

ALGORITMOS GENÉTICOS APLICADOS AL DISEÑO DE CONTROLADORES ROBUSTOS¹

Miguel A. Martínez, Javier Sanchis,
Xavier Blasco

*Grupo de Control Predictivo y Optimización Heurística
(CPOH)*

*Departamento de Ingeniería de Sistemas y Automática
Universidad Politécnica de Valencia*

Camino de Vera 14, 46022 - Valencia, España

e-mail: mmiranzo@isa.upv.es <http://ctl-predictivo.upv.es>

Resumen: La estrategia de optimización multiobjetivo denominada Programación Física (PP)* permite al diseñador expresar sus preferencias explícitamente para cada objetivo de diseño (tiempo de establecimiento, estabilidad, etc.) de una forma flexible y con un claro significado "físico". Estas preferencias se formulan a través de categorías del tipo *deseable, tolerable, inaceptable, etc.* asociadas a unos rangos numéricos que el diseñador fija para cada especificación. En este artículo se muestra cómo se puede plantear un problema de control robusto como un problema de optimización multiobjetivo y cómo se puede utilizar PP con Algoritmos Genéticos (AGs) para salvar los problemas que presenta esta técnica frente a funciones con numerosos mínimos locales. En el artículo se resuelve el *problema tipo* para control robusto de un proceso de masa-muelle y se comparan las soluciones a este problema desarrolladas por otros autores con las obtenidas empleando PP y AGs. *Copyright ©2006 CEA-IFAC.*

* Conocida en el mundo anglosajón como *Physical Programming* (PP)

Palabras Clave: Control Robusto, Optimización Multiobjetivo, Programación Física, Algoritmos Genéticos

1. INTRODUCCIÓN

Generalmente, el diseño de un controlador se diseña para cumplir un conjunto de especificaciones que suelen entrar en conflicto. Por ejemplo, conseguir que un controlador sea robusto ante variaciones del modelo limita la consecución de altas prestaciones, o si se pretende limitar las acciones de control se limitan también las prestaciones

dinámicas que se pueden conseguir. Así pues, el diseño de un controlador puede entenderse como la búsqueda del mejor compromiso entre todas las especificaciones exigibles, por lo que la optimización multiobjetivo (OM) es una alternativa razonable para el diseño de controladores.

En este contexto de optimización multiobjetivo no existe una solución ideal para todos los objetivos, pues como se ha dicho suelen entrar en conflicto, apareciendo los conceptos de conjunto de Pareto, el cual engloba al conjunto de buenas soluciones (ninguna de ellas domina sobre las demás),

¹ Financiado parcialmente por los proyectos de investigación del MEC: FEDER DPI2004-8383-C03-02 y DPI2005-07835

y frente de Pareto, determinado por los valores de las funciones a optimizar para los valores del conjunto de Pareto. Existen diversos algoritmos que tratan y resuelven este tipo de problemas (Miettinen, 1998), (Coello *et al.*, 2002), (Toscano and Coello, 2003). Una vez localizado el frente de Pareto, el diseñador debe seleccionar una de las soluciones de ese conjunto atendiendo a sus preferencias, este paso es subjetivo y no es sencillo. Los algoritmos llamados de *Toma de Decisión*² (TD) tratan de resolver este problema. Una forma tradicional y simple de resolver la OM (incluyendo la fase de decisión de la TD) pasa por realizar una suma ponderada de los objetivos convirtiendo el problema en una optimización de un sólo objetivo. El problema fundamental que aparece es el ajuste de los factores de ponderación que se resuelve, costosamente, mediante prueba y error y donde no siempre el diseñador es capaz de trasladar sus conocimientos o preferencias, resultando, generalmente una solución subóptima.

En este último caso, plantear las especificaciones de diseño en un lenguaje entendible por el diseñador, evitando seleccionar factores de peso sin un sentido físico, es el objetivo de una nueva metodología denominada PP (Messac, 1996). Usando esta nueva forma de plantear problemas de optimización multiobjetivo, el diseñador puede especificar sus preferencias respecto a los objetivos que quiere alcanzar, como por ejemplo: tiempo de establecimiento, máxima acción de control, etc. de una forma natural y flexible. Estas preferencias se fijan a través de las llamadas Funciones de Clase y rangos de pertenencia. Se consigue con ello que el planteamiento del problema de TD sea transparente al diseñador que sólo se tiene que preocupar de disponer de los algoritmos de cálculo de las especificaciones, de definir los rangos de preferencia de las mismas y de aplicar algún programa de optimización para la función objetivo.

Se podría argumentar que también con PP es necesario una labor de ajuste de parámetros o preferencias y desde ese punto de vista ser necesario realizar un trabajo similar al empleado en los algoritmos de suma ponderada. Sin embargo, en el caso de PP no se procede a un ajuste heurístico de prueba y error que no tiene en cuenta el sentido físico del problema, sino que se procede a una elección meditada y entendible por el diseñador, convirtiéndose dichos parámetros en parámetros de diseño.

Así pues, en este artículo se va a presentar inicialmente, en la sección 2, la técnica denominada PP y los conceptos asociados a las denominadas Funciones de Clase, que permiten al diseñador expresar sus preferencias en las magnitudes "físicamente"

entendibles. A continuación, en la sección 3, se presentarán los AGs como técnica de optimización no lineal seleccionada poniendo de manifiesto las ventajas de su utilización. Posteriormente, en la sección 4, se describirá el *problema tipo* para control robusto propuesto en la Conferencia Americana de Control (ACC) y se enumeran las especificaciones para dicho *problema tipo*. Por último, en la sección 5, se presentarán los resultados obtenidos que mejoran los propuestos originalmente, incluso aquellos que se deducen por aplicación de PP, pero con otras técnicas de optimización numéricas no lineales.

2. PROGRAMACIÓN FÍSICA

PP es una forma de plantear los problemas de optimización multiobjetivo incorporando información directa del problema a resolver en la fase de toma de decisiones. Una forma sencilla y ampliamente utilizada para resolver la optimización multiobjetivo es la formulación de un índice, normalmente cuadrático, con factores de peso para cada elemento que compone el índice. Esta formulación es dependiente de la elección de los factores de peso y de dicha elección dependerán los resultados obtenidos. En una fase posterior, el diseñador normalmente mediante un proceso de prueba y error modificará los factores de peso hasta obtener una solución considerada como "satisfactoria".

Para salvar los anteriores inconvenientes, el método de PP introduce una serie de novedades en la formulación del problema de optimización, tratando de incorporar el conocimiento que el diseñador tiene sobre las variables físicas que definen el problema o los valores deseados para las especificaciones que ha de cumplir la solución, de manera que orienten al proceso o algoritmo de optimización a una correcta selección de la solución. En una primera fase, el método PP traduce el conocimiento del diseñador sobre las variables, o especificaciones que definen un problema a una serie de funciones de clase, previamente establecidas, trasladando éste a un espacio de decisión estándar, donde todas las variables de clase están normalizadas y son independientes del problema inicial. En una segunda fase, y mediante algún método de optimización adecuado, se optimiza una función agregada de todas las funciones de clase presentes en el problema, las cuales dependen originalmente de las variables físicas o especificaciones. Con la función agregada se consigue convertir un problema multiobjetivo en un problema con un sólo objetivo.

Puesto que en el proceso de aplicación de PP estará envuelto un algoritmo de optimización de una función, en general, no lineal y sujeta a restricciones, se deberán tomar las precauciones necesarias

² *Decision Maker (DM)* en su expresión anglosajona.

para asegurar que PP sea lo suficientemente robusto y evitar quedar atrapado en mínimos locales, en el caso de que éstos existan. Los AGs, los cuales han demostrado su validez para solucionar problemas de optimización global (P.J. Fleming, 2002), serán empleados en este trabajo.

2.1 Funciones de Clase.

Siendo \mathbf{x} el vector de parámetros a optimizar, el diseñador debe decidir qué especificaciones desea aplicar definidas mediante las funciones $g_i(\mathbf{x})$, donde el índice i representa a cada especificación.

A partir de cada especificación $g_i(\mathbf{x})$ se definirá su Función de Clase $\bar{g}_i(g_i(\mathbf{x}))$ que representará, según la forma de dicha relación (entendido como curvatura), las preferencias del diseñador respecto a la especificación asociada.

Estas Funciones de Clase podrán ser *duras* o *suaves*, dependiendo de la forma que se asigne a la relación con la especificación correspondiente. La lista de Funciones de Clase a definir es la siguiente:

- Clases suaves:
 - Clase-1s: Más pequeño es mejor.
 - Clase-2s: Más grande es mejor.
 - Clase-3s: Un valor es mejor.
 - Clase-4s: Un rango es mejor.
- Clases duras:
 - Clase-1h: Menor de un valor.
 - Clase-2h: Mayor de un valor.
 - Clase-3h: Igual a un valor.
 - Clase-4h: Debe estar en el rango.

Elegida la Función de Clase para cada especificación, el diseñador seleccionará unos valores de g_i ($g_i^1 \dots g_i^5$), por ejemplo para la Clase-1s, determinando así los rangos de pertenencia de la variable de clase:

- *altamente deseable (AD)*
- *deseable (D)*
- *tolerable (T)*
- *indeseable (IND)*
- *altamente indeseable (AIND)*
- *inaceptable (INA)*

Estos rangos de pertenencia estarán definidos en las unidades físicas de la especificación a la que hacen referencia, por lo que el diseñador puede expresar sus preferencias de una forma más natural e intuitiva³.

³ El número de rangos de pertenencia a definir en un problema PP es tradicionalmente el que se presenta en este artículo. No obstante, es evidente que se podría plantear, con la misma metodología, modificaciones con un mayor o menor número de rangos adaptados a los deseos del diseñador.

Sin pérdida de generalidad⁴, se presentan en las figuras 1 y 2 ejemplos numéricos de las clases suaves en donde el diseñador ha expresado sus preferencias para cada una de las especificaciones g_i del problema. En concreto, la clase 1s de la figura 1 podría corresponder a una especificación de sobreoscilación como sigue:

- altamente deseable (AD):
 $\delta \leq 10\%$
- deseable (D):
 $10\% \leq \delta \leq 20\%$
- tolerable (T):
 $20\% \leq \delta \leq 30\%$
- indeseable (IND):
 $30\% \leq \delta \leq 40\%$
- altamente indeseable (AIND):
 $40\% \leq \delta \leq 50\%$
- inaceptable (INA):
 $\delta \geq 50\%$

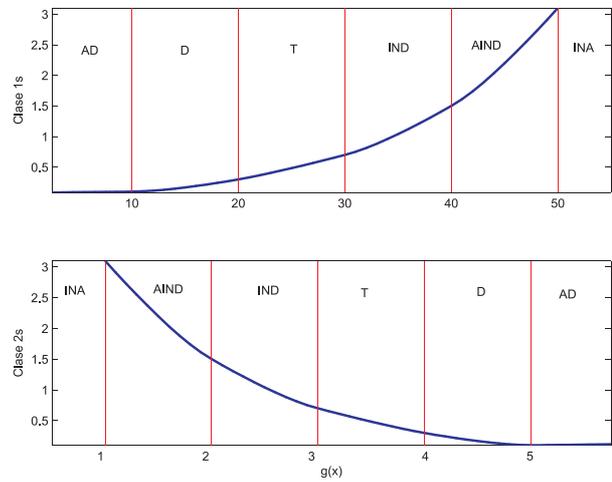


Figura 1. Clases 1s-2s.

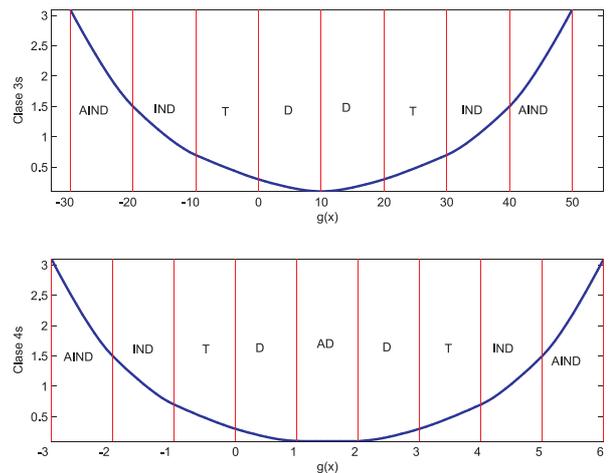


Figura 2. Clases 3s-4s.

De forma similar se definirían el resto de clases suaves. Puesto que los valores $g_i^1 \dots g_i^5$ tienen

⁴ Se podría elegir cualquier número de variables, tipo de especificaciones e intervalos de preferencia de cada g_i .

significado físico, el diseñador podrá cuantificar, por su experiencia, las preferencias asociadas a cada especificación g_i . La imagen de g_i , cualquiera que sea la Función de Clase suave, corresponde a una variable adimensional y estrictamente positiva (\bar{g}_i). La curvatura de dichas funciones tendrá un papel fundamental en el proceso de optimización posterior, por lo que su diseño será analizado en su momento en este artículo. También tendrán un papel fundamental las imágenes de los puntos $g_i^1 \dots g_i^5$, denominadas como $\bar{g}^1 \dots \bar{g}^5$, cuya principal característica es la coincidencia de valor, cualquiera que sea la Función de Clase suave que se defina⁵ y por tanto transparente para el usuario o diseñador que aplica el método. Esta característica permite un efecto de normalización de todas las variables g_i en el espacio imagen antes definido.

Respecto a las funciones de clase *duras*, los dos valores de la imagen, *posible* o *imposible*, representarían a las restricciones duras en un problema de optimización clásico (figura 3). De nuevo y sin pérdida de generalidad, la clase 1h de la figura podría corresponder a una especificación de tiempo de establecimiento del tipo

$$t_{est(98\%)} \leq 50 \text{seg.}$$

de manera que, si se cumple, la función de clase 1h toma el valor 0 y si no se cumple toma un valor suficientemente grande (en el caso ideal, infinito).

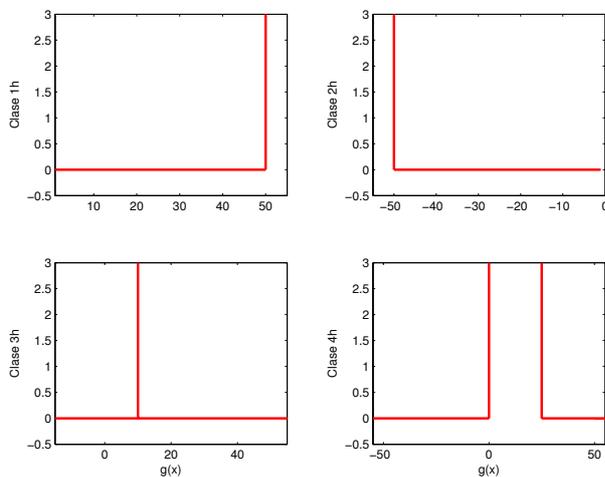


Figura 3. Clases 1h-4h.

2.2 Función agregada.

Un problema, desde el punto de vista de PP, se formula como el de optimización no lineal con restricciones de una función agregada o índice de coste dada por:

$$J(\mathbf{x}) = \log_{10} \frac{1}{n_{sc} \cdot \bar{g}^5} \left[\sum_{i=1}^{n_{sc}} \bar{g}(g_i(\mathbf{x})) \right] \quad (1)$$

El problema de minimización a resolver es por tanto:

$$x^* = \underset{x}{\text{arg}}[\text{mín} J(\mathbf{x})] \quad (2)$$

para el conjunto de clases suaves (n_{sc}) y sujeto a las restricciones de las clases duras y a las restricciones del vector \mathbf{x} .

En esta formulación, \mathbf{x} representa la variable multidimensional de parámetros, cuyo valor adecuado se determinará en el proceso de optimización. En general, y en función del problema a resolver, dicho vector de parámetros tiene una correspondencia con las especificaciones g_i planteadas. En ocasiones, los propios parámetros pueden corresponder directamente con estas especificaciones, en cuyo caso la especificación g_i representaría directamente a alguno de los elementos del vector \mathbf{x} . Obviamente, el problema también puede estar sujeto a restricciones duras de cada componente del vector \mathbf{x} .

El problema (2) así planteado, conforma un problema de optimización no lineal que, en teoría podría resolverse por cualquier algoritmo de optimización no lineal implementado convenientemente con variantes de los métodos del Gradiente o Newton.

La formulación del índice (1) en base logarítmica, pretende expandir el rango de búsqueda, logrando así una reducción apreciable en el número de iteraciones del optimizador seleccionado. Dicho índice representa la adición de tantos términos como especificaciones consideradas, con la importante cualidad de que todos ellos ponderan de la misma forma en el índice. Además, gracias a las especiales características con las que se han conformado las funciones de clase suave, dichos términos son independientes de las variables físicas de las que proceden.

2.3 El problema de los mínimos locales.

La estructura de la función agregada que se ha definido en la ecuación (1), por las particulares relaciones que pueden relacionar los parámetros \mathbf{x} con las especificaciones g_i y de éstas con las variables o funciones de clase \bar{g}_i , no permite asegurar que el problema de optimización formulado tenga un sólo mínimo.

Por ejemplo, un problema de optimización con una sola especificación en la función agregada y donde la relación entre los parámetros a determinar y dicha especificación sea del tipo cuadrático,

⁵ De ahí que para estos valores se prescindiera del subíndice.

nos permitiría llegar a la conclusión de que, ante cualquier función de clase del tipo suave, aparecerán 2 mínimos locales (Messac, 1996). Lógicamente, en el caso de funciones agregadas de varias especificaciones, el problema se hace mucho más complejo y la potencial presencia de múltiples mínimos locales hace que la solución sea muy sensible a la elección del método de optimización no lineal.

2.4 Representación matemática de las funciones de clase.

Como se acaba de comentar, la presencia de mínimos locales en la función agregada definida en la ecuación (1) depende tanto de la relación entre los parámetros \mathbf{x} y las especificaciones $g_i(\mathbf{x})$, como de la relación entre estas últimas y las Funciones de Clase $\bar{g}_i(g_i(\mathbf{x}))$. La especificaciones $g_i(\mathbf{x})$ dependen del problema concreto que se quiere resolver y su relación vendrá impuesta. En cambio, $\bar{g}_i(g_i(\mathbf{x}))$ sí se puede definir de manera que se puedan cumplir ciertas propiedades que faciliten el trabajo al algoritmo de optimización.

Con estas consideraciones, se definirán las funciones de clase suaves con unas propiedades matemáticas que den robustez al proceso de búsqueda del óptimo por parte del algoritmo de optimización no lineal. Las propiedades que cada Función de Clase debe poseer son las siguientes (Messac, 1996):

- Estrictamente positivas.
- Continuas en su primera derivada.
- Estrictamente positivas en su segunda derivada.
- Todas las propiedades anteriores deben cumplirse con independencia de la selección de los rangos de preferencia.

Para la clase 1s:

- Primera derivada estrictamente positiva.
- $\lim_{g_i \rightarrow -\infty} \bar{g}_i(g_i) = 0$.

Para la clase 2s:

- Primera derivada estrictamente negativa.
- $\lim_{g_i \rightarrow \infty} \bar{g}_i(g_i) = 0$.

Para la clase 3s y 4s:

- Primera derivada posee un sólo cero.

Con estas propiedades, es posible desarrollar un método para la construcción de dichas funciones. A continuación, se presenta el desarrollo para las funciones de clase 1s, el cual puede ser extendido a las demás clases sin problemas.

En lo que sigue, se va a usar la siguiente notación:

1. Se definirán las coordenadas de los puntos extremos de los intervalos como:

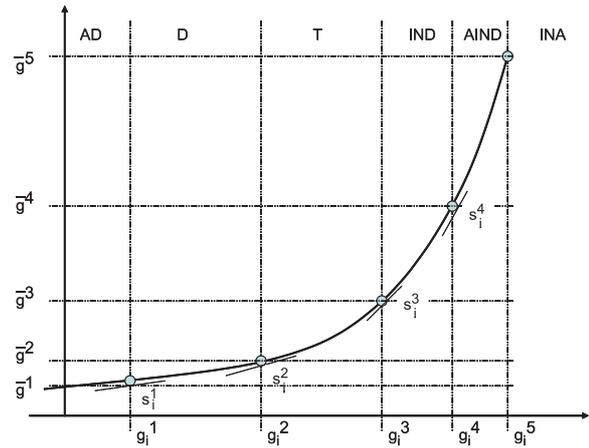


Figura 4. Función de Clase 1s para una especificación i genérica.

$$(g_i^k, \bar{g}^k); k = [1..5] \quad (3)$$

2. Se definirán las coordenadas de los puntos interiores de cada intervalo como:

$$(g_{i(k)}, \bar{g}_{i(k)}); k = [1..5] \quad (4)$$

En este caso, la función $\bar{g}_{i(k)}$ lleva subíndice i debido a que su valor es dependiente de la especificación i , cosa que no ocurre así en sus extremos.

En la figura 4, se observa que la curvatura puede ser representada por dos tipos de funciones genéricas. La primera puede ser representada por una exponencial decreciente en el intervalo 1 ($g_{i(k)} \leq g_i^1$), definida por:

$$\bar{g}_{i(1)} = \bar{g}^1 \exp\left[\frac{s_i^1}{\bar{g}^1}(g_{i(1)} - g_i^1)\right] \quad (5)$$

Para los intervalos 2, 3, 4 y 5 la función genérica toma la forma de un segmento *spline* que puede ser definido por sus puntos extremos (g_i^k, \bar{g}^k) y $(g_i^{k-1}, \bar{g}^{k-1})$, y pendiente en dichos puntos (s_i^{k-1}, s_i^k) .

El desarrollo de una *spline* que posee una segunda derivada estrictamente positiva comienza por definir (para cada región $k \in [2 \dots 5]$):

$$\frac{d^2 \bar{g}_{i(k)}}{dg_{i(k)}^2} = (\lambda_i^k)^2 [a(\xi_{i(k)})^2 + b(\xi_{i(k)} - 1)^2] \quad (6)$$

$$\lambda_i^k = g_i^k - g_i^{k-1} \quad (7)$$

$$\xi_{i(k)} = \frac{g_{i(k)} - g_i^{k-1}}{\lambda_i^k} \quad (8)$$

$$0 \leq \xi_{i(k)} \leq 1 \quad (9)$$

Donde a y b deben ser constantes reales y estrictamente positivas.

La integración de (6) conduce a la expresión de la primera derivada y la integración de esta

última a la propia función en el segmento *spline* considerado:

$$\begin{aligned} \frac{d\bar{g}_{i(k)}}{dg_{i(k)}} &= (\lambda_i^k)^3 \left[\frac{a}{3} (\xi_{i(k)})^3 + \frac{b}{3} (\xi_{i(k)} - 1)^3 \right] + d(10) \\ \bar{g}_{i(k)} &= (\lambda_i^k)^4 \left[\frac{a}{12} (\xi_{i(k)})^4 + \frac{b}{12} (\xi_{i(k)} - 1)^4 \right] + \\ &+ c\lambda_i^k \xi_{i(k)} + d \end{aligned} \quad (11)$$

Las constantes a, b, c, d , para la región $k \in [2 \dots 5]$ considerada, de la expresión (11), pueden ser determinadas usando:

- Las preferencias del diseñador para la especificación g_i , (reflejadas en la selección por parte de éste de los extremos g_i^{k-1} y g_i^k).
- Los valores de dichos extremos en el conjunto imagen de la Función de Clase: \bar{g}^{k-1} y \bar{g}^k . Aplicación de la ecuación (11) en sus extremos.
- Las pendientes s_i^{k-1} y s_i^k para los puntos anteriores. Aplicación de la ecuación (10) en sus extremos.

Bajo estas consideraciones, las constantes antes mencionadas se calcularán a través de las siguientes expresiones:

$$a = \frac{3[3s_i^k + s_i^{k-1}] - 12\bar{s}_i^k}{2(\lambda_i^k)^3} \quad (12)$$

$$b = \frac{12\bar{s}_i^k - 3[s_i^k + 3s_i^{k-1}]}{2(\lambda_i^k)^3} \quad (13)$$

$$c = 2\bar{s}_i^k - \frac{1}{2}[s_i^k + s_i^{k-1}] \quad (14)$$

$$d = \bar{g}^{k-1} - \lambda_i^k \frac{4\bar{s}_i^k - [s_i^k + 3s_i^{k-1}]}{8} \quad (15)$$

donde

$$\bar{s}_i^k = \frac{(g^k - g^{k-1})}{\lambda_i^k} = \frac{\bar{g}^k}{\lambda_i^k} \quad (16)$$

Usando las ecuaciones anteriores para a y b es posible determinar los límites de las pendientes s_i^{k-1} y s_i^k para cada región k , de tal forma que presenten valores estrictamente positivos, resultando:

$$(s_i^k)_{max} = 4\bar{s}_i^k - 3s_i^{k-1} \quad (17)$$

$$(s_i^k)_{min} = \frac{4\bar{s}_i^k - s_i^{k-1}}{3} \quad (18)$$

$$\begin{aligned} \Delta s_i^k &= (s_i^k)_{max} - (s_i^k)_{min} = \\ &= \frac{8}{3}[\bar{s}_i^k - s_i^{k-1}] \end{aligned} \quad (19)$$

Esta última expresión permitiría determinar las pendientes sucesivas para cada punto k de cada una de las especificaciones g_i .

2.5 Algoritmo de cálculo de las funciones de clase.

Para un número de especificaciones seleccionado (n_{sc}), el cálculo de las imágenes de los puntos extremos de cada intervalo g_i^k y de sus pendientes asociadas s_i^k resulta de aplicar el siguiente algoritmo:

- Para cada una de las especificaciones $i \in [1 \dots n_{sc}]$ hacer:
 1. Inicializar ⁶
 - a) $\beta_i = 1.5$
 - b) $\alpha = 0.1$
 - c) $\bar{g}^1 = 0.1$
 - d) $\bar{g}^1 = \bar{g}^1$
 2. Para $k \in [2 \dots 5]$ hacer:
 - a) $\bar{g}^k = \beta_i \cdot n_{sc} \cdot \bar{g}^{k-1}$
 - b) $\bar{g}^k = \bar{g}^{k-1} + \bar{g}^k$
 - c) $\lambda_i^k = g_i^k - g_i^{k-1}$
 - d) $\bar{s}_i^k = \frac{\bar{g}^k}{\lambda_i^k}$
 3. $s_i^1 = \alpha \cdot \bar{s}_i^2$ (cálculo de la pendiente de punto (g_i^1, \bar{g}^1)).
 4. Para $k \in [2 \dots 5]$ hacer:
 - a) $s_i^k = (s_i^k)_{min} + \alpha \Delta s_i^k$
 - b) Calcular los parámetros a y b
 5. Si algún a o b calculado es negativo, hacer $\beta_i = \beta_i + 0.5$, volver al paso 2, y repetir el proceso hasta que a y b sean positivos.
- Calcular el máximo de los parámetros β_i y ajustar o reconciliar con ese valor todos los valores imagen \bar{g}^k .

Una correcta aplicación del algoritmo y el posterior ajuste conduce a que todas y cada una de las especificaciones g_i coincidirán en sus puntos imagen \bar{g}^k para los puntos g_i^k que delimitan las regiones.

Es importante resaltar que en la clase 2s, el algoritmo anterior debe modificarse en el paso 4.a) con la expresión:

$$s_i^k = (s_i^k)_{max} + \alpha \Delta s_i^k \quad (20)$$

Para construir las funciones de clase 3s y 4s deben emplearse los algoritmos definidos para las clases 1s y 2s de forma independiente.

Para la clase 4s la región definida entre los puntos extremos de la región AD puede definirse por una cúbica, cuyos 4 parámetros se calculan gracias a los valores asignados a la función y a su pendiente en dichos extremos.

⁶ Inicialmente, β_i toma un valor positivo pequeño y es un factor que se va modificando en el algoritmo para asegurar que los parámetros a y b sean positivos. La constante α también toma un valor inicial pequeño

3. OPTIMIZACIÓN NO LINEAL CON ALGORITMOS GENÉTICOS

El problema que se quiere resolver es un problema de minimización de la función agregada que puede ser multimodal sujeta a una serie de restricciones. Para obtener la solución se debe utilizar una técnica de optimización que no quede atrapada en los posibles mínimos locales. Los algoritmos genéticos han demostrado su capacidad en este tipo de situaciones siendo la opción que se adopta en este artículo.

Un algoritmo genético (AG) es una técnica de optimización que realiza la búsqueda de la solución del problema de optimización imitando los mecanismos de evolución de las especies (Goldberg, 1989), (Holland, 1975). En este tipo de algoritmos, un conjunto de individuos (que llamaremos población) cambia de generación en generación sufriendo un proceso de evolución adaptándose cada vez mejor al entorno. En un problema de optimización, disponemos de una función a optimizar (maximizar o minimizar) y una zona donde buscar el óptimo (espacio de búsqueda), a cada punto del espacio de búsqueda le corresponde un valor de la función anterior, el objetivo es encontrar el punto que optimice (minimice o maximice) dicha función.

Trasladado el problema de optimización a un AG, los distintos puntos del espacio de búsqueda se identifican con los posibles individuos distintos de una población. Como ocurre con la genética natural, cada individuo se caracteriza por su o sus cromosomas, en el problema de optimización el cromosoma de un individuo está constituido por las coordenadas del punto en el espacio de búsqueda $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Siguiendo con el símil, cada una de las coordenadas correspondería a un gen.

El valor de la función de coste de un individuo (un punto del espacio) debe interpretarse como el grado de adaptación al entorno de dicho individuo. La evolución de una población se produce a través de la definición de diferentes tipos de operadores, conocidos como operadores genéticos. Los operadores genéticos clásicos son los conocidos como operador de selección, operador de cruce y operador de mutación. Cada uno de estos operadores trabaja por asignación de diferentes probabilidades. En el caso de la selección, cada individuo tiene mayor probabilidad de permanecer en la siguiente generación y su función objetivo asociada es mejor que la de los demás. En cuanto al cruce y la mutación se definen las probabilidades respectivas P_c y P_m ⁷.

Definido este marco general, aparecen diversas variantes en la implementación de los AGs que les confieren unas mejores prestaciones; variando el tipo de codificación de los genes, variando los mecanismos de selección, cruce o mutación, incorporando nuevos operadores genéticos (Goldberg, 1989), (Michalewicz, 1996), etc.

La implementación que se ha realizado para este problema presenta las siguientes características:

1. La codificación se realiza con números reales (Michalewicz, 1996), cada gen tiene un valor real.
2. La función de coste utilizada para la operación de selección no es directamente $J(\mathbf{x})$. Se realiza una operación de 'ranking' (Blasco, 1999), (Back, 1996) que consiste en ordenar de mayor a menor valor los individuos según su valor $J(\mathbf{x})$. Se cambia el valor de $J(\mathbf{x})$ por su posición en dicha clasificación, cada individuo tiene un nuevo valor $J'(\mathbf{x})$ que corresponde a su número de orden. Con esta operación se consigue que los individuos claramente dominantes no predominen demasiado rápido en la población, agotando prematuramente el algoritmo (este aspecto es especialmente importante en el caso de poblaciones con pocos individuos).
3. La operación de selección que se realiza se conoce como *Muestreo Universal Estocástico* (Baker, 1987). Garantiza que la probabilidad de supervivencia ($P(\mathbf{x}_i)$) de un individuo (\mathbf{x}_i) viene dada por la ecuación (21).

$$P(\mathbf{x}_i) = \frac{J'(\mathbf{x}_i)}{\sum_{j=1}^{N_{ind}} J'(\mathbf{x}_j)} \quad (21)$$

Donde N_{ind} es el número de individuos.

4. Para el cruce de los cromosomas se utiliza el operador de recombinación intermedia (Mühlenbein and Schlierkamp-Voosen, 1993). Los individuos hijos (\mathbf{x}'_1 y \mathbf{x}'_2) se obtiene realizando la siguiente operación sobre los individuos padres (\mathbf{x}_1 y \mathbf{x}_2):

$$\begin{aligned} \mathbf{x}'_1 &= \alpha_1 \cdot \mathbf{x}_1 + (1 - \alpha_1)\mathbf{x}_2 \\ \mathbf{x}'_2 &= \alpha_2 \cdot \mathbf{x}_2 + (1 - \alpha_2)\mathbf{x}_1 \\ \alpha_1 &\in [-d, 1 + d] \\ \alpha_2 &\in [-d, 1 + d] \end{aligned}$$

Este cruce se puede realizar sobre todo el cromosoma, o sobre cada uno de los genes por separado. En este segundo caso los parámetros aleatorios $\alpha_{1,2}$ se tiene que generar para cada gen incrementando el coste computacio-

⁷ Para una profundización general de los conceptos e implementación de los AG se recomienda acudir a algún

texto específico como los referenciados con anterioridad en esta apartado.

nal pero ofreciendo una mayor capacidad de exploración.

En el algoritmo genético implementado se han ajustado:

- los valores aleatorios $\alpha_1 = \alpha_2$, y se generan para cada cromosoma.
- $d = 0$

La probabilidad de cruce se ha ajustado a $P_c = 0.8$.

5. La operación de mutación, variación aleatoria de alguna parte del cromosoma, se produce con una probabilidad de $P_m = 0.1$. El cambio provocado en ese parámetro es generado por una distribución normal, de media el valor del parámetro antes de la mutación y de desviación típica el 20% del rango de búsqueda de ese parámetro. Con esto se consigue que el resultado de la mutación tenga más probabilidad de estar cerca del valor original.

Empleando funciones de test multimodales (Rastrigin, Schewefel o Griewank, entre otras) se puede obtener un ajuste estándar para los parámetros de diseño del AG (Blasco, 1999). De este modo se pueden ofrecer algunos de éstos (p. e. número de generaciones y de individuos) como parámetros a modificar por un usuario no familiarizado con esta técnica. No obstante, si se requiere un ajuste óptimo de un AG o la optimización de problemas de mayor complejidad será necesario un ajuste más específico de todos los parámetros.

4. SOLUCIÓN AL PROBLEMA TIPO DE CONTROL ROBUSTO

4.1 Descripción del problema tipo

El *problema tipo* de control robusto propuesto en la Conferencia Americana de Control (ACC) (Wie and Bernstein, 1990), (Wie and Bernstein, 1991), (Wie and Bernstein, 1992a), basado en un proceso masa-muelle plantea el problema de diseño de un controlador para alcanzar un compromiso entre maximizar la estabilidad y prestaciones robustas del sistema y, al mismo tiempo, minimizar el esfuerzo de control.

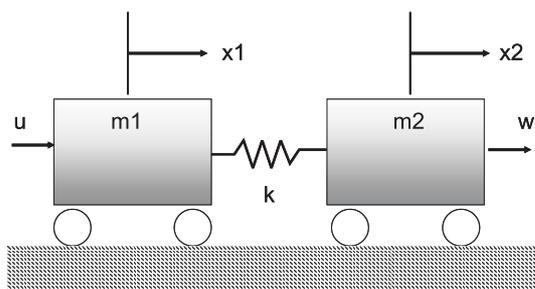


Figura 5. Sistema con 2 masas y un muelle con incertidumbre en los parámetros.

La figura 5 presenta el proceso descrito en el *problema tipo*. Se trata de un estructura flexible consistente en dos masas conectadas con un muelle. El modelo en espacio de estados del sistema es:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k/m_1 & k/m_1 & 0 & 0 \\ k/m_2 & -k/m_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/m_2 \end{bmatrix} w$$

$$y = x_2 + v$$

donde:

- x_1 y x_3 son, respectivamente, la posición y la velocidad de la masa 1.
- x_2 y x_4 son, respectivamente, la posición y la velocidad de la masa 2.

Los valores nominales para las dos masas m_1 y m_2 y para la constante k del muelle son la unidad. La acción de control u es la fuerza aplicada a la masa 1, mientras que y , la posición de la masa 2 afectada por ruido de medida, es la variable controlada. Además, existe una perturbación w que actúa sobre la masa 2.

De los tres escenarios de control propuestos en (Wie and Bernstein, 1992b), en el presente artículo se va a abordar el primero de ellos descrito, originalmente como:

- El sistema en bucle cerrado debe ser estable para $m_1 = m_2 = 1$ y $k \in [0.5 \dots 2]$.
- Para un impulso unitario en la perturbación w en $t = 0$, la variable controlada debe alcanzar un tiempo de establecimiento máximo de 15 segundos para el sistema nominal ($m_1 = m_2 = k = 1$).
- Se deben alcanzar márgenes de fase y ganancia razonables con un razonable ancho de banda.
- El bucle cerrado debe ser relativamente insensible a ruidos de alta frecuencia en la medida.
- El esfuerzo de control debe ser minimizado.
- La complejidad del controlador debe ser minimizada.

Hay que hacer notar que en esta definición del problema de control robusto existen ciertos aspectos que están completamente determinados, como incertidumbre de los parámetros, dinámica deseada para el sistema nominal, mientras que otros, ruido de medida, medida del esfuerzo de control, etc. son completamente abiertos a la interpretación del diseñador.

4.2 Objetivos de diseño

Tomando como punto de partida una determinada complejidad en el controlador⁸ y siendo \mathbf{x} el vector de parámetros a optimizar formado por dichos coeficientes, las especificaciones u objetivos de diseño $g(\mathbf{x})$, deben ser cantidades que el diseñador desea maximizar, minimizar, llevar a un cierto valor, etc. Para el *problema tipo* que nos ocupa, se utilizan seis funciones que devuelven dichas especificaciones para un determinado controlador. Su descripción es la siguiente:

1. **Tiempo de establecimiento nominal** (t_{est}^{nom}): Con la misma interpretación que en (Stengel and Marrison, 1992) y (Messac and Wilson, 1999), se asume que la variable controlada alcanza el régimen permanente frente a un impulso unitario en la perturbación w cuando está acotada por ± 0.1 unidades.
2. **Tiempo de establecimiento para el peor caso** (t_{est}^{max}): Con la interpretación anterior es el tiempo de establecimiento máximo, para un controlador dado, evaluado para el peor caso, $k = 0.5$ o $k = 2$.
3. **Estabilidad y prestaciones robustas** ($Re(\lambda)_{max}$): En (Stengel and Marrison, 1992) se muestra que los márgenes de fase y de ganancia para el peor caso son indicadores de poca calidad para evaluar la robustez. En lugar de éstos, se usarán los polos de bucle cerrado para el peor caso dado un controlador determinado. De esta forma se evalúa $Re(\lambda)_{max}$ como

$$Re(\lambda)_{max} = \max_{k \in [0.5 \dots 2]} Re(\lambda[A(k)])$$

donde A es la matriz del sistema en bucle cerrado.

4. **Sensibilidad al ruido** ($noise_{max}$): Para un determinado rango de frecuencias, la sensibilidad al ruido de medida ofrece un ratio de la amplificación del ruido de medida respecto a una pendiente de $-20db/dec$.

$$noise_{max} = \max_{k \in [0.5, 2]} \left(\left| \frac{u(jw)}{v(jw)} \right| / \left| \frac{1}{jw} \right| \right)$$

$$w \in [100 \dots 10000]$$

5. **Esfuerzo de control nominal** (u_{nom}): Acción de control máxima que se produce cuando aparece un impulso unitario en la perturbación, para el caso nominal.
6. **Esfuerzo de control máximo** (u_{max}): Acción de control máxima que se produce cuando aparece un impulso unitario en la perturbación, cuando hay incertidumbre en el sistema.

Existen por tanto, seis algoritmos o funciones que toman, como parámetros de entrada, los coeficientes del numerador y denominador del controlador y devuelven cada uno de los seis valores descritos anteriormente. Estas funciones se pueden utilizar para evaluar las prestaciones de las distintas propuestas de control elaboradas por otros autores y comparar con los resultados utilizando PP.

4.3 Preferencias y funciones de clase

Con el objetivo de comparar los resultados obtenidos con distintas soluciones del problema de control robusto propuestas por otros autores, se diseñarán controladores con funciones de transferencia estrictamente propias y de diferente complejidad.

Para todos los casos se han elegido funciones de clase 1s. Las tablas 1, 2, 3 presentan las preferencias o puntos extremos de las funciones de clase para las 6 especificaciones y para diferentes órdenes de controladores. Como se puede observar, los valores extremos exigibles en las especificaciones son, en general, cada vez más severos, según se aumenta la complejidad del controlador.

Como ejemplo, la figura 6 presenta la Función de Clase de la especificación de tiempo de establecimiento nominal (t_{est}^{nom}) resultante para los puntos extremos definidos en la tabla 1.

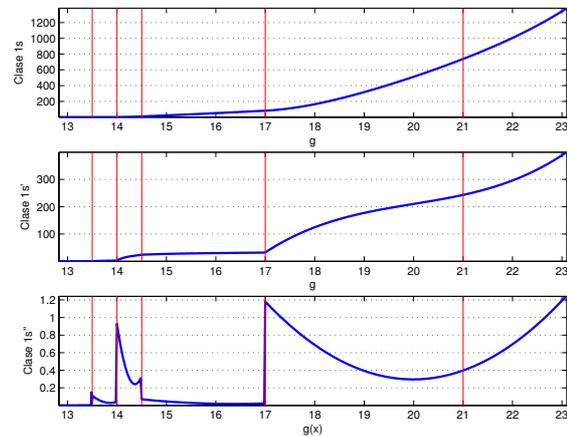


Figura 6. Función de Clase 1s y primera y segunda derivada para t_{est}^{nom}

Tabla 1. Preferencias para el diseño de un controlador (2,3)

	g_1^1	g_1^2	g_1^3	g_1^4	g_1^5
$Re(\lambda)_{max}$	-0.001	-0.0005	-0.0001	-5e-5	-1e-5
u_{max}	7	8	8.5	9	12
t_{est}^{max}	1e3	2e3	3e3	4e3	5e3
$noise_{max}$	5	10	30	40	50
u_{nom}	1	2	3	4	5
t_{est}^{nom}	13.5	14	14.5	17	21

⁸ Corresponde al orden de los polinomios del numerador y denominador seleccionados *a priori*.

Tabla 2. Preferencias para el diseño de un controlador (3,4)

	g_1^1	g_1^2	g_1^3	g_1^4	g_1^5
$Re(\lambda)_{max}$	-0.01	-0.005	-0.001	-0.0005	-0.0001
u_{max}	0.8	0.85	0.95	1	2
t_{est}^{max}	15	40	80	90	100
$noise_{max}$	1.8	2	2.2	2.5	3
u_{nom}	0.9	1.2	2	2.5	3
t_{est}^{nom}	14	14.2	14.4	14.6	15

Tabla 3. Preferencias para el diseño de un controlador (4,5)

	g_1^1	g_1^2	g_1^3	g_1^4	g_1^5
$Re(\lambda)_{max}$	-0.01	-0.005	-0.001	-0.0005	-0.0001
u_{max}	0.85	0.90	1	1.5	2
t_{est}^{max}	14	16	18	21	25
$noise_{max}$	0.5	0.9	1.2	1.4	1.5
u_{nom}	0.5	0.7	1	1.5	2
t_{est}^{nom}	10	11	12	14	15

5. RESULTADOS.

En este apartado se analizarán los resultados obtenidos al resolver el *problema tipo* de control robusto utilizando PP, al que se le ha incorporado como optimizador un AG. Son muchas las propuestas que proponen resultados a este *problema tipo* de control (Stengel and Marrison, 1992). Muy especialmente se pretende comparar con diseños obtenidos también a través de PP pero cuya optimización se lleva a cabo por métodos numéricos tradicionales de programación no lineal (Messac and Wilson, 1999), demostrando que utilizando un AG, se superan las limitaciones allí apuntadas.

En la tabla 4, se presentan 6 controladores de diferente complejidad⁹ propuestos por otros autores, que resuelven total o parcialmente el problema de control del sistema masa muelle. Para las preferencias especificadas en las tablas 1, 2, y 3, se aplicó PP optimizado con un AG, dando lugar a los controladores las tablas 5. La comparación entre ellos debe hacerse mediante el análisis de las seis especificaciones propuestas.

Los valores de las especificaciones para cada uno de los controladores anteriores se presentan en la tabla 6. Cada valor obtenido, aparece en uno de los rangos definidos por el diseñador gracias a los puntos extremos (tabla 7).

Analizando los resultados de las tablas 6 y 7 se puede deducir que:

- Los controladores diseñados utilizando PP (Messac and Wilson, 1999) mejoran, en general, a sus homólogos B23, W34, J45 diseñados por otros métodos. No obstante, para el caso de los controladores (4,5) dicha mejoría no es generalizada, de manera que el controlador

J45 es mejor en las especificaciones t_{est}^{max} y t_{est}^{nom} .

- Los controladores utilizando PP mediante optimización con AG son mejores en todas las especificaciones que sus homólogos. Hay que hacer notar que, incluso diseñando controladores de menor complejidad, se alcanzan mejores prestaciones (por ejemplo, con los controladores PPGA23* PPGA23**, PPGA34** donde se han exigido especificaciones mucho más restrictivas correspondientes a órdenes de controladores (3,4) y (4,5) respectivamente).

La razón de esta mejoría es consecuencia de que las soluciones propuestas por (Messac and Wilson, 1999) quedan atrapadas en un mínimo local¹⁰, no ocurriendo así con los nuevos controladores propuestos en este artículo.

Las figuras 7 y 8 muestran la salida del proceso cuando aparece un impulso unitario en la perturbación. En ellas se observa cómo el controlador PPGA34 presenta mejores prestaciones que otros controladores de igual complejidad.

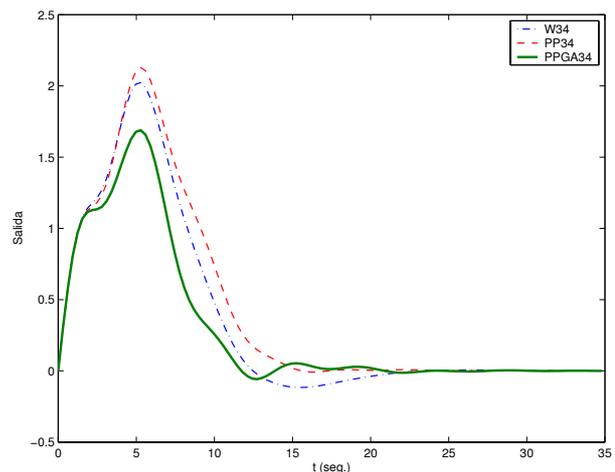


Figura 7. Tiempo de establecimiento nominal para los diseños W34, PP34 y PPGA34.

Las especificaciones alcanzadas no se consiguen a costa de un gran esfuerzo de control, (figuras 9 y 10), puesto que el controlador PPGA34 también tiene en cuenta las preferencias de esta especificación. En cuanto a la sensibilidad al ruido de medida (en el rango $[100 \dots 10000] rad/sec.$) la figura 11 muestra como el controlador PPGA34 consigue atenuar más que sus homólogos W34 y PP34.

6. CONCLUSIONES

Se ha desarrollado una metodología de diseño de controladores que permite incorporar, de forma

⁹ Se denota como (m,n) los grados de los polinomios numerador y denominador del controlador

¹⁰ O como en el caso del controlador M34 donde ni siquiera se llega a un mínimo local.

Tabla 4. Distintas soluciones para el *benchmark* de control robusto

Propuesta	(m, n)	f.d.t.
Byrns (B23) (Byrns and Calise, 1990)	(2, 3)	$\frac{40.4s^2 + 110.696s + 33.7946}{s^3 + 164.9258s^2 + 152.6928s + 140.0193}$
Messac (M23) (Messac and Wilson, 1999)	(2, 3)	$\frac{-12.5s^2 + 12.8375s + 3.1211}{s^3 + 21.8124s^2 + 26.44s + 30.1605}$
Wie (W34) (Wie and Bernstein, 1990)	(3, 4)	$\frac{-2.13s^3 - 5.327s^2 + 6.273s + 1.015}{s^4 + 4.68s^3 + 12.94s^2 + 18.36s + 12.68}$
Messac (M34) (Messac and Wilson, 1999)	(3, 4)	$\frac{-0.66s^3 - 4.101s^2 + 4.558s + 0.627}{s^4 + 3.416s^3 + 10.15s^2 + 13.52s + 9.281}$
Jayasuriya (J45) (Jayasuria <i>et al.</i> , 1992)	(4, 5)	$\frac{302e3s^4 + 1.052e6s^3 + 1.157e6s^2 + 8.401e5s + 2.978e5}{s^5 + 47.55s^4 + 2234s^3 + 3.152e4s^2 + 1.783e5s + 3.691e5}$
Messac (M45) (Messac and Wilson, 1999)	(4, 5)	$\frac{-1.14s^4 - 6.212s^3 + 7.353s^2 + 1.037s + 0.0154}{s^5 + 5.481s^4 + 15.01s^3 + 23.41s^2 + 14.31s + 0.1629}$

Tabla 5. Controladores obtenidos con PP y AGs

Propuesta	(m, n)	Preferencias	f.d.t.
PPGA23	(2, 3)	(2, 3)	$\frac{-0.0025s^2 + 0.8747s + 0.1176}{s^3 + 2.0767s^2 + 2.6282s + 1.3372}$
PPGA34	(3, 4)	(3, 4)	$\frac{-0.3226s^3 - 2.276s^2 + 4.79s + 0.7539}{s^4 + 2.075s^3 + 8.664s^2 + 11.32s + 7.825}$
PPGA45	(4, 5)	(4, 5)	$\frac{-0.1513s^4 - 2.9872s^3 + 7.9883s^2 + 1.8056s + 0.101}{s^5 + 3.6964s^4 + 14.1408s^3 + 20.0359s^2 + 13.491s + 0.9424}$
PPGA23*	(2, 3)	(3, 4)	$\frac{-1.5704s^2 + 3.1911s + 0.52}{s^3 + 5.2347s^2 + 7.2333s + 5.2436}$
PPGA23**	(2, 3)	(4, 5)	$\frac{-1.0122s^2 + 2.9275s + 0.4813}{s^3 + 4.8721s^2 + 6.476s + 4.5968}$
PPGA34**	(3, 4)	(4, 5)	$\frac{-0.6069s^3 - 0.1801s^2 + 3.6853s + 0.5740}{s^4 + 3.5681s^3 + 9.3442s^2 + 10.1662s + 5.6949}$

Tabla 6. Comparación de las especificaciones para cada controlador.

Controlador	$Re(\lambda)_{max}$	u_{max}	t_{est}^{max}	$noise_{max}$	u_{nom}	t_{est}^{nom}
B23	0.1416	-	-	-	0.5127	20.8100
M23	-0.0025	0.5669	21.000	12.2448	0.4548	20.5030
PPGA23	-0.0251	0.4381	25.1250	0.0091	0.3309	11.1838
W34	-0.0427	0.6793	22.125	2.1317	0.5595	16.7756
M34	-0.0542	0.6681	18.375	0.6620	0.5127	13.3580
PPGA34	-0.0166	0.7194	11.625	0.3241	0.5872	10.8731
PPGA23*	-0.0219	0.5259	15.000	1.5698	0.4170	10.8731
J45	-0.2254	47.76	6.375	337072	0.4450	6.21
M45	-0.0168	0.5873	18.375	1.1424	0.4622	12.1157
PPGA45	-0.0157	0.5581	12.0	0.1551	0.4664	10.5624
PPGA34**	-0.0233	0.5461	15.0	0.6075	0.4475	10.5624
PPGA23**	-0.0052	0.5223	15.25	1.0121	0.4126	10.8731

Tabla 7. Comparación de las especificaciones para cada controlador. AD-Altamente Deseable, D-Deseable, T-Tolerable, IND-Indeseable, AIND-altamente indeseable, INA-Inaceptable

Controlador	$Re(\lambda)_{max}$	u_{max}	t_{est}^{max}	$noise_{max}$	u_{nom}	t_{est}^{nom}
B23	INA	INA	INA	INA	AD	AIND
M23	AD	AD	AD	T	AD	AIND
PPGA23	AD	AD	AD	AD	AD	AD
W34	AD	AD	D	T	AD	IND
M34	AD	AD	D	AD	AD	AD
PPGA23*	AD	AD	AD	AD	AD	AD
PPGA34	AD	AD	AD	AD	AD	AD
J45	AD	IND	AD	IND	AD	AD
M45	AD	AD	IND	T	AD	IND
PPGA23**	D	AD	D	T	AD	D
PPGA34**	AD	AD	D	D	AD	D
PPGA45	AD	AD	AD	AD	AD	D

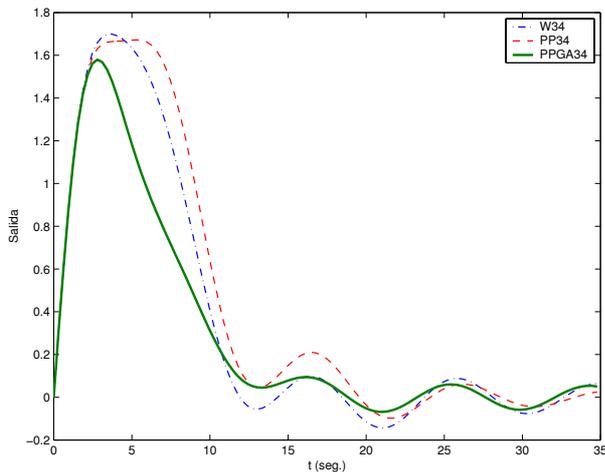


Figura 8. Tiempo de establecimiento para el peor caso ($k = 0.5$) para los diseños W34, PP34 y PPGA34.

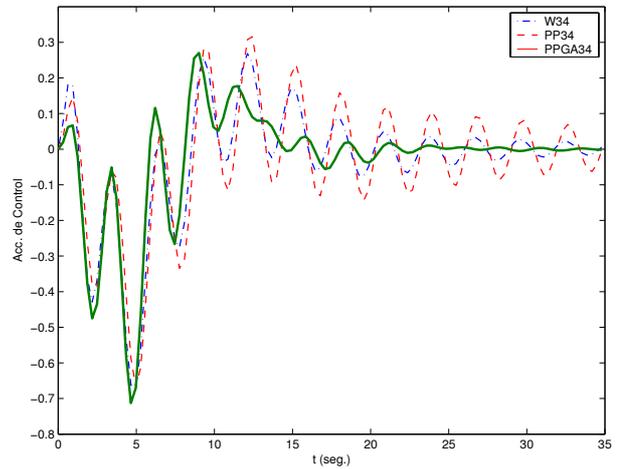


Figura 10. Acción de control para el peor caso ($k = 0.5$) para los diseños W34, PP34 y PPGA34.

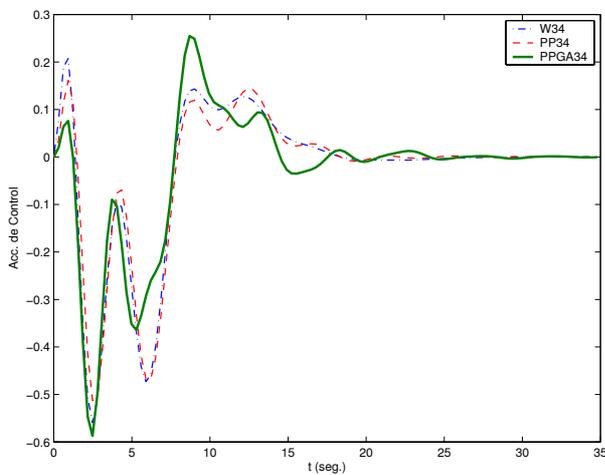


Figura 9. Acción de control nominal para los diseños W34, PP34 y PPGA34.

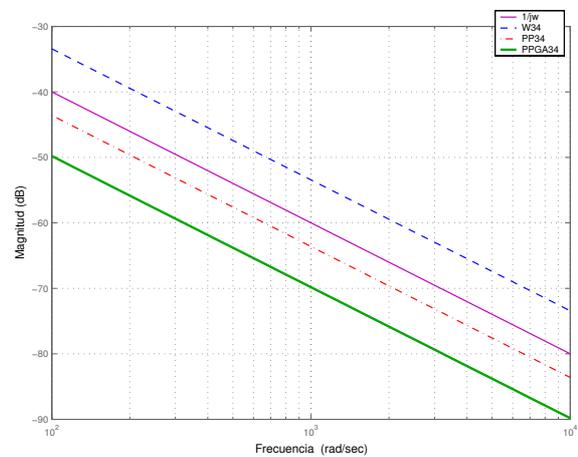


Figura 11. Sensibilidad al ruido de medida para los diseños W34, PP34 y PPGA34.

relativamente sencilla, distintos tipos de especificaciones y, por tanto, realizar diseños de controladores robustos. La idea fundamental para la incorporación de las especificaciones y preferencias de diseño está basada en el empleo de la metodología de PP que permite trasladar los deseos del diseñador a una estructura basada en Funciones de Clase que, agregadas en una función de coste, permite ser minimizada por alguna técnica de optimización suficientemente potente y obtener los parámetros de diseño óptimos.

Los AG, como técnica de optimización no lineal, han demostrado su eficacia en la obtención de soluciones que mejoran las presentadas originalmente (incluso con la técnica de PP, pero con métodos de optimización numéricos) en su aplicación al *problema tipo* de control robusto de la Conferencia Americana de Control (ACC).

Se pretenderá, en futuros trabajos, extender esta técnica a otros controladores y a otras especificaciones, dado que, desde un punto de vista práctico, no hay restricciones en su aplicabilidad. La metodología es utilizable también en otros campos distintos del diseño de controladores, siempre y cuando se basen en la optimización de criterios o especificaciones.

REFERENCIAS

- Back, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford University Press. New York.
- Baker, J.E. (1987). *Reducing Bias and Inefficiency in the Selection Algorithms*. pp. 14–21. Vol. Proceedings of the Second International Conference on Genetic Algorithms. Grefenstette, J.J. (ed.) ed.. Lawrence Erlbaum Associates., Hillsdale, NJ.
- Blasco, F.X. (1999). Control predictivo basado en modelos mediante técnicas de optimización heurística. PhD thesis. Universidad Politécnica de Valencia. Valencia.
- Byrns, E.V. and A.J. Calise (1990). Fixed-order dynamic compensators for h2/hinty benchmark problem. In: *Proceedings of the American Control Conference*. San Diego, CA. pp. 963–965.
- Coello, C.A., D. Van Veldhuizen and G.B. Lamont (2002). *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers.
- Goldberg, D.E. (1989). *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley.
- Holland, J.H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.
- Jayasuria, S., O. Yaniv, O.D.I. Nwokah and Y. Chait (1992). Benchmark problem solution by quantitative feedback theory. *Journal of Guidance, Control and Dynamics* **15**(5), 1087–1093.
- Messac, A. (1996). Physical programming: effective optimization for computational design. *AIAA Journal* **34**(1), 149–158.
- Messac, A. and B. H. Wilson (1999). Physical programming for computational control. *AIAA Journal* **36**(1), 219–226.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer series Artificial Intelligence. 3rd edition ed.. Springer.
- Miettinen, Kaisa M. (1998). *Nonlinear multiobjective optimization*. Kluwer Academic Publishers.
- Mühlenbein, H. and D. Schlierkamp-Voosen (1993). Predictive models for the breeder genetic algorithm. i. continuous parameter optimization. *Evolutionary Computation. The MIT Press* **1**(1), 25–49.
- P.J. Fleming, R. C. Purshouse (2002). Evolutionary algorithms in control systems engineering: a survey. *Control Eng. Practice* **10**, 1223–1241.
- Stengel, R. and C. Marrison (1992). Robustness of solutions to a benchmark control problem. *Journal of Guidance, Control and Dynamics* **15**(5), 1060–1067.
- Toscano, G. and C. Coello (2003). The micro genetic algorithm 2: Towards on-line adaptation in evolutionary multiobjective optimization. *Evolutionary Multi-Criterion Optimization. Second International Conference* **2632**, 252–266.
- Wie, B. and D. Bernstein (1990). A benchmark problem for robust control design. In: *Proceedings of the American Control Conference*. San Diego, CA. pp. 961–962.
- Wie, B. and D. Bernstein (1991). A benchmark problem for robust control design. In: *Proceedings of the American Control Conference*. MA. pp. 1929–1930.
- Wie, B. and D. Bernstein (1992a). A benchmark problem for robust control design. In: *Proceedings of the American Control Conference*. Chicago, IL. pp. 2047–2048.
- Wie, B. and D. Bernstein (1992b). Benchmark problems for robust control design. *Journal of Guidance, Control and Dynamics* **15**(5), 1057–1059.