

## Algoritmos Evolutivos y su empleo en el ajuste de controladores del tipo PID: Estado Actual y Perspectivas

Gilberto Reynoso-Meza\*, Javier Sanchis, Xavier Blasco, Miguel Martínez

*Departamento de Ingeniería de Sistemas y Automática, Universitat Politècnica de València, Camino de Vera, nº 14, 46022, Valencia, España.*

### Resumen

Los controladores PID continúan siendo una solución fiable, robusta, práctica y sencilla para el control de procesos. Actualmente constituyen la primera capa de control de la gran mayoría de las aplicaciones industriales. De ahí que un número importante de trabajos de investigación se han orientado a mejorar su rendimiento y prestaciones. Las líneas de investigación en este campo van desde nuevos métodos de ajuste, pasando por nuevos tipos de estructura hasta metodologías de diseño integrales. Particularizando en el ajuste de parámetros, una de las formas de obtener una solución novedosa consiste en plantear un problema de optimización, el cual puede llegar a ser no-lineal, no-convexo y con restricciones. Dado que los algoritmos evolutivos han mostrado un buen desempeño para solucionar problemas complejos de optimización, han sido utilizados en diversas propuestas relacionadas con el ajuste de controladores PID. Este trabajo muestra un revisión de estas propuestas y las prestaciones obtenidas en cada caso. Así mismo, se identifican algunas tendencias y posibles líneas de trabajo futuras. *Copyright © 2013 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.*

### Palabras Clave:

Controlador PID, PID convencional, PID borroso, PID fraccionario, Algoritmos Evolutivos, Optimización

### 1. Introducción

Los controladores PID representan a día de hoy, una solución fiable, robusta y práctica para el control de procesos en la industria (Åström y Hägglund (2001, 2005)). Pese a su sencillez, existe un continuo interés de la comunidad científica por el desarrollo de nuevos métodos de ajuste de sus parámetros (Åström y Hägglund (2005)). Estos avances son de especial interés para la industria, dado que los cambios requeridos en las instalaciones existentes y la capacitación requerida de los operadores es mínima, comparada con las técnicas más sofisticadas de control (Tan et al. (2004); Stewart y Samad (2011)).

Pese a que la mayoría de los controladores del tipo PID funcionan moderadamente bien con parámetros por defecto o con reglas de sintonía sencillas, pueden existir lazos que por sus particularidades necesiten un ajuste diferente para garantizar las prestaciones deseadas. En el caso de controladores más complejos pero basados en la estructura PID, por ejemplo borrosos

o fraccionarios, no se cuenta con un conjunto de reglas de sintonía con la misma madurez del controlador PID convencional.

La investigación actual apunta a mejorar las prestaciones de controladores PID robustos (Vilanova y Alfaro (2011)) atendiendo a múltiples requerimientos, especificaciones y objetivos de diseño, en la mayoría de los casos conflictivos entre sí (Ang et al. (2005); Li et al. (2006)). Un mecanismo con el que se puede abordar dicho problema es definiendo un problema de optimización, donde en la función de coste se pretende reflejar el balance entre los objetivos deseados por el diseñador.

Aunque se puede abordar dicho problema de diseño con métodos de optimización tradicionales (ver por ejemplo Cabello y Grossmann (2011)), en general los problemas planteados para alcanzar mejores prestaciones suelen ser no-lineales y no-convexos. Además la incorporación de restricciones, imprescindible en problemas de diseño en contexto realista, suele complicar aún más el proceso de optimización. Los algoritmos evolutivos (EA's por sus siglas en inglés) se han utilizado como una alternativa válida para resolver estos planteamientos en diversos campos de la ingeniería como exponen Saridakis y Dentsoras (2008) y Roy et al. (2008). Dada su naturaleza de optimizadores globales y su flexibilidad para manejar funciones de coste altamente no-lineales, los EA's también han sido usado en la ingeniería de control con mucho éxito, como se describe

\* Autor en correspondencia.

Correos electrónicos: [gilreyme@upv.es](mailto:gilreyme@upv.es) (Gilberto Reynoso-Meza),  
[jsanchis@isa.upv.es](mailto:jsanchis@isa.upv.es) (Javier Sanchis), [xblasco@isa.upv.es](mailto:xblasco@isa.upv.es) (Xavier Blasco),  
[miranzo@isa.upv.es](mailto:miranzo@isa.upv.es) (Miguel Martínez)

URL: <http://cpoh.upv.es> (Gilberto Reynoso-Meza)

en Fleming y Purshouse (2002).

En este documento se presentará una revisión del empleo de EA's para el ajuste de parámetros de controladores basados en la estructura del PID. Se asume entonces que se hablará de aquellos casos en los que la naturaleza del problema requiere de una técnica de ajuste más compleja (mediante un proceso de optimización) con el fin de garantizar las prestaciones deseadas del diseñador. En este contexto, se analizan las aportaciones de los EA's en el ajuste de controladores basados en la estructura del PID, que pueden ser convencionales, borrosos o fraccionarios. El resto de este documento se divide como sigue: en la sección 2 serán expuestas tres vías por las que el diseño puede ser abordado como un problema de optimización; en la sección 3 se presentará una breve introducción a los EA's y algunas de sus generalidades; en la sección 4 se presentarán aplicaciones relevantes sobre como han sido empleados los EA's con los controladores PID, para finalmente presentar las conclusiones y reflexiones de este trabajo.

## 2. Diseño planteado como problema de optimización

Sin pérdida de generalidad, un problema de diseño puede plantearse como el siguiente problema de optimización:

$$\min_{\mathbf{x}} \mathbf{J}(\mathbf{x}) = [J_1(\mathbf{x}), \dots, J_m(\mathbf{x})] \quad (1)$$

sujeito a:

$$\mathbf{g}(\mathbf{x}) \leq 0 \quad (2)$$

$$\mathbf{h}(\mathbf{x}) = 0 \quad (3)$$

$$x_i \leq \bar{x}_i, i = [1, \dots, n] \quad (4)$$

Donde  $\mathbf{x} \in \mathfrak{R}^n$  es el vector de variables de decisión;  $\mathbf{J}(\mathbf{x}) \in \mathfrak{R}^m$ , es el vector de objetivos de diseño;  $\mathbf{g}(\mathbf{x})$ ,  $\mathbf{h}(\mathbf{x})$  el vector de restricciones de desigualdad e igualdad respectivamente, y  $x_i \leq \bar{x}_i$  la cota de búsqueda del espacio de decisión para la variable  $x_i$ .

En la Figura 1 se muestran las diferentes formas que podría adoptar el anterior problema de diseño y como podría resolverse. En primer lugar ha de definirse el problema de diseño como tal. Lo anterior implica identificar las variables de decisión, los objetivos de diseño y las restricciones (si las hay) del problema. Cuando  $m = 1$  se habla de un problema mono-objetivo (SOP por sus siglas en inglés) mientras que si  $m \geq 2$  se habla de un problema multi-objetivo (MOP por sus siglas en inglés), donde se busca una solución de compromiso entre los objetivos de diseño.

Tras lo anterior, se define el problema de optimización; ello refiere a la forma en que se traduce el problema anterior en una expresión que el optimizador pueda interpretar. En el caso de un SOP, el índice a optimizar será el mismo (y único) objetivo de diseño. En el caso de un MOP, su resolución se puede abordar desde dos perspectivas acorde con Mattson y Messac (2005): empleando funciones de agregación (*aggregate objective function*, AOF) o calculando un conjunto de soluciones para seleccionar después la más indicada atendiendo a las preferencias del diseñador (*generate first, choose later*, GFCL).

En el caso de técnicas MOP/AOF, se busca especificar todos los requerimientos de diseño y las preferencias del diseñador en un sólo índice. Con el planteamiento anterior el optimizador calcula una solución única para implementar. Un ejemplo tradicional es usar factores de ponderación para indicar la importancia relativa que confiere el diseñador a los objetivos de diseño. Algunas técnicas clásicas para abordar este tipo de problemas se describen en Miettinen (1998) y en Marler y Arora (2004).

La MOP/GFCL es una estrategia que busca determinar un conjunto de soluciones potenciales, para que el diseñador, a partir de ellas, elija la más indicada de acuerdo a sus preferencias de diseño. Lo anterior se justifica con el hecho de que, cuándo se consideran simultáneamente todos los objetivos, generalmente no hay una solución mejor que las otras, pues no existe una solución única que mejore a todas las demás en todos los objetivos.

La fase de optimización se refiere al uso/ejecución del algoritmo de optimización; como se ha comentado con anterioridad, en este trabajo de revisión se asume que se ha detectado la necesidad de efectuar un ajuste mediante optimización empleando EA's. Los casos SOP y MOP/AOF derivan en la optimización de una función de coste mono-índice, y se espera en la mayoría de los casos (ver apartado 3.3.2) una solución única. En el caso SOP, se espera que el único objetivo de diseño sea suficiente para encontrar una solución práctica para el diseñador. En el caso de MOP/AOF, se espera que el intercambio de prestaciones deseado entre objetivos de diseño sea adecuadamente reflejado en el diseño de la función de coste mono-índice. En el caso de MOP/GFCL, se busca (por lo general) un conjunto de soluciones que aproxime lo que se conoce como conjunto de Pareto  $X_P$ , y su imagen en el espacio de objetivos, frente de Pareto  $J_P$  (Miettinen (1998)).

La etapa de selección de la solución es exclusiva de la rama MOP/GFCL. En los casos SOP y MOP/AOF, no existe esta etapa, pues la solución que ofrece el optimizador supone una solución con las prestaciones deseadas que han sido definidas *a priori*. En el caso MOP/GFCL, la búsqueda de una solución con las prestaciones deseadas se efectúa (en general) *a posteriori*, analizando el conjunto de soluciones y el intercambio de sus prestaciones. Este proceso adicional de selección consume tiempo, y en algunos casos puede no ser justificado, por lo que la técnica MOP/AOF resulta más práctica. Sin embargo, en muchas ocasiones el diseño de la función de coste mono-índice, garantizando el intercambio de prestaciones deseado, puede requerir un proceso de prueba y error. Pero una vez formulada a satisfacción del diseñador, no requiere mas la intervención del mismo, minimizando así el tiempo de implementación. Abordar el diseño como un MOP/GFCL implica resolver un problema de optimización aún más complejo que se justificará solamente si es necesario efectuar un análisis profundo en el intercambio de prestaciones. Acorde con Bonissone et al. (2009) esto puede ser de gran ayuda cuando se requiere explicar y justificar el proceso de selección. En tal caso, la visualización del intercambio de prestaciones y rendimiento entre los objetivos puede ser valiosa para el diseñador, como señalan Lotov y Miettinen (2008) y Blasco et al. (2008). En Figueira et al. (2005) se presenta un

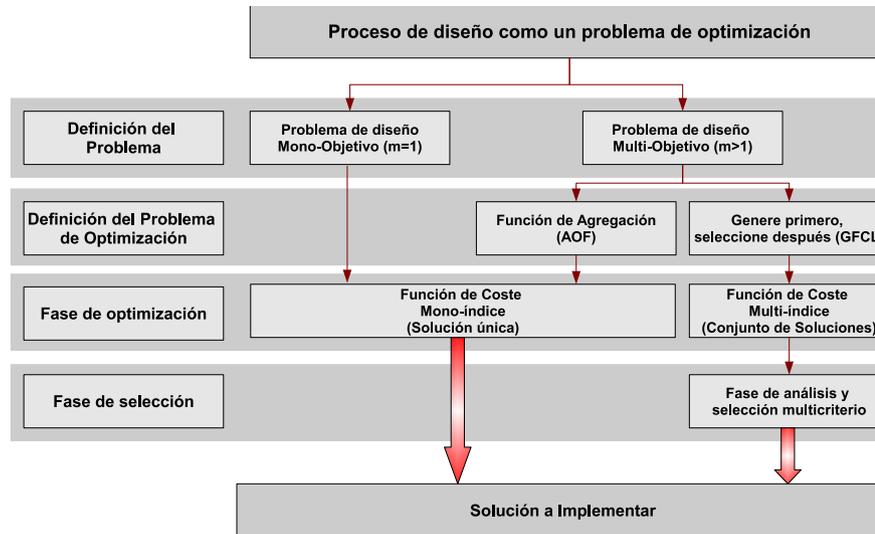


Figura 1: Metodología de diseño por medio de la optimización.

amplio compendio de técnicas (y software) para la selección multi-criterio y el análisis de datos multi-dimensionales.

Por sencillez, se hará referencia a la técnica MOP/AOF simplemente como AOF, mientras que a la técnica MOP/GFCL como MOOD (diseño mediante optimización multi-objetivo por sus siglas en inglés) como es referido en Reynoso-Meza et al. (2012a). Cabe destacar que dentro de la comunidad en computación evolutiva, el término optimización multi-objetivo es utilizado para especificar aquellos algoritmos que aproximan un conjunto de Pareto. En este trabajo hemos decidido hacer la distinción empleando los términos problema multi-objetivo (definición del problema) y la optimización multi-índice (definición del problema de optimización) por motivos de claridad.

A continuación, se comentarán brevemente los tipos de estructura que podrían emplear los EA's, así como algunas de las características deseables en ellos con relación al problema que se resuelve. Dichas características deseables ofrecen una pauta para apreciar las tendencias actuales en el desarrollo de EA's eficientes para el diseño en ingeniería.

### 3. Algoritmos evolutivos

Por EA's entenderemos (indistintamente) aquellos algoritmos que se basan o bien en las leyes de la selección natural (*evolutionary algorithms*) o que son inspirados por el comportamiento de ciertas especies (*nature inspired*). Los primeros emplean un esquema en el que los individuos más aptos (soluciones) en una población (conjunto de soluciones potenciales) tienen la mayor probabilidad de sobrevivir en el proceso de evolución. Los segundos, tratan de emular el comportamiento de algunos organismos para conducir el proceso de búsqueda en el espacio de decisión. Se comentan a continuación los algoritmos que parecen ser los más frecuentemente usados acorde con los resultados de la presente revisión.

#### 3.1. Tipos de algoritmos para optimización mono-índice

De acuerdo con este trabajo, las estrategias más populares (para la ingeniería de control) en los que se basan los diferentes EA's parecen ser los Algoritmos Genéticos (*genetic algorithms*, GA), la optimización por enjambre de partículas (*particle swarm optimization*, PSO) y la Evolución Diferencial (*Differential Evolution*, DE). No existe una técnica evolutiva mejor que las otras, dado que cada una de ellas tiene sus ventajas y desventajas. A continuación, serán expuestas brevemente tales estrategias.

##### Algoritmo I: GA.Básico

- 1 : Generar población inicial  $P|_0$  con  $N_p$  cromosomas.
- 2 : Evaluar  $P|_0$ .
- 3 : MIENTRAS criterio de terminación no alcanzado
- 4 : Leer contador de generaciones  $G$ .
- 5 : Crear nueva población  $P|_G^*$  a partir de  $P|_G$ .
- 6 : Seleccionar cromosomas padres
- 7 : Efectuar el operador de cruce para crear un hijo a partir de los padres.
- 8 : Efectuar el operador mutación en el hijo.
- 9 : Seleccionar los cromosomas más aptos de  $P|_G^* \cup P|_G$  para generar población  $P|_{G+1}$
- 10 :  $G = G + 1$ .
- 11 : FIN-MIENTRAS

#### 3.1.1. Algoritmos Genéticos

Los algoritmos genéticos (GA por sus siglas en inglés) (Holland (1975); Goldberg (1989)) se basan en un símil con la evolución de las especies. Cada individuo (posible solución del problema de optimización) está caracterizado por un cromosoma (codificación de esa solución). El algoritmo genético se basa en hacer evolucionar un conjunto de individuos (población) para conseguir que sus individuos se acerquen a la solución óptima con el paso de cada generación (iteración del algoritmo). El

mecanismo de evolución básico consiste en ir generando nuevos individuos modificando los cromosomas de los individuos de la población existente, bien mezclándolos entre si (cruce), bien efectuando un cambio aleatorio (mutación). Además de la generación de nuevos individuos, existe un proceso de selección para determinar los individuos que siguen en la población. En la selección se promueve la supervivencia de los individuos que tienen mejor función objetivo. Existen multitud de implementaciones diferentes basados en este mecanismo (Konnak et al. (2006)). En el Algoritmo I, se presenta uno de los posibles pseudo-código para un GA básico.

---

**Algoritmo II: PSO.Basico**

---

- 1 : Generar el enjambre inicial  $P|_0$  con  $N_p$  partículas.
  - 2 : Evaluar  $P|_0$ .
  - 3 : Inicializar para cada partícula su mejor posición conocida  $\mathbf{x}^j|_{best}$ .
  - 4 : Inicializar la mejor posición conocida del enjambre  $\mathbf{x}^{swarm}|_{best}$ .
  - 5 : Inicializar la velocidad de cada partícula  $\dot{\mathbf{x}}^j|_0$ .
  - 6 : MIENTRAS criterio de terminación no alcanzado
  - 7 : Leer contador de generaciones  $G$ .
  - 8 : HACER para cada partícula  $\mathbf{x}^j \in P|_G$
  - 9 : Actualizar su velocidad  $\dot{\mathbf{x}}^j|_G$  (Ecuación 5)
  - 10 : Actualizar su posición  $\mathbf{x}^j|_G$  (Ecuación 6)
  - 11 : Actualizar la mejor posición conocida  $\mathbf{x}^j|_{best}$
  - 12 : Actualizar el conocimiento colectivo del enjambre  $\mathbf{x}^{swarm}|_{best}$
  - 13 : FIN-HACER
  - 14 :  $G = G + 1$ .
  - 15 : FIN-MIENTRAS
- 

### 3.1.2. Enjambre de partículas

La optimización por enjambre de partículas (PSO por sus siglas en inglés) (Kennedy y Eberhart (1995); Coello (2011)) busca un compromiso entre conocimiento individual de cada partícula y conocimiento colectivo del enjambre. Cada partícula está determinada por su posición  $\mathbf{x}$  y velocidad  $\dot{\mathbf{x}}$  en la generación  $k$ :

$$\dot{\mathbf{x}}^j|_k = \dot{\mathbf{x}}^j|_{k-1} + \varphi_1\beta_1(\mathbf{x}^j|_{best}) + \varphi_2\beta_2(\mathbf{x}^{swarm}|_{best}) \quad (5)$$

$$\mathbf{x}^j|_k = \mathbf{x}^j|_{k-1} + \Delta\mathbf{x}^j|_k \quad (6)$$

$$\Delta\mathbf{x}^j|_k \approx \dot{\mathbf{x}}^j|_k \quad (7)$$

donde  $\varphi_1, \varphi_2$  son factores empleados para ponderar el conocimiento local y global;  $\beta_1, \beta_2$  son números aleatorios;  $\mathbf{x}^{swarm}|_{best}$  es la mejor posición (solución) conocida por el enjambre (conocimiento colectivo) mientras que  $\mathbf{x}^j|_{best}$  es la mejor posición conocida para la partícula  $j$  (conocimiento individual). En el Algoritmo II, se presenta el pseudo-código para un PSO estándar.

### 3.1.3. Evolución Diferencial

El algoritmo de Evolución Diferencial (DE por sus siglas en inglés) (Storn y Price (1997); Mezura-Montes et al. (2008); Das y Suganthan (2010)) es utilizado principalmente por su simplicidad y compactibilidad (Das y Suganthan (2010)). Existen muchas versiones primales del algoritmo; la versión original (y

de alguna forma estándar) del algoritmo (Storn y Price (1997)) emplea tres operadores: mutación (ecuación 8), recombinación (ecuación 9) y selección (ecuación 10).

$$\mathbf{v}^i|_k = \mathbf{x}^{r_1}|_k + F(\mathbf{x}^{r_2}|_k - \mathbf{x}^{r_3}|_k) \quad (8)$$

$$\mathbf{u}^i_{k,j} = \begin{cases} \mathbf{v}^i_{k,j} & \text{si } rand(0, 1) \leq Cr \\ \mathbf{x}^i_{k,j} & \text{de otra forma} \end{cases} \quad (9)$$

$$\mathbf{x}^i|_{k+1} = \begin{cases} \mathbf{u}^i|_k & \text{si } f(\mathbf{u}^i|_k) \leq f(\mathbf{x}^i|_k) \\ \mathbf{x}^i|_k & \text{de otra forma} \end{cases} \quad (10)$$

donde  $r_1 \neq r_2 \neq r_3 \neq i$ ,  $F$  se conoce como el factor de escalado y  $Cr$  es la probabilidad para llevar a cabo la recombinación. En el Algoritmo III, se presenta el pseudo-código para un DE básico.

---

**Algoritmo III: DE.Basico**

---

- 1 : Generar población inicial  $P|_0$  con  $N_p$  individuos.
  - 2 : Evaluar  $P|_0$ .
  - 3 : MIENTRAS criterio de terminación no alcanzado
  - 4 : Leer contador de generaciones  $G$ .
  - 5 : HACER para cada  $\mathbf{x}_i \in P|_G$
  - 6 : Generar un vector mutante (ecuación 8)
  - 7 : Generar un vector hijo (ecuación 9)
  - 8 : Efectuar una comparación por pares Padre-Hijo (ecuación 10)
  - 9 : FIN-HACER
  - 10 :  $G = G + 1$ .
  - 11 : FIN-MIENTRAS
- 

---

**Algoritmo IV: MOEA.Basico**

---

- 1 : Generar población inicial  $P|_0$  con  $N_p$  individuos.
  - 2 : Evaluar  $P|_0$ .
  - 3 : Construir la aproximación inicial al conjunto de Pareto  $X_p^*|_0$
  - 4 : MIENTRAS criterio de terminación no alcanzado
  - 5 : Leer contador de generaciones  $G$ .
  - 6 : Generar población  $P^*|_G$  a partir de  $P^*|_G$  usando algún esquema evolutivo.
  - 7 : Construir la aproximación al conjunto de Pareto  $X_p^*|_G$  con  $X_p^*|_{G-1} \cup P^*|_G$
  - 8 : Actualizar la población  $P|_{G+1}$  con  $P^*|_G \cup P|_G$
  - 9 :  $G = G + 1$ .
  - 10 : FIN-MIENTRAS
- 

### 3.2. Tipos de algoritmos para optimización multi-índice

Los EA's anteriores suelen ser usados ampliamente para aproximar frentes de Pareto en la optimización multi-índice. Sin embargo, algunos cambios estructurales en la etapa de selección de los individuos en cada generación debe ser efectuada. Es decir, ahora debe llevarse a cabo una optimización que mejore con cada iteración la aproximación al frente de Pareto. Lo anterior puede llevarse a cabo de varias formas. Por mencionar algunas, usando indicadores de dispersión, indicadores de calidad del frente o por medio de archivos externos. En el algoritmo IV se muestra un código de uso general para un EA orientado a

resolver optimización multi-índice/multi-objetivo (MOEA por sus siglas en inglés). Una revisión del estado actual de dichos algoritmos puede ser consultado en Coello Coello (2006) y en Zhou et al. (2011)

### 3.3. Características deseables en los algoritmos evolutivos: tendencias actuales y líneas de investigación.

Cualesquiera la técnica evolutiva o bio-inspirada empleada para la optimización en un problema de diseño, algunas características deseables por el diseñador son requeridas. Las mismas dependen del tipo de problema que se busca resolver, y forman una pauta para tener una imagen global en las tendencias de investigación sobre EA's. Destacan como características deseables (sin orden particular) la convergencia, la diversidad de soluciones, el manejo de preferencias, el tratamiento de restricciones, la escalabilidad, la optimización en entornos dinámicos, la optimización robusta y la optimización de funciones con alto coste computacional.

#### 3.3.1. Convergencia

La convergencia es quizá la primer característica que se desea en un EA para garantizar que encontrará la solución óptima deseada. En esta línea, es común hablar de la capacidad de explotación (*exploitation*) versus la capacidad de búsqueda (*exploration*) de un algoritmo (Eiben y Schippers (1998)). La primera refiere a, habiéndose encontrado una buena solución, emplearla para encontrar la solución óptima; la segunda, a la capacidad de continuar explorando el espacio de búsqueda, para encontrar soluciones en otras áreas y evitar caer en mínimos locales.

Es sabido que el rendimiento de los EA's en este punto suele ser sensible al ajuste de los parámetros internos del optimizador (probabilidad de mutación en GA, ponderación del conocimiento global y local en PSO, factor de escalado en DE, por ejemplo). La calidad de la solución (y el tiempo que se tarde en encontrarla) puede llegar a depender enormemente en los ajustes del mismo. En algunas ocasiones se requiere de un conocimiento profundo del problema y del optimizador para tener una idea del ajuste en los parámetros del EA.

Trabajos recientes en el diseño de EA's suelen incluir análisis exhaustivos en la sensibilidad de sus parámetros empleando funciones de benchmark conocidas; en su caso, se pueden ofrecer guías para la selección de los parámetros internos. Así mismo, dentro del marco de las competiciones en optimización que organiza el Congreso en Computación Evolutiva de la IEEE, se busca promover algoritmos *todo propósito* con parámetros generales. En algunos casos, se han propuesto algoritmos auto-ajustables, para minimizar el número de parámetros por ajustar. Esto con el fin de ofrecer herramientas fáciles de usar a los usuarios interesados. En el Apéndice A se muestra un breve ensayo con dos algoritmos de dicho repositorio en un problema de ajuste de controladores.

Por otro lado, técnicas usadas con regularidad para mejorar la convergencia del EA involucran el empleo de técnicas clásicas de optimización local. Lo anterior puede efectuarse dentro del mismo proceso evolutivo, o al final usando la mejor solución

encontrada por el EA como aproximación inicial. Emplearlos dentro del proceso evolutivo podría por un lado, acelerar la convergencia a solución, pero por otro, inducir la misma a mínimos locales por la presión en el proceso evolutivo por encontrar una solución.

#### 3.3.2. Diversidad de las Soluciones

En algunas ocasiones es deseado encontrar no una, sino un conjunto de soluciones óptimas, incluso para los problemas del tipo SOP y AOF. Tal caso puede referirse a la necesidad de resolver un problema multi-modal, donde diferentes vectores de decisión apuntarán hacia la misma solución óptima. Por tanto, se desea contar con ese conjunto de soluciones, diferentes entre sí en el espacio de decisión. La intención de obtener tantas soluciones óptimas (ya sean óptimo globales u óptimo locales) para seleccionar de entre ellas la más conveniente puede justificarse cuando es relevante el significado físico del vector de decisión. Es decir, que el espacio de decisión representa variables físicas cuya selección conviene ser evaluada por factores de preferencia. Para más detalles puede consultarse el trabajo de Das et al. (2011).

En el caso de MOOD, además de poder ser deseable la diversidad con respecto al espacio de decisión, suele ser deseada (exigida muchas veces en realidad) la diversidad de soluciones con respecto al espacio de objetivos (frente de Pareto). Lo anterior para que el diseñador pueda contar con una imagen global fidedigna del intercambio de prestaciones y poder apreciar diferentes grados de compromiso y efectuar mejor su selección.

En ambos casos, ya se busque diversidad de soluciones en el espacio de decisión o en el espacio de objetivos, se debe tener especial cuidado en el criterio de diversidad elegido. Un criterio de diversidad mal definido podría entrar en conflicto con la capacidad de convergencia del algoritmo. Lo anterior le demandará más tiempo en el proceso de selección e inclusive, la necesidad de volver a ejecutar el proceso de optimización. Por otro lado, puede orientar al EA a seleccionar soluciones que, aunque diversas, serán de poco interés para el diseñador. Esta propiedad de tener un conjunto de soluciones pero relevantes para el diseñador, se conoce como pertinencia de la solución (*pertinency*) donde puede resultar deseable incorporar las preferencias del diseñador en alguna etapa del proceso de optimización.

#### 3.3.3. Manejo de Preferencias

El manejo de preferencias para reforzar la pertinencia consiste en la capacidad del algoritmo para evolucionar hacia soluciones útiles para el diseñador. Dicha capacidad está relacionada con la forma en que se define el problema de optimización. Como ha señalado Fonseca y Fleming (1998b), muchas veces el punto de vista de como llevar a cabo la optimización se hace con respecto al optimizador, y no al diseñador. En el caso de un SOP, este problema no es relevante, dado que no hay objetivos que compitan entre sí. Sin embargo, en el caso de un MOP, se puede dirigir al optimizador hacia zonas de poco interés práctico para el diseñador.

Cuando se aborda el problema de diseño por AOF, se buscará garantizar que la función de agregación refleje adecuada-

mente el balance de prestaciones deseado. Tomando como base el ejemplo clásico de asignar un vector de pesos a los índices de interés, se observa como en ocasiones el intercambio de prestaciones de la solución resultante no satisface del todo al diseñador. Cuando esto sucede se suele intentar nuevamente con un vector de pesos diferentes hasta lograr el balance que le parece el adecuado. Suele ser normal elaborar varios ensayos para encontrar una correcta (o preferible) combinación (Fonseca y Fleming (1998a)). En el caso de diseño por MOOD, el diseñador tiene la opción de analizar el conjunto de soluciones, y centrar su atención en la zona que cumpla con sus preferencias. Sin embargo, se debe prestar atención a que el algoritmo evoluciones hacia las regiones del frente de Pareto que con prácticas, relevantes y de interés para el diseñador. Es decir, que en ciertos casos el nivel de degradación de un objetivo no justifica la mejora de los otros, por lo que la solución estará fuera de toda consideración.

Métodos para incorporar las preferencias del diseñador en problemas de optimización, y hacer más transparente su formulación, son discutidos en trabajos como los de Miettinen (1998); Marler y Arora (2004) para AOF y de Coello (2000) para MOOD. Algunas técnicas habituales suelen ser *goal attainment*, *lógica borrosa* y *physical programming* (Messac (1996); Sanchis et al. (2010)).

- *Goal attainment*. Se define un valor a alcanzar para cada objetivo, y se minimiza esa distancia.
- *Lógica borrosa*. Son empleadas funciones de membresía para especificar las relaciones deseadas.
- *Physical programming*. Expresa preferencias de valor para cada objetivo por medio de etiquetas como *Intolerable*, *Tolerable* y *Deseable*; después se emplea una transformación de dominio, donde todos los objetivos serán comparables entre sí.

Una de las formas habituales de mejorar la pertinencia de la solución es mediante la incorporación de restricciones en el problema de diseño. Esta capacidad de manejar restricciones es a su vez, otra característica potencialmente deseable en el optimizador, por lo que será discutida a continuación.

### 3.3.4. Tratamiento de Restricciones

Se trata de la capacidad para manejar las restricciones impuestas por el diseñador durante el proceso de optimización. La misma es relevante cuando se considera que la mayoría de los problemas de ingeniería incluyen diferentes niveles de restricción, que pueden ir desde restricciones de operación, implementación, físicas, entre otras. El tratamiento de restricciones en los algoritmos evolutivos es un tema que ha sido constantemente abordado (Fonseca y Fleming (1998a); Coello (2002); Mezura-Montes y Coello (December 2011)). Algunas técnicas habituales son:

- Reglas de factibilidad. Un ejemplo sencillo y clásico es el expuesto por Deb (2000). El mismo consiste en tres reglas sencillas:

1. Al comparar dos soluciones factibles, aquella que tenga la mejor función objetivo es seleccionada.
2. Al comparar una solución factible con una no-factible, la primera es seleccionada sobre la segunda.
3. Al comparar dos soluciones no-factibles, aquella con el menor grado de violación de las restricciones es seleccionada.

Las funciones de penalización determinan un tipo de regla de factibilidad.

- *$\epsilon$ -constrained*. Éste método emplea un ordenamiento lexicográfico donde la minimización de la violación de la restricción precede a la minimización de la función de coste del problema. Dicho mecanismo, junto con un mecanismo de adaptabilidad<sup>1</sup> ganó la competición en optimización real con restricciones (Mallipeddi y Suganthan (2009)) del Congreso en Computación Evolutiva de la IEEE en 2010.
- Filosofía MOP. Esto es, incorporar las restricciones como nuevos objetivos de diseño, que deben ser cumplidos en cierta medida (pertinencia de soluciones).

### 3.3.5. Escalabilidad

Se refiere principalmente a la capacidad de algoritmo para tratar con problemas de optimización de cualquier tamaño con tiempos de cómputo y recursos computacionales razonables para el diseñador. Esto es ser eficiente (computacionalmente hablando) para encontrar soluciones con pocas variables de decisión ó cientos de ellas (*large scale optimization*). Puede ocurrir que ciertos algoritmos ofrezcan soluciones con recursos computacionales razonables para pocas variables, pero que sean poco prácticos para un número mayor. El lector interesado puede consultar más información en Lozano et al. (2011).

En el caso de la optimización multi-índice, la escalabilidad también refiere al número de objetivos por optimizar. Una vez más, puede ocurrir que algoritmos sean eficientes para pocos objetivos, pero que a medida que crece el número de objetivos (*many objectives optimization*), crezca de forma exponencial el tiempo de cómputo necesario para aproximar un frente de Pareto. Un MOOD con cuatro índices o más para optimizar suele considerarse como *many objectives optimization*. En ese supuesto puede ocurrir que los mecanismos de convergencia y diversidad de un EA entren en conflicto. En tales casos, se ha encontrado que los algoritmos pueden llegar a perder su efectividad, al grado que simples búsquedas estocásticas pueden ser competitivos con ellos, como señala Corne y Knowles (2007). El lector interesado puede consultar el trabajo de Ishibuchi et al. (2008).

### 3.3.6. Optimización en Entornos Dinámicos

La optimización en entornos dinámicos refiere principalmente a que los objetivos de diseño varían en función del tiempo y del vector de variables de decisión, es decir:

<sup>1</sup>Código disponible en <http://www.ints.info.hiroshima-cu.ac.jp/~takahama/eng/index.html> para optimización mono-objetivo.

$$\min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{J}(\mathbf{x}, t) = [J_1(\mathbf{x}, t), \dots, J_m(\mathbf{x}, t)] \quad (11)$$

sujeto a:

$$\mathbf{g}(\mathbf{x}, t) \leq 0 \quad (12)$$

$$\mathbf{h}(\mathbf{x}, t) = 0 \quad (13)$$

$$\underline{x}_i \leq x_i \leq \bar{x}_i, i = [1, \dots, n] \quad (14)$$

En ocasiones el enfoque estático es suficiente, es decir, ejecutar un nuevo problema de optimización cada vez que sea necesario. Sin embargo, en ocasiones lo anterior puede no ser suficiente y otros tipos de estrategia pueden ser necesarias. Cruz et al. (2011) ofrecen una revisión extensa del área.

### 3.3.7. Optimización Robusta

Se habla de optimización robusta cuando existen (por ejemplo) incertidumbres en el(los) modelo(s) que se usa(n) en el proceso de optimización o alta sensibilidad entre las variables de decisión y el valor óptimo deseado (Beyer y Sendhoff (2007)).

En lo referente al primer caso el área de ingeniería de control está habituada a considerar imprecisiones en los modelos que emplea para el ajuste de controladores. Es sabido lo necesario que es garantizar cierto nivel de robustez para que el controlador sea capaz de mantener su rendimiento ante dinámicas no modeladas en el modelo nominal. Normalmente lo anterior suele ser resuelto incorporando restricciones en el problema de optimización (máximo valor de la función de sensibilidad por ejemplo). Sin embargo en otros casos, esta robustez puede estar vinculada con la confiabilidad del sistema, donde se busca garantizar determinado rendimiento ante diversos escenarios (diseño para evitar la falla en el sistema).

Con respecto a la sensibilidad de las variables, en ocasiones es deseable que el algoritmo pueda incorporar la flexibilidad del diseñador por considerar soluciones sub-óptimas (y preferirlas en último caso) si las mismas presentan un mayor grado de robustez en el espacio de objetivos. En otras palabras, considerar que tan sensible es la solución a un cambio en los valores del vector de decisión. Muchas veces lo anterior es de importancia cuando las variables de decisión representan variables físicas, donde la consideración de las tolerancias es un factor importante que no puede ser pasado por alto.

### 3.3.8. Optimización con alto coste computacional

En ocasiones la función a optimizar requiere es de alto coste computacional, por lo que es deseable un algoritmo que con pocas evaluaciones de la misma pueda ser capaz de converger a solución. En el caso de la ingeniería de control, lo anterior puede estar relacionado con simulaciones complejas que sea necesario ejecutar para calcular los índices a optimizar. Las tendencias para afrontar esta problemática se orientan principalmente a generar una superficie aproximada *on-the-fly* del espacio de objetivos, de menor coste computacional. Una de las técnicas populares es el uso de redes neuronales, que es entrenada periódicamente dentro del proceso evolutivo del optimizador. Revisiones extensas en el tema son expuestas en los trabajos

de Shi y Rasheed (2010) y Santana-Quintero et al. (2010) para AOF y MOOD respectivamente.

En otros casos, este alto coste computacional puede ser relativo; es decir, con respecto a la potencia de cálculo del que se dispone (recursos computacionales limitados) para determinada aplicación. En esa línea, se han propuesto EA's *compactos* (cEA's por sus siglas en inglés) como una alternativa para efectuar la optimización con recursos computacionales limitados. Algunos ejemplos pueden consultarse en los trabajos Harik et al. (1999) y Mininno et al. (2011).

### 3.3.9. Conclusiones sobre tendencias y características deseables en los EA's

Se han comentado una serie de características deseables de los EA's, que serán exigidas en mayor o menor grado por el diseñador. Tal exigencia estará relacionada con el tipo de problema que se necesita resolver y el tipo de solución que se desea encontrar.

Los puntos anteriores ofrecen una pauta para apreciar hacia donde se orienta la investigación en EA's y el tipo de soluciones que buscan ofrecer al diseñador. Acorde con Roy et al. (2008) las cinco tendencias más populares para abordar problemas de diseño por optimización son: optimización multi-disciplinaria (emplear variables de decisión de distintas disciplinas dentro del mismo problema de optimización), la optimización robusta, la optimización con incertidumbres, la optimización basada en confiabilidad y la optimización multi-objetivo. Como se puede apreciar, los EA's se orientan hacia ellos para ofrecer herramientas al diseñador.

### 3.4. Disponibilidad

Existen una serie de herramientas disponibles para ser empleadas en el diseño por optimización; ello facilita la labor del diseñador, ya que puede centrarse, si así lo desea, en la definición del problema y no en el desarrollo de aplicaciones. Algunas soluciones pueden encontrarse a través de Matlab© central<sup>2</sup>. Adicionalmente, se encuentran disponibles *toolboxes* para Matlab© de GA<sup>3</sup> y para el algoritmo DE<sup>4</sup> (diversas plataformas). Así mismo, se cuenta con un repositorio<sup>5</sup> de algoritmos participantes en las diferentes competiciones de optimización del Congreso en Computación Evolutiva de la IEEE. El repositorio cuenta con algoritmos tanto para SOP, MOP, optimización dinámica y multi-modal. Este repositorio representa un buen punto de inicio para probar herramientas existentes. Dependiendo del caso, se puede buscar incorporar en ellas alguna característica deseable adicional, o mejorar alguna ya existente.

En el caso de algoritmos multi-objetivo, existen soluciones como MOEA/D de Zhang y Li (2007)<sup>6</sup> (actualmente uno de

<sup>2</sup><http://www.mathworks.com/matlabcentral/fileexchange/>

<sup>3</sup>*Genetic Algorithm toolbox* para Matlab© disponible en <http://www.sheffield.ac.uk/acse/research/ecrg/gat>

<sup>4</sup><http://www.icsi.berkeley.edu/~storn/code.html>

<sup>5</sup><http://www.ntu.edu.sg/home/epsugan/>

<sup>6</sup><http://cswww.essex.ac.uk/staff/zhang/IntrotoResearch/MOEAAd.htm>

los más potentes) y NSGA-II de Deb et al. (2002)<sup>7</sup> (quizás el más popular en la actualidad; el Toolbox de optimización de Matlab© incluye una variante del mismo). Así mismo, aunque no implementa EA's, el ACADO© Matlab© *toolbox* para control automático<sup>8</sup> implementa soluciones clásicas para MOP del área de control.

En el caso de MOOD, existen visualizadores para la etapa de selección como VIDEO de Kollat y Reed (2007), útil para analizar hasta 4 dimensiones; diagramas de dispersión ( SCp ), *Parallel Coordinates* ( PAC ) de Inselberg (1985)<sup>9</sup> y *Level Diagrams* ( LD ) de Blasco et al. (2008) y Reynoso-Meza et al. (2013)<sup>10</sup>. Una herramienta de visualización que además incluye librerías para analizar datos multi-dimensionales es TULIP desarrollada por INRIA<sup>11</sup>.

A continuación, se comentará como se han utilizado las técnicas descritas anteriormente (y otras) para mejorar el rendimiento de los controladores del tipo PID.

#### 4. Aplicaciones en controladores PID

En su mayoría, los EA's han sido empleados para ajustar los parámetros de controladores PI, PID, PD+I, de uno y dos grados de libertad; también han sido empleados para estructuras no convencionales del PID, como son controladores PID borrosos o de orden fraccionario, descritos en Podlubny (1999). Otras aplicaciones para mejorar el rendimiento de este tipo de controladores se ha llevado a cabo en una capa superior, utilizando los EA's para obtener un diseño multi-disciplinar, que involucre tanto al diseño de control, como el diseño mecánico y electrónico (por ejemplo). Existe una tendencia mayor en justificar el empleo de los EA's por su capacidad de optimizadores globales, mejorando la convergencia (por medio de su capacidad de exploración) de las soluciones; sin embargo, su flexibilidad para lidiar con restricciones no lineales y emplear funciones de coste con mayor significado para el diseñador son también suficientes para justificar su uso. Se comentarán con particular interés aplicaciones de los últimos diez años, dado que aplicaciones previas han sido tratadas en el compendio elaborado por Fleming y Purshouse (2002).

Como dato de interés, en Koza et al. (2003), un algoritmo basado en *genetic programming* (GP) (Koza y Poli (2005)) ha mostrado ser capaz de *redescubrir* el controlador PID para aplicaciones específicas. Los algoritmos basados en GP son algoritmos que son capaces de construir expresiones, cuya complejidad depende de los operadores básicos que se seleccionan en el GP. Pueden ser empleados para modelar procesos o para emular comportamientos (como los de un controlador). En el anterior trabajo, destaca como este tipo de algoritmos determinan que las mejores estructuras para control (y a la vez sencillas) se acercan a los controladores PID.

El empleo de EA's para el ajuste de controladores PID parece haberse popularizado principalmente bajo el esquema AOF, como puntualizan Li et al. (2006). Existen una gran variedad de trabajos al respecto, definiendo diferentes tipos de funciones de coste; En el caso de las técnicas MOOD, poco a poco comienzan a presentarse como alternativa para el ajuste de controladores.

Tabla 1: Codificación de controladores con estructura proporcional, integral y derivativa.

Estructura	Controlador
P	Proporcional.
PI	Proporcional - Integral.
PID	Proporcional - Integral - Derivativo.
PID/N	PID con filtro de la derivada.
PI <sup>1</sup>	PI con ponderación de la referencia.
PID <sup>2</sup>	PID con ponderación de la referencia y la derivada.
PID <sub>NxN</sub>	Arreglo de N controladores PID para un proceso NxN.
PID <sub>Fuzzy</sub>	PID borroso
PI <sup>λ</sup> D <sup>μ</sup>	PID fraccionario

En las tablas 2, 3 y 4 se aprecia un resumen de las aplicaciones para PID convencional, PID borroso y PID fraccionario respectivamente. En todo caso, se comentan las características del problema de diseño (número de variables de decisión, número de objetivos, número de restricciones de desigualdad, y número de restricciones de igualdad); el tipo de algoritmo usado, la estrategia para el manejo de restricciones y el manejo de preferencias (adicional al empleo de restricciones). Se han elegido las anteriores como características deseables mínimas para problemas de control, mientras que propiedades como convergencia y diversidad son asumidas como garantizadas. Esto implica también que trabajos donde: 1) se expone simplemente la mejora en convergencia para un caso particular; 2) la mejora para aproximar un frente de Pareto de un EA ó 3) la simple prueba de concepto de que un EA resuelve un problema son omitidos intencionalmente. Solo análisis profundos en tales aspectos son considerados, para que sirvan de guía a posibles usuarios del área de control. Cualquier otra característica deseable de interés será comentada particularmente. Esto con el fin de identificar claramente las capacidades de los EA's para cubrir otras características deseables del proceso de optimización (sección 3.3). Se emplea la codificación de la tabla 1 para identificar la estructura del controlador PID empleado en cada caso.

##### 4.1. Ajuste de controladores convencionales del tipo PID

Como es de esperar, los controladores PID convencionales son los más extensamente explotados por el uso de EA's. Se encuentran trabajos orientados en diseño para SOP y MOP; además, se cuentan aportaciones para diseño multi-disciplinar y para recursos computacionales limitados. En la tabla 2 se muestra el resumen de las aplicaciones que se comentan a continuación.

<sup>7</sup><http://www.iitk.ac.in/kangal/codes.shtml>

<sup>8</sup>Disponible en <http://www.acadotoolkit.org/>.

<sup>9</sup>Herramienta disponible en el *statistics toolbox* de Matlab©.

<sup>10</sup>Disponible para Matlab© en: <http://www.mathworks.com/matlabcentral/fileexchange/24042> y <http://www.mathworks.com/matlabcentral/fileexchange/39458>

<sup>11</sup>Disponible en <http://tulip.labri.fr/TulipDrupal/>

Tabla 2: Resumen de aplicaciones en ajuste de controladores convencionales del tipo PID. En algunos casos, el número de objetivos puede incluir restricciones. Un asterisco antecedendo a la estructura de control específica una optimización mono-índice. La codificación del tipo de estructura de control se muestra en la tabla 1.

Estructura	Aplicación / Proceso	Referencia	Problema de Diseño			Detalles sobre:			
			$J(x)$	$x$	$g(x)-h(x)$	Optimizador	Restricciones	Preferencias	Características Deseables
*PID	Electromecánico	Gaing (2004)	4	3	(0,0)	PSO	-	-	Sección 3.3.1
*PID <sub>2x2</sub>	Aeronáutico	Juang et al. (2008)	5	12	(4,0)	GA	Penalización	Método Gráfico	
*PID	SISO	Zhang et al. (2009)	3	3	(1,0)	GA	Penalización		Sección 3.3.1
*PID <sub>NxN</sub>	Químico	Iruthayarajan y Baskar (2009)	2	6	(0,0)	Varios	-	-	Sección 3.3.5
*PID <sub>SL</sub>	Robótica	Lin et al. (2011)	1	3	(0,0)	PSO	-	-	Sección 3.3.6
*PID <sub>NxN</sub>	Químico	Menhas et al. (2012b)	1	6	(0,0)	PSO's	-	Índices Auxiliares	
*PID <sub>2x2</sub>	Mecánico	Menhas et al. (2012a)	2	6	(0,0)	PSO	-	Índices Auxiliares	Sección 3.3.1
*PID <sup>2</sup>	Eléctrico	Elgammal y Sharaf (2012)	5	5	(0,0)	PSO	-	-	
*PID	SISO	Romero-Pérez et al. (2012)	1	4	(3,0)	DE	Penalización	-	Sección 3.3.8
*PID	Mecatrónico	Avigad et al. (2003)	1	6	(0,0)	GA	-	-	Multi-disciplinar
*PID	Mecatrónico	Behbahani y de Silva (2008)	1	7	(4,0)	GA	Penalización	Acorde a caso ideal	Multi-disciplinar
*PID	Mecatrónico	Jiachuan et al. (2005)	1	-	(0,0)	GP	-	-	Multi-disciplinar
*PI	Químico	Lamanna et al. (2009)	3	5	(7,2)	GA	Penalización	Usando un caso Ideal.	Multi-disciplinar
PID <sup>2</sup> /N, PI <sup>1</sup>	SISO, MIMO	Herreros et al. (2002)	4	(7,4)	(1,0)	GA	MOP	Visualización 3D, SCp; Comparación de conceptos.	Sección 3.3.5
PI <sup>1</sup>	SISO	Tavakoli et al. (2007)	4	3	(0,0)	GA	-	Visualización 3D, 2D; Ofrece una metodología para ajuste.	Sección 3.3.5
PID/N	SISO	Huang et al. (2008)	3	4	(4,0)	Ad hoc	Penalización	Visualización SCp; Comparación con objetivos en el dominio del tiempo.	
PID	MIMO, Aeronáutico	Hung et al. (2008)	3	27	Mixed	Ad hoc	-	Visualización SCp; Análisis con otras técnicas.	
PID <sup>2</sup> /N	SISO	Reynoso-Meza et al. (2009)	15	7	(6,0)	GA	MOP	Visualización LD; Selección conforme a preferencias.	Sección 3.3.5
PI <sup>1</sup>	SISO	Reynoso-Meza et al. (2011a)	3	3	(2,0)	DE	MOP	Visualización LD; Análisis en el intercambio de prestaciones.	
PI	Químico	Reynoso-Meza et al. (2012c)	7	4	(3,0)	DE	MOP	Visualización LD; Análisis en el intercambio de prestaciones.	Sección 3.3.5
PID	Eléctrico	Sidhartha y Panda (2011)	2	3	(0,0)	NSGA-II	-	Selección conforme a un sistema de inferencia borroso.	
PID, I-PD	Medicina	Algoul et al. (2011)	2	3	(3,0)	GA	Penalización	Visualización SCp; Comparación de conceptos.	
P + PI (3)	Químico	Xue et al. (2010)	6	8	(1,0)	NSGA-II	Penalización	Visualización SCp; Nuevo indicador incluido en la etapa de selección.	Sección 3.3.5
PID	Químico, Aeronáutico	Zhao et al. (2011)	2	(12, 27)	(1,0)	PSO	MOP	Visualización SCp; Selección con base a Norma-1	
SS, PID(2)	Aeronáutico	Reynoso-Meza et al. (2012b)	5	16, 6	(1,0)	DE	MOP	Visualización LD; Comparación de alternativas y conceptos de diseño.	Sección 3.3.5
PID	Robótica	Ayala y dos Santos Coelho (2012)	2	6	(0,0)	NSGA-II	-	Visualización 2D; Selección acorde a la media armónica.	
PID <sup>1</sup>	Químico	Tan et al. (2005)	6	4	(4,0)	GA	En la Simulación	Visualización PAC; Selección acorde a criterio ISE.	Sección 3.3.5
PD	Mecatrónico	Rao y Tiwari (2009)	5	8	(13,0)	NSGA-II	Reglas de factibilidad	Visualización SCp; Selección acorde a la solución utópica.	Multi-disciplinar

#### 4.1.1. Optimización Mono-Índice

En el caso de ajuste de PID de estructura convencional, se cuentan trabajos como el de Gaing (2004) que aborda requerimientos en el dominio del tiempo. Esto se hace definiendo una función de coste en términos de sobre-impulso, tiempo de elevación, tiempo de establecimiento y error en régimen permanente. En dicho trabajo se pone un énfasis particular en mostrar un estudio sobre las propiedades de convergencia de algoritmos del tipo PSO sobre los GA para aplicaciones de regulación automática de voltaje.

En Juang et al. (2008) se usa un GA para ajustar los parámetros de un controlador PID, con aplicación específica a un *Twin Rotor MIMO System*. Se definen objetivos en el dominio del tiempo, y se desarrolla una etapa para traducir especificaciones de diseño en objetivos de optimización. Esta definición se logra seleccionando la respuesta deseada en el tiempo (mejorando así la pertinencia de las soluciones) por métodos gráficos. Éste es un ejemplo de como incorporar la pertinencia de soluciones en la formulación del problema de optimización. Dicho esquema pretende ofrecer un grado diferente de interpretabilidad de las soluciones, y con ello, aumentar su pertinencia.

En Zhang et al. (2009) se emplea un GA para optimizar una función que agrupa 3 objetivos de diseño en el dominio del tiempo (IAE, esfuerzo de control y tiempo de elevación) y una restricción (sobre-impulso, incorporado como una penalización en la función de coste). El trabajo se centra en el desarrollo de un nuevo sistema de selección en cada generación, para seleccionar a los individuos más aptos y mejorar la convergencia. Con ello, se busca evitar la convergencia prematura del algoritmo; es decir, mejorar la capacidad de exploración sin sacrificar la capacidad de explotación.

En Iruthayarajan y Baskar (2009), se ajusta un controlador PID para procesos MIMO, empleando como función de coste el IAE. El trabajo plantea la comparación de diferentes EA's y el algoritmo CMA-ES<sup>12</sup>, que es un algoritmo perteneciente al grupo de EDA's (*Estimation distribution algorithms*); tales algoritmos suelen ser considerados como un punto intermedio entre algoritmos puramente estocásticos y determinísticos (Hansen (2006)). Este trabajo es importante en el sentido que empieza a bordar la escalabilidad en el ajuste de controladores PID, al considerar procesos MIMO.

En Lin et al. (2011) un PID con esquema auto-aprendizaje es ajustado en línea; en la capa de aprendizaje es utilizado un algoritmo del tipo PSO para ajustar continuamente los coeficientes de aprendizaje. El esquema es probado en simulación con un manipulador de dos grados de libertad. En este caso, el EA es empleado para mejorar el rendimiento de una capa superior del controlador PID. En este trabajo se aborda la capacidad de los EA's para la optimización dinámica.

En Menhas et al. (2012b) se presenta una comparación de diversos esquemas de algoritmos del tipo PSO, y se compara la calidad de soluciones que ofrecen para el ajuste de controladores PID en procesos MIMO. Se emplean dos funciones de

coste con objetivos de diseño en el dominio del tiempo (IAE e ITAE), así como diferentes estructuras de control. Este trabajo es interesante, pues se aprecia un estudio de convergencia de diferentes versiones del PSO, para después proponer un algoritmo nuevo. En Menhas et al. (2012a) dicha propuesta es presentada, con mecanismos de cooperación y co-evolución, para mejorar las propiedades de convergencia del algoritmo. El trabajo se orienta al ajuste de dos controladores PID para un sistema de pulverizado. Se evalúan diferentes funciones de coste con vectores de peso para agregar objetivos de diseño en el dominio del tiempo: IAE/Sobreimpulso, ITAE/Sobreimpulso. La pertinencia de las soluciones obtenidas se evalúa con un conjunto adicional de índices de rendimiento.

En Elgammal y Sharaf (2012) un controlador PID de dos grados de libertad es ajustado, para ser empleado en el regulador del sistema de recarga de una batería híbrida fotovoltaica. Un total de 5 objetivos de diseño son definidos, acorde a los requerimientos especificados en el sistema de recarga: el tiempo de carga, la distorsión armónica, eficiencia de la batería, el error global de todos los reguladores y la energía proveniente de la batería híbrida. Los 5 objetivos de diseño son agregados usando un vector lineal de pesos y la función de coste es optimizada empleando un algoritmo basado en PSO. El esquema es evaluado en un prototipo real con buenos resultados.

La capacidad de los EA's para optimizar funciones de coste complejas con recursos computacionales limitados es explotado en Romero-Pérez et al. (2012); se emplea un DE para implementar un algoritmo de auto-ajuste de controladores PID con filtro de la derivada. El mismo se ejecuta en línea, ajustando el controlador con base a un modelo de primer orden con retardo calculado por el mismo algoritmo de auto-ajuste.

Un área de interés donde se han empleado EA's junto con PID convencionales es en el de optimización multi-disciplinaria. Entenderemos por diseño multi-disciplinar aquél que considera, en el mismo problema de optimización, aspectos de control, mecánica, geometría y otros aspectos del dispositivo en cuestión. El primer trabajo en abordar esto parece ser Avigad et al. (2003), donde se plantea de forma integral el diseño físico con los parámetros de un controlador PID. Se emplean como ejemplos tipo un manipulador de un eslabón y dos eslabones.

En Behbahani y de Silva (2008), se presenta una metodología para el diseño de sistemas mecatrónicos basada en lo que se denomina un *índice multi-criterio del coeficiente de diseño mecatrónico* (MDQ). Dado que el espacio de decisión es muy amplio y complejo, al considerar diferentes disciplinas en el diseño de sistemas mecatrónicos, se emplean algoritmos genéticos para abordar el problema de diseño. Un caso de estudio es presentado, donde se desarrolla el diseño integral de una máquina industrial para el corte de pescado, considerando variables de diseño mecánico, instrumentación y parámetros de un PID.

En Jiachuan et al. (2005), el esquema de diseño mecatrónico se aborda empleando *bond graphs* y GP; así mismo, se realiza una representación del controladores P, PI, PID, PD con *bond graphs*. Posteriormente, se procede a efectuar el diseño integral para aplicaciones mecatrónicas. Se presentan dos casos de estudio, el diseño de un sistema de suspensión de cuarto de vehículo y de un sistema microelectromecánico. Se mostró como es po-

<sup>12</sup>Código disponible para diferentes plataformas en [http://www.lri.fr/~hansen/cmaes\\_inmatlab.html](http://www.lri.fr/~hansen/cmaes_inmatlab.html)

sible, empleando el mismo lenguaje de diseño y con una herramienta GP, el llevar a cabo un diseño integral del sistema físico y del controlador.

Finalmente en Lamanna et al. (2009) se considera el diseño simultáneo de proceso y control de una torre sulfatadora de jugo de caña de azúcar. Como objetivo de optimización de empleo una función de coste que agrupó costos de construcción, costos de operación y costos de operación relacionado con la acción de control. El espacio de variables de decisión consideró dimensionamiento de la torre sulfatadora y los parámetros de un controlador PI. Se empleó un algoritmo genético para resolver dicho problema de diseño, mientras que para aumentar la pertinencia de soluciones se incorporaron restricciones físicas, de controlabilidad y balance de masa mediante un mecanismo de penalización.

#### 4.1.2. Optimización Multi-Índice

En el caso de diseño multi-objetivo en el ajuste de los parámetros del controlador, existen trabajos que abordan de formas diferentes el proceso MOOD. En Herreros et al. (2002) se propone un problema de optimización de 4 objetivos para un controlador PID de dos grados de libertad con filtro de la derivada. Los objetivos empleados son definidos en el dominio de la frecuencia, usando normas de Lebesgue para evaluar el rechazo a perturbación, seguimiento de la referencia, el valor máximo de la función de sensibilidad y la atenuación del ruido en la medida. Así mismo, se incluyen restricciones para evitar los efectos de la saturación. En este caso, se muestra la flexibilidad que gana el diseñador, al poder estudiar el intercambio de prestaciones entre soluciones, para elegir la más preferente.

En Tan et al. (2005) un controlador PI para una planta de pH es ajustado, atendiendo a 6 objetivos. Se emplea un algoritmo basado en GA's, y la visualización se lleva a cabo por medio de *parallel coordinates*. En este caso, se muestra como la optimización no está limitada a 2-3 objetivos simultáneos; así mismo, el empleo de herramientas gráficas permiten el emplear un mayor número de dimensiones de diseño y elegir la solución más adecuada según las preferencias.

En Tavakoli et al. (2007) se emplea una metodología integral MOOD para el ajuste de controladores PI con ponderación de la referencia. Se emplean como objetivos el máximo valor de la función de sensibilidad, la integral del error y la variación total de la acción de control. La metodología parte de un modelo FOPDT para desarrollar la optimización y la etapa de selección. Se trata de un trabajo donde se desarrollan los primeros avances en el desarrollo de reglas de ajuste basadas en la optimización multi-objetivo y el análisis del frente de Pareto.

En Reynoso-Meza et al. (2009), se ajusta un controlador PID de dos grados de libertad con filtro de la derivada. El proceso bajo consideración fue un modelo no-lineal caja negra, propuesto en el concurso de ingeniería de control de CEA en el año 2009. Dicho modelo no-lineal presentaba diferentes comportamientos dinámicos de acuerdo a la zona de operación. Un total de 15 objetivos de diseño fueron optimizados simultáneamente, incluyendo indicadores clásicos de rendimiento y de robustez. Se observa que los métodos de visualización pueden ser determinantes para llevar a cabo el análisis cuando muchos objetivos

son considerados.

En Xue et al. (2010) un conjunto de 4 controladores PI es propuesto para el problema del gasificador ALSTOM (Dixon y Pike (2006)). Dicho gasificador es un benchmark MIMO muy popular<sup>13</sup>, donde una gran variedad de estructuras de control han sido desarrolladas y evaluadas. El MOP consistió en 6 objetivos, cada uno relacionado con la integral del valor absoluto para diferentes escenarios de perturbación y carga. Se presenta un estudio detallado de las soluciones (visualización por medio de *parallel coordinates*); con objeto de elegir la solución más adecuada según las preferencias, un nuevo índice fue incluido en la selección multi-criterio. Este es un claro ejemplo de la flexibilidad de la optimización multi-objetivo: el análisis multi-criterio puede involucrar nuevos indicadores no usados durante la fase de optimización.

En Reynoso-Meza et al. (2011a) el problema de optimización no-convexa para controladores PI con ponderación de la referencia definido por Åström et al. (1998) es abordado como un MOP. El propósito general fue analizar el intercambio de prestaciones entre la ganancia integral, la función de sensibilidad y la función complementaria de sensibilidad para cambio en referencia. En este trabajo por medio del MOOD se busca tener otro nivel de entendimiento en el intercambio de prestaciones del controlador. Dicho trabajo fue extendido posteriormente al caso de controladores PI para procesos MIMO Reynoso-Meza et al. (2012c), empleando como indicador de robustez general el *biggest log modulus* (BLT) definido en Luyben (1986).

En Sidhartha y Panda (2011), un controlador PID es ajustado para el control de un sistema de transmisión AC. El objetivo general fue encontrar un controlador capaz de rechazar perturbaciones con el mínimo esfuerzo de control. Un sistema de inferencia borroso es utilizado para facilitar la selección de lo que es considerado la mejor solución de compromiso.

En Zhao et al. (2011) un controlador PID multivariable es ajustado para procesos MIMO. Dos objetivos son definidos, usando una AOF para la estabilidad robusta, y rechazo a perturbación y otro para la integral del error cuadrado. El análisis en el intercambio de prestaciones se lleva a cabo por medio de diagramas de dispersión (suficiente al emplear solo dos objetivos). Se selecciona el controlador que minimiza la norma 1 en el frente aproximado. Mientras tanto, en Ayala y dos Santos Coelho (2012) se ajusta un controlador PID para un manipulador de dos grados de libertad, empleando dos objetivos: uno para la posición y otro para el esfuerzo de control. Definen como la mejor solución de compromiso la media armónica de los objetivos de diseño en el frente calculado.

Una característica del enfoque MOOD, es el de poder comparar la totalidad de prestaciones que pueden ofrecer dos diferentes estructuras de control, y elegir la más conveniente para el diseñador. Esto se puede realizar durante la fase de selección multi-criterio y puede ser útil cuando se evalúa por ejemplo la capacidad de un controlador complejo con uno simple.

En ésta línea, en Algoul et al. (2011), ajustan y evalúan un controlador PID y un I-PD para el control óptimo de quimiote-

<sup>13</sup>Los modelos Simulink están disponibles en <http://www-staff.1boro.ac.uk/~e1rd2/>

rapias en el tratamiento de cáncer. El controlador es empleado para suministrar dosis del medicamento por un período de tiempo determinado. Tres objetivos fueron definidos: maximizar la eliminación de células cancerosas, minimizar la toxicidad y la concentración de medicamento. El MOOD es empleado para evaluar y observar las diferencias en el tratamiento. La visualización de los frentes de Pareto fue fundamental para poder analizar y dar significado físico al comportamiento e intercambio de prestaciones que cada controlador tenía en un paciente virtual modelo. El anterior es un ejemplo para mostrar las capacidades del MOOD para tener una idea general sobre el comportamiento en el intercambio de prestaciones de una solución de acuerdo a un caso nominal, para seleccionar la más adecuada para un paciente específico.

En Reynoso-Meza et al. (2012b), se compara un controlador PID y un controlador por realimentación del estado en un proceso no-lineal MIMO (simulación y experimental). La comparación de ambos controladores se lleva a cabo en un espacio de cuatro objetivos de diseño, observando debilidades y fortalezas de cada uno de ellos en lo referente a robustez y rendimiento.

Con respecto al diseño multi-disciplinar, en Rao y Tiwari (2009), un rodamiento magnético de doble acción y su controlador son diseñados de manera conjunta con un marco MOOD unificado. Un MOEA basado en GA es utilizado para optimizar un MOP de 5 objetivos de diseño, 13 restricciones, 6 variables de decisión geométrica y 2 parámetros de decisión de un controlador PD. Una visualización por medio de diagramas de dispersión es utilizado, eligiendo la solución más cercana a la solución utópica. En este caso, este trabajo podría ser un ejemplo de una integración total de diseño mecánico-y-de-control.

#### 4.1.3. Observaciones en el empleo de EA's para el ajuste de controladores PID convencionales

De lo expuesto anteriormente y la tabla 2 se pueden hacer las siguientes observaciones:

- Para SOP, AOF los algoritmos menos populares son los del tipo DE; para MOOD, los más populares son del tipo GA (particularmente NSGA-II).
- Son pocos los trabajos que explotan la totalidad de grados de libertad del PID convencional: ganancias (proporcional, integral y derivada), ponderaciones (de la referencia y de la derivada) y su filtro (en la medida y de la derivada).
- Es más común encontrar la incorporación de restricciones en el planteamiento del problema de optimización cuando se llevan a cabo un MOOD.
- Así mismo, es interesante hacer notar que aún se aprecian trabajos donde la robustez del controlador no es una restricción a considerar, sino que en muchos casos solo se verifica.
- No parece explotarse el uso de EA's para optimización a mayor escala y optimizar un conjunto de controladores

PID para procesos MIMO. Los casos presentados se limitan generalmente a ejemplos de procesos 2 entradas, 2 salidas.

- El uso de preferencias en optimización multi-índice se limita al empleo de restricciones y/o cotas en los frentes de Pareto.
- En lo general, no se destaca un análisis profundo en el intercambio de prestaciones cuando se aproxima un frente de Pareto. La mayoría de los casos, se aplica alguna regla específica para seleccionar la alternativa de diseño.
- Los EA's se emplean en optimización multi-disciplinar, multi-objetivo, optimización robusta (en lo referente a incertidumbres en el modelo), pero no se encuentran trabajo en optimización para confiabilidad.

#### 4.2. Ajuste de controladores borrosos del tipo PID

En el caso de los controladores PID borrosos, los EA's no aparecen tan ampliamente usados como en el caso de los PID convencionales. Esto pese a que los EA's han sido ampliamente usados en sistemas borrosos de forma general, como describen los trabajos de Cordón (2011) y de Fazzolar et al. (2013).

En Pan et al. (2011), un controlador es ajustado en términos de rendimiento en el dominio del tiempo, en condiciones de operación en red. Lo anterior permitió optimizar el ajuste del controlador considerando retardos aleatorios en la red. En este caso, se muestra la flexibilidad de los EA's para calcular un ajuste óptimo (dada una función de coste) considerando condiciones de operación que representaban mejor la realidad. Este es un ejemplo interesante de optimización para la confiabilidad, pues se optimiza un escenario donde el controlador sea capaz de mantener sus prestaciones ante incertidumbres (retardos aleatorios).

En Oh et al. (2012), una estructura en cascada de controlador Fuzzy PD es ajustado comparando un DE y un GA. se utiliza como función de coste el indicador IAE. El ajuste se realiza para un péndulo invertido rotacional y un sistema bola-viga: el primero sólo en simulación, el segundo en simulación y con el proceso real. Se incorpora ruido en la medida para evaluar la robustez del ajuste en simulación y la estructura de control.

En Bingul y Karahan (2011) se comparan 3 diferentes funciones de coste para el ajuste de los parámetro del controlador PID-borroso para un manipulador. Se emplea un algoritmo del tipo PSO, con funciones de coste basadas en el error cuadrático, el error absoluto y el absoluto con esfuerzo de control. Se identificó la importancia de una adecuada selección de la función de coste, ya que no es garantía que cumpla con las preferencias del diseñador. Por ello se analizaron las tres (error cuadrático, error absoluto, y error absoluto con esfuerzo de control) y se observaron otros muchos indicadores de rendimiento y robustez para determinar hacia qué tipo de soluciones convergen.

En Fazendeiro et al. (2007) se emplea un proceso de diseño MOOD para el ajuste de un controlador PD borroso para el suministro de anestesia en pacientes. El uso del esquema es justificado por la necesidad de tener una estructura interpretable

Tabla 3: Resumen de aplicaciones en ajuste de controladores borrosos del tipo PID. En algunos casos, el número de objetivos puede incluir restricciones. Un asterisco antecediendo a la estructura de control específica una optimización mono-índice. La codificación del tipo de estructura de control se muestra en la tabla 1.

Estructura	Aplicación / Proceso	Referencia	Problema de Diseño			Detalles sobre:			
			$J(x)$	$x$	$g(x)-h(x)$	Optimizador	Restricciones	Preferencias	Características Deseables
*PID <sub>Fuzzy</sub>	SISO	Pan et al. (2011)	2	3,5	(0,0)	GA,PSO	-	-	Sección 3.3.7 (Confiabilidad)
*PID <sub>Fuzzy</sub>	Péndulo invertido	Oh et al. (2012)	1	6	(0,0)	GA,DE	-	Comparando otros índices	
*PID <sub>Fuzzy</sub>	Robótica	Bingul y Karahan (2011)	1	60	(0,0)	PSO	-	Comparando Funciones de coste y otros índices	
PID <sub>Fuzzy</sub>	Biomédico	Fazendeiro et al. (2007)	2	40	(0,0)	SPEA	-	Analizando el Frente y comparando con otros índices.	

para el controlador, de manera que se pueda comprender la forma en que toma sus decisiones. El empleo de la técnica MOOD es justificada al servir de ayuda para apreciar el intercambio de prestaciones entre los dos objetivos de diseño: calidad de la acción de control e interpretabilidad del controlador. En la etapa de selección se consideran varias soluciones y se analizan sus prestaciones; se incluyen otros índices como sobreimpulso y tiempo de estabilización para valorar la pertinencia de las soluciones en un conjunto amplio de pacientes simulados. El trabajo es de interés por centrarse en un proceso de diseño integral y analizar la sensibilidad del mismo para aportar soluciones de interés. Así mismo, por centrarse en una optimización basada en la confiabilidad del sistema, al analizar varios escenarios para el modelo a controlar (pacientes).

De la tabla 3 se aprecian las siguientes observaciones:

- Es poco lo que ha filtrado el proceso de diseño MOOD en el ajuste de controladores PID borrosos.
- En todo caso, es de hacer notar la carencia en la incorporación de restricciones en el ajuste de los controladores. Lo anterior puede deberse a que dada la magnitud del espacio de decisión, la incorporación de restricciones podría complicar aún más el problema de optimización.
- Ha filtrado poco aún los planteamientos de optimización dinámica, robusta, multi-modal y multi-disciplinar.

#### 4.3. Ajuste de controladores fraccionarios del tipo PID

En el caso de los controladores PID fraccionarios (Podlubny (1999)), se encuentra que poco ha filtrado el empleo de EA's. En Lee y Chang (2010) se ajusta un controlador de este tipo, empleando un índice de error cuadrático. Como se señala en el trabajo, el ajuste de controladores PID fraccionarios suele ser complicado, por la teoría analítica detrás de ellos. Por ello se diseña un algoritmo basado en GA para determinar los parámetros óptimos a partir de la función de coste.

En la misma línea, en Biswas et al. (2009) el controlador se ajusta con un DE, empleando en la función de coste un AOF que agrupa la parte real, imaginaria y la fase del sistema en lazo cerrado. La función de coste es probada para un conjunto de plantas. El esquema de ajuste aquí presentado es interesante, pues combina técnicas clásicas (colocación de polos) y EA's. En

un primer caso, se traducen las especificaciones deseadas (aumentando la pertinencia de soluciones) en requerimientos en la colocación de polos del sistema; posteriormente, el EA optimiza las ganancias necesarias para cumplir con la estabilidad del sistema.

En Zamani et al. (2009) se usa un PSO, y se ha diseñado una función de coste que agrupa 6 objetivos de diseño en el dominio del tiempo y dos en el dominio de la frecuencia; adicionalmente, una restricción para la estabilidad del bucle cerrado por medio de penalización es incorporada. Se logra ajustar el controlador con un razonable margen de robustez. Se observa que no es trivial el incorporar más objetivos de diseño y agruparlos.

En Hajiloo et al. (2012) se ajusta un PID fraccionario usando el proceso MOOD. El planteamiento es de interés, debido a que aborda un enfoque de optimización robusta para la confiabilidad. Los objetivos de diseño incluyen probabilidad de fallo en el sistema ante simulaciones estocásticas (Monte Carlo) de diferentes modelos. Se ajustan las 5 variables de decisión del controlador minimizando simultáneamente la acción de control y las probabilidades de fallo del criterio ITSE, en la frecuencia de respuesta, en la robustez y en la inestabilidad. Se considera una cota en los objetivos de robustez en el sistema, para garantizar soluciones pertinentes para el diseñador. El análisis de las prestaciones se lleva a cabo mediante el LD y se compara con los de un PID convencional.

En la tabla 4 se muestra un resumen de las aplicaciones. Es poco lo que aún ha filtrado el empleo de los EA's en este tipo de controladores. Se hace notar que en lo referente al ajuste de controladores fraccionarios con EA's no existe un sesgo hacia el uso de una técnica u otra. Así mismo, es un área de oportunidad, para incorporar mecanismo para el tratamiento de restricciones, preferencias para diferentes planteamientos del problema de optimización.

## 5. Conclusiones y reflexiones

En el presente documento se ha hecho una revisión de la forma en que los EA's han apoyado al diseñador para el ajuste de controladores del tipo PID. Aunque se cuentan con reglas de ajuste sencillas (principalmente en el caso de controladores PID convencionales) en algunos casos el ajuste requiere de técnicas más complicadas, donde un problema de optimización puede

Tabla 4: Resumen de aplicaciones en ajuste de controladores fraccionarios del tipo PID. En algunos casos, el número de objetivos puede incluir restricciones. Un asterisco antecedido a la estructura de control especifica una optimización mono-índice. La codificación del tipo de estructura de control se muestra en la tabla 1.

Estruc-tura	Aplicación / Proceso	Referencia	Problema de Diseño			Detalles sobre:			
			$J(x)$	$x$	$g(x)-h(x)$	Optimi-zador	Restric-ciones	Preferencias	Características Deseables
*PI <sup>λ</sup> D <sup>μ</sup>	SISO	Lee y Chang (2010)	1	5	(0,0)	GA	-		
*PI <sup>λ</sup> D <sup>μ</sup>	SISO	Biswas et al. (2009)	3	5	(0,1)	DE	-	Por método de las raíces	
*PI <sup>λ</sup> D <sup>μ</sup>	Eléctrico	Zamani et al. (2009)	8	5	(0,0)	PSO	-		
*PI <sup>λ</sup> D <sup>μ</sup>	SISO	Hajiloo et al. (2012)	5	5	(0,3)	GA de Matlab	Filosofía MOP	Analizando el frente de Pareto.	Sección 3.3.7 (confiabilidad)

ser la opción conveniente. Cuando este problema sea no-lineal y no-convexo, los EA's son una opción válida.

Se han presentado dos formas en las que puede ser abordado el diseño cuando se emplea un proceso de optimización: optimizando una función de coste mono-índice u optimizando una función de coste multi-índice. La primera de ellas, busca una única solución en el espacio de decisión, mientras que la segunda calcula un conjunto de soluciones potenciales; en el último caso, para que el diseñador pueda compararlas y elegir la más adecuada según sus preferencias. La fortaleza de la primera es que, una vez que ha definido el MOP, puede ser empleada sin el diseñador, mientras que la segunda requiere de la intervención final del mismo. Ésta última en contraparte permite analizar el intercambio en prestaciones y lograr otro nivel de involucramiento en el MOP. Se podría decir que la estrategia MOOD se reservará para un subconjunto de estos controladores que requieren de un ajuste mediante optimización aún más complejo, donde el diseñador tiene interés en analizar el intercambio de prestaciones.

Se ha mostrado como bajo ambos esquemas, los EA's han servido de soporte para mejorar el rendimiento de diferentes tipos de controladores PID. El empleo de un EA u otro está ligado al tipo de características (algunas deseables, otras exigidas) que puede ofrecer para resolver el problema en cuestión. La capacidad de convergencia que ofrecen los EA's, dadas sus capacidades de explotación y exploración, son las que en un primer momento atrajeron la atención de los diseñadores. Sin embargo, otras características son actualmente demandadas por el diseñador marcando una pauta para el desarrollo de nuevos EA's. Entre dichas características destacan: diversidad de soluciones, manejo de preferencias, manejo de restricciones, escalabilidad, manejo de incertidumbres, manejo de robustez.

En el caso de los PID convencionales, se observa que los EA's han participado activamente en el ajuste de sus parámetros atendiendo a múltiples requerimientos. Se ha observado que son pocos los trabajos que ajustan todas las posibles variables de decisión del PID (ganancias, ponderaciones y filtros), por lo que algunos trabajos podrían orientarse al respecto.

Dentro de las cinco áreas emergentes en optimización (Roy et al. (2008)), se encuentran trabajos orientados al diseño multi-disciplinar, multi-objetivo, sin embargo, aún son pocos los basados en confiabilidad, incertidumbres y robustez (en lo referente a sensibilidad de la solución). Ésta última será interesante considerarla dentro de un marco multi-disciplinar, donde sea

considerado el ajuste del PID junto con otras variables de decisión, con mayor significado físico.

En el caso de MOOD, se ha explorado poco el empleo de preferencias adicionales; es importante considerarlo al ver que muchas aplicaciones caen en el ámbito de *many objectives optimization*, y es muy fácil contar con un conjunto enorme de soluciones, lo cual también implica un problema para el diseñador y su proceso de selección.

En el caso de los PID borrosos, salta a la vista el tema de la escalabilidad, ya que un PID borroso contará con más variables de decisión para ajustar. Se usa poco el empleo de restricciones en estos controladores, lo que es importante para mejorar la pertinencia de soluciones. Es de esperar investigación en esta área orientada a la definición de problemas de optimización que incorporen las restricciones, y a algoritmos que sean capaces de manejar decenas de variables de decisión con restricciones.

Respecto a los controladores PID fraccionarios, se observa como son un área que empieza a desarrollarse y donde los EA's empiezan a ofrecer soluciones. En este caso, ha sido interesante encontrar trabajos que abordan una comparación multi-índice de controladores PID fraccionarios con PID convencionales. Es decir, comparar las fortalezas y debilidades de ambos conceptos de diseño para resolver un problema de control. Se podría decir que ahora mismo esta área está centrada en la convergencia de soluciones, pero poco a poco se irá orientando hacia otras características deseables para el algoritmo.

En general, se aprecia no ha sido explotada la capacidad de los EA's para optimización a gran escala. Probar la viabilidad de los EA's para ajustar simultáneamente un conjunto de PID para procesos MIMO puede presentar una línea de trabajo a seguir. Así mismo, en muchos casos aún se pueden contar implementaciones donde la robustez no es asegurada, sino verificada con pocos ejemplos.

Finalmente, se puede concluir que existen varias herramientas disponibles, incorporando muchas de las características deseadas en los EA's (ver Apéndice A), que requieren poco ajuste de sus parámetros internos. Estas pueden servir de excelente punto de partida para el diseñador interesado solamente en ser un usuario de las mismas. Si estas herramientas no fuesen suficientes para un problema en particular (dadas las preferencias, deseos y tipo de soluciones que desea el diseñador) se podrá siempre mejorar el algoritmo o a trabajar en alguno de ellos. Es entonces cuando se promueve el estudio e investigación en EA's. En particular, el área de confiabilidad e incerti-

dumbres puede ser de interés, para garantizar el desempeño de los controladores del tipo PID.

## Apéndice A. Ensayo tipo

Se describe aquí un breve ensayo para comparar diferentes algoritmos evolutivos y el tipo de aportación que pueden ofrecer para el ajuste de controladores PID convencionales.

Se evaluarán los algoritmos que obtuvieron primer y segundo puesto en la competición organizada por el Congreso en Computación Evolutiva de la IEEE en 2011. La competición gira en torno a la optimización de aplicaciones reales, incluyendo problemas del ámbito químico, eléctrico, energético y aeroespacial como describen Das y Suganthan (2011). Los algoritmos a evaluar son el basado en GA (GA-MPC) de Elsayed et al. (2011) y en DE (DE-TCR) de Reynoso-Meza et al. (2011b); ningún algoritmo basado en PSO participó en la competición. Dichos algoritmos incluyen ajustes de sus parámetros internos por defecto, por lo que no han sido ajustados y han sido utilizados tal y como han sido descargados<sup>14</sup>.

### Apéndice A.1. Definición del problema

Se busca ajustar un controlador PI para el control y suministro de insulina. Se emplea el modelo de primer orden con retardo suministrado en Kamath et al. (2009):

$$G(s) = \frac{-7,31}{73,14s + 1} e^{-20,32s} \quad (\text{A.1})$$

Se desea generar un PI que sea capaz de rechazar una perturbación de insulina debido a una ingesta con un contenido de 80g de glucosa. Dicha perturbación en el sistema se puede modelar con un segundo modelo de primer orden, acorde con Kaveh y Shtessel (2008). Se pretende diseñar un controlador PID que sea capaz de mantener el nivel de glucosa en la sangre dentro de los límites de seguridad de 120 mg/dL y 70 mg/dL. Así mismo, dicha perturbación debe ser rechazada en un tiempo menor a 200 minutos (al  $\pm 2\%$ ). La concentración inicial de glucosa en la sangre es de 80 mg/dL. Por supuesto, una mínima robustez en el sistema es deseada y se define la misma con un valor de la función de sensibilidad  $M_s \leq 1,6$ .

### Apéndice A.2. Planteamiento

Se puede plantear el ajuste del controlador por medio de reglas de ajuste conocidas. Sin embargo, el ajuste de dichas reglas de sintonía debe probarse después en simulación ya que pueden no ofrecer los resultados deseados por el diseñador. Esto se debe no a que dichas reglas de ajuste sean inadecuadas, sino a que han sido diseñadas en un dominio de objetivos diferentes al que desea el diseñador. Es decir, el diseñador está interesado en objetivos de diseño en el dominio del tiempo, principalmente.

Se puede plantear un problema de optimización para el ajuste del controlador como el que describe Åström et al. (1998),

donde se maximiza la ganancia de la acción integral sujeta a cierta valor máximo de la función de sensibilidad. Sin embargo, una vez más lo anterior no garantiza que se cumplirán los objetivos de diseño, pues el problema de optimización está planteado en un dominio diferente (el de la frecuencia).

Por tanto, se plantea el problema de optimización siguiente:

$$\min_{x \in \mathbb{R}^2} J(K_c, T_i) = T_{estabilizacion} \quad (\text{A.2})$$

sujeto a:

$$T_{estabilizacion} \leq 200min \quad (\text{A.3})$$

$$70 \frac{mg}{dL} < C(t) < 120 \frac{mg}{dL} \quad (\text{A.4})$$

$$M_s \leq 1,6 \quad (\text{A.5})$$

$$\underline{x}_i \leq x_i \leq \bar{x}_i, i = [1, \dots, n] \quad (\text{A.6})$$

Donde  $K_c$  es la ganancia proporcional y  $T_i$  la constante de tiempo integral de un controlador PI convencional;  $C(t)$  la concentración de glucosa en la sangre y  $M_s$  el valor máximo de la función de sensibilidad. Los objetivos y restricciones serán calculados a partir de una simulación, donde el sistema de la ecuación A.1 es sometido a una perturbación de 80 g de Glucosa en el tiempo  $t=0$ . Se evalúan 4 ajustes:

1. Un ajuste empleando la función *fmincon* de Matlab © planteando el problema de optimización de Åström et al. (1998). Como aproximación inicial se usará un controlador ajustado por el método de Skogestad (2003).
2. Un ajuste empleando la función *fmincon* de Matlab © planteando un problema de optimización como el de la ecuación A.2. Como aproximación inicial se usará un controlador ajustado por el método de Skogestad (2003).
3. Un ajuste empleando el algoritmo GA-MPC de Elsayed et al. (2011) planteando un problema de optimización como el de la ecuación A.2.
4. Un ajuste con el algoritmo DE-TCR expuesto en Reynoso-Meza et al. (2011b) planteando un problema de optimización como el de la ecuación A.2.

En los casos de los algoritmos evolutivos, se deberá hacer una valoración estadística, debido a sus propiedades estocásticas. Por tanto, se llevarán a cabo 51 ejecuciones independientes de cada algoritmo seleccionándose la mediana para la comparación. En la tabla A.5 se exponen los resultados estadísticos de las 51 ejecuciones. Se emplea un mecanismo simple de penalización basado en las reglas de factibilidad para ambos casos y un máximo de 2000 evaluaciones de la función objetivo son empleadas.

En la figura A.2 se muestran los resultados. En el caso 1, se observa que no se cumplen los requerimientos de diseño. Una vez más, esto no quiere decir que el método de ajuste es deficiente; solo muestra que ha sido planteado en un dominio de objetivos diferente del que desea el diseñador. En el caso 2, se observa que el optimizador *fmincon* no encuentra una solución, a partir de la aproximación inicial, que cumpla con los requerimientos. En los casos de los algoritmos evolutivos, se observa como ambos encuentran soluciones satisfactorias. Entre ellos,

<sup>14</sup>El GA se encuentra disponible en: <http://www.ntu.edu.sg/home/epnsugan/>; el DE en: <http://www.mathworks.com/matlabcentral/fileexchange/39217>.

Tabla A.5: Desempeño de los algoritmos evolutivos en el ensayo. El asterisco (\*) indica que se ha calculado a partir del subconjunto de soluciones factibles.

Estadístico	GA-MPC	DE-TCR
Tasa Éxito	51/51	29/51
Mínimo	164 min	167 mins
Mediana	168 min	176 min
Máximo	174 min	No Factible
Media	168.25 min	170.41* min
Desviación	2.077	3.191*

se observa que el basado en GA-MPC es superior al DE-TCR, resultado consistente con el ranking de la competición en optimización del CEC-2011.

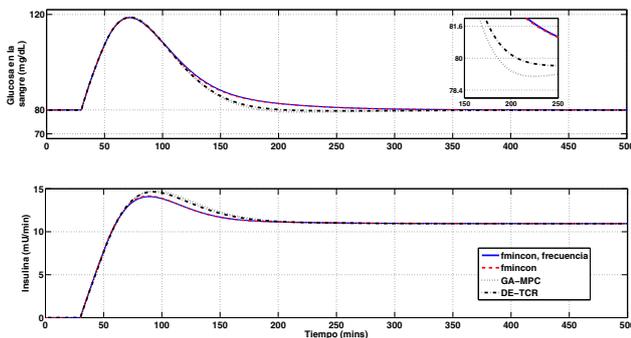


Figura A.2: Desempeño de los optimizadores utilizados.

Un ensayo más extenso deberá incluir una variedad mayor de algoritmos y procesos; así como la evaluación de diferentes métodos para el tratamiento de restricciones. Sin embargo, dada la amplitud de dicho estudio, se propone como trabajo futuro.

## English Summary

### Evolutionary Algorithms for PID controller tuning: Current Trends and Perspectives

#### Abstract

PID controllers are a reliable, robust, practical and easy to implement control solution for industrial processes. They provide the first control layer for a vast majority of industrial applications. Owing to this, several researches invest time and resources to improve their performance. The research lines in this field scope with new tuning methods, new types of structures and integral design methods. For tuning methods, improvements could be fulfilled stating an optimization problem, which could be non-linear, non-convex and highly constrained. In such instances, evolutionary algorithms have shown a good performance and have been used in various proposals related with PID controllers tuning. This work shows a review of these proposals and the benefits obtained in each case. Some trends and

possible future research lines are also identified.

#### Keywords:

PID controller, Conventional PID, Fuzzy PID, Fractional order PID, Evolutionary algorithms, Optimization.

## Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo del Ministerio de Economía y Competitividad (Gobierno de España) mediante los proyectos TIN2011 - 28082, ENE2011 - 25900; la Generalitat Valenciana mediante la iniciativa GV/2012/073 y la Universitat Politècnica de València a través de la becas FPI-2010/19 y la iniciativa de investigación PAID-06-11.

## Referencias

- Algoul, S., Alam, M., Hossain, M., Majumder, M., 2011. Multi-objective optimal chemotherapy control model for cancer treatment. *Medical and Biological Engineering and Computing* 49, 51–65, 10.1007/s11517-010-0678-y.
- Ang, K. H., Chong, G., Li, Y., July 2005. PID control system analysis, design, and technology. *Control Systems Technology, IEEE Transactions on* 13 (4), 559 – 576.
- Åström, K., Hägglund, T., 2001. The future of PID control. *Control Engineering Practice* 9 (11), 1163 – 1175.
- Åström, K. J., Hägglund, T., 2005. *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709.
- Åström, K., Panagopoulos, H., Hägglund, T., 1998. Design of PI controllers based on non-convex optimization. *Automatica* 34 (5), 585 – 601.
- Avigad, G., Moshaiov, A., Brauner, N., June 2003. Towards a general tool for mechatronic design. In: *Control Applications, 2003. CCA 2003. Proceedings of 2003 IEEE Conference on*. Vol. 2. pp. 1035 – 1040 vol.2.
- Ayala, H. V. H., dos Santos Coelho, L., 2012. Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Systems with Applications* 39 (10), 8968 – 8974.
- Behbahani, S., de Silva, C., Feb. 2008. System-based and concurrent design of a smart mechatronic system using the concept of mechatronic design quotient (MDQ). *Mechatronics, IEEE/ASME Transactions on* 13 (1), 14 – 21.
- Beyer, H.-G., Sendhoff, B., 2007. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering* 196 (33-34), 3190 – 3218.
- Bingul, Z., Karahan, O., 2011. A fuzzy logic controller tuned with PSO for 2 dof robot trajectory control. *Expert Systems with Applications* 38 (1), 1017 – 1031.
- Biswas, A., Das, S., Abraham, A., Dasgupta, S., 2009. Design of fractional-order  $PI^{\lambda}D^{\mu}$  controllers with an improved differential evolution. *Engineering Applications of Artificial Intelligence* 22 (2), 343 – 350.
- Blasco, X., Herrero, J., Sanchis, J., Martínez, M., 2008. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences* 178 (20), 3908 – 3924.
- Bonissone, P., Subbu, R., Lizzi, J., Aug. 2009. Multicriteria decision making (medm): a framework for research and applications. *Computational Intelligence Magazine, IEEE* 4 (3), 48 – 61.
- Caballero, J. A., Grossmann, I. E., 2011. Una revisión del estado del arte en optimización. *Revista Iberoamericana de Automática e Informática Industrial* 4 (1), 5 – 23.
- Coello, C., 2000. Handling preferences in evolutionary multiobjective optimization: a survey. In: *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. Vol. 1. pp. 30 – 37 vol.1.
- Coello, C. A. C., 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering* (191), 1245 – 1287.
- Coello Coello, C., Feb. 2006. Evolutionary multi-objective optimization: A historical view of the field. *Computational Intelligence Magazine, IEEE* 1 (1), 28 – 36.

- Coello, C., 2011. An introduction to multi-objective particle swarm optimizers. In: Gaspar-Cunha, A., Takahashi, R., Schaefer, G., Costa, L. (Eds.), *Soft Computing in Industrial Applications*. Vol. 96 of *Advances in Intelligent and Soft Computing*. Springer Berlin / Heidelberg, pp. 3–12, 10,1007/978-3-642-20505-7\_1.
- Cordón, O., 2011. A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning* 52 (6), 894–913.
- Corne, D. W., Knowles, J. D., 2007. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. GECCO '07. ACM, New York, NY, USA, pp. 773–780.
- Cruz, C., González, J., Pelta, D. A., 2011. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing* 15, 1427–1448.
- Das, S., Maity, S., Qu, B.-Y., Suganthan, P., 2011. Real-parameter evolutionary multimodal optimization - a survey of the state-of-the-art. *Swarm and Evolutionary Computation* 1 (2), 71–88.
- Das, S., Suganthan, P. N., 2010. Differential evolution: A survey of the state-of-the-art. *Evolutionary Computation*, IEEE Transactions on PP (99), 1–28.
- Das, S., Suganthan, P., 2011. Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems. Tech. rep., Jadavpur university and Nanyang Technological University.
- Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* 186 (2-4), 311–338.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6 (2), 124–141.
- Dixon, R., Pike, A., 2006. ALSTOM benchmark challenge II on gasifier control. *Control Theory and Applications, IEE Proceedings - 153* (3), 254–261.
- Eiben, A., Schippers, C., 1998. On evolutionary exploration and exploitation. *Fundamenta Informaticae* 35 (1), 35–50.
- Elgammal, A., Sharaf, A., 2012. Self-regulating particle swarm optimized controller for (photovoltaic-fuel cell) battery charging of hybrid electric vehicles. *Electrical Systems in Transportation, IET* 2 (2), 77–89.
- Elsayed, S., Sarker, R., Essam, D., 2011. GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems. In: *Evolutionary Computation (CEC), 2011 IEEE Congress on*. pp. 1034–1040.
- Fazendeiro, P., de Oliveira, J., Pedrycz, W., 2007. A multiobjective design of a patient and anaesthetist-friendly neuromuscular blockade controller. *Biomedical Engineering, IEEE Transactions on* 54 (9), 1667–1678.
- Fazzolar, M., Alcalá, R., Nojima, Y., Ishibuchi, H., Herrera, F., 2013. A review of the application of multi-objective evolutionary fuzzy systems: Current status and further directions. *IEEE Transactions on Fuzzy Systems* 21 (1), 45–65.
- Figueira, J., Greco, S., Ehrgott, M., 2005. *Multiple criteria decision analysis: State of the art surveys*. Springer international series.
- Fleming, P., Purshouse, R., 2002. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice* (10), 1223–1241.
- Fonseca, C., Fleming, P., 1998a. Multiobjective optimization and multiple constraint handling with evolutionary algorithms-I: A unified formulation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 28 (1), 26–37.
- Fonseca, C., Fleming, P., 1998b. Multiobjective optimization and multiple constraint handling with evolutionary algorithms-II: Application example. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 28 (1), 38–47.
- Gaing, Z.-L., 2004. A particle swarm optimization approach for optimum design of PID controller in AVR system. *Energy Conversion, IEEE Transactions on* 19 (2), 384–391.
- Goldberg, D., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.
- Hajiloo, A., Nariman-zadeh, N., Moeini, A., 2012. Pareto optimal robust design of fractional-order PID controllers for systems with probabilistic uncertainties. *Mechatronics* 22 (6), 788–801.
- Hansen, N., 2006. The CMA evolution strategy: a comparing review. In: Lozano, J., Larranaga, P., Inza, I., Bengoetxea, E. (Eds.), *Towards a new evolutionary computation*. Advances on estimation of distribution algorithms. Springer, pp. 75–102.
- Harik, G., Lobo, F., Goldberg, D., 1999. The compact genetic algorithm. *Evolutionary Computation, IEEE Transactions on* 3 (4), 287–297.
- Herreros, A., Baeyens, E., Perán, J. R., 2002. Design of PID-type controllers using multiobjective genetic algorithms. *ISA Transactions* 41 (4), 457–472.
- Holland, J. H., 1975. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U. Michigan Press, Oxford, England.
- Huang, L., Wang, N., Zhao, J.-H., 2008. Multiobjective optimization for controller design. *Acta Automatica Sinica* 34 (4), 472–477.
- Hung, M.-H., Shu, L.-S., Ho, S.-J., Hwang, S.-F., Ho, S.-Y., 2008. A novel intelligent multiobjective simulated annealing algorithm for designing robust PID controllers. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 38 (2), 319–330.
- Inselberg, A., 1985. The plane with parallel coordinates. *The Visual Computer* 1, 69–91.
- Iruthayarajan, M. W., Baskar, S., 2009. Evolutionary algorithms based design of multivariable PID controller. *Expert Systems with applications* 3 (36), 9159–9167.
- Ishibuchi, H., Tsukamoto, N., Nojima, Y., 2008. Evolutionary many-objective optimization: A short review. In: *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*. IEEE Congress on. pp. 2419–2426.
- Jiachuan, W., Zhun, F., Terpeny, J., Goodman, E., 2005. Knowledge interaction with genetic programming in mechatronic systems design using bond graphs. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 35 (2), 172–182.
- Juang, J.-G., Huang, M.-T., Liu, W.-K., 2008. PID control using presearched genetic algorithms for a mimo system. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 38 (5), 716–727.
- Kamath, S., George, V. I., Vidyasagar, S., 2009. A comparative study of different types of controllers used for blood glucose regulation system. *The Canadian Journal of Chemical Engineering* 87 (5), 812–817.
- Kaveh, P., Shtessel, Y. B., 2008. Blood glucose regulation using higher-order sliding mode control. *International Journal of Robust and Nonlinear Control* 18 (4-5), 557–569.
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. Vol. 4. pp. 1942–1948 vol.4.
- Kollat, J. B., Reed, P., 2007. A framework for visually interactive decision-making and design using evolutionary multi-objective optimization (VI-DEO). *Environmental Modelling & Software* 22 (12), 1691–1704.
- Konak, A., Coit, D. W., Smith, A. E., 2006. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety* 91 (9), 992–1007, special Issue - Genetic Algorithms and Reliability.
- Koza, J., Keane, M., Streeter, M., 2003. What's AI done for me lately genetic programming's human-competitive results. *Intelligent Systems, IEEE* 18 (3), 25–31.
- Koza, J., Poli, R., 2005. Genetic programming. In: Burke, E. K., Kendall, G. (Eds.), *Search Methodologies*. Springer US, pp. 127–164, 10.1007/0-387-28356-0\_5.
- Lamanna, R., Vega, P., Revollar, S., Alvarez, H., 2009. Diseño simultáneo de proceso y control de una torre sulfatadora de jugo de caña de azúcar. *Revista Iberoamericana de Automática e Informática Industrial* 6 (3), 32–43.
- Lee, C.-H., Chang, F.-K., 2010. Fractional-order PID controller optimization via improved electromagnetism-like algorithm. *Expert Systems with Applications* 37 (12), 8871–8878.
- Li, Y., Ang, K. H., Chong, G., 2006. PID control system analysis and design. *Control Systems, IEEE* 26 (1), 32–41.
- Lin, C.-M., Li, M.-C., Ting, A.-B., Lin, M.-H., 2011. A robust self-learning PID control system design for nonlinear systems using a particle swarm optimization algorithm. *International Journal of Machine Learning and Cybernetics* 2, 225–234.
- Lotov, A., Miettinen, K., 2008. Visualizing the Pareto frontier. In: Branke, J., Deb, K., Miettinen, K., Slowinski, R. (Eds.), *Multiobjective Optimization*. Vol. 5252 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pp. 213–243.
- Lozano, M., Molina, D., Herrera, F., 2011. *Soft Computing: Special Issue on scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems*. Vol. 15. Springer-Verlag.

- Luyben, W. L., 1986. Simple method for tuning SISO controllers in multivariable systems. *Industrial and Engineering Chemistry Process Design* (25), 654 – 660.
- Mallipeddi, R., Suganthan, P., 2009. Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization. Tech. rep., Nanyang Technological University, Singapore.
- Marler, R., Arora, J., 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* (26), 369 – 395.
- Mattson, C. A., Messac, A., 2005. Pareto frontier based concept selection under uncertainty, with visualization. *Optimization and Engineering* 6, 85–115, 10.1023/B:OPTE.0000048538.35456.45.
- Menhas, M. I., Fei, M., Wang, L., Qian, L., 2012a. Real/binary co-operative and co-evolving swarms based multivariable PID controller design of ball mill pulverizing system. *Energy Conversion and Management* 54 (1), 67 – 80.
- Menhas, M. I., Wang, L., Fei, M., Pan, H., 2012b. Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design. *Expert Systems with Applications* 39 (4), 4390 – 4401.
- Messac, A., 1996. Physical programming: effective optimization for computational design. *AIAA Journal* 34 (1), 149–158.
- Mezura-Montes, E., Coello, C. A. C., December 2011. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation* 1 (4), 173 – 194.
- Mezura-Montes, E., Reyes-Sierra, M., Coello, C., 2008. Multi-objective optimization using differential evolution: A survey of the state-of-the-art. *Advances in Differential Evolution* (SCI 143), 173 – 196.
- Miettinen, K. M., 1998. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers.
- Mininno, E., Neri, F., Cupertino, F., Naso, D., feb. 2011. Compact differential evolution. *Evolutionary Computation, IEEE Transactions on* 15 (1), 32 –54.
- Oh, S.-K., Kim, W.-D., Pedrycz, W., 2012. Design of optimized cascade fuzzy controller based on differential evolution: Simulation studies and practical insights. *Engineering Applications of Artificial Intelligence* 25 (3), 520 – 532.
- Pan, I., Das, S., Gupta, A., 2011. Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay. *ISA Transactions* 50 (1), 28 – 36.
- Podlubny, I., jan. 1999. Fractional-order systems and  $PI^{\lambda}D^{\mu}$ -controllers. *Automatic Control, IEEE Transactions on* 44 (1), 208 –214.
- Rao, J. S., Tiwari, R., 2009. Design optimization of double-acting hybrid magnetic thrust bearings with control integration using multi-objective evolutionary algorithms. *Mechatronics* 19 (6), 945 – 964.
- Reynoso-Meza, G., Blasco, X., Sanchis, J., 2009. Diseño multiobjetivo de controladores PID para el benchmark de control 2008-2009. *Revista Iberoamericana de Automática e Informática Industrial* 6 (4), 93 – 103.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J., september 2011a. Handling control engineer preferences: Getting the most of PI controllers. In: *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*. pp. 1–8.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J., june 2011b. Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In: *Evolutionary Computation (CEC), 2011 IEEE Congress on*. pp. 1551 –1556.
- Reynoso-Meza, G., Blasco, X., Sanchis, J., March 2012a. Optimización evolutiva multi-objetivo y selección multi-criterio para la ingeniería de control. In: *X Simposio CEA de Ingeniería de Control*.
- Reynoso-Meza, G., García-Nieto, S., Sanchis, J., Blasco, X., 2012b. Controller tuning using multiobjective optimization algorithms: a global tuning framework. *IEEE Transactions on Control Systems Article in press*.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J. M., 2012c. Multiobjective evolutionary algorithms for multivariable PI controller tuning. *Expert Systems with Applications* 39, 7895 – 7907.
- Reynoso-Meza, G., Blasco, X., Sanchis, J., Herrero, J. M., 2013. Comparison of design concepts in multi-criteria decision-making using level diagrams. *Information Sciences* 221 (0), 124 – 141.
- Romero-Pérez, J. A., Arrieta, O., Padula, F., Reynoso-Meza, G., García-Nieto, S., Balaguer, P., 2012. Estudio comparativo de algoritmos de auto-ajuste de controladores PID. resultados del benchmark 2010-2011 del grupo de ingeniería de control de cea. *Revista Iberoamericana de Automática e Informática Industrial* 9 (2), 182 – 193.
- Roy, R., Hinduja, S., Teti, R., 2008. Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals - Manufacturing Technology* 57 (2), 697 – 715.
- Sanchis, J., Martínez, M. A., Blasco, X., Reynoso-Meza, G., 2010. Modelling preferences in multiobjective engineering design. *Engineering Applications of Artificial Intelligence* 23, 1255 – 1264.
- Santana-Quintero, L., Montaña, A., Coello, C., 2010. A review of techniques for handling expensive functions in evolutionary multi-objective optimization. In: Tenne, Y., Goh, C.-K. (Eds.), *Computational Intelligence in Expensive Optimization Problems. Vol. 2 of Adaptation Learning and Optimization*. Springer Berlin Heidelberg, pp. 29–59.
- Saridakis, K., Dentsoras, A., 2008. Soft computing in engineering design - a review. *Advanced Engineering Informatics* 22 (2), 202 – 221, network methods in engineering.
- Shi, L., Rasheed, K., 2010. A survey of fitness approximation methods applied in evolutionary algorithms. In: Tenne, Y., Goh, C.-K. (Eds.), *Computational Intelligence in Expensive Optimization Problems. Vol. 2 of Adaptation Learning and Optimization*. Springer Berlin Heidelberg, pp. 3–28.
- Sidhartha, Panda, 2011. Multi-objective PID controller tuning for a facts-based damping stabilizer using non-dominated sorting genetic algorithm-II. *International Journal of Electrical Power and Energy Systems* 33 (7), 1296 – 1308.
- Skogestad, S., 2003. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control* 13 (4), 291 – 309.
- Stewart, G., Samad, T., 2011. Cross-application perspectives: Application and market requirements. In: Samad, T., Annaswamy, A. (Eds.), *The Impact of Control Technology*. IEEE Control Systems Society, pp. 95 – 100.
- Storn, R., Price, K., 1997. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341 – 359.
- Tan, W., Liu, J., Fang, F., Chen, Y., 2004. Tuning of PID controllers for boiler-turbine units. *ISA Transactions* 43 (4), 571 – 583.
- Tan, W., Lu, F., Loh, A., Tan, K., 2005. Modeling and control of a pilot pH plant using genetic algorithm. *Engineering Applications of Artificial Intelligence* 18 (4), 485 – 494.
- Tavakoli, S., Griffin, I., Fleming, P. J., September 2007. Multi-objective optimization approach to the PI tuning problem. In: *Proceedings of the IEEE congress on evolutionary computation (CEC2007)*. pp. 3165 – 3171.
- Vilanova, R., Alfaro, V. M., 2011. Control pid robusto: una visión panorámica. *Revista Iberoamericana de Automática e Informática Industrial* 8 (3), 141 – 158.
- Xue, Y., Li, D., Gao, F., 2010. Multi-objective optimization and selection for the PI control of ALSTOM gasifier problem. *Control Engineering Practice* 18 (1), 67 – 76.
- Zamani, M., Karimi-Ghartemani, M., Sadati, N., Parniani, M., 2009. Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Engineering Practice* 17 (12), 1380 – 1387.
- Zhang, J., Zhuang, J., Du, H., Wang, S., 2009. Self-organizing genetic algorithm based tuning of PID controllers. *Information Sciences* 179 (7), 1007 – 1018.
- Zhang, Q., Li, H., december 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on* 11 (6), 712 – 731.
- Zhao, S.-Z., Iruthayarajan, M. W., Baskar, S., Suganthan, P., 2011. Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization. *Information Sciences* 181 (16), 3323 – 3335.
- Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., Zhang, Q., 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1 (1), 32 – 49.