra que para procesos sobreamortiguados y con un tiempo de retardo las siguientes ideas referentes a la síntesis  $M_P$  pueden guiar la selección del modelo del proceso,  $\tilde{p}(s)$ , y del controlador  $q_r(s)$ :

- Un controlador de primer orden se comporta igual o de mejor manera que un controlador de mayor orden. Esto parece ser cierto cuando la incertidumbre en la ganancia,  $K$ , y el tiempo de retardo,  $T$ , del proceso,  $p(s)$ , es mayor al  $\pm 25\%$ , y este comportamiento mejora conforme dicha incertidumbre vaya siendo mayor.
- La elección de un modelo de primer orden para modelar un proceso sobreamortiguado de orden mayor, y el consecuente diseño de un controlador basado en ese modelo, da un comportamiento igual o mejor al basado en un modelo del proceso del orden correcto, cuando los parámetros de incertidumbre son lo suficientemente grandes.
- La elección óptima de la ganancia,  $K$ , corresponde al límite superior del rango de incertidumbre, y esto es a pesar de que el punto de operación del proceso pueda estar la mayoría del tiempo en el valor intermedio de dicho

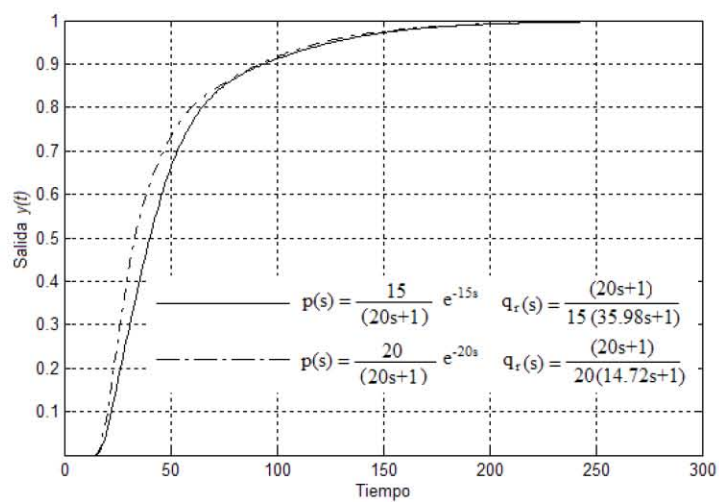


Figura 1.15: Comparación de las respuestas para el proceso con  $K = 15$  y  $T = 15$ , de los sistemas descritos por  $\tilde{p}(s) = \frac{15}{(20s+1)} e^{-15s}$  y  $q_r(s) = \frac{(20s+1)}{15(35.98s+1)}$ , y las expresiones (1.43) y (1.44).

parámetro. Esta idea complementa también la idea expuesta por la tabla 1.1.

## Capítulo 2

# Control por Modelo Interno de Dos Grados de Libertad

### 2.1. Justificación del Control por Modelo Interno de Dos Grados de Libertad

El esquema del sistema de Control por Modelo Interno de Un Grado de Libertad (CMI 1GL) contempla que la perturbación,  $d(s)$ , altera directamente a la variable de salida del proceso,  $y(s)$ . *Sin embargo, cuando dicha perturbación no afecta directamente a la variable del proceso, sino que tal efecto es alterado por la presencia de una función de transferencia,  $p_d(s) \neq 1$ , entonces la respuesta de lazo cerrado proporcionada por el sistema de primer orden a una entrada en la perturbación puede no ser del todo satisfactoria.* A continuación, se explicará a través de un ejemplo la idea anteriormente expuesta, y la conveniencia de implantar un sistema de Control por Modelo Interno de Dos Grados de Libertad (CMI 2GL).

Se considera el proceso con modelo exacto descrito por

$$\tilde{p}(s) = p(s) = \frac{5e^{-5s}}{5s + 1}, \quad (2.1)$$

donde la perturbación es modificada por

$$p_d(s) = \tilde{p}_d(s) = \frac{1}{5s + 1} e^{-5s}. \quad (2.2)$$

El controlador CMI 1GL que asegura la especificación de amplificación máxima de ruido a altas de frecuencias de 20 es

$$q_r(s) = \frac{(5s + 1)}{5(0.25s + 1)}. \quad (2.3)$$

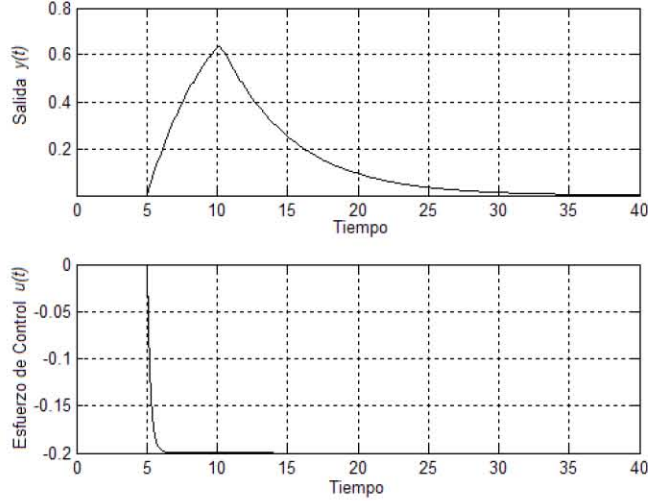


Figura 2.1: Respuesta del sistema CMI 1GL a una entrada tipo escalón en la perturbación,  $d(t)$ .

La función de transferencia que relaciona a la variable de salida,  $y(s)$ , con la perturbación,  $d(s)$ , es

$$\frac{y(s)}{d(s)} = (1 - \tilde{p}(s)q_r(s))\tilde{p}_d(s) = \left(1 - \frac{e^{-5s}}{(0.25s + 1)}\right) \frac{e^{-5s}}{(5s + 1)}. \quad (2.4)$$

Por otro lado, la función de transferencia entre la acción de control (esfuerzo de control),  $u(s)$ , y la perturbación,  $d(s)$ , es:

$$\frac{u(s)}{d(s)} = -\tilde{p}_d(s)q_r(s) = -\frac{e^{-5s}}{5(0.25s + 1)}. \quad (2.5)$$

La Figura 2.1 muestra las respuestas a una entrada escalón en la perturbación,  $d(t)$ , de la variable de salida del proceso,  $y(t)$ , y del esfuerzo de control,  $u(t)$ , para el sistema de control anteriormente expuesto. Como puede observarse, el efecto de la perturbación en  $y(t)$  es prolongado, siendo esto un comportamiento no deseable. Esto se debe a que el esfuerzo de control tarda aproximadamente cinco unidades de tiempo antes de llegar a un valor estable; mientras la respuesta a la perturbación sigue creciendo.

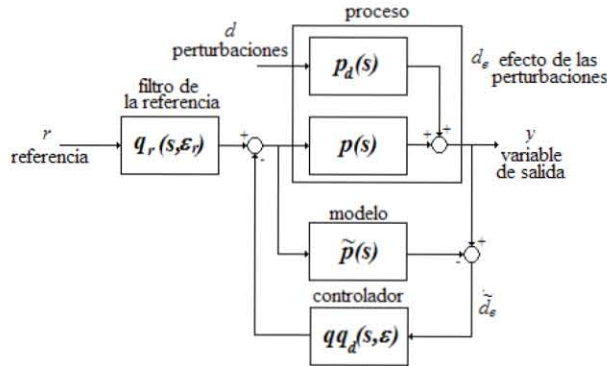


Figura 2.2: Esquema de Control por Modelo Interno de Dos Grados de Libertad (CMI 2GL).

## 2.2. Esquema de Control por Modelo Interno de Dos Grados de Libertad

La Figura 2.2 muestra el esquema de Control por Modelo Interno de Dos Grados de Libertad (CMI 2GL) para una planta cualquiera. En este caso, es necesario el diseño de dos controladores: primero el controlador  $qq_d(s, \varepsilon)$  que será diseñado para rechazar la acción de la perturbación,  $d(s)$ ; y el controlador  $q_r(s, \varepsilon_r)$ , el cual ajustará la respuesta del sistema de control a los cambios en la referencia,  $r(s)$ . Generalmente se conoce en la industria a este último controlador como *filtro de la referencia*. Así, la ventaja de este diseño consiste en la habilidad de modificar las respuestas a la referencia y a la perturbación de manera completamente independiente.

Para un proceso con modelo exacto ( $p(s) = \tilde{p}(s)$ ), la función de transferencia entre la variable de salida,  $y(s)$ , y la referencia,  $r(s)$ , es

$$\frac{y(s)}{r(s)} = p(s)q_r(s, \varepsilon_r), \quad (2.6)$$

y la función de transferencia entre la variable de salida,  $y(s)$ , y la perturbación,  $d(s)$ , es

$$\frac{y(s)}{d(s)} = (1 - \tilde{p}(s)qq_d(s))p_d(s). \quad (2.7)$$

Para el esfuerzo de control,  $u(s)$ , la función de transferencia entre esta variable y la referencia,  $r(s)$ , es

$$\frac{u(s)}{r(s)} = q_r(s), \quad (2.8)$$

y con respecto a la perturbación,  $d(s)$ , es

$$\frac{u(s)}{d(s)} = -qq_d(s)p_d(s). \quad (2.9)$$

### 2.2.1. Diseño de los Controladores $q_r(s)$ y $qq_d(s)$ para Procesos con Modelo Exacto

Nuevamente, en ésta y las secciones siguientes se resumirán la ideas de diseño y ajuste de un sistema de control que opere bajo la estrategia CMI 2GL, según lo propuesto por Brosilow y Joshep.

#### Diseño del Filtro de la Referencia $q_r(s)$

*El filtro de la referencia,  $q_r(s)$ , se diseña de acuerdo con las ideas expuestas para la construcción del controlador CMI de un grado de libertad.* Sin embargo, como generalmente no existe ruido en la referencia, entonces no es necesario especificar un límite en  $\varepsilon_r$  para reducir la amplificación de ruido a altas frecuencias. De cualquier forma no se recomienda un valor muy pequeño de  $\varepsilon_r$ , ya que esto puede producir saturación en el esfuerzo de control,  $u(s)$ , y con ello se podría degradar la respuesta del sistema de control.

#### Diseño del Controlador en la Realimentación $qq_d(s)$

*El diseño de este controlador incluye dos etapas para su realización: primero el diseño de un término  $q(s)$ , y después el diseño de un término  $q_d(s)$ .* El controlador se obtiene a partir de la expresión que resulta al multiplicar ambos términos. El procedimiento de diseño de ambas etapas se describe a continuación:

- Seleccionar  $q(s)$  de acuerdo con las ideas de diseño del controlador CMI de un grado de libertad. Es decir,  $q(s)$  invierte una parte del modelo de proceso  $\tilde{p}(s)$ , además de contar con un filtro pasobajas  $1/(\varepsilon s + 1)^\rho$ , donde  $\rho$  es el orden relativo de la parte invertida del proceso por  $q(s)$ .

- Seleccionar  $q_d(s)$  como

$$q_d(s, \alpha) = \frac{\sum_{i=0}^n \alpha_i s^i}{(\varepsilon s + 1)^n}; \alpha_o \equiv 1, \quad (2.10)$$

donde  $n$  es el número de polos de  $\tilde{p}_d(s)$  a ser cancelados por los ceros de  $(1 - \tilde{p}(s)qq_d(s))$ .

- Seleccionar un valor inicial de prueba para la constante del filtro  $\varepsilon$ .
- Encontrar los valores de  $\alpha_i$  resolviendo la ecuación (2.11) para cada uno de los polos de  $\tilde{p}_d(s)$  que han de ser removidos de la respuesta de la variable del proceso,  $y(s)$ , a la perturbación,  $d(s)$ ,

$$(1 - \tilde{p}(s)qq_d(s, \alpha))|_{s=-1/\tau_i} = 0; i = 1, 2, \dots, n, \quad (2.11)$$

donde  $\tau_i$  es la constante de tiempo asociada con el  $i$ -ésimo polo de  $\tilde{p}_d(s)$ .

Si cualquiera de los polos aparecen en la forma de complejos conjugados, entonces tanto la parte real como la imaginaria de la ecuación son igualadas a cero para cualquiera de los polos conjugados. El conjunto de ecuaciones dado por la expresión (2.11) es un conjunto de ecuaciones lineales con los parámetros  $\alpha_i$  como valores desconocidos, sin importar si los polos de  $\tilde{p}_d(s)$  son reales o complejos.

Si  $\tilde{p}(s)$  contiene polos repetidos ( $\tilde{p}_d(s) = 1/(\tau_j s + 1)$ ), entonces las derivadas de (2.11) son igualadas a cero, hasta un orden menor que el número de polos repetidos.

$$(1 - \tilde{p}(s)qq_d(s, \alpha))|_{s=-1/\tau_i} = 0. \quad (2.12)$$

$$\frac{d^k}{ds^k}(1 - \tilde{p}(s)qq_d(s, \alpha))|_{s=-1/\tau_i} = 0; k = 1, 2, \dots, r - 1. \quad (2.13)$$

- Ajustar el valor de  $\varepsilon$ , y recalculer los valores de  $\alpha_i$  hasta alcanzar el nivel deseado de amplificación de ruido.

Para demostrar el diseño del controlador en la realimentación  $qq_d(s)$ , se retoma el sistema anteriormente descrito. Recordando que

$$\tilde{p}(s) = p(s) = \frac{5e^{-5s}}{5s + 1},$$

y

$$\tilde{p}_d(s) = p_d(s) = \frac{e^{-5s}}{5s + 1}.$$



Se escoge a la parte  $q(s)$  del controlador  $qq_d(s)$

$$q(s) = \frac{(5s + 1)}{5(0.25s + 1)}, \quad (2.14)$$

como anteriormente se hizo en el sistema CMI 1GL para  $q_r(s)$ . Ahora, la parte  $q_d(s)$  del controlador  $qq_d(s)$  tiene la forma

$$q_d(s) = \frac{\alpha s + 1}{\varepsilon s + 1}, \quad (2.15)$$

ya que sólo se necesita eliminar un polo de  $p_d(s)$ . Escogemos un valor prueba de la constante del filtro  $\varepsilon = 0.25$ . De la ecuación (2.11), y considerando que  $p_d(s)$  tiene el polo en  $s = -1/5 = -0.2$ , tenemos que

$$1 - \frac{e^{1(-0.2\alpha + 1)}}{(0.25 * (-0.2) + 1)^2} = 0, \quad (2.16)$$

de donde  $\alpha = 3.33$ . Por lo tanto el controlador en la realimentación,  $qq_d(s)$ , es

$$qq_d(s) = \frac{(5s + 1)(3.33s + 1)}{5(0.25s + 1)^2}. \quad (2.17)$$

En la Figura 2.3 se muestran las respuestas del sistema de Dos Grados de Libertad, de la variable de salida,  $y(t)$ , y del esfuerzo de control,  $u(t)$ , a una entrada tipo escalón en la perturbación,  $d(t)$ . También se observan las mismas respuestas para el sistema de Un Grado de Libertad. Aquí es posible observar la evidente mejoría en la respuesta de la variable de salida a la acción de la perturbación, aunque también es claro que para esto se requiere de un mayor esfuerzo en la acción de control.

El diseño anterior de  $qq_d(s)$  consideró un valor arbitrario de la constante de tiempo del filtro de  $\varepsilon = 0.25$ . Sin embargo, si se desea cumplir con una cierta especificación de amplificación de ruido permisible a altas frecuencias de  $qq_d(s)$ , deberá de calcularse el valor de  $\varepsilon$  que asegure esta restricción. *Si se pide que la amplificación de ruido a altas frecuencias no sea 20 veces mayor a la amplificación del ruido a bajas frecuencias (condición generalmente especificada en la industria), entonces debe cumplirse con la condición*

$$|qq_d(\infty)/qq_d(0)| \leq 20. \quad (2.18)$$

Para nuestro sistema, el cumplir con la condición arriba señalada exige de un valor de la constante de tiempo igual a  $\varepsilon = 0.97$ . Con ello el controlador en la realimentación  $qq_d(s)$  es

$$qq_d(s, \varepsilon) = \frac{(5s + 1)(3.80s + 1)}{5(0.97s + 1)^2}. \quad (2.19)$$

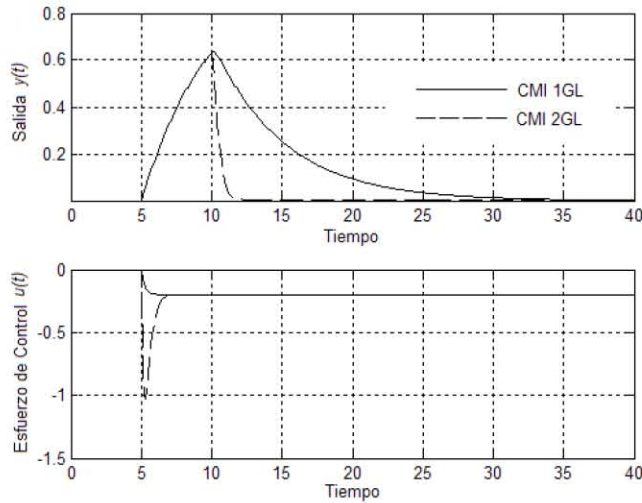


Figura 2.3: Comparación de las respuestas a la perturbación para los sistemas CMI 1GL y CMI 2GL.

En la Figura 2.4 se muestran las respuestas de la variable de salida y del esfuerzo de control para los tres sistemas presentados. Aquí se ve como el cumplir con la especificación de amplificación máxima de ruido a altas frecuencias (con  $\varepsilon = 0.97$ ) retrasa un poco la respuesta de la variable de salida a una entrada escalón en la perturbación, aunque dicho sistema sigue comportándose significativamente mejor que el sistema de Un Grado de Libertad.

**Ejemplo 5** Se toma el caso de un proceso subamortiguado, o sea

$$p(s) = \frac{e^{-s}}{s^2 + 0.2s + 1}, \quad (2.20)$$

y en donde la perturbación,  $d(s)$ , es modificada por la misma expresión (es decir,  $p_d(s) = p(s)$ ). El controlador CMI de Un Grado de Libertad que cumple con la especificación de amplificación de ruido a altas frecuencias de 20 es

$$q_r(s) = \frac{s^2 + 0.2s + 1}{(0.22s + 1)^2}. \quad (2.21)$$

El controlador  $q_d(s)$  del sistema de Dos Grados de Libertad que cumple con la misma especificación de ruido es

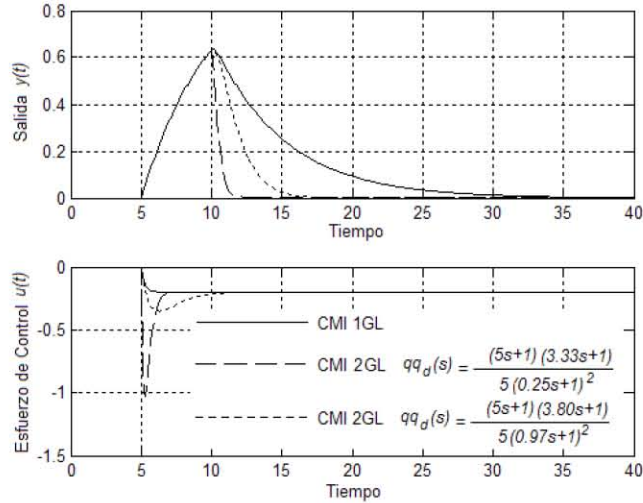


Figura 2.4: Respuestas del sistema CMI 1GL, y de los sistemas CMI 2GL descritos por las ecuaciones (2.17) y (2.19).

$$qq_d(s) = \frac{(s^2 + 0.2s + 1)(2.4s^2 + 0.32s + 1)}{(0.6s + 1)^4}. \quad (2.22)$$

En la Figura 2.5 se muestra la respuesta de los dos sistemas a un entrada escalón en la perturbación. Como es posible apreciar, la respuesta del sistema CMI 2GL es significativamente mejor que la del sistema CMI 1GL. Así, este ejemplo demuestra la ventaja en el comportamiento de un sistema de dos grados de libertad sobre uno de un solo grado, cuando se busca controlar un proceso subamortiguado ( $0 \leq \zeta \leq 1$ ).

### 2.3. Ajuste del Control por Modelo Interno de Dos Grados de Libertad

El ajuste de un sistema de Control por Modelo Interno de Dos Grados de Libertad (CMI 2GL), requiere de la modificación de las constantes de tiempo de los filtros pasabajas,  $\varepsilon_r$  y  $\varepsilon$ , del filtro de la referencia,  $q_r(s)$ , y del controlador en la realimentación,  $qq_d(s)$ , respectivamente. En el caso del controlador  $qq_d(s)$ , se buscará un buen rechazo del sistema a la perturbación,  $d(s)$ ; el filtro  $q_r(s)$  será

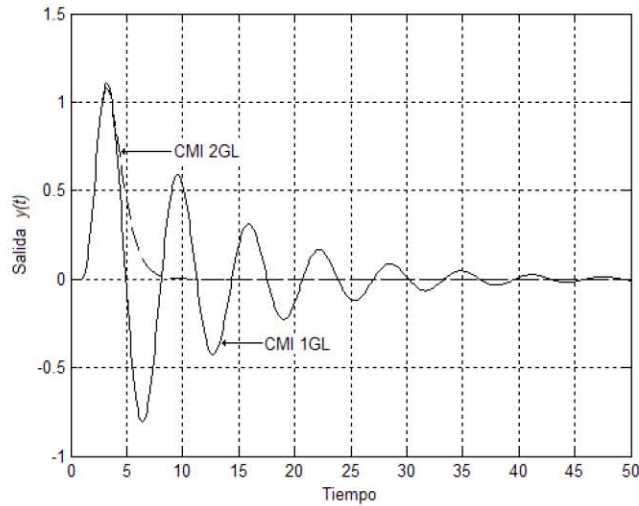


Figura 2.5: Respuestas de los sistemas CMI 1GL y CMI 2GL, para el proceso  $p_d(s) = p(s) = \frac{e^{-s}}{s^2+0.2s+1}$ .

ajustado para alcanzar cierto comportamiento deseado en la respuesta del sistema,  $y(s)$ , a una entrada en la referencia,  $r(s)$ , de acuerdo con el procedimiento de ajuste de un sistema CMI 1GL.

Una observación importante es que el sistema CMI 1GL es un caso particular del sistema CMI 2GL, ya que si  $qq_d(s) = q_r(s)$  ambos sistemas son iguales. También, cuando  $p_d(s)$  está representada por una red donde  $|p_d(i\omega)| \leq 1$  para  $\omega \geq 0$ , entonces se puede obtener un comportamiento significativamente mejor del sistema en lazo cerrado de Dos Grados de Libertad si se hace que  $qq_d(s) = q_r(s)$  y se disminuye el valor de  $\varepsilon$  respecto al valor de  $\varepsilon_r$ .

### 2.3.1. Ajuste $M_P$ para Procesos Estables y/o Sobreamortiguados con Incertidumbre

El ajuste del controlador en la realimentación,  $qq_d(s)$ , debe estar basado en la función de transferencia que relacione a la variable de salida,  $y(t)$ , con la perturbación,  $d(t)$ . Para el sistema de Dos Grados de Libertad esta función es de la forma

$$\frac{y(s)}{d(s)} = \frac{(1 - \tilde{p}(s)qq_d(s))p_d(s)}{1 + (p(s) - \tilde{p}(s))qq_d(s)}. \quad (2.23)$$

Dicha función de transferencia también se denomina función de sensibilidad,  $S$ . Sin embargo, esta función de transferencia raramente se utiliza para ajustar la constante de tiempo  $\varepsilon$  del controlador en la realimentación,  $qq_d(s)$ , ya que a diferencia de la función de sensibilidad complementaria,  $CS$ , utilizada en el ajuste  $M_P$  de un sistema de Un Grado de Libertad, *no existe relación entre un límite máximo del pico de la función de sensibilidad y cierta robustez en el lazo cerrado, de tal suerte que se cumpla con cierto comportamiento de las respuestas en el dominio del tiempo*. Pero, si se realiza una pequeña modificación de la función de sensibilidad, podemos obtener una expresión matemática donde la mismas ideas del ajuste  $M_P$  de un sistema CMI 1GL sean aplicables en el ajuste del controlador  $qq_d(s)$ . Definimos entonces la función de sensibilidad parcial,  $PS$ , como

$$\text{función de sensibilidad parcial} = PS = \frac{p(s)qq_d(s)p_d(s)/p_d(0)}{(1 + (p(s) - \tilde{p}(s))qq_d(s))}. \quad (2.24)$$

La definición anterior está motivada por la siguiente idea: si se toma el segundo término de la función de sensibilidad  $\frac{-\tilde{p}(s)qq_d(s)p_d}{1+(p(s)-\tilde{p}(s))qq_d(s)}$ , eliminamos el signo negativo, reemplazamos el modelo  $\tilde{p}(s)$  en el numerador con  $p(s)$  y tomamos a  $qq_d(s)$  como el controlador CMI, entonces este término (la función de sensibilidad parcial,  $PS$ ) es igual a la expresión de la función de sensibilidad complementaria filtrada por  $p_d(s)$ . Esta idea sugiere la posibilidad de aplicar el algoritmo del ajuste  $M_P$  a la ecuación (2.24).  $p_d(s)$  es dividida entre su ganancia,  $p_d(0)$ , de tal suerte que la ganancia en estado estable de la función de sensibilidad parcial sea 1.

Hay que restringir el uso de la función de sensibilidad parcial a sistemas estables y/o sobreamortiguados, ya que para tales sistemas la función de sensibilidad parcial se comporta muy similarmente a la función de sensibilidad complementaria,  $CS$ . Si  $q_d = p_d = 1$ , la función de sensibilidad parcial se reduce a la función de sensibilidad complementaria. Sin embargo,  $q_d(s)$  es una red de adelanto y la magnitud máxima con modelo perfecto de la función de sensibilidad parcial es la unidad sólo si el proceso es sobreamortiguado.

**Ejemplo 6** *Sea un SPOTR incierto descrito por la función de transferencia*

$$p(s) = \frac{K}{(5s + 1)}e^{-Ts}, \text{ con } 4 \leq K, T \leq 6. \quad (2.25)$$

*Sea también*

$$p_d(s) = \frac{1}{(5s + 1)}. \quad (2.26)$$

*Si se toma al modelo del proceso como*

$$\tilde{p}(s) = \frac{5}{(5s+1)}e^{-5s}, \quad (2.27)$$

entonces la expresión del controlador en la realimentación adquirirá la forma

$$qq_d(s) = \frac{(5s+1)(\alpha s+1)}{5(\varepsilon s+1)^2}. \quad (2.28)$$

Aplicando el ajuste  $M_P$  sobre la función de sensibilidad parcial  $PS$  se obtiene  $\varepsilon = 3$ , con su respectivo  $\alpha = 4.70$ . En la Figura 2.6 se muestra la curva de máxima ganancia de la función de sensibilidad parcial para la  $qq_d(s)$  descrita por los valores de parámetros  $\varepsilon$  y  $\alpha$  anteriores. Igualmente, en la Figura 2.7 están representadas la respuesta del mismo sistema de control a entradas tipo escalón en la perturbación para los procesos  $K, T = 4$  y  $K, T = 6$ .

Después de realizar el ajuste en el controlador de la realimentación, es necesario ajustar el valor de la constante de tiempo,  $\varepsilon_r$ , en el filtro de la referencia,  $q_r(s)$ . Especificando un valor máximo  $M_P$  de 1.05, y utilizando la función de sensibilidad complementaria,  $CS$ , se obtiene que  $\varepsilon_r = 5.17$ . La Figura 2.8 muestra la respuesta del sistema de control a una entrada de tipo escalón unitario en la referencia, para los procesos  $K, T = 4$  y  $K, T = 6$ . Finalmente, en la Figura 2.9 se muestra la curva de máxima ganancia de la función de sensibilidad parcial cuando  $p_d(s) = 1$ . En ella se observa que la estabilidad relativa del lazo en realimentación es buena, ya que el  $M_P$  (pico) es sólo de 1.8.

**Ejemplo 7** En este ejemplo se conserva la expresión del proceso,  $p(s)$ , y de su modelo,  $\tilde{p}(s)$  (ecuaciones (2.25) y (2.27)); pero se modifica la constante de tiempo de  $p_d(s)$  a tener un valor mayor, de tal manera que

$$p_d(s) = \frac{1}{(20s+1)}.$$

El ajuste de  $qq_d(s)$  y de  $q_r(s)$  procede como en el caso anterior, obteniéndose para dichas expresiones

$$qq_d(s) = \frac{(5s+1)(9.58s+1)}{5(3.64s+1)^2}, \quad (2.29)$$

y

$$q_r(s) = \frac{(5s+1)}{5(5.17s+1)}. \quad (2.30)$$

Las Figuras 2.10, 2.11, 2.12, y 2.13 muestran que:

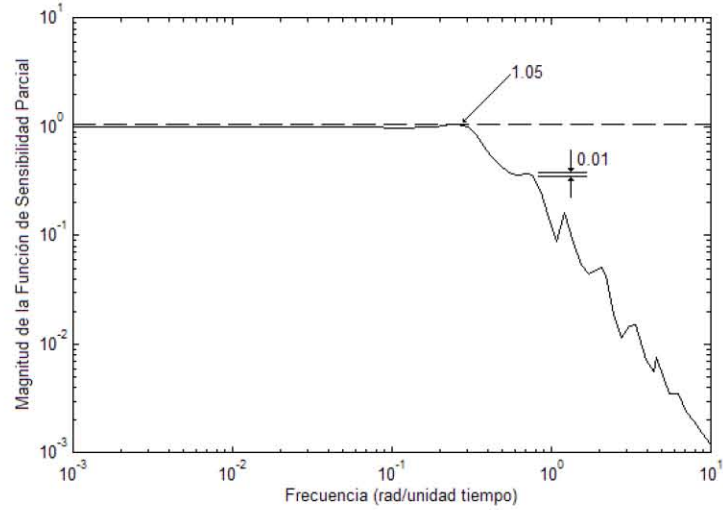


Figura 2.6: Curva de magnitud máxima de la función de sensibilidad parcial cuando  $qq_d(s) = \frac{(5s+1)(4.70s+1)}{5(3s+1)^2}$ , ejemplo 6.

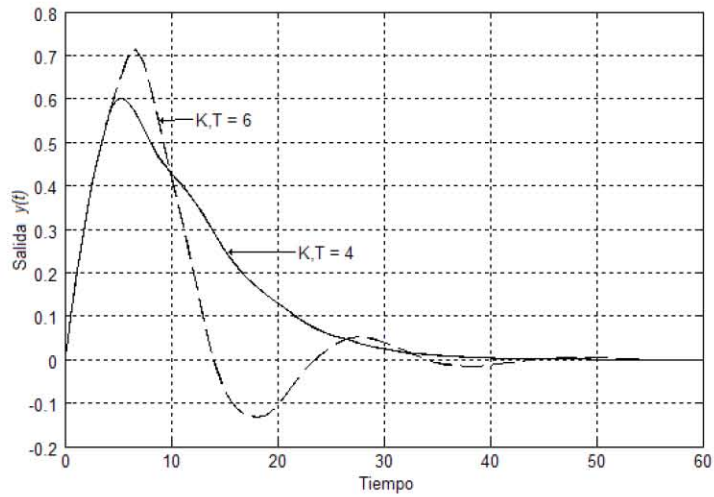


Figura 2.7: Respuestas del sistema de control a una entrada escalón en la perturbación, cuando  $qq_d(s) = \frac{(5s+1)(4.70s+1)}{5(3s+1)^2}$ , ejemplo 6.

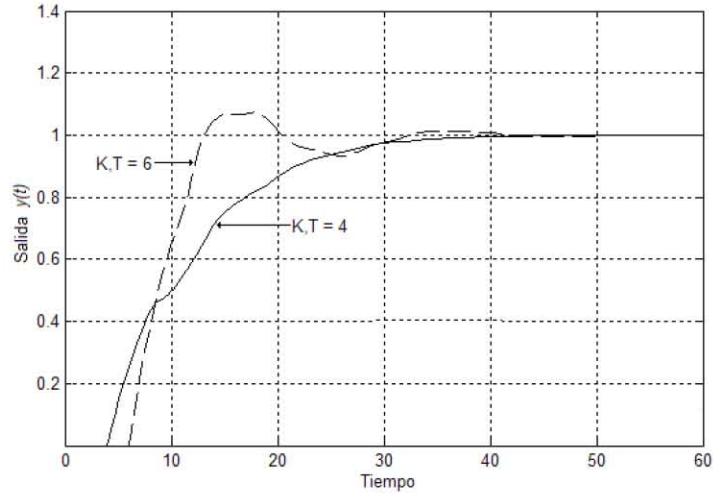


Figura 2.8: Respuesta a la referencia, con  $\varepsilon_r = 5.17$ , ejemplo 6.

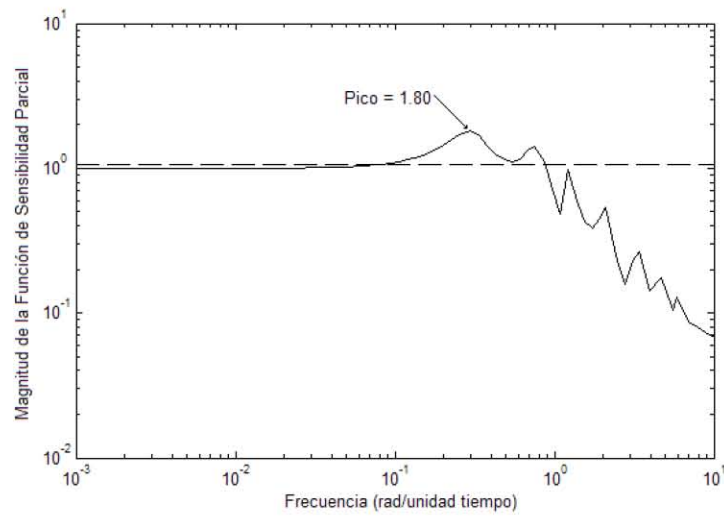


Figura 2.9: Magnitud máxima de la función de sensibilidad parcial, cuando  $p_d(s) = 1$ , ejemplo 6.



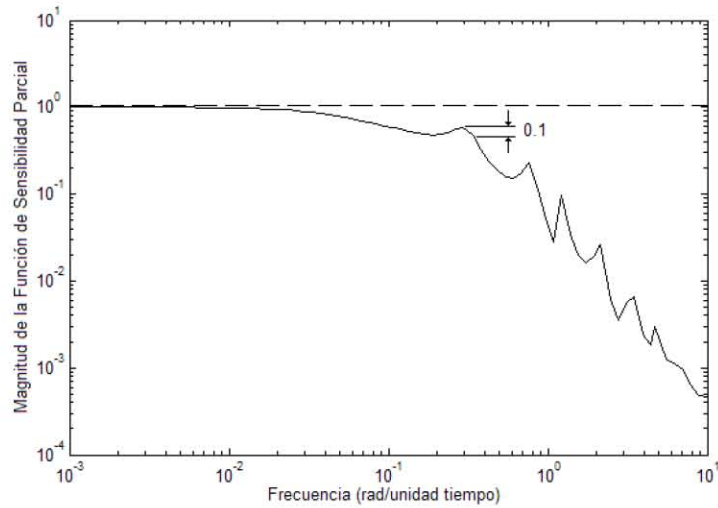


Figura 2.10: Curva de máxima magnitud de la función de sensibilidad parcial, cuando  $qq_d(s) = \frac{(5s+1)(9.58s+1)}{5(3.64s+1)^2}$ , ejemplo 7.

- *el sobrepaso máximo de la respuesta de la salida a una entrada escalón en la perturbación es menor que la obtenida en el ejemplo anterior, gracias a una acción de filtrado más fuerte debida a la mayor constante de tiempo en  $p_d(s)$ , y*
- *el pico de la curva máxima de la función de sensibilidad parcial con  $p_d(s) = 1$  es 3.58 contra 1.80 del caso anterior. Esto también se debe a la más fuerte acción de filtrado de  $p_d(s)$ .*

Por lo tanto, y como se vió en los ejemplos anteriores, es necesario verificar la estabilidad relativa del lazo a partir del máximo de la función de sensibilidad parcial cuando  $p_d(s) = 1$ . Es recomendable no aceptar diseños en donde dicho máximo sea mayor a  $5^1$ .

<sup>1</sup>Una magnitud máxima de 5 para la función de sensibilidad complementaria corresponde a un factor de amortiguamiento  $\zeta$  de 0.1, para un sistema de segundo orden.

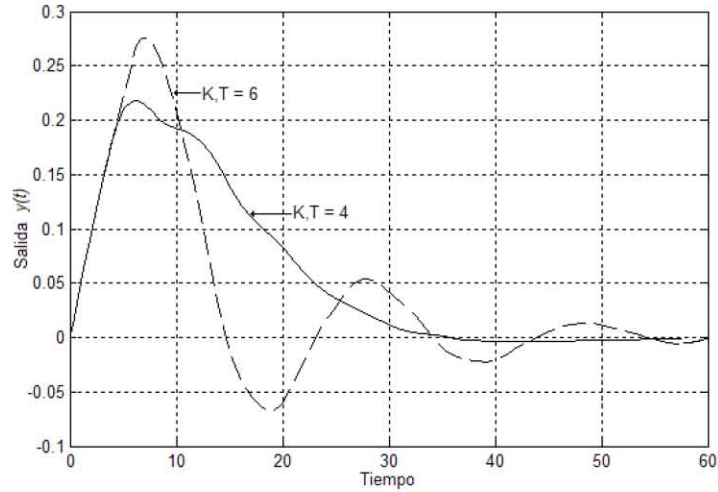


Figura 2.11: Respuesta del sistema a una entrada tipo escalón en la perturbación, cuando  $qq_d(s) = \frac{(5s+1)(9.58s+1)}{5(3.64s+1)^2}$ , ejemplo 7.

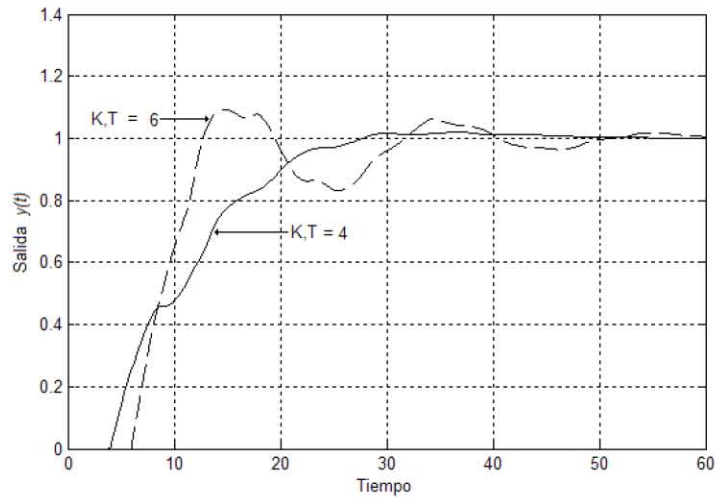


Figura 2.12: Respuesta a la referencia, con  $\varepsilon = 3.64$  y  $\varepsilon_r = 5.17$ , ejemplo 7.

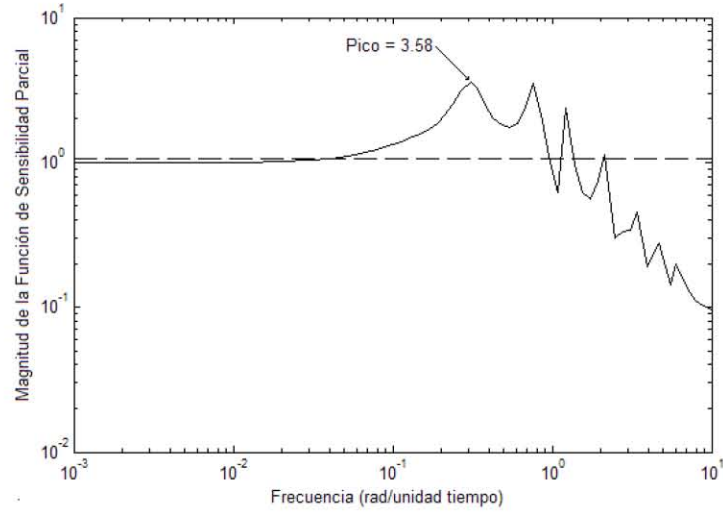


Figura 2.13: Magnitud máxima de la curva de sensibilidad parcial, cuando  $p_d(s) = 1$  y  $\varepsilon = 3.64$ , ejemplo 7.

### 2.3.2. Ajuste $M_P$ para Procesos Estables y Subamortiguados con Incertidumbre

En este caso, no es posible cumplir con una especificación arbitraria de la estabilidad relativa del lazo, usando para ello la función de sensibilidad parcial. Por lo tanto, el problema consiste en hallar el valor de la constante  $\varepsilon$  en  $qq_d(s)$  que da el mejor intercambio entre la velocidad de la respuesta y presencia de oscilaciones, para un proceso incierto dado,  $p(s)$ , y su modelo,  $\tilde{p}(s)$ . *Esta idea es concordante con el ajuste de  $\varepsilon$ , tal que se minimice la curva de máxima ganancia de la función de sensibilidad,  $S$ .*

**Ejemplo 8** Se toma el proceso descrito por

$$p(s) = \frac{K}{s^2 + 0.2s + 1} e^{-s}, 0.4 \leq K \leq 1.6, \quad (2.31)$$

y los modelos

$$\tilde{p}(s) = \frac{1}{s^2 + 0.2s + 1} e^{-s}; \quad \tilde{p}_d(s) = p_d(s) = \frac{1}{s^2 + 0.2s + 1}. \quad (2.32)$$

Ajustando  $\varepsilon = 0.6$ , se define el controlador en la realimentación  $qq_d(s) = \frac{(s^2 + 0.2s + 1)(2.40s^2 + 0.31s + 1)}{(0.6s + 1)^2}$ . En la Figura 2.14 se presenta la curva de máxima magnitud de la función de sensibilidad,  $S$ , para este sistema. Posteriormente,

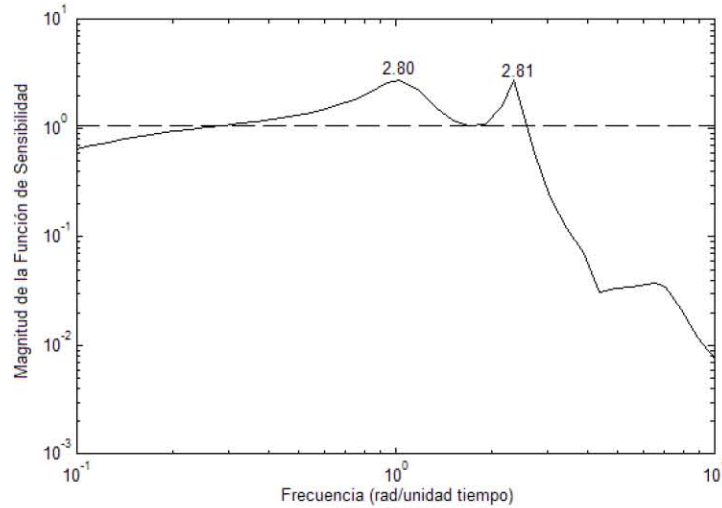


Figura 2.14: Magnitud máxima de la función de sensibilidad para el sistema descrito por  $\varepsilon = 0.6$ , ejemplo 8.

en la Figura 2.15 se muestra la respuesta de salida del sistema a entradas escalón en la perturbación, con los procesos definidos por los valores de ganancia  $K = 0.4$  y  $K = 1.6$ .

Para valores de la constante del filtro  $\varepsilon = 0.7$  y  $\varepsilon = 0.55$ , se obtienen los controladores descritos por  $qq_d(s) = \frac{(s^2+0.2s+1)(2.57s^2-0.16s+1)}{(0.7s+1)^2}$ , y  $qq_d(s) = \frac{(s^2+0.2s+1)(2.29s^2+0.51s+1)}{(0.55s+1)^2}$ , respectivamente. Las Figuras 2.16 a 2.19 muestran tanto las curvas de máxima magnitud de la función de sensibilidad y la respuestas en el tiempo a entradas escalón en la perturbación, para los dos sistemas señalados.

Comose observa, el sistema definido por  $\varepsilon = 0.7$  es el que muestra las mejores respuestas en el tiempo<sup>2</sup>, a pesar de no ser la que produce un curva máxima de la función de sensibilidad que sea mínima (como sucede en el caso de  $\varepsilon = 0.6$ ). Las respuestas asociadas con el sistema descrito por  $\varepsilon = 0.55$  son muy oscilatorias. Por ello, para sistemas subamortiguados, la elección óptima de  $\varepsilon$  es generalmente un poco mayor al valor de dicho parámetro que disminuye la magnitud de la curva máxima de la función de sensibilidad,  $S$ .

<sup>2</sup>Presentan menor oscilación en comparación con las de los otros dos sistemas; en específico para el proceso con ganancia  $K = 1.6$ .

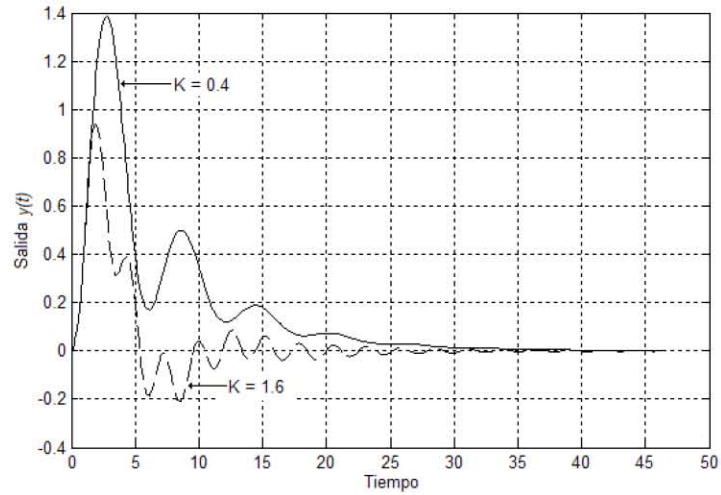


Figura 2.15: Respuesta del sistema descrito por  $\varepsilon = 0.6$  a una entrada escalón en la perturbación, ejemplo 8.

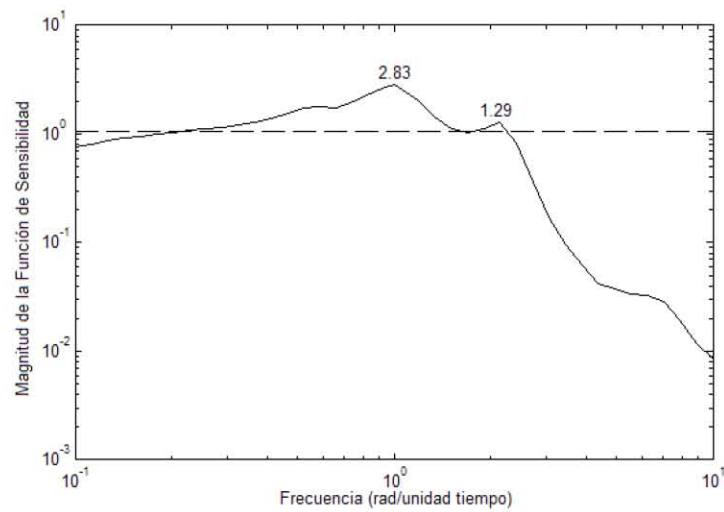


Figura 2.16: Magnitud máxima de la función de sensibilidad para el sistema descrito por  $\varepsilon = 0.7$ , ejemplo 8

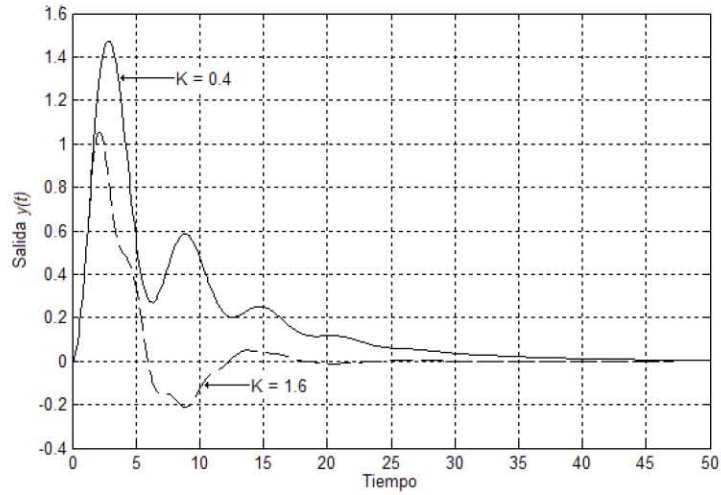


Figura 2.17: Respuesta del sistema descrito por  $\varepsilon = 0.7$  a una entrada escalón en la perturbación, ejemplo 8.

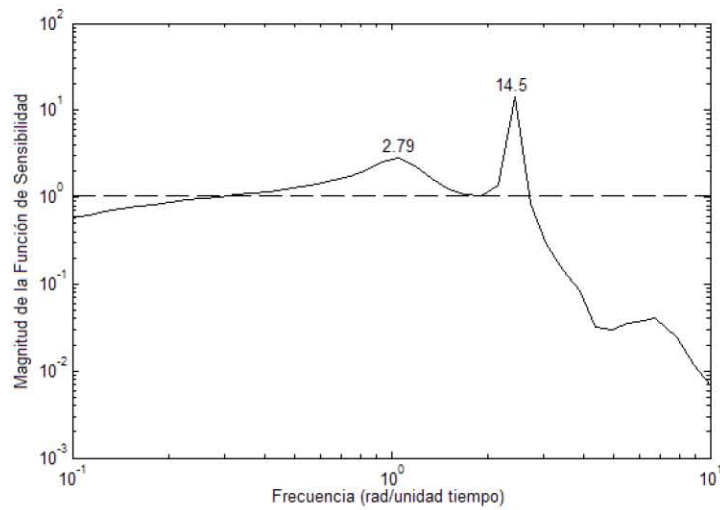


Figura 2.18: Magnitud máxima de la función de sensibilidad para el sistema descrito por  $\varepsilon = 0.55$ , ejemplo 8.

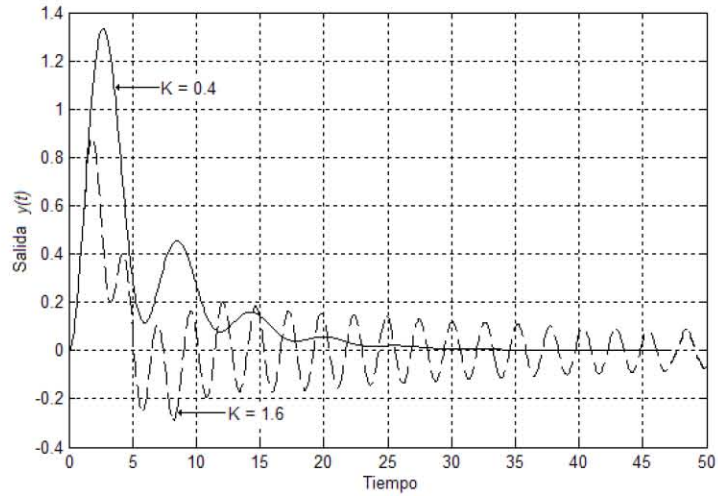


Figura 2.19: Respuesta del sistema descrito por  $\varepsilon = 0.55$  a una entrada escalón en la perturbación, ejemplo 8.

## 2.4. Síntesis del Control por Modelo Interno de Dos Grados de Libertad

### 2.4.1. Síntesis $M_P$ para Sistemas Estables y/o Sobreamortiguados con Incertidumbre

En el caso de la síntesis  $M_P$  de sistemas de Un Grado de Libertad, se vio que la utilización de un modelo  $\tilde{p}(s)$  de primer orden del proceso  $p(s)$ , y el consecuente diseño del controlador,  $q_r(s)$ , a partir de dicho modelo, arroja un sistema de control que se comporta igual o significativamente mejor, al sistema que toma el orden correcto del proceso; siempre y cuando la incertidumbre en el mismo proceso sea grande. Para sistemas de control de Dos Grados de Libertad mostraremos que esta idea es aún aplicable.

**Ejemplo 9** Se considera el proceso descrito por

$$p(s) = Ke^{-Ts}/(12s + 1)(8s + 1); p_d(s) = 1/(24s + 1)(16s + 1); 10 \leq K, T \leq 20.$$

La Tabla 2.1 muestra las expresiones de  $q_d(s)$ ,  $q_r(s)$ , y  $\tilde{p}(s)$ , para distintos sistemas de control. Obsérvese que el primer diseño corresponde al mejor caso para un sistema de Un Grado de Libertad. En el ajuste de todos los controladores

en la realimentación  $qq_d(s)$  requirió de una oscilación máxima de la función de sensibilidad parcial,  $PS$ , de 0.1. Debe recordarse también que la estabilidad relativa del lazo de realimentación puede medirse a partir de la curva máxima de  $PS$ , cuando  $p_d(s) = 1$ . Así, el pico máximo de dicha función es 2.67 para el caso del segundo controlador, 3.76 para el tercero, y 3.69 para el cuarto, contra 1.05 del sistema de Un Grado de Libertad.

**Tabla 2.1 Controladores y Modelos del Ejemplo 9**

	$qq_d(s, \varepsilon)$	$q_r(s, \varepsilon_r)$	$\tilde{p}(s)$
1GL, $q_d = 1, \varepsilon = \varepsilon_r$	$\frac{(20s+1)}{20(14.72s+1)}$	$\frac{(20s+1)}{20(14.72s+1)}$	$\frac{20e^{-20s}}{(20s+1)}$
2GL, modelo 2do.orden	$\frac{(12s+1)(8s+1)(340.25s^2+37.21s+1)}{20(10.77s+1)^4}$	$\frac{(12s+1)(8s+1)}{20(10.74s+1)^2}$	$\frac{20e^{-20s}}{(12s+1)(8s+1)}$
2GL, $q_d = 1, \varepsilon \neq \varepsilon_r$	$\frac{(20s+1)}{20(3.86s+1)}$	$\frac{(20s+1)}{20(19.81s+1)}$	$\frac{20e^{-20s}}{(20s+1)}$
2GL, mejor diseño	$\frac{(20s+1)(28.23s+1)}{20(12.14s+1)^2}$	$\frac{(20s+1)}{20(22.48s+1)}$	$\frac{20e^{-20s}}{(20s+1)}$

La Figura 2.20 muestra la respuesta de los sistemas listados en la tabla anterior a entradas escalón en la perturbación. Aquí es posible observar cómo todos los sistemas de Dos Grados de Libertad se comportan significativamente mejor al mejor diseño de Un Grado de Libertad. Igualmente, debe notarse cómo la selección de modelos del proceso y del controlador de primer orden proporcionan sistemas de control que se comportan mejor, a aquellos en donde se elige el orden correcto del proceso; esto cuando la incertidumbre es grande. La Figura 2.21 muestra las curvas de mínima ganancia de la función de sensibilidad parcial para los sistemas del primero y último renglón. Las curvas correspondientes a los otros dos sistemas caen entre estas dos curvas. Es visible nuevamente que el objetivo de la síntesis  $M_P$  es mover la curvas de mínima ganancia (en este caso de  $PS$ ) hacia la derecha.

La Figura 2.22 muestra la respuesta de los sistemas primero (1GL) y último (2GL) a entradas escalón en la referencia, para el proceso de respuesta más lenta ( $K = 10, T = 10$ ). En semejanza a las curvas mínimas de  $PS$ , la respuesta de los otros dos sistemas cae entre ambas líneas.

Como fue posible observar, el mejor desempeño de los sistemas de Dos Grados de Libertad en el rechazo a la perturbación implica una reducción en la estabilidad relativa del lazo de realimentación. Por ello, quizá en algunas aplicaciones este costo en estabilidad, y la también más complicada implantación del sistema, pueden justificar la no aplicación de esta estrategia de control sobre el esquema de Un Grado de Libertad.



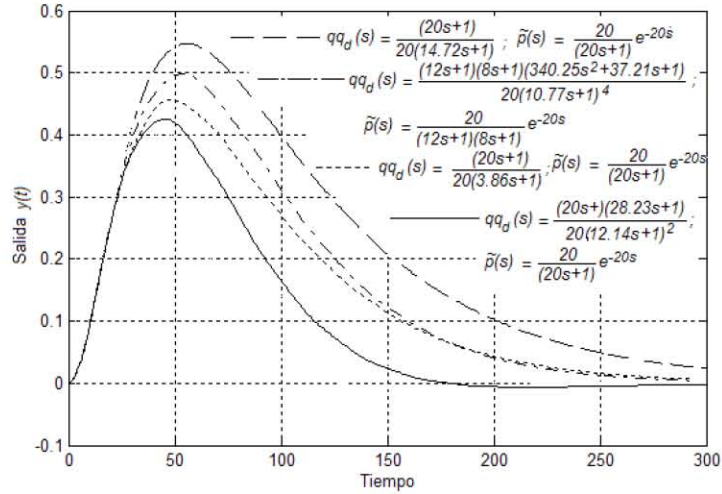


Figura 2.20: Respuestas más lentas (con  $K, T = 10$ ) de los sistemas de control de la tabla 2.1, cuando se aplica una entrada escalón en la perturbación.

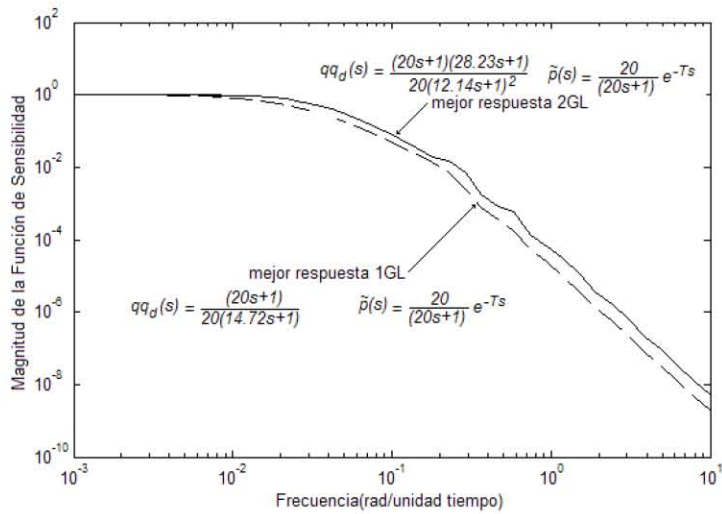


Figura 2.21: Curva mínima de la función de sensibilidad parcial de los mejores diseños de 1GL y 2GL (primer y último sistema de la tabla 2.1).

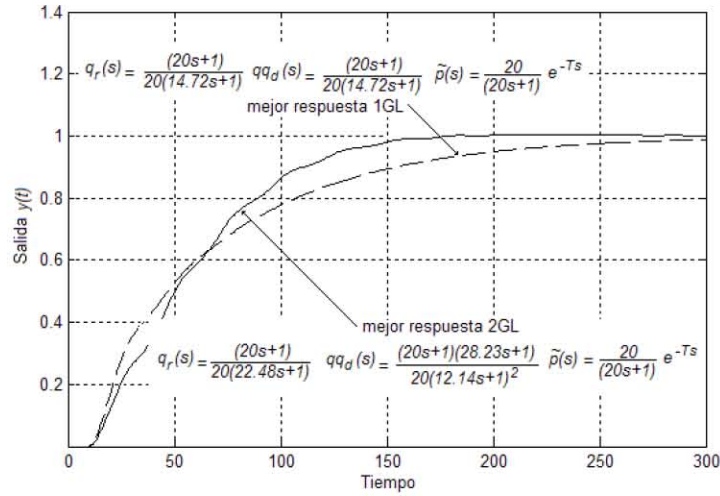


Figura 2.22: Respuestas más lentas a entrada escalón en la referencia de algunos controladores del ejemplo 9 ( $K, T = 10$ ).

### 2.4.2. Síntesis $M_P$ para Sistemas Estables y Subamortiguados con Incertidumbre

El comportamiento de los procesos inciertos subamortiguados requiere de un enfoque distinto en la síntesis del sistema de control. En este caso, se buscará seleccionar la combinación del modelo,  $\tilde{p}(s)$ , y del controlador,  $qq_d(s)$ , que mejore la velocidad de respuesta del proceso más rápido, esperando que esto también acelere la respuesta de los otros procesos. La idea anterior se debe a que los tiempos de asentamiento de las respuestas más rápida y más lenta son prácticamente iguales, de acuerdo a lo mostrado en el ejemplo 8.

**Ejemplo 10** En este caso se utiliza el mismo proceso definido en el ejemplo 8. Recordemos entonces que

$$p(s) = Ke^{-s}/(s^2 + 0.2s + 1); 0.4 \leq K \leq 1.6; p_d(s) = 1/(s^2 + 0.2s + 1),$$

$$\tilde{p}_d(s) = p_d(s) = 1/(s^2 + 0.2s + 1).$$

Después de buscar exhaustivamente la ganancia del modelo que mueve a la función de sensibilidad,  $S$ , a la derecha sin incrementar los picos máximos, se obtuvo una ganancia de  $K = 1.3$ , y una constante de tiempo del filtro del controlador  $qq_d(s)$ ,  $\varepsilon = 0.45$ . La Figura 2.23 compara las curvas máximas de magnitud

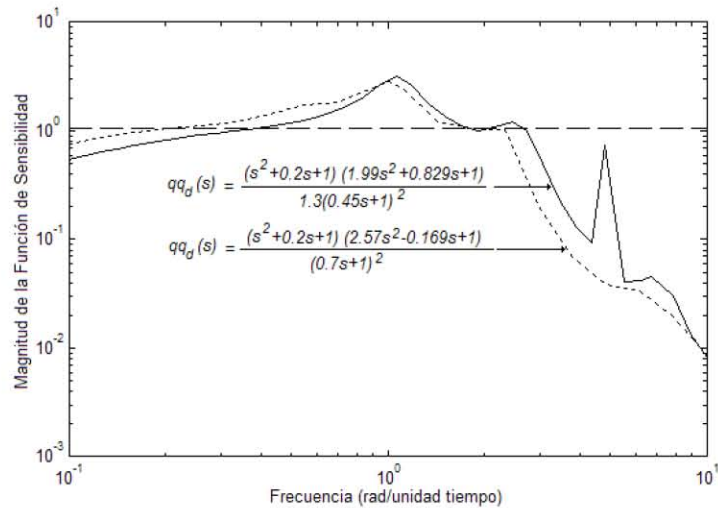


Figura 2.23: Comparación de la curva máxima de la función de sensibilidad del sistema con ganancia media ( $K = 1$ ) y del sistema óptimo ( $K = 1.3$ ).

de la función de sensibilidad de los sistemas obtenidos en el ajuste (ejemplo 8) y en la síntesis. Como puede observarse, los máximos son prácticamente iguales; aunque la curva correspondiente al sistema obtenido en este ejemplo está más hacia la derecha, y presenta también un tercer pico, el cual es el causante de limitar la ganancia  $K$  a un valor de 1.3.

Para observar que la respuesta a la perturbación en el sistema obtenido en esta sección es más rápida que la obtenida en el ejemplo 8, se muestra en las Figuras 2.24 y 2.25 la respuesta de ambos sistemas, para ganancias del modelo,  $p(s)$ , en los límites de la descripción de incertidumbre,  $K = 0.4$  y  $K = 1.6$ .

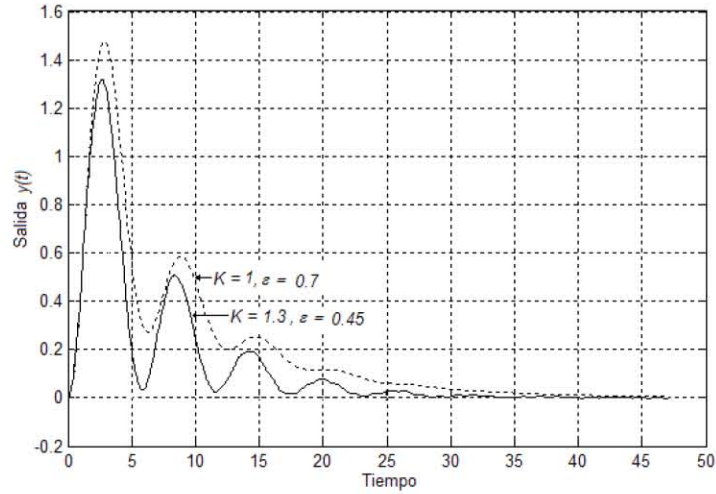


Figura 2.24: Comparación de las respuestas de lazo cerrado en el tiempo a entradas escalón en la perturbación, para una ganancia del proceso igual a 0.4, para los modelos de rango medio y óptimo.

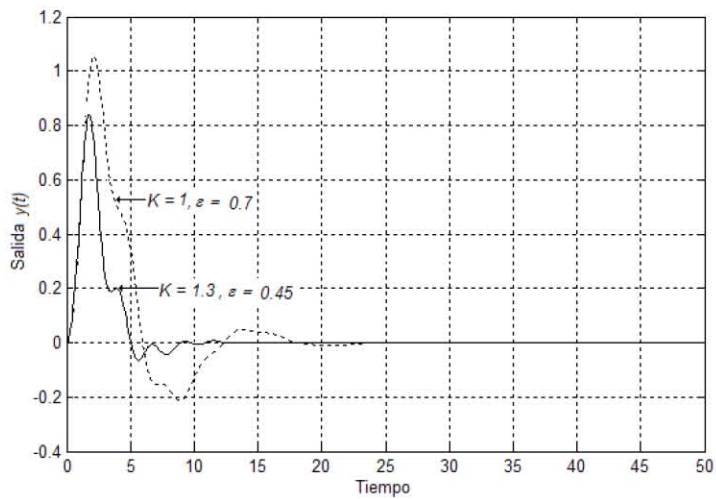


Figura 2.25: Comparación de las respuestas de lazo cerrado en el tiempo a entradas escalón en la perturbación, para una ganancia del proceso igual a 1.6, para los modelos de rango medio y óptimo.



## Capítulo 3

# Implantación de Sistemas de Control en Tiempo Real con LabVIEW Real-Time 7.1 y la tarjeta PCI-7041/6040E

### 3.1. Sistemas de Tiempo Real

En la vida diaria existen situaciones en las cuales el tiempo en el que o durante el cual una acción determinada debe de realizarse juega un papel primordial. Por ejemplo, si se piensa en un sistema de control que verifica la correcta operación de un equipo determinado (una máquina o un proceso), y que en el caso de la presencia de una falla en el equipo deba generar una señal de alarma y posiblemente también realizar toda una serie de acciones, a fin de evitar el daño o destrucción del equipo, e igualmente no comprometer la integridad física de los operadores presentes en el entorno inmediato. En este caso, el tiempo en que deberán de realizarse estas acciones es un factor crítico, ya que un tiempo muy prolongado en la ejecución de las acciones para contrarrestar la falla puede producir el resultado no deseado. Por ello es necesario asegurar que el sistema de control actuará dentro de un límite máximo de tiempo, con el propósito de lograr su correcto funcionamiento y su efectivo control de la falla en el equipo. El tiempo real, por lo tanto, implica una actividad donde el tiempo representa una variable de importancia crítica.

**Definición 3** *Un sistema de tiempo real es aquél sistema o aquella actividad de procesamiento de la información, que debe responder a estímulos de entrada externamente generados, dentro de un periodo de tiempo finito y especificado.*

Por lo tanto, el correcto funcionamiento de un sistema de tiempo real no solamente depende de la obtención de los resultados esperados del sistema, sino también de que estos resultados se presenten en el tiempo en que deben de ser producidos. A dicha característica de los sistemas de tiempo real se le conoce como *determinismo*. Nótese que el concepto de sistema de tiempo real no sólo se limita a aplicaciones de control de procesos, y al contrario se aplica también en otros muchos campos, como son los sistemas de información, las telecomunicaciones o los sistemas de procesamiento de datos en general, así como en la respuesta a eventos (como en el caso del ejemplo anteriormente descrito).

**Definición 4** *Para el control de lazo cerrado de procesos o máquinas, un sistema de control en tiempo real es aquél que monitorea a un proceso, compara el estado actual del proceso con el estado deseado (se obtiene la señal de error dentro del sistema de control.), calcula la acción de control a realizar y ejecuta dicha acción sobre el proceso, con el propósito de alterar la respuesta del mismo proceso bajo control. El tiempo que lleva ejecutar este ciclo es de consideración crítica y se conoce como tiempo de ciclo del lazo de control. El tiempo de ciclo del lazo de control varía de acuerdo a la complejidad del sistema y el proceso bajo control.*

### 3.1.1. Clasificación de Sistemas de Tiempo Real

Dentro de los sistemas de tiempo real se reconocen dos categorías:

- los *sistemas de tiempo real duros*, o
- los *sistemas de tiempo real suaves*.

En los *sistemas de tiempo real duros* es imperativo que el sistema cumpla con la ejecución de sus tareas dentro del tiempo especificado, y en el caso de respuesta a eventos, no es posible que el sistema deje de actuar ante la ocurrencia de uno de ellos. En los *sistemas de tiempo real suaves* el tiempo de respuesta es todavía importante; pero el sistema seguirá funcionando sin mayor problema a pesar de que ocasionalmente no se cumpla con la restricción en el tiempo de ejecución, o en el caso de eventos, se pierda la respuesta a alguno de ellos. En general, un sistema de tiempo real estará en algún punto entre estas dos categorías, dependiendo del grado de relevancia del determinismo para el sistema.

### 3.1.2. Características de los Sistemas de Tiempo Real

La característica más importante de los sistemas de tiempo real es el determinismo. *El determinismo mide el cumplimiento del intervalo de tiempo especificado dentro del que una tarea o acción debe de ejecutarse y completarse.* Aunque esto es deseable, puede no siempre cumplirse y existir un cierto error. A este error se le denomina "*jitter*", y se calcula como la diferencia máxima entre cualquier tiempo de ejecución que revase a la restricción y el tiempo de ejecución especificado.

*Una segunda característica es la confiabilidad y seguridad de operación de un sistema de tiempo real.* Ya que la operación de un proceso está confiada a un sistema de control autónomo, es imprescindible que dicho sistema no falle; y si lo hace, que lo haga de una manera controlada. También, el sistema debe de asegurar que una probable equivocación humana no ponga en peligro la estabilidad del proceso, sin que ello signifique necesariamente una nula interacción con el operador.

*La tercera característica de importancia es la implantación eficiente del sistema.* Dado que en una aplicación de tiempo real el tiempo es una consideración crítica, debe buscarse la implantación eficiente del sistema con el objetivo de cumplir con tales restricciones en el tiempo. En los sistemas de tiempo real, la eficiencia desempeña una papel de mayor relevancia, comparativamente a otros sistemas.

*La cuarta característica es la existencia de un reloj de tiempo real, el cual se utiliza como la base de tiempo para la operación de todo el sistema.*

## 3.2. ¿Cómo Implantar un Sistema de Tiempo Real?

En la práctica, para implantar un sistema de tiempo real se tienen varias posibilidades. Entre ellas están:

- construir un *sistema analógico*,
- utilizar *hardware digital dedicado*, y
- realizar un *sistema basado en computadora*.

Un *sistema analógico* representa una buena opción cuando se trabaja con señales de muy alta frecuencia; o cuando una operación sencilla, como la amplificación o filtrado de una señal, debe realizarse. Sin embargo, esta solución



ofrece poca flexibilidad al desear modificar la operación del sistema, ya que ello implica el cambio de componentes (generalmente eléctricos y/o electrónicos) y el rediseño del circuito que constituye al sistema.

Un *hardware digital dedicado* es óptimo cuando el sistema necesita realizar operaciones a muy alta velocidad, cuando la aplicación es simple y por lo tanto se requiere de un circuito de baja complejidad, o cuando las señales con las que se trabaja son digitales. La flexibilidad para modificar la operación de un sistema basado en hardware digital es relativa, dependiendo del tipo de tecnología utilizada: hoy en día existen muchos circuitos reprogramables por software (los FPGAs, por ejemplo), aunque en otros casos se precisará de modificar los componentes y circuitos del sistema.

### 3.2.1. Los Sistemas de Tiempo Real Basados en Computadora

Las soluciones consistentes en sistemas basados en computadora han estado ganando en los últimos años aceptación entre los desarrolladores de sistemas de tiempo real, y en específico en las aplicaciones de control de lazo cerrado de procesos. Su creciente popularidad se debe a diversos factores:

- Primeramente está el bajo costo de los componentes de hardware (procesadores, memoria, unidades de almacenamiento, etcétera) requeridos para construir un sistema de cómputo completo y funcional, y la tendencia marca que estos costos seguirán disminuyendo en el futuro.
- Además, la modularidad en la construcción de un sistema de cómputo permite el diseñar y desarrollar un sistema adecuado a las necesidades, contando además con la posibilidad de adquirir los mismos componentes de diversos fabricantes, lo que conlleva la reducción en los tiempos y costos de construcción del sistema.
- También, la continua evolución de lenguajes de programación de alto nivel y sistemas operativos de tiempo real más potentes permite la realización en software de aplicaciones cada vez más complejas.
- Al mismo tiempo este software permite una mayor flexibilidad en el rediseño y modificación de la operación del sistema.
- Finalmente, se han desarrollado interfaces de hardware (amplificadores, convertidores digitales/analógicos y analógico/digitales, etcétera) que hacen posible la adquisición y procesamiento de señales externas y reales con una computadora.

Dentro de las desventajas está la imposibilidad de alcanzar velocidades de procesamiento muy altas, limitante provocada principalmente por las características del hardware.

### 3.2.2. Descripción de un Sistema de Control de Lazo Cerrado en Tiempo Real Basado en Computadora

En un sistema de control de lazo cerrado en tiempo real, la computadora se desempeña básicamente como el controlador del proceso, además de añadir características de tiempo real a la implantación del controlador. La conjunción de varios elementos de hardware y de software permiten entonces que el control y monitoreo de un proceso o máquina sea realizado por una computadora, en forma completamente autónoma y con el cumplimiento de las restricciones de tiempo impuestas a la operación de todo el sistema de control. Es lógico pensar que la computadora requerirá trabajar con señales físicas reales y externas a ella, como también habilitar alarmas en casos de emergencia, proveer una interfaz de comunicación clara y oportuna con el operador, y posiblemente registrar en una base de datos información experimental recabada (para la generación posterior de reportes) o consultarlos de la misma base. En la Figura 3.1 se muestran todas estas ideas.

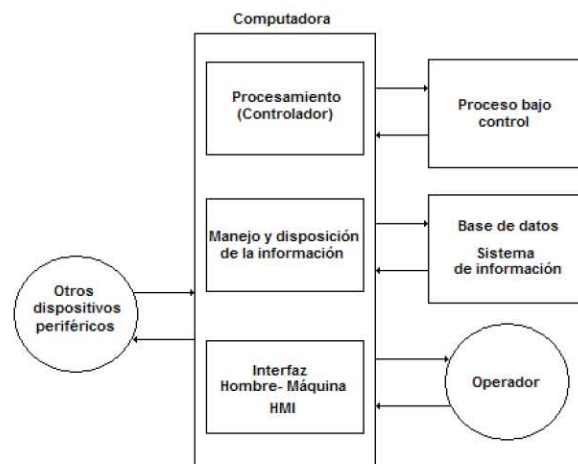


Figura 3.1: Diagrama de un sistema de control en tiempo real basado en una computadora.

#### El Hardware

En la actualidad las computadoras digitales han tomado la tutela en muchos campos y aplicaciones. Sin embargo, una computadora digital no puede operar directamente sobre una señal física real, generalmente analógica. Por ello, es importante considerar que para utilizar una computadora digital dentro de un sistema de control de lazo cerrado en tiempo real se debe:

- trabajar con muestras de la señal analógica, y
- acondicionar dicha señal, para poder operar sobre ella.

El muestreo de una señal contempla la determinación de con qué regularidad tendrá que adquirirse una nueva muestra de la señal. Esto define una *frecuencia de muestreo*  $f_s$ , la cual no puede ser una cantidad arbitraria, sino que debe de cumplir con el criterio de Nyquist. Según dicho criterio, *el valor mínimo de la frecuencia de muestreo  $f_s$  debe ser igual al doble de la componente de máxima frecuencia de la señal.*

*El acondicionamiento de la señal consiste en una serie de alteraciones que se necesitará realizar a la señal analógica de entrada, de tal suerte que la computadora digital pueda operar sobre ella (o sea, convertirla en una señal digital.).* Igualmente, las señales de salida ya procesadas por la computadora digital deberán también de acondicionarse, con el propósito de generar una salida utilizable, generalmente también de tipo analógica. A continuación se menciona las principales etapas en el acondicionamiento de una señal.

- *Transductor.* Aquí una señal física no eléctrica (como la presión o la temperatura) se convierte en una señal eléctrica (corriente o voltaje), o al contrario, una señal eléctrica se convierte en una señal no eléctrica. A los transductores del primer tipo se les denomina *sensores*, y a los segundos *actuadores*.
- *Linealización.* Generalmente usada a la entrada de una computadora digital, la linealización permite corregir la característica no lineal de algunos sensores. Aunque existe cierto hardware que desempeña esta función, dicha etapa también puede implantarse vía software.
- *Amplificación.* En esta etapa, una señal eléctrica es modificada en su amplitud, pero no en su frecuencia o en su fase. Lo anterior se hace por muchos factores: para alcanzar un nivel de señal con el que sea posible trabajar, ya que algunas señales pueden ser muy pequeñas y casi imperceptibles (del orden de miliamperes o milivolts); o para disminuir la presencia del ruido en la señal (aumentando la relación señal a ruido, o *SNR*).
- *Filtrado.* Un filtro elimina componentes en frecuencia de una señal que no son deseables, generalmente ruido. Dentro de las distintas clases de filtro existentes, de acuerdo a la banda que rechazan, están: pasobaja, pasoalta, pasobanda y supresor de banda o filtro muesca.
- *Aislamiento.* Cuando el nivel de una señal eléctrica externa es demasiado elevado para los circuitos internos de una computadora digital, o cuando simplemente se desea aislar una señal de entrada o salida de la computadora, puede utilizarse cualquier dispositivo que contenga un aislamiento de tipo magnético u óptico (como los amplificadores de aislamiento u optoacopladores).

- *Convertidores analógico/digital y digital/analógico.* Como su nombre lo dice, el primero cuantiza y convierte a una representación digital una señal analógica, y el segundo convierte una representación digital a un nivel correspondiente de señal analógica.

No es necesario que todas estas etapas estén presentes en un sistema real, y la presencia de ellas dependerá naturalmente de la aplicación de que se trate.

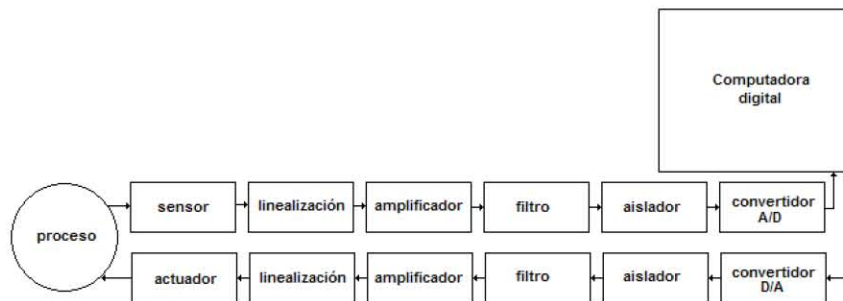


Figura 3.2: Etapas de acondicionamiento de señal para un sistema de computadora.

Para el procesamiento de la señal se emplean elementos de hardware comunes en todas las computadoras digitales: un procesador, memoria volátil (RAM), memoria no volátil (ROM), entre otros.

## El Software

**Sistemas Operativos de Tiempo Real** Existen dos clases de sistemas operativos:

- los *sistemas operativos de propósito general* (como Windows<sup>MR</sup> o Linux), y
- los *sistemas operativos de tiempo real*.

*Una tarea, o proceso, es cualquier actividad realizada por la computadora.* Consiste de un programa, datos asociados con el programa, y recursos de la computadora tales como espacio en memoria o dispositivos de entrada/salida requeridos para ejecutar el programa. El sistema operativo es la base sobre la

que esta tarea o proceso se puede ejecutar, al proveer los medios para que el hardware y el software de la computadora se comuniquen entre ellos.

Una de las principales diferencias entre los sistemas operativos de tiempo real y los de propósito general es la habilidad de los primeros en garantizar un peor caso en el tiempo de espera de ejecución de una tarea o interrupción. En un sistema operativo de propósito general, una interrupción externa es colocada al final de la cola de todas las interrupciones pendientes, y es atendida cuando el sistema operativo haya terminado con la tarea actual y haya servido también a todas las otras interrupciones pendientes. Por otro lado, un sistema operativo de tiempo real puede detener el proceso actual para servir inmediatamente a la interrupción. *En esencia, un sistema operativo de tiempo real garantiza que la respuesta a un evento se presente dentro de un determinado intervalo.* Para hacer esto, los sistemas operativos de tiempo real asignan prioridades a las tareas o procesos, y a las interrupciones. Si una interrupción de más alta prioridad ocurre, el sistema operativo detiene el proceso actual y sirve a la interrupción.

En la mayoría de los sistemas operativos de tiempo real una tarea puede estar en cualquiera de los siguientes cuatro estados:

- *Estado de ejecución*, cuando la computadora realiza la tarea.
- *Estado en espera de ejecución*, si la tarea está lista para ejecutarse pero aguarda a ser ejecutada.
- *Estado de bloqueo*, cuando la tarea no es posible de ejecutarse porque está en espera de la ocurrencia de un evento o de algún recurso no disponible de momento.
- *Estado durmiente*, si la tarea no es requerida por el sistema en tiempo real.

El *planeador de tareas* del sistema operativo de tiempo real es el responsable de controlar la transición de las tareas entre los distintos estados. En particular, existen dos estrategias con las que el planeador asigna el tiempo de procesamiento a varias tareas que deben realizarse en el mismo momento.

- *Asignación por round robin.* Aquí todas las tareas tienen igual prioridad, y la lista de tareas en espera de ejecución se configura como una cola de primeras entradas/primeras salidas. El planeador da a cada tarea un tiempo de procesamiento igual, parando la ejecución de la tarea al cumplirse dicho tiempo, y asignándole los recursos de procesamiento a la primera tarea que está en lista de espera. El intervalo asignado a cada tarea es determinado por el reloj de tiempo real, el cual interrumpe al planeador a intervalos regulares. Cuando el planeador recibe tal interrupción, realiza un cambio de tarea, parando la tarea en ejecución e iniciando la siguiente tarea en la lista de espera (Figura 3.3).

- *Asignación basada en prioridad de la tarea.* En esta estrategia las tareas tienen asignadas distintas prioridades, de tal forma que la tarea con mayor prioridad que no esté en estado de bloqueo o durmiente siempre se estará ejecutando, hasta que termine de realizarse, o deba de esperar por un evento o un recurso. La asignación de prioridades puede ser estática, en donde se especifican las prioridades antes de que el sistema arranque; o dinámica, si se permite el cambio de las mismas ya que el sistema ha arrancado (Figura 3.4).

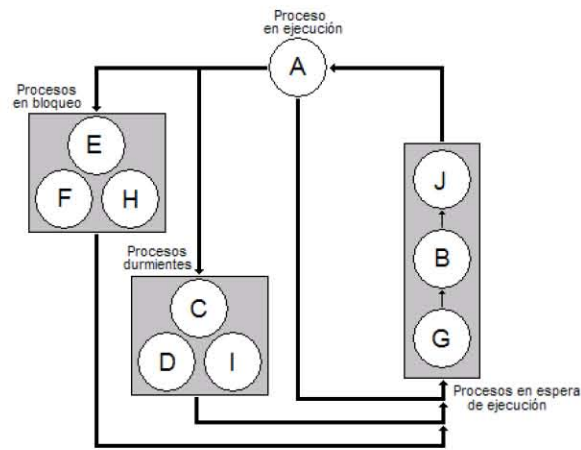


Figura 3.3: Estrategia Round Robin.

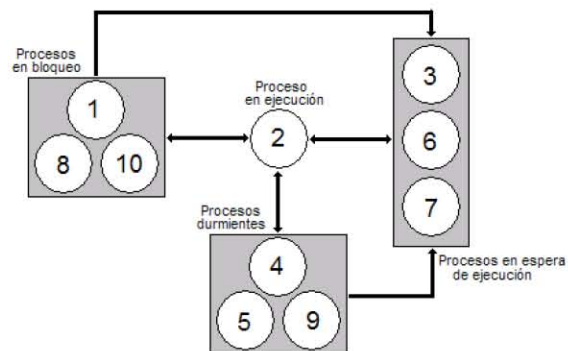


Figura 3.4: Estrategia por prioridades. El número indica la prioridad del proceso, con 1 como la prioridad máxima.

Una estrategia de asignación por prioridades puede representar un problema, porque *el proceso con la máxima prioridad puede monopolizar el tiempo de procesamiento si no se detiene nunca o se programa de tal suerte que no comparta dicho tiempo de procesamiento*. Es por ello que en la práctica se utiliza una combinación de las dos estrategias.

**Lenguajes de Programación de Alto Nivel y de Tiempo Real** La implantación del algoritmo de control en un sistema en tiempo real basado en computadora se realiza generalmente con la programación de la tarea o proceso respectivo en un lenguaje de alto nivel. Sin embargo, para que un lenguaje de alto nivel represente una verdadera posibilidad en la programación de un algoritmo de control, este lenguaje debe de cumplir con ciertas características.

*El requisito básico de un lenguaje de programación de alto nivel y de tiempo real es su capacidad de producir código objeto y código máquina de ejecución rápida y eficiente.* Esto a su vez se traducirá en un uso eficiente de la memoria, ya que programas que ocupan menor espacio de almacenamiento se ejecutarán más rápidamente. También, dentro de un mismo lenguaje pueden existir distintos compiladores que produzcan código objeto de eficiencia ampliamente variable. Por ello la elección correcta del lenguaje y de un compilador del mismo son de gran importancia.

*Sin embargo, en tareas cortas y simples, donde el programador pueda tomar ventaja de características especiales del problema o del procesador, el lenguaje ensamblador representa una ventaja en la obtención de código más eficiente.* Naturalmente la experiencia y habilidad del programador con el lenguaje ensamblador juega un papel relevante en esta situación, debido a que un programador promedio de lenguaje ensamblador generalmente no producirá código más eficiente que el obtenido por un buen compilador de lenguaje de alto nivel. *Igualmente, en proyectos grandes de programación, los cuales necesiten de la definición de cierta estructura y modularización del mismo en varias tareas, los lenguajes de alto nivel se desempeñarán de mejor forma.* Así, un sistema muy complejo puede ser dividido en procesos (o tareas) más pequeños, tales que puedan ser manejados más eficientemente para su mantenimiento y posible extensión en el futuro.

En específico, un lenguaje de alto nivel y de tiempo real debe de facilitar al programador:

- especificar el tiempo en el que las tareas o procesos deben ser realizados y terminados,
- responder en la forma deseada a las situaciones donde todos los requerimientos de tiempo no pudieron ser cumplidos,

- interactuar con hardware dedicado a la obtención de señales externas y a la generación de señales internas, y
- manipular cantidades reales y de punto flotante representativas de las variables físicas.

Por otro lado, *los lenguajes de programación de alto nivel y de tiempo real tienen la necesidad de crear software sin defectos y robusto*, dado que el sistema de control opera de forma autónoma y sin supervisión alguna del operador. Por ello es imprescindible verificar exhaustivamente el código para la búsqueda de fallas en la programación, y ello requerirá contar con herramientas que faciliten la depuración del código. Además, en algunos casos la operación del sistema de control está distribuida en distintas máquinas o procesadores, de tal manera que la ejecución de las tareas o procesos en un elemento del sistema debe hacerse en forma concurrente con las tareas o procesos en ejecución en otro elemento. Por lo tanto, el lenguaje de programación debe de facilitar la especificación y exhibición de esta concurrencia.

### 3.3. El Lenguaje Gráfico de Programación LabVIEW

LabVIEW<sup>MR 1</sup> es un lenguaje gráfico de programación de alto nivel desarrollado por National Instruments por primera vez en el año 1986. Lo que en un inicio se concibió como un entorno de programación dedicado a facilitar la construcción de sistemas basados en una computadora para la adquisición, análisis y presentación de datos, se ha convertido con el paso de los años en una herramienta muy útil en diversos campos, como son el control y la automatización de procesos, la realización de experimentos para pruebas a equipos e instrumentos, y la programación de procesadores digitales de señales, entre otras aplicaciones.

Todo programa desarrollado en LabVIEW se denomina también *instrumento virtual* (o en inglés *virtual instrument*), dado que el aspecto final de un programa en LabVIEW asemeja la operación de los instrumentos tradicionales, como osciloscopios, generadores de señales, multímetros digitales, etcétera. Por ello, *una de las grandes ventajas de este lenguaje es, además de su interfaz de programación gráfica, sencilla e intuitiva, su gran capacidad de integración de sistemas en una sola computadora*, en vez de un sistema seguramente complejo por el involucramiento de varios equipos interconectados para su funcionamiento.

---

<sup>1</sup>Acrónimo del inglés *Laboratory Virtual Instrument Engineers Workbench*.



Un instrumento virtual en LabVIEW consiste siempre en dos partes. La primera de ellas es el *panel frontal*, la cual generalmente forma la interfaz de comunicación con el usuario, ya que el usuario es capaz de especificar entradas al programa a través de *controles*, y observar los resultados obtenidos a través de *indicadores*. La otra parte la constituye el *diagrama de bloques*, en donde se especifica el funcionamiento del programa por medio de *bloques*<sup>2</sup> y la conexión entre los mismos, de tal manera que el orden de ejecución del programa queda definido de acuerdo a la forma en que los datos fluyen dentro del programa. En la Figura 3.5 se muestra un ejemplo de un programa realizado en LabVIEW 7.1, versión del lenguaje utilizada en este trabajo, donde se señalan los elementos antes mencionados.

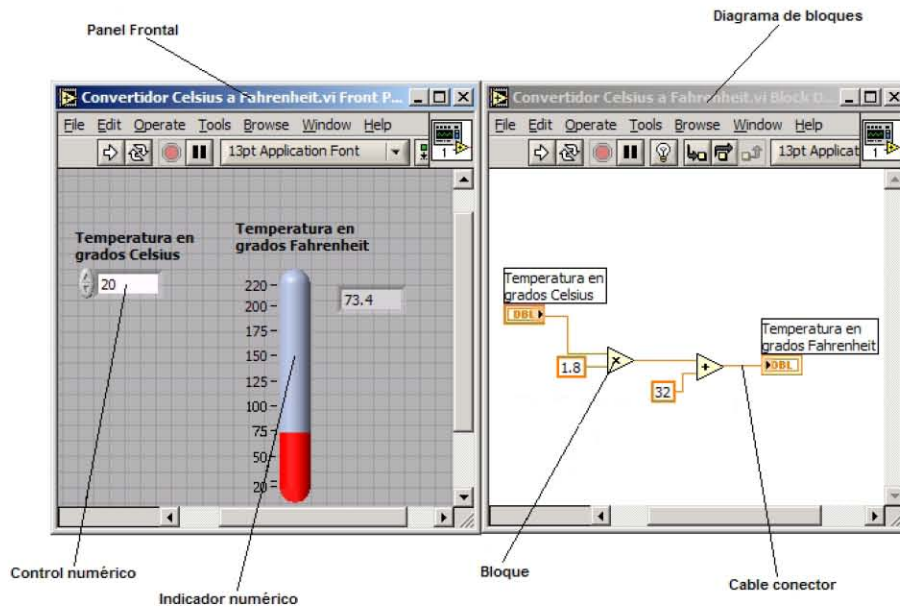


Figura 3.5: Ejemplo de un convertidor de grados Celsius a grados Fahrenheit, programado en LabVIEW 7.1.

LabVIEW<sup>MR</sup> Real-Time es un conjunto de plataformas de desarrollo que trabajan con LabVIEW como lenguaje de programación para la implantación de sistemas en tiempo real. Todas consisten en un software de desarrollo común (LabVIEW 7.1 y LabVIEW Real-time Module 7.1, para nuestro caso) a todas las distintas plataformas de hardware provistas por National Instruments, como son las series de tarjetas de tiempo real PCI, PXI, y los módulos de tiempo real

<sup>2</sup>En LabVIEW 7.1, un bloque es un elemento de programación que desempeña una tarea en específico, como una suma o una resta.

FieldPoint<sup>MR</sup> y Compact RIO<sup>MR</sup>. En específico, para la aplicación desarrollada se escogió el hardware PCI-7041/6040E, perteneciente a la serie de tarjetas de tiempo real PCI. Este equipo fue adquirido por el Departamento de Control de la Facultad de Ingeniería de la UNAM, y es apropiado para la implantación de proyectos de tiempo real pequeños o experimentales en una computadora personal.

### 3.4. Descripción de la Plataforma de Desarrollo LabVIEW Real-Time 7.1 y PCI 7041/6040E

Todas las plataformas de desarrollo de LabVIEW Real-Time están basadas en una misma arquitectura de hardware y de software. Una computadora personal funciona como *sistema de desarrollo*<sup>3</sup>, en donde se diseña y programa la aplicación en tiempo real (en nuestro caso, el algoritmo de un controlador); y desde donde se puede probar y depurar la misma operación, que se ejecuta ya en una *tarjeta objetivo de tiempo real*. *El medio por el cual se comunican el sistema de desarrollo y la tarjeta de tiempo real es, en este caso en particular, la memoria compartida*<sup>4</sup>.

En la siguiente tabla se resumen las características y funciones que desempeñan tanto el sistema de desarrollo como la tarjeta de tiempo real.

Sistema de desarrollo (Máquina anfitrión)	Tarjeta objetivo de tiempo real (PCI-7041/6040E)
Es el entorno para la programación, prueba y depuración de la aplicación.	Aquí se ejecuta la aplicación.
Es un equipo que funciona bajo un sistema operativo multitarea, como Window <sup>MR</sup> o Linux.	Es un equipo que funciona bajo un sistema operativo de tiempo real <sup>5</sup> .
Software: LabVIEW <sup>MR</sup> 7.1 y LabVIEW <sup>MR</sup> Real-Time Module 7.1.	Software: LabVIEW <sup>MR</sup> Real-Time Engine 7.1

El LabVIEW Real-Time Module 7.1 proporciona todos los mecanismos para la descarga de la aplicación en la tarjeta objetivo de tiempo real, así como de comunicación entre la tarjeta y el sistema de desarrollo en las etapas de prueba y depuración del código. También contiene ciertas herramientas y elementos de

<sup>3</sup>También denominada máquina anfitrión, o "host".

<sup>4</sup>Memoria a la cual tienen acceso la máquina anfitrión y la tarjeta de tiempo real.

<sup>5</sup>La PCI-7041 utiliza en sistema operativo en tiempo real de Venturcom Phar Lap Embedded Tool Suite (ETS).

programación propios de una aplicación en tiempo real. El LabVIEW Real-Time Engine 7.1 es una versión especial de LabVIEW capaz de ejecutar la aplicación desarrollada en un equipo que trabaje con un sistema operativo de tiempo real, que en nuestro caso es la tarjeta PCI-7041/6040E.

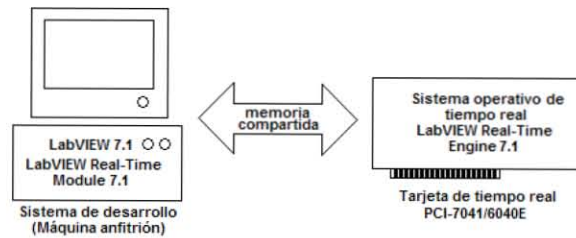


Figura 3.6: Arquitectura de la plataforma de desarrollo de sistemas en tiempo real con LabVIEW.

### La Tarjeta de Tiempo Real PCI7041/6040E

La PCI-7041/6040E es una tarjeta conectada al bus PCI de la máquina anfitrión, integrada por dos subtarjetas permanentemente unidas:

- la *tarjeta procesadora 7041*, y
- la *tarjeta multifunción de adquisición de datos 6040E*.



Figura 3.7: Tarjeta de tiempo real PCI-7041/6040E.

La parte procesadora contiene los mismos elementos básicos de una computadora personal cualquiera: un procesador, buses de control, de datos y de

información, memoria volátil y no volátil. *Es en esta parte donde la aplicación de tiempo real se ejecuta, de forma completamente independiente y continua, mientras que la tarjeta reciba alimentación eléctrica*<sup>6</sup>. Ya que en el procesador se ejecuta un sistema operativo de tiempo real, la aplicación opera de acuerdo a las prioridades especificadas por el programador, resultando lo anterior en un alto grado de determinismo de la aplicación.

La tarjeta multifunción de adquisición de datos (DAQ, por sus siglas en inglés) 6040E incluye medios de conectividad con 16 entradas analógicas, 2 salidas analógicas, 1 puerto digital de 8 bits, y señales de temporizador y/o contador. Para entradas y salidas analógicas, la tarjeta trabaja con un amplificador de instrumentación de ganancia programable, un multiplexor, y un convertidor analógico/digital y otro digital/analógico. Dado que está conectada permanentemente a la tarjeta procesadora, entonces es posible operar en la aplicación de tiempo real sobre señales externas reales.

## El Software de LabVIEW

**LabVIEW<sup>MR</sup> Real-Time Module 7.1** LabVIEW Real-Time Module 7.1 extiende el lenguaje gráfico de programación de LabVIEW a la construcción de aplicaciones en tiempo real. Con él se programa en el mismo entorno característico de LabVIEW, además de que agrega elementos de programación típicos de las aplicaciones en tiempo real (como el acceso al reloj de tiempo real o asignación de prioridades, por ejemplo). También realiza la descarga automática del programa del sistema de desarrollo a la tarjeta, además de proveer la comunicación entre ambos sistemas para la prueba y depuración del código de la aplicación.

**LabVIEW<sup>MR</sup> PID Control Toolset** Este conjunto de herramientas incluye bloques programados en LabVIEW destinados a la creación de controladores PID, de lógica difusa, o con base en algoritmos avanzados de control.

**LabVIEW<sup>MR</sup> Real-Time Engine 7.1** Ésta es una versión de LabVIEW 7.1, diseñada para ejecutarse sobre un sistema operativo de tiempo real. Con ello se asegura que la ejecución de la aplicación cumpla con las restricciones en tiempo y prioridades especificadas por el programador.

---

<sup>6</sup> Así, a pesar de la máquina anfitrión falle y detenga su operación, la aplicación que se está ejecutando en el momento en la tarjeta de tiempo real continuará ejecutándose.

### 3.5. Construcción Típica de un Sistema de Control en Tiempo Real con LabVIEW Real-Time 7.1 y PCI-7041/6040E

#### 3.5.1. Conexión de Señales Analógicas a la Tarjeta PCI-7041/6040E

##### Tipos de Fuentes de Señales Analógicas

Las señales analógicas pueden ser de dos tipos:

- *señales analógicas de fuente flotante, o*
- *señales analógicas de fuente aterrizada.*

En las fuentes flotantes, el voltaje de la señal no está referido a ninguna tierra común, tal como la tierra física o la tierra del edificio. Algunos ejemplos de fuentes flotantes de señales analógicas son las baterías, los termopares, los transformadores o los amplificadores de aislamiento. En una fuente aterrizada, el voltaje de la señal está referido a una tierra del sistema, como la tierra física o la tierra del edificio. Ejemplos comunes de este tipo de fuentes son aquellos equipos que se conectan al enchufe en una pared, como los generadores de señales y las fuentes de poder.

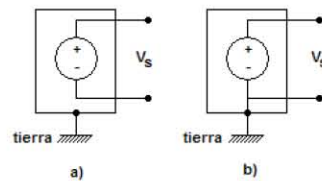


Figura 3.8: Fuentes de voltaje a) flotante y b) referida a tierra.

##### Clasificación de Sistemas de Medición

Como ya se mencionó, la PCI-7041 tiene integrada una tarjeta multifunción 6040E para la adquisición de datos. En la medición de señales analógicas,

algunos elementos presentes en la tarjeta de adquisición de datos se ven involucrados: multiplexores, un amplificador de instrumentación y un convertidor analógico/digital. La 6040E posee 16 canales de entrada analógica, todos conectados a un multiplexor que a su vez está conectado a un solo amplificador de instrumentación de ganancia programable. Finalmente, un convertidor analógico/digital transforma un nivel de voltaje a una palabra digital, apta para ser procesada por la computadora. Existen tres modos de emplear el amplificador de instrumentación, a fin de medir una señal analógica de entrada, que a su vez caracterizan al sistema de medición empleado:

- modo *diferencial*,
- modo *común referido*, y
- modo *común no referido* (o también conocido como *modo pseudodiferencial*).

El modo *diferencial* es parecido a las señales de tipo *flotantes*, ya que la medición se hace con respecto a una tierra flotante distinta a la tierra del sistema de medición. Ninguna de las entradas de un sistema de medición diferencial está atada a una referencia fija, tal como la tierra física. La 6040E implementa un sistema de medición diferencial de hasta 8 entradas, tal como se muestra en la Figura 3.9.

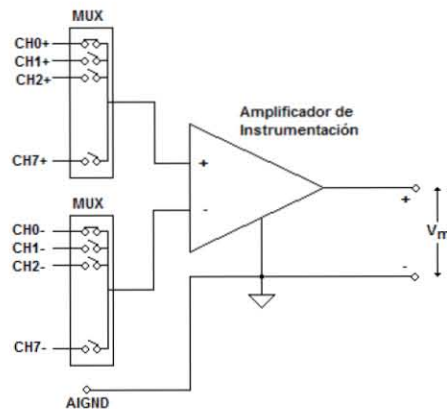


Figura 3.9: Sistema de medición diferencial de 8 entradas.

Un sistema de medición diferencial ideal mide solamente la diferencia de potencial entre sus terminales (+) y (-), rechazando totalmente cualquier voltaje presente en ambas entradas del amplificador de instrumentación respecto a la

tierra del mismo (es decir, rechaza el voltaje en modo común). Esto es muy útil, ya que el ruido se presenta generalmente como voltaje de esta forma. Sin embargo, los sistemas reales no presentan un rechazo absoluto del voltaje en modo común; y al contrario se mide su mayor grado de rechazo conforme es mayor su *factor de rechazo de modo común*<sup>7</sup>.

En un sistema de medición en modo común referido todas las señales son medidas respecto a una misma tierra, que en este caso es la tierra física, misma tierra del sistema de medición. En la Figura 3.10 se muestra esta configuración para la 6040E, donde *AIGND* representa la tierra del sistema de medición.

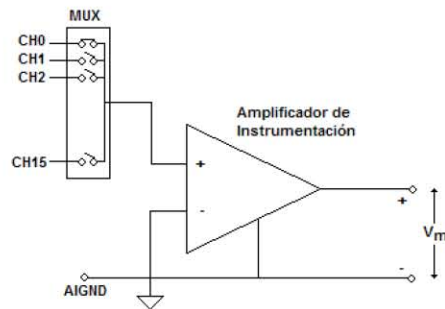


Figura 3.10: Sistema de medición de hasta 16 canales, en modo común referido.

Algunos dispositivos de adquisición de datos ocasionalmente usan el modo común no referido, que es una variación del modo común referido. En un sistema de medición de este tipo, todas las señales son medidas respecto a una referencia común (*AISENSE*, para la 6040E); pero el potencial en este nodo puede variar con respecto a la tierra del sistema de medición (*AIGND*).

Una señal flotante puede ser medida con cualquiera de las configuraciones anteriormente descritas, aunque se prefiere el modo diferencial ya que reduce el efecto del ruido presente como voltaje en modo común. Para una señal aterrizada debe evitarse el uso del modo común referido, debido a que la señal se degradará a causa de la presencia de lazos de corriente en tierra, dada la real diferencia entre los potenciales de la tierra de la señal y la tierra del sistema.

<sup>7</sup>En inglés CMRR, o Common-Mode Rejection Ratio, que se calcula por medio de  $CMRR(dB) = 20 \log\left(\frac{\text{Ganancia Diferencial}}{\text{Ganancia en Modo Común}}\right)$ .

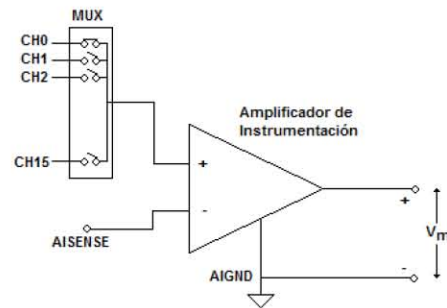


Figura 3.11: Sistema de medición de hasta 16 canales en modo común no referido.

Configuración de entrada	Tipo de señal	
	Señal de fuente flotante	Señal de fuente aterrizada
Diferencial		
Común referido		<p><b>NO RECOMENDADA</b></p> <p>Pérdidas por lazos en tierra, Vg, son adheridas a la señal medida.</p>
Común no referido		

Figura 3.12: Resumen de conexiones de entrada analógica.



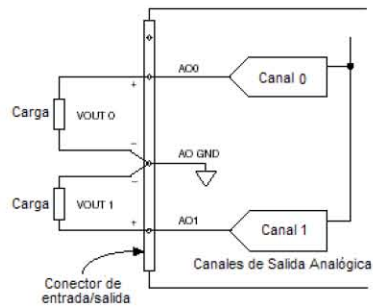


Figura 3.13: Conexiones para salida analógica.

### Generación de Señales Analógicas de Salida

La 6040E puede generar señales de voltaje en dos modos.

- *Generación punto a punto*, en donde el nivel de la señal es más importante que la razón a la cual cambia dicha señal. Así, la tarjeta actualiza el valor del voltaje de salida a un nuevo nivel, en el momento en que sea necesario hacer esto.
- *Generación con buffer*, la cual permite producir señales en donde tanto la forma de onda como la razón de cambio de la señal son de igual importancia. Ejemplo de ellas es una señal senoidal.

La Figura 3.13 muestra la forma de conectar señales analógicas de salida a cargas externas a la 6040E. *A00* y *A01* son los nombres de los dos canales de salida analógica de la tarjeta de adquisición de datos, y en ambas la referencia es el nodo *A0GND*.

### 3.5.2. Buenas Técnicas de Programación de una Aplicación en Tiempo Real

Cuando una computadora es el centro de la operación de un sistema de control, es posible que realice varias tareas al mismo tiempo, más allá de la simple implantación del controlador. Por ejemplo, la computadora podría también ser habilitada en la generación de reportes del funcionamiento del sistema completo, o en la recolección y almacenamiento de datos reales para su posterior análisis. *Sin embargo, siempre existirán prioridades entre las distintas tareas que ejecute la computadora, siendo algunas de importancia crítica y otras no crítica.* Por lo

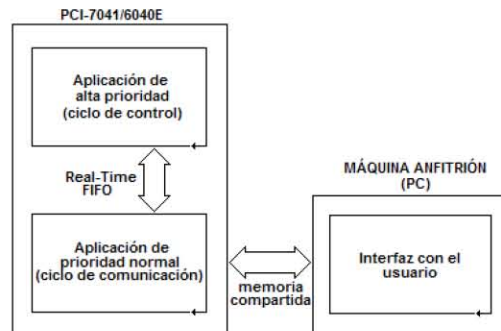


Figura 3.14: Esquema de la programación de un sistema de control en tiempo real en LabVIEW Real-Time.

tanto, la programación de todas las tareas debe dejar bien claro cuales son las prioridades asignadas a cada una de ellas; a fin de que las tareas de alta prioridad, que generalmente deben de cumplir con cierta restricción en el tiempo, se cumplan en el plazo determinado.

*Para la implantación de un sistema de control en tiempo real, el tiempo de ejecución del ciclo de control es un factor crítico, y por ello la programación debe asegurar que dicho ciclo posea la más alta prioridad. En LabVIEW Real-Time existen varias formas de cumplir con la idea anterior. Aquí se explicará la más común de ellas en LabVIEW 7.1, esquemáticamente representada en la Figura 3.14.*

Generalmente se especifican tres rutinas o ciclos, dos de ellos ejecutándose en la tarjeta de tiempo real PCI-7041/6040E, y uno en la máquina anfitrión. En la tarjeta de tiempo real trabajan paralelamente

- un *ciclo de control*, y
- un *ciclo de comunicación*.

En la máquina anfitrión se construye la *interfaz con el usuario*.

El ciclo de control contiene todas aquellas operaciones propias del controlador, y tiene asignada la más alta prioridad. Así se asegura la correcta ejecución del controlador, dentro del límite de tiempo especificado para ello (determinismo). Paralelamente trabaja un ciclo de comunicación, cuya principal función es ser puente para la transmisión de datos entre el ciclo de control y la máquina anfitrión. Dichos datos representan información del estado del proceso, de la magnitud de la acción de control emprendida, y del valor de la referencia; información que es de natural interés conocer para el usuario. *Sin embargo, el*

*conocimiento de esta información en tiempo real puede no ser del todo necesario, y al contrario podría afectar al determinismo del ciclo de control. Por ello la transmisión de estos datos tiene asignada una prioridad más baja.*

La comunicación entre ambos ciclos se realiza con un *buffer de primeras entradas y primeras salidas (FIFO)*, conocido como *Real-Time FIFO*. El buffer es un espacio en memoria, de cierto tamaño fijo, y en donde los primeros datos que se van escribiendo en él también son los primeros en leerse. Así, mientras se asegure que el buffer no se llene<sup>8</sup>, será posible recuperar los datos que en un momento determinado no pudieron ser leídos.

La máquina anfitrión ejecuta la interfaz con el usuario, donde es posible variar el valor de la referencia y visualizar el estado del proceso, y la magnitud y forma del esfuerzo de control. Igualmente, este programa puede realizar un análisis sobre los datos, almacenarlos en una unidad de memoria de alta densidad, y generar reportes de la operación del proceso. La transmisión de los datos se realiza directamente con el ciclo de comunicación, ejecutándose en la tarjeta de tiempo real. *El medio para este intercambio de datos es la memoria compartida.*

## 3.6. Herramientas de Tiempo Real del LabVIEW Real-Time Module 7.1

En esta sección se comentarán brevemente algunas de las herramientas del LabVIEW Real-Time Module 7.1 especialmente diseñadas para la programación de tareas de tiempo real, como las encontradas en un sistema de control de lazo cerrado o en cualquier otro tipo de sistema.

### 3.6.1. Estructura *Timed Loop*

Para aplicaciones que requieren ejecutar múltiples tareas de tiempo crítico en paralelo, se utiliza la estructura<sup>9</sup> *Timed Loop*. *El Timed Loop (o Lazo Temporizado) opera como un ciclo while común, además de contar con algunas características adicionales referentes al tiempo de ejecución de cada iteración y a propiedades típicas de tiempo real.* Así, dentro de un *Timed Loop* se especifican las acciones o instrucciones a seguir dentro de una tarea. En un mismo VI

<sup>8</sup>Ya que el buffer tiene un tamaño lo suficientemente grande, o la velocidad de lectura de los datos es lo suficientemente rápida.

<sup>9</sup>En LabVIEW<sup>MR</sup>, se denominan estructuras a los ciclos iterativos o a las sentencias de control de la ejecución, de las rutinas escritas en un lenguaje de programación estructurado, como son: ciclos while y for, sentencia case, etcétera.

pueden incluirse hasta 128 ciclos *Timed Loop*, cada uno relacionado a una diferente tarea con prioridad y periodo de ejecución único. Así mismo, recuperan la información concerniente a la ejecución de la última iteración, con lo que es posible verificar y realimentar el estado del comportamiento en tiempo real de la tarea.

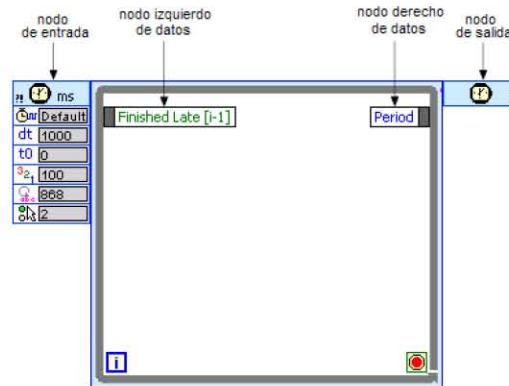


Figura 3.15: Estructura *Timed Loop*.

La estructura *Timed Loop* está formada por:

- un *nodo de entrada*,
- un *nodo izquierdo de datos*,
- un *nodo derecho de datos*, y
- un *nodo de salida*.

En el nodo de entrada es posible configurar al ciclo, de acuerdo con los valores suministrados a las entradas del nodo, o bien por medio de los valores proporcionados en la ventana de diálogo, mostrada en la Figura 3.16. En el nodo izquierdo de datos se proporciona información acerca del tiempo de ejecución y el estado de la iteración previa, teniéndose la opción de conectar estos datos izquierdos al nodo derecho de datos para poder configurar futuras iteraciones. El nodo de salida regresa los errores recibidos en el nodo de entrada, o genera un mensaje de error si existe algún problema en la ejecución del ciclo.

Dentro de las opciones de configuración del ciclo *Timed Loop* están:

- la base de tiempo a utilizar,

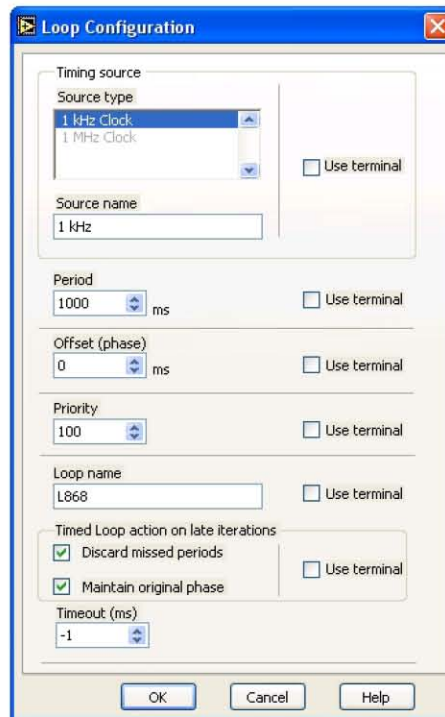


Figura 3.16: Ventana de diálogo del nodo de entrada del *Timed Loop*.

- el periodo y offset<sup>10</sup> de la ejecución de cada iteración,
- la prioridad<sup>11</sup>,
- en nombre del ciclo, y
- el modo de operación.

### 3.6.2. Comunicación entre Tareas con Real-Time FIFOs

Una forma posible de transferir datos entre distintas tareas o programas es a través de *variables globales o locales*. Sin embargo, este medio de intercambio de datos es vulnerable a la pérdida de los mismos, ya que se pueden realizar muchas escrituras antes de hacer una lectura, por citar sólo un ejemplo. En

<sup>10</sup>El tiempo que el ciclo espera para iniciar la iteración, respecto a la referencia de tiempo inicial de ejecución.

<sup>11</sup>Entre mayor sea el número, mayor es la prioridad.

el caso de las variables globales, un solo VI puede acceder a la variable en un cierto instante, con lo que bloquea el alcance de esa variable a otros programas, a pesar de contar éstos con mayor prioridad. *Lo anterior provocaría la inversión en el cumplimiento de la prioridad y consecuentemente aumentaría el jitter.*

*Con los Real-Time FIFOs, una lectura y una escritura pueden ser realizadas en el mismo tiempo.* Además, el Real-Time FIFO actúa como una cola, de tal suerte que los nuevos datos almacenados no sobrescriben a aquellos elementos ya presentes; a menos que el FIFO se llene, lo que sobrescribiría el dato más viejo. Por lo tanto, mientras la lectura de los datos en el FIFO sea lo suficientemente rápida o el tamaño del mismo sea grande, no se perderán los datos viejos.

Un Real-Time FIFO es fácil de usar. Su uso involucra la creación del FIFO en un VI y el paso de la referencia del mismo a otro VI. Después, las escrituras al FIFO son realizadas en un VI, mientras las lecturas son hechas en el otro programa. Finalmente, el FIFO es borrado. La Figura 3.17 muestra un ejemplo sencillo en la utilización del FIFO, desde su creación hasta su borrado, aunque aquí el dato es transferido desde y hacia el mismo VI.

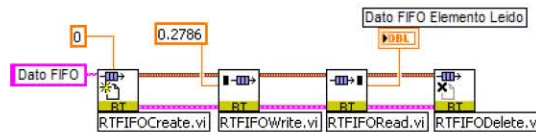


Figura 3.17: Ejemplo de uso del Real-Time FIFO.

### 3.6.3. Comunicación entre la Tarjeta PCI y la Máquina Anfitrión con Memoria Compartida

Para el intercambio de datos entre una tarjeta PCI y la máquina anfitrión se requiere el uso de la *memoria compartida*, que es un sector del bus PCI al cual tienen acceso directo los procesadores de ambos sistemas (la PCI-7041/6040E tiene 512 [KB] de memoria compartida). En específico existen dos métodos para la lectura y escritura de datos en la memoria compartida, ambos aplicables a arreglos de variables de precisión simple<sup>12</sup>:

- el método de *lectura y escritura simple*, y
- el método de *lectura y escritura incremental*.

<sup>12</sup>En LabVIEW, representación de 16 bits para números reales o de punto flotante.

La diferencia básica entre ambos métodos puede inferirse de sus nombres, tomando en cuenta que generalmente las instrucciones respectivas se desarrollan dentro de un ciclo iterativo (*while* o *for*), con motivo de lograr una transferencia continua de la información. En el primero, se lee y/o escribe un arreglo completo de datos de precisión simple en una sola iteración; mientras que en el segundo se requerirán varias iteraciones para alcanzar el mismo objetivo, ya que el arreglo se lee o escribe en partes e incrementalmente (dato por dato), según sea el caso.

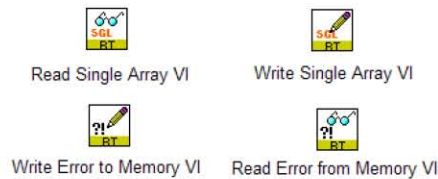


Figura 3.18: Algunos VIs que operan en la memoria compartida, en el modo de lectura y escritura simple.

### 3.7. LabVIEW Execution Trace Toolkit 7.1

El LabVIEW<sup>MR</sup> Execution Trace Toolkit 7.1 es un software para la medición del tiempo de ejecución y la observación del desempeño de un programa ejecutándose en tiempo real. En general, podemos medir cuánto tiempo se lleva la realización del programa, así como verificar el cumplimiento de la condición de tiempo real y el comportamiento del sistema conforme a las prioridades asignadas a cada tarea dentro del programa.

Este software incluye a los bloques (*Execution Trace Tool VIs*) para la captura y carga de las sesiones, y un visor (*LabVIEW Execution Trace Tool*, Figura 3.19) donde se analizarán tales sesiones. Los VIs se agregan al diagrama de bloques del programa a estudiar. En la Figura 3.20 se muestra un ejemplo de cómo utilizar estos bloques. El *TraceTool Start Trace VI* inicia la captura; posteriormente se comienza la ejecución de las tareas a analizar, y finalmente el *TraceTool Stop Trace and Send VI* detiene la captura y manda la sesión a la computadora con dirección IP indicada en la terminal *Trace Host Network Address*. Así, el visor mostrará los resultados de la captura, y en él podrán medirse los tiempos de ejecución y el desempeño total del sistema.

La ventana del *LabVIEW Execution Trace Tool* está dividida en dos partes:

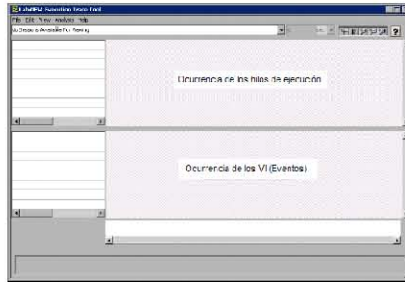


Figura 3.19: LabVIEW Execution Trace Tool.

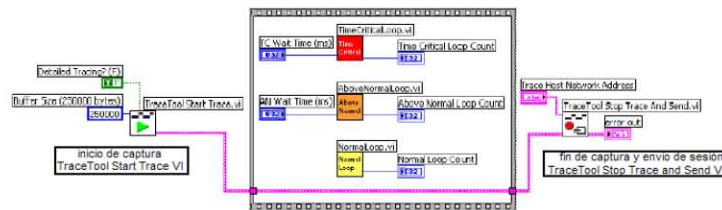


Figura 3.20: Utilización típica de los bloques *TraceTool Start Trace VI* y *TraceTool Stop Trace and Send VI*, del *Execution Trace Tool VIs*.

- en la superior se observa la *ocurrencia (o eventos) de los hilos de ejecución*<sup>13</sup>, y
- en la inferior la *ocurrencia (o eventos) de los VIs*.

Así, la parte superior muestra la ejecución de los hilos en el sistema operativo de tiempo real de la tarjeta objetivo. A la izquierda se muestran el nombre de cada hilo, precedido por un número identificador de espacio en memoria, y también se especifica el sistema de ejecución donde el hilo corre. Cada hilo se despliega en un color diferente, para distinguir la prioridad de ejecución. Este color coincide también con las tonalidades de las prioridades en la vista de ocurrencia de eventos.

La vista de eventos muestra todos los VIs cargados en la memoria de la tarjeta de tiempo real, listados en la parte izquierda e identificados también por el número de espacio en memoria asignado al VI por el sistema operativo

<sup>13</sup>Los *hilos de ejecución* son la forma en que un programa se divide así mismo en dos o más tareas que corren simultáneamente. Varios hilos pueden ser ejecutados en paralelo en muchos sistemas de computadora. Generalmente esto ocurre a partir de *dividir el tiempo de procesamiento* (donde un solo procesador cambia continuamente entre los diferentes hilos) o por *procesamiento múltiple* (cuando los hilos son ejecutados en distintos procesadores).



de tiempo real. Se observa también cuándo estos VIs se ejecutaron respecto al tiempo, cuya escala va de izquierda a derecha.

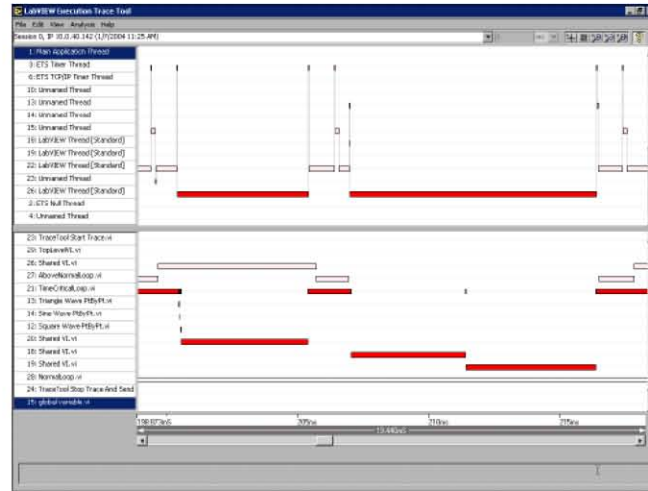


Figura 3.21: Aspecto de una sesión con el LabVIEW Execution Trace Toolkit 7.1, de una aplicación en tiempo real.

## Capítulo 4

# Diseño e Implantación del Controlador Virtual Determinístico por Modelo Interno

En este capítulo se tratará el diseño e implantación del Controlador Virtual Determinístico por Modelo Interno (CVDMI). Primero se presentarán los diseños de Un Grado y Dos Grados de Libertad del sistema de control. Posteriormente se describirán la construcción y programación del sistema, buscando el desempeño en tiempo real. Para ello se ha hecho uso de LabVIEW y LabVIEW Real-Time Module 7.1, los cuales nos permitirán una construcción modular, versátil, y con una interfaz de usuario muy amigable.

La planta a controlar constará del equipo simulador de procesos PCS327. A continuación se dará una breve descripción de este equipo.

### 4.1. Simulador de Procesos PCS327

La planta PCS327 es, por un lado, un equipo simulador procesos de sistemas de primer, segundo y hasta tercer orden, con polos repetidos y ajustables a través de interruptores (expresados por constantes de tiempo  $\tau$  iguales a 10 [ms] o 1[s]), o ubicados en el origen. También se pueden simular procesos con retardo en el tiempo  $T$ , o adicionar perturbaciones a la entrada o salida del proceso. Por otro lado, es una simulador de controladores PID, donde pueden ajustarse por

separado las constantes de cada una de las tres estrategias: proporcional, integral y derivativa. También permite simular características no lineales, como saturación, zona muerta, histéresis, entre otras. Así, es posible simular el comportamiento de sistemas de control PID, con la posibilidad de variar distintos parámetros y ver el efecto de estas modificaciones en el desempeño del sistema de lazo cerrado. La Figura 4.1 muestra este equipo.

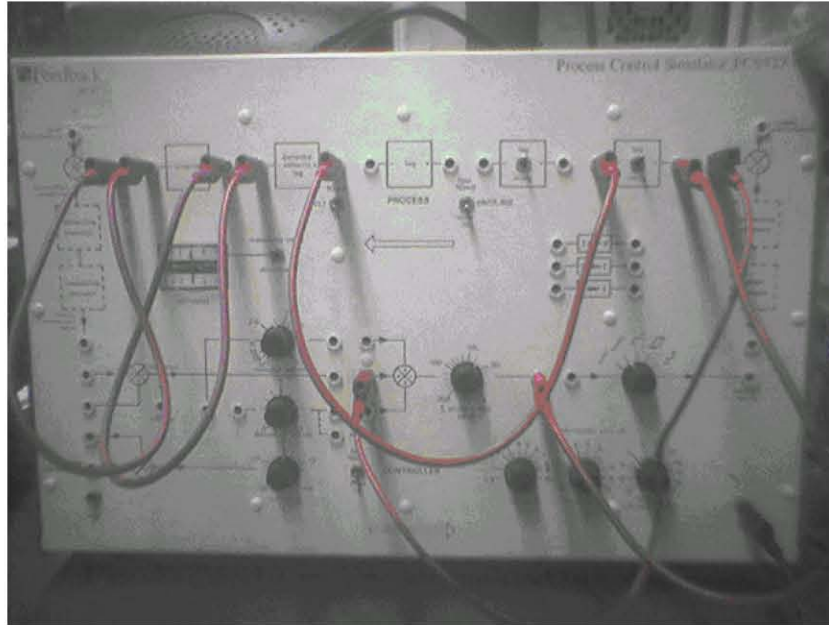


Figura 4.1: Simulador de procesos PCS327.

Internamente la planta PCS327 está formada por amplificadores operacionales, en distintas configuraciones, los cuales al conectarse en cascada definen un sistema en particular. El simulador es alimentado con un voltaje de 120 [VAC]; además contiene varios nodos de tierra física.

## 4.2. Ajuste del Controlador Virtual Determinístico Por Modelo Interno

Para controlar se necesita primero un proceso. En el caso del diseño de un sistema bajo el esquema de Control por Modelo Interno, se necesita contar primero con una descripción matemática del proceso: su modelo y su incertidumbre.

### 4.2.1. Modelo e Incertidumbre del Proceso a Controlar

Para el diseño de los controladores  $q_r(s)$  y  $qq_d(s)$ <sup>1</sup>, partimos de la descripción del proceso siguiente:

$$p(s) = \frac{K}{(s+1)^2} e^{-Ts}; 0.5 \leq K \leq 2.0, 0.01 \leq T \leq 1 \text{ [seg]}. \quad (4.1)$$

Como se observa el proceso es un sistema de segundo orden con polos repetidos (críticamente amortiguado), con rangos de incertidumbre tanto en la ganancia,  $K$ , como en el tiempo de retardo,  $T$ .

### 4.2.2. Ajuste de Un Grado de Libertad

Para el ajuste del controlador  $q_r(s)$  en un sistema de control de Un Grado de Libertad, se necesita proponer primero un modelo del proceso,  $\tilde{p}(s)$ , y la parte de éste a invertir. Utilizando el programa IMCTUNE<sup>2</sup>, desarrollado en Matlab, se realizaron los cálculos de la constante de tiempo del controlador,  $\varepsilon_r$ , para el cumplimiento de la especificación de amplificación máxima de ruido en altas frecuencias y del ajuste  $M_P$ , como también de las curvas máxima y mínima de la función de sensibilidad complementaria,  $CS$ , para estimar tanto la respuesta más rápida como la más lenta.

#### Primer Diseño de Un Grado de Libertad

En este primer diseño, se tomó aquel modelo del proceso,  $\tilde{p}(s)$ , en donde los valores de la ganancia,  $K$ , y del tiempo de retardo,  $T$ , están ubicados a la mitad del rango de incertidumbre. Es decir,

$$\tilde{p}(s) = \frac{1.25}{(s+1)^2} e^{-0.5s}. \quad (4.2)$$

Sea el controlador  $q_r(s)$  aquél obtenido al invertir la ganancia  $K = 1.25$  y los dos polos repetidos  $(s+1)^2$ . Cumplir con la especificación de amplificación de ruido a altas frecuencias en  $q_r(s)$  requiere de un valor de la constante de tiempo del filtro,  $\varepsilon_r$ , igual a 0.22; por lo tanto

$$q_r(s) = \frac{(s+1)^2}{1.25(0.22s+1)^2}. \quad (4.3)$$

Para el ajuste, considerando un valor de  $M_P = 1.05$ , se obtiene la expresión del controlador

<sup>1</sup>Según se trate de un esquema de Uno o Dos Grados de Libertad.

<sup>2</sup>Para consultar un guía sobre este programa, véase el Apéndice G de Brosilow y Joseph.

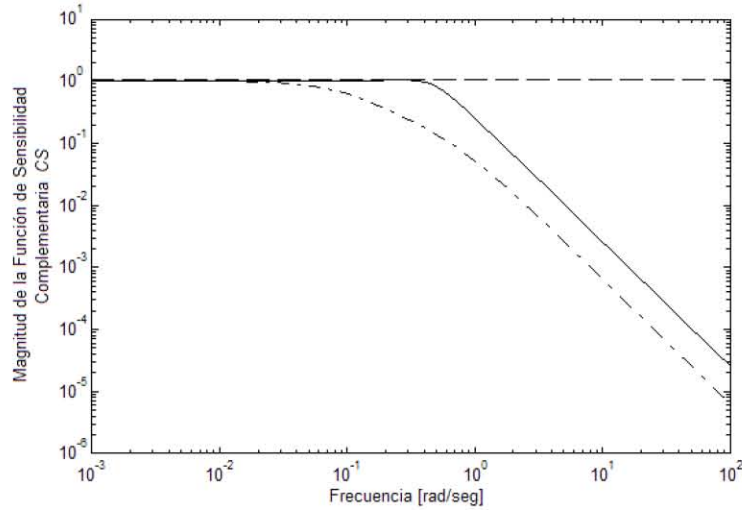


Figura 4.2: Magnitud máxima y mínima de la función de sensibilidad complementaria, para el primer diseño de Un Grado de Libertad.

$$q_r(s) = \frac{(s + 1)^2}{1.25(2.47s + 1)^2}. \quad (4.4)$$

En la Figura 4.2 se muestran las curvas de magnitud máxima y mínima de la función de sensibilidad complementaria,  $CS$ , para el esquema de Un Grado descrito por las ecuaciones (4.2) y (4.4). De esta misma Figura, obtenemos que el máximo  $M_P = 1.05$  ocurre en la frecuencia  $\omega_{M_p} = 0.27 \text{ [rad/seg]}$ , con los parámetros del proceso  $K = 2$  y  $T = 1 \text{ [seg]}$ .

Las respuestas más rápida y más lenta del sistema se obtienen de los valores de los parámetros del proceso,  $p(s)$ , en la recta asintótica a altas frecuencias de la curvas máxima y mínima de magnitud, respectivamente. Los valores de  $K$ ,  $T$ , y la frecuencia de corte,  $\omega_{corte}$ , en estos dos casos son:

	$K$	$T \text{ [seg]}$	$\omega_{corte}$
<b>Respuesta más rápida</b>	2	0.24	0.51[rad/seg]
<b>Respuesta más lenta</b>	0,5	0.01	0.25[rad/seg]

Las Figuras 4.3 y 4.4 muestran la respuesta de la variable del proceso,  $y(t)$ , a entradas escalón unitario en la referencia,  $r(t)$ , y en la perturbación,  $d(t)$ , cuando esta última actúa directamente sobre la salida del proceso. También son visibles los esfuerzos de control,  $u(t)$ , aplicados por el controlador  $q_r(s)$ .

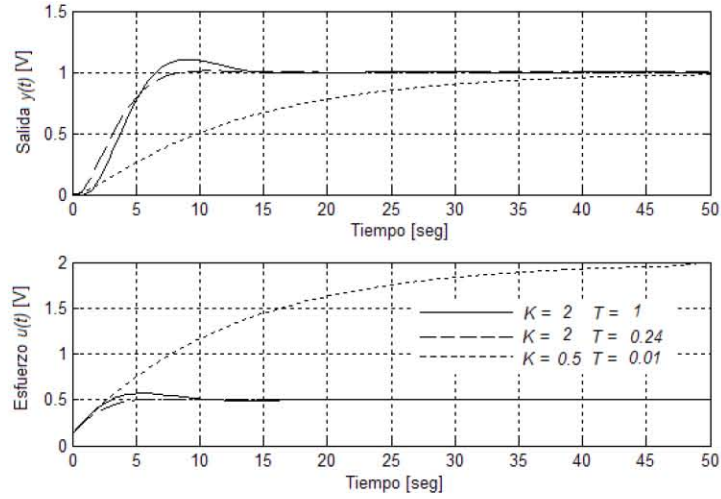


Figura 4.3: Respuesta a una entrada en la referencia, del sistema correspondiente al primer diseño de Un Grado de Libertad.

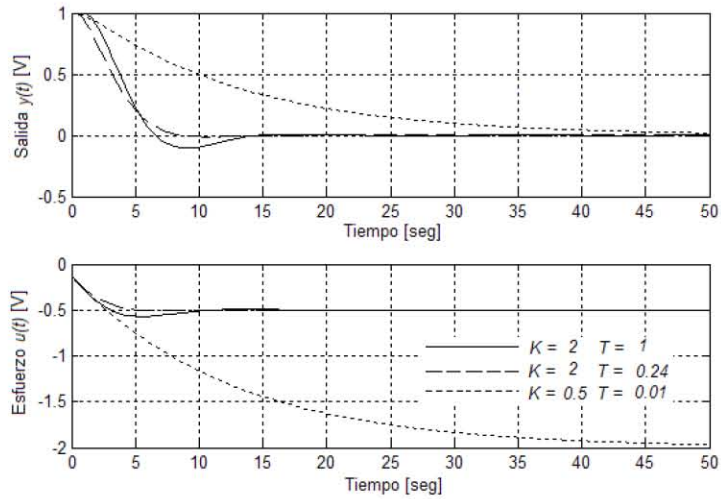


Figura 4.4: Respuesta a una entrada en la perturbación, del sistema correspondiente al primer diseño de Un Grado de Libertad.

## Segundo Diseño de Un Grado de Libertad

Ahora se considera el mismo modelo,  $\tilde{p}(s)$ , del primer diseño,

$$\tilde{p}(s) = \frac{1.25}{(s+1)^2} e^{-0.5s}. \quad (4.5)$$

Sea el controlador  $q_r(s)$  aquél obtenido al invertir la ganancia y sólo un polo, cuya constante de tiempo es igual a la suma de las constantes de los polos repetidos:  $(2s+1)$ . *Es decir, se buscó acelerar las respuestas de todo el sistema de control al utilizar una expresión de menor grado para el controlador CMI, en combinación con el modelo del proceso dado por (4.5).* Para cumplir con la especificación de amplificación de ruido a altas frecuencias en  $q_r(s)$ , se requiere de un valor de la constante de tiempo del filtro,  $\varepsilon_r$ , igual a 0.10; por lo tanto

$$q_r(s) = \frac{(2s+1)}{1.25(0.10s+1)}. \quad (4.6)$$

En el ajuste del máximo de la función de sensibilidad complementaria en  $M_P = 1.05$ , se obtiene la expresión de controlador

$$q_r(s) = \frac{(2s+1)}{1.25(3.30s+1)}. \quad (4.7)$$

Para los diseños primero y segundo, se muestran las curvas de magnitud máxima y mínima de la función de sensibilidad complementaria en la Figura 4.5. En ella es posible apreciar cómo las trazas máxima y mínima de  $CS$  para el segundo diseño están ubicadas más hacia la derecha que las correspondientes al primer diseño; *por lo tanto esto provocará respuestas más rápidas en el sistema definido por las ecuaciones (4.5) y (4.7).* De esta misma figura, obtenemos que el máximo  $M_P$  para un valor de 1.05 ocurre en la frecuencia  $\omega_{M_p} = 0.54$  [rad/seg], con los parámetros del proceso  $K = 2$  y  $T = 1$ [seg].

Los valores de  $K$ ,  $T$ , y la frecuencia de corte,  $\omega_{corte}$ , para las respuestas más rápida y más lenta son:

	$K$	$T$ [seg]	$\omega_{corte}$
<b>Respuesta más rápida</b>	2	0.08	0.97[rad/seg]
<b>Respuesta más lenta</b>	0.5	0.01	0.51[rad/seg]

Finalmente, las Figuras 4.6 y 4.7 muestran, para este segundo diseño, la respuesta de la variable del proceso,  $y(t)$ , a entradas escalón unitario en la referencia,  $r(t)$ , y en la perturbación,  $d(t)$ , cuando esta última actúa directamente sobre la salida del proceso. También se presentan los esfuerzos de control,  $u(t)$ , aplicados por el controlador  $q_r(s)$ . En estas mismas gráficas es posible apreciar cómo la respuesta del sistema basado en el segundo diseño es más rápida que la correspondiente al sistema basado en el primer diseño; por ejemplo, en la

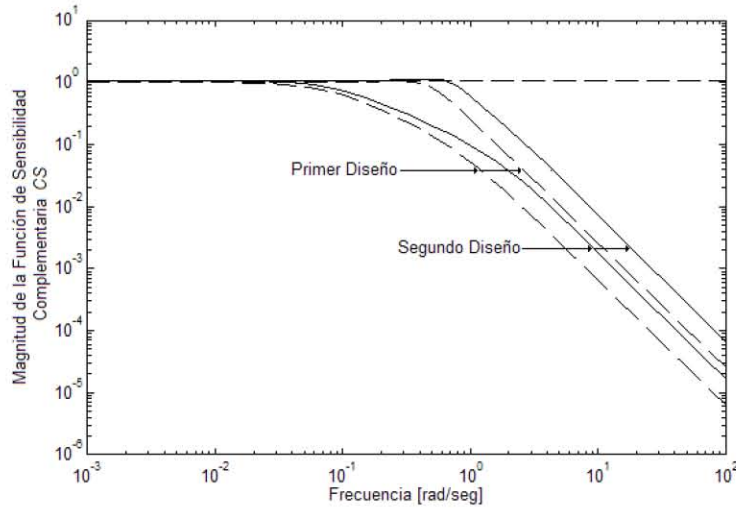


Figura 4.5: Máxima y mínima magnitud de  $CS$ , para los diseños primero y segundo de Un Grado de Libertad.

respuesta para los parámetros del proceso  $K = 2$  y  $T = 1[\text{seg}]$ , tanto en la respuesta a la referencia como en la respuesta a la perturbación. Las frecuencias de corte corroboran también este hecho.

### Tercer Diseño de Un Grado de Libertad

En este diseño se mantienen los grados del modelo,  $\tilde{p}(s)$ , y del controlador,  $q_r(s)$ , del segundo diseño. Sin embargo, elijeremos los valores de  $K$  y  $T$  iguales al límite superior de los rangos de incertidumbre respectivos, con la idea de acelerar aún más la respuesta del sistema. Así, el modelo es

$$\tilde{p}(s) = \frac{2}{(s+1)^2} e^{-s}, \quad (4.8)$$

y el controlador,  $q_r(s)$ , que cumple con la condición de amplificación a altas frecuencias es

$$q_r(s) = \frac{(2s+1)}{2(0.10s+1)}. \quad (4.9)$$

En el ajuste de  $M_P = 1.05$  se requiere que  $\varepsilon_r = 0.81$ ; cumpliéndose este máximo a una frecuencia  $\omega_{corte} = 0.44 [\text{rad/seg}]$ , para valores de los parámetros  $K$  y  $T$  de 2 y 1[seg], respectivamente. Entonces,



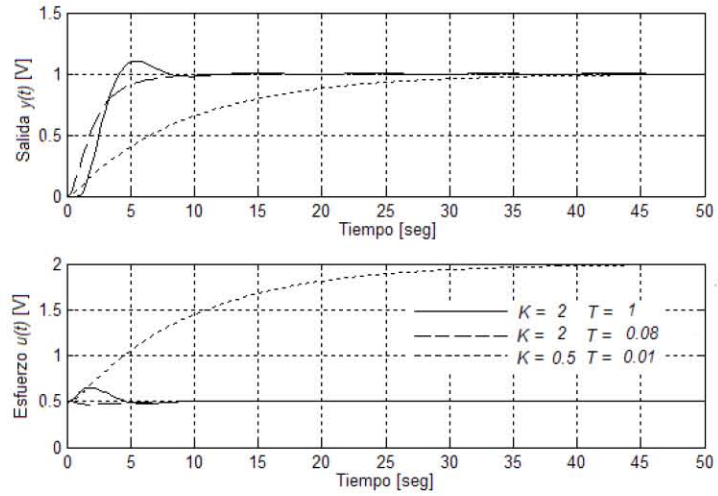


Figura 4.6: Respuesta a una entrada en la referencia, del sistema correspondiente al segundo diseño de Un Grado de Libertad.

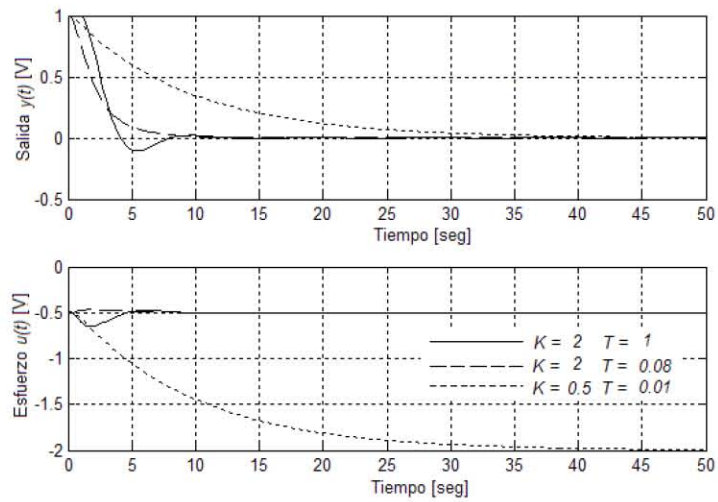


Figura 4.7: Respuesta a una entrada en la perturbación, del sistema correspondiente al segundo diseño de Un Grado de Libertad.

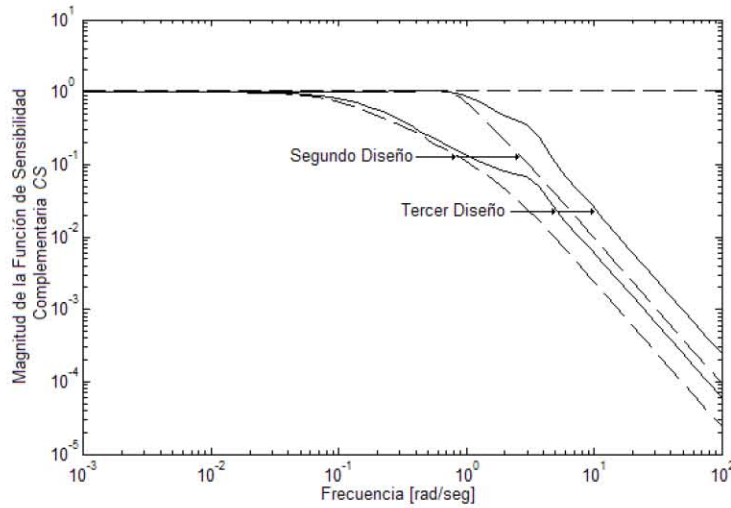


Figura 4.8: Máxima y mínima magnitud de  $CS$ , para los diseños segundo y tercero de Un Grado de Libertad.

$$q_r(s) = \frac{(2s + 1)}{2(0.81s + 1)}. \quad (4.10)$$

La Figura 4.8 presenta las curvas de  $CS$  para los diseños segundo y tercero. *Nuevamente es posible apreciar cómo este último diseño produce que dichas curvas estén ubicadas más hacia la derecha; ello implicará que el sistema responderá más rápidamente a entradas en la referencia.*

En el tercer diseño, los valores de  $K$ ,  $T$ , y la frecuencia de corte,  $\omega_{corte}$ , para las respuestas más rápida y más lenta son:

	$K$	$T$ [seg]	$\omega_{corte}$
<b>Respuesta más rápida</b>	2	0.03	1.55 [rad/seg]
<b>Respuesta más lenta</b>	0.5	0.01	0.77 [rad/seg]

Las Figuras 4.9 y 4.10 muestran la respuesta de la variable del proceso,  $y(t)$ , a entradas escalón unitario en la referencia,  $r(t)$ , y en la perturbación,  $d(t)$ , para este tercer diseño. Nuevamente la perturbación está presente directamente a la salida del proceso. También se presentan los esfuerzos de control,  $u(t)$ , aplicados por el controlador  $q_r(s)$ . Si se centra la atención en la respuesta del sistema, para este tercer diseño, cuando  $K = 2$  y  $T = 1$ [seg], se logrará apreciar cómo el sistema reacciona más rápidamente a la referencia y a la perturbación, en

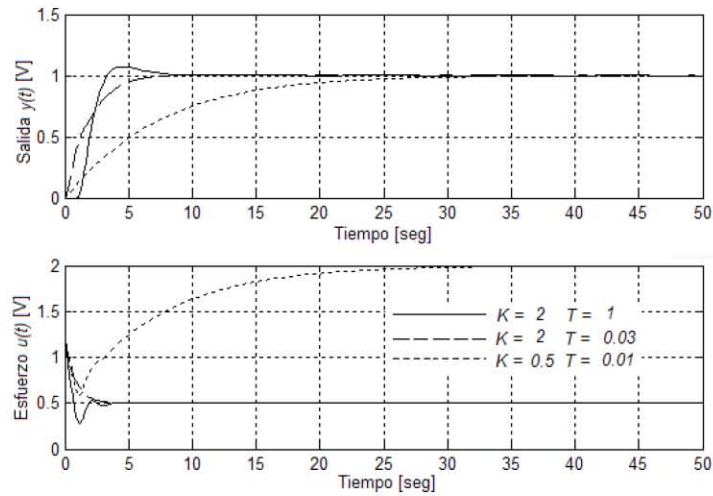


Figura 4.9: Respuesta a una entrada en la referencia, del sistema correspondiente al tercer diseño de Un Grado de Libertad.

comparación del caso correspondiente al segundo diseño. También, los valores de las frecuencias de corte verifican este comportamiento.

Comparando el desempeño de los tres diseños, se observa que el último produce las respuestas más rápidas, independientemente de la incertidumbre del proceso. Sin embargo, *este comportamiento requiere también de esfuerzos de control más agresivos (obsérvese la magnitud del esfuerzo de control en el tiempo inicial), y quizá un actuador real no sería capaz de seguir estas acciones*. En la implantación del sistema, la planta trabaja directamente con las señales de voltaje proporcionadas por la tarjeta; así, a pesar de que una acción de control pueda variar rápida y repentinamente su valor, el sistema podrá generar tales cambios sin mayor problema. Para la implantación del CVDMI, se escogió el tercer diseño.

### 4.2.3. Ajuste de Dos Grados de Libertad

El Control por Modelo Interno de Dos Grados de Libertad busca mejorar la respuesta del sistema a entradas en la perturbación,  $d(t)$ , cuando ésta no actúa directamente sobre la salida del proceso,  $p(s)$ . *Por ello, en esta sección se buscará lograr un diseño que muestre un mejor rechazo a la acción de la perturbación, estando ella presente a la entrada del proceso*.

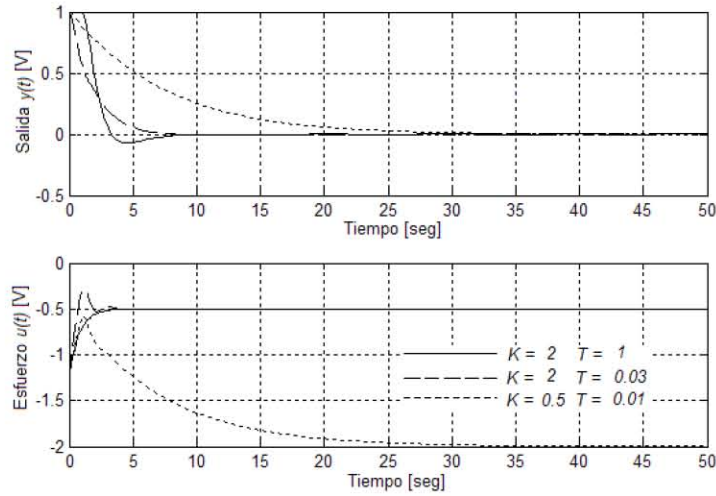


Figura 4.10: Respuesta a una entrada en la perturbación, del sistema correspondiente al segundo diseño de Un Grado de Libertad.

### Primer Diseño de Dos Grados de Libertad

En este primer diseño se tomará como expresión del modelo del proceso a

$$\tilde{p}(s) = \frac{2}{(s+1)^2} e^{-s}. \quad (4.11)$$

De aquí, se obtiene que las expresiones del controlador en la realimentación,  $qq_d(s)$ , para cumplir con la especificación de pico máximo  $M_P = 1.05$  y de oscilaciones no mayores a 0.1 en la curva máxima de la función de sensibilidad parcial  $PS$ , y del filtro de la referencia,  $q_r(s)$ , que cumple con los mismos dos criterios para la curva máxima de la función de sensibilidad complementaria,  $CS$ , son

$$qq_d(s) = \frac{(s+1)^2(0.89s^2 + 1.88s + 1)}{2(0.55s + 1)^4} \quad (4.12)$$

y

$$q_r(s) = \frac{(s+1)^2}{2(1.68s + 1)^2}. \quad (4.13)$$

Nótese cómo para el cálculo de  $qq_d(s)$  se consideró la eliminación de los dos polos de  $p_d(s) = \frac{1}{(s+1)^2}$ . La Figura 4.11 muestra las curvas mínimas de  $PS$  de este diseño y el mejor diseño de Un Grado de Libertad<sup>3</sup>, representado por las

<sup>3</sup>Recordemos que un sistema de Un Grado de Libertad puede verse como un caso particular de un sistema de Dos Grados de Libertad, donde  $qq_d(s) = q_r(s)$ .

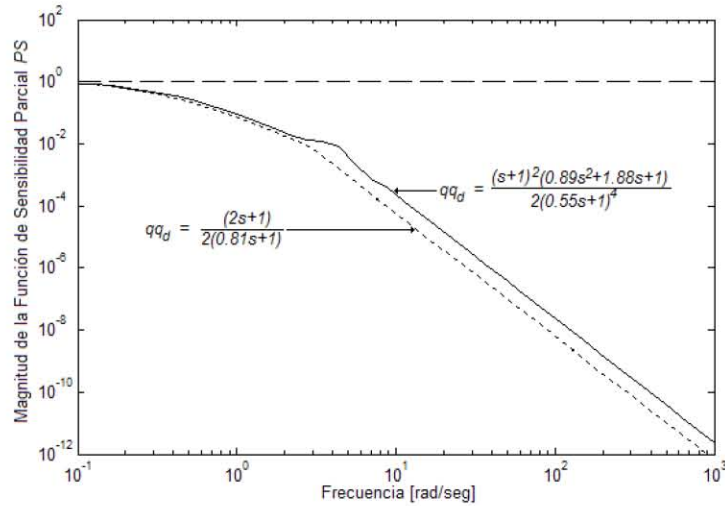


Figura 4.11: Mínima magnitud de  $PS$  de los sistemas de Un Grado de Libertad, expresado por las ecuaciones (4.8) y (4.10), y de Dos Grados de Libertad, representado por las ecuaciones (4.11) y (4.12).

ecuaciones (4.8) y (4.10).

El hecho de que la curva mínima de  $PS$  del sistema de Dos Grados de Libertad aquí expuesta esté más hacia la derecha supone un mejor rechazo a la perturbación. Esta idea se comprueba en la Figura 4.12, donde se observan las respuestas de estos dos sistemas a una entrada tipo escalón unitario en la perturbación,  $d(t)$ , para los parámetros del proceso,  $p(s)$ , que significan la respuesta más lenta ( $K = 0.5$  y  $T = 0.01$ ).

Finalmente, la Figura 4.13 muestra la respuesta de ambos sistemas a una entrada escalón unitario en la referencia,  $r(t)$ , para el mismo par de parámetros del proceso.

## Segundo Diseño de Dos Grados de Libertad

Ahora sea el modelo del proceso

$$\tilde{p}(s) = \frac{2}{(2s+1)}e^{-s}, \quad (4.14)$$

esto es, un modelo de primer orden. Las expresiones del controlador en la realimentación,  $qq_d(s)$ , para cumplir con la especificación de pico máximo  $M_P = 1.05$  y de oscilaciones no mayores a 0.1 en la curva máxima de la función de

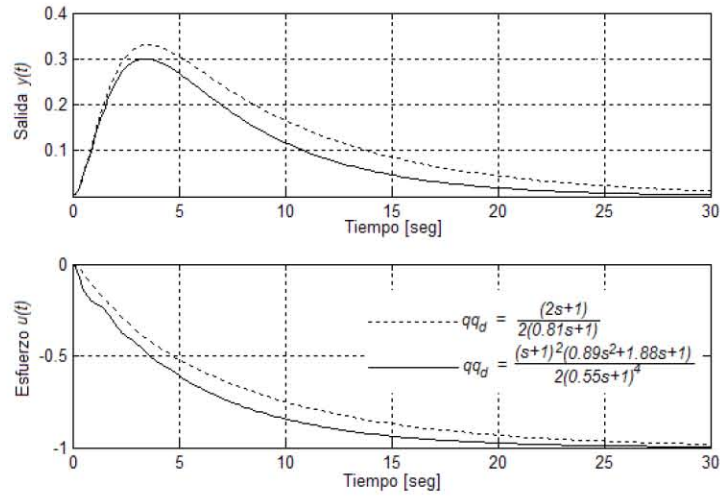


Figura 4.12: Respuesta a la perturbación de los sistemas de Un Grado de Libertad, expresado por las ecuaciones (4.8) y (4.10), y de Dos Grados de Libertad, representado por las ecuaciones (4.11) y (4.12).

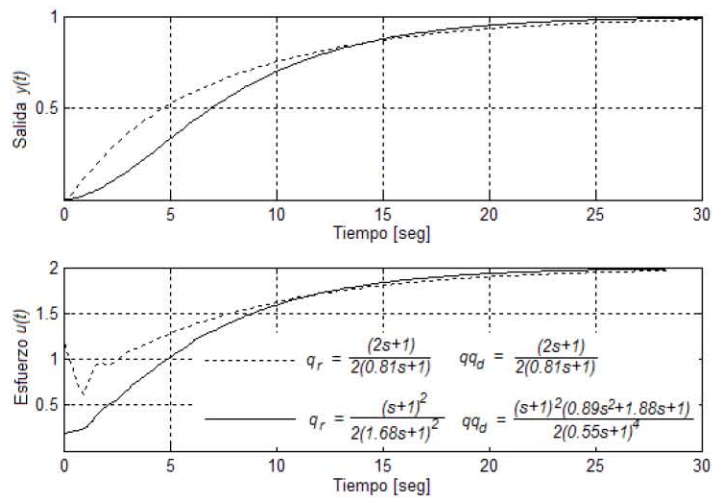


Figura 4.13: Respuesta a la referencia de los sistemas de Un Grado de Libertad, expresado por las ecuaciones (4.8) y (4.10), y de Dos Grados de Libertad, representado por las ecuaciones (4.11), (4.12) y (4.13).

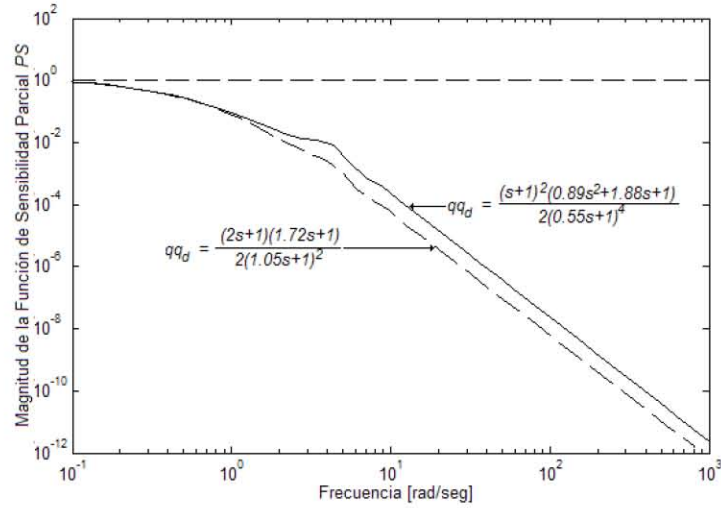


Figura 4.14: Mínima magnitud de  $PS$  de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones (4.11) y (4.12), (4.14) y (4.15).

sensibilidad parcial,  $PS$ , y del filtro de la referencia,  $q_r(s)$ , que cumple con los mismos dos criterios para la curva máxima de la función de sensibilidad complementaria,  $CS$ , son

$$qq_d(s) = \frac{(2s+1)(1.72s+1)}{2(1.05s+1)^2} \quad (4.15)$$

y

$$q_r(s) = \frac{(2s+1)}{2(2.50s+1)}. \quad (4.16)$$

Nótese cómo para el cálculo de  $qq_d(s)$  se consideró la eliminación de un solo polo de  $p_d(s)$ , cuya constante de tiempo es igual a la suma de las constantes individuales de los polos repetidos:  $(2s+1)$ . La Figura 4.14 muestra las curvas mínimas de  $PS$  de este diseño y el diseño anterior de Dos Grados de Libertad, representado por las ecuaciones (4.11) y (4.12). En la Figura 4.15 se observan las respuestas de estos dos sistemas a una entrada tipo escalón unitario en la perturbación,  $d(t)$ , para los parámetros del proceso,  $p(s)$ , que significan la respuesta más lenta ( $K = 0.5$  y  $T = 0.01$ ). Finalmente, la Figura 4.16 muestra la respuesta de ambos sistemas a una entrada escalón unitario en la referencia,  $r(t)$ , para el mismo par de parámetros del proceso.

Como es posible apreciar en las respuestas a la referencia y a la perturbación, estos dos sistemas presentan un comportamiento muy similar. Sin embargo, *el segundo diseño de Dos Grados de Libertad supone una implantación más*

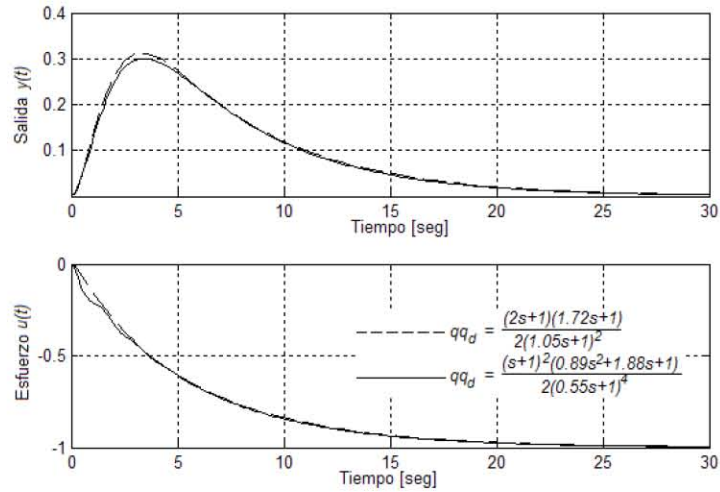


Figura 4.15: Respuesta a la perturbación de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones 4.11 y 4.12, y 4.14 y 4.15.

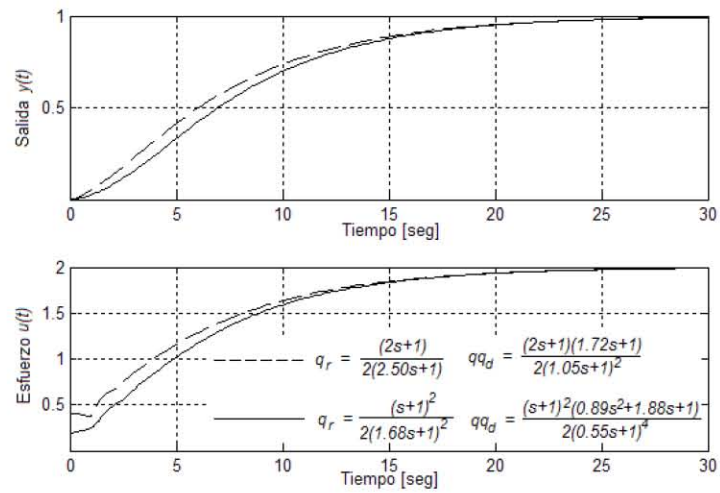


Figura 4.16: Respuesta a la referencia de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones (4.11), (4.12) y (4.13), y (4.14), (4.15) y (4.16).



sencilla, al contar con expresiones del modelo y de los controladores de orden reducido.

### Tercer Diseño de Dos Grados de Libertad

Volviendo al modelo de segundo orden

$$\tilde{p}(s) = \frac{2}{(s+1)^2} e^{-s}, \quad (4.17)$$

y retomando el mejor diseño de Un Grado de Libertad, hagamos  $qq_d(s) = q_r(s)$ , pero considerando valores distintos de las constantes de tiempo,  $\varepsilon$  y  $\varepsilon_r$ . Sean entonces

$$qq_d(s) = \frac{(2s+1)}{2(0.30s+1)} \quad (4.18)$$

y

$$q_r(s) = \frac{(2s+1)}{2(0.81s+1)}. \quad (4.19)$$

La Figura 4.17 muestra las curvas mínimas de  $PS$  de este diseño y el primer diseño de Dos Grados de Libertad, representado por las ecuaciones (4.11) y (4.12). En la Figura 4.18 se observan las respuestas de estos dos sistemas a una entrada tipo escalón unitario en la perturbación,  $d(t)$ , para los parámetros del proceso,  $p(s)$ , que significan la respuesta más lenta ( $K = 0.5$  y  $T = 0.01$ ). Finalmente, la Figura 4.19 muestra la respuesta de ambos sistemas a una entrada escalón unitario en la referencia,  $r(t)$ , para el mismo par de parámetros del proceso.

Este último diseño presenta el mismo rechazo a la perturbación que el primer diseño de Dos Grados de Libertad; pero también presenta una mejor respuesta a la referencia, como se destaca en la Figura 4.19. Además, las expresiones de los controladores son de menor orden, y por lo tanto de más fácil implantación. Por estas dos razones, éste es el diseño preferido para la estrategia de Dos Grados de Libertad.

### 4.3. Implantación del Controlador Virtual Determinístico por Modelo Interno

En esta sección se mencionará la construcción e implantación del Sistema Controlador Virtual Determinístico por Modelo Interno, de acuerdo con los

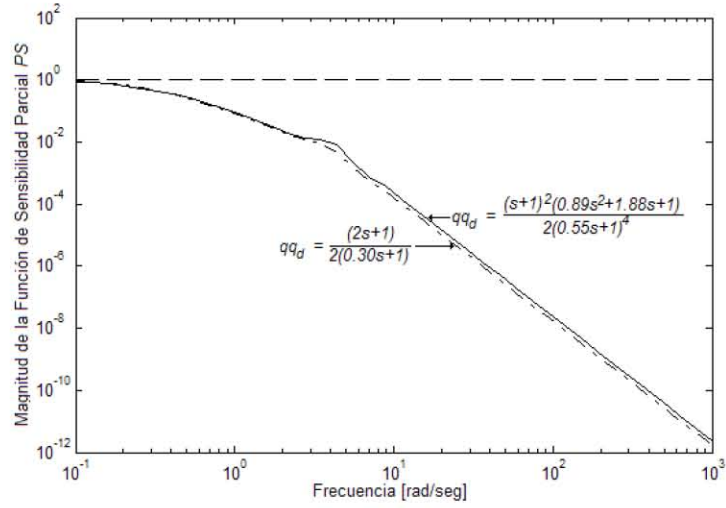


Figura 4.17: Mínima magnitud de  $PS$  de los sistemas de Dos Grado de Libertad, expresados por las ecuaciones (4.11) y (4.12), y (4.17) y (4.18).

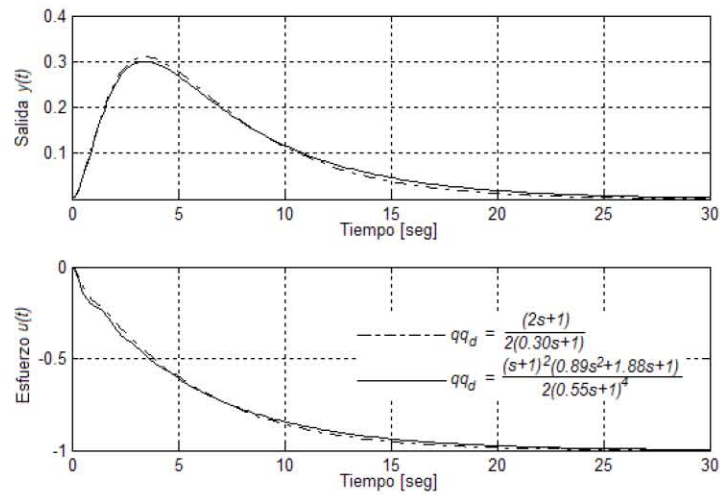


Figura 4.18: Respuesta a la perturbación de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones (4.11) y (4.12), y (4.17) y (4.18).

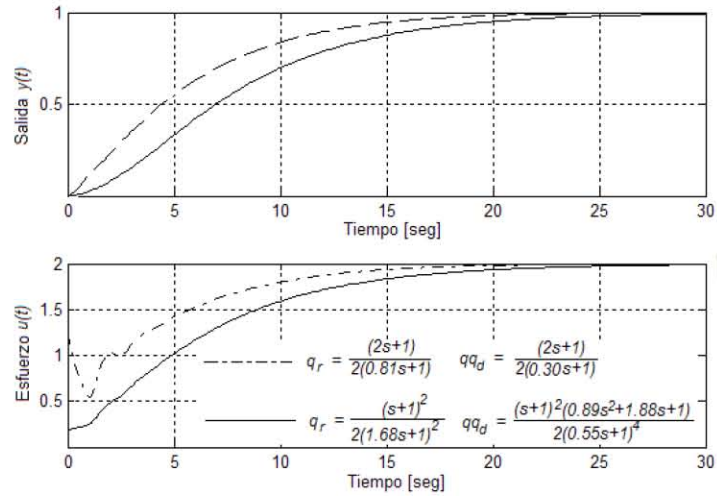


Figura 4.19: Respuesta a la referencia de los sistemas de Dos Grados de Libertad, expresados por las ecuaciones (4.11), (4.12) y (4.13), y (4.17), (4.18) y (4.19).

diseños de los esquemas de Control por Modelo Interno de Uno y Dos Grados de Libertad anteriormente obtenidos. Utilizando el software LabVIEW y LabVIEW Real-Time Module 7. 1, junto con la tarjeta de tiempo real PCI-7041/6040E, se explicarán y comentarán la programación de las distintas tareas del sistema: de aquellas ejecutándose en la tarjeta de tiempo real y de aquellas ejecutándose en la máquina anfitrion.

### 4.3.1. Programación de la Tarjeta de Tiempo Real

En la tarjeta de tiempo real se ejecutan dos tareas del CVDMI: el lazo de control y el lazo de comunicación. Igualmente, es necesario configurar dos acciones a realizar por la tarjeta multifunción de adquisición de datos de la PCI-7041/6040E: una de muestreo de una señal analógica y otra de generación del mismo tipo de señal, con el propósito de poder medir y conocer el estado actual del proceso controlado y de enviar la acción pertinente de control, respectivamente. También, se emplean los Real-Time FIFOs para el intercambio de datos entre los dos lazos, para evitar afectar el determinismo del lazo de control. Finalmente, y para obtener un desempeño más ordenado del sistema, se sincronizaron los tiempos de inicio para cada iteración de los lazos de control y de comunicación.

## Configuración de los Canales Analógicos de Entrada y de Salida

En LabVIEW se denomina *tarea (task)* a una actividad de medición o generación de una señal. De hecho, en nuestro sistema podemos ver al muestreo de la variable del proceso y a la escritura del esfuerzo de control como dos tareas adicionales.

Para configurar una tarea de medición, es necesario definir ciertas características de la misma, las cuales variarán de acuerdo con el tipo de señal (analógica o digital) con que se trabaje. Tratándose de la adquisición de una señal analógica, se debe especificar:

- la *terminal* o el canal físico a utilizar por la tarea,
- el *modo del canal* o el sistema de medición a utilizar<sup>4</sup>,
- el *rango* de la medición,
- la *frecuencia de muestreo*, y
- el *modo de muestreo*.

El rango de medición determina la ganancia del amplificador de instrumentación presente a la entrada de la 6040E. *El modo de muestreo establece si la tarea se realiza continuamente o si tiene una duración finita.* En el primer modo se obtiene sólo una muestra, la cual se lee de inmediato, y ello quita la necesidad de almacenar dicha muestra en memoria (un buffer). El segundo modo obtiene varias muestras antes de ser leídas, por lo que deben de ser almacenadas temporalmente en memoria. Para las generación de señales analógicas se aplican también las ideas anteriores, a excepción del modo del canal.

*Para aplicaciones de tiempo real, también se acepta el modo de muestreo Hardware-Timed Single Point, en donde se adquiere y lee, o se escribe, muestra a muestra (sin buffer) de la señal, utilizando como base de tiempo de la frecuencia de muestreo a una señal de reloj generada por hardware.* Este modo es comúnmente utilizado en aplicaciones de control en tiempo real, donde se requiere la entrada y salida de datos en un periodo determinístico de tiempo.

Siguiendo la discusión anterior, la tarea de adquisición para el muestreo de la variable de proceso,  $y(t)$ , tiene las siguientes propiedades.

<b>Terminal</b>	AI0
<b>Modo del Canal</b>	Diferencial
<b>Rango</b>	-10 a 10 [V]
<b>Frecuencia de Muestreo</b>	1 [kHz]
<b>Modo de Muestreo</b>	Hardware-Timed Single Point

---

<sup>4</sup>Diferencial, común referido y común no referido

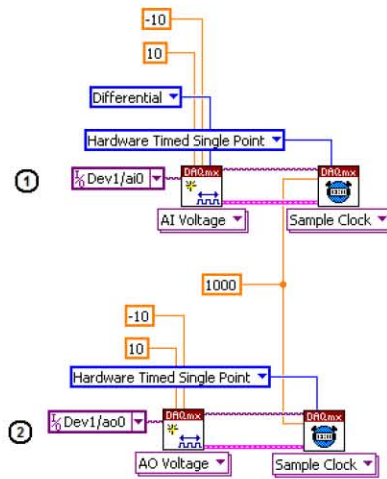


Figura 4.20: Configuración del canal para: (1) el muestreo de la variable del proceso, y (2) la escritura del esfuerzo de control.

La tarea para la generación del esfuerzo de control,  $u(t)$ , se configura de acuerdo a las siguientes características.

<b>Terminal</b>	AO0
<b>Rango</b>	-10 a 10 [V]
<b>Frecuencia de Muestreo</b>	1 [kHz]
<b>Modo de Muestreo</b>	Hardware-Timed Single Point

En la Figura 4.20 se muestra el diagrama de bloques correspondiente a la programación de las dos tareas mencionadas.

### Lazos de Control y Comunicación

La operación en tiempo real del CVDMI requiere de que el lazo de control se ejecute con una cierta periodicidad. Se desea que dicho lazo se realice en cada momento de adquisición de una nueva muestra de la variable de proceso,  $y(t)$ . *Por lo tanto, la frecuencia de muestreo de la tarea de adquisición anteriormente expuesta definirá el periodo de ejecución de interés de nuestro ciclo de control.* Se debe recordar que dicha frecuencia era igual a 1 [kHz], por lo que el periodo de ejecución del lazo de control es de 1 [ms].

Para programar esta idea en LabVIEW 7.1, utilizando la estructura *Timed Loop*, se requiere del bloque *DAQmx Create Timing Source VI* en su instancia *Sample Clock*. Este bloque crea una fuente de tiempo para el *Timed Loop*

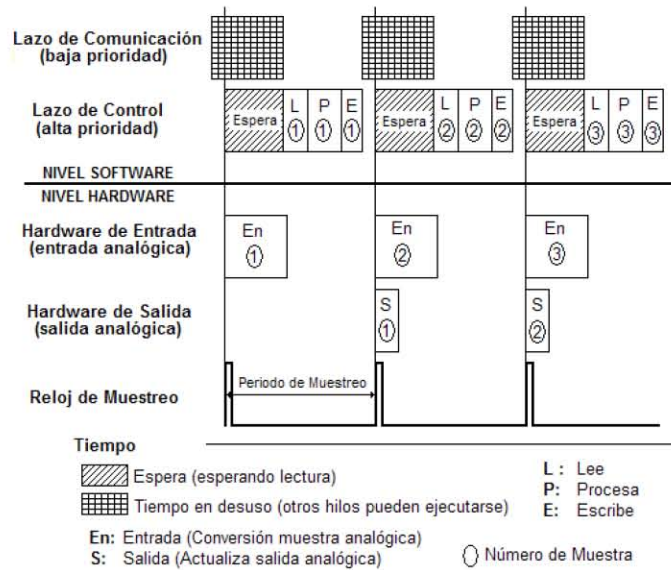


Figura 4.21: Diagrama de tiempo del Sistema Controlador Virtual Determinístico por Modelo Interno.

que contiene la operaciones de entrada/salida y del algoritmo de control; así la presencia de una nueva muestra de la variable del proceso iniciará la siguiente iteración de este ciclo. Esta misma base de tiempo se utiliza para el *Timed Loop* que contiene al ciclo de comunicación, conforme a lo mostrado en la Figura 4.22.

La conexión de bloques expuesta en la Figura 4.22 presenta otras ventajas. Primero los *Timed Loops* informan sobre si la iteración anterior se completó en el tiempo indicado. Por otro lado, el valor de salida (que representa al esfuerzo de control,  $u(t)$ ) se actualiza a la llegada de los pulsos de reloj que controlan el muestreo de la variable del proceso,  $y(t)$  (véase la Figura 4.21), por lo que la lectura de una nueva muestra y la escritura de el nuevo esfuerzo están sincronizadas. Por último, el bloque *Synchronize Timed Loops Starts VI* sincroniza los tiempos de inicio de cada iteración de los ciclos de control y comunicación. Sin embargo, la alta prioridad del lazo de control (200, sobre 100 del lazo de comunicación) asegura que éste se ejecute primero, y mientras espera al inicio de la siguiente iteración permite que el ciclo de comunicación realice sus operaciones.

En las Figuras 4.23 y 4.24 se muestran nuevamente los esquemas de Control por Modelo Interno de Uno y Dos Grados de Libertad, en donde también se señala la parte de los mismos (marcada dentro del área sombreada) que debió implantarse dentro del lazo de control.

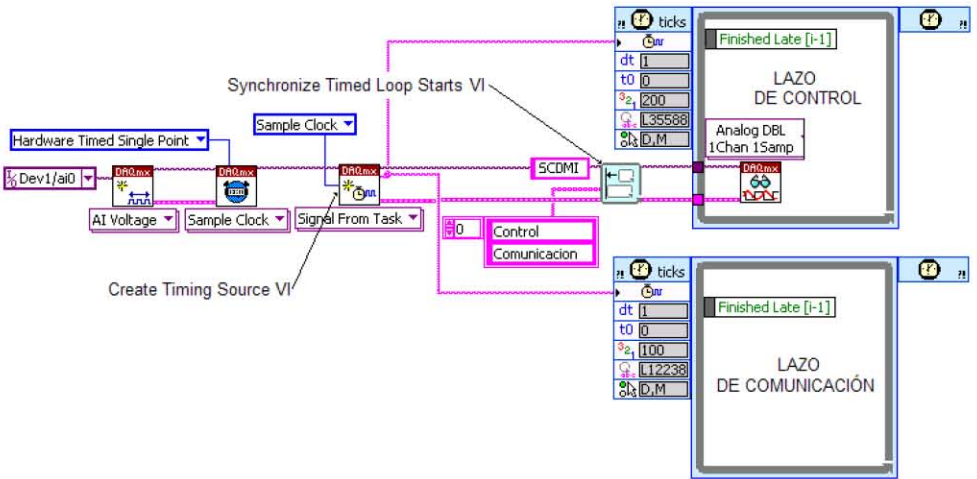


Figura 4.22: Conexión de bloques, para asegurar un comportamiento determinado del lazo de control, de acuerdo a la frecuencia de muestreo de la variable del proceso.

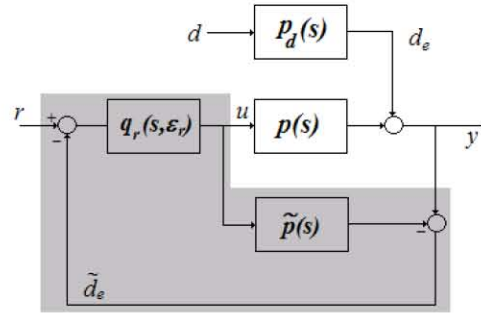


Figura 4.23: Implantación en el lazo de control de esquema CMI 1GL.

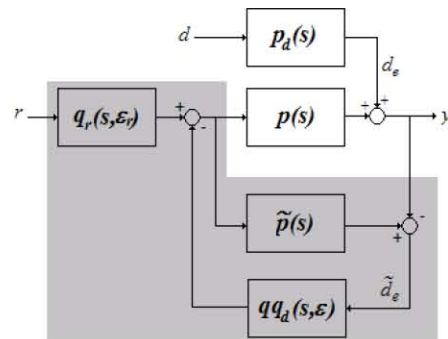


Figura 4.24: Implantación en el lazo de control de esquema CMI 2GL.



El CVDMI puede elegir entre las dos estrategias del Control por Modelo Interno, a través del uso de una estructura *Case*<sup>5</sup>. La Figura 4.25 presenta la programación del algoritmo de Control Por Modelo Interno de Un Grado de Libertad. En ella se debe reconocer a:

1. la lectura de la muestra de la variable del proceso,  $y(t)$ ,
2. la salida del modelo del proceso,
3. la referencia,  $r(t)$ ,
4. el controlador  $q_r(s)$ ,
5. la saturación del esfuerzo de control,
6. el modelo del proceso,  $\tilde{p}(s)$ , y
7. la escritura del esfuerzo de control,  $u(t)$ .

La Figura 4.26 muestra un diagrama similar, pero en este caso para el esquema de Dos Grados de Libertad. Aquí, se identifican a:

1. la muestra de la variable del proceso,  $y(t)$ ,
2. la salida del modelo del proceso,
3. la referencia,  $r(t)$ ,
4. el controlador  $q_r(s)$ ,
5. el controlador  $qq_d(s)$ ,
6. la saturación del esfuerzo de control,
7. el modelo del proceso,  $\tilde{p}(s)$ , y
8. el esfuerzo de control,  $u(t)$ .

Los bloques (*Transfer Function VI*, *Saturation VI*, y *Delay VI*) con los cuales se implantaron las expresiones del modelo del proceso,  $\tilde{p}(s)$ , y de los controladores  $q_r(s)$  y  $qq_d(s)$  son parte del LabVIEW PID Control Toolset. Estos bloques permiten la programación de esquemas de control lineal, desde el esquema de realimentación simple hasta estrategias de control más complejas que ésta.

El sistema de control también proporciona un paro seguro del proceso, en caso de suceder un error en la ejecución del programa o si el usuario así lo desea. La Figura 4.27 expone estas ideas. En este diagrama se observa

---

<sup>5</sup>Verdadero (True) en el caso de Un Grado de Libertad, y Falso (False) en el caso de Dos Grados de Libertad.

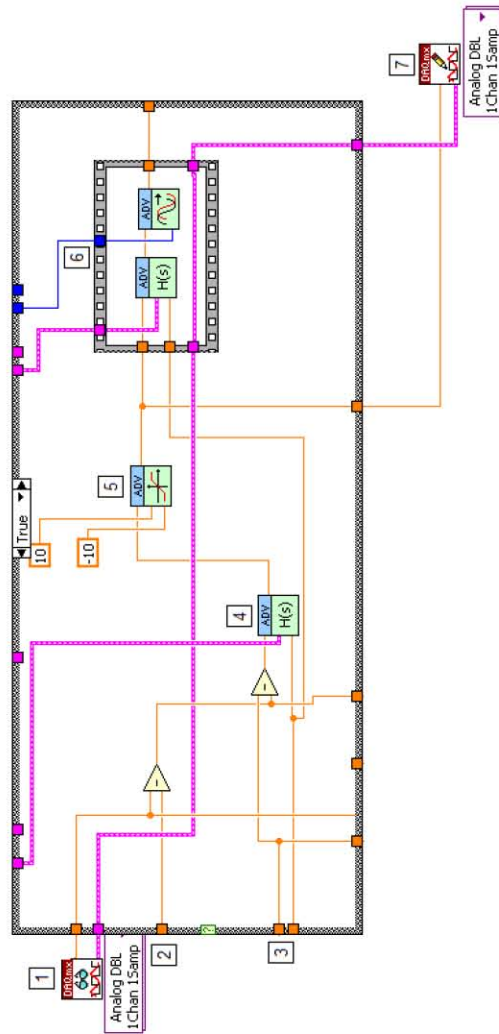


Figura 4.25: Programación del algoritmo CMI 1GL.

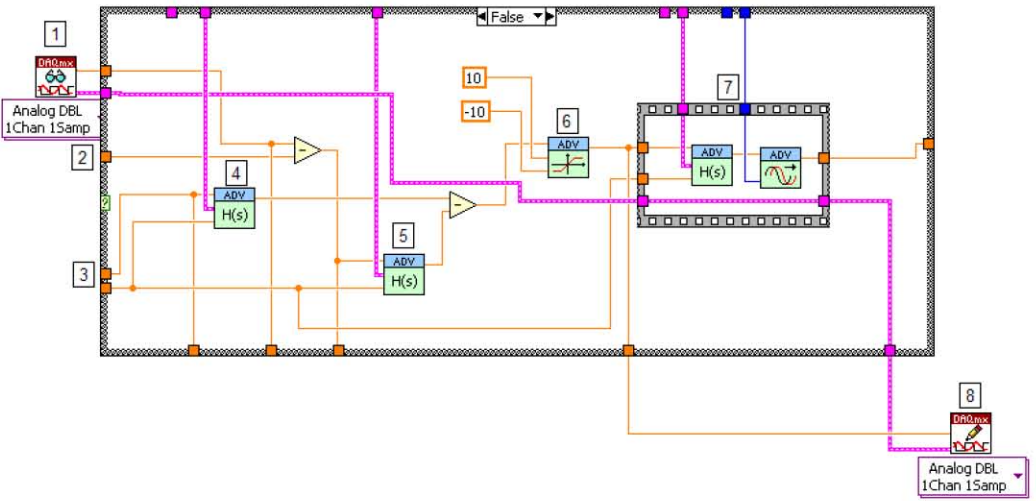


Figura 4.26: Programación del algoritmo CMI 2GL.

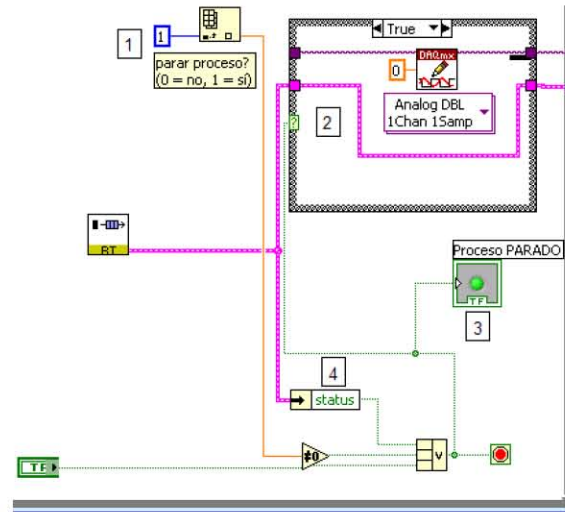


Figura 4.27: Paro del proceso.

1. una señal de paro del proceso dada por el usuario desde la interfaz, y que llega aquí después de pasar por la memoria compartida y por un *Real-Time FIFO*;
2. la acción de paro del proceso, al escribir 0 [V] en el canal de salida analógica;
3. una variable local para pasar la señal de paro al lazo de comunicación; y
4. la verificación del estado del programa, para ver si ocurrió algún error que detenga la operación del sistema.

En la Figura 4.28 se muestra el lazo de comunicación. Las actividades que en él se realizan son:

1. la lectura del valor de la referencia  $r(t)$ , la señal de paro del proceso (con 0 no para el proceso, con 1 sí lo detiene), la elección de la estrategia de control (0 = CMI 1GL, 1 = CMI 2GL), y el valor de la perturbación;
2. la escritura en un FIFO de la referencia, la señal de paro, y la estrategia;
3. la lectura del otro FIFO de los valores de la referencia, la variable del proceso,  $y(t)$ , el esfuerzo del control,  $u(t)$ , y el efecto estimado de la perturbación,  $\hat{d}_e(t)$ , provenientes del lazo de control;

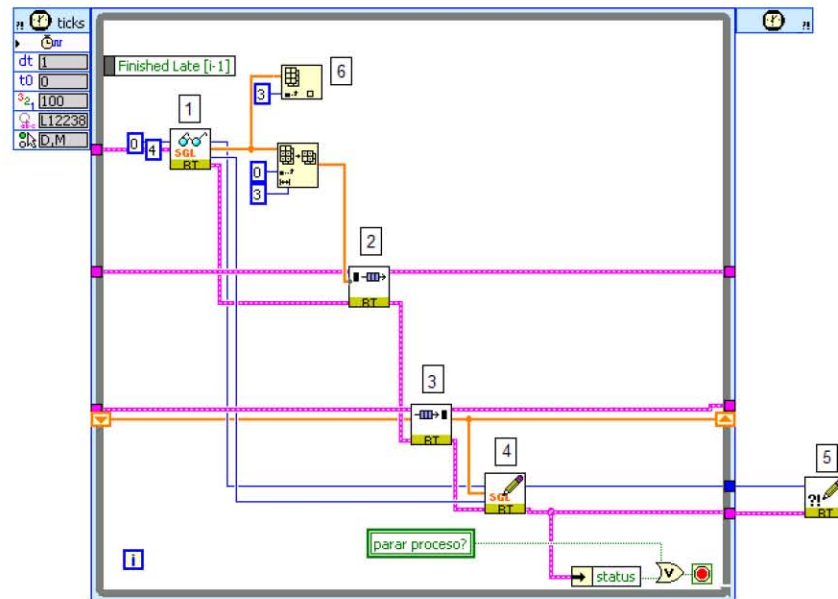


Figura 4.28: Ciclo de comunicación.

4. la escritura de la ocurrencia de un error en la memoria compartida, a ser informada al usuario desde la interfaz; y
5. la recuperación de la perturbación  $d(t)$  a escribir en el canal analógico de salida  $AO1$ .

Como se deduce del diagrama del lazo de control, se definen dos FIFOs. Esta idea se expone en la Figura 4.29. En la *FIFO salida* el lazo de control escribe datos, mientras el lazo de comunicación los lee. En la *FIFO entrada* sucede exactamente lo contrario.

### 4.3.2. Programación de la Máquina Anfitrión

#### Interfaz de Usuario

En este programa el usuario maneja y opera el CVDMI actuando sobre el proceso, modificando la referencia y observando la respuesta de la variable bajo control, el esfuerzo de control aplicado, y el efecto estimado de la perturbación. En la Figura 4.30 se observa el panel de usuario, el cual incluye un control numérico para cambiar la referencia, y un indicador gráfico para apreciar todas

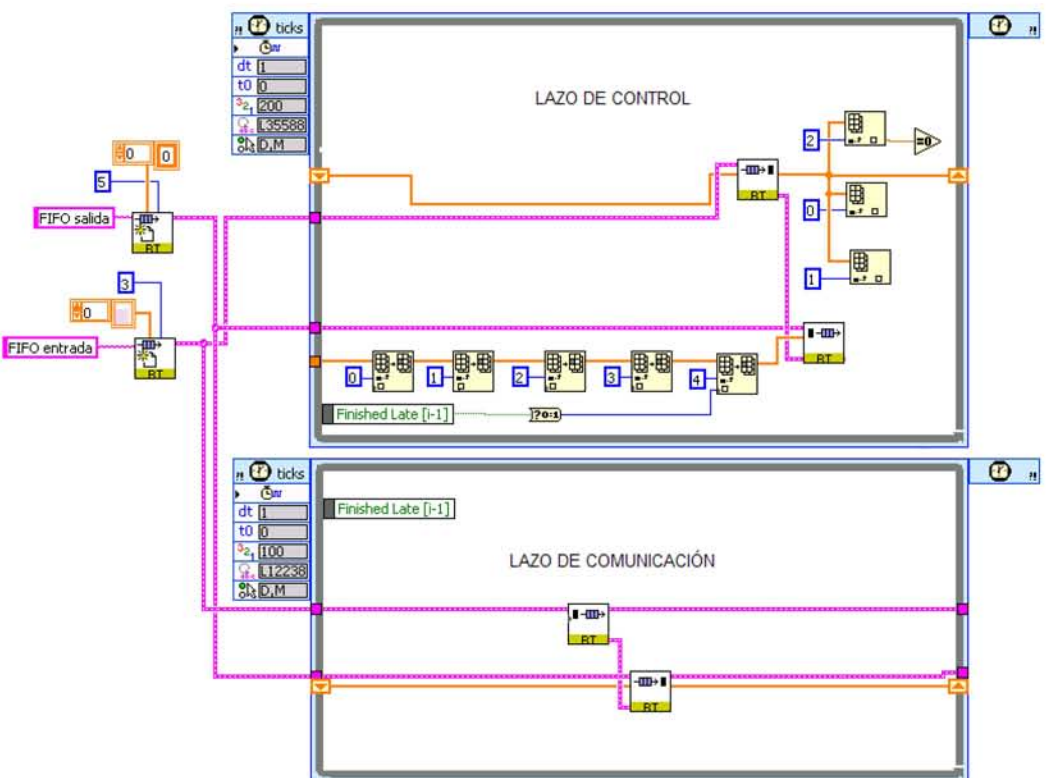


Figura 4.29: Comunicación vía FIFOs entre los ciclos de comunicación y control.

las señales mencionadas. También está el control selector entre un esquema de Control por Modelo Interno de Uno o de Dos Grados de Libertad. Por último, y únicamente con intención demostrativa, se cuenta con un control numérico para enviar una perturbación tipo escalón a la entrada o salida del proceso, según esté conectado el canal de salida analógica *AO1* al proceso.

La Figura 4.31 muestra parte del diagrama de bloques detrás de la interfaz de usuario. De aquí se observa:

1. la escritura de la referencia, señal de paro y selección de estrategia a la memoria compartida,
2. la lectura de datos provenientes del lazo de comunicación ejecutándose en la tarjeta de tiempo real (referencia, salida, esfuerzo, efecto de  $d(t)$  y estado de operación) a través la memoria compartida,
3. la lectura de algún mensaje de error en la comunicación por la memoria compartida,
4. la visualización de los datos de interés en el indicador gráfico, y
5. un tiempo de espera de 100 [ms], que marca el periodo aproximado de ejecución de este ciclo.

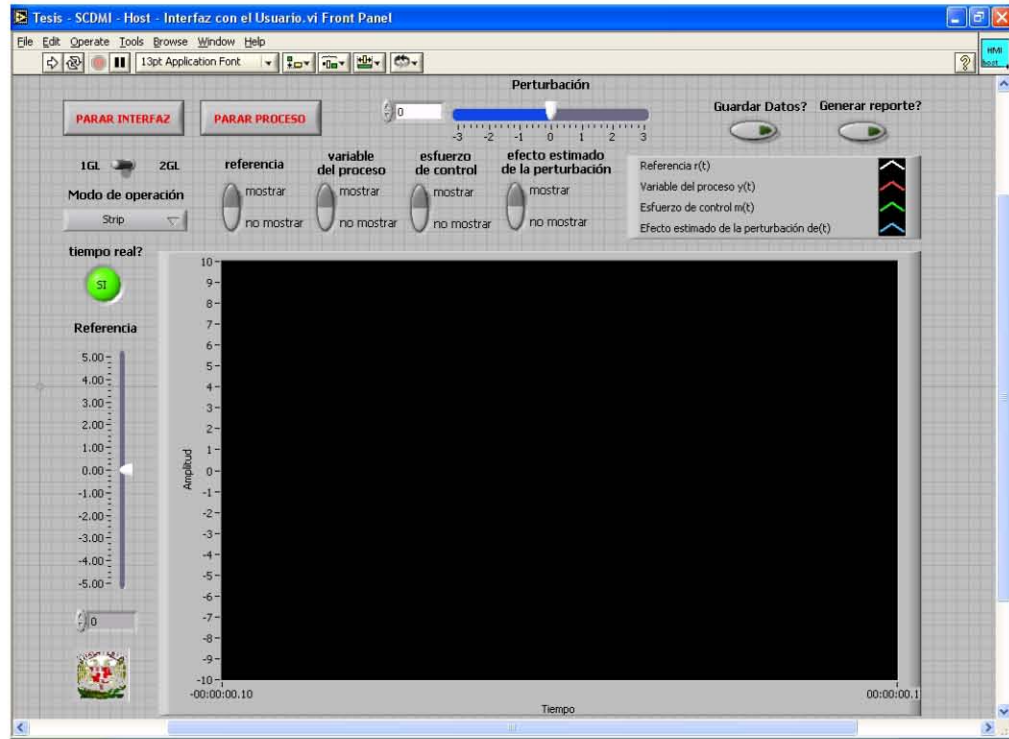
### Almacenamiento de Datos y Generación de Reportes

La interfaz del CVDMI da también la posibilidad de almacenar las variables en un archivo para su posterior análisis; así como de generar reportes donde se indica la fecha y hora del mismo, junto con la presentación del indicador gráfico de la interfaz.

El diagrama en la Figura 4.32 muestra cómo se almacenan las variables en un archivo, al apretar el botón *Guardar Datos?* en el panel de usuario. Primero es necesario crear o abrir el archivo. Después, dentro del ciclo, se irán guardando los datos provenientes de la memoria compartida, siempre que el botón *Guardar Datos?* esté activado. Al finalizar la ejecución de la interfaz, se guarda y cierra el archivo en memoria.

También desde la interfaz se pueden crear reportes de operación del sistema al pulsar el botón *Generar Reporte?*. El código dentro del diagrama de bloques que realiza esta tarea se muestra en la Figura 4.33.

Figura 4.30: Panel de la interfaz de usuario.





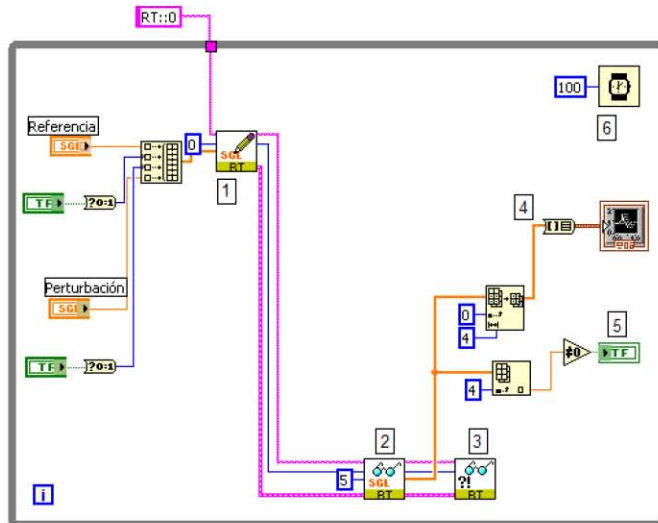


Figura 4.31: Parte del diagrama de bloques de la interfaz de usuario, encargada de la lectura y escritura en memoria compartida.

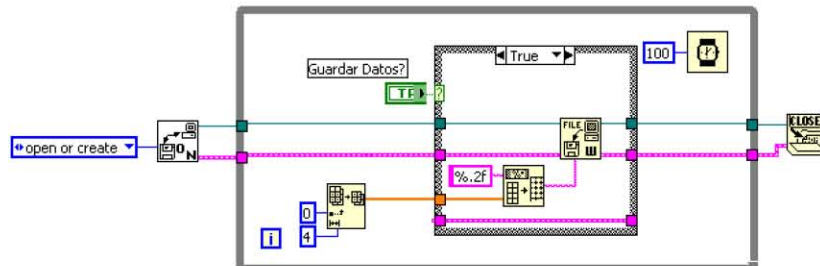


Figura 4.32: Almacenamiento de datos en un archivo.

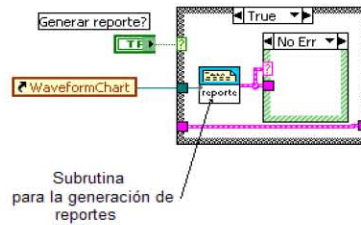


Figura 4.33: Detalle del diagrama de bloques de la interfaz, encargada de la generación de reportes.

Como es posible observar en la Figura 4.33, una subrutina es la encargada de desempeñar las acciones necesarias para la creación de los reportes. El diagrama de la Figura 4.34 presenta la construcción de esta subrutina, llamada *Genera Reporte VI*. En ella:

1. se crea el reporte,
2. se agrega el encabezado al mismo,
3. se anexa la fecha y hora,
4. se adiciona la imagen del indicador gráfico del panel frontal de la interfaz,
5. se guarda el reporte, y
6. se borra el mismo.

## 4.4. Operación del Controlador Virtual Determinístico por Modelo Interno

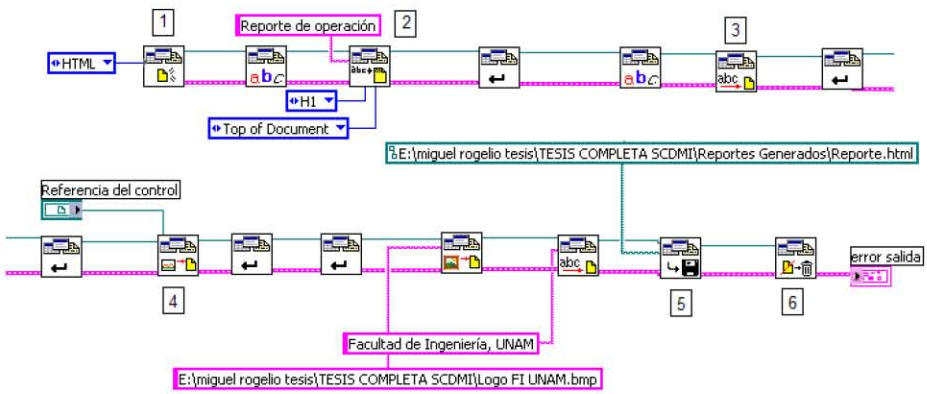
En las Figuras 4.35, 4.36, 4.37 y 4.38 se muestra la operación del CVDMI a entradas tipo escalón en la referencia y la perturbación, para un caso específico de parámetros del proceso, con las estrategias de Control por Modelo Interno de Uno y Dos Grados de Libertad. En particular, debe compararse la operación del sistema de control de 1GL (Figuras 4.35 y 4.36) con lo predicho en las Figuras 4.9 y 4.10, según los parámetros  $K = 2$  y  $T = 1[seg]$ . Aquí logra apreciarse cómo el comportamiento del CVDMI es muy similar al previsto en el diseño.

Lo anterior puede decirse para el caso CMI 2GL, de acuerdo con lo mostrado en las Figuras 4.18, 4.19, 4.37 y 4.38.

### 4.4.1. Desempeño en el Tiempo del Controlador Virtual Determinístico por Modelo Interno

Para medir el desempeño en el tiempo de nuestro programa, fue necesario agregar los bloques *TraceTool Start Trace VI* y *TraceTool Stop Trace and Send VI* del *LabVIEW Execution Trace Tool*, conforme a los mencionado en el capítulo 3. Se muestra en la Figura 4.39 la adición de dichos bloques, a fin de recabar los datos a ser vistos después en la sesión dentro del *LabVIEW Trace Tool*.

Figura 4.34: Subrutina Genera Reporte VI.



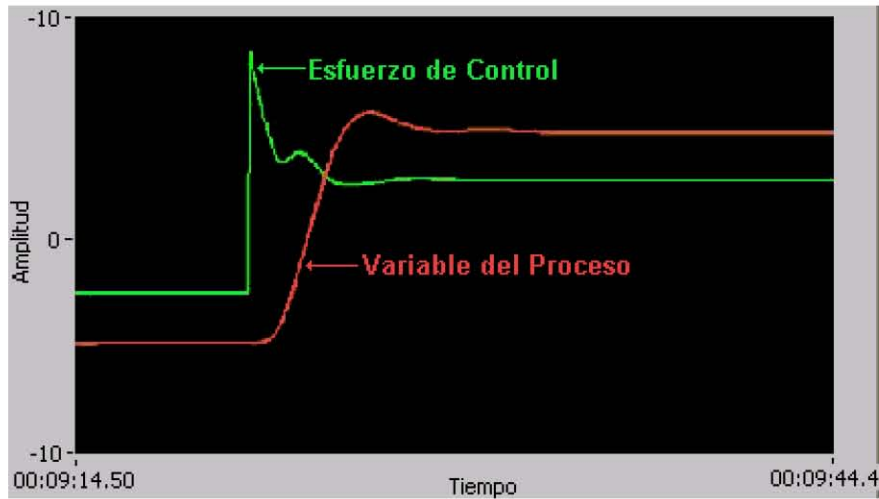


Figura 4.35: Respuesta del CVDMI a un cambio escalón en la referencia, bajo la estrategia CMI 1GL, para el proceso definido por  $K = 2$  y  $T = 1[seg]$ .

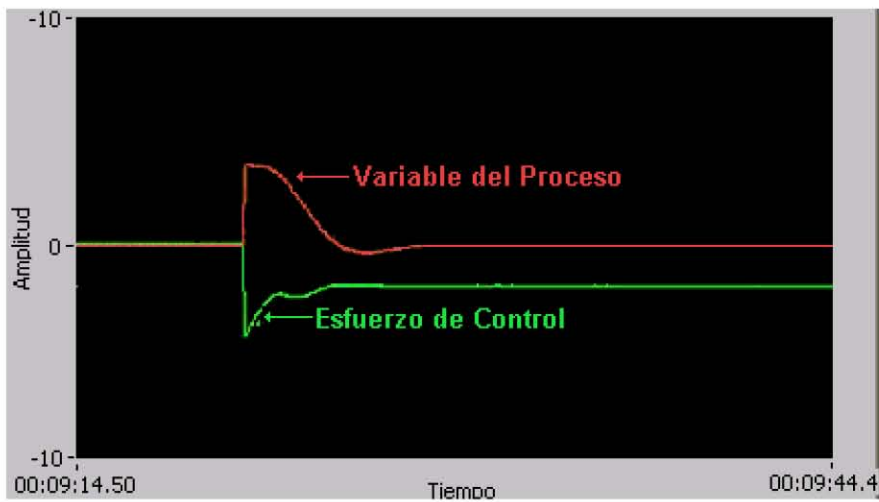


Figura 4.36: Respuesta del CVDMI a una perturbación tipo escalón, actuante a la salida del proceso,  $p(s)$ , bajo la estrategia CMI 1GL, para los valores de parámetros  $K = 2$  y  $T = 1[seg]$ .

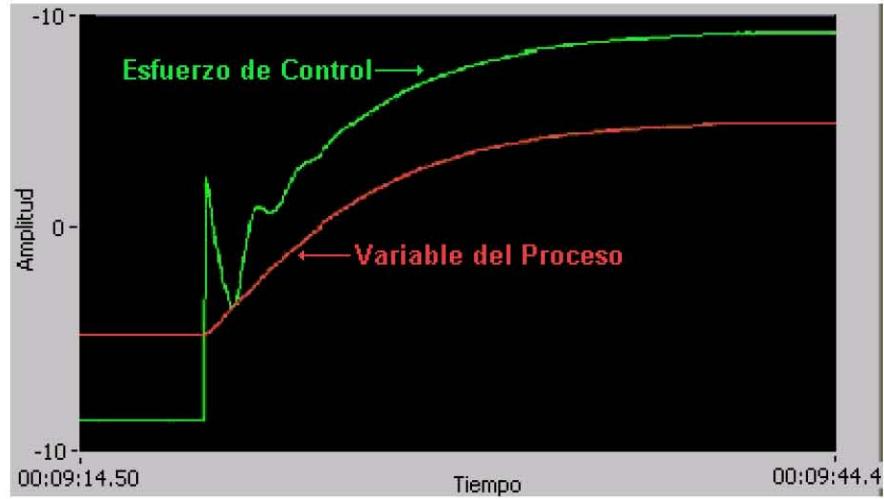


Figura 4.37: Respuesta del CVDMI a un cambio escalón en la referencia, bajo la estrategia CMI 2GL, para el proceso definido por  $K = 0.5$  y  $T = 0.01[\text{seg}]$ .

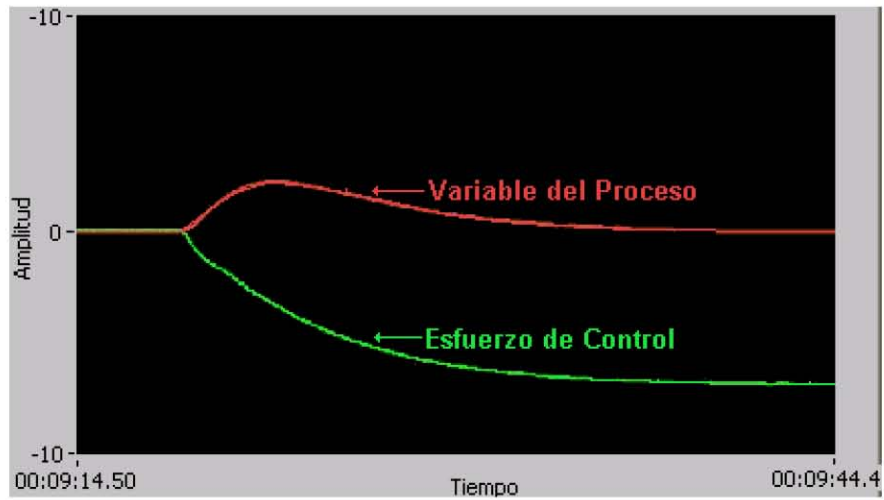


Figura 4.38: Respuesta del CVDMI a una perturbación tipo escalón, actuante a la entrada del proceso,  $p(s)$ , bajo la estrategia CMI 2GL, para los valores de parámetros  $K = 0.5$  y  $T = 0.01[\text{seg}]$ .



Figura 4.39: Adición de los bloques (1) *TraceTool Start Trace VI* y (2) *TraceTool Stop Trace and Send VI*, al diagrama de bloques de los lazos de control y comunicación.

Las Figuras 4.40 y 4.41 muestran la medición en el periodo de ejecución de nuestro lazo de control, para los casos de Un Grado y Dos Grados de Libertad, respectivamente. En la vista de hilos de ejecución, se aprecian las tareas que representan a los ciclos de control y comunicación. El intervalo entre el inicio de una iteración del lazo de control y la siguiente iteración está alrededor de 1 [ms], tal como se observa en ambas Figuras, conforme al tiempo que se había deseado.

Igualmente se resalta la precedencia del ciclo de control sobre el lazo de comunicación: cuando el primero ha terminado de ejecutarse, libera el tiempo de procesamiento al segundo. Se debe recordar la sincronización de los tiempos de inicio de los lazos, de acuerdo con lo expuesto en secciones anteriores.

## 4.5. Conclusiones

El desarrollo del presente trabajo contempló dos grandes objetivos:

- primero, conocer una nueva estrategia de control, más allá de las técnicas e ideas del control clásico, capaz de afrontar la presencia de incertidumbre para un proceso; y

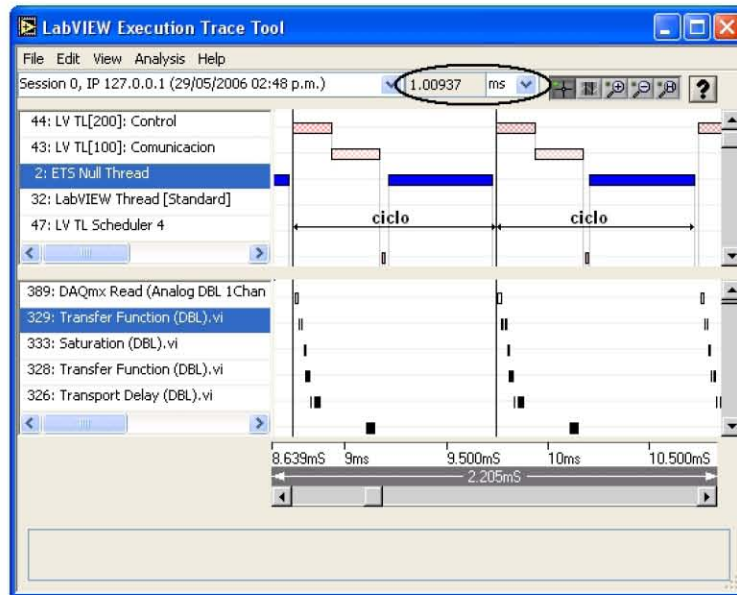


Figura 4.40: LabVIEW Execution Trace Tool con la sesión correspondiente al sistema de Un Grado de Libertad.

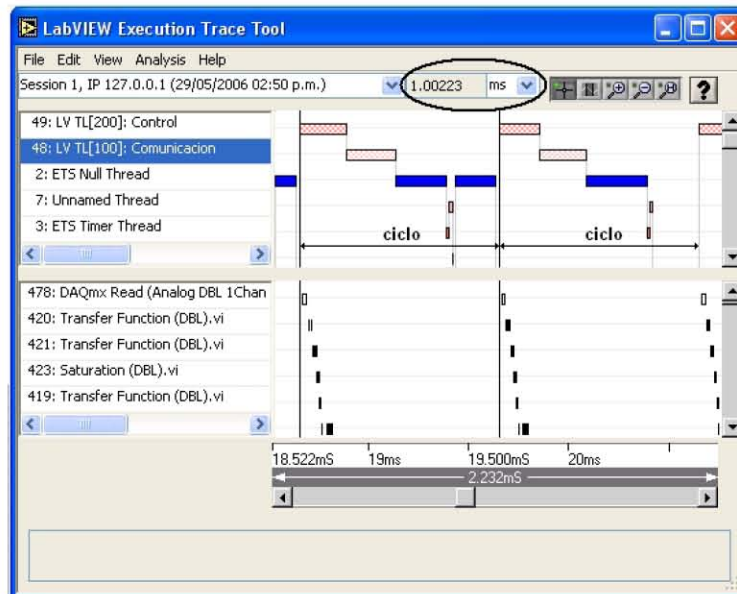


Figura 4.41: LabVIEW Execution Trace Tool con la sesión correspondiente al sistema de Dos Grados de Libertad.

- segundo, aprender y usar una tecnología orientada a la implantación de sistemas en tiempo real, a fin de construir un sistema controlador que opere bajo cualquier estrategia, como aquélla estudiada en el primer punto.

Como pudo observarse en los primeros dos capítulos, el diseño de sistemas bajo el esquema de Control por Modelo Interno de Uno y Dos Grados de Libertad se apoya en muchas de las ideas propias del control clásico para el ajuste de controladores. En específico, el diseño en el dominio de la frecuencia juega un papel principal en el ajuste  $M_P$  y la síntesis  $M_P$ , a través del uso de las funciones de sensibilidad complementaria  $CS$ , de sensibilidad parcial  $PS$  y de sensibilidad  $S$ , según se trate del esquema de Un Grado, de Dos Grados, o de procesos subamortiguados, respectivamente. Por otro lado, y a pesar de la necesidad de contar todavía para el diseño del sistema con una expresión matemática que describa al proceso a controlar, el Control por Modelo Interno se comporta de manera satisfactoria si se cuenta con dicha descripción y con una correcta estimación de la incertidumbre paramétrica de la misma. Debemos mencionar aquí que el comportamiento del Controlador Virtual Determinístico por Modelo Interno, comparado con las simulaciones realizadas en el programa IMCTUNE, fue muy similar. Por ello, *esta estrategia de control representa una solución real en situaciones donde se cuenta con un proceso que cumple con las condiciones anteriormente mencionadas.*

Los sistemas en tiempo real se encuentran hoy en un número considerable de aplicaciones, entre ellas el control. Sin embargo, la implantación de tales sistemas puede no ser trivial, al tenerse que considerar diversos factores y criterios. Por ejemplo, en el caso de utilizar un sistema de cómputo en la implantación de un sistema de control en tiempo real, deberá de asegurarse que todo el hardware sea compatible, que la elección del software sea la correcta, y que el equipo de medición, sensado y los actuadores sea el adecuado, por mencionar sólo algunas ideas. Lo anterior necesariamente requerirá del involucramiento de distintas personas: ingenieros en programación, eléctricos, electrónicos, en control, etcétera. Y ello también implicará largos tiempos de desarrollo. *En esta idea se centra la fortaleza de un programa como LabVIEW, LabVIEW Real-Time Module y su hardware asociado: tratándose de un interfaz de programación gráfica, sencilla, y orientada a aplicaciones especiales como el tiempo real, puede construirse un sistema real en poco tiempo y con un mínimo involucramiento de personas.* El resultado puede repercutir entonces en menores costos y mayor productividad.

En relación a la operación en tiempo real de nuestro sistema de control, fue posible verificar en la sección anterior el alto grado de determinismo alcanzado. Así, *LabVIEW conforma una buena opción en el desarrollo de un sistema en tiempo real cualquiera*, siempre y cuando la programación sea tal que se asegure un alto desempeño en el sistema. *Igualmente, gracias al LabVIEW PID Control Toolset fue posible programar un algoritmo de control distinto a las estrategias tradicionales, como el control PID.* Por lo tanto, parece haber pocos impedimentos para la implantación de una estrategia de control lineal cualquiera; un



programador experimentado en LabVIEW y en el área de control seguramente podrá hacer frente a la mayoría de las situaciones que se presentan en la vida cotidiana.

## Apéndice A

# Bloque de Conexiones BNC-2120

El bloque BNC-2120 facilita la conexión de los canales de entrada y de salida, analógicos o digitales, de una tarjeta de adquisición de datos de la series E o S de National Instruments, a cualquier dispositivo externo de interés. El bloque contiene los siguientes elementos:

- Ocho conectores BNC para entrada analógica (AI) en modo diferencial, con las siguientes características adicionales:
  - conector de termopar,
  - referencia de temperatura, y
  - terminales en tornillo para medición de resistores.
- Dos conectores BNC de salida analógica (AO).
- Terminales en tornillo para entrada/salida digital (DIO), con leds indicadores de estado.
- Terminales en tornillo para conexiones de entrada/salida de señales de tiempo.
- Dos conectores BNC para uso definido por el usuario.
- Un generador de funciones con la siguientes salidas:
  - Señal de onda cuadrada de frecuencia ajustable, compatible con lógica TTL.
  - Señal senoidal o triangular, ajustable en frecuencia y magnitud.
- Un codificador.

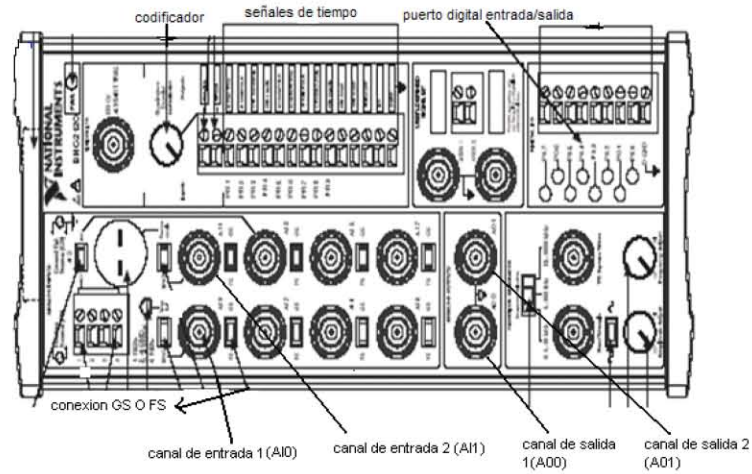


Figura A.1: Bloque conector BNC-2120.

El BNC-2120 tiene un conector entradas/salidas de 68 pines, adecuado para tarjetas multifunción DAQ de las series E y S. Naturalmente ello requerirá del empleo del cable conector adecuado: SHC-68-68EP.

Se puede utilizar el BNC-2120 para medir señales analógicas de entrada, provenientes de fuentes flotantes o referidas a una tierra. Para medir señales flotantes, se necesita mover el interruptor *FS/GS*, que está abajo de cada conector BNC de *AI*, a la posición *FS*. En esta posición, la terminal negativa del amplificador de instrumentación ubicado a la entrada del DAQ se conecta a tierra a través de un resistor de  $4,99 \text{ } [\Omega]$ . Para medir señales referidas, se puede colocar el interruptor *FS/GS* en cualquier posición; sin embargo es preferible utilizar la opción *GS*, a fin de evitar lazos en tierra. Las ideas anteriores se resumen en la Figura A.2.

El BNC-2120 conecta los canales de salida analógica *AO* del dispositivo de la serie E o S con los conectores BNC *AO0* y *AO1*. *AO GND* es la referencia de estos dos conectores.

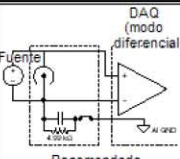
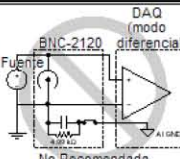
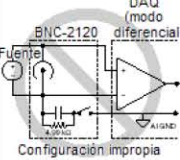
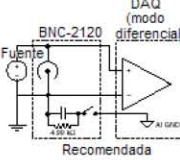
Configuración BNC-2120	Fuentes de Señal	
	Fuente Flotante	Fuente Referida a Tierra
FS	 <p>Recomendada</p>	 <p>No Recomendada</p>
GS	 <p>Configuración impropia</p>	 <p>Recomendada</p>

Figura A.2: Resumen de opciones de configuración del BNC-2120.



# Bibliografía

Bishop, Robert; *Learning with LabVIEW<sup>TM</sup> 7 Express*, Estados Unidos, Pearson Prentice Hall, 2004.

Brosilow, Coleman y Babu Joshep; *Techniques of Model-Based Control*, Estados Unidos, Prentice Hall PTR, 2002.

Burns, Alan y Andy Wellings; *Real-Time Systems and their Programming Languages*, Inglaterra, Addison-Wesley, 2001.

Lawrence Peter y Konrad Mauch; *Real-Time Microcomputer System Design: An Introduction*, Estados Unidos, McGraw Hill, 1987.

Marlin, Thomas; *Process Control: Designing Processes and Control Systems for Dynamic Performance*, Estados Unidos, Mc Graw Hill, 1995.

National Instruments Corporation; *DAQ Getting Started Guide*, junio 2005.

—; *Installation Guide BNC-2120*, junio 2003.

—; *LabVIEW<sup>TM</sup> Execution Trace Toolkit User Manual*, abril 2004.

—; *LabVIEW<sup>TM</sup> Measurements Manual*, abril 2003.

—; *LabVIEW<sup>TM</sup> Real-Time Module User Manual*, abril 2004.

—; *PID Control Toolset User Manual*, noviembre 2001.

—; *PCI E Series User Manual*, julio 2002.

—; *PCI-7041 User Manual*, octubre 2002.