



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD ZACATENCO

CONTROL AUTOMÁTICO

**Sintonización de controladores PID mediante un algoritmo genético
multiobjetivo (NSGA-II)**

**Tesis que presenta
José Rubén Felipe Lagunas Jiménez**

Para obtener el Grado de

Doctor en Ciencias

**En la Especialidad de
Control Automático**

Directores de la Tesis:

**Dr. Juan Carlos Martínez García
Dr. Guillermo Fernández Anaya**

México, D.F.

Abril 2004

A la memoria de mi papá Sr. Rubén Lagunas Silva†, mi mamá Sra. Aurelia Jiménez Jiménez†, mi hermano Ing. Armando Lagunas Jiménez†.

A mis otros hermanos y a mi tía Pinita.

A mi esposa Yela Angélica Báez C. y a mis hijos.

A mis asesores Dr. Juan Carlos Martínez García y Dr. Guillermo Fernández Anaya.

Con especial afecto y agradecimiento a los Doctores Carlos A. Coello Coello, Rubén Garrido Moctezuma, Alberto Soria López, Moisés Bonilla Estrada, Basilio del Muro Cuellar y Daniel U. Campos.

Mi Agradecimiento a la Universidad Autónoma de Campeche y al Programa de Mejoramiento de Profesores (PROMEP-SEP) por su apoyo económico.

Índice general

1. Conceptos Básicos	15
1.1. Introducción	15
1.1.1. Optimización global	16
1.1.2. Problema de optimización multiobjetivo	16
1.2. Estrategias de optimización global	24
1.3. Técnicas de optimización multiobjetivo	25
1.3.1. Métodos a priori	25
1.3.2. Métodos a posteriori	27
1.3.3. Métodos progresivos	28
2. Algoritmos Genéticos Aplicados a Optimización Multiobjetivo	31
2.1. Introducción	31
2.1.1. Diversidad en la población	32
2.2. Algoritmos genéticos de primera generación	34
2.2.1. NSGA (Nondominated Sorting Genetic Algorithm)	34
2.2.2. NPGA (Niche-Pareto Genetic Algorithm)	34
2.2.3. MOGA (Multi-objective Genetic Algorithm)	35
2.3. Algoritmos genéticos de segunda generación	36
2.3.1. NSGA-II (Nondominated Sorting Genetic Algorithm-II)	36
2.3.2. SPEA (Strength Pareto Evolutionary Algorithm)	45
2.3.3. SPEA-2	46

2.3.4. PAES (Pareto Archived Evolution Strategy)	46
2.3.5. Algoritmos MRCD y MRCD min-max.	47
3. Controladores PID	50
3.1. Controladores PID	50
3.1.1. Introducción	50
3.1.2. Objetivos	53
3.2. Los enfoques más comunes para síntesis de controladores PID	54
3.3. Procesos controlados con un controlador nuevo PI-PD Smith predictor usando autosintonizado, propuesto por Kaya.	63
3.3.1. Procedimiento para determinar los controladores del esquema de control propuesto por Kaya	65
3.4. Método propuesto por Skogestad para sintonizar controladores PID (2001)	70
3.5. Controlador PID de dos grados de libertad	76
3.5.1. Controladores PID comerciales	78
4. Aplicación de un Algoritmo Genético simple	80
4.1. Introducción	80
4.2. Planteamiento del problema	81
4.3. Control PID basado en la combinación H_2/H_∞	83
4.4. Ejemplo ilustrativo.	86
4.5. Enfoque con algoritmos genéticos	89
4.5.1. Procedimiento computacional	91
4.5.2. Resultados experimentales	92
4.5.3. Problema resuelto como Problema multiobjetivo	94
4.6. Conclusiones	97
5. Sintonizado de un Controlador PID Robusto usando NSGA-II	98
5.1. Introducción	98
5.2. Diseño del controlador PID	99

5.2.1. Plantas de prueba	105
5.2.2. Evaluación de resultados	107
5.2.3. Valores de los parámetros del NSGA-II	112
5.2.4. Valores de los parámetros de los controladores PID	113
5.3. Conclusiones	142
6. Sintonización de Controladores PID de Plantas Seleccionadas	143
6.1. Introducción	143
6.2. Evaluación de resultados	144
6.3. Análisis de resultados	155
7. Conclusiones y Perspectivas de Trabajos Futuros	156
7.1. Discusiones	156
7.2. Conclusiones	157
7.3. Perspectivas de trabajos futuros	159

Índice de figuras

1-1. Frente de Pareto del problema de Schaffer	20
1-2. Solución ideal	22
2-1. Distancia de agrupamiento	40
2-2. Procedimiento del NSGA-II	43
3-1. Método de Ziegler-Nichols	61
3-2. Esquema de control propuesto por Kaya	65
3-3. Esquema de control usado por Skogestad	71
3-4. Estructura de dos grados de libertad	77
4-1. Sistema de control	82
4-2. Interfaz gráfica	90
4-3. Salida del sistema de control	93
4-4. Frente de Pareto	96
4-5. Salida del proceso	96
5-1. Sistema de control en lazo cerrado con un controlador PID	99
5-2. Estimación de la norma \mathcal{L}_1 de $g(t)$	104
5-3. Sintonización PID	111
5-4. Planta 1	116
5-5. Planta 1	117
5-6. Planta 2	118

5-7. Planta 2	119
5-8. Planta 3	120
5-9. Planta 3	121
5-10. Planta 4	122
5-11. Planta 4	123
5-12. Planta 5	124
5-13. Planta 5	125
5-14. Planta 6	126
5-15. Planta 6	127
5-16. Planta 7	128
5-17. Planta 7	129
5-18. Planta 8	130
5-19. Planta 8	131
5-20. Planta 10	132
5-21. Planta 10	133
5-22. Planta 11	134
5-23. Planta 11	135
5-24. Planta 12	136
5-25. Planta 12	137
5-26. Planta 13	138
5-27. Planta 13	139
5-28. Planta 14	140
5-29. Planta 14	141
6-1. Planta Seleccionada 1	146
6-2. Planta Seleccionada 1	147
6-3. Planta seleccionada 1	148
6-4. Planta Seleccionada 2	149
6-5. Planta Seleccionada 2	150

6-6. Planta Seleccionada 2	151
6-7. Planta Seleccionada 3	152
6-8. Planta Seleccionada 3	153
6-9. Planta Seleccionada 3	154

Índice de cuadros

3.1. Sintonización mediante óptimo simétrico	58
3.2. Sintonización mediante Kappa-tau(KT)	60
3.3. Sintonización por Ziegler-Nichols	62
3.4. Controladores PID comerciales	79
4.1. Parámetros de los controladores PID1 y PID2.	95
5.1. Plantas de prueba	106
5.2. Rango de los parámetros	114
5.3. Parámetros de los controladores	114
5.4. Índices de desempeño de la Planta 1	117
5.5. Índices de desempeño de la Planta 2	119
5.6. Índices de desempeño de la Planta 3	121
5.7. Índices de desempeño de la Planta 4	122
5.8. Índices de desempeño de la Planta 5	125
5.9. Índices de desempeño de la Planta 6	127
5.10. Índices de desempeño de la Planta 7	128
5.11. Índices de desempeño de la Planta 8	130
5.12. Índices de desempeño de la Planta 10	132
5.13. Índices de desempeño de la Planta 11	135
5.14. Índices de desempeño de la Planta 12	136
5.15. Índices de desempeño de la Planta 13	138

5.16. Índices de desempeño de la Planta 14	140
6.1. Plantas seleccionadas	143
6.2. Parámetros contr. plantas selecc.	144
6.3. Rango de los parámetros plantas selecc.	144
6.4. Índices de desempeño de la Planta Seleccionada 1	147
6.5. Índices de desempeño de la Planta Seleccionada 2	150
6.6. Índices de desempeño de la Planta Seleccionada 3	152

Introducción

Desde hace cientos de años se ha tenido necesidad del control automático, al principio en aplicaciones rudimentarias, pero poco a poco y conforme el hombre fue automatizando sus procesos de trabajo y satisfactores de vida, también se han sofisticado las aplicaciones del control. Aunque su empleo proviene de cientos de años atrás, sólo hasta hace relativamente poco tiempo se ha desarrollado de manera importante, debido al avance de la ciencia y la tecnología en su conjunto. El control automático se manifiesta en diferentes formas: como controladores de un solo lazo, como software en un controlador lógico programable, en juguetes, máquinas herramientas, y en una infinidad de aplicaciones de la vida actual. Los planteamientos del control como una respuesta a las necesidades del hombre, tienen en gran medida mucha similitud con la naturaleza del pensamiento de éste y de la forma de ver y analizar los problemas. La sociedad y el hombre, que forma parte de ella, tiene que tomar decisiones en su vida diaria. Es por esto que un problema en particular se puede plantear como una serie de objetivos y compromisos bajo ciertas condiciones. En el caso del control automático, muchos problemas se pueden plantear como un conjunto de índices de desempeño, también denominadas funciones objetivo, bajo ciertas restricciones. Por ejemplo, en muchos problemas prácticos se requiere que un sistema de control tenga la capacidad de manejar diferentes objetivos de operación (entre los que se podrían mencionar, estabilidad en lazo cerrado, señales de control bajas e insensibilidad en el modelo ante variaciones en sus parámetros). Estos objetivos podrán estar en muchas ocasiones en conflicto y deberán ser manejados simultáneamente de manera adecuada. La sofisticación de la tecnología aplicada a las computadoras, asociada a

los refinamientos del *software*, ha permitido probar la efectividad de muchas estrategias computacionales aplicadas a la optimización de problemas de control multiobjetivo. Algunas, basadas en técnicas tradicionales (analíticas), y otras en algoritmos de búsqueda estocástica como pueden ser las técnicas evolutivas, entre otras.

Los algoritmos evolutivos son algoritmos de optimización y búsqueda estocástica inspirados en los procesos de evolución natural. Actualmente, el enfoque evolutivo incluye un grupo de estrategias computacionales, entre las cuales se encuentran: los Algoritmos Genéticos (AGs), la Programación Evolutiva, la Programación Genética y las Estrategias Evolutivas. Estas estrategias resultan ser una herramienta poderosa para resolver problemas de optimización multiobjetivo, particularmente los algoritmos genéticos multiobjetivo, los cuales realizan procesos de búsqueda aleatoria en un espacio de búsqueda válido. Los algoritmos genéticos multiobjetivo, son una alternativa para resolver problemas de control planteados como problemas de optimización multiobjetivo con restricciones. El planteamiento de muchos problemas de control automático, sobre todo en procesos industriales, como una serie de objetivos y restricciones, ha despertado el interés de numerosos investigadores por la aplicación de las técnicas evolutivas. Las especificaciones de diseño se pueden plantear como funciones objetivo, las cuales serán, en algunos casos minimizadas y en otros maximizadas.

Los objetivos de esta investigación se pueden presentar en dos partes. La primera concierne a la aplicación de un algoritmo genético simple en la sintonización de un controlador PID. Dicho controlador es aplicado a un problema de control de posición de un servomecanismo de Corriente Directa (C.D.), utilizando control mixto H_2/H_∞ y aplicando una estrategia de solución numérica. En esta primera parte también se tiene el caso de una aplicación de optimización global, donde sólo se tiene un objetivo a minimizar. La segunda parte (y más importante) concierne el uso del algoritmo genético multiobjetivo NSGA-II (del inglés *Nondominated Sorting Genetic Algorithm-II*), para determinar los parámetros de un controlador. Específicamente se trata de un controlador PID de dos grados de libertad, dentro de un esquema de control básico aplicado a una amplia gama

de procesos. Primero se usa el NSGA-II para sintonizar trece controladores PID robustos, utilizados en el control de trece plantas propuestas originalmente por K. J. Åström, H. Panagopoulos y T. Hägglund, en el artículo “*Design of PI Controllers based on Non-Convex Optimization*” [4]. Después, se sintonizan tres controladores PID robustos para tres plantas seleccionadas, las cuales fueron tomadas del artículo “*Obtaining controller parameter for a new PI-Pd Smith predictor using autotuning*” [53]. El objetivo de utilizar estas nuevas plantas se debe a la dificultad para controlarlas y a la vez mostrar la versatilidad de los controladores PID y el procedimiento de sintonización basado en algoritmos genéticos. Para seleccionar los parámetros es necesario plantear los requerimientos de diseño que contemplan principalmente: el desempeño del sistema a controlar y la robustez ante las incertidumbres y variaciones del proceso. Partiendo de estos requerimientos se procede a construir adecuadamente las funciones objetivo y las restricciones, que contienen los parámetros del controlador, y las cuales serán optimizadas por medio del algoritmo genético. La decisión de utilizar el NSGA-II, se justifica debido a que es un algoritmo de aplicación general y su implantación computacional es gratuita, además de ser uno de los algoritmos que más prestigio tiene en la actualidad [19]. El resultado que entrega el algoritmo genético multiobjetivo es un conjunto de soluciones no-dominadas (óptimas), en el sentido de Pareto. Cada solución del frente de Pareto es un vector que contiene los parámetros del controlador y dado que todas las soluciones son buenas (en un problema de optimización global solo hay una solución), queda a criterio del usuario seleccionar la más conveniente. Los requerimientos para diseñar los controladores están en función de:

- Un adecuado seguimiento de la señal de referencia.
- Rechazo de perturbación en la carga.
- Atenuación de ruido en la medición.
- Robustez ante incertidumbres y variaciones de los parámetros de la planta (proceso).

- Acotamiento de la señal de control.

La investigación sobre algoritmos genéticos es reciente y uno de los primeros trabajos que utiliza el concepto de óptimo de Pareto fue el algoritmo genético multiobjetivo (*MOGA*) (desarrollado por Fonseca y Fleming en 1993 [29]). Este algoritmo ha tenido mucha aceptación en Ingeniería de control [28], concretamente para sintonizar controladores PID. Más recientemente, Vlachos, Williams y Gomm (1999) aplicaron un algoritmo genético simple para sintonizar un controlador PI descentralizado aplicado a procesos multivariable [76]. En el área de la Electrónica se puede mencionar una aplicación del algoritmo genético multiobjetivo (*VEGA* [67]), usado en el diseño de circuitos electrónicos digitales. En este trabajo realizado por Coello-Coello y Hernández-Aguirre [17] y [18], el objetivo principal es minimizar el número de compuertas correspondientes a una tabla de verdad propuesta. En el año 2002, los investigadores R. Kasat, D. Kunzru, D.N. Saraf y S.K. Gupta publicaron un artículo llamado “*Multiobjective Optimization of Industrial FCC Units Using Nondominated Sorting Genetic Algorithm*”, aplicado a proceso de refinación de petróleo, en el cual se usa el NSGA-II para resolver un problema de optimización multiobjetivo [52]. Otra excelente referencia bibliográfica que reúne varias aplicaciones de problemas de control multiobjetivo es el libro titulado “*Multiobjective Optimisation and Control*”, escrito por G.P. Liu, J.B. Yang y J.F. Whidborn (2003), [45].

Como antecedente principal de este trabajo de investigación se tiene la tesis doctoral titulada “*Diseño de Controladores Robustos Multiobjetivo por medio de Algoritmos Genéticos*” realizada por Alberto Herreros López en la Universidad de Valladolid, España, [36]. En dicha tesis el autor utiliza dos algoritmos genéticos multiobjetivo de su autoría, llamados MRCD y MRCD min-max.

La presente tesis está organizada como sigue: En el Capítulo 1 se presentan los conceptos básicos de optimización global y multiobjetivo, y se presenta una revisión breve de algunos algoritmos de optimización analíticos. Dentro del Capítulo 2 se describen brevemente algunos algoritmos genéticos; de estos últimos se describe detalladamente el NSGA-II. En el Capítulo 3 se presentan los conceptos básicos sobre la teoría de contro-

ladores PID. En el Capítulo 4 se muestra una aplicación de los algoritmos genéticos simples a un problema de control de posición formulado como un problema de optimización global. El problema tratado concierne al control de posición de un servomecanismo de Corriente Directa. Como complemento de la aplicación del algoritmo genético simple al problema de control de posición, al final del Capítulo 4 se presentan los resultados del mismo problema planteado en este capítulo, sólo que tratado como un problema de optimización multiobjetivo, utilizando el algoritmo genético NSGA-II. El Capítulo 5 trata sobre el diseño de controladores PID robustos utilizando el algoritmo NSGA-II. Los resultados obtenidos en el Capítulo 5 se comparan con los obtenidos por Herreros en 13 plantas de prueba, ver [36]. En el Capítulo 6 se sintonizan tres controladores PID robustos para tres plantas seleccionadas, tomadas del artículo de Kaya [53]. En el Capítulo 7 se presentan las discusiones, conclusiones y las perspectivas de trabajos futuros.

Resumen

Durante el desarrollo de esta investigación, se produjeron dos artículos utilizando algoritmos genéticos simples. El primero, denominado: *Experimental Evaluation of a Mixed H_2/H_∞ -Based PID using Genetic Algorithms*, fue presentado en el congreso Internacional IASTED (The International Association of Science and Technology for Development), dentro de las conferencias Circuits, Signal & Systems (CSC 2003) en Cancún, México. El segundo artículo titulado: *Mixed H_2/H_∞ -based PID control via genetic algorithms: An experimental evaluation*, fue presentado en el congreso Internacional IEE European Control Conference (ECC 03) en la Universidad de Cambridge, United Kingston.

En la parte más importante de esta investigación se usó un algoritmo genético multiobjetivo (NSGA-II) para sintonizar controladores PID, con una estructura de dos grados de libertad, aplicados a diferentes procesos. El algoritmo en cuestión, se usó para resolver numéricamente un problema de optimización multiobjetivo, que permite encontrar los valores de los parámetros de los controladores PID. El algoritmo genético multiobjetivo

minimiza una serie de índices de desempeño (los cuales contemplan algunos requerimientos de diseño de un esquema de control) sujetos a ciertas restricciones. Un algoritmo genético multiobjetivo a diferencia de un algoritmo genético simple, converge a un conjunto de soluciones óptimas llamado conjunto óptimo de Pareto.

Capítulo 1

Conceptos Básicos

1.1. Introducción

Problemas con objetivos múltiples aparecen de forma natural en muchos problemas de optimización y constituyen un área que ha adquirido gran relevancia en el mundo de la Investigación Científica y en aplicaciones de Ingeniería (química, petrolera e industrial en general).

El uso de algoritmos evolutivos (AEs) ha despertado mucho interés por ser una técnica que permite encontrar soluciones de problemas multiobjetivo de manera relativamente fácil, en comparación con estrategias analíticas tradicionales, que son más complejas en su aplicación y en algunos casos no llegan a alcanzar resultados satisfactorios. Los AEs son algoritmos de optimización y búsqueda, inspirados en los procesos de la evolución natural, que permiten encontrar la solución de problemas de optimización multiobjetivo, entregando como resultado un conjunto de soluciones no-dominadas que son, al menos sub-óptimas y representan buenos compromisos entre todas las funciones objetivo del problema (ver por ejemplo [78]). Este conjunto de soluciones óptimas, también conocidas como soluciones no-dominadas, pertenecen a un conjunto llamado conjunto óptimo de Pareto. Actualmente, el enfoque evolutivo incluye un grupo de estrategias computacionales entre las cuales se encuentran los Algoritmos Genéticos, la Programación Evo-

lutiva, la Programación Genética y las Estrategias Evolutivas.

A continuación se presentan algunas definiciones, las cuales son necesarias para abordar el estudio de la optimización multiobjetivo (tomadas en su mayoría de [14]).

1.1.1. Optimización global

Es el proceso de encontrar un mínimo o un máximo global (donde $\min\{f(\mathbf{x})\} = -\max\{f(\mathbf{x})\}$), en un espacio de búsqueda S . El Problema de Optimización Global [14], puede ser definido como:

Definición 1.1 (*Mínimo Global*): Dada una función $f : \Omega \subseteq S = \mathbb{R}^n \rightarrow \mathbb{R}$, $\Omega \neq \emptyset$, para $\mathbf{x} \in \Omega$, el valor $f^* \triangleq f(\mathbf{x}^*) > -\infty$ es llamado mínimo global si y sólo si:

$$\forall \mathbf{x} \in \Omega : f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad (1.1)$$

Entonces \mathbf{x}^* es la solución mínima global, f es la función objetivo y Ω es un subconjunto de S , que es el subespacio de búsqueda válido después de aplicar restricciones al problema de optimización.

Los problemas de optimización global difieren de los problemas de optimización multiobjetivo (MOP por sus siglas en inglés), ya que en éstos últimos la solución no es única y el conjunto de soluciones (conjunto de óptimos de Pareto), al ser evaluadas, producen vectores cuyos componentes representan compromisos en el espacio de funciones objetivo. Esto quiere decir que prácticamente no se puede encontrar el mínimo de cada uno de los objetivos al mismo tiempo y por lo tanto el diseñador tendrá que tomar una decisión, seleccionando alguna o algunas soluciones aceptables, de acuerdo a cada problema en particular.

1.1.2. Problema de optimización multiobjetivo

El problema de optimización multiobjetivo, también conocido como optimización multicriterio, se puede definir en palabras como sigue:

Se desea hallar un vector de variables de decisión que optimiza un vector de funciones (funciones objetivo), y satisface ciertas restricciones. Las funciones objetivo describen matemáticamente los criterios de desempeño del problema que usualmente están en conflicto unos con otros. El término “optimizar” significa encontrar una solución entre todos los valores aceptables de las funciones objetivo (aceptables para el usuario). Antes de definir formalmente el problema de optimización multiobjetivo, se presentan algunas definiciones básicas.

Variables de decisión Son cantidades numéricas cuyos valores son seleccionados en un problema de optimización. Éstas son representadas como x_j , para $j = 1, 2, \dots, n$.

El vector de n variables de decisión se representa por:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad (1.2)$$

que también puede ser escrito como:

$$\mathbf{x} = [x_1, x_1, \dots, x_n]^T.$$

Restricciones

En la mayoría de los problemas de optimización se presentan restricciones. Éstas pueden aparecer como consecuencia de las características del ambiente (o entorno) de la investigación (las limitaciones físicas, restricciones de tiempo, etc.). Las restricciones se deben satisfacer con el fin de tener soluciones válidas. Todas las restricciones describen dependencias entre las variables de decisión y las constantes (o parámetros), involucrados en el problema. Las restricciones se representan matemáticamente en forma de desigualdades como:

$$g_i(\mathbf{x}) \geq 0, \quad i = 1, 2, \dots, m$$

o en forma de igualdades:

$$h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, p$$

El valor de p debe ser menor que n , porque si $p \geq n$, se tendría una situación de sobre restricción y no se tendría ningún grado de libertad para el proceso de optimización. El grado de libertad se define como $n - p$. Las restricciones se pueden tener en forma explícita (expresiones algebraicas), o implícita, en cuyo caso se representan mediante algún algoritmo.

Funciones objetivo

Las funciones objetivo son designadas como: $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})$, donde k es el número de funciones objetivo en el MOP que está siendo resuelto. Las funciones objetivos forman un vector de funciones $\mathbf{f}(\mathbf{x})$, definido como:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_k(\mathbf{x}) \end{bmatrix}, \quad (1.3)$$

que también puede ser escrito como:

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})].$$

El conjunto de las n -tuplas de números reales es denotado por \mathbb{R}^n , llamado espacio n -Euclidiano. A continuación se mencionan dos espacios Euclidianos:

-Espacio n -dimensional de las variables de decisión, en donde cada eje corresponde a un componente del vector \mathbf{x} .

-Espacio k -dimensional de las funciones objetivo, en el cual cada eje corresponde a un componente del vector $\mathbf{f}(\mathbf{x})$.

Problema de optimización multiobjetivo general

Ahora puede ser definido formalmente un problema de optimización multiobjetivo general como se muestra a continuación:

Definición 1.2 *Problema de optimización multiobjetivo general: Determinar el vector $\mathbf{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ que satisfaga las siguientes m restricciones de desigualdad:*

$$g_i(\mathbf{x}) \geq 0 \quad i = 1, 2, \dots, m, \quad (1.4)$$

las p restricciones de igualdad:

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \dots, p \quad (1.5)$$

y optimice el vector de funciones:

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})]^T \quad (1.6)$$

Las restricciones dadas por las Ecuaciones (1.4) y (1.5), determinan la región válida Ω , y algún punto \mathbf{x} en Ω define una solución factible o válida. El vector de funciones $\mathbf{f}(\mathbf{x})$ es una función que mapea el conjunto Ω en el conjunto Λ , que contiene todos los valores posibles de las funciones objetivo. Para ilustrar la definición anterior se presenta a continuación un ejemplo de un problema de optimización propuesto por Schaffer [68], usado como problema de prueba en algoritmos genéticos multiobjetivo [24].

Problema 1.1 *Se desea encontrar el conjunto de soluciones óptimas, del problema de optimización multiobjetivo planteado por Schaffer [68], cuyas funciones objetivo son las siguientes:*

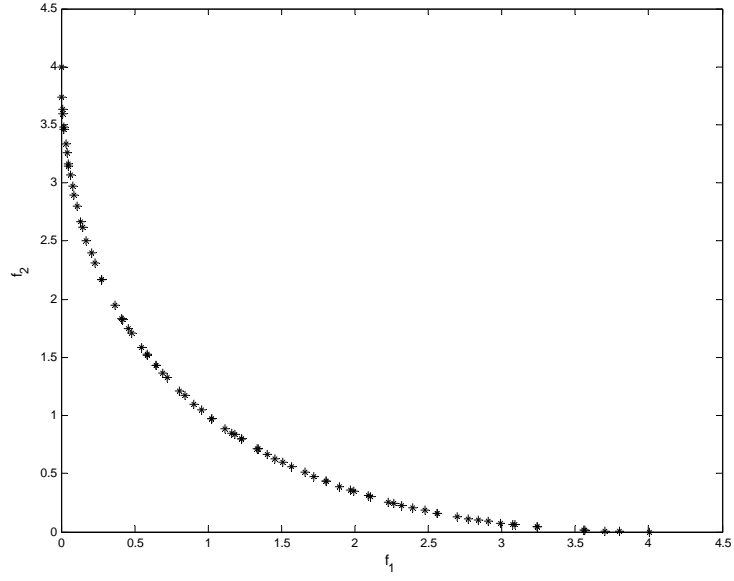


Figura 1-1: Frente de Pareto del problema de Schaffer

$$\begin{cases} f_1(x) = x^2, \\ f_2(x) = (x - 2)^2, \end{cases}$$

en un espacio de búsqueda de $[-10^3, 10^3]$.

La variable de decisión en este caso es un escalar x y las funciones objetivo se representan vectorialmente como:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}$$

La solución gráfica del problema de optimización se muestra en la Figura (1-1), la cual fue obtenida por medio del NSGA-II, utilizado en esta tesis, donde se aprecia el frente de Pareto, y que no se puede obtener un valor mínimo único para las dos funciones y que por lo tanto es necesario seleccionar el valor que más se adecuó a los requerimientos del usuario. La solución del problema se da en [24] y es $x = [0, 2]$.

En algunos problemas de optimización multiobjetivo se presenta el caso de que: una sola variable de decisión optimiza todas las funciones objetivo. A este caso se le llama solución ideal o utópica (ver la Figura 1-2).

A continuación se presentan dos definiciones más antes de mostrar formalmente las definiciones de óptimo de Pareto, conjunto de Pareto, frente de Pareto y otros conceptos fundamentales utilizados en optimización multiobjetivo.

Vector ideal

Definición 1.3 *Vector ideal: Si se tiene un vector de variables como el siguiente:*

$$\mathbf{x}^{o(i)} = [\mathbf{x}_1^{o(i)}, \mathbf{x}_2^{o(i)}, \dots, \mathbf{x}_n^{o(i)}]^T$$

que optimiza (maximiza o minimiza) a la i -ésima función objetivo $f_i(x)$, es decir, el vector $\mathbf{x}^{o(i)} \in \Omega$ es tal que:

$$f_i(\mathbf{x}^{o(i)}) = \underset{x \in \Omega}{\text{óptimo}} f_i(\mathbf{x})$$

entonces, el vector $\mathbf{f}^o = [\vec{f}_1^o, \vec{f}_2^o, \dots, \vec{f}_k^o]^T$ (donde \mathbf{f}_i^o , denota el óptimo de i -ésima función objetivo) es ideal para un MOP, y el punto correspondiente en \mathbb{R}^n es una solución ideal (utópica).

Convexidad

La definición formal de convexidad se muestra a continuación:

Definición 1.4 *Convexidad: Un conjunto convexo v se define como sigue:*

$$\text{si } \mathbf{x}_1, \mathbf{x}_2 \in v \text{ entonces } \mathbf{x} \in v$$

donde

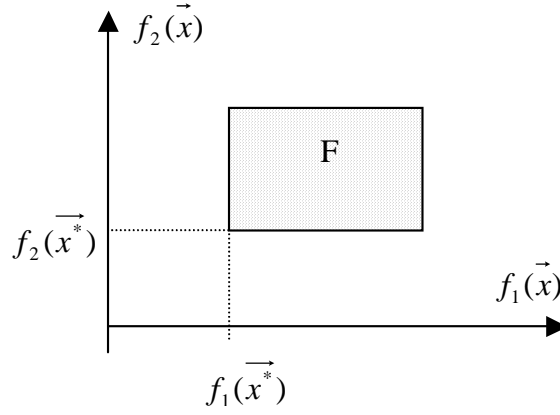


Figura 1-2: Solución ideal

$$\mathbf{x} = \theta \mathbf{x}_1 + (1 - \theta) \mathbf{x}_2, \quad 0 \leq \theta \leq 1$$

Para ilustrar lo anterior se puede decir que un conjunto convexo es una colección de puntos, en donde, si se toman dos puntos del conjunto de manera arbitraria \mathbf{x}_1 y \mathbf{x}_2 , y los unimos mediante una línea, entonces la línea que los une pertenece también al conjunto [63].

Óptimo de Pareto

Cuando se tiene más de una función objetivo, la noción de óptimo cambia, ya que en problemas multicriterio el objetivo es llegar a un buen compromiso, en lugar de una solución única como en optimización global. La noción de óptimo en problemas multiobjetivo fue originalmente propuesta por Edgeworth en 1881 y después generalizada por Pareto en 1896 (ver [73]). El concepto de dominancia de Pareto se presenta formalmente a través de la siguiente definición.

Definición 1.5 *Pareto dominante:* Un vector $\mathbf{u} = (u_1, \dots, u_k)$ se dice que domina a un vector $\mathbf{v} = (v_1, \dots, v_k)$ (denotado por $\mathbf{u} \preceq \mathbf{v}$) si y sólo si \mathbf{u} es parcialmente menor que \mathbf{v} , es decir, $\forall i \in \{1, 2, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, 2, \dots, k\} : u_i < v_i$.

A continuación se da una definición formal de óptimo de Pareto.

Definición 1.6 *Óptimo de Pareto:* Una solución $\mathbf{x}^* \in \Omega$ es óptimo de Pareto, si y sólo si no existe otro $\mathbf{x} \in \Omega$ tal que $f_i(\mathbf{x})$ domine a $f_i(\mathbf{x}^*)$, donde $i \in \{1, 2, \dots, k\}$

Conjunto de óptimos de Pareto

Definición 1.7 *Conjunto de óptimos de Pareto:* dado un problema de optimización multiobjetivo (MOP) con un vector de funciones $\mathbf{f}(x)$, el conjunto de óptimos de Pareto (P^*) se define como:

$$P^* := \{x \in \Omega \mid \neg \exists x' \in \Omega \mathbf{f}(x') \preceq \mathbf{f}(x)\}$$

Frente de Pareto

Definición 1.8 *Frente de Pareto:* Para un MOP dado con un vector de funciones $\mathbf{f}(x)$ y un conjunto de óptimos de Pareto (P^*), el frente de Pareto (PF^*) se define como:

$$PF^* := \{\mathbf{u} = \mathbf{f} = (f_1(x), \dots, f_k(x)) \mid x \in P^*\}$$

No-dominancia débil

Definición 1.9 *No-dominancia débil:* Un punto $\mathbf{x}^* \in \Omega$ es una solución débilmente no-dominada si no hay una $\mathbf{x} \in \Omega$ tal que $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$, para $i = 1, \dots, k$.

No-dominancia fuerte

Definición 1.10 *No-dominancia fuerte:* Un punto $\mathbf{x}^* \in \Omega$ es es una solución fuertemente no-dominada si no existe un $\mathbf{x} \in \Omega$ tal que $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$, para $i = 1, \dots, k$ y para al menos un valor de i que se cumpla: $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$.

Condiciones de Kuhn-Tucker

Definición 1.11 *Condiciones de no inferioridad de Kuhn-Tucker: Si una solución $\mathbf{x} \in \Omega$ es no inferior para un MOP general, entonces existen $w_l \geq 0$, $l = 1, 2, \dots, k$ (w_r es estrictamente positiva para alguna $r = 1, 2, \dots, k$), y $\lambda_i \geq 0$, $i = 1, 2, \dots, m$, tal que:*

$$\begin{cases} \mathbf{x} \in \Omega \\ \lambda_i g_i(\mathbf{x}) = 0 \quad i = 1, \dots, m \end{cases}$$

y

$$\sum_{l=1}^k w_l \nabla f_l(\mathbf{x}) - \sum_{i=1}^m \lambda_i \nabla g_i(\mathbf{x}) = 0 \quad (1.7)$$

donde $g_i(\mathbf{x})$, es la i -ésima restricción.

Estas condiciones son necesarias para que una solución sea no inferior, y cuando todas las $f_l(\mathbf{x})$ son cóncavas y Ω es un conjunto convexo, entonces también son condiciones suficientes.

1.2. Estrategias de optimización global

Los métodos de optimización global se clasifican principalmente en tres categorías: enumerativos, determinísticos y estocásticos. Las estrategias enumerativas son las técnicas de búsqueda más sencillas. En un espacio de búsqueda finito se evalúa cada una de las soluciones posibles. Sin embargo esta opción es ineficiente cuando los espacios de búsqueda son grandes. En cuanto a los algoritmos determinísticos se encuentran principalmente los siguientes: Greedy, Hill-climbing y Branch & Bound [14]. Estas estrategias han sido usadas ampliamente en una gran variedad de casos. Sin embargo, existen problemas de optimización global dimensionalmente grandes, discontinuos, multimodales y/o NP-Completos, que no se pueden resolver con estas técnicas [14]. Debido a que en el mundo real se presentan problemas de optimización global como los mencionados anteriormente,

las técnicas enumerativas y determinísticas resultan inoperantes en su resolución. Para estos casos se han utilizado con mucho éxito los enfoques de búsqueda estocástica y optimización tales como *Simulated Annealing* (SA) [58], *Tabu Search* [42], método de Monte Carlo [61] y Computación Evolutiva ([34], [50] y [7]).

1.3. Técnicas de optimización multiobjetivo

A continuación se revisarán brevemente algunas técnicas tradicionales analíticas, para la resolución de Problemas de Optimización Multiobjetivo (MOP). Primero se mencionará una clasificación de estos enfoques, dependiendo de los criterios que aplique el usuario respecto a la toma de decisiones y los compromisos de cada problema en particular (ver [20] y [74]). Esta clasificación es la siguiente: Algoritmos **a Priori**, **a Posteriori** y **Progresivos**. En los primeros, el usuario (persona o grupo de personas que toman decisiones), debe especificar sus preferencias, expectativas y opiniones antes de ejecutar el algoritmo de optimización. En los métodos a posteriori, después de que se ha encontrado el conjunto de óptimos de Pareto, se presenta éste a la persona encargada de la toma de decisiones (usuario), para que seleccione las mejores alternativas de acuerdo a su criterio. Finalmente, en los métodos progresivos el usuario tiene una participación activa durante el proceso de solución del MOP en cuestión. El conjunto de óptimos de Pareto no es completo, ya que depende de la intervención de la persona que toma las decisiones, basado en el conocimiento del problema, durante el proceso de solución. A continuación se presentarán algunos métodos de acuerdo a la clasificación anterior.

1.3.1. Métodos a priori

Método de programación de metas

Este es uno de los primeros métodos usados para la resolución de problemas de optimización multiobjetivo, fue propuesto por Charnes y Cooper (1961) [22], e Igiri (1965) [39], como un enfoque para usarse en modelos lineales, (este método ha sido aplicado

exitosamente en procesos industriales [14]). En este método el diseñador tiene que asignar metas que desee alcanzar por cada uno de los objetivos. Estos valores se agregan al problema como restricciones adicionales. La función objetivo trata de minimizar la desviación absoluta entre el valor alcanzado y el objetivo. En su forma más simple este enfoque puede ser formulado como sigue:

$$\min \sum_{i=1}^k |f_i(\mathbf{x}) - T_i|, \text{ sujeto a } \mathbf{x} \in \Omega, \quad (1.8)$$

donde T_i representa la meta propuesta por el diseñador para la i -ésima función objetivo $f_i(\mathbf{x})$ y Ω denota el espacio de búsqueda válido.

Método lexicográfico

En este método, la persona que toma decisiones ordena las funciones objetivo de acuerdo a su importancia en el problema, quedando en primer lugar la más importante. Este método evalúa la primera función objetivo (la más importante) sujeta a las restricciones originales. El valor que se obtiene como consecuencia de la minimización de la primera función objetivo se convierte en una restricción más cuando se evalúa la segunda función y este procedimiento se continúa aplicando hasta evaluar la última. Este método se presenta a continuación:

$$\begin{aligned} \min_{x \in \Omega} f_1(x) &= \alpha_1 \\ \min_{x \in \Omega} \{f_2(x) | f_1(x) \leq \alpha_1\} &= \alpha_2 \\ &\vdots \\ \min_{x \in \Omega} \{f_n(x) | f_1(x) \leq \alpha_1, \dots, f_{n-1}(x) \leq \alpha_{n-1}\} \end{aligned} \quad (1.9)$$

La formulación del problema puede escribirse como sigue:

$$\begin{aligned} & \min_{x \in \Omega} f_n(x) \\ \text{sujeto a } & \min_{x \in \Omega} f_k(x) \leq \alpha_k \quad k = 1, 2, \dots, n-1 \end{aligned}$$

La aplicación de este método obtiene buenos resultados cuando el diseñador conoce suficientemente las prioridades de los objetivos a optimizar (los detalles de este método se pueden ver en [62] y [65]).

1.3.2. Métodos a posteriori

Estas técnicas no requieren de información previa proveniente del usuario. Las ideas principales sobre estas estrategias están basadas en las condiciones de Kuhn-Tucker para soluciones no inferiores (Cohon and Marks, 1975 [20]). Entre los métodos más conocidos se pueden mencionar los siguientes:

Combinación lineal de pesos

En 1963 [77], Zadeh descubrió que tomando como base la tercera condición de Kuhn-Tucker sobre soluciones no inferiores se podía plantear un problema de optimización multiobjetivo como un problema de optimización global (un solo objetivo), tomando una función objetivo como la suma de las funciones objetivo (ponderadas por un parámetro w_i). Lo anterior se puede formular mediante la siguiente expresión:

$$\begin{aligned} & \min \sum_{i=1}^k w_i f_i(\mathbf{x}) \\ \text{sujeto a } & \mathbf{x} \in \Omega \end{aligned}$$

donde $w_i \geq 0$ para todo i y es estrictamente positivo para al menos un objetivo. Se puede notar que variando los valores de w_i se obtienen diferentes soluciones del conjunto de óptimos de Pareto.

Método de la Restricción (ϵ)

El método de la restricción (ϵ), ha sido uno de los mas utilizados para resolver problemas de optimización multiobjetivo, convexos y no convexos [49]. Este enfoque reduce un problema multiobjetivo a uno global, minimizando alguna de las funciones objetivo y manejando como restricciones el resto de las otras. Este problema se puede formular como se muestra a continuación:

$$\begin{aligned} & \min_{x \in \Omega} f_l(x) \\ & \text{sujeto a } f_j(x) \leq \epsilon_j \text{ para todo } j = 1, \dots, k, \ j \neq l \\ & x \in \Omega \end{aligned} \tag{1.10}$$

En [49] se demuestra que con el método de la restricción ϵ se obtiene una solución no-dominada débil.

1.3.3. Métodos progresivos

Los métodos progresivos contemplan básicamente tres etapas: primero se encuentra una solución no-dominada; posteriormente se pone a consideración del usuario la solución encontrada, para que de acuerdo a sus preferencias se realicen los cambios necesarios en los objetivos. Finalmente se repiten los dos pasos anteriores hasta encontrar resultados satisfactorios. A continuación se describe uno de estos métodos.

Método del paso (STEP)

La idea básica es converger hacia la mejor solución en el sentido min-max, en no más de k pasos, donde k es el número de objetivos. Esta técnica es útil, principalmente para resolver problemas lineales. Inicia de un punto ideal y se desarrolla en seis etapas, como plantea Cohon en 1978 [21] y [14].

1. Se construye una tabla de soluciones marginales (si es solución única, deberá ser estrictamente no-dominada), por medio de la optimización de cada función objetivo

de manera separada.

2. Se calcula, por cada función objetivo:

$$\alpha_i = \frac{M(i) - m(i)}{M(i)} \left[\sum_{j=1}^J c(i, j) \right]^2, \quad (1.11)$$

donde:

$M(i) = \max f_i(\mathbf{x})$, $m(i) = \min f_i(\mathbf{x})$, y $c(i, j)$ = coeficiente de costo del i -ésimo objetivo lineal.

Se hace el índice de iteración igual a cero ($k = 0$)

3. Se calcula $\Pi(i) = \alpha(i) / \sum \alpha(i)$ y se resuelve el problema min-max. La solución es llamada $x(k)$.
4. Se presenta la solución al usuario, quien procede a tomar las siguientes decisiones:
 - a) Si la solución es satisfactoria, se detiene el algoritmo
 - b) Si la solución no es satisfactoria y $k < p - 1$, ir al paso 5
 - c) Si la solución no es satisfactoria y $k > p - 1$, entonces se detiene. Esto quiere decir que se requiere un procedimiento diferente, o al menos una redefinición del problema.
5. El usuario selecciona un objetivo satisfecho para la solución y determina la cantidad en la cual ésta puede ser decrementada con el fin de incrementar los otros objetivos. Si esto no se puede hacer, entonces se requiere otro enfoque.
6. Se define una nueva restricción que disminuya las pretenciones del objetivo seleccionado en el paso 5. Se hace $\alpha(i) = 0$ para ese objetivo, se incrementa k por uno y se va al paso 3.

Una crítica a esta técnica, es que se asume que no existe una mejor solución, si ésta no es encontrada después de haberse ejecutado los seis pasos de este algoritmo.

En el siguiente capítulo serán tratados los métodos de optimización multiobjetivo, basados en técnicas estocásticas, concretamente los algoritmos genéticos multiobjetivo.

Capítulo 2

Algoritmos Genéticos Aplicados a Optimización Multiobjetivo

2.1. Introducción

Los algoritmos genéticos multiobjetivo, en cuanto a sus principios básicos, se pueden clasificar en aquellos que utilizan en forma directa los conceptos de Pareto y los que no [30]. Los primeros se dice que utilizan métodos directos y los segundos métodos indirectos.

Los métodos indirectos o no basados en Pareto son aquellos en los que el algoritmo evolutivo no incorpora el concepto de óptimo de Pareto en su mecanismo de selección. Un ejemplo es el *Vector Evaluated Genetic Algorithm* (VEGA) propuesto por Schaffer a mediados de los 80s [67], [68] y [69]. VEGA divide la población de un algoritmo genético en tantas subpoblaciones como objetivos existan y en la etapa de selección, cada individuo es seleccionado de acuerdo al objetivo relevante en su subpoblación. Por ejemplo, en la subpoblación 1 se eligen a los mejores individuos con respecto a la primera función objetivo. Después se mezclan aleatoriamente todos los individuos y se procede a la cruce y la mutación como en un algoritmo genético simple.

Los métodos directos tienen su origen en la idea de Goldberg (1989) [34], sobre el concepto de nivel de no-dominancia de Pareto (niveles de jerarquía). El primer nivel de

una población lo componen el conjunto de soluciones no-dominadas (desde el punto de vista de Pareto) de dicha población y se les asigna el rango más alto. El segundo nivel lo componen el conjunto de soluciones no-dominadas de toda la población a la que se han quitado los individuos del primer nivel, y este procedimiento se aplica sucesivamente hasta ordenar toda la población. La función selección tendrá como índice de bondad de un individuo el nivel en que se encuentre. Esto da una idea clara de la distancia del vector de funciones de un individuo respecto del frente de Pareto. Lo anterior, aunado a los operadores de diversidad, constituyen la característica más importantes de un AG multiobjetivo moderno. A continuación se comenta brevemente la importancia de mantener la diversidad en la población de individuos, con el fin de alcanzar el frente de Pareto de un problema de optimización multiobjetivo.

2.1.1. Diversidad en la población

A diferencia de los algoritmos genéticos simples que buscan una solución única, los algoritmos genéticos multiobjetivo tratan de encontrar tantos elementos del conjunto de Pareto como sea posible. Para esto se deben satisfacer principalmente las siguientes metas:

- Convergencia de las soluciones hacia el conjunto de óptimos de Pareto
- Mantener la diversidad en las soluciones dentro del conjunto de óptimos de Pareto

Para poder cumplir con estas metas, los autores que han propuesto los diferentes algoritmos genéticos multiobjetivo utilizan herramientas que mantienen la diversidad en la población, para evitar la convergencia hacia una solución única (algo que tiende a ocurrir después de un número elevado de generaciones por efecto del ruido estocástico). El primer intento para mantener la diversidad en la población surge con la idea de Holland al introducir el concepto de agrupamiento (del inglés *crowding*), donde se intentan identificar situaciones donde los individuos se concentran en espacios o nichos (esto último en referencia a lo que sucede en la naturaleza, donde diferentes especies de animales

se agrupan en función de sus características), compitiendo por los recursos y con ésto disminuyendo las expectativas de vida y la tasa de nacimiento. Pero el fundamento de los algoritmos genéticos multiobjetivo actuales se basa en los conceptos de Goldberg y Richardson (1987) [33], que permite obtener un valor modificado de aptitud para cada uno de los individuos. Para esto se ordena la población en diferentes subpoblaciones (nichos), y la aptitud de una solución en particular, depende de su ubicación dentro de la subpoblación y de la cantidad de soluciones vecinas que ésta tenga. Al número de soluciones vecinas que una solución tenga se le llama contador de nicho y la cercanía entre soluciones se puede medir dentro del espacio de las funciones objetivo mediante una métrica Euclidiana. El valor de bondad o aptitud del individuo i se obtiene de la siguiente expresión:

$$F_{si} = \frac{f_i}{\sum_{j=1}^M \phi(d_{ij})},$$

donde M es el número de individuos localizados en la vecindad del i -ésimo individuo. $\phi(d_{ij})$ se define como la función de compartimiento (del inglés *sharing function*) y su valor se obtiene de:

$$\phi(d_{ij}) = \begin{cases} 1 - (\frac{d_{ij}}{\sigma_{share}})\alpha, & d_{ij} < \sigma_{share} \\ 0 & \text{otro caso} \end{cases}$$

donde normalmente $\alpha = 1$; d_{ij} es una medida de la distancia entre los individuos i y j y σ_{share} es un parámetro que determina la cercanía permitida entre individuos.

A continuación se presentan algunos de los algoritmos genéticos multiobjetivo más relevantes, basados en los principios de Pareto.

2.2. Algoritmos genéticos de primera generación

2.2.1. NSGA (Nondominated Sorting Genetic Algorithm)

Esta estrategia fue propuesta por Srinivas y Deb en 1994 [72]. Se basa en la idea original de Goldberg [34] sobre el uso de niveles de dominación. Se diferencia básicamente de un AG simple en el índice de bondad elegido. Este índice ha sido tomado aplicando el nivel de dominación para obtener el conjunto de Pareto junto con la función índice de bondad compartido para obtener diversidad en la población y así lograr el máximo número de individuos asociados al frente de Pareto. A continuación se muestra el pseudo código de este algoritmo, el cual se encuentra en [72] y [14].

Algoritmo 2.1 *NSGA (Nondominated Sorting Genetic Algorithm)*

```
Se inicializa la población
Se evalúan los individuos de acuerdo a las funciones objetivo
Asignación de jerarquía basado en la dominancia de Pareto en cada capa o frente
Se obtiene el número de contador de nicho
Se asigna el valor de bondad compartido (Shared fitness)
for  $i = 1$  to  $G$ ;  $G$ : número de generaciones
    Se aplica selección mediante muestreo estocástico universal
    Se aplica cruzamiento y mutación
    Se evalúan los individuos de acuerdo a las funciones objetivo
    Se asigna el rango basado en la dominancia de Pareto en cada capa
    Se obtiene el número de contador de nicho
    Se asigna el valor de bondad compartido
end loop ( $i$ )
```

2.2.2. NPGA (Niche-Pareto Genetic Algorithm)

Este método fue propuesto por Horn y Nafpliotis en 1993 (ver [41]), en donde el cambio principal a un AG simple se realiza en la etapa de selección. En la etapa de selección se utiliza la técnica de torneo binario, aplicando los conceptos de nivel de dominación y

contador de nicho. Esta técnica especial de torneo inicia con la selección aleatoria de dos individuos, éstos se comparan con otros individuos, que están dentro de un subconjunto de la población total (típicamente un 10 %) [14]. Si uno de los dos individuos domina a los individuos del subconjunto de la población, entonces es el ganador del torneo. Si los dos dominan o son dominados, entonces se aplica el número del contador de nicho y gana el individuo que esté en un nicho con menor densidad. Fuera de la etapa de selección esta estrategia contiene todos los demás operadores de un algoritmo genético multiobjetivo convencional, como se muestra a continuación:

Algoritmo 2.2 *NPGA (Niche-Pareto Genetic Algorithm)*

```

Se inicializa la población
Se evalúan los individuos de acuerdo a las funciones objetivo
for  $i = 1$  to  $G$ 
    Se aplica selección de torneo especial
        si solamente el individuo 1 es dominado: se selecciona el candidato 2
        si solamente el individuo 2 es dominado: se selecciona el candidato 1
        si los dos candidatos dominan o son dominados (empate):
            selecciona al individuo con el menor número de contador de nicho
    Se aplica cruzamiento en un punto
    Se aplica mutación
    Se evalúan los individuos de acuerdo a las funciones objetivo
end loop ( $i$ ).
```

2.2.3. MOGA (Multi-objective Genetic Algorithm)

En 1993 Fonseca y Fleming [29] propusieron una variación a la técnica de Goldberg, que llamaron “Multi-objective Genetic Algorithm” (MOGA), en el cual la jerarquía de un individuo es igual al número de individuos de la población que lo dominan más uno (los de mayor jerarquía tienen un valor de aptitud igual a uno). Es claro que el conjunto de óptimos de Pareto de la población tienen rango igual a uno. El valor de bondad de un individuo está dado por:

$$\text{Ind.bondad}_{\text{individuo}} = \frac{1}{1 + \text{No. ind. que lo dom.}} \quad (2.1)$$

Este algoritmo utiliza la función índice de bondad compartido (*fitness sharing*), con el objetivo de que no se acumule la población. El espacio donde se mide la distancia de dos individuos es el espacio de las funciones objetivo. Con la intención de que éstos se repartan a lo largo del frente de Pareto, también se aplica el operador “restricción en la reproducción” para tener diversidad e incrementar la velocidad de convergencia del algoritmo.

2.3. Algoritmos genéticos de segunda generación

A continuación serán tratados los algoritmos genéticos multiobjetivo de segunda generación. Se hará mayor énfasis en el NSGA-II, por ser el que se utiliza en esta tesis. La elección de éste se debe principalmente a que es un algoritmo que es representativo del estado del arte y además su implantación computacional es de dominio público.

2.3.1. NSGA-II (Nondominated Sorting Genetic Algorithm-II)

Se trata de un algoritmo genético multiobjetivo de segunda generación propuesto en el año 2000 por Kalyanmoy Deb, A. Pratap, S. Agarwal y T. Meyarivan [24]. Es una versión mejorada de su antecesor, el NSGA, desarrollado también por Deb. Básicamente, el NSGA-II mejora a la versión anterior en tres aspectos fundamentales: mejora el proceso de ordenamiento de las soluciones no-dominadas; en esta etapa, el NSGA tiene una complejidad computacional¹ de $O(MN^3)$, donde M es el número de objetivos y N el tamaño de la población, por $O(MN^2)$ del NSGA-II. La segunda modificación concierne la adición de elitismo y finalmente, el NSGA-II no requiere del parámetro σ_{share} para incrementar la variedad en la población, y para el que no suelen definirse métodos sistematizados de

¹A $O(x)$ se le suele denominar notación “o grande” y denota la complejidad computacional del algoritmo.

elección (en [31] y [32] se da una propuesta para calcular σ_{share}). Otro aspecto importante del NSGA-II es el manejo de restricciones en problemas de optimización multiobjetivo, las cuales se suelen presentar en problemas prácticos.

La mecánica del algoritmo NSGA-II se describirá a continuación explicando primero las diferentes fases que lo componen y después todo en su conjunto. Primero se mencionará la parte que corresponde al ordenamiento de la población en diferentes capas o frentes utilizando el concepto de no-dominancia.

Enfoque rápido de ordenamiento no-dominado

Por claridad se menciona primero un mecanismo simple de ordenamiento de una población de tamaño N en diferentes niveles no-dominados. Después se describe el proceso rápido propuesto por Deb para el NSGA-II.

Para encontrar el primer frente no-dominado se requiere comparar cada uno de los individuos de la población con el resto de la misma para determinar si es no-dominado. Esto requiere $O(MN)$ comparaciones por cada solución (individuo). Para encontrar el primer frente no-dominado se requiere de una complejidad de $O(MN^2)$. Para encontrar el segundo frente no-dominado se separan de la población los individuos del primer frente y se aplica el mismo procedimiento utilizado en el primer caso. Los siguientes niveles son encontrados de la misma forma, requiriéndose para ello un esfuerzo computacional de $O(MN^3)$. A continuación se describe el mecanismo rápido utilizado en el NSGA-II.

Para cada elemento de la población se determina el valor de dos entidades: 1) cuenta de dominación n_p , que es el número de individuos que dominan al individuo p y 2) S_p , que es un conjunto que contiene a los individuos que son dominados por p . Esto requiere $O(MN^2)$ comparaciones. Todas las soluciones del primer frente no-dominado tendrán su cuenta de dominación igual a cero, puesto que ninguna otra solución las domina. Por cada solución del primer frente se revisa cada integrante de su conjunto S_p (denominado por q) y se reduce en uno su grado de dominación. Después de este proceso todos los elementos (q), cuyos contadores de dominación alcancen el valor de cero, serán colocados

en una lista Q . Todos los integrantes de Q pertenecerán al segundo frente no-dominado. Ahora se continúa con el procedimiento anterior con cada miembro de Q y se forma el tercer frente no-dominado. Este proceso se mantiene, hasta que todos los frentes son identificados. A continuación se presenta el seudo código original propuesto por Deb y sus estudiantes [24]:

Algoritmo 2.3 *Enfoque rápido de ordenamiento no-dominado (P), (NSGA-II)*

```

Por cada  $p \in P$ 
     $S_p = \emptyset$ 
     $n_p = 0$ 
    por cada  $q \in P$ 
        if  $(p \prec q)$  then;    sí  $p$  domina a  $q$ 
             $S_p = S_p \cup \{q\}$ ; Se incluye a  $q$  en el conjunto de soluciones dominadas por  $p$ 
        else if  $(q \prec p)$  then
             $n_p = n_p + 1$ ; Incrementa el contador de dominación de  $p$ 
    if  $n_p = 0$  then;  $p$  pertenece al primer frente
         $p_{rank} = 1$ 
         $F_1 = F_1 \cup \{p\}$ 
 $i = 1$ ; inicializa el contador de frente
while  $F_i \neq \emptyset$ 
     $Q = \emptyset$ ; usada para almacenar los miembros del próximo frente
    por cada  $p \in F_i$ 
        por cada  $q \in S_p$ 
             $n_q = n_q - 1$ 
        if  $n_q = 0$  then; entonces  $q$  pertenece al siguiente frente
             $q_{rank} = i + 1$ 
             $Q = Q \cup \{q\}$ 
     $i = i + 1$ 
     $F_i = Q$ .

```

A continuación se describe la segunda fase, encargada de mantener la diversidad en la población.

Preservación de la diversidad en la población

Durante el proceso de convergencia hacia el conjunto de óptimos de Pareto es deseable mantener una gran diversidad de elementos en la población. En el NSGA se usa el enfoque conocido por su nombre en inglés como *Function Sharing* para dar diversidad a la población. Sin embargo, la función mencionada involucra un parámetro llamado *Sharing parameter* (σ_{share}). Este parámetro está relacionado directamente con la medición de la distancia entre los individuos de la población, y como ya se mencionó anteriormente, no existe un procedimiento sistematizado para encontrar el valor σ_{share} . En el NSGA-II se substituyó la función *Sharing* por el enfoque denominado *crowded-comparison*. Este parámetro no requiere ningún valor definido por el usuario, para mantener la diversidad en la población. Por claridad, antes de definir el operador *crowded-comparison* se definirá el concepto de estimación de densidad conocido como: *density-estimation metric*.

Estimación de densidad (density-estimation metric)

Para obtener una estimación de la densidad de los individuos alrededor de un punto en particular, se calcula la distancia promedio de dos puntos vecinos, uno en cada lado del punto de interés, utilizando el valor de sus funciones objetivo. Esta cantidad, $i_{distance}$, sirve como una estimación del perímetro de un cuboide tomando como vértice el punto vecino más cercano, como se muestra en la Figura 2-1.

En la Figura 2-1 el valor de la *crowding distance*, del i -ésimo individuo en su frente (marcados como círculos llenos), es la longitud promedio de sus lados. El proceso computacional para hallar la *crowding-distance*, requiere ordenar la población de acuerdo con el valor de cada función objetivo, en orden de magnitud ascendente. Después, por cada función objetivo se asignan valores de distancia infinitos a los individuos que están en los extremos (individuos con el menor y mayor valor). A todos las demás soluciones (individuos), se les asigna un valor de distancia, igual al valor absoluto de la diferencia (normalizada) del valor de las funciones correspondientes a los puntos adyacentes. Estos cálculos se hacen con todas las funciones objetivo. El valor *crowding-distance* total

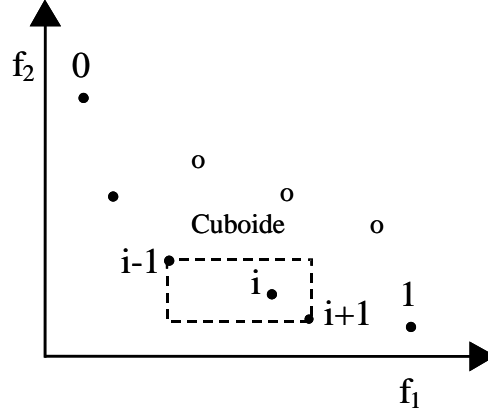


Figura 2-1: Distancia de agrupamiento

se calcula como la suma de los valores de todas las distancias de los elementos correspondientes a cada objetivo. Se normaliza cada función objetivo antes del cálculo de la *crowding-distance*. A continuación se muestra el procedimiento computacional para calcular la *crowding-distance* de todas las soluciones en un conjunto no-dominado (I).

Algoritmo 2.4 *Asignación de distancia de agrupamiento (I), (NSGA-II)*

$l = |I|$; número de soluciones en I
 para cada i , $I[i]_{distance} = 0$; se inicializa el contador de distancia
 por cada objetivo m
 $I = sort(I, m)$; se ordena usando el valor de cada función objetivo
 $I[1]_{distance} = I[l]_{distance} = \infty$; acotamiento para los valores extremos
 for $i = 2$ to $(l - 1)$; para todos los puntos

$$I[i]_{distance} = I[i]_{distance} + (I[i + 1].m - I[i - 1].m) / (f_m^{m\acute{a}x} - f_m^{m\acute{i}n}).$$

En el pseudocódigo anterior, $I[i].m$ se refiere al valor de la m -ésima función objetivo, correspondiente al individuo i -ésimo, en el conjunto I . Los parámetros $f_m^{m\acute{a}x}$ y $f_m^{m\acute{i}n}$ son el máximo y el mínimo valor de la m -ésima función objetivo. La complejidad computacional de este proceso está relacionada directamente con el proceso de ordenamiento, ya que se requieren M ordenaciones independientes de a lo más N individuos. Cuando toda la población está en el frente I , la complejidad que tiene es $O(MN \log N)$. Después que todos

los miembros del conjunto I tienen asignado un valor de distancia se pueden comparar 2 integrantes por su separación o proximidad con otros individuos. Una solución con valor de distancia pequeño tiene más vecinos y por lo tanto es menos deseable. A continuación se aplica el concepto anterior para definir el *crowded-comparison operator*.

Operador “crowded-comparison”

El operador *crowded-comparison* (\prec_n) guía el proceso de selección encaminando directamente al algoritmo hacia un frente óptimo de Pareto. Se asume que cada individuo i de la población tiene dos atributos:

1. Jerarquía de no-dominación (i_{rank});
2. Distancia de agrupamiento ($i_{distance}$)

ahora se define un orden parcial \prec_n como

$$\begin{aligned} i \prec_n j \text{ if } & (i_{rank} < j_{rank}) \\ \text{or } & ((i_{rank} = j_{rank}) \\ \text{and } & (i_{distance} > j_{distance})) \end{aligned}$$

Lo anterior significa que entre dos soluciones con diferentes rangos de no-dominación, se prefiere la solución con el menor rango (la mejor). En otro caso si dos soluciones pertenecen al mismo frente, entonces se prefiere la solución que está ubicada en una región de menor densidad. Después de haber presentado los conceptos de: procedimiento de ordenamiento rápido (*fast nondominated sorting procedure*), operador de estimación de distancia de agrupamiento (*fast crowded distance estimation*) y el operador de comparación de agrupamiento (*crowded comparison operator*), se está en condiciones de describir el NSGA-II completo.

Ciclo completo del NSGA-II

Inicialmente y de manera aleatoria, se crea una población P_0 . Esta población es ordenada en base al principio de no-dominación. A cada solución se le asigna un valor de

bondad (o jerarquía), igual al nivel de su dominación (el nivel 1 es el mejor, el 2 es el siguiente mejor y así sucesivamente). En el procedimiento anterior se aplica el principio de minimización del valor de aptitud. Al principio se aplica el tipo de selección de torneo (*tournament selection*) en forma usual, así como los operadores de cruzamiento y mutación, para crear una población de descendientes Q_0 de tamaño N . El concepto de elitismo se aplica al comparar a la población actual con la mejor población de individuos no-dominados encontrados previamente. El procedimiento difiere después de producir a la generación inicial, como se describe a continuación con la i -ésima generación.

Primero se obtiene una población formada por $R_t = P_t \cup Q_t$ de tamaño $2N$. Luego, la población R_t es ordenada por niveles de acuerdo a los principios de no-dominación. Así la combinación de las poblaciones previa y actual en R_t aseguran el proceso de elitismo. Después de que son ordenadas, la mejores soluciones pertenecen al conjunto F_1 y éstas van a ocupar un papel preponderante durante el proceso. Los siguientes mejores individuos pertenecerán al conjunto F_2 y así sucesivamente hasta formar el último frente F_l . Si el tamaño de F_1 es menor a N , entonces se deben considerar todos sus elementos para formar una nueva población P_{t+1} . Los espacios pendientes en la nueva población serán llenados por los subsiguientes niveles no-dominados. Este procedimiento continúa hasta que ningún conjunto más puede ser acomodado. Seguramente que la suma de los conjuntos F_1 a F_l es mayor que N . Para solucionar esta situación se ordena en forma descendente el último conjunto F_l usando el operador de *crowded-comparison* (\prec_n) para seleccionar las mejores soluciones y llenar exactamente el espacio de la nueva población de tamaño N . El NSGA-II se describe gráficamente en la Figura 2-2. A la nueva población P_{t+1} se le aplican los operadores de selección cruzamiento y mutación para formar una nueva población Q_{t+1} de tamaño N . Es importante decir que en la técnica de selección de torneo binario se aplica el operador *crowded-comparison* (\prec_n). Este operador requiere que se conozca la jerarquía y la distancia de agrupamiento (*crowded-distance*), de cada solución en la población. Estos últimos valores se calculan mientras se está formando la población P_{t+1} como se muestra en el siguiente algoritmo:

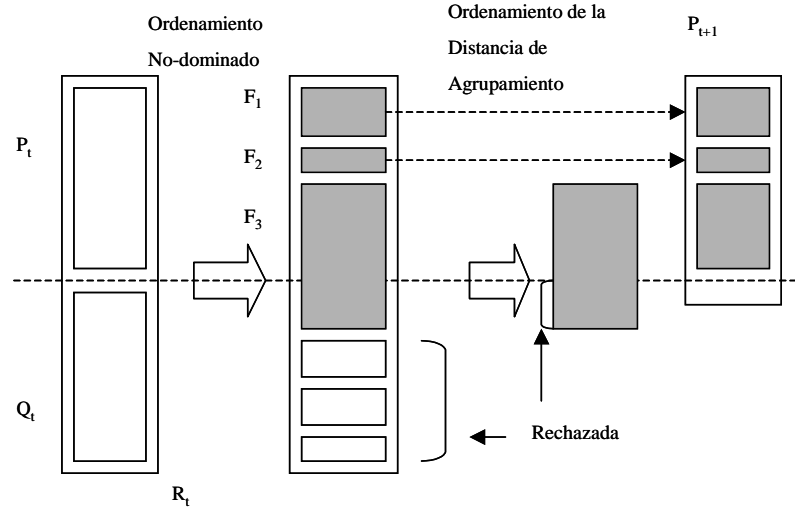


Figura 2-2: Procedimiento del NSGA-II

Algoritmo 2.5 *Procedimiento completo del NSGA-II*

$R_t = P_t \cup Q_t$; combina padres e hijos de la población

F =ordenamiento rápido no dominado(R_t); $F = (F_1, F_2, \dots)$, todos los frentes no dominados de R_t

$P_{t+1} = \emptyset$; $i = 1$

until $|P_{t+1}| + |F_i| \leq N$; hasta que la población de padres sea llenada

Asignación de la distancia de agrupamiento (F_i); calcula la distancia de agrupamiento en F_i

$P_{t+1} = P_{t+1} \cup F_i$; incluye el i -ésimo frente no-dominado en la población de padres

$i = i + 1$; revisa el próximo frente para inclusión

$sort(F_i, \prec_n)$; ordena en forma descendente usando \prec_n

$P_{t+1} = P_{t+1} \cup F_i[(1 : (N - |P_{t+1}|))]$; busca los primeros $(N - |P_{t+1}|)$ elementos de F_i

Q_{t+1} =reproducción (P_{t+1}); uso de selección, cruzamiento y mutación para crear la nueva pobl. Q_{t+1}

$t = t + 1$; incremento del contador de generación.

La complejidad computacional (temporal), del NSGA-II completo es $O(MN^2)$ cuya aportación principal es la fase de ordenación no-dominada.

Enfoque de manejo de Restricciones en el NSGA-II Con este método se usa la estrategia de selección de torneo binario, en el cual se toman dos soluciones de la población y se elige la mejor. Si el problema de optimización presenta restricciones, cada

una de las soluciones puede ser válida o inválida. En este caso se pueden presentar tres diferentes situaciones: 1) las dos soluciones son válidas, 2) una de ellas es válida y la otra no y 3) ninguna de las dos es válida. En un problema de optimización global (un objetivo), se usa una regla simple por cada caso.

Caso 1) Se elige la solución con mejor índice de bondad

Caso 2) Se elige la solución válida

Caso 3) Se elige la solución que presente la menor suma total de restricciones violadas.

En cuanto a optimización multiobjetivo, los casos 2 y 3 pueden ser usados como se presentaron anteriormente y el primero puede ser resuelto usando el operador *crowded-comparison*. Para mantener la modularidad en el NSGA-II sólo se modifica la definición de dominación, llamándole a ésta como “dominancia-restringida”. A continuación se presenta la definición de dominancia-restringida.

Definición 2.1 *Dominancia-restringida: Con la definición de dominancia-restringida (del inglés constrained-dominate), aplicada a i y j , se dice que i domina a j si se cumple alguna de las siguientes condiciones:*

1. *La solución i es válida y la solución j no.*
2. *Las dos soluciones son inválidas, pero la solución i tiene menor suma total de restricciones violadas.*
3. *Las dos soluciones son válidas y la solución i domina a la solución j .*

El efecto de usar el principio de dominación restringida, es que cualquier solución válida tiene una mejor jerarquía de no-dominación que cualquier solución inválida. Además todas las soluciones válidas son jerarquizadas de acuerdo con su nivel de no-dominación basados en los valores de las funciones objetivo. Sin embargo entre dos soluciones inválidas, la solución con menor valor a la violación de restricción tiene mejor jerarquía. También es importante mencionar que los cambios en la definición de no-dominación no incrementa el nivel de complejidad computacional del NSGA-II.

2.3.2. SPEA (Strength Pareto Evolutionary Algorithm)

Este algoritmo fue presentado por Zitzler y Thiele en 1999 [78]. Fue concebido como un medio de integración de los demás algoritmos evolutivos para optimización multiobjetivo existentes en aquel entonces (ver por ejemplo [14], [15] y [16]). SPEA usa un archivo que contiene las soluciones no-dominadas encontradas previamente (también se le conoce como conjunto externo no-dominado). En cada generación, los individuos no-dominados se copian a este conjunto. Para cada individuo de este conjunto se determina un valor, llamado fortaleza (del inglés *strength value* y se usa este concepto en el sentido de dominación), el cual es proporcional al número de soluciones que cierto individuo domina. El valor de bondad de cada uno de los integrantes de la población actual es calculado de acuerdo al valor de fortaleza de todas las soluciones externas no-dominadas que éste domina. Para proveer diversidad a la población se utiliza una técnica de clustering llamada método del enlace promedio (*average linkage method*), que se basa en penalizar a las soluciones vecinas más cercanas a la que se está evaluando. A continuación se presenta el pseudocódigo de este método.

Algoritmo 2.6 SPEA (*Strength Pareto Evolutionary Algorithm*)

```
Se inicializa una población  $P$ 
se crea un conjunto vacío externo  $E$ 
for  $i = 1$  to  $G$ 
    se copian las soluciones no-dominadas de  $P$  a  $E$ 
    se remueven los elementos de  $E$  que son dominados por otro miembro de  $E$ 
    podar  $E$  (usando clustering), cuando la capacidad máxima de  $E$  esté siendo excedida
    se calcula el valor bondad de cada individuo en  $P$  y en  $E$ 
    se usa torneo binario con reemplazo para seleccionar elementos de  $P + E$  (operación de unión)
        hasta que la piscina de padres (mating pool) esté llena
    se aplica cruzamiento y mutación.
fin del ciclo ( $i$ )
```

2.3.3. SPEA-2

Fue dado a conocer por Zitzler, Laumanns y Thiele en el año 2001 [79]. Presenta tres principales diferencias con respecto a su antecesor el SPEA: 1) incorpora una estrategia de bondad de grano-fino (*fine-grained fitness*), que por cada individuo toma en cuenta el número de individuos que domina y el número de individuos que lo dominan, 2) Uso de una técnica de estimación del vecino más cercano que guía la búsqueda de una manera más efectiva, y 3) presenta un método de truncamiento (*enhanced archive truncation*), de las soluciones que están en los extremos del archivo externo o población. Esto garantiza que las soluciones se mantengan acotadas. El SPEA-2 reduce la complejidad del algoritmo de *clustering* utilizado.

2.3.4. PAES (Pareto Archived Evolution Strategy)

Este método fue presentado por Knowles y Corne en el año 2000 [59] y consiste en una estrategia de evolución (1 + 1) (esto es, un solo padre genera un solo hijo), en combinación con un archivo histórico que almacena algunas de las soluciones no-dominadas encontradas previamente. Este archivo es usado como un conjunto de referencia contra el cual se compara cada individuo mutado. A continuación se presenta el pseudocódigo de este método:

Algoritmo 2.7 PAES (*Pareto Archived Evolution Strategy*)

```
Se inicializa una población con un único padre ( $p$ ), y se le incluye en un archivo externo
se muta  $p$ , para obtener un individuo hijo( $c$ ), y se realiza la evaluación de su aptitud (fitness)
  if (  $p$  domina a  $c$  ) se desecha  $c$ 
  else if (  $c$  domina a  $p$  )
    se reemplaza  $p$  por  $c$  y se incluye en el archivo
  else if (  $c$  es dominada por algún otro miembro del archivo ) se desecha  $c$ 
  else se prueba ( $p$ ,  $c$  y el archivo), para determinar cuál llegará a ser la nueva solución actual y si se
    incluirá a  $c$  en el archivo
mientras no se cumpla un número de repeticiones preestablecido, regresar a la segunda línea de este
procedimiento (donde se muta a  $p$ ).
```


PAES utiliza como novedad un enfoque para mantener la diversidad que consiste de un procedimiento de vecindad que divide el espacio de objetivos en una forma recursiva. Cada solución es ubicada en una cierta parte de una malla, basada en los valores de sus objetivos (que son usadas como sus coordenadas). Por ejemplo, si el problema tiene dos funciones objetivo, el eje de las abscisas corresponderá a la función uno y el eje de las ordenadas a la función dos. También se mantiene un mapa de la malla, indicando cuántas soluciones hay en cada rejilla. Una de las bondades de este método consiste en presentar un grado de complejidad casi lineal en el proceso de preservación de diversidad, por un grado N^2 en el caso del enfoque de nichos.

2.3.5. Algoritmos MRCD y MRCD min-max.

Los Algoritmos MRCD y MRCD min-max son dos algoritmos genéticos multiobjetivo, desarrollados en la tesis de Herreros [36], para la resolución de problemas de diseño multiobjetivo y diseño robusto multiobjetivo, planteados en forma paramétrica. El acrónimo MRCD proviene de la denominación inglesa *Genetic Algorithm for multiobjective Robust Control Design* (ver [37]). Son algoritmos que usan de forma directa los conceptos de optimización de Pareto para hallar su frente y los óptimos asociados, y son algoritmos a posteriori ya que tratan de obtener el conjunto completo de dichos óptimos, dejando para una segunda fase la elección de uno de ellos.

Algoritmo MRCD El algoritmo MRCD está dotado de operadores que modifican de forma dinámica los límites del espacio de búsqueda, y de una estructura de “Algoritmo paralelo de isla”. Esto quiere decir que la población se divide en pequeñas subpoblaciones y los operadores genéticos se aplican en forma paralela a cada una de estas subpoblaciones y sólo después de un número dado de generaciones hay un intercambio de individuos entre ellas. A continuación se presenta el Pseudocódigo del algoritmo MRCD, tomado de [36].

Algoritmo 2.8 MRCD

inicio

```

 $gen \leftarrow 0$ 
repetir
     $gen \leftarrow gen + 1$ 
    desde  $k = 1$  a  $k = Nsubpop$ 
    inicio
         $[poblac(k), ondad(k)] \leftarrow InPob(espacio(k))$ 
    fin
    desde  $k = 1$  a  $k = Nsubpop$ 
    inicio
         $espacio(k) \leftarrow Mod\Omega(poblac(k), gen)$ 
         $[selec(k), elite(k)] \leftarrow Select(poblac(k), bondad(k))$ 
         $[newpoblac(k), newbondad(k)] \leftarrow Mecl(poblac(selec(k))..$ 
             $bondad(select(k), espacio(K))$ 
         $elite(k) \leftarrow Filt(poblac(elite(k)), bondad(elite(k)))$ 
         $poblac \leftarrow [newpoblac(k), poblac(elite(k))]$ 
         $bondad \leftarrow [newbondad(k), bondad(elite(k))]$ 
    fin
    desde  $k = 2$  a  $k = Nsubpop$ 
    inicio
         $[poblac(k), bondad(k)] \leftarrow Migra(poblac(k), bondad(k), poblac(k - 1), bondad(k - 1))$ 
    fin
     $poblac(1) \leftarrow Migra(poblac(1), bondad(1), poblac(Nsubpop), bondad(Nsubpop))$ 
hasta (condición parada)

fin.

```

Los límites del espacio de búsqueda serán modificados (por medio de la función $Mod\Omega$) ciclicamente en relación del número de generaciones transcurridas y del operador de migración. La modificación de estos límites será función de los valores máximos y mínimos de los parámetros de la población.

Algoritmo MRCD min-max

Este algoritmo es una variante sobre el algoritmo MRCD simple. Con él se desea resolver el típico problema de min-max, donde el vector de funciones objetivo debe ser minimizado respecto a una serie de parámetros y maximizado con respecto a otra serie.

El objetivo final de este algoritmo es su aplicación a control multiobjetivo de sistemas con incertidumbre paramétrica. Este problema se puede plantear como la minimización del vector de funciones objetivo con respecto a los parámetros del controlador, y la maximización de dichas funciones con respecto a los parámetros de la incertidumbre. La estructura del algoritmo es la misma que la del algoritmo MRCD, siendo su principal diferencia el operador filtro de la élite, $Filt(\cdot)$. A continuación se describe brevemente las principales diferencias de este algoritmo con respecto al MCRD simple.

Población inicial

La población en este caso está compuesta por individuos divididos en dos clases de parámetros: la parte m que corresponde a aquellos con respecto a los cuales se va a minimizar el vector de funciones objetivo y la parte M que corresponde a aquellos con respecto a los cuales se va a maximizar. Suponiendo que el número de los primeros sea p y el de los segundos q , el fenotipo (parámetros) de un individuo queda de la siguiente forma:

$$k = [m|M]^T = [m_1, \dots, m_p | M_1, \dots, M_q]$$

El operador $Mod\Omega$ sólo afecta al espacio m , ya que el espacio M corresponde a los parámetros de la incertidumbre y éstos son fijos.

Filtro en la élite

Ésta es la diferencia principal entre este algoritmo MRCD min-max y el simple. La idea es modificar la parte M de los individuos de la élite con el fin de conseguir maximizar el vector de funciones objetivo.

Capítulo 3

Controladores PID

3.1. Controladores PID

3.1.1. Introducción

La idea de retroalimentación ha demostrado ser muy poderosa en la ingeniería. Su uso ha tenido consecuencias drásticas en el desempeño de sistemas (e. g. Bennett, 1979, 1993 [8] y [9]). Prácticamente todo el crédito se le puede dar a un tipo simple de estrategia de control: a la que caracteriza a los controladores proporcionales, integrales y derivativos mejor conocidos como controladores PIDs.

Los controladores PIDs son los más usados en los procesos industriales. Sus principales propiedades son: se retroalimenta a la planta bajo control, la eliminación del error estacionario a partir del efecto integrador y algunas ventajas adicionales debido a su efecto derivativo como agregar amortiguamiento a la acción de control. En los procesos industriales más del 95 % de los lazos de control son del tipo PID, incluso la mayoría de ellos son controladores PI debido a la eliminación del efecto derivativo ya que produce inconvenientes debido al ruido generado principalmente a nivel del sensor (o sensores). En la actualidad a pesar de los avances en la teoría de control y en las tecnologías de control de las últimas décadas, los PIDs se siguen usando frecuentemente. La aparición de

las computadoras como elementos de control ha hecho que en muchos casos los PIDs sean implantados como programas de cómputo, incorporándose nuevos elementos a su estructura básica, tales como la sintonización automática o la adaptación de sus parámetros a los cambios de la planta que debe controlar.

En procesos industriales de gran escala, pueden haber cientos de controladores PIDs o PIs y tienen que ser sintonizados individualmente para garantizar un desempeño adecuado y robusto. En la mayoría de los casos el procedimiento de sintonizado es hecho manualmente, lo cual es muy tedioso y consume más tiempo del necesario. El desempeño del sistema resultante depende principalmente de la experiencia y el conocimiento que tenga el ingeniero del proceso. En la práctica es reconocido que muchos lazos de control están pobremente sintonizados. En consecuencia las técnicas de sintonizado automático se hacen más atractivas a investigadores e ingenieros de procesos. Por sintonizado automático (o auto-sintonizado), se entiende un método que permite que el controlador sea calibrado de manera automática, es decir, que de manera automática se calibren los parámetros del controlador para obtener un desempeño óptimo del sistema de lazo cerrado, cuando lo demande un operador externo o una señal externa. Típicamente, el usuario aprieta un botón o envía un comando al controlador.

A pesar de la existencia de teoría, con que actualmente se cuenta, sobre PIDs (ver por ejemplo [23]), en muchas aplicaciones industriales todavía se siguen utilizando las reglas de Ziegler-Nichols (Z-N) para sintonizar controladores PID. Ésto es una situación bastante inadecuada, porque es conocido que las reglas (Z-N) dan resultados muy pobres en muchos casos, por ejemplo éstas dan una respuesta oscilatoria ante cambios de referencia (Åström y Hägglund 2001 [5]).

En el caso del uso de PIDs, para el buen desempeño de sistemas industriales se requiere de especificaciones de diseño complicadas para los controladores PID, algunas de éstas son las siguientes:

- Diseñar un controlador robusto que mantenga las variables del proceso razonablemente cerca de los valores deseados.

- Diseñar un controlador que mantenga las variables del proceso tan cerca como sea posible de las especificaciones dadas.
- Diseñar un controlador donde la variables de salida del proceso pueda seguir variaciones en la referencia.
- Diseñar un controlador que mantenga las variables del proceso dentro de un rango dado.

Además, el controlador PID debe enfrentar problemas como los siguientes:

- Los sistemas tienen no-linealidades tales como direccionalidad dependiente del actuador y dinámicas no modeladas.
- El controlador debe trabajar con varios tipos de incertidumbre, como errores de modelado y perturbaciones externas a la planta (proceso).
- Puede ser necesario un sintonizado subóptimo para enfrentar cambios en el sistema producidos por el tiempo y el desgaste.

Como resultado de estas dificultades, en pocas ocasiones los controladores PID son sintonizados óptimamente y además, los ingenieros necesitan establecer un compromiso entre el desempeño y el tiempo disponible para lograrlo. Esto hace que el sintonizado sea más subjetivo, y dependa del ingeniero que lo realice.

Un inconveniente de los métodos clásicos de sintonización de PIDs es su gran sensibilidad a cambios en el modelo de la planta [26] y[4]. Cuando la planta es un sistema con múltiples entradas y múltiples salidas (MIMO), los métodos clásicos de sintonización de PIDs consideran las entradas y las salidas como independientes, sin tener en cuenta los posibles acoplamientos por lo que los controladores obtenidos generalmente tienen un desempeño pobre.

3.1.2. Objetivos

Los siguientes criterios de desempeño son los objetivos que se desean alcanzar en el presente trabajo utilizando métodos heurísticos:

1. Optimizar la rapidez de respuesta ante una señal de referencia.
2. Minimizar la sensibilidad ante cambios en la estructura de la planta para obtener robustez frente a estos cambios.
3. Atenuar el efecto de perturbación en la carga.
4. Minimizar el posible efecto del ruido sobre la señal de medición.
5. La amplitud de la señal de salida de control no debe sobrepasar de un valor dado para evitar saturaciones en el actuador.

Los objetivos anteriores corresponden a las siguientes especificaciones para el diseño de un controlador PID:

- Seguimiento de la señal de referencia (referencia constante).
- Atenuación de perturbación en la carga.
- Robustez respecto de las incertidumbres del modelo.
- Atenuación del ruido en la medición.
- Límites de la señal de control (amplitud de la señal considerando todo el espectro de frecuencia).

La importancia de cada especificación está en función de la aplicación de control. En el control de procesos, la atenuación de la perturbación de carga es fundamental para mantener la referencia requerida. Al ser este ruido de baja frecuencia se modela como una señal escalón en la entrada de la planta. En otras aplicaciones, como en control de

motores y robótica, un buen seguimiento de la señal de referencia es muy importante. Otro requerimiento fundamental es la robustez respecto a las posibles incertidumbres de la planta. Muchos de los métodos clásicos para la sintonización de PIDs, como la regla de Z-N, proporcionan PIDs que carecen de robustez, lo que origina que pequeños cambios en la planta desestabilicen al sistema en lazo cerrado.

3.2. Los enfoques más comunes para síntesis de controladores PID

En esta sección se expone un breve repaso de algunos de los métodos de síntesis de controladores PI(D) basados en modelo. El objetivo es dar una idea de cómo funcionan estos métodos de sintonización.

Síntesis basada en modelo

La propiedad clave de los métodos basados en modelo es que la descripción del proceso es precisamente un modelo matemático. Por lo que estos métodos dependen de alguna técnica de identificación que debe dar un modelo de estructura fija y simple, debido a la necesidad de obtener reglas simples de sintonización. Cuando métodos basados en modelo son usados manualmente es de gran ayuda representar el proceso con un modelo que permita conocer qué tan sensible es a variaciones en sus parámetros, para poder seleccionar entonces los posibles parámetros del controlador. Cuando estos parámetros son usados en un autosintonizador, es importante verificar que los resultados de identificación sean accesibles al usuario, ya que éstos pueden ser una fuente importante de información para hacer diagnósticos sobre el proceso.

El método Haalman

El método Haalman [35] refiere a un PID ideal de un grado de libertad, es decir, de acuerdo a la siguiente ley de control de un PID ideal:

$$C_{PID_1}(s) = K \left(1 + \frac{1}{sT_i} + sT_d \right),$$

donde K denota la ganancia proporcional, T_i y T_d denotan las ganancias integral y derivativa respectivamente.

La idea es comenzar de un modelo de primer o segundo orden con un retardo puro de tiempo y seleccionar los parámetros del controlador, proponiendo que la función de transferencia de lazo abierto $L(s)$ sea:

$$L(s) = \frac{2e^{-sL}}{3Ls},$$

la cual corresponde a una frecuencia de corte ω_c de $\frac{2}{3L}$ y un margen de fase φ_m de 50° aproximadamente; la razón $2/3$ se obtiene al minimizar el promedio del error al cuadrado, cuando se aplica una señal escalón en la entrada de referencia del sistema de control [3]. El objetivo es hacer que la función de transferencia de lazo cerrado se comporte como $L(s)/(1 + L(s))$. Una vez que $L(s)$ ha sido definida, los parámetros del controlador son calculados aplicando la relación $R(s) = L(s)/M(s)$ donde $M(s)$ es la función de transferencia del modelo del proceso. Si se usa un modelo de primer orden con retardo ($M(s) = e^{-sL}/(1 + sT)$) se obtiene un controlador PI cuyos parámetros son determinados a partir de las siguientes expresiones:

$$K = \frac{2T}{3L}, \quad T_i = T$$

Mientras que para un modelo de segundo orden con retardo ($M(s) = e^{-sL}/(1 +$

$sT_1)(1 + sT_2))$ se obtiene un controlador PID y las formulas de sintonización son:

$$K = \frac{2(T_1 + T_2)}{3L}, \quad T_i = T_1 + T_2, \quad T_d = \frac{T_1 T_2}{T_1 + T_2}.$$

El método Haalman es adecuado para procesos con respuesta sobreamortiguada y un retraso significativo. En efecto, siendo ω_c inversamente proporcional a L , la respuesta puede ser muy rápida si L es pequeña. Sin embargo, cuando se aplica éste metodo, se presenta la desventaja de que los polos y ceros del controlador cancelan los polos y ceros del proceso, acarreando como consecuencia, en algunos casos, efectos indeseados en el funcionamiento de lazo cerrado [3].

Una versión modificada del método ha sido propuesta en [66], el cual funciona para un modelo de primer orden con retardo y un controlador PI, un margen de fase mínima φ_m y una frecuencia de corte máxima ω_c , tomando las condiciones más restrictivas. Esto guía a:

$$K = \min \left(\frac{T(\pi/2 - \varphi_m)}{L}, T\omega_c \right), \quad T_i = T,$$

donde φ_m y ω_c son los parámetros de diseño. Una regla para seleccionarlos es fijar φ_m a un mínimo razonable (digamos 50°), mientras que ω_c puede ser calculado con el fin de imponer que el tiempo (esperado) de establecimiento de lazo cerrado, la cual es igual a $5\omega_c$, sea “ β veces más pequeño” que la del modelo del proceso. En el caso de un modelo de primer orden con retardo, donde el tiempo de establecimiento del modelo puede ser expresado como $L + 5T$, ésto significa que

$$\omega_c = \frac{5\beta}{L + 5T}$$

donde β puede variar de 4 a 10. Ésto puede ser interpretado como un factor de aceleración. El método de Haalman modificado no requiere necesariamente de definir la función $L(s)$; en este método se usan los parámetros φ_m y ω_c para alcanzar el desempeño de lazo cerrado deseado.

El método del óptimo simétrico

El método del óptimo simétrico [3], también refiere al PID ideal de un grado de libertad $C_{PID_1}(s)$. Contiene varias ideas que han sido ampliamente desarrolladas en las últimas décadas. Lo más importante es suponer que el modelo del proceso es de la forma:

$$M(s) = \mu \frac{e^{-sL}}{\prod_{h=1}^n (1 + sT_h) \prod_{k=1}^m (1 + sT_k)}$$

donde para un modelo de primer orden con retraso $m = 1$ y para un modelo de segundo orden con retraso $m = 2$, más algunos polos adicionales por las dinámicas no modeladas. También se supone que las constantes de tiempo son dominantes, es decir:

$$T_k \gg \sum_{h=1}^n (1 + sT_h) \quad \forall k.$$

La cantidad

$$T_{um} = L + \sum_{h=1}^n (1 + sT_h)$$

puede ser interpretada como la constante de tiempo de una función de transferencia que representa las dinámicas no modeladas.

El método del óptimo simétrico toma al siguiente modelo aproximado

$$\widehat{M}(s) = \mu \frac{e^{-sL}}{(1 + sT_{um}) \prod_{k=1}^m (1 + sT_k)}$$

y se diseña el controlador para que la frecuencia de corte sea $1/2T_{um}$ (reduciendo la demanda cuando las diferencias entre el modelo y el proceso es importante) y la magnitud de lazo abierto $|R(j\omega)\widehat{M}(j\omega)|$ tiene una pendiente de -20 dB/dec en el intervalo de frecuencia $1/4mT_{um}$ a $1/T_{um}$. Este es un método de síntesis de controladores PI(D) basado en características. Para un modelo de segundo orden con retardo donde $T_1 \gg T_2$, las formulas de sintonización para el método del óptimo simétrico se muestran en la Tabla 3.1

	K	T_i	T_d
PI	$\frac{T_1}{2\mu T_{um}}$	$4T_{um}$	
PID ($T_2 \geq 4T_{um}$)	$\frac{T_1 T_2}{8\mu T_{um}^2}$	$16T_{um}$	$4T_{um}$
PID ($T_2 \geq 8T_{um}$)	$\frac{T_1(T_2+4T_{um})}{8\mu T_{um}^2}$	$T_2 + 4T_{um}$	$\frac{4T_2 T_{um}}{T_2+4T_{um}}$

Cuadro 3.1: Sintonización mediante óptimo simétrico

El método del óptimo simétrico tiene buen desempeño si el retraso del proceso es pequeño y la constante de tiempo T_k debe también dominar a L . Se pueden generar sobrepasos en la respuesta a la señal de referencia, lo cual se puede evitar si se filtra la señal de referencia o se usa una señal de referencia ponderada [3].

El método Dahlin o λ -sintonizado

Dado un modelo de primer orden con retraso, el objetivo del método Dahlin es hacer que la función de transferencia de la señal de referencia a la variable controlada sea similar a un modelo de primer orden con ganancia unitaria, el mismo retraso que el modelo del proceso y una constante de tiempo específica, la cual será el parámetro de diseño. Denotando esta constante de tiempo con λ (lo cual motiva el nombre del método) éste corresponde a sintonizar el controlador para que éste pueda ser aproximado por:

$$R(s) = \frac{1 + sT}{\mu(1 + s\lambda - e^{-sL})}.$$

Si el término e^{-sL} es reemplazado por su aproximación de Padé (1,0), es decir por $1 - sL$, esta aproximación es un PI. Conversamente, si una aproximación de Padé (1,1) es usada, es decir $(1 - sL/2)/(1 + sL/2)$, se obtiene un PID. En síntesis, las formulas de sintonización del método de Dahlin son:

$$K = \frac{T}{\mu(L + \lambda)}, \quad T_i = T$$

para el PI, y:

$$K = \frac{T + L/2}{\mu(L + \lambda)}, \quad T_i = T + L/2, \quad T_d = \frac{TL/2}{T + L/2}$$

para el PID. El método está referido a un PID $C_{PID_1}(s)$ ideal de un grado de libertad y está basado en modelo. Es una buena técnica pero requiere una selección razonable de λ . Sin embargo, la experiencia ha mostrado que una λ grande (es decir un peor desempeño) se requiere cuando las diferencias entre el modelo y el proceso es más significativo. Por lo tanto, para hacer este método realmente aplicable, es necesario tener alguna información sobre el error de modelado.

El método Kappa-tau (KT)

Este método calcula los parámetros de un controlador PID ISA de dos grados de libertad (donde ISA es el acrónimo de *Instrument Society of America*), salvo N y en el caso de la derivación de la salida ($c = 0$) dado por la siguiente ley de control

$$U = K(bY_{sp} - Y + \frac{1}{sT_i}(Y_{sp} - Y) + \frac{sT_d}{1 + st_d/N}(cY_{sp} - y)).$$

Este método requiere identificar un modelo de primer orden con retraso, si el proceso no involucra un integrador, en caso de que sí lo involucre, entonces se debe identificar un modelo de primer orden con retraso más un factor $1/s$.

La información usada es entonces dada por los parámetros del modelo μ , T y L , dependiendo de la presencia o ausencia del término $1/s$ y por el uso de un PI o un PID. Una condición adicional es necesaria sobre el margen de magnitud M_s , definido como

$$M_s := \max_{\omega} \left| \frac{1}{1 + L(j\omega)} \right|,$$

para el cual dos valores son recomendados, 1.4 para un sintonizado conservador y de 2.0 para un sintonizado más agresivo. Dado lo anterior y definiendo la ganancia normalizada

Integral	No	No	No	No	Si	Si
Control	PI	PI	PID	PID	PI	PI
M_s	1.4	2.0	1.4	2.0	1.4	2.0
A_0	0.29	0.78	3.8	8.4	0.41	0.81
A_1	-2.7	-4.1	-8.4	-9.6	-0.23	-1.1
A_2	3.7	5.7	7.3	9.8	0.019	0.76
B_0	8.9	8.9	5.2	3.2	5.7	3.4
B_1	-6.6	-6.6	-2.5	-1.5	1.7	0.28
B_2	3.0	3.0	-1.4	-0.93	-0.69	-0.0089
C_0			0.89	0.86		
C_1			-0.37	-1.9		
C_2			-4.1	-0.44		
D_0	0.81	0.48	0.4	0.22	0.33	0.78
D_1	0.73	0.78	0.18	0.65	2.5	-1.9
D_2	1.9	-0.45	2.8	0.051	-1.9	1.2

Cuadro 3.2: Sintonización mediante Kappa-tau(KT)

α del proceso y el retraso normalizado τ como

$$\alpha = \mu \frac{L}{T}, \quad \tau = \frac{L}{L + T}$$

los parámetros del controlador PI(D) son calculados como sigue:

$$K = \frac{A_0}{\alpha} e^{(A_1\tau + A_2\tau^2)}, \quad T_i = LB_0 e^{(B_1\tau + B_2\tau^2)},$$

$$T_d = LC_0 e^{(C_1\tau + C_2\tau^2)}, \quad b = D_0 e^{(D_1\tau + D_2\tau^2)}$$

donde los valores de los coeficientes A_i , B_i , C_i y D_i están dados en la Tabla 3.2, tomada de [3].

Estos coeficientes fueron obtenidos usando polos dominantes en el diseño de muchos y variados procesos, e interpolando luego los resultados, para obtener relaciones de sintonizado compactas. Éste es un método basado en modelo con la peculiaridad de usar interpolación. Una observación importante es que el retraso normalizado, llamado a veces “índice de controlabilidad”, puede ser tomado como una medida de qué tan difícil de con-

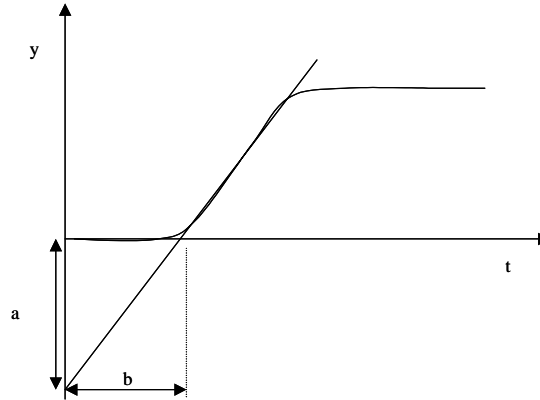


Figura 3-1: Método de Ziegler-Nichols

trolar es el proceso. Este método es muy buena herramienta, simple de usar y adecuada para muchos casos.

El método Ziegler-Nichols

Las reglas de Ziegler-Nichols [81], son el primer ejemplo de un método para auto-sintonizado de PID's. La idea es imponer una tasa de decaimiento de 0.25 a la respuesta a escalón, suponiendo que el proceso puede ser modelado por un integrador más un retraso al menos en la banda de interés.

Ziegler y Nichols proponen algunas formulas para calcular los parámetros del PID, sobre la base de valores característicos de la respuesta a escalón del proceso en malla abierta o la oscilación constante inducida por un regulador proporcional de ganancia adecuada.

El método consiste en determinar las cantidades a y b basadas en la respuesta en malla abierta de la Figura 3-1.

Entonces los parámetros pueden ser calculados de la Tabla 3.3.

	K	T_i	T_d
P	$0.5K_u$		
PI	$0.4K_u$	$0.8T_u$	
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$

Cuadro 3.3: Sintonización por Ziegler-Nichols

Métodos basados en relevador

Una amplia gama de nuevos métodos de sintonización de PID's usan información dada por experimentos de un relevador, es decir uno o más puntos de la curva de Nyquist del proceso. Todos estos métodos están basados en características, y en la gran mayoría de los casos también siguen características. Un ejemplo histórico puede ser encontrado en [1], otro está en [44]. En efecto, la manera más natural de usar un punto de la curva de Nyquist del proceso caracterizada por una frecuencia ω_1 , una magnitud P_1 y una fase φ_1 , es imponer que:

$$R(j\omega_1)P_1e^{j\varphi_1} = e^{j(\varphi_m - \pi)}$$

Esta ecuación de variable compleja lleva dos parámetros reales del regulador y significa que la curva de Nyquist de malla cerrada $L(j\omega)$ contiene el punto $e^{-j(\varphi_m - \pi)}$, es decir el lazo tendrá frecuencia de corte ω_1 y margen de fase φ_m . Para sintonizar un PID de un grado de libertad, se requiere una tercera ecuación, la cual es típicamente obtenida por imponer la tasa T_i/T_d , es decir tomando $T_i = \alpha T_d$ donde α viene a ser un parámetro de diseño usado para limitar la ganancia del regulador de altas frecuencias.

El principal interés en usar métodos basados en relevador es que la frecuencia de corte emerge como un resultado del experimento del relevador, es en esta frecuencia en la cual surgen las oscilaciones. Como tal, la cantidad adoptada de histéresis del relevador viene a ser relevante para determinar la frecuencia de corte del relay, el problema es que la relación entre estas dos cantidades está lejos de ser trivial. Insertar un retraso en serie con el relay puede ayudar en este aspecto, porque modificando retraso durante el experimento podemos hacer que la oscilación se presente en una frecuencia prescrita.

Esto significa que los parámetros de diseño son la frecuencia de corte y el margen

de fase, lo cual es una selección ampliamente aceptada. El problema está en que pueden requerirse varias modificaciones al retraso, resultando en un largo tiempo de sintonizado de la fase.

Una ventaja muy importante de estos métodos basados en relay, es que las características de la curva de Nyquist de malla abierta pueden ser impuestas exactamente, de tal manera que garanticen que la curva contiene el punto $e^{-j(\varphi_m - \pi)}$. Sin embargo el problema es que es necesario el desarrollo total de esta curva, pero teniendo una manera para imponer al menos su desarrollo local alrededor de la frecuencia de corte con muy buena precisión. Éste es un buen método.

3.3. Procesos controlados con un controlador nuevo PI-PD Smith predictor usando autosintonizado, propuesto por Kaya.

Este método se aplica a procesos de alto orden o con grandes retardos de tiempo, los cuales son modelados como funciones de transferencia de plantas de bajo orden, con retardos de tiempo grande. Con el uso del controlador PI-PD propuesto por Kaya [53], también se obtienen buenos resultados en plantas con integrador más retardo de tiempo y plantas inestables más retardo de tiempo. Los antecedentes de este método se basan en trabajos previos de controladores PI-PD, los cuales se pueden encontrar por ejemplo en [54], [55] y [6]. Los aspectos más relevantes del presente método se pueden resumir en dos puntos. Primero, el método está basado en la propuesta de tres parámetros físicos, la constante de amortiguamiento relativo ζ , la frecuencia natural ω_0 y la constante de tiempo de lazo cerrado τ . Los dos primeros se aplican a plantas con funciones de transferencia de segundo orden más un retardo de tiempo, una planta de primer orden con integrador más un retardo de tiempo. El tercer parámetro se aplica al caso de una planta inestable de primer orden más un retardo de tiempo. Los valores de la constante de tiempo τ y la

frecuencia natural ω_0 se obtienen del valor del tiempo de establecimiento deseado. Segundo, en el caso de plantas de primer orden inestables con retardo de tiempo, se contempla solamente un controlador proporcional dentro de un lazo interno. La configuración del controlador nuevo PI-PD Smith predictor se muestra en la Figura 3-2, donde G_{c1} es un controlador PI, G_{c2} es un controlador PD y G_d es un controlador PD (se convierte en un controlador proporcional para el modelo de primer orden inestable con retardo), para rechazar la perturbación d . G_{c2} se usa para estabilizar un proceso inestable o un proceso con integrador y también para modificar la ubicación de los polos de un proceso estable. Los otros dos controladores, G_{c1} y G_d se usan para obtener un buen desempeño en los problemas de control de seguimiento y regulación respectivamente. Para el caso cuando $G_{c2} = G_d$ se tiene el controlador Smith predictor estándar. Las señales $r(s)$ y $c(s)$, corresponden a la entrada y salida del sistema de control respectivamente. En cuanto a los otros 4 bloques del sistema, $G(s)$ es el modelo del proceso a controlar sin retardo, e^{-Ls} es el retardo del proceso. Los bloques $G_m(s)$ y e^{-Lms} son los parámetros del modelo. Si se asume que son iguales los parámetros del proceso y del modelo, esto es, si $G(s) = G_m(s)$ y $L = L_m$ entonces la respuesta del sistema de control a la entrada de referencia r y la señal de disturbio d se puede formular como sigue:

$$C(s) = T_r(s)R(s) + T_d(s)D(s),$$

donde:

$$T_r(s) = \frac{G_m G_{c1} e^{-Lms}}{1 + G_m (G_{c1} + G_{c2})} \quad (3.1)$$

y:

$$T_d(s) = \frac{G_m [1 + G_m (G_{c2} + G_{c1} - G_{c1} e^{-Lms})] e^{-Lms}}{[1 + G_m (G_{c1} + G_{c2})] [1 + G_d G_m e^{-Lms}]}. \quad (3.2)$$

De la Figura 3-2, $G(s)$ representa al proceso real a controlar y e^{-Ls} el retardo correspondiente al proceso. $G_m(s)$ y e^{-Lms} se obtienen mediante un proceso de identificación

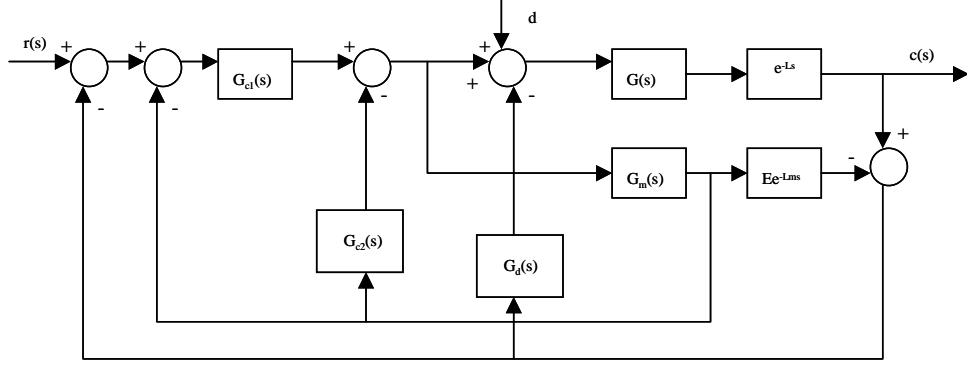


Figura 3-2: Esquema de control propuesto por Kaya

(en el idioma inglés “relay feedback test”), los detalles para obtener los parámetros se pueden ver en [56] y [57]. Las funciones de transferencia utilizadas como modelo en el esquema de control son las siguientes:

$$G_1(s) = \frac{K_m e^{-L_m s}}{(T_{1m}s + 1)(T_{2m}s + 1)},$$

$$G_2(s) = \frac{K_m e^{-L_m s}}{s(T_m s + 1)},$$

$$G_3(s) = \frac{K_m e^{-L_m s}}{T_m s - 1}.$$

A continuación se presenta el procedimiento para determinar los controladores del esquema de control propuesto por Kaya.

3.3.1. Procedimiento para determinar los controladores del esquema de control propuesto por Kaya

Caso1: Procesos que pueden ser modelados por una planta de segundo orden más retardo

En este caso la parte del modelo de la planta sin retardo es:

$$G_m(s) = \frac{K_m}{(T_{1m}s + 1)(T_{2m}s + 1)}$$

y los controladores G_{c1} y G_{c2} tiene la siguiente forma:

$$G_{c1} = K_p(1 + \frac{1}{T_i s}), \quad (3.3)$$

$$G_{c2} = K_f(1 + T_f s). \quad (3.4)$$

En este caso no es necesario el controlador G_d por tratarse de una planta estable y sin integrador. A continuación se presentan solamente las expresiones para determinar los parámetros de los controladores G_{c1} y G_{c2} . Se hace $T_i = T_m$ y $T_f = T_m$. Se sustituyen éstos junto con $G_m(s)$, G_{c1} y G_{c2} en (3.1). La parte libre de retardo de la función de transferencia de lazo cerrado (3.1) se convierte en:

$$T_r(s) = \frac{K_m K_p}{T_{1m} T_{2m} s^2 + T_{1m}(1 + K_f K_m)s + K_m K_p} = \quad (3.5)$$

$$= \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (3.6)$$

donde ω_0 y ζ son la frecuencia natural del sistema y la relación de amortiguamiento, respectivamente (especificados en el diseño). Los parámetros K_p y K_f se encuentran comparando el lado izquierdo y derecho de la Ecuación (3.5). Los valores de K_p y K_f que se obtienen son:

$$K_p = \frac{T_{1m} T_{2m} \omega_0^2}{K_m},$$

$$K_f = \frac{2T_{2m}\zeta\omega_0 - 1}{K_m}$$

y:

$$T_i = T_{1m},$$

$$T_f = T_{1m}.$$

Como ya se mencionó anteriormente; el valor de ω_0 se obtiene de los valores del tiempo de establecimiento T_s y la constante de amortiguamiento relativo ζ , propuestos por el usuario. Por otro lado T_s es directamente proporcional a la constante de tiempo τ de un sistema de segundo orden amortiguado, como se presenta a continuación:

$$T_s = k\tau = \frac{k}{\zeta\omega_0}$$

En [53] se da un valor adecuado de k de entre 2.5 y 3.0. Usando $k = 2.5$ se puede determinar ω_0 como sigue:

$$\omega_0 = \frac{2.5}{T_s\zeta} \quad (3.7)$$

Con las expresiones anteriores se puede determinar el valor de los controladores G_{c1} y G_{c2} usados en el esquema de control correspondiente al primer caso, a partir de ζ y ω_0 .

Caso 2: Cuando el proceso puede ser modelado por medio de una planta de primer orden con integrador más retardo de tiempo.

En este caso la parte del modelo de la planta sin retardo es:

$$G_m(s) = \frac{K_m}{s(T_ms + 1)}$$

Los controladores G_{c1} y G_{c2} tienen la misma estructura del caso 1, dados por (3.3) y (3.4). También se hace $T_i = T_m$ y $T_f = T_m$. Se sustituyen éstos junto con $G_m(s)$, G_{c1} y G_{c2} en (3.1). La parte libre de retardo de la función de transferencia de lazo cerrado (3.1) se convierte en:

$$T_r(s) = \frac{1}{(T_m/K_m K_p)s^2 + K_m(T_m K_f/K_m K_p)s + 1} = \quad (3.8)$$

$$= \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad (3.9)$$

entonces, se pueden obtener los parámetros del controlador comparando el lado izquierdo y el derecho de la Ecuación (3.8). Los parámetros K_p y K_f son:

$$K_p = \frac{T_m \omega_0^2}{K_m}$$

y:

$$K_f = \frac{2\zeta\omega_0}{K_m}.$$

Nuevamente ω_0 se obtiene de (3.7). En este caso se requiere del controlador $G_d(s)$, que está dado por:

$$G_d(s) = K_d(1 + T_d s). \quad (3.10)$$

El controlador se requiere para tener un buen rechazo de la señal de perturbación. Se diseña, tomando en cuenta la estabilización de la segunda parte de la ecuación característica de (3.2):

$$1 + G_d G_m e^{-Lms}.$$

También se da por hecho la relación $T_d = \alpha L_m$, donde α es una constante de proporcionalidad. En base a lo anterior, la expresión para K_d es:

$$K_d = \frac{\pi/2 - \phi_{pm}}{K_d L_m \sqrt{(1 - \alpha)^2 \alpha^2 + (\pi/2 - \phi_{pm})^2 \alpha^2}} \quad (3.11)$$

El autor del método propone el valor del margen de fase ϕ_{pm} , y aclara que ϕ_{pm} no es el margen de fase de la función de transferencia del sistema en lazo abierto. También menciona los valores adecuados de $\phi_{pm} = 64^\circ$ y $\alpha = 0.4$.

Caso 3: Procesos que pueden ser modelados por una función de transferencia de primer orden inestable más retardo de tiempo.

En este caso la parte del modelo de la planta sin retardo es:

$$G_m(s) = \frac{K_m}{(T_m s - 1)}$$

Los controladores G_{c1} y G_{c2} tienen la misma estructura de los casos 1 y 2, dados por (3.3) y (3.4). Ahora se hace $T_i = T_m + 2T_f$ y $K_m T_f = 2$. Se sustituyen éstos junto con $G_m(s)$, G_{c1} y G_{c2} en (3.1). La parte libre de retardo de la función de transferencia de lazo cerrado (3.1) se convierte en:

$$T_r(s) = \frac{1}{(T_i/K_m K_p)s + 1} = \frac{1}{\tau s + 1},$$

donde τ es la constante de tiempo en lazo cerrado. Si se hace $K_m K_p = 1$, entonces $\tau = T_i$. Tomando en cuenta que se hizo $T_i = T_m + 2T_f$ anteriormente, $T_f = (\tau - T_m)/2$. Así los parámetros del controlador están dados por:

$$K_p = \frac{1}{K_m};$$

$$K_f = \tau;$$

$$K_f = \frac{2}{K_m};$$

$$T_f = \frac{\tau - T_m}{2}.$$

El valor de la constante de tiempo en lazo cerrado τ es proporcionado por el usuario

a través del tiempo de establecimiento T_s y tomando nuevamente $k=2.5$, se tiene:

$$\tau = \frac{T_s}{2.5}$$

En cuanto al controlador G_d , el autor recomienda usar un controlador proporcional, es decir, $G_d = K_d$. Esta ganancia K_d se obtiene considerando la estabilización de la segunda parte de la ecuación característica de (3.2):

$$1 + K_d G_m e^{-Lms} = 0,$$

como se menciona en [53]. Usando un criterio óptimo de margen de fase, la expresión de K_d es:

$$K_d = \frac{1}{K_m} \sqrt{\frac{T_m}{L_m}},$$

aplicando la restricción $L_m/T_m < 1$.

Las expresiones presentadas anteriormente se aplican a tres procesos propuestos en [53], utilizados en el Capítulo 6, con el objeto de comparar los resultados de este método con los de la tesis.

3.4. Método propuesto por Skogestad para sintonizar controladores PID (2001)

El método propuesto por Skogestad [71] se aplica a procesos integrales, con retardo de tiempo y con retardo de tiempo puro, obteniéndose un buen desempeño para la respuesta de referencia (setpoint), y rechazo de perturbación. El método se puede resumir en dos puntos:

- A partir del modelo del proceso a controlar se obtiene un modelo de primero o segundo orden más un retardo de tiempo. El autor propone una técnica llamada

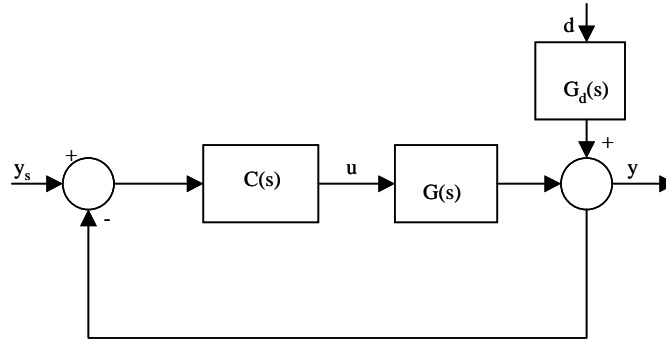


Figura 3-3: Esquema de control usado por Skogestad

“half-rule” para obtener estos modelos.

- Las reglas para sintonizar los controladores PID o PI se derivan del modelo de primero o segundo orden obtenido en el paso anterior. Un controlador PID se obtiene a partir de un modelo de segundo orden y un PI a partir de uno de primero. Las reglas para sintonizar los controladores retoman las bases de los métodos llamados en el idioma inglés “direct synthesis for setpoints” [70] o equivalentemente “Internal Model Control approach for setpoints” [64].

A continuación se presenta el esquema de control que utiliza Skogestad para probar los controladores sintonizados con su método, y el cual, como él mismo lo dice, es uno de los más sencillos del mundo, pero a la vez de los más robustos y que mejor desempeño tienen.

Del esquema de control anterior y_s , u , y y d son las señales de referencia, de control, de salida del sistema y de disturbio, respectivamente. El bloque $C(s)$ es el controlador, $G(s)$ es el proceso a controlar y $G_d(s) = G(s)$ funciona dentro del esquema de control, como si la señal de disturbio actuara directamente sobre la señal de control u . A continuación se presenta la estructura del controlador utilizado por Skogestad.

PDI Serie:

$$C(s) = Kc\left(\frac{T_I s + 1}{T_I s}\right)(T_D s + 1) = \frac{Kc}{T_I s}(T_I T_D s^2 + (T_I + T_D)s + 1) \quad (3.12)$$

El modelo propuesto en el primer paso es:

$$g(s) = \frac{k}{(T_1 s + 1)(T_2 s + 1)} e^{-\theta s} \quad (3.13)$$

donde:

- El parámetro k es la ganancia de la planta
- T_1 es la constante de tiempo dominante
- θ es el tiempo de retardo total
- T_2 es la segunda constante de tiempo. Es opcional y se recomienda cuando $T_2 > \theta$

Aproximación del tiempo total usando la regla “half-rule”

Esta regla aproxima el valor total de retardo de tiempo del proceso a controlar, considerando el retardo del propio proceso, más el efecto de retardo que aporta una constante de tiempo negativa en el numerador (constante de tiempo de respuesta inversa), así como del efecto de las constantes de tiempo más pequeñas del proceso. Para hacer esta aproximación el autor toma los dos primeras aproximaciones del término de primer orden de la serie de Taylor de $e^{-\theta s}$ como sigue:

$$e^{-\theta s} \approx 1 - \theta s \quad \text{y} \quad e^{-\theta s} = \frac{1}{e^{\theta s}} \approx \frac{1}{1 + \theta s}$$

Derivación de las reglas para sintonizar el controlador PID

Como ya se mencionó anteriormente: las reglas para obtener los parámetros de los controladores PID y PI se obtienen usando el método “direct synthesis for setpoints” o equivalentemente “Internal Model Control approach for setpoints”. Para el sistema de la Figura 3-3, la respuesta del sistema a una señal de referencia es:

$$\frac{y(s)}{y_s(s)} = \frac{G(s)C(s)}{1 + G(s)C(s)}, \quad (3.14)$$

donde se considera que la medición de la salida y es exacta. La idea central del método “direct synthesis for setpoints” es proponer una respuesta de lazo cerrado y determinar el controlador correspondiente. De (3.14) se tiene:

$$C(s) = \frac{1}{G(s)} \frac{1}{\frac{1}{(y/y_s)_{deseado} - 1}}. \quad (3.15)$$

El autor considera un modelo $G(s)$ de segundo orden con retardo:

$$(y/y_s)_{deseado} = \frac{1}{T_c s + 1} e^{-\theta s}. \quad (3.16)$$

Se tiene que mantener el retardo en la respuesta deseada ya que éste es inevitable. Sustituyendo (3.16) y (3.13) en (3.15) se obtiene un controlador “Smith Predictor”:

$$C(s) = \frac{(T_1 s + 1)(T_2 s + 1)}{k} \frac{1}{T_c s + 1 - e^{-\theta s}} \quad (3.17)$$

T_c es la constante de tiempo de lazo cerrado deseada y es el único parámetro del controlador que se sintoniza. A continuación se introduce en (3.17) una aproximación del retardo $e^{-\theta s}$ por medio del término de primer orden de la serie de Taylor, esto es $e^{-\theta s} \approx 1 - \theta s$. Ahora la expresión para el controlador es:

$$C(s) = \frac{(T_1 s + 1)(T_2 s + 1)}{k} \frac{1}{(T_c + \theta)s}, \quad (3.18)$$

que tiene la forma de un controlador PID serie. Por comparación de términos entre las Ecuaciones 3.18 y 3.12, se tiene:

$$K_c = \frac{1}{k} \frac{T_i}{(T_c + \theta)}, \quad (3.19)$$

$$T_I = T_1,$$

y:

$$T_D = T_2.$$

Lo anterior quiere decir que el controlador PID requerido para una respuesta de salida correspondiente a un sistema de primer orden con retardo, cuya constante de tiempo es T_c contiene las constantes T_1 , T_2 y la ganancia k del modelo de segundo orden, obtenido en el primer paso de este método. Se puede apreciar que al hacer $T_I = T_1$ se cancela el efecto de la constante de tiempo de primer orden del sistema a controlar. Esto da una buena respuesta de referencia (setpoint).

Modificación del tiempo integral para mejorar el rechazo a la perturbación

La técnica que se utilizó para obtener una respuesta de primer orden con retardo de fase se hizo en base a la respuesta de setpoint y por lo tanto con esto se obtiene un buen desempeño de seguimiento. Sin embargo se presenta un tiempo de establecimiento muy grande para la perturbación d . Lo que el autor del método propone es reducir el valor del tiempo integral. Sin embargo no debe ser tan pequeño, porque esto ocasionaría un desempeño pobre (que incluye pequeñas oscilaciones), en cuanto al seguimiento de la señal de referencia. De lo anterior es conveniente encontrar una relación aceptable entre la constante de tiempo dominante del proceso y el retardo de tiempo, de tal manera que el controlador consiga una buena respuesta del sistema tanto para la señal de referencia como para el rechazo de la perturbación. A continuación se hace un análisis para encontrar un valor aceptable de T_i .

Como ya se mencionó anteriormente, en [71] se analiza el efecto del valor del tiempo integral en el desempeño del sistema de control que utiliza, así como su robustez. El análisis que se hace parte de un ejemplo de un sistema de primer orden con retardo $e^{-\theta s}$, donde $\theta = 1$ y una constante de tiempo $T_1 = 30$. En este tipo de casos, donde $T_1 \gg \theta$, es

donde se presenta principalmente un rechazo pobre de la perturbación de entrada cuando se iguala T_I con T_1 . Durante el desarrollo del ejemplo muestra algunos casos al variar el valor de T_i , desde $T_I = T_1$, hasta $T_I = 2$. Cuando se hace $T_I = T_1$ se obtiene una buena respuesta de referencia pero un tiempo de establecimiento grande a la perturbación d . Con $T_I = 8\theta$ se obtiene un buen resultado, tanto para el problema seguimiento como en el tiempo de establecimiento de la señal de perturbación. El peor caso se presenta cuando se hace $T_i = 2$: aparecen pequeñas oscilaciones en la señal de salida y . El autor del artículo menciona que las oscilaciones son prácticamente independientes del retardo y por lo tanto replantea el modelo del ejemplo, considerándolo sin retardo para encontrar un valor adecuado de T_I , como se describe a continuación:

$$g(s) = k \frac{e^{-\theta s}}{T_I s + 1} = k \frac{1}{T_I s + 1} \approx \frac{k}{T_I s} = \frac{k'}{s},$$

donde se aplica la segunda aproximación, debido a que la frecuencia resultante de las oscilaciones ω_0 es tal que $(T_1 \omega_0)^2$ es mucho mayor que 1. Con un controlador $PI = K_c(1 + \frac{1}{T_I s})$ el polinomio característico $1 + gc$ es:

$$\frac{T_I}{k'K_c} s^2 + T_I s + 1,$$

que tiene forma estándar de segundo orden, $T_0^2 s^2 + 2T_0 \zeta s + 1$, con:

$$T_0 = \sqrt{\frac{T_I}{k'K_c}} \quad (3.20)$$

y

$$\zeta = \frac{1}{2} \sqrt{k'K_c T_I}.$$

Las oscilaciones ocurren para $\zeta < 1$. Aunque se pueden tolerar algunas oscilaciones, para un funcionamiento robusto (ver [71]) se hace $\zeta = 1$. O también:

$$K_c T_I = \frac{4}{k'}. \quad (3.21)$$

Insertando el valor de K_c de (3.19), entonces se obtiene el T_I modificado, para los casos cuando $T_I = T_1$ es muy grande:

$$T_I = 4(T_c + \theta) \quad (3.22)$$

Resumiendo el método propuesto por Skogestad [71]: para encontrar el valor de los parámetros del controlador PID se utilizan las siguientes expresiones:

$$K_c = \frac{1}{k} \frac{T_i}{(T_c + \theta)} = \frac{1}{k'} \frac{1}{(T_c + \theta)}, \quad (3.23)$$

$$T_I = \min\{T_1, 4(T_c + \theta)\}, \quad (3.24)$$

$$T_D = T_2, \quad (3.25)$$

donde T_c es el único parámetro que se requiere sintonizar para tener una respuesta deseada de primer orden en lazo cerrado. Las reglas son las mismas para sintonizar un controlador PID o un PI (con la diferencia de que $T_2 = 0$).

Finalmente, en [71] se propone un valor de $T_c = \theta$ para tener una respuesta rápida y un buen rechazo a la perturbación. Respecto a estabilidad y robustez, éstas son favorecidas por valores grandes de T_c .

3.5. Controlador PID de dos grados de libertad

En general un sistema de control tiene diferentes requerimientos de diseño [3]. Debe tener buena respuesta transitoria ante los cambios de la señal de referencia, atenuar perturbaciones y ser inmune a problemas de ruido en las variables medidas. Cuando un

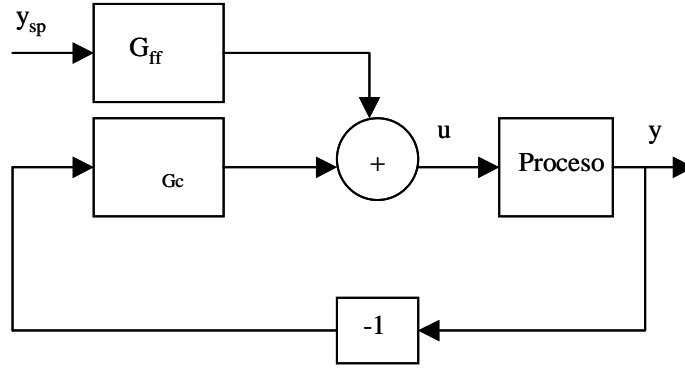


Figura 3-4: Estructura de dos grados de libertad

sistema de control en lazo cerrado funciona con sólo una señal error ($e = r - y$), se conoce como un sistema con un grado de libertad. Una estructura con dos grados de libertad presenta mayor flexibilidad para satisfacer los requerimientos de diseño, concretamente ante los cambios de la señal de referencia [4]. La estructura con dos grados de libertad se muestra en la Figura 3-4.

En este esquema G_{ff} es la parte del controlador que es afectada únicamente por la señal de referencia y G_c es la función de transferencia del controlador cuya entrada es la salida del proceso (y). A continuación se presentan las expresiones para G_{ff} y G_c :

$$G_{ff} = K(b + \frac{1}{sT_i} + csT_d),$$

$$G_c = K(1 + \frac{1}{sT_i} + sT_d).$$

En el dominio del tiempo, la señal de control u está dada por:

$$u(t) = (e_p + \frac{1}{T_i} \int_0^t e(s)ds + Td\frac{de_d}{dt}),$$

donde el error para la parte proporcional es:

$$e_p = by_{sp} - y,$$

para la parte derivativa:

$$ed = cy_{sp} - y$$

y la parte integral:

$$e = y_{sp} - y.$$

En [3] se afirma que los parámetros b y c sólo tienen influencia en lo que concierne a los cambios en la señal de referencia y no en lo que se refiere a la atenuación de perturbación o a los efectos del ruido sobre las variables medidas. Lo anterior quiere decir que esta estructura permite desacoplar los efectos de cambio en la señal de referencia con el de perturbación.

3.5.1. Controladores PID comerciales

Para finalizar este capítulo se muestran los esquemas de control PID comerciales más comunes:

Forma estándar o ISA:

$$U = K(by_{sp} - Y + \frac{1}{sT_i}(Y_{sp} - Y) + \frac{sT_d}{1 + st_d/N}(cY_{sp} - y))$$

forma serie:

$$U = K'((b + \frac{1}{sT'_i})\frac{1 + scT'_d}{1 + sT'_d/N}Y_{sp} - (1 + \frac{1}{sT'_i})\frac{1 + scT'_d}{1 + sT'_d/N}Y)$$

y la forma paralela:

Controlador	Estructura	b	c	N	P. muestreo
Allen Bradley PLC 5	I, II	1	1	no usa	func. carga
Bailey Net 90	II, III	0.0 ó 1.0	0.0 ó 1.0	10	0.25
Fisher Controls Provox	II	1.0	0.0	8	0.1, 0.25 ó 1
Fisher Controls DPR, 900, 910	II	0.0	0.0	8	0.2
Fisher Porter Micro DCI	II	1.0	0.0 ó 1.0	no usa	0.1
Foxboro model 761	II	1.0	0.0	10	0.25
Honeywell TDC	II	1.0	1.0	8	0.33, 0.5 ó 1
Moore Products Type 352	II	1.0	0.0	1-30	0.1
Alfa Laval Automation ECA40	II	0.0	0.0	8	0.2
Taylor Mod. 30	II	0.0 ó 1.0	0.0	17 ó 20	0.25
Toshiba TOSDIC	II	1.0	1.0	3.3-10	0.2
Turnbull TCS 6000	II	1.0	1.0	no usa	0.036-1.56
Yokogawa SLPC	II	0.0 ó 1.0	0.0 ó 1.0	10	0.1

Cuadro 3.4: Controladores PID comerciales

$$U = K''(bY_{sp} - Y) + \frac{K_i''}{s}E + \frac{K_d''s}{1 + sK_d''/NK''}(cY_{sp} - Y),$$

donde el parámetro N limita la ganancia de alta frecuencia del término derivativo, para evitar la amplificación del ruido. A continuación se presenta la Tabla 3.4, donde se muestran algunos controladores comerciales y los parámetros que utilizan [3].

En la Tabla 3.4, la columna denominada Estructura, se refiere a la estructura del controlador, donde I corresponde a la forma estándar o ISA, II a la forma serie y III a la forma paralela. Los parámetros b y c son utilizados como una ponderación de la referencia de entrada (del inglés setpoint); finalmente, N es el parámetro que limita la ganancia de alta frecuencia del término derivativo.

Capítulo 4

Aplicación de un Algoritmo Genético simple

4.1. Introducción

El problema que se aborda en este capítulo es un problema de control de posición de un servomecanismo de Corriente Directa (C.D.), formulado como un problema de optimización global. Se utiliza la técnica de control mixta H_2/H_∞ para sintonizar un controlador PID. La ley de control proporcional, integral y derivativa (control PID), es una herramienta muy poderosa que permite dar solución a la mayoría de las demandas que se presentan en la industria. Como se menciona en [13], los diseños de control basados en el método H_2/H_∞ se utilizan frecuentemente cuando se desea un funcionamiento robusto en sistemas sometidos a perturbaciones en los parámetros y/o presencia de señales de disturbio no conocidas. Sin embargo también se menciona que la aplicación de esta estrategia resulta difícil de llevar a la práctica. Por esta razón es común fijar la estructura de un controlador con el fin de tratar la estrategia H_2/H_∞ como un problema de optimización numérico en un espacio de vectores que contienen los parámetros del controlador. El vector que se obtiene con los parámetros reales representa una solución al problema de optimización y en particular, para este caso el valor de las ganancias de un

controlador de estructura fija. Se sabe también que en la actualidad existen herramientas computacionales, tales como los algoritmos evolutivos que permiten resolver numéricamente muchos problemas de control óptimo, en particular, los algoritmos genéticos (AG's), originalmente inspirados en los procesos de selección natural y genético evolutiva (ver por ejemplo las ideas originales planteadas por Holland en 1975 [38]). Para el problema planteado en este capítulo, el algoritmo genético determina los parámetros de un controlador PID utilizando la metodología H_2/H_∞ siguiendo un procedimiento de solución numérico. Los resultados que se presentan en la primera parte de este capítulo son congruentes con los resultados presentados en [13]. Sin embargo, los valores de los parámetros resultan grandes para ser aplicados a una planta experimental, debido principalmente a las limitaciones físicas de los componentes que intervienen en el control, así como por problemas de ruido. Lo anterior motivo a replantear el problema, adicionando una restricción más al problema de optimización. A esta restricción se ha denominado restricción de saturación y tiene como objetivo limitar el valor de los parámetros del controlador PID.

Como complemento de la aplicación del algoritmo genético simple al problema de control de posición, al final de este capítulo se presentan los resultados del mismo problema, sólo que tratado como un problema de optimización multiobjetivo con dos funciones objetivo y una restricción. El algoritmo genético utilizado fue el NSGA-II.

4.2. Planteamiento del problema

Considere el sistema de control retroalimentado que se muestra en la Figura 4-1, donde:

- r representa la señal de referencia de entrada;
- y_d es la señal de salida;
- d representa la señal de disturbio;

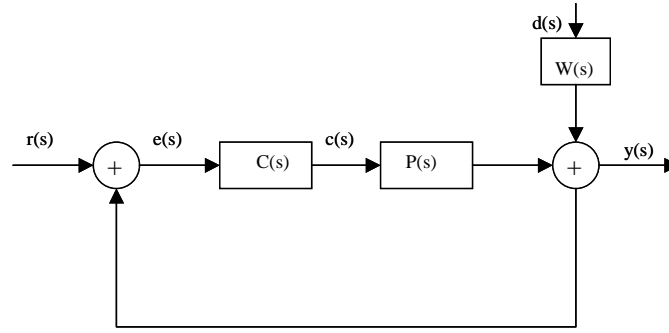


Figura 4-1: Sistema de control

- e representa la señal de entrada de error de seguimiento;
- P denota un sistema lineal invariante en el tiempo (LTI) de una entrada una salida (SISO);
- C denota un controlador LTI SISO;
- W denota una función de ponderación (perfil frecuencial de la perturbación).

El problema de control se define como sigue:

Definición 4.1 *Problema de control de seguimiento óptimo (PCSO): Tomando en cuenta el error de seguimiento mostrado en la Figura 4-1, se desea hallar un controlador C que minimice la señal de error de seguimiento e para una señal de referencia especificada r , considerando la atenuación de la señal de disturbio y la estabilidad interna en lazo cerrado.*

Eligiendo la integral del cuadrado de una señal dada, la definición de PCSO puede ser reformulada en términos formales como sigue:

Definición 4.2 *Problema de control de seguimiento óptimo (PCSO): Se toman una función de transferencia $P(s)$ SISO y una $W(s)$ dadas. Se considera un nivel de atenuación de disturbio $\gamma > 0$. Se desea hallar un controlador C tal que:*

$$\min_C J, \quad \text{con } J := \int_0^\infty e^2(t) dt \quad (4.1)$$

sujeto a la restricción:

$$\sup_{d(t) \in L_2} \frac{\|y_d(t)\|_2}{\|d(t)\|_2} = \left\| \frac{1}{1 + P(s)C(s)} \right\|_\infty \leq \gamma \quad (4.2)$$

mientras se asegura la estabilidad interna en lazo cerrado. L_2 ejemplifica el espacio de todas las funciones reales evaluadas como integrales de Lebesgue. La señal de error se define como:

$$e(s) = \frac{r(s)}{1 + P(s)C(s)} \quad (4.3)$$

En la Definición 4.2, $\|\cdot\|_\infty$ simboliza la norma H_∞ de la función de transferencia (ver por ejemplo [27] y [75]): También es importante concluir que el problema que se considera en la definición anterior es un caso típico de un problema de control mixto H_2/H_∞ . De hecho, el problema corresponde a la minimización de la norma H_2 de una señal (4.1), bajo una restricción en norma H_∞ de (4.2).

4.3. Control PID basado en la combinación H_2/H_∞

Ahora se fija la estructura del controlador como un controlador PID:

$$C(s) = k_1 + \frac{k_2}{s} + k_3 s$$

donde: k_1 , k_2 y k_3 denotan las ganancias proporcional, integral y derivativa del controlador respectivamente. Si se supone que el dominio de los parámetros $\{k_1, k_2, k_3\}$ garantiza la estabilidad del sistema en lazo cerrado (este dominio de los parámetros puede ser caracterizado por el criterio de Routh-Hurwitz), y aplicando el Teorema de Parseval (ver por ejemplo [60] y [40]), tenemos que (4.1) se puede reescribir como:

$$J := \int_0^\infty e^2(t)dt = \min_{k_1, k_2, k_3} \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} e(s)e(-s)ds \quad (4.4)$$

$$= \min_{k_1, k_2, k_3} \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \frac{r(-s)r(s)}{(1 + P(-s)C(-s))(1 + P(s)C(s))} ds \quad (4.5)$$

$$= \min_{k_1, k_2, k_3} \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \frac{B(s)B(-s)}{A(s)A(-s)} ds \quad (4.6)$$

donde $A(s)$ y $B(s)$ son polinomios Hurwitz. Ahora se toma m como el grado de $A(s)$ y se asume que el grado de $B(s)$ es $m - 1$. Así, los polinomios de $A(s)$ y $B(s)$ están dados por:

$$A(s) = \sum_{k=0}^m a_k s^k = a_m \prod_{i=1}^m (s - z_i) \quad (4.7)$$

y

$$B(s) = \sum_{k=0}^{m-1} b_k s^k,$$

donde $\{z_1, z_2, \dots, z_m\}$ es el conjunto de ceros de $A(s)$. También se asume que $a_0 \neq 0$ y $a_m > 0$.

En lo que sigue se puede cambiar J en (4.4) por $J_m(k_1, k_2, k_3)$ para indicar estrictamente su dependencia sobre m , k_1 , k_2 y k_3 .

Como se muestra en [13] el problema de optimización (4.4) se puede resolver en este caso a través del Teorema del Residuo (para mayor detalle, consultar [40], [43] y [51]), el cual permite obtener una forma explícita para J en términos de los parámetros del controlador. En efecto

$$J_m(k_1, k_2, k_3) = \frac{a_0 d_{m-1} Q_{m-1} [d_{m-2} Q_{m-2} - d_{m-3} Q_{m-3} + \dots + (-1)^{m-1} d_1 Q_1] + (-1)^{m-1} a_m d_0 Q_0}{(-1)^{m-1} 2 a_m a_0 [a_{m-1} Q_{m-1} - a_m (a_{m-2} Q_{m-2} - a_{m-5} Q_{m-3} + a_{m-7} Q_{m-4} + \dots)]} \quad (4.8)$$

donde: $d_l = \sum_{i,j=0}^{m-1} (-1)^i b_i b_j$, con $i + j + 2l$. Q_i , $i = 1, 2, \dots, m-2$ se forman de $|\Omega|$, con:

$$\Omega := \begin{bmatrix} a_{m-1} & a_{m-3} & a_{m-5} & \cdots & \cdots & 0 & 0 \\ a_m & a_{m-2} & a_{m-4} & \cdots & \cdots & \vdots & \vdots \\ 0 & a_{m-1} & a_{m-3} & a_{m-5} & \cdots & \vdots & \vdots \\ \vdots & a_m & a_{m-2} & a_{m-4} & \cdots & \vdots & \vdots \\ \vdots & 0 & a_{m-1} & a_{m-3} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & a_1 & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & a_2 & a_0 \end{bmatrix},$$

mediante la cancelación de la primera, $(m-1)$ -ésima y m -ésima columnas y el primero $(i+1)$ -ésimo y m -ésimo renglones, y Q_0 y Q_{m-1} están dadas por:

$$\begin{aligned} Q_0 &= a_2 Q_1 - a_4 Q_2 + a_6 Q_3 - a_8 Q_4 + \dots \\ Q_1 &= a_{m-2} Q_{m-2} - a_{m-4} Q_{m-3} + a_{m-6} Q_{m-4} - \dots \end{aligned}$$

y en lo que respecta a la evaluación de la restricción:

$$\left\| \frac{W(s)}{1 + P(s)C(s)} \right\|_{\infty} \leq \gamma \quad (4.9)$$

es obtenida directamente usando un método numérico (ver por ejemplo la rutina incluida en el toolbox [48] de MATLABTM).

Resumiendo, el problema de control PID basado en H_2/H_{∞} (PID _{H_2/H_{∞}} CP) se define como sigue:

Definición 4.3 (PID _{H_2/H_{∞}} CP): dadas las funciones de transferencia $P(s)$ y $W(s)$. Así

como una constante de atenuación dada γ , hallar un controlador PID ($C(s) = k_1 + \frac{k_2}{s} + k_3s$), asumiendo que el dominio de parámetros de $\{k_1, k_2, k_3\}$ garantiza la estabilidad interna del sistema en lazo cerrado, tal que $J_m(k_1, k_2, k_3)$ sea minimizado y $\left\| \frac{W(s)}{1+P(s)C(s)} \right\|_\infty \leq \gamma$ sea satisfecho. $J_m(k_1, k_2, k_3)$ y $\left\| \frac{W(s)}{1+P(s)C(s)} \right\|_\infty \leq \gamma$ se definen en (4.8) y (4.9) respectivamente.

4.4. Ejemplo ilustrativo.

Con el fin de ilustrar el controlador PID basado en la estrategia H_2/H_∞ , se identificó el modelo de una plataforma experimental de un servosistema de CD. El modelo que se obtuvo de la identificación (por mínimos cuadrados [2]) del proceso es el siguiente:

$$P(s) = \frac{317}{s(s + 5.57)} = \frac{y(s)}{u(s)}, \quad (4.10)$$

donde $u(s)$ denota la señal de control y $y(s)$ denota la señal de salida. La salida (posición angular) es medida en número de revoluciones, mientras que la señal de entrada está dada en Voltios. La señal de referencia es la siguiente:

$$r(s) = \frac{1}{s}$$

y se tomó como función de ponderación la siguiente función de transferencia:

$$W(s) = \frac{1}{s + 1}$$

Restricción de estabilidad interna.

Primero, se tiene que:

$$e(s) = r(s) - y_d(s) = T_e(s)r(s),$$

donde:

$$T_e(s) := \left(\frac{1}{1 + P(s)C(s)} \right) = \frac{s^2(s + 5.57)}{s^3 + (5.57 + 317k_3)s^2 + 317k_1s + 317k_2}$$

consecuentemente, el polinomio característico denotado por p_c está dado por:

$$p_c = s^3 + (5.57 + 317k_3)s^2 + 317k_1s + 317k_2 \quad (4.11)$$

Ahora, el criterio de Routh-Hurwitz permite concluir que el dominio de los parámetros de $\{k_1, k_2, k_3\}$ que asegura la estabilidad interna en lazo cerrado está caracterizado por:

$$k_2 > 0; \quad (4.12)$$

$$k_3 > -5.57/317;$$

$$k_1 > k_2/(5.57 + 317k_3);$$

Problema de optimización H_2

Tomando en cuenta (4.4) y (4.8) se tiene:

$$\begin{aligned} \int_{-j\infty}^{j\infty} e(s)e(-s)ds &= \int_{-j\infty}^{j\infty} \frac{r(-s)r(s)}{(1 + P(-s)C(-s))(1 + P(s)C(s))} ds \\ &= \int_{-j\infty}^{j\infty} \frac{(\frac{1}{-s})(\frac{1}{-s})}{(1 + \frac{317}{-s(-s+5.57)}(k_1 + \frac{k_2}{-s} - k_3s))(1 + \frac{317}{s(s+5.57)}(k_1 + \frac{k_2}{s} + k_3s))} ds \\ &= \int_{-j\infty}^{j\infty} \frac{B(s)B(-s)}{A(s)A(-s)} ds, \end{aligned}$$

donde:

$$B(s) := b_0 + b_1s + b_2s^2 \quad \text{y} \quad A(s) = a_0 + a_1s + a_2s^2 + a_3s^3,$$

con:

$$a_0 = 317k_2, \quad a_1 = 317k_1, \quad a_2 = (5.57 + 317k_3), \quad a_3 = 1, \quad b_0 = 0, \quad b_1 = 5.57, \quad b_2 = 1.$$

se hace mención que para este ejemplo $m = 3$, y por lo tanto:

$$\begin{aligned} J_3(k_1, k_2, k_3) &= \frac{b_2^2 a_0 a_1 + (b_1^2 - 2b_0 b_2) a_0 a_3 + b_0^2 a_2 a_3}{2a_0 a_3 (-a_0 a_3 + a_1 a_2)} \\ &= 0,00001577287066 \left(\frac{31700k_1 + 3102,49}{-k_2 + 5,57k_1 + 317k_1 k_3} \right) \end{aligned} \quad (4.13)$$

Problema de optimización H_∞ .

Retomando la Ecuación 4.9 y sustituyendo valores, se tiene:

$$\begin{aligned} \frac{W(s)}{1 + P(s)C(s)} &= \\ &= \frac{s^2(s + 5.57)}{s^4 + (6.57 + 317k_3)s^3 + (317k_1 + 317k_3 + 5.57)s^2 + (317k_1 + 317k_2)s + 317k_2} \end{aligned} \quad (4.14)$$

El problema de control PID basado en H_2/H_∞ ($\text{PID}_{H_2/H_\infty}$ CP)

Resumiendo, para nuestro ejemplo el problema $\text{PID}_{H_2/H_\infty}$ CP se define como sigue (fijamos $\gamma = 0.1$):

Problema 4.1 Sean $P(s) = \frac{317}{s(s+5.57)}$, $W(s) = \frac{1}{s+1}$ y $r(s) = \frac{1}{s}$ dados. Sea $\gamma = 0.1$. Considerando el dominio de los parámetros para efectos de estabilidad $\{k_1, k_2, k_3\}$, (de 4.12):

$$\begin{aligned}
k_2 &> 0; \\
k_3 &> -5.57/317; \\
k_1 &> k_2/(5.57 + 317k_3);
\end{aligned}$$

se desea hallar un conjunto $\{k_1, k_2, k_3\}$ que cumpla con la condición anterior tal que (ver 4.13 y 4.14):

$$\min_{\{k_1, k_2, k_3\}} J_3\{k_1, k_2, k_3\} = \min_{\{k_1, k_2, k_3\}} \left(0.00001577287066 \left(\frac{31700k_1 + 3102.49}{-k_2 + 5.57k_1 + 317k_1k_3} \right) \right) \quad (4.15)$$

y:

$$\left\| \frac{W(s)}{1 + P(s)C(s)} \right\|_{\infty} =$$

$$\left\| \frac{s^2(s + 5.57)}{s^4 + (6.57 + 317k_3)s^3 + (317k_1 + 317k_3 + 5.57)s^2 + (317k_1 + 317k_2)s + 317k_2} \right\|_{\infty} \leq 0.1 \quad (4.16)$$

sean satisfechas.

Ahora podemos resolver el problema propuesto, entendiendo que $\{k_1^*, k_2^*, k_3^*\}$ se denota como la solución del problema de optimización propuesto.

4.5. Enfoque con algoritmos genéticos

Existe una gran cantidad de publicaciones sobre algoritmos genéticos (AG's) y sus aplicaciones en control automático y procesamiento de señales (ver por ejemplo [46], [47], [60], [34] y [11]). Para nuestro propósito, integramos un algoritmo genético simple

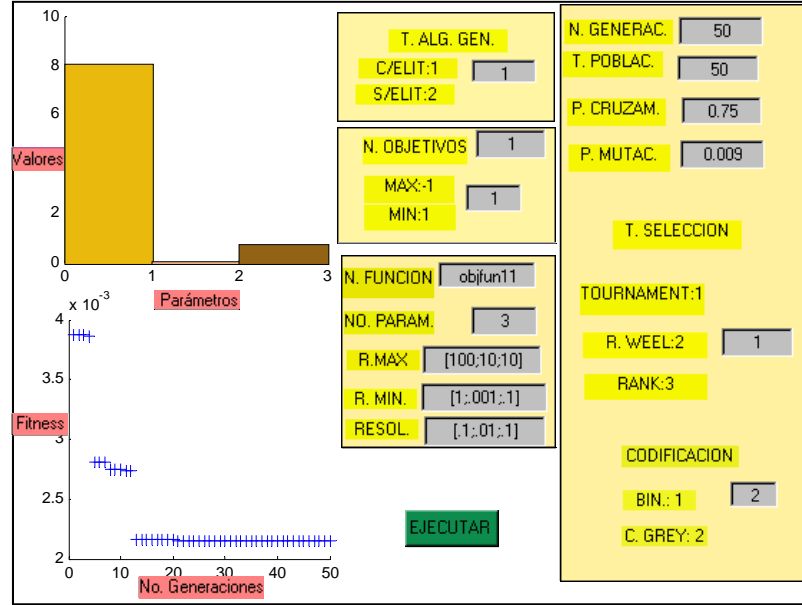


Figura 4-2: Interfaz gráfica

tomando como base algunos bloques de una herramienta basada en MATLABTM– llamada FlexToolsTM [12]. Después de estudiarse detalladamente este toolbox de MATLABTM formamos un algoritmo genético de propósito general con los componentes básicos de éste y con nuevas y más versátiles funciones, como por ejemplo la inclusión de codificación con código Gray y una interfaz gráfica que permite el uso del nuevo algoritmo de manera más amigable. La interfaz gráfica se muestra en la Figura 4-2. En la parte superior izquierda de la interfaz se presenta en forma de barras, el valor de los parámetros a optimizar y en la parte inferior izquierda se muestran los valores de aptitud (fitness) que toma la función objetivo a través de las generaciones preestablecidas. Tanto la función de aptitud (esto es, para nuestro caso $J_m(k_1, k_2, k_3)$), como la restricción de atenuación de disturbio fueron escritas en lenguaje de MATLABTM, cada función recibe como argumentos $\{k_1, k_2, k_3\}$. Es obvio que la población genética inicial pertenezca al dominio de parámetros caracterizada por (4.12). A continuación se describe brevemente la aplicación del algoritmo genético utilizado.

4.5.1. Procedimiento computacional

Con el fin de construir la solución de $(\text{PID}_{H_2/H_\infty} \text{CP})$ a través de la aproximación de un algoritmo genético se presenta el siguiente procedimiento:

1. Fijamos los descriptores del algoritmo genético mencionado arriba (número de generaciones, tamaño de la población, probabilidad de mutación, etc.).
2. El Ag crea aleatoriamente una población de individuos (cromosomas), que contienen las ganancias del controlador PID $\{k_1, k_2, k_3\}$, restringidas de tal forma que pertenezcan al dominio de los parámetros que satisfagan los requisitos de estabilidad interna del sistema.
3. Se calcula la aptitud de cada cadena genética codificada, tomando directamente $J_m(k_1, k_2, k_3)$ como la función de aptitud.
4. Aplicamos selección (se utiliza la opción de torneo binario), para elegir el mejor subconjunto de la población de ganancias del controlador PID.
5. El AG selecciona de manera aleatoria un número de cadenas binarias y les aplica el operador de cruzamiento con el fin de dar variedad a la población.
6. Dependiendo del valor de la probabilidad de mutación, cada una de las cadenas de la población pueden o no ser mutadas.
7. Verificamos la condición de atenuación de disturbio de cada miembro de la población elegida. Se penalizan miembros que no cumplen con esta condición.
8. En el procedimiento se repiten los pasos del 3 al 7 hasta que se alcanza un número preestablecido de generaciones.

Los valores de probabilidad de cruzamiento y de mutación se eligen como se indica en [7].

4.5.2. Resultados experimentales

Continuando con nuestro ejemplo ilustrativo. El conjunto de los descriptores del algoritmo genético se fijan con los siguientes valores:

- Número de generaciones: 20.
- Tamaño de la población: 80.
- Probabilidad de cruzamiento: 0.8.
- Número de puntos de cruce: 2.
- Probabilidad de mutación: 0.070.
- Técnica de selección : Torneo binario.
- Elitismo: Si.

Con el fin de hallar $\{k_1^*, k_2^*, k_3^*\}$, tomamos un subconjunto de parámetros que cumplen con la condición de estabilidad, caracterizado mediante $k^1 \in [0,8]$, $k^2 \in [0,8]$ y $k^3 \in [0,8]$. y después se aplica el procedimiento propuesto en este trabajo, obteniendo el siguiente resultado (se ejecutó el algoritmo más de 30 veces obteniéndose resultados muy parecidos):

$$k_1^* = 8, \quad k_2^* = 0.0073, \quad k_3^* = 8$$

$$\left\| \frac{s^2(s + 5.57)}{s^3 + (5.57 + 317k_3)s^2 + 317k_1s + 317k_2} \right\|_{\infty} \leq 0.0669$$

La solución obtenida computacionalmente muestra algunas particularidades: k_1^* y k_3^* toman los valores extremos de sus respectivos rangos, mientras que k_2^* tiende a cero. Debido a la naturaleza de $J_3\{k_1, k_2, k_3\}$, k_1^* y k_3^* deben ser grandes para garantizar un valor pequeño de $J_3\{k_1, k_2, k_3\}$. En cuanto al valor de k_2^* confirma que la acción integral no es necesaria, ya que el modelo del servosistema de CD contiene acción integral. Es

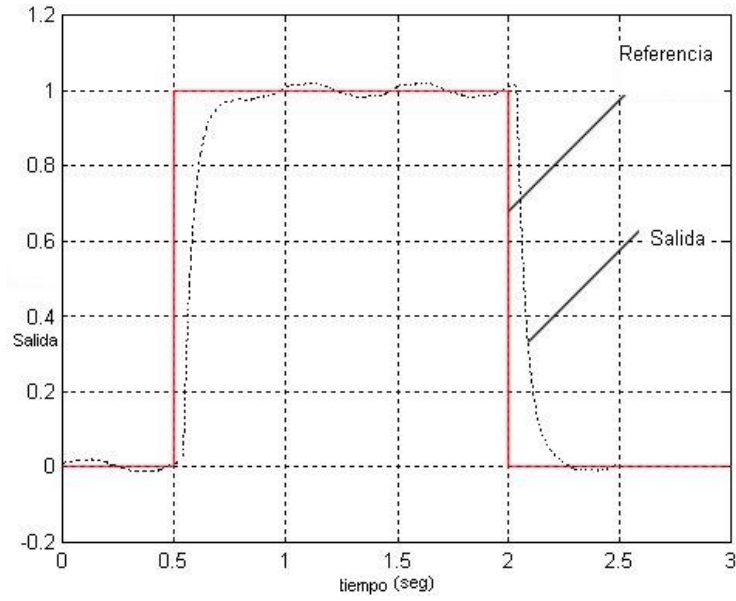


Figura 4-3: Salida del sistema de control

necesario decir que un valor grande de k_1^* incrementa el sobre impulso de la respuesta. Esta situación no es deseable, por lo que es necesario proporcionar amortiguamiento. Esto se consigue aumentando el valor de la ganancia derivativa k_3^* . Sin embargo por problemas de ruido principalmente, es recomendable aplicar valores pequeños de k_3^* . Por estas razones las ganancias del controlador PID fueron ajustadas a los siguientes valores:

$$k_1^* = 8, \quad k_2^* = 0.0073, \quad k_3^* = 0.4.$$

La Figura 4-3 muestra el comportamiento del sistema real correspondiente al PID con las 3 ganancias propuestas. Es importante remarcar que la señal de disturbio es atenuada, aunque sigue presente a la salida.

En la segunda etapa de esta aplicación se reformula el problema de control como un problema de optimización global con dos restricciones, pero manteniendo la misma planta y el mismo procedimiento del ejemplo anterior. La nueva restricción es llamada restricción de saturación, la cual es formulada como sigue:

$$\left\| T_{ur}(s) \frac{1}{s} \right\| = \left\| \frac{W(s)}{1 + P(s)C(s)} \frac{1}{s} \right\|_{\infty} \leq \alpha \quad (4.17)$$

donde $T_{ur}(s)$ es la función de transferencia que relaciona la señal de control $u(t)$ con la señal de referencia $r(t)$, para una entrada de referencia escalón unitario.

La adición de la segunda etapa, restringe el espacio de búsqueda válido del problema de optimización tratado anteriormente. Los valores de las ganancias del controlador que se obtienen en este caso son muy parecidos a los que se obtienen con un procedimiento práctico de prueba y error, realizados por un experto. A continuación se presenta el resultado de aplicar el algoritmo genético simple al problema planteado con dos restricciones:

$$k_1^* = 8, \quad k_2^* = 0.0070, \quad k_3^* = 0.423$$

El resultado anterior se obtuvo después de ejecutar el algoritmo genético más de 30 veces, obteniéndose prácticamente el mismo en todos los casos.

A continuación se presenta el problema de control de posición de C.D. tratado en este capítulo, el cual es propuesto como un problema de optimización multiobjetivo.

4.5.3. Problema resuelto como Problema multiobjetivo

Con el fin de introducir la estrategia de sintonización de controladores PID mediante un algoritmo genético multiobjetivo (NSGA-II), el cual es tratado en el Capítulo 5 de esta tesis, se presentan los resultados del problema de control de posición de C.D. después de aplicar el NSGA-II a dicho problema.

El problema de control se plantea como un problema de optimización multiobjetivo, el cual minimiza dos funciones objetivo sujeto a una restricción. La Función Objetivo 1 (Obj 1) corresponde a la Ecuación 4.15 y la Función Objetivo 2 (Obj 2) se presenta a continuación:

$$Obj2 = \left\| T_{ur} \frac{1}{s} \right\|_{\infty}$$

La restricción está dada por la Ecuación 4.16.

Los Parámetros del algoritmo genético multiobjetivo son:

- Tamaño de la población: 20.
- Número de generaciones: 80.
- Probabilidad de cruzamiento: 0.80.
- Número de puntos de cruce: 2.
- Probabilidad de Mutación: 0.09.
- Técnica de selección : Torneo binario.
- Elitismo: Si

A continuación se muestra la Tabla 4.1, la cual contiene los parámetros de dos controladores, seleccionados del conjunto de óptimos de Pareto del problema de optimización.

PID	k_p	k_I	k_d	Obj 1	Obj 2
PID1	4.95	0.0121	0.2	0.0074	0.2
PID2	7.93	0.0248	0.21	0.0068	0.21

Cuadro 4.1: Parámetros de los controladores PID1 y PID2.

En la Figura 4-4 se muestra la gráfica del frente de Pareto del problema de optimización y en la Figura 4-5 la respuesta de salida del proceso del problema de control, mostrado en la Figura 4-1.

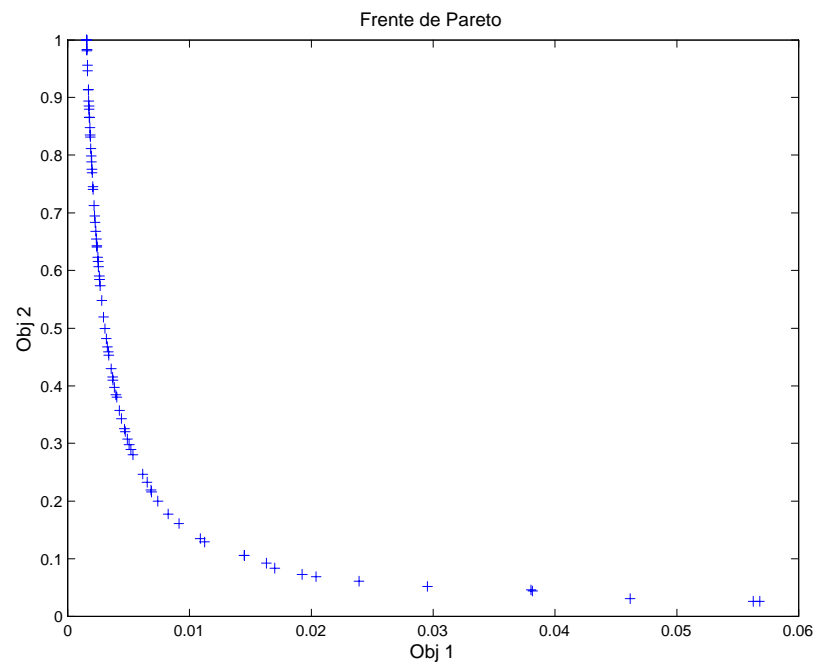


Figura 4-4: Frente de Pareto

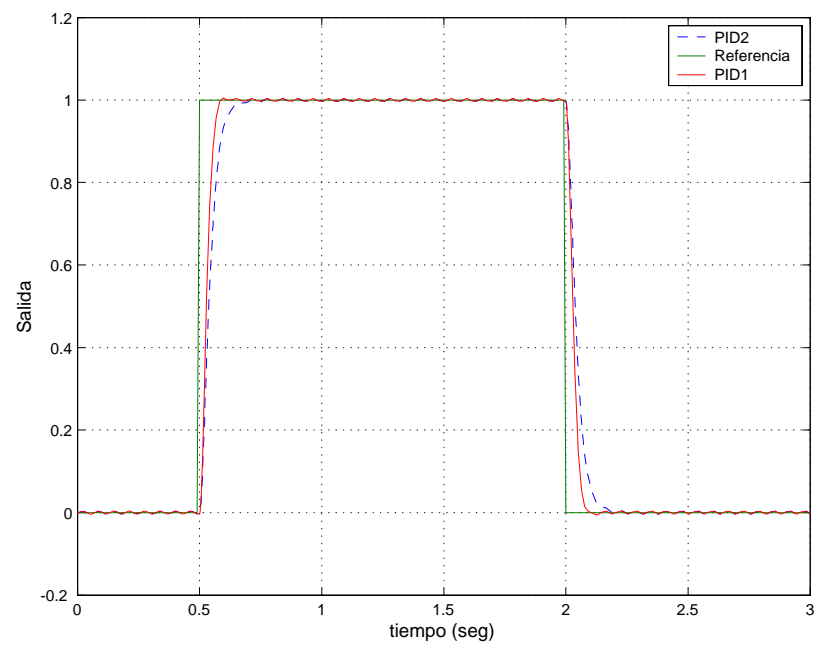


Figura 4-5: Salida del proceso

4.6. Conclusiones

En relación a nuestra evaluación experimental. Comparamos el funcionamiento que se obtuvo, aplicando el controlador al sistema real, contra el resultado que se obtiene con el controlador en un esquema de control simulado. Ambos resultados son similares, pero en el esquema de control simulado no se presenta un efecto apreciable de la perturbación. Esto es obvio dado que la diferencia, es debida a la naturaleza no lineal del sistema real.

En general, podemos concluir que el enfoque mediante algoritmos genéticos para resolver problemas de control, basados en la mezcla H_2/H_∞ (considerando controladores PID como el propuesto), es una buena elección cuando las dinámicas de la planta real son muy cercanas a las de un sistema lineal. En nuestro estudio, aplicamos un algoritmo genético estándar para determinar las ganancias del controlador. En la primera etapa se encontraron valores de las ganancias que no podían ser aplicados a un servomotor real, principalmente por la limitante en el valor de la ganancia derivativa. Esto se pudo apreciar experimentalmente con nuestro ejemplo ilustrativo (porque un valor grande de la ganancia derivativa produce un efecto indeseable que se manifiesta en forma de vibración en el motor de CD). Lo anterior se pudo superar con la adición de una segunda restricción al problema de optimización, como se puede ver en los resultados mostrados anteriormente y que coinciden con los que se obtuvieron durante el proceso de sintonización a prueba y error, realizado por un experto con la planta experimental. También se puede agregar que para este problema en particular, los resultados obtenidos con el algoritmo genético estándar son parecidos a los que se obtienen con el algoritmo genético multiobjetivo.

Capítulo 5

Sintonizado de un Controlador PID Robusto usando NSGA-II

5.1. Introducción

Para efecto de esta investigación se utiliza un controlador PID comercial de forma ISA con dos grados de libertad, el cual contiene siete parámetros y es sintonizado mediante un algoritmo genético multiobjetivo, el NSGA-II, que fue detallado en el Capítulo 5 de esta tesis. En este capítulo se plantea un diseño multiobjetivo de PIDs a partir de la minimización de normas de diferentes funciones de transferencia de lazo cerrado. Se desea minimizar la rapidez de respuesta ante una señal de referencia (problema de regulación), minimizar la sensibilidad ante cambios en la estructura de la planta, atenuar el efecto de la perturbación en la carga y minimizar el posible efecto del ruido en la señal de salida del sistema. A este algoritmo genético se le adicionó una interfaz que permite interactuar con MATLABTM (utilizando MATLAB-Engine), con el fin de que en este *software* sean evaluadas las funciones objetivo y las restricciones, que permiten la sintonización del controlador PID para cada planta en particular. Esta decisión se tomó en virtud de que MATLABTM contempla entre sus librerías las funciones que permiten evaluar normas de Sistemas Lineales Invariantes en el Tiempo, las cuales se usan en el proceso de evaluación

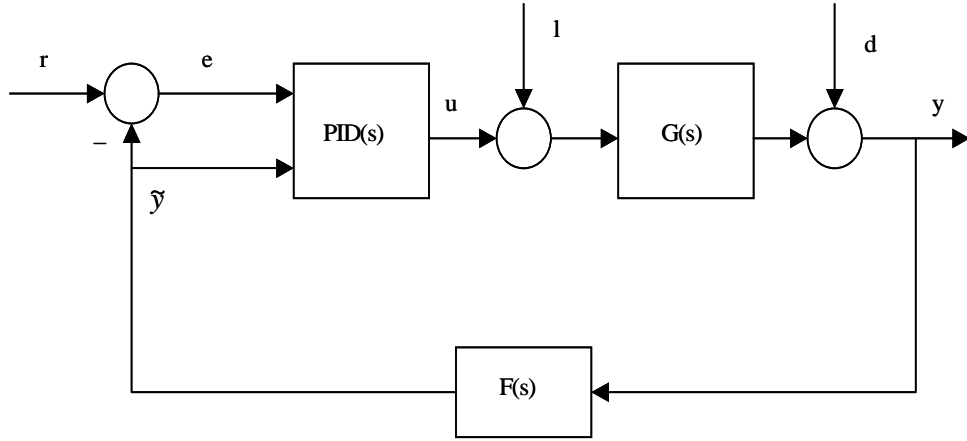


Figura 5-1: Sistema de control en lazo cerrado con un controlador PID

de los objetivos y restricciones de los problemas de optimización tratados en este capítulo. Para efecto de evaluar los resultados, éstos se comparan con los obtenidos por Herreros [36] y en algunos casos con los diseñados con el método propuesto por Skogestad en su artículo [71]. En la Figura 5-1 se muestra el esquema de control utilizado, incluyendo el controlador mencionado anteriormente:

5.2. Diseño del controlador PID

Con respecto al esquema de control mostrado en la Figura 5-1 :

r denota la señal de referencia;

e es la señal de error;

\tilde{y} denota la señal de salida filtrada;

u es la señal de control;

l representa una perturbación de carga;

d es la señal de ruido de la salida del sistema;

y es la señal de salida del sistema;

$G(s)$ es la planta a controlar;

$F(s)$ es un filtro pasabajas para minimizar el efecto del ruido en la señal de salida; $PID(s)$ denota el controlador.

Las especificaciones de diseño de controladores PID están orientadas principalmente en tres sentidos: la optimización del tiempo de respuesta del sistema para alcanzar un valor de referencia deseado, la minimización del efecto de posibles ruidos en la señal de control u y a la salida y , así como la robustez ante variaciones en la estructura de la planta. Se trata de unificar estos tres objetivos en la optimización de una serie de normas funcionales de las transferencias existentes entre las diferentes entradas y salidas del sistema en lazo cerrado. El sistema de control está sujeto a perturbaciones, considerándose principalmente el que afecta a la carga (l) y el ruido sobre la señal de medición (d). En cuanto a la perturbación de carga, ésta se modela como una función escalón y el ruido a la salida como una señal de alta frecuencia. A partir de esta consideración, la señal de salida es filtrada en la etapa de retroalimentación. El modelo del filtro utilizado es el siguiente:

$$F(s) = \frac{1}{1 + sT_f} \quad (5.1)$$

y el modelo del controlador:

$$u(s) = k(e_p(s) + \frac{1}{T_i s} e(s) + \frac{T_d s}{1 + T_d s/N} e_d(s)),$$

$$e(s) = r(s) - \tilde{y}(s),$$

$$e_p(s) = br(s) - \tilde{y}(s) = be(s) + (b - 1)\tilde{y}(s),$$

$$e_d(s) = cr(s) - \tilde{y}(s) = ce(s) + (c - 1)\tilde{y}(s),$$

donde $b > 0$, $c > 0$. Por lo que el modelo del controlador está dado por:

$$PID(s) = \begin{bmatrix} C_1(s) \\ C_2(s) \end{bmatrix} = \begin{bmatrix} k(b + \frac{1}{T_i s} + \frac{cT_d s}{1+T_d s/N}) \\ k((b-1) + \frac{1}{T_i s} + \frac{(c-1)T_d s}{1+T_d s/N}) \end{bmatrix} \quad (5.2)$$

Ésta es la estructura de un controlador PID de la forma ISA y es una estructura muy usada en los PID's comerciales. Los parámetros b y c no tienen efecto sobre la respuesta del sistema de control en lo que se refiere a la perturbación de carga y el ruido en la señal de medición. Su efecto se presenta en la respuesta ante los cambios de referencia en la entrada de control (corresponde a una ponderación de la referencia) [3]. Esto es útil, considerando que en la mayoría de los casos la señal de referencia es un escalón. El sistema en lazo cerrado se puede representar matricialmente [26], en donde cada uno de sus elementos es una función de transferencia de la forma $T_{zw}(s)$, donde w es la entrada y z la salida. A continuación se presenta el sistema de lazo cerrado en forma matricial:

$$\begin{bmatrix} e(s) \\ y(s) \\ u(s) \end{bmatrix} = T(s) \begin{bmatrix} r(s) \\ l(s) \\ d(s) \end{bmatrix} = \begin{bmatrix} T_{er}(s) & T_{el}(s) & T_{ed}(s) \\ T_{yr}(s) & T_{yl}(s) & T_{yd}(s) \\ T_{ur}(s) & T_{ul}(s) & T_{ud}(s) \end{bmatrix} \begin{bmatrix} r(s) \\ l(s) \\ d(s) \end{bmatrix} \quad (5.3)$$

Considerando el sistema de control de la Figura 5-1 y de (5.3) se tiene:

$$T(s) = \frac{1}{1 + F(s)C(s)G(s)} \begin{bmatrix} 1 - F(s)G(s)C'(s) & -F(s)G(s) & -F(s) \\ G(s)C(s) & G(s) & 1 \\ C''(s) & 1 & -F(s)C(s) \end{bmatrix}$$

$$= \frac{1}{1 + F(s)G(s)k(1 + \frac{1}{T_i s} + d')} \begin{bmatrix} 1 - F(s)G(s)k(b + cd' - d' - 1) & -F(s)G(s) & -F(s) \\ G(s)C(s) & G(s) & 1 \\ k(b + \frac{1}{T_i s} + d'c) & 1 & -F(s)C(s) \end{bmatrix}$$

donde $d'(s)$ se define como sigue:

$$d'(s) := \frac{T_d s}{1 + T_d s/N}.$$

Especificaciones para el diseño del controlador PID.

El criterio para el diseño del controlador PID debe satisfacer las siguientes especificaciones:

- Seguimiento de la señal de referencia.
- Atenuación del efecto de la señal de disturbio (l) que actúa sobre la señal de control (u).
- Robustez frente a las incertidumbres de la planta.
- Atenuación del efecto del ruido en la salida (ruido en la medición) (d).
- Limitación de la señal de control (u).

La importancia de cada especificación depende de la aplicación. En el control de procesos es importante mantener la señal de referencia a pesar de posibles disturbios en la carga. Estos se pueden modelar de manera aceptable como un escalón. En control de motores y robótica es importante un buen seguimiento a la señal de referencia. Otro aspecto importante se refiere al hecho de que cuando se diseña un controlador, éste se basa en el modelo de una planta, que puede presentar incertidumbre paramétrica o estructural, por lo que hay que considerar esta situación para el diseño del controlador. A continuación se describen cada uno de los puntos tratados anteriormente.

Perturbaciones de carga Este efecto se va a medir por medio de la integral de la señal de salida para una entrada escalón aplicada en la señal de control. Aplicando el principio de superposición, es decir, haciendo cero los valores de la referencia y el ruido de salida, se tiene:

$$J_c = \int_0^\infty |y(t)|^p dt, \quad p \in \mathbb{N} \quad (5.4)$$

normalmente se elige 1 o 2 para p .

Dado que $y(t)$ en (5.4) se puede expresar como la transformada inversa de Laplace de $T_{yl}(s)\frac{1}{s}$ donde $T_{yl}(s)$ es la función de transferencia entre la perturbación l y la salida y , (5.4) se puede expresar como:

$$\left(\left\| T_{yl} \frac{1}{s} \right\|_p \right)^{1/p} \quad (5.5)$$

donde $\|\cdot\|_p$ expresa la norma- p . En este trabajo se toma $p = 1$ (norma \mathcal{L}_1). El valor de esta norma se define como (ver [80]):

$$\int_0^\infty |g(t)| dt = h_1 + h_2 + h_3 + h_4 + \dots$$

La norma \mathcal{L}_1 mide los picos de una señal $g(t)$ en el dominio temporal como se muestra en la Figura 5-2 (se puede encontrar información para construir un método numérico de cómputo en [10] y [80]).

Robustez frente a las incertidumbres de la planta Esta especificación se puede medir con la función de sensibilidad en lazo cerrado $S(s) = \frac{1}{1+L(s)}$, donde $L(s)$ es la función de transferencia en lazo abierto del sistema de control mostrado en la Figura 5-1. $L(s)$ está dada por:

$$\begin{aligned} L(s) &= F(s)G(s)C(s) \\ &= F(s)G(s)k\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1+T_d s/N}\right). \end{aligned}$$

También, $S(s)$ está dada por la función de transferencia $T_{yd}(s) = \frac{1}{1+L(s)}$, mostrada anteriormente. El valor máximo de la ganancia de la sensibilidad (de la respuesta en frecuencia) se obtiene mediante la evaluación de la norma H_∞ de $T_{yd}(s)$, representada

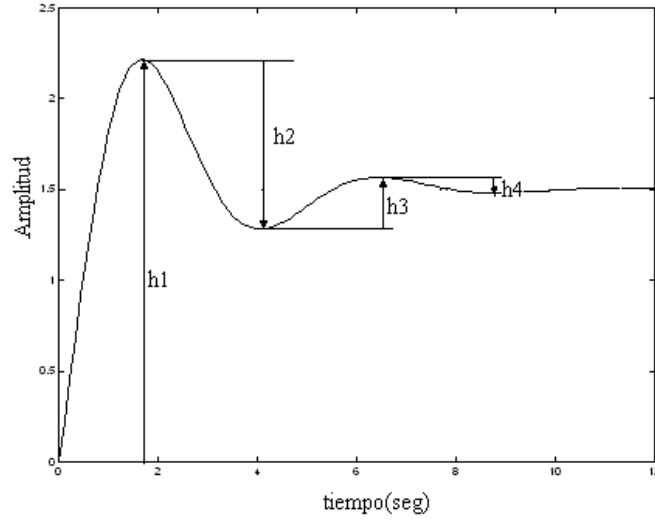


Figura 5-2: Estimación de la norma \mathcal{L}_1 de $g(t)$

por $M_s(s) = \|T_{yd}(s)\|_\infty$. Para un sistema SISO el inverso de $\|T_{yd}(s)\|_\infty$ corresponde a la distancia mínima de la curva de Nyquist del sistema en lazo abierto ($F(s)G(s)C(s)$), al punto crítico -1 . En [4] se recomienda un rango aceptable para $M_s(s) = [1.1, 2]$.

Seguimiento de la señal de referencia Este aspecto se mide por las características de la respuesta del sistema a una señal de referencia de tipo escalón, considerándose entre los puntos más importantes: el tiempo de subida, el tiempo de establecimiento, el sobrepico y el error en estado estacionario. Para caracterizar la respuesta temporal del sistema de control ante una señal de referencia, se utiliza el siguiente índice de desempeño:

$$J_t = \left(\int_0^\infty |te(t)|^p dt \right)^{1/p} \quad (5.6)$$

donde $e(t)$ es la salida a una entrada escalón en $r(t)$. Así, si $e(t)$ es la respuesta impulso de un sistema con una función de transferencia $T_{er}(s)\frac{1}{s}$, entonces $te(t)$ es la respuesta impulso a un sistema con una función de transferencia $\frac{d}{ds}(T_{er}(s)\frac{1}{s})$ (aplicando el teorema de la diferenciación real). Por lo tanto el índice de desempeño J_t es equivalente a:

$$J_f = \left\| \frac{d}{ds} \left(T_{er} \frac{1}{s} \right) \right\|_p \quad (5.7)$$

en este caso se hace $p = 2$.

Restricción de la señal de control Es importante mantener la señal de control $u(t)$ dentro de ciertos valores, sobre todo, en aplicaciones prácticas donde se pueden tener problemas de saturación en el actuador del sistema. De lo anterior se ha incorporado al problema de optimización multiobjetivo una restricción, llamada restricción de saturación ($r1$); ésta mide el valor más grande de $u(t)$, y está formulada por:

$$r1 = \|T_{ur}(s)\|_\infty < \infty \quad (5.8)$$

donde $u(t)$, es la transformada inversa de Laplace de la función de transferencia $T_{ur}(s)$.1.

5.2.1. Plantas de prueba

Con el objetivo de mostrar el procedimiento de sintonización de controladores PID comerciales con la metodología utilizada en esta investigación, se utilizan las plantas de prueba con las que trabajó Herreros en su tesis doctoral [36], tomadas a su vez del artículo “*Design of PI Controllers based on Non-Convex Optimization*”, ver [4]. Estas son presentadas en la Tabla 5.1.

Características de las plantas de prueba

Los sistemas $G_1(s)$ y $G_2(s)$ representan procesos que son relativamente fáciles de controlar. El sistema $G_3(s)$ tiene un retardo de tiempo grande, y $G_4(s)$ es un modelo integral. El Sistema $G_5(s)$ tiene un cero en el semiplano derecho del plano complejo “ s ” y el sistema $G_6(s)$ tiene polos complejos con un amortiguamiento relativo $\zeta = 0.33$. $G_7(s)$ representa un sistema con retardo puro, siendo $w_{90} = \pi/2$ y $w_{180} = \pi$. $G_8(s)$ es un

$G_1(s) = \frac{1}{(s+1)^3}$
$G_2(s) = \frac{1}{(s+1)(1+0.2s)(1+0.04s)(1+0.008s)}$
$G_3(s) = \frac{e^{-15s}}{(s+1)^3}$
$G_4(s) = \frac{1}{s(s+1)^2}$
$G_5(s) = \frac{1-2s}{(s+1)^3}$
$G_6(s) = \frac{9}{(s+1)(s^2+2s+9)}$
$G_7(s) = e^{-s}$
$G_8(s) = \frac{e^{-s}}{s}$
$G_{10}(s) = \frac{100}{(s+10)^2} \left(\frac{1}{s+1} + \frac{0.5}{s+0.5} \right)$
$G_{11}(s) = \frac{150}{(s+10)^2(s+1)}$
$G_{12}(s) = \frac{9}{(s+1)(s^2+as+9)}$
$G_{13}(s) = \frac{(s+6)^2}{s(s+1)^2(s+36)}$
$G_{14}(s) = \frac{a}{(s+a)(s-1)}$

Cuadro 5.1: Plantas de prueba

modelo con integrador puro y un tiempo de retardo. En este sistema se tiene $w_{90} = 0$ y $w_{180} = \pi/2$. El Sistema $G_9(s)$ tratado en [4] es un modelo descrito por ecuaciones diferenciales parciales, por lo que no se incluye en estas plantas de pruebas. Sin embargo se prefiere conservar la nomenclatura original de las plantas del artículo [4], como se hace también en [36]. La planta $G_{10}(s)$ presenta dos dinámicas rápidas con una constante de tiempo de $0.1s$. Un modo con una constante de tiempo de $1s$ y dos modos lentos con constante de tiempo igual a $20s$. El comportamiento estático es dominado por los modos lentos, que tienen una ganancia a baja frecuencia igual a 10. La respuesta al escalón es dominada por la constante de tiempo lenta, pero son los modos rápidos los determinantes del desempeño del sistema en lazo cerrado. El sistema $G_{11}(s)$ se obtiene removiendo el modo más lento de $G_{10}(s)$. El modelo representado por $G_{12}(s)$ tiene dos polos oscilatorios con amortiguamiento relativo $\zeta = a/6$. Conforme a toma valores más pequeños, el sistema se hace más difícil de controlar, hasta llegar al caso extremo cuando a toma el valor cero. $G_{13}(s)$ es un sistema condicionalmente estable. Este sistema no satisface el principio de monotonía porque el retardo de fase no es monótono. La curva de Nyquist cruza el eje real negativo en dos ocasiones, en $s = -0.0191$ y $s = -0.1656$ y

se tiene $w_{90} = 0$ y $w_{180} = 1.69$. Finalmente, $G_{14}(s)$ es un sistema inestable. Su fase no decrece monótonamente. El valor de la fase es igual a π para bajas y altas frecuencias. La fase mínima ocurre para $w = \sqrt{a}$ y es igual a $\pi - \arctan \frac{(a-1)\sqrt{a}}{2a}$.

5.2.2. Evaluación de resultados

Como se mencionó en la introducción de este capítulo, los resultados obtenidos se comparan con los dados por Herreros y también para este objeto se diseñaron algunos controladores basados en el método de Skogestad. Este método originalmente no contempla ruido a la salida del sistema, sin embargo, debido a la robustez de los controladores PID diseñados con este método, éstos pueden manejar un nivel de ruido a la salida comparable con los que usó Herreros en su tesis [36].

En las gráficas que se presentan en este capítulo, las respuestas en el tiempo se obtuvieron aplicando un escalón unitario a la entrada. Como perturbación en la carga se aplicó también un escalón de magnitud igual a 0.5 y a la salida del sistema se aplica una señal de alta frecuencia. Las señales referenciadas con PID1 corresponden a los resultados obtenidos, PID2 corresponde a los resultados de Herreros [36], y finalmente, en algunos casos PID3 a los controladores diseñados con el método de Skogestad. En las figuras del frente de Pareto y sus proyecciones, corresponden a las soluciones óptimas que se obtuvieron al aplicar el algoritmo genético multiobjetivo NSGA-II a cada planta en particular (más de 5 ejecuciones por cada planta). A continuación se presenta una metodología para obtener las “mejores soluciones” (a nuestro juicio) del frente de Pareto, proporcionado por el NSGA-II.

Regla para seleccionar las mejores soluciones

Para efectos de sintonización de controladores PID tratados en este trabajo desde el punto de vista de un problema de optimización multiobjetivo, se minimizan tres funciones objetivo con dos restricciones. El primer objetivo minimiza una señal de disturbio, modelada por un escalón que está presente en la señal de control. Este objetivo se representa

por la función de transferencia $T_{yl}(s)$, donde y es la señal de salida y l es la señal de disturbio presente en la señal de control. En el primer objetivo se minimiza $\|T_{yl}(s)\frac{1}{s}\|_1$, donde $\frac{1}{s}$ es necesario para eliminar el cero en $T_{yl}(s)$. El segundo objetivo $\|\frac{d}{ds}T_{er}(s)\frac{1}{s}\|_2$ está relacionado con el seguimiento de la señal de salida con respecto a la referencia en el dominio temporal. El tercer objetivo contempla la robustez frente a las incertidumbres de la planta, este puede ser medido con la función de sensibilidad ($S(s)$), en lazo cerrado, representado por $T_{yd}(s)$. El pico de ganancia de sensibilidad viene dado por su norma H_∞ . En un sistema SISO, el valor de la ganancia de sensibilidad corresponde a la inversa de la mínima distancia de la curva de Nyquist del sistema en lazo abierto al punto crítico -1 . Para obtener robustez frente a incertidumbres de la planta su valor debe ser pequeño. De lo anterior, el objetivo 3 está dado por $\|T_{yd}(s)\|_\infty$. En cuanto a las restricciones, la primera reduce el espacio de búsqueda a las soluciones que proporcionan estabilidad a las plantas en lazo cerrado, verificando que todos los polos del determinante del sistema $T(s)$ sean estables. La segunda restricción se utiliza para acotar la señal de control u a valores deseados y está representado por la función de transferencia $T_{ur}(s)$. Una manera de acotar la señal de control u es evaluando la norma H_∞ de $T_{ur}(s) \cdot 1$, es decir se trata de acotar u a una entrada impulso, considerando que si u tiene un valor acotado para dicha señal, entonces también será acotada para otra señal diferente finita. Esta restricción se plantea de la siguiente forma:

$$\|T_{ur}(s) \cdot 1\|_\infty < \alpha$$

El procedimiento para seleccionar las mejores soluciones, después de aplicar el algoritmo genético, es el siguiente:

1. El valor del tercer objetivo ($\|T_{yd}(s)\|_\infty$) se selecciona en el rango $[1.08 - 1.70]$, este intervalo es recomendado en [4].
2. Se toman los valores más pequeños correspondientes a los objetivos 1 y 2, cumpliendo con el paso 1.

3. Se prefiere el valor más pequeño del objetivo 2, es decir, le damos mayor peso al objetivo 2, ya que éste, está relacionado con el problema de seguimiento de la señal de entrada (si se desea esta opción).

Programa en MATLABTM para seleccionar las mejores soluciones del frente de Pareto.

Después de ejecutarse el algoritmo genético multiobjetivo (NSGA-II) en una plataforma PentiumTM a una velocidad de procesamiento de 1.4 GHz. El algoritmo crea entre otros, dos archivos de salida llamados *final_var* y *plot*, que contienen los valores de los parámetros óptimos (frente de Pareto) y de los objetivos, respectivamente. Se hizo un programa en MATLABTM versión 6.5 Release 13 para simplificar la búsqueda de las mejores soluciones del frente de Pareto. Para esto se usan *final_var* y *plot*. Primero el programa concatena los datos de los dos archivos, para tener en una sola matriz los parámetros y los objetivos. Segundo, se seleccionan las soluciones para las cuales el valor del Objetivo 3 (objetivo que mide M_s), está en el rango [1.09-1.70]. Después el usuario selecciona entre las opciones: mejor respuesta de la señal de referencia o mejor atenuación a la perturbación. Al final el programa entrega las 4 mejores soluciones (los 7 parámetros del controlador PID más el valor de los 3 objetivos). La intención de tener más de una respuesta es para satisfacer el espíritu de la estrategia usada, ya que una de las características principales de la optimización multiobjetivo es alcanzar más de una solución óptima.

Valores de los parámetros usados para el NSGA-II

Como se mencionó en la introducción de este capítulo, las funciones objetivo y las restricciones son evaluadas en MATLABTM. Por lo tanto, es necesario hacer una función en lenguaje de MATLABTM con los objetivos y las restricciones del problema de control en particular. Se hicieron las modificaciones necesarias al programa fuente del NSGA-II, aprovechando las librerías de “MATLAB Engine” para que después de compilarse,

sea un programa ejecutable que pueda abrir MATLB^{TM} e interactuar con este lenguaje pasándole los parámetros del controlador PID como argumentos de la función creada previamente y ser evaluados en éste. En cada generación, el NSGA-II pasa individualmente a MATLB^{TM} cada uno de los elementos de la población (7 parámetros codificados por cada cromosoma), los cuales son evaluados a través de las funciones objetivo y las restricciones y estos resultados son regresados al NSGA-II para ser utilizados en el proceso genético completo. Dentro del programa fuente del NSGA-II se hace referencia a la función de MATLB^{TM} con el nombre de $F.m$ (nombre creado arbitrariamente). Por lo que cuando se desea encontrar las soluciones para sintonizar algún controlador PID en particular se tiene que crear una función en MATLB^{TM} con este nombre. Los archivos que entrega el algoritmo genético son entre otros *output.out* y *plot.out*. En éstos se presentan las soluciones de la última generación y el frente de Pareto respectivamente. A continuación se muestra el procedimiento completo (Figura 5-3) usando un diagrama de flujo y una breve explicación de los bloques que lo componen:

1. Se escriben los objetivos y las restricciones (del esquema de control de la planta deseada), en MATLB^{TM} y se salva con el nombre $F.m$. Los objetivos y las restricciones del problema están en función de los 7 parámetros del controlador (Ti, Tf, Td, N, k, b, c) que también aparecen en el archivo *mlb.txt*.
2. Se construye en *simulink*TM el sistema a controlar para ser utilizado posteriormente en el paso 6.
3. Se construye un archivo de texto que contiene los datos de los parámetros del NSGA-II. Estos son: número de parámetros (reales y/o binarios), número de objetivos, número de restricciones, probabilidad de cruzamiento, probabilidad de mutación, precisión y rango de cada uno de los parámetros.
4. En el sistema operativo “MS-DOS” (símbolo del sistema), se ejecuta el algoritmo genético NSGA-II. El archivo ejecutable se llama *nsga2u.exe*. Con fines prácticos se hizo un archivo de texto (mencionado en el paso 3), el cual contiene los datos

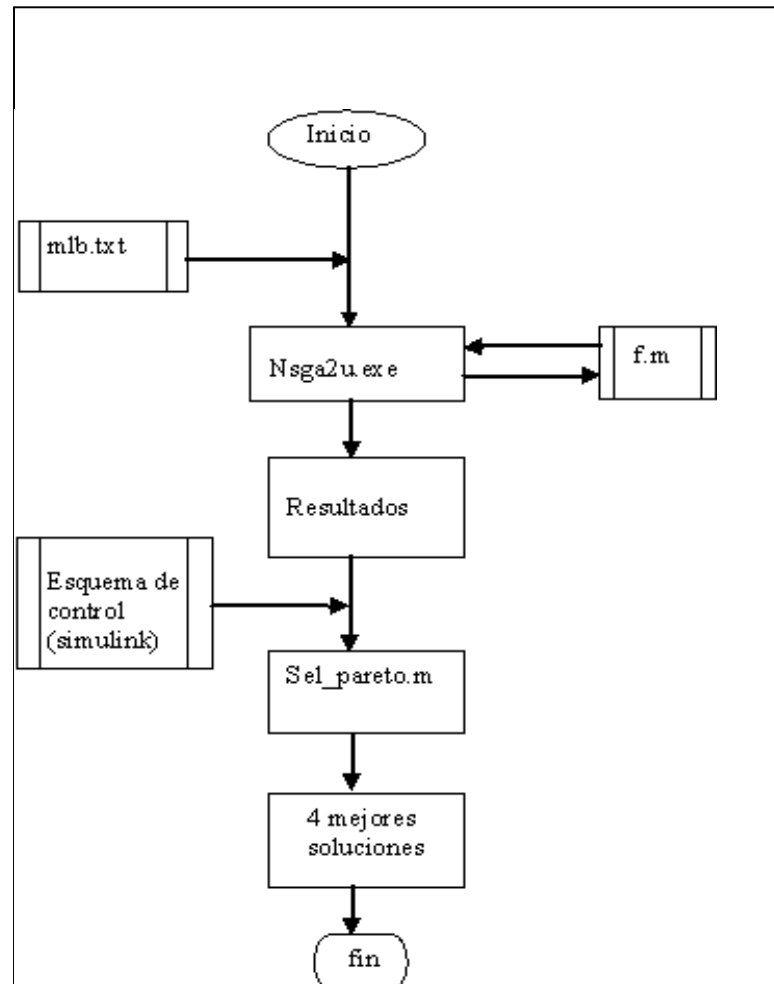


Figura 5-3: Sintonización PID

que necesita el algoritmo genético. Este último archivo llamado *mlb.txt* debe estar en la misma carpeta donde se encuentra *nsga2u.exe*. Suponiendo que estos archivos estén en la carpeta *Temp.*, entonces se escribe como: $C:\backslash Temp\backslash nsga2u < mlb$.

5. Después de ejecutarse el programa *nsga2u.exe*, éste genera varios archivos de salida, entre ellos *plot.out* y *final_var.out* . El primero contiene los valores de los objetivos optimizados del frente de Pareto y el segundo los valores de los parámetros (Ti, Tf, Td, N, k, b, c) correspondientes a las soluciones no-dominadas. Las restricciones junto con los objetivos y los parámetros del controlador están en el archivo *output.out*.
6. Para seleccionar las mejores soluciones del frente de Pareto, se hizo una función llamada *sel_pareto.m*, la cual utiliza los datos de los archivos *plot.out* y *final_var.out* para mostrar una figura con los mejores cuatro controladores. La función *sel_pareto.m* presenta la opción al usuario de seleccionar entre una mejor respuesta de la señal de referencia o atenuación de perturbación de carga. Durante la ejecución de *sel_pareto*, éste hace uso del modelo creado en simulinkTM. También, dentro de *sel_pareto* se llama a una función llamada *pid1.m*, que recibe como argumentos los 7 parámetros del controlador en cuestión (cada uno de los cuatro controladores) y regresa las funciones de transferencia del controlador usadas por el modelo construido en simulinkTM (paso 2).
7. Si el usuario desea, también puede ver gráficamente el frente de Pareto, graficando los datos del archivo *plot.out*.

5.2.3. Valores de los parámetros del NSGA-II

El NSGA-II fue utilizado usando codificación real y codificación binaria, obteniéndose prácticamente los mismos resultados. El valor de los parámetros del NSGA-II para los dos casos, fueron los siguientes:

Codificación binaria

- Tamaño de la población: 100.
- Número de generaciones: 200.
- Probabilidad de cruzamiento: 0.80.
- No. de puntos de cruzamiento: 1.
- Probabilidad de Mutación: 0.09.

Codificación Real

- Tamaño de la población: 100.
- Número de generaciones: 200.
- Probabilidad de cruzamiento: 0.80.
- Índice de distribución de cruzamiento real (SBX): 10.
- Probabilidad de Mutación: 0.09.
- Índice de distribución de mutación real: 50.

Los índices de distribución de cruzamiento real y mutación real son utilizados en el NSGA-II por los operadores de cruzamiento y mutación respectivamente cuando se usa la opción de codificación real. Estos operadores producen descendientes basados en una distribución de probabilidad que son similares a los que se producen utilizando operadores binarios [25].

5.2.4. Valores de los parámetros de los controladores PID

Para iniciar la búsqueda de los 7 parámetros que sintonizan cada uno de los controladores PID, mediante el algoritmo genético multiobjetivo, se requiere dar a éste los rangos iniciales de los parámetros del problema a optimizar. Para efecto de sintonizar los controladores incluidos en este capítulo se utilizaron los rangos de la Tabla 5.2.

PID	T_i	T_f	T_d	N	k	b	c
	0-50	0-2	0-20	0-120	0-50	0-1	01

Cuadro 5.2: Rango de los parámetros

$C(s)$	T_i	T_f	T_d	N	k	b	c	Obj 1	Obj 2	Obj3
PID_1	2.6328	0.003	0.782	43.4	3.695	0.713	0.015	0.714	0.99	1.46
PID_2	0.4110	0.008	0.132	19.1	11.60	0.162	0.002	0.039	0.13	1.49
PID_3	8.1881	0.019	1.919	10.9	0.3968	0.073	0.0002	26.08	67.8	1.56
PID_4	8.1414	0.001	1.328	26.2	0.6734	0.523	0.0043	12.09	3.7	1.33
PID_5	1.2160	0.006	2.808	0.9	0.1543	0.072	0.0175	9.475	13.4	1.43
PID_6	0.4420	0.008	0.185	1.38	0.3831	0.227	0.0064	1.328	1.1	1.32
PID_7	0.2043	0.008	4.075	1.01	0.0884	0.103	0.0030	2.699	1.9	1.22
PID_8	8.2153	0.009	0.046	1.01	0.3907	0.598	0.8740	21.14	2.89	1.44
PID_{10}	0.5540	0.008	0.133	1.14	2.6673	0.259	0.0545	0.239	0.25	1.64
PID_{11}	0.7992	0.008	0.119	15.7	4.5576	0.726	0.0035	0.175	0.09	1.37
PID_{12}	0.0004	1.001	15.68	5.48	0.0001	0.482	0.9423	23.32	5.33	1.39
PID_{13}	2.5910	0.257	1.209	10.7	1.3843	0.563	0.6669	2.622	0.94	1.72
PID_{14}	0.4421	0.008	0.156	33.8	22.102	0.183	.00014	0.021	0.11	1.16

Cuadro 5.3: Parámetros de los controladores

Finalmente, en la Tabla 5.3 se presentan los valores de cada uno de los 7 parámetros que componen los controladores PID usados para controlar las trece plantas propuestas en este capítulo. También se proporciona el valor de cada función objetivo. Las soluciones que se presentan para cada una de las plantas, fueron seleccionadas aplicando el criterio de mejor respuesta de referencia de entrada. Estas soluciones fueron tomadas del frente de Pareto, contenidas en los archivos *plot.out* (valores de las funciones objetivo) y *final.out* (soluciones correspondientes a los valores de las funciones objetivo).

A continuación se muestran los frentes de Pareto de las trece plantas ($G_1(s) - G_8(s)$ y $G_{10}(s) - G_{14}(s)$), los cuales se graficaron tomando los valores de las tres funciones objetivo, contenidos en *plot.out*, que es uno de los archivos de salida del NSGA-II. En estas gráficas de los frentes de Pareto se puede apreciar el resultado del proceso de optimización multiobjetivo en el espacio de las funciones objetivo. De las gráficas del frente de Pareto

y sus proyecciones, el Objetivo 1 (Obj 1) corresponde a la función objetivo que minimiza $\|T_{yl}(s)\frac{1}{s}\|_1$. El segundo Objetivo (Obj 2) minimiza a $\|\frac{d}{ds}T_{er}(s)\frac{1}{s}\|_2$ y el tercer Objetivo minimiza a $\|T_{yd}(s)\|_\infty$. También se presentan las gráficas de la respuesta en el tiempo de cada una de las plantas controladas (utilizando SimulinkTM de MAT-LABTM), donde se puede apreciar el cambio de la señal de salida (y) ante una entrada de tipo escalón y el efecto de una perturbación (escalón de valor igual a 0.5), aplicada en la señal de control u . También se adicionó una señal de alta frecuencia en la salida del proceso. Las señales denominadas bajo PID1 corresponden a las producidas por los controladores obtenidos en esta tesis, las denominadas bajo PID2 a los controladores sintonizados por Herreros y PID3 en algunas casos (cuando el tipo de sistema lo permite), a los controladores diseñados con el método de Skogestad. Con la finalidad de comparar el desempeño de los controladores obtenidos con los de Herreros y Skogestad, se utilizan algunos criterios de desempeño para evaluar la respuesta temporal, considerandose tanto la respuesta a la señal de referencia como el efecto de la perturbación en la carga. Para evaluar la respuesta a la señal de referencia se utilizan los siguientes índices de desempeño: $\int_0^\infty |e(t)| dt$, $\int_0^\infty t |e(t)| dt$ y PI . Para evaluar el efecto de la perturbación de carga se emplea $\int_0^\infty |e(t)| dt$. Los índices de desempeño usados en este capítulo son ampliamente citados en la literatura técnica, salvo PI , por lo cual se dará una breve explicación del mismo. El índice PI es igual a la integral del error al cuadrado más una función de penalización de sobretiro, denominada OvS . Este índice está dado por:

$$\int_0^\infty \tilde{x}^2 + OvS(r(t), x(t), \delta) dt$$

donde \tilde{x} es la señal de error, $r(t)$ es la señal de referencia y $x(t)$ es la señal de salida del proceso.

La función de penalización OvS calcula el sobretiro de la señal de salida del proceso, para una entrada de referencia tipo escalón, fuera de un intervalo δ propuesto por el usuario. Esta función toma un valor diferente de cero cuando la señal de salida del proceso alcanza a la señal de referencia y no permanece dentro del intervalo δ .

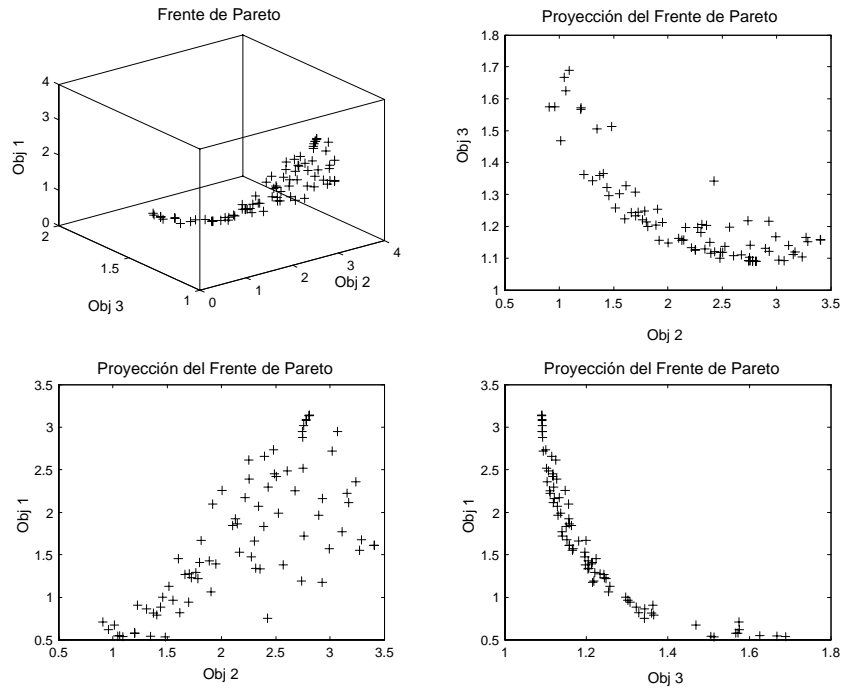


Figura 5-4: Planta 1

En la Figura 5-4 se muestra el frente de Pareto del problema de optimización multiobjetivo, utilizado para determinar los parámetros del controlador PID, aplicado al control de la Planta 1. Después de aplicar el algoritmo genético multiobjetivo al problema de optimización, se obtuvo un conjunto de soluciones no-dominadas, conteniendo cada una de ellas los 7 parámetros del controlador. Después se aplicó el programa `sel_pareto` y se obtuvieron 4 soluciones, presentándose una de ellas en la Tabla 5.3 (PID_1). Con estos parámetros se simuló el esquema de control utilizado en este capítulo, Figura 5-1, y el resultado correspondiente se presenta gráficamente en la Figura 5-5. En esa figura, se presentan también los resultados de los controladores sintonizados por Herreros y Skogestad. En la Tabla 5.4 se muestran los valores de los índices de desempeño, descritos anteriormente, para evaluar de manera cuantitativa el desempeño del controlador PID_1 con respecto a los otros dos controladores. Analizando los resultados se puede observar que para los cambios ante la entrada de referencia, el mejor resultado es proporcionado

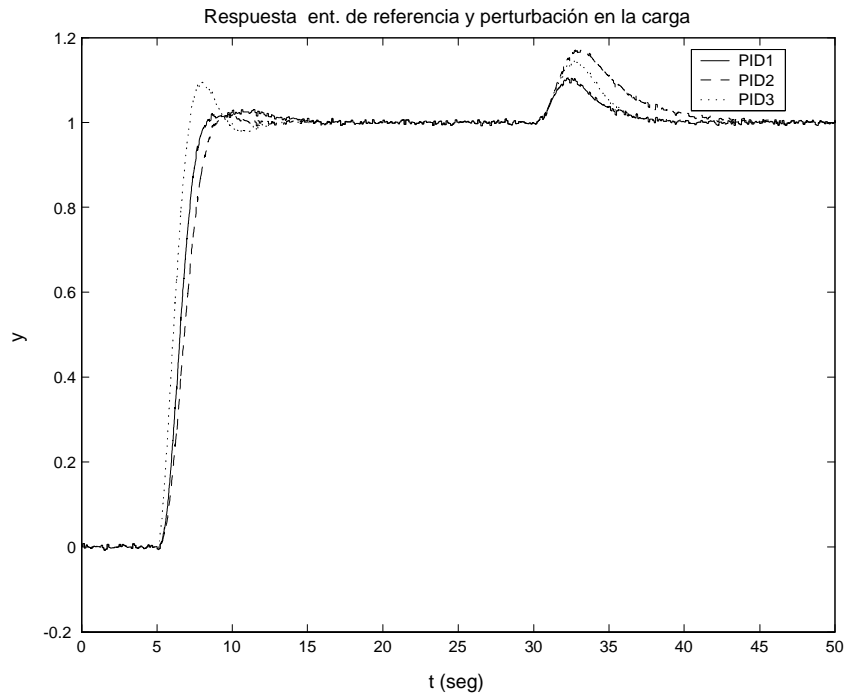


Figura 5-5: Planta 1

por PID2, dado que aunque el controlador PID3 presenta valores menores en los dos primeros índices, sin embargo éste presenta un gran sobretiro, evaluado por el tercer criterio de desempeño (PI). En cuanto a la perturbación de carga, aplicada en un tiempo $t = 30s$, se puede apreciar en la misma tabla que, el controlador más eficiente para atenuar el efecto de la perturbación es el controlador PID1. Esto se puede corroborar de manera gráfica en la Figura 5-5, donde la señal denominada PID1 presenta un valor de amplitud máximo, menor al de las otras dos señales, así como un tiempo de establecimiento menor.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	2.050	2.769	756.80	0.501
PID2	1.949	2.379	720.60	1.689
PID3	1.286	1.452	6148.0	1.0

Cuadro 5.4: Índices de desempeño de la Planta 1

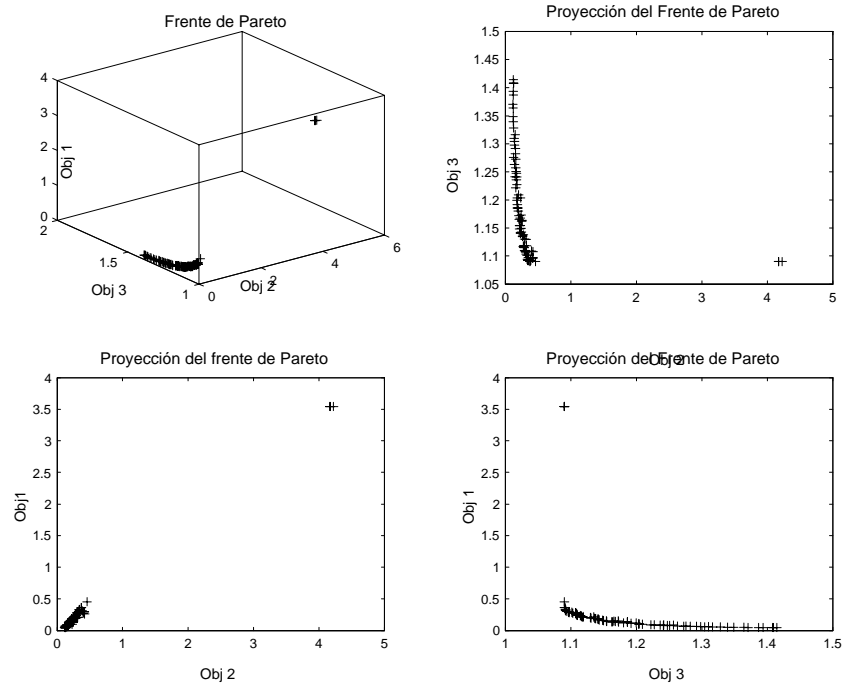


Figura 5-6: Planta 2

La Figura 5-6 muestra el frente de Pareto correspondiente al problema de optimización multiobjetivo utilizado para sintonizar al controlador PID de la Planta 2. Mediante el mismo procedimiento referido en el caso anterior, se seleccionó una de las soluciones del conjunto de óptimos de Pareto, la cual es presentada en la Tabla 5.3 (PID_2). En la Figura 5-7 se muestran las señales de salida de la Planta 2 correspondientes a los controladores PID1, PID2 y PID3 donde se puede observar que el mejor desempeño en cuanto al cambio de entrada de referencia corresponde al controlador PID2. Por otra parte, se puede observar en la misma figura que el controlador PID1 atenúa mejor el efecto de la perturbación de carga, aplicada en un tiempo $t = 10s$. En Tabla 5.5, los dos primeros índices reflejan que la respuesta a la entrada de referencia, debida al controlador PID3 es la más rápida, sin embargo presenta ésta un gran sobre tiro (ver el valor del índice PI). En cuanto al efecto de la perturbación de carga, PID1 es el controlador que tiene mejor desempeño.

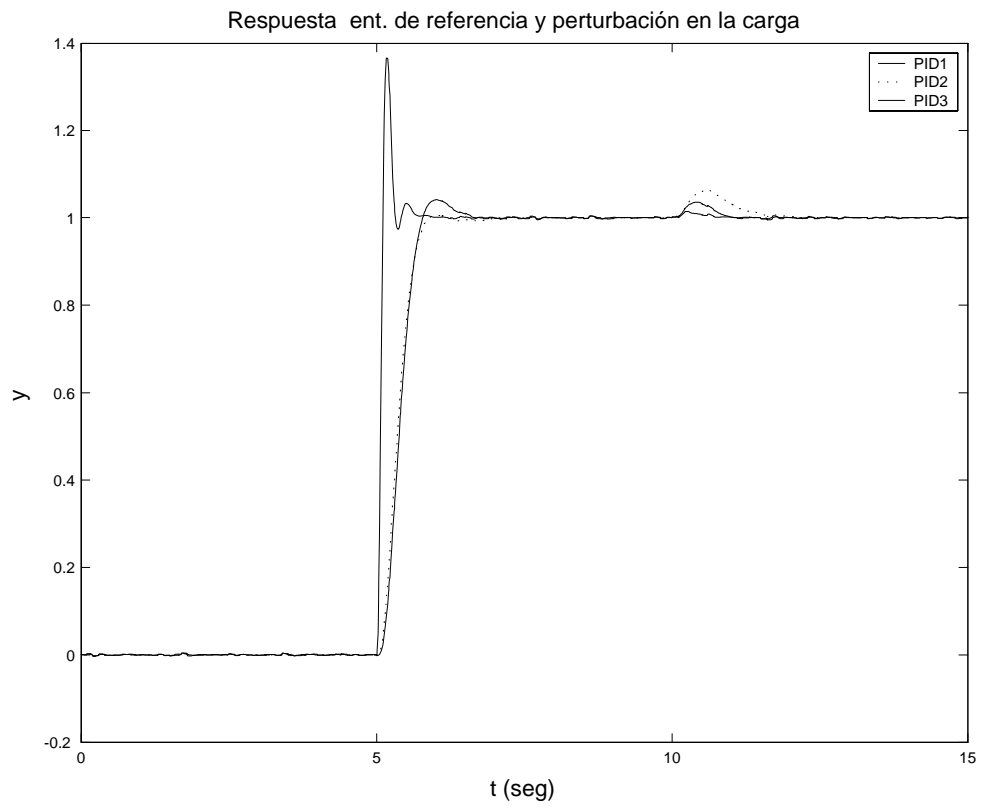


Figura 5-7: Planta 2

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	0.417	0.1170	1102.0	0.03811
PID2	0.389	0.0990	218.10	0.1063
PID3	0.116	0.0156	2597.0	0.0125

Cuadro 5.5: Índices de desempeño de la Planta 2

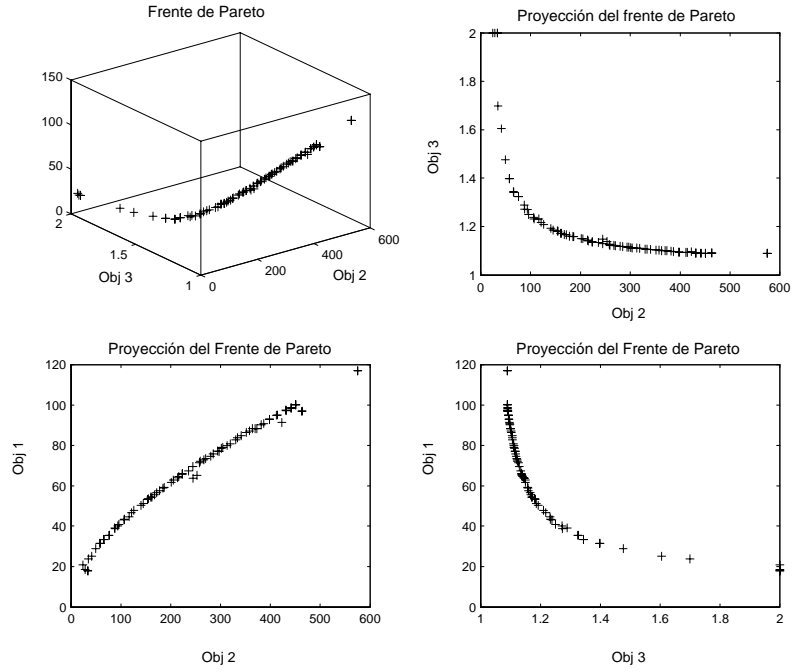


Figura 5-8: Planta 3

La Figura 5-8 muestra el frente de Pareto correspondiente al problema de optimización multiobjetivo utilizado para sintonizar al controlador PID de la Planta 3. Los parámetros de este controlador se encuentran en la Tabla 5.3 como PID_3 . Los resultados de la simulación se presentan en la Figura 5-9, donde se puede apreciar claramente un mejor desempeño del controlador PID1 tanto en la parte correspondiente al cambio de referencia, que es más rápido, como en la atenuación de la perturbación de carga, al alcanzar el estado estacionario en un tiempo más corto. También podemos ver en la gráfica, la respuesta inversa al inicio de las señales de referencia y de perturbación de carga, dado que el retardo que presenta la Planta 3 se aproximó por un modelo de Padé de primer orden, ésto adiciona al modelo un cero en el semiplano derecho del plano complejo (modelo de fase no-mínima). Los valores de los índices que evalúan el desempeño de los controladores de la Planta 3, están contenidos en la Tabla 5.6.

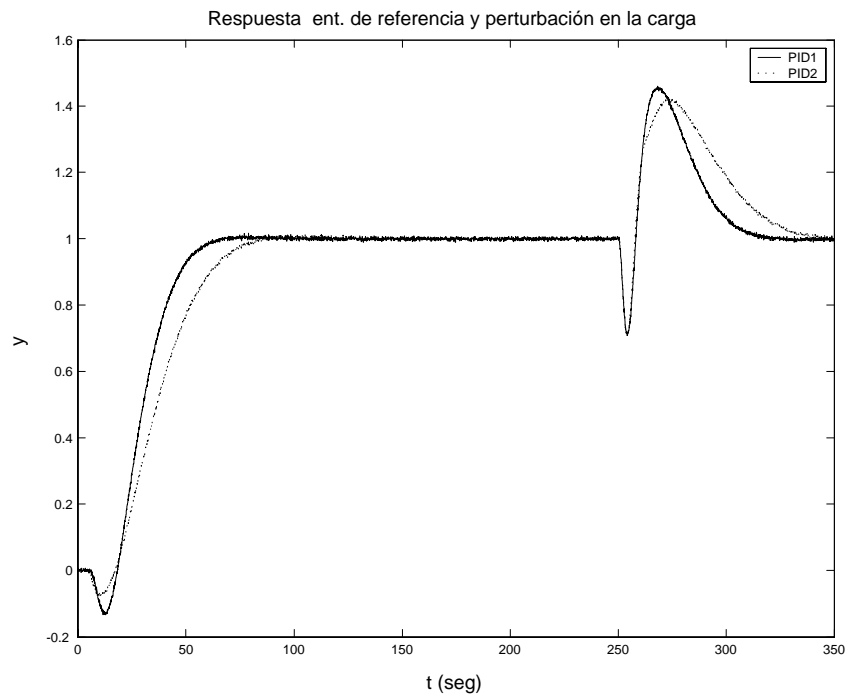


Figura 5-9: Planta 3

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	29.03	716.00	2544	26.63
PID2	35.36	1014.0	3356	35.28

Cuadro 5.6: Índices de desempeño de la Planta 3

A continuación se presentan los resultados de la Planta 4.

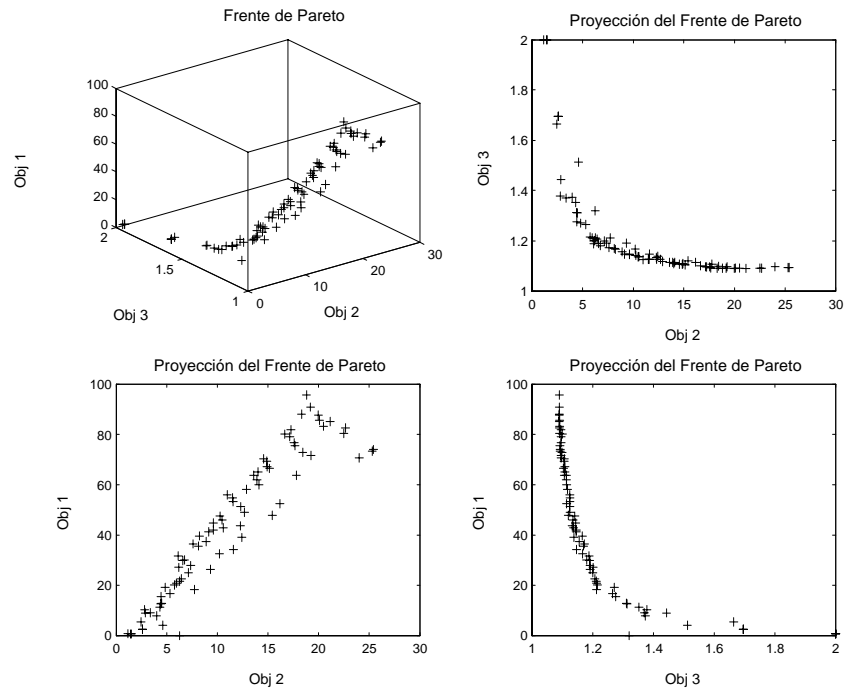


Figura 5-10: Planta 4

La Figura 5-10 muestra el frente de Pareto del problema de optimización multiobjetivo concerniente a la Planta 4, utilizada en este capítulo. El vector de parámetros elegido se encuentra también en la Tabla 5.3 llamado PID_4 . Los resultados de la simulación, empleando los parámetros mencionados, se presentan en la Figura 5-11. En esta última figura se puede ver que el controlador PID1 tiene el mejor desempeño con respecto a PID2 y PID3; considerando tanto la respuesta a la entrada de referencia como la perturbación de carga. Lo anterior puede ser avalado por los resultados de la Tabla 5.7.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	3.70	12.01	1071.00	18
PID2	4.93	18.94	1511.00	17
PID3	1.69	3.620	41080.0	23

Cuadro 5.7: Índices de desempeño de la Planta 4

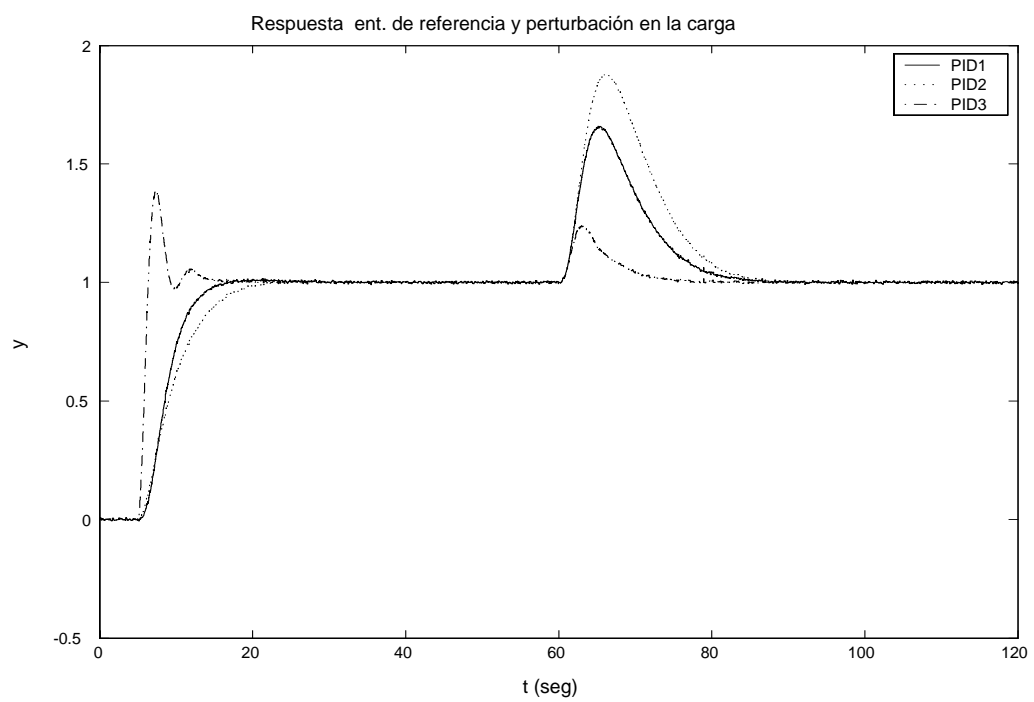


Figura 5-11: Planta 4

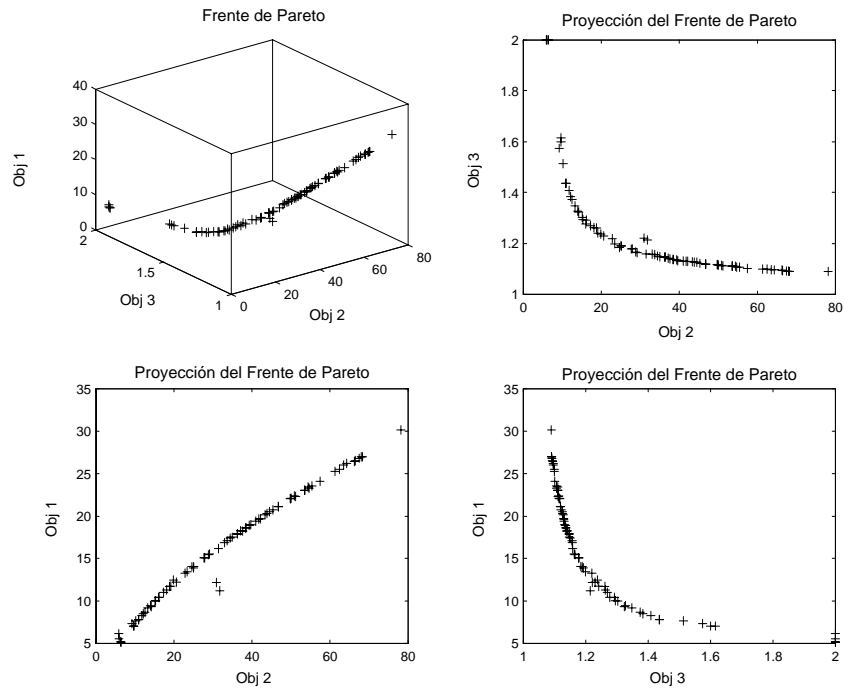


Figura 5-12: Planta 5

En la Figura 5-12 se muestra el frente de Pareto concerniente a la Planta 5. En la Figura 5-13 se presentan las gráficas que muestran el desempeño de los controladores PID1, PID2 y PID3. PID1 utiliza los parámetros de la solución elegida del conjunto de óptimos de Pareto reportada en la Tabla 5.3 como PID_5 . De la Figura 5-13 y de los valores de los índices de desempeño presentados en la Tabla 5.8 se puede decir que: el controlador PID3, de Skogestad, tiene el mejor desempeño, teniendo el inconveniente de presentar una amplitud mayor en la respuesta inversa, ya que la Planta 5 tiene un cero en el semiplano derecho del plano complejo.

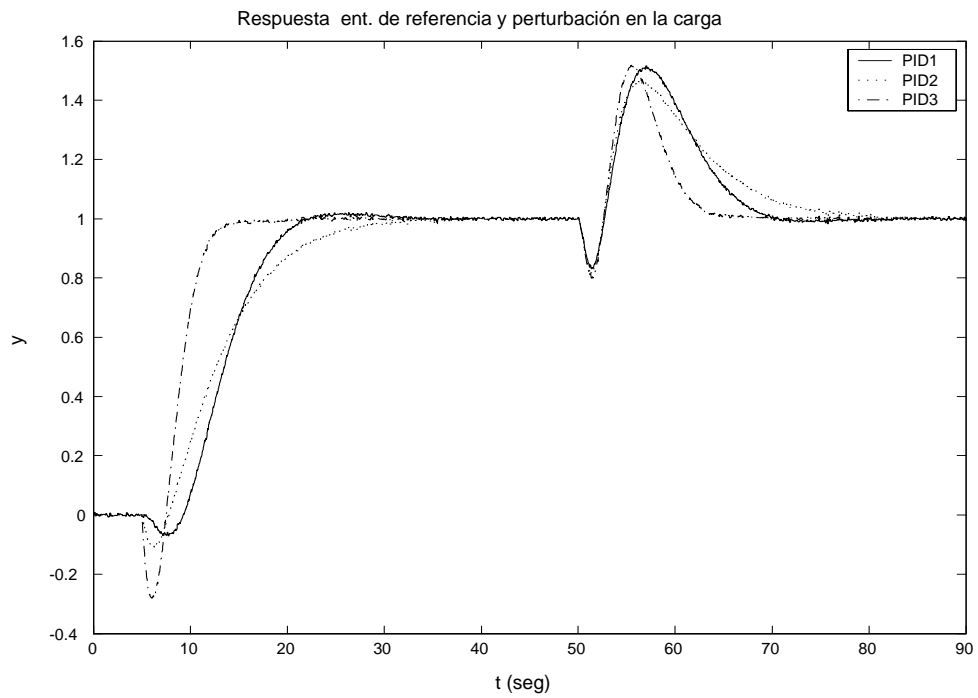


Figura 5-13: Planta 5

A continuación se presentan los índices de desempeño de la Planta 5.

Contr.	Desempeño de los controladores			
	Entrada de referencia			Perturbación en la carga
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	9.22	93.68	3788	9.193
PID2	9.26	102.4	3325	10.18
PID3	5.00	37.50	2335	6.23

Cuadro 5.8: Índices de desempeño de la Planta 5

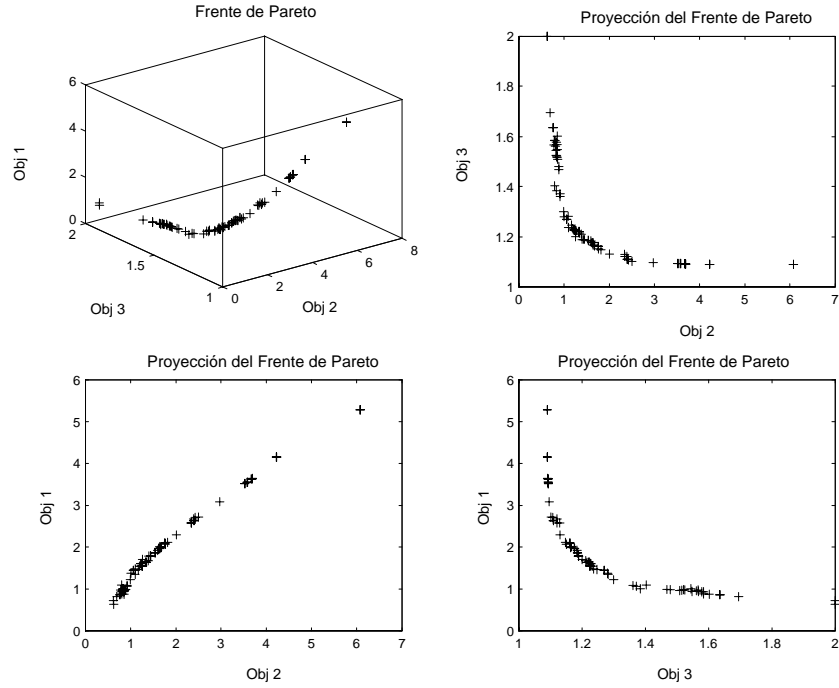


Figura 5-14: Planta 6

La Figura 5-14 presenta el frente de Pareto concerniente a la Planta 6. Referente a los resultados obtenidos con los controladores PID1 y PID2 (Herreros), utilizados como controladores de la Planta 6, se puede decir que: en base la Figura 5-15 y a la Tabla 5.9; el controlador PID1 tiene un mejor desempeño en la respuesta de entrada de referencia, considerando que aunque la señal denominada como PID2 es un poco más rápida (los valores de los índices $\int_0^\infty |e(t)| dt$ y $\int_0^\infty t |e(t)| dt$ son mejores para PID2), tiene sin embargo un mayor sobretiro. En cuanto a la atenuación de la perturbación de carga, aplicada en un tiempo $t = 30s$, también es ligeramente mejor el controlador PID1. Los parámetros del controlador PID1 fueron tomados del conjunto de soluciones óptimas de Pareto, reportado en la Tabla 5.3 como PID_6 .

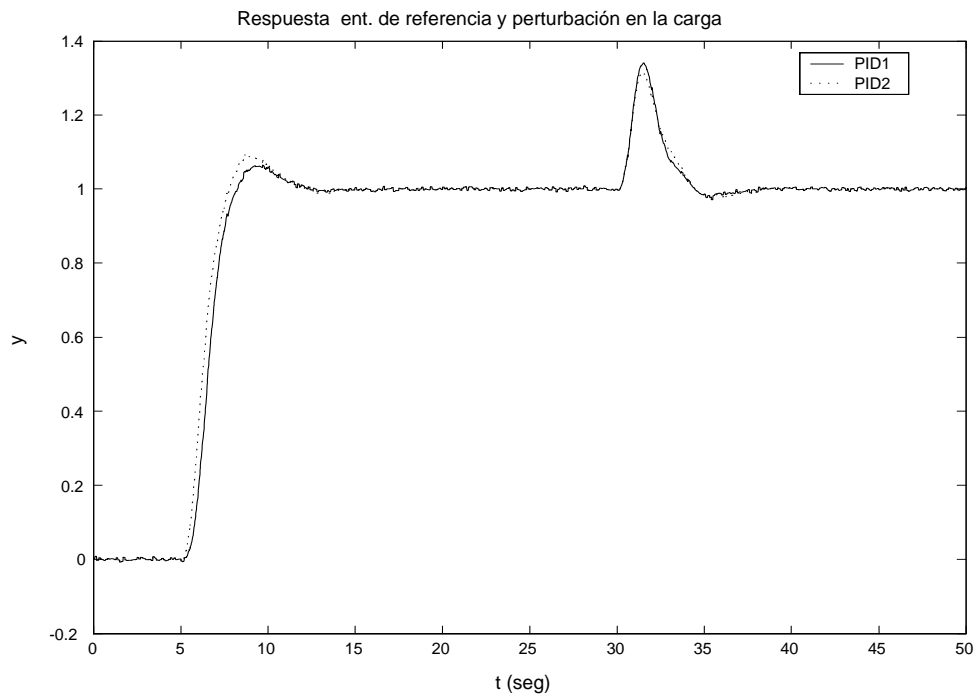


Figura 5-15: Planta 6

A continuación se presentan los índices de desempeño de la Planta 6.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	1.775	11.17	6561	1.31
PID2	1.570	10.06	7000	1.34

Cuadro 5.9: Índices de desempeño de la Planta 6

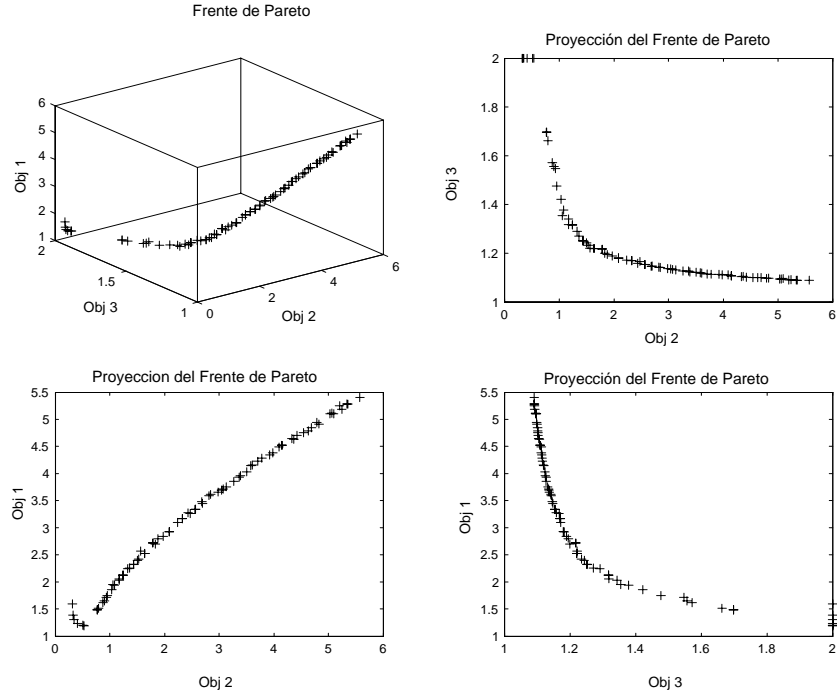


Figura 5-16: Planta 7

La Figura 5-16 muestra el frente de Pareto del problema de optimización multiobjetivo, utilizado para determinar los parámetros del controlador PID, aplicado al control de la Planta 7. La Planta 7 esta representada por un modelo con retardo puro (Tabla 5.1), el cual fue aproximado por un modelo de Padé de primer orden, esto conlleva a un modelo de fase no-mínima. En la Figura 5-17, se puede ver la respuesta de salida de la Planta 7 ante una entrada de referencia y, la adición de una señal escalón a la entrada de control, la cual simula una perturbación en la carga. En la Tabla 5.10 se reportan los valores de los índices de desempeño de los controladores PID1 y PID2.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	2.61	19.52	3478	2.757
PID2	2.79	19.33	4646	2.283

Cuadro 5.10: Índices de desempeño de la Planta 7

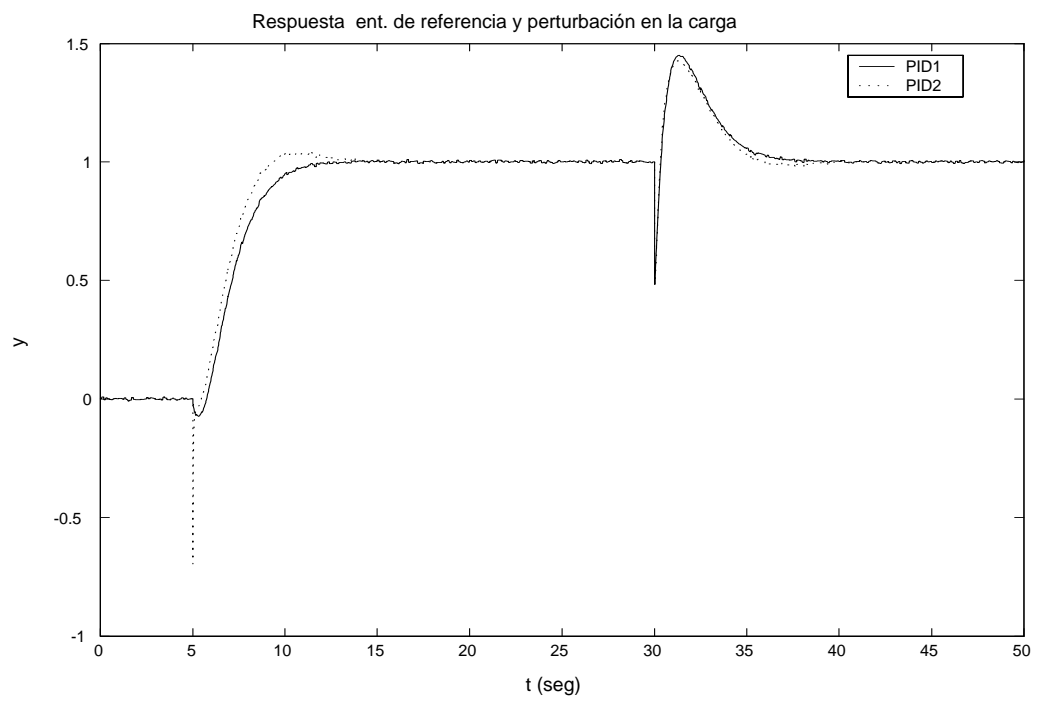


Figura 5-17: Planta 7

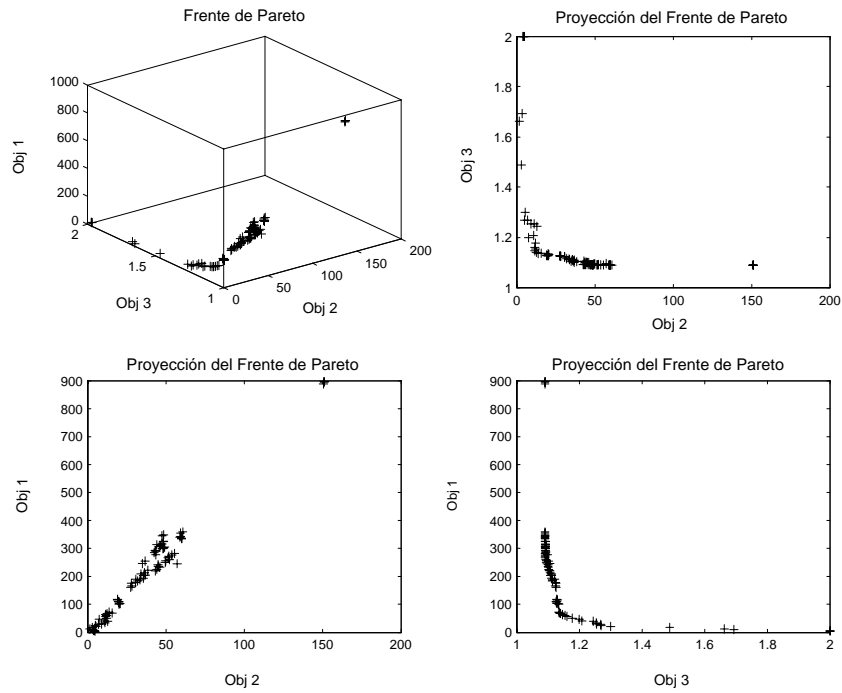


Figura 5-18: Planta 8

En la Figura 5-18 se muestra el frente de Pareto concerniente a la Planta 8, ésta corresponde a un modelo con integrador puro más retardo (Tabla 5.1). En la Figura 5-19, se puede ver la respuesta de salida de la Planta 8 con los controladores PID1 y PID2, ante una entrada de referencia y, la adición de una señal escalón a la entrada de control, aplicada en un tiempo $t = 50s$. Los parámetros del controlador PID1 fueron seleccionados del conjunto de óptimos de Pareto. En la Tabla 5.11 se reportan los valores de los índices de desempeño de los controladores PID1 y PID2.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	3.616	34.92	1219	21.32
PID2	5.378	58.22	1624	25.76

Cuadro 5.11: Índices de desempeño de la Planta 8

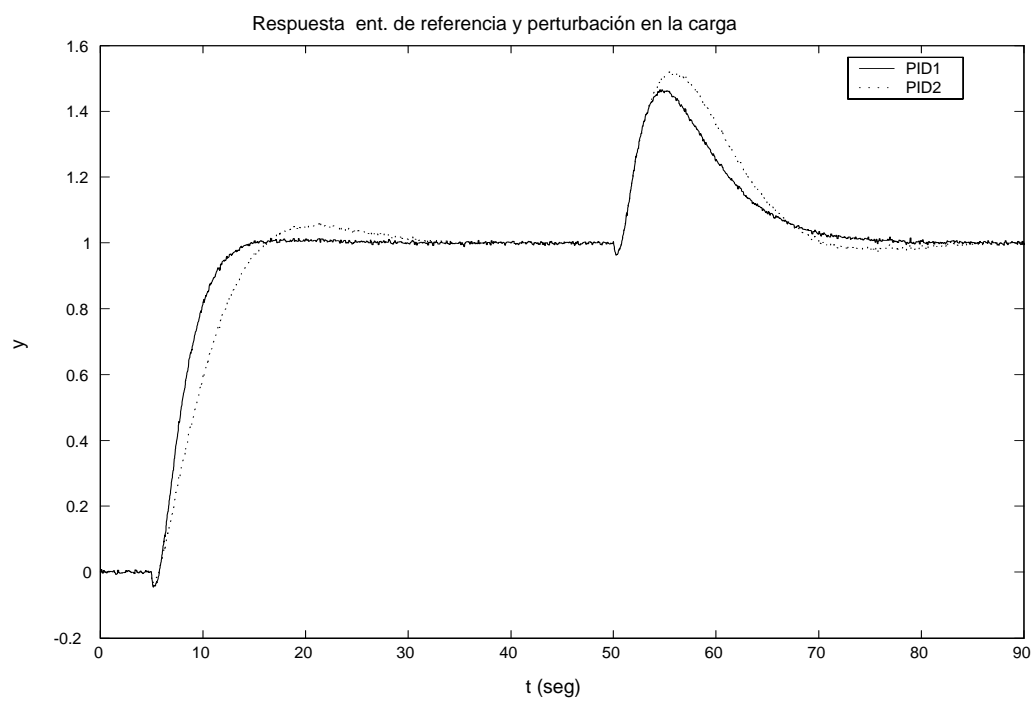


Figura 5-19: Planta 8

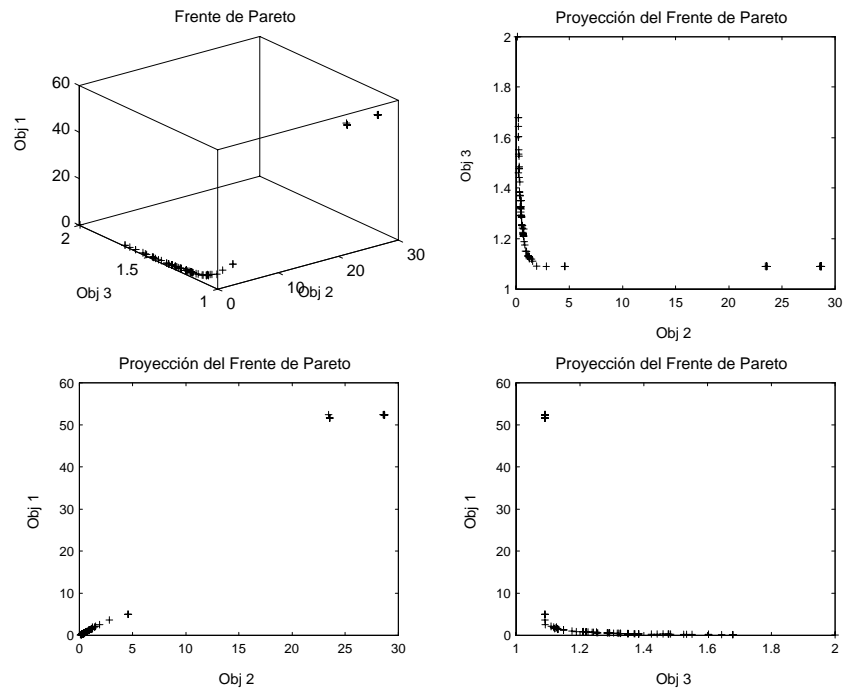


Figura 5-20: Planta 10

En la Figura 5-20 se presenta el frente de Pareto, correspondiente al problema de optimización multiobjetivo utilizado para sintonizar al controlador PID de la Planta 10. En el caso de la Planta 10 el desempeño de los controladores PID1 y PID2 es muy parecido. Ésto se puede apreciar en la Figura 5-21, donde es necesario disponer de información adicional para realizar una comparación sobre el desempeño de los dos controladores. En la Tabla 5.12 se reportan los valores de los índices de desempeño de los controladores PID1 y PID2.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	0.5603	3.072	158.1	0.2095
PID2	0.5548	5.114	275.5	0.3395

Cuadro 5.12: Índices de desempeño de la Planta 10

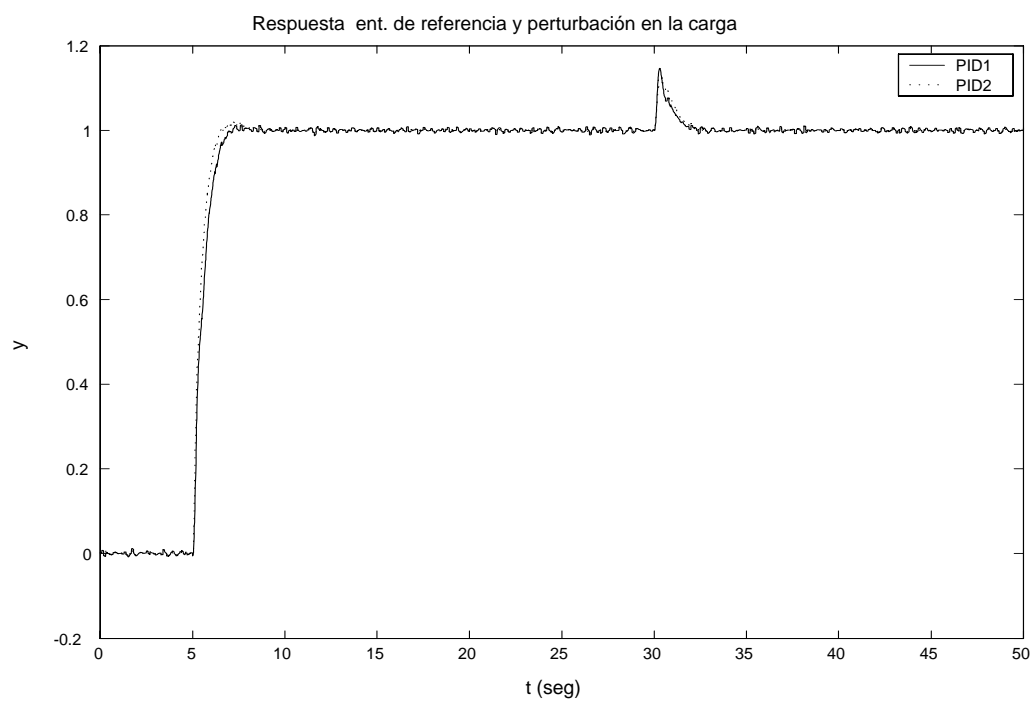


Figura 5-21: Planta 10

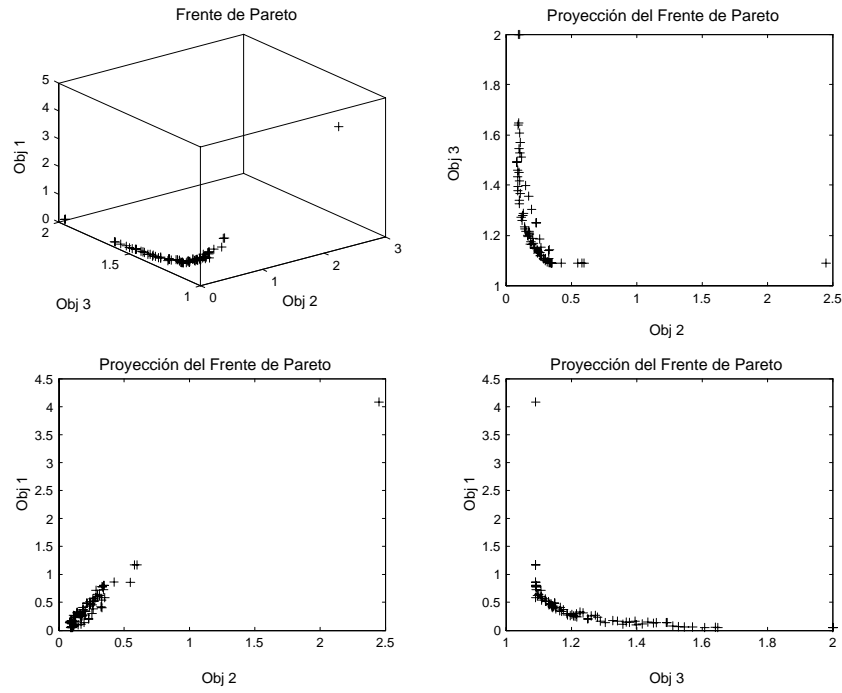


Figura 5-22: Planta 11

En la Figura 5-22 se muestra el frente de Pareto correspondiente a la Planta 11. En la Figura 5-23 se puede apreciar el comportamiento de los controladores PID1 y PID2, usados en la Planta 11. En la misma figura, se puede observar que: el controlador PID1 atenúa más a la señal de perturbación de carga que el controlador PID2. Ésto, puede ser corroborado por el criterio de evaluación correspondiente ($\int_0^\infty |e(t)| dt$), mostrado en la Tabla 5.13. La respuesta del sistema al escalón, aplicado en la entrada de referencia, es muy parecida para los dos controladores, siendo necesario recurrir a la Tabla 5.13, para identificar al controlador con mejor desempeño. Los parámetros del controlador PID1 fueron obtenidos del conjunto de soluciones óptimas de Pareto.

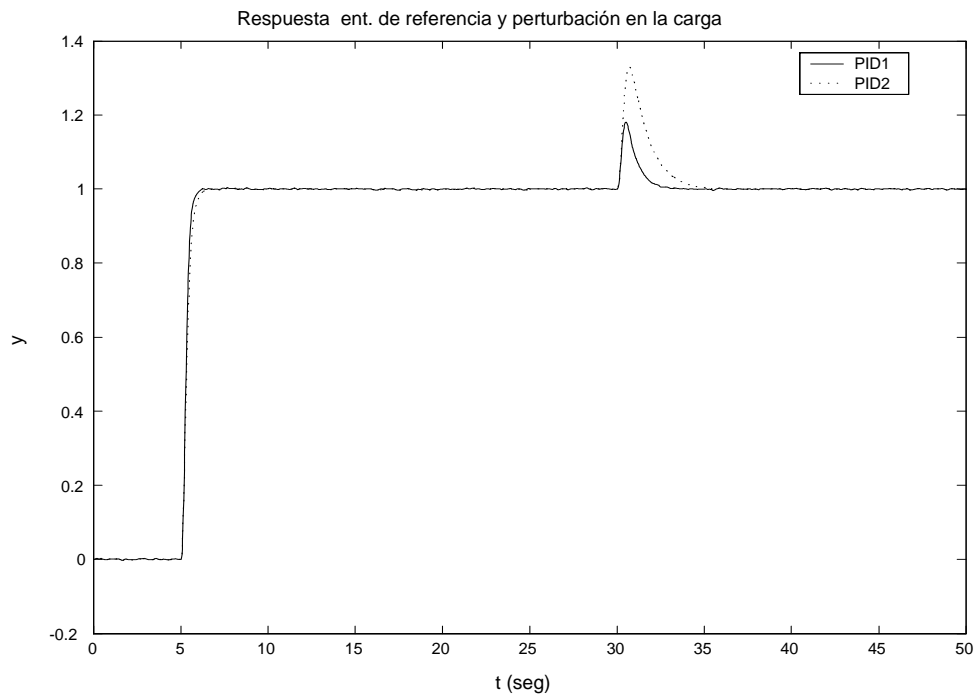


Figura 5-23: Planta 11

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	0.376	2.756	119.0	0.2133
PID2	0.448	3.400	133.4	0.5712

Cuadro 5.13: Índices de desempeño de la Planta 11

A continuación se presentan los resultados de la Planta 12

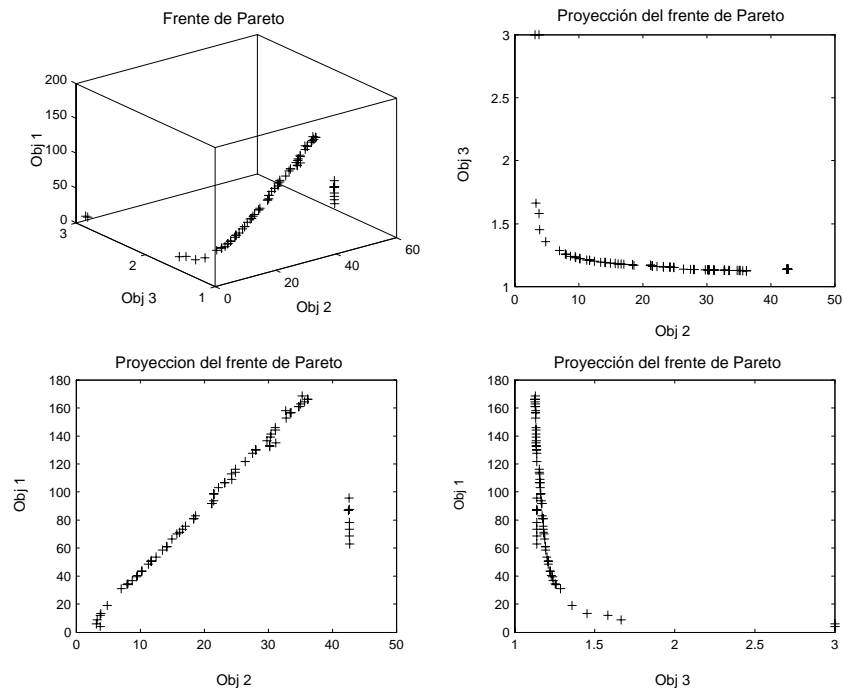


Figura 5-24: Planta 12

La Planta 12, contenida en la Tabla 5.1, tiene dos polos oscilatorios ($G_{12}(s) = \frac{9}{(s+1)(s^2+as+9)}$). La Figura 5-25 muestra el desempeño de los controladores PID1 y PID2 aplicados a esa planta cuando se hace $a = 0$. En la Tabla 5.14 se reportan los valores de los índices de desempeño de los controladores PID1 y PID2. La Figura 5-24 muestra el frente de Pareto correspondiente a la Planta 12.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	4.757	53.94	2758	7.51
PID2	5.371	59.17	1806	8.62

Cuadro 5.14: Índices de desempeño de la Planta 12

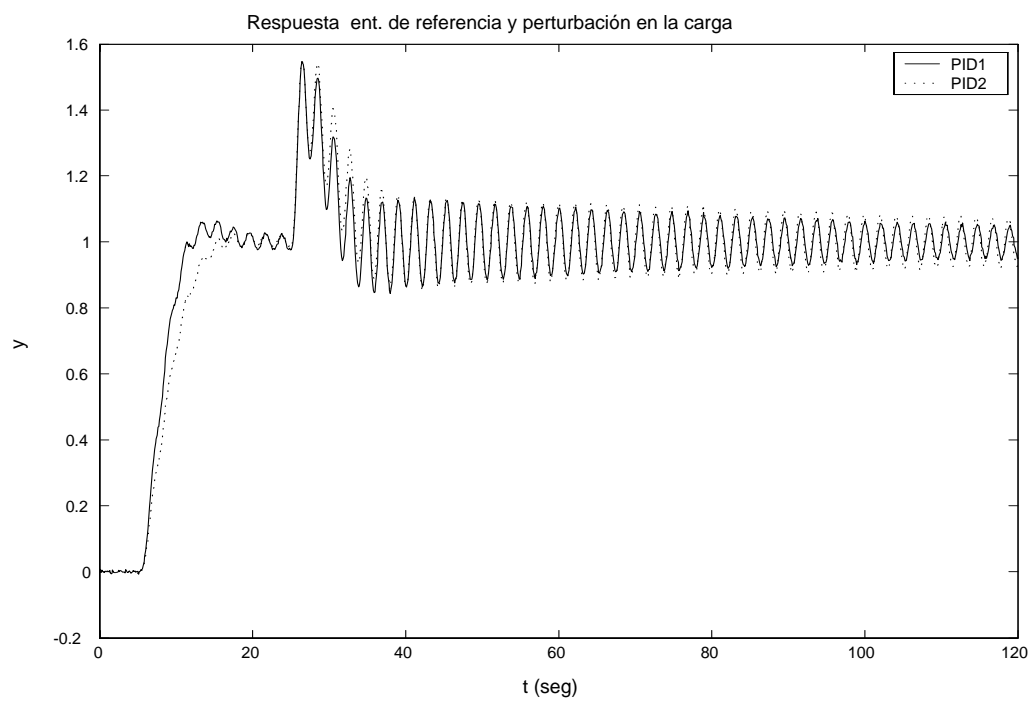


Figura 5-25: Planta 12

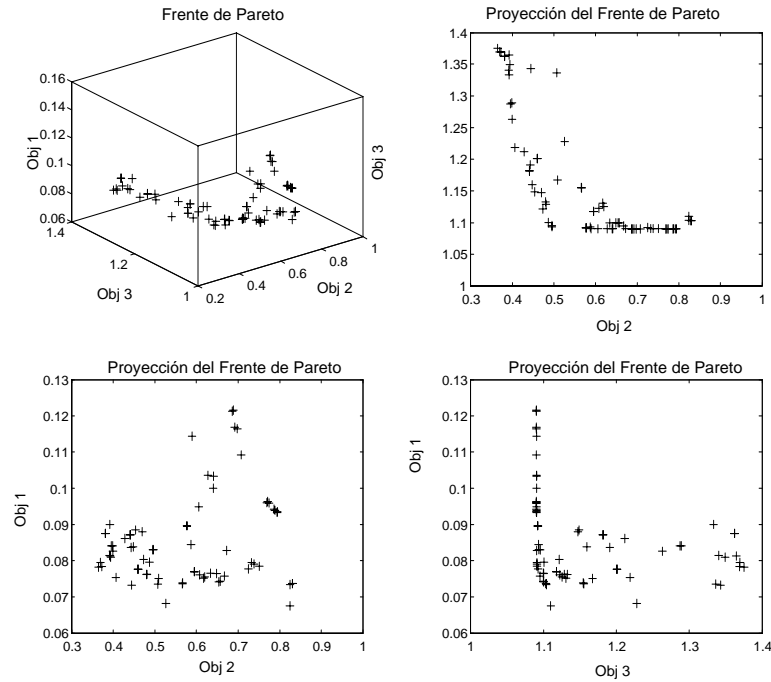


Figura 5-26: Planta 13

La Figura 5-26 contiene el frente de Pareto del problema de optimización multiobjetivo, aplicado a la selección del controlador PID, usado con la Planta 13. El conjunto de parámetros elegido se encuentra en la Tabla 5.3 y la respuesta temporal del sistema ante la entrada de referencia y perturbación de carga, es mostrada en la Figura 5-27. En la Tabla 5.15 se reportan los valores de los índices de desempeño de los controladores PID1 y PID2.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	0.9321	5.354	936	0.877
PID2	0.9337	5.249	333	0.022

Cuadro 5.15: Índices de desempeño de la Planta 13

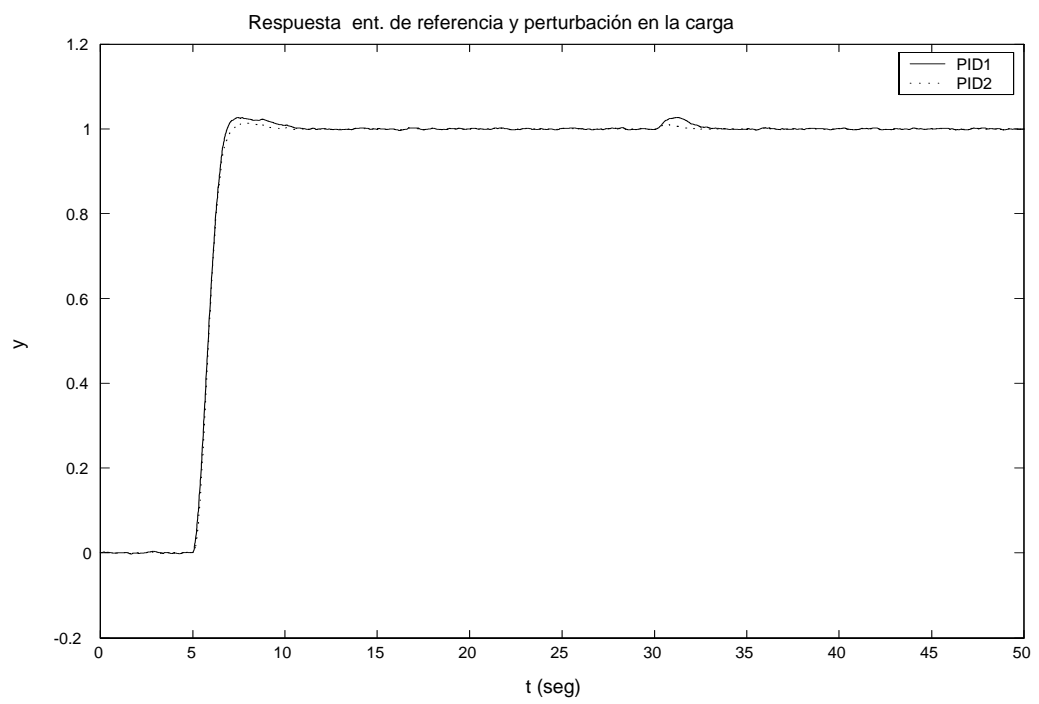


Figura 5-27: Planta 13

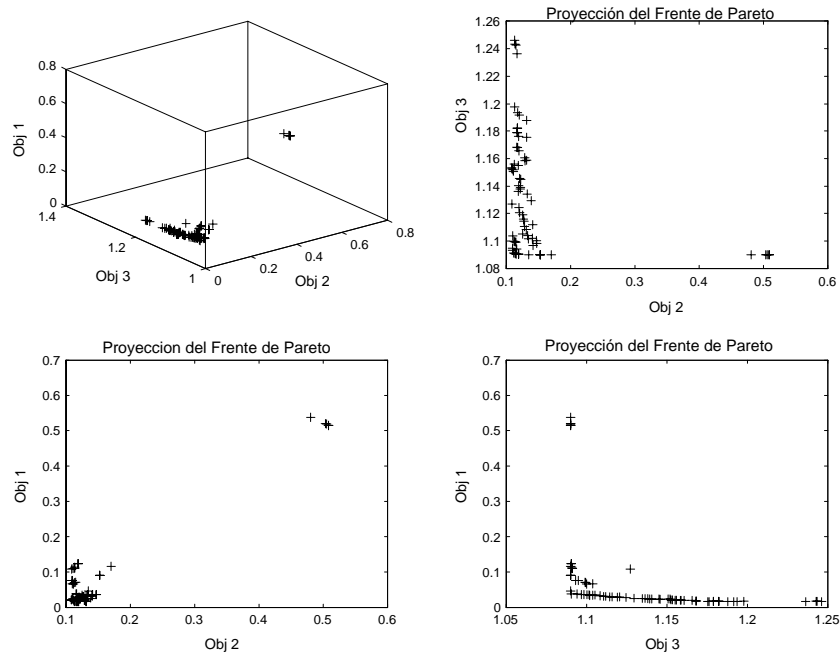


Figura 5-28: Planta 14

Finalmente, se presentan las figuras correspondientes a la Planta 14, tratada en este capítulo. La Figura 5-28 presenta el frente de Pareto del Problema de optimización, propuesto para sintonizar al controlador PID usado en esta planta. La Figura 5-29 presenta la respuesta temporal del sistema de control, Esquema 5-1, para los controladores PID1 y PID2. En la Tabla 5.15 se reportan los valores de los índices de desempeño de los controladores PID1 y PID2.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID1	0.3763	1.991	655.7	0.02129
PID2	0.5238	5.102	139.0	0.2118

Cuadro 5.16: Índices de desempeño de la Planta 14

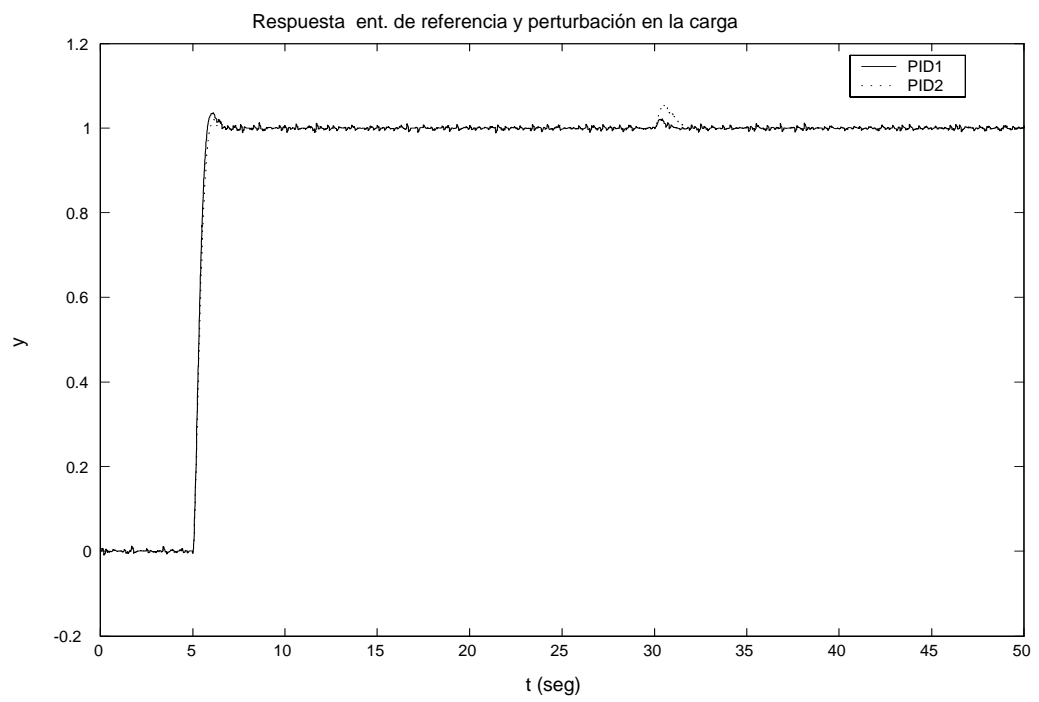


Figura 5-29: Planta 14

5.3. Conclusiones

Como se puede observar en las figuras que muestran las señales de salida del proceso para el conjunto de plantas tratadas en este capítulo, el controlador PID serie de dos grados de libertad representa una excelente opción para el control de procesos, ya que los modelos propuestos en [4] representan una colección variada de plantas. También es relevante mencionar que la estrategia de sintonización usada, tratada como un problema de optimización donde se minimiza una serie de funciones objetivo sujeto a restricciones mediante un algoritmo genético multiobjetivo (en este caso el NSGA-II), es fácil de aplicar y sólo podría tener el inconveniente del tiempo de ejecución del algoritmo genético, que para efecto de la sintonización de los controladores obtenidos, con el número de generaciones y el tamaño de la población mencionados, requiere de aproximadamente 90 minutos en promedio por cada controlador.

Capítulo 6

Sintonización de Controladores PID de Plantas Seleccionadas

6.1. Introducción

En el capítulo anterior se aplicó el procedimiento de sintonización de controladores PID mediante un algoritmo genético multiobjetivo (NSGA-II) a 13 plantas de prueba. Para el caso anterior ya se tenía conocimiento del valor de los parámetros, ya que éstos son presentados en la tesis doctoral de Herreros [36]. Ahora se presentan tres plantas con un grado de dificultad mayor a las del capítulo anterior, tomadas del artículo “*Obtaining controller parameter for a new PI-Pd Smith predictor using autotuning*” de Ibrahim Kaya [53]. Se aplica el mismo procedimiento, es decir el mismo controlador PID comercial de dos grados de libertad y los mismo criterios de desempeño. En la Tabla 6.1 se presentan las plantas tratadas en este capítulo.

$G_1(s) = \frac{4e^{-14.69s}}{(21.625s+1)(11.34s+1)}$
$G_2(s) = \frac{e^{-6.75s}}{s(10s+1)}$
$G_3(s) = \frac{4e^{-2s}}{(4s-1)}$

Cuadro 6.1: Plantas seleccionadas

PID	Ti	Tf	Td	N	k	b	c	Obj 1	Obj 2	Obj3
PID1	32.54	0.0068	9.629	13.66	0.4106	0.944	0.746	83.90	59.32	1.89
PID 2	43.42	0.0005	9.854	117.0	0.1176	0.728	0.779	379.2	20.51	1.95
PID 3	21.82	0.00024	0.892	26.78	0.5144	0.330	0.234	43.28	3.71	2.00

Cuadro 6.2: Parámetros contr. plantas selecc.

PID	T_i	T_f	T_d	N	k	b	c
	0-50	0-2	0-20	0-120	0-50	0-1	0-1

Cuadro 6.3: Rango de los parámetros plantas selecc.

La planta $G_1(s)$ es de segundo orden más un retardo de tiempo, es la más fácil de controlar. $G_2(s)$ es un modelo de primer orden con integrador más un retardo de tiempo relativo grande. La planta $G_3(s)$ es un modelo de primer grado, inestable y además presenta un retardo de tiempo.

6.2. Evaluación de resultados

En la Tabla 6.2 se muestran los valores de los siete parámetros y los tres objetivos que se obtuvieron al aplicar el algoritmo genético multiobjetivo NSGA-II a las tres plantas de prueba seleccionadas. Los resultados presentados en la Tabla 6.2 son los que se utilizaron para sintonizar los controladores PID usados en las plantas $G_1(s)$ - $G_3(s)$. Los rangos de los parámetros que se utilizaron para sintonizar los controladores están dados en la Tabla 6.3.

Como en el capítulo anterior, también se muestran los frentes de Pareto de las tres plantas ($G_1(s) - G_3(s)$), los cuales se dibujaron tomando los valores de las tres funciones objetivo, contenidos en *plot.out*, que es uno de los archivos de salida del NSGA-II. En estas gráficas de los frentes de Pareto se puede apreciar el resultado del proceso de optimización multiobjetivo en el espacio de las funciones objetivo. De las gráficas del frente de Pareto y sus proyecciones, el Objetivo 1 (Obj 1) corresponde a la función objetivo que minimiza $\|T_{yt}(s)\frac{1}{s}\|_1$. El segundo Objetivo (Obj 2) minimiza a $\|\frac{d}{ds}T_{er}(s)\frac{1}{s}\|_2$ y el tercer

Objetivo minimiza a $\|T_{yd}(s)\|_{\infty}$. También se presentan las gráficas de la respuesta en el tiempo de cada una de las plantas controladas (utilizando SimulinkTM de MATLABTM), donde se puede apreciar el cambio de la señal de salida (y) ante una entrada de tipo escalón y el efecto de una perturbación (escalón de valor igual a -0.2 para $G_1(s)$ y -0.1 para $G_2(s)$ y $G_3(s)$) aplicada en la señal de control u . También se adicionó una señal alta frecuencia en la señal de salida. En este capítulo se adicionaron las gráficas de Nyquist, del problema de control en lazo abierto de las tres plantas, donde se pueden observar de forma gráfica los valores de M_s , representados por el objetivo 3 (Obj3). Las señales bajo la denominación PID corresponden a los controladores obtenidos, las denominadas Kaya a los controladores sintonizados por el método de Kaya [53] y Skog., en algunas casos (cuando el tipo de sistema lo permite), a los controladores diseñados con el método de Skogestad [71]. Las gráficas correspondientes a las plantas $G_1(s)$ (Planta Seleccionada 1), $G_2(s)$ (Planta Seleccionada 2) y $G_3(s)$ (Planta Seleccionada 3), se muestran en las Figuras (6-1-6-9).

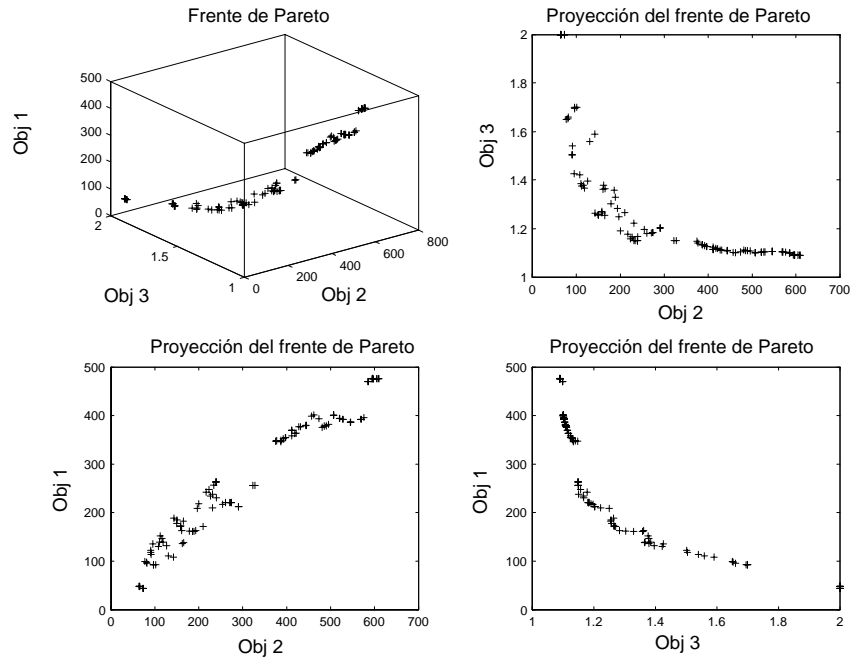


Figura 6-1: Planta Seleccionada 1

En la Figura 6-1 se muestra el frente de Pareto del problema de optimización multiobjetivo, utilizado para determinar los parámetros del controlador PID, aplicado al control de la Planta Seleccionada 1. Como en el capítulo anterior, se seleccionó también una solución del conjunto de óptimos de Pareto, presentándose ésta en la Tabla 6.2 (controlador PID1). Con estos parámetros se simuló el problema de control utilizado en este capítulo, usado también en el capítulo anterior, Figura 5-1, y el resultado correspondiente se presenta gráficamente en la Figura 6-2. En esa figura, se presentan también los resultados de los controladores sintonizados por Herreros y Skogestad. Como información complementaria se muestra también la curva de Nyquist de la función de transferencia de lazo abierto del problema de control correspondiente a esta planta, Figura 6-3, teniendo un Margen de Fase de 55.5° y un Margen de Ganancia de 7.57 dB. En la Tabla 6.4 se muestran los valores de los índices de desempeño de los tres controladores, con el objeto de evaluar cuantitativamente el desempeño del controlador PID con respecto

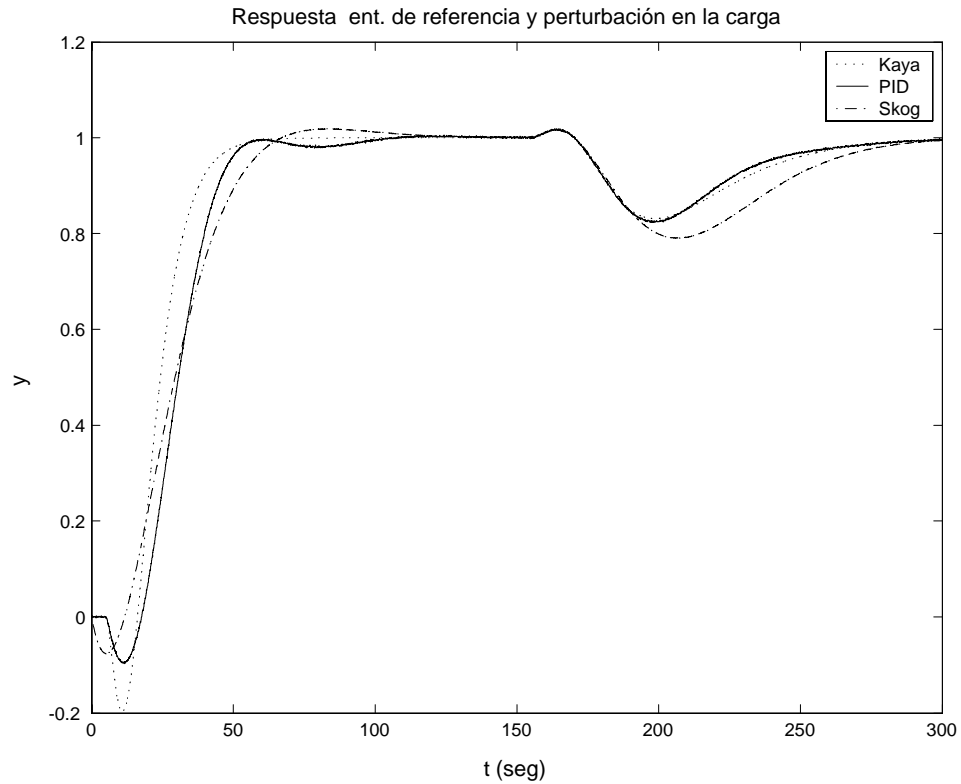


Figura 6-2: Planta Seleccionada 1

a los otros dos controladores. Analizando los resultados se puede observar que para los cambios ante la entrada de referencia, el mejor resultado es proporcionado por Kaya. En cuanto a la perturbación de carga, aplicada en un tiempo $t = 150s$, el controlador PID presenta un tiempo de establecimiento menor con respecto a los otros dos controladores. Esto se puede corroborar de manera gráfica en la Figura 6-2.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID	27.41	712.0	1103.0	83.54
Kaya	22.72	394.6	984.60	92.8
Skog.	32.72	653.6	1267.0	127.4

Cuadro 6.4: Índices de desempeño de la Planta Seleccionada 1

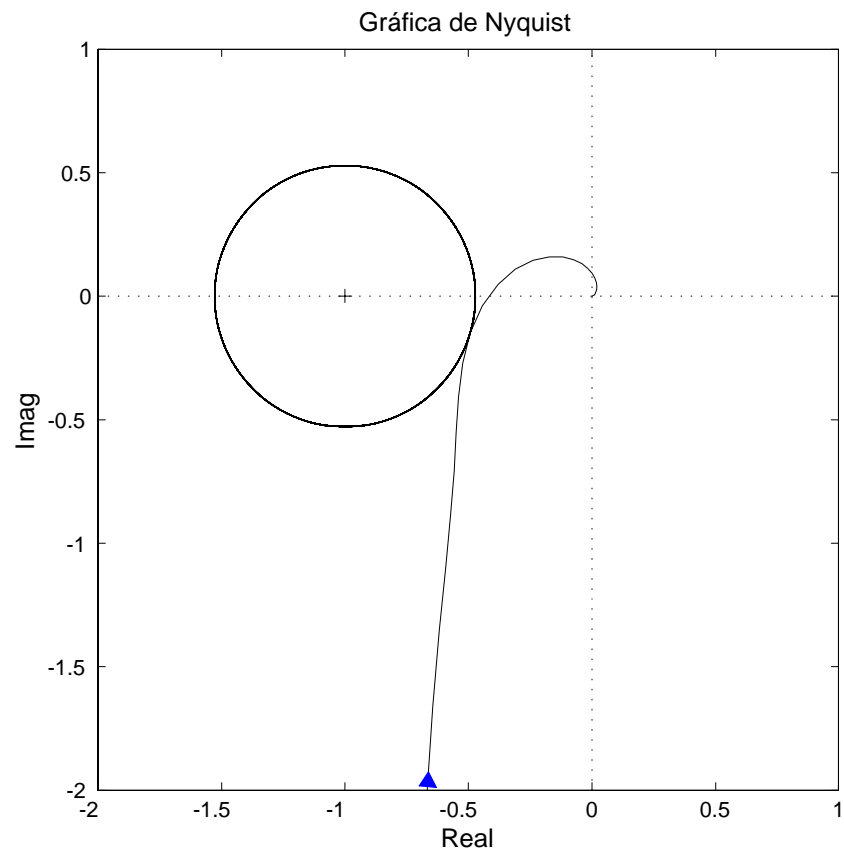


Figura 6-3: Planta seleccionada 1

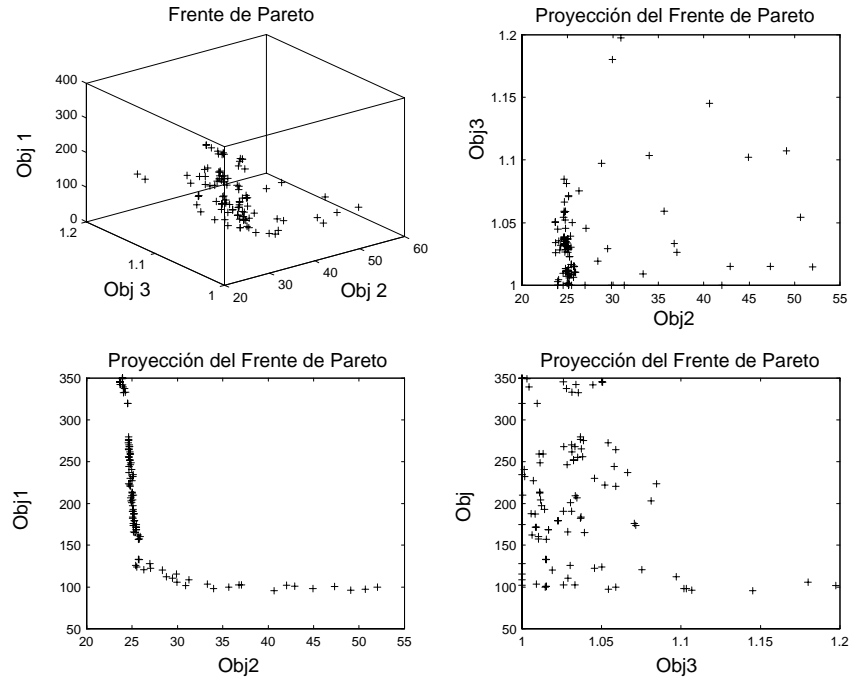


Figura 6-4: Planta Seleccionada 2

La Figura 6-4 muestra el frente de Pareto correspondiente al problema de optimización multiobjetivo utilizado para sintonizar al controlador PID de la Planta Seleccionada 2. Como en el caso anterior, se seleccionó también una de las soluciones del conjunto de óptimos de Pareto, la cual es presentada en la Tabla 6.2 (controlador PID2). En la Figura 6-6 se muestra la curva de Nyquist, teniéndose un Margen de Fase de 42.8° y un Margen de Ganancia de 7.59 dB. En la Figura 6-5 se muestran las señales de salida de la Planta Seleccionada 2 correspondientes a los controladores PID y Kaya donde se puede observar que el mejor desempeño corresponde al controlador de Kaya, tanto en la respuesta a la entrada de referencia como a la perturbación de carga (aplicada en un tiempo $t = 150s$). En Tabla 6.5, se muestran los valores de los índices de desempeño de la Planta seleccionada 2 .

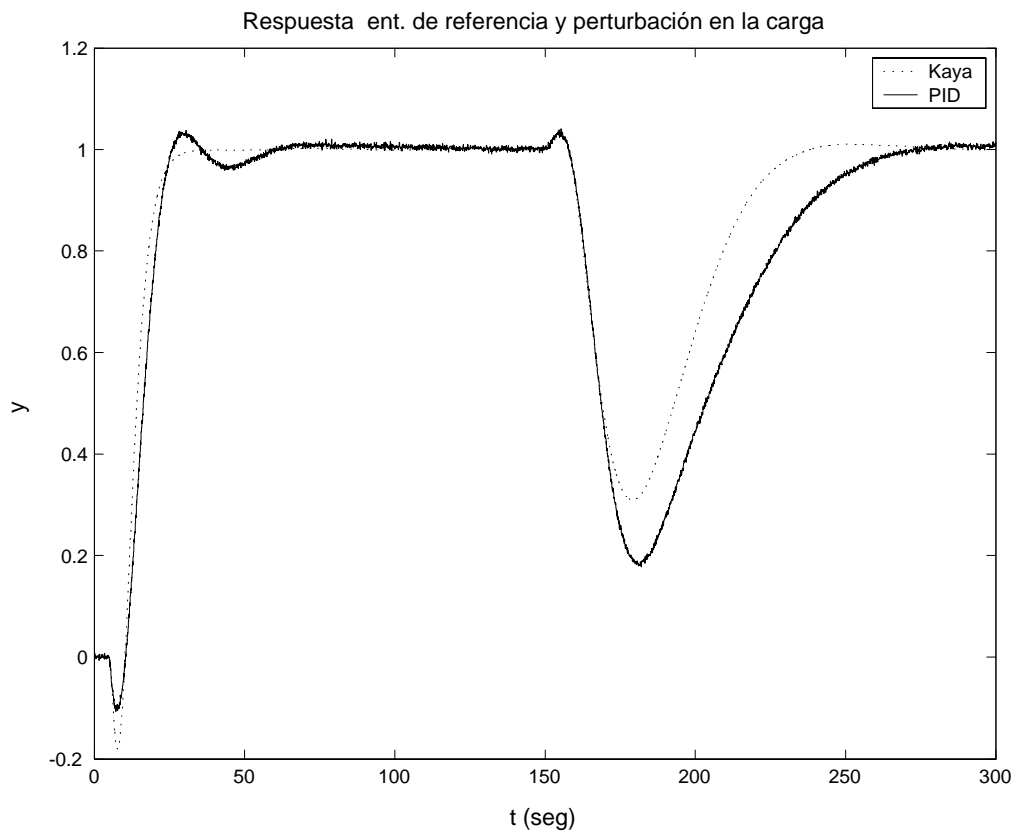


Figura 6-5: Planta Seleccionada 2

A continuación se presentan los valores de los índices de desempeño de la Planta Seleccionada 2.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	PI	$\int_0^\infty e(t) dt$
PID	13.2	204.8	514	376.9
Kaya	10.7	114.8	459	254.4

Cuadro 6.5: Índices de desempeño de la Planta Seleccionada 2

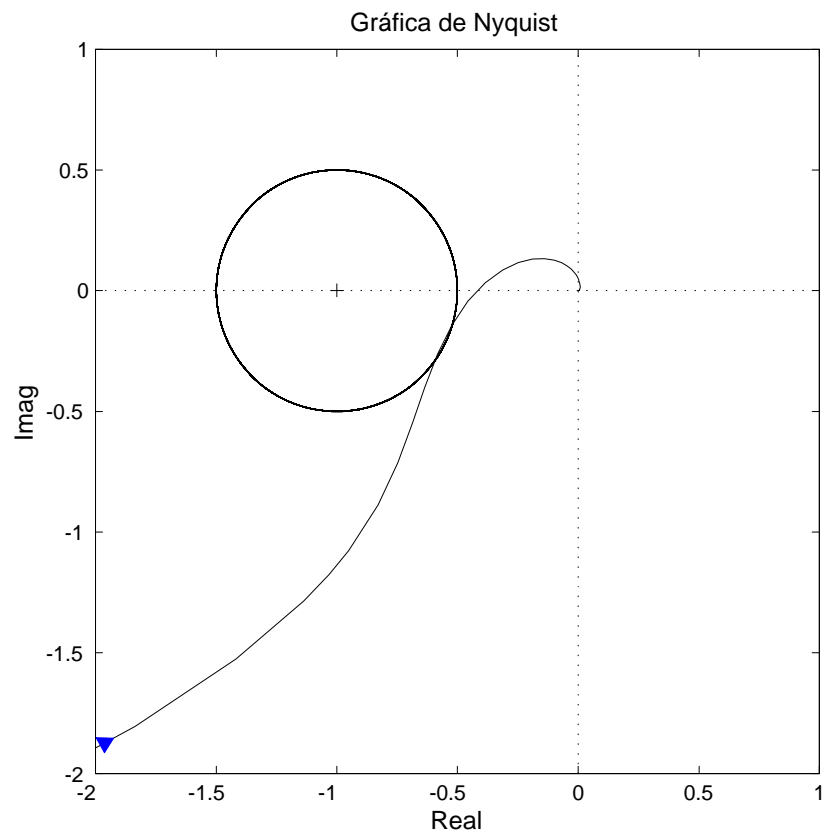


Figura 6-6: Planta Seleccionada 2

A continuación se presentan los resultados de la Planta Seleccionada 3.

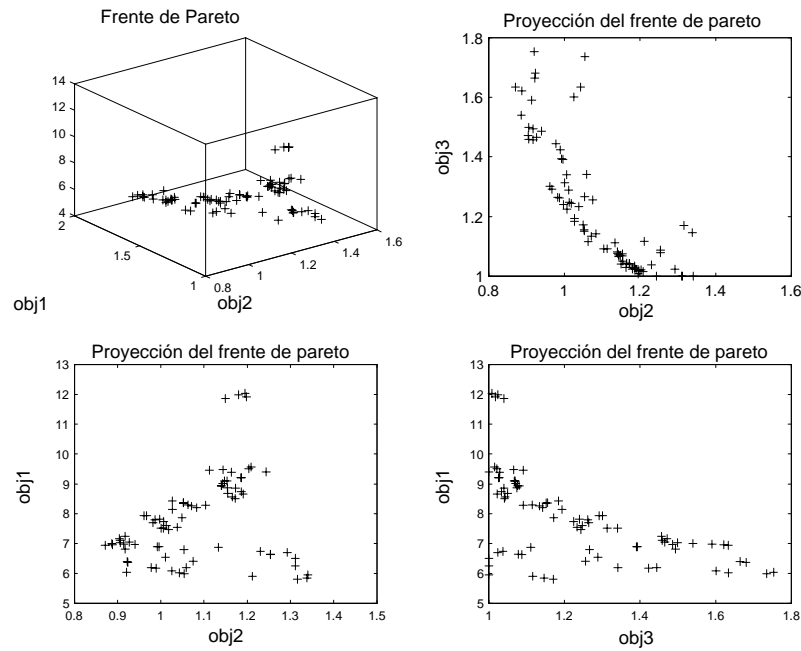


Figura 6-7: Planta Seleccionada 3

En la Figura 6-7 se muestra el frente de Pareto concerniente a la Planta Seleccionada 3. En la Figura 6-8 se presentan las gráficas que muestran el desempeño de los controladores PID y Kaya. PID utiliza los parámetros de la solución elegida del conjunto de óptimos de Pareto reportada en la Tabla 5.3 como *PID3*. En la Figura 6-9 se muestra la curva de Nyquist, teniéndose un Margen de Fase de 29.5° y en este caso, el problema (la curva de Nyquist cruza el eje real negativo del plano complejo 2 veces) presenta dos valores para el Margen de Ganancia 6.33 dB y -5.42 dB. En la Tabla 6.6 se presentan los valores de los índices de desempeño de este problema.

Contr.	Desempeño de los controladores			
	Entrada de referencia		Perturbación en la carga	
	$\int_0^\infty e(t) dt$	$\int_0^\infty t e(t) dt$	<i>PI</i>	$\int_0^\infty e(t) dt$
PID	4.14	32.72	145.1	43.38
Kaya	4.00	30.00	150.0	44.72

Cuadro 6.6: Índices de desempeño de la Planta Seleccionada 3

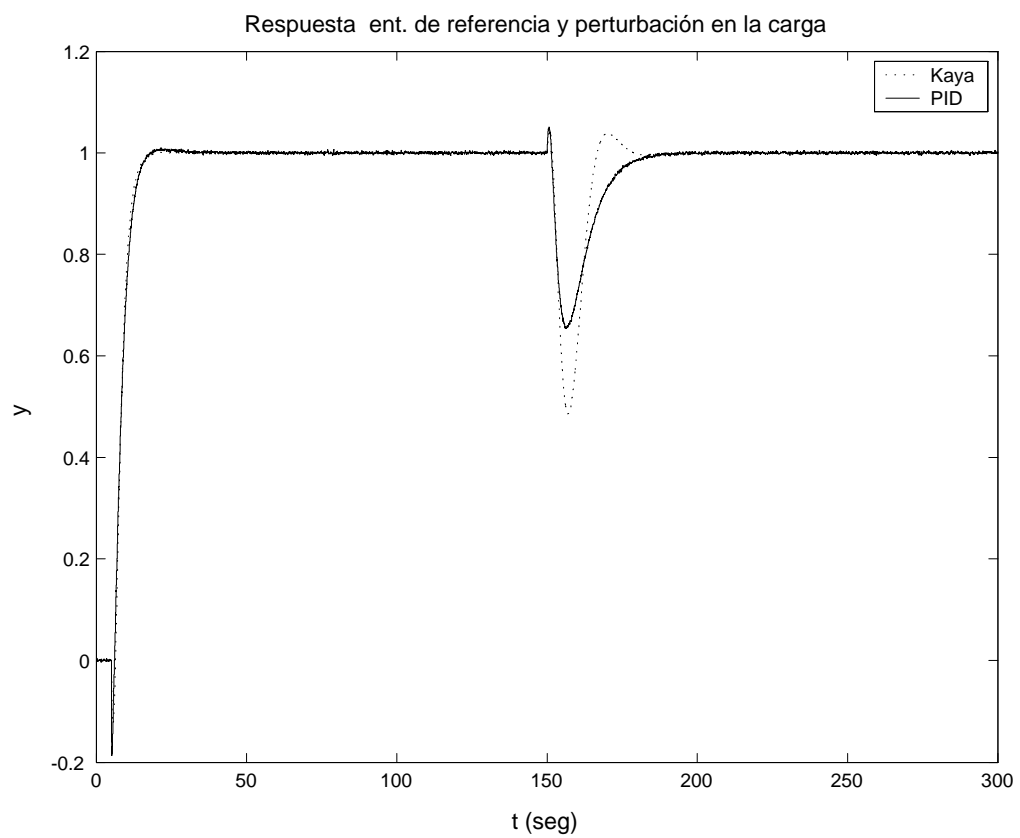


Figura 6-8: Planta Seleccionada 3

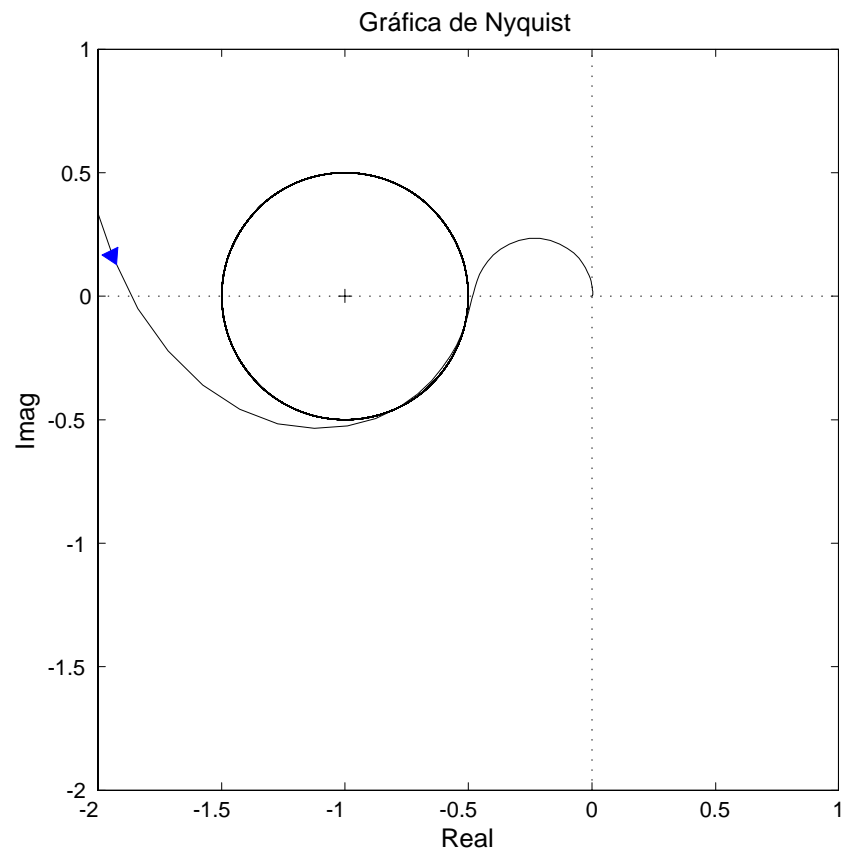


Figura 6-9: Planta Seleccionada 3

6.3. Análisis de resultados

Como se puede apreciar en las Figuras 6-2, 6-5 y 6-8, la respuesta temporal obtenida usando un controlador PID, tiene deficiencias para controlar procesos con retardos de tiempo grandes como los presentados en el artículo de Kaya [53]. Estas deficiencias también se pueden hacer patentes por medio de las Tablas 6.4 6.5 y 6.6; donde se muestran los valores de los índices desempeño, que evalúan la respuesta de la salida del proceso, ante una entrada de referencia y perturbaciones de carga en cada una de las tres Plantas Seleccionadas. Esto se debe principalmente al retraso de fase adicional, contribuido por el retardo de tiempo que tiende a desestabilizar al sistema en lazo cerrado [53]. El problema de estabilidad se puede solucionar decrementando la ganancia del controlador, pero esto haría la respuesta del sistema muy lenta. Lo anterior sin embargo no es una limitación de la estrategia utilizada para sintonizar los controladores PID (optimización multiobjetivo), si no más bien debido a las propias limitaciones de la estructura de dicho controlador.

Capítulo 7

Conclusiones y Perspectivas de Trabajos Futuros

7.1. Discusiones

Se seleccionó el NSGA-II por ser una herramienta de dominio público y citada en múltiples reportes sobre investigación evolutiva multiobjetivo. Es un algoritmo robusto, de aplicación general, que puede ejecutarse en diferentes plataformas, aunque fue totalmente probado por sus autores usando el sistema operativo Linux. En nuestro caso se prefirió correrlo bajo el sistema operativo Windows-XP de Microsoft para que pudiera ejecutarse junto con MATLABTM en el mismo entorno.

En cuanto a la aplicación de un algoritmo genético simple a un problema de control H_2/H_∞ se obtuvieron resultados satisfactorios, sobre todo cuando se aplicó la restricción que permite acotar la señal de control.

Comparando los resultados obtenidos de las 13 plantas de prueba con los obtenidos por Herreros, resultan muy buenos, sobre todo considerando que usamos un algoritmo de aplicación general. Además, se filtraron los resultados del frente de Pareto para que sólo dé los mejores cuatro resultados de acuerdo a los siguientes dos criterios: mejor respuesta a cambios de señal de referencia (escalón) y mejor atenuación a perturbaciones en la señal

de control.

El método de Skogestad usado para sintonizar controladores PID es muy sencillo, sin embargo, permite encontrar controladores muy robustos, su desempeño es menor al de los controladores sintonizados con la técnica desarrollada en este trabajo. La ventaja del método de Skogestad es su sencillez. Este método no se aplica a plantas inestables o procesos oscilatorios y el método propuesto si funciona bien en estos casos.

En el caso de las plantas propuestas por Kaya, nuestros resultados no son tan buenos, sin embargo lo interesante es mostrar el amplio rango de aplicación que tienen los controladores PID y lo fácil que es su sintonización, aplicando los algoritmos genéticos multiobjetivo.

Por otro lado, respecto de la convergencia del algoritmo genético evolutivo NSGA-II, se sabe que en la práctica no hay manera de saber si se ha llegado o no al verdadero frente de Pareto (eso aplica a cualquier algoritmo evolutivo multiobjetivo). Un criterio posible de paro es cuando ya no se producen soluciones que dominen a las mejores que se tienen hasta el momento. Si no hay mejoras después de un cierto número de iteraciones se puede presuponer que ya convergimos pero obviamente no hay ninguna garantía de eso. Esa es la mayor debilidad de las heurísticas, pues cuando se detienen no hay garantía de que hayan alcanzado el óptimo.

7.2. Conclusiones

Los resultados que se obtuvieron en las diferentes plantas de prueba, concuerdan con la opinión de muchos expertos en control automático, acerca de la versatilidad de los controladores PID, sobre todo en la aplicación de procesos industriales. Aun más si se utiliza una variedad del controlador PID convencional como el controlador PID ISA de dos grados de libertad. También se puede constatar la sencillez del problema de control planteado como un problema de optimización multiobjetivo, el cual minimiza una serie de normas que representan los índices de desempeño y las restricciones del problema de

control para ser resuelto numéricamente por medio de un algoritmo genético multiobjetivo (NSGA-II).

- La estructura de dos grados de libertad permite desacoplar el problema de seguimiento de referencia y el de perturbación en la señal de control. Se puede concluir que este objetivo se cumple para todas las plantas de prueba utilizadas en esta tesis.
- El resultado exitoso de un algoritmo genético multiobjetivo de aplicación general, en este caso el NSGA-II, a problemas de sintonizado de controladores PID.
- Este método de sintonización basado en el controlador PID comercial de dos grados de libertad tiene la ventaja de que se puede usar para una amplia gama de plantas, por ejemplo plantas con tiempos muertos, plantas con polos complejos, plantas inestables, etc.
- Uso del método propuesto por Kaya

La intención de mostrar esta propuesta moderna de sintonizado de un controlador, es realzar la versatilidad del controlador PID comercial tratado en esta tesis. Como se pudo apreciar en el Capítulo 6 los resultados que se obtuvieron con la aplicación del PID a las plantas propuesta por Kaya no son mejores que los de éste, aunque en el caso de la planta 2 el resultado es similar. Sin embargo como ya se mencionó en el Capítulo 3, el método propuesto por Kaya es especial para el tratamiento de tres procesos particulares, es decir se aplica a plantas que pueden ser modeladas como funciones de transferencia de segundo orden más retardo, de primer orden con integrador más retardo y de primer orden inestable más retardo. Por lo que no abarca tantos casos como el controlador PID comercial, como puede ser una planta con polos complejos.

- La automatización del proceso de búsqueda de las mejores soluciones del frente de Pareto, a través de la función *sel_pareto* ha demostrado ser una herramienta muy útil para determinar los mejores controladores PID propuestos en esta tesis. A

propósito se obtiene más de una solución para estar de acuerdo con el espíritu de la estrategia basada en optimización multiobjetivo, donde existe más de una solución óptima y su elección depende del usuario.

7.3. Perspectivas de trabajos futuros

- Abordar el problema de identificación con algoritmos genéticos multiobjetivo para plantas lineales y no-lineales SISO, simultáneamente con el problema de sintonización de controladores PID.
- El uso de algoritmos genéticos multiobjetivo a problemas de sintonización de PID's para plantas lineales SISO y MIMO en tiempo discreto.
- Comparar los resultados obtenidos en esta tesis con otros trabajos para sintonización de PID's basados en algoritmos de “inteligencia artificial”, como pueden ser: control difuso, redes neurales, etc.. En particular, es interesante comparar con otros esquemas de optimización evolutiva, como algoritmos meméticos y algoritmos basados en el sistema inmunológico, etc.
- La aplicación de algoritmos genéticos multiobjetivo a problemas de sintonización de PID's para plantas lineales y no-lineales MIMO.
- El desarrollo de esquemas basados en algoritmos genéticos multiobjetivo para problemas de sintonización de controladores con estructura más general y compleja que los PID's, a plantas lineales y no-lineales MIMO.
- El desarrollo de metodologías híbridas, las cuales usen diferentes técnicas de sintonizado de manera integrada para plantas con estructura variable o fija y de tipo muy general (de dimensión finita o infinita, SISO o MIMO). Por ejemplo, usando algoritmos basados en agentes evolutivos cooperativos para coordinar simultánea-

mente otras técnicas como algoritmos genéticos multiobjetivo, control difuso, redes neurales, etc.

- Implantación de estos algoritmos genéticos multiobjetivo, en programas que se ejecuten en paralelo o en clusters, para mejorar la velocidad de convergencia.
- Investigar la posibilidad de implementar otro método de optimización para los parámetros del algoritmo genético multiobjetivo, con la idea de que se mejore el desempeño del algoritmo genético multiobjetivo de manera automática, es decir cerrando un lazo de retroalimentación entre el algoritmo genético y el método de optimización, donde este último juega el papel del controlador.

Bibliografía

- [1] K.J. Åström and T. Häggglund. *Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins*. Automatica, 20, pp. 645-651, 1984.
- [2] K.J. Åström and B. Wittenmark. *Computer Controlled System: Theory and Desing*. Second Edition, Prentice Hall, Englewood Cliffs, New Yersey, USA, 1990.
- [3] K.J. Åström and T. Häggglund. *PID Controllers: Theory, Design, and Tuning*. Second Edition, Instrument Society of America, Reserch Triangle Park, NC, 1995.
- [4] K. J. Åström, H. Panagopoulos and T. Häggglund. *Design of PI Controllers based on Non-Convex Optimization*. Automatica, Vol. 34, No. 5 pp. 585-601, 1998.
- [5] K.J. Åström and T. Häggglund. *The future of PID control*. Control Engineering Practice. no. 9 pp. 1163-1175, 2001.
- [6] D.P. Atherton and A.F. Boz. *Using standard forms for controllers design*. UKACC Int. Conference on Control'98, Swansea, UK, pp. 1066-1071, 1998.
- [7] T. Bäck . *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- [8] S. Bennett. *A History of Control Engineering 1800-1930*. London: Peter Peregrinuos, 1979.

- [9] S. Bennett. *A History of Control Engineering 1930-1955*. London: Peter Peregrinus, 1993.
- [10] Daniel U. Campos-Delgado and Kemin Zhou. *Mixed $\mathcal{L}_1/H_2/H_\infty$ control design: numerical optimization approaches*. International Journal of Control. Vol 76, no. 7, pp. 687-697, 2003.
- [11] F.E. Cellier. *Continuous System Modeling*. Springer-Verlag, New York, USA, 1991.
- [12] FlexTools(Ga) M 2.1 Flexible Intelligence Group, June 1995.
- [13] Bor-Sen Chen, Yu-Min Cheng, and Ching-Hsiang Lee. *A Genetic Approach to Mixed H_2/H_∞ Optimal PID Control*. IEEE Control Systems. pp. 51-59, October 1995.
- [14] Carlos A. Coello-Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.
- [15] Carlos A. Coello-Coello. *A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques*. Knowledge and Information Systems. An International Journal. 1(3). pp.269-308, 1999
- [16] Carlos A. Coello-Coello. *An Update Survey of Evolutionary-Based Multiobjective Optimization Techniques: State of the Art and Future Trends*. In 1999 Congress on Evolutionary Computation. Washington, D.C. IEEE Service Center. pp. 3-13, 1999.
- [17] Carlos A. Coello Coello, Hernández Aguirre and B.P. Buckles. *Evolutionary Multiobjective Design of Combinational Logic Circuits*. In Lohn, J., Stoica, A., Keymeulen, D. and Colombano, S., editors, Proceedings of the Second *NASA/DoD Workshop on Evolvable Hardware*, pp. 161-170, Los Alamitos, California. IEEE Computer Society, 2000.

- [18] Carlos A. Coello Coello and Hernández Aguirre. *Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization Approach*. Artificial Intelligence for Engineering, Design, Analysis and Manufacture, (in press), 2001.
- [19] Carlos A. Coello Coello and Ricardo Landa Becerra. *Evolutionary Multiobjective using a Cultural Algorithm*. In 2003 IEEE Swarm Intelligence Symposium, pp. 6-13, IEEE Service Center, Indianapolis, Indiana, USA, April 2003.
- [20] J.L. Cohon and D.H. Marks. *A Review and Evaluation of Multiobjective Techniques*. Water Resources Research, 11(2). pp. 208-220, 1975.
- [21] J.L. Cohon. *Multiobjective Programming and Planning*. Academic Press, 1978.
- [22] A. Charnes and W.W. Cooper. *Management Models and Industrial Applications of Linear Programming*. Volume 1. John Wiley, New York, 1961.
- [23] A. Datta, Ming-Tzu Ho and S. Bhattacharyya. *Structure and Synthesis of PID Controllers*. Advances in Industrial Control, Springer-Verlag, 2000.
- [24] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2 April, pp. 182-197, 2002.
- [25] Kalyanmoy Deb and Goyal Mayank. *Optimizing Engineering Designs Using a Combined Genetic Search*. Proceeding of the Seventh International Conference on Genetic Algorithms. Michigan State University, pp. 521-528, 1997.
- [26] John C. Doyle, Bruce A. Francis and Allen R. Tannenbaum. *Feedback Control Theory*. Macmillan, 1992
- [27] B.A. Francis. *A Course in H_∞ Control Theory*. Springer-Verlag, Berlin, 1986.
- [28] P.J. Fleming, R.C. Purshouse. *Evolutionary algorithms in control systems engineering: a survey*. Control Engineering Practice. no.10 pp.1223-1241, 2002.

- [29] C.M. Fonseca and P.J. Fleming. *Genetic algorithms for multiobjective optimization: formulation, discussion and generalization*. In Proc. 5th Int. Conf. Genetic Algorithms. S. Forrest, Ed. San Mateo, CA.: Morgan Kaufmann, pp. 416-423, 1993.
- [30] C.M. Fonseca and P.J. Fleming. *An Overview of Evolutionary Algorithms in Multiobjective Optimization*. Evolutionary Computation, 3(1), pp. 1-16, 1995.
- [31] C.M. Fonseca and P.J. Fleming. *Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part 1: A Unified Formulation*. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol. 28, No. 1, pp. 38-47, January 1998.
- [32] C.M. Fonseca and P.J. Fleming. *Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part II: Application Example*. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol. 28, No. 1, pp. 26-37, January 1998.
- [33] D.E. Goldberg and J. Richardson. *Genetics algorithms with sharing for multimodal function optimization*. Grefenstette, J.J., editor, Genetics Algorithms and their Applications: Proceedings of the second International Conference on Genetics Algorithms, pp. 41-49, Hillsdale, New Jersey. Lawrence Erlbaum, 1987.
- [34] D.E. Goldberg. *Genetics Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison wesley, 1989.
- [35] A. Haalman. *Adjusting Controllers for a Deadtime Process*. Control Engineering, July, pp. 71-73, 1965.
- [36] Alberto Herreros López. *Diseño de controladores Robustos Multiobjetivo por medio de Algoritmos Genéticos*. Tesis Doctoral, Departamento de Ingeniería de Sistemas y Automática, Universidad de Valladolid, España, 2000.

- [37] A. Herreros López, E. Baeyens Lázaro and J. R. Perán Gonzalez. *Design of Multiobjective Robust Controllers Using Genetic Algorithms*. In Annie S. Wu, editor, Proceedings of the 1999 Genetic and evolutionary Computation Conference. Workshop Program, pp. 131-132, Orlando Florida, July 1999
- [38] J.H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Harbor, MI: Univ. Mich. Press, 1975.
- [39] Y. Ijiri. *Management Goals and Accounting for Control*. North-Holland, Amsterdam, 1965.
- [40] Mo Jamshidi, Leandro dos Santos Coelho, Renato A. Krohling and Peter J. Fleming. *Robust Control Systems with Genetic Algorithms*. CRC PRESS, 2002
- [41] J. Horn and N. Nafpliotis. *Multiobjective Optimization Using the Niche Pareto Genetic Algorithm*. Technical Report IlliGA1 Report 93005. University of Illinois at Urbana Champaign, Urbana, Illinois, 1993.
- [42] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, Boston Massachusetts, 1997.
- [43] E.I. Jury and A.G. Dewey. *A General Formulation of the total Square Integrals for Continuous Systems*. IEEE Trans. Autom. Control, AC-10, pp. 119-120, January 1965.
- [44] A. Leva. *PID Autotuning Algorithm Based on Relay Feedback*. IEE Proceedings-D, 140 (5), pp. 328-338, 1993.
- [45] G P Liu, J B Yang. *Multiobjective Optimisation and Control*. RESEARCH STUDIES PRESS LTD. Baldock. Hertfordshire, England, 2003
- [46] K.F. Man, K.S. Tang, S. Kwong and W.A. Halang. *Genetic Algorithms for control and Signal Processing*. Springer-Verlag, Advances in Industrial Control, London, UK, 1997.

- [47] K.F. Man, K.S. Tang and S. Kwong. *Genetic Algorithms: Concepts and Designs*. Springer-Verlag London, 1999.
- [48] MATLAB. *Control System Toolbox*. The MathWorks, Version 6.5.0.180913a, Release 13.
- [49] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers. Boston, Massachusetts, 1998.
- [50] Z. Michalewicz. *Genetic Algorithms+Data Structures=Evolution Programs*. Springer-Verlag, third edition, 1996.
- [51] P.V. O’Neil. *Advanced Engineering Mathematics*. Brook/Cole Publishing Company, CA, USA, 1995.
- [52] Rahul B. Kasat, D. N. Kunzru, D.N. Saraf, and Santosh K. Gupta. *Multiobjective Optimization of Industrial FCC Units Using Elitist Nondominated Sorting Genetic Algorithm*. Ind, Eng. Chem. Res. 41, pp. 4761-4776, 2002.
- [53] Ibrahim Kaya. *Obtaining controller parameter for a new PI-Pd Smith predictor using autotuning*. Journal of Process Control, 13 pp. 465-472, 2003.
- [54] Ibrahim Kaya, D.P. Atherton. *A new PI-PD Smith predictor for control of processes with long dead time*. 14th IFAC World Congress. Vol. C, pp. 283-288, 1999.
- [55] Ibrahim Kaya, D.P. Atherton. *A PI-PD controller design for integrating processes*. Proc. of American Control Conference. ACC’99. pp. 258-262, 1999.
- [56] Ibrahim Kaya, D.P. Atherton. *Parameter estimation from relay autotuning with asymmetric limit cycle data*. J. Process Control 11 (4), pp. 429-439, 2001.
- [57] Ibrahim Kaya. Relay Feedback. *Identification and Model Based Controller Desig.* Dphil Thesis, University of Sussex, Brighton, UK, 1999.

- [58] S. Kirkpatrick, C. Gellatt and M. Vecchi. *Optimization and Simulated Annealing*. Science. 220(4589), pp. 671-680, 1983.
- [59] J. D. Knowles and D.W. Corne. *The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation*. In 1999 Congress on Evolutionary Computation. Washington, D.C. IEEE Service Center, pp.98-105, 1999.
- [60] A. Krohling, J. P. Rey. *Design of Optimal Disturbance Rejection PID Controllers Using Genetic Algorithms*. IEEE Trans. Evolut. Computation, 5 pp. 78-82, 2001.
- [61] A. Osyczka. *Multicriterion optimization in engineering with FORTRAN programs*. Ellis Horwood Limited, 1984.
- [62] S.S. Rao. *Multiobjective Optimization in Structural Design with Uncertain Parameters and Stochastic Processes*. AIAA Journal, 22(11), pp. 1670-1678, 1984.
- [63] S.S. Rao. *Engineering Optimization*. John Wiley and Sons, thierd edition. 1996.
- [64] D.E. Rivera, M Morari, S. Skogestad. *Internal Model Control. 4. PID controller design*. Ind. Eng. Chem. Res. pp. 252-265. 25(1) 1986.
- [65] G.V. Sarma, L. Sellami and K.D. Houam. *Application of Lexicographic Goal Programming in Production Planning-Two case studies*. Opsearch, 30(2), pp. 141-162, 1993.
- [66] Scattolini; R. and N. Schiavoni. *Regolatori PID e Metodi Classici di Taratura (in Italian)*. Proc. ANIPLA Workshop on Advanced PID Regulators for Industrial Processes, Milano (I), 1995.
- [67] J.D. Schaffer. *Multiple objective optimization with vector evaluated genetic algorithm*. PhD thesis, Vanderbilt University, Nashville, Tennessee, 1984.
- [68] J.D. Schaffer. *Multiple objective optimization with vector evaluated genetic algorithm*. Proceedings of the First International Conference on Genetic Algorithm, J.J. Grefensttete, Ed. Hillsdale, NJ: Lawrence Erlbaum, pp. 93-100, 1985.

- [69] J.D. Schaffer. *Multiobjective Learning via Genetic Algorithm*. Proceedings of the 9th International Joint Conference on Artificial Intelligence(IJCAI-85), Los Angeles California.AAAI, pp. 593-595, 1985.
- [70] C.A. Smith, A.B. Corripio. *Principles and Practice of Automatic Process Control*. John Wiley and Sons, New York, 1985.
- [71] Sigurd Skogestad. *Simple analytic rules for model reduction and PID controller tuning*. Journal of Process Control 13 pp. 291-309, 2003.
- [72] N. Srinivas and K. Deb. *Multiobjective Function Optimization Using Nondominated Sorting Genetic Algorithm*. Evol. Comput. Vol. 2, no. 3, pp. 221-248, Fall 1995.
- [73] W. Stadler. *Fundamentals of multicriteria optimization*. Stadler, W., Editor, Multicriteria Optimization in Engineering and the Sciences. Plenum Press, New York, 1988.
- [74] D.A.Van Veldhuizen and G.B. Lamont. *Multiobjective Optimization with messy Genetic Algorithms*. Proceedings of the 2000 ACM Symposium on Applied Computing, Villa Olmo, Como, Italy. ACM, pp. 470-476, 2000.
- [75] M. Vidyasagar. *Control Systems Synthesis: A Factorization Approach*. MIT Press, Cambridge, MA, 1985.
- [76] C. Vlachos, D. Williams and J.B. Gomm. *Genetic approach to decentralised PI controller tuning for multivariable processes*. IEE Proc.-Control Theory Appl., Vol. 146, No. 1, pp. 58-64, January 1999.
- [77] L.A. Zadeh. *Optimality and Nonscalar-Valued Performance Criteria*. IEEE Transactions on Automatic Control, AC-8(1), pp. 59-60, 1963.
- [78] E. Zitzler and L.Thiele. *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. IEEE Transactions on Evolutionary Computation. Vol. 3, No. 4, pp. 257-271, November 1999.

- [79] E. Zitzler, M. Laumanns and L. Thiele. SPEA2: Improving the *Strength Pareto Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.
- [80] Kemin Zhou and John C. Doyle. *Essentials of Robust Control*. Prentice Hall, 1998.
- [81] J.G. Ziegler and N.B. Nichols. *Optimum Settings for Automatic Controllers*. Trans. ASME, 64, pp. 759-768, 1942.