UNIVERSIDAD DE VALLADOLID

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

DISEÑO DE CONTROLADORES ROBUSTOS MULTIOBJETIVO POR MEDIO DE ALGORITMOS GENÉTICOS

TESIS DOCTORAL

ALBERTO HERREROS LÓPEZ

Ingeniero Industrial

2000

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
Escuela Técnica Superior de Ingenieros Industriales

DISEÑO DE CONTROLADORES ROBUSTOS MULTIOBJETIVO POR MEDIO DE ALGORITMOS GENÉTICOS

Autor:

ALBERTO HERREROS LÓPEZ
Ingeniero Industrial

Directores:

Dr. D. Enrique Baeyens Lázaro
Profesor Titular de Universidad
Dr. D. José Ramón Perán González
Catedrático de Universidad

Valladolid, 2000

Tribunal

Presidente: ______________________________

Vocal: ______________________________

Vocal: ______________________________

Vocal: ______________________________

Secretario: ______________________________

Calificación de la Tesis: ______________________

A mis padres y hermanos

Agradecimientos

Desearía hacer llegar mi agradecimiento a todas aquellas personas que me han animado y apoyado durante el desarrollo de la investigación que ha conducido a esta Tesis Doctoral.

En primer lugar, a mis Directores de Tesis, Enrique Baeyens Lázaro y José Ramón Perán González, quienes me abrierón las puertas al campo de la Automática teórica. Su apoyo y consejo durante estos años ha sido inestimable para llevar a buen término este trabajo.

Al profesor Mario Rotea, quien me acogió en su Departamento de la Universidad de Purdue. Mi sincero agradecimiento por su apoyo, consejo, interés y colaboración, tanto en el ámbito personal como investigador.

Finalmente, no quiero acabar este breve apartado sin recordar a mis compañeros del Departamento de Ingeniería de Sistemas y Automática y del Departamento de Aeronautics and Astronautics de la Universidad de Purdue. Son quizá demasiados para poder reflejar sus nombres en este breve apartado, pero su amistad y colaboración diaria ha sido fundamental para mi trabajo.

Resumen

El presente trabajo es un estudio sobre el diseño de controladores multiobjetivo y controladores robustos multiobjetivo por medio del uso de algoritmos genéticos (GAs) multiobjetivos.

La teoría de control moderna formula los problemas de control como problemas de optimización. En muchas ocasiones los objetivos que se desean de un controlador son multiples lo que conduce a un problema de optimización multiobjetivo, cuya solución no es única. Los objetivos a optimizar se miden por normas de diferentes funciones de transferencia del sistema en lazo cerrado. Existen diferentes métodos de control mixto para distintas combinaciones de normas, por ejemplo el control $\mathcal{H}_2/\mathcal{H}_\infty$ o el $\ell_1/\frac{\mathcal{H}_2}{\mathcal{H}_\infty}$. Sin embargo, dichas optimizaciones generan problemas no convexos de difícil solución, que dan como resultado controladores de un gran orden donde no se puede fijar la estructura ni añadir restricciones. En la mayoría de los casos dichos métodos reducen el problema a la obtención de un límite superior, como por ejemplo en el caso del uso de los métodos basados en inecuaciones lineales matriciales (LMIs).

Las principales aportaciones de este trabajo son:

- La formulación de un problema de control multiobjetivo como un problema de optimización de un vector de funciones respecto a los parámetros del controlador, cuya estructura ha sido elegida a priori.

- La formulación de un problema de control robusto multiobjetivo como un problema de optimización min-max de un vector de funciones entre los parámetros del controlador, cuya estructura ha sido elegida a priori, así como los parámetros de la incertidumbre.

- El estudio de los algoritmos genéticos (GAs) multiobjetivo realizados hasta la fecha. Creación de dos GAs multiobjetivos, con los operadores necesarios, para la resolución de los dos problemas propuestos, algoritmos MRCD y MRCD min-max.

- La evaluación de dichos algoritmos frente a funciones test mutiobjetivo, y frente a algoritmos basados en LMIs en la resolución de problemas de control mixto clásicos.
- El planteamiento de un método general de sintonización de PIDs multiobjetivos y aplicación del mismo a sistemas de ensayo SISO y MIMO usando el algoritmo MRCD.
- El diseño de controladores multiobjetivo (por medio del algoritmo MRCD) y controladores robusto multiobjetivo (por medio del algoritmo MRCD min-max) para una planta industrial, comparando sus resultados.

Índice General

Capítulo 1

Introducción y objetivos.

1.1 Introducción.

Muchos sistemas artificiales y naturales utilizan el control por realimentación. Éste consiste en utilizar la señal de salida del sistema, modificándola con un controlador, para introducirla de nuevo en la planta y lograr los objetivos de control deseados. La teoría de control moderna tiende a formular el diseño de controladores como un problema de optimización de unos índices relacionados con las especificaciones y la robustez del sistema realimentado. La herramienta para definir estos índices son las normas de señales y sistemas.

Las principales normas de señales son la norma-1, la integral del valor absoluto de la señal, la norma-2, la integral del cuadrado de la señal que refleja la potencia o energía de la misma, y la norma-∞, el máximo valor de la señal que refleja la amplitud de la misma. Las señales en tiempo continuo que tengan la norma-i menor que infinito pertenecen al espacio normado $\mathcal{L}_i$, y las de tiempo discreto al espacio normado ℓ_i. Estos espacios normados son lineales. Un sistema es un conjunto de señales divididas en dos grupos, con una relación causal, la señales de entrada y las señales de salida. El sistema se llama SISO si tiene una entrada y una salida y MIMO si tiene varias entradas y varias salidas. Se considera que las entradas y salidas de un sistema son señales pertenecientes a un espacio normado. Una norma inducida, norma-(i, j), de un sistema es la máxima relación entre la norma-j de una señal de entrada perteneciente al espacio $\mathcal{L}_j$ (o ℓ_j) y la norma-i de la señal de salida correspondiente. Las normas de sistemas más empleadas son la norma $\mathcal{H}_2$, que mide la norma-2 de la salida del sistema a una entrada impulso, la norma

$\mathcal{H}_\infty$ que corresponde con la norma inducida norma-$(2,2)$ y la norma $\mathcal{L}_1$ que corresponde con la norma inducida por la norma-(∞,∞), ver apéndice B para un estudio más detallado de estas normas.

El primer diseño de controladores formulado como un problema de optimización fue el control óptimo LQG, o la optimización de la norma $\mathcal{H}_2$. En este problema se obtienen las especificaciones deseadas garantizando estabilidad a la planta ante ruidos (señales estocásticas). Este método considera que se conoce un modelo exacto de la planta, y por ello los controladores obtenidos con realimentación de la señal de salida y filtro de Kalman no son robustos, es decir, no conservan sus propiedades ante cambios en el modelo de la planta. La optimización $\mathcal{H}_\infty$ se construyó principalmente para conseguir controladores robustos en plantas con incertidumbre. Se formula en el dominio de la frecuencia y sus ecuaciones son muy similares a las del control LQG. Esta optimización fue tachada de conservadora por considerar que la incertidumbre del sistema no tiene una estructura determinada, por lo que se elaboró la teoría del valor singular estructurado o teoría μ para plantas con incertidumbre estructurada, ver [ZDG96]. Con el fin de trabajar con especificaciones temporales se desarrolló la optimización $\mathcal{L}_1$ (ℓ_1 en discreto), donde se trabaja con señales limitadas en magnitud, pertenecientes a $\mathcal{L}_\infty$, ver [DDB95].

Sin embargo, el diseño de un controlador raramente responde a la optimización de un solo objetivo, y normalmente los objetivo requeridos suelen ser contradictorios. Un ejemplo típico es el diseño de un controlador que cumpla con unas especificaciones temporales, que sea robusto ante una incertidumbre en la planta y que no sature la señal de control.

Los problemas donde se desea optimizar varios objetivos en conflicto entre sí, dentro de un espacio de búsqueda, no tienen una solución única. Esto es debido a que, dados dos puntos del espacio de búsqueda, cada uno de ellos puede mejorar al otro con respecto a objetivos diferentes. La solución de este problema, según la optimización de Pareto, es el conjunto de los puntos del espacio de búsqueda no dominados, llamado conjunto de óptimos de Pareto, [BT80]. Un punto del espacio de búsqueda es no dominado si no existe otro punto en dicho espacio que optimice todos los objetivos deseados a la vez respecto del primero. Se llamará frente de Pareto a los objetivos asociados al conjunto de óptimos de Pareto.

En los últimos años, la teoría de control moderno ha desarrollado una serie de problemas mixtos de control donde se intenta dotar al controlador de varias propiedades. El primero fue el control mixto $\mathcal{H}_2/\mathcal{H}_\infty$, donde se combinan las especificaciones deseadas usando la norma $\mathcal{H}_2$

y la robustez del controlador con la norma $\mathcal{H}_\infty$, ver [BH89, KR91]. Esta combinación parece estar justificada por la analogía en la resolución de ambos problemas en términos de la ecuación de Riccati, ver [ZDG96]. Otros intentos de control mixto son el control $\mathcal{L}_1/\mathcal{H}_2, \ell_1/\mathcal{H}_2, \mathcal{L}_1/\mathcal{H}_\infty$ y otras combinaciones, tanto en sistemas SISO como MIMO, cuya realización es mucho más compleja, ver [VJ97].

Todos los problemas de control mixto citados tienen una serie de características comunes que hace difícil su resolución:

- Los problemas de optimización mixta de dos normas son reducidos a problemas de optimización simple de una norma con una restricción en la otra, lo que lleva a encontrar un único controlador como solución. Si se desea encontrar el conjunto de soluciones del problema, el conjunto de óptimos de Pareto, se deberá repetir dicho problema para diferentes valores en la restricción, tantas veces como controladores se desee.

- Los problemas planteados no son convexos, y su solución exacta es muy compleja, exigiendo en casi todos los casos reducir el problema a la minimización de una cota superior para poder usar técnicas de optimización convexa. Un método ampliamente usado es la aproximación del problema a uno convexo planteado a base de inecuaciones lineales matriciales, ver el apéndice D donde se plantea el problema $\mathcal{H}_2/\mathcal{H}_\infty$ en dichos términos.

- Cada problema mixto se resuelve de forma diferente, siendo muy compleja la optimización de más de dos objetivos. La optimización $\mathcal{H}_2$ y $\mathcal{H}_\infty$ se formulan en ecuaciones de estado, mientras que la $\mathcal{L}_1$ se formula como programación lineal de dimensión infinita en el dominio de la frecuencia, siendo difícil su combinación, ver [Vou94, SDV95].

- El orden de los controladores obtenidos es muy elevado, pudiendo ser incluso mayor que el de la propia planta. En muchos caso el orden del controlador para la solución exacta es infinito. Esto hace que sea necesario un segundo paso donde se reduce dicho orden para poder poder implantar físicamente el controlador.

- No es posible fijar la estructura del controlador, ni límites de sus parámetros. Por ello, estos métodos son difíciles de usar para sintonizar controladores PID (Proporcional-Integral-Derivativa), por ejemplo. Los métodos basados en inecuaciones lineales matriciales (LMI) generan controladores con unos parámetros muy elevados que pueden producir saturación en la señal de control.

- En algunos casos, los controladores que se obtienen son muy frágiles, en el sentido de que un pequeño cambio en sus parámetros modifica mucho sus propiedades.

El objetivo de este trabajo es la solución de problemas de diseño de control multiobjetivo y control robusto multiobjetivo, planteándolos como un problema de optimización paramétrica multiobjetivo y un problema de min-max paramétrico muultiobjetivo.

Se parte de una planta genérica con incertidumbre que se desea controlar por medio de un controlador con una estructura dada a priori donde alguna de sus características, como por ejemplo el orden, pueden ser variable. Con ayuda de la transformación fraccional lineal (LFT), ver apéndice A, la planta con incertidumbre se divide en dos bloques funcionales claramente diferenciados, el bloque planta nominal y el bloque incertidumbre, con entradas y salidas que los unen. A esta planta se le añade el controlador en realimentación con la estructura deseada y se plantean los objetivos en función de diferentes normas entre las diversas entradas y salidas del sistema y posibles restricciones, tanto en los valores de los parámetros del controlador como en las normas objetivo.

El problema de control robusto multiobjetivo en forma paramétrica se puede formular de dos formas diferentes, dependiendo principalmente del tipo de incertidumbre del sistema. El primer camino es la búsqueda de los parámetros del controlador, cuya estructura ha sido elegida a priori, que minimizan los objetivos deseados y cumplan las restricciones para la planta nominal. En este caso la incertidumbre se elimina del diagrama de bloques, mediante la aplicación del criterio de pequeña ganancia, pero no así sus entradas y salidas hacia el sistema, que servirán para conseguir objetivos de robustez y especificaciones que minimicen su efecto en la salida del sistema. Este planteamiento se usa en el caso de incertidumbre no estructurada, al igual que los problemas mixtos antes mencionados. El segundo camino es la búsqueda de los parámetros del controlador, cuya estructura ha sido elegida a priori, que minimizan los objetivos deseados y cumplan las restricciones en el peor caso posible de la incertidumbre. Este planteamiento sólo se podrá realizar cuando la incertidumbre sea paramétrica, en cuyo caso depende de un número finito de parámetros reales.

Estos dos caminos en la resolución del problema corresponden en el primer caso a un problema de minimización de unos objetivos respecto a los parámetros del controlador, y en el segundo caso a un problema de min-max, es decir, minimización de unos objetivos respecto de los parámetros de un controlador y maximización de los mismos objetivos respecto de los parámetros de la incertidumbre de la planta. En ambos casos, los controladores son un conjunto de parámetros a optimizar y los objetivos, normas entre señales de entrada y salida, no son más que funciones de los parámetros del controlador que devuelven un valor. Este planteamiento da una flexibilidad total en la elección y combinación de normas a optimizar, y en la estructura y restricciones de los controladores.

Se necesita una herramienta potente para poder abordar los problemas de minimización y min-max multiobjetivo propuestos. Las principales dificultades que se van a tener que afrontar son las siguientes:

- Es un problema de optimización multiobjetivo cuya solución es un conjunto de controladores, el correspondiente al frente de Pareto del problema.

- El problema no es convexo, existiendo en su solución muchos subóptimos.

- Los límites del espacio de búsqueda de los parámetros del controlador no están en muchos casos bien definidos.

- Existe redundancia en los parámetros del controlador de forma que con varios controladores diferentes se puede obtener los mismos objetivos.

- Es preciso fijar una estructura del controlador a priori, hecho que en muchos casos es una ventaja pero en otros puede ser una restricción.

Las herramientas elegidas para la solución de ambos problemas han sido los algoritmos genéticos multiobjetivo, que han experimentado un gran desarrollo en los últimos años dando muy buenos resultados en los problemas de optimización paramétrica multiobjetivo, [FF95a, VV99a, Hor97, Deb99, CC99].

Los algoritmos genéticos (GAs) son un subconjunto de ciertos procedimientos heurísticos basados en las leyes de supervivencia denominados algoritmos evolutivos (EAs), que se han desarrollado ampliamente en las últimas dos décadas, ver [Bäc97]. Son algoritmos de muestreo dirigido, tales que, a partir de un espacio de búsqueda y un objetivo a optimizar muestrean de forma uniforme dicho espacio, en una primera fase. Cada punto muestreado es valorado con respecto al objetivo a optimizar, y se vuelve a muestrear el espacio de búsqueda pero esta vez no de forma uniforme sino teniendo en cuenta los resultados del muestreo anterior. Tras una serie de muestreos dirigido, el algoritmo se centra en puntos cada vez más próximos al óptimo global. En los algoritmos genéticos este proceso se realiza a través de tres operadores principales; selección, cruce y mutación. Los puntos muestreados, individuos de una población, son valorados respecto al objetivo a optimizar. De ellos se elige a los mejores respecto al objetivo y se forma una nueva población a partir del cruce y mutación de los individuos de la subpoblación seleccionada. El proceso continúa hasta que el algoritmo converge a la supuesta solución global. El éxito de estos algoritmos se basa en el equilibrio entre la explotación de los mejores individuos encontrados y la exploración del resto del espacio de búsqueda, con el fin de no descartar la posibilidad de que el verdadero óptimo global esté en otra zona. Bajo ciertas condiciones, cuando las poblaciones encontradas sean monótonas, es posible demostrar la convergencia en probabilidad al óptimo global, ver [Bäc96].

Los algoritmos genéticos fueron desarrollados en un principio para problemas de optimización global, pero su dinámica de muestreo direccionado los hacen muy útiles en problemas de optimización multiobjetivo. Para ello se han modificado los operadores y objetivos del GA simple con el fin de conseguir como resultado el conjunto de puntos óptimos de Pareto. Esto se logra con dos movimientos, convergencia de la población hacia el conjunto de óptimos y diversidad en la misma para conseguir todos los óptimos. Diferentes autores han desarrollado algoritmos que tratan de resolver este problema creando operadores especiales y modificando los objetivos a optimizar para conseguir en ellos convergencia y diversidad en la población final, ver [Mah95] y sus referencias.

Los teoremas de “Nada Es Gratis (No Free Lunch o NFL)” demuestran que dos algoritmos de muestreo dirigido tienen una media similar en su índice de bondad para resolver todo tipo de problemas. Ello implica que para que un algoritmo sea efectivo en la resolución de un tipo concreto de problemas debe tener información sobre dicho problema. La forma de valorar dos GAs en la resolución de una clase de problemas es el uso de funciones test que hagan patentes las dificultades que la resolución de dicho problema implica. Existen numerosas funciones test para problemas de optimización global y para optimización multiobjetivo.

Los antecedentes históricos de aplicación de algoritmos genéticos a problemas de control se remontan a la última década. Se han centrado sobre todo en la sintonización de PIDs u otros controladores de estructura fija en sistemas SISO. Existen también ejemplos de problemas de control multiobjetivo usando estas técnicas, y en algunos casos se ha abordado problemas de control robusto. Sin embargo, el planteamiento general del control multiobjetivo para un sistema MIMO con un controlador genérico definido en alguna de sus formas canónicas no ha sido planteado, probablemente por la dificultad que encierra el que no estén bien definidos los límites del espacio de búsqueda de los parámetros a optimizar. Las aplicaciones de algoritmos genéticos (GAs) a problemas de min-max y en concreto a control robusto de plantas con incertidumbres paramétricas no existen, aunque se han desarrollado aproximaciones donde se discretiza la planta con incertidumbre como un conjunto finito de plantas con distintos parámetros.

1.2 Objetivos.

El objetivo general de este trabajo es doble. En primer lugar se desea resolver de forma general el problema de control multiobjetivo de una planta nominal buscando los controladores (sus parámetros) que minimizan, aplicando la optimización de Pareto, los objetivos deseados y cumplan con unas restricciones. En segundo lugar se desea resolver de forma general el problema de control multiobjetivo de una planta con incertidumbre paramétrica buscando los controladores (sus parámetros) que minimizan, en el sentido de Pareto, los objetivos deseados y cumplan unas restricciones, para el peor caso posible de incertidumbre. La resolución de ambos problemas se realizará gracias al diseño de dos algoritmos genéticos (GA) multiobjetivo que desarrollarán operadores destinados a lograr la máxima eficacia en la resolución de ambos problemas. Tras ello se evaluarán los algoritmos con respecto a diferentes funciones test y con respecto a problemas de control mixto clásicos, y se aplicarán a casos prácticos.

De forma más detallada, los objetivos son:

1. Estudio de las dificultades que implica el diseño de control mutiobjetivo de una planta nominal. Planteamiento de dicho problema como la minimización de Pareto de los objetivos respecto de los parámetros del controlador, cuya estructura ha sido definida a priori.

2. Estudio de las dificultades que implica el diseño de control mutiobjetivo de una planta con incertidumbre paramétrica. Planteamiento del problema como un min-max entre los

parámetros del controlador, cuya estructura ha sido elegida a priori, y los de la incertidumbre.

3. Estudio de los algoritmos genéticos multiobjetivos realizados hasta la fecha y, en concreto, los operadores que utilizan para conseguir la convergencia y diversidad de la población.

4. Diseño de un algoritmo genético multiobjetivo con nuevos operadores que se ajusten al problema de minimización propuesto, algoritmo MRCD.

5. Diseño de un algoritmo genético multiobjetivo que resuelva un problema min-max, algoritmo MRCD min-max.

6. Evaluación de los algoritmos MRCD mediante funciones test multiobjetivo y comparación con los algoritmos existentes.

7. Evaluación de los algoritmos MRCD en problemas clásicos de control mixto con diferentes plantas de ensayo, comparándolos con otras técnicas de diseño de control robusto, en concreto, con las inecuaciones lineales matriciales (LMI).

8. Planteamiento de un método general de sintonización de controladores PIDs como un problema de optimización multiobjetivo, aplicándolo a diversos sistemas de ensayo SISO y MIMO. Resolución mediante el algoritmo MRCD y comparación de los resultados con los obtenidos por medio de otras técnicas.

9. Aplicación a un problema de control industrial donde la planta tiene incertidumbre paramétrica. Utilización del algoritmo MRCD y el algoritmo MRCD min-max para la sintonización de controladores PIDs, comparando los resultados.

1.3 Organización de la memoria.

La presente memoria está dividida en ocho capítulos y seis apéndices. En el capítulo 2 se plantea las dos formulaciones del diseño de controladores para una planta con incertidumbre donde se desea optimizar varios objetivos. Se describen las diversas aproximaciones desarrolladas en la teoría moderna de control para plantas con incertidumbre estructurada y no estructurada. Por último, se plantean nuevas aproximaciones basadas en la optimización paramétrica multiobjetivo y en un problema min-max entre los parámetros del controlador y la incertidumbre.

En el capítulo 3 se define la optimización global y la multiobjetivo basada en la optimización de Pareto. Se describen las técnicas clásicas de resolución de problemas multiobjetivo, y se detallan diferentes algoritmos estocásticos para resolver un problema de optimización global, en especial, los algoritmos genéticos (GA). Tras ello, se describen los nuevos operadores e índices de bondad necesarios para conseguir que un GA pueda resolver un problema de optimización multiobjetivo. Se estudian los principales GA multiobjetivos diseñados hasta la fecha, y se da una panorámica general de las aplicaciones realizadas en problemas prácticos multiobjetivos y, en especial, en el área de control.

En el capítulo 4 se describen los GA multiobjetivos desarrollados en esta memoria, el algoritmo MRCD y el algoritmo MRCD min-max. Se estudian los problemas que deben resolver, y se analizan los operadores desarrollados para conseguirlo, así como la dinámica general de dichos algoritmos.

En el capítulo 5 se estudian diversas teorías sobre la convergencia de un GA, y las funciones test necesarias para su evaluación, en especial, diez funciones test multiobjetivo. Con estas funciones se compara la bondad del algoritmo MRCD respecto a las verdaderas soluciones de los problemas, y respecto de las soluciones alcanzadas por otros GA multiobjetivos. Luego se formulan dos problemas de diseño mixtos clásicos, que se resuelven para trece sistemas de ensayo usando el algoritmo MRCD, con diferentes parametrizaciones, y las inecuaciones lineales matriciales. Se comparan los resultados. Por último, se resuelve un mismo problema a partir del algoritmo MRCD con controladores de distinto orden y distinta parametrización y se comparán los resultados.

En el capítulo 6 se plantea un método general de sintonización de PIDs como un problema de diseño multiobjetivo. Se aplica a trece sistemas de ensayo SISO y los resultados se comparan con los obtenidos en el artículo [APH98]. Se vuelve a repetir el problema para sistemas MIMO, con distintos grados de acoplamiento, empleado dos controladores PID desacoplados y cuatro acoplados.

En el capítulo 7 se presenta una planta industrial, compuesta de dos entradas y dos salidas, con subsistemas de primer orden con retardo cuyos parámetros tienen incertidumbre. Se plantea un problema de minimización de la sensibilidad y la sensibilidad complementaria en diferentes bandas de frecuencia respecto a los parámetros de dos controladores PID desacoplados. El problema se plantea con el algoritmo MRCD considerando que los parámetros de la planta son los nominales, y con el algoritmo MRCD min-max considerando que los parámetros tienen incertidumbre. Se comparan los resultados.

En el capítulo 8 se presentan las conclusiones de la tesis y se proponen futuras líneas de investigación.

El apéndice A describe las transformaciones fraccional lineal (LFT) que permite separar una planta con incertidumbre en dos bloques, la planta nominal y la incertidumbre. El apéndice B estudia las diferentes normas para señales y sistemas, analizando las relaciones entre ellas. El apéndice C describe las formas canónicas controlables, diagonal y balanceada usadas como forma de parametrización de los controladores. El apéndice D describe el problema mixto clásico $\mathcal{H}_2/\mathcal{H}_\infty$ y su resolución a partir de inecuaciones lineales matriciales. Y el apéndice E describe la estructura de un controlador PID (Proporcional-Integral-Derivativo), definiendo las diferentes formas en las que se puede expresar.

Capítulo 2

Diseño de control robusto multiobjetivo.

2.1 Introducción.

En la teoría de control moderna, el diseño de controladores se formula como un problema de optimización de una serie de normas de determinadas funciones de transferencia de lazo cerrado que miden las especificaciones del sistema, ver apéndice B. El primer intento fue la optimización de la norma $\mathcal{H}_2$ o control LQG, que consigue controladores estables en lazo cerrado, que atenúan el efecto de ruido en las entradas. Como más tarde se demostró, estos controladores no son robustos a incertidumbres en la planta. A partir de ahí se desarrolló la optimización $\mathcal{H}_\infty$ para conseguir controladores robustos ante cambios en el modelo de la planta, y la $\mathcal{L}_1$ (ℓ_1 en discreto) para conseguir especificaciones temporales.

Normalmente, las especificaciones deseadas no pueden determinarse con una sola norma. Así, la minimización de la norma $\mathcal{H}_2$ es buena para la atenuación de efecto de ruidos, pero no es valida, como sí lo es la norma $\mathcal{H}_\infty$, para conseguir robustez. La norma $\mathcal{L}_1$ (ℓ_1 en discreto) es valida para minimizar la relación entre los picos de las señales de entrada y salida en el dominio temporal. Este hecho llevó a plantear el problema de diseño como una optimización mixta de distintas normas de otras tantas funciones de transferencia del sistema en lazo cerrado. Desgraciadamente, las normas a optimizar habitualmente están en conflicto, ya que de otra forma tendrían información redundante, con lo que el problema de optimización mixto se convierte en un problema de optimización multiobjetivo cuya solución no es única, ver capítulo 3.

Los problemas de optimización de normas han sido formulados en diferentes espacios. Mientras que la norma $\mathcal{H}_\infty$ y $\mathcal{H}_2$ se formulan en el espacio de estado, basándose en la solución de la ecuación de Riccati (ver [ZDG96]), la norma $\mathcal{L}_1$ se formula como un problema de programación lineal de dimensión infinita en frecuencia, ver [DDB95]. Los problemas mixtos donde se mezclan este tipo de normas no son convexos y tienen difícil solución. En los últimos años, muchos problemas mixtos han sido planteados como la búsqueda de un límite superior, solución de un problema convexo, formulando con ayuda de inecuaciones lineal matricial (LMI), ver apéndice D. Los controladores obtenidos por estos métodos son de un orden muy elevado, mayor incluso que los de la planta a controlar, y para su implementación práctica se precisa la reducción de este orden. Además, estos métodos no son válidos para conseguir controladores con una estructura fija, o con restricciones en el valor de sus parámetros.

En este trabajo se plantea el diseño multiobjetivo de controladores como un problema de optimización multiobjetivo con respecto a los parámetros de los mismos, cuya estructura ha sido definida a priori. Este procedimiento da una gran flexibilidad en la elección de objetivos, ya que nunca van a ser incompatibles, en la elección de la estructura de los controladores, y en las restricciones para el valor de sus parámetros.

Este capítulo está dividido en las siguientes secciones. En la sección 2.2 se plantea el problema general de diseño de controladores en una planta con incertidumbre, para el caso en el que se desconozca la estructura de dicha incertidumbre (no estructurada), y para el caso en el que sí se conozca la misma (incertidumbre estructurada). La sección 2.3 revisa los métodos clásicos de tratamiento de diferentes problemas de optimización mixta para el caso de incertidumbre estructurada y no estructurada. La sección 2.4 plantea el problema de diseño multiobjetivo y el problema de diseño robusto multiobjetivo como un problema de optimización paramétrica multiobjetivo y como un problema min-max. La sección 2.5 plantea las conclusiones del capítulo.

2.2 Formulación general del diseño robusto multiobjetivo.

Uno de las principales aplicaciones del diseño de control robusto es encontrar un controlador para una familia de plantas que dé estabilidad a la misma y cumpla con una serie de especificaciones. Con ayuda de la transformación fraccional lineal (LFT), ver apéndice A, una planta con incertidumbre se puede dividir en dos bloques conectados por sus entradas y salidas; y de la misma forma se puede plantear una planta y un controlador en realimentación. Usando esta herramienta el diagrama de bloques general de una planta con incertidumbre realimentada por un controlador queda tal y como se muestra en la figura 2.1; donde $G_n(s)$ es el modelo de la planta nominal, Δ_G representa la incertidumbre y K el controlador.

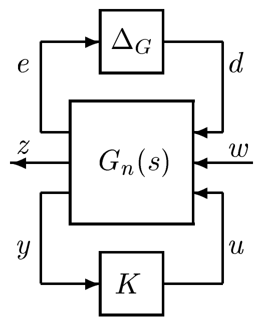


Figura 2.1: Planta nominal, incertidumbre y controlador en realimentación.

Como se indica en la sección A.4.2, este modelo se puede plantear como una doble LFT

$$z = \mathcal{F}_u(\mathcal{F}_l(G_n, K), \Delta_G)w = \mathcal{F}_l(\mathcal{F}_u(G_n, \Delta_G), K)w.$$

La incertidumbre es un bloque variable donde no se conocerá más que un límite de su tamaño, medido mediante alguna norma inducida ($\mathcal{H}_\infty$ ó $\mathcal{L}_1$), en el peor de los casos, o bien ese límite y su estructura. El controlador es un bloque a determinar para conseguir las especificaciones de diseño. En un problema de diseño de controladores hay dos tipos principales de objetivos: por un lado los relacionados con el comportamiento dinámico de la señales características del sistema y por otro lado los objetivos de robustez, que garantizan un mínimo comportamiento dinámico incluso en presencia de incertidumbre. Todas estas especificaciones pueden formularse en términos de minimización de normas, $\mathcal{H}_\infty, \mathcal{H}_2$ y $\mathcal{L}_1$ o sus equivalentes en discreto, de bloques de la matriz de transferencia, que pueden ser filtrados mediante funciones peso de entrada-salida, ver figura B.1 y apéndice B. La minimización de varias normas al mismo tiempo es normalmente conflictiva por responder a especificaciones opuestas, por lo que el problema se convierte en un problema de optimización multiobjetivo cuya solución no es única, ver capítulo 3.

El problema de diseño de controladores robustos puede ser planteado por dos caminos,

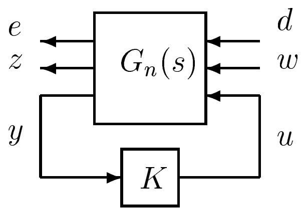


Figura 2.2: Planta nominal y controlador en realimentación.

en función del conocimiento que se tenga de la incertidumbre. El primer planteamiento del problema se suele emplear cuando se desconoce la estructura de la incertidumbre, incertidumbre no estructurada. En este caso, tras descomponer el sistema en tres bloques por medio de la LFT; planta, controlador e incertidumbre; se elimina el bloque incertidumbre del sistema general. El sistema quedaría definido por

$$\begin{bmatrix} e \\ z \end{bmatrix} = \mathcal{F}_l(G_n, K) \begin{bmatrix} d \\ w \end{bmatrix}$$

donde el efecto de la incertidumbre queda reflejado en las entradas d y salidas e de la misma en el sistema. Los objetivos a optimizar van ir encaminados en diversos sentidos. Por un lado, conseguir unas especificaciones con la minimización de la norma $\mathcal{H}_2$ de la función de transferencia correspondiente. Por otro lado, aumentar la robustez del sistema, hecho que se puede conseguir minimizando $\|T_{ed}(s)\|_\infty$, por la aplicación del teorema de pequeña ganancia, ver [ZDG96]. Otros objetivos pueden evitar la saturación del controlador minimizando la norma $\|T_{uw}\|_p$, donde p puede ser la norma $\mathcal{L}_1$.

Al eliminar la incertidumbre, el problema se convierte en un diseño de controladores multiobjetivo, siendo los objetivos las diferentes normas entre otras tantas funciones de transferencia en lazo cerrado. La formulación general de un problema de diseño de controladores multiobjetivo para una planta nominal viene dada por,

$$\min_K \begin{bmatrix} \|W_1[\mathcal{F}_l(G_n, K)]_{ind_1} V_1\|_{p_1} \\ \vdots \\ \|W_n[\mathcal{F}_l(G_n, K)]_{ind_n} V_n\|_{p_n} \end{bmatrix}, \tag{2.1}$$

donde $V_i(s)$ y $W_i(s)$ son los filtros, $[\cdot]_{ind_i}$ es el bloque de la matriz de transferencia elegido y p_i la norma que se aplicará.

El segundo planteamiento no elimina el bloque incertidumbre, transformándose en un problema min-max multiobjetivo. Se deberá encontrar el controlador, K, que para el peor caso de incertidumbre minimice unas determinadas normas de los bloques de la matriz de transferencia elegidos. El problema de diseño robusto de controladores multiobjetivo viene dado por

$$\min_{K} \max_{\Delta_G} \begin{bmatrix} \|W_1[\mathcal{F}_l(\mathcal{F}_u(G_n, \Delta_G), K)]_{ind_1} V_1\|_{p_1} \\ \vdots \\ \|W_n[\mathcal{F}_l(\mathcal{F}_u(G_n, \Delta_G), K)]_{ind_n} V_n\|_{p_n} \end{bmatrix}, \tag{2.2}$$

donde $V_i(s)$ y $W_i(s)$ son los filtros, $[\cdot]_{ind_i}$ es el bloque de la matriz de transferencia elegido y p_i la norma que se aplicará. Esta segunda definición sólo tendrá validez práctica cuando se conozca la estructura de la incertidumbre, como por ejemplo en el caso de que la incertidumbre sea paramétrica, ver sección A.5.

En la figura 2.3 se muestra un ejemplo típico del sistema general planteado en la figura 2.1. En él la incertidumbre se descompone en una forma nominal, con el uso de filtros, $\Delta_G = V_1 \Delta W_1$, siendo $\|\Delta(jw)\|_\infty < 1$. Las nuevas entradas serán $e' = W_1 e$ y $d' = V_1^{-1} d$. Un problema típico $\mathcal{H}_2/\mathcal{H}_\infty$ consiste en la minimización de la norma $\mathcal{H}_2$ entre la salida de la incertidumbre nominal w' y la salida del sistema filtrada z', y la minimización de la norma $\mathcal{H}_\infty$ entre la salida de la incertidumbre nominal d' y su entrada e', ver apéndice D.

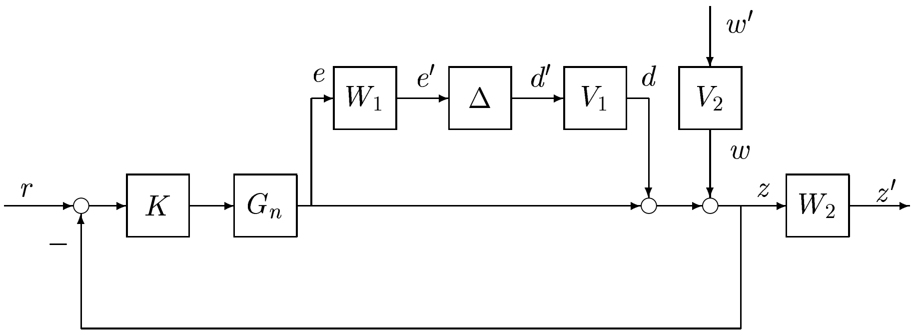


Figura 2.3: Sistema con incertidumbre multiplicativa.

2.3 Revisión de control robusto multiobjetivo.

La clasificación de las distintas técnicas empleadas en el control robusto viene dada por el conocimiento de la estructura de la incertidumbre. Así, se tendrán dos tipos de problemas principales, los de incertidumbre no estructurada y los de incertidumbre estructurada, ver sección 2.2. La aplicación de control multiobjetivo se ha desarrollado mucho más en el caso de problemas con incertidumbre no estructurada.

2.3.1 Incertidumbre no estructurada.

La teoría de control moderna ha tratado los problemas de diseño de controladores como problemas de optimización de normas, $\mathcal{H}_2$, $\mathcal{H}_\infty$ ó $\mathcal{L}_1$ para el caso continuo, de las funciones de transferencia entre las entradas-salidas elegidas, ver sección 2.2. La primera teoría de este tipo fue la optimización $\mathcal{H}_2$, o de LQG, de la que una buena referencia es [Kal60]. Esta optimización obtiene las especificaciones de comportamiento dinámico deseadas mientras se garantiza la estabilidad del sistema en lazo cerrado en presencia de incertidumbre en la planta. Los ruidos pueden ser modelados como procesos estocásticos con un espectro característico y considerar como objetivo la reducción de la potencia en la señal regulada. Sin embargo, la teoría LQG no garantiza márgenes de robustez si es aplicada usando observadores o filtros de Kalman. La teoría de optimización $\mathcal{H}_\infty$ fue desarrollada por Zames [Zam81]. Esta optimización permite la resolución de un problema de estabilidad robusta, consiguiéndose estabilidad en lazo cerrado en presencia de incertidumbre en el sistema. Tiene dos grandes ventajas, la norma $\mathcal{H}_\infty$ se define en el dominio de la frecuencia y la solución de la optimización $\mathcal{H}_\infty$ es muy similar a la de LQG, ya que usa ecuaciones de Riccati. Sin embargo, la optimización $\mathcal{H}_\infty$ puede ser tachada de conservadora por considerar las incertidumbres como no estructuradas, hecho que ha llevado a desarrollar la teoría de síntesis μ. Por otro lado, la optimización $\mathcal{H}_\infty$ considera que las señales de entrada, salida son del tipo $\mathcal{L}_2$, cuadráticamente integrables, hecho que raramente es cierto. Recientemente se ha desarrollado la optimización $\mathcal{L}_1$, o ℓ_1 en el caso discreto. Este método considera señales entrada, salida del tipo $\mathcal{L}_\infty$, y posibilita conseguir especificaciones temporales, donde la señal esté limitada en magnitud. En [ZDG96, DFT92, Fra87] se explica con detalle la optimización $\mathcal{H}_\infty$, y en [DDB95] la optimización ℓ_1.

De la mezcla de especificaciones donde intervengan estos tres tipos de normas se generan problemas mixtos donde los objetivos a optimizar van a estar en conflicto entre sí, ya que sino serían redundantes, lo que da lugar a problemas de optimización multiobjetivo donde la solución no es única, ver sección 3.2.

Los problemas mixtos que se tratarán en esta sección van a convertir el problema multiobjetivo, dos objetivos, de la ecuación (2.1) como un problema de un solo objetivo y una restricción en el otro objetivo. Esto lleva a obtener una sola solución del problema, siendo necesario para conseguir todas las soluciones del mismo, repetirlo para distintos valores de la restricción. El método y espacio donde se resuelven los distintos problemas dependen de las normas a optimizar, no siendo posible plantear una solución general. Así, el problema $\mathcal{H}_\infty$ y $\mathcal{H}_2$ se resuelve en el espacio de estado aplicando la ecuación de Riccati y el problema $\mathcal{L}_1$ se formula como un problema de programación lineal de dimensión infinita. Esto hace que no sea fácil resolver un problema con más de dos objetivos.

El orden de los controladores obtenidos con este tipo de técnicas es igual o mayor al de la planta, siendo en muchos casos imprescindible su reducción para poderlo utilizar de forma práctica, [ZDG96]. Por otro lado, alguno de estos controladores tienen problemas de fragilidad, ver [KB97], es decir, son muy sensibles a pequeños cambios en sus parámetros.

Diseño de control $\mathcal{H}_2/\mathcal{H}_\infty$.

La forma más frecuente de la optimización $\mathcal{H}_2/\mathcal{H}_\infty$ es una formulación convexa usando inecuaciones matriciales, ver apéndice D. El modelo general adoptado es el de la figura 2.2 donde se considera que la entrada d está limitada en potencia y la w es un ruido blanco. El problema mixto se reduce a un problema de minimización de una norma $\mathcal{H}_2$ con restricciones en la norma $\mathcal{H}_\infty$,

$$\begin{array}{l} \min_K \|T_{zw}(s)\|_2 \\ \text{s.a. } \|T_{ed}(s)\|_\infty < \gamma \end{array}$$

donde $T_{ij}(s)$ representa una función de transferencia entre la entrada j y la salida i. El objetivo del problema es conseguir reducir el efecto del ruido blanco, con la minimización de la norma $\mathcal{H}_2$, a la vez que se exige robustez al sistema con una restricción en la norma $\mathcal{H}_\infty$, aplicando el teorema de pequeña ganancia, ver [ZDG96]. Este problema fue planteado por primera vez por [BH89] como una minimización de la cota superior de la norma $\mathcal{H}_2$ sujeta a una restricción en la norma $\mathcal{H}_\infty$, en un sistema con un bloque de entrada $w := w = d$ y dos de salida $[e, z]'$. En [ZGG+94, DZGB94] se plantea el caso contrario, dos bloques de entrada $[d, w]'$ y uno de salida $z := z = e$. El problema planteado en [BH89] fue convertido en un verdadero problema de optimización convexa con restricciones en [KR91] para el caso continuo y en [KKR93] para el caso discreto. En [BK94] se aplica los resultados de [KR91] a dos ejemplos numéricos y se demuestra que las especificaciones $\mathcal{H}_2$ conseguidas pueden ser peores que las obtenidas con el control clásico control $\mathcal{H}_\infty$, [DGKF89]. El primer intento para conseguir la solución de la norma $\mathcal{H}_2$ y no de una cota superior es en [RVDS92], aunque implica el planteamiento de unas ecuaciones matriciales no lineales acopladas, de difícil solución. En [WR92] se intenta aumentar el orden del controlador, de forma que pueda llegar a ser infinito, y en [RK91] se obtiene las condiciones suficientes para la resolución del problema como un sistema con realimentación de estado, considerando controladores de mayor orden que la planta. En [Meg94] se demuestra que el problema general tiene dimensión infinita y que no tiene una realización en espacio de estado exponencialmente estable a menos que la restricción $\mathcal{H}_\infty$ sea redundante. En [WR95, SRBS97] se obtiene un controlador, con restricción en su orden, como solución general del sistema, figura 2.2, para el caso continuo, y en [Szn94a, RS98] para el caso discreto. El método empleado es una optimización convexa de dimensión finita, con el uso de la parametrización de Youla, [ZDG96]. Una dificultad asociada a este procedimiento es la sensibilidad del problema a cambios en el orden del controlador.

Debido al desarrollo de los métodos de puntos interiores para inecuaciones lineales matriciales (LMI), y su aplicación con éxito a numerosos campos de la teoría de control [BEGFB94], el problema mixto $\mathcal{H}_2/\mathcal{H}_\infty$ tiende a formularse mediante LMIs. En [CW95] con ayuda de la parmetrización de Youla se plantea el problema como una inecuación bilineal matricial que se reduce a una LMI. En [SGC97] se da una relación de problemas mutiobjetivo que pueden ser resueltos por LMIs, estando entre ellos la optimización $\mathcal{H}_2/\mathcal{H}_\infty$. Los objetivos se formulan mediante funciones de Lyapunov y el problema se resuelven con LMIs. Un ejemplo práctico es la función de librería de Matlab [GNLC95], donde se considera que el sistema tiene un bloque de entrada $w := w = d$ y dos de salida $[e, z]'$ y se plantea la solución del problema

$$\min_K \alpha\|T_{zw}\|_2 + \beta\|T_{ew}\|_\infty,$$

con $\alpha \geq 0$ y $\beta \geq 0$. Las normas $\mathcal{H}_\infty$ y $\mathcal{H}_2$ permanecen por debajo de unos valores prefijados y los polos de lazo cerrado debe quedar en una región elegida, ver apéndice D

Diseño de control $\ell_1/\frac{\mathcal{H}_2}{\mathcal{H}_\infty}$.

El problema de control óptimo ℓ_1, es decir, la minimización la relación entre los picos de las señales de salida y entrada en discreto, fue formulado por primera vez en [Vid86]. La síntesis de controladores como optimización de esta norma, requiere la resolución de problemas de programación lineal de un tamaño infinito, ver [DDB95, DP87a]. Un problema de programación lineal (LP) viene definido como

$$\begin{array}{ll} \min_x & c^T x \\ \text{s.a} & Ax = b \\ & x_i \geq 0; \;\; i = 1, \ldots, n \end{array} \tag{2.3}$$

donde $x, c \in \mathbb{R}^n$ y $A \in \mathbb{R}^{m \times n}$. En [DDB95] se proponen tres métodos para la solución del problema de programación lineal infinita utilizando la ecuación primal (2.3) y su dual. Estos son: reducción a variables finitas, reducción a restricciones finitas y aumento del retardo.

La optimización $\mathcal{L}_1$, tiempo continuo, es aún más compleja. En [DP87b] se demuestra que el controlador óptimo $\mathcal{L}_1$ para un problema general; con varias variables que regular, entradas y señales de medida; es irracional o de dimensión infinita. En [BS94] se demuestra que a partir de la optimización ℓ_1 se pueden conseguir controladores subóptimos que estén tan cercanos al óptimo $\mathcal{L}_1$ como se desee, aunque con un grado de McMillan muy elevado, ver [ZDG96].

La formulación del problema general $\mathcal{H}_2/\ell_1$, figura 2.2, considera que la señal de entrada w es un ruido blanco y que las señales entrada-salida $d, e \in \mathcal{L}_\infty$, o en discreto $d, e \in \ell_\infty$. El problema queda planteado como

$$\begin{array}{l} \min_K \|T_{zw}\|_2 \\ \text{s.a.} \;\; \|T_{ed}\|_1 < \gamma. \end{array}$$

Y para la optimización mixta $\ell_1/\mathcal{H}_\infty$ se considera que las señales de entrada-salida $w, z \in \mathcal{L}_\infty$, o en discreto $w, z \in \ell_\infty$, y las $d, e \in \mathcal{L}_2$, o en discreto $d, e \in \ell_2$. El problema queda planteado como

$$\begin{array}{l} \min_K \|T_{zw}\|_1 \\ \text{s.a.} \;\; \|T_{ed}\|_\infty < \gamma. \end{array}$$

Una parte de las optimizaciones mixtas $\mathcal{H}_2/\ell_1$ o $\mathcal{H}_\infty/\ell_1$, se basan en formular el problema como una LP donde estén incorporadas las restricciones de una y otra norma. Estos problemas están formulados para el caso SISO, donde la entrada $w := w = d$ y la salida $z := z = e$. En [Vou94] se plantea la optimización $\mathcal{H}_2/\ell_1$ para discreto. En [SDV95] se plantea la optimización $\ell_1/\mathcal{H}_2$, es decir, se minimiza la norma ℓ_1 con restricción en la $\mathcal{H}_2$. En [EDDB93, ED96] se plantean problemas similares.

En [Szn94b] se plantea el problema de optimización mixta $\ell_1/\mathcal{H}_\infty$ para caso MIMO discreto, y en [SB94] el problema $\mathcal{L}_1/\mathcal{H}_\infty$ para caso MIMO continuo. En ambos casos se utiliza la parametrización de Youla del sistema más el controlador. La formulación del problema será

$$\mu^0 = \inf_{Q\in\mathcal{RH}_\infty} \|T_{zw}\|_1 = \inf_{Q\in\mathcal{RH}_\infty} \|V_{11} + V_{12}QV_{21}\|_1$$
$$\text{s.a.}\|T_{ed}\|_\infty = \|T_{11} + T_{12}QT_{21}\|_\infty \leq \gamma$$

donde $\mathcal{RH}_\infty$ es el subespacio de matrices de transferencia racionales estables y propias, ver [ZDG96]. El controlador deberá cumplir que $\|T_{ed}\|_\infty \leq \gamma$ y $\|T_{zw}\|_1 \leq \mu^0 + \varepsilon$, siendo $\varepsilon > 0$. En [RS98, RS99] se plantea la optimización mixta $\ell_1/\mathcal{H}_\infty$ como un problema de optimización convexa.

En [SDV98] se plantean dos optimizaciones $\mathcal{H}_2/\ell_1$ para un sistema MIMO. La primera tiene un bloque de entrada $w := w = d$, y dos de salida $[e, z]'$, y la segunda tiene dos bloques de entrada $[d, w]'$ y uno de salida $z := z = e$. En [SHB95] se formula una aproximación de la optimización $\mathcal{H}_2/\mathcal{L}_1$ a partir de LMIs, y en [HC97] se resuelve el mismo problema a partir de la ecuación de Riccati.

Otras optimizaciones mixtas.

Las posibilidades de optimización mixta no se acaban con las mencionadas. Un ejemplo es la optimización $\mathcal{H}_2/ESPR$ desarrollada en [Shi94]. Este problema es una variación del problema $\mathcal{H}_2/\mathcal{H}_\infty$ propuesto en [KR91]. A partir de sistema de la figura 2.2, con un bloque de entrada $w : w = d$ y dos de salida $[e', z']'$ se desea

$$\min_K \|T_{zw}\|_2$$
$$\text{s.a. } T_{ew} \in \text{ESPR}.$$

Un sistema $G(s)$ pertenece a ESPR (Real positiva estrictamente extendida) si 1) todos sus elementos son analíticos para $Re[s] > 0$, y 2) $G(s) + G(s)^*$ es definida no negativa para $Re[s] > 0$ y 3) $D + D' > 0$, donde $D = G(\infty)$.

En [YD95] se optimiza ℓ_1/LMI objetivos, es decir la norma $\|T_{zw}(z)\|_1$ con unas restricciones en forma de inecuaciones lineales matriciales. Finalmente, en [MGL91, YS97], con un bloque de entrada $w := w = d$ y otro de salida $z := z = e$, se optimiza una función entropía de T_{zw} con una restricción en la norma $\|T_{zw}\|_\infty$. La entropía de $T_{zw}(s)$ viene definida como

$$E(T_{zw}, \gamma) := -\frac{\gamma^2}{2\pi}\int_{-\infty}^{\infty} \ln(\det[I - \gamma^{-2}T_{zw}^*(jw)T_{zw}(jw)])dw.$$

Conclusión.

Los principales inconvenientes que se pueden atribuir al control mixto por los métodos propuestos en este punto son los siguiente.

- La formulación de problemas mixtos se plantean de forma general como la optimización de una norma con una restricción. La solución del problema es un único controlador, siendo necesaria la repetición del problema para distintas restricciones con el fin de obtener todos los controladores posibles, ver sección 3.3.
- La solución óptima es muy compleja, siendo preciso en la mayoría de los casos obtener un límite superior.
- El orden del controlador es mayor que el de la planta y en muchos casos infinito. Es, por tanto, preciso reducir el mismo para poderlo implantar físicamente. Este hecho lleva consigo un fenómeno asociado de fragilidad, ver [KB97].
- En la resolución del problema no se pueden introducir restricciones respecto a estructura o valor del los parámetros del controlador deseado.
- El número máximo de normas a optimizar será de dos, siendo muy complicado el planteamiento de nuevos problemas.

2.3.2 Incertidumbre estructurada.

La optimización $\mathcal{H}_\infty$ es conservadora porque considera que no se tiene ninguna información sobre la estructura de la incertidumbre; hecho que muchas veces, como en el caso de la incertidumbre paramétrica, no es cierto. Esta razón ha llevado al estudio de diferentes técnicas que tratan de subsanar este problema.

Valor singular estructurado.

Esta técnica va a considerar que la incertidumbre del sistema, puesto como modelo LFT, figura 2.1, es una matriz diagonal por bloques de números complejos.

$$\begin{gathered}\boldsymbol{\Delta} = \{diag[\delta_1, I_{k_1}, \ldots, \delta_S I_{k_S}, \Delta_1, \ldots, \Delta_F] : \delta_i \in \mathbb{C}, \Delta_i \in \mathbb{C}^{k_{S+i} \times k_{S+i}}\} \\ \mathbf{B}\boldsymbol{\Delta} = \{\Delta \in \boldsymbol{\Delta} : \overline{\sigma}(\Delta) \leq 1\}\end{gathered} \tag{2.4}$$

siendo S y F dos número enteros no negativos.

Definición 2.1 *Para una $M \in \mathbb{C}^{n\times n}$, $\mu_{\mathbf{\Delta}}(M)$ está definido como*

$$\mu_{\mathbf{\Delta}}(M) := \frac{1}{\min\{\overline{\sigma}(\Delta) : \Delta \in \mathbf{\Delta}, det(I - M\Delta) = 0\}}$$

cuando $\Delta \in \mathbf{\Delta}$ no hace que $I - M\Delta$ sea singular, $\mu_{\mathbf{\Delta}} := 0$ en caso de que si que sea singular.

De esta definición se obtiene que $\mu_{\mathbf{\Delta}}(M) = \max_{\Delta \in \mathbf{B\Delta}} \rho(M\Delta)$, donde ρ es el radio espectral, ver [ZDG96, DPZ91]. El cálculo de $\mu_{\mathbf{\Delta}}(\cdot)$ se realiza por medio de sus límites inferior y superior,

$$\rho(M) \leq \mu_{\mathbf{\Delta}}(M) \leq \overline{\sigma}(M),$$

siendo $\overline{\sigma}$ el máximo valor singular.

$$\text{Si } \mathbf{\Delta} = \{\delta I : \delta \in \mathbb{C}\}(S = 1, F = 0, k_1 = n) \text{ entonces } \mu_{\mathbf{\Delta}}(M) = \rho(M)$$
$$\text{Si } \mathbf{\Delta} = \mathbb{C}^{n\times n}(S = 0, F = 1, k_1 = n) \text{ entonces } \mu_{\mathbf{\Delta}}(M) = \overline{\sigma}(M).$$

Para conseguir reducir estos márgenes, se define los siguientes conjuntos de matrices

$$\mathbf{Q} = \{Q \in \mathbf{\Delta} : Q^*Q = I_n\}$$
$$D = \left\{ \begin{array}{c} diag[D_1, \ldots, D_S, d_1 I_{m_1}, \ldots, d_{F-1} I_{m_{F-1}}, I_{m_F}] : \\ D_i \in \mathbb{C}^{r_i \times r_i}, D_i = D_i^* > 0, d_j \in \mathbb{R}, d_j > 0 \end{array} \right\}$$

Y como para cualquier $\Delta \in \mathbf{\Delta}$, $Q \in \mathbf{Q}$ y $D \in \mathbf{D}$ se cumple

$$\begin{gathered} Q^* \in \mathbf{Q} \\ Q\Delta \in \mathbf{\Delta} \\ \Delta Q \in \mathbf{\Delta} \\ \overline{\sigma}(Q\Delta) = \overline{\sigma}(\Delta Q) = \overline{\sigma}(\Delta) \\ D\Delta = \Delta D \end{gathered}$$

se puede afirmar que

$$\max_{Q \in \mathbf{Q}} \rho(QM) \leq \mu_{\mathbf{\Delta}}(M) \leq \inf_{D \in \mathbf{D}} \mu_{\mathbf{\Delta}}(DMD^{-1}).$$

El límite inferior es una igualdad, pero $\rho(UM)$ tiene muchos máximos locales, teniendo que buscarse la solución para cada estructura concreta de Δ, ver [ZDG96].

La optimización μ al igual que la optimización $\mathcal{H}_\infty$ tienen como objetivo la estabilidad robusta del sistema, siendo poco efectiva en la reducción del efecto de ruidos. La razón es que la optimización μ y $\mathcal{H}_\infty$ intenta reducir el pico en frecuencia del sistema a costa de aumentar la sensibilidad en un gran margen de frecuencias, lo que provoca su ineficiencia para reducir ruidos. Por ello, en [Pag99a, Pag99b] se estudia la optimización $\mathcal{H}_2$ robusta en tiempo continuo y en frecuencia, respectivamente, de un sistema con incertidumbre estructurada. Se trata de conseguir especificaciones en forma de normas $\mathcal{H}_2$ y condiciones de robustez en plantas con incertidumbre estructurada.

Incertidumbre paramétrica.

Este tipo de incertidumbre es un caso particular de la incertidumbre definida en la ecuación (2.4), donde la matriz Δ es diagonal y real, ver sección A.5. Muchos de los problemas de incertidumbre paramétrica se pueden catalogar como de incertidumbres lineales afines o multilineales afines respecto de los coeficientes del polinomio característico del sistema, ver [Bar94].

Definición 2.2 *Se define el polinomio característico de un sistema con incertidumbre como*

$$p(s,q) = \sum_{i=0}^{n} a_i(q)s^i,$$

siendo $q = [q_1, \ldots, q_n], \quad n \in \mathbb{Z}^+$ *el vector de incertidumbre. Un sistema tiene incertidumbres lineales afines cuando los coeficientes de polinomio característico tienen la forma* $a_i(q) = \alpha_i^T q + \beta_i$. *Un sistema tiene incertidumbres multilineal si fijadas todas las incertidumbres menos una los coeficientes del polinomio característico son lineales afines respecto de la incertidumbre variable.*

El estudio de la estabilidad y diseño de sistemas con incertidumbre paramétrica de este tipo se realiza con ayuda del "Teorema de Kharitonov" y el "Teorema de las aristas" ("Egde Theorem") entre otros, ver [Bar94]. Estos teoremas se basan en que los valores extremos del polinomio característico, para una s dada, corresponden con alguno de los extremos del espacio de las incertidumbres. En cualquier caso, la aplicación práctica de este tipo de técnicas sólo resulta válida cuando el orden de la incertidumbre es bajo. En [Ack93] se trabaja con sistemas de este tipo para conseguir controladores con varios objetivos.

El cálculo de μ en sistemas de incertidumbre paramétrica puede ser planteado de forma pareja al caso general de incertidumbre estructurada. Sin embargo, existe una diferencia fundamental en el grado de complejidad computacional. Como se demuestra en [BYDM93], el calculo de μ en un problema con incertidumbre real pura o incertidumbre mixta real/compleja es NP-hard, ver [GJ79, Vid98]. No es éste el único problema de control que tiene este tipo de complejidad computacional, ver [Vid98, Nem93, BT97]. La segunda propiedad de μ para el caso de incertidumbre real es su continuidad, mientras que no lo es en el caso general. En [YND91] se estudia la obtención de las cotas superior e inferior de μ para el caso de incertidumbre real, aplicándose el algoritmo "Branch and Bound", ver sección 3.4.

Otra forma de tratar los problemas de análisis y diseño de sistemas con incertidumbre paramétrica son los métodos estocásticos robustos. El problema de análisis de sistemas se realiza por medio de "Simulación de Monte Carlo" en el dominio de la incertidumbre, ver [Ray91, RS93]. Estos métodos tratan de obtener la probabilidad de que el sistema con su incertidumbre cumplan con una determinada propiedad, ver [Vid97, TBD96]. Los problemas de diseño se plantean en la forma general (2.2), es decir buscar el mejor controlador para el peor caso de incertidumbre. En [MS94, Mar95, MS95] se usan técnicas estocásticas robustas para analizar el mejor controlador de varios propuestos, hallados por otros métodos.

Conclusión.

El control mixto en problemas con incertidumbre parámetrica está muy poco desarrollado. La resolución es muy compleja por dar lugar a problemas en muchos casos computacionalmente irresolubles. Alguno autores ha intentado su estudio a partir de técnicas estocásticas dando pie al llamado control robusto estocástico, aunque incluso en este caso no han tratado de forma directa el problema mixto.

2.4 Formulación paramétrica del diseño multiobjetivo y el diseño robusto multiobjetivo.

El problema de diseño multiobjetivo de una planta nominal (2.1) es muy difícil de resolver planteado como un problema en ecuaciones de estado, o como un problema de programación lineal infinita. Además, su resolución implica reducir el problema de optimización multiobjetivo correspondiente, cual solución es un conjunto de controladores, a uno de optimización simple con restricciones, cuya solución es un controlador, ver sección 3.3. La obtención de todas las soluciones del problema implica repetir el proceso variando las restricciones.

El planteamiento utilizado este trabajo es formular el problema de diseño multiobjetivo como un problema de optimización de los diferentes objetivos respecto a los parámetros de controlador, cuya estructura ha sido elegida a priori. Tras fijar la estructura del controlador dependiente de un número de parámetros finito N, $K(s, k_1, \ldots, k_N)$ donde $k_i \in \mathbb{R}$, el problema se plantea como,

$$\min_{\{k_1,\ldots,k_N\}} \begin{bmatrix} \|W_1[\mathcal{F}_l(G_n, K(k_i))]_{ind_1} V_1\|_{p_1} \\ \vdots \\ \|W_n[\mathcal{F}_l(G_n, K(k_i))]_{ind_n} V_n\|_{p_n} \end{bmatrix}, \tag{2.5}$$

donde $V_i(s)$ y $W_i(s)$ son los filtros, $[\cdot]_{ind_i}$ es el bloque de la matriz de transferencia elegido y p_i la norma que se aplicará.

Este planteamiento evita las incompatibilidades entre las normas, ya que en él éstas no son más que funciones de los parámetros k_i. Además se puede fijar la estructura del controlador y su orden, así como introducir restricciones en el valor de sus parámetros. Es, por tanto, un problema con una estructura mucho más abierta que los problemas mixtos clásicos.

El problema de control robusto multiobjetivo va a ser planteado de forma similar. Se parte de una planta con una incertidumbre paramétrica, es decir dependiente de un conjunto de parámetros finito $\Delta(\delta_1, \ldots, \delta_M)$, con $\delta_i \in \mathbb{R}$. El controlador en realimentación tiene una estructura fijada dependiente de un número también finito de parámetros $K(s, k_1, \ldots, k_N)$, con $k_i \in \mathbb{R}$. Se busca la combinación de parámetros del controlador que minimizan los objetivos planeados en la peor combinación posible de los parámetros de la planta,

$$\min_{\{k_1,\ldots,k_N\}} \max_{\{\delta_1,\ldots,\delta_M\}} \begin{bmatrix} \|W_1[\mathcal{F}_l(\mathcal{F}_u(G_n, \Delta(\delta_i)), K(k_j))]_{ind_1} V_1\|_{p_1} \\ \vdots \\ \|W_n[\mathcal{F}_l(\mathcal{F}_u(G_n, \Delta(\delta_i)), K(k_j)]_{ind_n} V_n\|_{p_n} \end{bmatrix}, \tag{2.6}$$

donde $V_i(s)$ y $W_i(s)$ son los filtros, $[\cdot]_{ind_i}$ es el bloque de la matriz de transferencia elegido y p_i la norma que se aplicará. Por este procedimiento se va a poder abordar un problema que hasta la fecha ha sido muy poco tratado debido a su complejidad.

El planteamiento de ambos problemas precisa de una herramienta potente capaz de conseguir la convergencia de N parámetros del controlador en el primer caso, y N parámetros del controlador y M de la incertidumbre en el segundo. La herramienta elegida ha sido los algoritmos genéticos multiobjetivo, que serán tratados en el capítulo 3.

2.5 Conclusión.

La teoría de control moderna se formula como la optimización de normas de funciones de transferencia entre diferentes señales de entrada y salida del sistema en lazo cerrado. Como normalmente se desean varios objetivos, el problema es multiobjetivo lo que implica que su solución es un conjunto de controladores. Los métodos que optimizan la norma $\mathcal{H}_2$ y $\mathcal{H}_\infty$ utilizan el espacio de estado, mientras que los que minimizan la norma $\mathcal{L}_1$ usan programación lineal infinita, por lo que su mezcla en un problema mixto es complicada. La optimización de problemas mixtos de dos normas es muy complejo, siendo necesario reducirlo a la búsqueda de un límite superior. Los problemas mixtos planteados con estos métodos reducen el problema multiobjetivo a uno de optimización simple con restricciones, donde sólo se obtiene un controlador. Los controladores obtenidos son de orden muy alto y no es posible fijar restricciones en su estructura o valor de sus parámetros.

En esta memoria se plantea el control multiobjetivo como un problema de optimización multiobjetivo con respecto a los parámetros del controlador, con una estructura definida previamente, y el control robusto multiobjetivo como un problema de min-max entre los parámetros del controlador y los de la incertidumbre. Ambas formulaciones son totalmente flexibles tanto en objetivos como restricciones. Las herramientas elegida para su resolución han sido los algoritmos genéticos multiobjetivo, ver capítulo 3.

Capítulo 3

Optimización multiobjetivo usando Algoritmos Evolutivos (EAs).

3.1 Introducción.

La formulación matemática de muchos problemas en diferentes áreas de conocimiento viene dada por la resolución de un problema de optimización. En caso de que la función sea convexa, la solución puede alcanzarse por muchos métodos de cálculo, por ejemplo el método del gradiente, pero si la función no es convexa estos métodos sólo llegan a un óptimo local, que depende del punto inicial elegido en la búsqueda. Este hecho ha llevado al desarrollo de muchos algoritmos, deterministas y estocásticos, que trata de obtener con la máxima probabilidad posible el óptimo global del problema. Dentro de ellos merecen una especial atención los algoritmos evolutivos (EAs), inicialmente propuestos por Holland [Hol92] en la década de los sesenta. Los algoritmos genéticos (GAs) son los EAs más ampliamente usado en este trabajo. Estos son algoritmos de muestreo dirigido, tales que en un primer paso muestrean el espacio de búsqueda de forma uniforme en busca de posibles soluciones. Tras examinar el valor de los puntos muestreados respecto a los objetivos, se muestrea de nuevo el espacio de búsqueda pero teniendo en cuenta los resultados obtenidos en el muestreo anterior. Los EAs utilizan técnicas que simulan la evolución natural de las especies para realizar este muestreo dirigido. Sus principales operadores son la selección, cruce y mutación. El desarrollo de estos algoritmos y su aplicación a distintos campos de conocimiento ha sido enorme en las últimas dos décadas, viéndose reflejado en gran cantidad de congresos, libros, artículos en revistas y aplicaciones prácticas de todo tipo, como se verá en la tabla 3.1.

Cuando la formulación matemática de un problema se plantea como la optimización de varios objetivos a la vez, el problema se denomina optimización multiobjetivo. Los objetivos a optimizar normalmente están en contradicción, ya que sino serían redundantes. La resolución de este problema no es un único punto, como en el caso de la optimización global, sino un conjunto de puntos que se llamarán conjunto de óptimos de Pareto.

Los métodos convencionales para resolver el problema de optimización multiobjetivo pasan por reducirlo a un problema de optimización global. Esto lleva a obtener una sola solución del conjunto de óptimos de Pareto, que depende en buena medida de las suposiciones dadas a priori sobre la forma de dicha solución.

Una de las propiedades de los EAs es el proceso de búsqueda en paralelo, ya que su dinámica se puede interpretar como la convergencia en paralelo de un conjunto de posibles soluciones hacia un objetivo. Numerosos autores han desarrollado operadores para convertir los originales EAs, en especial los algoritmos genéticos (GAs), en algoritmos que converjan hacia un conjunto de soluciones, ver el capítulo 3 de [Mah95] y sus referencias. Todos estos operadores intentan obtener diversidad en la solución. Los algoritmos con diversidad han sido utilizados para la búsqueda del conjunto de óptimos de Pareto en un problema multiobjetivo. Los buenos resultados obtenidos con estos algoritmos han hecho que su uso sea cada vez mayor, constituyendo en nuestros días unas de las ramas de los EAs con más desarrollo.

El control moderno formula el diseño de controladores como un problema de optimización de una serie de normas de otras tantas funciones de transferencia del sistema en lazo cerrado. En la sección 2.4 se ha definido el problema de diseño multiobjetivo como un problema de minimización de los objetivos respecto a los parámetros del controlador, cuya estructura ha sido fijada a priori, ver ecuación (2.5). Así mismo, el diseño robusto multiobjetivo de una planta con incertidumbre paramétrica ha sido definido como un problema de min-max entre los parámetros del controlador y los parámetros de la incertidumbre, ver ecuación (2.6). La herramienta propuesta en este trabajo para resolver ambos problemas va a ser los algoritmos genéticos (GAs) multiobjetivo. Algunos autores han aplicado estos algoritmos en casos particulares del problema (2.5), principalmente para la sintonización de PID en sistemas SISO, con resultados muy positivos, [CF96, CF95, MTKH97]. El problema (2.6) aún no ha sido abordado.

Este capítulo está organizado de la siguiente forma. En la sección 3.2 se define el concepto de óptimo global para problemas simples y óptimo de Pareto para problemas multiobjetivo. En la sección 3.3 se estudian los métodos clásicos de resolución de un problema multiobjetivo, que consisten básicamente en la conversión del mismo a una optimización global. En la sección 3.4 se explican diferentes algoritmos de optimización global. En la sección 3.5 se describen los algoritmos evolutivos (EAs) como un caso especial de algoritmos estocásticos. En la sección 3.6 se analizan diferentes modificaciones a los EAs simples para conseguir diversidad en la solución. En la sección 3.7 se analizan los EAs aplicados en la optimización multiobjetivo, los cuales usan operadores para conseguir diversidad. En la sección 3.8 se describe el desarrollo de los EAs en las dos últimas décadas, viendo con mayor atención las aplicaciones a problemas multiobjetivo y a problemas de control robusto. En la sección 3.9 se dan diversas conclusiones sobre la posibilidades de los EAs en la optimización global y multiobjetivo.

3.2 Optimización global y optimización multiobjetivo (MO).

La optimización global es el proceso que encuentra el mínimo (o máximo) global en un espacio de búsqueda. En [Bäc96] se define formalmente la optimización global como:

Definición 3.1 *Dada una función $f(\cdot) : \Omega \subseteq \mathbb{R}^k \mapsto \mathbb{R}, \Omega \neq \emptyset$, donde $x \in \Omega$, el valor $f^* := f(x^*) > -\infty$ es llamado mínimo global si y solo si*

$$\forall x \in \Omega : f(x^*) \leq f(x).$$

Entonces, x^* es la solución global mínima, $f(\cdot)$ es la función objetivo, y el conjunto Ω es el espacio de búsqueda .

A partir de esta definición se puede concluir que el problema de optimización global de un simple objetivo tiene una solución única en el espacio del objetivo.

Un problema de optimización multiobjetivo (MOP) queda definido como:

Definición 3.2 *En general, un MOP minimiza un vector de funciones $f(\cdot)$ respecto de un vector variable x,*

$$\min_x f(x)$$

donde $f(x) = [f_1(x), \ldots, f_n(x)]$ y $x = [x_1, \ldots, x_k]$. La solución de un MOP minimiza los componentes de vector $f(x) \in \Lambda \subseteq \mathbb{R}^n$ donde $x \in \Omega \subseteq \mathbb{R}^k$.

Por tanto, el problema tiene k variables de decisión, m restricciones, n objetivos y las funciones $f(\cdot) : \Omega \mapsto \Lambda$ es la correspondencia entre el espacio de búsqueda Ω y el espacio de las funciones objetivo Λ. De la definición de MOP se infiere que la solución del problema no es única ya que los diversos objetivos pueden estar en conflicto con lo que la minimización de uno de ellos lleva a la maximización del otro. De hecho, la optimización global de un problema multiobjetivo es un problema NP-completo, ver [Bäc96, GJ79]. Sea, por ejemplo, el problema de minimización de una función de dos dimensiones $f(x) \in \Lambda \subseteq \mathbb{R}^2$ en un espacio de búsqueda $x \in \Omega \subseteq \mathbb{R}^k$. Los cuatros puntos A, B, C y D dibujados en la figura 3.1 son cuatro posibles soluciones del problema. A la hora de elegir cual de ellos es mejor sólo podemos afirmar que el punto D es peor que el B por ser mayor que él en todas los componentes, pero los puntos A, B y C no son comparables. Esta especial característica de la optimización multiobjetivo es lo que lleva al desarrollo de la optimización de Pareto. A continuación se definen, ver [Gol89a, VV99a], varios conceptos esenciales en la optimización de Pareto.

Definición 3.3 (Pareto Dominante) *En un problema de minimización, un vector $u = [u_1, \ldots, u_n] \in \Lambda \subseteq \mathbb{R}^n$ domina a otro vector $v = [v_1, \ldots, v_n] \in \Lambda \subseteq R^n$, se denomina $u \preceq v$, si y sólo si u es parcialmente menor que v.*

$$u \preceq v \iff i = \{1, \ldots, n\} : (\forall i)(u_i \leq v_i) \wedge (\exists i)(u_i < v_i)$$

Definición 3.4 (Óptimo de Pareto) *Una solución del espacio de búsqueda, $x \in \Omega$ es óptimo de Pareto con respecto de este espacio (Ω) si y sólo si no existe otro $x' \in \Omega$ tal que $v = f(x')$ domine a $u = f(x)$.*

Definición 3.5 (Conjunto de óptimos de Pareto) *Dado un MOP con un vector de funciones $f(\cdot)$, el conjunto de óptimos de Pareto ($\mathcal{P}^*$) se define como:*

$$\mathcal{P}^* := \{x \in \Omega | \ (\neg\exists x' \in \Omega)(f(x') \preceq f(x))\}.$$

Definición 3.6 (Frente de Pareto) *Para un MOP con un vector de funciones $f(\cdot)$ y un conjunto de óptimos de Pareto ($\mathcal{P}^*$), el frente de Pareto ($\mathcal{PF}^*$) se define como:*

$$\mathcal{PF}^* := \{u = f(x) | \ x \in \mathcal{P}^*\}.$$

Definición 3.7 (Mínimo global de un MOP) *Dado un vector de funciones $f(\cdot) : \Omega \subseteq \mathbb{R}^k \mapsto \mathbb{R}^n,\ \ \Omega \neq \emptyset, k \geq 2$, el conjunto $\mathcal{PF}^* := f(x^*)$ es llamado mínimo global si y sólo si*

$$\forall x \in \Omega : f(x^*) \preceq f(x),$$

donde $x^* \in \Omega$ es llamado conjunto de soluciones de mínimo global, $f(\cdot)$ es el vector de funciones objetivos y Ω el espacio de búsqueda.

A raíz de estas definiciones se deduce que el mínimo global de un MOP es el frente de Pareto de dicho problema y las soluciones de mínimo global forman el conjunto de óptimos de Pareto. Un problema de optimización de Pareto donde el espacio de búsqueda sea $\mathbb{R}^k$ tiene, en general, infinitas soluciones. Se puede restringir el espacio de búsqueda a un número finito de elementos, por ejemplo, en el problema planteado en la figura 3.1, siendo el espacio de búsqueda $\Omega = \{x_A, x_B, x_C, x_D\}$ el conjunto de óptimos de Pareto es $\mathcal{P}^* = \{x_A, x_B, x_C\}$, y el frente de Pareto $\mathcal{FP}^* = \{f(x_A), f(x_B), f(x_C)\}$. Esto se deduce debido a que $f(x_B)$ domina a $f(x_D)$ con lo que este elemento (x_D) no es un óptimo de Pareto.

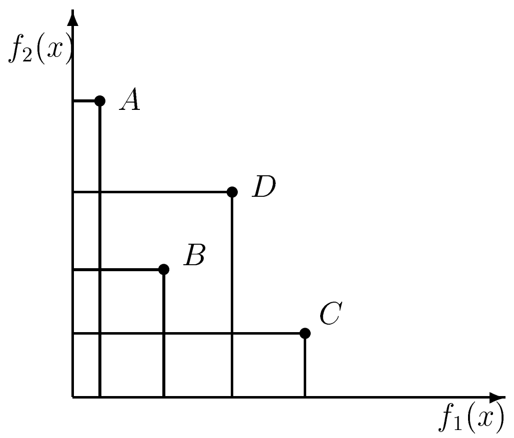


Figura 3.1: Problema optimización de una función dos dimensiones.

3.3 Métodos clásicos de resolución de un problema de optimización multiobjetivo (MOP).

Los métodos clásicos de resolución de un problema de optimización multiobjetivo (MOP) tienden a convertir dicho problema en otro de optimización global.

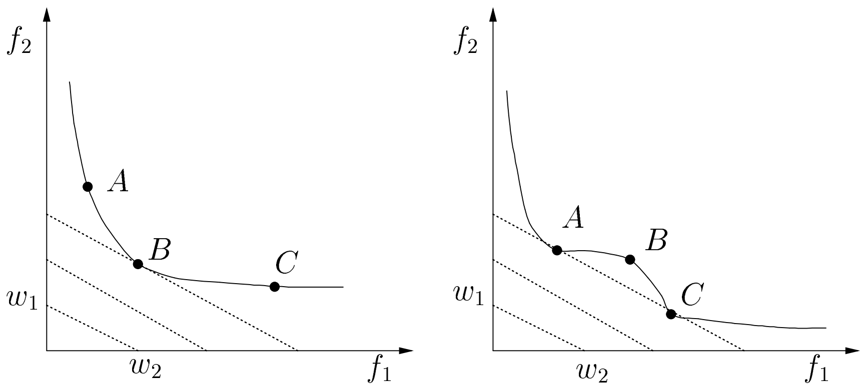


Figura 3.2: Optimización función suma con frente de Pareto convexo y no convexo.

3.3.1 Método de optimización de suma con pesos.

En este método el MOP es convertido en un problema de minimización de una suma convexa de funciones objetivo

$$\begin{array}{c} \min_x f_{sum} = \sum_{i=1}^{n} w_i \cdot f_i(x) \\ x \in \Omega \end{array} \tag{3.1}$$

donde $\{w_i \geq 0, \sum_{i=1}^{n} w_i = 1\}$, ver [BB91]. La solución de este problema es un punto del frente de Pareto, función de los pesos, $\{w_i, \quad i = 1, \ldots, n\}$ elegidos. Este método sólo es útil cuando el frente de Pareto es convexo, siendo complicada la elección de pesos a priori. Supongamos que el MOP es de dos dimensiones como los planteados en la figura 3.2. En esta figura se plantea un problema con el frente de Pareto convexo y otro con el frente de Pareto no convexo. La solución del problema por este método es el punto intersección entre la recta de pendiente $(\pi/2 + \arctan(\frac{w_1}{w_2}))$ con el frente de Pareto. Para el caso convexo la intersección es el punto B, pero para el caso no convexo serán los puntos A y C, no pudiéndose encontrar nunca el punto B con este método.

3.3.2 Método de optimización de máximo con pesos.

En este método se debe efectuar una optimización de "min-max", ver [BB91]. Se define un vector formado por los términos del vector de funciones objetivo multiplicados diferentes pesos. El MOP se convierte en la minimización con respecto al vector x del máximo valor del vector definido, para unos pesos dados a priori,

$$\min_{x \in \Omega} \max\{f_1(x) \cdot w_1, \ldots, f_n(x) \cdot w_n\}, \tag{3.2}$$

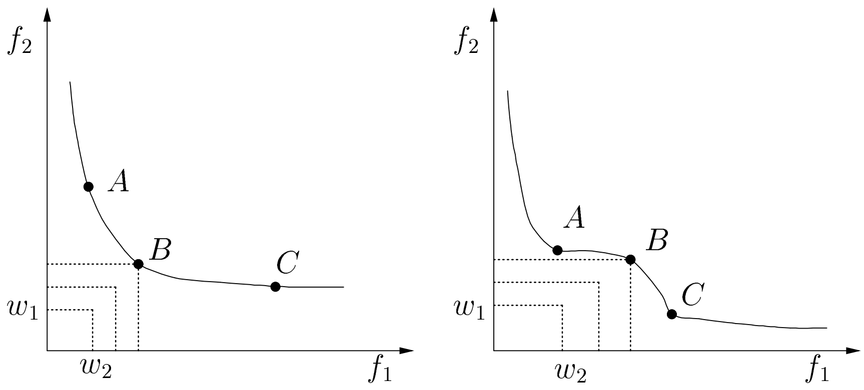


Figura 3.3: Optimización "min-max" con frente de Pareto convexo y no convexo.

donde $\{w_i \geq 0, \ \sum_{i=1}^{n} w_i = 1\}$. Un ejemplo de este método para el caso de dos dimensiones con un frente de Pareto convexo y no convexo es el mostrado en la figura 3.3. La solución del problema es el corte de la recta de pendiente $(\arctan(\frac{w_1}{w_2}))$ con el frente de Pareto. Este método sirve tanto para frentes de Pareto convexo como no convexo, pero su realización implica un problema de "min-max". Otra variación de este método se produce al fijar un vector constante que se resta al vector de funciones objetivos, de forma que la formulación del problema queda

$$\min_{x \in \Omega} \max\{(f_1(x) - \gamma_1) \cdot w_1, \ldots, (f_n(x) - \gamma_n) \cdot w_n\}. \tag{3.3}$$

3.3.3 Método de perturbación ϵ.

Este método reduce el MOP a un problema de optimización para un simple objetivo con restricciones

$$\begin{array}{ll} \min_x & f_k(x) \\ \text{s.a.} & \quad f_j(x) \leq \epsilon_j \quad \forall j \neq k \\ & x \in \Omega. \end{array} \tag{3.4}$$

Este método sirve para un frente de Pareto convexo o no convexo, y la figura 3.4 muestra un ejemplo de su aplicación en el caso de dos dimensiones

$$\begin{array}{ll} \min_x & f_2(x) \\ \text{s.a.} & f_1(x) \leq \epsilon \\ & x \in \Omega. \end{array}$$

Es éste, sin duda, el método más utilizado en el diseño clásico de controladores multiobjetivos, tal y como se muestra en la sección 2.3.

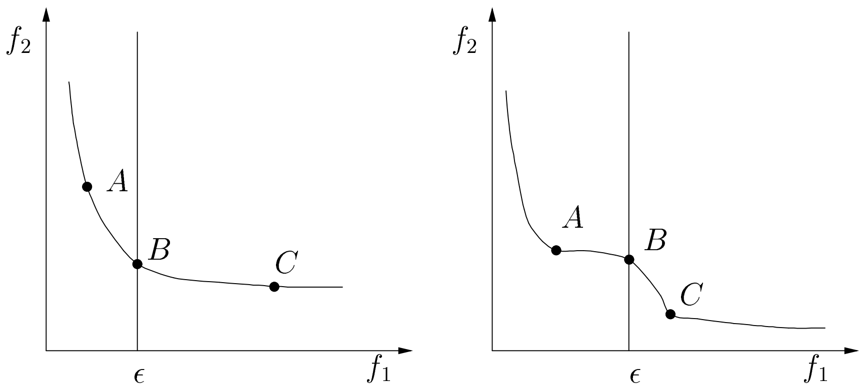


Figura 3.4: Optimización ϵ de frente de Pareto convexo y no convexo.

3.3.4 Aproximación a un punto utópico.

En este método se considera un punto utópico (f^*) al que se desea llegar con la función objetivo. El problema se replantea como

$$\min_{x\in\Omega} \|f^* - f(x)\| \tag{3.5}$$

donde se elige una métrica, la euclídea por ejemplo, como medida. Tiene el inconveniente de tener que definir a priori el punto utópico.

3.3.5 Soluciones óptimas lexicográficas.

Este método convierte el problema de optimización de Pareto, donde todas las componentes del vector de funciones $f(\cdot)$ tienen la misma importancia, en un problema donde las componentes de dicho vector tienen una importancia decreciente, ver [BT80]. Así, el problema de optimización lexicográfica se formula como una secuencia de n minimizaciones, P_i

$$\begin{array}{ll} (P_1) & \min_{x\in\Omega} f_1(x) = \alpha_1 \\ (P_2) & \min_{x\in\Omega}\{f_2(x) \;\mid\; f_1(x) \le \alpha_1\} = \alpha_2 \\ \vdots & \vdots \\ (P_n) & \min_{x\in\Omega}\{f_n(x) \;\mid\; f_1(x) \le \alpha_1, \ldots, f_{n-1}(x) \le \alpha_{n-1}\}. \end{array} \tag{3.6}$$

La formulación del problema (P_n) puede escribirse como

$$\begin{array}{l} \min_{x\in\Omega} f_n(x) \\ \text{s.a.} \quad f_k(x) \le \alpha_k \quad k = 1, \ldots, n-1. \end{array}$$

La formulación de este problema rompe el espíritu del un verdadero MOP y sólo es aplicable a los casos donde el orden de prioridad de los objetivos esté claro.

3.3.6 Conclusión.

Varias son las causas que hacen que estos métodos no sean eficaces para hallar el conjunto de óptimos de Pareto de un MOP.

- Estos algoritmos sólo dan un punto del frente de Pareto como solución, siendo necesaria la repetición del algoritmo un gran número de veces para obtener todas la soluciones.
- Muchos de estos algoritmos necesitan un conocimiento previo del problema para ser viables. Es el caso del valor de las funciones peso o del punto utópico.
- Alguno de estos algoritmos son muy sensibles a la forma del frente de Pareto.
- Estos algoritmos no son válidos en problemas con incertidumbre o problemas estocásticos.
- Los algoritmos de optimización clásicos no son válidos para resolver problemas de un sólo objetivo con un espacio de búsqueda discreto, luego tampoco lo serán para resolver MOP con espacio de búsqueda discreto.

Estas razones son algunas de las causas que han potenciado el desarrollo de algoritmos evolutivos (EAs), en especial algoritmos genéticos (GAs) para la resolución de MOPs, como se mostrará a largo de este capítulo.

3.4 Algoritmos de optimización global.

El problema de optimización global de un simple objetivo, definido en 3.1, puede resolverse por algoritmos enumerativos, determinísticos y estocásticos. En esta sección se dará una visión general de estos métodos, dejando para el siguiente punto el estudio de los algoritmos estocásticos que se usan en este trabajo, los algoritmos evolutivos (EAs).

3.4.1 Algoritmos enumerativos.

Estos algoritmos consisten en la división del espacio de búsqueda en un número finito de partes y el estudio de la función en cada uno de las partes divididas. La idea es muy simple pero no es eficiente en caso de un espacio de búsqueda de dimensiones elevado. Un ejemplo de este tipo de métodos es el algoritmo de "Búsqueda en Rejilla (Grid Search)", ver [Tim84]. Este algoritmo consiste en el mallado del espacio a estudiar y el análisis del mismo en los puntos de cruce de dicha red. De esa forma, al contrario que con una búsqueda aleatoria pura, se tiene un espectro continuo de todo el espacio de búsqueda. Esta técnica normalmente va acompañada de otras que analizan con más profundidad las zonas del espacio de búsqueda donde la función a optimizar tiene mejor valor. Es el caso del método "Múltiples puntos iniciales (Multi-start)", que busca óptimos locales a partir de buenos puntos subóptimos iniciales obtenidos con otros métodos, por ejemplo con el método "Grid Search".

3.4.2 Algoritmos deterministas.

Algoritmos basados en cálculo.

Los métodos basados en cálculo utilizan el concepto de gradiente, ya sea de forma directa o indirecta, ver [Gol89a]. De forma indirecta si calculan los puntos del espacio de búsqueda donde la derivada de la función a optimizar es nula, hallando el óptimo global como el mayor óptimo local. De forma directa si analizan el gradiente de la función a partir de un punto inicial, hallando el camino hasta el óptimo local. Un ejemplo de este segundo tipo es el algoritmo "Escalando Colinas (Climbing-Hill)", cuyo pseudocódigo está en la figura 3.5. Se parte de un punto aleatorio $x_c \in \Omega \subseteq \mathbb{R}^k$ y se selecciona N puntos cercanos a x_c. La función $S(x)$ representa el conjunto de puntos cercanos a x, con una métrica como medida de la distancia. Se elige de ellos el que mejor función objetivo tenga y se compara con el valor $f(x_c)$. Si se ha conseguido mejorar la función objetivo el proceso continúa, sino se considera que se ha llegado a un óptimo local. Este bucle se repite $Tmax$ veces para conseguir otros tantos óptimos locales, y se considera óptimo global al máximo de ellos.

Algoritmo "Ramificar y Podar (Branch and Bound)".

Otra técnica es el algoritmo "Branch and Bound" para minimizar. Este método está basado en la búsqueda de los mínimos de las cotas superior e inferior de la función para distintas particiones hechas en el espacio de búsqueda, ver pseudocódigo en figura 3.6. Se divide al espacio de búsqueda Ω en un conjunto de particiones $\{\Delta_i\}$ y se hallan los mínimos de las cotas inferior y

```
inicio
  t ← 0
  repetir
    local ← FALSE
    seleccionar un punto x_c aleatoriamente
    evaluar f(x_c)
    repetir
      seleccionar {x_1, ..., x_N} ∈ S(x_c)
      f(x_n) = min_{i=1,...,N} f(x_i)
      si f(x_n) < f(x_c), x_c ← x_n
      sino local ← TRUE
    hasta local
    t ← t + 1
  hasta (t > Tmax)
fin
```

Figura 3.5: Pseudocódigo del algoritmo "Climbing-Hill".

superior de la función $f(\cdot)$ del conjunto de dichas particiones. Se considera que ambos mínimos de $f(\cdot)$ en una partición Δ, $UB(f(\Delta))$ y $LB(f(\Delta))$, están en los extremos de la misma. Las particiones que tengan mínima cota inferior se vuelven a dividir y se repite el proceso hasta que la diferencia entre los dos mínimos, superior e inferior, sean menor que un valor ϵ dado. El mínimo global de la función se considera que está entre los dos mínimos, superior e inferior, hallados.

```
inicio
  {Δ_i} ← Δ_1
  UB ← min_i UB(f(Δ_i))
  LB ← min_i LB(f(Δ_i))
  repetir
    Tomar particiones Δ ∈ {Δ_i} con LB(f(Δ)) = LU
    Partir Δ por la bisectriz del mayor lado en Δ_a y Δ_b
    Añadir Δ_a y Δ_b a {Δ_i}
    Quitar Δ de {Δ_i}
    UB ← min_i UB(f(Δ_i))
    LB ← min_i LB(f(Δ_i))
  hasta UB − LB < ε
fin
```

Figura 3.6: Pseudocódigo del algoritmo "Branch and Bound".

Conclusión.

Estos métodos no son robustos por estar basados en la continuidad de la función $f(\cdot)$ en el espacio de búsqueda Ω. Los algoritmos como el de "Branch and Bound", que en el fondo son variaciones del método de "Grid Search" son poco eficientes cuando las dimensiones de Ω son elevadas. Así, para espacios de búsqueda de gran dimensión, funciones discontinuas, multimodales o problemas NP-Completos, ver [GJ79], los algoritmos deterministas van a ser ineficientes.

3.4.3 Algoritmos estocásticos.

Búsqueda estocástica.

La búsqueda estocástica pura consiste en evaluar N puntos del espacio de búsqueda de forma aleatoria y considerar que la solución es el mejor de los encontrados. La búsqueda estocástica orientada consiste en obtener puntos de forma aleatoria a partir de los puntos anteriormente encontrados, [Tim84, Rub92].

Algoritmo "Cristalización Simulada (Simulated Annealing)".

El algoritmo "Simulated Annealing" simula la cristalización de un elemento, ver pseudocódigo en figura 3.7 y [AK89, LA87, KGV83]. Es bastante parecido al algoritmo "Climbing-Hill", figura 3.5, pero ahora la probabilidad de escoger el mejor movimiento no es uno. De un punto dado podemos ir a uno mejor, si $f(x_n) < f(x_c)$, o a otro peor con una probabilidad que es función de T. La variable T simula la temperatura, y al disminuir la posibilidad de ir a un punto peor sea menor, asegurándose la convergencia del algoritmo. La aleatoriedad en la ruta hace que este algoritmo sea más eficiente que el algoritmo "Climbing-Hill" para encontrar el óptimo global, ver [Mic92].

Algoritmo "Búsqueda Tabú (Tabu search)".

El algoritmo "Tabu search" es también una variante estocástica del algoritmo "Climbing Hill", ver pseudocódigo figura 3.8 y [Glo89, Glo90]. El conjunto $S(x)$ es el conjunto de los puntos cercanos a x en función de una métrica y el conjunto T es el conjunto tabú, que en principio está vacío. Este conjunto va a cambia en cada iteración con una función no Markoviana $g(\cdot)$, es

```
inicio
  t ← 0
  iniciar temperatura T
  seleccionar x_c aleatoriamente
  evaluar f(x_c)
  repetir
    repetir
      x_n ∈ S(x_c)
      si f(x_n) < f(x_c), x_c ← x_n
      sino  si aleatorio[0,1) < exp{(f(x_c) − f(x_n))/T}, x_c ← x_n
    hasta (condición de fin)
    T ← g(T,t)
    t ← t + 1
  hasta (condición de parada)
fin
```

Figura 3.7: Pseudocódigo del algoritmo "Simulated Annealing".

decir, una función del histórico de T, ver [Pap91]. En este conjunto se va a ir introduciendo toda la experiencia acumulada en las anteriores generaciones. En cada iteración se elige N puntos cercanos a x_c que cumpla con las restricciones del conjunto tabú. Si la mínima función objetivo de los nuevos puntos es menor que la de x_c, se cambia el punto x_c por el mínimo. El algoritmo se acabará cuando hayan pasado $kmax$ iteraciones sin cambios o las restricciones del conjunto tabú no dejen elegir puntos cercanos a x_c.

```
inicio
  T ← ∅
  k ← 0
  seleccionar el punto x_c aleatoriamente.
  repetir
    seleccionar {x_1, ..., x_N} ∈ {S(x_c)}
    f(x_n) = min_{i=1,...,N} f(x_i)
    si f(x_n) < f(x_c),
      x_c ← x_n
      k ← 0
    sino k ← k + 1
    T ← g(T)
  hasta (k > kmax) ∧ ({S(x) − T} = ∅)
fin
```

Figura 3.8: Pseudocódigo del algoritmo "Tabu Search".

Algoritmo de "Monte Carlo".

Los métodos de "Monte Carlo" son simulaciones de eventos estocásticos que emplean una búsqueda pura aleatoria donde cada punto elegido es independiente del antecesor y posterior. Se van almacenando las mejores soluciones encontradas, ver [Rub81].

Algoritmos evolutivos (EAs).

Se tratarán más ampliamente en la sección 3.5.

Conclusión.

Los algoritmos de optimización estocástica introducen un componente aleatorio en la búsqueda del óptimo global lo que mejora su capacidad de exploración del espacio de búsqueda y disminuye la posibilidad de caer en un óptimo local. Además, estos algoritmos pueden abordar problemas con funciones no continuas o con espacios de búsqueda discreto que los algoritmos basados en cálculo no pueden.

3.5 Algoritmos evolutivos (EAs).

Con este nombre se designa a una clase de métodos de búsqueda estocástica que intentan simular los procesos de evolución natural, ver el pseudocódigo de la figura 3.9. Los algoritmos evolutivos se pueden dividir en Algoritmos Genéticos (GAs), Holland [Hol92], Estrategias de Evolución (ESs), Schwefel [Sch81] , Programación Evolutiva (EP), Fogel [FOW66] y Programación Genética (GP), Koza [Koz92].

En un espacio de búsqueda Ω, se eligen un conjunto de puntos como posibles soluciones del óptimo global. Este conjunto se llamará población inicial, y sus elementos individuos de la población. Los individuos de la población serán evaluados con respecto a la función objetivo, asignándose a cada individuo un índice de bondad. En una serie de iteraciones, llamadas generaciones, se selecciona parte de la población por medio del operador `seleccionar`$(\cdot)$. El criterio de selección es función del índice de bondad. Mezclando la información de la subpoblación seleccionada, se crea una nueva población de individuos. Está población es modificada de forma aleatoria por medio del operador `modificar`$(\cdot)$, y se evalúa su índice de bondad. El proceso se repite hasta una condición de parada.

La idea implícita en este algoritmo es que la información sobre el óptimo global está en el conjunto de la población inicial. Es necesario mezclar la información de los mejores individuos para obtener el óptimo deseado. El operador `modificar`$(\cdot)$ añade nueva información no contenida en la población inicial. Se puede, por tanto, definir estos algoritmos como procesos de optimización en paralelo de una población inicial. Esta característica hace que los EAs sean más efectivos que los algoritmos que partiendo de un punto aleatorio trazan un camino, por ejemplo el algoritmo "Simulated Annealing". Los EAs optimizan a ciegas, ya que sólo precisan del valor de la función objetivo, no su gradiente como en los métodos basados en cálculo. Por ello son de gran efectividad en optimización de funciones no continuas. Por otro lado, la exacta definición de los operadores `seleccionar`$(\cdot)$, `mezclar`$(\cdot)$ y `modificar`$(\cdot)$ dan mucha libertad al programador para acomodar el EA al tipo específico de problema que se desee resolver, por ejemplo los MOP. Tres buenas referencias sobre EAs son [Gol89a, Sch95, Mic92], y ha sido editado recientemente un libro monográfico sobre todos los temas relacionados con EAs en [Bäc97].

inicio
  $t \leftarrow 0$
  `iniciar` $P(t)$
  $bondad(t) \leftarrow evaluar(P(t))$
  Mientras (no condición parada) hacer
  inicio
    $t \leftarrow t+1$
    $P_{sel}(t) \leftarrow seleccionar(P(t))$
    $P(t+1)(t) \leftarrow mezclar(P_{sel}(t))$
    $P(t+1) \leftarrow modificar(P(t+1))$
    $bondad(t+1) \leftarrow evaluar(P(t+1))$
  fin
fin

Figura 3.9: Pseudocódigo de un Algoritmo Evolutivo (EA).

3.5.1 Mecanismo de un algoritmo genético (GA) simple.

Los GAs simples tiene claramente la estructura de un típico algoritmo evolutivo (EA), ver figura 3.9. A continuación se describirán alguno de los operadores típicos de un GA, siendo buenos libros de consulta [Gol89a, Mit99, Mic92], o artículos generales [BBM93b, BBM93c, SP94].

Espacio de búsqueda. Es el espacio Ω, ver definición 3.1 donde se busca la solución óptima. Los problemas donde no esté bien definido van a ser difíciles de tratar, ya que un espacio muy grande frena la convergencia y uno muy pequeño no ofrece garantías de englobar al óptimo.

Índice de Bondad. Es el factor que valora lo "bueno" que un individuo x es, siendo proporcional a $f(x)$ para la maximización e inversamente proporcional para la minimización.

Métodos de selección. Sirven para elegir a un porcentaje de la población, normalmente el 50%, como futuros padres de la siguiente generación. Un mismo individuo puede ser elegido varias veces. Hay diferentes métodos de selección, todos ellos basados en el índice de bondad de los individuos. El método de la ruleta, ver [Gol89a], asigna al individuo x_i una franja en el intervalo $[0,1]$,

$$P_i = \left[\frac{\sum_{j=1}^{i-1} bondad(x_j)}{\sum_{j=1}^{N} bondad(x_j)}, \frac{\sum_{j=1}^{i} bondad(x_j)}{\sum_{j=1}^{N} bondad(x_j)}\right],$$

donde N es el número de individuos de la población. Se genera un número aleatorio de [0,1] y se selecciona al individuo en cuya franja haya quedado el número aleatorio. El método del torneo binario, ver [GD91], elige dos individuos a azar de la población, seleccionado al mejor de ellos. Tanto uno como otro proceso se repiten tantas veces como individuos seleccionados se requieran.

Operador cruce y mutación. El operador cruce mezcla la información de dos individuos seleccionados. Para ello a un individuo se le asignan dos valores, su fenotipo que es el valor paramétrico de ese individuo, $x \in \Omega$, y su genotipo que es una codificación de su valor paramétrico. Existen varios tipos de codificación de la información de un individuo. La binaria crea como genotipo de un individuo una cadena de ceros y unos que corresponde a la unión de los valores en binario de cada componente de x. Si la componente $x(i)$ tiene como límites del espacio de búsqueda $LB \leq x(i) \leq UB$, y se elige una precisión de 6 dígitos la correspondencia entre los límites de fenotipo y genotipos de un individuo será $LB \rightarrow 000000$ y $UB \rightarrow 111111$. De forma similar se trabajaría con el código Gray. La codificación en coma flotante, ver [Mic92], convierte cada componente de $x \in \Omega$ en su respectivo valor en coma flotante. El genotipo de un individuo de la siguientes generación va a ser el cruce de genotipos de dos individuos seleccionados, padres, con una probabilidad de p_{cruce}, o serán iguales a los genotipos de sus padres con una probabilidad $1 - p_{\text{cruce}}$. Así, con un punto de corte (elegido aleatoriamente), el cruce será

$$\begin{array}{c|lcc|l} \multicolumn{2}{c}{\text{Padres (genotipo)}} & & \multicolumn{2}{c}{\text{Hijos (genotipo)}} \\ \hline 0101100 & 11001 & \Longleftrightarrow & 0101100 & 10000 \\ 1110010 & 10000 & & 1110010 & 11001. \end{array}$$

donde los genotipos tienen 12 dígitos. Se pueden aplicar de forma similar varios puntos de corte, ver [Gol89a], o el llamado cruce uniforme “Uniform Crossover”, ver [Sys89].

Tras ello se aplica el operador mutación a los nuevos genotipos. Este operador consiste en el cambio aleatorio de los dígitos del genotipo de individuo con una probabilidad de p_{muta}. Por último se descodifican el genotipo de los nuevos individuos para hallar su correspondiente fenotipo, o sea su valor paramétrico en el espacio de búsqueda Ω.

Élite. La élite es un operador añadido a los GA simples, ver [DJ75], que respeta los mejores individuos de una generación como individuos de la siguiente.

Restricciones. Cuando se tiene un problema general con restricciones

$$\begin{gathered} \min_x f(x) \\ g_i(x) \leq 0; \quad i = 1, \ldots, m \end{gathered}$$

los GA simples deben transformarlo en un problema sin ellas de la forma

$$min_x f(x) + r \cdot \sum_{i=1}^{m} \phi(g_i(x)) \tag{3.7}$$

donde r es el coeficiente de penalización y $\phi(\cdot)$ la función de penalización, ver [Gol89a, RPLH89]. La elección de r y $\phi(\cdot)$ es complicada por lo que en los GAs multiobjetivo se suelen formular las restricciones como nuevos objetivos.

3.5.2 Mecanismos de otros algoritmos evolutivos (EAs).

Los EAs utilizados como herramienta de este trabajo son los GAs, pero en este apartado se describirá de forma breve los mecanismos de otros tipos de EAs.

Estrategias de Evolución (ESs). Fueron desarrolladas como métodos para resolver problemas de optimización de parámetros, Schwefel [Sch81]. El fenotipo de un individuo viene representado por un par de vectores en coma flotante $v = (x, \sigma)$. El vector x representa el fenotipo real y σ es un vector que indica una desviación estándar. Las ESs más simples parten de un solo individuo cuyo fenotipo x se va modificando de forma que $x^{t+1} = x^t + N(0, \sigma)$, donde $N(0, \sigma)$ es la vector que representa un número aleatorio Gaussiano de media cero y varianza σ. Si la función objetivo del nuevo individuo es mejor que la de su predecesor, se cambia uno por otro. Con esta estructura se asegura la convergencia del algoritmo hacia el óptimo global.

De este simple algoritmo se desarrollaron otros dos, el $(\mu + \lambda) - ES$ y el $(\mu, \lambda) - ES$. En el algoritmo $(\mu + \lambda) - ES$, μ individuos producen λ nuevos individuos. La nueva población de $(\mu + \lambda)$ individuos es reducida por medio de un proceso de selección a μ individuos. En el algoritmo $(\mu, \lambda) - ES$ los μ individuos de una generación crean $\lambda > \mu$ nuevos individuos, de los cuales sólo μ son seleccionados como individuos de la nueva población. En este segundo algoritmo la vida de un individuo es de una generación. En estos dos algoritmos se aplica los operadores cruce y mutación. El operador cruce mezcla la información guardada en dos individuos seleccionados

$$\begin{aligned}
(x^1, \sigma^1) &= ((x_1^1, \ldots, x_n^1)(\sigma_1^1, \ldots, \sigma_n^1)) \\
(x^2, \sigma^2) &= ((x_1^2, \ldots, x_n^2)(\sigma_1^2, \ldots, \sigma_n^2)),
\end{aligned}$$

con el fin de crear un nuevo individuo (x, σ). La mezcla puede ser de forma discreta

$$(x, \sigma) = ((x_1^{q_1}, \ldots, x_n^{q_n})(\sigma_1^{q_1}, \ldots, \sigma_n^{q_n}))$$

donde $q_i = 1$ o $q_i = 2$ aleatoriamente, o bien como promedio

$$(x, \sigma) = (((x_1^1 + x_1^2)/2, \ldots, (x_n^1 + x_n^2)/2), ((\sigma_1^1 + \sigma_1^2)/2, \ldots, (\sigma_n^1 + \sigma_n^2)/2)).$$

La aplicación del operador mutación tiene la forma

$$\begin{aligned}
\sigma' &= \sigma \cdot e^{N(0, \Delta\sigma)} \\
x' &= x + N(0, \sigma'),
\end{aligned}$$

donde $\Delta\sigma$ es un parámetro predeterminado.

Programación Evolutiva (EP). Son un conjunto de algoritmos desarrollados para la creación de sistemas con inteligencia artificial en el sentido de ser capaces de predecir un evento futuro, llamado a_n, a partir de la experiencia acumulada de una serie de n eventos pasados, $a_0, \ldots, a_{n-1}$, ver [FOW66]. Para ello se formula como sistema una máquina de estados finitos. El fenotipo de un individuo será una matriz con los valores entrada-salida para la transición entre los diferentes estados de la máquina. La función objetivo para un individuo se obtiene como una relación entre el evento $\hat{a}_n$ previsto en la simulación de la máquina con los parámetros del individuo y el verdadero evento a_n. El operador cruce es similar al del algoritmo $(\mu, \lambda) - ES$. Este tipo de algoritmos se han generalizado para problemas de optimización numérica típicos, ver [Mic92].

GA híbridos. Este tipo de algoritmos combinan el GA simple con otros algoritmos de optimización estocástica o determinista. Entre ellos destacan la combinación de GAs con el método de "Grid Search", [CLD97, KNM95], y la de los GAs con el método de "Simulated Annealing", [WG96, JL96]. Estos métodos añaden a la estructura típica de un GA una fase donde se aplica el otro algoritmo.

3.6 Operadores para obtener diversidad en la solución de un algoritmo evolutivo (EA).

Los algoritmos genéticos (GAs) simples están planteados para resolver problemas de optimización global con una sola solución, ver sección 3.5.1. No es éste el caso de los problemas de optimización multiobjetivo (MOP), ver sección 3.2, ni de otros problemas similares donde la solución es un conjunto de puntos. Entre ellos se puede citar la optimización de funciones multimodales, es decir, funciones con varias soluciones óptimas. Otro ejemplo similar es la resolución de problemas donde la función a optimizar no sea estacionaria, lo cual hará que el GA deba cambiar de óptimo en el tiempo. Un tercer ejemplo es la simulación de un problema ecológico, donde se desea simular el comportamiento de diversos tipos de especies sin cruzarse unas con otras.

Para conseguir que un GA pueda resolver un MOP debe tener operadores que aproximen a la población inicial hacia el conjunto de óptimos de Pareto, y operadores que den diversidad a la población para conseguir el mayor número de elementos de dicho conjunto.

3.6.1 Diversidad.

Se llama convergencia prematura al fenómeno por el cual un EA converge en pocas generaciones hacia un punto del espacio de búsqueda. En el caso de optimización global, este punto será probablemente un óptimo local. Tres son los factores que influyen en esta convergencia prematura, la presión en la selección, el ruido en la selección y la ruptura en los operadores, ver [Mah95].

Presión en la selección ("Selection Pressure"). Es el resultado de grandes diferencias en los índices de bondad en una población finita, ver sección 3.5.1. En [Bäc94] se llama presión de selección al ratio entre el valor del índice de bondad del individuo óptimo y el de la media de la población. Los individuos con peor índice de bondad tienden a desaparecer, lo que lleva a una convergencia prematura cuando el óptimo global se encuentra en una zona rodeada de puntos con bajo índice de bondad.

Ruido en la selección ("Selection Noise"). Este fenómeno es fruto de la alta varianza del método de selección elegido. Con ciertos métodos de selección, el método de la ruleta por ejemplo [Mah95], ante varias zonas en el espacio de búsqueda Ω con alto índice de bondad, la población converge en solo una de ellas de forma aleatoria.

Ruptura de operadores ("Operator Disruption"). El operador cruce sirve para unir información útil de varios individuos con el fin de conseguir un fenómeno de optimización paralela de todos los individuos, ver sección 3.5.1. Este efecto, deseable, puede ir unido a otro indeseable donde el operador cruce rompe información útil al crear nuevos individuos. En [Mah95] se analiza este efecto negativo para los operadores cruce y mutación, siendo mucho mayor en el primero.

3.6.2 Métodos generales para conseguir diversidad en la población.

En esta sección, y basándose en el estudio realizado en [Mah95], se citarán una serie de técnicas desarrolladas con el fin de conseguir disminuir los problemas planteados en el apartado anterior, o al menos alguno de ellos, y por tanto lograr diversidad en la población en el curso de las sucesivas generaciones de un GA. La mayor parte de estas técnicas no logran de forma general su objetivo, la diversidad de la población, por lo que normalmente se usan como complementos a otro tipo de operadores, por ejemplo los métodos de nicho que se verán en la próxima sección.

Reducción del ruido de selección. El método de la ruleta tiene una gran varianza y han sido varias las modificaciones que sobre él se ha hecho para conseguir reducirlo, ver [Mah95]. Uno de ellos es el muestreo estocástico universal, "Stochastic Universal Sampling", el cual asegura que un individuo con índice de bondad bueno no pueda perderse.

Mezcla directa. Responden a un conjunto de técnicas que tratan de impedir que la población tienda a un solo punto del espacio de búsqueda. Una de ellas es el aumento de la probabilidad de mutación, que conlleva como consecuencia negativa la mala convergencia del algoritmo. Otra es el filtrado de la población para eliminar individuos iguales o muy parecidos.

Reinicialización. Para frenar la presión de selección se han desarrollado una serie de técnicas que incorporan a la población nuevos individuos en cada generación, [FF95b], consiguiendo mayor velocidad de convergencia. En los algoritmos micro genéticos toda la población menos la élite es sustituida por nuevos individuos, [Kar91, DBB94].

Algoritmos genéticos (GAs) paralelos. Son diferentes técnicas que dividen a la población en subpoblaciones intentando que cada una de ellas analice una parte de espacio de búsqueda, ver [CP97].

Los GAs paralelos de granulado basto ("Coarse-Grained Parallel GA") o "Algoritmo paralelo de isla" dividen a la población en un número pequeño de subpoblaciones. Todos los operadores se aplican de forma paralela en estas subpoblaciones y sólo tras un número dado de generación hay un intercambio de individuos entre ellas, migración. Esta migración puede ser controlada de forma que no haya repetición de individuos, siendo un buen porcentaje el 10% aproximadamente cada 5 generaciones, ver [Tan89].

Los GAs paralelos distribuidos cruzan a cada individuo con otro cercano a él en el espacio de búsqueda. Si el hijo tiene un índice de bondad mayor que el primero de los padres le reemplaza. Así, un individuo sólo es reemplazado por otro mejor, lo que da a este algoritmo mucha robustez y se podrá asegurar su convergencia, aunque suele ser lento. En [RVM96] se aplican este tipo de algoritmos a MOP.

Los GAs de grano fino ("Fine Grained GA") simulan una red de dos o tres dimensiones, asignando un individuo de la población inicial a cada cruce de la red. El operador cruce se realiza entre individuos colindantes en la red. Con esto se consigue que cada zona de la red analice una zona del espacio de búsqueda.

Algoritmos Genéticos (GAs) Termodinámicos. Son métodos donde se emplean técnicas similares a los algoritmos de "Simulated Annealing", ver sección 3.4.3. Un ejemplo es la aplicación del método de selección mediante torneo Boltzmann, ver [Gol90, Mah93b], donde se introduce un componente aleatorio en el operador selección consiguiendo una distribución de Boltzmann en la población.

Restricción en la reproducción ("restricted mating"). A partir de la idea de que las diferentes especies no se cruzan entre sí, esta técnica sólo deja mezclar individuos próximos en el espacio de búsqueda, consiguiendo nuevos individuos cercanos a los primeros. Con ello se mejora la ruptura de operadores pero no la presión y ruido de selección, [DG89], por lo que su aplicación debe ser reforzada con otras técnicas, [FF95b].

3.6.3 Métodos de nicho ("niching methods").

Estos métodos son técnicas que permiten la formación y mantenimiento de subpoblaciones estables dentro del espacio de búsqueda a través de diferentes generaciones, siendo mucho más eficientes que los descritos en el apartado anterior. Los nichos creados pueden ser espaciales, cuando se realiza una división espacial del espacio de búsqueda y nichos secuenciales si cambian en el tiempo. Los primeros se aplican a MOP y problemas de optimización de funciones multimodales y los segundos a problemas donde el óptimo varíe con el tiempo, ver [BBM93a, DG91].

Método del agrupamiento, ("crowding").

En este tipo de algoritmo sólo una parte de la población va a ser reemplazada y lo será por individuos cercanos a los primeros en su genotipo, aplicando por ejemplo la distancia de Hamming, ver [Cav70, DJ75]. Este algoritmo no genera diversidad en la población, sino que preserva la diversidad existente en la población inicial. En [Mah93a] se estudian este tipo de algoritmos con funciones test, y en [Mah94] se propone el algoritmo de agrupamiento determinístico ("Deterministic Crowding"). En él se agrupa a la población por parejas, se cruzan y cada padre podrá ser reemplazado, por torneo binario, por el hijo más parecido a él en genotipo. Este operador es muy usado en los problemas de optimización multimodal.

Método de "índice de bondad repartido" ("fitness sharing").

El primero en plantear esta técnica fue Holland en [Hol75, Hol92], proponiendo que los individuos tuvieran un índice de bondad modificado compuesto de dos medidas, la función objetivo y el índice de la densidad de población del nicho al que ese individuo pertenece. Fuerón Goldberg y Richardson [GR87] los que crearon por primera vez una función, llamada función de "índice de bondad repartida", que tuvo en cuenta ambos objetivos. La función "índice de bondad repartida", $f_{sh}(\cdot)$, para un individuo i es su función objetivo dividido por el "contador de nicho", función que refleja la densidad de población del nicho donde se encuentra el individuo i,

$$f_{sh}(i) = \frac{f(i)}{\sum_{j=1}^{N} sh(d(i,j))}, \tag{3.8}$$

donde N es el número de individuos de la población, $f(\cdot)$ es la función objetivo y $sh(\cdot)$ es una función de la distancia entre dos individuos. La definición de la función $sh(\cdot)$ es

$$sh(d(i,j)) = \begin{cases} 1 - \left(\frac{d(i,j)}{\sigma_{sh}}\right)^{\alpha}, & \text{Si } d < \sigma_{sh}; \\ 0, & \text{Otro caso.} \end{cases} \tag{3.9}$$

donde $d(i,j)$ es la distancia entre los individuos i y j, σ_{sh} llamado radio de nicho es la distancia estimada entre dos nichos, y α es una constante que frecuentemente tiene el valor de uno.

Este método precisa que se elija un espacio donde medir la distancia entre individuos y el radio de los nichos. En [Deb89, DG89] se demuestra que la elección los parámetros del individuo, fenotipos, como espacio de medida es más eficiente que la elección de los parámetros codificados, genotipos. En el primer caso se aplica como métrica la euclídea y en el segundo la distancia de Hamming. En los MOP se puede elegir como espacio de medida el de los vectores de funciones, siendo muy útil para obtener el frente de Pareto de dichos vectores, y en consecuencia el conjunto de óptimos de Pareto asociado, ver [HNG94, HN93].

En [OGC91] se demuestra que el método de selección por torneo da problemas con la función de "índice de bondad repartido", problemas solucionables si los "contadores de nicho" son una estimación de la población futura, ("continuously update share"). En [GDH92] se plantea el "contador de nicho" en forma de muestreo, reduciendo el tiempo de computación. En [YG93], se propone un algoritmo de grupos ("clustering") que delimite los nichos de la población. La reproducción de individuos sólo se realizará entre individuos del mismo grupo. En [Spe94] se añade al genotipo del individuo un conjunto de bits que codifican su nicho, restringiéndose el operador cruce a individuos del mismo nicho, ("simple subpopulation schema").

3.7 Algoritmos genéticos (GAs) aplicados a optimización multiobjetivo.

Los EAs más usados para su aplicación a MOP son los algoritmos genéticos (GAs). Éstos deben conseguir dirigir a una población inicial de individuos del espacio de búsqueda Ω hacia el conjunto de óptimos de Pareto, ver definición 3.5. Muchos de estos algoritmos emplean operadores para obtener diversidad, ver sección 3.6, debido a que la solución del problema no es única. En [FF95a] se dividen los GAs para la solución de un MOP en dos grandes grupos, los que utilizan de forma directa el concepto de frente de Pareto, ver definición 3.6, y los que no lo utilizan. En [VV99a, VVL98] se dividen estos algoritmos en tres clases, los algoritmos a priori, progresivos y a posteriori. Se consideran algoritmos a priori a aquellos que suponen conocimiento sobre la forma de la solución del problema, por ejemplo las técnicas clásicas de la sección 3.3. Se consideran algoritmos a posterior a los que tienen como objetivo la obtención del conjunto de óptimos de Pareto, sin presuponer nada, eligiendo luego el elemento que se desee. Los algoritmos progresivos cambian el objetivo de forma interactiva en el curso de las generaciones. Este artículo clasifica una gran cantidad de GAs en estos tres diferentes grupos, viéndose el gran desarrollo que han experimentado en los últimos años. En [ZT98a, ZDT99, ZT99] se plantea una clasificación de los principales algoritmos, estudiando su eficacia por medio de funciones test. Otras buenas referencias sobre el estado del arte de este tipo de algoritmos son [Hor97, Deb99, CC99]. A continuación, siguiendo las directrices de [FF95a], se van a clasificar los GAs en dos grandes grupos, los que emplean métodos directos y los que emplean métodos indirectos.

3.7.1 Métodos indirectos.

La mayor parte de estos algoritmos están basados en las técnicas de optimización multiobjetivo clásicas, ver sección 3.3.

En [Sch84, Sch85] se presenta por primera vez un algoritmo para conseguir la resolución de un problema de optimización multiobjetivo, el "Algoritmo Genético para la Evaluación de Vectores", VEGA. Su novedad está en modificar el operador selección de forma que se repita tantas veces como objetivos haya, siendo diferente el objetivo a optimizar en cada paso. Con ello se consiguen subpoblaciones seleccionadas para cada componente del vector objetivo. Las subpoblaciones obtenidas se mezclan de forma aleatoria y se aplica el operador cruce y mutación sobre ellas. El tamaño de las subpoblaciones seleccionadas tiene un efecto sobre la población final similar al que tienen los pesos en el algoritmo suma (3.1). Este algoritmo tiene buen comportamiento, en general, pero tiende a mover a la población hacia los extremos del frente de Pareto y no a su parte central. El mismo autor intento modificar este algoritmo con normas heurísticas sin resultado.

En [Fou85] se propone otro algoritmo parecido que en una segunda versión usa la optimización lexicográfica y el método de selección de torneo. En [IM98] se crea un algoritmo donde se selecciona individuos con respecto a una función suma (3.1), sobre la que se hacen varían aleatoriamente los parámetros peso. En [CCC99] se plantea el algoritmo MOSES ("Multiobjective Optimization of Systems in Engineering Sciences") donde se proponen dos métodos basados en las técnicas de min-max, ver sección 3.3. El primero requiere la formulación a priori de un punto utópico, y el segundo obtiene una aproximación de este punto a partir de la población existente. Hay otros muchos algoritmos que usan una combinación lineal o no lineal de las componentes del vector objetivo como índice de bondad, ver [VVL98].

3.7.2 Métodos directos.

Estos métodos parten de la definición en [Gol89a] de la llamada función "nivel de dominación". El primer "nivel" de una población lo componen el conjunto de óptimos de Pareto de dicha población. El segundo nivel lo componen el conjunto de óptimos de Pareto de toda la población a la que se ha quitado los individuos del primer nivel, y así sucesivamente. La función selección tendrá como índice de bondad de un individuo el nivel en el que se encuentre. Este único valor da una idea clara de la distancia del vector de funciones de un individuo respecto del frente de Pareto, y unido a operadores de diversidad son la clave de los más relevantes GAs aplicados a MOP creados hasta la fecha.

Algoritmo genético multiobjetivo (MOGA) (Fonseca y Fleming).

Este algoritmo [FF93, FF95b] plantea un procedimiento donde se crea la función "grado de dotación" o función "ranking". El valor de dicha función para un individuo es el número de individuos de la población que le dominan más uno. El conjunto de óptimos de Pareto de la población tienen "ranking" uno. En [GD91] se compara esta función con la función "nivel de dominación", llegando a la conclusión que la primera puede ejercer más presión de selección, ver sección 3.6. Este algoritmo aplica la función "fitness sharing" (3.8), para conseguir que no se acumule la población. El espacio donde se mide la proximidad de dos individuos es el espacio de sus vectores de funciones, con el fin de que estos se repartan a lo largo de todo el frente de Pareto. En [Deb99] se critica esta elección por impedir encontrar individuos diferentes, x_i, x_j, con el mismo vector de funciones $f(x_i) = f(x_j)$. Se aplica el operador "restricción en la reproducción", ver sección 3.6, como complemento para la obtención de diversidad y para aumentar la velocidad del algoritmo. En [FF98a, FF98b] estos mismos autores desarrollan de forma teórica las bases de su algoritmo y lo aplican a una turbina de gas como caso práctico.

Algoritmo genético con selección de Pareto-Nicho (NPGA) (Horn, Nafploitis y Goldberg).

La principal innovación de este algoritmo [HNG94, HN93] es aplicar un método de selección de torneo binario donde se mezcla el concepto de función de "nivel de dominación" con el de "contador de nicho" (3.7), ver figura 3.10. Los dos candidatos en el proceso de selección por torneo binario se comparan con otros t_{dom} individuos elegidos aleatoriamente. Si uno de los candidatos domina a los t_{dom} y el otro no, se elige al primero. Si hay empate, tanto por dominar como por no dominar, el individuo que esté en un nicho con menor densidad de población es el elegido. La densidad del nicho se mide por el "contador de nicho". El número de individuos con los que se compara t_{dom}, y el radio de nicho σ_{sh} son dos magnitudes fundamentales en este algoritmo, y en los artículos citados se estudian sus valores óptimos. En [HN93] se sugieren diferentes espacios de medida al aplicar la función "contador de nicho". Estos son el fenotipo, el genotipo o el vector de objetivos, considerándose más efectivo el último caso. También se mencionan los problemas y soluciones de la mezcla entre la función "contador de nicho" y el método de torneo binario, ver [OGC91]; y la posibilidad de usar un contador de nicho basado en muestreo que disminuya la carga computacional, ver [GDH92].

inicio
  `Barajar(individuo)`
  $candidato_1 \leftarrow individuo[1];$
  $candidato_2 \leftarrow individuo[2];$
  $candidato_1 dominado \leftarrow falso;$
  $candidato_2 dominado \leftarrow falso;$
  for $K \leftarrow 3$ **to** $t_{dom} + 3$
  Inicio
    $candidato_K \leftarrow individuo[K]$
    Si $ranking(candidato_K) < ranking(candidato[1])$
      $candidato_1 dominado \leftarrow verdadero;$
    Si $ranking(candidato_K) < ranking(candidato[2])$
      $candidato_2 dominado \leftarrow verdadero;$
  Fin
  Si $candidato_1 dominado$ **Y** $\neg candidato_2 dominado$
    Retornar $candididate_2$
  Sino Si $\neg candidato_1 dominado$ **Y** $candidato_2 dominado$
    Retornar $candididate_1$
  Sino Si $ContaNicho(condidato_1) > ContaNicho(candidato_2)$
    Retornar $candididate_2$
  Sino Retornar $candidato_1$
end

Figura 3.10: Estructura del operador de selección Pareto-Nicho.

Aproximación de Pareto guiada (SPEA) (Ziztler y Theile).

Este reciente algoritmo [ZT98a, ZT98b] usa el concepto de individuo no dominado, y de élite como el conjunto de los individuos no dominados desde la población inicial. A este conjunto se llamará "conjunto de Pareto", y formará parte de la población en las operaciones de selección, cruce y mutación. Por otro lado, el conjunto de Pareto será filtrado en cada generación para controlar el número de individuos, ver el esquema de este algoritmo en la figura 3.11.

El índice de bondad aplicado a la población, incluido el "conjunto de Pareto", será el siguiente. Los individuos óptimos de Pareto de este conjunto tendrán como índice de bondad el número de individuos de la población a los que dominan. Con ello los individuos que estén aislados tendrán menor índice de bondad. Un individuo del resto de la población tendrá co-

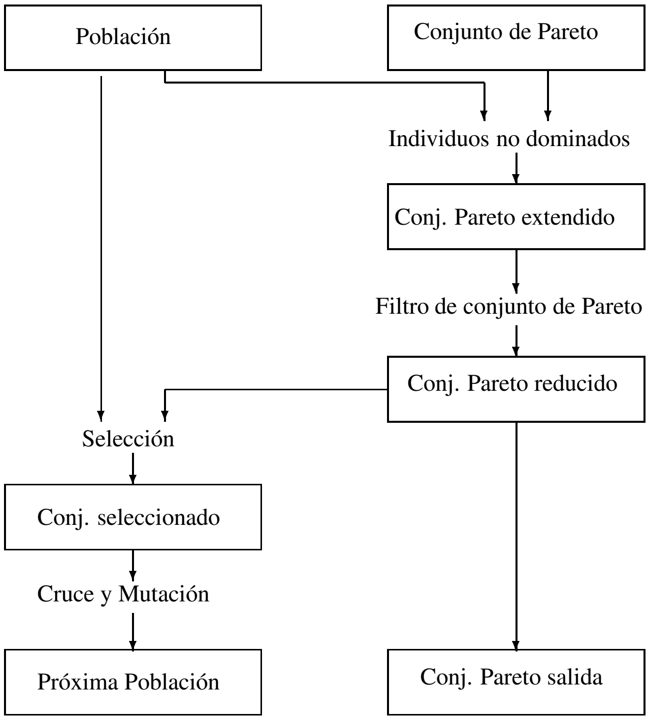


Figura 3.11: Esquema general de un algoritmo SPEA.

mo índice de bondad la suma de los siguientes términos; el número de individuos que no sean óptimos de Pareto existentes, más el valor de los índices de bondad de los individuos óptimos que le dominen más uno, ver figura 3.12.

Este índice de bondad consigue dos propósitos, por un lado valorar el hecho de que un individuo óptimo de Pareto esté aislado, y por otro lado, devaluar a los individuos que se hayan dominados por varios óptimos de Pareto o que estén en una zona de alta densidad de población. Ambos propósitos son útiles para conseguir diversidad en la población y una convergencia progresiva a lo largo de todo el frente de Pareto.

Algoritmo genético clasificador de no dominados (NSGA) (Srinivas, Deb).

Este algoritmo [Deb99] recoge directamente la idea de Golberg sobre la función del "nivel de dominación",ver [Gol89a]. Sólo se diferencia de un GA simple, ver sección 3.5.1, en el índice de bondad elegido. Este índice ha sido tomado aplicando la función "nivel de dominación", para conseguir obtener los individuos asociados al frente de Pareto, junto con la función "índice de bondad repartido" (3.8), para obtener diversidad en la población y de ese modo obtener el

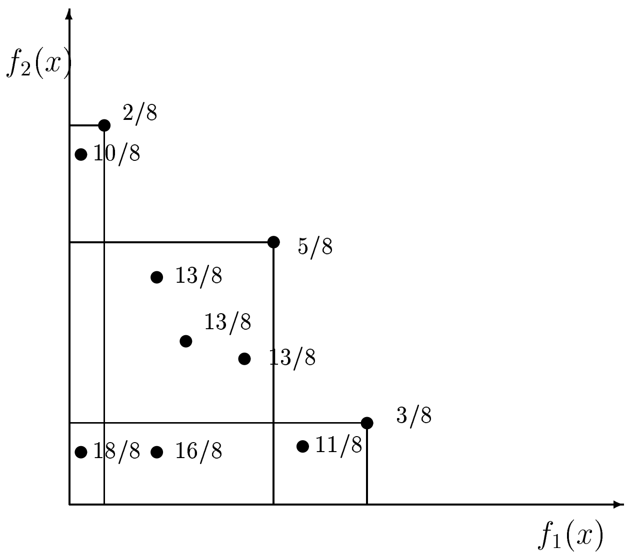


Figura 3.12: Índices de bondad de la población en el algoritmo SPEA.

máximo número de individuos asociados al frente de Pareto.

Algoritmos genéticos basados en la teoría de esquemas (MOMGA) (Van Veldhuizen).

Es otro reciente algoritmo [VV99b] basado en la teoría de esquemas, usada para explicar la convergencia de los GA, ver capítulo 5. Esta estrategia fue aplicada para GA simples en el algoritmo "messy GA" y en el "fast messy GA" [Gol89b, DG91], y Van Veldhuizen aplica la misma estrategia en GA para optimización multiobjetivo. El método es de gran utilidad para trabajar con las llamadas funciones deceptivas, donde el óptimo global se haya rodeado de subóptimos siendo complicado que el algoritmo converja hacia él. En [VV99b] se compara los resultados de su algoritmo y los mencionados anteriormente.

3.8 Aplicaciones prácticas de los algoritmos evolutivos (EAs).

En las últimas dos décadas el desarrollo de los algoritmos evolutivos (EA) ha sido espectacular. Como muestra de este proceso se pueden citar diferentes congresos internacionales como "Internacional Conference on Genetic Algorithms" (ICGA), "Genetic and Evoltionary Computation Conference" (GECCO), "IEEE International Confenceres on Evolutionary Programming", "Annual Conference on Evolutionary Programming" y "Parallel Problem Solving from Nature"

GA en inteligencia artificial, [Ala98b].
GA en arte y música, [Ala98a].
GA en biociencias, incluido medicina [Ala98c].
GA en diseño asistido por ordenador, [Ala98e].
GA en física y química, [Ala98d].
GA en control, [Ala98g].
GA en ciencias de la computación, [Ala98f].
GA en bases de datos, [Ala98h].
GA en economía y finanzas, [Ala98i].
GA en ingeniería, [Ala98j].
GA en lógica borrosa, [Ala98k].
GA en sistemas inmunes, [Ala98l].
GA en logística, [Ala98m].
GA en redes neuronales, [Ala98n].
GA en optimización, [Ala98q].
GA en óptica y procesamiento de imagen, [Ala98p].
GA en operaciones de búsqueda, [Ala98o].
GA en ingeniería de potencia, [Ala98r].
GA en búsqueda de proteínas, [Ala98s].
GA en señal y proceso de imágenes, [Ala98t].
GA en diseño de VLSI, [Ala98u].

Tabla 3.1: Áreas de aplicación de los GAs.

(PPSN). Existen, por otro lado, tres revistas especializadas en EAs, "Evolutionary Computation" (MIT Press), "IEEE Transaction on Evolutionary Computation" y "Genetic Programming and Evolvable Machines" (Kluwer Academic Publishers), así como numerosos grupos por red Internet, como por ejemplo,

- "`comp.ai.genetic`",

- "`GA-List_Request@aic.nrl.navy.mil`",

- "`pascal@icsc.ab.ca`".

Las aplicaciones prácticas de los EAs se han extendido a numerosos áreas de conocimiento. En el departamento de "Information Technology and Production Economics" de la Universidad de Vaasa hay una serie de artículos internos que clasifican libros, artículos y tesis basados en aplicaciones de los EAs a diferentes áreas, ver figura 3.1.

En estos artículos se estudia de forma minuciosa la distribución por fecha, país, tema y revista donde han sido publicados, y es una buena muestra del desarrollo de las aplicaciones de los EAs. Los diversos problemas a los que se aplican los EAs tienen como característica común el ser problemas de resolución difícil o imposible por los métodos convencionales, y buscan en los EAs una forma de llegar a una solución relativamente buena. Dos son los campos de aplicación en los que se va a entrar con más detalle, los EAs aplicados a problemas multiobjetivo y los EAs aplicados a control.

3.8.1 Aplicaciones de los algoritmos evolutivos (EAs) a problemas multiobjetivo.

Muchos problemas físicos se formulan de forma matemática como un problema de optimización multiobjetivo. Esto se debe a que se desea la optimización de más de un objetivo, estando estos en conflicto entre si. En este caso el problema se puede plantear como un problema de optimización de Pareto, ver sección 3.2, donde la solución es el conjunto de óptimos de Pareto. Como se vio en la sección 3.7, los EAs plantean muchas ventajas en las obtención del conjunto de óptimos de Pareto y en consecuencia en la obtención del frente de Pareto de los objetivos. Es por ello que la aplicación de los EAs a este campo tanto desde el punto de vista teórico como desde las aplicaciones prácticas se ha desarrollado fuertemente. Prueba de esta inquietud es la creación de secciones en los congresos de EAs especificas para este problema y de grupos de trabajo a través de la red Internet.

Un ejemplo de ello es el grupo del Laboratorio Nacional de Informática Avanzada del la Universidad de Veracruz (México), que ha recopilado en la dirección de Internet

- `www.lania.mx/ ccoello/EM00/EM00bib.html`

un total de más de trescientos cincuenta referencias de artículos, publicaciones en congresos, tesis y artículos internos sobre MOP resueltos a través de EAs, muchos de ellos con aplicaciones prácticas a algún campo de la ingeniería.

Dentro de los campos de la ingeniería donde se utiliza la optimización multiobjetivo a base de EAs se pueden destacar los siguientes. En Ingeniería Aerodinámica se usan para el diseño de la estructura de las alas de los aviones y formas de misiles, [And96, ABJ99, Oba98, VQ97]. En Ingeniería de Potencia para la resolución de problemas de distribución de corriente eléctrica, [BA98]. En la Organización de la Producción se aplican para el cálculo de almacenamiento e inventario, [IM98, SF96]. En Electrónica son utilizados en el diseño de circuitos integrados VLSI, [AHO96]. En Robótica se usan en el cálculo del contrapeso a los brazos del robot, [CCH95], en el control, en la simulación de su movimiento, en su aprendizaje, y en el control de la trayectoria. En Economía se han empleado en la resolución de algunos problemas financieros, [MB98]. Otras áreas donde se han aplicado los EAs para la resolución de problemas multiobjetivo son el reconocimiento de patrones, el diseño de distribución de calor, el control de sistemas digitales, como por ejemplo semáforos, el diseño de motores, el cálculo de estructuras y el diseño de sistemas electromagnéticos. El apartado de los EAs aplicados a control multiobjetivo, motivo de este trabajo, será tratado en el siguiente punto. En suma, los EAs son una buena herramienta para la resolución de los problemas multiobjetivo y la aplicación que pueden tener en las diversas áreas de la ingeniería es muy grande.

3.8.2 Aplicaciones de algoritmos evolutivos (EAs) a problemas de control multiobjetivo y/o robusto.

El desarrollo de EAs aplicados a control se ha dado principalmente en la década de los 90. En [Ala98g], se da un estudio detallado de esta evolución, donde se citan 640 referencias sobre documentos de control, 51 sobre robótica, 142 sobre control borroso ("fuzzy") y 62 sobre identificación de sistemas. Además, se agrupan por tipo de documento, año de edición, objetivos más comunes, distribución geográfica y principales autores. Por ejemplo, la distribución por tipos de documentos sería la siguiente; libros (4), artículo de revista (240), artículo de conferencias (590), tesis doctorales (15), tesis de master (9) y otros. El mismo autor en [Ala98v] hace un estudio similar sobre documentos relacionados con EAs que son de libre acceso a través de la red de internet. Ambos artículos nos dan una idea del gran desarrollo que han experimentado las aplicaciones de los EAs al control en la última década. Existen congresos sobre aplicación directa de los EAs a ingeniería de sistemas, como por ejemplo "Genetic Algorithms in Engineering Systems: Innovations and Applications" (GALESIA), y cada vez son más las revistas y congresos de control donde se incorporan aplicaciones hechas con EAs.

Las aplicaciones de los EAs al control de sistemas puede dividirse en dos grandes grupos. El primero de ellos es la aplicación de EAs para el diseño de controladores puros. Es decir que a base de algoritmos genéticos (GAs), estrategias de evolución (ES), programación de evolución (EP) y programación genética (GP), ver sección 3.5, se intenta hallar el valor de los parámetros y/o estructura de un controlador, [Ala98g]. El segundo grupo de aplicaciones son las llamadas aplicaciones híbridas. En ellas se utilizan los EAs para la obtención de parámetros de otro tipo de controladores no lineales, principalmente redes neuronales [Ala98n] y controladores borrosos, [Ala98k].

Artículos más relevantes.

En [FZL99] se aplica los EAs al problema de análisis de un sistema con incertidumbre paramétrica, ver sección 2.3.2 y [Bar94]. Este planteamiento hace que la ecuación característica del sistema sea función de una serie de parámetros $q \in Q$

$$P(z,q) = \left\{ p(z,q) = \sum_{i=0}^{n} a_i(q) z^i, q \in Q \right\} = 0.$$

Con ayuda de los EAs se conseguirá mostrar el grado de estabilidad del sistema.

En [ZHD97] se critica la efectividad de los GAs y los algoritmos "Simulated Anneling", ver sección 3.4.3 frente a los algoritmos deterministas "Branch and Bound", ver sección 3.4.2, en el análisis de un sistema con incertidumbre paramétrica. Está demostrado que este problema es NP-hard, ver sección 2.3.2. El estudio consiste en la aplicación de uno y otro método en un problema con un número suficiente de incertidumbre (32). La conclusión a la que llega es que los métodos determinista consiguen resultados más exactos, aunque con un costo computacional mayor. Sin embargo, se puede criticar que no todos los GAs son iguales ni tienen la misma efectividad ante un determinado problema.

En [KK96] se parte del típico problema de control óptimo, donde se intenta minimizar la función

$$J = \int_{t_0}^{t_r} [x^T(t) Q x(t) + u^T(t) R u(t)] dt,$$

siendo el control por realimentación de estado, $u(t) = -k * x(t)$. El problema se formula como un problema multiobjetivo donde se prescinde de los pesos Q y R y donde se da a la realimentación una estructura no lineal.

Varios de los ejemplos del libro [MTKH97] son diferentes artículos [MTK96, TMG96, GGM96] donde se exponen otras tantas aplicaciones a control. Estas aplicaciones son, el diseño de un filtro IIR, el diseño de un controlador $\mathcal{H}_\infty$ y el diseño de un controlador por redes neuronales. En todos estos ejemplos los fenotipos de los individuos tienen un carácter jerárquico.

Con ello no solo se puede modificar el valor de los parámetros del controlador, sino su propia estructura. Por ejemplo, dos individuos pueden representar a dos filtros con distinto número de polos y ceros. Con esta novedad, se plantea el problema de diseño de un controlador $\mathcal{H}_\infty$ como el diseño de los filtros de entrada y de salida óptimos para obtener un buen controlador aplicando el método $\mathcal{H}_\infty$ clásico.

En [DWC95] se plantea el diseño de un controlador que cumpla unas restricciones respecto a la norma $\mathcal{H}_\infty$ y otras seis funciones objetivo más. Estos objetivos se formulan como un conjunto de inecuaciones $\phi_i \leq \varepsilon_i, (i = 1, \ldots, 7)$. Se desea la búsqueda de filtros a la entrada y la salida que consigan obtener por el método $\mathcal{H}_\infty$ clásico un controlador que cumpla con las inecuaciones dadas. En [CL99] se plantea un problema similar, es decir, la obtención de un controlador por medio del método $\mathcal{H}_\infty$ clásico, donde un GA es usado para hallar los parámetros de los filtros de entrada y salida óptimos.

En [CC98] se considera un interesante problema abordado por más autores, la obtención de un controlador que optimice una norma $\mathcal{H}_\infty$ y tenga una estructura dada. Un controlador obtenido por las técnicas $\mathcal{H}_\infty$ clásicas tiene que tener al menos el mismo orden que la planta. Si la estructura del controlador se define con anterioridad, por ejemplo un controlador PID, la optimización $\mathcal{H}_\infty$ por métodos clásicos no es posible. Estos autores plantean esta optimización por un GA, analizando además otro interesante problema, el espacio de búsqueda donde seleccionar la población inicial del GA. En [CCL95] los mismos autores plantean la obtención de un controlador PID que optimice una norma mixta $\mathcal{H}_2/\mathcal{H}_\infty$. Este artículo convierte el problema en la minimización de la norma $\mathcal{H}_2$ sujeto a una restricción en la norma $\mathcal{H}_\infty$, como los métodos clásicos de la sección 3.3, de forma que el resultado es un único controlador. En [KCC99] se plantea un problema similar, diseño de un controlador con una estructura fija y que optimice un índice mixto $\mathcal{H}_2/\mathcal{H}_\infty$, a partir del empleo de estrategias de evolución (ES). El problema es plantado como la optimización de la norma $\mathcal{H}_2$, sujeto a una restricción en la norma $\mathcal{H}_\infty$.

En [CF96, CF95] se propone la aplicación del algoritmo multiobjetivo MOGA, ver sección 3.7.2 al control en una turbina de gas. El problema se plantea como la búsqueda de un controlador que cumpla con una serie especificaciones planteadas en forma de inecuaciones. En [SKF99] se propone la aplicación de otro nuevo algoritmo, llamado "STUDGA", para controlar la misma turbina.

En [WG96] se plantea el diseño de controladores multiobjetivo a partir del algoritmo "Simulated Annealing", ver sección 3.4.3. El problema multiobjetivo se plantea en forma de un conjunto de inecuaciones de los diferentes objetivos a minimizar, $\phi_i(p) \leq \varepsilon_i$, donde p son los parámetros del controlador. La función energía por la que se mueve el algoritmo "Simulated Anneling" depende de todas las inecuaciones,

$$E(p) = \max\left\{\frac{\phi_i(p) - \varepsilon_i}{\omega_i} : \quad i = 1, \ldots, n\right\},$$

donde $\{\omega_i, \quad i = 1, \ldots, n\}$ son pesos positivos definidos en función de la importancia de cada objetivo. El resultado es un único controlador, que depende de los pesos elegidos, de forma similar a si se utiliza el método clásico "máximo con pesos", ver sección 3.3.

En [KTK96] se analiza un punto importante y novedoso, el uso de GA para la resolución de un algoritmo "min-max". Se plantea el diseño de un controlador tipo I-PD para una planta con incertidumbre en sus parámetros. Un controlador tipo I-PD tiene una acción integral en el camino directo y una acción proporcional y derivativa en la realimentación. El problema se formula como un problema "min-max" (2.6),

$$\min_{q \in Q} \max_{\theta \in \Theta} J(q, \theta),$$

donde q son los parámetros del controlador, θ los parámetros de la planta y $J(\cdot)$ la función que se desea optimizar. El método consiste en discretizar el conjunto $Q = \{q| \quad q_l \leq q \leq q_u\}$ de la forma que se llegue a tener un número finito de plantas $Q_d := \{q_i| \quad i = 1, \ldots, N\}$. El GA propuesto maximiza la función de $J(\cdot)$ para cada valor q_i encontrándose una aproximación del problema "min-max". Los mismos autores en [HYKK99] vuelven a plantear un problema parecido.

En [JL96] se propone un algoritmo híbrido entre un GA y un algoritmo "Simulated Anneling" para diversas aplicaciones de control. El autor desarrolla un GA donde en la fase de mutación aplica un criterio similar al aplicado en el algoritmo "Simulated Anneling". Este algoritmo es usado en la identificación de parámetros de sistemas, y en el control de sistemas no lineales usando redes neuronales. En [TLMSS95] se plantea un algoritmo muy parecido que se utiliza en identificación de sistemas.

En el libro [CP99] cuyo tema es el diagnóstico de fallos dedica un capítulo a la identificación de sistemas para la detección de fallos, usando GAs. El problema es la identificación de un modelo, que será usado de observador de estado, y que debe ser insensible a la incertidumbre de la planta y sensible a los fallos en los sensores de forma que ambos fenómenos no puedan confundirse. La formulación matemática del problema es la minimización de varios índices, que se plantean en forma de inecuaciones $\phi_i \leq \varepsilon_i, (i = 1, \ldots, n)$. La resolución del problema se realiza mediante GAs. Uno de los autores de libro anterior usa en [LP96] los GAs para la control por asignación de polos. En [GP97] se usan también GAs para la estimación de fallos.

Conclusión.

Los algoritmo genéticos han sido aplicados a múltiples áreas en las últimas dos décadas y en concreto en el área que este trabajo estudia, el control robusto multiobjetivo. Hay numerosos artículos que tratan este tipo de problemas pero desde un punto de vista particular, aplicándolo a casos concretos con plantas SISO y controladores en la mayoría de los casos con estructura de PID. Las principales carencias que se ha encontrado en los artículos analizados son las siguientes:

- Falta un planteamiento del problema de control multiobjetivo general (2.5) y su resolución con un controlador definido en alguna de sus formas canónicas estándar. Con ello se conseguirá poder comparar los resultados de la aplicación de estos métodos con la obtenida con las técnicas de control mixto vistas en la sección 2.3.

- Los límites de los parámetros de los controladores son en muchos casos desconocidos, lo que supone un grave problema para poder aplicar GAs. Pocos son los autores que han estudiado la forma de superar este problema.

- El problema de control robusto planteado como un problema min-max no ha sido prácticamente tratado.

3.9 Conclusión.

Muchos problemas de diferentes áreas de conocimiento se formulan a partir de un problema de optimización. Si el problema sólo tiene un objetivo y no es convexo se estará ante un problema de optimización global. Si el problema tiene varios objetivos se estará ante un problema de optimización multiobjetivo cuya solución no es única.

En ambos casos, los métodos clásicos de resolución no son muy efectivos. En la optimización global los métodos clásicos convergen hacia un óptimo local, función en muchos casos del punto de partida empleado. En la optimización multiobjetivo los métodos clásicos reducen el problema a uno de optimización global, con lo que se pierde información. Para ambos problemas, los EAs son una buena alternativa. En la optimización global consiguen encontrar una buena solución con un coste computacional no muy elevado, y en la optimización multiobjetivo consiguen encontrar de forma conjunta una buena aproximación a todo el conjunto de óptimos de Pareto.

Ambos hechos son la principal razón de la gran popularidad de los EAs en las dos últimas décadas, y de las numerosas aplicaciones que de los mismos se han hecho en múltiples áreas de conocimiento, una de las cuales es el control de sistemas. Hay numerosos artículo que aplican EAs al problema de control robusto y/o multiobjetivo pero casi siempre en casos concretos. Se precisa una formulación general del problema para que se pueda comparar esta técnica con otro tipo de técnicas de control mixto. Por otro lado, quedan problemas por resolver como el desconocimiento del límite del espacio de búsqueda inicial y la resolución de un problema min-max con GAs.

Capítulo 4

Mecanismos de los algoritmos MRCD.

4.1 Introducción

Los algoritmos MRCD simple y el MRCD min-max son dos algoritmos genéticos (GAs) multiobjetivo, ver sección 3.7, desarrollados en esta tesis para la resolución de los problemas de diseño multiobjetivo y diseño robusto multiobjetivo planteados en su forma paramétrica, ver sección 2.4. El acrónimo MRCD proviene de la denominación inglesa "Genetic Algorithm for **M**ultiobjective **R**obust **C**ontrol **D**esign", con la que fue presentado por primera vez en la conferencia GECCO'99, ver [HLBLPG99]. Según las clasificaciones propuestas en la sección 3.7.2, los algoritmos MRCD 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.

El algoritmo MRCD simple ha sido diseñado para obtener los parámetros de controladores multiobjetivo con una estructura dada a priori, ecuación (2.5). Los límites del espacio de búsqueda de los parámetros de dichos controladores son desconocidos, por lo que se precisan nuevos operadores que consigan analizar el mayor área posible, sin impedir la convergencia del algoritmo. 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", ver sección 3.6.1, para poder asignar a cada subpoblación unos límites iniciales diferentes. Para conseguir mayor libertad en la elección del espacio de búsqueda inicial se formula un objetivo fijo a optimizar, el grado de estabilidad del sistema en lazo cerrado. Con él, a partir de una población inicial de controladores que inestabilizan la planta el algoritmo encontrará controladores que la estabilicen e irá convergiendo hacia los óptimos de Pareto.

El algoritmo MRCD min-max está pensado para la resolución de problemas de diseño de control en plantas con incertidumbre paramétrica, ver ecuación 2.6. Pocos son los EAs propuestos para resolver un problema de min-max y este algoritmo lo intenta con un esquema muy similar al del MRCD simple. El espacio de búsqueda en este caso estará dividido en dos tipos de parámetros, $\{m|M\}$; los parámetros m son aquellos que deben ser modificados para obtener un mínimo y los parámetros M deben ser modificados para obtener un máximo en el vector de funciones objetivo. El algoritmo está diseñado para modificar los parámetros m, con una parte M dada, con el fin de obtener un mínimo. En cada ciclo, se modifica la parte M de las mejores soluciones con el fin de obtener un máximo. En este juego se consigue una doble convergencia de las dos partes en las que se haya dividido el espacio de búsqueda.

El capítulo está dividido en las siguientes secciones. En la sección 4.2 se explica los objetivos que deben resolver los algoritmos y los problemas que implican su convergencia. En la sección 4.3 se explica de forma pormenorizada el mecanismo del algoritmo MRCD simple, y en la sección 4.4 se explica el mecanismo del MRCD min-max incidiendo en sus diferencias con el primero. En la sección 4.5 se describe el lenguaje de programación usado para implementar estos algoritmos y el entorno gráfico desarrollado para facilitar al usuario su uso en el diseño de controladores. Esta aplicaciones han sido desarrollados en el entorno de MATLAB. En la sección 4.6 se dan una serie de conclusiones sobres los algoritmos propuestos.

4.2 Objetivos de los algoritmos MRCD y MRCD min-max.

El principal objetivo del algoritmo MRCD simple es la solución de un problema de control multiobjetivo para una planta nominal en su planteamiento paramétrico (2.5), y el principal objetivo el algoritmo MRCD min-max es la solución de problemas de control multiobjetivo de plantas con incertidumbre paramétrica en la formulación propuesta en (2.6).

Para la resolución de ambos problemas es preciso fijar a priori una parametrización del controlador. En la mayoría de los casos se establece el orden y estructura del controlador deseado, aunque diversos autores proponen controladores donde el orden y/o estructura vienen definidos por los parámetros a optimizar. En [MTK96, TMG96, GGM96] a este tipo de estructura se la denomina estructura jerárquica. En los apéndices C se describen las formas canónicas más usadas en esta tesis y en el apéndice E las diversas formas de un controlador PID (Proporcional-Integral-Derivativo)

Los límites del espacio de búsqueda de los parámetros del controlador son desconocidos a priori. Este desconocimiento es un grave inconveniente en la aplicación de GA a los problemas propuestos. Un gran espacio de búsqueda provoca una mala convergencia y un pequeño espacio no garantiza que se esté buscando en el área correcta. Pese a ser un problema capital en el uso de los GAs para diseño de controladores, pocos autores lo han tratado suficientemente. En [CC98] se propone la búsqueda de un espacio inicial a base del uso de técnicas de “Simulación de Monte Carlo”, ver sección 3.4.3, lo que puede aumentar de forma considerable el tiempo computacional del algoritmo.

Los algoritmos MRCD plantean nuevos operadores y un objetivo fijo para conseguir explorar el máximo espacio de búsqueda posible sin perder propiedades de convergencia:

Límites dinámicos del espacio de búsqueda. Los límites serán modificados en función del valor de los parámetros de los individuos de la población. Con ello se puede definir un gran espacio de búsqueda inicial que en sucesivas generaciones se irá optimizando.

Algoritmo paralelo en isla. Se divide a la población en subpoblaciones con distintos límites en el espacio de búsqueda. Con ello se pretende una exploración por secciones del espacio de búsqueda, dando lugar al desarrollo de nichos. Este operador puede ser empleado para la disminución del tiempo computacional del algoritmo.

Migración. De forma cíclica se mezclan los individuos de las subpoblaciones, lo que acarrea un cambio en los límites de los espacios de búsqueda.

Objetivo fijo: Medida de la estabilidad. Las normas en sistemas inestables tienen el mismo valor, infinito. Este índice hace posible medir la bondad de dos controladores que inestabilicen la planta, lo que da una gran libertad en la elección del espacio de búsqueda inicial. Este índice puede ser la máxima distancia del polo más inestable al eje imaginario, en el caso continuo, o al circulo unidad en el caso discreto.

El algoritmo MRCD trata de obtener todo el frente de Pareto como solución del problema para lo cual se aplican diversas variantes de los operadores que obtienen diversidad estudiados en las secciones 3.6 y 3.7. Entre ellas, se han elegido el método de selección binaria por nichos y el operador “restricción en la reproducción”.

En la mayor parte de los casos el número de parámetros del controlador va a ser muy elevado, un controlador de una entrada y dos salidas con seis polos representado en alguna forma canónica tiene 22 parámetros. Por ello el operador cruce genera un punto de corte en cada zona del genotipo (parámetros codificados) correspondiente a un parámetro.

Para dar mayor robustez al algoritmo se respetan las mejores soluciones encontradas en la generación anterior, pero éstas se filtran para evitar una convergencia prematura.

La resolución de un problema min-max a partir de GAs ha sido muy poco tratado en la literatura. En [Ack93] se aplica técnicas basadas en el “Teorema de Kharitonov” para conseguir controladores robustos en plantas de orden reducido. En [KTK96, HYKK99] se discretiza una planta con incertidumbre como un conjunto de plantas y se resuelve el problema para todas ellas, comprobando a la vista de los resultados cual es la peor incertidumbre.

El algoritmo MRCD min-max trata de resolver este problema de una forma muy simple. Se define un espacio de búsqueda con los parámetros de la planta e incertidumbre. Se aplica el algoritmo MRCD simple sobre la parte de parámetros correspondientes al controlador, tomando como incertidumbre alguna de las existentes en la población. En cada ciclo se modifica las incertidumbres de los mejores controladores con el fin de maximizar el vector de funciones. Con ello se consigue que los mejores controladores lo sean para las peores incertidumbres.

4.3 Mecánica del algoritmo MRCD.

Este algoritmo se puede clasificar como un algoritmo evolutivo para resolución de problemas de optimización multiobjetivo paramétricos por métodos directos, usando la nomenclatura propuesta en [FF95a]. Es decir, un algoritmo que usa de forma expresa el concepto conjunto de óptimos de Pareto y el de frente de Pareto asociado. Por otro lado, es un algoritmo a posteriori ya que trata de encontrar una aproximación a todo el conjunto de óptimos de Pareto al mismo tiempo, dejando para una segunda fase la elección del punto (controlador en este caso) deseado, ver [VV99b].

El algoritmo MRCD ha sido diseñado para la resolución del problema planteado en (2.5), es decir, la búsqueda de los parámetros de un controlador con una estructura dada a priori, que minimice una serie de objetivos y cumplan con unas restricciones. Varios operadores de este algoritmo están diseñados para poder explorar un gran espacio de búsqueda ya que los límites de dicho espacio en el problema planteado no están definidos. Por otro lado, este algoritmo tiene operadores que consiguen como objetivo una aproximación al conjunto de óptimos de Pareto.

```
inicio
  gen ← 0;
  repetir
    gen ← gen + 1;
    desde k = 1 a k = Nsubpop
    inicio
      [poblac(k), bondad(k)] ← InPob(espacio(k));
    fin
    desde k = 1 a k = Nsubpop
    inicio
      espacio(k) ← ModΩ(poblac(k), gen);
      [selec(k), elite(k)] ← Selec(poblac(k), bondad(k));
      [newpoblac(k), newbondad(k)] ← Mecl(poblac(selec(k)), ...
                                        bondad(selec(k)), espacio(k));
      elite(k) ← Filt(poblac(elite(k)), bondad(elite(k)))
      poblac ← [newpoblac(k), poblac(elite(k))];
      bondad ← [newbondad(k), bondad(elite(k))];
    fin
    desde k = 2 a k = Nsubpop
    inicio
      [poblac(k), bondad(k)] ← Migra(poblac(k), bondad(k), ...
                                        poblac(k − 1), bondad(k − 1));
    fin
    poblac(1) ← Migra(poblac(1), bondad(1), poblac(Nsubpop), bondad(Nsubpop));
  hasta(condición parada)
fin
```

Figura 4.1: Pseudocódigo del algoritmo MRCD

4.3.1 Población inicial. Espacio de búsqueda variable.

La población inicial se obtiene del muestreo uniforme de un espacio de búsqueda Ω. En la aplicación de este algoritmo a problemas de control multiobjetivo (2.5) no se requiere que los controladores de la población inicial estabilicen el sistema en lazo cerrado. Ello es debido a que la primera función a optimizar en este tipo de problemas es una restricción que mide el grado de estabilidad del sistema en lazo cerrado, ver sección 4.2. El algoritmo MRCD encuentra controladores que estabilizan el sistema a partir de una población inicial que puede no hacerlo.

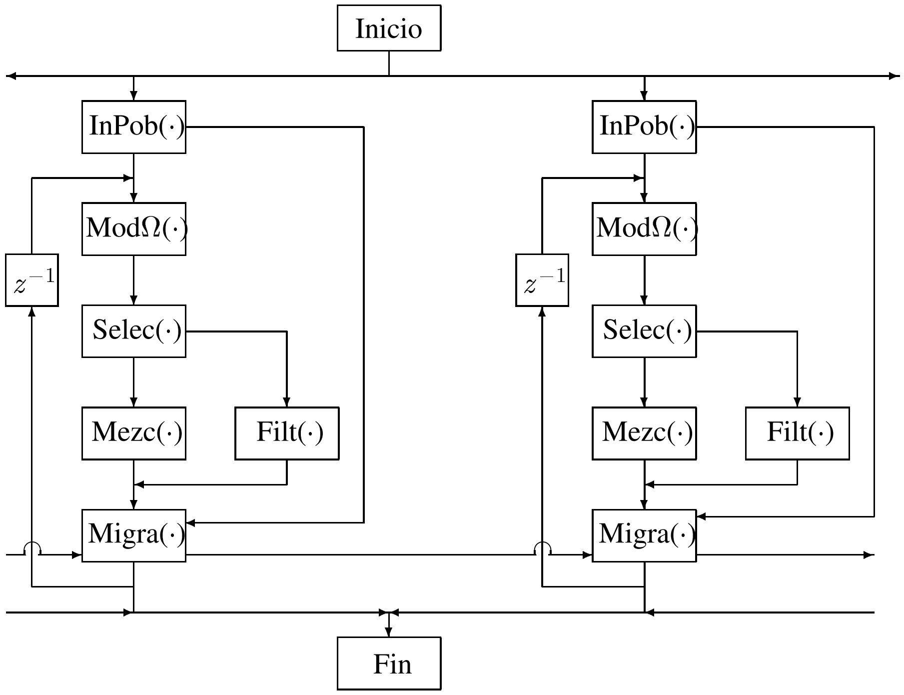


Figura 4.2: Diagrama de flujo del algoritmo MRCD.

Los límites del espacio de búsqueda Ω serán modificados cíclicamente en relación del número de generaciones transcurridas y del operador migración que se verá más adelante. La modificación de estos límites va a ser función de los valores máximos y mínimos de los parámetros de la población. Dado un vector de parámetros a optimizar, $k = [k_1, \ldots, k_n]^T$, los límites del parámetro k_i, $[k_i^{max}, k_i^{min}]$, pueden variar de la forma

$$k_i^{min} = \min(k_i(1), \ldots, k_i(N)) - \delta \cdot T$$
$$k_i^{max} = \max(k_i(1), \ldots, k_i(N)) + \delta \cdot T,$$

siendo

$$\delta = \max(k_i(1), \ldots, k_i(N)) - \min(k_i(1), \ldots, k_i(N)),$$
$$T \in [0, 1], \quad i \in \{1, \ldots, n\},$$

donde N es el número de individuos de la población y T una constante del algoritmo.

Dos son los principales objetivos conseguidos con este operador en su aplicación al problema (2.5). El primero es la posibilidad de elegir un gran espacio de búsqueda inicial de controladores de todo tipo, que conforme el algoritmo converja a una zona de controladores que estabilicen la planta en lazo cerrado se ira modificando. La segunda es la convergencia final hacia la solución óptima, dando al algoritmo mucha precisión.

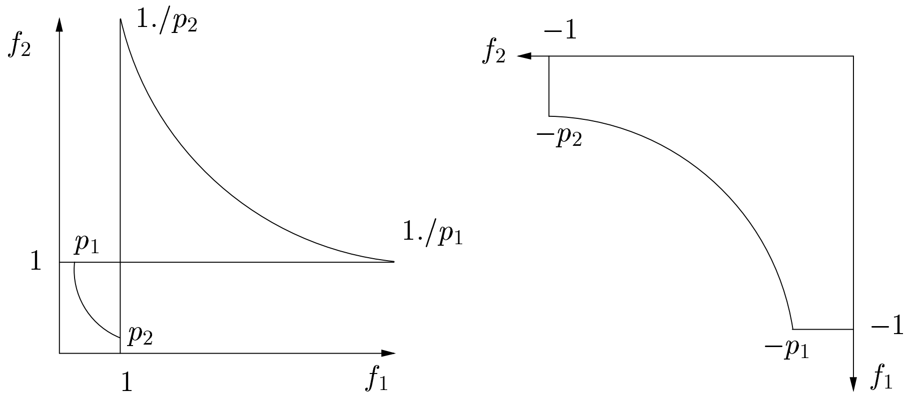


Figura 4.3: Frente de Pareto del vector de funciones normalizado.

Para poder aplicar este operador, el genotipo de los individuos de la población deben ser modificado en cada generación con respecto al nuevo espacio de búsqueda. En este algoritmo se emplea el código binario o Gray para formar el genotipo de los individuos.

4.3.2 Escalado.

Los algoritmos que usan de forma directa el concepto de frente de Pareto usan la distancia entre dos individuos. Esta distancia puede ser definida en el espacio del fenotipo (parámetros), genotipo (parámetros codificados), o en el vector de funciones objetivo. El algoritmo MRCD usa principalmente el espacio del vector de funciones objetivos como medida de la distancia entre dos individuos. Por ello es importante poner en una misma escala todas las componentes de dicho vector para que sean comparables. El algoritmo MRCD resuelve un problema de maximización, pudiéndose formular un problema de minimización de dos formas, como la maximización de las inversas del vector de funciones, o como la maximización del vector de funciones multiplicado por -1, ver figura 4.3. La primera forma es útil en casos donde haya individuos con valores muy diferentes en una función objetivo, ya que los pequeños valores relativos se convierten en grandes valores. En el caso de maximización de la función inversa se toma como factor de escala el mínimo valor de cada objetivo, que corresponde con el máximo valor de su inversa. En el caso de maximización de la función negativa se toma como escala el máximo valor absoluto de cada objetivo. De esta forma las funciones objetivo quedan normalizadas tal y como se muestra en la figura 4.3

4.3.3 Método de selección.

El método de selección empleado en el algoritmo MRCD es el método de Pareto-Nicho definido en el pseudocódigo de la figura 3.10, ver [HNG94, HN93]. La única diferencia está en el índice de bondad elegido, que en este caso es la función “ranking”, definida en [FF93, FF95b], en vez de la función “índice de dominación” definida en [Gol89a], ver sección 3.7.2.

4.3.4 Estructura “Algoritmo paralelo de isla”.

El algoritmo MRCD tiene una estructura de GA paralelo, ver sección 3.6.1, con el fin de conseguir flexibilidad en la elección de los límites del espacio de búsqueda. Esta estructura consiste en trabajar de forma paralela con un conjunto no muy grande de subpoblaciones, ver figura 4.2 y figura 4.1. Cada subpoblación tiene su propio espacio de búsqueda, y sólo habrá relación entre estas subpoblaciones a través de una migración al final de cada generación, operador `Migra`$(\cdot)$. El operador migración cambia los peores individuos de una subpoblación por los mejores individuos de otra, migración interna, o por individuos buscados de forma aleatoria a partir del operador `InPob`$(\cdot)$, migración externa. El tanto por ciento de una o otra migración es un parámetro del algoritmo.

La migración interna provoca futuros cambios en los límites del espacio de búsqueda que tiende a homogeneizar las diferentes subpoblaciones. Con ello se intenta que en el curso de las generaciones los límites de las subpoblaciones tiendan a un óptimo. La migración externa sirve para introducir al algoritmo nueva información sobre los espacios de búsqueda modificados.

Esta estructura puede servir para mejorar el tiempo de cálculo del algoritmo, ya que permite hallar el vector de funciones de los nuevos individuos de las diferentes subpoblaciones de forma simultanea. Esta es la operación que más tiempo de cálculo precisa.

4.3.5 Cruce, mutación y restricción en la mezcla.

El operador `Mezcl`$(\cdot)$ se compone básicamente de los operadores cruce y mutación descritos en la sección 3.5.1 para el mecanismo de los GA simple. Dichos operadores generan los genotipos (parámetros codificados) de los elementos de una nueva población a partir del cruce y mutación de los genotipos de los individuos seleccionados por el operador `Selec`$(\cdot)$, elegidos por parejas, a los que se llamarán padres.

El operador `Mezcl`$(\cdot)$ aplica "restricción en la reproducción" (ver sección 3.5.1) al seleccionar a dos individuos como padres. Para ello se elige a un primer individuo de forma aleatoria y a otros T posibles candidatos como segundo individuo. Aquel de los T individuos que esté más cerca del primero en cuanto a su fenotipo (parámetros) será elegido. El valor de T es un parámetro del algoritmo y puede ser un 5% de la población.

Para cruzar los genotipos de los padres se genera un punto de cruce en cada parte del mismo correspondiente a un parámetro del fenotipo del individuo, ver [Gol89a]. De los dos genotipos generados en el cruce se elige uno de forma aleatoria que se mutará y se descodificará para hallar el fenotipo del nuevo miembro de la población. Con ello se pretenden conseguir mayor diversidad en la población.

4.3.6 Restricciones.

El algoritmo MRCD plantea las restricciones como un nuevo objetivo a optimizar. Estos objetivos deben tomar un valor que mida su proximidad al cumplimiento de la restricción o un valor límite en caso de que las cumplan. Con ello se evita el uso de funciones "penalti" (3.7) cuyos resultados son bastante pobres.

En las aplicaciones a control multiobjetivo (2.5) la restricción de estabilidad en lazo cerrado es fija y de suma importancia. El valor de todos los objetivos de un sistema inestable va a ser infinito menos el de la restricción de estabilidad. Un posible valor de esta restricción es la máxima distancia de los polos inestables del sistema en lazo cerrado al eje imaginario en casos de sistemas continuos, o al circulo unidad en caso de sistemas discretos.

4.3.7 Élite.

La élite está compuesta de los "mejores" individuos. obtenidos desde la población inicial, ver el mecanismo de los GA simple en la sección 3.5.1. El algoritmo MRCD incorpora a parte de la élite, filtrada por el operador `Filt`$(\cdot)$, como individuos de la siguiente generación. El filtro de la élite se realiza para evitar convergencia prematura, y está basado en la técnica de nichos. Se elige un individuo por nicho en cada una de las élites de las diferentes subpoblaciones, ver sección 3.6.3. El espacio para medir la distancia entre nichos, radio de nicho, puede ser los parámetros de los individuos o su vector de funciones objetivo.

Otro tipo de filtro en la élite es el provocado por las restricciones en los valores del vector de funciones objetivo. Para fijarlas es preciso tener conocimiento sobre la zona del frente de Pareto que se desea. Los individuos que no cumplan estas restricciones no podrán pertenecer a la élite. Con este filtro se puede exigir, por ejemplo, que todos los individuos de la élite cumplan las restricciones del problema.

4.4 Mecánica del algoritmo MRCD min-max.

Este algoritmo es una variante sobre el algoritmo MRCD simple, descrito en la sección 4.3. 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, ver ecuación (2.5). 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, $\texttt{Filt}(\cdot)$. A continuación se describen las variaciones del algoritmo MRCD min-max sobre el MRCD simple.

4.4.1 Población inicial. Espacio de búsqueda variable.

En el algoritmo MRCD min-max el fenotipo de un individuo está dividido 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 individuos queda de la siguiente forma,

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

En la resolución de (2.6) la población inicial puede estar formada por controladores que no estabilicen la planta en lazo cerrado. El operador $\texttt{Mod}\Omega(\cdot)$ se aplicará sólo al espacio m, los parámetros del controlador, ya que el espacio M, los parámetros de la incertidumbre de la planta, tiene unos límites fijos.

El genotipo de los individuos (parámetros codificados) se dará en código binario o Gray.

4.4.2 Escalado.

Se aplica de forma similar al algoritmo MRCD simple.

4.4.3 Método de selección.

Se aplica de forma similar al algoritmo MRCD simple. La función "ranking" se aplica con el criterio de minimizar el vector de funciones objetivo, es decir, se trata de obtener los individuos con mejores parte m.

4.4.4 Estructura "Algoritmo paralelo de isla".

Se aplica de forma similar al algoritmo MRCD simple.

4.4.5 Cruce, mutación y restricción en la mezcla.

El operador mezcla de dos individuos actúa en la parte m, con los mismos criterios que el MRCD simple. A los individuos resultantes se les asignará una de las partes M de los dos individuos padres, elegida aleatoriamente. Se usa el operador "restricción en la reproducción". En esta parte del algoritmo se desea obtener nuevos individuos con mejores parte m, y por tanto dominantes respecto a la minimización del vector de funciones objetivo.

4.4.6 Restricciones.

Se plantean, al igual que en el MRCD simple, como nuevos objetivos que miden la proximidad del individuo a cumplirlas, llegando a tener un valor límite en el caso de que las cumplan.

4.4.7 Filtro en la élite.

Esta es la principal diferencia 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.

El mecanismo para conseguirlo es el siguiente. Durante un número de ciclos t_{max} la parte M de los individuos de la élite es modificada con el fin de maximizar el vector de funciones objetivo. Dichas modificaciones se harán por alguno de los tres siguientes caminos, elegidos aleatoriamente:

- Se elige un vector aleatorio M.
- Se elige la parte M de otro individuos de la población.
- Se cruza y muta la parte M de ese individuo con la parte M de otro individuo de la población.

Si el individuo con la nueva parte M tiene un vector de funciones dominante sobre el antiguo, con respecto a la maximización, se modifica la antigua parte M por la nueva. Este filtro hace que la élite se forme de individuos que no son sensibles a cambios en la parte M. Por nuestra experiencia, el número de ciclos t_{max} no debe ser mayor de dos o tres para que el algoritmo funcione correctamente.

Sólo se filtra la élite para no incrementar el tiempo de cálculo de este algoritmo con respecto al MRCD simple. Todo buen individuo llegará a ser parte de la élite y en ese momento será filtrado.

4.5 Implantación de los algoritmos MRCD.

Entorno de programación. El entorno de programación elegido para el desarrollo de los algoritmos MRCD ha sido el entorno de programación de MATLAB, ver [The97]. La razón es la de hacer compatible los algoritmos MRCD con los paquetes de programación sobre control en general, y control robusto en particular realizados en dicho entorno. Dentro de ellos caben destacar los paquetes de "Control System", "μ syntesis", "Robust Control", "LMI control" y "System Identification".

Lenguaje de programación usado. Con el fin de mejorar el tiempo de cálculo de estos algoritmos, uno de los principales problemas de los EAs, las principales funciones de los algoritmos MRCD han sido programadas en lenguaje de programación "C", y compiladas a través del compilador "CMEX" como funciones de MATLAB. Con ello se une a las ventajas mencionadas de la programación en entorno MATLAB la velocidad de un programa compilado. Dentro de la estructura del algoritmo MRCD dibujada en la figura 4.2, los operadores `InPob`$(\cdot)$ y `Mezc`$(\cdot)$

son los encargados de llamar a las funciones del entorno MATLAB para conseguir los vectores objetivos.

Entorno gráfico. Los algoritmos MRCD ha sido plateados para el diseño de controladores. Con este fin se ha realizado un entorno gráfico que facilite al usuario inexperto su utilización. Este entorno consiste en una serie de pantallas realizadas con ayuda de las utilidades de MATLAB para la introducción de los parámetros que el algoritmo MRCD necesita y para analizar los resultados.

La figura 4.5 es la pantalla principal de este entorno. Está dividida en dos partes, una parte izquierda para la introducción de parámetros propios del algoritmo MRCD, y la parte derecha para la introducción de los datos del problema de control a resolver. Dentro de los parámetros del algoritmo se pueden asignar el número de generaciones, número de individuos, tipo de filtro, y todos los parámetros descritos en la sección 4.3

Los datos del problema de control son la estructura del modelo planta, controlador, las restricciones del problema y las normas entre diferentes entradas y salida a minimizar. Para introducir la estructura planta, controlador se ha usado el programa SIMULINK, del entorno MATLAB, donde el usuario debe dibujar de forma gráfica su diagrama de bloques. En el caso del algoritmo MRCD, los parámetros de la planta serán constantes y los del controlador constantes o términos de un vector K. En el algoritmo MRCD min-max tanto planta como controladores pueden tener parámetros variables, D y K respectivamente. El vector K representa a los parámetros del controlador, y el vector D a los parámetros de la incertidumbre de la planta. Se deberán definir unos límites en los espacios de búsqueda inicial de los diferentes parámetros D y K en las diferentes subpoblaciones. Esta representación da una gran libertad a la hora de elegir todo tipo de estructuras en la planta y el controlador. La optimización de la estabilidad en lazo cerrado siempre debe de ser un objetivo (restricción) a optimizar, como se vio en la sección 4.2. El problema (2.5) para el MRCD simple es la búsqueda de los parámetros K que minimizan las normas elegidas, y en el problema (2.6) para el MRCD min-max la búsqueda de los parámetros K que minimizan dichas normas para el peor caso D.

Tiempo de cálculo. El tiempo de cálculo es un de lo mayores problemas de los EAs. Los algoritmos MRCD tienen un tiempo de cálculo polinómico, debido a la necesidad de comparar

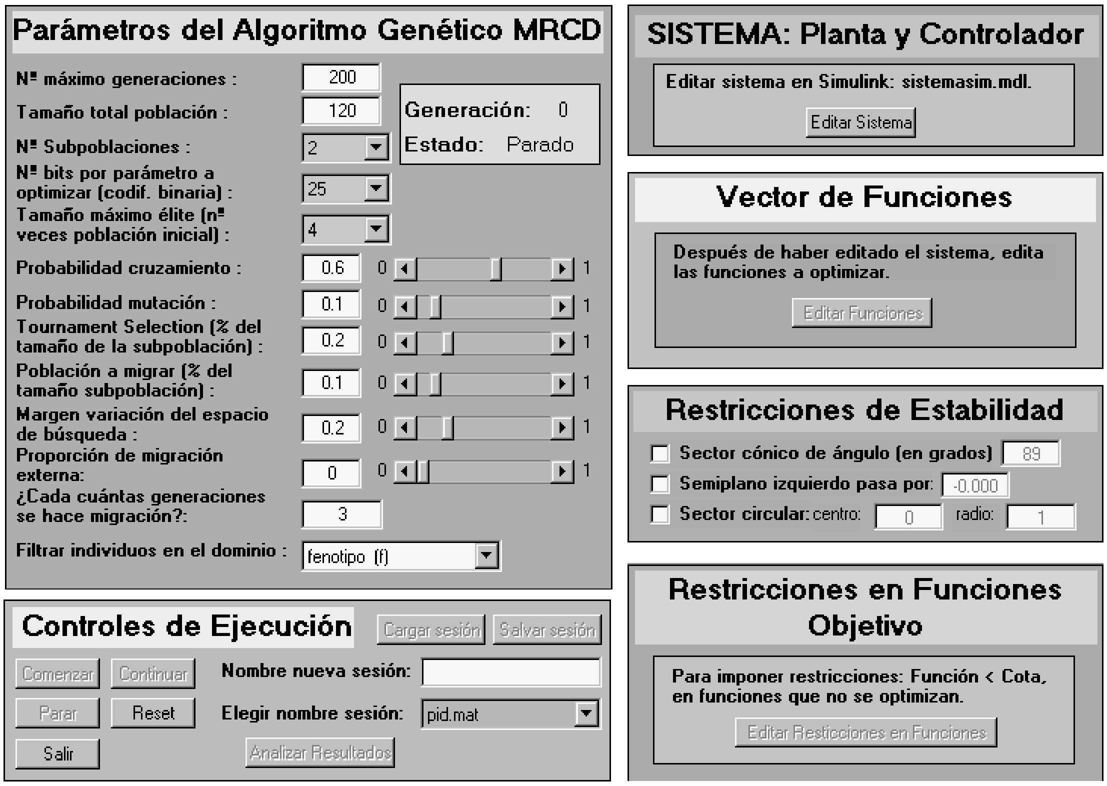


Figura 4.4: Pantalla principal de entorno gráfico.

todos los individuos de una subpoblación entre sí en la función “ranking”. Al estar dividida la población en subpoblaciones el costo de esta operación es menor. Pero el gran peso en el tiempo de cálculo de este algoritmo es la búsqueda del vector de funciones objetivo. Dependiendo del tipo de norma y orden del sistema el tiempo de cálculo va a ser diferente, siendo necesario llamar a las funciones correspondientes tantas veces como individuos tenga la población. El algoritmo MRCD es un “Algoritmo paralelo en isla”, por lo que puede ser usado para obtener los vectores de funciones de las diferentes subpoblaciones en paralelo. Esta puede ser una línea futura de investigación.

4.6 Conclusión.

En este capítulo se describen los algoritmos MRCD, que se pueden catalogar como GAs de optimización multiobjetivo cuya solución es una aproximación al conjunto de los óptimos de Pareto. El algoritmo MRCD simple ha sido diseñado para resolver el problema (2.5), donde los límites del espacio de búsqueda del controlador se desconocen a priori. Para paliar en lo posible este problema el MRCD simple usa una estructura de “Algoritmo paralelo en isla” y un espacio de búsqueda variable, que hace que los límites de dicho espacio se optimicen en el curso de las generaciones. El algoritmo MRCD min-max trata de resolver un problema min-max con GAs, hecho altamente novedoso. Con él se intenta resolver el problema de control robusto multiobjetivo plateado en (2.6), y su estructura es muy parecida al algoritmo MRCD simple menos en el filtro de la élite. Para ambos algoritmos se ha diseñado un entorno gráfico que facilita su uso por usuarios sin grandes conocimientos de GAs.

Capítulo 5

Evaluación del Algoritmo MRCD.

5.1 Introducción.

En la actualidad no existe una teoría exacta de la convergencia de los algoritmos evolutivos (EAs). La hipótesis de los "Bloques Constructivos (Building Blocks)", ver [Gol89a], trata de explicar el aumento de las buenas combinaciones de parámetros para el objetivo deseado por medio del operador selección y la posible destrucción de las mismas por los operados cruce y mutación, pero no tiene en cuenta la posible creación de nuevas combinaciones por medio de estos dos últimos operadores. Esta hipótesis no puede explicar el llamado "ruido colateral", que consiste en que individuos con una parte de sus parámetros buenos y con valores de la función objetivo relativamente altos, pueden entorpecer la optimización del conjunto de todos los parámetros, al introducir ruido en los parámetros en los que no son buenos. Los teoremas de "No Free Lunch" (NFL), ver [WM97] demuestran que un EA debe de llevar incorporada información del problema que quiere resolver para ser eficaz, lo que hace que la comparación en términos absolutos entre la bondad de dos EAs no tenga sentido, siendo necesaria su comparación ante un problema concreto.

Con estas premisas la única opción para poder valorar la bondad de un algoritmo es la aplicación de funciones test. Existe un gran catálogo de funciones test para problemas de optimización multiobjetivo, de las cuales se ha elegido siete funciones sin restricciones y tres con ellas para poder medir la eficiencia del algoritmo MRCD. El resultado de la prueba ha sido muy satisfactorio, igualando y mejorando en la mayoría de los casos a los algoritmos existentes.

Para ver las propiedades del algoritmo MRCD frente a problemas de control multiobjetivo se ha planteado un problema mixto $\mathcal{H}_2/\mathcal{H}_\infty$ con realimentación de estado y otro con realimentación a la salida. Este problema se ha resuelto para trece sistemas seleccionados, ver [APH98], para analizar los resultados. Con el fin de tener un criterio de medida se ha obtenido un límite superior de los respectivos frentes de Pareto con ayuda de las técnicas de inecuaciones lineales matriciales (LMI), ver apéndice D. Los resultados han sido también muy satisfactorios, consiguiendo con el algoritmo MRCD minimizar el límite superior del frente de Pareto obtenido con las técnicas de LMIs. Los controladores dinámicos se han diseñado con tres tipos de parametrizaciones; controlable, diagonal y balanceada, ver apéndice C, siendo la controlable la que mejores resultados ha dado. Por último se ha tratado de ver la eficacia del algoritmo MRCD al trabajar con controladores de diferentes ordenes, llegando a la conclusión que el algoritmo MRCD obtiene buenos controladores de orden reducido.

Este capítulo está dividido en las siguientes secciones. En la sección 5.2 se detallan las hipótesis sobre la convergencia de los EAs. En la sección 5.3 se analizan las funciones test, en general, y la funciones test multiobjetivo. En la sección 5.4 se prueba al algoritmo MRCD con diez funciones test para problemas de optimización multiobjetivo. En la sección 5.5 se formula dos problemas mixtos $\mathcal{H}_2/\mathcal{H}_\infty$ para trece diferentes sistemas y se analizan los resultados obtenidos mediante el algoritmo MRCD. En la sección 5.6 se formula un problema que es resuelto con controladores dinámicos de diferentes ordenes. En la sección 5.7 se proponen las conclusiones del capítulo.

5.2 Fundamentos teóricos de los algoritmos evolutivos (EAs).

Existen varias teorías respecto a los fundamentos teóricos que rigen la convergencia de los EAs, más en particular de los GAs. En ellas se pretende dar una explicación a la razón por la que un GA simple converge, y sobre las ventajas que su uso puede representar con respecto a otros tipos de algoritmos, como por ejemplo los métodos de “Climbing Hill” o los métodos de gradiente, ver sección 3.4. Existen numerosos libros donde se intenta dar una respuesta a estas cuestiones, [Hol92, Gol89a, Vos99, Mic92], pero de forma general este tema está aún abierto. En esta sección se tratará brevemente las diversas teorías para explicar la convergencia de un EA y las críticas que se pueden formular sobre las mismas.

5.2.1 Teoría de esquemas.

Esta teoría fue planteada por el padre de los GAs, Holland [Hol92], y está explicada ampliamente en [Gol89a]. La formulación de esta teoría se hizo pensando en un GA simple, ver sección 3.5, donde el genotipo de un individuo (parámetros codificados) está compuesto por una cadena de unos y ceros, que corresponden a la codificación binaria o en código Gray del fenotipo (parámetros) del individuo. En cualquier caso, esta teoría se puede generalizar para cualquier tipo de genotipo, como luego se verá.

Un esquema es una cadena de longitud l, la misma que la longitud del genotipo del individuo, y que puede tener tres tipos de caracteres, $\{0, 1, *\}$. Un esquema representa el conjunto de cadenas binarias que tienen los 0 y 1 en las mismas posiciones del esquema. Por ejemplo, el esquema $\{10{*}{*}1\}$ representa el conjunto de cuatro cadena binarias, $\{10001, 10011, 10101, 10110\}$. Se definen dos propiedades en un esquema, su longitud y su orden. La longitud definida de un esquema, $\delta(\cdot)$, es la distancia entre el índice del primer y último bit especificado. Por ejemplo $\delta(1*****0*) = 7-1 = 6$, o bien $\delta(1*******) = 1-1 = 0$. El orden de un esquema, $o(\cdot)$, es el número de bit especificados. Por ejemplo, $o(1*******) = 1$, o bien $o(11111111) = 8$.

Se define índice de bondad de un esquema H, $f(H)$, a la media de los índices de bondad de todos sus componentes. Un "Bloque Constructivo" ("Building Block") es un esquema de bajo orden, corta longitud definida y alto índice de bondad. Los "Bloques Constructivos" representan similitudes entre los genotipos de los individuos que serán significativos en la solución del problema. El proceso de selección elegirá a individuos cuyos fenotipos tengan esquemas de alto índice de bondad. El operador cruce tiene mayor posibilidad de romper esquemas con gran longitud definida que los que no la tienen, y el operador mutación tiene más posibilidades de modificar esquemas con orden alto. De estas apreciaciones se deduce que los "Bloques Constructivos" tienen gran posibilidad de incrementarse de población en población. Holland [Hol92] estimó que en una población de n individuos puede tener n^3 esquemas útiles. A este fenómeno lo definió como "paralelismo implícito", e implica que una gran población puede encontrar una solución en un tiempo computacional polinómicamente menor que una pequeña población.

Basándose en el problema "Two-Armed Bandit", ver [Gol89a], ampliamente estudiado en la teoría estadística de decisión y en control adaptativo se desarrollo una teoría, llamada "Teoría de

Esquemas", para explicar la convergencia de un GA. Se supone un esquema H, con un índice de bondad $f(H)$. Se define $m(H,t)$ como el número de cadenas (genotipos) que tienen ese esquema dentro de la población en un tiempo (t). El número de cadenas se va a conseguir con dicho esquema en el instante $(t+1)$ gracias al operador selección, suponiendo que la selección es proporcional al índice de bondad, es

$$m(H,t+1) = m(H,t)\frac{f(H)}{\overline{f}},$$

donde $\overline{f}$ es el índice de bondad medio de la población. Dos son las causas por la que las nuevas cadenas pertenecientes al esquema H pueden desaparecer, la aplicación de los operadores cruce y mutación. La posibilidad de que el operador cruce, con un solo punto de cruce, rompa un esquema en una cadena es proporcional a la longitud definida de dicho esquema,

$$ps \geq 1 - p_{cruce}\frac{\delta(H)}{l-1},$$

donde p_{cruce} es la probabilidad de cruce y l la longitud de la cadena. Esta inecuación refleja el hecho de que el operador cruce pueda no romper un esquema cuando el punto de cruce está en una determinada posición. El operador mutación también puede romper un esquema en una cadena, y en este caso, la probabilidad de que un esquema sobreviva en una determinada cadena es proporcional al orden del esquema,

$$pMs \approx 1 - o(H)p_{muta}, \quad p_{muta} \ll 1,$$

donde p_{muta} es la probabilidad de mutación. Combinando estos resultados y simplificando términos se llega a que los individuos que poseen un determinado esquema H en un instante $t+1$ son

$$m(H,t+1) \geq m(H,t)\frac{f(H)}{\overline{f}}\left[1 - p_c\frac{\delta(H)}{l-1} - o(H)p_m\right].$$

A partir de aquí Golderg [Gol89a] expuso los postulados de la hipótesis de los "Bloques Constructivos" ("Building Block Hypothesis"): "Los esquemas cortos, de bajo orden crecen exponencialmente en el curso de las generaciones".

Esta teoría puede ser aplicada para el caso de GAs que trabajen con valores reales como genotipos, en vez de con cadenas binarias, y a los EAs en general. En este caso se dispone de cadenas de m parámetros con valor real. Dando a cada parámetro un espacio de búsqueda definido por los límites $l_i \leq p_i \leq u_i$, un esquema S se define como

$$S \in [\alpha_i, \beta_i],$$

donde $l_i \leq \alpha_i \leq \beta_i \leq u_i$, pudiéndose aplicar el mismo razonamiento que en el caso anterior. Con ello se puede desarrollar una hipótesis de convergencia similar a la dada para otro tipo de algoritmos evolutivos (EAs).

La "Teoría de esquemas" describe de forma simplificada importantes aspectos del comportamiento de los GAs. En los genotipos de la población inicial hay una serie de esquemas con alto índice de bondad que van a ser determinantes en la convergencia del GA. Una alta probabilidad de mutación provoca la ruptura de esquemas de gran orden, y una alta probabilidad de cruce provoca la ruptura de esquemas con gran longitud definida. El proceso de selección hace que el algoritmo converja con un ratio proporcional a la división entre el mejor índice de bondad y la media del índice de bondad de la población.

Los parámetros de la probabilidad de cruce y de mutación deben ser elegidos de forma que exista un equilibrio entre la exploración de nuevos esquemas y la explotación de los existentes. La sola explotación de los esquemas existentes, por el operador selección y cruce, lleva a la convergencia prematura del algoritmo, y una exploración masiva a base de un gran índice de mutación degenera una mala convergencia. Es una filosofía muy parecida al problema "Two-Armed Bandit", donde se dispone de una máquina con dos opciones de premio, cuyos márgenes se desconocen. Siendo n el número de veces que se apuesta a la supuesta peor opción y N el número de apuestas totales, se llega a la conclusión que el número de apuestas óptimo n^* para perder el menor dinero posible es $N - n \approx e^{cn^*}$, ver detalles en [Gol89a]. Es decir, se debe ir incrementando las apuestas a la supuesta mejor opción de forma exponencial al número de apuestas a la supuesta peor opción. Es el equilibrio entre la explotación y la exploración de un espacio de búsqueda.

5.2.2 Crítica de la "Teoría de esquemas".

La hipótesis de "Bloques Constructivos" trata a los diferentes esquemas de una población como si fueran independientes, con lo cual se gana en simplicidad pero se comenten varios fallos:

Convergencia colateral Cuando una población empieza a converger hacia un esquema, el muestreo de otros esquemas no será uniforme. Así, si el esquema $111 * \ldots *$ tiene un buen índice de bondad y la población tiende hacia él, los elementos que tengan el esquema $***000*\ldots*$ va a tender a ser $111000*\ldots*$, con lo que ya no se podrá hablar de las propiedades del esquema $***000*\ldots*$ como tal, por tener un sesgo. A este fenómeno se le llama también "ruido colateral".

Alta varianza en el índice de bondad Si el índice de bondad de un esquema tiene alta varianza la estimación de éste no será exacta. Por ejemplo, sea la función objetivo

$$f(x) = \begin{cases} 2 & \text{Si } x \in 111*\ldots*, \\ 1 & \text{Si } x \in 0*\ldots*, \\ 0 & \text{En otro caso.} \end{cases}$$

La varianza del índice de bondad del esquema $1*\ldots*$ es alta, y la población va a converger sólo a una subpoblación de este esquema, por lo que el índice de bondad de dicho esquema no da gran información.

Creación de nuevos esquemas. En la hipótesis del "Bloques Constructivos" se habla del efecto positivo del operador selección en la búsqueda de esquemas con buen índice de bondad, y del efecto negativo de los operadores cruce y mutación en la ruptura de los esquemas encontrados. Sin embargo, el operador cruce puede crear esquemas de longitud definida mayor a partir de esquemas más pequeños.

En [Gol89a] se define como función deceptiva, de forma general, a la que origina problemas en la convergencia de un GA. Este tipo de funciones suelen estar asociadas a funciones multimodales (con varios subóptimos) o con un óptimo aislado rodeado de puntos con bajo índice de bondad. En ellas existe un óptimo global y otros puntos que sin serlo atraen a la población. La convergencia de GA simple con este tipo de problemas puede tender hacia un subóptimo, pudiendo dar peores resultados que un algoritmo de muestreo puro, como por ejemplo la "Simulación de Monte Carlo", ver sección 3.4.3. La "Teoría de esquemas" trata principalmente de dar una explicación a este tipo de fenómenos. Sin embargo, el hecho de que la función a optimizar sea deceptiva, ver [Gol89a] no es causa necesaria ni suficiente para que un GA converjan mal, como se indica en [Mit99], habiendo ejemplos de funciones deceptivas donde los GAs actúan muy bien y otros de funciones no deceptivas donde esto no sucede, como las función "Royal Road" que más tarde se estudiarán. El comportamiento de los GAs en estas funciones demuestra que su dinámica es más compleja que la mostrada en la "Teoría de esquemas", teniéndose que hablar de comportamiento defectivo de un GA, en vez de funciones deceptivas.

Funciones "Royal Road". Son funciones creadas para analizar el comportamiento de los GAs y verificar la exactitud de la "Teoría de esquemas". La hipótesis de "Bloques Constructivos" sugiere que en el curso de las generaciones aumentan los esquemas de longitud definida corta, con bajo orden y alto índice de bondad. Los esquemas de mayor longitud definida y mayor orden se crearán por la combinación de "Bloques Constructivos". La función $R_1(x)$ ha sido diseñada para comprobar este efecto, ver [Mit99], definiéndose para una gran cadena x como

$$R_1(x) = \sum_i c_i \delta_i(x), \quad \text{donde } \delta_i(x) = \begin{cases} 1 & \text{Si } x \in s_i \\ 0 & \text{En otro caso.} \end{cases}$$

Un ejemplo de esta función es el caso de una cadena con 32 bits, donde se tiene la siguiente composición de esquemas.

$$\begin{aligned}
s_1 &= 1111****************************; & c_1 &= 8 \\
s_2 &= ****1111************************; & c_2 &= 8 \\
s_3 &= ********1111********************; & c_3 &= 8 \\
s_4 &= ************1111****************; & c_4 &= 8 \\
s_5 &= ****************1111************; & c_5 &= 8 \\
s_6 &= ********************1111********; & c_6 &= 8 \\
s_7 &= ************************1111****; & c_7 &= 8 \\
s_8 &= ****************************1111; & c_8 &= 8
\end{aligned}$$

$$s_{opt} = 11111111111111111111111111111111$$

Si la hipótesis de "Bloques Constructivos" fuera válida, esta función sería muy sencilla de optimizar por medio de GAs, ya que se encontrarían "Bloques Constructivos" cortos, de bajo orden y buen índice de bondad, y como unión de los mismo se llegaría la solución final. En [Mit99] se analiza el problema comparándolo con diversas versiones estocásticas del algoritmo "Climbing Hill" estudiado en la sección 3.4.2. El resultado es aparentemente contradictorio ya que las versiones del algoritmo "Climbing Hill" encuentran la solución en un periodo de tiempo diez veces menor que con el GA simple.

La razón de esta mala convergencia se centra en la convergencia colateral no tenida en cuenta en la hipótesis de "Bloques Constructivos". Supongamos que los esquemas s_1 y s_2 están dentro de la población inicial, y el esquema s_3 no lo está. Los dos primeros crecerán exponencialmente en el curso de las generaciones. Pero este crecimiento llevará implícito el crecimiento colateral de individuos que no pertenezcan al esquema s_3, ya que este esquema no está en la población inicial. De esa forma, la explotación de esquemas contenidos en la población inicial crea, cuando los esquemas no están relacionados, un perjuicio en la exploración de esquemas no contenidos en dicha población inicial. A este fenómeno se le denomina "falsa correlación", ya que parece como si la pertenencia al esquema s_1 y s_2 implicará no pertenecer a s_3. Este efecto será de gran utilidad en la interpretación de resultados tratados con los algoritmos MRCD.

5.2.3 Modelo matemático exacto de un GA.

La "Teoría de esquemas" predice los cambios en el número de esquemas de una población a la siguiente, pero no predice directamente la composición de la población, la velocidad de convergencia o la distribución del índice de bondad. En [Vos99] se construye un modelo matemático "exacto" para GAs simples. Con él se pretende probar propiedades de los operadores selección, cruce y mutación del algoritmo.

El algoritmo propuesto para el estudio es muy similar al GA simple descrito en el pseudocódigo de la figura 3.9. Se parte de una población inicial aleatoria de cadenas binarias de longitud l, a partir de aquí:

1. Calcular el índice de bondad $f(x)$ de cada cadena.
2. Elegir dos padres de la población con probabilidad proporcional al índice de bondad relativo al de la media de la población.
3. Cruzar a los dos padres obteniendo dos hijos, de los cuales se elegirá uno aleatoriamente.
4. Mutar la cadena seleccionada con una probabilidad p_m.
5. Ir al paso 2 hasta que la nueva población se complete.
6. Ir al paso 1.

Cada cadena del espacio de búsqueda se identifica por un entero $i \in [0, 2^l - 1]$, por ejemplo para $l = 8$ la cadena 00000111 está asociado al entero 7. La población tiene asociados dos vectores $p(t)$ y $s(t)$, de tamaño 2^l. La componente $p_i(t)$ es proporcional a los elementos que la cadena i tiene en la población en el instante t. La componente $s_i(t)$ es la probabilidad que tiene la cadena i en el instante t de ser elegida. Definiendo una matriz F de dos dimensiones tal que $F_{i,j} = 0, i \neq j$, y $F_{i,i} = f(i)$, se puede relacionar $s(t)$ y $p(t)$ de la siguiente forma

$$s(t) = \frac{Fp(t)}{\sum_{j=0}^{2^l-1} F_{i,j} p_j(t)}.$$

A partir de estas premisas y tras una serie de cálculos más o menos laborioso se llega a formular el problema como

$$G(s(t)) \sim s(t+1) \quad G_p(p(t)) = p(t+1),$$

siendo

$$G = F \circ \mathcal{M} \quad G_p(x) = \mathcal{M}(Fx/|Fx|).$$

El operador "$\sim$" indica que uno y otro término se diferencian en un factor escalar y el operador "$\circ$" indica composición. La componente $\mathcal{M}_{i,j}$ de la matriz $\mathcal{M}$ es la probabilidad de paso de la cadena i a la j por efecto de la mutación y cruce, y la matriz F está relacionada únicamente con el operador selección, como se vio en la relación entre $s(t)$ y $p(t)$.

Utilizando este modelo, se ha estudiado la dinámica del operador $G(\cdot)$ sobre el vector de estado s. El problema general no ha sido resuelto pero en su lugar se han obtenido algunos resultados para la dinámica de los operadores F y $\mathcal{M}$ independientemente. El operador F hace converger a la población hacia individuos que maximizan el valor de la función objetivo, mientras que el operador $\mathcal{M}$ lleva a la población hacia una situación de máxima dispersión de sus individuos. Existe un efecto opuesto entre el operador cruce y mutación ($\mathcal{M}$) y el operador selección (F) que refleja el proceso de búsqueda de los GAs que se mueve entre la exploración y la explotación de la información contenida en la población.

Otros autores han tratado de modelar un GA simple usando métodos de mecánica estadística, aplicando las leyes de presión y temperatura que sirven para explicar de forma macroscópica fenómenos microscópicos de relación entre partículas. Ver los detalles de estos procedimientos de modelado exacto de GAs en [Mit99] y sus referencias.

5.2.4 Convergencia de un algoritmo evolutivo (EA).

En [Bäc96] se demuestra la convergencia con una probabilidad uno de un EA hacia un óptimo global en los siguientes términos. Se tiene una variable de decisión x perteneciente a un espacio de búsqueda Ω, una función de índice de bondad $\Phi(x)$ y un número de generaciones t. Un EA converge hacia la solución óptima global si se cumple que para $\{\forall x \in \Omega, \quad x' \in \Omega\}$, x' puede ser estructurada desde x a base de mutación y cruce, y que la secuencia de poblaciones $\{P(0), P(1), \ldots\}$ es monótona, lo que quiere decir que

$$\forall t : \min(\Phi(x(t+1))| \ x(t+1) \in P(t+1)) \leq \min(\Phi(x(t))| \ x(t) \in P(t)).$$

La restricción de población monótona se cumple en el algoritmo estrategia de evolución simples $((1+1)-ES)$, ver sección 3.5, donde un padre no es reemplazado más que por un hijo con mejor índice de bondad que él.

En [VV99b] se plantea un teorema similar para el caso del problema de optimización multiobjetivo. Si $\forall x \in \Omega, \quad x' \in \Omega$, x' puede ser estructurada desde x a base de cruce y mutación, y la población $P(t+1)$ mantiene o mejora el frente de Pareto de la población $P(t)$, siendo $P(0) \neq \emptyset$, entonces el algoritmo converge hacia el conjunto de óptimos de Pareto con probabilidad uno. En [Rud98] se demuestra que un algoritmo $(1+1)-ES$, ver sección 3.5.2, converge a un elemento del frente de Pareto con probabilidad uno. Este algoritmo es robusto pero puede ser lento.

Ambas conclusiones sólo son aceptables desde un punto de vista teórico, ya que la seguridad de obtener un óptimo se da cuanto $t \to \infty$.

5.2.5 Teoremas "No Free Lunch" (NFL).

Son una serie de teoremas desarrollados en [WM97] que muestran la razón por la que los algoritmos de muestreo dirigido deben incorporar información sobre el problema que se desea resolver para que sean efectivos. Un algoritmo de muestreo dirigido es aquel que parte de una función de distribución uniforme en el muestreo de un espacio de búsqueda, y que irá modificando dicha función a raíz de los resultados obtenidos en el muestreo anterior. Dentro de este tipo de algoritmo están todos los EAs, los algoritmos de "Búsqueda Tabú" y los algoritmos de "Simulated Anneling" vistos en la sección 3.4.3.

Los teoremas NFL demuestran que dos algoritmos A y B de muestreo dirigido tienen como media las mismas características en la resolución de todos los problemas de optimización posibles. Esto implica que un algoritmo dirigido necesita información añadida sobre el problema para el que ha sido diseñado, y que las buenas cualidades que tenga en ese tipo de problemas vienen unidas a malas cualidades en la resolución de otro tipo de problemas, para los cuales no ha sido diseñado. Por tanto, la evaluación de dos EAs no tiene sentido, debiéndose evaluar con respecto a un problema concreto.

5.3 Funciones test para problemas multiobjetivo.

Tras la demostración de los teoremas NFL, ver sección 5.2, la comparación global (para todo tipo de problemas) de las especificaciones de dos EAs no tiene sentido. Estos teoremas implican que los EAs deben incorporar conocimiento del tipo de problema que se quiere resolver para que sean efectivos. Este hecho ha llevado a valorar las especificaciones de varios EAs a partir de los resultados que los mismos ofrecen ante diferentes funciones test. Con los resultados de los algoritmos ante estas funciones test se puede inferir las propiedades que los mismos poseen. Por supuesto, el hecho de que un algoritmo funcione bien ante una función test no da la plena seguridad de que lo vaya a hacer con otra función equivalente, lo que hace que su uso y eficiencia sean discutidos.

Existen gran número de funciones test en el estudio de EAs aplicados a la optimización global. Dentro de las más conocidas están las cinco funciones test de De Jong [Gol89a] y las de Michalewicz [Mic92]. Las cinco funciones test de De Jong tienen una o varias de las siguientes características: son continuas o discontinuas, convexas o no convexas, unimodales o multimodales, cuadráticas o no cuadráticas, de baja o de alta dimensión, y deterministas o no deterministas. Con los resultados del análisis de los EAs por estas funciones, se puede tener una idea del comportamiento que los mismos tendrán ante funciones con similares características.

En [VV99b] se muestran treinta funciones test formuladas por diferentes autores para los problemas de optimización multiobjetivo. Muchas de ellas han sido diseñadas basándose en las funciones test de problemas con un sólo objetivo. Las características que estas funciones presentan pueden residir en la forma de su óptimo de Pareto (espacio de búsqueda), o en la forma de su frente de Pareto (funciones objetivos). Ambos conjuntos pueden ser conexos o no conexos, simétricos o no simétricos, y tener una dimensión de parámetros determinada. Estas funciones pueden tener restricciones en el espacio de búsqueda, y su frente de Pareto puede ser cóncavo o convexo.

Deb, ver [Deb98], analiza las funciones cuyos óptimos globales son más difíciles de hallar por EAs: funciones multimodales, deceptivas, óptimo aislado y ruido colateral, ver sección 5.2. Este autor propone una forma de traducir estas funciones al caso de problemas multiobjetivo. Los principales problemas que pueden presentar la convergencia hacia el frente de Pareto de un EAs multiobjetivo son: su convexidad o no convexidad, su continuidad y su distribución no uniforme. Además hay que tener en cuenta las restricciones respecto del espacio de búsqueda. La formulación general que Deb propone para una función test con dos objetivos y un espacio de búsqueda de dimensión N es,

$$\begin{array}{l} \min_x f_1(x) = f_1(x_1, \ldots, x_m) \\ \min_x f_2(x) = g(x_{m+1}, \ldots, x_N) h(f_1(x_1, \ldots, x_m), g(x_{m+1}, \ldots, x_N)). \end{array} \tag{5.1}$$

A partir de esta formulación, Deb deduce las siguientes características del problema multiobjetivo propuesto,

- La convexidad y discontinuidad del frente de Pareto puede ser modificada con la variación de la función $h(\cdot)$.
- La convergencia hacia el verdadero frente de Pareto viene determinada por el tipo de función $g(\cdot)$ elegida, ya sea porque esta función es deceptiva o multimodal.
- La diversidad en el frente de Pareto viene determinada en la elección de la función $f_1(\cdot)$, ya sea por no ser lineal o ser multidimensional.

En el citado artículo se dan ejemplos de funciones test multiobjetivo que poseen la estructura mencionada.

5.3.1 Funciones test elegidas.

En [VV99b] se citan treinta funciones test para problemas de optimización multiobjetivo. De ellas el autor elige siete funciones sin restricciones en el espacio de búsqueda, ver tabla 5.1, y tres con restricciones en dicho espacio de búsqueda, ver tabla 5.2. En la citada tesis se compara el algoritmo formulado por el autor (MOMGA) con varios de los GA multiobjetivos planteados en la sección 3.7.2, MOGA, NPGA y NSGA. Con estos resultados se realiza un estudio sobre la bondad del algoritmo MOMGA respecto a los otros.

En esta sección se va a describir las características que poseen las funciones test elegidas, empezado por las que no tienen restricciones.

Función de Schaffer. Es un clásico en los problemas multiobjetivo por ser Schaffer quien desarrollo el primer GA multiobjetivo (VEGA), ver sección 3.7.1. El frente de Pareto tiene una expresión analítica, siendo convexo y continuo. Puede dar problemas por los grandes límites del espacio de búsqueda y es útil para ver la diversidad en la población a lo largo del frente de Pareto, ver sección 3.6.1.

Función de Fonseca. Su espacio de búsqueda es de dimensión variable, pero la forma del frente de Pareto es similar para todas las dimensiones, siendo cóncavo y continuo.

Función de Poloni. Es un problema de maximización con dos objetivos y un espacio de búsqueda de dos dimensiones. Tanto el conjunto de óptimos de Pareto como el frente de Pareto están divididos en dos zonas bien diferenciadas.

Función de Kursawe. Es un problema de dos funciones objetivos y un espacio de búsqueda de dimensión variable. El conjunto de óptimos de Pareto está dividido en varias zonas de diferente extensión. El frente de Pareto está dividido en tres zonas bien diferenciadas.

Función de Viennet (1). Tiene tres vectores objetivos compuestos de funciones no lineales y un espacio de búsqueda de dos dimensiones. Tanto el conjunto de óptimos de Pareto como el frente de Pareto están divididos en zonas de distinta extensión.

Función de Deb. Es una función diseñada por el método general descrito en el apartado anterior (5.1). Tanto el conjunto de óptimos de Pareto como el frente de Pareto están divididos en regiones del mismo tamaño.

Función de Viennet(2). Tiene tres vectores objetivos compuestos por funciones no lineales y un espacio de búsqueda de dos dimensiones. La forma del conjunto de óptimos de Pareto y la del frente de Pareto es atípica.

Las funciones test con restricciones en el espacio de búsqueda son las siguientes.

Función de Binh. Es una función de dos dimensiones en el espacio de búsqueda y en el de vectores objetivos. Tiene un frente de Pareto convexo y las restricciones del espacio de búsqueda no son muy duras, ya que en una búsqueda aleatoria de 40000 puntos, 36921 de ellos cumplen con las mismas.

Función de Osyczka. La principal característica de esta función es que tiene restricciones severas. Tiene seis restricciones tales que en una búsqueda aleatoria de 40000 puntos sólo 1389 cumplen con las mismas. Su frente de Pareto tiene una forma bastante irregular, con diversas áreas de distintos tamaños.

Función de Viennet (3). Tiene un frente de Pareto de tres dimensiones dividido en varias zonas y un espacio de búsqueda de dos dimensiones de una forma peculiar. En una búsqueda aleatoria de 40000 puntos, 6378 cumplen con las restricciones.

5.4 Evaluación del algoritmo MRCD con funciones test.

El algoritmo MRCD ha sido creado para el diseño de controladores robustos multiobjetivo, ver capítulo 4. Por ello va a ser evaluado respecto a las funciones test propuestas en [VV99b] en esta sección, y respecto a otros métodos de diseño de controladores multiobjetivo en la sección 5.5.

Las funciones test planteada tienen límites definidos en el espacio de búsqueda, que por otro lado no son muy grandes excepto en la función de Shaffer. Ello hace que el opearador "Algoritmo paralelo en isla", ver sección 4.3 no sea interesante para la resolución de estos problemas. Por ello, los límites iniciales de las cinco subpoblaciones propuestas van a ser iguales a los límites de cada problema. En las 20 primeras generaciones no se ha aplicado el operador dinámico de variación del espacio de búsqueda, ver sección 4.3 y en las restantes sí se ha hecho. En las 100 primeras generaciones se ha aplicado filtro en la élite con respecto al frente de Pareto, ver sección 4.3, quitando este filtro a partir de entonces. La condición de parada propuesta ha sido conseguir 1000 individuos en la élite, hecho que en ningún caso a supuesto más de 150 generaciones. La aplicación del operador élite sin filtro sólo tiene sentido en una última fase de convergencia, ya que un número excesivo en la élite puede provocar convergencia prematura. Se ha tomado una probabilidad de cruce $p_{cruce} = 0.7$, una probabilidad de mutación $p_{muta} = 0.1$, y la $t_{dom} = 5$, ver el pseudocódigo 3.10.

Las figuras 5.1-5.10 muestran los resultados obtenidos. Las figuras de la derecha muestran los vectores objetivos del frente de Pareto, ($\Box$), y vectores objetivo aleatorio obtenidos en los límites del espacio de búsqueda inicial, 40000 puntos ($\cdot$), que permiten valorar la dificultad de cada problema. Las figuras de la izquierda muestran el conjunto de óptimos de Pareto ($\cdot$) asociados al frente de Pareto ($\square$).

En [VV99b] se dan las verdaderas soluciones de las funciones test elegidas y se comparan, en algunas de dichas funciones, los diferentes resultados obtenidos por medio de los algoritmos MOGA, MOMGA, NPGA y NSGA, sección 3.7.2. A continuación se compara los resultados obtenidos con el algoritmo MRCD, figuras 5.1-5.10, con los verdaderos frentes de Pareto y en los casos que proceda, con los resultados de los algoritmos mencionados.

Función de Shaffer. Su máximo problema es el gran espacio de búsqueda. Los algoritmos NPGA y NSGA apenas si consiguen llegar al verdadero frente de Pareto con varios individuos, el algoritmo MOGA llega pero no lo cubre en su totalidad y el MOMGA lo cubre entero. El algoritmo MRCD al modificar los límites del espacio de búsqueda de forma dinámica consigue sin problemas abarcar todo el frente de Pareto.

Función de Fonseca Todos los algoritmos, excepto el NSGA, llegan sin problemas al frente de Pareto, pese a que un muestreo uniforme de 40000 puntos no consigue puntos próximos a dicho frente. El algoritmo MRCD tiende hacia el verdadero frente de Pareto sin problemas.

Función de Poloni. Todos los algoritmos consiguen un gran número de individuos en los dos frentes de Pareto, pero sólo el algoritmo MOMGA consigue dos puntos de un tercer frente. El algoritmo MRCD también consigue puntos en este tercer frente.

Función de Kursawe. En ella los algoritmos MOMGA y NPGA llegan al verdadero frente de Pareto y los algoritmos MOGA y NSGA consiguen la forma del mismo, pero con un pequeño desplazamiento. El algoritmo MRCD consiguen individuos en todas los partes en las que queda dividido el conjunto de óptimos de Pareto pero al igual que el algoritmo MOGA y NSGA su frente de Pareto queda algo desplazado.

Función de Viennet (1). En ella el algoritmo MRCD llega sin problema a todas las partes del verdadero frente de Pareto y los óptimos de Pareto asociados.

Función de Deb En ella el algoritmo MRCD llega a todas las partes del verdadero frente de Pareto al igual que el resto de los otros algoritmos.

Función de Viennet (2). El algoritmo MRCD distribuye a los individuos de forma homogénea en las diversas ramas del verdadero frente de Pareto.

Función de Bihn, Osyscka y Viennet (3). En todas ellas el algoritmo MRCD incorpora las restricciones como otro objetivo, ver sección 4.3. Este objetivo tiene como valor la máxima distancia que separa a un individuo de cumplir alguna de las restricciones. Si el individuo cumple con las restricciones el valor del objetivo es cero, o infinito, dependiendo de si se está maximizando o minimizando. La población inicial se elige aleatoriamente, lo que hace que en las funciones Osyscka y Viennet (3) no haya un solo individuo que cumpla con las restricciones. Pese a ello, y con ayuda del nuevo objetivo las poblaciones convergen sin problemas hacia los verdaderos frentes de Pareto, con una población que cumple con los objetivos de problema.

La conclusión a la vista de los resultados obtenidos es que el algoritmo MRCD es capaz de resolver las funciones test sin grandes problemas hallando puntos a lo largo de todo el verdadero frente de Pareto de forma uniforme. La influencia que los diversos operadores tienen en esta convergencia es la siguiente:

- El operador "Algoritmo paralelo en isla" no ha sido de gran ayuda en este caso por estar los límites del espacio de búsqueda bien definidos.
- El operador variación dinámica de los límites del espacio de búsqueda ha servido para obtener una mayor precisión en la solución, sobre todo en la función Shaffer donde el límite de espacio de búsqueda es muy grande.
- El operador élite, y su filtro por nichos en el frente de Pareto, ha hecho que se obtenga una población uniforme a lo largo de todos los frentes de Pareto.
- La formulación de las restricciones como un objetivo más ha hecho que se haya podido resolver problemas donde sólo un 3.47% de los individuos pertenecientes al espacio de búsqueda cumplían las mismas.

Estas funciones test son, en suma, ejemplos de las buenas cualidades del algoritmo MRCD en la resolución de problemas de optimización multiobjetivo, y muestran la bondad del mismo en comparación con los más conocidos algoritmos realizados hasta la fecha.

Nombre	Definición	Límites
Schaffer	$F=(f_1(x,y),f_2(x,y));$ $f_1(x,y)=x^2,$ $f_2(x,y)=(x-2)^2$	$-10^5 \le x \le 10^5$
Fonseca	$F=(f_1(x),f_2(x));$ $f_1(x)=1-exp(-\sum_{i=1}^{n}(x_i-\frac{1}{\sqrt{n}})^2),$ $f_2(x)=1-exp(-\sum_{i=1}^{n}(x_i+\frac{1}{\sqrt{n}})^2)$	$-4 \le x_i \le 4; i=1,2,3$
Poloni	$F=(f_1(x),f_2(x));$ $f_1(x,y)=-[1+(A_1-B_1)^2+(A_2-B_2)^2],$ $f_2(x,y)=-[(x+3)^2+(y+1)^2]$	$-3.1416 \le x,y \le 3.1416,$ $A_1 = 0.5sin(1)-2cos(1) +sin(2)-1.5cos(2),$ $A_2 = 1.5sin(1)-cos(1) +2sin(2)-0.5cos(2),$ $B_1 = 0.5sin(x)-2cos(x) +sin(y)-1.5cos(y),$ $B_2 = 1.5sin(x)-cos(x) +2sin(y)-0.5cos(y)$
Kursawe	$F=(f_1(x),f_2(x));$ $f_1(x)=\sum_{i=1}^{n-1}\left(-10exp\left((-0.2)\cdot\sqrt{x_i^2+x_{i+1}^2}\right)\right),$ $f_2(x)=\sum_{i=1}^{n}(\|x_i\|^{0.8}+5sin(x_i)^3)$	$-5 \le x_i \le 5; i=1,2,3$
Viennet (1)	$F=(f_1(x,y),f_2(x,y),f_3(x,y));$ $f_1(x,y)=0.5\cdot(x^2+y^2)+sin(x^2+y^2),$ $f_2(x,y)=\frac{(3x-2y+4)^2}{8}+\frac{(x-y+1)^2}{27}+15,$ $f_3(x,y)=\frac{1}{(x^2+y^2+1)}-1.1exp(-x^2-y^2)$	$-3 \le x,y \le 3$
Deb	$F=(f_1(x,y),f_2(x,y));$ $f_1(x,y)=x,$ $f_2(x,y) = (1+10y) \cdot[1-\left(\frac{x}{1+10y}\right)^{\alpha}-\frac{x}{1+10y}sin(2\pi qx)]$	$0 \le x,y \le 1,$ $q=4,$ $\alpha=2$
Viennet (2)	$F=(f_1(x,y),f_2(x,y),f_3(x,y));$ $f_1(x,y)=\frac{(x-2)^2}{2}+\frac{(y+1)^2}{13}+3,$ $f_2(x,y)=\frac{(x+y-3)^2}{36}+\frac{(-x+y+2)^2}{8}-17,$ $f_3(x,y)=\frac{(x+2y-1)^2}{175}+\frac{(2y-x)^2}{17}-13$	$4 \le x,y \le 4$

Tabla 5.1: Funciones test sin restricciones en el espacio de búsqueda

Nombre	**Definición**	**Restricciones**
Bihn	$F=(f_1(x,y),f_2(x,y));$ $f_1(x,y)=4x^2+4y^2,$ $f_2(x,y)=(x-5)^2+(y-5)^2$	$0\leq x\leq 5, 0\leq y\leq 3,$ $0\geq (x-5)^2+y^2-25,$ $0\geq -(x-8)^2-(y+3)^2+7.7$
Ozsyczka	$F=(f_1(x),f_2(x));$ $f_1(x) = -(25(x_1-2)^2+(x_2-2)^2+(x_3-1)^2 +(x_4-4)^2+(x_5-1)^2),$ $f_2(x)=x_1^2+x_2^2+x_3^2+x_4^2+x_5^2$	$0\leq x_1,x_2,x_6\leq 10,$ $1\leq x_3,x_5\leq 5,$ $0\leq x_4\leq 6,$ $0\leq x_1+x_2-2,$ $0\leq 6-x_1-x_2,$ $0\leq 2-x_2+x_1,$ $0\leq 2-x_1+3x_2,$ $0\leq 4-(x_3-3)^2-x_4,$ $0\leq (x_5-3)^2+x_6-4$
Viennet (3)	$F=(f_1(x,y),f_2(x,y),f_3(x,y));$ $f_1(x,y)=\frac{(x-2)^2}{2}+\frac{(y+1)^2}{13}+3,$ $f_2(x,y)=\frac{(x+y-3)^2}{175}+\frac{(2y-x)^2}{17}-13,$ $f_3(x,y)=\frac{(3x-2y+4)^2}{8}+\frac{(x-y+1)^2}{27}+15$	$-4\leq x,y\leq 4,$ $y<-4x+4,$ $x>-1,$ $y>x-2$

Tabla 5.2: Funciones test con restricciones en el espacio de búsqueda

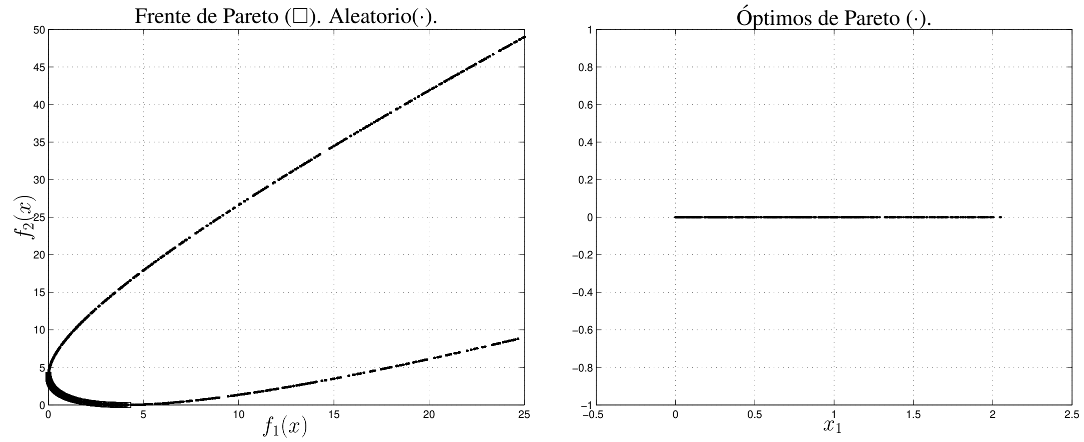


Figura 5.1: Función test de Shaffer

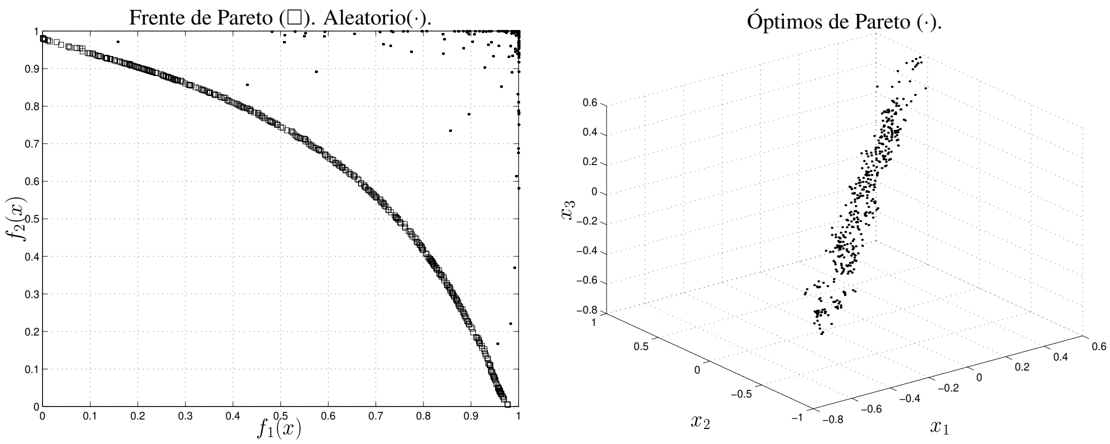


Figura 5.2: Función test de Fonseca

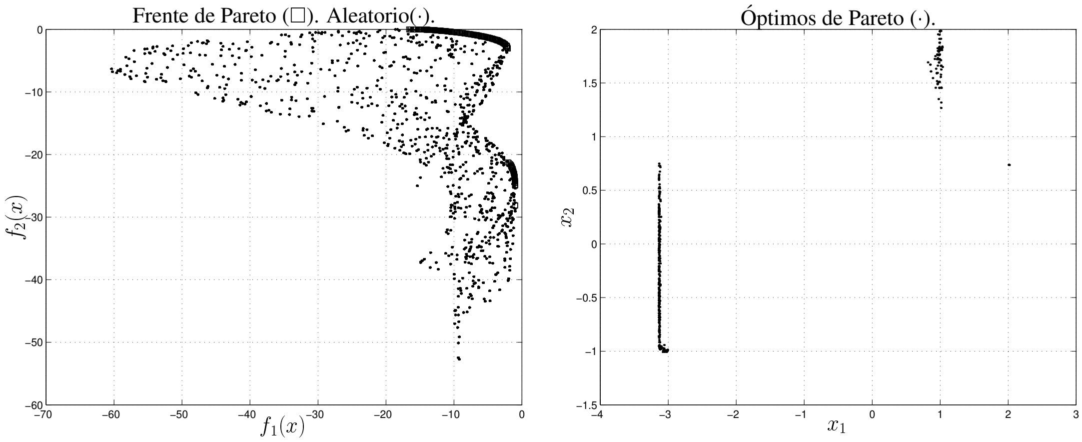


Figura 5.3: Función test de Poloni

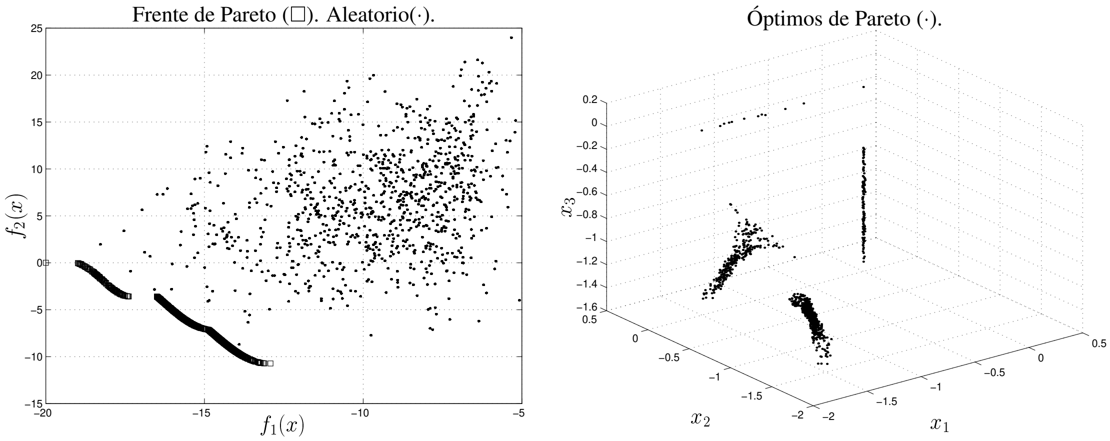


Figura 5.4: Función test de Kursawe

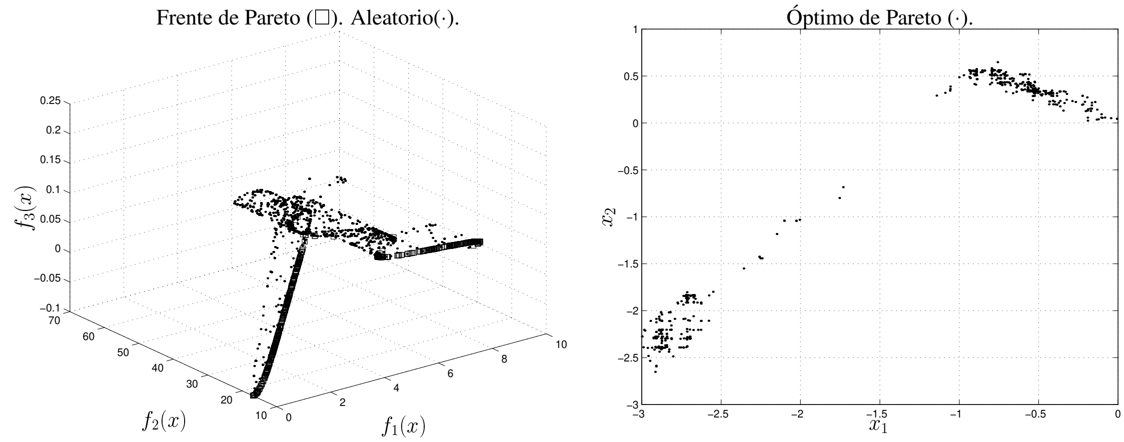


Figura 5.5: Función test de Viennet (1)

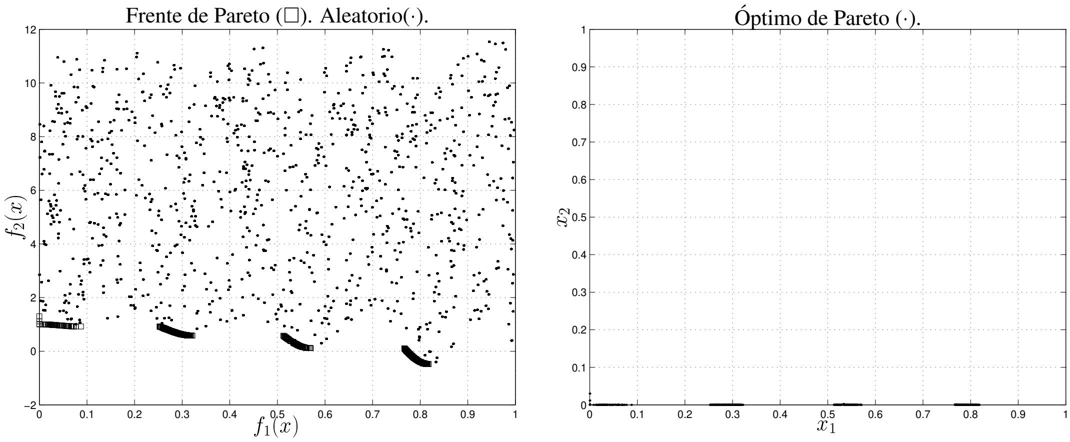


Figura 5.6: Función test de Deb

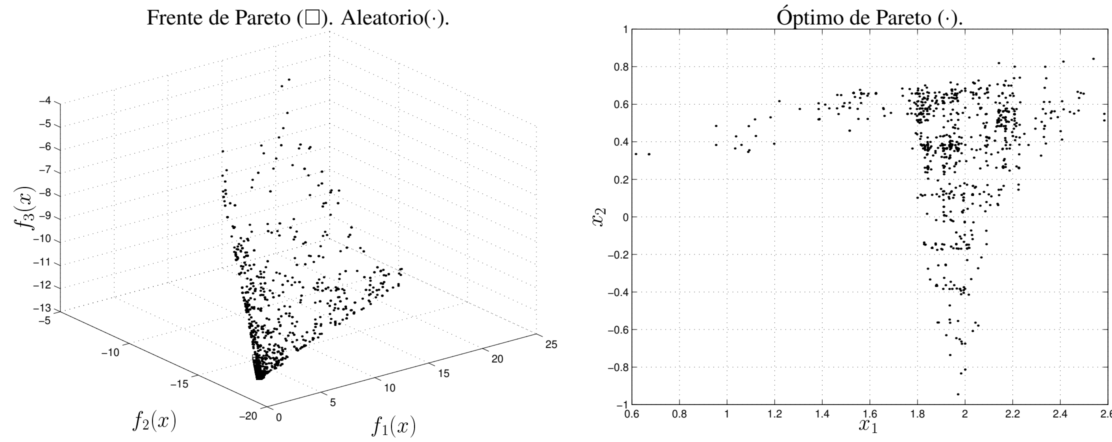


Figura 5.7: Función test de Viennet's (2)

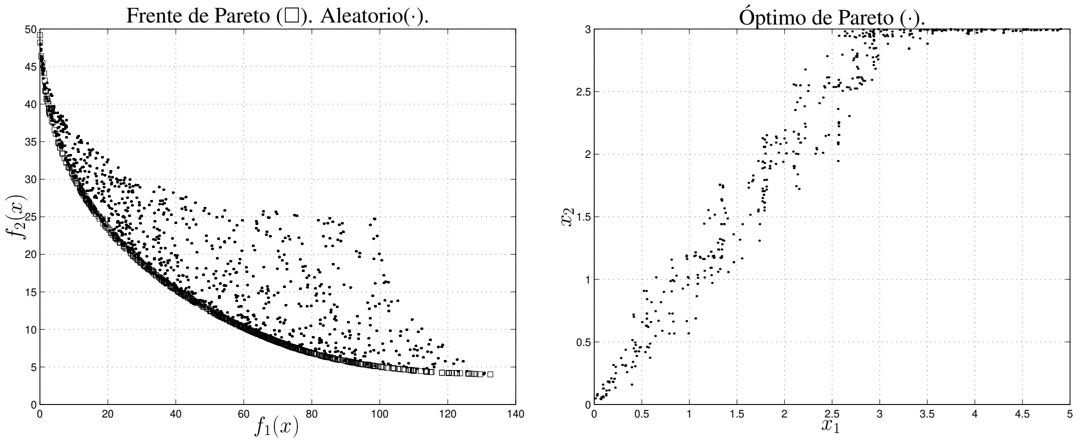


Figura 5.8: Función test de Binh

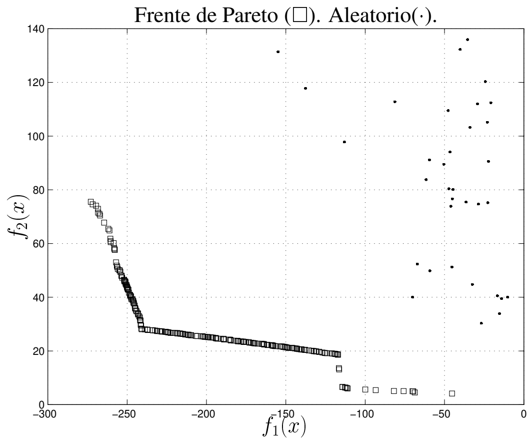


Figura 5.9: Función test de Osyczka

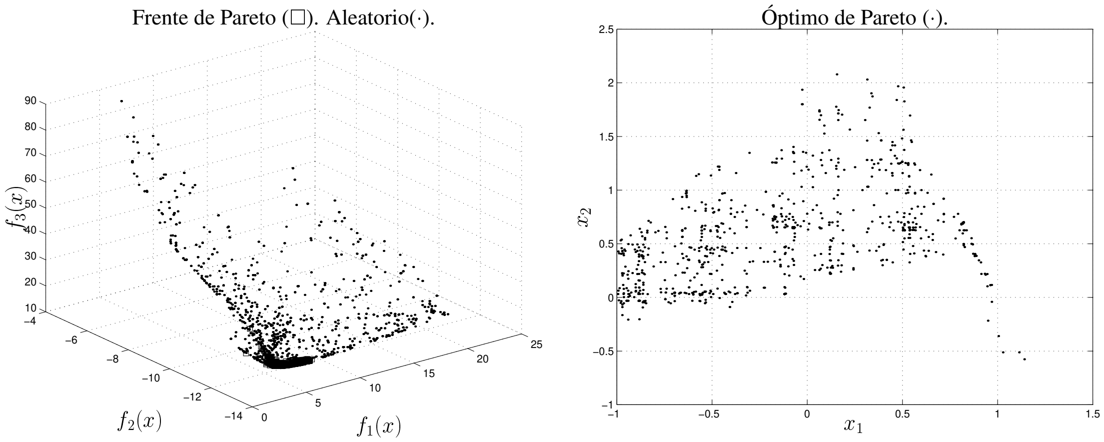


Figura 5.10: Función test de Viennet (3)

5.5 Evaluación del algoritmo MRCD con sistemas de ensayo.

El algoritmo MRCD ha sido creado para el diseño multiobjetivo. En la sección 5.4 se ha usado para la resolución de problemas multiobjetivo puros, y en ésta se usará para analizar sus resultados ante problemas de diseño multiobjetivo mixto, $\mathcal{H}_2/\mathcal{H}_\infty$. Para poder comparar la bondad de los resultados obtenidos, se va a hallar una cota superior de los diferentes frentes de Pareto por medio de algoritmos que usan las técnicas de ecuaciones lineales matriciales (LMIs), ver apéndice D.

Con el fin de que el estudio sea lo más significativo posible se han seleccionado trece sistemas de ensayo, propuestos en [APH98] para la sintonización de PIs por métodos de optimización convexa y que volverán a ser usadas en el capítulo 6. Son sistemas SISO (una entrada, una salida), cuyas ecuaciones de transferencia están en la tabla 5.3. A continuación se detallan las características de dichos sistemas y los problemas que puede presentar su control.

Sistemas $G_1(s) - G_6(s)$. Los sistemas $G_1(s)$ y $G_2(s)$ son relativamente fáciles de controlar. El sistema $G_3(s)$ tiene un gran tiempo de retraso, y el modelo $G_4(s)$ es un modelo integral. El sistema $G_5(s)$ es de fase no mínima con un cero en la parte izquierda del plano s, y el sistema $G_6(s)$ tiene dos polos complejos conjugados con un amortiguamiento de $\zeta = 0.33$. Todos estos sistemas excepto el sistema $G_6(s)$ tienen un diagrama de Bode decreciente de forma monótona.

Sistemas $G_7(s) - G_8(s)$. Estos sistemas tienen un tiempo relativamente grande de retardo relativo, como la función $G_3(s)$, siendo $w_{90} = \pi/2$ y $w_{180} = \pi$. Ambos sistemas tienen un diagrama de Bode decreciente de forma monótona.

Sistemas $G_{10} - G_{11}$. En estos sistemas hay unas constantes de tiempo lentas, que condicionan el comportamiento temporal del sistema, y otras rápidas, que son fundamentales en el comportamiento del sistema en lazo cerrado. El primero tiene dos constantes de tiempo rápidas, $0.1s$ y $1s$ y una tercera lenta $20s$. El segundo reemplaza la constante de tiempo más lenta.

Sistemas G_{12} Este sistema es oscilatorio como el $G_6(s)$, y su amortiguamiento es de $\zeta = a/6$. El disminuir a se hace más inestable llegando a un valor límite en $a = 0$.

$G_1(s) = \frac{1}{(s+1)^3}$
$G_2(s) = \frac{1}{(s+1)(1+0.2s)(1+0.04s)(1+0.008s)}$
$G_3(s) = \frac{e^{-15s}}{(s+1)^3}$
$G_4(s) = \frac{1}{s(s+1)^2}$
$G_5(s) = \frac{1-2s}{(s+1)^3}$
$G_6(s) = \frac{9}{(s+1)(s^2+2s+9)}$
$G_7(s) = e^{-s}$
$G_8(s) = \frac{e^{-s}}{s}$
$G_{10} = \frac{100}{(s+10)^2}\left(\frac{1}{s+1} + \frac{0.5}{s+0.05}\right)$
$G_{11}(s) = \frac{150}{(s+10)^2(s+1)}$
$G_{12}(s) = \frac{9}{(s+1)(s^2+as+9)}$
$G_{13}(s) = \frac{(s+6)^2}{s(s+1)^2(s+36)}$
$G_{14}(s) = \frac{a}{(s+a)(s-1)}$

Tabla 5.3: Funciones de transferencia de los sistemas de ensayo.

Sistema $G_{13}(s)$. Este sistema es condicionalmente estable ya que la fase no es monótonamente decreciente con lo que la curva de Nyquist cruza el eje real negativo en dos puntos $s = 0.0191$ y $s = -0.1656$, siendo $w_{90} = 0$ y $w_{180} = 1.69$.

Sistema $G_{14}(s)$. Es un sistema inestable dependiendo del valor de a. La fase no decrece de forma monótona siendo de valor π para altas y bajas frecuencias. La fase mínima se produce a la frecuencia $w = \sqrt{a}$, siendo su valor $\pi - \arctan \frac{(a-1)\sqrt{a}}{2a}$.

5.5.1 Planteamiento del problema.

A partir de estas funciones se va a plantear dos problemas para probar la bondad del algoritmo MRCD frente al diseño multiobjetivo. Ambos consisten en el diseño de controladores mixto $\mathcal{H}_2/\mathcal{H}_\infty$, ver sección 2.3.1. En el primer problema se proponen controladores en realimentación de estado y en el segundo controladores con realimentación a la salida. En ambos casos se comparan los resultados con el límite superior del mismo problema obtenido por algoritmos que usan inecuaciones lineales matriciales (LMI), ver apéndice D. El uso de LMIs para la solución del límite superior de problemas mixtos es, sin duda, uno de los métodos más usados en los últimos años como queda de manifiesto en la sección 2.3.1.

Se propone un sistema definido en ecuaciones de estado donde la matriz A del mismo tenga una incertidumbre multiplicativa. Eliminando dicha incertidumbre y respetando las conexiones planta e incertidumbre, el problema se convierte en el típico primer planteamiento de diseño multiobjetivo (2.5). En la figura 5.11 se dibuja el sistema propuesto con un controlador en realimentación de estado (bloque izquierdo) y con un controlador a la salida (bloque derecho).

Los objetivos deseados en el diseño por cualquiera de las dos realimentaciones vienen dados por

$$\min_{K} \begin{bmatrix} \|T_{yw}(s)\|_2 \\ \|T_{ew}(s)\|_\infty \end{bmatrix}.$$

La minimización de la norma $\mathcal{H}_2$ reduce el efecto en la salida del sistema de un posible ruido de la entrada incertidumbre, y la minimización de la norma $\mathcal{H}_\infty$ da robustez al mismo, mediante la aplicación del teorema de pequeña ganancia, ver [ZDG96]. El problema queda simplificado al tener las dos funciones de transferencia con las que se trabaja la misma entrada w.

La elección de este problema es puramente académica con el fin de comparar los resultados obtenidos por el algoritmo MRCD y el límite superior hallado por las técnicas LMI. De hecho, una de las ventajas que tiene el diseño de controladores por los algoritmos MRCD es la posibilidad de fijar límites a sus parámetros, restricción que en este caso no se aplica. En este problema se supone que los límites no existen para ver la evolución del algoritmo. En la tabla 5.2 se ha desarrollado las funciones test de la tabla 5.1 en la representación de estado que se utilizará para el problema propuesto. Los sistemas con retardo han sido sustituidos por su aproximación de Padè de primer orden, [Fri96].

El algoritmo MRCD convierte las restricciones del problema en nuevos objetivos, ver sección 4.3. En este caso la única restricción es conseguir controladores que estabilicen la planta en lazo cerrado. El problema propuesto para maximizar con el algoritmo MRCD es el siguiente,

$$\max_{K} \begin{bmatrix} \left\{ \begin{array}{l} \text{SI } \max(\text{Re}(\lambda(T(s)))) < 0 \to 0, \\ \text{SINO } -\max(\text{Re}(\lambda(T(s)))). \end{array} \right\} \\ -\|T_{yw}(s)\|_2 \\ -\|T_{ew}(s)\|_\infty \end{bmatrix} \tag{5.2}$$

donde $\text{Re}(\lambda(T(s)))$ devuelve el valor real de los polos del sistema en lazo cerrado. Con este

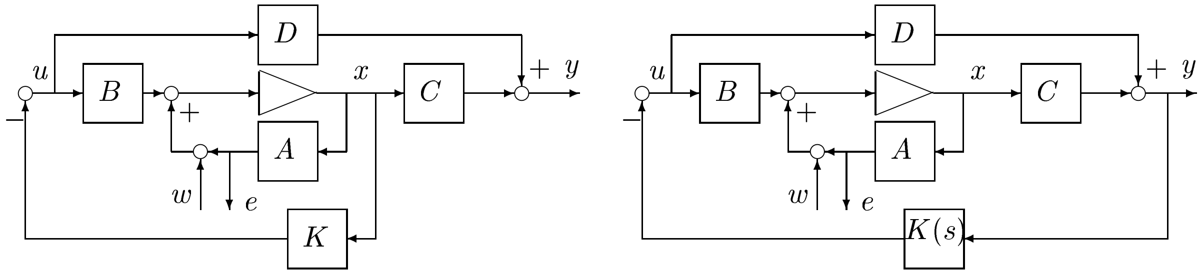


Figura 5.11: Controladores de estado y de salida.

objetivo se consigue distinguir entre controladores que inestabilicen la planta en lazo cerrado en función de lo grandes que sean los polos inestables que genere. Este hecho da una gran libertad al elegir los límites del espacio inicial, ya que no se precisa que la población inicial este constituida por controladores que estabilicen la planta.

5.5.2 Estructura de los controladores.

Los GAs son algoritmos que optimizan una serie de parámetros, por lo que es necesario fijar una estructura a priori del controlador para poder plantear el problema, ver sección 4.2. Esta estructura va a ser decisiva a la hora de elegir los límites del espacio de búsqueda de los parámetros, que es uno de los problemas más importantes del diseño de controladores con GAs. Varios de los operadores del algoritmo MRCD han sido diseñados para conseguir combatir este grave problema, éstos son: "Algoritmo paralelo en isla", modificación dinámica de los límites del espacio de búsqueda, formulación de restricciones como objetivos, ver sección 4.2. Será en el diseño mixto propuesto donde se intentará analizar su eficacia.

Tipos de controladores. Los controladores para la resolución del problema mixto propuesto han sido divididos en dos grandes grupos, controladores con realimentación de estado y controladores con realimentación a la salida. Los primeros son controladores estáticos, existiendo una única parametrización posible. En cada sistema propuesto el controlador estático tiene un número de parámetros igual al número de variables de estado del sistema. Los controladores con realimentación a la salida han de ser dinámicos, y el orden elegido es el mismo que el orden de la planta a la que deben controlar. Se ha tomado este orden por ser el que necesitan las técnicas basadas en LMIs para poder hallar el límite superior del problema mixto. Así, los controladores buscados por el algoritmo MRCD y los algoritmos basados en LMIs tienen el mismo orden.

$G_1(s) = \left[\begin{array}{ccc\|c} -3 & -3 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \end{array}\right]$
$G_2(s) = 1.0e+5 * \left[\begin{array}{cccc\|c} -0.0038 & -0.3351 & -1.8938 & -1.5625 & 1e-5 \\ 1e-1 & 0 & 0 & 0 & 0 \\ 0 & 1e-5 & 0 & 0 & 0 \\ 0 & 0 & 1e-5 & 0 & 0 \\ \hline 0 & 0 & 0 & 1e-5 & 0 \end{array}\right]$
$G_3(s) = \left[\begin{array}{cccc\|c} -3.1333 & -3.4000 & -1.4000 & -0.1333 & 1.0000 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \\ \hline 0 & 0 & -1.0000 & 0.1333 & 0 \end{array}\right]$
$G_4(s) = \left[\begin{array}{ccc\|c} -2 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \end{array}\right]$
$G_5(s) = \left[\begin{array}{ccc\|c} -3 & -3 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & -2 & 1 & 0 \end{array}\right]$
$G_6(s) = \left[\begin{array}{ccc\|c} -3 & -11 & -9 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 9 & 0 \end{array}\right]$
$G_7(s) = \left[\begin{array}{c\|c} -2 & 1 \\ \hline 4 & -1 \end{array}\right]$
$G_8(s) = \left[\begin{array}{cc\|c} -2 & 0 & 1 \\ 1 & 0 & 0 \\ \hline -1 & 2 & 0 \end{array}\right]$
$G_{10} = \left[\begin{array}{cccc\|c} -21.0500 & -121.0500 & -106.0000 & -5.0000 & 1.0000 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \\ \hline 0 & 0 & 150.0000 & 55.0000 & 0 \end{array}\right]$
$G_{11}(s) = \left[\begin{array}{ccc\|c} -21 & -120 & -100 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 150 & 0 \end{array}\right]$
$G_{12}(s) = \left[\begin{array}{ccc\|c} -1.2000 & -9.2000 & -9.0000 & 1.0000 \\ 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ \hline 0 & 0 & 9.0000 & 0 \end{array}\right]$
$G_{13}(s) = \left[\begin{array}{cccc\|c} -38 & -73 & -36 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 12 & 36 & 0 \end{array}\right]$
$G_{14}(s) = \left[\begin{array}{cc\|c} 0 & 1 & 1 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \end{array}\right]$

Tabla 5.4: Ecuaciones de estado de los problemas modelo

Formas paramétricas. Un punto de gran importancia en la descripción de los controladores dinámicos es la forma paramétrica empleada para su definición. Se han empleado tres formas canónicas diferentes para resolver los mismos problemas con el fin de estudiar las diferencias en su convergencia. Las estructuras elegidas son la forma canónica controlable, la forma canónica diagonal y una forma canónica balanceada, ver apéndice C. En la forma canónica diagonal es preciso establecer ciertas restricciones sobre la estructura del sistema, específicamente, si los polos del controlador son reales o complejos conjugados, ver sección 4.2. Tal y como se muestra en el apéndice C la forma canónica controlable de un sistema SISO tiene como parámetros a_j, c_j y d. La forma canónica diagonal tiene como parámetros los polos p_j, y los parámetros s_j, b_j y d. Los polos podrán ser reales o complejos conjugados en función de los parámetros s_j de la siguiente forma

$$\begin{aligned} &\text{SI } s_j < 0.5 \rightarrow [p_{2\cdot j}, p_{2\cdot j+1}] \\ &\text{SINO } [p_{2\cdot j} \pm i \cdot p_{2\cdot j+1}]. \end{aligned} \tag{5.3}$$

La forma balanceada tiene los parámetros σ_j (valores singulares de Hankel, ver [ZDG96]), c_j y d. En todos los casos $j = 1 \dots n$, donde n es el orden del controlador.

Límites iniciales del espacio de búsqueda. Una vez fijada la estructura es preciso determinar los límites iniciales del espacio de búsqueda. El problema propuesto va a requerir unos límites muy grandes, lo que va a ser útil para analizar la eficacia de los operadores del algoritmo MRCD. En el caso de controladores estáticos para realimentación de estado se han usado los límites iniciales fijados en la tabla 5.5. En la tabla 5.6 contiene los límites del espacio de búsqueda de los respectivos controladores dinámicos siguiendo la nomenclatura antes mencionada, ver apéndice C. En ambas tablas se muestra como las cinco diferentes subpoblaciones (Sub_i) parten con límites muy diferentes. El hecho de que la población inicial no sea estable no es problema por la transformación de esa restricción en un objetivo a optimizar, ver ecuación 5.2.

Parámetros del algoritmo MRCD. El algoritmo MRCD parte en todos los casos de una población de 100 individuos divididos en 5 subpoblaciones. La probabilidad de cruce es de $p_{cruce} = 0.7$ y la de mutación de $p_{muta} = 0.1$, siendo la constante $t_{dom} = 0.5$, ver pseudocódigo 3.10.

	Sub_1	Sub_2	Sub_3	Sub_4	Sub_5
$[k_i^{min}, k_i^{max}]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e4, 1e4]$	$[-1e5, 1e5]$	$[-1e6, 1e6]$

Tabla 5.5: Limites de espacio de búsqueda inicial de los parámetros de controlador estático.

	Sub_1	Sub_2	Sub_3	Sub_4	Sub_5
$[a_i^{min}, a_i^{max}]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e4, 1e4]$	$[-1e5, 1e5]$	$[-1e6, 1e6]$
$[b_i^{min}, b_i^{max}]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e4, 1e4]$	$[-1e5, 1e5]$	$[-1e6, 1e6]$
$[c_i^{min}, c_i^{max}]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e4, 1e4]$	$[-1e5, 1e5]$	$[-1e6, 1e6]$
$[d^{min}, d^{max}]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e4, 1e4]$	$[-1e5, 1e5]$	$[-1e6, 1e6]$
$[p_i^{min}, p_i^{max}]$	$[-1e2, 1e2]$	$[-1e2, 1e2]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e3, 1e3]$
$[\sigma_i^{min}, \sigma_i^{max}]$	$[-1e2, 1e2]$	$[-1e2, 1e2]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e3, 1e3]$
$[s_i^{min}, s_i^{max}]$	$[0, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$

Tabla 5.6: Límites del espacio de búsqueda inicial para los controladores dinámicos.

5.5.3 Evaluación de los resultados.

Las figuras 5.12-5.24 muestran los frentes de Pareto para la realimentación de estado, figuras de las izquierda, y para la realimentación a la salida, figuras de la derecha, obtenidas con el algoritmo MRCD y con dos funciones de la librería “LMI Control Toolbox” de Matlab, ver apéndice D. En las figuras del la izquierda los puntos $(\cdot)$ representan el frente de Pareto hallado por los algoritmos MRCD y la línea continua el límite superior del frente de Pareto hallado por la función `msfsyn`$(\cdot)$. En las figuras de la derecha los $(*)$ representan el frente de Pareto hallado por el algoritmo MRCD con el controlador en su forma controlable, los $(+)$ el frente de Pareto para un controlador en su forma diagonal, los $(\circ)$ el frente de Pareto para un controlador en su forma balanceada y la línea continua el límite superior de frente de Pareto hallado por la función `hinfmix`$(\cdot)$.

La tabla 5.10 muestra los valores de la optimización simple $\mathcal{H}_\infty$ y la $\mathcal{H}_2$ con el valor de los parámetros de los controladores en realimentación de estado que lo consiguen. Estos datos dan información sobre la posición de la curva de frente de Pareto, y sobre la escala de valores que se deberán manejar. Hay un gran desequilibrio en el valor de los parámetros lo que va a generar sistemas mal condicionados, ver [TB97], con los que va a ser difícil trabajar. Las tablas 5.12-5.18 muestran los límites inferior y superior de los diferentes parámetros de individuos pertenecientes al frente de Pareto de la última generación.

Realimentación de estado.

Con esta realimentación, todos los controladores estáticos obtenidos por el algoritmo MRCD mejoran las respectivas cotas superiores del frente de Pareto hallados por la función $\mathtt{msfsyn}(\cdot)$, ver apéndice D. En la parte central de frente de Pareto esta diferencia es mayor, minimizándose al tender a los extremos.

Los límites inferior y superior de los respectivos parámetros k_i de los controladores asociados al frente de Pareto de la última generación están representados en la tabla 5.12. En ella se ve que los valores son muy altos al igual que los son los controladores óptimos de la norma $\mathcal{H}_2$ y $\mathcal{H}_\infty$ mostrados en las tablas 5.10 y 5.11.

En el sistema $G_2(s)$ la función $\mathtt{msfsyn}(\cdot)$ no pudo encontrar ningún elemento del frente de Pareto, probablemente por tener valores muy altos mientras en el algoritmo MRCD sí pudo. En el sistema $G_8(s)$ el algoritmo MRCD encontró un único punto óptimo, que obtienen los valores mínimos de las normas $\mathcal{H}_2$ y $\mathcal{H}_\infty$ calculadas independientemente y que pueden verse en las tablas 5.10 y 5.11. El punto solución se muestra en la tabla siguiente.

k_1	k_2	$\|T_{yw}\|_2$	$\|T_{ew}\|_\infty$
$4.842e8$	$9.684e8$	2	2.2361

Esta situación se produce porque los índices elegidos no son competitivos. Es un ejemplo de la potencia del algoritmo MRCD al conseguir converger a un solo punto desde un espacio de búsqueda inicial tan grande. En el sistema $G_{10}(s)$ el frente de Pareto y su límite superior pasan de ser cóncavo a convexo en el mismo punto, aunque siempre la curva hallada por el algoritmo MRCD es menor.

La diversidad de controladores a lo largo del frente de Pareto es muy buena en todos los casos, consiguiendo curvas de densidad uniforme menos en los extremos de las mismas. El hecho de filtrar a la élite con respecto a los parámetros del controlador hace que el frente de Pareto tenga una gran variedad de controladores. En la tabla 5.7 se muestran controladores de varios sistemas donde un pequeña modificación en los parámetros k_i motiva un pequeño cambio en las normas $[\mathcal{H}_2, \mathcal{H}_\infty]$ y de la posición de los polos del sistema en lazo cerrado $T(s)$. Ello prueba que estos controladores no son frágiles, ver [KB97], es decir que una leve modificación

	$[k_1, \ldots, k_n]$	$polos(T(s))$	$[\mathcal{H}_2, \mathcal{H}_\infty]$
$G_3(s)$	$[2.66e7, 2.06e9, 1.67e9]$	$[-0.186, -0.633, -76.4, -2.66e7]$	$[0.998, 4.25]$
$G_3(s)$	$[8.61e7, 1.54e9, 1.03e9]$	$[-0.221, -0.467, -17.2, -8.61e7]$	$[1.1, 3.9]$
$G_4(s)$	$[1.13e6, 4.14e9, 8.32e9]$	$[-2.01, -3.67e3, -1.13e6]$	$[0.499, 5.02]$
$G_4(s)$	$[3.17e6, 5.39e9, 7.49e9]$	$[-1.39, -1.7e3, -3.17e6]$	$[0.6, 3.83]$
$G_6(s)$	$[8.26e4, 4.95e8, 5.04e9]$	$[-10.2, -6.49e3, -7.61e4]$	$[2, 32.4]$
$G_6(s)$	$[4.4e4, 4.92e8, 4.54e9]$	$[-9.24, -2.2e4 \pm 2.73e3i]$	$[2.1, 29.4]$

Tabla 5.7: Ejemplos de controladores estáticos cercanos en $[\mathcal{H}_2, \mathcal{H}_\infty]$.

en sus parámetros no provoca un cambio sustancial en sus prestaciones. El procedimiento de convergencia del algoritmo MRCD, que dirige una población de controladores hacia el frente de Pareto solución garantiza que los controladores obtenidos no van a ser frágiles. Un controlador frágil es un punto aislado en dentro del espacio de los vectores objetivo, y por tanto muy difícil de localizar por un GA.

Realimentación a la salida.

El problema de realimentación a la salida ha sido realizado por cuatro diferentes procesos, y sus resultados son mostrados en la parte izquierda de las figuras 5.12-5.24. Dichos procesos son, la optimización por el algoritmo MRCD de los parámetros de un controlador dinámico en su forma canónica controlable ($*$), los de un controlador en su forma diagonal ($+$) y los de un controladores en su forma balanceada ($\circ$); así como los controladores encontrados por la función `hinfmix`$(\cdot)$, ver apéndice D.

Los tres problemas planteados por el algoritmo MRCD constituye tres problemas diferentes con un frente de Pareto común. El espacio de búsqueda no será equivalente y las diferentes parametrizaciones actúan de forma diferente en los operadores cruce y mutación, lo que modifica su posible convergencia. A la vista de los resultados, los frentes de Pareto conseguido con la forma controlable mejoran los conseguidos con las otras parametrizaciones. Se pueden formular diferentes hipótesis sobre las posibles causas:

1. La forma controlable es más abierta para la representación de todo tipo de controladores. La forma diagonal tiene el impedimento de los parámetros s_i, ver (5.3, para determinar

su estructura, y la forma balanceada sólo puede describir controladores estables.

2. La forma controlable tiene más posibilidades de explorar el espacio de búsqueda ya que todos los parámetros están relacionados, modificándose mucho las propiedades del controlador con la variación de cualquiera de ellos.

3. La forma controlable tiene menos “ruido colateral”, ver sección 5.2. En la forma diagonal o balanceada es más fácil encontrar controladores subóptimos donde varios de sus parámetros no intervengan, por ser polos muy rápidos. El crecimiento en la población de estos controladores dará lugar a optimizar sólo parte de los parámetros, de forma similar a la función “Royal Road”.

4. Los límites del espacio de búsqueda inicial no son estrictamente equivalentes, pudiendo conseguir más diversidad en la población inicial con los controladores representados en su forma canónica controlable, cuyos polos dependen de la relación entre el valor de sus parámetros.

Las tablas 5.13-5.18 muestran los límites inferior y superior de los parámetros de los controladores dinámicos asociados a los frentes de Pareto de la última generación. Sus valores son muy altos, lo que es una prueba de la eficacia del movimiento dinámico de los límites del espacio de búsqueda.

Los frentes de Pareto obtenidos con el algoritmo MRCD en su formulación controlable han minimizado las cotas superiores obtenida con la función `hinfmix`$(\cdot)$ en todos los casos, excepto en algunas zonas con grandes valores de la norma $\mathcal{H}_\infty$. Todos los frentes de Pareto tienen la forma de los frentes obtenidos en la realimentación de estado. Por ejemplo el sistema $G_{10}(s)$ sigue teniendo dos partes, una cóncava y convexa. Los sistemas $G_3(s)$ y $G_5(s)$ tienen un límite superior muy alto y en el sistema $G_2(s)$, la función `hinfmix`$(\cdot)$ no puede encontrar dicho límite. En las funciones $G_2(s)$, $G_3(s)$, $G_{10}(s)$ y $G_{13}(s)$ los frentes con los tres tipos de controladores son parecidos. Los frentes de $G_7(s)$ son iguales por ser el controlador de primer orden y por tanto no influir el tipo de parametrización.

La tabla 5.8 muestra los polos del controlador y del sistema en lazo cerrado para controladores con distinta parametrización y con la misma posición en el frente de Pareto asociado. Dichos controladores tienen distintos polos pero parecida función de transferencia en lazo cerrado y los mismos objetivos. De esta tabla se deduce que los distintos problemas no solo tienen infinitas soluciones a lo largo del frente de Pareto, sino que para un punto de dicho frente de Pareto hay múltiples soluciones.

La tabla 5.9 muestra los mismos datos que la tabla 5.8 para el caso de controladores cuyos puntos el vector de funciones están más o menos próximo al verdadero frente de Pareto. Los controladores balanceados tienen polos muy rápidos, avalando la hipótesis de que en ese caso existe "ruido colateral". De hecho al incrementar el espacio inicial de búsqueda de los parámetros p_i y σ_i, ver tabla 5.6, los frentes de Pareto obtenidos por los controladores diagonal y balanceado son sensiblemente peores, por su convergencia a polos rápidos. Sin embargo, los controladores en su forma diagonal mostrados en esta tabla no tienen polos más rápidos que los controladores en su forma controlable.

Conclusión.

Las conclusiones a las que se ha llegado en la realización de estos dos problemas de control son las siguientes:

- Los problemas de diseño de controladores en realimentación tienen múltiples soluciones para un mismo valor en el vector objetivo.
- La repetición de un problema con el algoritmo MRCD proporciona el mismo frente de Pareto si no se varía ninguno de sus parámetros.
- La utilización de diferentes estructuras en el controlador dinámico influye en la posición del frente de Pareto obtenido y en las características de sus correspondientes controladores.
- La modificación de los límites del espacio de búsqueda influye en el resultado final de forma diferente en las distintas parametrizaciones, pudiendo producir "ruido colateral".
- A la vista de los resultados, la forma canónica controlable tiene menos sensibilidad a los cambios en los límites del espacio de búsqueda y tiene más capacidad para explorar dicho espacio.
- En los dos problemas planteados, y con los trece sistemas de ensayo, el algoritmo MRCD minimiza el cota superior calculada con los algoritmos basados en LMIs.

	$polos(K(s))$	$polos(T(s))$	$[\mathcal{H}_2, \mathcal{H}_\infty]$
$G_1(s)[cont.]$	$[-25.9, -96, -923]$	$[-0.449, -1.1, -2.08, -25, \ldots]$	$[1.79, 1.71]$
$G_1(s)[diag.]$	$[-1.44 \pm -4.68i, -6.04]$	$[-0.433, -1.43 \pm -0.633i, -1.3 \pm 4.66i, \ldots]$	$[1.79, 1.98]$
$G_1(s)[bal.]$	$[-32.2, -4.57e3, -2.17e6]$	$[-0.348, -1.31 \pm -1.05i, -32.2, \ldots]$	$[1.79, 2.23]$
$G_3(s)[cont.]$	$[-0.37, -0.736, -24.6, -2.26e3]$	$[-0.0592, -0.391, -0.306 \pm -0.274i, \ldots]$	$[2, 3.68]$
$G_3(s)[diag.]$	$[-11.2 \pm -17.8i, -6.65 \pm -80.8i]$	$[-0.075, -0.322, -1.35 \pm -1.38i, \ldots]$	$[2, 3.82]$
$G_3(s)[bal.]$	$[-12.1, -2.2e5, -1.09e8 \pm 2.16e8i]$	$[-0.0772, -0.313, -1.3 \pm -1.43i, \ldots]$	$[2, 3.82]$
$G_{11}(s)[cont.]$	$[-1.86, -49.9, -2.19e4]$	$[-0.875, -2.46, -9.41 \pm -5.02i, \ldots]$	$[99.9, 34]$
$G_{11}(s)[diag.]$	$[-34.6, -10.7 \pm 140i]$	$[-1.07, -9.28 \pm -6.14i, -35.9, \ldots]$	$[99.4, 35.3]$
$G_{11}(s)[bal.]$	$[-59.4, -1.84e5, -2.49e7]$	$[-1.1, -9.62 \pm -5.67i, -60.1, \ldots]$	$[99.7, 33.9]$

Tabla 5.8: Ejemplos de controladores dinámicos cercanos en $[\mathcal{H}_2, \mathcal{H}_\infty]$.

	$polos(K(s))$	$polos(T(s))$	$[\mathcal{H}_2, \mathcal{H}_\infty]$
$G_6(s)[cont.]$	$[-22.5 \pm -36.4i, -97.9]$	$[-2.3, -1.76 \pm -3.05i, -21.4 \pm 34.8i]$	$[4.35, 9.52]$
$G_6(s)[diag.]$	$[-4.75 \pm -7.94i, -19.7]$	$[-2.97, -1.29 \pm -2.99i, -3.49 \pm 7.46i]$	$[5.27, 9.96]$
$G_6(s)[bal.]$	$[-487, -4.04e7, -7.45e8]$	$[-0.873, -1.06 \pm -3.52i, -487]$	$[6.59, 9.81]$
$G_6(s)[cont.]$	$[-34.8, -15.9 \pm -89.8i]$	$[-1.61 \pm -3.72i, -17.4 \pm -7.31i]$	$[2.38, 38]$
$G_6(s)[cont.]$	$[-6.17, -1.8 \pm 14.2i]$	$[-2.6 \pm -2.22i, -2.54 \pm -4.06i]$	$[4.33, 39]$
$G_6(s)[cont.]$	$[-98.8, -3.51e7, -5.38e8]$	$[-0.686, -1.04 \pm -5.59i, -99.1]$	$[4.65, 39]$
$G_{11}(s)[cont.]$	$[-8.2, -31.8 \pm -338i]$	$[-3.51 \pm -5.58i, -11.1 \pm 12.8i]$	$[42.5, 199]$
$G_{11}(s)[diag.]$	$[-28.3, -10.2 \pm 134i]$	$[-4.52, -9.7 \pm -10.5i, -25.7]$	$[43.9, 198]$
$G_{11}(s)[bal.]$	$[-40.2, -2.7e5, -2.92e8]$	$[-3.38, -6.76 \pm -11.1i, -44.3]$	$[51.4, 194]$
$G_{14}(s)[cont.]$	$[-1.89e3 \pm -4.81e3i]$	$[-0.0739, -1.59e3, -1.1e3 \pm -4.7e3i]$	$[0.0071, 1]$
$G_{14}(s)[diag.]$	$[-284 \pm -185i, -284 \pm 185i]$	$[-18, -110, -220 \pm -469i]$	$[0.0463, 1.06]$
$G_{14}(s)[bal.]$	$[-61.8, -2.48e5]$	$[-0.684, -30.6 \pm 44.9i, -2.48e5]$	$[0.138, 1.09]$

Tabla 5.9: Ejemplos de controladores dinámicos cercanos en $\mathcal{H}_\infty$.

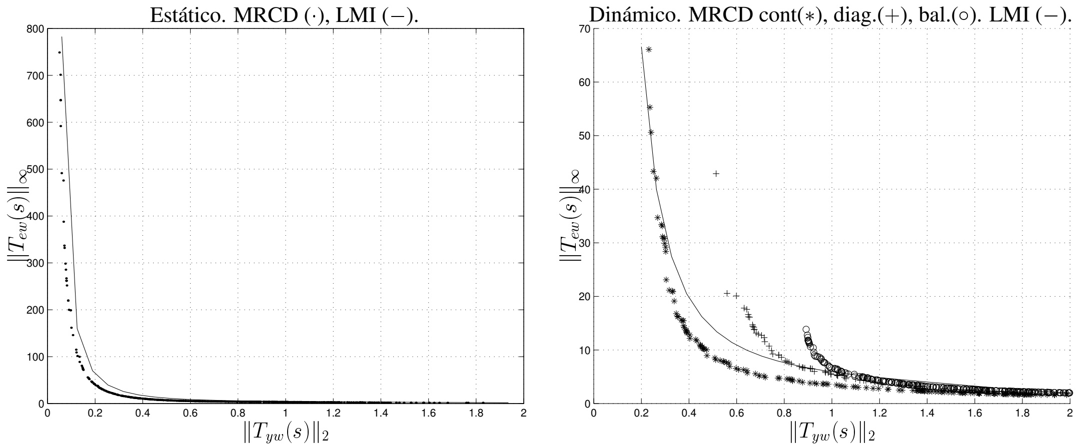


Figura 5.12: Realimentación de estado y de salida de $G_1(s)$

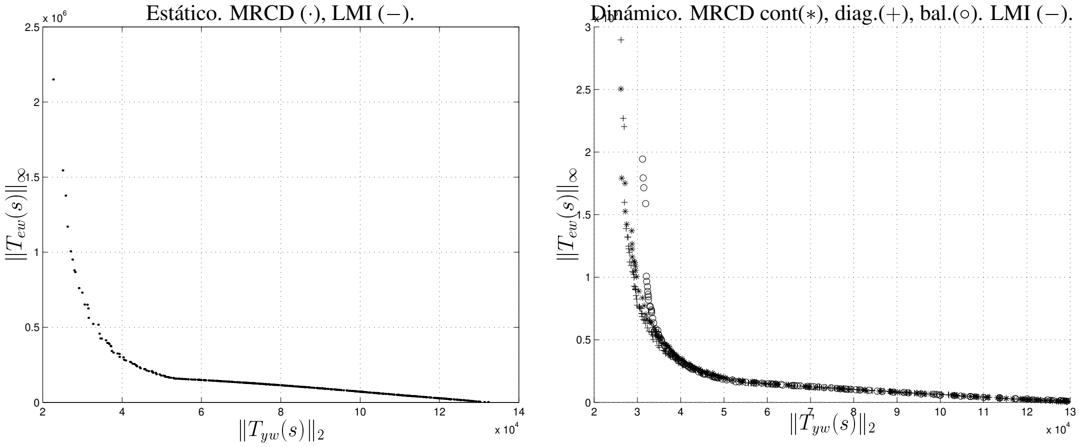


Figura 5.13: Realimentación de estado y de salida de $G_2(s)$

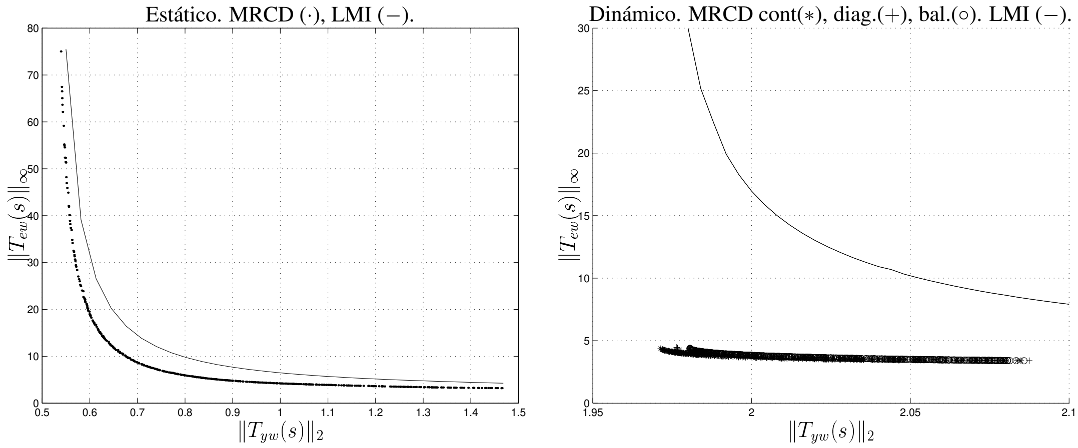


Figura 5.14: Realimentación de estado y de salida de $G_3(s)$

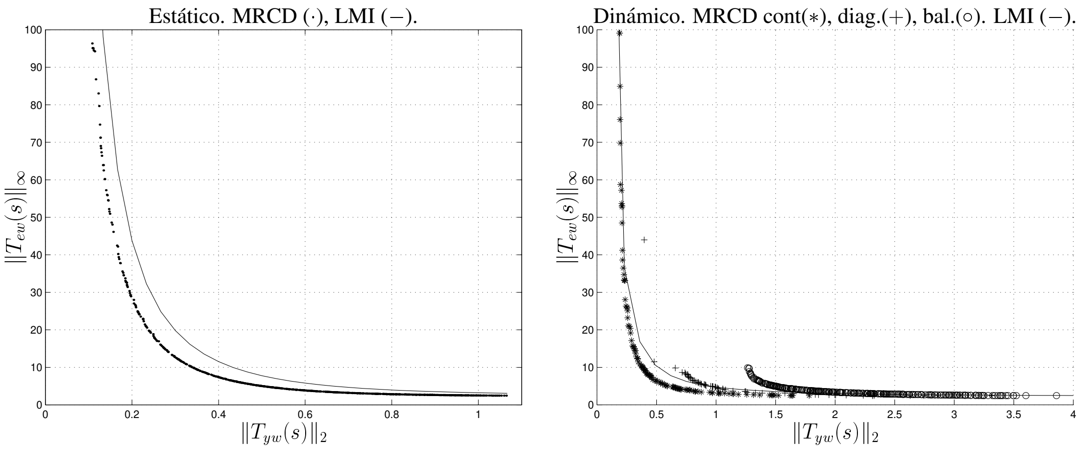


Figura 5.15: Realimentación de estado y a la salida de $G_4(s)$

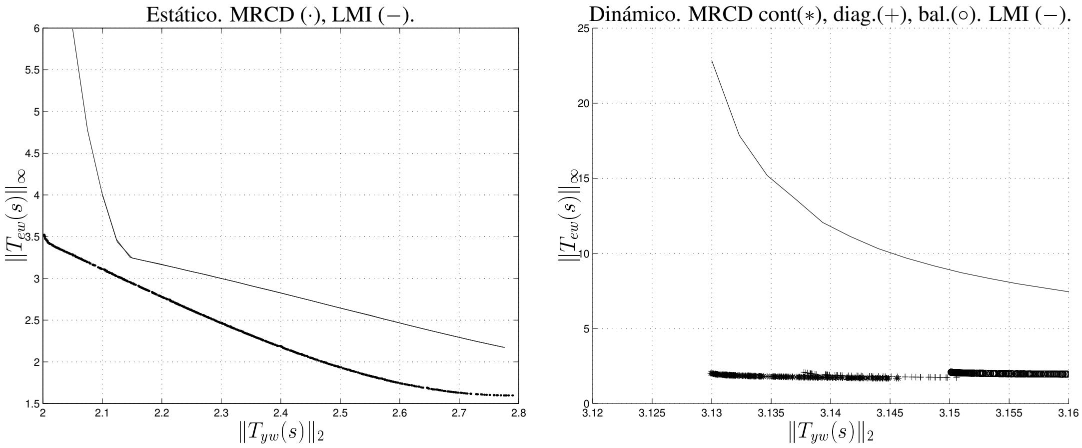


Figura 5.16: Realimentación de estado y de salida $G_5(s)$

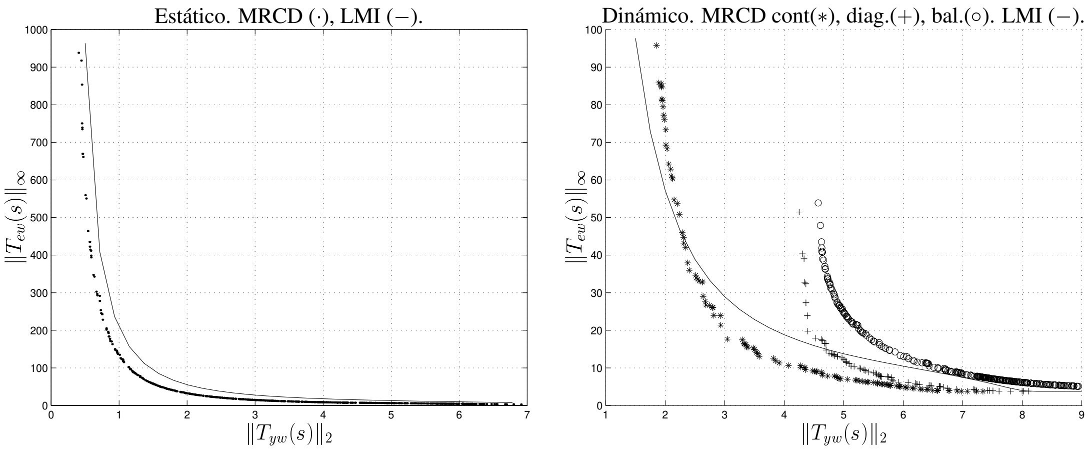


Figura 5.17: Realimentación de estado y de salida de $G_6(s)$

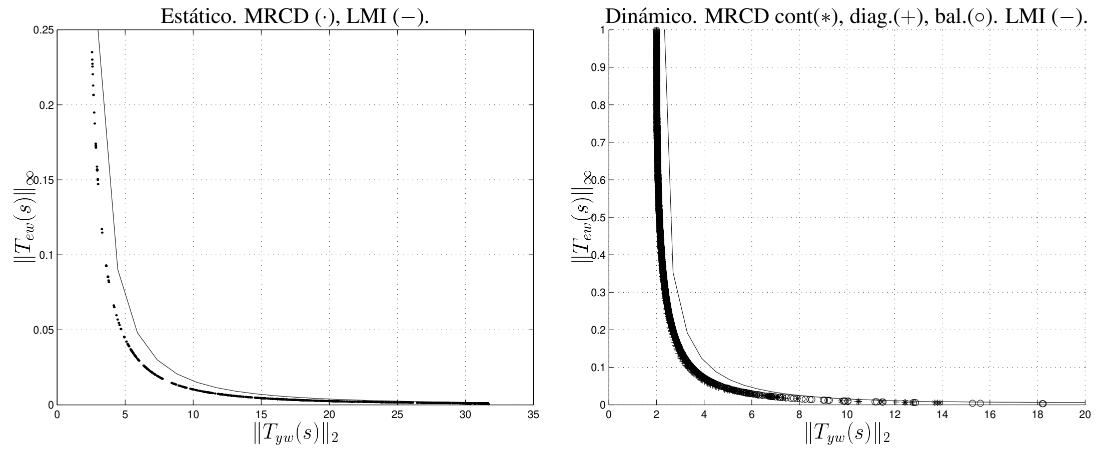


Figura 5.18: Realimentación de estado y de salida de $G_7(s)$

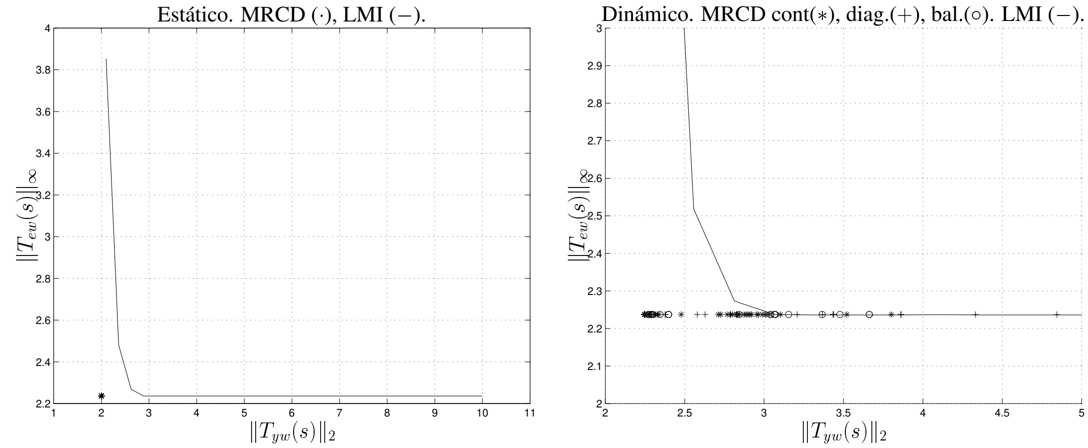


Figura 5.19: Realimentación de estado y de salida $G_8(s)$

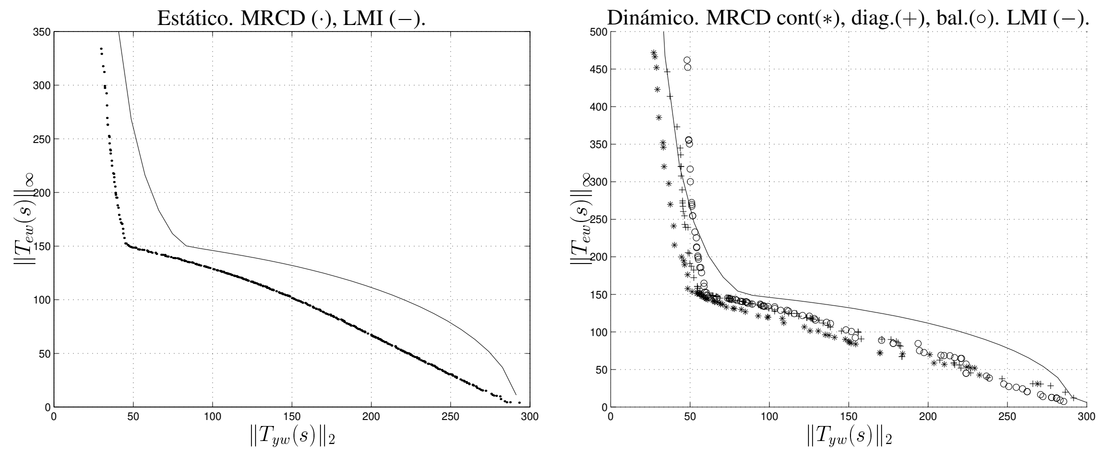


Figura 5.20: Realimentación de estado y de salida de $G_{10}(s)$

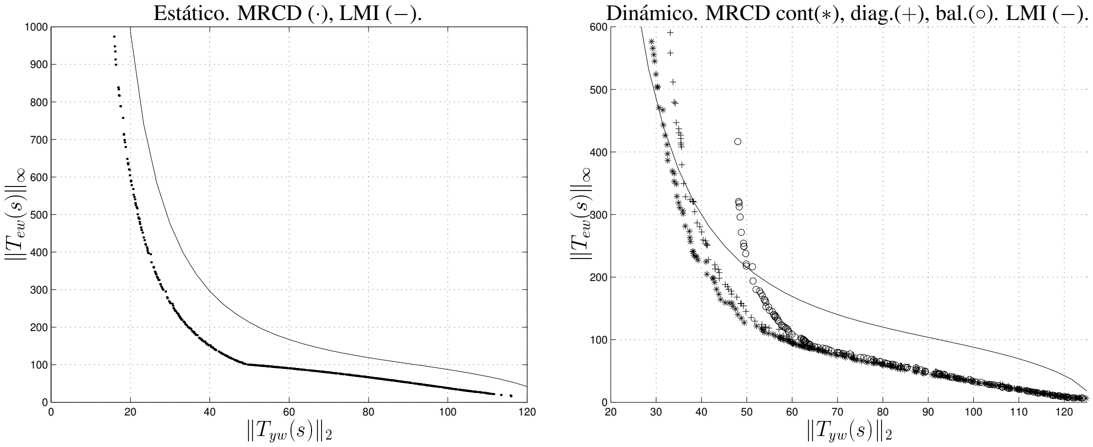


Figura 5.21: Realimentación de estado y de salida de $G_{11}(s)$

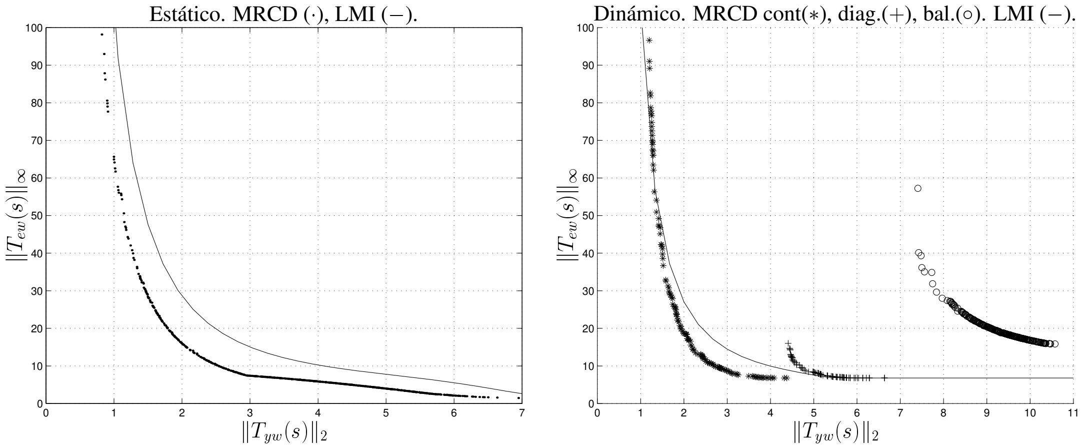


Figura 5.22: Realimentación de estado y de salida de $G_{12}(s)$

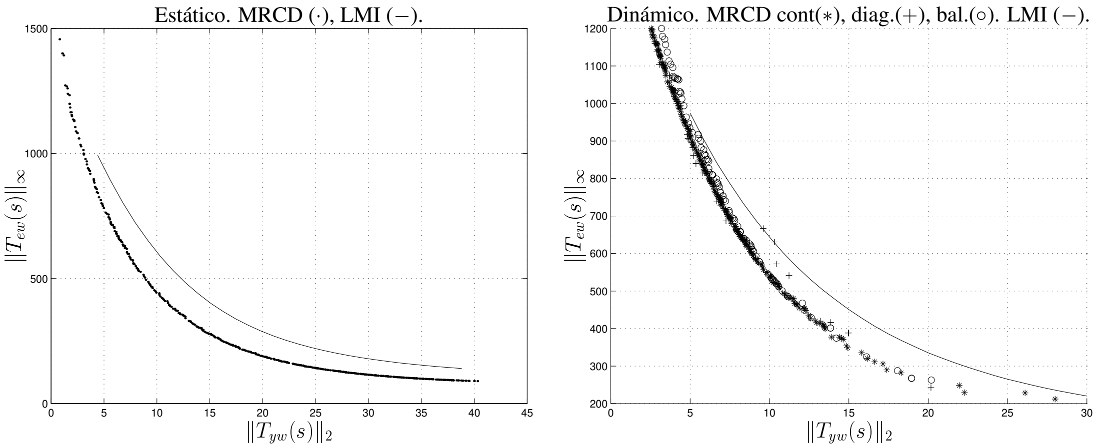


Figura 5.23: Realimentación de estado y de salida de $G_{13}(s)$

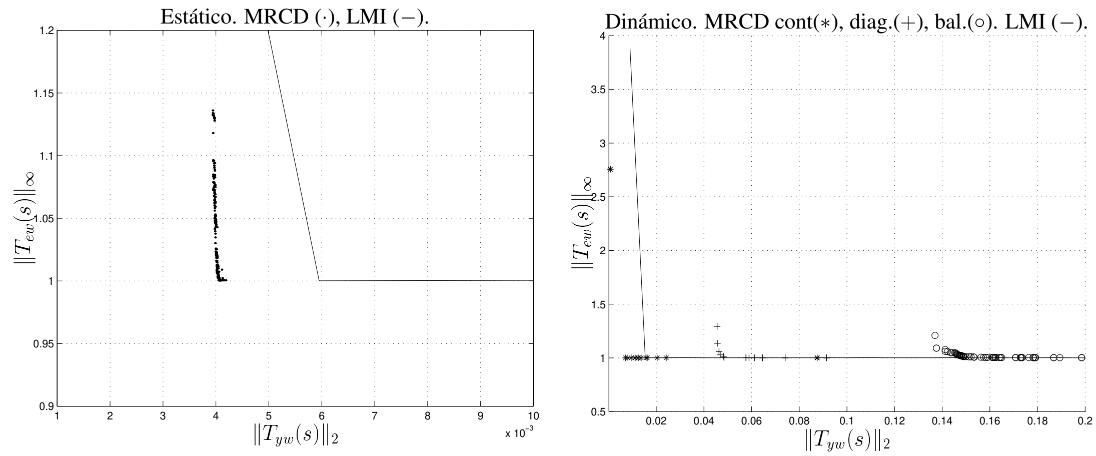


Figura 5.24: Realimentación de estado y de salida de $G_{14}(s)$

	$\min_{K_\infty} \mathcal{H}_\infty$	K_∞
$G_1(s)$	1.5968	$1.0e5 * [1.4830, 1.7029, 0.4981]$
$G_2(s)$	7.4918	$1.0e9 * [0.0115, 1.0069, 5.6912, 4.6957]$
$G_3(s)$	3.2227	$1.0e5 * [1.3356, 2.5150, 1.2783, 0.0786]$
$G_4(s)$	2.4495	$1.0e4 * [5.9228, 4.7497, 0.0000]$
$G_5(s)$	1.5968	$1.0e5 * [1.4830, 1.7029, 0.4981]$
$G_6(s)$	1.4674	$1.0e5 * [1.8935, 7.1721, 5.6899]$
$G_7(s)$	$5.5904e-4$	$3.1341e+6$
$G_8(s)$	2.2361	$1.0e4 * [3.1246, 0.0000]$
$G_{10}(s)$	1.6176	$1.0e6 * [0.9042, 5.2025, 4.5559, 0.2149]$
$G_{11}(s)$	1.42	$1.0e6 * [1.2255, 7.0057, 5.8358]$
$G_{12}(s)$	1.4592	$1.0e5 * [0.7997, 6.7540, 6.1389]$
$G_{13}(s)$	89.8332	$1.0e4 * [0.5555, 1.0673, 0.5264, 0.0000]$
$G_{14}(s)$	1	$1.0e6 * [0.0154, 2.5064]$

Tabla 5.10: Valores $\mathcal{H}_\infty$ óptimos con sus correspondientes controladores.

5.6 Orden óptimo para los controladores dinámicos.

Los controladores solución de los problemas de control mixto por métodos clásicos son de orden muy elevado o incluso infinito, ver sección 2.3.1. Ello implica que se debe reducir el orden los controladores obtenidos por estos métodos, ver [ZDG96], para poderse implantar físicamente. En el caso de la resolución de problemas mixtos con el algoritmo MRCD, el orden del controlador que se desea como solución deber ser fijado a priori. En esta sección de tratará de mostrar que los algoritmos MRCD son buenos para la consecución de controladores de orden relativamente bajo, con muy buenas propiedades de resiliencia (no fragilidad), y que aumentar el orden del controlador no necesariamente implica obtener mejores resultados.

Para intentar demostrar esta relación se va a usar un problema planteado en la tesis de Mario Rotea [Rot90], donde se analiza un problema mixto $\mathcal{H}_2/\mathcal{H}_2$ entre diferentes entrada y salidas de un sistema MIMO, ver figura 5.25. Se pretende obtener el conjunto de controladores dinámicos $K(s)$ que minimice el siguiente vector multiobjetivo

$$\min_{K(s)} \begin{bmatrix} \|T_{z_1w_1}(s)\|_2 \\ \|T_{z_2w_2}(s)\|_2 \end{bmatrix}.$$

	$\min_{K_2} \mathcal{H}_2$	K_2
$G_1(s)$	0.019	$1.0e+14*[0.0000, 0.0023, 4.3400]$
$G_2(s)$	$1.8110e4$	$1.0e12*[0.0000, 0.0005, 0.0657, 4.0801]$
$G_3(s)$	0.5179	$1.0e10*[0.0000, 0.0048, 6.1606, 0.8205]$
$G_4(s)$	0.0198	$1.0e14*[0.0000, 0.0006, 1.2185]$
$G_5(s)$	2.0044	$1.0e10*[0.0000, 4.4826, 2.2366]$
$G_6(s)$	0.1784	$1.0e15*[0.0000, 0.0010, 1.9356]$
$G_7(s)$	2.0001	$3.7622e-4$
$G_8(s)$	2.0000	$1.0e17*[4.7294, 9.4588]$
$G_{10}(s)$	3.2247	$1.0e15*[0.0000, 0.0035, 6.4728, 2.3729]$
$G_{11}(s)$	2.9728	$1.0e13*[-0.0000, -0.0008, -1.5570]$
$G_{12}(s)$	0.1784	$1.0e15*[0.0000, 0.0007, 1.2463]$
$G_{13}(s)$	0.0075	$1.0e18*[0.0000, 0.2488, 2.9854, 8.9562]$
$G_{14}(s)$	$1.3576e-4$	$1.0e17*[0.0000, 1.4409]$

Tabla 5.11: Valores $\mathcal{H}_2$ óptimos con sus correspondientes controladores.

	$[k_1^{min}, k_1^{max}]$	$[k_2^{min}, k_2^{max}]$	$[k_3^{min}, k_3^{max}]$	$[k_4^{min}, k_4^{max}]$
$G_1(s)$	$[3.04e4, 1.46e9]$	$[4.06e6, 4.33e9]$	$[3.36e8, 7.83e9]$	
$G_2(s)$	$[1.88e3, 2.78e6]$	$[1.52e6, 3.49e8]$	$[1.56e8, 5.93e9]$	$[4.05e8, 1.45e10]$
$G_3(s)$	$[2.07e4, 7.43e8]$	$[5.99e6, 4.47e9]$	$[7.26e8, 3.49e9]$	$[9.5e7, 5.11e8]$
$G_4(s)$	$[6.60e3, 3.58e7]$	$[1.94e7, 1.55e10]$	$[1.56e8, 1.95e10]$	
$G_5(s)$	$[1.70e+6, 1.54e9]$	$[8.11e8, 1.97e9]$	$[2.53e8, 6.62e8]$	
$G_6(s)$	$[3.21e3, 1.64e8]$	$[4.66e6, 3.79e9]$	$[3.43e8, 1.13e10]$	
$G_7(s)$	$[6.87, 2.06e3]$			
$G_8(s)$	$[4.842e8, 4.842e8]$	$[9.684e8, 9.684e8]$		
$G_{10}(s)$	$[5.13e3, 1.54e8]$	$[5.55e6, 1.01e9]$	$[3.30e8, 2.45e9]$	$[2e7, 3.85e8]$
$G_{11}(s)$	$[4.79e3, 1.60e8]$	$[1.37e7, 1.04e10]$	$[2.69e8, 2.11e10]$	
$G_{12}(s)$	$[5.05e3, 9.25e7]$	$[7.17e6, 5.36e9]$	$[3.10e8, 1.23e10]$	
$G_{13}(s)$	$[7.86e3, 1.31e6]$	$[1.91e7, 4.22e8]$	$[1.81e8, 8.84e8]$	$[2.25e8, 1.45e9]$
$G_{14}(s)$	$[6.63e4, 9.11e4]$	$[4.11e9, 4.18e9]$		

Tabla 5.12: Límites de los parámetros k_i en la élite.

	$[-a_4^{mix}, -a_4^{max}]$	$[-a_3^{min}, -a_3^{max}]$	$[-a_2^{min}, -a_2^{max}]$	$[-a_1^{min}, -a_1^{max}]$
$G_1(s)$		$[-3.23e6, -5.03e5]$	$[-5.44e5, -1.36e4]$	$[-1.25e4, -42.6]$
$G_2(s)$	$[-1.25e6, -131]$	$[-8.14e5, -1e3]$	$[-6.99e4, -7.14e3]$	$[-6.86e3, -37.8]$
$G_3(s)$	$[-3.33e4, -1.99e3]$	$[-2.24e5, -4.14e4]$	$[-1.76e5, -2.59e4]$	$[-3.8e4, -41.5]$
$G_4(s)$		$[-5.13e6, -6.98e5]$	$[-4.31e6, -1.09e4]$	$[-6.54e4, -88]$
$G_5(s)$		$[-1.65e5, -9.21e4]$	$[-7.49e3, -3.86e3]$	$[-65.8, -25.9]$
$G_6(s)$		$[-3.39e5, -1.16e4]$	$[-2e4, -1.24e3]$	$[-246, -36.5]$
$G_7(s)$				$[-2.71e7, 2.22e5]$
$G_8(s)$			$[-6.69e4, -2.55e3]$	$[-3.34e5, -152]$
$G_{10}(s)$	$[-2.05e6, 3.45e4]$	$[-5.33e7, -6.61e5]$	$[-1.73e7, -1.61e5]$	$[-5.77e6, -119]$
$G_{11}(s)$		$[-4.53e6, -4.54e4]$	$[-2.85e6, -4.59e4]$	$[-9.93e4, -18.4]$
G_{12}		$[-7.24e6, -1.32e6]$	$[-1.41e5, -2.78e4]$	$[-1.71e3, -111]$
G_{13}	$[-2.39e6, 9.13e4]$	$[-8.37e6, -7.17e4]$	$[-7.52e6, -1.14e5]$	$[-2.26e3, -66.1]$
G_{14}			$[-2.67e7, -2.35e5]$	$[-4.46e3, -325]$

Tabla 5.13: Límite de los parámetros a_i en la élite de la forma controlable.

	$[c_1^{min}, c_1^{max}]$	$[c_2^{min}, c_1^{max}]$	$[c_3^{min}, c_3^{max}]$	$[c_4^{min}, c_4^{max}]$	$[d^{min}, d^{max}]$
$G_1(s)$	$[-4.87e6, 3.52e6]$	$[-7.66e7, -6.47e5]$	$[-3.65e7, -5.35e5]$		$[-43.4, 0.48]$
$G_2(s)$	$[-5.06e4, 7.49e5]$	$[-4.98e4, 4.5e4]$	$[-1.52e5, 3.86e4]$	$[-1.7e5, 1.09e3]$	$[-59.3, 0.0228]$
$G_3(s)$	$[-6.47e3, 846]$	$[2.67e3, 3.29e4]$	$[2e4, 3.63e5]$	$[2.56e4, 4.05e5]$	$[0.315, 0.566]$
$G_4(s)$	$[-2.71e7, 4.51e6]$	$[-1.13e8, 2.02e5]$	$[-3.66e7, -1.77e6]$		$[-28.5, 1.02]$
$G_5(s)$	$[-7.41e3, 4.43e3]$	$[-1.03e4, 9.07e3]$	$[-2.35e4, -1.39e4]$		$[0.0541, 0.169]$
$G_6(s)$	$[-4.9e5, 5.57e5]$	$[-1.11e6, -867]$	$[-1.65e6, -1.51e3]$		$[-59.5, 0.262]$
$G_7(s)$	$[-5.69e6, 1.95e6]$				$[-1.83, 3.63]$
$G_8(s)$	$[-3.44e3, 6.05e3]$	$[-2.54e4, 9.52e3]$			$[-0.932, -0.00691]$
$G_{10}(s)$	$[-1.91e7, 7.4e6]$	$[-3.53e7, 1.62e7]$	$[-1.62e8, 1.2e7]$	$[-1.43e7, -5.32e5]$	$[-58.7, -0.0432]$
$G_{11}(s)$	$[-8.69e5, 2.35e6]$	$[-2.58e6, 5.32e5]$	$[-1.51e6, -1.46e4]$		$[-60.7, 0.0627]$
$G_{12}(s)$	$[-1.53e7, 1.08e7]$	$[-3.45e7, -1.5e5]$	$[-4.38e7, -6.72e5]$		$[-63.2, -1.59]$
$G_{13}(s)$	$[-2.11e8, 1.7e8]$	$[-4.6e8, 3.67e8]$	$[-1.89e8, 1.83e8]$	$[-1.17e9, -3.96e5]$	$[-1.44e3, 12.5]$
G_{14}	$[-1.32e9, -3.1e6]$	$[-5.68e10, -3.01e7]$			$[-112, 17.4]$

Tabla 5.14: Límite de los parámetros c_i y d de la élite de la forma controlable.

	$[s_1^{min}, s_1^{max}]$	$[s_2^{min}, s_2^{max}]$	$[p_1^{min}, p_1^{max}]$	$[p_2^{min}, p_2^{max}]$	$[p_3^{min}, p_3^{max}]$	$[p_4^{min}, p_4^{max}]$
$G_1(s)$	$[0.5, 0.9]$		$[-113, -0.486]$	$[-120, -4.23]$	$[-90.1, -2.93]$	
$G_2(s)$	$[0.4, 0.9]$	$[0.3, 0.9]$	$[-467, -9.68]$	$[-111, 179]$	$[-177, -8.32]$	$[-131, 195]$
$G_3(s)$	$[0.6, 0.7]$	$[0.7, 0.9]$	$[-9.73, -5.22]$	$[-411, 66.3]$	$[-20, -1.21]$	$[-38.8, -9.58]$
$G_4(s)$	$[0.5, 0.9]$		$[-10.5, -1.03]$	$[7.55, 25.7]$	$[-114, -1.65]$	
$G_5(s)$	$[0.2, 0.2]$		$[-20.1, -19]$	$[-60.3, -58.7]$	$[-9.83, -8.8]$	
$G_6(s)$	$[0.5, 0.9]$		$[-5.39, -1.76]$	$[-10, 22.3]$	$[-44.4, -2.5]$	
$G_7(s)$			$[-732, -5.09]$			
$G_8(s)$	$[0.6, 0.9]$		$[-208, -133]$	$[10.9, 36.3]$		
$G_{10}(s)$	$[0.5, 0.9]$	$[0.1, 0.9]$	$[-40.2, -2.97]$	$[-208, 99.3]$	$[-29, -1.15]$	$[-42.5, 80.1]$
$G_{11}(s)$	$[0.5, 0.9]$		$[-26.7, -0.231]$	$[-333, 118]$	$[-158, -3]$	
$G_{12}(s)$	$[0.6, 0.9]$		$[-4.49, -3.76]$	$[8.54, 9.24]$	$[-46.1, -23.9]$	
$G_{13}(s)$	$[0.1, 0.9]$	$[0.1, 0.9]$	$[-163, -14.1]$	$[-668, 665]$	$[-328, -50.1]$	$[-278, -78.8]$
$G_{14}(s)$	$[0.5, 0.9]$		$[-306, -106]$	$[-301, -122]$		

Tabla 5.15: Límite de los parámetros s_i y p_i de la élite de la forma diagonal.

	$[b_1^{min}, b_1^{max}]$	$[b_2^{min}, b_2^{max}]$	$[b_3^{min}, b_3^{max}]$	$[b_4^{min}, b_4^{max}]$	$[d^{min}, d^{max}]$
$G_1(s)$	$[-6.35e4, -11.8]$	$[-9.91e3, 5.07e4]$	$[-732, 1.25e4]$		$[-400, -0.261]$
$G_2(s)$	$[-4.88e3, 2.36e4]$	$[-8.71e3, 3.71e4]$	$[-1.93e3, 4.67e4]$	$[-2.58e3, 4.51e3]$	$[-752, -0.206]$
$G_3(s)$	$[-340, 664]$	$[-332, 477]$	$[166, 594]$	$[-169, 158]$	$[-2.47, 16]$
$G_4(s)$	$[-37.2, 2.11e4]$	$[-3.56e4, 385]$	$[-170, 265]$		$[-1.38e3, -7.68]$
$G_5(s)$	$[-214, -168]$	$[3.92e3, 4.06e3]$	$[-65.8, -46.8]$		$[-52, -50.2]$
$G_6(s)$	$[-83.3, 28.5]$	$[-472, 154]$	$[-53.7, 111]$		$[-26.9, -6]$
$G_7(s)$	$[-99.4, 21.3]$				$[-0.996, 0.109]$
$G_8(s)$	$[9.11e3, 1.07e4]$	$[-3.79e3, -1.81e3]$			$[-69, -45.5]$
$G_{10}(s)$	$[-4.86e4, 58]$	$[-1.1e3, 1.22e3]$	$[-182, 140]$	$[-98.5, 134]$	$[-386, 0.767]$
$G_{11}(s)$	$[-2.12e4, 729]$	$[-1.28e4, 231]$	$[-740, 3.03e3]$		$[-59.5, 55.8]$
$G_{12}(s)$	$[162, 227]$	$[5.82, 62.1]$	$[284, 511]$		$[-44.4, -30.1]$
$G_{13}(s)$	$[-1.16e5, 2.65e5]$	$[-6.08e5, 4.46e5]$	$[-9.6e5, 1.59e6]$	$[7.9e3, 4.38e7]$	$[-1.73e5, -837]$
$G_{14}(s)$	$[6.49e5, 4.36e7]$	$[3.42e6, 5.69e7]$			$[-2.15e5, -2.92e4]$

Tabla 5.16: Límite de los parámetros b_i y d de élite de la forma diagonal.

	$[\sigma_1^{min}, \sigma_1^{max}]$	$[\sigma_2^{min}, \sigma_2^{max}]$	$[\sigma_3^{min}, \sigma_3^{max}]$	$[\sigma_4^{min}, \sigma_4^{max}]$
$G_1(s)$	$[0.0473, 9.09]$	$[0.0282, 18.9]$	$[0.37, 13.5]$	
$G_2(s)$	$[0.158, 134]$	$[1.82, 103]$	$[0.186, 156]$	$[0.0347, 98.8]$
$G_3(s)$	$[8.26, 8.48]$	$[107, 107]$	$[8.28, 8.5]$	$[4.18, 5.02]$
$G_4(s)$	$[5.44, 6.55]$	$[9.7, 11.7]$	$[4.78, 6.12]$	
$G_5(s)$	$[9, 42.1]$	$[0.889, 22.5]$	$[16.9, 21.5]$	
$G_6(s)$	$[2.03, 71.4]$	$[3.33, 133]$	$[19.5, 25.7]$	
$G_7(s)$	$[5.32, 13.1]$			
$G_8(s)$	$[3.9, 16.5]$	$[0.00818, 22.2]$		
$G_{10}(s)$	$[0.0256, 13.1]$	$[0.95, 11.2]$	$[0.417, 13.2]$	$[7.02, 32.8]$
$G_{11}(s)$	$[7.21, 16.7]$	$[0.0379, 9.18]$	$[0.171, 10.5]$	
$G_{12}(s)$	$[23.8, 30.4]$	$[101, 115]$	$[3.34, 8.35]$	
G_{13}	$[1.45, 364]$	$[0.724, 394]$	$[3.98, 827]$	$[514, 2.25e4]$
$G_{14}(s)$	$[30.8, 78]$	$[1.01e3, 1.48e3]$		

Tabla 5.17: Límite de los parámetros σ_i de la élite en la forma balanceada.

	$[c_1^{min}, c_1^{max}]$	$[c_2^{min}, c_2^{max}]$	$[c_3^{min}, c_3^{max}]$	$[c_4^{min}, c_4^{max}]$	$[d^{min}, d^{max}]$
$G_1(s)$	$[-4.12e4, 5.65e4]$	$[-1.28e4, 1.74e3]$	$[-9.91e3, 3.59e4]$		$[-55.6, -29.6]$
$G_2(s)$	$[-1.8e4, 2.3e5]$	$[-5.88e4, 4.41e4]$	$[-9.58e4, 2.5e4]$	$[-3.2e4, 1.45e5]$	$[-310, 11.5]$
$G_3(s)$	$[-5.22e3, -465]$	$[1.38e3, 1.79e4]$	$[-1.36e5, -7.1e3]$	$[9.27e3, 2.2e4]$	$[-191, -189]$
$G_4(s)$	$[23.7, 614]$	$[-3.8e3, -32.1]$	$[51.2, 3.2e3]$		$[-3.05, 0.195]$
$G_5(s)$	$[190, 1.19e5]$	$[3.18e3, 1.44e4]$	$[-1.34e4, -716]$		$[-92.3, 6.9]$
$G_6(s)$	$[6.76e4, 2.35e5]$	$[584, 3.49e4]$	$[152, 2.3e5]$		$[-439, -61.7]$
$G_7(s)$	$[-1.1e5, -902]$				$[10.4, 25.3]$
$G_8(s)$	$[6.01, 1.6e3]$	$[-1.07e3, -410]$			$[-32.7, 34.7]$
$G_{10}(s)$	$[-3.06e5, 9.42e5]$	$[-1.28e5, 6.49e5]$	$[-1.19e5, 4.15e4]$	$[49, 8.97e3]$	$[-82.1, -21.9]$
$G_{11}(s)$	$[-8.96e4, 3.84e3]$	$[-2.52e4, 2.71e4]$	$[-1.13e3, 2.71e3]$		$[-46, 5.73]$
$G_{12}(s)$	$[175, 366]$	$[335, 5.34e4]$	$[448, 2.45e4]$		$[-297, -271]$
$G_{13}(s)$	$[-5.54e6, 7.23e6]$	$[-1.45e6, 1.52e6]$	$[-1.46e7, 1.78e7]$	$[275, 4.08e3]$	$[-4.81e4, -1.25e3]$
$G_{14}(s)$	$[-1.5e5, -1e3]$	$[334, 396]$			$[-2.88e3, -1.93e3]$

Tabla 5.18: Límite de los parámetros c_i y d de la élite en la forma balanceada.

En la citada tesis se propone un ejemplo donde la planta a optimizar $G(s)$ es un sistema de tercer orden con seis entradas y seis salidas, mostrado en la ecuación 5.4, con el formato

$$\begin{bmatrix} \dot{x} \\ z_1 \\ z_2 \\ y \end{bmatrix} = T(s) \begin{bmatrix} x \\ w_1 \\ w_2 \\ u \end{bmatrix} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \begin{bmatrix} x \\ w_1 \\ w_2 \\ u \end{bmatrix}.$$

Las dimensiones de las diferentes entradas y salidas son las siguientes: $w_1(2)$, $w_2(2)$, $u(2)$, $z_1(3)$, $z_2(2)$ y $y(1)$. El controlador dinámico del sistema es de la forma $u = K(s)y$, siendo por tanto un controlador con una entrada y dos salidas. La solución de este problema obtenida en la mencionada tesis, tras reducir en lo posible el orden de los controladores es de tercer orden en los casos extremos del frente de Pareto, donde sólo se optimiza uno de los objetivos, y de sexto orden en cualquier otro punto del frente.

$$T(s) = \left[\begin{array}{ccc|cccccc} -0.3908 & -0.4565 & 1.2657 & 0.0488 & 2.2212 & 1.4077 & -1.4924 & -0.4275 & -0.8694 \\ 1.4453 & -1.0491 & -1.2077 & 0.3608 & -0.1063 & 0.9723 & -0.6944 & -0.4470 & -0.0956 \\ -0.1288 & 0.6744 & 1.0324 & 0.3564 & 0.4689 & -1.6050 & -0.6635 & -0.9172 & -1.1833 \\ \hline 0.9420 & -0.6970 & -0.0450 & 0 & 0 & 0 & 0 & 0.5185 & -0.1652 \\ 0.5211 & 0.1187 & 0.3606 & 0 & 0 & 0 & 0 & -2.2230 & 0.9414 \\ 0.0144 & -0.5872 & 1.8972 & 0 & 0 & 0 & 0 & 0.3899 & -1.0126 \\ -1.5567 & -0.0914 & -1.0345 & 0 & 0 & 0 & 0 & 1.9816 & 1.5995 \\ -1.9432 & 1.3575 & 1.5108 & 0 & 0 & 0 & 0 & 0.9964 & 0.3310 \\ 0.2852 & 0.5781 & 0.7574 & -0.6479 & -0.3569 & 0.5157 & 0.5373 & 0 & 0 \end{array}\right] \tag{5.4}$$

El problema planteado para ser resuelto con el algoritmo MRCD incorpora una restricción adicional como un nuevo objetivo: el controlador $K(s)$ debe estabilizar el sistema en lazo cerrado, y como medida de este objetivo se tomó la parte real del polo más inestable del sistema en lazo cerrado, ver ecuación 5.5. Se ha elegido dos posibles estructuras para el controlador, una controlable y otra diagonal, y se ha repetido el problema para controladores de orden dos, tres y seis. Como se menciona en el apéndice C la forma canónica de un controlador de orden n, con m entradas y p salidas necesita $n(m + p) + mp$ parámetros, que en este caso quedan reducidos a $n(m+p)$ por considerar que los parámetros d_i son ceros. Los límites del espacio de búsqueda inicial para los distintos parámetros están en la tabla 5.19. En el caso del controlador de sexto orden en su forma diagonal fue necesario reducir los límites de $[p_i^{min}, p_i^{max}] = [1, 1]$ para evitar el efecto de "ruido colateral" y que los polos del controlador fueran muy grandes. Se ha tomado una probabilidad de cruce $p_{cruce} = 0.7$, una probabilidad de mutación $p_{muta} = 0.1$ y la $t_{dom} = 5$, ver el pseudocódigo 3.10. El número de generaciones para la convergencia de los controladores de segundo y tercer orden fue de 200 frente a las 1000 necesarias para la convergencia de los controladores de sexto orden. Ello se debe al incremento en el número de parámetros a optimizar, 18 parámetros en los controladores de sexto orden.

$$\max_{K} \begin{bmatrix} \left\{ \begin{array}{l} \text{SI } \max(\text{Re}(\lambda(T(s)))) < 0 \to 0, \\ \text{SINO } -\max(\text{Re}(\lambda(T(s)))). \end{array} \right\} \\ -\|T_{z_1w_1}(s)\|_2 \\ -\|T_{z_2w_2}(s)\|_2 \end{bmatrix} \tag{5.5}$$

Los frentes de Pareto obtenidos se muestran en la figura 5.26. La figura de la derecha muestra los resultados para la forma controlable y la de la izquierda para la forma diagonal. Los puntos $(\cdot)$ corresponden a controladores de segundo orden, las cruces $(+)$ a controladores de tercer orden y los estrellas $(*)$ a controladores de sexto orden. La línea continua son los resultados obtenidos en la tesis [Rot90]. En la tabla 5.20 se analizan los polos de los distintos controladores en torno al objetivo $\|T_{z_1w_1}(s)\|_2 = 4$ y $\|T_{z_2w_2}(s)\|_2 = 4.1$, y en la tabla 5.21 los polos de la función en lazo cerrado para los controladores de la tabla 5.20.

Del análisis de las gráficas y tablas se obtienen los siguientes resultados:

- Los controladores con forma controlable y diagonal tienen diferentes polos, aunque en el caso de segundo y tercer orden tiene un frente de Pareto equivalente.
- Los controladores obtenidos son resilientes, es decir, pequeños cambios en sus parámetros producen pequeños cambios en el vector de objetivos.
- Controladores de bajo orden, tiene unas características muy buenas cercanas a las óptimas propuestas en la mencionada tesis.
- El aumento del orden del controlador controlable (orden 6) mejora, aunque no mucho, el frente de Pareto pero su convergencia es más lenta.
- El aumento del orden del controlador en su forma diagonal (orden 6) empeora el frente

	Sub_1	Sub_2	Sub_3	Sub_4	Sub_5
$[a_i^{min}, a_i^{max}]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e4, 1e4]$	$[-1e5, 1e5]$	$[-1e6, 1e6]$
$[b_i^{min}, b_i^{max}]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e4, 1e4]$	$[-1e5, 1e5]$	$[-1e6, 1e6]$
$[c_i^{min}, c_i^{max}]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e4, 1e4]$	$[-1e5, 1e5]$	$[-1e6, 1e6]$
$[d^{min}, d_i^{max}]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$
$[p_i^{min}, p_i^{max}]$	$[-1e2, 1e2]$	$[-1e2, 1e2]$	$[-1e2, 1e2]$	$[-1e3, 1e3]$	$[-1e3, 1e3]$
$[s_i^{min}, s_i^{max}]$	$[0, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$

Tabla 5.19: Límites del espacio de búsqueda para los controladores dinámicos.

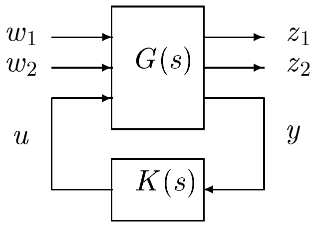


Figura 5.25: Sistema para control mixto $\mathcal{H}_2/\mathcal{H}_2$.

de Pareto, convergiendo en subóptimos, y siendo necesario ajustar los límites del espacio de búsqueda inicial para evitar el efecto "ruido colateral".

Se puede concluir que el algoritmo MRCD tiene más eficacia en controladores de orden reducido obteniendo similares resultados a los conseguidos por controladores de orden superior. Este hecho no sólo es debido al aumento de número de parámetros a optimizar sino a la mayor tendencia de los controladores de orden superior para converger a puntos subóptimos. Por tanto, el algoritmo MRCD obtiene controladores de bajo orden con muy buenas características, que es lo que se desea en el control de sistemas. La forma controlable vuelve a ser mejor que la diagonal en convergencia, ver sección 5.5, consiguiendo llegar en su controlador de sexto orden a valores muy cercanos a los obtenidos en la tesis mencionada.

5.7 Conclusión.

Dado los escasos conocimientos teóricos sobre la convergencia de los algoritmos evolutivos (EAs) y los teoremas de NFL, la forma idónea de ver la bondad de un algoritmo es el uso

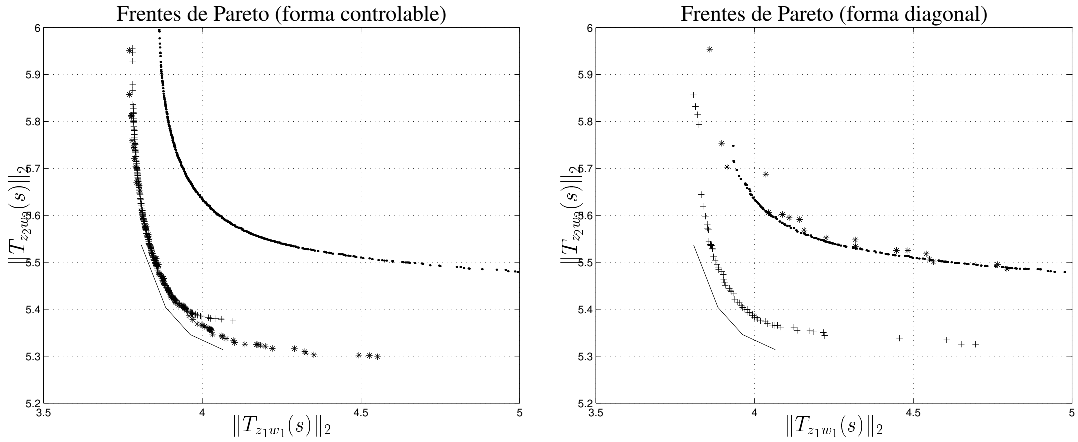


Figura 5.26: Frentes de Pareto con controladores de $2^o(\cdot)$, $3^o(+)$ y $6^o(*)$ orden.

$Orden$	$polos(K(s))$	$[\mathcal{H}_2, \mathcal{H}_\infty]$
$2^o orden[cont.]$	$[-6.41, -2.87e4]$	$[4, 5.63]$
$2^o orden[cont.]$	$[-5.96, -2.76e4]$	$[4.1, 5.58]$
$2^o orden[diag.]$	$[-6.42, -602]$	$[4, 5.63]$
$2^o orden[diag.]$	$[-5.76, -532]$	$[4.1, 5.58]$
$3^o orden[cont.]$	$[-2.22 \pm 1.07i, -8.3e3]$	$[4, 5.38]$
$3^o orden[cont.]$	$[-1.88 \pm 1.06i - 9.7e3]$	$[4.1, 5.38]$
$3^o orden[diag.]$	$[-1.09, -3.8, -250]$	$[4, 5.38]$
$3^o orden[diag.]$	$[-0.937, -3.55, -96.8]$	$[4.08, 5.36]$
$6^o orden[cont.]$	$[-0.221 \pm -0.464i, -0.549, -1.81, -3.75, -2.25e5]$	$[4, 5.37]$
$6^o orden[cont.]$	$[-0.185 \pm -0.39i, -0.654, -1.8, -3.98, -2.28e5]$	$[4.1, 5.33]$
$6^o orden[diag.]$	$[-8.46, -8.69 \pm 0.766i, -9.27, -9.47, -11]$	$[3.91, 5.7]$
$6^o orden[diag.]$	$[-8.48, -8.75 \pm 0.769i, -9.27, -9.48, -11]$	$[4.09, 5.6]$

Tabla 5.20: Polos de los controladores e índice de bondad para $[\mathcal{H}_2, \mathcal{H}_\infty]$ cercanos.

$Orden$	$polos(T(s))$	$[\mathcal{H}_2, \mathcal{H}_\infty]$
$2^o orden[cont.]$	$[-0.482 \pm -1.01i, -1.22, -4.64, -2.87e4]$	$[4, 5.63]$
$2^o orden[cont.]$	$[-0.46 \pm -1.03i, -1.5, -3.94, -2.76e + 004]$	$[4.1, 5.58]$
$2^o orden[diag.]$	$[-0.484 \pm 1.01i, -1.21, -4.65, -602]$	$[4, 5.63]$
$2^o orden[diag.]$	$[-0.455 \pm 1.03i, -1.55, -3.71, -532]$	$[4.1, 5.58]$
$3^o orden[cont.]$	$[-0.501, -0.518 \pm -1.09i, -1.65 \pm 2.33i, -8.3e3]$	$[4, 5.38]$
$3^o orden[cont.]$	$[-0.462, -0.503 \pm -1.09i, -1.35 \pm -2.18i, -9.7e3]$	$[4.1, 5.38]$
$3^o orden[diag.]$	$[-0.338, -0.509 \pm -1.09i, -1.97 \pm 1.77i, -250]$	$[4, 5.38]$
$3^o orden[diag.]$	$[-0.252, -0.5 \pm -1.11i, -1.82 \pm -1.86i, -96.8\pm]$	$[4.08, 5.36]$
$6^o orden[cont.]$	$[-0.201, -0.113 \pm -0.461i, -0.488 \pm 1.13i, -2.02, \ldots]$	$[4, 5.37]$
$6^o orden[cont.]$	$[-0.183, -0.118 \pm -0.405i, -0.479 \pm 1.11i, -1.6 \pm 1.7i, \ldots]$	$[4.1, 5.33]$
$6^o orden[diag.]$	$[-0.936, -0.518 \pm -0.998i, -5.47, -7.44 \pm -3.41i, \ldots]$	$[3.91, 5.7]$
$6^o orden[diag.]$	$[-0.479 \pm -1.04i, -1.31, -4.76, -7.85 \pm -3.31i, \ldots]$	$[4.09, 5.6]$

Tabla 5.21: Polos del sistema para los controladores de la tabla 5.20.

de funciones test. En este capítulo se han elegido diez funciones test multiobjetivo, siete sin restricciones y tres con ellas, y se han resuelto por el algoritmo MRCD. El resultado ha sido muy positivo pudiendo afirmar que el algoritmo MRCD iguala y, en la mayor parte de los casos, mejora a los algoritmos existentes.

Con el fin de comprobar la eficiencia del algoritmo MRCD en el control multiobjetivo se ha diseñado dos problemas mixtos $\mathcal{H}_2/\mathcal{H}_\infty$, uno en realimentación de estado y otro de salida, para trece diferentes sistemas. Los resultados han sido comparados con los límites superiores del frente de Pareto obtenidos gracias a las técnicas de LMIs. Los frentes de Pareto obtenidos con el algoritmo MRCD minimizan en todos los casos a los obtenidos con las técnicas LMIs. Para implementar los controladores dinámicos se ha utilizado tres parametrizaciones; controlable, diagonal y balanceada; siendo mejor la forma controlable por ser menos sensible a los límites iniciales del espacio de búsqueda y tener menos "ruido colateral". Por último se ha valorado el algoritmo MRCD al modificar el orden del controlador dinámicos que se desea optimizar llegándo a la conclusión de que el algoritmo MRCD es ideal para hallar buenos controladores de bajo orden y que la forma controlable tiene mejores propiedades de convergencia que la forma diagonal.

Capítulo 6

Sintonización de controladores PIDs (Proporcionales, Integrales y Derivativos).

6.1 Introducción.

Los controladores PID (Proporcional, Integral y Derivativo) son los más usados en los procesos industriales. Sus principales propiedades son su implementación en realimentación, la eliminación del error estacionario a partir de su efecto integrador y su anticipación hacia el futuro debido a su efecto derivativo. 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 por producir inconvenientes en plantas con ruido. Pese a los avances en la teoría de control y en las tecnologías de control de las últimas décadas, los PIDs han sobrevivido a estos cambios. La introducción de los computadores como elementos de control ha hecho que los PIDs se transformen en un programa, incorporándose nuevos elementos a su estructura original, tales como la sintonización automática o la adaptación de sus parámetros a los cambios en la planta que debe controlar.

Existen muchos métodos para sintonizar los parámetros de un PID, por ejemplo es sobradamente conocido el método de Ziegler-Nichols. Un inconveniente de todos estos métodos clásicos de sintonización de PIDs es su gran sensibilidad a cambios en el modelo de la planta.

Cuando la planta es un sistema MIMO (varias entradas y varias salidas) los métodos clásicos de sintonización de PIDs consideran las entradas y salidas como independientes, sin tener en cuenta los posibles acoplamientos, por lo que los controladores obtenidos pueden ser malos. Los métodos de diseño vistos en el capítulo 2 no son válidos para este tipo de controladores por tener una estructura fija, lo que lleva a que sea necesario la resolución del problema para un controlador genérico y su aproximación a una estructura PID, hecho que en la mayoría de los casos no será posible. La sintonización de PIDs es aún un tema abierto pese a ser uno de los primeros problemas planteados en la teoría de control.

La formulación paramétrica del diseño multiobjetivo (2.5) es totalmente consecuente con la sintonización de PIDs. En este capítulo se va a plantear un diseño multiobjetivo de PIDs a partir de la minimización de una serie de normas de diferentes funciones de trasferencia que miden los objetivos deseados. Un primer objetivo que se desea optimizar es la rapidez de respuesta ante una señal de referencia. El segundo objetivo es minimizar la sensibilidad ante cambios en la estructura de la planta para obtener robustez. Un tercer objetivo es atenuar el efecto de un error en la señal de control y minimizar el posible efecto del ruido sobre la señal de salida. Por último, se pide que la amplitud de la señal de control no pase de un valor dado para evitar saturaciones.

Una vez plateado el problema se va a obtener controladores para trece sistemas de ensayo SISO, propuestos en [APH98], y con los que ya se ha trabajado en la sección 5.5. Se proponen dos problemas, el control del sistema mediante PIs y su control mediante PIDs. Los ensayos se harán de forma estándar, fijando los mismos límites iniciales del espacio de búsqueda para todos los casos. Los resultados se comparan con los obtenidos en dicho artículo, donde se optimiza los parámetros de controladores PIs frente a una serie de objetivos.

Tras los resultados obtenidos en los sistemas SISO se pretende ampliar el estudio a sistemas MIMO. Para ello se combinan cuatro subsistemas SISO de los propuestos para crear varios sistemas MIMO dos entradas dos salidas con distinto grado de acoplamiento. Dichos sistema van a ser controlados por dos PIDs desacoplados o por cuatro PIDs acoplados. En el caso de los cuatro PIDs acoplados, con dos filtros, se deberá sintonizar 26 parámetros al mismo tiempo.

Los resultados de este capítulo han sido publicados como ponencia en el congreso “IFAC Workshop on Digital Control. Past, Present and Future of PID Control”, [HLBLPG00].

La estructura del capítulo es la siguiente, en la sección 6.2 se plantean los objetivos deseados en su formulación matemática usando normas entre entradas y salidas del sistema en lazo cerrado. En la sección 6.3 se discute las razones por las que el algoritmo MRCD es válido en la resolución de este problema. En la sección 6.4 se plantea trece sistemas de ensayo SISO para los que se resuelve el problema por medio de controladores PIs y PIDs. En la sección 6.5 se plantean varios sistemas MIMO de dos entradas, dos salidas para los que se resuelve el mismo problema. En la sección 6.6 se dan las conclusiones del capítulo.

6.2 Diseño de controladores PIDs como un problema multiobjetivo.

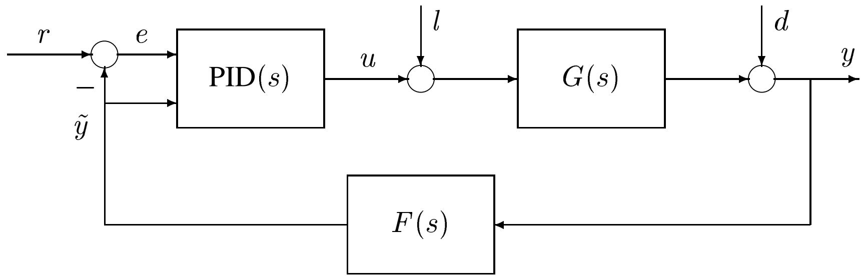


Figura 6.1: Diagrama de bloques del sistema en lazo cerrado con un controlador PID.

Las especificaciones que se tratarán de conseguir en el diseño de controladores PID están orientadas en tres sentidos; la optimización del tiempo de respuesta del sistema en lazo cerrado para llegar al punto de referencia deseado, la minimización del efecto de posibles ruidos en la planta y la robustez ante variaciones en la estructura de la planta. Se tratará de unificar estos tres objetivos en la optimización de una serie de normas entre las diferentes entradas y salidas del sistema en lazo cerrado, formulando el problema como en (2.5).

Se considera el diagrama de bloques descrito en la figura 6.1. El proceso que se desea controlar es modelado como un sistema lineal con una función de transferencia $G(s)$. La entrada y salida del proceso se representan por $u(t)$ y $y(t)$. La señal de referencia viene representada por $r(t)$. Normalmente la señal de referencia en los procesos industriales es constante y puede cambiar cada cierto tiempo en función de las condiciones de trabajo, hecho por lo que se puede modelar como una secuencia de funciones escalón.

El sistema de control está sujeto a ruidos, considerándose como básicos el ruido $l(t)$ de la acción de control y el ruido $d(t)$ de la salida del sistema. El ruido en la acción de control se modela como una función escalón y el ruido salida como un proceso estocástico, por ejemplo ruido blanco. Por considerar que la señal de salida lleva ruido, dicha señal se filtra en su realimentación de la forma

$$F(s) = \frac{1}{1 + sT_f}. \tag{6.1}$$

La señal de salida filtrada es denomina como $\tilde{y}(t)$.

El modelo de controlador PID que se han considerado en este capítulo tiene la siguiente estructura

$$\begin{aligned}
u(s) &= k\left(e_p(s) + \frac{1}{T_i s}e(s) + \frac{T_d s}{1 + T_d s/N}e_d(s)\right) && (6.2)\\
e(s) &= r(s) - \tilde{y}(s) && (6.3)\\
e_p(s) &= br(s) - \tilde{y}(t) \\
&= be(s) + (b-1)\tilde{y}(s), && (6.4)\\
e_d(s) &= cr(s) - \tilde{y}(s) \\
&= ce(s) + (c-1)\tilde{y}(s), && (6.5)
\end{aligned}$$

donde $0 \leq b, \;\; c \leq 1$. Por lo tanto su función de transferencia viene dada por

$$\begin{aligned}
\text{PID}(s) &= \begin{bmatrix} C_1(s) \\ C_2(s) \end{bmatrix} \\
&= \begin{bmatrix} k\left(b + \frac{1}{T_i s} + \frac{cT_d s}{1+T_d s/N}\right) \\ k\left(b - 1 + \frac{(c-1)T_d s}{1+T_d s/N}\right) \end{bmatrix} && (6.6)
\end{aligned}$$

Esta estructura es la forma ISA de un PID, ver apéndice E, y es más flexible que la estructura del PID básico, que responde al caso particular $b = c = 1$. La variación de los parámetros b y c no tiene ninguna influencia en el efecto de los ruidos en la señal de control y en la señal de salida, pero sí la tiene en las variaciones en la referencia. Cuando b tiene pequeños valores se limita el efecto que el cambio de referencia tiene en la acción proporcional y lo mismo ocurre cuando c tiene pequeños valores con la acción derivativa. Esta propiedad va a ser importante debido a que la señal de referencia está formada en la mayoría de los casos por escalones, ver [AH94].

El sistema en lazo cerrado va a estar compuesto de funciones de transferencia de la forma $T_{zw}(s)$, donde w es la entrada y z la salida. Con una notación matricial, la función de transferencia del sistema en lazo cerrado se puede escribir de la forma

$$\begin{bmatrix} e \\ y \\ u \end{bmatrix} = T(s) \begin{bmatrix} r \\ l \\ d \end{bmatrix} = \begin{bmatrix} T_{er} & T_{el} & T_{ed} \\ T_{yr} & T_{yl} & T_{yd} \\ T_{ur} & T_{ul} & T_{ud} \end{bmatrix} \begin{bmatrix} r \\ l \\ d \end{bmatrix} \tag{6.7}$$

Los objetivo para los problemas de diseño pueden ser expresados en función de los términos de la matriz de funciones de transferencia.

6.2.1 Especificaciones para el diseño de controladores PIDs.

Cualquier formulación de un problema de diseño de controladores PID debe satisfacer las siguientes especificaciones:

- Seguimiento de la señal de referencia.
- Atenuación del ruido de la señal de control.
- Robustez respecto de las incertidumbres del modelo.
- Atenuación del ruido de salida.
- Límites del actuador.

La importancia de cada especificación está en función de la aplicación de control. En el control de procesos, la atenuación el ruido en la señal de control 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, por ejemplo en control de motores, robótica, un buen seguimiento de la señal de referencia es muy importante.

Otro objetivo fundamental es la robustez respecto a las posibles incertidumbres de la planta. Muchos de los métodos clásicos para la sintonización de PIDs, por ejemplo la regla de Ziegler-Nichols ver [AH94], proporcionan PIDs que carecen de robustez, lo que origina que pequeños cambios en la planta desestabilicen el sistema.

Ruido en la señal de control.

Este efecto se va a medir por la integral de la señal de salida para una señal escalón en la entrada de control, suponiendo que no existen señales en ninguna otra entrada al sistema

$$J_p = \int_0^{\infty} |y(t)|^p dt, \; p \in \mathbb{N}. \tag{6.8}$$

Normalmente se elige el valor 1 o 2 para p.

Dado que $y(t)$ en (6.8) se puede expresar como la inversa de la transformada de Laplace de $T_{yl}\frac{1}{s}$ donde T_{yl} es la función de transferencia entre la entrada l y la salida y la ecuación (6.8) se puede expresar como

$$J_p = \left(\left\| T_{yl}\frac{1}{s} \right\|_p \right)^{1/p}, \tag{6.9}$$

y $\|\cdot\|_p$ expresa la norma-p, $\mathcal{L}_p[0,\infty)$, del impulso de respuesta del sistema. Para $p = 2$ la norma-2, $\mathcal{L}_2[0,\infty)$, de la salida es igual a la norma $\mathcal{H}_2$ del sistema, y para $p = 1$ dicha norma-1 es la norma $\mathcal{L}_1$, ver apéndice B.

Las normas formuladas están bien definidas ya que el cero en el origen de la función T_{yl} se cancela con el término $\frac{1}{s}$. Por otro lado, $T_{yl}(s)$ es una función de transferencia estrictamente propia con lo que las norma son siempre finitas. En la mayoría de las aplicaciones es preferible la norma $\mathcal{L}_1$, ya que la norma $\mathcal{H}_2$ produce oscilaciones en las respuestas del sistema de lazo cerrado.

Robustez frente a las incertidumbres de la planta.

Este efecto puede ser medido 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. Para el diagrama de bloques mostrado en la figura 6.1, $L(s)$ viene dado por

$$L(s) = (C_1(s) + C_2(s))G(s)F(s). \tag{6.10}$$

El pico en la ganancia de la sensibilidad viene dado por su norma $\mathcal{H}_\infty$, $\|S(s)\|_\infty = |S(j\omega)|$, que para un sistema SISO corresponde con 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 en la planta su valor debe ser pequeño. La sensibilidad debe ser siempre mayor que 1 porque $|S(j\infty)| = 1$. La función de sensibilidad es la función de transferencia en lazo cerrado T_{yd}, definida en (6.7).

Seguimiento de la señal de referencia.

Este efecto se mide por diferentes características de la señal de respuesta del sistema a una entrada de referencia escalón, tales como el tiempo de subida, el tiempo de establecimiento, el sobrepico o el error estacionario.

Estas mismas especificaciones pueden expresarse con índices integrales como los definidos en (6.8). Puesto que un cambio en la señal de referencia produce un gran aumento del error, se puede definir el objetivo

$$J_{tp} = \left(\int_0^{\infty} |te(t)|^p dt \right)^{1/p} \tag{6.11}$$

donde $e(t)$ es la salida a una señal escalón en la entrada $r(t)$. Como $e(t)$ es la respuesta impulso de un sistema con la función de transferencia $T_{er}(s)\frac{1}{s}$, y $te(t)$ es la respuesta impulso a un sistema con una función de transferencia $T_{er}(s)\frac{1}{s}$, el objetivo integral J_{tp} es equivalente a la norma-p, $\mathcal{L}_p(0, \infty)$,

$$J_{tp} = \left\| \frac{d}{ds} \left(T_{er} \frac{1}{s} \right) \right\|_p . \tag{6.12}$$

Atenuación del ruido de salida.

El ruido de salida se modela como un proceso estocástico con una densidad de potencia espectral $\Phi_n(\omega)$, ver [Fri89]. El objetivo es atenuar el efecto de este ruido a la salida, y se puede formular usando la norma $\mathcal{H}_2$ de la función de transferencia filtrada

$$\|T_{yd}(s)W(s)\|_2, \tag{6.13}$$

donde el filtro $W(s)$ corresponde a la banda de frecuencias que se desea atenuar.

Límites en el actuador.

Cada señal del sistema físico de control tiene unos rangos máximos. El objetivo de diseño es en este caso no sobrepasar estos límites con el fin de no saturar los límites de actuador. Este objetivo se puede formular como una restricción en la norma-∞, $\mathcal{L}_\infty(0, \infty)$, de la señal de control para una entrada de referencia escalón,

$$\left\| T_{ur} \frac{1}{s} \right\|_\infty < \alpha, \tag{6.14}$$

donde α es el nivel de saturación.

6.2.2 Problema de optimización multiobjetivo.

A partir de los objetivos antes señalados se va a poder plantear un problema de optimización multiobjetivo para el diseño de controladores PIDs. Considerar el diagrama de bloques de la figura 6.1, donde $G(s)$ es la planta, $K(s)$ es el controlador PID y $F(s)$ el filtro de la señal de salida, definidos por la ecuaciones (6.6) y (6.1) respectivamente. Se desea sintonizar los parámetros tanto del PID como del filtro.

El problema de diseño se convierte en un problema de optimización multiobjetivo en los siguientes términos: Encontrar el conjunto de controladores $K(s)$ y los filtros $F(s)$ que minimicen el siguiente vector de objetivos:

$$\begin{bmatrix} \left\| T_{yl}(s)\frac{1}{s} \right\|_p \\ \left\| \frac{d}{ds}\left(T_{er}(s)\frac{1}{s} \right) \right\|_q \\ \| T_{yd}(s) \|_\infty \\ \| T_{yd}(s) W(s) \|_2 \end{bmatrix} \tag{6.15}$$

donde $p, q \in \{1, 2\}$, y que cumpla con las restricción $\|T_{ur}\frac{1}{s}\|_\infty < \alpha$. En la mayoría de los casos $p = 1$ y $q = 2$, pero se pueden elegir otros valores dependiendo del cada caso particular.

6.3 Ventajas del uso del algoritmo MRCD.

En la sección anterior se ha definido el diseño de controladores PID multiobjetivo con el planteamiento general definido en la ecuación (2.1).

El principal factor a tener en cuenta al elegir el método idóneo para la resolución del problema planteado es la estructura fija del controlador que se desea. Este hecho lleva a formalizar el problema como una optimización multiobjetivo respecto a los parámetros del controlador, ver ecuación (2.5). Si se utilizarán métodos similares a los vistos en la sección 2.3 el resultado sería un controlador de orden elevado y sin estructura fija. Tras ello, se requiere un segundo paso donde se intentaría obtener un PID con características similares al controlador encontrado, hecho que no será posible en la mayor parte de los casos.

En [APH98] se propone un problema de optimización no convexa de una serie de objetivos con respecto a los parámetros de un PI en una planta SISO. En ese artículo, se convierte el problema multiobjetivo en un problema de un simple objetivo con restricciones, de forma similar a los métodos formulados en la sección 3.3. Tras un laborioso proceso consigue encontrar un sólo controlador y no el conjunto de óptimos de Pareto. El método formulado en dicho artículo no puede ampliarse a un controlador PID, o a un sistema MIMO, por ser su desarrollo mucho más complejo.

La formulación de la sintonización de PIDs como un problema de optimización de una serie de normas respectos de los parámetros del mismo, ecuación (2.5), y la aplicación para su resolución del algoritmo MRCD proporcionan un marco de trabajo totalmente abierto. Por un lado, el algoritmo obtiene un conjunto de controladores próximos al conjunto de óptimos de Pareto como solución. Por otro lado, es posible cambiar cualquier objetivo y/o la estructura del controlador sin ningún problema. En este capítulo se han sintonizado controladores PI y PID para trece plantas SISO, comparando los resultados. Luego se han sintonizado dos PIDs desacoplados y cuatro acoplados para varios sistemas MIMO con distinto grado de acoplamiento. El algoritmo ha sido capaz de sintonizar a la vez hasta 24 parámetros.

6.4 Sistemas de ensayo SISO (una entrada, una salida).

En esta sección se plantea la sintonización de los parámetros de controladores PIs y PIDs para minimizar los objetivos definidos en (6.15) en trece diferentes sistemas de ensayo. Para ello se usará el algoritmo MRCD, ver capitulo 4. La estructura de los PIDs es la mostrada en la ecuación (6.6). Los controladores PI se obtienen cuando $T_d = 0$, lo cual hace que los parámetros c, T_d y N sean irrelevantes.

Los sistemas de ensayo han sido propuestos en [APH98] para el diseño de PIs con técnicas de optimización no convexa. En este artículo se diseñan controladores PIs que optimicen parte de los objetivos propuestos en (6.15). Se ha respetado el nombre de los sistemas con el fin de que los resultados obtenidos en el citado artículo y los obtenidos con el algoritmo MRCD sean fácilmente comparables. En la tabla 5.3 se muestra la función de transferencia de los citados sistemas, y en la sección 5.5 se describe las características de cada sistema junto con los problemas que puede presentar su control.

El sistema $G_{12}(s)$ tiene como constante $a = 0$, lo que hace que sea críticamente estable y por tanto muy difícil de controlar. El sistema $G_{14}(s)$ tiene como constante $a = 4$, siendo por tanto un sistema inestable que será difícil de controlar por un PI. Se ha tomado la aproximación de

Padè de segundo orden, ver [Fri96], para diseñar los controladores correspondientes a sistemas con retardo; $G_3(s)$, $G_7(s)$ y $G_8(s)$.

Se va a asumir, al igual que en [APH98], que no existe ruido de medida, con lo que el problema tiene tres objetivos y una restricción. Al plantear el problema con el algoritmo MRCD se añade otra restricción, el controlador debe estabilizar la planta en lazo cerrado. La medida de estabilidad de la planta va a ser el valor de la parte real del polo en lazo cerrado más inestable. Agrupando las dos restricciones del problema y planteado el mismo como un problema de maximización del valor negativo de los objetivos, el mismo se puede formular en los siguientes términos: Encontrar el conjunto de controladores y filtros $(K(s), F(s))$ que maximicen (en el sentido de Pareto) los siguientes objetivos

$$\begin{bmatrix} \left\{ \begin{array}{l} \text{SI Re}(\lambda(T(s))) < 0 \text{ Y } \|T_{ur}\frac{1}{s}\|_\infty < \alpha, \rightarrow 0 \\ \text{SINO } -\max(\text{Re}(\lambda(T(s))), \alpha - \|T_{ur}\frac{1}{s}\|_\infty). \end{array} \right\} \\ -\left\|T_{yl}(s)\frac{1}{s}\right\|_1 \\ -\left\|\frac{d}{ds}(T_{er}(s)\frac{1}{s})\right\|_2 \\ -\left\|T_{yd}(s)\right\|_\infty \end{bmatrix} \tag{6.16}$$

donde Re$(\lambda(T))$ devuelve la parte real de los polos del sistema. Para los ejemplos mostrados se ha tomado $\alpha = 5$.

Los límites del espacio de búsqueda inicial son los mostrados en la tabla 6.1, y los parámetros del algoritmo MRCD son; probabilidad de cruce $p_{cruce} = 0.7$, probabilidad de mutación $p_{muta} = 0.1$ y la constante $t_{dom} = 5$, ver pseudocódigo 4.1 y 3.10.

6.4.1 Evaluación de resultados.

El problema planteado en la ecuación (6.16) ha sido resuelto por medio del algoritmo MRCD para el caso en el que el controladores sea un PID con la estructura definida en (6.15), y para el caso particular donde la estructura sea un PI. El problema se ha repetido con los trece sistemas de ensayo definidos en la tabla 5.3 y las gráficas 6.2-6.27 muestran los resultados.

Para cada uno de los sistemas de ensayo se muestran dos figuras. La primera está dividida en cuatro secciones que contienen el frente de Pareto en tres dimensiones y sus correspondientes proyecciones, de los vectores objetivos asociados a los controladores PIs $(\cdot)$ y PIDs $(+)$ pertenecientes a la élite. La segunda gráfica, dividida también en cuatro secciones, representa la respuesta temporal y la curva de Nyquist para dos controladores PIs seleccionados, con sensibilidades en torno a 1.4 y 1.8, y para dos controladores PIDs, con sensibilidades en torno a 1.3 y 1.7.

En la respuesta temporal se dibuja la señal de salida $y(t)$ y la señal de control $u(t)$ del sistema en lazo cerrado ante una señal escalón inicial en la entrada $r(t)$, y una señal escalón intermedia en la perturbación a la señal de control, $d(t)$, ver figura 6.1. La magnitud del escalón en $r(t)$ es uno y la magnitud de $d(t)$ depende de cada caso, siendo para todos ellos del mismo orden que los empleados en [APH98]. Su valor se puede deducir de la posición estacionaria de la señal $u(t)$. En el diagrama de Nyquist se dibuja la curva del sistema más el controlador en lazo abierto para $w > 0$, y el máximo círculo centrado en el punto crítico $-1 + j0$, que no encierra a algún punto de dicha curva. Notar que el radio de dicho circulo es la inversa de la norma $\mathcal{H}_\infty$ de la sensibilidad, $\|T_{yd}\|_\infty$.

La tabla 6.2 contiene los parámetros PI y PID elegidos, y el valor del vector objetivo asociado a cada uno de ellos.

Comparación con los resultados de [APH98]. La elección de los sistemas de ensayo ha estado motivada por el artículo [APH98], donde se optimizan PIs con varios de los objetivos propuestos en (6.16). En general, la comparación de la respuesta temporal y las curvas de Nyquist correspondientes a los PIs obtenidos por el algoritmo MRCD y los obtenidos en dicho artículo es muy positiva. En todos los caso se iguala o mejora la respuesta temporal, y a su vez se obtiene menor sensibilidad, ya que se ha elegido PIs con un valor de norma $\mathcal{H}_\infty$ de la sensibilidad en torno a 1.4 y 1.8. Un ejemplo es el sistema $G_6(s)$ donde se disminuye el sobrepico, sin aumentar el tiempo de establecimiento y disminuyendo la sensibilidad.

Por encima de los casos concretos, el algoritmo MRCD da una flexibilidad que no tiene el método propuesto por dicho artículo. Con el algoritmo MRCD se puede plantear un problema con idénticos objetivos para un controlador PI, para uno PID, en un sistema SISO y, como se verá en la próxima sección, en un sistema MIMO.

Efecto positivo de la acción derivativa. La gráficas correspondientes a los frentes de Pareto de los diferentes sistemas de ensayo, muestra de forman general que los frentes pertenecientes a controladores PIDs minimizan los obtenidos por controladores PIs. Las dos principales propiedades de los sistemas controlados por PIDs frente a los controlados por PIs son su mayor rapidez y su menor sensibilidad a ruidos en la planta. La causa de ello es la mayor versatilidad de los controladores PIDs, que añaden a la acción proporcional e integral de los PIs una acción derivativa. Esta acción hace que el controlador PID no sólo reaccione ante la señal de error y de su integral, sino que también lo puede hacer por la variación de dicha señal, anticipándose a un posible error futuro. Si la acción derivativa es muy elevada se puede saturar el actuador, por lo que se ha añadido una restricción respecto al máximo de la acción de control ante una entrada escalón, ver ecuación (6.16).

La diferencia entre ambos tipos de controladores es más acusada en algunos sistemas; por ejemplo el sistema G_1, $G_2(s)$ o G_{13}; donde con un controlador PID se aumenta la velocidad de respuesta, se disminuye el sobrepico, se disminuye la sensibilidad y el efecto del error. Un caso singular es el sistema $G_{14}(s)$ que al ser inestable es difícil de controlar por un PI y sin embargo se controla sin problemas con un PID.

Efecto negativo de la acción derivativa. Los sistemas para los cuales el efecto derivativo no ha sido positivo han sido aquellos con grandes retardos $G_3(s)$, $G_7(s)$, y el sistema críticamente estable $G_{12}(s)$. Para hallar los controladores PID de sistemas con retardo se ha usado la aproximación de Padè de segundo orden, y la norma $\mathcal{H}_\infty$ de la sensibilidad obtenida en estos casos es menor que la real debido a perturbaciones de alta frecuencia, ver figuras 6.7 y 6.15. Dichas perturbaciones se traducen en ruidos en la respuesta temporal producidos por la acción derivativa, que no es útil para sistemas con retardo.

Los sistemas críticamente estables, con polos complejos conjugados en el eje imaginario, son muy oscilatorios y el efecto derivativo no consigue corregirlo, sino que lo potencia, ver figuras 6.22 y 6.23. La acción derivativa es negativa y el algoritmo tiende a eliminarla, no siendo fácil debido a la forma en la que se ha definido el controlador, ver tabla 6.2.

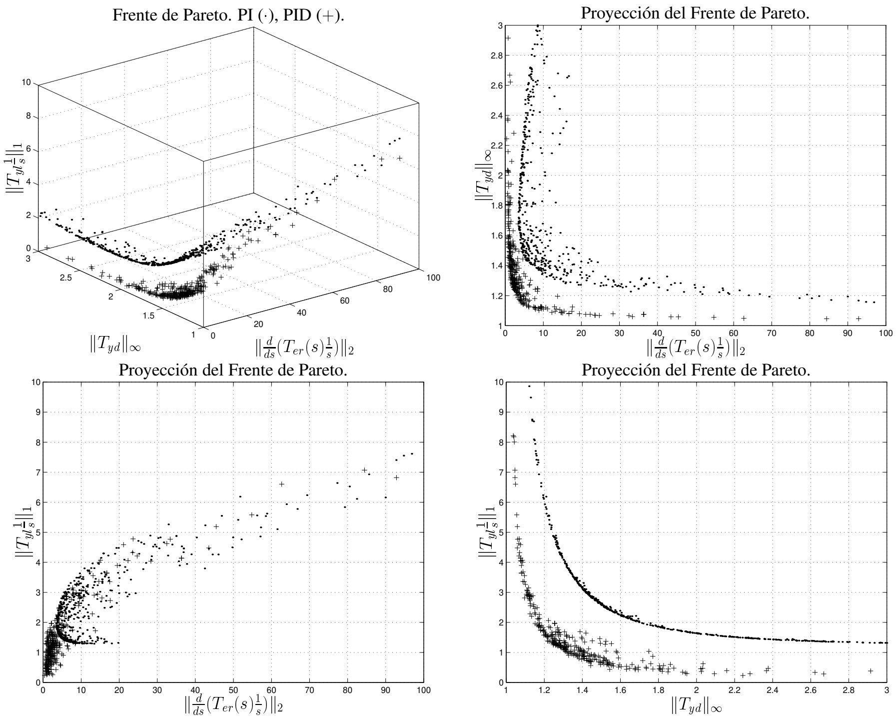


Figura 6.2: Frente de Pareto asociado a la élite de PIs y PIDs para $G_1(s)$.

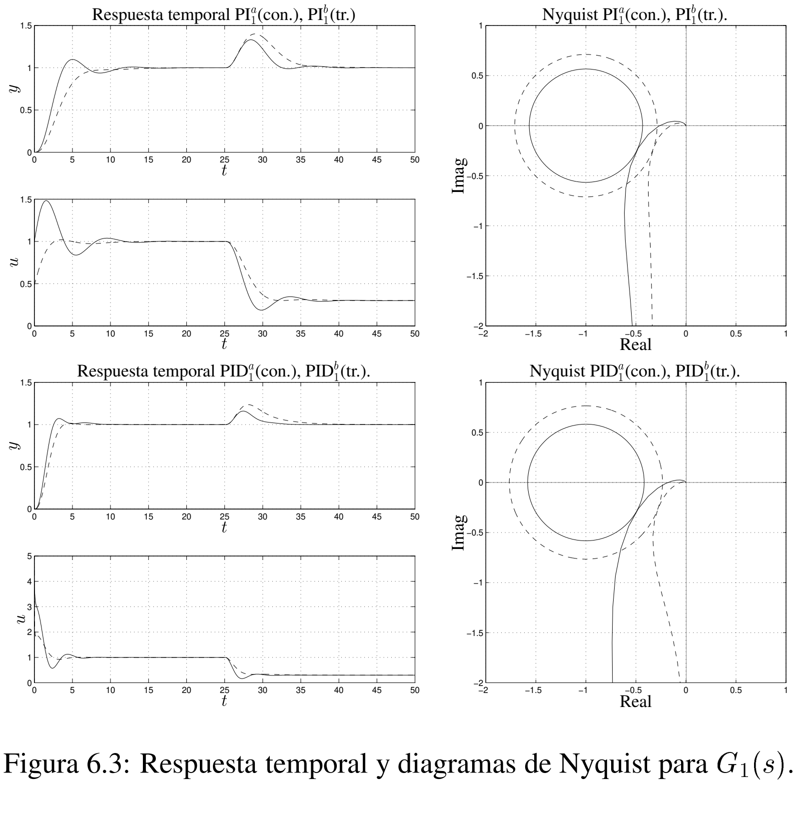


Figura 6.3: Respuesta temporal y diagramas de Nyquist para $G_1(s)$.

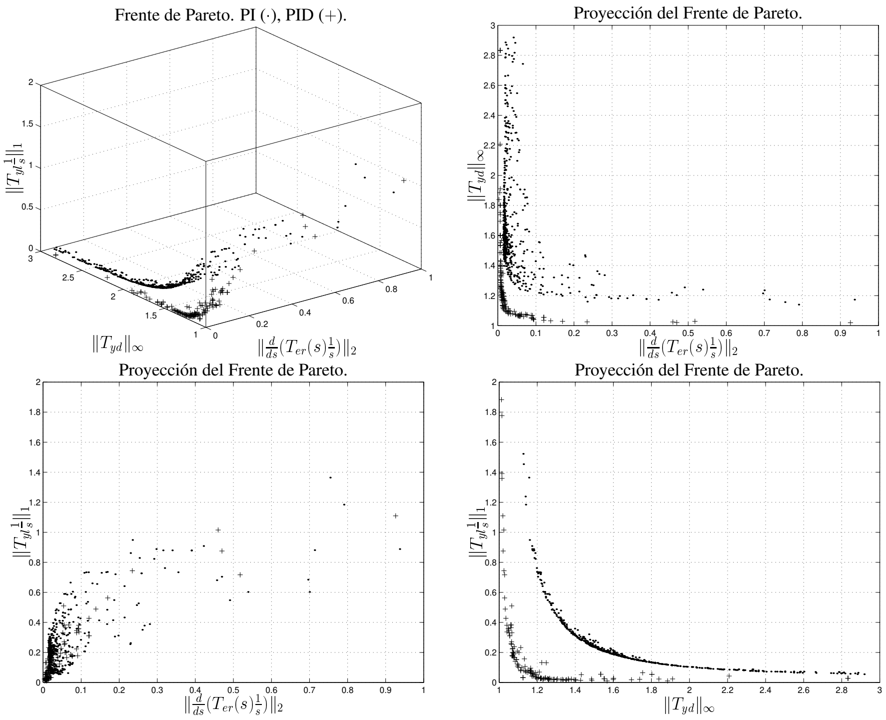


Figura 6.4: Frente de Pareto asociado a la élite de PIs y PIDs para $G_2(s)$.

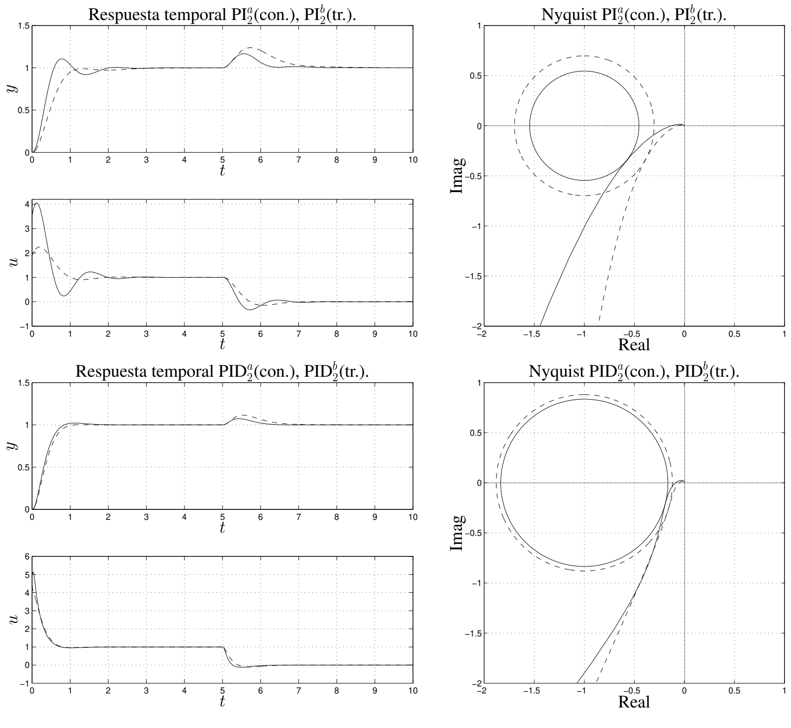


Figura 6.5: Respuesta temporal y diagramas de Nyquist para $G_2(s)$.

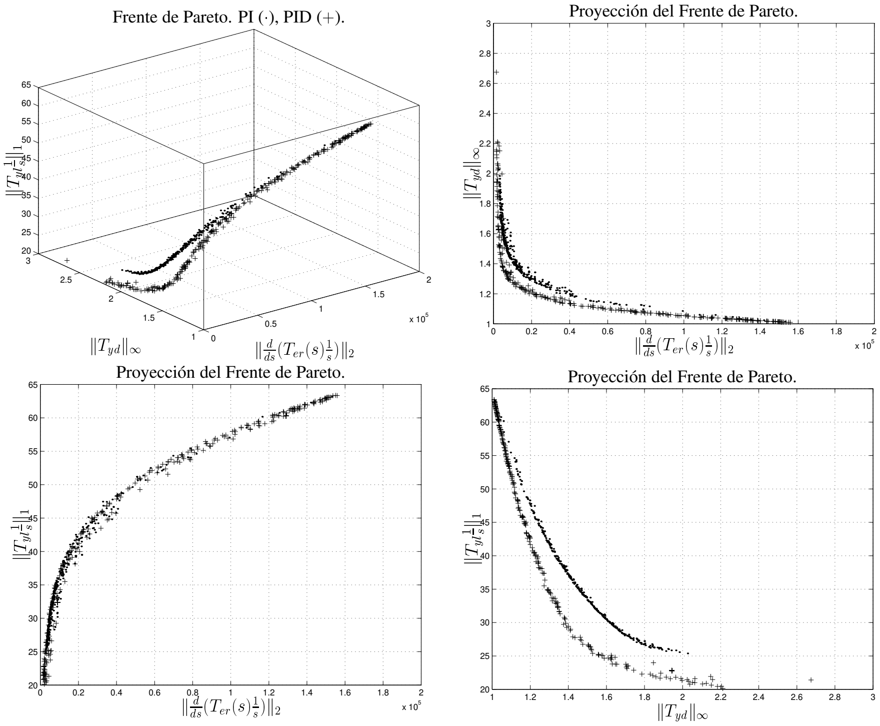


Figura 6.6: Frente de Pareto asociado a la élite de PIs y PIDs para $G_3(s)$.

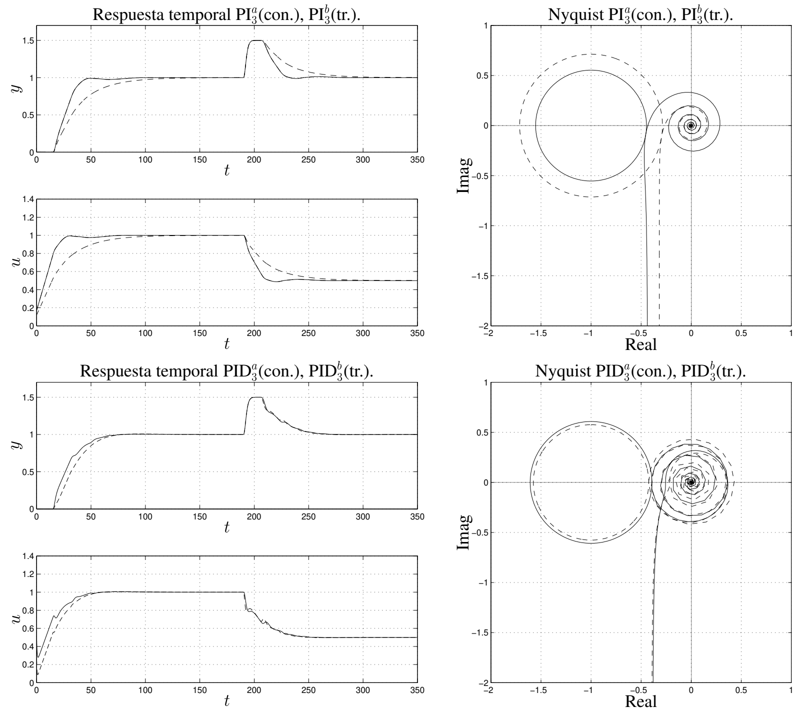


Figura 6.7: Respuesta temporal y diagramas de Nyquist para $G_3(s)$.

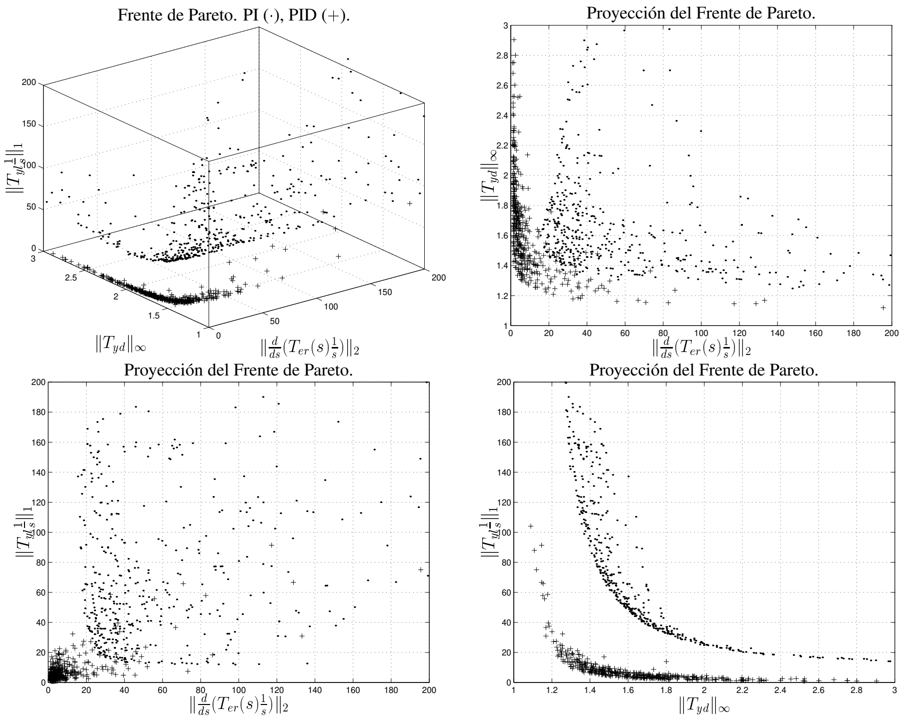


Figura 6.8: Frente de Pareto asociado a la élite de PIs y PIDs para $G_4(s)$.

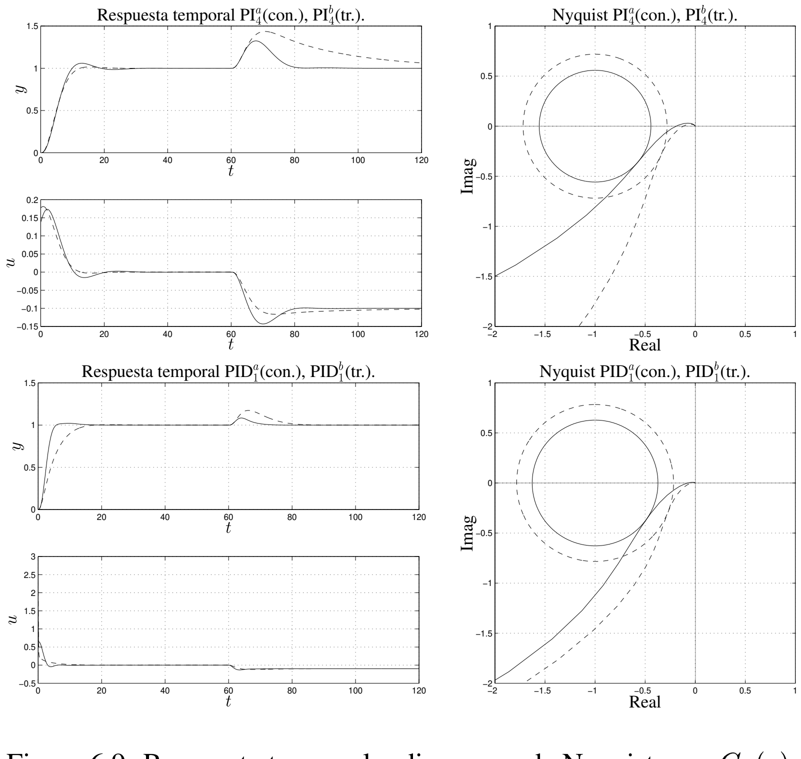


Figura 6.9: Respuesta temporal y diagramas de Nyquist para $G_4(s)$.

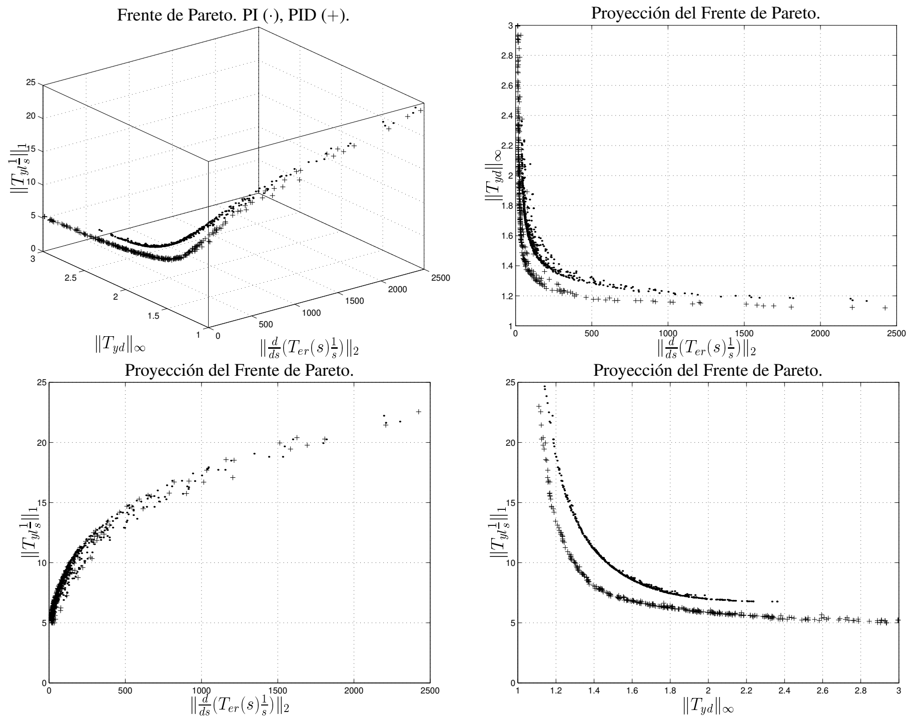


Figura 6.10: Frente de Pareto asociado a la élite de PIs y PIDs para $G_5(s)$.

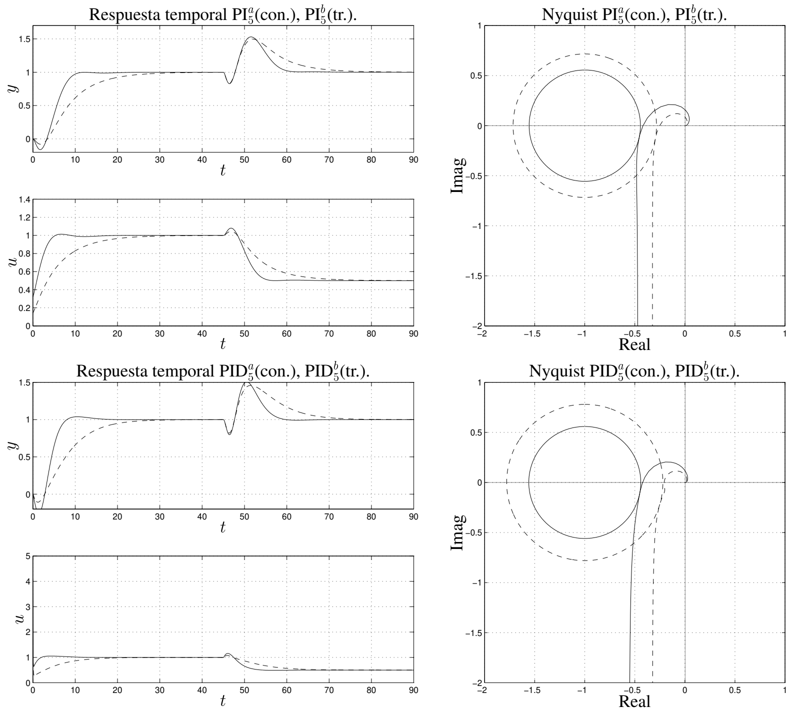


Figura 6.11: Respuesta temporal y diagramas de Nyquist para $G_5(s)$.

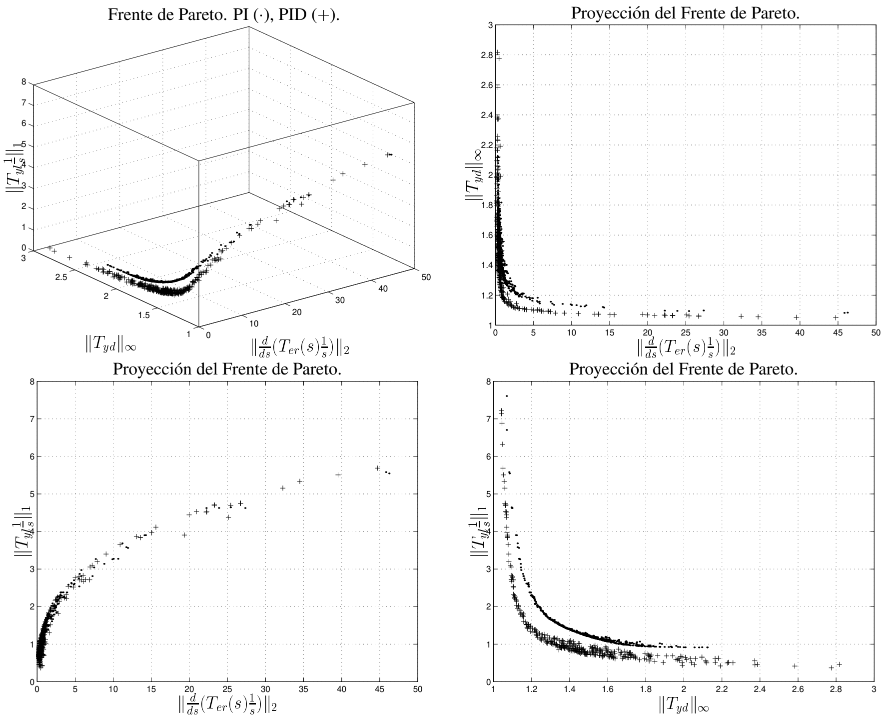


Figura 6.12: Frente de Pareto asociado a la élite de PIs y PIDs para $G_6(s)$.

$[b^{min}, b^{max}]$	$[c^{min}, c^{max}]$	$[K^{min}, K^{min}]$	$[T_i^{min}, T_i^{max}]$	$[T_d^{min}, T_d^{max}]$	$[N^{min}, N^{max}]$	$[T_f^{min}, T_f^{max}]$
$[0, 1]$	$[0, 1]$	$[-20, 20]$	$[0, 20]$	$[0, 20]$	$[1, 100]$	$[1, 100]$

Tabla 6.1: Límites iniciales del espacio de búsqueda.

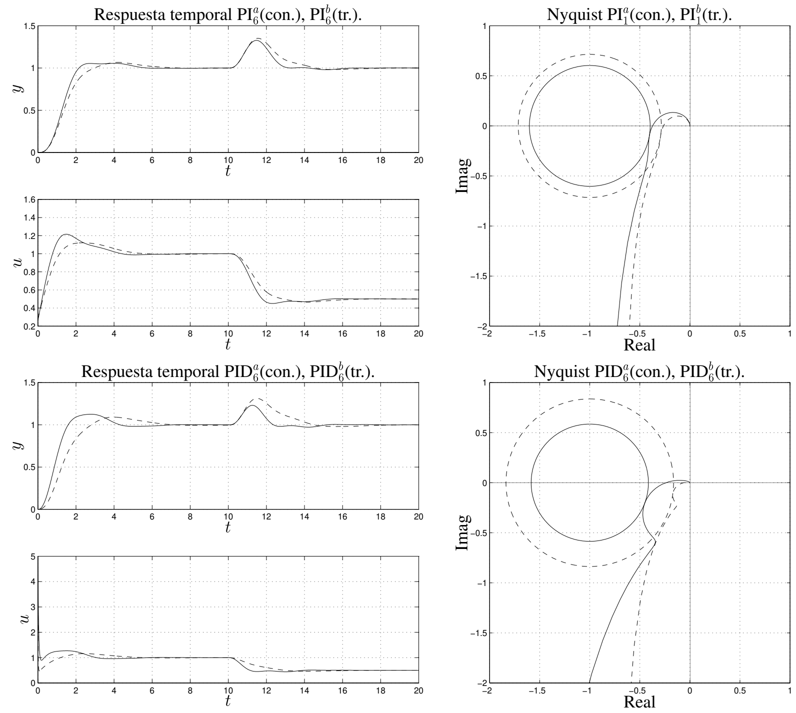


Figura 6.13: Respuesta temporal y diagramas de Nyquist para $G_6(s)$.

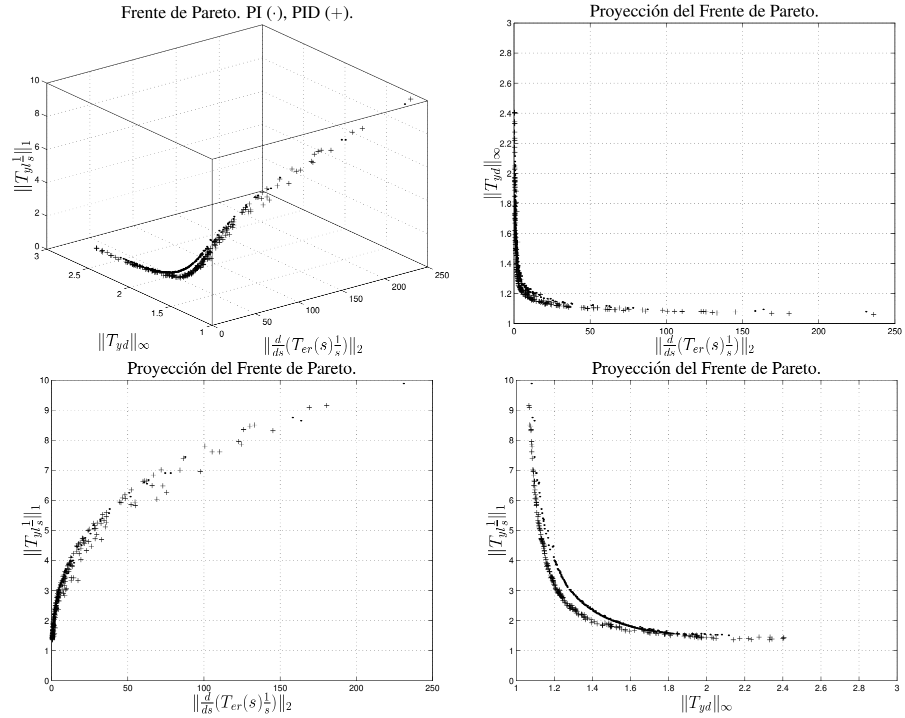


Figura 6.14: Frente de Pareto asociado a la élite de PIs y PIDs para $G_7(s)$.

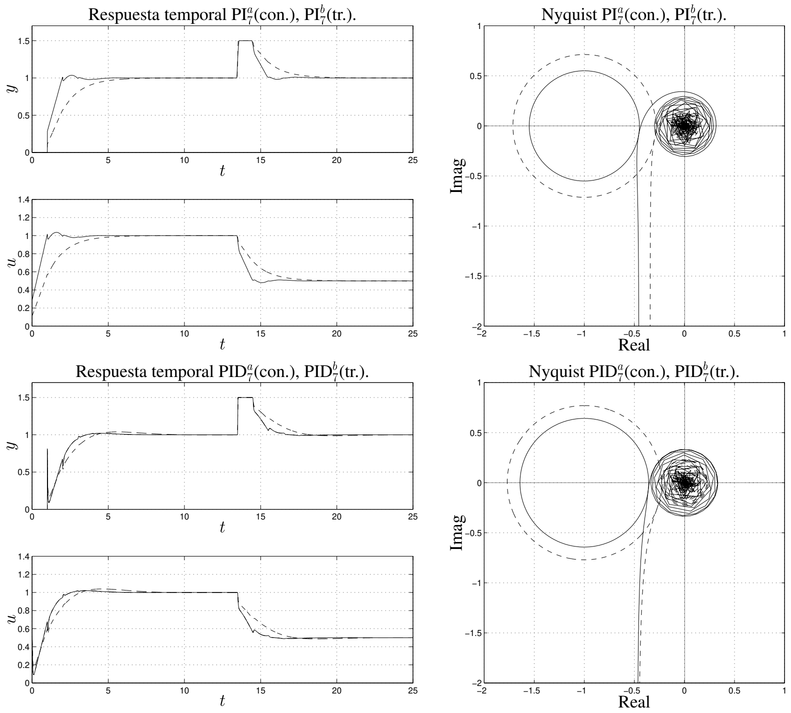


Figura 6.15: Respuesta temporal y diagramas de Nyquist para $G_7(s)$.

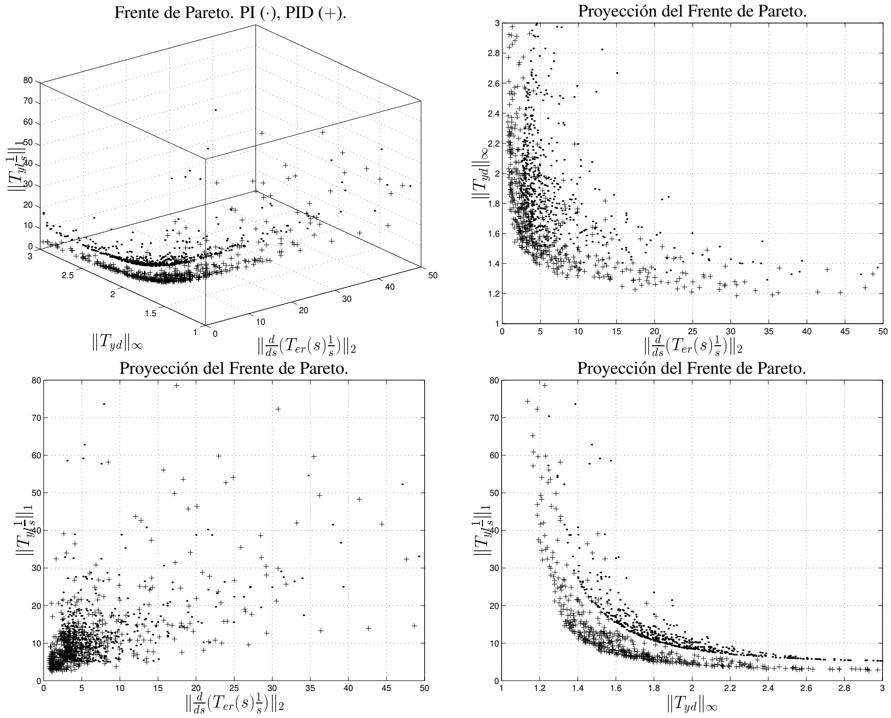


Figura 6.16: Frente de Pareto asociado a la élite de PIs y PIDs para $G_8(s)$.

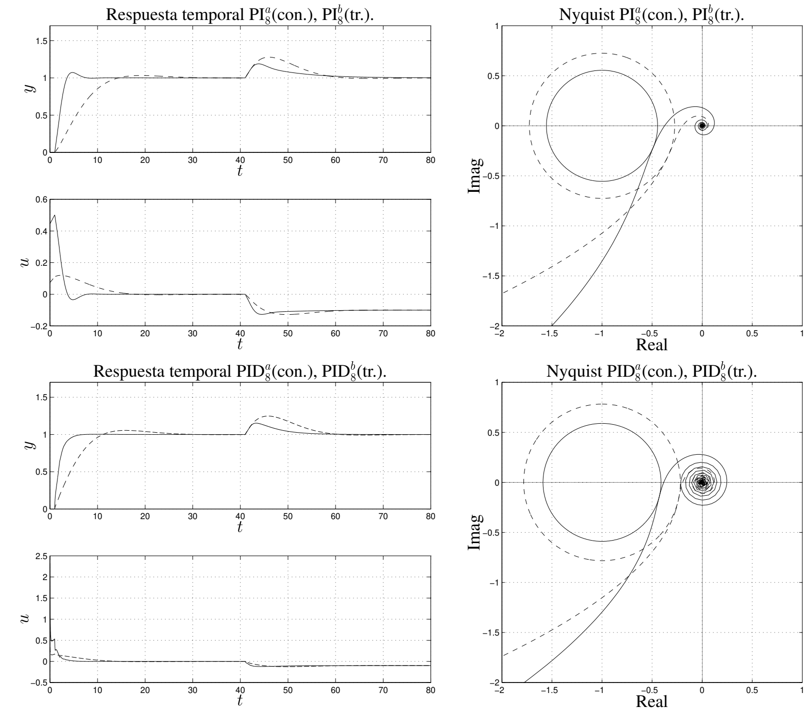


Figura 6.17: Respuesta temporal y diagramas de Nyquist para $G_8(s)$.

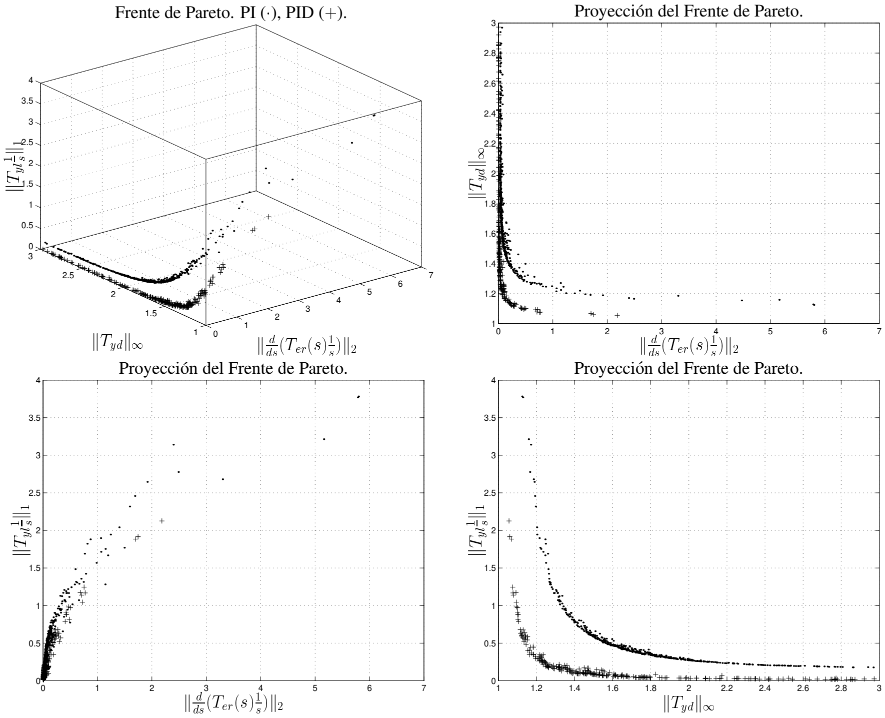


Figura 6.18: Frente de Pareto asociado a la élite de PIs y PIDs para $G_{10}(s)$.

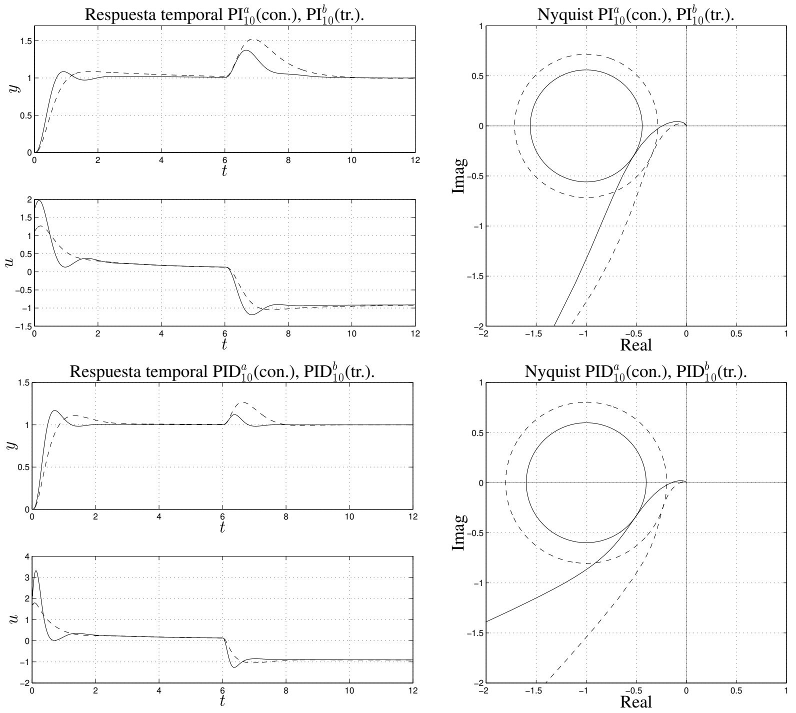


Figura 6.19: Respuesta temporal y diagramas de Nyquist para $G_{10}(s)$.

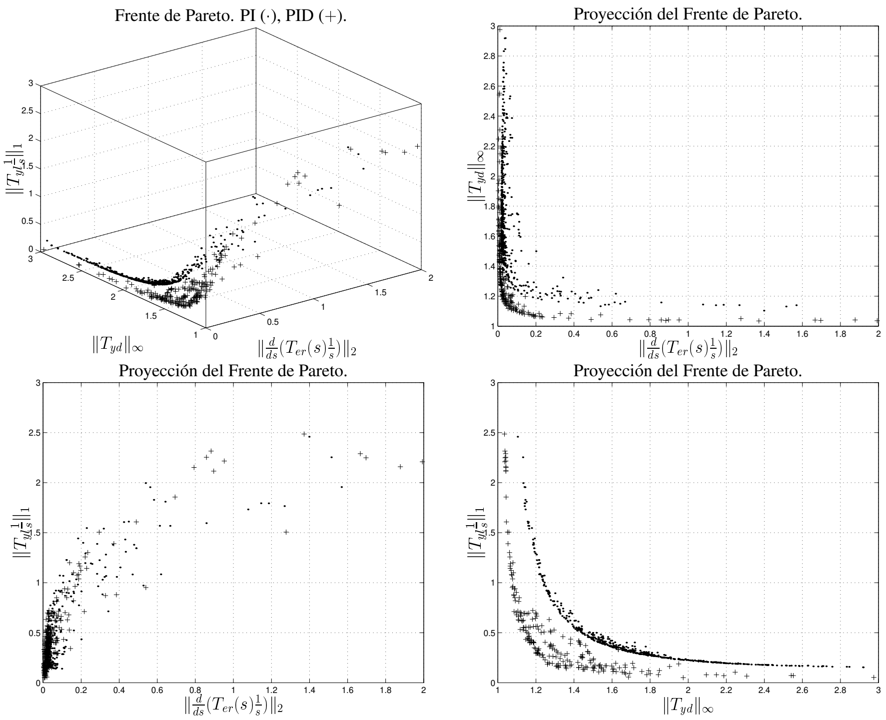


Figura 6.20: Frente de Pareto asociado a la élite de PIs y PIDs para $G_{11}(s)$.

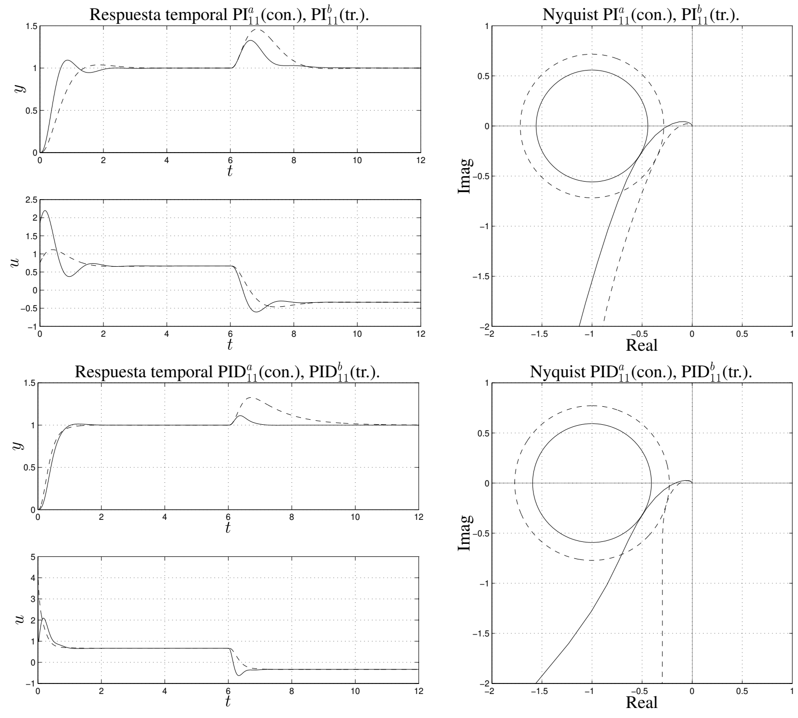


Figura 6.21: Respuesta temporal y diagramas de Nyquist para $G_{11}(s)$.

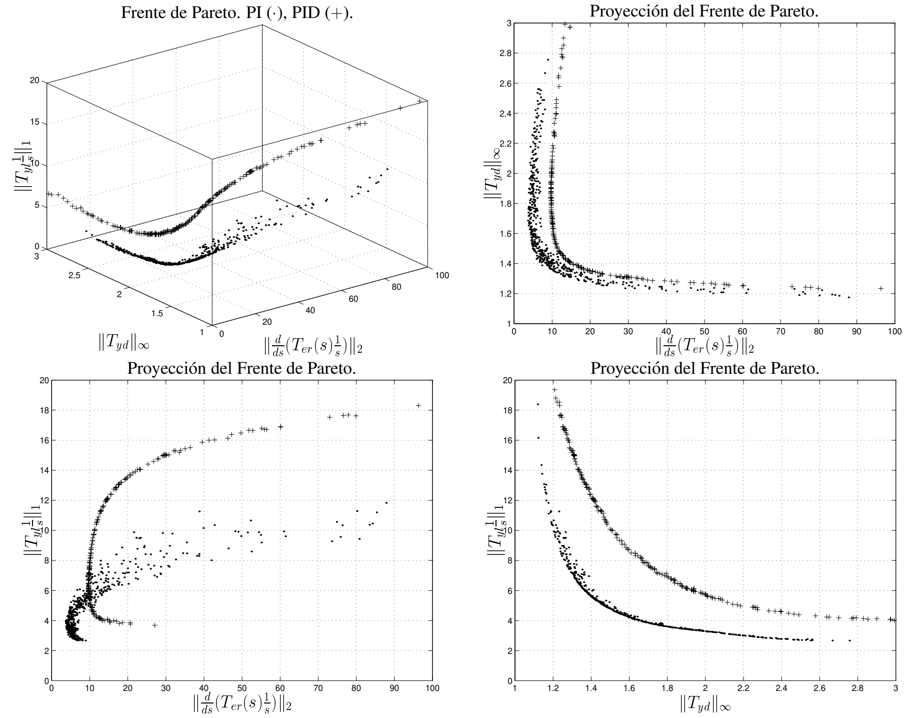


Figura 6.22: Frente de Pareto asociado a la élite de PIs y PIDs para $G_{12}(s)$.

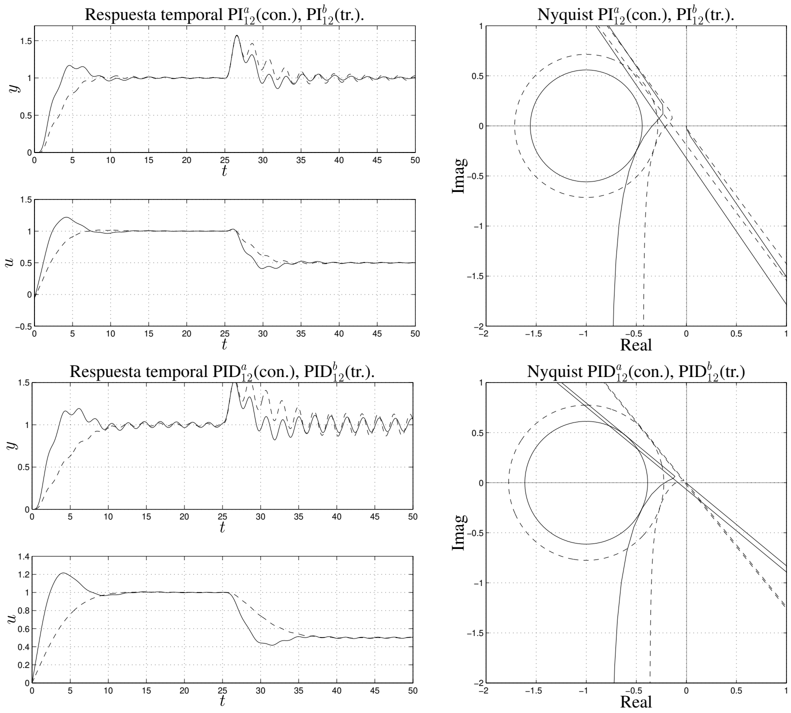


Figura 6.23: Respuesta temporal y diagramas de Nyquist para $G_{12}(s)$.

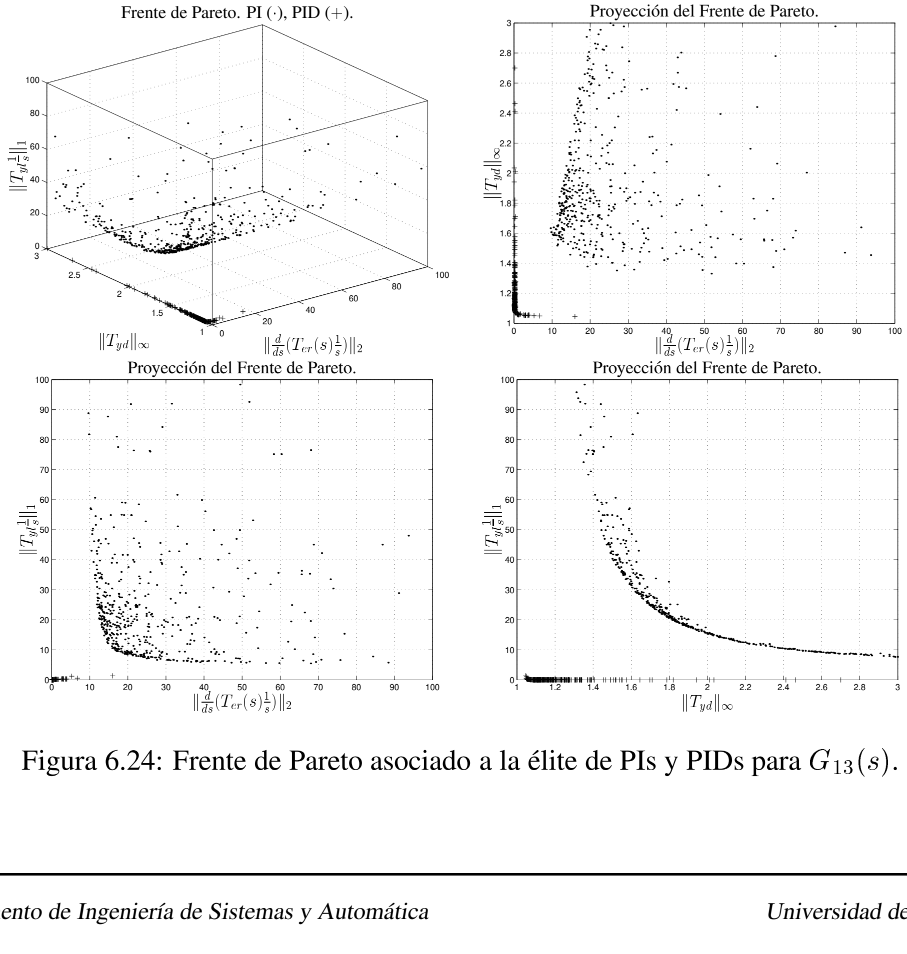


Figura 6.24: Frente de Pareto asociado a la élite de PIs y PIDs para $G_{13}(s)$.

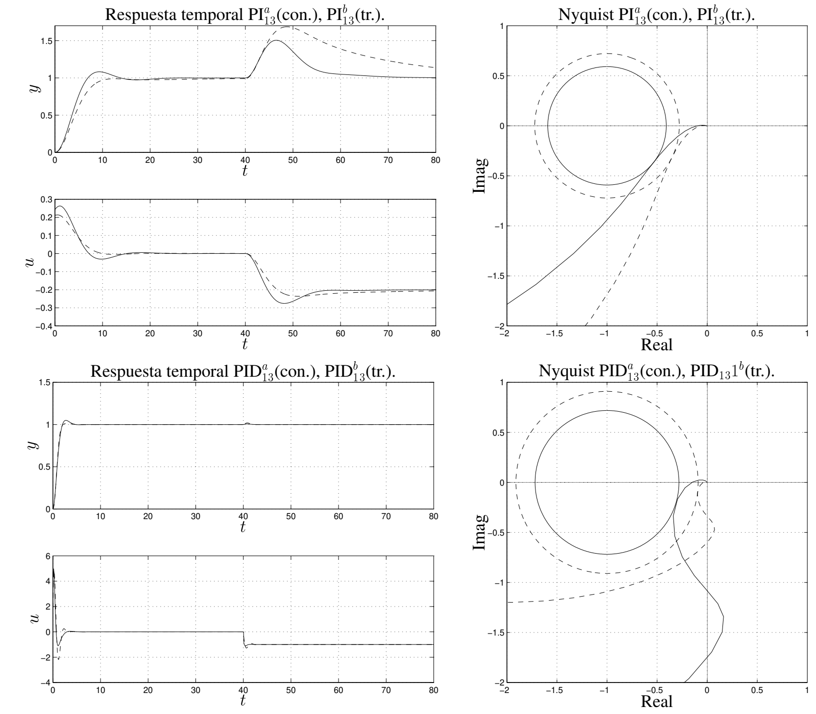


Figura 6.25: Respuesta temporal y diagramas de Nyquist para $G_{13}(s)$.

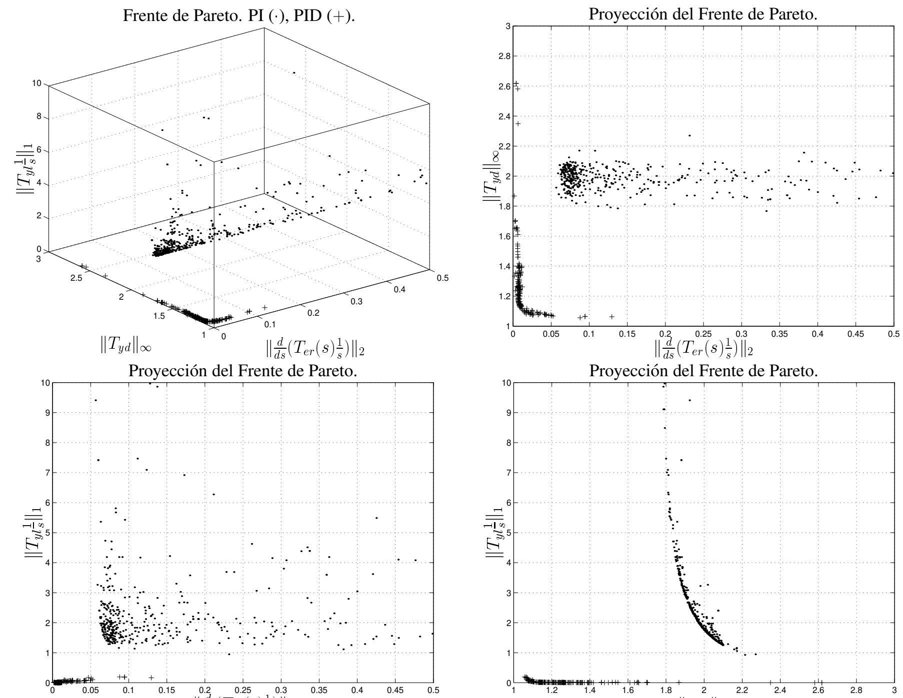


Figura 6.26: Frente de Pareto asociado a la élite de PIs y PIDs para $G_{14}(s)$.

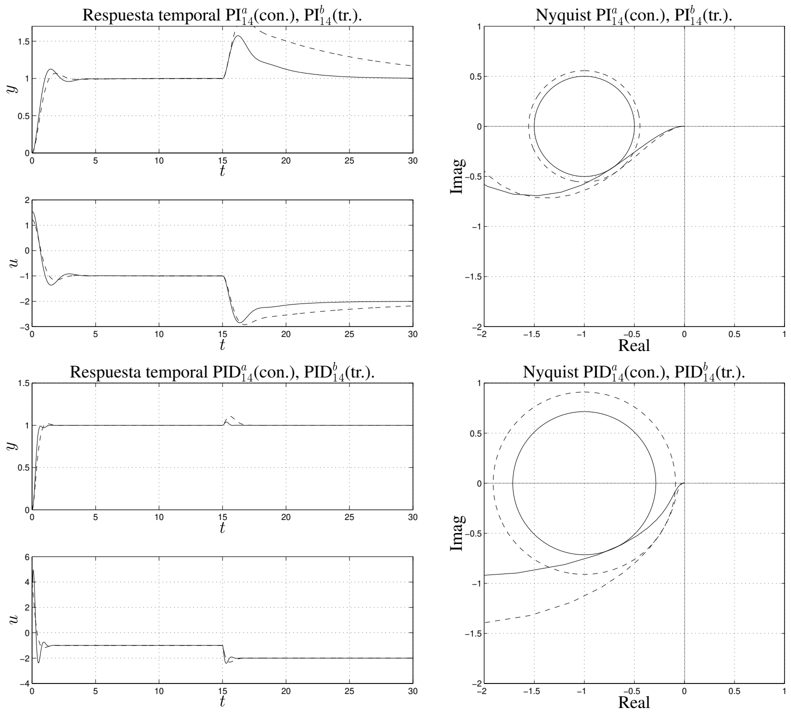


Figura 6.27: Respuesta temporal y diagramas de Nyquist para $G_{14}(s)$.

6.5 Sistemas de ensayo MIMO (dos entradas dos salidas).

En esta sección se generaliza el planteamiento de sintonización de PIDs mostrado en la sección 6.2 a sistemas MIMO (dos entradas, dos salidas). Los sistemas MIMO planteados tienen la forma de la figura 6.3, donde las plantas de lazo directo ($G_{1,1}(s)$ y $G_{2,2}$) serán los sistemas $G_1(s)$ y $G_6(s)$, ver tabla 6.3, y los sistemas cruzados ($G_{1,2}(s)$ y $G_{2,1}$) van a ser $G_2(s)$ y $G_5(s)$, con distinto valor en su ganancia con el fin de modificar la dificultad del problema. Los sistemas propuestos se muestran en la tabla 6.3. Las ecuaciones del sistema en lazo abierto serán

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_{1,1}(s) & G_{1,2}(s) \\ G_{2,1}(s) & G_{2,2}(s) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

Las señales de salida se filtrarán antes de realimentarlas para eliminar el ruido,

$$\begin{bmatrix} y'_1 \\ y'_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{T_{f(1,1)}s+1} & 0 \\ 0 & \frac{1}{T_{f(2,2)}s+1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

Se desea que las señales de salida sigan a una referencia y minimicen el resto de objetivos propuestos en 6.15. Para ello se proponen dos tipos de controladores compuestos de PIDs. El primero consiste en dos controladores PIDs desacoplados, que lo podrán ser de forma directa, cuya señal de control es

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \mathrm{PID}_{1,1} & 0 \\ 0 & \mathrm{PID}_{2,2} \end{bmatrix} \begin{bmatrix} e_1 \\ y'_1 \\ \hline e_2 \\ y'_2 \end{bmatrix},$$

y de forma inversa, cuya señal de control es

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 & \mathrm{PID}_{1,2} \\ \mathrm{PID}_{2,1} & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ y'_1 \\ \hline e_2 \\ y'_2 \end{bmatrix}.$$

En ambos casos la estructura de los controladores PIDs es la dada en (6.6), que corresponden a la de un PID estándar ISO, ver apéndice E. Los métodos tradicionales usan este tipo de estructura, sintonizando cada PID por separado, como si los sistemas no estuvieran acoplados. El segundo tipo de controlador propuesto es el formado de cuatro PIDs acoplados,

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \mathrm{PID}_{1,1} & \mathrm{PID}_{1,2} \\ \mathrm{PID}_{2,1} & \mathrm{PID}_{2,2} \end{bmatrix} \begin{bmatrix} e_1 \\ y_1' \\ \hline e_2 \\ y_2' \end{bmatrix}.$$

La eficacia de un controlador u otro la dará el grado de acoplamiento que la planta tenga, como se vera en la evaluación de resultados.

En [OR94] se mide la dificultad para controlar un sistema MIMO a través de la realimentación entre una entrada j y una salida i elegida. La herramienta usada es un ratio, llamado ganancia relativa entre ambas, que compara la ganancia de lazo abierto con la ganancia con todos los bucles en realimentación, λ_{ij}. Tras una serie de cálculos se llega a la conclusión de que si la matriz de ganancias viene dada por $\lim_{s\to 0} G(s) = K$, los términos de la matriz de ganancia relativa pueden ser expresados como,

$$\lambda_{ij} = [(K^{-1})^T]_{ij} \cdot [K]_{ij}.$$

Los valores de λ_{ij} son una medida de la bondad en un determinado bucle entre la entrada j y salida i:

- ($\lambda_{ij} = 0$). La entrada no influye en la salida.
- ($0 < \lambda_{ij} < 1$). La ganancia directa es menor que la ganancia en realimentación, lo que implica que los otros lazos ayudan en la realimentación. La realimentación por esa entrada, salida puede ser efectiva, sobre todo si $\lambda_{ij} \leq 0.5$.
- ($\lambda_{ij} > 1$). La ganancia directa es mayor que la ganancia en realimentación, lo que implican que los otros lazos no ayudan a la realimentación entre esa entrada y salida. Por ello se puede desestabilizar el sistema si se abren los otros lazos.
- ($\lambda_{ij} < 0$). La ganancia en lazo abierto es de signo opuesto a la de lazo cerrado, lo que implica que los otros lazos en realimentación tienen un efecto contrario al lazo directo y además son superiores a él. Es el peor caso posible.

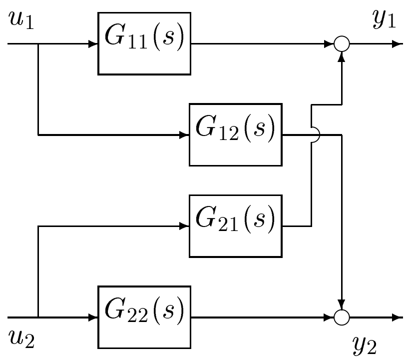


Figura 6.28: Planta MIMO con cuatro subsistemas acoplados.

Los sistemas planteados en la tabla 6.3 tienen las siguientes matrices de ganancia relativa,

$$\begin{array}{ll} \Lambda_{15} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \Lambda_{16} = \begin{bmatrix} 1.333 & -0.333 \\ -0.333 & 1.333 \end{bmatrix} \\ \Lambda_{17} = \begin{bmatrix} 5.26 & -4.26 \\ -4.26 & 5.26 \end{bmatrix} & \Lambda_{18} = \begin{bmatrix} -2.27 & 3.27 \\ 3.27 & -2.27 \end{bmatrix} \end{array} \tag{6.17}$$

El sistema $G_{15}(s)$ es el caso perfecto para emplear dos PIDs desacoplados directos ya que el sistema es diagonal. Los $G_{16}(s)$ y $G_{17}(s)$ va a ser abordables por dos PIDs desacoplados directos o por cuatro acoplados. Dentro de ellos el sistema $G_{17}(s)$ es mucho más sencillo de controlar por tener una ganancia relativa menor. El sistema $G_{18}(s)$ va a ser controlable por dos PID desacoplados inversos o por cuatro PIDs acoplados. La aplicación de dos PIDs desacoplados directos es una mala opción por ser negativa la diagonal de la matriz de ganancias relativas.

El problema se va a resolver con el uso del algoritmo MRCD aplicado a la optimización multiobjetivo propuesta en (6.16). Pese a ser un caso MIMO el planteamiento es el mismo sin más que generalizar las normas a los correspondientes sistemas MIMO. Los parámetros que se deben encontrar son los dos parámetros de los filtros y los parámetros de los PIDs, 12 en el caso de PIDs desacoplados y 24 en el caso de PIDs acoplados.

6.5.1 Evaluación de los resultados.

Los problemas propuestos son los siguientes:

1. Optimizar los objetivos (6.16) del sistema $G_{15}(s)$ para dos PIDs desacoplados directos,

figuras 6.29 y 6.30.

2. Optimizar los objetivos (6.16) del sistema $G_{16}(s)$ para dos PIDs desacoplados directos, figuras 6.31 y 6.32, y para cuatro PIDs acoplados, figuras 6.33 y 6.34.

3. Optimizar los objetivos (6.16) del sistema $G_{17}(s)$ para dos PIDs desacoplados directos, figuras 6.35 y 6.36, y para cuatro PIDs acoplados, figuras 6.37 y 6.38.

4. Optimizar los objetivos (6.16) del sistema $G_{18}(s)$ para dos PIDs desacoplados de forma directa, figura 6.39, dos PIDs desacoplados de forma inversa, figuras 6.40 y 6.41, y cuatro PIDs acoplados, figuras 6.42, 6.43. En el caso de PIDs desacoplados de forma directa no se ha encontrado controladores razonablemente buenos.

Las figuras que representan frentes de Pareto superponen el encontrado para el sistema $G_{15}(s)$, $(\cdot)$, con el del sistema correspondiente, $(+)$. Con ello se puede ver la diferencia entre el caso más favorable para el control y el caso correspondiente. En cada problema propuesto se ha elegido dos controladores de diferentes zonas del frente de Pareto para mostrar su respuesta temporal y su respuesta en frecuencia. Para la respuesta temporal se excita el sistema en lazo cerrado con una entrada escalón unitaria en $r(t)$ y a mitad de la simulación una perturbación escalón de magnitud 0.5 en la acción de control, $l(t)$. La respuesta de la salida 1 se dibuja en línea continua y la de la salida 2 en trazos. Para la respuesta frecuencial se obtienen los valores singulares en las distintas frecuencias para la sensibilidad (línea continua), y para la sensibilidad complementaria (línea a trazos) del sistema en lazo cerrado. Los controladores elegidos tienen una sensibilidad en torno a 1.4 y a 1.8. El valor de los parámetros de los controladores elegidos y el de sus objetivos correspondientes se muestran en la tabla 6.4.

El problema 1 muestra unos resultados, figuras 6.29 y 6.30, comparables a los obtenidos en los respectivos sistemas SISO, lo que implica que el planteamiento del problema (6.16) sigue siendo valido en sistemas MIMO.

El problema 2 muestra resultados similares para dos PIDs desacoplados y para cuatro PIDs acoplados, siendo estos resultados muy parecidos a los del problema 1, figuras 6.31, 6.33 y 6.32, 6.34. Este hecho se podía suponer por no ser la ganancia relativa muy grande, ver (6.17). La realimentación cruzada tiene ganancia relativa negativa, por lo que va a ser mucho menos eficiente.

El problema 3 muestra como tanto los resultados para los dos PIDs desacoplados como los obtenidos para cuatro PIDs acoplados son peores que los del problema 1, ver figuras 6.35, 6.37 y 6.36, 6.36. Las ganancias relativas directas del sistema son ahora mucho mayores, y las cruzadas negativas. Con ello se puede prever que será más difícil la realimentación en el lazo directo, y que el efecto de la realimentación cruzada en el PID acoplado no servirá de mucho.

El problema 4 tiene una difícil solución con PIDs desacoplados de forma directa, ya que las ganancias relativas directas de la planta son negativas (6.17). Los PIDs asociados al frente de Pareto en este caso son muy lentos o con una gran sensibilidad, ver figura 6.40. Este problema se puede resolver por PIDs desacoplados de forma inversa y con PIDs acoplados, siendo sus resultados similares, ver figuras 6.40 y 6.42 y 6.41, 6.43. La explicación es la misma del problema 3 pero en este caso la realimentación mejor es la cruzada.

La conclusión final es que la realimentación con PIDs desacoplados tiene que realizarse entre las entradas y salidas más favorables, siendo muy difícil el control del sistema si se realiza entre entradas y salidas con ganancia relativa negativa. La realimentación con PIDs acoplados funciona bien en cualquier caso, pero no consigue sustanciales ventajas frente a la realimentación desacoplada que ha elegido bien los lazos de realimentación.

6.6 Conclusiones

La sintonización de los parámetros de un PID requiere una serie de especificaciones que están en conflicto entre sí, por ejemplo rapidez de respuesta, robustez, atenuación de ruidos. Se trata, por tanto, de un problema de diseño multiobjetivo. Los controladores PID tienen una estructura fija, función de una serie de parámetros, lo que hace que la sintonización de PIDs se pueda formular como una optimización multiobjetivo respecto de dichos parámetros, ver ecuación (2.5). En este capítulo se ha planteado un problema de optimización que intenta reducir el tiempo de respuesta del sistema, la sensibilidad del mismo a perturbaciones en la planta, minimizar el efecto de ruido en la señal de control y evitar la saturación de los actuadores. Para ello se ha formulado un problema de minimización multiobjetivo de otras tantas normas que miden dichos efectos.

Para comprobar la bondad de la formulación del problema planteado se han diseñado controladores PI y PID para trece sistemas de ensayo SISO con diferentes propiedades. Los resultados obtenidos han sido satisfactorios y en el caso de los PIs han sido comparados con los obtenidos

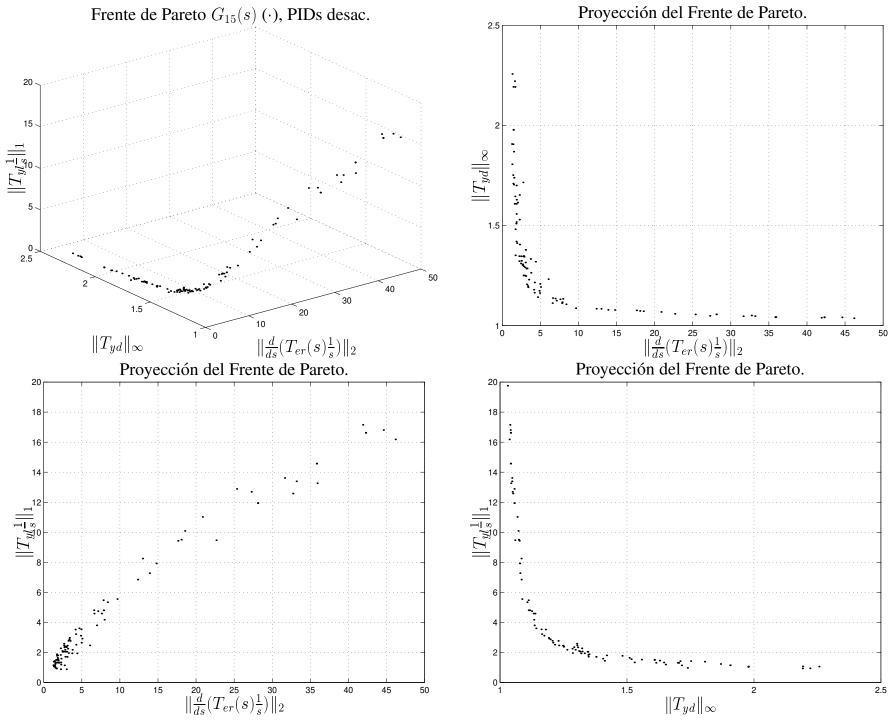


Figura 6.29: Frente de Pareto asociado a $G_{15}(s)$, PIDs desacoplados.

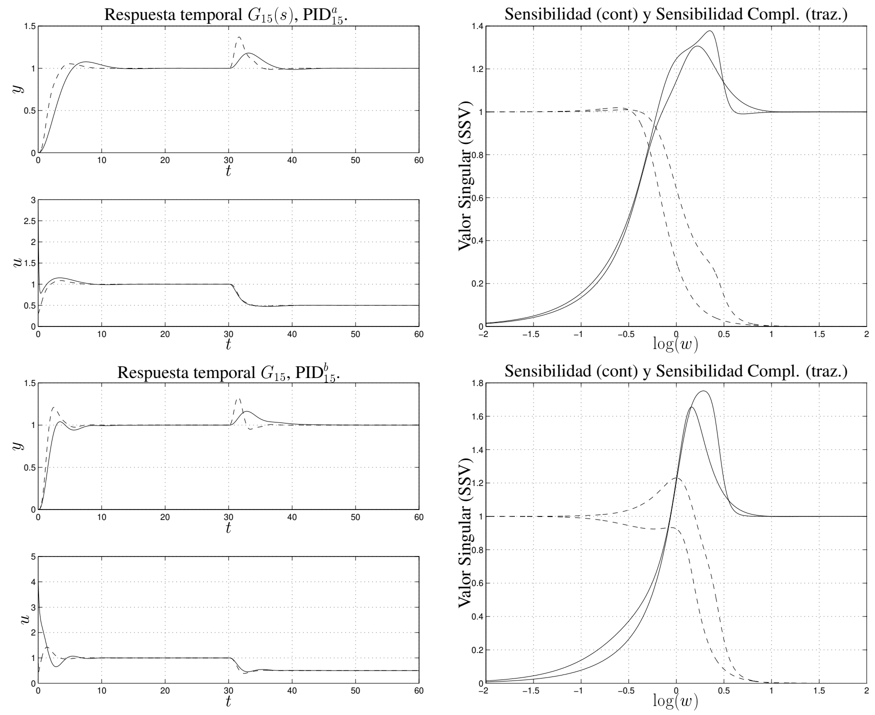


Figura 6.30: Respuesta temporal y sensibilidades de $G_{15}(s)$, PIDs desacoplados.

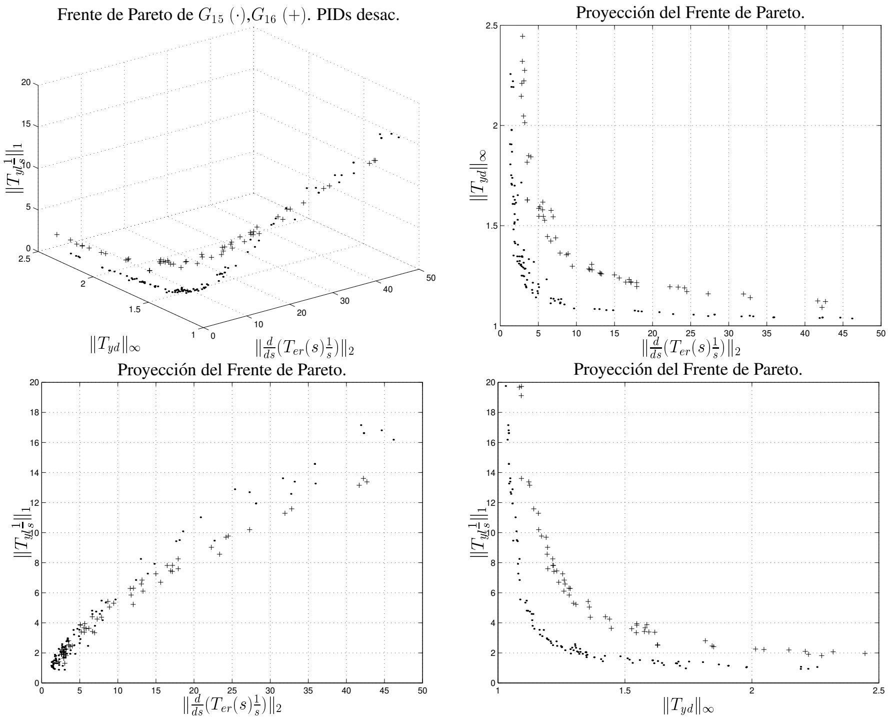


Figura 6.31: Frente de Pareto asociado a $G_{16}(s)$, PIDs desacoplados.

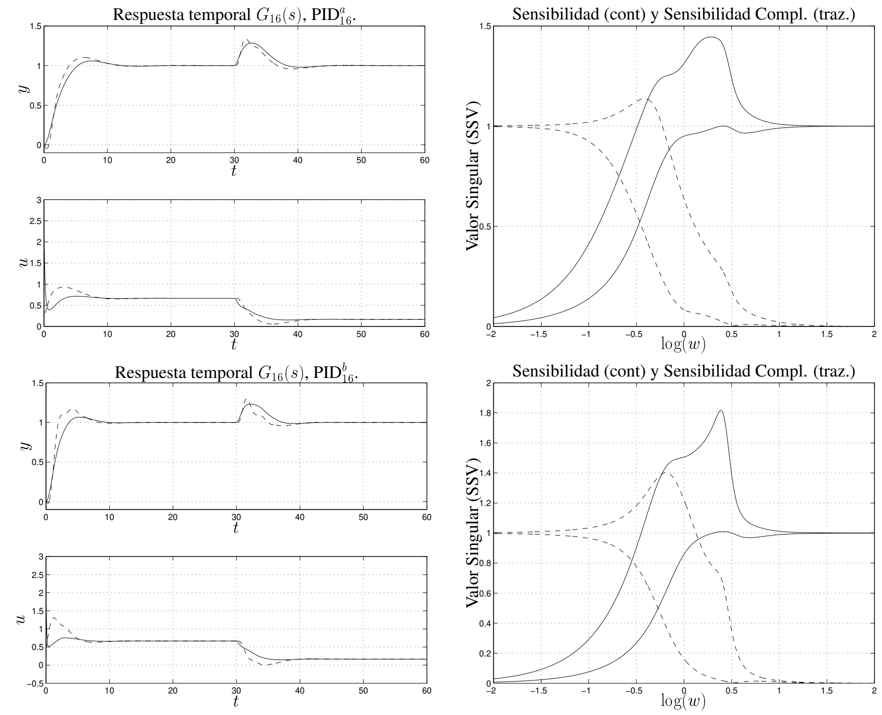


Figura 6.32: Respuesta temporal y sensibilidades de $G_{16}(s)$, PIDs desacoplados.

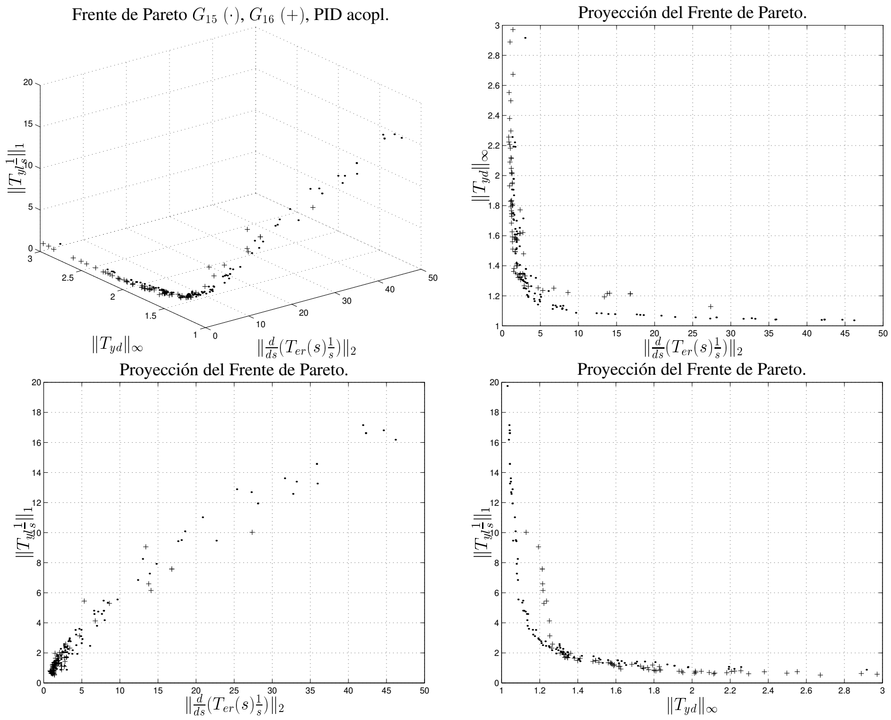


Figura 6.33: Frente de Pareto asociado a $G_{16}(s)$, PIDs acoplados.

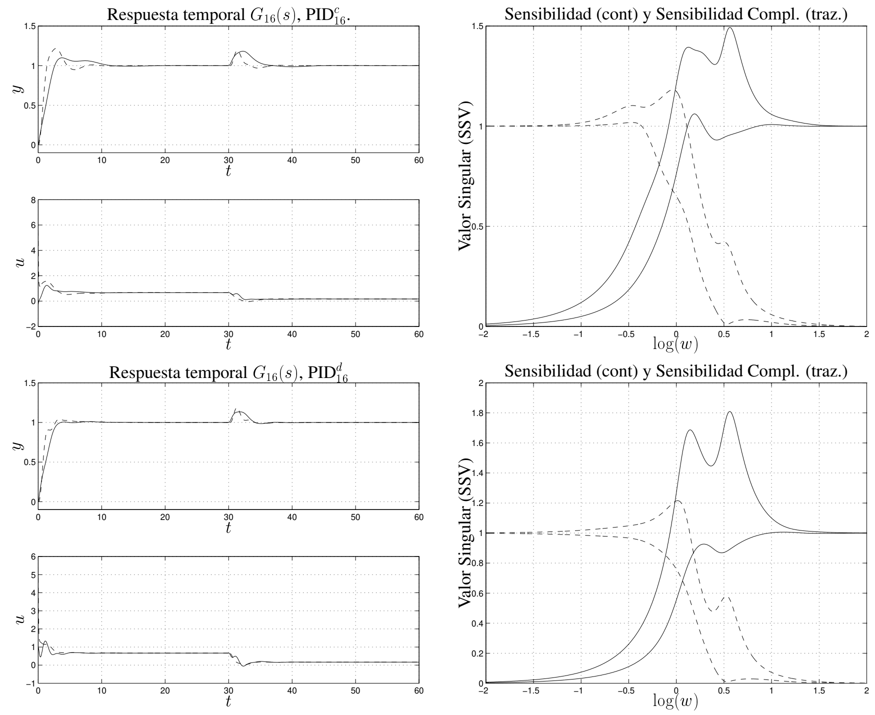


Figura 6.34: Respuesta temporal y sensibilidades de $G_{16}(s)$, PIDs acoplados.

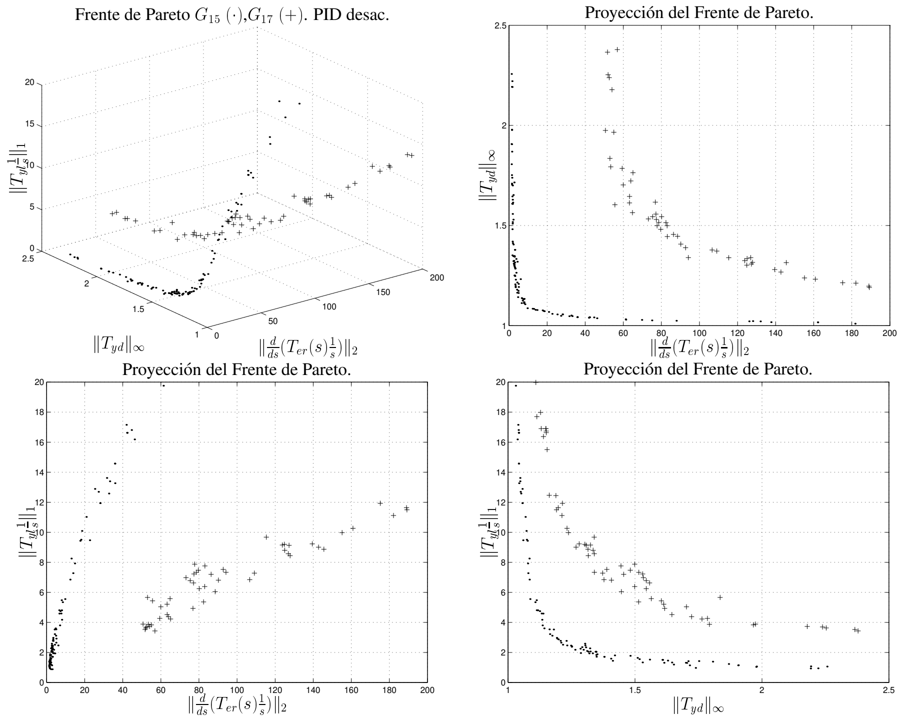


Figura 6.35: Frente de Pareto asociado a $G_{17}(s)$, PIDs directo desacoplados.

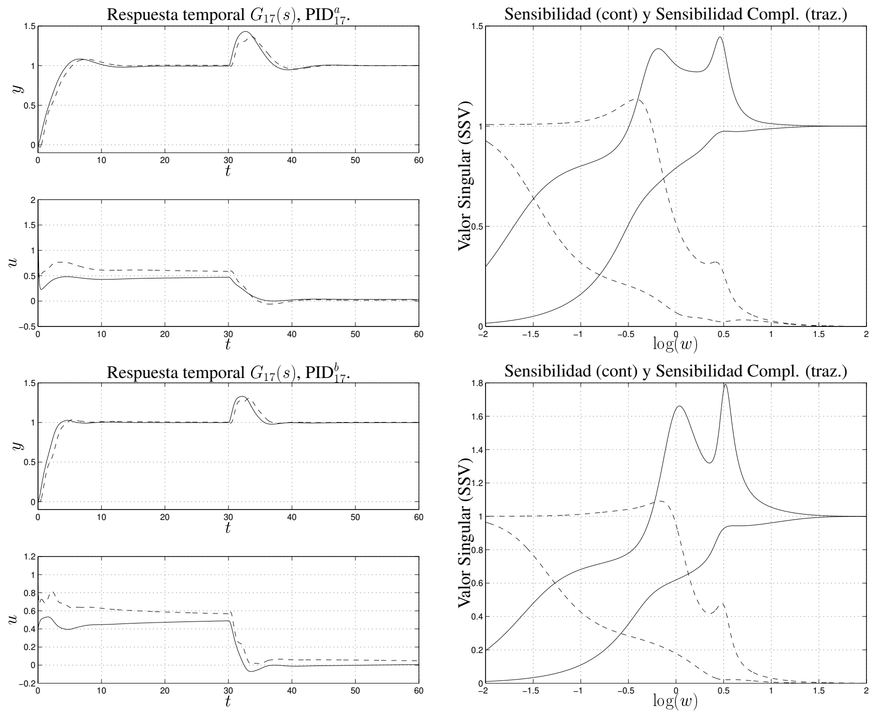


Figura 6.36: Respuesta temporal y sensibilidades de $G_{17}(s)$, PIDs directo desacoplados.

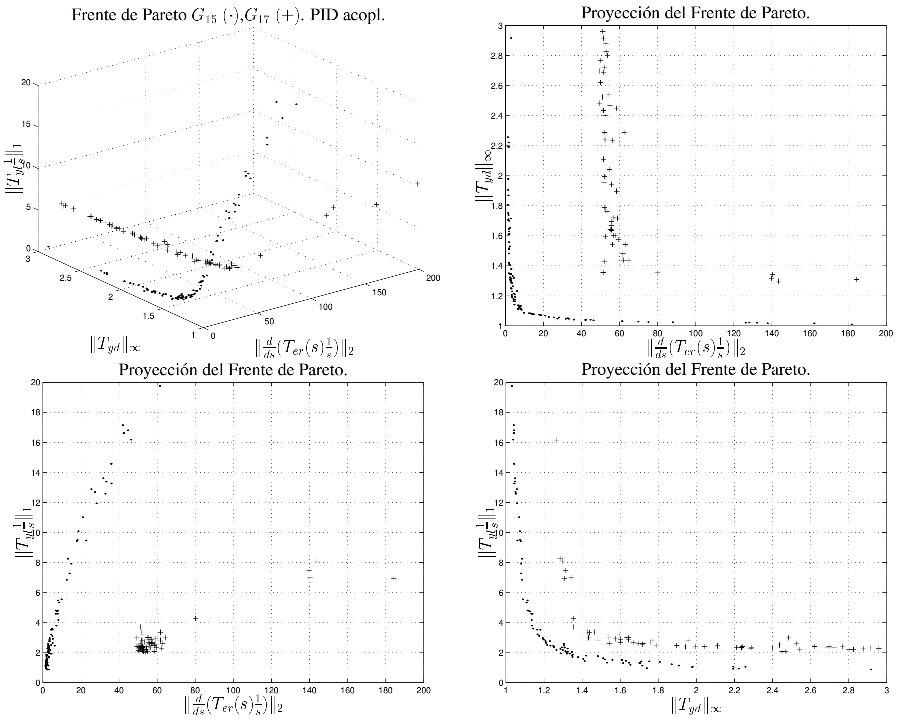


Figura 6.37: Frente de Pareto asociado a $G_{17}(s)$, PIDs acoplados.

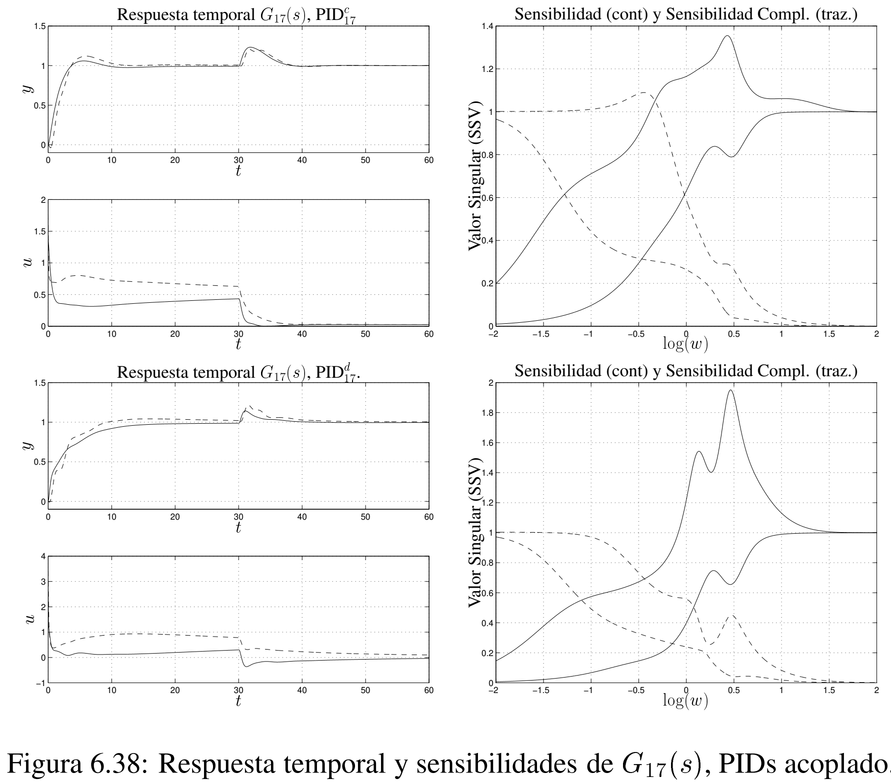


Figura 6.38: Respuesta temporal y sensibilidades de $G_{17}(s)$, PIDs acoplado.

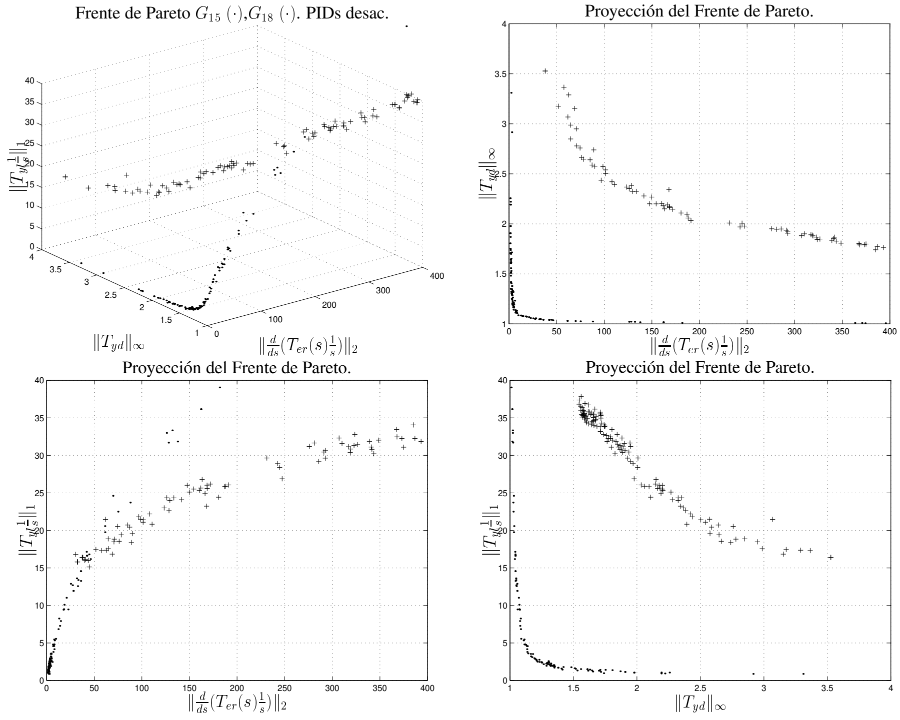


Figura 6.39: Frente de Pareto asociado a $G_{18}(s)$, PIDs desacoplados directa.

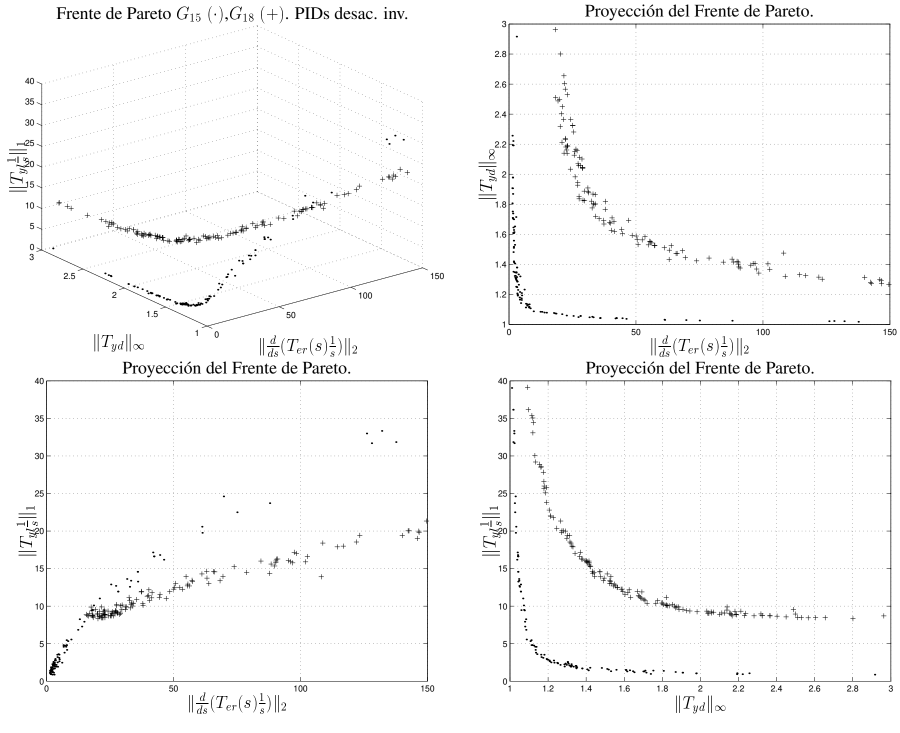


Figura 6.40: Frente de Pareto asociado a $G_{18}(s)$, PIDs desacoplados inverso.

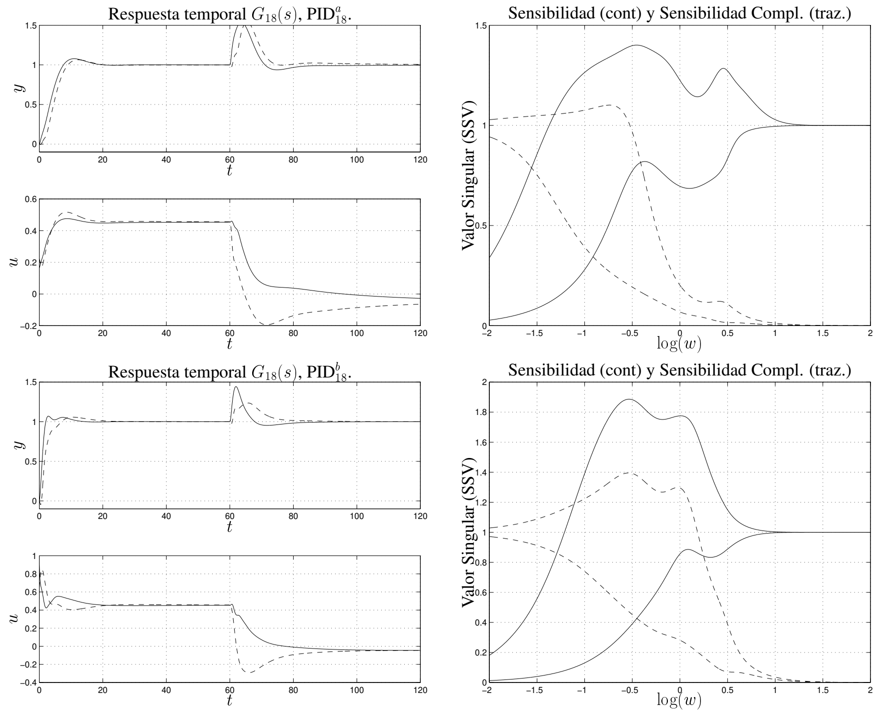


Figura 6.41: Respuesta temporal y sensibilidades de $G_{18}(s)$, PIDs desacoplados inverso.

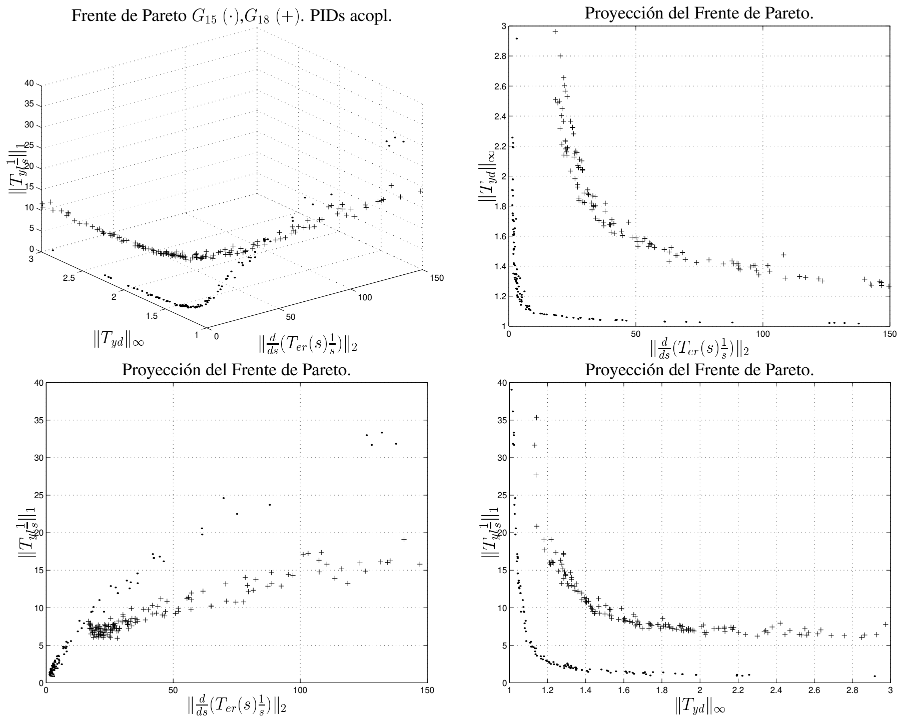


Figura 6.42: Frente de Pareto asociado a $G_{18}(s)$, PIDs acoplados.

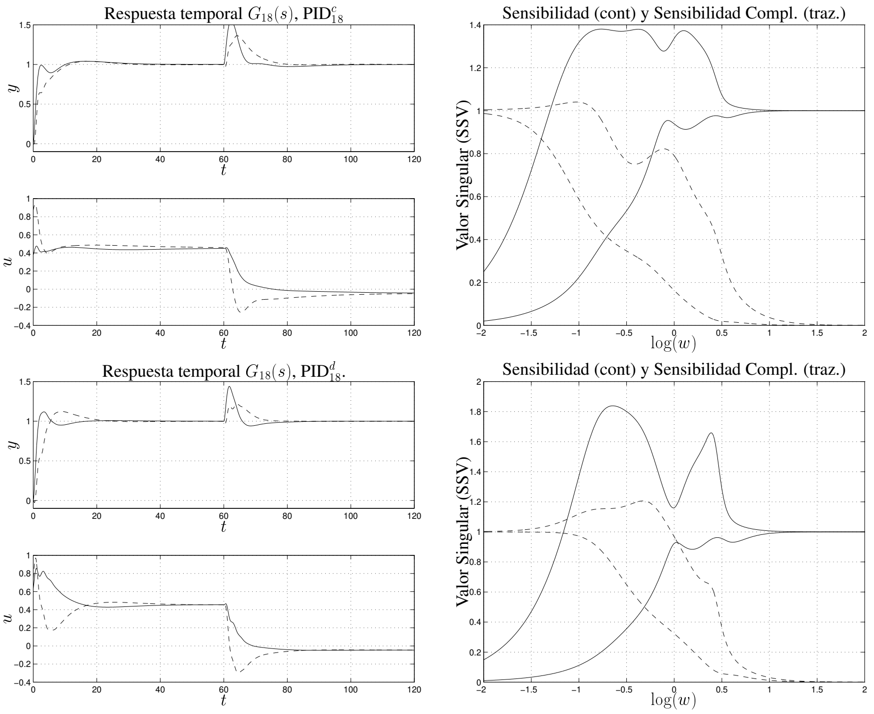


Figura 6.43: Respuesta temporal y sensibilidades de $G_{18}(s)$, PIDs acoplados.

en el artículo [APH98], donde se plantea un problema equivalente. Se ha comparado las diferencias entre los resultados obtenidos con los PIs y PIDs, dando una explicación a las mismas.

La misma formulación del problema ha sido empleada en la sintonización de los parámetros de dos PIDs desacoplados y cuatro PIDs acoplados usados en el control de cuatro plantas MIMO, dos entradas dos salidas. Dichas plantas han sido modeladas con diferente grado de acoplamiento, lo que influye de forma directa en la dificultad del problema de control. Los resultados son muy buenos, viendo como el algoritmo MRCD consigue la convergencia de los 24 parámetros de los cuatro PIDs acoplados.

En suma, el planteamiento de la sintonización de PIDs como un problema de optimización multiobjetivo tiene buenos resultados tanto en sistemas SISO como en sistemas MIMO. Con el algoritmo MRCD se consigue sintonizar los parámetros de dos PID desacoplados o de cuatro PIDs acoplados sin excesivo problema, lo que supone una ventaja sustancial respecto a la sintonización de cada PID de una planta MIMO de forma independiente.

$K(s)$	b	c	K	T_i	T_d	$1/N$	T_f	$\|\frac{d}{ds}\frac{T_{er}(s)}{s}\|_2$	$\|T_{yd}(s)\|_\infty$	$\|\frac{T_{yl}}{s}\|_1$
PI_1^a	0.8383		1.1943	2.2055			0.0102	3.7175	1.7665	1.8995
PI_1^b	0.7394		0.6583	2.0255			0.0108	9.4943	1.4059	3.0773
PID_1^a	0.7769	0.0718	3.3809	2.4607	0.6564	0.1934	0.0101	0.8254	1.7174	0.7288
PID_1^b	0.9373	0.0587	1.8763	3.1646	0.6829	0.0411	0.0364	1.6314	1.3060	1.6866
PI_2^a	0.6901		5.1178	0.6663			0.0119	0.0160	1.8324	0.1302
PI_2^b	0.6652		2.8469	0.8019			0.0109	0.0336	1.4341	0.2817
PID_2^a	0.4615	0.0009	10.3610	0.5249	0.1315	0.0469	0.0112	0.0097	1.1981	0.0513
PID_2^b	0.5955	0.0100	6.5212	0.6934	0.1171	0.0617	0.0111	0.0129	1.1364	0.1063
PI_3^a	0.5455		0.3105	7.5732			0.1660	$4.62e3$	1.8	26.7
PI_3^b	0.7303		0.1634	6.1199			0.0585	$1.12e4$	1.4	37
PID_3^a	0.8523	0.1180	0.2580	7.7381	4.8699	0.0872	1.2807	6346	1.3	32.3
PID_3^b	0.2117	0.0821	0.2118	6.2832	5.2553	0.0535	0.0447	9126	1.3	32.4
PI_4^a	0.4774		0.2877	9.495			0.026	36.115	1.794	32.99
PI_4^b	0.8282		0.2144	30.19			0.018	33.925	1.391	133.63
PID_4^a	0.5448	0.0353	1.1218	5.4549	1.1682	0.0318	0.0280	4.4305	1.5936	4.8733
PID_4^b	0.4525	0.2843	0.4919	8.9132	1.2975	0.0522	0.0103	27.0978	1.2759	18.1919
PI_5^a	0.9936		0.3191	2.0429			0.0492	52.32	1.79	7.53
PI_5^b	0.8088		0.1737	1.7492			0.0167	212.65	1.38	11.14
PID_5^a	0.951	0.334	0.49	2.354	0.763	0.0452	0.0251	27.158	1.79	6.21
PID_5^b	0.905	0.421	0.223	2.032	1.0366	0.0617	0.0183	158.92	1.28	10.30
PI_6^a	0.499		0.5185	0.4676			0.0124	0.5802	1.657	1.01
PI_6^b	0.8491		0.3371	0.4151			0.0156	0.8036	1.4	1.39
PID_6^a	0.5196	0.4758	1.0084	0.4634	0.3867	0.1087	0.0126	0.3572	1.7079	0.6441
PID_6^b	0.9770	0.2074	0.4052	0.4594	0.3994	0.0369	0.0151	0.8229	1.1951	1.3399
PI_7^a	0.9713		0.2924	0.3992			0.0180	0.5927	1.7971	1.5928
PI_7^b	0.8930		0.1246	0.2703			0.0146	2.0893	1.3998	2.3787
PID_7^a	0.8390	0.9697	0.4566	0.6132	0.3088	0.1169	0.4079	0.8731	1.5434	1.7895
PID_7^b	0.2396	0.1126	0.3933	0.7866	0.4078	0.0739	0.7143	4.4498	1.2950	2.3726
PI_8^a	0.8005		0.5549	9.8212			0.0132	2.4520	1.8004	17.7327
PI_8^b	0.2623		0.2814	7.8167			0.0218	54.7359	1.3787	29.3481
PID_8^a	0.6871	0.6602	0.6340	6.5789	0.3087	0.2229	0.0170	1.7747	1.6899	10.4258
PID_8^b	0.4265	0.2287	0.3092	7.3817	0.3108	0.3386	0.0262	31.8128	1.2778	25.6925
PI_{10}^a	0.7111		2.4238	0.8860			0.0118	0.0368	1.7887	0.3765
PI_{10}^b	0.8281		1.3415	0.9892			0.0123	0.1892	1.3984	0.7979
PID_{10}^a	0.2542	0.0027	7.1732	0.3066	0.1241	0.0183	0.0142	0.0132	1.6701	0.0575
PID_{10}^b	0.5892	0.0119	2.7346	0.6364	0.1366	0.0290	0.0133	0.0479	1.2428	0.2604
PI_{11}^a	0.7445		2.4951	0.7247			0.0112	0.0182	1.7913	0.2905
PI_{11}^b	0.6296		1.1976	0.6603			0.0114	0.0672	1.3956	0.5648
PID_{11}^a	0.0640	0.0194	7.8141	0.4316	0.1182	0.0539	0.0103	0.0201	1.6863	0.0567
PID_{11}^b	0.9610	0.6094	2.0839	1.0986	0.0936	0.5328	0.0106	0.0128	1.2966	0.5272
PI_{12}^a	0.2783		-0.2378	-0.5			0.1066	4.7026	1.7894	3.6337
PI_{12}^b	0.3543		-0.1556	-0.5962			0.0659	9.9383	1.4003	5.5974
PID_{12}^a	0.6448	0.8819	0.0012	0.0024	30.6487	0.4769	0.5596	10.3564	1.6310	8.4888
PID_{12}^b	0.9182	0.8803	0.0006	0.0030	10.1999	0.6083	0.9626	35.0459	1.2909	15.4336
PI_{13}^a	0.6403		0.3788	9.6880			0.0106	14.6876	1.6917	25.5709
PI_{13}^b	0.7754		0.2717	23.4224			0.0156	29.0102	1.3868	84.2284
PID_{13}^a	0.0181	0.0006	101.6761	0.8932	0.4673	0.0192	0.0102	0.2226	1.3919	0.0097
PID_{13}^b	0.0354	0.0001	41.7934	0.9257	0.3914	0.0107	0.0110	0.1769	1.0991	0.0228
PI_{14}^a	0.5387		2.8078	4.9846			0.0100	0.0790	1.9983	1.7752
PI_{14}^b	0.5249		2.2992	17.1710			0.0110	0.1115	1.7992	7.4658
PID_{14}^a	0.0985	0.0027	25.0394	0.3596	0.0868	0.0276	0.0102	0.0063	1.3997	0.0148
PID_{14}^b	0.4270	0.0006	8.8515	0.8546	0.1565	0.0231	0.0103	0.0152	1.0984	0.0966

Tabla 6.2: Parámetros y vectores objetivos de los controladores seleccionados.

Sistemas MIMO
$G_{15}(s) = \begin{bmatrix} G_1(s) & 0 \\ 0 & G_6(s) \end{bmatrix}$
$G_{16}(s) = \begin{bmatrix} G_1(s) & 0.5 \cdot G_2 s \\ 0.5 \cdot G_5(s) & G_6(s) \end{bmatrix}$
$G_{17}(s) = \begin{bmatrix} G_1(s) & 0.9 \cdot G_2 s \\ 0.9 \cdot G_5(s) & G_6(s) \end{bmatrix}$
$G_{18}(s) = \begin{bmatrix} G_1(s) & 1.2 \cdot G_2 s \\ 1.2 \cdot G_5(s) & G_6(s) \end{bmatrix}$

Tabla 6.3: Función de transferencia de sistemas MIMO.

$K(s)$	(ind)	$b_{(ind)}$	$c_{(ind)}$	$K_{(ind)}$	$T_{i(ind)}$	$T_{d(ind)}$	$1/N_{(ind)}$	$T_{f(ind)}$	$\Vert\frac{d}{ds}\frac{T_{er}(s)}{s}\Vert_2$	$\Vert T_{yd}(s)\Vert_\infty$	$\Vert\frac{T_{yl}}{s}\Vert_1$
PID^a_{15}	$(1,1)$	0.351	0.173	1.437	1.891	0.949	0.101	0.094	3.07	1.379	1.709
	$(2,2)$	0.352	0.106	0.623	0.961	0.291	0.888	0.630			
PID^b_{15}	$(1,1)$	0.904	0.349	2.256	3.211	0.52	0.369	0.069	1.408	1.752	1.423
	$(2,2)$	0.754	0.689	0.442	0.343	0.029	0.355	0.038			
PID^a_{16}	$(1,1)$	0.588	0.662	0.422	1.033	1.327	0.122	0.069	6.195	1.446	3.637
	$(2,2)$	0.206	0.017	0.861	1.569	0.320	0.074	0.921			
PID^b_{16}	$(1,1)$	0.812	0.480	0.485	0.877	0.752	0.106	0.033	3.528	1.817	2.806
	$(2,2)$	0.39	0.977	0.236	0.211	8.084	0.649	0.027			
PID^c_{16}	$(1,1)$	0.936	0.468	1.263	1.087	1.669	0.731	0.047	1.693	1.494	1.496
	$(1,2)$	0.588	0.247	1.122	5.13	0.204	0.25				
	$(2,1)$	0.913	0.552	-1.301	2.184	0.42	0.641				
	$(2,2)$	0.748	0.595	0.33	0.174	0.962	0.045	0.132			
PID^d_{16}	$(1,1)$	0.525	0.26	3.38	1.275	0.273	0.876	0.052	1.356	1.809	0.803
	$(1,2)$	0.199	0.897	0.711	5.072	4.213	0.764				
	$(2,1)$	0.928	0.107	-1.441	0.936	0.511	0.351				
	$(2,2)$	0.112	0.878	0.6	0.291	0.482	0.107	0.018			
PID^a_{17}	$(1,1)$	0.511	0.636	0.271	1.001	0.982	0.098	0.475	88.492	1.446	6.046
	$(2,2)$	0.609	0.467	0.531	1.393	0.202	0.390	0.112			
PID^b_{17}	$(1,1)$	0.018	0.929	0.301	0.583	8.656	0.722	0.067	53.451	1.793	3.873
	$(2,2)$	0.660	0.128	0.843	1.800	0.311	0.195	0.041			
PID^c_{17}	$(1,1)$	0.793	0.457	1.098	2.440	0.473	0.747	0.028	51.373	1.356	3.710
	$(1,2)$	0.45	0.153	1.226	15.703	0.191	0.169				
	$(2,1)$	0.127	0.956	-0.067	2.581	7.189	0.440				
	$(2,2)$	0.402	0.225	0.313	0.543	1.032	0.516	1.170			
PID^d_{17}	$(1,1)$	0.324	0.361	3.786	4.762	0.005	0.703	0.030	43.037	1.953	2.930
	$(1,2)$	0.573	0.138	0.829	12.958	10.713	0.414				
	$(2,1)$	0.013	0.453	-0.016	7.583	4.957	0.791				
	$(2,2)$	0.077	0.970	2.132	3.383	0.090	0.712	2.473			
PID^a_{18}	$(1,2)$	0.738	0.770	0.399	0.854	13.904	0.863	0.684	31.108	1.886	9.897
	$(2,1)$	0.957	0.762	0.336	1.562	1.136	0.441	0.469			
PID^b_{18}	$(1,2)$	0.165	0.193	0.204	1.009	2.988	0.208	0.170	90.313	1.402	16.012
	$(2,1)$	0.565	0.681	0.128	1.098	1.700	0.967	0.078			
PID^c_{18}	$(1,1)$	0.193	0.193	-0.031	5.909	3.579	0.999	1.401	57.222	1.380	11.062
	$(1,2)$	0.646	0.333	0.254	1.042	10.444	0.855				
	$(1,2)$	0.302	0.902	0.244	1.348	9.534	0.702				
	$(2,2)$	0.834	0.840	0.259	9.551	8.288	0.557	0.987			
PID^d_{18}	$(1,1)$	0.085	0.340	0.017	10.824	4.369	0.571	0.832	24.492	1.840	7.469
	$(1,2)$	0.727	0.616	0.307	0.705	6.410	0.671				
	$(2,1)$	0.832	0.948	0.316	0.935	9.092	0.845				
	$(2,2)$	0.263	0.336	0.372	3.715	9.148	0.918	0.241			

Tabla 6.4: Parámetros de los controladores para sistemas MIMO.

Capítulo 7

Diseño de controladores robustos multiobjetivo.

7.1 Introducción.

La mayoría de los procesos existentes en la industria son demasiados complejos como para hallar un modelo exacto a base de las leyes físicas que rigen su comportamiento. Por ello estos procesos son modelados a partir de los resultados empíricos obtenidos en su salida al ser excitados por determinadas señales de entrada. Esta aproximación es conocida como identificación de un proceso. Cuando el proceso es de múltiple entrada, múltiple salida (MIMO), el modelo del mismo lo constituye el conjunto de todas las funciones de transferencia entre las diferentes entradas y salidas. La identificación se complica si no es posible excitar al sistema de forma independiente para las diversas entradas, ya que se deberá determinar que parte de la señal de salida corresponde a cada señal de entrada.

Una modelo típico empleado en control de procesos químicos es un sistema de primer orden con retardo. Estos modelos tienen tres parámetros, ganancia, constante de tiempo y retardo que identifican la agregación de una serie de dinámicas del proceso. La identificación de estos parámetros nunca es exacta, lo que hace que tengan un margen de error. Por ello, los sistemas de primer orden con retardo que identifican un proceso son ejemplos claros de sistemas con incertidumbre paramétrica, ver sección A.5.

Para diseñar un controlador multiobjetivo en un sistema de este tipo existen dos caminos, ver sección 2.2. El primero es la descomposición del sistema en dos bloques funcionales, con ayuda de la transformación fraccional lineal (LFT), ver apéndice A, y la aplicación del problema de optimización (2.1). El segundo es el planteamiento directo de un problema de min-max (2.2) entre el controlador y la incertidumbre de la planta.

En este capítulo se va a diseñar controladores para una planta industrial usando las dos técnicas citadas, y se van a comparar los resultados obtenidos. Por un lado se optimizará una serie de objetivos respecto de los parámetros del controlador en el caso nominal, ecuación (2.5). Por otro se optimizará los mismos objetivos respecto de los parámetros del controlador para el peor caso de la incertidumbre, ecuación (2.6). Como se comentó en la sección 4.2 existe muy poca literatura donde se aborde el segundo problema de forma directa.

El modelo propuesto es una planta con dos entradas y dos salidas donde las funciones de transferencia de cada subsistema se pueden identificar como de primer orden con retardo, con un margen de error en el valor de sus parámetros. El controlador que se ha elegido está compuesto por dos PID, en su forma serie (ver apéndice E) desacoplados, de forma que cada señal de control sólo dependerá del valor de una de las realimentaciones. Los objetivos a optimizar son la minimización de la sensibilidad y la sensibilidad complementaria en diferentes franjas de frecuencia, ver [ZDG96], con una serie de restricciones en la posición de los polos del sistema en lazo cerrado.

La herramienta utilizada para la solución de estos problemas son el algoritmo MRCD y el algoritmo MRCD min-max, ver sección 4.4.

Los resultados de este capítulo han sido presentados en una ponencia en el congreso "Evolutionary Computation Conference (GECCO'99)", ver [HLBLPG99].

La estructura de este capítulo es la siguiente, en la sección 7.2 se plantea el problema de diseño multiobjetivo que se desea resolver En la sección 7.3 se resuelve el problema para el caso donde la planta tenga parámetros con un valor nominal, usando el algoritmo MRCD. En la sección 7.4 se resuelve el problema min-max para el caso donde la planta tenga incertidumbre paramétrica por medio del algoritmo MRCD min-max. En la sección 7.5 se formulan las conclusiones del capítulo.

7.2 Diseño de un controlador desacoplado para una planta acoplada.

7.2.1 Planteamiento del problema.

En la industria, la mayoría de los procesos químicos con una entrada y una salida, (SISO), pueden ser modelados por medio de tres parámetros; la ganancia estática, la constante de tiempo y el retardo.

Se considera el siguiente proceso en lazo abierto de dos entradas y dos salidas

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} p_{11}(s) & p_{12}(s) \\ p_{21}(s) & p_{22}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}, \tag{7.1}$$

donde cada una de las funciones de transferencia se representa por tres parámetros

$$p_{ij}(s) = \frac{C_{ij}e^{-sR_{ij}}}{sT_{ij}+1}. \tag{7.2}$$

En los procesos industriales, los modelos como (7.1-7.2) representan la agregación de muchos elementos dinámicos y se obtienen normalmente por experimentación (identificación). La identificación de los parámetros de la planta normalmente no es exacta, por lo que es interesante plantear cierto grado de incertidumbre en el modelo. Un método es considerar que los parámetros tienen un valor nominal con una incertidumbre paramétrica, tal y como se ha desarrollado en la sección A.5. Las incertidumbres paramétricas están formuladas de la forma:

$$\Theta_{ij} = \Theta_{ij}^{nom}(1+\Theta_{ij}^{\Delta}), \quad \Theta \in \{C, T, R\} \;\; i \in \{1, 2\}$$

Se va a considerar los siguientes parámetros nominales para nuestro proceso de dos entradas y dos salidas (7.1):

$$C^{nom} = \begin{bmatrix} 4 & 1 \\ 4 & 2 \end{bmatrix}, \; T^{nom} = \begin{bmatrix} 50 & 40 \\ 15 & 19 \end{bmatrix}, \; R^{nom} = \begin{bmatrix} 20 & 20 \\ 10 & 5 \end{bmatrix}. \tag{7.3}$$

En el problema propuesto se tiene una planta de dos entradas y dos salidas y el objetivo es el diseño de un controlador multivariable desacoplado formado de dos controladores PID. Las señales de control u_1 y u_2 se obtienen aplicando los errores de seguimiento a las referencias $r_1 - y_1$ y $r_2 - y_2$ como entradas a los reguladores PID_1 y PID_2, respectivamente. Las ecuaciones del controlador son

$$
\begin{aligned}
u_1 &= \mathrm{PID}_1(r_1 - y_1) \\
u_2 &= \mathrm{PID}_2(r_2 - y_2)
\end{aligned}
\tag{7.4}
$$

donde los controladores $\mathrm{PID}_i(s)|i \in \{1,2\}$ tienen la siguiente estructura en serie, tal y como se define en la sección E.2,

$$
\mathrm{PID}_i(s) = K_{C,i}\left(1 + \frac{1}{T_{I,i}s}\right)\left(\frac{1 + T_{D,i}s}{1 + \alpha_i T_{D,i}s}\right).
$$

El controlador desacoplado MIMO tiene por tanto ocho parámetros.

Los controladores desacoplados son fáciles de sintonizar en una planta cuyos términos cruzados sean pequeños [OR94], porque se puede trabajar como si las entradas y salidas fuesen independientes. Pero este hecho no ocurre en la planta elegida ya que la ganancia C_{22} es menor que la ganancia C_{21} con los que no es posible formular el problema de forma independiente. La matriz de ganancia relativa del sistema Λ, definida en [OR94] y que ya ha sido aplicada en la sección 6.6, es de la forma

$$
\Lambda = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.
\tag{7.5}
$$

Esto implica que el sistema va a ser relativamente fácil de controlar realimentando las entradas y salidas directas, y complicado si se realimentan la entradas y salidas cruzadas.

Como el controlador deseado tiene su estructura fija es muy difícil sintonizar sus parámetros por los métodos clásicos de control, como los abordados en el capítulo 2, sin embargo, éstas misma característica hace que sea un problema abordable por medio de GAs. Estamos ante un ejemplo claro para poder aplicar la formulación paramétrica del control multiobjetivo, ecuación (2.5), y la formulación paramétrica del control multiobjetivo robusto, ecuación (2.6).

7.2.2 Objetivos del problema.

En este capítulo se van a sintonizar diferentes controladores PID desacoplados por medio de los algoritmos MRCD y MRCD min-max, ver capítulo 4. El diseño de controladores será formulado como un problema de optimización de sensibilidades poderadas usando la norma $\mathcal{H}_2$, ver [ZDG96].

Se considera el sistema en lazo cerrado de la figura 7.1 donde P es la planta y K el controlador que se desea diseñar. La función de sensibilidad en lazo cerrado es la función de transferencia entre el ruido de entrada $d(t)$ a la salida $y(t)$,

$$T_{yd} = (I - PK)^{-1}.$$

La función de sensibilidad complementaria T cumple con la formula $S + T = I$ y es igual a la función de transferencia negativa entre el ruido n a la salida y,

$$T = I - S = -T_{yn} = PK(I + PK)^{-1}.$$

Se necesita una pequeña sensibilidad para atenuar las perturbaciones en la planta y una pequeña sensibilidad complementaria para atenuar el ruido en la medida y para la robustez, ver [ZDG96]. Pero ambos requerimientos son complementarios debido a que $S+T = I$. Afortunadamente no se requiere optimizar ambos objetivos en la misma banda de frecuencia, ya que la sensibilidad es importante a bajas frecuencias y la sensibilidad complementaria es importante a altas frecuencias. El método para conseguir ambos objetivos es la introducción de un filtro de frecuencia a ambas funciones para conseguir optimizar a cada una de ellas en una banda de frecuencia diferente.

El problema de optimización de la sensibilidad se puede formular como:

$$\begin{gathered}\min_{K} \begin{bmatrix} \|W_d T_{yd}\|_p \\ \|W_n T_{yn}\|_q \end{bmatrix} \\ \text{s.a. Restricciones}\end{gathered} \tag{7.6}$$

para un controlador K que estabilice la planta. Los pesos W_d y W_n son un filtro paso-bajo y uno pasa-altos, respectivamente, y p y q son las normas elegidas.

7.3 Optimización de la sensibilidad multiobjetivo para una planta nominal.

Se va a resolver la optimización de sensibilidad multiobjetivo con un frente de Pareto de dos objetivos, las dos sensibilidades, y con un frente de Pareto de tres objetivo, las dos sensibilidades y el sobrepico en la señal temporal de salida ante una señal escalón en la referencia.

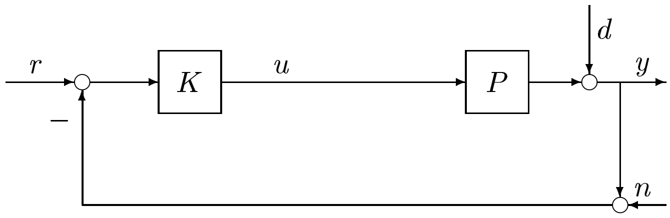


Figura 7.1: Sistema realimentado estándar.

7.3.1 Frente de Pareto de dos objetivos.

En esta sección se resuelve el problema de optimización multiobjetivo formulado en (7.6), como un problema de dos objetivos y restricciones. La planta P está definida en (7.1) con los parámetros nominales dados en (7.3). El controlador K es un controlador desacoplado PID (7.4), donde

$$K = \begin{bmatrix} \mathrm{PID}_1 & 0 \\ 0 & \mathrm{PID}_2 \end{bmatrix}.$$

El filtro pasa-bajos W_d, es un filtro de Butterworth de tercer orden

$$W_d(s) = \frac{10\omega_0^3}{s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3} I_{2\times 2},$$

y el filtro pasa-altos es

$$W_n(s) = \frac{s + \omega_0/10}{s + \omega_0} I_{2\times 2},$$

donde $\omega_0 = 10^{-2}$.

Se han añadido ciertas restricciones sobre la localización de los polos de lazo cerrado. Dichos polos deben pertenece a un sector cónico de la parte izquierda del semiplano negativo con un ángulo de $\theta < 7\pi/9$ radianes, además de tener una parte real menor de $\sigma = -1/100$. Con estas restricciones en la localización de los polos se garantiza un grado de estabilidad y un tiempo de respuesta. Por simplicidad, se ha utilizado la norma $\mathcal{H}_2$ para este problema.

$[K_C^{\max}, K_C^{\min}]$	$[T_I^{\max}, T_I^{\min}]$	$[T_D^{\max}, T_D^{\min}]$	$[\alpha^{\max}, \alpha^{\min}]$
[0.5,0]	[1,25]	[0.1,10]	[0,0.5]

Tabla 7.1: Límites iniciales de los parámetros de los controladores.

El problema multiobjetivo a resolver es:

$$\min_{K} \begin{bmatrix} \|W_d T_{yd}\|_2 \\ \|W_n T_{yn}\|_2 \end{bmatrix}$$

$s.a.$ K estabilizador

Polos en sección cónica

$\sigma < -1/100$ Y $\theta < 7\pi/9$

Para hallar los controladores se va a sustituir los sistemas con retardo por su aproximación de Padè de primer orden.

El problema se ha resuelto utilizando el algoritmo MRCD que va a tener tres objetivos. Dos de ellos corresponden a los objetivos de sensibilidad filtrada y sensibilidad complementaria filtrada y el tercero agrupa a todas las restricciones. Los límites de los parámetros en la población inicial son los mostrados en la tabla 7.1. La población inicial es de 100 individuos, la probabilidad de cruce es de $p_{cruce} = 0.7$, la de mutación $p_{muta} = 0.1$ y la constante $t_{dom} = 5$, ver algoritmos 4.1 y operador selección 3.9.

La figura 7.2 (bloque superior derecho) muestra el valor de la sensibilidad y sensibilidad complementaria filtrada de los individuos de la última generación, de donde se puede deducir el frente de Pareto. Esta curva da mucha información porque determina los límites de las especificaciones de diseño. Los individuos con una alta sensibilidad complementaria filtrada son rápidos pero tienen menor robustez y pueden generar grandes sobrepicos ante una señal escalón en la referencia. La tabla 7.2 muestra los parámetros de dos controladores, PID$_A$ y PID$_B$, de distintas zonas del frente de Pareto, y la figura 7.3 muestra las respuestas del sistema, y_1 e y_2, para dichos controladores a dos señales escalón unitario en las referencias. En la figura 7.4 (bloque superior) se muestra las normas de las sensibilidades poderadas mínima y máxima de los sistemas con dichos controladores.

	K_C	T_I	T_D	α	$\Vert W_d T_{yd} \Vert_2$	$\Vert W_n T_{yn} \Vert_2$
PID_A	$\begin{bmatrix} 0.1815 \\ 0.4199 \end{bmatrix}$	$\begin{bmatrix} 17.8284 \\ 14.7590 \end{bmatrix}$	$\begin{bmatrix} 1.5259 \\ 1.7680 \end{bmatrix}$	$\begin{bmatrix} 0.7243 \\ 0.4707 \end{bmatrix}$	0.3314	0.2725
PID_B	$\begin{bmatrix} 0.0217 \\ 0.1680 \end{bmatrix}$	$\begin{bmatrix} 6.5567 \\ 12.3928 \end{bmatrix}$	$\begin{bmatrix} 3.7668 \\ 2.5938 \end{bmatrix}$	$\begin{bmatrix} 0.3289 \\ 0.4879 \end{bmatrix}$	0.5179	0.1268
PID_C	$\begin{bmatrix} 0.0876 \\ 0.3241 \end{bmatrix}$	$\begin{bmatrix} 22.9523 \\ 18.1173 \end{bmatrix}$	$\begin{bmatrix} 9.2713 \\ 9.3610 \end{bmatrix}$	$\begin{bmatrix} 0.3304 \\ 0.1104 \end{bmatrix}$	0.4527	0.3074

Tabla 7.2: Parámetros de los controladores PID_i.

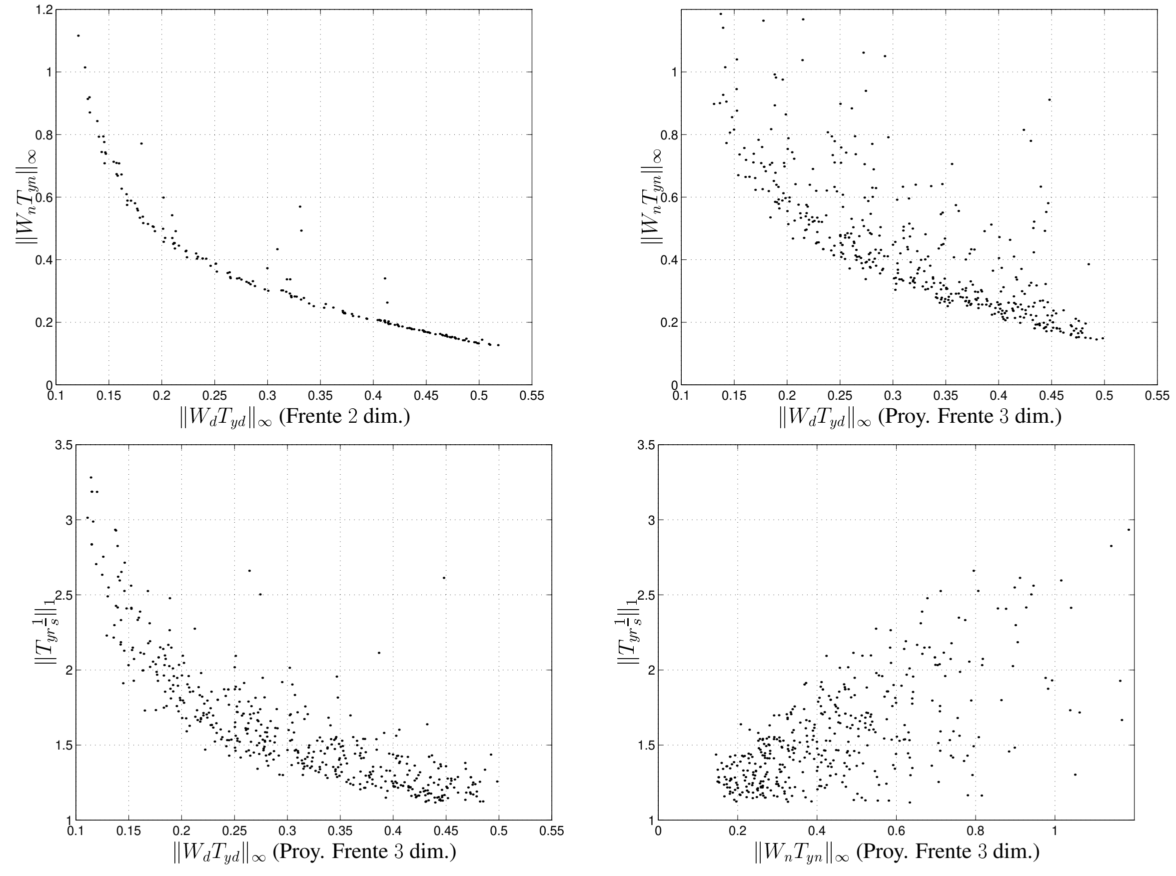


Figura 7.2: Frentes de Pareto de dos dimensiones y tres dimensiones.

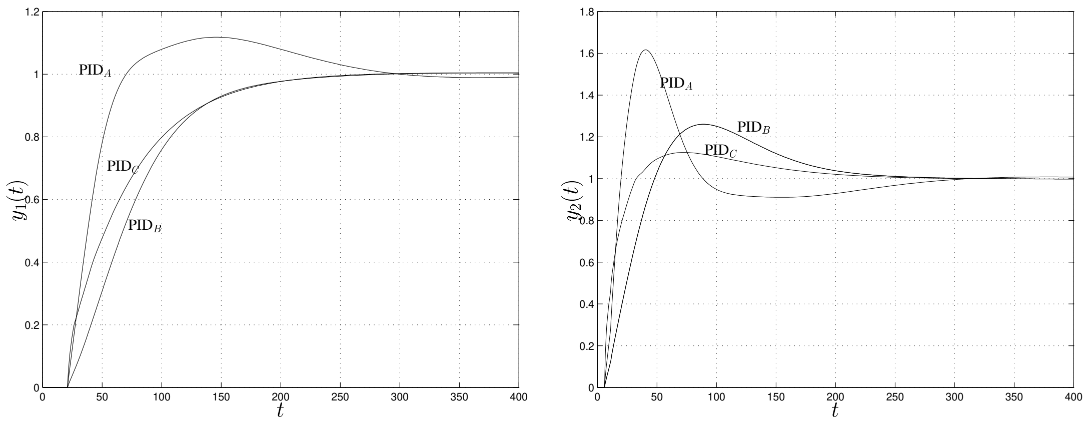


Figura 7.3: Respuestas de sistema $y_1(t)$ y $y_2(t)$ ante una entrada referencia.

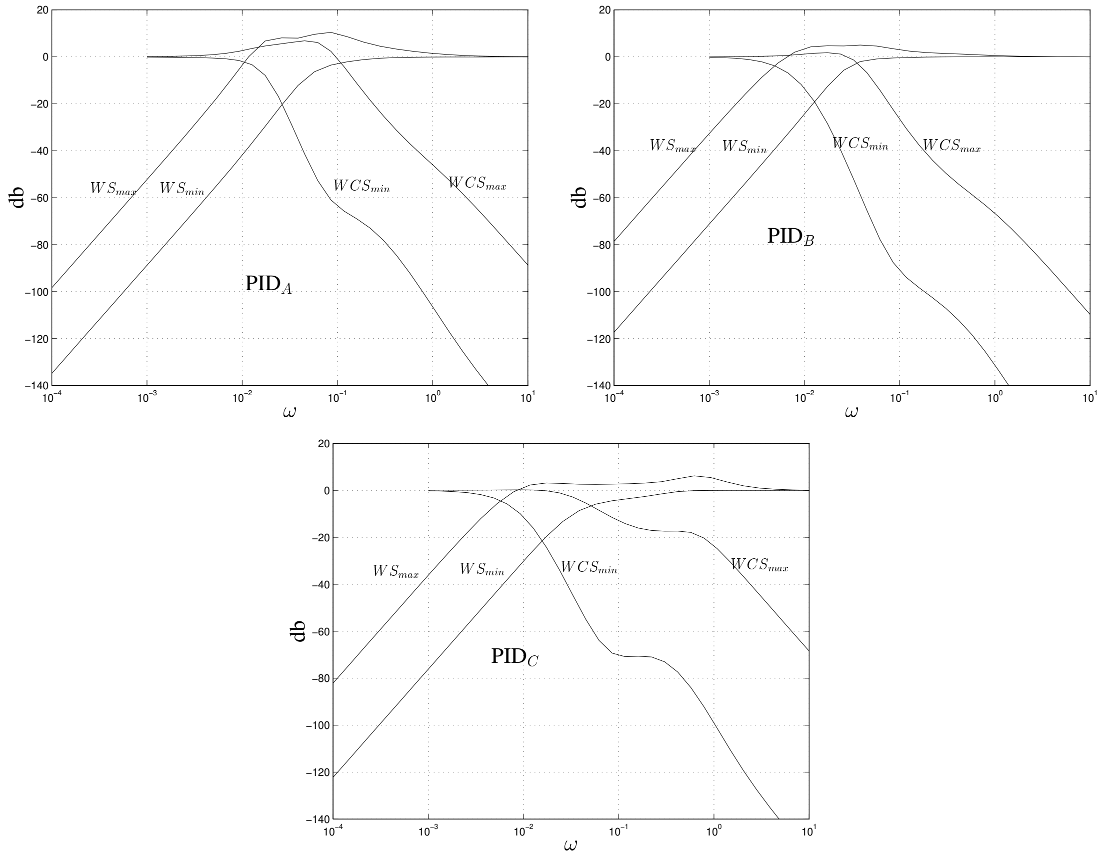


Figura 7.4: Sensibilidades filtradas máxima y mínima para cada controlador.

7.3.2 Frente de Pareto de tres objetivos.

El sobrepico de salida a una señal de entrada escalón es una propiedad importante en las especificaciones de un controlador, por este motivo se ha modificado el problema planteado en el apartado anterior con un nuevo objetivo, la minimización del sobrepico. La formulación matemática de este nuevo objetivo se puede realizar minimizando la norma-∞ de la señal de salida correspondiente a una entrada escalón, $||T_{yr}\frac{1}{s}||_\infty$, ver apéndice B.

El nuevo problema a optimizar tiene tres objetivos y restricciones respecto a la posición de los polos del sistema en lazo cerrado,

$$\min_K \begin{bmatrix} ||W_d T_{yd}||_2 \\ ||W_n T_{ny}||_2 \\ ||T_{yr}\frac{1}{s}||_\infty \end{bmatrix}$$

s.a. K estabilizador

Polos Lazo Cerrado en la sección cónica:

$\sigma < -1/100$ Y $\theta < 7\pi/18$.

Para resolver este nuevo problema se ha usado el algoritmo MRCD, que convierte las restricciones del mismo en un nuevo objetivo a optimizar, donde se mide la proximidad del sistema de lazo cerrado a cumplirlas. Para ello se mide la posición de los polos del sistema en lazo cerrado. El resultado en un frente de Pareto de tres dimensiones. En la figura 7.2 (bloque superior izquierdo y bloques inferiores) se muestra las proyecciones del frente de Pareto de los individuos de la última generación.

La densidad de la población final es mayor en la zona donde la sensibilidad complementaria filtrada es menor. La razón es la existencia de cierta correlación entre la sensibilidad complementaria filtrada y el sobrepico, ver figura 7.2. Se ha elegido el controlador PID$_C$ de los obtenidos en este nuevo problema por poseer menor sobrepico, como se ve en la respuesta temporal comparada de la figura 7.3. En la figura 7.4 (bloque inferior) se muestra las sensibilidades filtradas mínima y máxima del sistema con el nuevo controlador.

7.4 Optimización de sensibilidades multiobjetivo para una planta con incertidumbre.

En esta sección se asume que los parámetros (7.3) de la planta nominal propuesta en (7.1) tienen incertidumbre. Se desea diseñar un controlador PID desacoplado que garantice las especificaciones para el problema de sensibilidades filtradas propuesto en la sección 7.3.2, en el peor caso de la incertidumbre de los parámetros nominales de la planta.

Se asume que las incertidumbres en los parámetros de la planta son de la forma,

$$\Theta_{ij} = \Theta_{ij}^{nom}(1 + \Theta_{ij}^{\Delta}), \quad \Theta \in \{C, T, R\} \;\; i \in \{1, 2\}$$

y $|\Theta_{ij}^{\Delta}| \leq 0.1$. Por tanto la planta va a tener un 10% de incertidumbre en el valor de sus parámetros. El problema que se desea resolver se puede formular de forma general como un problema de optimización multiobjetivo min-max, tal y como se plantea en (2.6),

$$\min_{K} \max_{\Delta} \begin{bmatrix} \|W_d T_{yd}\|_2 \\ \|W_n T_{yn}\|_2 \\ \|T_{yr} \frac{1}{s}\|_\infty \end{bmatrix}$$

s.a. K estabilizador

Polos de Lazo cerrado en el sector cónico:

$\sigma < 1/100$ Y $\theta < 7\pi/18$.

Este problema es una minimización de Pareto para un vector de funciones, en el peor caso de una incertidumbre relativa Δ.

En la resolución de este problema se ha usado el algoritmo MRCD min-max, descrito en la sección 4.4. Se va a plantear un problema con cuatro objetivo, los tres del problema y un cuarto con las restricciones, que se intentarán minimizar respecto a los parámetros de controlador y maximizar respecto a los parámetros de la incertidumbre. El problema tiene veinte variables, doce de los términos de la incertidumbre y ocho de los parámetros del controlador.

Los límites del espacio de búsqueda inicial para los parámetros del controlador son los mismos que los empleados en el problema con la planta nominal, ver tabla 7.1. La población inicial es de 100 individuos, la probabilidad de cruce es de $p_{cruce} = 0.7$, la de mutación $p_{muta} = 0.1$, la constante $t_{dom} = 5$ para el operador selección y la constante $t_{max} = 2$ para el operador filtro de élite, ver sección 4.4 y operador selección 3.9.

K_C	T_I	T_D	α
$\begin{bmatrix} 0.0937 \\ 0.3123 \end{bmatrix}$	$\begin{bmatrix} 12.9344 \\ 24.4468 \end{bmatrix}$	$\begin{bmatrix} 0.1264 \\ 0.4196 \end{bmatrix}$	$\begin{bmatrix} 0.4306 \\ 0.4186 \end{bmatrix}$

Tabla 7.3: Parámetros del controlador PID$_D$.

La figura 7.5 muestra la sensibilidad filtrada y la sensibilidad complementaria filtrada para la población final. Esta solución refleja el vector objetivo para los mejores controladores en el peor caso de incertidumbre ($+$). Los vectores objetivos para los mismos controladores en el caso en los que la planta tenga sus parámetros nominales han sido marcados con ($\cdot$).

Esta claro que las especificaciones conseguidas con la peor incertidumbre del sistema no son tan buenas como las obtenidas con la planta nominal. En este ejemplo, la sensibilidad complementaria filtrada de la población son parecidas en ambos casos, no siendo así la sensibilidad filtrada.

En la figura 7.6 se compara la respuesta temporal del sistema, $y(t)$, ante una señal de entrada escalón en la referencia $r(t)$, para un controlador de la población final, PID$_D$. La figura representa su respuesta en el caso de que los parámetros de la planta sean los nominales, o los peores para el problema de optimización planteado. La tabla 7.3 representa los parámetros del controlador PID$_D$, la tabla 7.4 representa el valor de las incertidumbre que maximiza el problema para ese controlador y en la tabla 7.5 el valor de las sensibilidades para el caso nominal y para el peor caso. Se puede usar el concepto de ganancia relativa, [OR94], para ver como han empeorado las condiciones de la planta en el caso elegido. De la matriz de ganancias relativas de la planta nominal (7.5) se pasa, en el caso de la incertidumbre mostrada en la tabla 7.4, a una matriz de ganancias relativa bastante peor

$$\Lambda_\Delta = \begin{bmatrix} 3.2950 & -2.2950 \\ -2.2950 & 3.2950 \end{bmatrix},$$

por lo que se puede intuir que los resultados del control van a ser peores, como de hecho ocurre.

7.5 Conclusiones.

El objetivo de este capítulo es la utilización del algoritmo MRCD min-max para resolver de forma directa un problema de diseño multiobjetivo planteado en su forma paramétrica, ver ecua-

C^{Δ}	T^{Δ}	R^{Δ}
$\begin{bmatrix} -0.08636 & 0.09212 \\ 0.09590 & -0.05984 \end{bmatrix}$	$\begin{bmatrix} -0.09082 & 0.09246 \\ 0.09849 & 0.09788 \end{bmatrix}$	$\begin{bmatrix} -0.08620 & 0.06559 \\ 0.07695 & -0.09486 \end{bmatrix}$

Tabla 7.4: Peor caso de la incertidumbre paramétrica para el controlador PID_D.

	$\|W_d T_{yd}\|_2$	$\|W_n T_{yn}\|_2$
Nominal	0.4509	0.1037
Con incer.	0.6463	0.1099

Tabla 7.5: Sensibilidad filtrada para el controlador PID_D en el caso nominal y con la peor incertidumbre.

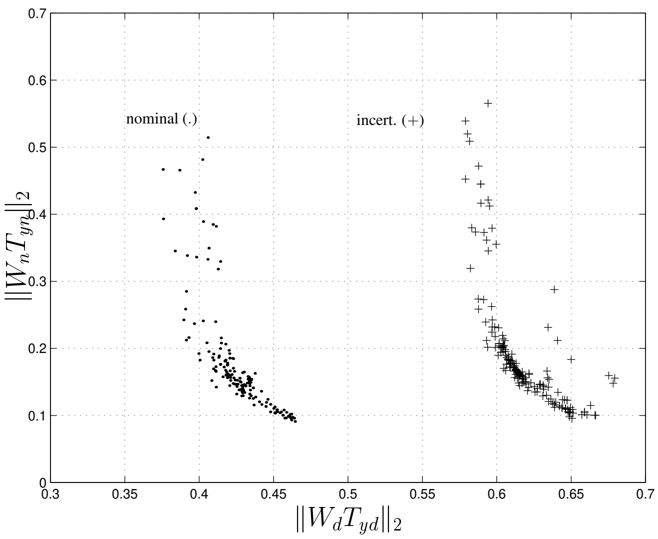


Figura 7.5: Población final y frente de Pareto para el problema min-max.

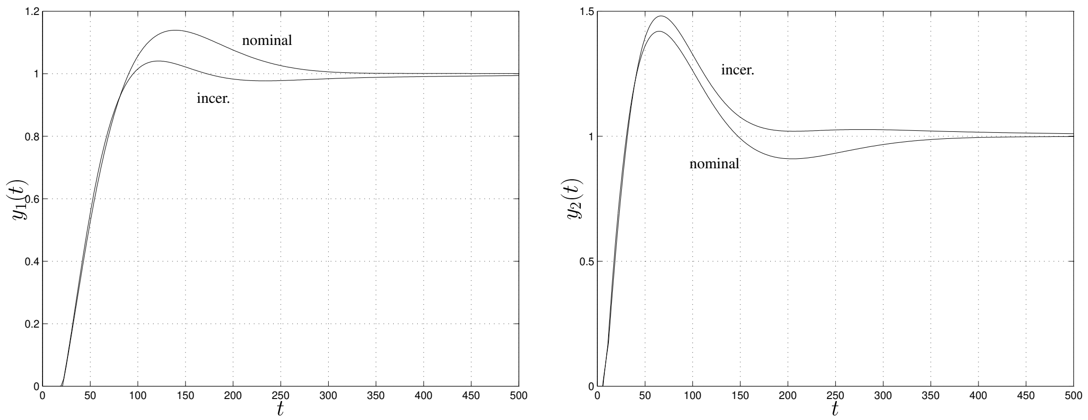


Figura 7.6: Respuestas de sistema $y_1(t)$ y $y_2(t)$ para el controlador PID_D.

ción (2.6). Este problema ha sido muy poco tratado en la literatura por su complejidad. La planta elegida para el diseño tiene dos entrada y dos salidas y sus subsistemas son de primer orden con retardo. La planta está fuertemente acoplada lo que hace que la variación en el valor de sus parámetros modifique mucho su dinámica. El controlador elegido ha sido un controlador formado por dos PID (forma serie) conectados en diagonal, es decir que cada uno de ellos depende de una realimentación. El problema plateado es la minimización de la sensibilidad y de la sensibilidad complementaria del sistema.

Se ha resuelto el problema cuando la planta tiene sus parámetros nominales por medio del algoritmo MRCD, y cuando la planta tiene incertidumbre en dichos parámetros con el algoritmo MRCD min-max. Se puede ver la diferencia en el frente de Pareto para el caso en el que los parámetros de la planta tienen sus valores nominales y en el que tienen los valores que maximizan el vector objetivo. El uso de este algoritmo proporciona controladores robustos para todo el rango de incertidumbre de la planta.

Capítulo 8

Conclusiones.

8.1 Contribuciones principales.

La teoría de control moderna formula el diseño de controladores como un problema de optimización, que en el caso de desear varios objetivos al mismo tiempo se convierte en un problema de optimización multiobjetivo. Las normas entre diferentes entradas y salidas del sistema planta-controlador en lazo cerrado son el medio matemático de formular las especificaciones y robustez que se desean conseguir, o sea, los objetivos del problema. La solución de un problema de optimización multiobjetivo no es única, sino un conjunto de soluciones llamado conjunto de óptimos de Pareto.

En los últimos años se han planteado numerosos problemas de optimización mixta, cuya solución es muy compleja y donde los controladores que se obtienen son de un orden muy elevado. En muchas ocasiones dichos controladores no se pueden implementar físicamente, principalmente por restricciones en su estructura y/o el valor de sus parámetros.

En este trabajo se ha planteado de forma general el problema de diseño de controladores que optimicen una serie de objetivos como un problema de minimización de dichos objetivos respecto a los parámetros del controlador, cuya estructura ha sido definida a priori. Por otro lado, se ha planteado el problema de diseño de controladores que optimicen una serie de objetivos en una planta con incertidumbre paramétrica como un problema de min-max entre los parámetros del controlador y los de la incertidumbre de la planta. Este planteamiento es flexible a todo tipo de restricciones en la estructura del controlador y a todo tipo de objetivos que se deseen optimizar.

Para resolver los problemas de optimización planteados se han desarrollado dos algoritmos, basados en los algoritmos genéticos (GAs) multiobjetivos a los que se ha dotado de nuevos operadores capaces de resolver de forma general los problemas planteados. Dichos algoritmos han sido evaluados respecto a otros GAs multiobjetivos y frente a otras técnicas de diseño mixto de controladores, las inecuaciones lineales matriciales (LMI), con muy buenos resultados en ambos casos. Como aplicación directa de estas técnicas se ha planteado la sintonización de PIDs multiobjetivo en diferentes plantas de ensayo SISO y MIMO, así como otros casos prácticos donde se comparan los resultados de los algoritmos MRCD y MRCD min-max.

De forma detallada las contribuciones de este trabajo son:

1. Planteamiento general de un problema de diseño de controladores multiobjetivo, con una estructura dada a priori, como un problema de minimización de sus parámetros.

2. Planteamiento general de un problema de diseño de controladores multiobjetivo en una planta con incertidumbre paramétrica como un problema min-max entre los parámetros del controlador, fijados a priori, y los parámetros de la incertidumbre.

3. Estudio de los algoritmos genéticos multiobjetivo realizados hasta la fecha.

4. Desarrollo de un algoritmo genético multiobjetivo, MRCD, con diferentes operadores específicos para los problemas de diseño de controladores multiobjetivo planteados en este trabajo. Estos operadores son:

   Algoritmo paralelo en isla. Diferentes subpoblaciones actúan en diferentes zonas del espacio de búsqueda, intercambiándose información a través de migración de individuos entre ellas.

   Límites dinámicos del espacio de búsqueda. Los movimientos de la población generan la modificación de los límites del espacio de búsqueda con el fin de poder explorar el mayor espacio de búsqueda posible en las primeras generaciones y conseguir precisión en la solución en las últimas. Este operador es fundamental en el problema planteado por no estar definidos dichos límites.

   Restricciones como objetivos. Para dar libertad al algoritmo se introducen las restricciones como nuevos objetivos, lo que lleva a conseguir controladores óptimos a partir de una población inicial de controladores que inestabilizan la planta.

Élite y filtro en la élite. Consiguen una convergencia más rápida evitando por el filtro caer en una convergencia prematura.

Estos operadores consiguen dar una gran libertad en la elección de los límites iniciales del espacio de búsqueda, y convierten la resolución de cualquier problema mediante un procedimiento normalizado.

5. Desarrollo del algoritmo MRCD min-max que aporta principalmente un nuevo operador "Filtro en la élite", que lleva a resolver el problema de min-max planteado. Dicho operador modifica los parámetros de la incertidumbre de la planta en los individuos de la élite, con el fin de maximizar los objetivos.

6. Desarrollo de un interfaz de usuario para facilitar el uso de dichos algoritmos a un usuario inexperto.

7. Evaluación del algoritmo MRCD a través de siete funciones test sin restricciones y tres con ellas frente a las verdaderos soluciones y frente a las soluciones obtenidos por otros algoritmos genéticos (GAs) multiobjetivos. Su resultado ha sido muy positivo.

8. Evaluación del algoritmo MRCD frente a las técnicas de inecuaciones lineales matriciales (LMIs) en un problema mixto $\mathcal{H}_2/\mathcal{H}_\infty$ para trece sistemas de ensayo, con resultado muy positivo. Los límites iniciales del espacio de búsqueda en todos los casos fueron los mismos, lo que no ha sido un impedimento para su convergencia, independientemente del tipo de sistema.

9. Estudio de la convergencia de controladores dinámicos con diferentes formas paramétricas y distinto orden al usar los algoritmos MRCD.

10. Planteamiento de la sintonización de PIDs en sistemas SISO y MIMO como un problema de diseño multiobjetivo. Aplicación de este planteamiento en trece sistemas SISO de ensayo, y en sistemas MIMO obtenidos de la combinación de los anteriores. Comparación con estudios similares sobre los mismos sistemas. Los resultados han superado los estudios anteriores y se ha demostrado las posibilidades de los algoritmos MRCD para sintonizar cuatro PIDs acoplados y dos filtros (24 parámetros) a la vez.

11. Comparación entre el algoritmo MRCD y el MRCD min-max en el diseño de un controlador para una planta con incertidumbre paramétrica. En los resultados se puede apreciar el efecto que produce la modificación de los parámetros de la planta, y las ventajas del uso del algoritmo MRCD min-max en el diseño de controladores robustos.

8.1.1 Líneas futuras de investigación.

En este trabajo se intenta demostrar que el problema de control multiobjetivo planteado como un problema de optimización se puede abordar de forma mucho más general de la que hasta ahora se ha hecho con el uso de los algoritmos genéticos multiobjetivos. Para ello se han desarrollado dos algoritmos genéticos que incorporan nuevos operadores para que la formulación de este problema sea lo más general y normalizada posible. Las líneas futuras de investigación pueden ir encaminadas en dos sentidos. Por un lado la mejora de los algoritmos MRCD optimizando su tiempo de cálculo y desarrollando nuevos operadores para dar solución a nuevos problemas. Por otro lado, la aplicación práctica a nuevos problemas. De forma detallada se pueden formular las siguientes líneas de investigación:

1. Estudio de nuevas formas de parametrización de sistemas, por ejemplo las bases ortonormales, viendo la influencia que tiene su aplicación en la convergencia de los algoritmos MRCD.

2. Planteamiento del problema de identificación como un problema de optimización multiobjetivo y aplicación del algoritmo MRCD en su resolución.

3. Reducción del tiempo de cálculo del algoritmo MRCD, aprovechando el hecho de que posee una estructura paralela para implementarlo como un algoritmo paralelo.

4. Planteamiento general de un algoritmo para el proceso iterativo y combinado de identificación y control de sistemas basado en el algoritmo MRCD.

5. Aplicación del algoritmo MRCD en tiempo real en el control adaptativo de una planta. Ello lleva implícito un proceso de control-identificación paralelo y una mayor velocidad del algoritmo.

6. Aplicación del algoritmo MRCD al control descentralizado multiobjetivo de sistemas.

7. Aplicación del algoritmo MRCD en el control de sistemas no lineales y/o híbridos.

Apéndice A

Transformación Fraccional Lineal, (LFT).

A.1 Introducción.

Una transformación fraccional lineal LFT es un operador matricial que juega un papel importante en la teoría de control actual, ya que permite la unificación de los problemas de diseño de controladores para plantas con incertidumbre, en su realización de estado. La formulación de una planta con incertidumbre vía LFT permite separar la parte nominal de la planta de la incertidumbre en dos bloques bien definidos. De igual forma se podrá dividir una planta y un controlador en realimentación. Esta realización permite disponer de información sobre las entradas y salidas entre los diferentes bloques lo que será de mucha ayuda en el análisis de la estabilidad de la planta y en el diseño de controladores. En [DPZ91, ZDG96] se da la definición, propiedades y campos de aplicación de este operador, y son innumerables los artículos dedicados a control robusto donde se aplica, sirva como ejemplo [LZD91, WDBG91, LZD96].

La estructura de este capítulo es la siguiente. En la sección A.2 se define las LFTs de forma matemática. En la sección A.3 se declara una serie de propiedades de las LFTs de gran utilidad para el uso de las mismas en el modelado de sistemas. En la sección A.4 se estudia de forma general las aplicaciones de las LFT al campo de control, y sobre todo en el tratamiento de sistemas con incertidumbre. En la sección A.5 se dan indicaciones para modelar sistemas con incertidumbre paramétrica usando LFTs.

A.2 Definición de una LFT.

Definición A.1 *Supongamos que M es una partición de matrices complejas*

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{C}^{(p_1+p_2)\times(q_1+q_2)}$$

si se toma $\mathbb{D}_1 \subset \mathbb{C}^{p_1 \times q_1}$ y $\mathbb{D}_2 \subset \mathbb{C}^{p_2\times q_2}$, entonces se define la Transformación Fraccional Lineal (LFT) inferior y superior como la correspondencia

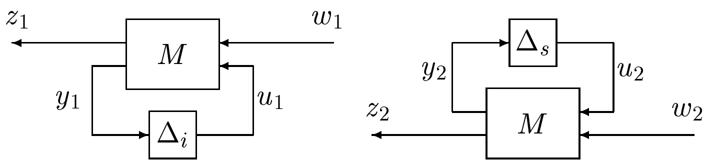
$$\mathcal{F}_l(M,\bullet) : \mathbb{D}_2 \mapsto \mathbb{C}^{p_1\times q_1}$$
$$\mathcal{F}_u(M,\bullet) : \mathbb{D}_1 \mapsto \mathbb{C}^{p_2\times q_2}$$

con

$$\mathcal{F}_l(M,\Delta_l) := M_{11}+M_{12}\Delta_l(I-M_{22}\Delta_l)^{-1}M_{21}\mathcal{F}_u(M,\Delta_u) := M_{22}+M_{21}\Delta_u(I-M_{11}\Delta_u)^{-1}M_{12}$$

siendo necesaria la existencia de ambas inversas para que el problema esté bien definido.

El nombre de inferior y superior viene dado por la representación que en diagrama de bloques tienen ambas transformaciones, dependiendo de la situación del bloque Δ.



El diagrama de la LFT inferior representa las ecuaciones

$$\begin{bmatrix} z_1 \\ y_1 \end{bmatrix} = M \begin{bmatrix} w_1 \\ u_1 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} w_1 \\ u_1 \end{bmatrix}$$
$$u_1 = \Delta_l y_1$$

mientras que el diagrama de la LFT superior representa las ecuaciones

$$\begin{bmatrix} y_2 \\ z_2 \end{bmatrix} = M \begin{bmatrix} u_2 \\ w_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} u_2 \\ w_2 \end{bmatrix}$$
$$u_2 = \Delta_u y_2.$$

El significado físico de las dos LFT propuesta es la función de transferencia de lazo cerrado entre $w_1 \mapsto z_1$, $T_{z_1 w_1} = \mathcal{F}_l(M, \Delta_l)$ y entre $w_2 \mapsto z_2$, $T_{z_2 w_2} = \mathcal{F}_u(M, \Delta_u)$. Se suele emplear la notación

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] := \mathcal{F}_u(\begin{bmatrix} A & B \\ C & D \end{bmatrix}, \Delta)$$

aunque su formulación puede resultar ambigua dependiendo del valor de Δ y de si se trata de una LFT inferior o superior, debiéndose aclarar por el contexto.

A.3 Propiedades de las LFTs.

Sea M, Q y G las siguientes partición de matrices:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \quad Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}, \quad G = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}.$$

La siguientes cinco propiedades son de gran utilidad para el diseño de controladores.

1. $\mathcal{F}_u(M, \Delta) = \mathcal{F}_l(N, \Delta)$, donde

   $$N = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} M \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} = \begin{bmatrix} M_{22} & M_{21} \\ M_{12} & M_{11} \end{bmatrix}$$

   donde las dimensiones de las matrices identidad son compatibles con las particiones de las matrices M y N.

2. Se supone que $\mathcal{F}_u(M, \Delta)$ es una matriz cuadrada, bien definida. Su inversa se puede formular como una LFT con respecto a Δ:

   $$(\mathcal{F}_u(M, \Delta))^{-1} = \mathcal{F}_u(N, \Delta)$$

   donde N tomará el valor

   $$N = \begin{bmatrix} M_{11} - M_{12} M_{22}^{-1} M_{21} & -M_{12} M_{22}^{-1} \\ M_{22}^{-1} M_{21} & M_{22}^{-1} \end{bmatrix}.$$

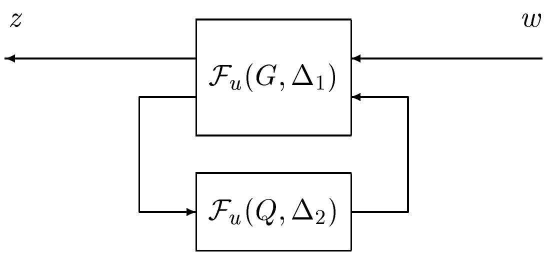
3. $\mathcal{F}_u(M,\Delta_1)+\mathcal{F}_u(Q,\Delta_2)=\mathcal{F}_u(N,\Delta)$, siendo

$$N=\left[\begin{array}{cc|c} M_{11} & 0 & M_{12} \\ 0 & Q_{11} & Q_{12} \\ \hline M_{21} & Q_{21} & M_{22}+Q_{22} \end{array}\right], \quad \Delta=\left[\begin{array}{cc} \Delta_1 & 0 \\ 0 & \Delta_2 \end{array}\right].$$

4. $\mathcal{F}_u(M,\Delta_1)\cdot\mathcal{F}_u(Q,\Delta_2)=\mathcal{F}_u(N,\Delta)$, siendo

$$N=\left[\begin{array}{cc|c} M_{11} & M_{12}Q_{21} & M_{12}Q_{22} \\ 0 & M_{11} & Q_{12} \\ \hline M_{21} & M_{22}Q_{21} & M_{22}Q_{22} \end{array}\right], \Delta=\left[\begin{array}{cc} \Delta_1 & 0 \\ 0 & \Delta_2 \end{array}\right].$$

5. Se supone que se tiene una estructura como la reflejada en la figura, donde las dimensiones de Δ_1 son compatibles con las de A.



La correspondencia entre $w \mapsto z$ viene dada por

$$\mathcal{F}_l(\mathcal{F}_u(G,\Delta_1),\mathcal{F}_u(Q,\Delta_2))=\mathcal{F}_u(\mathcal{F}_l(G,\mathcal{F}_u(Q,\Delta_2)),\Delta_1)=\mathcal{F}_u(N,\Delta)$$

$$N=\left[\begin{array}{cc|c} A+B_1Q_{22}L_1C_2 & B_2L_2Q_{21} & B_1+B_2Q_{22}L_1D_{21} \\ Q_{12}L_1C_2 & Q_{11}+Q_{12}L_1D_{22}Q_{21} & Q_{12}L_1D_{21} \\ \hline C_1+D_{12}L_2Q_{22}C_2 & D_{12}L_2Q_{21} & D_{11}+D_{12}Q_{22}L_1D_{21} \end{array}\right]$$

donde $L_1 := (I-D_{22}Q_{22})^{-1}$, $L_2 := (I-Q_{22}D_{22})^{-1}$, y $\Delta=\left[\begin{array}{cc} \Delta_1 & 0 \\ 0 & \Delta_2 \end{array}\right]$.

A.4 LFTs en el diseño de controladores.

Son muchas las aplicaciones de este operador y a continuación se darán las más interesantes desde el punto de vista de su aplicación a diseño de controladores.

A.4.1 Realización en espacio de estado.

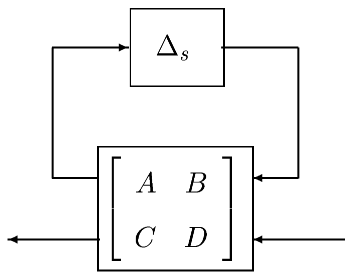
Si se define de forma general la matriz Δ como $\Delta \in \mathbb{R}^{n\times m}(s)$, o bien $\Delta \in \mathbb{R}^{n\times m}(z)$, las funciones de transferencia en realización de estado se pueden expresar como LFT. Dado una realización de estado del sistema continuo $G_n(s)$

$$\begin{bmatrix} \dot{x}(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(s) \\ u(s) \end{bmatrix},$$

y tomando como matriz $\Delta_s = \frac{1}{s}I$, la matriz de funciones de transferencia puede expresarse como

$$G_n(s) = D + C(sI - A)^{-1}B = \mathcal{F}_u(\begin{bmatrix} A & B \\ C & D \end{bmatrix}, \Delta_s). \tag{A.1}$$

El diagrama de bloques está representado en la siguiente figura.



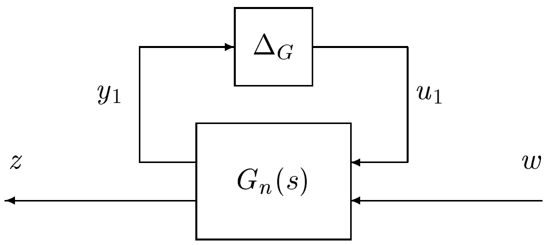
A Δ_s se la denomina estructura de frecuencias de un sistema en realización de estado. De forma similar se puede expresar la realización de un sistema de tiempo discreto, siendo $\Delta_z = \frac{1}{z}I$.

A.4.2 Problemas de diseño de controladores.

Representación de una familia de plantas.

Una de las principales aplicaciones del control robusto es la búsqueda de controlares validos para una familia de plantas. Un familia de plantas se definen como una planta nominal y una incertidumbre. Las LFTs proporcionan un marco adecuado para la representación de este problema. La familia de plantas representada por el diagrama de bloques de la figura se puede expresar vía LFT como

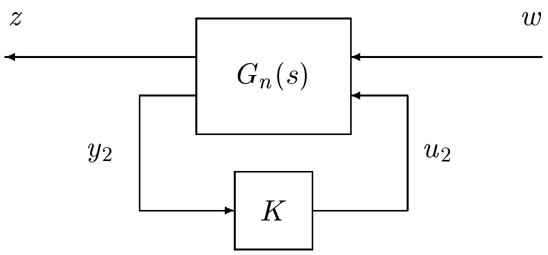
$$z = \mathcal{F}_u(G_n(s), \Delta_G)w. \tag{A.2}$$



Representación de una planta y un controlador en realimentación.

La planta nominal y el controlador en realimentación representado en la figura se puede expresar como

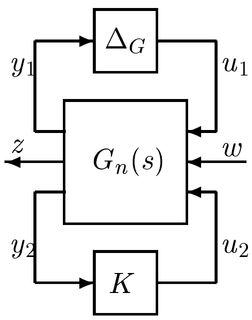
$$z = \mathcal{F}_l(G_n(s), K). \tag{A.3}$$



Planta con incertidumbre y controladores en realimentación.

Éste es el caso general del diseño de control robusto. La planta nominal, la incertidumbre y el controlador representado en la figura se puede expresar vía LFT como

$$z = \mathcal{F}_u(\mathcal{F}_l(G_n(s), K), \Delta_G) = \mathcal{F}_l(\mathcal{F}_u(G_n, \Delta_G), K) \tag{A.4}$$



La planta nominal $G_n(s)$ es otra LFT (A.1) al ser un sistema dinámico. Si la incertidumbre, el controlador o ambos fueran dinámicos podrían ser tratados vía LFT (A.1) igualmente. El conjunto planta nominal, controlador e incertidumbre puede ser simplificado con la aplicación de las propiedades de la sección A.3.

A.5 LFTs en el modelado de incertidumbres paramétricas.

El caso más común de incertidumbre tratado en esta tesis es el de incertidumbre paramétrica. Una planta tiene incertidumbre paramétrica cuando los parámetros de la misma tienen incertidumbre. La expresión general de un modelo con incertidumbre paramétrica viene dado como

$$\begin{gathered}\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \\ \delta = [\delta_1, \delta_2, \dots, \delta_n] \in \mathbb{R}^m. \end{gathered} \tag{A.5}$$

Este tipo de sistemas pueden representar un error en la medición de los parámetros, la posible variación de estos en el tiempo, o la linealización de un sistema no lineal en diferentes puntos de equilibrio, dependiendo del valor de δ, ver [KC98].

Un modelo LFT divide a la planta definida por la ecuación (A.5) en dos parte bien definidas, la parte nominal y la incertidumbre, tal y como se muestra en la figura A.1 y en la ecuación

$$\begin{gathered} z_\Delta = P_{11} w_\Delta + P_{12} \begin{bmatrix} x \\ u \end{bmatrix} \\ \begin{bmatrix} \dot{x} \\ y \end{bmatrix} = P_{21} w_\Delta + P_{22} \begin{bmatrix} x \\ u \end{bmatrix} \\ w_\Delta = \Delta z_\Delta \\ \Delta(\delta) = diag[\delta_1 I_{n_1}, \delta_2 I_{n_2}, \dots, \delta_m I_{n_m}] \\ dim(\Delta) = \textstyle\sum_{i=1}^m n_i, \quad n_i = dim(I_i). \end{gathered} \tag{A.6}$$

Minimizar el orden del sistema es minimizar el valor de $\dim(\Delta)$.

A.5.1 Formulación general de un modelo LFT.

El problema planteado es la obtención de un modelo LFT, figura A.1, de orden mínimo a partir de un sistema con incertidumbre paramétrica como el expuesto en (A.5).

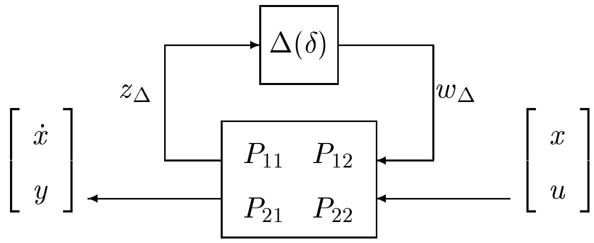


Figura A.1: Planta nominal e incertidumbre.

Cerrando el lazo en la ecuación (A.6) se puede escribir

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = [P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12} + P_{22}] \begin{bmatrix} x \\ u \end{bmatrix}. \tag{A.7}$$

Entre la ecuación (A.7) y (A.5) se deduce

$$\begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12} + P_{22} \tag{A.8}$$

$$P_{21}(I - \Delta P_{11})^{-1}\Delta P_{12} + P_{22}. \tag{A.9}$$

Si se divide la matriz $\begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix}$ en una parte nominal y otra con la incertidumbre

$$\begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} + \begin{bmatrix} A_\Delta & B_\Delta \\ C_\Delta & D_\Delta \end{bmatrix} \tag{A.10}$$

a partir de (A.9) se puede deducir que

$$\begin{aligned} \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} &= P_{22} \\ \begin{bmatrix} A_\Delta & B_\Delta \\ C_\Delta & D_\Delta \end{bmatrix} &= P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12} \\ &= P_{21}(I - \Delta P_{11})^{-1}\Delta P_{12}. \end{aligned} \tag{A.11}$$

El principal problema con este tipo de realización es la división de la planta en su parte nominal y su incertidumbre (A.10), que sólo se podrá plantear de forma directa en el caso de que los términos de las matrices sean lineales respecto de las incertidumbres. Si esto no fuera así, existen varias soluciones como el desarrollo en serie y la aproximación, problema abordado en [Bel98, BC98, BD98].

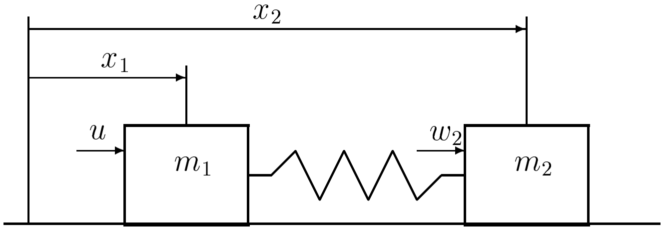


Figura A.2: Sistema masa muelle.

A.5.2 Modelo LFT a partir del diagrama de bloques.

Los pasos a dar para conseguir el modelo LFT de un sistema a partir de su diagrama de bloques serán explicados con el ejemplo del sistema masa/muelle propuesto en [WB92], problema muy utilizado en el estudio del control robusto.

Formulación del problema.

Supongamos que tenemos un sistema de la forma

$$\dot{x} = A(q)x + B(q)u$$
$$y = C(q)x + D(q)u$$

donde $q = [q_1, \ldots, q_n]$ son los parámetros con incertidumbre de la planta. Cada uno de estos parámetros tendrá un rango de incertidumbre conocido, $q_i \in [q_i^-, q_i^+]$.

El problema propuesto consiste en dos masas, m_1 y m_2, unidas por un muelle, figura A.2. La masa m_2 sufre una perturbación, fuerza w_2, y la masa m_1 tiene una entrada, fuerza u. Las variables x_1 y x_2 son las distancias respecto del origen de las dos masas. Las salidas del sistema son z que es la distancia real de la masa m_2 al origen, e y que es la misma distancia medida con una perturbación w_1. La idea de este modelo es la de anular a partir de la entrada u el efecto que sobre la masa m_2 ejerce la perturbación w_2, teniendo en cuenta que la distancia de la masa m_2, y, está siendo medida con una perturbación w_1.

Las ecuaciones del problema será

$$\ddot{x_1} + \frac{k}{m_1}(x_1 - x_2) = \frac{u}{m_1}$$
$$\ddot{x_2} + \frac{k}{m_2}(x_2 - x_1) = \frac{u}{m_2}$$

y sus ecuaciones de estado

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \hline \dot{x}_4 \\ z \\ y \end{bmatrix} = \left[\begin{array}{cccc|ccc} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{k}{m_1} & \frac{k}{m_1} & 0 & 0 & 0 & 0 & \frac{1}{m_1} \\ \frac{k}{m_2} & -\frac{k}{m_2} & 0 & 0 & 0 & \frac{1}{m_2} & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \hline u \\ w_1 \\ w_2 \end{bmatrix}.$$

El problema tiene tres parámetros que van a tener incertidumbre. En el caso de m_1 y m_2 está incertidumbre es un error en la medida y en el caso de $k(t)$ es una posible variación de su valor a lo largo del tiempo. En ambos casos se considera que el valor de estos parámetros va a estar dentro de un margen fijado

$$\begin{aligned} m_1 &\in [m_1^-, m_1^+] \\ m_2 &\in [m_2^-, m_2^+] \\ k(t) &\in [k^-, k^+] \end{aligned}$$

Normalización de los parámetros con incertidumbre.

Los parámetros con incertidumbre del sistema general son $[q_1, \ldots, q_n]$, donde $q_i \in [q_i^-, q_i^+]$. Se procederá a la normalización de la incertidumbre de dichos parámetros de forma que $q_i = q_{i0}(1 + \sigma_{q_i}\delta_{q_i})$, siendo

$$\begin{aligned} q_{i0} &= \tfrac{q^+ + q^-}{2} \\ \sigma_{q_i} &= \tfrac{q_i^+ - q_i^-}{q_i^+ + q_i^-} \\ \delta_{q_i} &\in [-1, 1]. \end{aligned}$$

En el ejemplo propuesto los parámetros se descompondrán en

$$\begin{aligned} m_1 &= m_{10}(1 + \sigma_{m_1}\delta_{m_1}) \\ m_2 &= m_{20}(1 + \sigma_{m_2}\delta_{m_2}) \\ k(t) &= k_0(1 + \sigma_k\delta_k(t)). \end{aligned}$$

Diagrama de bloques en su realización mínima.

El siguiente paso es la realización del diagrama de bloques de forma que cada parámetro con incertidumbre aparezca el mínimo número de veces posible. Con ello se conseguirá que el mo-

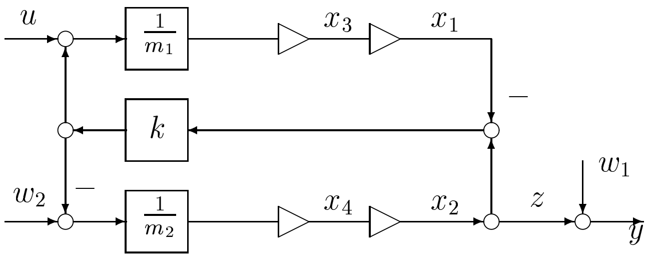


Figura A.3: Diagrama de bloques del sistema.

delo LFT tenga realización mínima, ver (A.6). Una realización mínima del problema propuesto es la representada en la figura A.3.

Diagrama de bloques de modelo LFT.

Un modelo LFT deberá estar dividido en planta e incertidumbre como se muestra en (A.6) y en la figura A.1. Para conseguirlo se deberá sustituir en el diagrama de bloques del paso anterior cada incertidumbre por su valor nominal.

En la figura A.4 están representadas en diagramas de bloques una incertidumbre normalizada $q_i = q_{i0}(1 + \sigma_{q_i}\delta_{q_i})$, y otra puesta en forma inversa $\frac{1}{q_j} = \frac{1}{q_{j0}(1+\sigma_{q_j}\delta_{q_j})}$. A partir de estas representaciones o de otras equivalentes se puede llegar a tener un diagrama de bloques donde estén representadas las entradas y salida de las incertidumbres del sistema. La figura A.5.2 representa el diagrama de bloques con las entradas y salidas del sistema y de las incertidumbres para el problema propuesto. La ecuación de estado de este diagrama de bloques en el orden dado en la figura A.3 es

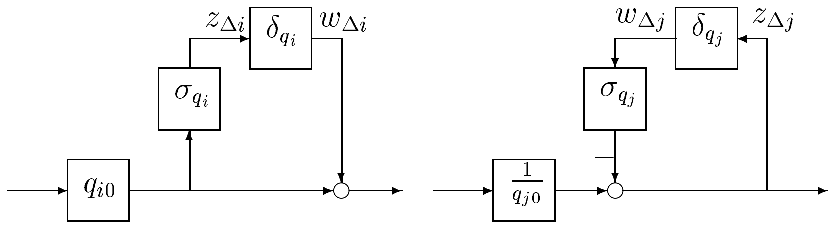


Figura A.4: Incertidumbre normalizada en forma directa e inversa.

$$
\left[\begin{array}{c} z_{\Delta 1} \\ z_{\Delta 2} \\ z_{\Delta 3} \\ \hline \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \hline z \\ y \end{array}\right]
=
\left[\begin{array}{ccc|cccc|ccc}
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{\sigma_k k_0}{m_{10}} & -\sigma_{m_1} & 0 & -\frac{k_0}{m_{10}} & \frac{k_0}{m_{10}} & 0 & 0 & 0 & 0 & \frac{1}{m_{10}} \\
-\frac{\sigma_k k_0}{m_{10}} & 0 & -\sigma_{m_2} & \frac{k_0}{m_{20}} & -\frac{k_0}{m_{10}} & 0 & 0 & 0 & \frac{1}{m_{10}} & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\frac{\sigma_k k_0}{m_{10}} & -\sigma_{m_1} & 0 & -\frac{k_0}{m_{10}} & \frac{k_0}{m_{10}} & 0 & 0 & 0 & 0 & \frac{1}{m_{10}} \\
-\frac{\sigma_k k_0}{m_{20}} & 0 & -\sigma_{m_2} & \frac{k_0}{m_{20}} & -\frac{k_0}{m_{20}} & 0 & 0 & 0 & \frac{1}{m_{20}} & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0
\end{array}\right]
\left[\begin{array}{c} w_{\Delta 1} \\ w_{\Delta 2} \\ w_{\Delta 3} \\ \hline x_1 \\ x_2 \\ x_3 \\ x_4 \\ \hline w_1 \\ w_2 \\ w_3 \end{array}\right]
$$

siendo la relación entre las entradas y salidas de las incertidumbres

$$
\begin{bmatrix} w_{\Delta 1} \\ w_{\Delta 2} \\ w_{\Delta 3} \end{bmatrix} = \begin{bmatrix} \delta_k & 0 & 0 \\ 0 & \delta_{m_1} & 0 \\ 0 & 0 & \delta_{m_2} \end{bmatrix} \begin{bmatrix} z_{\Delta 1} \\ z_{\Delta 2} \\ z_{\Delta 3} \end{bmatrix}.
$$

A partir de esta ecuación de estado es muy fácil la obtención de las matrices P_{ij} de la ecuación (A.6), con lo que el problema planteado estaría resuelto.

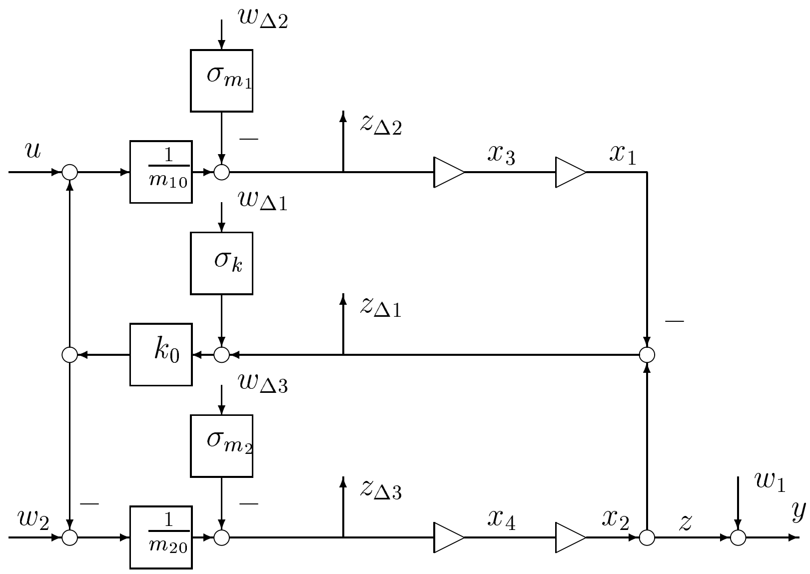


Figura A.5: Diagrama de bloques del sistema y las incertidumbres.

Apéndice B

Normas de señales y sistemas.

B.1 Normas de señales.

Una norma es una función con valores reales que asigna a cada elemento s de un espacio vectorial un número real $\|s\|$ llamado norma de s y que cumple con las siguientes propiedades:

1. $\|s\| \geq 0$ y $\|s\| = 0$ si y sólo si $s = 0$.

2. $\|s_1 + s_2\| \leq \|s_1\| + \|s_2\|$ para todo s_1 y s_2.

3. $\|\alpha s\| = |\alpha| \|s\|$ para todo $\alpha \in \mathbb{C}$.

Norma Euclídea de un vector $w \in \mathbb{R}^q$ viene definida como

$$|w| := \sqrt{w_1^2 + w_2^2, \ldots, w_q^2}$$

y si el vector es complejo $w \in \mathbb{C}^q$

$$|w| := \sqrt{w_1 \overline{w}_1 + w_2 \overline{w}_2 + \ldots + w_q \overline{w}_q}$$

Una señal es una función $s : T \to W$ donde $T \subseteq \mathbb{R}$ es el tiempo y W es un conjunto llamado espacio de la señal. Si el espacio $W \in \mathbb{R}^q$, una señal $s : T \to \mathbb{R}^q$ puede ser representada en cada instante de tiempo $t \in T$ como $s(t) = \left[\begin{array}{c} s_1(t), s_2(t), \ldots, s_q(t) \end{array} \right]'$.

Definición B.1 *Para una señal de tiempo discreto $s : T \to \mathbb{R}^q$ se definen las siguientes normas*

$$\begin{array}{c} \|s\|_\infty = \max_i \sup_{t \in T} |s_i(t)| \\ \|s\|_2 = \left\{ \sum_{t \in T} |s(t)|^2 \right\}^{1/2} \\ \|s\|_1 = \sum_{t \in T} |s(t)| \end{array}$$

o bien de forma general la $norma - p$, $1 \leq p < \infty$, *para una señal en tiempo discreto es*

$$\|s\|_p = \left\{ \sum_{t \in T} |s(t)|^p \right\}^{1/p}.$$

Definición B.2 *Para una señal de tiempo continuo* $s : T \to \mathbb{R}^q$ *se definen las siguientes normas*

$$\begin{aligned} \|s\|_\infty &= \max_i \sup_{t \in T} |s_i(t)| \\ \|s\|_2 &= \left\{ \int_{t \in T} |s(t)|^2 dt \right\}^{1/2} \\ \|s\|_1 &= \int_{t \in T} |s(t)| dt \end{aligned}$$

o bien de forma general la $p - norma$, $1 \leq p < \infty$, *para una señal en tiempo continuo es*

$$\|s\|_p = \left\{ \int_{t \in T} |s(t)|^p dt \right\}^{1/p}.$$

La norma de una señal da una idea de la magnitud de la misma. Estas normas son de más fácil interpretación en el caso de que $q = 1$. Para una señal $s : \mathbb{Z} \to \mathbb{R}$, es decir discreta, la norma-∞, $\|s\|_\infty$, es igual a la amplitud de la señal, la norma-2 es la potencia o energía de la señal y la norma-1 es la suma del valor absoluto de la señal.

No todas las señales tienen normas finitas, de forma que una norma $\|\cdot\|_p$ divide a las señales en dos conjuntos, el conjunto de las que tienen esa norma finita y las que no. De esta forma se tiene los siguientes espacios normalizados de señales para tiempo discreto

$$\begin{aligned} \ell_\infty &= \{s : T \to W | \ \ \|s\|_\infty < \infty\} \\ \ell_2 &= \{s : T \to W | \ \ \|s\|_2 < \infty\} \\ \ell_1 &= \{s : T \to W | \ \ \|s\|_1 < \infty\} \end{aligned}$$

y los siguientes espacios normalizados de señales para tiempo continuo

$$\begin{aligned} \mathcal{L}_\infty &= \{s : T \to W | \ \ \|s\|_\infty < \infty\} \\ \mathcal{L}_2 &= \{s : T \to W | \ \ \|s\|_2 < \infty\} \\ \mathcal{L}_1 &= \{s : T \to W | \ \ \|s\|_1 < \infty\}. \end{aligned}$$

Estos espacios son espacios normalizados lineales, lo que quiere decir que si s_1 y s_2 pertenecen a un espacio, entonces $s_1 + s_2$ y αs_1 también pertenecen a él, siendo α un número.

Normas de señales estacionarias.

A partir de las normas definidas con anterioridad se pueden inferir otras normas para señales estacionarias.

Definición B.3 *Se denomina norma-∞, o norma de pico, de estado estacionario a*

$$\|s\|_{ss\infty} \doteq \lim_{t\to\infty} \sup |s(t)| = \lim_{T\to\infty} \sup_{t\geq T} |s(t)|.$$

Esta norma mide la norma-∞ de la parte estacionaria de la señal, de forma que

$$\|s + s_{trans}\|_{ss\infty} = \|s\|_{ss\infty}, \quad \text{si} \quad \lim_{t\to\infty} s_{trans}(t) = 0.$$

Definición B.4 *Se define norma media cuadrada (RMS) de una señal s como*

$$\|s\|_{rms} \doteq \left(\lim_{T\to\infty} \frac{1}{T} \int_0^T s(t)^2 dt\right)^{1/2}$$

o bien para un vector de señales

$$\|s\|_{rms} \doteq \left(\lim_{T\to\infty} \frac{1}{T} \int_0^T s^T(t) s(t) dt\right)^{1/2}.$$

Esta norma nos da una idea de la media de la señal en el tiempo. Es una señal de medida del estado estacionario no influyendo en ella grandes pero infrecuentes valores de la señal.

Definición B.5 *Se define la norma media cuadrada (RMS) de una señal modelada como un proceso estocástico estacionario a*

$$\|s\|_{rms} = \mathcal{E}[s(t)^2]^{1/2},$$

o bien para un vector de señales

$$\|s\|_{rms} = (\sum_{i=1}^{n} \|s\|_{rms}^2)^{1/2}.$$

Esta norma no depende del tiempo al ser un proceso estacionario. Cuando la señal es ergódica, se puede definir su norma RMS de esta forma. Definiendo la autocorrelación de s y su potencia espectral como

$$R_s(\tau) \doteq \mathcal{E}[s(t)s(t+\tau)], \quad S_u(\omega) = \textstyle\int_{-\infty}^{\infty} R_s(\tau) e^{-j\omega\tau} d\tau,$$

se puede expresar la norma RMS de un proceso estocástico estacionario como

$$\|u\|_{rms}^2 = R_u(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_s(\omega) d\omega.$$

Definición B.6 *Se define la norma del valor de la media absoluta de una señal a*

$$\|s\|_{aa} \doteq \lim_{T\to\infty} \frac{1}{T} \int_0^T |s(t)|dt$$

o bien para un vector de señales

$$\|s\|_{aa} \doteq \lim_{T\to\infty} \sup \frac{1}{T} \int_0^n \sum_{i=1}^{T} |s_i(t)|dt$$

Para un proceso estocástico estacionario la norma valor medio absoluto y la norma de valor de pico estacionario se pueden expresar como

$$\|s\|_{aa} \doteq \mathcal{E}[|s(t)|], \quad \|s\|_{ss\infty} \doteq \inf\{a|\mathbf{Prob}(|u(t)| \geq a) = 0\}.$$

En este tipo de procesos se define la distribución de amplitud como

$$\mathbf{F}_s(a) \doteq \mathbf{Prob}(|u(t)| \geq a),$$

y se deduce que $\|u\|_{ss\infty} = \sup\{a|\mathbf{F}_s(a) > 0\}$.

Normas con pesos.

Las normas de señales con peso consisten básicamente en el filtrado de la señal a través de una función de forma quede resaltado las características que se deseen. Existen dos tipos de filtros, los filtros en frecuencia y en el tiempo. Así, la norma RMS de una señal estocástica estacionaria filtrada con un peso $W(j\omega)$ se define como

$$\|s\|_{W,rms} = \|Ws\|_{rms} = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} S_s(\omega)|W(j\omega)|^2 d\omega\right)^{1/2}$$

Un peso muy común es el derivador $W = s$, definiéndose tres diferentes normas.

Definición B.7 *Se definen las tres siguientes normas de derivadas de la señal*

$$\begin{aligned} \|s\|_{step_{rate}} &= \|\tfrac{ds}{dt}\|_\infty \\ \|s\|_{acc} &= \|\tfrac{d^2s}{dt^2}\|_\infty \\ \|s\|_{jerk} &= \|\tfrac{d^3s}{dt^3}\|_\infty. \end{aligned}$$

Algo similar ocurre con los pesos en el dominio del tiempo como por ejemplo la clásica norma ITAE (integral de tiempo multiplicada por el valor absoluto)

$$\|u\|_{itae} \doteq \int_0^\infty t|u(t)|dt.$$

Comparación entre normas.

A partir de las definición de las diferentes normas se desea la comparación entre las mismas. Para señales escalares se tiene

$$\|s\|_\infty \geq \|s\|_{rms} \geq \|s\|_{aa}$$

y para señales de n componentes se puede generalizar

$$\|s\|_\infty \geq \frac{1}{\sqrt{n}}\|s\|_{rms} \geq \frac{1}{n}\|s\|_{aa}.$$

También se puede comprobar que

$$\|s\|_{rms}^2 \leq \|s\|_{aa}\|s\|_\infty$$

B.2 Normas de sistemas SISO (una entrada, una salida).

En Ingeniería de Control normalmente se consideran sistemas donde las señales están divididas en dos grupos, señales de entrada y de salida, de tal forma que un sistema viene a representar la correspondencia entre un conjunto de señales de entrada y un conjunto de señales de salida, $H : \mathcal{U} \mapsto \mathcal{Y}$. Se va a asumir que tanto $\mathcal{U}$ como $\mathcal{Y}$ son conjuntos de señales normalizadas. Como ejemplo de señales de entrada, $\mathcal{U}$, se pueden mencionar las señales $\mathcal{L}_\infty$ (máximo limitado), las señales $\mathcal{L}_2$ (energía limitada), las señales impulso y las señales de ruido blanco (su media es cero y su varianza finita).

Definición B.8 *Se define norma inducida de un sistema $H : \mathcal{U} \mapsto \mathcal{Y}$ como*

$$\begin{aligned}
\|H\| &= \inf_M \{M | \|H(u)\| \leq M\|u\|, \ \ \forall u \in \mathcal{U}\} \\
&= \sup_{u\in\mathcal{U}, u\neq 0} \frac{\|H(u)\|}{\|u\|} \\
&= \sup_{u\in\mathcal{U}, \|u\|\leq 1} \|H(u)\| \\
&= \sup_{u\in\mathcal{U}, \|u\|=1} \|H(u)\|.
\end{aligned}$$

Dos ejemplos son la norma inducida $\mathcal{L}_\infty$ y la norma inducida de $\mathcal{L}_2$, equivalentes a la norma inducida media absoluta (aa) y la norma inducida de la media cuadrada (RMS) en señales *estacionarias,*

$$\|H\|_{(\infty,\infty)} := \sup_{u\in\mathcal{L}_\infty} \frac{\|H(u)\|_\infty}{\|u\|_\infty} = \sup_{\|u\|_{aa}\neq 0} \frac{\|H(u)\|_{aa}}{\|u\|_{aa}}$$
$$\|H\|_{(2,2)} := \sup_{u\in\mathcal{L}_2} \frac{\|H(u)\|_2}{\|u\|_2} = \sup_{\|u\|_{rms}\neq 0} \frac{\|H(u)\|_{rms}}{\|u\|_{rms}}.$$

Es decir, el sistema H va a estar limitado por un máximo valor M, lo que se puede interpretar como que el sistema es estable respecto al conjunto de señales de entrada.

Un sistema lineal, invariante en el tiempo y causal viene representado en tiempo continuo por una ecuación de convolución

$$y(t) = (Hu)(t) = h * u = \int_0^t h(t-\tau)u(\tau)d\tau \tag{B.1}$$

donde $h : \mathbb{R} \mapsto \mathbb{R}$ es una función que representa la respuesta del sistema a una entrada impulso de Dirac $u(t) = \delta(t), t = 0$.

La transformada de Fourier de la función $h(t)$ se define como $\hat{h}(jw) := \int_{-\infty}^{\infty} h(t)e^{jwt}dt$, y la transformada de Laplace se define como $H(s) := \int_{-\infty}^{\infty} h(t)e^{st}dt$. La función convolución se puede representar en los distintos dominios t, w y s como

$$y = h * u \iff \hat{y}(jw) = \hat{h}(jw)\hat{u}(jw) \iff Y(s) = H(s)U(s).$$

El sistema H será estable si la norma-1 de la salida a una entrada impulso es finita $h \in \mathcal{L}_1$, o si los polos de la función de transferencia $H(s)$ se encuentran el semiplano izquierdo.

Teorema B.1 *Sea H un sistema definido en (B.1), donde se supone $h \in \mathcal{L}_1$. Entonces*

1. *La norma inducida $\mathcal{L}_\infty$ de H viene dada por*

$$\|H\|_{(\infty,\infty)} = \|h\|_1$$

2. *La norma inducida $\mathcal{L}_2$ viene dada por $\|H\|_{(2,2)} = \max_{w\in\mathbb{R}} |\hat{h}(jw)|$*

El primer teorema responde a que si la respuesta a una entrada impulso de $h : \mathbb{R} \mapsto \mathbb{R}$ es $\|h\|_1 = \int_0^\infty |h(t)|dt < \infty$, el sistema es estable en el sentido de que cada entrada limitada produce una salida limitada, o bien que si $h \in \mathcal{L}_1$, para una entrada $u \in \mathcal{L}_\infty$ se produce una salida $y \in \mathcal{L}_\infty$. El segundo teorema se deduce del teorema de Paserval que demuestra que la norma-2 de una señal $u \in \mathcal{L}_2$ cumple la igualdad $\|u\|_2^2 = \frac{1}{2\pi}\|\hat{u}\|_2^2$.

Definición B.9 *Sea la función de transferencia de un sistema SISO estable $H(s)$ con una respuesta en frecuencia $\hat{h}(jw)$. La norma $\mathcal{H}_\infty$ de H se define como*

$$\|H\|_\infty := \max_{w\in\mathbb{R}} |\hat{h}(jw)|.$$

Es decir, la norma $\mathcal{H}_\infty$ de un sistema SISO es el mayor pico del diagrama de Bode de su respuesta en frecuencia $\hat{h}$. Y por lo dicho en el teorema B.1 la norma inducida $\mathcal{L}_2$ de un sistema H es igual a la norma $\mathcal{H}_\infty$ de su función de transferencia.

Definición B.10 *Sea la función de transferencia de un sistema SISO estable $H(s)$ con una respuesta en frecuencia $\hat{h}(jw)$. La norma $\mathcal{H}_2$ de H se define como*

$$\|H\|_2 := \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} H(jw)H(-jw)dw \right\}^{1/2}.$$

Por el teorema de Paseval se llega a la conclusión que la norma $\mathcal{H}_2$ de un sistema es la norma-2 de la respuesta del mismo a una entrada impulso $\delta(t), t = 0$, $\|H\|_2 = \|h\|_2$. Otra interpretación de la norma $\mathcal{H}_2$, está relacionada con las señales estocásticas estacionarias. La señal $u(t)$ es un ruido blanco si tiene de media $\mu := \mathcal{E}[u(t)] = 0$, y su covarianza $R_u(\tau) := \mathcal{E}[(u(t-\tau)-\mu)(u(t)-\mu)] = 0$ para toda $\tau \neq 0$. La norma $\mathcal{H}_2$ de un sistema $H(s)$ es la norma-2 de la covarianza de la señal de salida, cuando la entrada al sistema es un ruido blanco.

$$\|H(s)\|_2 = \|R_y(\tau)\|_2^2 = \left\{ \int_{-\infty}^{\infty} \mathcal{E}[y(t+\tau)y(t)]d\tau \right\}^{1/2}$$

B.3 Normas de un sistema MIMO (múltiple entrada, múltiple salida).

Un sistema H con m entrada y p salidas define la correspondencia entre un espacio de señales $\mathcal{U}^m$ de entrada y un espacio $\mathcal{Y}^p$ de salida, $H : \mathcal{U}^m \mapsto \mathcal{Y}^p$. Si el sistema es lineal, invariante en el tiempo y causal se puede representar por la función de convolución, y a partir de la transformada de Laplace

$$y(t) = h * u \Longleftrightarrow Y(s) = H(s)U(s)$$

En este caso $h(t)$ para $t \in \mathbb{R}$ y $H(s)$ para $s \in \mathbb{C}$ son dos matrices de dimensión $p \times m$. Cada término $[h(t)]_{ij}$ representa la salida $y_i(t)$ del sistema a una entrada $u_j(t) = \delta(t), t = 0$, y cada término $[H(s)]_{ij}$ la función de transferencia entre la salida y_i y la entrada u_j. El sistema será estable si todos los términos $[h(t)]_{ij}$ pertenecen a $\mathcal{L}_1$, o si los polos de todos los términos $[H(s)]_{ij}$ están en la parte izquierda del plano complejo. La norma inducida de un sistema viene definida de la misma forma a como se definió en B.8.

Definición B.11 *La descomposición en valores singulares estructurados (SVD) de una matriz $H \in \mathbb{R}^{p\times m}$ es la factorización de dicha matriz en $H = Y\Sigma U^T$, donde*

- $Y \in \mathbb{R}^{p\times p}$ *es ortogonal,* $Y^T Y = YY^T = I_p$,
- $U \in \mathbb{R}^{m\times m}$ *es ortogonal,* $U^T U = UU^T = I_m$,
- $\Sigma \in \mathbb{R}^{p\times m}$ *es diagonal,* $\Sigma = \begin{bmatrix} \Sigma' & 0 \\ 0 & 0 \end{bmatrix}$ *siendo*

$$\Sigma' = diag(\sigma_1, \ldots, \sigma_r)$$
$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0.$$

Con este método se puede descomponer una matriz H en una serie ordenada de números positivos $[\sigma_1, \sigma_2, \ldots, \sigma_r]$, únicos que se denominan valores singulares de la matriz H. A partir de la definición se puede comprobar que $H^T H = U\Sigma^2 U^T$ y que $HH^T = Y\Sigma^2 Y^T$. La forma practica de hallar los valores singulares de H es hallar los autovalores de la matriz $H^T H$ o de HH^T y ordenarlos de mayor a menor , $[\lambda_1 \geq \ldots \geq \lambda_r]$. Los valores singulares de H serán $\sigma_k = \sqrt{\lambda_k},\ \ k = 1, \ldots, r$. El número r es el rango de H.

Los valores singulares de una matriz se interpretan como los valores de la ganancia de dicha matriz en sus diferentes direcciones. Así, para $u \in \mathbb{R}^m$, la norma

$$\|Hu\|^2 = u^T H^T H u = u^T U \Sigma \Sigma^T U^T u = x^T \Sigma' \Sigma' x$$

donde $x = U^T x$. Por otro lado

$$\max_{\|u\|=1} \|Hu\|^2 = \max_{\|u\|=1} \|\Sigma' \Sigma' x\|^2 = \sum_{i=1}^{m} \sigma_i^2 |x_i|$$

donde se puede ver que es máximo cuando $x_1 = 1$ y $x_i = 0, \forall i \neq 1$. Es decir, cuando el vector u está en la dirección del mayor valor singular.

Dada la matriz de transferencia $H(s)$, de dimensión $p \times m$, su descomposición SVD para un valor $s = jw$ fijo será

$$H(jw) = Y(jw)\Sigma(jw)U^*(jw).$$

La matriz $H(jw) \in \mathbb{C}^{p\times m}$, en cada valor de w, tendrán una serie de valores singulares $\sigma_1(w) \geq \sigma_2(w) \geq \ldots \geq \sigma_r(w) > 0$, y

$$0 \leq \frac{\|H(jw)\hat{u}(jw)\|}{\|\hat{u}(jw)\|} \leq \sigma_1(w).$$

Se define la función $\overline{\sigma}(w) := \sigma_1(w)$ como el máximo valor singular del sistema a la frecuencia w.

Definición B.12 *Sea $H(s)$ la matriz de transferencia de un sistema MIMO estable. La norma $\mathcal{H}_\infty$ de $H(s)$ está definida como*

$$\|H(s)\|_\infty := \sup_{w\in\mathbb{R}} \overline{\sigma}(H(jw)).$$

Teorema B.2 *Para una matriz de transferencia estable $H(s)$ la norma inducida $\mathcal{L}_2$ asociadas a la correspondencia entre las señales de entrada y salida del sistema es equivalente a la norma $\mathcal{H}_\infty$ de $H(s)$.*

Los valores singulares del sistema a distintas frecuencias constituyen una generalización del diagrama de Bode en los sistemas SISO. La norma $\mathcal{H}_\infty$ es el máximo pico del estas curvas, y este máximo corresponde con la norma inducida $\mathcal{L}_2$.

Definición B.13 *Sea $H(s)$ una matriz de transferencia de un sistema MIMO estable de dimensiones $p \times m$. La norma $\mathcal{H}_2$ de $H(s)$ está definida como*

$$\|H(s)\|_2 = \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} traza[H^*(-jw)H(jw)]dw \right\}^{1/2}$$

donde la $traza(\cdot)$ de una matriz es la suma de los elementos de su diagonal. La norma-2 del vector salida por efecto de una entrada impulso $\delta(t), t = 0$ en u_i es

$$\|y^{(i)}\|_2^2 := \int_{-\infty}^{\infty} \sum_{j=1}^{p} |y_j^{(i)}(t)|^2 dt$$

ver definición (B.2), y la norma $\mathcal{H}_2$, tal y como se ha definido, es la suma de las normas-dos de las salida por efecto de señales impulso simultaneas en todas los canales de entrada,

$$\|H(s)\|_2^2 = \sum_{i=1}^{m} \|y^{(i)}\|_2^2.$$

Definición B.14 *Sea $H(s)$ una matriz de transferencia de un sistema MIMO estable, de m entradas y p salidas. Sea la matriz h, cuyos términos $[h]_{ij}$ son los elementos de la convolución $y(t) = h * u$. La norma $\mathcal{L}_1$ de $H(s)$ viene definida como*

$$\|H(s)\|_1 = \max_{1 \leq i \leq p} \sum_{j=0}^{m} \|[h]_{ij}\|_1$$

Esta norma es de difícil aplicación en el caso continuo, por lo que normalmente se aplicará su equivalente en discreto.

Teorema B.3 *Para una matriz de transferencia estable $H(s)$ la norma inducida $\mathcal{L}_\infty$ asociadas a la correspondencia entre las señales de entrada y salida del sistema es equivalente a la norma $\mathcal{L}_1$ de $H(s)$.*

Funciones peso como filtros.

El máximo valor singular de un sistema, $\overline{\sigma}(H(jw)), \forall w$, proporciona más información que la norma $\mathcal{H}_\infty$, como es obvio de la definición B.12. Al añadir filtros de entrada y salida al sistema, tal y como se muestra en la figura B.1, la norma $\mathcal{H}_\infty$ del sistema más los filtros, darán información sobre la forma de la función $\overline{\sigma}(H(jw)), \forall w$.

Sea el sistema $y(s) = H(s)u(s)$, y dos matrices de transferencia $V(s)$ y $W(s)$ estables, cuadradas y con las dimensiones del número de entradas y salidas de $H(s)$ respectivamente. Sean sus matrices inversas $V^{-1}(s)$ y $W^{-1}(s)$ igualmente estables. Las siguientes afirmaciones

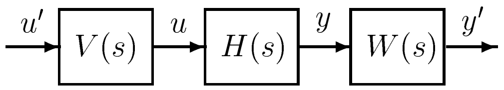


Figura B.1: Sistema con funciones peso de entrada y salida.

son equivalentes

$$\begin{array}{rcl}
\|H(s)\|_\infty \le \gamma & \iff & \overline{\sigma}(H(jw)) \le \|H(s)\|_\infty \le \gamma, \\
\|H(s)V(s)\|_\infty \le \gamma & \iff & \overline{\sigma}(H(jw)) \le \gamma\overline{\sigma}(V^{-1}(jw)), \\
\|W(s)H(s)\|_\infty \le \gamma & \iff & \overline{\sigma}(H(jw)) \le \gamma\overline{\sigma}(W^{-1}(jw)), \\
\|W(s)H(s)V(s)\|_\infty \le \gamma & \iff & \overline{\sigma}(H(jw)) \le \gamma\overline{\sigma}(V^{-1}(jw))\overline{\sigma}(W^{-1}(jw)).
\end{array}$$

Las matrices de transferencia $V(s)$ y $W(s)$ son los filtro de entrada y salida respectivamente.

Un sistema $V(s)$ cuyo sistema inverso exista,

$$\overline{\sigma}(V^{-1}) = \frac{1}{\underline{\sigma}(V(jw))},$$

siendo $\underline{\sigma}(\cdot)$ el menor valor singular. Luego con filtros de entrada y salida apropiados y viendo la norma $\mathcal{H}_\infty$ del conjunto se puede analizar la función $\overline{\sigma}(H(jw)), \forall w$.

Cálculo de las normas $\mathcal{H}_2$ y $\mathcal{H}_\infty$ de un sistema.

Los siguientes teoremas demostrados en [ZDG96] sirven para el cálculo de las normas $\mathcal{H}_2$ y $\mathcal{H}_\infty$.

Teorema B.4 *Sea la matriz de transferencia*

$$H(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

con A estable. La norma $\mathcal{H}_2$ de $H(s)$ es

$$\|H\|_2^2 = traza(B^* L_c B) = traza(C L_o C^*)$$

donde L_o y L_c son los grammianos de controlabilidad y observabilidad definidos por las ecuaciones de Lyapunov

$$A L_c + L_c A^* + B B^* = 0 \qquad\qquad A^* L_o + L_o A + C^* C = 0.$$

Teorema B.5 *Sea $\gamma > 0$ y una matriz de transferencia*

$$H(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

con A estable. Entonces, $\|H\|_\infty < \gamma$ si y solo si $\overline{\sigma}(D) < \gamma$ y M no tiene autovalores en el eje imaginario, donde

$$M := \begin{bmatrix} A + BR^{-1}D^*C & BR^{-1}B^* \\ -C^*(I + DR^{-1}D^*)C & -(A + BR^{-1}D^*C)^* \end{bmatrix}$$

y $R = \gamma^2 I - D^*D$.

Normas en sistemas MIMO discretos.

De forma muy parecida a como se han definidos las normas $\mathcal{H}_\infty$, $\mathcal{H}_2$ y $\mathcal{L}_1$ para sistemas continuos, se puede definir para sistemas discretos.

Definición B.15 *Sea $H(z)$ una matriz de transferencia de un sistema MIMO discreto estable. La norma $\mathcal{H}_\infty$ de $H(z)$ está definida como*

$$\|H(z)\|_\infty := \sup_{0 \leq \theta \leq \pi} \overline{\sigma}(H(e^{j\theta})).$$

Definición B.16 *Sea $H(z)$ una matriz de transferencia de un sistema MIMO discreto estable. La norma $\mathcal{H}_2$ de $H(z)$ está definida como*

$$\|H(z)\|_2 = \left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} trace[H(e^{j\theta})H(e^{-j\theta})]dw \right\}^{1/2}$$

Definición B.17 *Sea $H(z)$ una matriz de transferencia de un sistema discreto MIMO estable, de m entradas y p salidas. Sea la matriz h, cuyos términos $[h]_{ij}$ son los elementos de la convolución en tiempo discreto $y(z) = h * u$. La norma ℓ_1 de $H(z)$ viene definida como*

$$\|H(z)\|_1 = \max_{1 \leq i \leq p} \sum_{j=0}^{m} \|[h]_{ij}\|_1$$

La definición de convolución de discreto es

$$y(k) = h * u = \sum_{i=0}^{\infty} h(k-i)u(i)$$

donde $[h(k)]_{ij}$ es la salida $y_i(t), t = k$, correspondiente a una entrada impulso $u_j(t) = \delta(t), t = 0$.

Entropía de un sistema.

No es exactamente una norma pero está muy relacionado con las normas $\mathcal{H}_2$ y la $\mathcal{H}_\infty$. Para $\gamma > 0$ se define la γ-entropía de un sistema cuya matriz de transferencia es H como

$$I_\gamma(H) \doteq \begin{cases} -\frac{\gamma^2}{2\pi}\int_{-\infty}^{\infty} \log\det(I - \gamma^{-2}H(j\omega)H(j\omega)^*)d\omega & Si \quad \|H\|_\infty < \gamma \\ \infty & Si \quad \|H\|_\infty \geq \gamma \end{cases}.$$

A partir de esta definición se pueden inferir

$$\sqrt{I_\gamma(H)} \geq \|H\|_2 \geq \frac{\alpha}{\sqrt{-\log(1-\alpha^2)}}\sqrt{I_\gamma(H)},$$

donde $\alpha = \|H\|_\infty/\gamma < 1$. Si la norma $\|H\|_\infty$ tiene un valor cercano a γ, la norma $\|H\|_2$ del sistema tiene un valor cercano a la raíz cuadrada de la γ-entropía.

Apéndice C

Formas Canónicas.

C.1 Introducción.

Uno de los principales problemas en el tratamiento de los sistemas dinámicos lineales es su representación paramétrica efectiva. Un sistema planteado por medio de ecuaciones de estado $G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ tiene infinitas realizaciones. Las formas canónicas son diferentes realizaciones de un sistema dinámico lineal usando el menor número de parámetros. El número de parámetros reales independientes necesarios para definir un sistema de grado de McMillan (ver [ZDG96]) n, con m entradas y p salidas es de $n(p+m)+mp$, ver [CM97]. Este apéndice define tres formas canónicas muy conocidas, la forma canónica controlable, sección C.2, la forma canónica diagonal, sección C.3, y la balanceada, sección C.4.

C.2 Forma canónica controlable.

Un sistema de orden de McMillan n, con m entradas y p salidas puede ser representado por una forma canónica controlable tal y como se muestra en la ecuación C.1.

$$G(s)=\left[\begin{array}{ccccc|cccc} 0 & 1 & 0 & \dots & 0 & 0 & b_{1,1} & \dots & b_{1,m-1} \\ 0 & 0 & 1 & \dots & 0 & 0 & b_{2,1} & \dots & b_{2,m-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 & b_{n-1,1} & \dots & b_{n-1,m-1} \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 & 1 & b_{n,1} & \dots & b_{n,m-1} \\ \hline c_{1,1} & c_{1,2} & c_{1,3} & \dots & c_{1,n} & d_{1,1} & d_{1,2} & \dots & d_{1,m} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p,1} & c_{p,2} & c_{p,3} & \dots & c_{p,n} & d_{p,1} & d_{1,2} & \dots & d_{p,m} \end{array}\right] \tag{C.1}$$

El sistema puesto en esta forma implica que es controlable desde la primera entrada. Los parámetros a_i son los coeficientes del polinómico característico, los parámetros b_i y c_i tienen una estrecha relación con los ceros de los subsistemas entrada/salida y los parámetros d_i refleja la relación directa entrada/salida del sistema, ver [Fri89]. Existen otras formas canónicas controlable equivalentes y las llamadas formas canónicas observables que implican que el sistema sea observable desde la primera salida, ver [Fri89]. En el caso en el que el sistema tenga orden de McMillan n, una entrada $m = 1$ y una salidas $p = 1$, la ecuación (C.1) se reduce a

$$G(s)=d+\frac{c_1 s^{n-1}+\ldots+c_n}{s^n+a_1 s^{n-1}+\ldots+a_n}=\left[\begin{array}{cccc|c} 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -a_n & -a_{n-1} & \dots & -a_1 & 1 \\ \hline c_n & c_{n-1} & \dots & c_1 & d \end{array}\right]. \tag{C.2}$$

C.3 Forma canónica diagonal.

Un sistema con orden de McMillan n, con m entrada y p salidas puede ser representado por la ecuación C.3.

$$G(s) = \left[\begin{array}{ccccc|ccc} p_1 & 0 & \dots & \dots & 0 & b_{1,1} & \dots & b_{1,m} \\ 0 & \ddots & \dots & \dots & 0 & \dots & \dots & \dots \\ 0 & \dots & \begin{bmatrix} p_j & p_{j+1} \\ -p_{j+1} & p_j \end{bmatrix} & \dots & 0 & \dots & \dots & \dots \\ 0 & \dots & \dots & \ddots & 0 & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & p_n & b_{n,1} & \dots & b_{n,m} \\ \hline 1 & \dots & \dots & \dots & 1 & d_{1,1} & \dots & d_{1,m} \\ c_{1,1} & \dots & \dots & \dots & c_{1,n} & d_{2,1} & \dots & d_{2,m} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{p-1,1} & \dots & \dots & \dots & c_{p-1,n} & d_{p,1} & \dots & d_{p,m} \end{array}\right] \tag{C.3}$$

Los parámetros p_i y los $p_j \pm p_{j+1}j$ son las raíces del polinómico característico del sistema. En el caso de que haya raíces repetidas será necesario usar la forma de Jordan en vez de la forma diagonal, ver [Fri89]. Los parámetros b_i y c_i están directamente relacionados con los residuos de los subsistemas de entrada/salida y los parámetros d_i son la relación directa entre las entradas y salidas.

Para el caso concreto de un sistema con orden de McMillan n, una entrada $m = 1$ y una salida $p = 1$, la ecuación C.3 se convierte en

$$G(s) = d + \frac{b_1}{s - p_1} + \ldots + \frac{b_n}{s - p_n} = \left[\begin{array}{ccc|c} p_i & \dots & 0 & b_1 \\ 0 & \ddots & 0 & \dots \\ 0 & \dots & p_n & b_n \\ \hline 1 & \dots & 1 & d \end{array}\right]. \tag{C.4}$$

Esta forma canónica no es fija, ya que depende de si las raíces de polinómico característico son reales, dobles o complejas, lo cual puede representar un problema.

C.4 Forma canónica balanceada.

Esta forma está basada en los valores singulares de Hankel, ver [ZDG96], y en ella el Gramiano de controlabilidad del sistema es igual al Gramiano de obsevabilidad. Este hecho puede interpretarse como que existe una simetría entre la relación variables de entrada, variables de estado con la relación variables de estado, variables de salida. Esta simetría da buenos resultados a la hora de elegir esta forma canónica para identificar. Hay diversas realizaciones con esta forma de estado. En [CM97] los parámetros empleados son los siguientes:

$$\begin{array}{cccccc}
\sigma_1 & > \ldots > & \sigma_j & > \ldots > & \sigma_n & \sigma \in \mathbb{R} \\
U_1 & \ldots & U_j & \ldots & U_n & U_j \in \mathbb{R}^{p\times 1} \\
\tilde{B}_1 & \ldots & \tilde{B}_j & \ldots & \tilde{B}_n & \tilde{B}_j \in \mathbb{R}^{1\times m} \\
\tilde{D} & & & & & \tilde{D} \in \mathbb{R}^{p\times m}
\end{array}$$

donde $\sigma_n > 0$, $U_j^T U_j = 1$, $\tilde{B}_j \neq 0$ y el primer elemento distinto de cero de $\tilde{B}_j$ es positivo. La función de transferencia de un sistema estable con grado de McMillan n y valores singulares de Hankel diferentes, puede representarse por media de las matrices (A, B, C, D) como

$$\begin{gathered}
A = [a_{ij}]; \quad a_{ij} = \begin{cases} -\frac{1}{2\sigma_j}\tilde{B}_j\tilde{B}_j^T, & \text{for } i = j \\ \frac{1}{\sigma_i^2-\sigma_j^2}(\sigma_j\tilde{B}_i\tilde{B}_j^T - \sigma_i C_i^T C_j), & \text{for } i \neq j \end{cases} \\
B = \begin{bmatrix} \tilde{B}_1 \\ \vdots \\ \tilde{B}_n \end{bmatrix} \\
C = [C_1, \ldots, C_j, \ldots, C_n]; \quad C_j = U_j\sqrt{\tilde{B}_j\tilde{B}_j^T} \\
D = \tilde{D}.
\end{gathered} \tag{C.5}$$

Para un caso de múltiple salida ($p > 1$) el vector $U_j^T = [u_{j,1}, \ldots, u_{j,p}]$ puede ser parametrizado usando coordenadas polares

$$U_j = \begin{bmatrix} u_{j,q} \\ u_{j,2} \\ u_{j,3} \\ \vdots \\ u_{j,p-1} \\ u_{j,p} \end{bmatrix} = \begin{bmatrix} \cos(\phi_{j,p-1})\cos(\phi_{j,p-2})\cos(\phi_{j,p-3})\ldots\cos(\phi_{j,2})\cos(\phi_{j,1}) \\ \cos(\phi_{j,p-1})\cos(\phi_{j,p-2})\cos(\phi_{j,p-3})\ldots\cos(\phi_{j,2})\sin(\phi_{j,1}) \\ \cos(\phi_{j,p-1})\cos(\phi_{j,p-2})\cos(\phi_{j,p-3})\ldots\sin(\phi_{j,2}) \\ \vdots \\ \cos(\phi_{j,p-1})\sin(\phi_{j,p-2}) \\ \sin(\phi_{j,p-1}) \end{bmatrix}$$

para las siguientes bandas de parámetros: $-\pi/2 < \phi_{j,1} < 3\pi/2$ y $-(\pi/2) < \phi_{j,i} < \pi/2$ siendo $2 \leq i < p$. En esta parametrización, σ_j son los valores singulares de Hankel, la matriz C es una rotación de la matriz B. El caso de un sistema con n polos reales, una entrada $m = 1$ y una salida $p = 1$, el sistema puede ser definido a partir de los parámetros

$$\sigma_1 \dots \sigma_n$$
$$\tilde{b}_1 \dots \tilde{b}_n$$
$$\tilde{d}$$

siendo los parámetros $u_j = 1$. El problema de esta parametrización es la necesidad de que los sistemas representados en ella sean estable.

Apéndice D

Solución del problema $\mathcal{H}_2/\mathcal{H}_\infty$ usando inecuaciones lineales matriciales (LMIs).

El problema multiobjetivo $\mathcal{H}_2/\mathcal{H}_\infty$ es un intento de combinar las normas $\mathcal{H}_2$ y la norma $\mathcal{H}_\infty$. Son conocidas las buenas prestaciones de la norma $\mathcal{H}_2$ en la medida de las especificaciones de ante un ruido de entrada, mientras que la norma $\mathcal{H}_\infty$ consigue medir de forma efectiva la robustez del sistema, ver sección 2.3.1. Ambos requerimientos son complementarios y resulta natural su mezcla en un problema multiobjetivo.

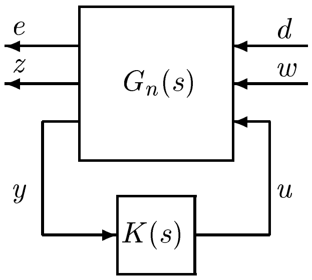


Figura D.1: El problema de control multiobjetivo $\mathcal{H}_2/\mathcal{H}_\infty$.

Si se considera el problema de control con realimentación de la figura D.1, la optimización

$\mathcal{H}_2/\mathcal{H}_\infty$ se formula de la forma siguiente: Encontrar un controlador $K(s)$ que estabilice la planta y que minimice el siguiente vector objetivo

$$\min_{K(s)} \begin{bmatrix} \|T_{zw}\|_2 \\ \|T_{ed}\|_\infty \end{bmatrix}.$$

Este problema es no convexo y no tiene una solución exacta analítica, ver sección 2.3.1. En muchos casos este problema se reduce al caso de una entrada simple $w = d$ y dos salidas d y z. La forma de resolverlo por medio de LMIs implica la transformación del problema multiobjetivo en un problema de optimización global de la función suma de ambos objetivos,

$$\alpha\|T_{zw}\|_2 + \beta\|T_{ew}\|_\infty. \tag{D.1}$$

Dada una planta generalizada:

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 u \\ z &= C_0 x + D_{01} w + D_{02} u \\ e &= C_1 x + D_{12} u \end{aligned}$$

Un controlador en realimentación el estado $u = Kx$ que minimice una banda superior del problema mixto $\mathcal{H}_2/\mathcal{H}_\infty$ se puede encontrar por la resolución del siguiente problema de optimización convexa LMI:

$$\min_{X,W,Q,\gamma} \alpha\gamma^2 + \beta\,\mathrm{trace}(Q)$$

sujeto a:

$$\begin{bmatrix} AX + XA^T + B_2W + W^TB_2 & B_1 & XC_0^T + W^TD_{02} \\ B_1^T & -I & D_{01}^T \\ C_0X + D_{02}Y & D_{01} & -\gamma^2 I \end{bmatrix} < 0$$

$$\begin{bmatrix} Q & C_1X + D_{12}W \\ XC_1^T + W^TD_{12} & X \end{bmatrix} < 0$$

Si $(X^*, W^*, Q^*, \gamma^*)$ es la solución óptima, entonces el controlador en realimentación de estado viene dado por $K = W^*(X^*)^{-1}$. Con esta solución se asegura que los límites de la norma $\mathcal{H}_2$ y $\mathcal{H}_\infty$ van a ser los siguientes

$$\|T_{ew}(s)\|_\infty \leq \gamma^*, \quad \|T_{zw}\|_2 \leq \sqrt{Trace(Q^*)}.$$

Hay muchas restricciones en los problemas lineales que pueden ser definidas en términos de LMIs, por ejemplo, el emplazamiento de los polos del sistema en lazo cerrado. Estas restricciones se pueden añadir de forma sencilla al problema, aunque su inclusión puede alejar la banda superior de la verdadera solución del problema mixto $\mathcal{H}_2/\mathcal{H}_\infty$. De forma similar se puede abordar el problema de realimentación a la salida, ver [BEGFB94].

En el paquete de MATLAB "LMI Control Toolbox", ver [GNLC95], se desarrollan dos funciones para la resolución del problema mixto $\mathcal{H}_2/\mathcal{H}_\infty$ por LMIs para el caso de problemas con realimentación de estado, `msfsyn`$(\cdot)$, y en el caso de problemas con realimentación de salida, `hinfmix`$(\cdot)$. Estas funciones son las usadas en esta tesis para comparar los resultados obtenidos en la resolución del problema mixto $\mathcal{H}_2/\mathcal{H}_\infty$ por medio del algoritmo MRCD con los límites superiores del mismo hallados por algoritmos basados en LMIs, ver sección 5.5.

Apéndice E

Formas normalizadas del controlador proporcional, integral y derivativo (PID).

E.1 Acciones de control.

Las acciones de control que un controlador PID son la suma de tres términos, la acción proporcional, la acción integral y la acción derivativa.

$$u(t) = u_P(t) + u_I(t) + u_D(t).$$

La acción proporcional u_P responde al error entre la señal referencia $r(t)$ y la señal de salida del proceso $y(t)$,

$$u_P(t) = Ke(t)$$

donde $e(t) = y(t) - r(t)$ es el error entre ambas señales. Una forma modificada de esta acción es

$$u_P(t) = K(br(t) - y(y))$$

que consigue suavizar el efecto de cambios brusco en la entrada. La acción integral sirve para controlar los sistemas con error en estado estacionario, siendo su de la forma

$$u_I(t) = k_i \int^t e(s)ds = \frac{K}{T_i} \int^t e(s)ds$$

Este efecto controla un error permanente con el mismo signo. La acción derivativa se usa como predictor, ya que reacciona ante la derivada del error,

$$u_D(t) = k_d \frac{de(t)}{dt} = KT_d \frac{de(t)}{dt}.$$

Para evitar el efecto producido por los cambios bruscos en la señal de referencia o los ruidos de la señal de salida la acción puramente derivativa se suele modificar por

$$U_D(s) = \frac{KT_d s}{1 + sT_d/N}(cr(s) - ry(s))$$

donde se incorpora un filtro para los ruidos de alta frecuencia de constante T_d/N y un parámetro para evaluar el peso de la entrada c.

Además de estas acciones de suele introducir un filtro en la señal medida para evitar ruidos

$$G_f(s) = \frac{1}{1 + sT_f}$$

E.2 Diferentes parametrizaciones de un controlador PID.

Forma normalizada ISA del PID. Teniendo en cuenta las acciones que va a llevar a cabo un PID la forma normalizada en que se puede presentar un controlador PID, también llamada forma ISA ("Instrument Society of America") es

$$U(s) = K\left(b + \frac{1}{sT_i} + c\frac{sT_d}{1 + sT_d/N}\right)R(s) + K\left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + sT_d/N}\right)Y(s) \qquad \text{(E.1)}$$

Si $b \neq 1$ o $c \neq 1$ la entrada al controlador no es el error entre la señal de referencia y la de salida, con lo que el PID tiene dos grados de libertad. Con ello se consigue que los ceros de la función de transferencia entre la señal de referencia y la de control sean diferentes a los de la función entre la señal de salida medida y la de control. Un PID normalizado tiene siete parámetros, la ganancia de control K, el tiempo integral T_i, el tiempo derivativo T_d, la máxima ganancia derivativa, los pesos de la referencia b y c y la constante de tiempo del filtro. Esta es la forma más general, que permite la existencia de ceros complejos conjugados, útiles en la optimización de sistemas oscilatorios.

Forma paralela del PID. Una pequeña modificación sobre la forma normalizada ISA es la forma paralela, que tiene como ecuación,

$$U(s) = k[bR(s) - Y(s)] + \frac{1}{sT_i}[R(s) - Y(s)] + \frac{sk_d}{1 + sT_d/N}[cR(s) - Y(s)]. \qquad \text{(E.2)}$$

Esta forma tiene como ventaja la separación clara de las tres acciones con distintas ganancias, pero en ella se pierde el concepto de tiempo de integración y tiempo derivativo. Aunque su forma matemática es la más sencilla de manejar, las constantes no tiene un significado físico interesante. Por supuesto, el valor de la constante $k = K$, $k_i = K/T_i$ y $k_d = K \cdot T_d$.

Forma serie del PID. La tercera forma corresponde al caso en el que $T_i > 4T_d$, para el que el PID tiene sólo ceros reales. En ese caso el PID se puede representar como,

$$U(s) = K' \left(1 + \frac{1}{sT_i'}\right) \left(\frac{1 + sT_d'}{1 + sT_d'\alpha}\right). \tag{E.3}$$

Esta forma es la típica en un controlador PID comercial. La razón es que en la forma (E.1) se necesita integrar la señal de control, hecho que es imposible de realizar de forma exacta con un instrumento físico. Por otro lado esta forma se asemeja a un controlador en adelanto-atraso con una constante de tiempo de retardo muy pequeña con lo que es más fácil de sintonizar, los ceros del PID son las inversos de T_i' y T_d'.

Los parámetros de un controlador serie y otro paralelo son diferentes cuando actúa el efecto derivativo. Para el caso en el que no haya filtro en la parte derivativa y $b = c = 1$ la relación entre dichos parámetros será:

$$\begin{aligned} K &= K' \tfrac{T_i' + T_d'}{T_i'} \\ T_i &= T_i' + T_d' \\ T_d &= \tfrac{T_i' T_d'}{T_i' + T_d'} \end{aligned}$$

y a la inversa

$$\begin{aligned} K' &= \tfrac{K}{2} \left(\sqrt{1 + \sqrt{1 - 4T_d/T_i}}\right) \\ T_i' &= \tfrac{T_i}{2} \left(\sqrt{1 + \sqrt{1 - 4T_d/T_i}}\right) \\ T_d' &= \tfrac{T_d}{2} \left(\sqrt{1 - \sqrt{1 - 4T_d/T_i}}\right). \end{aligned}$$

Apéndice F

Lista de símbolos

F.1 Abreviaturas

EA	Algoritmo evolutivo.
ES	Estrategias de evolución.
GA	Algoritmo Genético.
LMI	Inecuación lineal matricial.
LQG	Lineal cuadrático Gaussiano.
MIMO	Varias entradas, varias salidas.
MOP	Problema de optimización multiobjetivo.
MO	Optimización multiobjetivo
MOGA	“Multiobjective Optimization Genetic Algorithm” (nombre propio).
MOMGA	“Multiobjective messy Genetic Algorithm” (nombre propio).
NFL	“No Free Lunch” (nombre propio).
NP	No determinado polinómicamente.
NPSA	“Niche Pareto Genetic Algorithm” (nombre propio).
NSGA	“Nondominated Sorting Genetic Algorithm” (nombre propio).
PI	Proporcional-Integral
PID	Proporcional-Integral-Derivativo.
SISO	Una entrada, una salida.
SPEA	“Strength Parteto Evolutionary Algorithm” (nombre propio).
SSV	Valor singular estructurado.
VEGA	“Vector Evaluated Genetic Algorithm” (nombre propio).

F.2 Notación.

$T_{zw}(s)$	Función de transferencia entre el vector de entrada w y el de salida z.
$K, K(s)$	Controladores lineales de parámetros constantes estático y dinámicos.
Δ	Incertidumbre en el modelo de un sistema.
μ	SSV asociado a un determinado conjunto de incertidumbres.
$\mathcal{L}_1$	Espacio de Lebesgue de los vectores de funciones absolutamente integrables.
$\mathcal{L}_2$	Espacio de Lebesgue de los vectores de funciones cuadráticamente integrables.
$\mathcal{L}_\infty$	Espacio de Lebesgue de los vectores de funciones acotadas.
ℓ_1	Espacio de secuencias absolutamente sumables.
ℓ_2	Espacio de secuencias absolutamente integrables.
$\mathcal{H}_2$	Espacio de Hardy de funciones matriciales integrables cuadráticamente y analíticas en el semiplano derecho.
$\mathcal{H}_\infty$	Espacio de Hardy de las matrices de funciones acotada y analíticas en el semiplano derecho.
$\mathcal{RH}_\infty$	Subespacio de $\mathcal{H}_\infty$ formado por las matrices de funciones reales. Equivale al espacio de las matrices de transferencia estables.
$\|\cdot\|_i$	norma-i de un vector o matriz de funciones.
$\|\cdot\|_{i,j}$	norma inducida-(i,j) de una matriz de funciones.
A'	Transpuesta de la matriz A.
$\lambda(\cdot)$	Autovalores de una matriz.
$\sigma(\dot{})$	Valores singulares.
p_{cruce}	probabilidad de cruce.
p_{muta}	probabilidad de mutación.
t_{dom}	presión en la selección.

Bibliografía

[ABJ99] M.B. Anderson, J.E. Burkhalter, and R.M. Jenkins. Missile Aerodynamic Shape Optimization Using Genetic Algorithms. In *AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1999. AIAA Paper 99-0261.

[Ack93] J. Ackermann. *Robust Control, System with Uncertain Physical Parameters*. Springer-Verlag, 1993.

[AH94] K. Åström and Hägglund. *PID Controllers: Theory, Design, and Turning*. Instrument Society of America, second edition, 1994.

[AHO96] T. Arslan, D.H. Horrocks, and E. Ozdemir. Structural Synthesis of Cell-based VLSI Circuits using a Multi-Objective Genetic Algorithm. *IEE Electronic Letters*, 32(7):651–652, March 1996.

[AK89] E.H.L. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. John Wiley, Chichester, UK, first edition, 1989.

[Ala98a] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Art and Music. Technical Report 94-1-ART, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98b] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Artificial Intelligence. Technical Report 94-1-AI, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98c] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Bioscience, Including Medicine. Technical Report 94-1-BIO, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98d] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Chemistry and Physics. Technical Report 94-1-CHEMPHYS, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98e] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Computer Aided Design. Technical Report 94-1-CAD, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98f] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Computer Science. Technical Report 94-1-CS, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98g] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Control. Technical Report 94-1-CONTROL, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98h] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Databases. Technical Report 94-1-DB, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98i] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Economics and Finance. Technical Report 94-1-ECO, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98j] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Engineering. Technical Report 94-1-ENG, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98k] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Fuzzy Logic. Technical Report 94-1-FUZZY, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98l] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Immune Systems. Technical Report 94-1-IS, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98m] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Logistics. Technical Report 94-1-LOGISTICS, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98n] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Neuronal Network. Technical Report 94-1-NN, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98o] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Operation Research. Technical Report 94-1-OR, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98p] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Optics and Imagen Processing. Technical Report 94-1-OPTICS, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98q] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Optimization. Technical Report 94-1-OPTIMI, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98r] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in power Engineering. Technical Report 94-1-POWER, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98s] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Protein research. Technical Report 94-1-PROTEIN, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98t] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in Signal and Image Processing. Technical Report 94-1-SIGNAL, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98u] J.T. Alander. An Indexed Bibliography of Genetic Algorithms in VLSI desgn and Testing. Technical Report 94-1-VLSI, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[Ala98v] J.T. Alander. An Indexed Bibliography of Genetic Algorithms Papers Available via ftp and www. Technical Report 94-1-FTP, Dpt. of Information Technology, Univ. of Vaasa, 1998.

[And96] M.A. Anderson. Using Pareto Genetic Algorithms for Preliminary Subsonic Wing Design. In *6th AIAA/NASA/USAF Multidisciplinary Analysis and Optimization Symposium*, Bellevue, Washington, September 1996. AIAA Paper 96-4023.

[APH98] K.J. Åström, H. Panagopoulos, and Hägglund. Design of PI controllers based on non-convex optimization. *Automatica*, (5), 1998.

[BA98] J.L. Bernal Agustín. *Aplicación de Algoritmos Genéticos al Diseño Óptimo de Sistemas de Distribución de Energía Eléctrica*. PhD thesis, Department of Electrical Engineering, University of Zaragoza, Spain, January 1998. (In Spanish).

[Bäc94] T. Bäck. Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. In *Proceeding of the First IEEE Conference on Computational Intelligence*, pages 57–62. IEEE Congress on Computational Intelligence, 1994.

[Bäc96] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.

[Bäc97] T. et al. Bäck, editor. *Handbook of Evolutionary Computation*. IOP Publishing Ltd. and Oxford University Press, 1997.

[Bar94] J. Ross Barmish. *New Tools for Robustness of Linear Systems*. Macmillan, first edition, 1994.

[BB91] S.P. Boyd and C.H. Barratt. *Linear Controller Design. Limit of Performance*. Prentice Hall, New Jersey, 1991.

[BBM93a] D. Beaskey, D.R. Bull, and R.R. Martin. A Secuencial Niche Technique for Multimodal Function Optimization. *Evolutionary Computation*, 1:101–125, 1993.

[BBM93b] D. Beasley, D.R Bull, and R.R Martin. An Overview of Genetic Algorithms: Part 1, fundamentals. *University Computing*, 15(2):58–69, 1993.

[BBM93c] D. Beasley, D.R Bull, and R.R Martin. An Overview of Genetic Algorithms: Part 2, research topic. *University Computing*, 15(4):170–181, 1993.

[BC98] C. Belcastro and B.-C Chang. LFT Formulation for Multivariate Polynomial Ploblems. In *Proceeding of the American Control Conference*, 1998.

[BD98] C. Beck and R. D'Andrea. Computational Study and Comparisons of LFT Reducibility Methods. In *Proceedings of the American Control Conference*, pages 1013–1017, 1998.

[BEGFB94] S. Boyd, L. EL Ghaoui, E. Feron, and V. Balakrishnan. *Lineal Matrix Inequalities in System and Control Theory*. SIAM, first edition, 1994.

[Bel98] C.M. Belcastro. Parametric Uncertainty Modeling: An Overview. In *Proceedings of the American Control Conference*, 1998.

[BH89] D. Bernsein and W. Haddad. LQG Control with an $\mathcal{H}_\infty$ Performance Bound: A Riccati Equation Approach. *IEEE Transactions on Automatic Control*, 34:293–305, 1989.

[BK94] E. Baeyens and P.P. Khargonekar. Some Examples in Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control. In *Proceeding of the American Control Conference*, pages 1608–1612, 1994.

[BS94] F. Blanchini and M. Sznaier. Rational $\mathcal{L}_1$ Suboptimal Compensators for Continuous-Time Systems. *IEEE Transactions on Automatic Control*, 39:1487–1492, 1994.

[BT80] A. Ben-Tal. Characterization of Pareto and Lexicographic Optimal Solutions. In Fandel and Gal, editor, *Multiple Criteria Decision Making Theory and Application 177*, Lecture Note in Economics and Mathematical Systems, pages 1–11. Springer-Verlag, 1980.

[BT97] V. Blondel and J.N. Tsitsiklis. NP-hardness of some Lineal Control Design Problems. *SIAM Journal of Control and Optimization*, 35(6):2118–2127, 1997.

[BYDM93] R.D. Braatz, P.M. Young, J.C. Doyle, and M. Morari. Computational Complexity of μ Calculation. In *Proceeding of the American Control Conference*, pages 1682–1683, 1993.

[Cav70] D.J. Cavicchio. *Adaptive Search Using Simulated Evolution*. Unpublished doctoral dissertation, University of Michigan, Ann Arbor, 1970.

[CC98] B. Chen and Y. Cheng. A Structure-Specified H_∞ Optimal Control Design for Practical Applications: A Genetic Approach. *IEEE Transactions on Control Systems Technology*, 6(6):707–718, 1998.

[CC99] C.A. Coello Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1999. (Accepted for publication).

[CCC99] C.A. Coello Coello and Alan D. Christiansen. MOSES : A Multiobjective Optimization Tool for Engineering Design. *Engineering Optimization*, 31(3):337–368, 1999.

[CCH95] C.A. Coello, A.D. Christiansen, and A. Hernández. Multiobjective design optimization of counterweight balancing of a robot arm using genetic algorithms. In *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence*, pages 20–23, Herndon, Virginia, U.S.A., nov 1995. IEEE Computer Society Press.

[CCL95] B. Chen, Y. Cheng, and C. Lee. A Genetic Approach to Mixed H_2/H_∞ Optimal PID Control. *Control System*, pages 51–60, 1995.

[CF95] A. Chipperfield and P. Fleming. Gas Turbine Engine Controller Design Using Multiobjective Genetic Algorithms. In Conference Publication n 414, editor, *Genetic Algorithms in Engineering Systems: Innovation and Application*, pages 214–220, 1995.

[CF96] A. Chipperfield and P. Fleming. Multiobjective gas turbine engine controller design using genetic algorithms. *IEEE Transaction on Industrial Electronics*, (5), October 1996.

[CL99] A. Chritiansson and B. Lennartson. Weight selection for $\mathcal{H}_\infty$-control using genetic algorithms. In *Procceding of the 14th World Congress of IFAC*, pages 25–30, July 1999.

[CLD97] B.K.S. Cheung, A. Langevin, and H. Delmaire. Coupling Genetic Algorithm with a Grid Search Method to Solve Mixed Integer Nonlinear Programming Problems. *Computers Math. Applic.*, 34(12):13–23, 1997.

[CM97] C.T. Chou and J.M Maciejowski. System identification using balaced parametrization. *IEEE Transaction on Automatic Control*, 42(7):956–974, 1997.

[CP97] E. Cantú-Paz. A survey of Parallel Genetic Algorithms. Technical Report 95007, Illinois Genetic Algorithms Laboratory, 1997.

[CP99] J. Chen and R.J. Patton. *Robust Model- Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, first edition, 1999.

[CW95] X. Chen and J.T. Wen. A Linear Matrix Inequality Approach to the General Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control Problem. In *Proceeding of the American Control Conference*, pages 1443–1447, 1995.

[DBB94] G. Dozier, J. Bowen, and D. Bahler. Solving small and large scale constrain satisfaction problems using a heuristic -based microgeneric algorithms. In *Proceeding of the First IEEE Conference on Evolutionary Computation*, pages 306–311. IEEE World Congress on Computational Intelligence, 1994.

[DDB95] M.A. Dahleh and I.J. Diaz-Bobillo. *Control of Uncertain Systems: A linear Programming Approach*. Prentice-Hall, first edition, 1995.

[Deb89] K. Deb. *Genetic Algorithms in Multimodal Function Optimization*. Master thesis and tcga report n0. 89002, Tuscaloosa: University of Alabama, 1989.

[Deb98] K. Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Technical Report CI-49/98, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, 1998.

[Deb99] K. Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In *Proceedings of Evolutionary Algorithms in Engineering and Computer Science (EUROGEN'99)*, 1999.

[DFT92] J. C. Doyle, B. A. Francis, , and A. R. Tannenbaum. *Feedback Control Theory* . Macmillan, New York, 1992.

[DG89] K. Deb and D.E Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, 1989.

[DG91] K. Deb and D.E. Goldberg. A Messy Genetic Algorithm in C. Technical Report 91008, Illinois Genetic Algorithms Laboratory, 1991.

[DGKF89] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State Space Solution to Standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control Problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[DJ75] K.A. De Jong. *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*. University microfilms no. 76-9381, Doctoral dissertation, University of Michigan, 1975.

[DP87a] M.A. Dahleh and J.B. Pearson. ℓ_1-Optimal Feedback Controller for MIMO Discrete Time System. *IEEE Transactions on Automatic Control*, 32:314–322, 1987.

[DP87b] M.A. Dahleh and J.B. Pearson. $\mathcal{L}_1$-Optimal Feedback Controller for MIMO Continuous Time System. *IEEE Transactions on Automatic Control*, 32:889–895, 1987.

[DPZ91] J. Doyle, A. Packard, and K. Zhou. Review of LFTs, LMIs, and μ. In *Proceeding of the 30th Conference on Decision and Control*, Grighton, England, 1991. IEEE.

[DWC95] N.V. Dakev, J.F. Whidborne, and A.J. Chipperfield. $\mathcal{H}_\infty$ Design of an EMS Control System for a Maglev Vehicle Using Evolutionary Algorithms. In *Genetic Algorithms in Engineering Systems: Innovations and Applications*, , September 1995. IEE.

[DZGB94] J. Doyle, K. Zhou, K. Glover, and B. Bodenheimer. Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Performance Objectives I: Optimal Control. *IEEE Transactions on Automatic Control*, 39:1575–1587, 1994.

[ED96] N. Elia and Dahleh. ℓ_1 Minimization with Magnitude Constraints in the Frequency Domain. In *IFAC World Congress*, volume C, pages 397–402, 1996.

[EDDB93] N. Elia, M.A. Dahleh, and I.J. Diaz-Bobillo. Controller Design via Infinite-Dimensional Linear Programming. In *Proceeding of the American Control Conference*, pages 2165–2169, 1993.

[FF93] C.M. Fonseca and P.J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceeding of the Fifth International Conference in GA*, San Mateo, July 1993. CA: Morgan Kaufmann.

[FF95a] C.M. Fonseca and P.J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[FF95b] C.M. Fonseca and P.J. Fleming. Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mate Restriction. In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems, Innovations and Applications, GALESIA*, pages 45–52, University of Sheffield, UK, Sept 1995. IEE.

[FF98a] C.M. Fonseca and P.J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):26–37, 1998.

[FF98b] C.M. Fonseca and P.J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part II: A Application Example. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):38–47, 1998.

[Fou85] M.P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Grefenstette*, pages 141–153. , 1985.

[FOW66] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial Intelligence Through Simulated Evolution*. John Wiley, Chichester, UK, first edition, 1966.

[Fra87] B.A. Francis. *A Course in $\mathcal{H}_\infty$ Control Theory*. Springer-Verlag, first edition, 1987.

[Fri89] B. Friedland. *Control System Desing: An Introduction to State-Spaece Methods*. McGraw-Hill, first edition, 1989.

[Fri96] B. Friedland. *Advanced Control System Design*. Prentice Hall, first edition, 1996.

[FZL99] M.S. Fadali, Y. Zhang, and S.J. Louis. Robust stability analisis of discrete-time systems using genetic algorithms. *IEEE Transaction on Sytems, Man and Cybernetics,- Part A: Systems and Humans*, (5), Septembre 1999.

[GD91] D.E. Goldberg and K. Deb. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In G. RAWLINS, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, SAN MATEO, 1991.

[GDH92] D.E. Goldberg, K. Deb, and J. Horn. Massive Multimodality, Deception, and Genetic Algorithms. In R. Männer and B. Manderick, editors, *Parallel Problem solving from Nature*, pages 37–46. Amsterdam: Elsevier, 1992.

[GGM96] S.J. Gih, D.W. Gu, and K.F. Man. Multi-Layer Genetic Algorithms in Robust Control System Design. In *UKACC International Conference on Control*, page . Conference Publication n 427 IEE, September 1996.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Bell Laboratories. Murray Hill, New Jersey, first edition, 1979.

[Glo89] F. Glover. Tabu Search-Part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[Glo90] F. Glover. Tabu Search-Part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[GNLC95] P. Gahinet, A.Ñemirovski, A. Laub, and M. Chilali. *LMI Control Toolbox*. The MathWorks, Inc. , 1995.

[Gol89a] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[Gol89b] D.E. Goldberg. Messy genetic algorithms: Motivation, analysis, and fist result. *Complex Systems*, pages 493–530, 1989.

[Gol90] D.E. Goldberg. A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing. *Complex Systems*, 4:445–460, 1990.

[GP97] J.R. Gremling and K.M. Passino. Genetic Adaptive Failure Estimation. In *Proceeding of the American Control Conference.*, pages 908–912, Albuquerque, June 1997. AACC.

[GR87] D.E. Goldberg and J.J. Richardson. Genetic Algorithms with sharing for multimodal function optimization. In *Genetic Algorithms and their Applications: Proceeding of the Second ICGA*, Hillsdale, 1987. Lawrence Erlbaum Associates.

[HC97] W. Haddad and V.S. Chellaboina. Mixed-Norm $\mathcal{H}_2/\mathcal{L}_1$ Controller Synthesis via Fixed-Order Dynamic Compensation: A Riccati Equation Approach. In *Proceeding of the 36rd Conference on Decision and Control*, pages 452–457, 1997.

[HLBLPG99] 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*, pages 131–132, Orlando, Florida, July 1999.

[HLBLPG00] A. Herreros López, E. Baeyens Lázaro, and J.R. Perán Gonzalez. Design of pid Controllers Using Multiobjective Genetic Algorithms. In *Proceedings of the 2000 IFAC Workshop on Digital Control. Past, present and Future of PID Control.*, pages 309–315, Terrassa, Spain, April 2000.

[HN93] J. Horn and N.Ñafpliotis. Multiobjective Optimization Using the Niche Pareto Genetic Algorithm. Technical Report 93005, Illinois Genetic Algorithms Laboratory, 1993.

[HNG94] J. Horn, N.Ñafpliotis, and D.E. Goldberg. A Niche Pareto Algorithm for Multiobjective Optimization. In *First IEEE Conference on Evolutionary Computation*. IEEE World Congress on Computational Intelligence, 1994.

[Hol75] J. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, first edition, 1975.

[Hol92] J. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, second edition, 1992.

[Hor97] J. Horn. Multicriterion Decision making. In Thomas Bäck, David Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, volume 1, pages F1.9:1 – F1.9:15. IOP Publishing Ltd. and Oxford University Press, 1997.

[HYKK99] K. Hirata, Y. Yanase, T. Katayama, and T. Kawabe. A min-max design of robust i-pd controller for time-delay systems with parametric uncertainty. In *Proceeding of the 14th World Congress of IFAC*, pages 259–264, July 1999.

[IM98] H. Ishibuchi and T. Murata. Multi-Objective Genetic Local Search Algorithm and Its Application to Flowship Scheduling. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):392–403, August 1998.

[JL96] I.K. Jeong and J.J. Lee. Adaptive Simulated Annealing Genetic Algorithms for Control Applications. *International Journal of Systems Science*, 27(2):241–253, 1996.

[Kal60] R.E. Kalman. Contributions to the Theory of Optimal Control. *Bulletin Society of Mathematics Mexico*, 5:102–199, 1960.

[Kar91] C.L. Karr. Design of an adaptive fuzzy logic controller using a GA. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 450–457, 1991.

[KB97] L. H. Keel and S. P. Bhattacharyya. Robust, Fragile, or Optimal. *IEEE Transactions on Automatic Control*, 42(8):1098–1105, 1997.

[KC98] H. Kwatny and B.-C. Chang. Constructing Linear Families from Parameter-Dependent Nonlinear Dynamics. *IEEE Transactions on Automatic Control*, 43(8):1143–1147, 1998.

[KCC99] R.A. Krohling, L.S. Coelho, and A.A.R. Coelho. Evolutionary strategies for synthesis of mixed $\mathcal{H}_2/\mathcal{H}_\infty$ fixed-structure controllers. In *Proceeding of the 14th World Congress of IFAC*, pages 471–476, July 1999.

[KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by Simulated annealing. *Science*, 220:671–680, 1983.

[KK96] S. Kundu and S. Kawata. Genetic algorithms for optimal feedback control design. *Engng. Applic Artif. Intell.*, (4):403–411, 1996.

[KKR93] I. Kaminer, P.P. Khargonekar, and M.A. Rotea. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control for Discrete-Time Systems via Convex Optimization. *Automatica*, 29:57–70, 1993.

[KNM95] S. Kwong, A.Ñg, and K.F. Man. Improving Local Search in Genetic Algorithms for Numerical Global Optimization Using Modified GRID-point Search Technique. In *Genetic Algorithms in Engineering Systems: Innovations and Applications*, , September 1995. IEE.

[Koz92] J.R. Koza. *Genetic Programming.* MIT Press, first edition, 1992.

[KR91] P.P. Khargonekar and M.A. Rotea. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: A Convex Optimization Approach. *IEEE Transactions on Automatic Control*, 36:824–837, 1991.

[KTK96] T. Kawabe, T. Tagami, and T. Katayama. A Genetic Algorithm Based MinMax Optimal Design of Robust I-PD Controller. In *UKACC International Conference on Control*, pages 436–441. Conference Publication n 427, IEE, September 1996.

[LA87] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated Annealing: Theory and Applications*. D.Reidel, Dordreht, Holland, first edition, 1987.

[LP96] G.P. Liu and R.J. Patton. Robust Control Design Using Eigenstructure Assignment and Multiobjective Optimization. *International Journal of Systems Science*, 27(9):871–879, 1996.

[LZD91] W.M. Lu, K. Zhou, and J.C. Doyle. Stabilization of LFT Systems. In *Proceeding of the 30th Conference on Decision and Control*, Grighton, England, 1991. IEEE.

[LZD96] W.M. Lu, K. Zhou, and C. Doyle. Stabilization of Uncertain Linear Systems: An LFT Approach. *IEEE Transactions on Automatic Control*, 41(1):50–64, 1996.

[Mah93a] S.W. Mahfoud. Crowding and Preselection revisited. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature*, pages 27–36. Amsterdam: Elsevier, 1993.

[Mah93b] S.W. Mahfound. An Analysis of Boltzmann Tournament Selection. *Complex Systems*, 7(2):155–170, 1993.

[Mah94] S.W. Mahfoud. Crossover Interactions Among Niches. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 188–193, 1994.

[Mah95] S.W. Mahfound. *Niching Method for Genetic Algorithms*. PhD thesis, Illinois Genetic Algorithms Laboratory, 1995.

[Mar95] C.I. Marrison. *The Design of Control Laws for Uncertain Dynamic Systems Using Stochastic Robustness Metrics*. PhD thesis, Priceton University, 1995.

[MB98] S. Mullei and P. Beling. Hybrid Evolutionary Algorithms for a Multiobjective Financial Problem. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3925–3930. IEEE, October 1998.

[Meg94] A. Megretski. On the Order of the Optimal Controller in the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control. In *Proceeding of the 33rd Conference on Decision and Control*, pages 1358–1362, 1994.

[MGL91] D. Mustafa, K. Glover, and D.J.N. Limebeer. Solution of the $\mathcal{H}_\infty$ Distance Problem which Minimize an Entropy Integral. *Automatica*, 27:193–199, 1991.

[Mic92] Z. Michalewicz. *Genetic Algorithms + Data Structures= Evolution Programs*. Springer-Verlarg, third edition, 1992.

[Mit99] M. Mitchell. *An Introduction to Genetic Algorithm*. MIT press, fitht edition, 1999.

[MS94] C.I. Marrison and R.F. Stengel. The Use of Random Search and Genetic Algorithms to Optimize Stochastic Robustness Functions. In *Proceeding of the American Control Conference*, pages 1484–1489, 1994.

[MS95] C.J. Marrison and R.F. Stengel. Stochastic Robustness Synthesis Applied to a Benchmark Control Problem. *International Journal of Robust and Nonlinear Control*, 5:13–31, 1995.

[MTK96] K.F Man, K.S. Tang, and S. Kwong. Genetic Algorithms: concept and applications. *IEEE Trans. Industrial Electronics*, 43(5):519–534, 1996.

[MTKH97] K.F. Man, K.S. Tang, S. Kwong, and W. A. Halang. *Genetic Algorithms for Control and Signal Processing*. Springer, first edition, 1997.

[Nem93] A.Ñemirovskii. Several NP-hard Problem Arising in Robust Stability Analisys. *Mathematic of Control, Signal and System*, 6(2):99–105, 1993.

[Oba98] S. Obayashi. Multidisciplinary Design Optimization of Aircraft Wing Planform Based on Evolutionary Algorithms. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, La Jolla, California, October 1998. IEEE.

[OGC91] C.K. Oei, D.E. Goldberg, and S.J. Chang. Tournament Selection, Niching and The Preservation of the diversity. Technical Report 91011, Illinois Genetic Algorithms Laboratory, 1991.

[OR94] B.A. Ogunnaike and W.H. Ray. *Process Dynamics, Modeling, and Control*. Oxford University Press, New York, Oxford, 1994.

[Pag99a] F. Paganini. Convex Methods for Robust $\mathcal{H}_2$ Analysis of Continuous Time System. *Transactions on Automatic Control*, 44(2):239–253, 1999.

[Pag99b] F. Paganini. Frequency Domain Conditions for Robust $\mathcal{H}_2$ Performance. *Transactions on Automatic Control*, 44(1):38–50, 1999.

[Pap91] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc., third edition, 1991.

[Ray91] L.R. Ray. *Stochastic Robustness of Linear Multivariable Control System: Toward Comprensive Robustness Analysis*. PhD thesis, Pricenton University, 1991.

[RK91] M.A. Rotea and P.R. Khargonekar. $\mathcal{H}_2$ Control with an $\mathcal{H}_\infty$ Constrains: The State Feedback Case. *Automatica*, 27:307–316, 1991.

[Rot90] M. Rotea. *Multiple Objetive and Robust Control for Lineal Systems*. PhD thesis, Faculty of the Garduate School of the University of Minnesota, 1990.

[RPLH89] J.T. Richardson, M.R. Palmer, G. Liepins, and M. Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–197. Morgan-Kauffman, 1989.

[RS93] L.R. Ray and R.F. Stengel. A Monte Carlo Approach to the Analysis of Control System Robustness. *Automatica*, 29(1):229–236, 1993.

[RS98] H. Rotstein and M. Sznaier. An Exact Solution to General Four Block Discrete Time Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Problems via Convex Optimization. *IEEE Transactions on Automatic Control*, 43(10):1475–1480, 1998.

[RS99] H. Rotstein and B. Shklyar. Optimal Solution and Approximation to the $\ell_1/\mathcal{H}_\infty$ Control Problem. *IEEE Transactions on Automatic Control*, 44(6):1240–1243, 1999.

[Rub81] R. Rubinstein. *Simulation and the Monte Carlo Methods*. Wiley, New York, 1981.

[Rub92] R. Rubinstein. *Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks*. Krieger, Malabar. Florida, 1992.

[Rud98] G. Rudolph. On a multi-objetive evolutionary algorithm and its convergence to the pareto front. In *Proceeding of the 1998 IEEE Concerence on Evolutionary Computation*, 1998.

[RVDS92] D.B. Ridgely, L.S. Valavani, M.A. Dahlah, and G. Stein. Solutions of the General Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control Problem-Necessary Condition for Optimality. In *Proceeding of the American Control Conference*, pages 1348–1352, 1992.

[RVM96] J. Rowe, K. Vinsen, and N. Marvin. Parallel Genetic Algorithms for Multiobjective Function. In *Proceeding of the Second Nordic Workshop on Genetic Algorithms and their Application*, Vaasa, 1996. University of Vaasa.

[SB94] M. Sznaier and F. Blanchini. Mixed $\mathcal{L}_1/\mathcal{H}_\infty$ Controllers for MIMO Continuous Time Systems. In *IEEE Conference Decision Control*, volume 3, pages 2702–2707, 1994.

[Sch81] H.P. Schwefel. *Numerical Optimization for Computer Models*. John Wiley, Chichester, UK, first edition, 1981.

[Sch84] J.D. Schaffer. *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*. Unpublished doctoral dissertation, Vanderbilt University, Nashville, 1984.

[Sch85] J.D. Schaffer. Multiple Objective Optimization with Vector Evaluation Genetic Algorithms. In *Proceeding of an International Conference on Genetic Algorithms and their Applications*, pages 93–100. J. Grefenstette, ed, 1985.

[Sch95] H.P. Schwefel. *Evolution and Optimum Seeking*. John Wiley & Sons, Inc, New York, first edition, 1995.

[SDV95] M.V. Salapaka, M. Dahleh, and P. Voulgaris. Mixed Objective Control Synthesis: Optimal $\ell_1/\mathcal{H}_2$ Control. In *Proceeding of the American Control Conference*, pages 1438–1442, 1995.

[SDV98] M.V. Salapaka, M. Dahleh, and P.G Voulgaris. MIMO Optimal Control Design: The Interplay Between the $\mathcal{H}_2$ and the ℓ_1 Norms. *IEEE Transactions on Automatic Control*, 43(10):1374–1388, 1998.

[SF96] K.J. Shaw and P.J. Fleming. An Initial Study of Practical Multi-Objective Production Scheduling using Genetic Algorithms. In *Proceedings of the International Conference on Control'96*, University of Exeter, UK, September 1996.

[SGC97] C. Scherer, P. Gahinet, and M. Chilali. Multiobjective Output Feedback Control via LMI Optimization. *IEEE Transactions on Automatic Control*, 42(7):896–911, 1997.

[SHB95] M. Sznaier, M. Holmes, and J. Bu. Mixed $\mathcal{H}_2/\mathcal{L}_1$ Control with Low Order Controllers: A Linear Matrix Inequality Approach. In *Proceeding of the 34rd Conference on Decision and Control*, pages 1352–1357, 1995.

[Shi94] D. Shim. Mixed $\mathcal{H}_2/ESPR$ (Extended Strictly Positive Real) Control for State Feedback Case. In *Proceeding of the 33rd Conference on Decision and Control*, pages 3175–3180, 1994.

[SKF99] V. Silva, W. Khatib, and P. Fleming. Performance optimization of gas turbine engines using the stuga. In *Proceeding of the 14th World Congress of IFAC*, pages 367–372, July 1999.

[SP94] M. Srinivas and L.M. Patnaik. Genetic Algorithms: A survey. *Computer*, pages 17–26, 1994.

[Spe94] W.M. Spears. Simple Subpopulation Schemes. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 296–307, 1994.

[SRBS97] M. Sznaier, H. Rotsein, J. Bu, and A. Sideris. An Exact Solution To Continuous Time Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control. In *Proceeding of the American Control Conference*, pages 3500–3504, 1997.

[Sys89] G. Syswerda. Uniform Crossover in Genetic Algorithms. In *Proceeding of the Third International Concerence on Genetic Algorithms*, pages 2–9, 1989.

[Szn94a] M. Sznaier. An Exact Solution to The General 4-Block Discrete Time Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Problems via Convex Optimization. In *Proceeding of American Control Conference*, pages 2251–2256, 1994.

[Szn94b] M. Sznaier. Mixed $\ell_1/\mathcal{H}_\infty$ Controllers for MINO Discrete Time Systems. In *Proceeding of the 33rd Conference on Decision and Control*, pages 3187–3192, 1994.

[Tan89] R. Tanese. Distributed Genetic Algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 434–439, 1989.

[TB97] L.N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.

[TBD96] R. Tempo, E.W. Bai, and F. Dabbene. Probabilistic Robustness Analysis: Explicit Bound for the Minimum Number of Samples. In *Proceeding of the 35th Conference on Decision and Control*, pages 3424–3428, 1996.

[The97] The MathWorks, Inc. . *Matlab Reference Guide*, 1997.

[Tim84] G.T. Timmer. *Global Optimization: A Stochastic Approach*. Centrum Voor Wiskunde en Informatica. Amsterdam, first edition, 1984.

[TLMSS95] K.C. Tan, Y. Li, D.J. Murray-Smith, and K.C. Sharman. System Identification and Linearisarion Using Genetic Algorithms with Simulated Anneling. In *Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 164–169. IEE, September 1995.

[TMG96] K.S. Tang, K.F Man, and D.W. Gu. Structured Genetic Algorithm for Robust $\mathcal{H}_\infty$ Control Systems Design. *IEEE Trans. Industrial Electronics*, 43(5):575–582, 1996.

[Vid86] M. Vidyasagar. Optimal Rejection of Persistent Bounded Disturbances. *IEEE Transactions on Automatic Control*, 31:527–535, 1986.

[Vid97] M. Vidyasagar. *A Theory of Learning and Generalization*. Springer-Verlag, London, 1997.

[Vid98] M. Vidyasagar. Statistical Learning Theory and Randomized Algorithms for Control. *Control System*, 18(6):69–85, 1998.

[VJ97] B. Vroemen and B. Jager. Multiobjective control: An overview. In *Proceeding of the 36th Conference on Decision and Control*, San Diego, California, 1997. IEEE.

[Vos99] M.D. Vose. *The simple Genetic Algorithm*. MIT press, first edition, 1999.

[Vou94] P. Voulgaris. Optimal $\mathcal{H}_2/\ell_1$ Control: The SISO Case. In *Proceeding of the 33rd Conference on Control and Decision*, pages 3181–3186, 1994.

[VQ97] A. Vicini and D. Quagliarella. Inverse and Direct Airfoil Design Using a Multiobjective Genetic Algorithm. *AIAA Journal*, 35(9):1499–1505, September 1997.

[VV99a] D. Van Veldhuizen. *Multiobjetive Evolutionary Algorithms: Classification, Analyses, and New Innovations*. PhD thesis, Air University, 1999. AFIT/DS/ENG/99-01.

[VV99b] D.A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[VVL98] D.A. Van Veldhuizen and G.B. Lamont. Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.

[WB92] B. Wie and D.S. Bernstein. Benchmark Problems for Robust Control Design. *American Institute of Aeronautics and Astronautics*, 15(5):1057–1059, 1992.

[WDBG91] W. Wang, J. Doyle, C. Beck, and K. Glover. Model Reduction of LFT Systems. In *Proceeding of the 30th Conference on Decision and Control*, Grighton, England, 1991. IEEE.

[WG96] J.F. Whidborne and I. Gu, D.W.and Postlethwaite. Simulated Annealing for Multiobjective Control System Design. In *UKACC International Conference on Control*, pages 376–381. Conference Publication n 427, IEE, September 1996.

[WM97] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transaction on Evolutionary Computation*, (1), 1997.

[WR92] S.R. Wells and D.B. Ridgely. Using Increased Order Controllers in Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Optimization. In *Proceeding of the American Control Conference*, pages 1358–1362, 1992.

[WR95] D.E. Walker and D.B. Ridgely. Uniqueness of the General Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Optimal. In *Proceeding of the American Control Conference*, pages 1453–1457, 1995.

[YD95] P.M Young and M.A. Dahleh. Controller Design for Mixed ℓ_1/LMI Performance Objectives. In *Proceeding of the American Control Conference*, pages 1433–1437, 1995.

[YG93] X. Yin and N. Germay. A Fast Genetic Algorithm with Sharing Scheme Using Cluster Analysis in Multimodal Function Optimization. In R.F ALbrecht, C.S. Reeves, and N.C Steele, editors, *Artificial Neuronal Nets and Genetic Algorithms: Proceedings of the International Conference in Insbruck*, pages 450–457. Berlin:Springer-Verlag, 1993.

[YND91] P.M. Young, M.P. Newlin, and J.C. Doyle. Mu Analysis with Real Parametric Uncertainty. In *Proceeding of the 30th Conference on Decision and Control*, Grighton, England, 1991. IEEE.

[YS97] I. Yaesh and U. Shaked. Minimum Entropy Static Output Feedback Control with an $\mathcal{H}_\infty$ Norm Performance Bound. *IEEE Transactions on Automatic Control*, 42(6):853–858, 1997.

[Zam81] G. Zames. Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms and Approximate Inverses. *IEEE Transactions on Automatic Control*, 26(4):301–320, 1981.

[ZDG96] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control* . Prentice Hall, New Jersey, 1996.

[ZDT99] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Technical Report 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, February 1999.

[ZGG+94] K. Zhou, Glover, K. Glover, B. Bodenheimer, and J. Doyle. Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Performance Objectives I: Robust Control. *IEEE Transactions on Automatic Control*, 39(8):1564–1574, 1994.

[ZHD97] X. Zhu, Y. Huang, and J. Doyle. Genetic Algorithms and Simulation Annealing for Robustness Analysis. In *Proceeding of the American Control Conference.*, pages 3756–3760, Albuquerque, June 1997. AACC.

[ZT98a] E. Zitzler and L. Thiele. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 1998.

[ZT98b] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.

[ZT99] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.