

UNIVERSIDAD DE CANTABRIA



Departamento de Ingeniería Estructural y Mecánica
Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación

METODOLOGÍA DE SÍNTESIS ÓPTIMA DIMENSIONAL DE MECANISMOS MEDIANTE ALGORITMOS DE OPTIMIZACIÓN HÍBRIDOS

TESIS DOCTORAL

Autor: D. Ángel Sedano García

Directores de Tesis: Dr. D. Ramón Sancibrián Herrera
Dr. D. Fernando Viadero Rueda

Santander, Septiembre de 2013



Departamento de Ingeniería Estructural y Mecánica
Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación

METODOLOGÍA DE SÍNTESIS ÓPTIMA DIMENSIONAL DE MECANISMOS MEDIANTE ALGORITMOS DE OPTIMIZACIÓN HÍBRIDOS

TESIS DOCTORAL

Autor: D. Ángel Sedano García

Directores de Tesis: Dr. D. Ramón Sancibrián Herrera
Dr. D. Fernando Viadero Rueda

Santander, Septiembre de 2013

A Ángel e Irundina.
A Balbino y Delfina.
A Beatriz.
A Cristina.

Agradecimientos

Me gustaría agradecer al Área de Ingeniería Mecánica de esta escuela el apoyo y confianza que he recibido durante la realización de esta Tesis Doctoral. En especial a mis directores de Tesis, Ramón Sancibrián, cuya paciencia he puesto a prueba, por su dedicación, sugerencias y explicaciones, y a Fernando Viadero por permitirme y animarme a colaborar con este grupo. A mis compañeros ya doctores Pablo García y Ana de Juan por su paciencia y consejos.

No puedo olvidarme de agradecer a mi antiguo centro de trabajo, el departamento de mecatrónica de la Fundación Tekniker, y en especial a David Olasolo, Guillaume Collin, Steeve Gautier, Fernando Egaña, Luis Gerardo Uriarte, Javier Arzamendi y Mikel Zatarain por animarme y facilitarme el poder desarrollar esta Tesis Doctoral.

Agradecer también a la empresa en la que trabajo en la actualidad, M. Torres Ólvega Industrial, el permitirme disponer de recursos para la realización de esta Tesis Doctoral, y en especial a Carlos Arricibita por su comprensión y apoyo.

Gracias a mi familia y amigos por su comprensión y ánimos.

Gracias a todos.

Resumen

La síntesis óptima de mecanismos es un campo de investigación con múltiples métodos de solución propuestos para resolver un mismo problema. Estos métodos pueden variar en su planteamiento matemático, por ejemplo entre métodos basados en gradiente o métodos estocásticos. En el planteamiento formal de los objetivos, o incluso en la forma de evaluación del error. Este trabajo se centra en diferentes de estos aspectos, como implementación de un estimador de error, desarrollo de un método de gradiente, de un método estocástico y finalmente de la hibridación de métodos estocásticos y basados en gradiente.

Una de las principales características del método híbrido es el enfoque del mismo. Se plantean varias alternativas, como la utilización de un método basado en gradiente tras el método estocástico, y otras en las que se plantea la hibridación como un proceso de aprendizaje del individuo que se puede transmitir de una generación a otra de individuos (teoría Lamarkiana). Con este planteamiento se implementa el método basado en gradiente como parte del estimador de error, intentando representar el aprendizaje del individuo y su adaptación al medio a lo largo de su vida. Esta implementación se ha realizado para todos los individuos de cada generación, o únicamente para la élite de la misma.

La ventaja fundamental de la hibridación es la obtención de resultados precisos, cercanos a un mínimo de error global (método estocástico) y con precisión en los resultados con un coste computacional razonable (método de gradiente).

Los métodos propuestos se pueden utilizar con o sin restricciones y son aplicables a cualquier tipo de mecanismo, como se muestra en parte de los ejemplos expuestos a lo largo del trabajo.

Índice

Resumen	1
ÍNDICE	3
1 FUNDAMENTOS DEL TRABAJO Y ESTRUCTURA DE LA TESIS.....	8
1.1 FUNDAMENTO Y OBJETIVOS DE ESTA TESIS.	8
1.2 ESTRUCTURA DEL DOCUMENTO	9
2 ANTECEDENTES	12
2.1 SÍNTESIS CINEMÁTICA DE MECANISMOS.....	12
2.2 MÉTODOS EXACTOS.....	14
2.2.1 Métodos gráficos	14
2.2.2 Métodos analíticos.....	15
2.3 MÉTODOS DE SÍNTESIS ÓPTIMA.....	16
2.3.1 Estimadores de error.....	19
2.3.2 Métodos deterministas	22
2.3.3 Métodos probabilísticos	33
2.3.4 Métodos híbridos	48
2.3.5 Búsqueda Tabú	49
2.3.6 Colonias de Hormigas	49
2.3.7 Enjambre de Partículas	50
3 COMPARACIÓN DE CURVAS Y ESTIMACIÓN DEL ERROR.....	52
3.1 ESTIMADOR DE ERROR PROPUESTO: TRS.....	55
3.1.1 Trayectorias generada y deseada cerradas.	56
3.1.2 Trayectorias abiertas:	60
3.1.3 Trayectorias abierta (deseada) – cerrada (generada).....	61
3.1.4 Rotación de curvas y función objetivo	63
3.1.5 Escalado.....	64
3.1.6 Utilización de coordenadas polares.....	65
3.2 APLICACIÓN AL MECANISMO CUADRILÁTERO ARTICULADO.....	66
3.3 EJEMPLOS	68
3.4 CONCLUSIONES.....	76

4 SÍNTESIS ÓPTIMA: SÍNTESIS DE MECANISMOS BASADA EN MÉTODOS DE OPTIMIZACIÓN LOCAL.....78

4.1	DEFINICIÓN CINEMÁTICA DEL MECANISMO.....	79
4.2	PLANTEAMIENTO DEL PROBLEMA DE SÍNTESIS.....	81
4.3	RESOLUCIÓN DEL PROBLEMA DE SÍNTESIS.....	82
4.3.1	<i>Cálculo de matriz Jacobiana</i>	83
4.3.2	<i>Cálculo de tamaño de paso / módulo de la dirección de descenso</i>	84
4.3.3	<i>Verificación del ensamblaje</i>	88
4.3.4	<i>Criterios de convergencia-parada.....</i>	89
4.3.5	<i>Diagrama de flujo</i>	92
4.4	EJEMPLOS DE SÍNTESIS LOCAL	95
4.5	CONCLUSIONES.....	104

5 SÍNTESIS GLOBAL: SÍNTESIS DE MECANISMOS BASADA EN ALGORITMOS EVOLUTIVOS.....108

5.1	INTRODUCCIÓN.....	108
5.2	REVISIÓN DE LA HERRAMIENTA DE MATLAB® DE MÉTODOS EVOLUTIVOS	109
5.2.1	<i>Parámetros generales disponibles en la herramienta de algoritmos evolutivos de MATLAB®</i>	112
5.2.2	<i>Parámetros del operador de selección</i>	113
5.2.3	<i>Parámetros del operador de cruce</i>	114
5.2.4	<i>Parámetros del operador de mutación</i>	116
5.3	DEFINICIÓN DE PRUEBAS Y RESULTADOS CON LA HERRAMIENTA DE ALGORITMOS EVOLUTIVOS DE MATLAB®	117
5.3.1	<i>Prueba 1: Convergencia de los algoritmos evolutivos.....</i>	117
5.3.2	<i>Prueba 2: Estudio de proporción de cruce y mutación</i>	119
5.3.3	<i>Prueba 3: Influencia del tipo de cruce</i>	122
5.3.4	<i>Prueba 4: Influencia del cruce comparado con el tamaño de la población .</i>	123
5.3.5	<i>Prueba 5: Influencia de la tasa cruce mutación con tamaños de población mayores, cruce aritmético y 20 generaciones como máximo</i>	126
5.3.6	<i>Prueba 6: Media geométrica vs aritmética</i>	128
5.3.7	<i>Prueba 7: Tipos de cruce</i>	130
5.4	MÉTODO EVOLUTIVO DESARROLLADO PARA LA SÍNTESIS DIMENSIONAL DE MECANISMOS	131
5.4.1	<i>Programación</i>	131
5.4.2	<i>Definición de parámetros generales del programa.....</i>	132
5.4.3	<i>Generación de la población inicial.....</i>	132
5.4.4	<i>Cálculo de la aptitud de individuos</i>	133
5.4.5	<i>Escalado de la aptitud: probabilidad de un individuo de participar en operadores de cruce o mutación</i>	133
5.4.6	<i>Selección de padres</i>	134
5.4.7	<i>Operadores para la creación de una nueva generación de individuos..</i>	136
5.4.8	<i>Pruebas realizadas.....</i>	137
5.5	EJEMPLOS DEL MÉTODO PROGRAMADO CON 4 TRAYECTORIAS DIFERENTES.	144
5.5.1	<i>Trayectoria L.....</i>	145
5.5.2	<i>Elipse con punto anguloso</i>	148

5.5.3	<i>Gota de agua</i>	151
5.5.4	<i>Limason</i>	154
5.6	CONCLUSIONES.....	157
6	MÉTODOS HÍBRIDOS PARA SÍNTESIS DIMENSIONAL DE MECANISMOS. ..	160
6.1	INTRODUCCIÓN.....	160
6.2	ALGORITMO HÍBRIDO 1 (AH1): HIBRIDACIÓN EN EL ESTIMADOR DE ERROR	161
6.3	ALGORITMO HÍBRIDO 2 (AH2): HIBRIDACIÓN EN EL ESTIMADOR DE ERROR Y EN LA SOLUCIÓN FINAL.....	163
6.4	ALGORITMO HÍBRIDO 3 (AH3): OPTIMIZACIÓN LOCAL DE TODOS LOS INDIVIDUOS....	164
6.5	ALGORITMO HÍBRIDO 4 (AH4): OPTIMIZACIÓN LOCAL CENTRADA EN LA ÉLITE.	166
6.6	EJEMPLOS	169
6.6.1	<i>Trayectoria L</i>	170
6.6.2	<i>Elipse con punto anguloso</i>	172
6.6.3	<i>Gota de agua</i>	175
6.6.4	<i>Limason</i>	177
6.7	EJEMPLO CON RESTRICCIONES DE TAMAÑO.....	180
6.8	APLICACIÓN DEL MÉTODO 1 Y 3 A UN MECANISMO DE INYECCIÓN	189
6.9	CONCLUSIONES DE LOS EJEMPLOS EXPUESTOS.....	192
7	CONCLUSIONES Y TRABAJOS FUTUROS	200
7.1	CONCLUSIONES.....	200
7.2	TRABAJOS FUTUROS	203
ANEXO A		205
ANEXO B		207
ANEXO C		209
ANEXO D		211
ANEXO E		213
ANEXO F		217
ANEXO G		221
ANEXO H		225
REFERENCIAS		229

Capítulo 1

“La autocomplacencia personal es el fin del científico. La autocomplacencia colectiva es el fin de la investigación. Son la inquietud, la ansiedad, el descontento y la agonía de la mente lo que nutre la ciencia.”

Jacques-Lucien Monod

1 Fundamentos del trabajo y estructura de la tesis

1.1 Fundamento y objetivos de esta Tesis.

Este trabajo se enmarca dentro de una de las líneas de investigación del Grupo de Ingeniería Mecánica de la Universidad de Cantabria sobre la síntesis de mecanismos. Los trabajos principales previos en los que se fundamenta esta Tesis son (Sancibrián, et al., 2004), (Sancibrián, et al., 2006), la Tesis Doctoral de García (2011) y la Tesis Doctoral de De Juan (2011).

La meta última de esta línea de investigación es desarrollar una herramienta de ayuda para el diseñador en la que éste especifique únicamente los requisitos que un mecanismo debe cumplir. Mediante este planteamiento, la herramienta debe ser capaz no sólo de realizar un proceso de optimización, sino de ofrecer al diseñador distintas soluciones que cumplan los requisitos de diseño de manera adecuada y con una cierta tolerancia.

La metodología desarrollada hasta ahora se basa en métodos de síntesis óptima dimensional mediante el cálculo exacto del gradiente, aplicada con éxito a diferentes tipos de mecanismos planos y espaciales, así como en el desarrollo de esta metodología para problemas de orden 1 (síntesis de velocidades) y problemas de orden 0 y orden 1 de manera simultánea.

De forma complementaria al trabajo desarrollado hasta la fecha, el objetivo general de esta Tesis es contribuir y enriquecer la metodología con los siguientes objetivos parciales:

- Estudio y mejora de estimadores de error: hasta ahora el estimador de error utilizado es el de distancias medias cuadráticas, que no es capaz de tener en cuenta posibles desplazamientos, rotaciones u homotecia entre trayectorias objetivo y generadas. Un objetivo parcial de esta tesis es generar un estimador de error que tenga en cuenta estos aspectos y que no implique un aumento significativo del coste computacional.
- Ajustar el método de síntesis óptima en base al cálculo exacto de gradiente: mejorando aspectos como el cálculo de la dirección de

descenso, tamaño de paso y estudio de posibles estrategias de optimización (optimización por fases).

- Evaluación de métodos estocásticos y desarrollo de un método propio para el problema de síntesis dimensional de mecanismos.
- En caso de que los métodos estocásticos obtengan resultados aceptables, generar métodos híbridos que mantengan en lo posible las ventajas de los métodos estocásticos y de los métodos locales.

1.2 Estructura del documento

Este trabajo se divide en siete capítulos.

En el primer capítulo se expone la motivación y los objetivos que se deberían alcanzar con el mismo.

El segundo capítulo hace una revisión bibliográfica de diferentes aspectos de interés en la síntesis de mecanismos, tales como estimadores de error, métodos de síntesis óptima basada en gradiente y síntesis óptima basada en métodos estocásticos.

A lo largo del tercer capítulo se expone el trabajo realizado en la generación de un estimador de error que considere aspectos como la traslación, rotación y escalado entre el objetivo y la función generada.

En el cuarto capítulo se expone el desarrollo del método de síntesis óptima basada en gradiente desarrollado, explicando aspectos como la obtención de la dirección de optimización y sus estrategias de tamaño de paso (módulo del vector de cambio de variables).

La síntesis óptima en base a métodos evolutivos se desarrolla a lo largo del quinto capítulo. En él se analiza la viabilidad de este tipo de métodos y se propone un método propio para el problema de síntesis de mecanismos.

El sexto capítulo presenta diferentes propuestas de métodos híbridos en los que se combina el método de síntesis basado en el cálculo de gradiente presentado en el capítulo cuarto, con el método estocástico desarrollado en el quinto capítulo. Los métodos híbridos propuestos se

prueban para diferentes objetivos de trayectoria, con restricciones y con un mecanismo diferente al cuadrilátero articulado.

Finalmente el séptimo capítulo recopila las conclusiones de este trabajo y propone líneas de trabajo futuras que den continuidad al trabajo realizado. Tras estos capítulos se presenta un anexo recogiendo resultados de las pruebas realizadas con los métodos. Por último se presenta un bloque de referencias bibliográficas a las que se hace alusión en este documento.

Capítulo 2

“Véase ahora el poder de la verdad, el mismo experimento que a primera vista parecía mostrar una cosa, cuando se examina con más cuidado, nos demuestra lo contrario.”

Galileo Galilei

2 Antecedentes

2.1 Síntesis cinemática de mecanismos

Según el diccionario de la Real Academia Española se define síntesis como “Composición de un todo por la reunión de sus partes”. En el caso de la síntesis de mecanismos, se trata de componer un mecanismo organizando y definiendo cada una de sus partes, para cumplir con cierta funcionalidad u objetivo. El cumplimiento de la funcionalidad puede darse de forma exacta, o lo más aproximada posible mediante procesos de optimización. El diseño de un mecanismo se podría representar de acuerdo a lo expuesto en el trabajo de Noriega (Noriega, 2008), como se muestra en la Fig. 2.1.

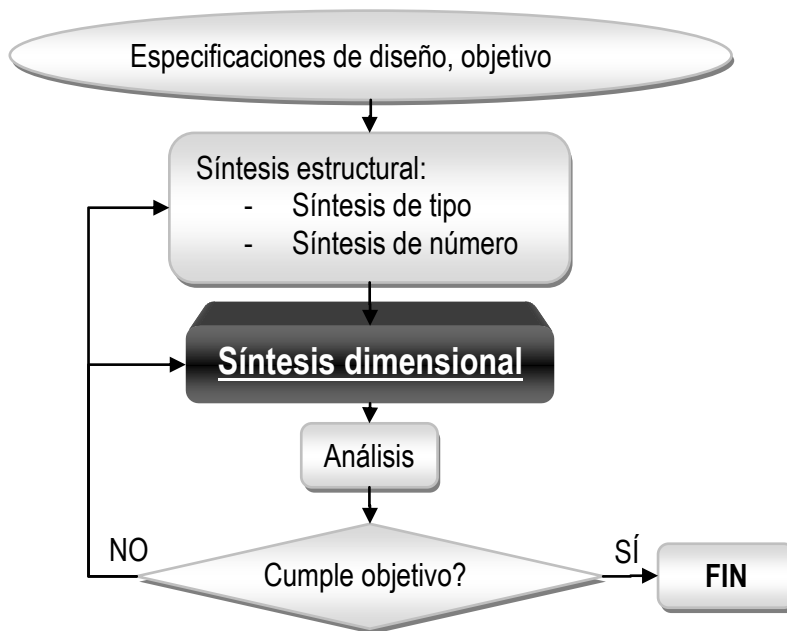


Fig. 2.1 Diagrama de flujo de un proceso de síntesis de mecanismos

Como punto de partida se debe definir qué funcionalidad se quiere obtener en el proceso de diseño. Esta funcionalidad puede hacer referencia a diferentes criterios, por ejemplo cinemáticos, dinámicos u otros.

Normalmente esta búsqueda de funcionalidad conlleva también ciertas restricciones en la solución, o restricciones de diseño.

Tras la definición de la funcionalidad, se debe definir el tipo de mecanismo y el número de elementos que lo componen, mediante técnicas de síntesis de tipo y número, completando así el proceso de síntesis estructural, que proporciona la definición topológica del mecanismo.

Una vez definida la topología del mecanismo se deben definir las dimensiones de cada uno de los componentes del mecanismo, mediante la síntesis dimensional. Así mediante el proceso de síntesis dimensional se definirán las variables de diseño (dimensiones de los elementos).

Comúnmente se puede clasificar la síntesis cinemática dimensional según el tipo de objetivo que debe alcanzar en:

- **Generación de trayectoria:** en este tipo de problema de síntesis cinemática dimensional se trata de dar dimensiones adecuadas a un mecanismo para que uno de sus puntos trace una trayectoria objetivo.
- **Generación de funciones:** en este caso la función objetivo consiste en obtener cierta relación de transmisión entre un elemento de entrada y otro de salida.
- **Guiado de sólido rígido:** la función objetivo en este tipo de síntesis es similar al de generación de trayectoria, pero en este caso no sólo uno de los puntos del mecanismo debe describir una trayectoria sino además debe hacerlo cumpliendo con que el elemento tenga cierta orientación a lo largo de la trayectoria.

Una vez finalizada la síntesis dimensional se debe evaluar la respuesta del mecanismo obtenido, para poderla comparar así con el objetivo fijado (funcionalidad). Esta comparación permite evaluar la idoneidad del mecanismo generado a lo largo del proceso. Si el mecanismo cumple con el objetivo se finaliza el proceso global de síntesis, si no cumple se debe iterar, bien modificando el diseño estructural o bien modificando las dimensiones del individuo evaluado.

Mediante el proceso de síntesis se encuentra un mecanismo que se comporta de la forma establecida en la función objetivo en ciertas posiciones o puntos. Así “la síntesis de mecanismos permite determinar el

mecanismo capaz de ofrecer una respuesta preestablecida de antemano” (Nieto, 1978). Las posiciones o puntos en los que se cumplen estas condiciones son los *puntos o posiciones de precisión*.

En la Fig. 2.1 se ha resaltado el recuadro de síntesis dimensional, ya que este trabajo está centrado en la misma, y más concretamente en la síntesis cinemática dimensional.

Como se recoge en el trabajo de Vallejo (Vallejo, 1993) “a lo largo de la Historia de la Ciencia de Máquinas y Mecanismos, han ido surgiendo diversos métodos de síntesis cinemática que se pueden clasificar atendiendo a diversas consideraciones”. Una posible clasificación es en función de las técnicas utilizadas en el proceso de síntesis, en métodos gráficos, métodos analíticos y métodos numéricos.

2.2 Métodos exactos

Los métodos exactos permiten realizar un ejercicio de síntesis de mecanismos con un objetivo en un número de puntos de precisión reducido y con una solución exacta para esos puntos de precisión. Sin embargo, presentan el problema de que no se puede garantizar que el orden de los puntos de precisión sea el deseado. Dentro de los métodos exactos se puede realizar una clasificación en dos grandes grupos: los métodos gráficos y los métodos analíticos.

2.2.1 Métodos gráficos

Los métodos gráficos permiten realizar un ejercicio de síntesis de forma rápida, visual e intuitiva, y fueron los primeros en desarrollarse. Sin embargo, presentan las restricciones de que sólo son válidos para mecanismos planos y con un número de puntos de precisión reducido.

De entre las múltiples contribuciones a este tipo de métodos se quieren destacar los trabajos de Burns (Burns, 1989), Kimbrell (Kimbrell, 1984), Farhang (Farhang, et al., 1988) (Farhang, et al., 1988), Sun (Sun & Waldron, 1981), Bagci (Bagci, 1977), Chuang (Chuang & Chiang, 1987), y Mitchiner (Mitchiner & Mabie, 1977) (Mitchiner & Mabie, 1977).

Uno de los métodos más populares de aplicación en problemas de generación de funciones y trayectorias, y que es aplicable a cualquier tipo de mecanismo plano es el Overlay, desarrollado por Svoboda (Svoboda, 1965) y Oblonsky (Oblonsky, 1980).

2.2.2 Métodos analíticos

Estos métodos permiten estudiar mediante ecuaciones analíticas que relacionan parámetros de diseño establecidos el comportamiento de los mecanismos. Parte de estos métodos son desarrollos analíticos basados en los métodos gráficos. Aunque la resolución de las ecuaciones en un principio podría hacer que estos métodos fuesen más tediosos que los métodos gráficos, el desarrollo de los ordenadores y su fácil implementación ha permitido que su uso se incrementase en detrimento de los métodos gráficos.

Al igual que los métodos gráficos, estos métodos tienen el inconveniente de que sólo permiten un número de puntos de precisión reducido. Esta restricción se debe a que el sistema de ecuaciones que describe el funcionamiento del mecanismo tiene un número reducido de ecuaciones, y este número es el máximo número de puntos de precisión que pueden considerar.

Existen dos grandes grupos de métodos analíticos: algebraicos y de números complejos. Dentro de los algebraicos se pueden destacar los trabajos de Kramer (Kramer & Sandor, 1970), Litvin (Litvin, 1980), Akhras (Akhras & Angeles, 1990), Dranga (Dranga & Manolescu, 1977), Oren (Oren & Kaufman, 1984), Akçali (Akçali & Dittrich, 1989), (Sutherland & Karwa, 1978) y Ahmad (Ahmad & Waldron, 1979). De los basados en números complejos se quieren citar los trabajos de Nieto (Nieto, 1978), Sandor (Sandor, 1984) y el método de Bloch.

Además de los métodos mencionados es de especial relevancia la ecuación de Freudenstein (Freudenstein, 1955), desarrollado en un principio para generación de funciones con un cuadrilátero articulado, pudiendo generalizarse a otros mecanismos; el método de los polinomios de Chebyshev o los métodos basados en la teoría de Burmester (Burmester, 1888).

2.3 Métodos de síntesis óptima

La utilización de métodos numéricos para resolver un problema de síntesis dimensional conlleva el uso de algoritmos de optimización, de forma que se minimice la diferencia entre el comportamiento mostrado por el mecanismo resultado de síntesis y el comportamiento objetivo.

Estos tipos de métodos son aplicables independientemente de la topología del mecanismo, ampliando la generalidad mostrada por los métodos analíticos. Permiten además un mayor número de puntos de precisión y por lo tanto un mayor número de variables de diseño. Sin embargo, presentan la desventaja de no poder garantizar una solución exacta aún cuando esta exista. Las ventajas expuestas junto con un incremento de la capacidad de procesamiento de los computadores ha favorecido un amplio desarrollo de este tipo de métodos en los últimos años.

El amplio abanico de métodos de optimización numérica es debido a que no existe un método que se adecue a cualquier tipo de problema. Así dependiendo del tipo de objetivo, linealidad del problema, continuidad de las variables de diseño, existencia de restricciones de diseño, etc., el método que mejor se adecue al problema puede variar.

Si se considera el problema de síntesis dimensional óptima se deben considerar las variables de diseño. Estas variables de diseño deben representar mediante valores numéricos los parámetros geométricos que definen el mecanismo, como tamaño, posición y orientación de todos sus elementos. Las variables de diseño se agrupan en el vector $\mathbf{w}$ mostrado en la Ecuación (2.1).

$$\mathbf{w}^T = [w_1 \ w_2 \ w_3 \ w_4 \ \dots \ w_n] \ \forall \mathbf{w} \in R \quad (2.1)$$

Se debe considerar también la función objetivo, normalmente minimizar el error que presente el mecanismo frente al objetivo buscado, y sujeto a ciertas condiciones o restricciones. Dentro de estas restricciones están por ejemplo las ecuaciones de restricción cinemática, que describen el comportamiento del mecanismo, y pudiéndose escribir como se presenta en la Ecuación (2.2), donde $\mathbf{q}$ representa un vector que contiene variables dependientes como las orientaciones de diferentes elementos del mecanismo.

$$\Phi(\mathbf{w}^T, \mathbf{q}) = 0 \quad (2.2)$$

Considerando lo dicho hasta ahora el problema de síntesis se puede presentar matemáticamente como se muestra en la Ecuación (2.3).

$$\begin{aligned} & \min_{\mathbf{w}} \text{Error}(f(\mathbf{w})) \\ & \text{Sujeto a } \Phi(\mathbf{w}^T, \mathbf{q}) = 0 \end{aligned} \quad (2.3)$$

Para resolver este problema se utilizan métodos de optimización iterativos, en los que en cada iteración se obtiene un conjunto de variables de diseño $\mathbf{w}^i$, de forma que se minimice el error de síntesis en cada iteración, hasta obtener un valor mínimo del error.

Se debe tener en cuenta que en el caso de síntesis cinemática dimensional el problema a resolver presenta un comportamiento no lineal, presentando una no linealidad elevada en muchos casos. Es además un problema no convexo de variable continua y con restricciones. Estas características conllevan una función objetivo que contiene un alto número de mínimos locales.

Debido a lo expuesto en el párrafo anterior, un aspecto del problema de síntesis que preocupa a los investigadores que trabajan en este tipo de problemas es el conocer si la solución obtenida tras el proceso de síntesis representa una solución de óptimo local u óptimo global; entendiendo que el óptimo global representa el mínimo error dentro del espacio de diseño.

Una característica que se debe maximizar en este tipo de métodos es su convergencia a la solución. Es decir, es necesario que el método consiga la mejor solución posible con el mínimo coste computacional.

Finalmente hay que reseñar que estos algoritmos deben cumplir criterios de optimalidad, o de que el método es capaz de garantizar que la solución encontrada representa un mínimo error frente al objetivo. En el caso de problemas sin restricciones esto equivale a que el gradiente se anule en el punto de la solución, y dependiendo del signo de la segunda derivada representará un máximo o un mínimo de la función de error. En el caso de que se esté resolviendo un problema con restricciones se deben cumplir las condiciones de Kuhn-Tucker, estando estas condiciones recogidas en

diversos libros sobre optimización como el presentado por Arora (Arora, 2004).

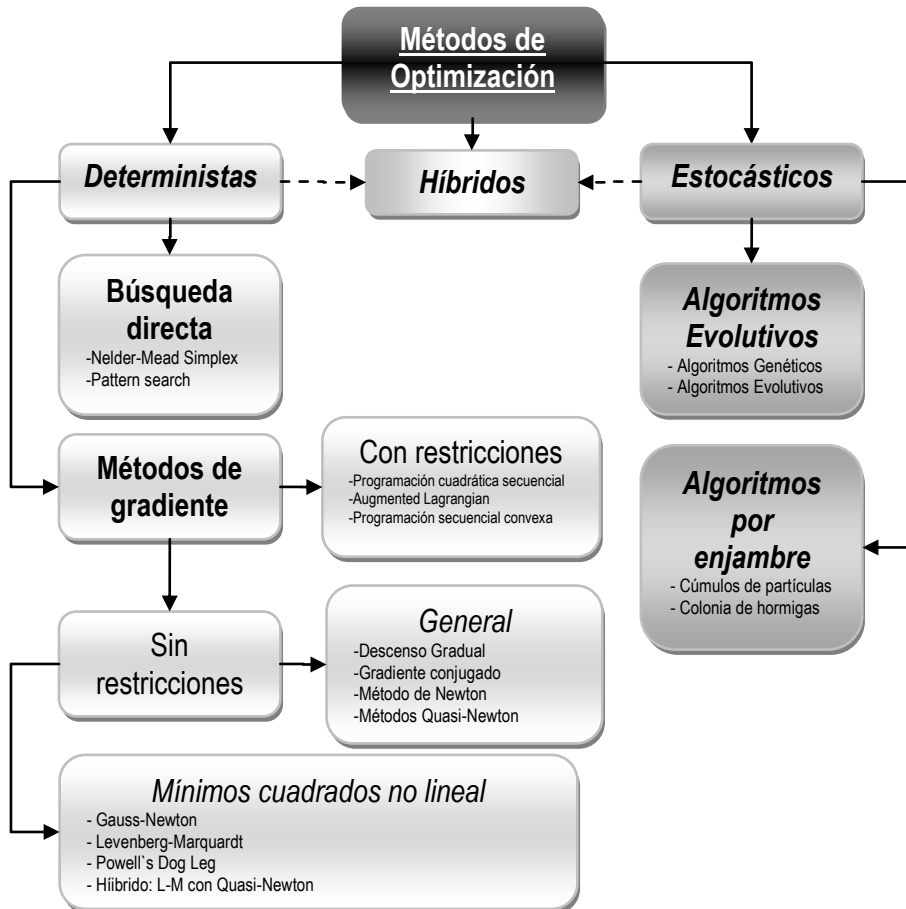


Fig. 2.2 Clasificación de métodos de síntesis óptima

2.3.1 Estimadores de error

Para que un problema de optimización permita mediante su solución obtener el resultado deseado, es fundamental que la función objetivo refleje la realidad de lo que se busca. Es decir, la función objetivo debe ser capaz de expresar matemáticamente cual es el objetivo a minimizar (optimizar). Así, en problemas de síntesis de mecanismos para generación de trayectorias, normalmente la función objetivo es un estimador de error entre la trayectoria objetivo y la trayectoria generada por el individuo evaluado.

Para conseguir una comparación de curvas adecuada, el método de comparación deben cumplir ciertos requisitos como identificación de curvas exactas, siendo capaz de identificar la forma de las curvas y no evaluar las coordenadas que se emplean para su descripción; progresividad, disminuyendo el error linealmente a medida que la trayectoria generada se acerque a la trayectoria deseada; conmutabilidad, obteniéndose el mismo error al comparar la curva generada con la deseada o viceversa; independencia de la posición, siendo capaz de identificar dos curvas como iguales aunque estén desplazadas; independencia de la orientación e independencia del escalado.

Haciendo una recopilación bibliográfica de los métodos de comparación de curva se pueden destacar los siguientes métodos:

2.3.1.1 *Método de Distancias Mínimas Cuadradas DMC*

Es un método de amplio uso, en el que se compara un número discreto de puntos, realizando la comparación entre un par de puntos, uno de la curva objetivo y otro de la curva a evaluar, calculando la distancia entre ambos y realizando un promedio de esta distancia al cuadrado para todos los puntos evaluados.

2.3.1.2 *Método de distancias mínimas cuadradas optimizando posición LSOP*

Este método fue desarrollado por Watanabe (Watanabe, 1992), y está basado en el método DMC, aplicando de forma previa ciertos ajustes sobre

la curva a evaluar, posicionándola respecto a la curva objetivo. El método se puede resumir en los siguientes pasos:

- Obtención de los puntos de las curvas: distribuyendo uniformemente los mismos a lo largo de cada una de las curvas.
- Re-escalado de las curvas: se busca que la longitud de la curva a evaluar sea la misma que la de la curva objetivo aplicando sobre la curva a evaluar un factor de escala.
- Reordenación de la curva a evaluar: en base a la curvatura en cada uno de los puntos se busca el punto de origen en la curva a evaluar que permita obtener el mínimo error en la comparación de curvaturas entre ambas curvas mediante una optimización del error de curvatura.
- Reposicionamiento y reorientación de curva a evaluar: se realiza un problema de optimización en el que se busca la traslación y giro de la trayectoria evaluada para reducir el error de DMC al mínimo.
- Cálculo de error de comparación: evaluado como el máximo valor de la comparación de todos los puntos homólogos.

Los principales problemas de este método son que sólo sirve para curvas planas cerradas, y que no se puede garantizar la progresividad del mismo, es decir, a medida que las curvas tienen mayor similitud no se puede garantizar que el estimador de error arroje un menor valor de error ya que basta con que uno de los puntos difiera más que en otra curva, aunque el resto de los puntos sean coincidentes.

2.3.1.3 Método minimizando el error perpendicular MPE

Este método también fue desarrollado por Watanabe (Watanabe, 1992), y es similar al anterior, aunque en los dos últimos pasos se procede de forma diferente, quedando los pasos a dar como se enuncia a continuación:

- Obtención de los puntos de las curvas: distribuyendo uniformemente los mismos a lo largo de cada una de las curvas.
- Re-escalado de las curvas: se busca que la longitud de la curva a evaluar sea la misma que la de la curva objetivo aplicando sobre la curva a evaluar un factor de escala.

- Reordenación de la curva a evaluar: en base a la curvatura en cada uno de los puntos se busca el punto de origen en la curva a evaluar que permita obtener el mínimo error en la comparación de curvaturas entre ambas curvas mediante una optimización del error de curvatura.
- Reposicionamiento y reorientación de curva a evaluar: en este caso el reposicionamiento y reorientación de la curva con respecto a la función objetivo a optimizar no compara las curvas por distancia media cuadrática, sino por distancia perpendicular cuadrática.
- Cálculo del error de comparación: tomado como el error máximo entre un par de puntos de todos los evaluados.

Este método al igual que el anterior no garantiza su progresividad, y sólo es válido para curvas planas cerradas.

2.3.1.4 Método de conversión en B-Spline o método BSC

El método BSC fue propuesto por Unruh y Krishnaswami (Unruh & Krishnaswami, 1995). Se trata de un método que permite comparar curvas planas cerradas. El método permite obtener un polígono de control de n puntos (puntos de precisión), a partir del cual y mediante una B-Spline que pase por estos puntos se puede reproducir la curva objeto de análisis en todos sus puntos.

Cabe destacar que para una curva determinada y un número de puntos de precisión, sólo existe un polígono de control único, y que ante pequeñas variaciones de la curva, el polígono de control se modifica, de forma que el polígono de control se puede considerar como una representación discreta de la curva, capaz de representar los infinitos puntos que definen la curva.

Con lo explicado en los párrafos anteriores, para comparar dos curvas cerradas se pueden convertir estas curvas en B-Splines, para así obtener sus polígonos de control, de forma que la comparación de las curvas se pueda realizar mediante la comparación de sus polígonos de control. Esto permite comparar las curvas en su globalidad, siendo un buen indicador de comparación entre formas de curvas.

Este método de comparación de curvas se puede resumir en los pasos expuestos a continuación:

- Obtención de los polígonos de control de cada una de las curvas a comparar.
- Una vez obtenidos los dos polígonos de control, se debe idear un método para comparar ambos polígonos. En un principio, y como primer paso de la comparación, se escalan los polígonos para que su perímetro sea idéntico, lo que se conoce como una fase de escalado. Tras igualar el tamaño, se debe comparar la forma de ambos polígonos, para lo que los autores comparan los siguientes parámetros:
 - Ratio área/perímetro
 - Ángulo interior mayor
 - Ángulo interior menor
 - Longitud del lado mayor
 - Longitud del lado menor
 - Ratio de aspecto del menor rectángulo que encierra el polígono
 - Distancia mínima al centroide
 - Distancia máxima al centroide
 - Distancia total al centroide

Este método presenta varios problemas ya que es incapaz de independizar el origen de la parametrización de la curva en polígono de control, ni el sentido de generación de la curva.

2.3.2 Métodos deterministas

A continuación se realizará una revisión bibliográfica de los distintos métodos deterministas. En este tipo de métodos es necesario partir de un vector de variables de diseño determinado. A partir de estas variables de diseño y aplicando uno de estos métodos siempre se obtendrá la misma solución final. Una de las ventajas de estos métodos es su alta convergencia y por lo tanto un bajo coste computacional para encontrar una solución. Sin embargo presentan el inconveniente que para obtener una buena solución en vector de variables de diseño iniciales debe presentar un buen comportamiento o tener cierta predisposición para obtener este buen resultado. Es decir, si el vector de variables de diseño inicial está cerca de un mínimo local de la función de error la solución obtenida será este mínimo, aunque dentro del espacio de trabajo exista una mejor solución. Una posibilidad para evitar este inconveniente es

utilizar diversos individuos de partida, representados por vectores de variables de diseño con diversos valores.

Dentro de los métodos deterministas se van a distinguir dos grandes grupos: los métodos de búsqueda directa y los métodos basados en el gradiente.

2.3.2.1 Métodos de búsqueda directa

Estos métodos están basados en la comparación de los valores de comportamiento obtenidos por la solución en cada iteración con los valores objetivo, sin usar derivadas de estos valores, por lo que también son conocidos como métodos de orden cero. Este tipo de métodos son de utilizad ante funciones objetivo no diferenciables o con derivadas discontinuas, siendo además métodos de aplicación muy general ya que sólo requieren evaluar la función objetivo y que sus valores sean continuos.

Dentro de los métodos de búsqueda directa se quiere destacar el método de Nelder-Mead Simplex (Nelder & Mead, 1965). El método permite encontrar la solución mediante la aplicación de un conjunto de puntos *simplex*, es decir, utilizando un politipo de $n+1$ dimensiones en el espacio de trabajo de n dimensiones de las variables de diseño, aplicando técnicas de reflexión, expansión y contracción.

2.3.2.2 Métodos basados en el gradiente

Se define el gradiente de una función como se muestra en la Ecuación (2.4).

$$\nabla f(\mathbf{w}) = \left\{ \frac{\partial f}{\partial w_1} \quad \frac{\partial f}{\partial w_2} \quad \dots \quad \frac{\partial f}{\partial w_n} \right\} \quad (2.4)$$

Este tipo de métodos tienen mayores exigencias que los de búsqueda directa, ya que requieren que la función objetivo sea continua y derivable, hasta segundo orden en alguno de los casos. Estas condiciones se suelen cumplir con funciones objetivo analíticas. En el caso de que la función objetivo no sea analítica o su derivación implique un elevado coste

computacional, se puede aproximar su derivación mediante métodos de derivación numérica.

Los métodos basados en el gradiente son métodos iterativos y normalmente están estructurados según lo mostrado en la Fig. 2.3:

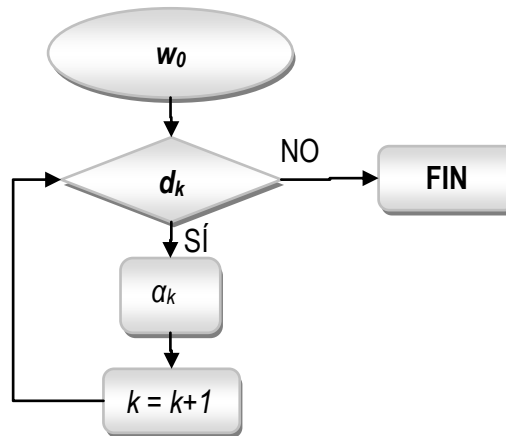


Fig. 2.3 Diagrama de flujo de la estructura básica de los métodos deterministas basados en gradiente

El proceso mostrado en la Fig. 2.3 se puede explicar en los siguientes puntos:

- Se toma un vector inicial de variables de diseño o solución inicial.
- Se calcula una dirección de variación de las variables de diseño, dirección de búsqueda d_k , que permita reducir el error entre el objetivo y el comportamiento mostrado por la solución en la iteración. Así para que d_k proporciona una dirección de reducción de error. Si no es capaz de proporcionar esta dirección es que el proceso ha llegado a un punto estacionario y se finaliza el proceso de iteración
- Tras determinar la dirección de descenso se debe estimar el tamaño de paso. Este tamaño de paso representa el "cuánto" se va a avanzar en esa dirección de descenso mediante el módulo del vector de descenso α_k .
- Como último paso de proceso iterativo se debe estimar una nueva solución para volver a comenzar el proceso iterativo. Esta nueva solución se obtiene modificando la solución inicial, o solución de la

última iteración mediante la dirección de búsqueda y el tamaño de paso.

El flujo de trabajo básico de todos los métodos de gradiente es el presentado en la Fig. 2.3. Los distintos métodos se distinguen por el procedimiento de búsqueda de dirección de descenso y del tamaño de paso. Con el fin de simplificar las expresiones no se tendrá en lo sucesivo el subíndice relativo a las iteraciones.

Una vez calculada la dirección de descenso existen dos metodologías para el cálculo del tamaño de paso:

- Búsqueda lineal
- Región de confianza

Los métodos de búsqueda lineal trabajan estudiando la variación de la función objetivo $f(\mathbf{w})$ en la dirección $\mathbf{d}$ a partir de los valores actuales de las variables de diseño. De esta forma el valor del tamaño de paso debe garantizar que siempre se reduce el valor de la función objetivo. Para conseguir esto el tamaño de paso no debe tener un valor muy reducido, lo que ralentizaría el proceso de convergencia, ni demasiado elevado, ya que se podrían pasar de largo mínimos de la función, lo que conlleva un mayor número de iteraciones y por lo tanto una convergencia más lenta.

Dentro de los métodos de búsqueda lineal se pueden distinguir dos grandes grupos de métodos, los de búsqueda exacta y los de búsqueda aproximada. La búsqueda exacta puede llegar a obtener el mínimo en esa dirección en una sola iteración, sin embargo conlleva usualmente un coste computacional elevado. Debido a este elevado coste computacional, son los métodos de búsqueda aproximada los usados con mayor asiduidad, ya que los métodos de búsqueda aproximada consiguen una reducción aceptable del error frente a la función objetivo con un coste computacional muy inferior.

En cuanto a los métodos basados en **rango de confianza**, se utiliza una aproximación lineal o cuadrática de la función objetivo entorno al individuo de partida de la iteración. Esta aproximación se muestra formalmente en la Ecuación (2.5).

$$f(\mathbf{w} + \mathbf{d}) \approx q(\mathbf{d})$$

$$\text{donde } q(\mathbf{d}) = f(\mathbf{w}) + \mathbf{d}^T \cdot \frac{df(\mathbf{z})}{d\mathbf{z}} \quad (2.5)$$

$$\text{ó } q(\mathbf{d}) = f(\mathbf{w}) + \mathbf{d}^T \cdot \frac{df(\mathbf{z})}{d\mathbf{z}} + \frac{1}{2} \mathbf{d}^T \cdot \frac{d^2 f(\mathbf{z})}{d\mathbf{z}^2} \cdot \mathbf{d}$$

Tanto la aproximación lineal como la cuadrática sólo son aceptables si $\|\mathbf{d}\|$ tiene un valor suficientemente reducido. Para que se cumpla esta condición la dirección $\mathbf{d}$, debe estar contenida dentro de la denominada región de confianza, como se muestra en la Ecuación (2.6).

$$\mathbf{d}^{RC} = \arg \min_{\mathbf{d} \in D} \{q(\mathbf{d})\} \quad (2.6)$$

donde $D = \{\mathbf{d} \mid \|\mathbf{d}\| \leq \Delta\}, \Delta > 0$

El valor de $\mathbf{d}$, obtenido mediante la Ecuación (2.6), debe rechazarse si no representa una dirección de descenso.

Para poder evaluar la validez de la región de confianza se define el parámetro r , con el que se compara la reducción en la función objetivo mediante la aproximación y la propia función objetivo, como se presenta en la Ecuación (2.7), siendo la región de confianza válida siempre que los valores de r sean elevados, pudiendo ampliar la región de confianza si r tiene un valor muy elevado y debiendo reducirla si el valor es pequeño. Más información sobre estos métodos se muestra en el trabajo de Frandsen (Frandsen, et al., 1999)

$$r = \frac{f(\mathbf{w}) - f(\mathbf{w} + \mathbf{d})}{q(\mathbf{0}) - q(\mathbf{d})} \quad (2.7)$$

Método del descenso máximo: también conocido como método de Cauchy. Es de los primeros desarrollados, y aunque su efectividad es reducida, ha servido como punto de partida a otros métodos basados en el gradiente. Este método asigna a la dirección de descenso el valor del gradiente cambiando de signo, tal y como se presenta en la Ecuación (2.8).

$$\mathbf{d} = -\nabla f(\mathbf{w}) \quad (2.8)$$

El método de Cauchy se mejoró en el método de los gradientes conjugados o método de Fletcher-Reeves, en el que se plantea como dirección de descenso una combinación entre la dirección de descenso del método de Cauchy combinada linealmente con la dirección de descenso de la iteración anterior, tal y como se muestra en la Ecuación (2.9).

$$\mathbf{d} = -\nabla f(\mathbf{w}) + \beta \cdot \mathbf{d}$$

$$\text{con } \beta = \frac{\|-\nabla f(\mathbf{w})\|^2}{\|-\nabla f(\mathbf{w})^2\|} \quad (2.9)$$

Este algoritmo consigue, aplicado a funciones objetivo cuadráticas, converger en un número de iteraciones menor o igual al de variables de diseño. En el caso de no ser la función objetivo cuadrática es necesario un mayor número de iteraciones.

Método de Newton y cuasi-Newton: Estos métodos exigen que la función objetivo sea derivable de segundo orden, ya que la dirección de descenso se define mediante su primera derivada o gradiente, y su segunda derivada o matriz Hessiana definida como se muestra en la Ecuación (2.10).

$$\mathbf{H} = \nabla^2 f(\mathbf{w}) = \begin{bmatrix} \frac{df}{dw_1 dw_1} & \frac{df}{dw_1 dw_2} & \cdots & \frac{df}{dw_1 dw_n} \\ \frac{df}{dw_2 dw_1} & \frac{df}{dw_2 dw_2} & \cdots & \frac{df}{dw_2 dw_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{df}{dw_n dw_1} & \frac{df}{dw_n dw_2} & \cdots & \frac{df}{dw_n dw_n} \end{bmatrix} \quad (2.10)$$

La función objetivo se aproxima mediante un desarrollo en serie de Taylor hasta el término de segundo orden entorno al punto de comienzo de iteración (individuo actual), tal y como se presenta en la Ecuación (2.11).

$$f(\mathbf{w}) \approx f(\mathbf{w}) + \nabla f(\mathbf{w})^T \cdot \mathbf{d} + \frac{1}{2} \cdot \mathbf{d}^T \cdot \mathbf{H} \cdot \mathbf{d} \quad (2.11)$$

Con esta aproximación y si la matriz Hessiana es definida positiva, la aproximación tiene un único mínimo que coincide con el punto de gradiente nulo, por lo que la dirección de descenso se puede obtener resolviendo el problema lineal mostrado en la Ecuación (2.12).

$$\mathbf{H} \cdot \mathbf{d} = -\nabla f(\mathbf{w}) \quad (2.12)$$

Este método puede no converger si se obtiene una solución intermedia que coincide con un punto de silla o máximo local de la función objetivo. Este problema se subsana mediante los métodos de Newton amortiguados (Damped Newton Methods), como el de Levenberg-Marquardt.

Además este método presenta el inconveniente de cálculo e inversión de la matriz Hessiana, $\mathbf{H}$, que implica un alto coste computacional. Con el fin de reducir este coste existen métodos cuasi-Newton en los que la inversa de la Hessiana se sustituye por una matriz aproximada $\mathbf{B}$, que se calcula a partir de las derivadas de $f(\mathbf{w})$. En función del método de aproximación de $\mathbf{B}$, existen diferentes métodos de cuasi-Newton, como el DFP (Davidson-Fletcher-Powell) o el BFGS (Broyden-Fletcher-Goldfarb-Shanno), pudiéndose encontrar más información a este respecto en diferente bibliografía como por ejemplo los trabajos de Arora (Arora, 2004), Reklaitis (Reklaitis, et al., 1983) o Belegundu (Belegundu & Chandrupatla, 1999).

Método de Gauss-Newton: Se trata de una variación del método de Newton para resolver problemas de mínimos cuadrados sin necesidad de utilizar derivadas segundas. Un problema de mínimos cuadrados se puede formular como se presenta en la Ecuación (2.13).

$$\min_{\mathbf{w}} F(\mathbf{w}) = \min_{\mathbf{w}} \frac{1}{2} \cdot f(\mathbf{w})^T \cdot f(\mathbf{w}) \quad (2.13)$$

Al poder plantear el problema de esta forma, el método trabaja con una linealización por serie de Taylor de $f(\mathbf{w})$, tal y como se muestra en la Ecuación (2.14).

$$f(\mathbf{w} + \mathbf{d}) \approx f(\mathbf{w}) + \mathbf{J}(\mathbf{z}) \cdot \mathbf{d} \quad (2.14)$$

Siendo $\mathbf{J}$ la matriz Jacobiana de $f(\mathbf{w})$, por lo que la función objetivo de la Ecuación (2.13) se podría reescribir como se presenta en la Ecuación (2.15).

$$\begin{aligned} F(\mathbf{w} + \mathbf{d}) &\approx F(\mathbf{w}) + \frac{1}{2} \cdot (f(\mathbf{w} + \mathbf{d}))^T \cdot f(\mathbf{w} + \mathbf{d}) = \\ &= F(\mathbf{w}) + \mathbf{d}^T \cdot \mathbf{J}^T \cdot f(\mathbf{w}) + \frac{1}{2} \cdot \mathbf{d}^T \cdot (\mathbf{J}^T \cdot \mathbf{J}) \cdot \mathbf{d} = \\ &= F(\mathbf{w}) + \mathbf{d}^T \cdot \nabla F(\mathbf{d}) + \frac{1}{2} \cdot \mathbf{d}^T \cdot (\mathbf{J}^T \cdot \mathbf{J}) \cdot \mathbf{d} \end{aligned} \quad (2.15)$$

Si se aplica ahora la condición de óptimo se obtiene la Ecuación (2.16).

$$(\mathbf{J}^T \cdot \mathbf{J}) \cdot \mathbf{d} = -\mathbf{J}^T \cdot f \quad (2.16)$$

Este método consigue convergencia cuadrática siempre que la matriz Jacobiana sea cuadrada y no singular, muy similar a un Newton-Raphson.

Método de Levenberg-Marquardt: Es un método amortiguado, siendo una modificación a partir del método Gauss Newton, pero a la vez es un método muy rápido. Para conseguir esto, se modifica la matriz pseudo-Hessiana del método de Gauss-Newton, $(\mathbf{J}^T \cdot \mathbf{J})$, adicionando una matriz diagonal con un término de amortiguamiento, obteniendo la dirección de búsqueda tal y como se presenta en la Ecuación (2.17)

$$(\mathbf{J}^T \cdot \mathbf{J} + \mu \cdot \mathbf{I}) \cdot \mathbf{d} = -\mathbf{J}^T \cdot f \quad \forall \mu > 0 \quad (2.17)$$

Siendo $\mathbf{I}$ una matriz identidad y μ el término de amortiguamiento. Si el término de amortiguamiento tiene un valor elevado, el tamaño de paso será reducido pero con una dirección muy similar a la de máximo descenso. Esta coincidencia de la máxima dirección de descenso con la dirección $\mathbf{d}$ obtenida garantiza una buena convergencia hacia mínimos globales de la función objetivo, incluso partiendo de soluciones alejadas del óptimo global. Por otro lado, términos de un valor reducido de amortiguamiento, μ , permiten tamaños de paso mayores con una convergencia cuadrática, como el método Gauss-Newton en las cercanías del mínimo.

Como μ controla tanto el tamaño de paso como la dirección de descenso, se suele actualizar dos veces en cada iteración.

A partir de este método Marquardt consigue obtener una estimación local de la curvatura de la función objetivo, y por tanto una estimación de la matriz Hessiana. Utilizando esta estimación de la matriz Hessiana en lugar de la matriz identidad en la Ecuación (2.32), se genera el método de Levenberg-Marquardt.

Método de los multiplicadores de Lagrange: Permite de manera indirecta resolver problemas de optimización no lineales y con restricciones, modificando el problema inicial mostrado en la Ecuación (2.3), en la Ecuación (2.18).

$$L(\mathbf{w}, \lambda) = f(\mathbf{w}) + \sum_{i=1}^m \lambda_i \cdot \Phi_i(\mathbf{w}) \quad (2.18)$$

La Ecuación (2.18) representa una función Lagrangiana, estando formada por la función objetivo y una combinación lineal de las ecuaciones de restricción con coeficientes λ . Siendo estos coeficientes los llamados multiplicadores de Lagrange. En esta ecuación si se busca el valor de $\mathbf{w}$ que haga la función objetivo mínima y cumpla simultáneamente las ecuaciones de restricción, se puede reducir el problema al sistema de ecuaciones mostrado en la Ecuación (2.19)

$$\left\{ \begin{array}{l} \frac{\partial L(\mathbf{w}, \lambda)}{\partial w_j} = 0 \\ \frac{\partial L(\mathbf{w}, \lambda)}{\partial \lambda_i} = 0 \equiv \Phi_i(\mathbf{w}) = 0 \end{array} \right\} \quad (2.19)$$

Métodos de la función de penalización: Estos métodos tienen en cuenta las restricciones como parte de la función objetivo, penalizando esta función con las ecuaciones de restricción no cumplidas.

El principal problema de este tipo de métodos es un mal condicionamiento de la matriz Hessiana debido a un aumento progresivo de los parámetros de penalización.

Métodos de la función Lagrangiana aumentada: este tipo de métodos salva el problema de mal condicionamiento de la matriz Hessiana, mediante la consideración de un valor máximo del factor de penalización. Así, se propone resolver el problema de optimización combinando el método de los multiplicadores de Lagrange con el de la función de penalización cuadrática, tal y como se muestra en la Ecuación (2.20).

$$A(\mathbf{w}, \lambda, r) = f(\mathbf{w}) + \sum_{i=1}^m \lambda_i \cdot \Phi_i(\mathbf{w}) + \mu \cdot \sum_{i=1}^m (\Phi_i(\mathbf{w}))^2 \quad (2.20)$$

Obteniendo el valor óptimo de los multiplicadores de Lagrange λ_i^* , el valor de $\mathbf{w}^*$ que minimiza la Lagrangiana aumentada, minimiza también la función $f(\mathbf{w})$ para cualquier valor del factor de penalización r . Existiendo además un valor finito máximo del factor de penalización μ , con el que la función aumentada proporcione un valor mínimo.

Mediante este razonamiento se puede sustituir el problema de búsqueda de $\mathbf{w}^*$, por la búsqueda de los multiplicadores λ_i^* , que permiten obtener un valor mínimo de la función aumentada de forma iterativa. Esto es posible si se considera condición de punto estacionario de la función Lagrangiana L , y de la función Lagrangiana aumentada A , mostradas en las Ecuaciones (2.21) y (2.22).

$$\frac{\partial L}{\partial w_i} = \frac{\partial f}{\partial w_i} + \sum_{j=1}^m \lambda_j \cdot \frac{\partial \Phi_j}{\partial w_i} = 0 \quad \forall i = 1, 2, \dots, n \quad (2.21)$$

$$\frac{\partial A}{\partial w_i} = \frac{\partial f}{\partial w_i} + \sum_{j=1}^m (\lambda_j + 2 \cdot \mu \cdot \Phi_j) \cdot \frac{\partial \Phi_j}{\partial w_i} = 0 \quad \forall i = 1, 2, \dots, n \quad (2.22)$$

Operando se puede obtener la Ecuación (2.23) a través de la que se pueden actualizar iterativamente los multiplicadores de Lagrange en función del factor de penalización.

$$\lambda_j^* = \lambda_j + 2 \cdot \mu \cdot \Phi_j \quad \forall j = 1, 2, \dots, m \quad (2.23)$$

Método del gradiente reducido generalizado: Es un método muy utilizado en el caso de tener que resolver problemas con restricciones. Este método se basa en el método del gradiente reducido, extendiéndolo para problemas no lineales y restricciones lineales, y ampliado a restricciones no lineales por Abadie (Abadie & Carpenter, 1969).

El algoritmo presenta la restricción de que el punto de partida debe cumplir las restricciones impuestas en el problema de optimización. Si se cumple esta condición el algoritmo modifica la solución en una dirección de descenso sin dejar de cumplir con las condiciones de restricción, repitiendo esta operación de forma iterativa hasta llegar a un punto en el que el algoritmo no puede encontrar una dirección de modificación del individuo en la que se pueda reducir la función objetivo.

Suponiendo un problema no lineal, con restricciones de igualdad y contorno, como el presentado en la Ecuación (2.24).

$$\begin{aligned} & \min_{\mathbf{w}} f(\mathbf{w}) \\ \text{sujeto a } & \Phi(\mathbf{w}) = \{\Phi_1 \quad \dots \quad \Phi_m\}^T = 0 \\ & \text{con } w_i^{(1)} \leq w_i \leq w_i^{(u)} \forall i = 1, 2, \dots, n \end{aligned} \quad (2.24)$$

Teniendo en cuenta que las variables $\mathbf{w}$ se pueden dividir en dos grupos, variables independientes, $\mathbf{x}$ (n-m variables), y variables dependientes $\mathbf{y}$ (m variables), se pueden diferenciar las funciones objetivo y de restricción como se muestra en la Ecuación (2.25).

$$\begin{aligned} df(\mathbf{w}) &= \nabla_{\mathbf{x}}^T f \cdot d\mathbf{x} + \nabla_{\mathbf{y}}^T f \cdot d\mathbf{y} \\ d\Phi(\mathbf{w}) &= \nabla_{\mathbf{x}}^T \Phi \cdot d\mathbf{x} + \nabla_{\mathbf{y}}^T \Phi \cdot d\mathbf{y} \end{aligned} \quad (2.25)$$

Al deberse cumplir que los nuevos individuos cumplan las ecuaciones de restricción se puede aplicar que $d\Phi = 0$. Por lo que sustituyendo en la Ecuación (2.25), se obtiene la Ecuación (2.26).

$$d\mathbf{y} = -[\nabla_{\mathbf{y}}^T \Phi]^{-1} \nabla_{\mathbf{x}}^T \Phi \cdot d\mathbf{x} \quad (2.26)$$

Y sustituyendo la Ecuación (2.26) en la Ecuación (2.25), se obtiene la Ecuación (2.27).

$$\frac{df(\mathbf{w})}{d\mathbf{x}} = \nabla_{\mathbf{x}}^T f - \nabla_{\mathbf{y}}^T f \cdot [\nabla_{\mathbf{y}}^T \Phi]^{-1} \nabla_{\mathbf{x}}^T \Phi = \mathbf{G}_R \quad (2.27)$$

Siendo $\mathbf{G}_R$ el *gradiente reducido generalizado*, GRG, que se utiliza en este método para determinar la dirección de búsqueda para actualizar las variables independientes.

Método del recocido simulado: Un método a medio camino entre los métodos deterministas y los métodos probabilísticos es el **método del recocido simulado**. Este método trata de simular el proceso de recocido metalúrgico, en el que mediante un calentamiento y un enfriamiento controlado se consigue que la estructura metalúrgica del material almacene la energía interna mínima. El método fue propuesto por Cerny (Cerny, 1985) y Kirkpatrick (Kirkpatrick, et al., 1983). De la misma forma el método del recocido simulado, a partir de un valor de solución inicial, busca una transición a otro individuo que permita reducir el error frente a la función objetivo mediante una probabilidad, que depende de parámetros que son análogos a la temperatura del material y de la energía interna del mismo. Esa probabilidad favorece siempre la variación del individuo hacia estados de menor energía (menor error), y nunca es negativa. Esto permite poder evitar el estancamiento en mínimos locales, al tener en cuenta el parámetro de temperatura, que es siempre decreciente. Esta característica de poder evitar estancamiento en mínimos locales se reduce a medida que el parámetro temperatura disminuye. Así el valor inicial del parámetro que emula la temperatura y la forma de reducirlo a lo largo del algoritmo (proceso de recocido), marca la efectividad del método.

2.3.3 Métodos probabilísticos

2.3.3.1 Generalidades de los métodos evolutivos

Los métodos evolutivos son métodos probabilísticos de búsqueda aplicables a obtener soluciones en problemas de optimización. Estos métodos están basados en algoritmos inspirados en las leyes de evolución de especies que se dan en la naturaleza. Estos algoritmos trabajan sobre una población de individuos diferenciados por su cromosoma. Así, cada uno de estos individuos tiene un cromosoma que le identifica, compuesto a

su vez por genes. Estos genes varían a lo largo de las diferentes generaciones de la población atendiendo a operadores que emulan las leyes naturales de selección y reproducción, cruzamiento y mutación, permitiendo que los individuos se adapten mejor a un medio dado a medida que avanza el número de generaciones.

Una referencia importante en la descripción de las leyes de evolución natural es la proporcionada por Charles Darwin (Darwin, 1859). Darwin propone que ante un medio con recursos limitados, los individuos genéticamente mejor adaptados al medio serán los más fuertes, con mayor posibilidad de supervivencia, y por lo tanto con mayores probabilidades de descendencia, que heredará las características genéticas de dichos individuos. Haciendo una analogía con los problemas de optimización, el medio vendría definido mediante una función objetivo que permitiría definir la aptitud de cada uno de los individuos, y ciertas restricciones que obligarían a estos individuos a evolucionar dentro de un espacio de búsqueda limitado.

Siguiendo con la analogía, cada generación del método estaría compuesta por np individuos con su cromosoma asociado. Este cromosoma compuesto por genes, que define una posición en el espacio de búsqueda n -dimensional de la función objetivo $f(\mathbf{w})$. De esta forma, cada una de las variables de diseño estará definida por un gen que forma el cromosoma, es decir, el cromosoma está formado por las diferentes variables de diseño codificadas. Esta codificación puede tener una representación binaria, como se ha hecho en el enfoque clásico de algoritmos genéticos, entera o real. En este trabajo la descripción de los diferentes genes se ha realizado mediante números reales, teniendo el cromosoma una estructura de sucesión de números reales, tantos como variables de diseño, genes, contenga el cromosoma.

Llegado este punto, se debe diferenciar entre genotipo, constitución genética de un individuo, y fenotipo, características que el individuo tiene en base a su genotipo. Así, el fenotipo, no sólo refleja la aptitud genética de un individuo, sino que también refleja las capacidades adquiridas a lo largo de su vida. En la teoría de la evolución desarrollada por Darwin, los individuos de sucesivas generaciones se adaptan cada vez mejor a su medio y restricciones (función objetivo y restricciones en una optimización matemática), gracias a la mejora de su genotipo a lo largo de las generaciones, por las leyes de selección, reproducción, y mutación ya citadas. Sin embargo, las capacidades adquiridas no se heredan de una

generación a otra. Lamarck sin embargo, desarrolló una teoría similar a la Darwiniana, pero contrapuesta, ya que según Lamarck, los caracteres adquiridos se traspasan de una generación a otra. En términos de teoría evolutiva, se ha demostrado que la teoría desarrollada por Lamarck no es válida en la naturaleza. Sin embargo, en problemas de optimización, el combinar ambas teorías puede ser beneficioso. Así, en este trabajo, se va a mostrar un método de optimización evolutivo, basado en la teoría Darwiniana, pero que puede incluir también la teoría desarrollada por Lamarck, es decir, se puede considerar el “aprendizaje” o mejora de aptitudes de un individuo a lo largo de su vida (en la generación que corresponda), y posteriormente transmitir este “aprendizaje” o mejora de aptitudes a la siguiente generación.

Tal y como se viene describiendo hasta ahora, se debe tener en cuenta que los métodos evolutivos son métodos iterativos. Así a partir de una población inicial $P(0)$, se van generando T generaciones de individuos, en las que el número de individuos es constante, y a través de las cuales, la función objetivo se consiga cada vez con un menor error, obteniendo el mejor individuo en la generación $P(T)$. Para crear generaciones sucesivas, a partir de la primera generación, se utilizan operadores genéticos, tal y como se explicará de forma esquemática en los siguientes puntos.

2.3.3.2 Esquema básico de un algoritmo evolutivo.

El esquema básico de un posible algoritmo evolutivo se presenta en la Fig. 2.4:

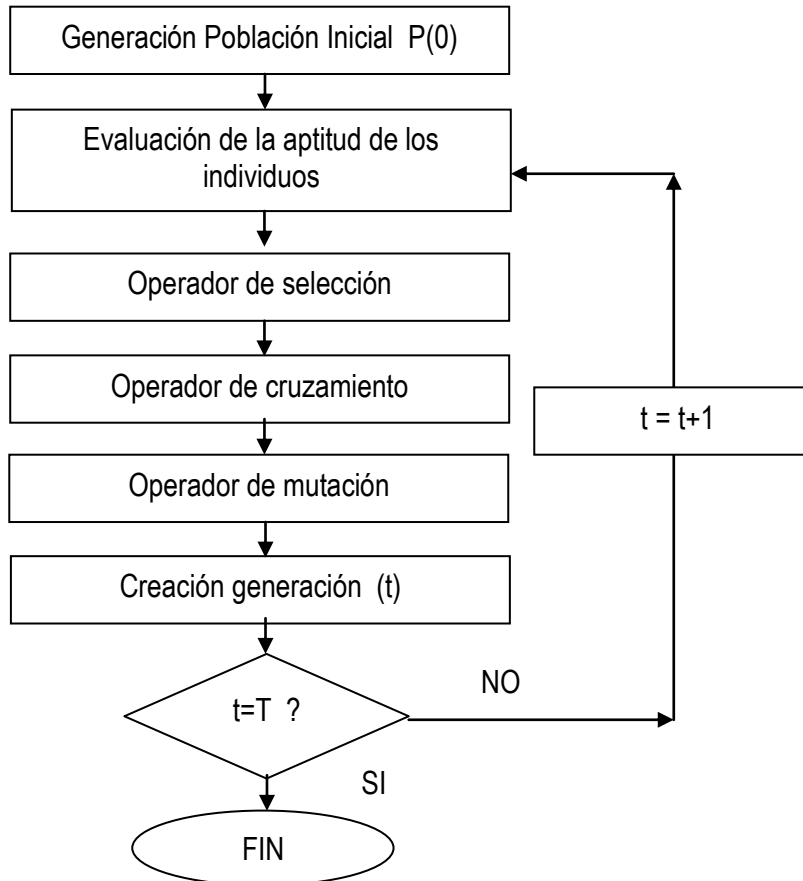


Fig. 2.4 Esquema básico de algoritmo evolutivo

Según este esquema, a partir de una población inicial, generada de forma aleatoria, se clasifican los individuos según su aptitud (adaptación a la función objetivo), y en función de esta clasificación se agrupan para generar nuevos individuos según operadores como el operador de cruce o mutación. Así de forma secuencial se van generando individuos cada vez mejor adaptados al entorno propuesto (que se ajusten mejor a la función objetivo impuesta, así como a las posibles restricciones).

Se van a describir a continuación los diferentes operadores básicos. Así, el **operador de selección** clasifica los distintos individuos de una

población y les asigna una probabilidad de selección en función de su aptitud, teniendo un valor de probabilidad asignado mayor cuanto mayor sea la aptitud del individuo. Se debe tener en cuenta que el valor de probabilidad asignado debe normalizarse, para que el sumatorio de probabilidades de los individuos de una generación no supere la unidad.

En algunos algoritmos, al contrario que en la naturaleza, los individuos con un valor de probabilidad más elevado pasan directamente a la siguiente generación como élite. Aunque esto no se refleja en el algoritmo simplificado presentado, se implementará en el método evolutivo desarrollado, tal y como se explicará posteriormente.

Para el operador de cruzamiento o cruce, y con los individuos de la generación y sus valores de probabilidad correspondientes, se seleccionan parejas de individuos según una función de probabilidad, por ejemplo una función de Gauss u otra (se utiliza también el método Montecarlo). En esta selección pueden aparecer individuos varias veces, sobre todo los individuos más aptos, que tienen un mayor valor de probabilidad asignado, y que por lo tanto tienen más probabilidades de participar en el operador de cruce. El número de individuos seleccionados para participar en el operador de cruce depende del número de individuos a generar por este operador, y debe ser, como regla general el doble de los individuos a generar.

En el **operador de cruzamiento**, se toman de forma aleatoria dos individuos de los seleccionados como padres, y se “cruzan” sus cromosomas. Existen varias formas de cruzamiento, desde sustituir parte del cromosoma de uno de los progenitores por los genes del otro progenitor, conocido como cruzamiento de punto único; cruzar los genes uno a uno como media aritmética ponderada; cruzar los genes uno a uno como media geométrica ponderada; etc.

Finalmente y cerrando el primer ciclo de la secuencia, se aplica el **operador de mutación**. Este operador toma individuos de la población inicial y varía de forma aleatoria uno o varios genes. Este operador tiene una probabilidad de actuación, que debe garantizar la diversidad de individuos, para evitar que se converja a una solución o grupo de individuos en una zona del espacio de búsqueda de forma pronta, y que no se evalúen individuos en otras zonas del espacio de búsqueda.

Así la tasa de cruzamiento y mutación, quitando la tasa de élite si es utilizada, deben generar los np (o $np - np_{elite}$) individuos de cada generación, influyendo el valor de estas tasas en la convergencia del problema de forma directa. En algunos métodos estas tasas varían a lo largo de las generaciones, aumentando la tasa de mutación a medida que aumentan las generaciones, para garantizar la diversidad de la población (dentro del espacio de búsqueda).

2.3.3.2.1 *Cromosomas: algunas propuestas para su generación.*

No sólo una definición adecuada de la función objetivo tiene una gran importancia a la hora de abordar un problema de síntesis de mecanismos mediante algoritmos evolutivos, sino también la forma en la que se defina el cromosoma de cada uno de los individuos. En los primeros programas de algoritmos evolutivos (genéticos) era habitual su codificación binaria, es decir, una fila de genes con alelos que tomen un valor 0 ó 1. Más recientemente se vienen utilizando también otras codificaciones de los genes, como un vector fila de valores enteros, o reales, que describen ciertas características del individuo. De forma estricta, sólo se suele hablar de algoritmos genéticos si la codificación de los cromosomas es binario, aunque en el trabajo que se presenta se utiliza una codificación real, por lo que se habla de algoritmos evolutivos, o genéticos indistintamente, ya que es sólo una discusión formal.

La codificación en números binarios presenta el problema de que lo representado por los alelos de los diferentes genes debe ser una propiedad que se represente con un valor discreto, 0 ó 1, por ejemplo la existencia o no de una articulación del mecanismo en cierto punto, la existencia o no de un actuador, etc., pero no una variable que cuantifique por ejemplo la longitud de una barra de cuadrilátero. Para solventar esto, se pueden representar los diferentes genes con alelos que representen números enteros con notación binaria, pero de nuevo, se encuentra la dificultad de que sólo se pueden tomar valores enteros, y en cierto intervalo. Esto a su vez se puede solventar escalando hasta cierto decimal y representado ese valor entero en notación binaria, lo que aumenta el número de dígitos necesarios para la codificación considerablemente.

Así, una representación real de los genes, aporta la posibilidad de generar operadores de cruzamiento y mutación más sencillos, e implica un mejor acondicionamiento del método para una posible combinación con métodos

de optimización numérica local, siendo esta la solución estudiada en este trabajo.

2.3.3.2.2 Selección, cruzamiento y mutación

Como ya se ha comentando en párrafos anteriores, el número de generaciones t , variará desde 0 hasta T . En cada una de las generaciones, se evalúa la función objetivo de cada uno de los np individuos, obteniendo un valor para dicha función $f(\mathbf{w})_{pt}$, donde el subíndice np hace referencia al identificador de individuo dentro de la generación indicada por el subíndice t . Este valor $f(\mathbf{w})_{pt}$ marcará la adaptación de cada individuo a la función objetivo, y una mejor adaptación indicará una mayor probabilidad de selección s_{pt} . El operador que marque esa probabilidad de selección puede variar, desde una normalización, que en el caso de una síntesis dimensional de mecanismos, en la que la función objetivo se debería minimizar, ya que sería una evaluación del error, esta normalización se podría plasmar matemáticamente como:

$$s_{it} = \frac{\frac{1}{f(\mathbf{w})_{it}}}{\sum_1^{np} \frac{1}{f(\mathbf{w})_{it}}} \quad (2.28)$$

Existen otros operadores como el descrito por Rao (Rao & Pan, 1991), o la selección por rango. Tomando el ejemplo anterior, en el que se trata de minimizar el error en una síntesis dimensional de un mecanismo, se daría rango desde 1 hasta np a los diferentes individuos de la generación t , teniendo rango 1 el individuo con menor error y rango np el individuo con mayor error. Una vez fijado el rango, se le asignaría la probabilidad mostrada en la siguiente expresión:

$$s_{it} = \frac{np + 1 - i}{\frac{np \cdot (np + 1)}{2}} \quad (2.29)$$

Esta forma de asignar la probabilidad permite un reparto más uniforme de la misma (mayor variabilidad en elección, al asignar una probabilidad de selección más uniforme entre individuos, sin grandes variaciones en los valores de probabilidad, y menor probabilidad de llegar a un mínimo local

de forma prematura), y además no depende, por ejemplo, de que la función objetivo esté escalada.

Una vez asignados los np valores s_{it} de probabilidad, se procede a la selección de los individuos que se van a utilizar en los operadores de cruce y mutación. Para esto se pueden utilizar diferentes métodos, como el método de la ruleta, el método de Montecarlo, etc. Uno de los métodos más utilizados es el de la ruleta. En este método, y a modo de explicación gráfica, cada uno de los individuos ocupa un sector en un círculo, proporcional a su probabilidad asignada, de forma que tomando un punto de origen, se podría recorrer el círculo mediante un barrido en ángulo, obteniendo para 0° un valor de 0 y para 359.99999° un valor de 1, marcando este valor entre 0 y 1 la probabilidad acumulada de selección de un individuo.

Así, generando un número aleatorio entre 0 y 1, se recorrería el círculo hasta conseguir la probabilidad acumulada marcada por el número aleatorio generado, tantas veces como individuos se deban seleccionar (como se muestra en la Fig. 2.5, para la selección de 2 individuos), para generar mediante operadores de cruzamiento y mutación (y población élite si se utiliza en el método evolutivo) la población $P(t+1)$. Definiéndose de esta forma un operador de selección.

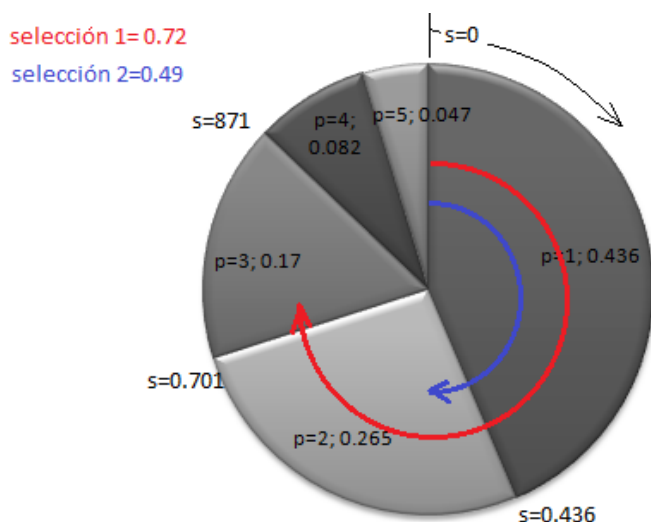


Fig. 2.5 Esquema de selección de 2 individuos mediante el método de la ruleta

El método de selección de la ruleta, tiene como posible inconveniente el no seleccionar los individuos más aptos, y por lo tanto retrasar el proceso de selección. Sin embargo, garantiza la selección de individuos poco aptos, pero con alguna característica interesante. Además si se utiliza la selección de élite, el inconveniente presentado desaparece.

Una vez seleccionados los progenitores, se generará la nueva población en base a los operadores de cruzamiento y de mutación.

El operador de cruzamiento toma dos individuos de los progenitores seleccionados, y a partir de estos progenitores se genera uno (o dos en algunos algoritmos) individuos para la nueva generación. El número de individuos generados por este operador dependerá de la tasa de cruzamiento definida.

En el caso de utilizar cromosomas binarios, existen dos formas habituales de cruzamiento; el cruzamiento de punto único y el cruzamiento uniforme. En el caso de cruzamiento de punto único se genera un número aleatorio entre 1 y el número de genes del cromosoma. Así se tomará de uno de los progenitores los genes desde 1 hasta el número generado y del otro los genes desde el número generado hasta el número máximo de genes.

En el cruzamiento uniforme, una vez seleccionados los dos progenitores, se elige de forma aleatoria un número de veces igual al número de genes del cromosoma 0 ó 1, y en función de si es 0 ó 1 ese gen será igual al del primer o segundo progenitor respectivamente. A modo de ejemplo se presenta un posible cruzamiento uniforme en la Fig. 2.6.

Progenitor 1	1	1	1	0	0	1	1
Progenitor 2	0	0	0	1	1	0	0
Secuencia aleatoria	1	1	0	0	1	0	0
Nuevo individuo	0	0	1	0	1	1	1

Fig. 2.6 Ejemplo de cruzamiento uniforme

En caso de métodos con notación no binaria, tanto real como entera, se pueden aplicar además de este método, otros como medias aritméticas con o sin ponderación aleatoria, medias geométricas, etc.

El operador de mutación en lugar de combinar dos progenitores, trata de variar un solo progenitor en uno o varios genes de forma aleatoria. Este operador es utilizado para diversificar la población objeto de análisis, y evitar así dejar zonas del espacio de diseño sin explorar, es decir, es un operador que trata de impedir que se obtengan soluciones optimizadas localmente mediante exploración del espacio de diseño.

Con notaciones binarias y mediante la tasa de mutación, se define el número de alelos a modificar, para a continuación sustituir en el número de alelos seleccionados, y con una selección aleatoria de los alelos a modificar dentro del cromosoma, su valor por 0 ó 1 de forma también aleatoria, o bien por el valor inverso al existente en el progenitor.

En caso de utilizar notación entera o real, se puede sustituir el gen o los genes por un valor aleatorio dentro del rango de búsqueda.

Con los individuos generados mediante este operador, más los generados por el operador de cruzamiento y los posibles individuos élite, se genera la nueva población, para empezar de nuevo el proceso de evaluación de la aptitud de la población, operador de selección, operador de cruzamiento y operador de mutación, para así generar una nueva población, hasta alcanzar el número máximo de poblaciones, y obtener el individuo “óptimo” como parte de la última población.

Los operadores descritos hasta el momento, son operadores utilizados en el algoritmo genético simple. Además de estos operadores, y si se consideran los métodos evolutivos, se puede hablar de otros operadores como el elitismo, modificación Lamarckiana, etc. Algunos de estos operadores serán descritos a continuación.

2.3.3.2.3 *Mejoras del algoritmo genético básico: Elitismo y modificación Lamarckiana*

El elitismo pretende conservar los mejores individuos dentro de una población en la siguiente población, a fin de que individuos con gran

potencial no desaparezcan al aplicarse los operadores genéticos de cruzamiento y mutación.

El añadir elitismo al método evolutivo, puede tenerse en cuenta con varias estrategias. Una de ellas es archivar la élite de cada generación y dejar que el método evolutivo siga desarrollando las diferentes generaciones de forma paralela, para tras la generación y evaluación de la última generación buscar el mejor individuo entre los individuos de esta población y los individuos élite almacenados. Otra estrategia, es definir una parte de la generación $(t+1)$, con los mejores individuos de la generación t . Mediante esta estrategia se garantiza que el valor de aptitud del individuo con mejor aptitud en la generación $(t+1)$ tiene al menos el valor del individuo mejor adaptado de la generación t . De esta forma, tras generar y evaluar la última generación, la solución vendrá dada por su individuo con mejor aptitud.

Lo descrito hasta ahora del elitismo puede llevar a pensar que utilizar este operador mejoraría la convergencia del método evolutivo. Rudolph (*Rudolph, 1994*) demuestra que un algoritmo genético simple no siempre converge a una solución con aptitud óptima, pero si lo hace si se le añade elitismo.

Además del elitismo, se puede utilizar también la modificación Lamarckiana para mejorar la velocidad de convergencia y mejorar la aptitud del individuo óptimo. Esta modificación trata de reflejar la teoría de Lamarck, que supone que los caracteres adquiridos a lo largo de la vida del individuo se transmiten a otras generaciones. Esto se puede reflejar en los métodos evolutivos suponiendo que los caracteres son adquiridos por el individuo de una generación t mediante una optimización local antes de generar la generación $t+1$ y utilizando el individuo optimizado no sólo en el elitismo, sino también en los operadores de selección, cruzamiento y mutación.

La modificación Lamarckiana se puede aplicar en todos los individuos de todas las generaciones, pero esto aumenta considerablemente el coste computacional del método evolutivo, por lo que también se contemplan otras opciones como optimizar localmente los individuos de la generación T , o bien optimizar localmente los individuos élite de cada generación, o ir aumentando el número de individuos que se someten a optimización local a medida que aumente el número de generaciones, para mejorar la exploración en las primeras fases del método evolutivo, y mejorar la

explotación a medida que las generaciones tienen individuos con una aptitud más elevada, para acelerar de esta forma la convergencia.

La modificación Lamarckiana es un modo de hibridación en métodos de optimización, y se debe tener en cuenta que una codificación binaria combinada con un método matemático, y el uso de una codificación en base a números reales, necesitaría un conversor de codificación binaria a real y viceversa, por lo que se penalizaría el método evolutivo. Debido a esto, y si se piensa en utilizar este tipo de hibridación u otra, es recomendable utilizar una codificación real, ya que evita la conversión del cromosoma para su adaptación al método de optimización local.

2.3.3.2.4 *Revisión bibliográfica de los métodos evolutivos*

Con el incremento en la velocidad de procesamiento de los equipos informáticos y la reducción de tiempo de cálculo, se han generado métodos llamados evolutivos o basados en algoritmos genéticos. Estos fueron probados por primera vez por Holland en (Holland, 1973) y (Holland, 1975), cuyo trabajo es incluido en el trabajo de Goldberg (Goldberg, 1989), teniendo una aplicación exitosa en diferentes problemas de optimización. Si se centra el uso de esta técnica en la síntesis de mecanismos, Fang (Fang, 1994), y Kunjur junto a Krishnamurty (Kunjur & Krishnamurty, 1997), presentaron algunos resultados aplicando técnicas evolutivas.

La ventaja principal de estos primeros métodos evolutivos era su simplicidad de implementación, así como un bajo coste computacional, ya que sólo evaluaban error, sin necesidad de cálculo de derivadas, como puede tenerla un método de optimización local. Otra ventaja adicional, es que no es necesario conocer de forma detallada el espacio de trabajo, sabiendo si es continuo o no, si hay mínimos locales, etc., que suele ser exigido si se utilizan métodos de optimización local.

Una referencia con un gran número de citas es la proporcionada por Cabrera (Cabrera, et al., 2002). Aquí se presenta un método evolutivo, en el cual el problema a resolver es de síntesis de trayectoria, pudiéndose expresar el problema como se presenta en la Ecuación (2.30):

$$\begin{aligned} \min & \left(f(p_1(X), p_2(X), p_3(X), \dots, p_n(X)) \right) \\ \text{sujeto a: } & g_j(X) \leq 0 \quad j = 0, 1, 2, \dots, n \quad x_i \in [li_i, ls_i] \quad \forall x_i \\ & \in X \end{aligned} \tag{2.30}$$

Donde f es la función a minimizar, y hay que evaluarla con cada individuo X , para los distintos objetivos p_i , cumpliendo las restricciones g_j .

Comenzando a solucionar el problema mediante la generación de una población inicial, que generalmente es una secuencia de parámetros del mecanismo, asignando a cada uno de estos parámetros un valor de forma “aleatoria”, dentro del espacio de trabajo. Cada individuo de la generación es una solución potencial al problema.

Los genes de cada individuo pueden ser codificados de diferentes formas, en un principio se hicieron codificaciones binarias, en las que cada gen podía tomar 2 posibles valores 0 o 1. Sin embargo Cabrera utiliza una codificación en la que a cada gen se le puede asignar un valor real. Así se puede expresar un cromosoma como un conjunto de genes, agrupando los distintos reales de los genes en un vector cromosoma como el presentado en la Ecuación (2.31):

$$X = [x_1 \ x_1 \ x_1 \ x_1 \ \dots \ x_1] \ \forall x \in R \quad (2.31)$$

Con esta forma de representar cada individuo, por su cromosoma, se genera una población inicial, con un número determinado de individuos, con su cromosoma asociado. A partir de esta población inicial y utilizando operadores como selección natural, cruce, o mutación, se generarán sucesivas generaciones hasta obtener una solución al problema.

Cabrera utiliza los operadores de selección, cruce y mutación como se pasa a detallar a continuación: para la selección se debe tener en cuenta que dos individuos son elegidos con una función probabilística de la población inicial (operador de selección) y forman una pareja para la reproducción. La función de probabilidad puede tomar diferentes formas, uniforme, Gauss, etc. En este caso la distribución utilizada es la uniforme, en la que a cada individuo se le asigna una probabilidad de apareamiento directamente proporcional al valor de su aptitud, de forma que los individuos más aptos, los que mejor se adaptan a la función objetivo, tienen una mayor probabilidad de apareamiento, y con ello de reproducción. Así se puede dividir la población inicial en población élite y resto de la población.

Cabrera utiliza para la reproducción un individuo de la élite y dos individuos de la población no perteneciente a la élite, generándose el individuo resultante para la reproducción, V , como se muestra en la Ecuación (2.32):

$$\begin{aligned} X_i : i \in [1, NP] \\ V = X_{\text{élite}} + F(X_{r1} - X_{r2}) \end{aligned} \quad (2.32)$$

Donde $X_{\text{élite}}$ es el mejor individuo de la población de NP individuos, X_{r1} y X_{r2} son dos individuos elegidos aleatoriamente de la población y F es un valor real que controla la variación de los mejores individuos.

Así para el cruce V es cruzado con otro individuo aleatoriamente elegido de la población actual para generar un individuo de la nueva generación. Por lo general en esta operación los genes de los progenitores son intercambiados para generar su descendencia (utilizando el operador de cruzamiento de punto fijo anteriormente explicado o el operador de cruzamiento uniforme). El operador de cruce en el desarrollo de Cabrera puede tomar 2 alternativas: de un lado grupos de genes del cromosoma pueden ser elegidos aleatoriamente de V y del individuo elegido de la población generando con estos grupos de genes el cromosoma del individuo hijo, o bien se pueden elegir aleatoriamente los genes a tener en cuenta del individuo elegido y completar el cromosoma del individuo hijo con los genes de V .

En caso que el individuo resultante de la reproducción, hijo, sea mejor que su ancestro pasará a formar parte de la nueva población, Si no es así el individuo hijo será desechado para la nueva población, manteniéndose el ancestro, de forma que el número de individuos que forman la población no crece ni decrece.

Finalmente Cabrera considera el operador de mutación dentro de su trabajo como un cambio aleatorio en uno de los genes del cromosoma del individuo que se genera en la reproducción. Es decir, algunos individuos de la reproducción son afectados por mutación. Por lo general el número de individuos que sufre mutación es mucho menor del que no la sufre, pero la mutación es necesaria para evitar que la solución converja hacia un mínimo local.

Por otro lado también es importante la definición de la función objetivo. Cabrera la define en función de los cuadrados de la distancia euclídea entre la trayectoria objetivo y la generada en los puntos de interés, y tiene

en cuenta un segundo parámetro para comprobar si el mecanismo cumple las condiciones de Grashof, y el orden adecuado. Todo esto lo engloba en una función error en la que pondera estos 3 factores con diferentes coeficientes, penalizando a los individuos que no cumplen las condiciones de Grashof o presentan problemas de orden.

En el caso de las condiciones de Grashof, puede considerarse correcto penalizar a los individuos que no las cumplen en el caso de que la función objetivo sea una trayectoria cerrada; sin embargo, si el objetivo es una trayectoria abierta, penalizar a mecanismos que no las cumplan es ampliar las posibilidades de no hallar el mínimo global que cumpla la función objetivo, tal y como comenta el autor en un artículo posterior (Cabrera, et al., 2006). En este artículo también propone el algoritmo POEMA (Pareto optimum evolutionary multiobjective algorithm), en el que se mantiene en cada generación una proporción adecuada entre individuos dominantes y no dominantes (élite y no élite), para preservar así la diversidad dentro de cada generación. En este artículo cambia la función objetivo añadiéndole restricciones en alguna de las variables de diseño.

Otra referencia bibliográfica interesante es la escrita por Laribi (Laribi, et al., 2004). En este trabajo se mezcla la técnica de algoritmos genéticos con lógica difusa. Los operadores de lógica difusa monitorizan la variación de los diferentes genes y van variando el espacio de trabajo (valores límites de los genes) según lo obtenido en pasos anteriores de la optimización. En concreto se controla la variación de las variables y el error durante 30 generaciones. A partir de la generación 30, y en base a cómo hayan cambiado las variables y el error durante estas generaciones se varían los límites de variación por lógica difusa. Si el error se reduce y una variable no presenta una dispersión importante en sus valores, se mantiene prácticamente constante. Esto querrá decir que para una solución óptima la variable objeto de estudio tendrá un valor similar, por lo que el rango de variación de dicha variable se puede reducir.

Sin embargo, si la variable de diseño tiene fluctuaciones importantes se deberá variar el espacio de trabajo para ampliarlo y hacer la búsqueda del óptimo más eficiente.

En los últimos años las aportaciones a este tipo de métodos son numerosas, queriéndose destacar aquí la tesis doctoral de Fernandez de Bustos (Fernández de Bustos, 2003), el trabajo de Gatti (Gatti & Mundo, 2006), el trabajo de Liu (Liu & McPhee, 2005), el trabajo de Habibi (Habibi,

et al., 2008), el trabajo de Szydlowski (Szydlowski & Nelson, 2007), los trabajos de Erkaya (Erkaya & Uzmay, 2008) (Erkaya & Uzmay, 2008), el de Starosta (Starosta, 2008), el de acharyva (Acharyva & Mandal, 2009), el de Al-Dwairi (Al-Dwairi, et al., 2010), y el de Bulatovic (Bulatovic & Dordevic, 2009).

2.3.4 Métodos híbridos

Los métodos híbridos tratan de obtener la ventaja de un reducido coste computacional, obtenido con los métodos de optimización local, y la posibilidad de encontrar un óptimo global proporcionado por los métodos globales. Ante el cómo realizar la hibridación se deben responder varias preguntas.

Dónde realizar la hibridación es una de las primeras preguntas que se pueden plantear. La forma más simple posible sería aplicar un método evolutivo con un número reducido de generaciones, para así contener el gasto computacional, y posteriormente aplicar un método de optimización global sobre el mejor individuo, o los individuos de élite. Esta no es la única posibilidad, y por ejemplo, en este trabajo se va a proponer también métodos híbridos en los que la optimización local se aplique en el estimador de error, como una mejora o adaptación del individuo al método. Esto estaría en la línea de la teoría Lamarkiana, ya que el individuo final tendría todas las capacidades posibles (cumplir con la función objetivo) que permita un método de optimización local. Esto también tiene implicaciones, ya que si se aplica sobre todos los individuos podría presentarse un problema de coste computacional excesivamente elevado, por lo que se debería responder a la siguiente pregunta.

Cuándo se quiere aplicar la hibridación, es decir a todos los individuos, sólo al individuo resultado del método global o a un porcentaje de la población. A lo largo de este trabajo se han planteado varias estrategias para responder a esta pregunta, desde a todos los individuos de la última generación, pasando por todos los individuos del método evolutivo, hasta únicamente a la élite de cada generación.

Otra cuestión que se debe responder es el **cómo**, es decir, qué tipo de método local se va a aplicar, tanto el tipo de método como el tipo de optimización. La respuesta a esta pregunta puede originar que únicamente se busque el mejor posicionamiento, giro y escalado del individuo para que

la trayectoria generada y objetivo se parezcan lo máximo posible, hasta una optimización local que incluya la morfología completa del individuo. Cuanto más complejo sea el método y tenga en cuenta un mayor número de variables, mayor conste computacional tendrá asociado.

Finalmente una última pregunta que podría ser interesante responder es el **por qué**. Así de los métodos híbridos propuestos en este trabajo son diferentes planteamientos con una justificación determinada. Así por ejemplo existe un primer método híbrido en el que se aplica una modificación del estimador de error, TRS. Este estimador de error no deja de ser una optimización local de la posición, giro y escalado del mecanismo para que se adapte lo mejor posible a la función objetivo. La causa de este método es tratar de explorar las posibles capacidades del individuo, es decir, que grado de adaptación al medio (cumplimiento de la función objetivo) puede conseguir el individuo a lo largo de su vida.

Algunas de las contribuciones en el campo de la síntesis de mecanismos son el trabajo de Lin (Lin, 2010), el de Khorshidi (Khorshidi, et al., 2011), el de Peñuñuri (Peñuñuri, et al., 2011) y un artículo presentado como parte del trabajo de esta tesis (Sedano, et al., 2012) entre otras.

2.3.5 Búsqueda Tabú

Este algoritmo propuesto por Glover (Glover, 1986) es muy similar a un método de gradiente dotado de memoria. Así el método recuerda que áreas del espacio de diseño ha visitado, e impide una re-exploración de las mismas haciendo que estas áreas sean tabú. En este método se utiliza un único individuo y se va mutando por diferentes caminos del espacio de trabajo.

2.3.6 Colonias de Hormigas

Este método está basado en el comportamiento de las hormigas cuando buscan comida. Las hormigas vagan aleatoriamente hasta encontrar la comida y una vez encontrada, en su camino de vuelta a la colonia dejan un rastro de feromonas que atrae a otras hormigas. Esta feromona se evapora y deja de tener efecto pasado cierto tiempo. Así, cuanto mayor sea la fuente de alimento, más hormigas recorrerán el mismo camino, y por lo tanto mayor será el rastro de feromonas. Por el contrario, si el

alimento es escaso, una vez se acabe, el rastro de las feromonas desaparecerá tras un plazo de tiempo reducido. Un método de este tipo fue desarrollado por Dorigo (Dorigo, 1992), y en dicho método el tamaño de la fuente de alimento está asociada con la aptitud de la zona de trabajo para encontrar un individuo óptimo. Otro trabajo reseñable en esta línea es el presentado en la tesis doctoral de McDougall (McDougall, 2008).

2.3.7 Enjambre de Partículas

Esta técnica de optimización trata de emular el comportamiento de aves o insectos en enjambre, y fue propuesta por Kennedy (Kennedy, 1997). Así, en un enjambre cada uno de los individuos vuela dentro del espacio de trabajo, tratando cada uno de los individuos del enjambre de imitar a individuos de su alrededor que obtienen mejores resultados.

Capítulo 3

“El mundo es números”

Pitágoras

3 Comparación de curvas y estimación del error.

Un aspecto importante en la síntesis dimensional de mecanismos, es la formulación de la función objetivo que debe ser capaz de permitir evaluar correctamente la diferencia entre lo deseado y lo conseguido por el mecanismo evaluado. Esta diferencia se debe traducir en un estimador de error que proporcione las mejores características al algoritmo de optimización incrementando su robustez, eficiencia, precisión, y a la vez siendo lo más general posible, es decir, que abarque la mayor cantidad de casos posibles en la síntesis dimensional de generación de trayectorias.

Un estimador de error ampliamente utilizado en la síntesis de mecanismos para la comparación de trayectorias es el de Distancia Media Cuadrática, DMC (Mean Square Distance). Sin embargo, el principal inconveniente de la DMC es que no tiene en cuenta una posible traslación, rotación o escalado (homotecia) entre la función objetivo y la conseguida por el mecanismo que se pretende evaluar. Por lo tanto, con la DMC como estimador de error un mecanismo que consiguiese la reproducción exacta de la función objetivo podría tener asociado un error muy superior a otro que esté posicionado correctamente y que sin embargo no reproduce exactamente esta función.

Por tanto está claro que debe mejorarse la estimación de error entre la curva deseada y la generada por el mecanismo propuesto. Para ello se deben cumplir una serie de requisitos ya expuestos por Sánchez Martín en su Tesis doctoral (Sánchez Marín, 2000) y que reproducimos aquí algunas de ellas, ya que fundamentan esta parte de este trabajo de tesis doctoral. Estos requisitos son los siguientes:

1. Identificación de curvas exactas: Un estimador de error en la síntesis de generación de trayectorias debe identificar la forma de las dos curvas y no las coordenadas empleadas en su descripción. Las coordenadas son una herramienta que nos permiten su identificación pero no deben ser el fin en sí mismo, como sucede en otros procedimientos de estimación de error (p. ej. la DMC).
2. Progresividad: Un estimador de error debe ser progresivo e ir disminuyendo de la forma más aproximadamente lineal a medida que la trayectoria generada se acerque a la trayectoria deseada. El concepto de "linealidad" aplicado a la progresividad no deja de ser algo subjetivo, ya que el error debe representarse con respecto a un número muy amplio de variables entre los cuales no se puede garantizar la linealidad. Sin embargo, aunque se maneje este

concepto de forma cualitativa, debe tratarse en la medida de lo posible alcanzar dicho comportamiento.

3. Conmutabilidad: Este requisito se refiere a que el resultado del estimador de error debe ser siempre el mismo independientemente de que se compare la curva deseada con la generada o la generada con la deseada.
4. Independencia de la posición: Si dos curvas son iguales pero están trasladadas la una con respecto a la otra, el estimador de error debe evaluarlas como idénticas. Igualmente, aunque las curvas no sean iguales el estimador de error debe proporcionar un valor independiente del sistema de referencia empleado. Este valor además debe cumplir con el requisito de progresividad mencionado anteriormente.
5. Independencia de la orientación: Si dos curvas son iguales pero presentan distinta orientación, es decir una esta rotada con respecto a la otra, el estimador de error debe evaluarlas como idénticas. De la misma forma, aunque las curvas no sean exactamente idénticas el estimador de error debe evaluarlas en función del requisito de progresividad.
6. Independencia del escalado (homotecia): Si la diferencia entre las dos curvas es debido únicamente a un factor de escala aplicado sobre sus coordenadas el estimador de error debe identificarlas como idénticas. Igualmente debe cumplirse el principio de progresividad cuando las dos curvas no sean exactamente idénticas.

La condición 1 es el aspecto fundamental del estimador de error. Es decir, identificar y evaluar la forma de la curva y poder compararla con otra debe ser prioritario en la estimación del error. La condición 2 permite al algoritmo evolucionar de forma progresiva hacia su objetivo, proporcionando una valoración proporcional a la distancia que le separa de la solución óptima. Si este requisito de proporcionalidad no se da el algoritmo perdería robustez y eficiencia. Por otro lado, la conmutabilidad garantiza la misma valoración del error independientemente de cuál sea la curva generada y deseada. La condición de la independencia de la posición es debida a que cualquier mecanismo que no tenga restricciones suplementarias puede trasladarse en el plano reduciendo el error entre su trayectoria generada y la deseada. De la misma manera, la condición de la independencia de la orientación establece que el mecanismo puede rotar para adecuarse a la orientación de la curva deseada. Finalmente, la independencia del

escalado permite al mecanismo escalar sus dimensiones, empleando un factor de escala, para aproximar su trayectoria generada a la trayectoria deseada.

Las tres últimas condiciones hacen que las variables de diseño puedan clasificarse a su vez en tres grupos:

1. Variables de translación: son aquellas que su modificación sólo producen una translación del mecanismo.
2. Variables de orientación: son aquellas variables del mecanismo que su modificación sólo producen la rotación del mismo.
3. Variables de escalado: Son aquellas variables cuya modificación produce una variación proporcional en las dimensiones del mecanismo y como consecuencia el escalado de la trayectoria generada.

Como se verá más adelante, las variables de traslación y orientación son variables geométricas del propio mecanismo, mientras que las variables de escalado son variables adicionales que modifican proporcionalmente las dimensiones.

Debido a lo expuesto en el párrafo anterior, se ha propuesto un nuevo estimador de error, TRS, que considera la posible traslación, rotación y/o escalado del mecanismo evaluado, y que se describirá en detalle en el siguiente punto.

3.1 Estimador de error propuesto: TRS

Como ya se ha dicho anteriormente la falta de un estimador de error adecuado en una rutina de síntesis de generación de trayectorias, puede conllevar a que se evalúen erróneamente mecanismos con bajo potencial para conseguir una buena solución como adecuados. Por el contrario se puede asignar un valor de aptitud bajo a mecanismos con gran potencial para conseguir alcanzar la función objetivo.

El valor del error de síntesis proporcionado por la DMC utilizada como estimador de error depende de la posición relativa entre la trayectoria objetivo (deseada) y la trayectoria generada por el mecanismo en estudio. El uso de la DMC puede proporcionar por ejemplo un valor de error elevado ante dos trayectorias iguales pero desplazadas una respecto a la otra. Al igual que de la posición relativa en desplazamiento, este estimador de error también tiene dependencia con el giro y la escala entre ambas trayectorias. Por lo tanto se considera interesante conseguir un estimador de error que consiga el error mínimo entre la trayectoria generada y deseada, mediante la traslación giro y escalado del mecanismo.

A todas estas características se debe sumar el coste computacional de evaluar el error. La evaluación del error en problemas de optimización, ya sea mediante métodos deterministas o probabilísticos, debe realizarse un número elevado de veces por iteración. Por tanto, un coste computacional elevado de la evaluación del error puede conllevar una pérdida importante de la eficiencia del algoritmo. Es importante, por tanto, no utilizar métodos multivariable que además supongan un gasto computacional elevado.

Debido a esto, este trabajo propone un estimador de error, el estimador TRS (Traslación-Rotación-eScalado), en el que se evaluará el error en tres pasos.

1. Traslación: en un primer paso se traslada el centro de gravedad de la trayectoria generada, para hacerlo coincidir con el centro de gravedad de la trayectoria deseada.
2. Rotación: se reduce el error entre las curvas mediante la rotación relativa.
3. Escalado: se aplica un coeficiente de escala para aproximar la curva generada a la curva deseada.

En función de la disposición de ambas curvas deseada y generada se pueden dar varios casos:

1. Trayectorias cerradas. En este caso tanto la curva generada como la deseada son cerradas (ver Fig. 3.1) y por lo general se establecen para que ambas tengan el mismo número de puntos de precisión ($p=c$, siendo p el número de puntos de precisión que describen la trayectoria deseada y c el número de puntos de precisión de la trayectoria generada), ya que esto mejora la eficacia del algoritmo.
2. Trayectorias abierta-abierta. En este caso ambas curvas son abiertas (ver Fig.3.1) pero se cumple la condición $c \geq p$. Es decir, la curva generada puede contener un mayor número de puntos de precisión que la curva deseada. De esta forma se puede seleccionar el tramo de la curva generada que mejor se adapta a la deseada.
3. Trayectoria deseada abierta - generada cerrada. La curva deseada es abierta y la generada es cerrada. Puede darse esta situación cuando el mecanismo generado permite la rotación completa del elemento de entrada. En este caso también se cumple la condición $c \geq p$. Esta condición, al igual que en el caso anterior, permite seleccionar el tramo de la curva cerrada que reduce el error de síntesis.

A continuación se describe la obtención del estimador de error para cada uno de estos casos.

3.1.1 Trayectorias generada y deseada cerradas.

Se definen las trayectorias deseada y generada tal y como se muestra en la Fig. 3.1. En ambos casos el número de puntos de precisión es el mismo ($p=c$) y por simplificar se denotan ambos como p . En el caso de la trayectoria deseada los puntos de precisión se denotan como d^i y se emplean coordenadas cartesianas para su definición. La curva del mecanismo generado mediante el algoritmo evolutivo (individuo) o mediante optimización local se define por los puntos de precisión g^i . En forma matricial se puede expresar para cada curva los puntos de precisión como,

$$\begin{aligned} \mathbf{d}^{iT} &= [x_d^i \ y_d^i] \\ \mathbf{g}^{iT} &= [x_g^i \ y_g^i] \end{aligned} \left\} i = 1, 2, \dots, p \quad (3.1)$$

La mayoría de los algoritmos empleados en síntesis dimensional de mecanismos emplean la distancia Euclídea como función objetivo. En este trabajo también se parte de dicha expresión, pero se van a introducir modificaciones que permitan cumplir con los requisitos previamente

mencionados, para conseguir así el estimador propuesto, TRS. Para ello se emplea la fórmula mostrada en la Ecuación (3.2).

$$E = \frac{1}{2} \sum_{i=1}^p \left[(x_g^i - x_d^i)^2 + (y_g^i - y_d^i)^2 \right] = \frac{1}{2} \sum_{i=1}^p \left[(\mathbf{g}^i - \mathbf{d}^i)^T (\mathbf{g}^i - \mathbf{d}^i) \right] \quad (3.2)$$

Esta función objetivo depende de la posición relativa de las dos curvas. Por ejemplo, en el caso de la Fig. 3.1 el error dado por la Ecuación (3.2) puede disminuirse o aumentarse notablemente si la curva generada se desplaza hacia la curva deseada o se aleja de ella. A efectos prácticos sólo implicaría la traslación del mecanismo, lo que en general no es ningún problema desde el punto de vista del diseño de estos sistemas. Por tanto la Ecuación (3.2) puede proporcionar un error elevado cuando realmente no es así en función de la distancia relativa de las dos curvas. En consecuencia es conveniente trasladar la curva para que el estimador proporcione el menor error posible. Para realizar el traslado de la curva generada hacia la deseada es necesario seleccionar un punto de referencia que reduzca dicho error entre ambas curvas. Un punto ideal para conseguir este objetivo es el centroide de ambas curvas, siendo este el punto utilizado en el TRS. Las coordenadas del centroide pueden ser determinadas fácilmente como se muestra en la Ecuación (3.3)

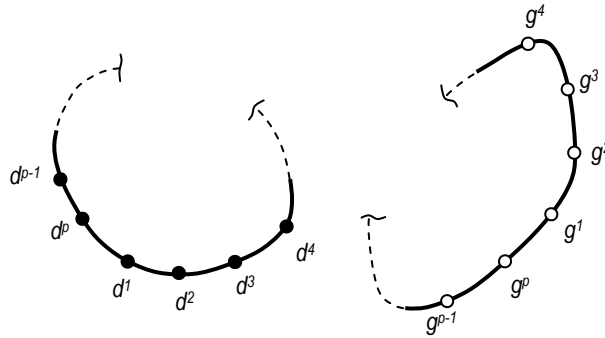


Fig. 3.1 Trayectoria deseada (d) y generada (g).

$$\begin{aligned} x_{cd} &= \frac{\sum_{i=1}^p x_d^i}{p}, & y_{cd} &= \frac{\sum_{i=1}^p y_d^i}{p}, & \mathbf{d}_c &= \frac{1}{p} \sum_{i=1}^p \mathbf{d}^i \\ x_{cg} &= \frac{\sum_{i=1}^p x_g^i}{p}, & y_{cg} &= \frac{\sum_{i=1}^p y_g^i}{p}, & \mathbf{g}_c &= \frac{1}{p} \sum_{i=1}^p \mathbf{g}^i \end{aligned} \quad (3.3)$$

Realizando a continuación un cambio de coordenadas para llevar ambas curvas a un origen común como se muestra en la Ecuación (3.4)

$$\begin{aligned} \left. \begin{aligned} x_{0d}^i &= x_d^i - x_{cd} \\ y_{0d}^i &= y_d^i - y_{cd} \end{aligned} \right\} \mathbf{d}_0^i &= \mathbf{d}^i - \mathbf{d}_c \\ \left. \begin{aligned} x_{0g}^i &= x_g^i - x_{cg} \\ y_{0g}^i &= y_g^i - y_{cg} \end{aligned} \right\} \mathbf{g}_0^i &= \mathbf{g}^i - \mathbf{g}_c \end{aligned} \quad (3.4)$$

La Fig. 3.2 muestra la traslación de ambas curvas. La estimación del error puede formularse ahora como,

$$E_0 = \frac{1}{2} \sum_{i=1}^p \left[(x_{0g}^i - x_{0d}^i)^2 + (y_{0g}^i - y_{0d}^i)^2 \right] = \frac{1}{2} \sum_{i=1}^p \left[(\mathbf{g}_0^i - \mathbf{d}_0^i)^T (\mathbf{g}_0^i - \mathbf{d}_0^i) \right] \quad (3.5)$$

Evidentemente la Ecuación (3.5) reduce el error y por tanto es más precisa que la Ecuación (3.4) empleada habitualmente. Sin embargo todavía depende del orden escogido para la numeración de los puntos de precisión. En otras palabras, la fórmula de la Ecuación (3.5) proporciona la comparación de puntos de precisión con el mismo superíndice, lo cual es una selección arbitraria realizada previamente por el diseñador. Nada impide que se pueda obtener un error más reducido cuando se cambia esta numeración. Evidentemente esto conduce a seleccionar p combinaciones distintas de error que ordenados en forma matricial resultan como se muestra en la Ecuación (3.6),

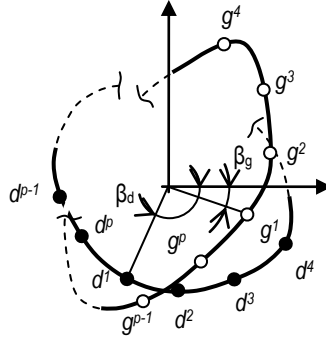


Fig. 3.2 Trayectoria deseada (d) y generada (g) trasladadas a un origen común.

$$\mathbf{E} = \begin{bmatrix} \mathbf{g}_0^1 - \mathbf{d}_0^1 & \mathbf{g}_0^2 - \mathbf{d}_0^1 & \dots & \mathbf{g}_0^{p-1} - \mathbf{d}_0^1 & \mathbf{g}_0^p - \mathbf{d}_0^1 \\ \mathbf{g}_0^2 - \mathbf{d}_0^2 & \mathbf{g}_0^3 - \mathbf{d}_0^2 & \dots & \mathbf{g}_0^p - \mathbf{d}_0^2 & \mathbf{g}_0^1 - \mathbf{d}_0^2 \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{g}_0^{p-1} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^p - \mathbf{d}_0^{p-1} & \dots & \mathbf{g}_0^{p-3} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^{p-2} - \mathbf{d}_0^{p-1} \\ \mathbf{g}_0^p - \mathbf{d}_0^p & \mathbf{g}_0^1 - \mathbf{d}_0^p & \dots & \mathbf{g}_0^{p-2} - \mathbf{d}_0^p & \mathbf{g}_0^{p-1} - \mathbf{d}_0^p \end{bmatrix}_{p \times p} \quad (3.6)$$

Cada columna de la Ecuación (3.6) representa las diferentes combinaciones para la estimación del error. Cada una de estas columnas da lugar a un estimador del error con la siguiente expresión mostrada en la Ecuación (3.7).

$$\begin{aligned} F_1 &= \sum_{i=1}^p (\mathbf{g}_0^i - \mathbf{d}_0^i)^2 \\ F_2 &= \sum_{i=1}^{p-1} (\mathbf{g}_0^{i+1} - \mathbf{d}_0^i)^2 + (\mathbf{g}_0^1 - \mathbf{d}_0^p)^2 \\ F_3 &= \sum_{i=1}^{p-2} (\mathbf{g}_0^{i+2} - \mathbf{d}_0^i)^2 + \sum_{i=p-1}^p (\mathbf{g}_0^{i-p+3} - \mathbf{d}_0^i)^2 \\ F_4 &= \sum_{i=1}^{p-3} (\mathbf{g}_0^{i+3} - \mathbf{d}_0^i)^2 + \sum_{i=p-2}^p (\mathbf{g}_0^{i-p+4} - \mathbf{d}_0^i)^2 \\ &\dots \\ F_p &= \sum_{i=1}^1 (\mathbf{g}_0^{i+p-1} - \mathbf{d}_0^i)^2 + \sum_{i=2}^p (\mathbf{g}_0^{i-1} - \mathbf{d}_0^i)^2; \end{aligned} \quad (3.7)$$

Que pueden resumirse en una única expresión como se muestra en la Ecuación (3.8),

$$F_j = \sum_{i=1}^{p-j+1} (\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i)^2 + \sum_{i=p-j+2}^p (\mathbf{g}_0^{i+j-p-1} - \mathbf{d}_0^i)^2; \quad j = 1, 2, \dots, p \quad (3.8)$$

Donde j representa la referencia al estimador de forma que se puede reescribir como se muestra en la Ecuación (3.9),

$$\mathbf{F}^T = [\mathbf{F}_1 \quad \mathbf{F}_2 \quad \dots \quad \mathbf{F}_p] \quad (3.9)$$

El vector de la Ecuación (3.9) contiene los p estimadores de error pero solo el que presenta el valor mínimo será seleccionado para la función objetivo, como se muestra en la Ecuación (3.10)

$$F_m = \min(\mathbf{F}) \quad (3.10)$$

3.1.2 Trayectorias abiertas:

En este nuevo caso se definen las trayectorias deseada y generada tal y como se muestra en la Fig. 3.3. Ahora en general el número de puntos de precisión p para la trayectoria deseada no coincide con el número de puntos de precisión c para la trayectoria generada, pero siempre cumple la condición $c \geq p$. En el caso de la trayectoria deseada los puntos de precisión se siguen denotando como $\mathbf{d}^i$ mientras que los puntos generados se denotan ahora como $\mathbf{g}^r$. Por tanto,

$$\left. \begin{aligned} \mathbf{d}^{iT} &= [x_d^i \quad y_d^i]; \quad i = 1, 2, \dots, p \\ \mathbf{g}^{rT} &= [x_g^r \quad y_g^r]; \quad r = 1, 2, \dots, c \end{aligned} \right\} \quad c \geq p \quad (3.11)$$

De forma similar al caso anterior se determina el centroide de ambas curvas y se trasladan a un origen común. Las posibles combinaciones que permiten valorar el error se expresan matricialmente como se muestra en la Ecuación (3.12),

$$\mathbf{E} = \begin{bmatrix} \mathbf{g}_0^1 - \mathbf{d}_0^1 & \mathbf{g}_0^2 - \mathbf{d}_0^1 & \dots & \mathbf{g}_0^{c-p} - \mathbf{d}_0^1 & \mathbf{g}_0^{c-p+1} - \mathbf{d}_0^1 \\ \mathbf{g}_0^2 - \mathbf{d}_0^2 & \mathbf{g}_0^3 - \mathbf{d}_0^2 & \dots & \mathbf{g}_0^{c-p+1} - \mathbf{d}_0^2 & \mathbf{g}_0^{c-p+2} - \mathbf{d}_0^2 \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{g}_0^{p-1} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^p - \mathbf{d}_0^{p-1} & \dots & \mathbf{g}_0^{c-2} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^{c-1} - \mathbf{d}_0^{p-1} \\ \mathbf{g}_0^p - \mathbf{d}_0^p & \mathbf{g}_0^{p+1} - \mathbf{d}_0^p & \dots & \mathbf{g}_0^{c-1} - \mathbf{d}_0^p & \mathbf{g}_0^c - \mathbf{d}_0^p \end{bmatrix}_{p \times (c-p+1)} \quad (3.12)$$

Cada columna de la matriz de la Ecuación (3.12) proporciona un estimador de error con la expresión mostrada en la Ecuación (3.13),

$$F_1 = \sum_{i=1}^p (\mathbf{g}_0^i - \mathbf{d}_0^i)^2 \quad (3.13)$$

$$F_2 = \sum_{i=1}^p (g_0^{i+1} - d_0^i)^2$$

$$F_3 = \sum_{i=1}^p (g_0^{i+2} - d_0^i)^2$$

...

$$F_j = \sum_{i=1}^p (g_0^{i+j-1} - d_0^i)^2; \quad j = 1, 2, \dots, c - p + 1$$

...

$$F_{c-p+1} = \sum_{i=1}^p (g_0^{i+c-p} - d_0^i)^2$$

Estas expresiones pueden resumirse como,

$$F_j = \sum_{i=1}^p (g_0^{i+j-1} - d_0^i)^2; \quad j = 1, 2, \dots, c - p + 1 \quad (3.14)$$

Igual que en el caso anterior (ver ecuación (3.10)) debe seleccionarse el estimador de error que proporciona el error mínimo.

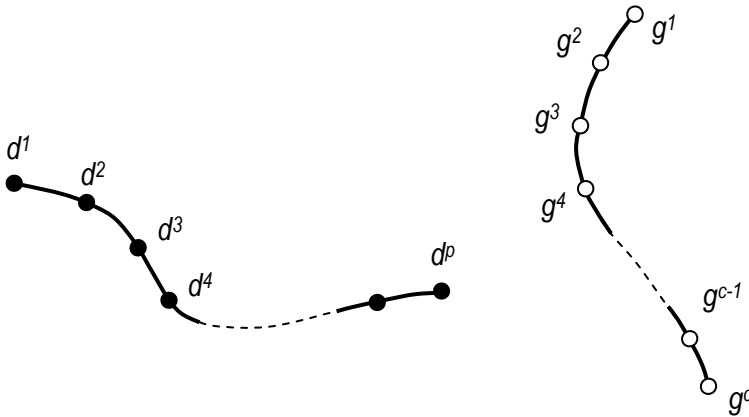


Fig. 3.3 Trayectoria deseada (d) y generada (g).

3.1.3 Trayectorias abierta (deseada) – cerrada (generada)

La Fig. 3.4 muestra las dos trayectorias. El proceso seguido es el mismo que para los casos anteriores. En primer lugar se trasladan ambas curvas

determinando el centroide y realizando un cambio de coordenadas. En segundo lugar se establece la matriz que establece las distintas combinaciones de errores, como se muestra en la Ecuación (3.15),

$$\mathbf{E} = \begin{bmatrix} \mathbf{g}_0^1 - \mathbf{d}_0^1 & \mathbf{g}_0^2 - \mathbf{d}_0^1 & \dots & \mathbf{g}_0^{c-1} - \mathbf{d}_0^1 & \mathbf{g}_0^c - \mathbf{d}_0^1 \\ \mathbf{g}_0^2 - \mathbf{d}_0^2 & \mathbf{g}_0^3 - \mathbf{d}_0^2 & \dots & \mathbf{g}_0^c - \mathbf{d}_0^2 & \mathbf{g}_0^1 - \mathbf{d}_0^2 \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{g}_0^{p-1} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^p - \mathbf{d}_0^{p-1} & \dots & \mathbf{g}_0^{p-3} - \mathbf{d}_0^{p-1} & \mathbf{g}_0^{p-2} - \mathbf{d}_0^{p-1} \\ \mathbf{g}_0^p - \mathbf{d}_0^p & \mathbf{g}_0^{p+1} - \mathbf{d}_0^p & \dots & \mathbf{g}_0^{p-2} - \mathbf{d}_0^p & \mathbf{g}_0^{p-1} - \mathbf{d}_0^p \end{bmatrix}_{p \times c} \quad (3.15)$$

Los estimadores de error obtenidos se muestran en la Ecuación (3.16),

$$\begin{aligned} F_1 &= \sum_{i=1}^p (\mathbf{g}_0^i - \mathbf{d}_0^i)^2 \\ F_2 &= \sum_{i=1}^p (\mathbf{g}_0^{i+1} - \mathbf{d}_0^i)^2 \\ &\dots \\ F_j &= \sum_{i=1}^p (\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i)^2; \quad 1 \leq j \leq c-p+1 \\ F_j &= \sum_{i=1}^{c-j+1} (\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i)^2 + \sum_{i=c-j+2}^p (\mathbf{g}_0^{i+j-c-1} - \mathbf{d}_0^i)^2; \quad c-p+1 \leq j \leq c \\ &\dots \\ F_c &= \sum_{i=1}^1 (\mathbf{g}_0^{i+c-1} - \mathbf{d}_0^i)^2 + \sum_{i=2}^p (\mathbf{g}_0^{i-1} - \mathbf{d}_0^i)^2; \end{aligned} \quad (3.16)$$

Que se pueden agrupar como se presenta en la Ecuación (3.17),

$$\left. \begin{aligned} F_j &= \sum_{i=1}^p (\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i)^2 & 1 \leq j \leq c-p+1 \\ F_j &= \sum_{i=1}^{c-j+1} (\mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i)^2 + \sum_{i=c-j+2}^p (\mathbf{g}_0^{i+j-c-1} - \mathbf{d}_0^i)^2 & c-p+1 \leq j \leq c \end{aligned} \right\} \quad (3.17)$$

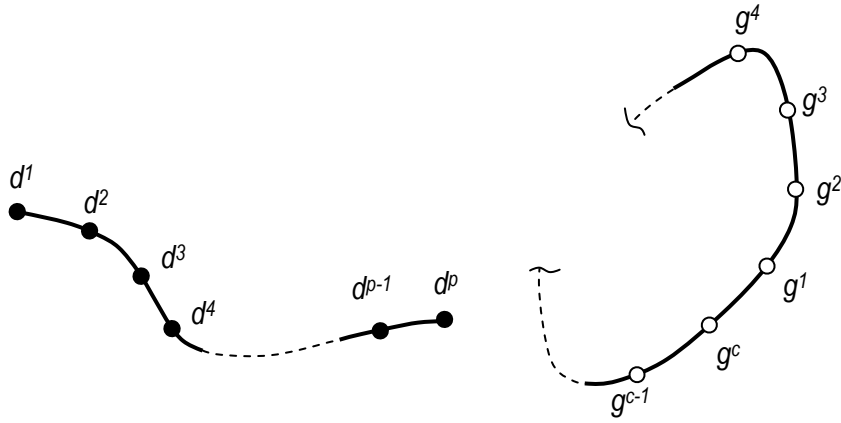


Fig. 3.4 Trayectoria deseada (d) y generada (g).

3.1.4 Rotación de curvas y función objetivo

Las Ecuaciones (3.8), (3.14) y (3.17) proporcionan una mejor comparación de las curvas con un error más ajustado que el proporcionado por la Ecuación (3.5). Sin embargo la posición relativa de ambas curvas puede modificarse mediante rotación para obtener una mejora en la estimación del error. En la Fig. 3.2 se ha establecido el primer punto de precisión de cada curva como referencia para determinar su orientación. Si los ángulos β_d y β_g se modifican mediante la rotación de una de las curvas la estimación del error también se modifica. A efectos prácticos esto supondría la rotación del mecanismo lo que no es en general un problema para el diseño. Tomando como referencia el caso de curvas cerradas, se aplica la rotación sobre la curva generada dando lugar a la modificación del estimador de error como el presentado en la Ecuación (3.18),

$$F_m(\beta_g) = \sum_{i=1}^{p-j+1} [\mathbf{A} \mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i]^2 + \sum_{i=p-j+2}^p (\mathbf{A} \mathbf{g}_0^{i+j-p-1} - \mathbf{d}_0^i)^2; \quad j = 1, 2, \dots, p \quad (3.18)$$

donde,

$$\mathbf{A}(\beta_g) = \mathbf{A} = \begin{bmatrix} \cos \beta_g & -\sin \beta_g \\ \sin \beta_g & \cos \beta_g \end{bmatrix} \quad (3.19)$$

es la matriz de rotación para proporcionar el giro de las coordenadas de los puntos de precisión. El estimador de error proporcionado por la Ecuación (3.21) es ahora la función objetivo de un problema de

optimización local con una única variable de diseño que es el ángulo β_g . La implementación numérica de la Ecuación (3.18) y la optimización conduce a una estimación del error precisa y eficiente que puede ser incorporada en algoritmos de optimización deterministas y estocásticos. La función objetivo para el caso de curvas cerrada-cerrada será,

$$F_j(\beta_g) = \sum_{i=1}^{p-j+1} [\mathbf{A} \mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i]^2 + \sum_{i=p-j+2}^p (\mathbf{A} \mathbf{g}_0^{i+j-p-1} - \mathbf{d}_0^i)^2; \quad j = 1, 2, \dots, p \quad (3.20)$$

La función objetivo para el casos de curvas abierta-abierta sería,

$$F_m(\beta_g) = \sum_{i=1}^p (\mathbf{A} \mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i)^2; \quad j = 1, 2, \dots, c - p + 1 \quad (3.21)$$

Y para el caso de curvas abierta-cerrada,

$$F_m(\beta_g) = \sum_{i=1}^p (\mathbf{A} \mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i)^2 \quad 1 \leq j \leq c - p + 1$$

$$F_m(\beta_g) = \sum_{i=1}^{c-j+1} (\mathbf{A} \mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i)^2 + \sum_{i=c-j+2}^p (\mathbf{A} \mathbf{g}_0^{i+j-c-1} - \mathbf{d}_0^i)^2 \quad c - p + 1 \leq j \leq c \quad (3.22)$$

3.1.5 Escalado

El último paso consiste en el escalado de la curva generada para adaptarla a la función deseada, con lo que el nuevo estimador de error, TRS, estaría completado. Para ello se introduce un coeficiente de escala s , que multiplica a las coordenadas de los puntos de precisión transformando la función objetivo para el caso de curvas cerradas como,

$$F_m(\beta_g, s) = \sum_{i=1}^{p-j+1} [s \mathbf{A} \mathbf{g}_0^{i+j-1} - \mathbf{d}_0^i]^2 + \sum_{i=p-j+2}^p (s \mathbf{A} \mathbf{g}_0^{i+j-p-1} - \mathbf{d}_0^i)^2; \quad j = 1, 2, \dots, p \quad (3.23)$$

Estas expresiones muestran que el problema debe resolverse como una optimización de dos variables. Sin embargo, a lo largo de este trabajo se han realizado pruebas que muestra que una optimización independiente por cada variable conlleva buenos resultados con reducciones de coste computacional. Teniendo en cuenta el gran número de iteraciones a realizar, sobre todo en una optimización mediante métodos evolutivos, la solución de realizar dos optimizaciones consecutivas, primero rotando y luego escalando es el método seleccionado.

3.1.6 Utilización de coordenadas polares

En el caso anterior se han empleado coordenadas cartesianas para el proceso de traslación-rotación-escalado, estimador TRS propuesto. Sin embargo, merece la pena investigar la influencia que puede tener la utilización de coordenadas polares frente a las coordenadas cartesianas en los procesos de rotación y escalado.

Para ello, tras la fase de traslación, se aborda la fase de giro del mecanismo. Al igual que en el proceso anterior esta fase consiste en minimizar el error cuadrático resultante de comparar la posición de los puntos de la trayectoria objetivo con la generada después de girar el mecanismo un ángulo β . Para conseguir este objetivo se dan los siguientes pasos:

1. Se transforman a coordenadas polares (módulo y ángulo) los puntos de la trayectoria ya trasladada. Se toma como origen de las coordenadas polares el centro de gravedad anteriormente determinado, tal y como se muestra en la Fig. 3.5. Se incrementa un valor arbitrario β todos los ángulos de los puntos de la trayectoria a evaluar, manteniendo el valor de sus módulos. Esto equivale a girar la trayectoria (y el mecanismo) el mismo ángulo β .
2. Tras esto, y mediante un proceso de optimización del ángulo β , para minimizar el error estructural en función del mismo, se obtiene el giro óptimo de la trayectoria obtenida, y por lo tanto también el giro del mecanismo.
3. Finalmente y como fase de escalado, se trabaja con el módulo de los vectores en coordenadas polares obtenidos anteriormente. Teniendo en cuenta estos módulos girados, se puede calcular la trayectoria obtenida con el mecanismo en función del factor de escala aplicado. Con esto se puede evaluar el error estructural como media de errores cuadráticos entre trayectoria objetivo y trayectoria generada por el mecanismo objeto de estudio, considerando nuevamente y como en el caso de giro el ángulo inicial de la barra de entrada. Optimizando este factor de escala para minimizar el error estructural, se obtiene el valor de escala a aplicar al mecanismo.

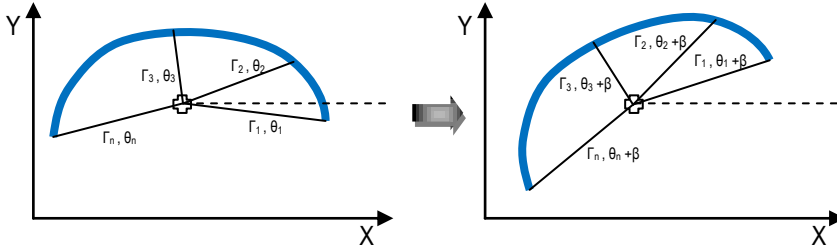


Fig. 3.5 Aproximación de giro del mecanismo y su trayectoria

3.2 Aplicación al mecanismo cuadrilátero articulado.

Una curva plana o función generada por un mecanismo depende de número de variables de diseño que lo definan. Esta curva puede ser especificada en términos de sus coordenadas cartesianas de la siguiente manera.

$$\mathbf{P}(x_g, y_g) = \mathbf{f}(\mathbf{w}_0, \mathbf{q}_{in}, \mathbf{m}, q_0) \quad (3.24)$$

donde $\mathbf{P}$ es un vector que representa las coordenadas cartesianas de los puntos de precisión obtenidos sobre la curva, $\mathbf{w}_0$ es un vector con las dimensiones geométricas de las barras y $\mathbf{q}_{in}$ son las entradas del mecanismo en cada uno de los grados de libertad. El vector $\mathbf{m}$ representa las coordenadas de un punto de referencia en el mecanismo y q_0 es la variable que modifica la orientación del bastidor. Para reducir el error entre la curva deseada y la curva generada se aplica el estimador TRS, realizando los tres pasos previamente mencionados, es decir, traslación, rotación y escalado. Esto exige dividir las variables de diseño en cuatro subconjuntos para formar un nuevo vector.

$$\mathbf{w}^T = [\mathbf{w}_t^T \ \mathbf{w}_r^T \ \mathbf{w}_s^T \ \mathbf{w}_0^T] \quad (3.25)$$

donde los subíndices t, r y s indican traslación, rotación y escalado respectivamente. Cada uno de estos pasos requiere la optimización del mecanismo con respecto al subconjunto de variables correspondientes. Con el objeto de tener mayor claridad, a partir de aquí se presenta el caso del mecanismo cuadrilátero articulado representado en la Fig. 3.6. Sin embargo, todo lo presentado en este apartado puede ser utilizado en cualquier tipo de mecanismo plano, independientemente del número barras y número y tipo de pares cinemáticos. Las variables de diseño del mecanismo de la Fig. 3.6 serán,

$$\mathbf{w}^T = [x_0 \ y_0 \ \theta_0 \ \theta_{20} \ L_1 \ L_2 \ L_3 \ L_4 \ L_5 \ \beta] \quad (3.26)$$

y las variables correspondientes a cada una de las fases del estimador de error TRS serán:

$$\begin{aligned} \mathbf{w}_t^T &= [x_0 \quad y_0] \\ \mathbf{w}_r^T &= [\theta_0] \\ \mathbf{w}_s^T &= [(s \mathbf{w}_0)^T] \end{aligned} \quad (3.27)$$

donde s representa el factor de escala. El factor de escala s es una nueva variable de diseño que multiplica todas las variables geométricas que tienen influencia sobre la escala de la curva generada pero que no modifican su forma. Es decir,

$$\mathbf{w}_0^T = [L_1 \quad L_2 \quad L_3 \quad L_4 \quad L_5] \quad (3.28)$$

De esta forma la función $\mathbf{P}$ no se ve modificada salvo en su escala.

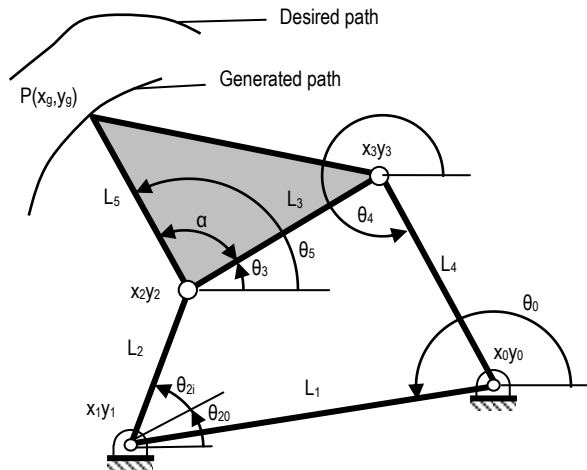


Fig. 3.6 Esquema de un cuadrilátero articulado.

Es importante resaltar que cada una de las fases de estimación del error involucra a la anterior. La experiencia obtenida de las pruebas realizadas demuestra que la inclusión de la traslación en el proceso de rotación mejora la eficiencia de este último. Del mismo modo la inclusión de la traslación y rotación en el proceso de escalado mejora el resultado obtenido.

3.3 Ejemplos

Para mostrar de forma representativa la mejora que supone el estimador de error propuesto TRS, frente al MSD se ha realizado una batería de ensayos. Los ensayos se han llevado a cabo sobre dos trayectorias objetivo, una abierta sin puntos singulares y otra cerrada con un único punto singular (gota de agua). Las trayectorias utilizadas se presentan en la Tabla 3.1 en base a las coordenadas x e y de los puntos de precisión empleados en el proceso de estimación de error.

	Arco Abierto		Gota	
<i>Punto nº</i>	<i>X1</i>	<i>Y1</i>	<i>X2</i>	<i>Y2</i>
1	10	0	7.03	5.99
2	9.89	0.52	6.95	5.45
3	9.78	1.04	6.77	5.03
4	9.46	1.54	6.4	4.6
5	9.14	2.03	5.91	4.03
6	8.61	2.49	5.43	3.56
7	8.09	2.94	4.93	2.94
8	7.39	3.33	4.67	2.6
9	6.69	3.72	4.38	2.2
10	5.85	4.02	4.04	1.67
11	5	4.33	3.76	1.22
12	4.05	4.54	3.76	1.97
13	3.09	4.76	3.76	2.78
14	2.07	4.86	3.76	3.56
15	1.05	4.97	3.76	4.34
16	0	4.97	3.76	4.91
17	-1.05	4.97	3.76	5.47
18	-2.07	4.86	3.8	5.98
19	-3.09	4.76	4.07	6.4
20	-4.05	4.54	4.53	6.75
21	-5	4.33	5.07	6.85
22	-5.95	4.12	5.5	6.84
23	-6.91	3.90	5.89	6.83
24	-7.86	3.69	6.41	6.8
25	-8.81	3.48	6.92	6.58

Tabla 3.1 Trayectorias objetivo

De forma gráfica se presentan las trayectorias de la Tabla 3.1 en la figura 3.7.

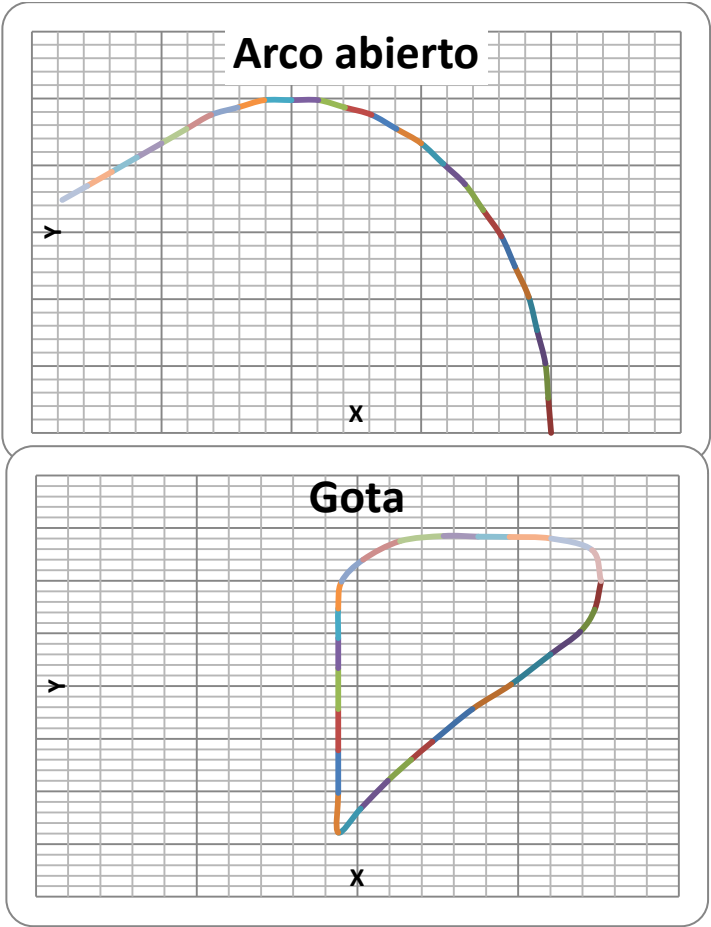


Fig. 3.7 Trayectorias objetivo

Para los dos casos (esto es, curva abierta y curva cerrada) se han generado aleatoriamente 10 (mecanismos del A al J) mecanismos a través de sus variables de diseño. En el caso del arco abierto las dimensiones de los mecanismos generados inicialmente se presentan en la Tabla 3.2.

Original	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	B
A	-6.07	5.43	-2.96	-0.64	9.18	10.92	16.08	15.58	8.32	1.82
B	6.21	3.72	1.38	1.89	5.50	12.49	14.97	8.97	15.26	1.25
C	5.82	6.28	0.44	-1.72	3.30	12.71	13.28	17.27	7.10	0.47
D	3.10	-7.43	-1.37	-3.13	11.39	13.87	15.72	13.50	3.25	-0.16
E	-2.94	2.44	1.24	1.11	3.54	11.10	5.63	12.64	5.59	-0.65
F	-3.00	-6.47	-2.00	0.99	5.56	9.33	8.82	6.85	13.84	1.73
G	7.74	4.51	1.55	-2.80	4.10	9.12	18.91	14.42	3.62	-1.65
H	-4.11	1.99	-1.39	0.83	7.79	17.22	17.04	8.05	13.24	0.51
I	-6.20	-0.82	-1.48	-1.98	5.67	13.70	17.53	10.02	6.31	-0.31
J	1.56	2.14	1.25	2.73	5.54	10.93	8.11	11.65	8.15	2.21

Tabla 3.2 Mecanismos iniciales a evaluar para la trayectoria de arco abierto

En el caso la trayectoria de gota los mecanismos generados inicialmente de forma aleatoria se presentan en la Tabla 3.3 (mecanismos del K al T).

Original	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
K	-8.27	1.05	-2.91	0.32	8.95	13.49	13.93	13.41	9.23	-0.95
L	7.45	-8.48	-0.85	1.30	9.08	17.66	16.75	22.07	12.17	-0.50
M	-0.04	5.10	2.38	-0.01	8.80	21.89	19.32	12.90	4.30	4.70
N	-1.10	0.48	2.98	-2.62	6.92	11.00	11.76	11.08	4.82	0.22
O	2.93	5.90	1.22	1.32	0.37	3.73	7.72	10.05	12.33	-1.33
P	-7.68	-7.53	-2.94	-2.88	2.30	11.20	14.09	9.25	19.34	-1.39
Q	1.56	1.21	-0.76	-2.29	3.44	7.94	9.43	10.44	1.43	-0.33
R	-6.21	0.30	1.22	0.52	4.04	9.80	9.15	5.18	5.08	1.98
S	3.10	-4.98	-1.80	2.13	5.75	9.86	8.64	9.80	8.73	-3.94
T	4.82	-9.35	-0.22	2.90	8.14	13.57	13.28	10.57	17.26	0.14

Tabla 3.3 Mecanismos iniciales a evaluar para la trayectoria gota

Trayectoria de arco abierto				Trayectoria de gota			
	TRS	DMC	Comparación		TRS	DMC	Comparación
A	324.09	1458.58	22.22%	K	40.04	5531.69	0.72%
B	14.73	2569.81	0.57%	L	32.02	1892.94	1.69%
C	319.52	1423.28	22.45%	M	48.96	2382.07	2.06%
D	319.41	5861.37	5.45%	N	40.19	1304.78	3.08%
E	4.08	1424.04	0.29%	O	18.97	2296.54	0.83%
F	178.81	1549.06	11.54%	P	6.49	1193.17	0.54%
G	159.19	2037.33	7.81%	Q	49.45	636.75	7.77%
H	195.78	871.30	22.47%	R	44.00	1694.45	2.60%
I	259.00	689.92	37.54%	S	41.37	4866.90	0.85%
J	230.55	2661.61	8.66%	T	21.37	1436.44	1.49%

Tabla 3.4 Comparación de los resultados con cada estimador

Los mecanismos presentados en la Tabla 3.2 y en la Tabla 3.3 son evaluados con el estimador de error propuesto, TRS, y con el DMC frente a las trayectorias presentadas en la Tabla 3.1. En la Tabla 3.4 se muestra esta evaluación y una comparativa porcentual entre el error obtenido con el estimador de error propuesto TRS y el DMC.

El estimador TRS involucra cambios en la definición de los mecanismos iniciales, es decir, modifica su posición inicial, lo rota y lo escala, por lo que los mecanismos tras la estimación de error han modificado tanto sus coordenadas de posición como sus dimensiones. Los resultados de la modificación se muestran en la Tabla 3.5.

En el caso del arco abierto, los las nuevas variables de diseño de los mecanismos modificados tras el estimador de error se presentan en la Tabla 3.5.

<i>Mec.</i>	x_0	y_0	ϑ_0	ϑ_{20}	L_1	L_2	L_3	L_4	L_5	B
A	-24.41	-8.25	-2.58	0.13	17.08	20.30	29.90	28.98	15.47	1.82
B	37.71	5.07	0.00	0.52	15.03	34.12	40.87	24.49	41.68	-5.03
C	14.63	11.28	-0.84	-1.71	5.77	22.22	23.21	30.19	12.42	-5.81
D	2.02	6.65	-1.47	-1.83	5.98	7.28	8.25	7.09	1.71	-0.16
E	32.94	-11.34	0.55	0.41	33.01	103.59	52.55	117.91	52.14	-0.65
F	-2.19	-2.01	-2.95	0.04	3.62	6.07	5.73	4.46	9.00	1.73
G	5.47	0.70	-2.62	-0.69	5.09	11.34	23.50	17.92	4.50	-1.65
H	4.20	-1.82	-0.01	2.21	6.40	14.14	14.00	6.61	10.87	0.51
I	-2.07	7.30	-2.14	-1.74	4.53	10.95	14.02	8.01	5.05	-0.31
J	-2.44	-0.76	2.54	-1.09	4.05	8.00	5.94	8.53	5.96	2.21

Tabla 3.5 Mecanismos resultantes tras evaluación de error para arco abierto

En el caso la trayectoria de gota los mecanismos generados tras estimador de error se presentan en la Tabla 3.6.

<i>Mec.</i>	x_0	y_0	ϑ_0	ϑ_{20}	L_1	L_2	L_3	L_4	L_5	B
K	5.89	4.94	2.28	1.49	1.10	1.65	1.71	1.64	1.13	-0.95
L	6.54	6.40	0.59	1.74	1.90	3.69	3.50	4.61	2.54	-0.50
M	5.32	4.57	2.44	1.82	0.77	1.91	1.69	1.13	0.38	4.70
N	5.49	5.38	1.83	1.76	1.53	2.43	2.60	2.45	1.07	0.22
O	24.26	0.73	0.39	0.99	0.59	5.97	12.35	16.06	19.71	4.95
P	-6.34	4.91	3.00	0.04	1.38	6.74	8.48	5.57	11.64	-1.39
Q	5.18	4.95	0.40	1.89	0.96	2.22	2.64	2.92	0.40	-0.33
R	5.62	4.27	2.71	2.02	0.68	1.65	1.54	0.87	0.86	1.98
S	4.17	5.23	-0.65	2.02	0.77	1.31	1.15	1.30	1.16	2.35
T	7.54	5.97	1.38	0.98	1.67	2.78	2.72	2.17	3.54	0.14

Tabla 3.6 Mecanismos resultantes tras evaluación de error para trayectoria gota

Estas modificaciones se han generado aplicado una traslación en x e y una variación de ϑ_0 , y de ϑ_{20} y un escalado del mecanismo, según las Tablas 3.7 y 3.8. En estas tablas se muestran en las primeras dos columnas las coordenadas de traslación. En la tercera columna se muestra el giro experimentado por el mecanismo durante la rotación. En la cuarta columna se presenta la posición inicial de la barra de entrada. Finalmente, en la

última columna se muestra la escala aplicada al acabar el proceso de aplicar el estimador TRS.

Mec.	Disp. X	Disp. Y	$\Delta\vartheta_0$	$\Delta\vartheta_{20}$	Escala
A	-18.338	-13.683	0.378	0.770	1.860
B	31.498	1.356	-1.380	-1.380	2.731
C	8.818	4.994	-1.287	0.017	1.748
D	-1.086	14.084	-0.092	1.300	0.525
E	35.887	-13.773	-0.697	-0.697	9.330
F	0.816	4.457	-0.953	-0.953	0.650
G	-2.270	-3.808	-4.169	2.114	1.243
H	8.309	-3.809	1.384	1.384	0.821
I	4.126	8.118	-0.657	0.243	0.799
J	-4.003	-2.901	1.283	-3.815	0.732

Tabla 3.7 Variables modificadas en mecanismos evaluados para trayectoria de arco abierto

Mec.	Disp. X	Disp. Y	$\Delta\vartheta_0$	$\Delta\vartheta_{20}$	Escala
K	14.159	3.886	5.194	1.172	0.123
L	-0.916	14.879	1.446	0.441	0.209
M	5.363	-0.525	0.066	1.826	0.087
N	6.584	4.903	-1.146	4.383	0.221
O	21.331	-5.161	-0.829	-0.326	1.599
P	1.335	12.437	5.939	2.923	0.602
Q	3.617	3.734	1.156	4.172	0.279
R	11.824	3.975	1.496	1.496	0.169
S	1.067	10.204	1.150	-0.107	0.133
T	2.714	15.315	1.596	-1.923	0.205

Tabla 3.8 Variables modificadas en mecanismos evaluados para trayectoria de gota

En la Fig. 3.8 y la Fig. 3.9 se presentan gráficos en los que en rojo se dibujan las trayectorias objetivo descritas en la Tabla 3.1. Los mecanismos dibujados con el acoplador de color verde, son los mecanismos originales descritos en la Tabla 3.2 y la Tabla 3.3, cuyas trayectorias se dibujan en azul. Los mecanismos modificados tras aplicar el estimador de error TRS se presentan con el acoplador en color amarillo, estando estos mecanismos descritos en la Tabla 3.5 y Tabla 3.6. La trayectoria generada por los mecanismos modificados por el estimador de error se muestra en negro.

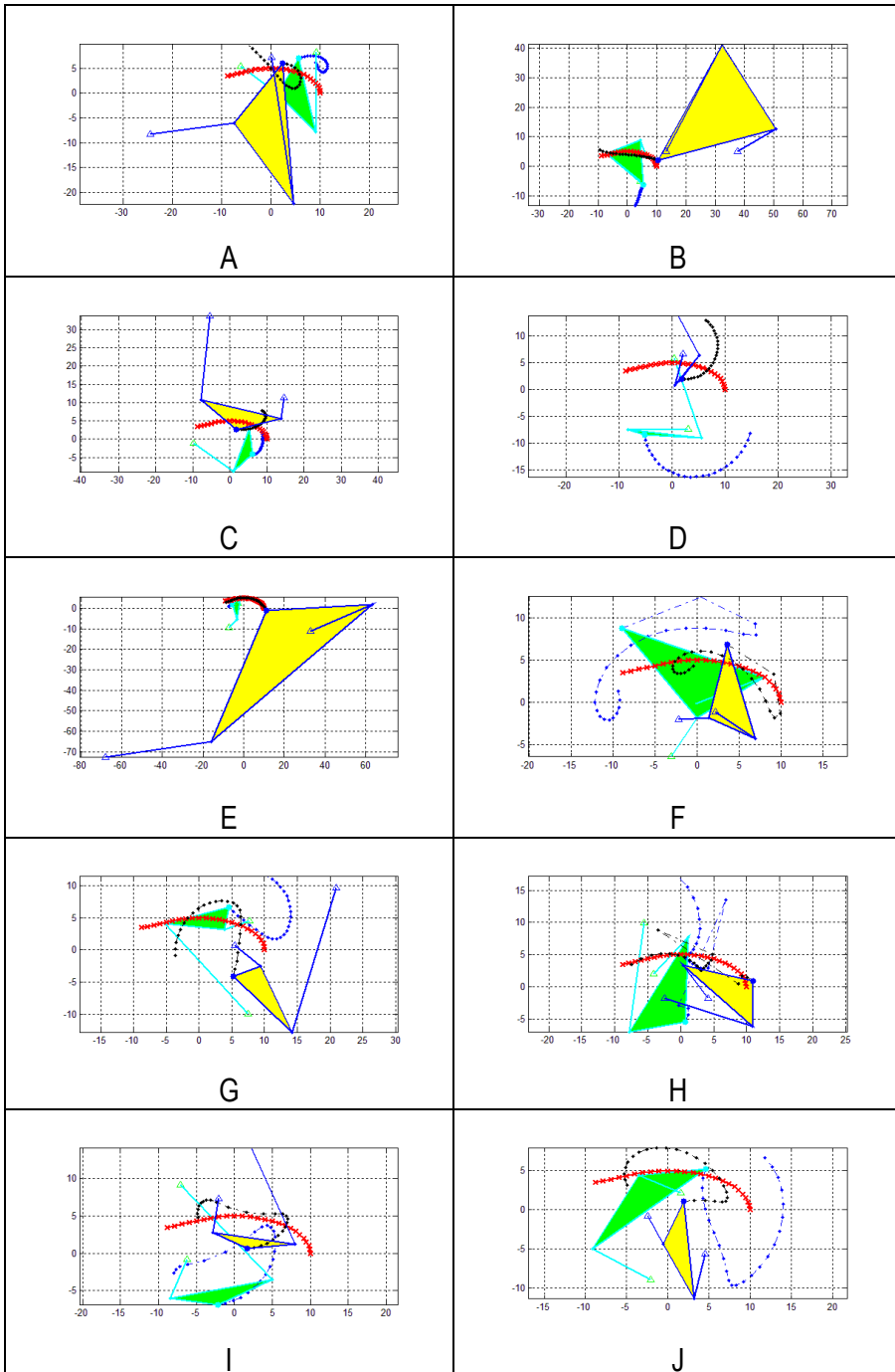


Fig. 3.8 Arco abierto (objetivo en rojo), generada por mecanismo (seguidor verde) original (azul) y por mecanismo tras estimador de error propuesto (negro)

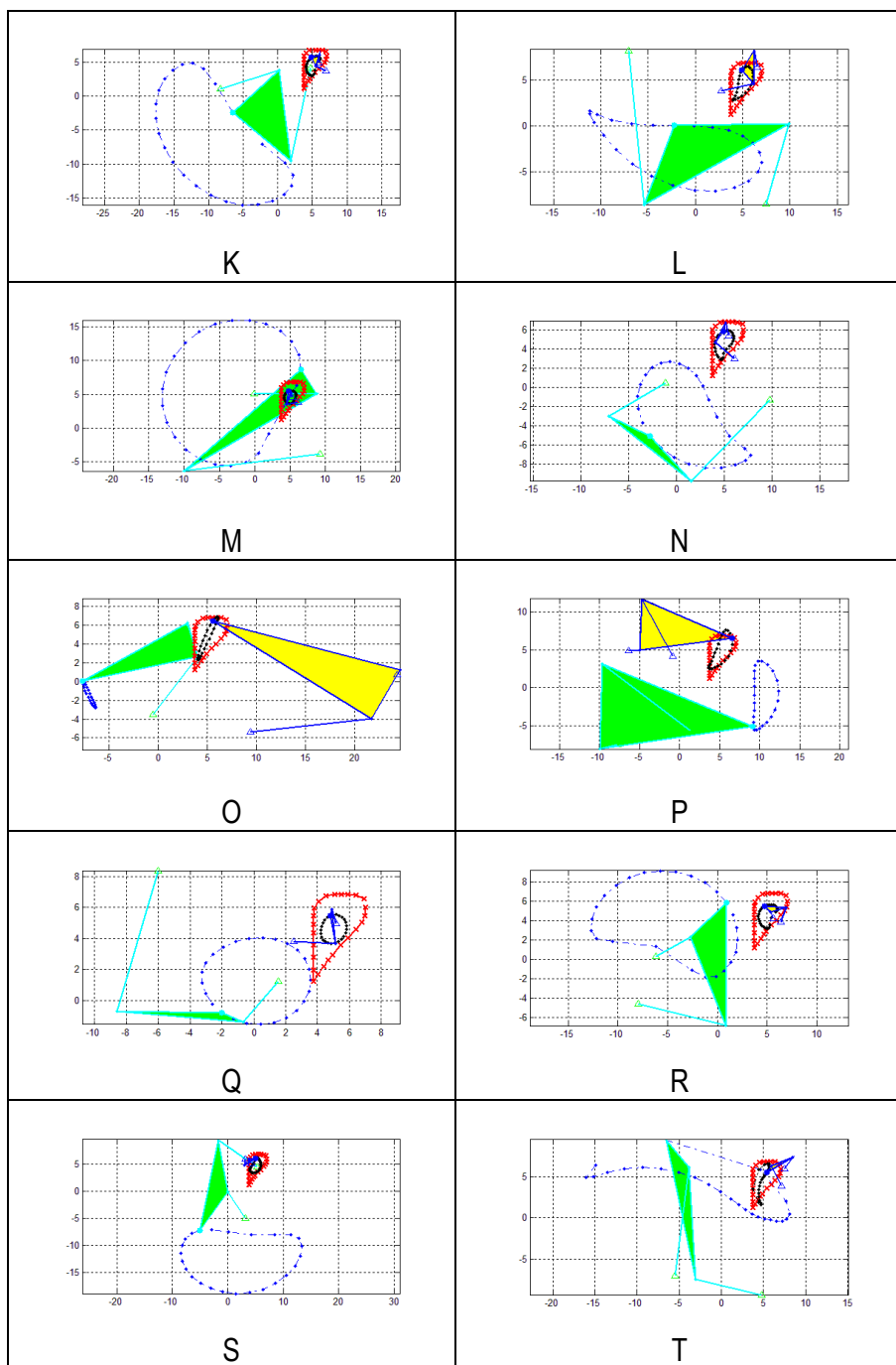


Fig. 3.9 Trayectoria gota (objetivo en rojo), generada por mecanismo (seguidor verde) original (azul) y por mecanismo tras estimador (seguidor amarillo) de error propuesto (negro)

Como se puede apreciar en las diferentes figuras de la Fig. 3.8 y la Fig. 3.9, los mecanismos tras aplicar el estimador de error TRS son los originales trasladados, girados y escalados. Aunque las dimensiones de los mecanismos al finalizar el proceso difieren de las dimensiones originales no se modifican la relación de longitudes entre las barras del mecanismo. En otras palabras, el cambio de dimensiones es únicamente debido a la aplicación de la escala mostrada en la última columna de las Tablas 3.7 y 3.8. De la misma forma, se puede apreciar que mecanismos como el “E” en la Fig. 3.8, o el “P” en la Fig. 3.9, cumplen prácticamente con la curva objetivo al aplicarse el estimador de error TRS, tal y como indica el error mostrado en la Tabla 3.4. Sin embargo, el error según DMC es muy superior, y conlleva clasificar el mismo mecanismo (sin trasladar, rotar y escalar) como un mecanismo que no cumple con el objetivo, es decir, cuya aptitud es baja.

3.4 Conclusiones

El estimador de error propuesto, TRS, cumple con las condiciones descritas al comienzo de este capítulo. Este estimador de error, no conlleva cambios entre la relación de longitudes de las diferentes barras que lo componen, es decir, no cambian la morfología del mecanismo, pero sí en su posición, giro y tamaño. Sin embargo, la evaluación de la aptitud del mecanismo, cuantificando la validez del mismo para cumplir con la trayectoria objetivo es mucho más precisa, no despreciando mecanismos que podrían cumplir con el objetivo simplemente con trasladarlos, rotarlos y/o aplicarles un factor de escala diferente.

El estimador propuesto tiene un coste computacional muy reducido, por lo que puede ser utilizado de forma intensiva. Esta propiedad permite su uso en métodos evolutivos, en los que la aplicación intensiva del estimador de error haría inviable el uso de estimadores de error con un coste computacional medio o elevado.

El estimador de error TRS, permite en el caso de un proceso de síntesis local, partir de un mecanismo cuyo potencial para cumplir con la función objetivo sea muy superior a los estimadores de error utilizados habitualmente. Teniendo en cuenta que en el estimador TRS el traslado, giro y escalado del mecanismo inicial es secuencial y no aditivo (el ajuste de la rotación del mecanismo no implica modificaciones en su traslación, por ejemplo), es posible vislumbrar que en un proceso de síntesis por métodos locales podría ser recomendable utilizar una estrategia por fases, en la que en una primera fase se optimicen las variables de diseño que involucren traslación del mecanismo; en una segunda fase las que involucren la traslación y giro del mecanismo; en una tercera fase la traslación, giro y escalado del mecanismo; y finalmente en una cuarta fase todas las variables de diseño.

Capítulo 4

“El verdadero problema es que los programadores han pasado demasiado tiempo preocupándose por la eficiencia en el lugar equivocado y en el momento equivocado; la optimización prematura es la raíz de todo mal (o al menos de la mayor parte de él) en la programación.”

Donald Knuth

4 Síntesis óptima: síntesis de mecanismos basada en métodos de optimización local.

El estimador de error propuesto, TRS, descrito en el capítulo 3, podría aplicarse, como allí se ha comentado, tanto en técnicas de optimización local, como en algoritmos evolutivos.

Aplicar el estimador de error explicado en un método de optimización local implicaría conseguir una mejor aptitud del individuo de forma previa a abordar el problema de optimización local como tal, es decir, tomando una solución inicial de partida, el estimador TRS, permitiría realizar operaciones de traslado, rotación y escalado del mecanismo para que la trayectoria generada por el mismo sea lo más próxima posible a la trayectoria deseada o trayectoria objetivo, así como el punto de inicio y fin de la trayectoria generada para aprovechar al máximo la aptitud del mecanismo de partida.

Una vez modificado el mecanismo de partida como se ha explicado, se comenzaría el proceso de optimización local. Durante la optimización local el estimador de error utilizado es el de distancia media cuadrática, DMC. El estimador de distancia media cuadrática no tiene en cuenta en sí mismo procesos de traslación, rotación y homotecia de las trayectorias para su comparación. Sin embargo, abordando un proceso de optimización local por fases se pueden tener en cuenta estos procesos de forma incremental.

En la optimización local por fases se plantea una primera fase que consiste en el traslado de la trayectoria generada, lo que implica modificar las variables $\mathbf{w}_t$, descritas en el capítulo 3.

En una segunda fase se aborda una optimización de traslación y rotación de forma conjunta, por lo que las variables de diseño a modificar son $\mathbf{w}_t$ y $\mathbf{w}_r$.

En una tercera fase se aborda una optimización en la traslación, rotación y escalado de la trayectoria generada, por lo que el grupo de variables de diseño a optimizar se incrementaría con $\mathbf{w}_s$, con lo que se recogerían en estas tres primeras fases todas las variables correspondientes a las fases del estimador de error, pero de forma incremental y conjunta, lo que

permite obtener mejores resultados, a cambio de un coste computacional más elevado que el estimador TRS.

Finalmente y como una cuarta fase de la optimización local se tienen en cuenta las variables de diseño correspondientes al estimador de error, así como las variables geométricas, por lo que se aborda una optimización de todas las variables de diseño de forma conjunta.

En los siguientes puntos se expondrá el proceso de síntesis local desarrollado en este trabajo con mayor detalle, abordando el planteamiento cinemático del mecanismo, la definición de la función objetivo y el proceso de optimización en sí mismo.

4.1 Definición cinemática del mecanismo

El problema de síntesis óptima dimensional trata de obtener un mecanismo que genere una trayectoria con la mayor similitud posible a una trayectoria deseada. El mecanismo generado se debe representar mediante un vector de parámetros que permitan definir el tamaño, posición y orientación de cada uno de los elementos que son parte de dicho mecanismo. Este vector o conjunto de parámetros son las variables de diseño, y tal y como se ha descrito con anterioridad son representadas mediante el vector $\mathbf{w}$, que en el caso de un cuadrilátero articulado puede estar construido como se muestra en la Ecuación (4.1).

$$\mathbf{w}^T = [x_0 \ y_0 \ \theta_1 \ \theta_{20} \ L_1 \ L_2 \ L_3 \ L_4 \ L_5 \ \beta] \quad (4.1)$$

Además de las variables de diseño, que definen el mecanismo, el estudio de la cinemática del mismo implica conocer las posiciones de sus distintos elementos a lo largo de la trayectoria generada. Este conjunto de posiciones se recogen en un vector $\mathbf{q}$, denominado coordenadas generalizadas.

Las coordenadas generalizadas pueden expresarse con un conjunto de parámetros diferentes, es decir, existen distintos tipos de coordenadas generalizadas en función del conjunto de parámetros que contenga el vector $\mathbf{q}$, así como de si los parámetros que están contenidos en el mismo son independientes o dependientes entre sí. Debido a esto, la cinemática del mecanismo se puede expresar con diferentes tipos de coordenadas generalizadas, siendo una decisión no trivial el tipo de coordenadas a

utilizar, ya que esta elección influye en la eficiencia e implementación de la formulación de la cinemática del mecanismo.

Si se selecciona una representación en base a coordenadas independientes, se tiene la ventaja de que el número de coordenadas a utilizar es mínimo y coincidente con el número de grados de libertad de mecanismo. Por el contrario, en mecanismos de cadena cerrada, con las coordenadas independientes puede no describirse de forma unívoca la posición del mecanismo, por lo que su uso queda limitado.

La utilización de coordenadas dependientes implica un mayor número de coordenadas a utilizar, pero sin embargo permiten definir de forma unívoca la posición de todos los elementos del mecanismo, así como una formulación más sencilla y generalizable a diferentes cadenas cinemáticas. Debido a estas ventajas, este ha sido el tipo de coordenadas generalizadas elegido para el desarrollo de este trabajo. Por otro lado se debe considerar que las variables dependientes elegidas para la descripción de la cinemática del problema deben aportar información directa sobre parámetros de interés para el diseño, siendo responsabilidad del diseñador no añadir variables dependientes que no aporten información de interés para el problema de síntesis.

Las ecuaciones de restricción cinemática definen el comportamiento cinemático del mecanismo. En la definición descrita por estas ecuaciones, se relacionan las variables independientes con las dependientes y las dependientes entre sí. El número de ecuaciones a generar para describir el comportamiento del mecanismo depende de los grados de libertad del mismo, así como de las coordenadas generalizadas seleccionadas por el diseñador. Así el número de ecuaciones de restricción necesarias es la diferencia del número de coordenadas generalizadas y los grados de libertad del mecanismo. Estas ecuaciones están agrupadas dentro de un vector Φ , pudiéndose escribir de forma compacta como se muestra en la Ecuación (4.2).

$$\Phi(\mathbf{w}^T, \mathbf{q}) = 0 \quad (4.2)$$

Las coordenadas utilizadas en este trabajo son las coordenadas naturales del mecanismo, longitudes de barras y ángulos, y las coordenadas cartesianas de uno de los pares cinemáticos fijos y del seguidor. Esto permite un planteamiento de ecuaciones de restricción cerrando los lazos del mecanismo y utilizando condiciones de sólido rígido, por lo que la formulación de las mismas es sencilla y compacta, así como obtener

directamente información útil para evaluar la aptitud del mecanismo, como son las coordenadas cartesianas del seguidor (trayectoria descrita por el mecanismo).

4.2 Planteamiento del problema de síntesis

El problema de síntesis se puede abordar como un problema de optimización, en el que a través de la modificación de unas variables de diseño se intenta conseguir que el mecanismo alcance ciertos parámetros funcionales. Estos parámetros funcionales recogen los requisitos que debe cumplir el mecanismo objeto de síntesis, y en un problema plano se podrían definir hasta tres parámetros funcionales para cada elemento e instante de tiempo, siendo las coordenadas cartesianas de uno de sus puntos y un ángulo.

Los parámetros funcionales se pueden clasificar en dos grandes grupos, los deseados y los generados, dependientes estos últimos de las variables de diseño $\mathbf{w}$, y del valor de las coordenadas generalizadas. La comparación de estos dos grupos de parámetros funcionales mediante cierta expresión matemática será la función objetivo a minimizar durante el proceso de síntesis. En este caso esa función objetivo es una función de error, como la TRS, descrita en el capítulo 3, o la función DCM (distancia cuadrática media).

Así, en un problema de síntesis dimensional para generación de trayectorias, la función de error vendrá definida por la diferencia entre una trayectoria objetivo y la trayectoria descrita por el seguidor del mecanismo objeto de optimización. La trayectoria objetivo se definirá como una serie de puntos a alcanzar por el seguidor. Comparando la trayectoria de seguidor de un mecanismo propuesto con la trayectoria objetivo, se obtendría el error alcanzado por el mecanismo propuesto en cada uno de los puntos de la trayectoria. Sin embargo, la función objetivo debe describirse mediante un escalar, por lo que para el caso del DMC, se definirá la función de error como una media cuadrática del error en cada punto, según la expresión recogida en las Ecuaciones (3.1) y (3.2), y que se vuelven a mostrar a continuación:

$$\left. \begin{aligned} \mathbf{d}^{iT} &= \begin{bmatrix} x_d^i & y_d^i \end{bmatrix} \\ \mathbf{g}^{iT} &= \begin{bmatrix} x_g^i & y_g^i \end{bmatrix} \end{aligned} \right\} i = 1, 2, \dots, p \quad (3.1)$$

$$E = \frac{1}{2} \sum_{i=1}^p \left[(x_g^i - x_d^i)^2 + (y_g^i - y_d^i)^2 \right] = \frac{1}{2} \sum_{i=1}^p \left[(\mathbf{g}^i - \mathbf{d}^i)^T (\mathbf{g}^i - \mathbf{d}^i) \right] \quad (3.2)$$

El mecanismo objeto de estudio se describirá mediante sus ecuaciones de restricción, mostradas en la Ecuación (4.2). Estas ecuaciones de restricción dependerán de la posición de los diferentes elementos representadas por las coordenadas generalizadas $\mathbf{q}$, que se obtendrán mediante el análisis de la cinemática del sistema. Por otro lado, estas ecuaciones también dependerán de los parámetros del sistema $\mathbf{w}$, o variables de diseño, que representan las longitudes de los diferentes elementos, así como la posición en el sistema de referencia global de alguno de sus elementos no móviles, en posición de uno de sus puntos y ángulo respecto a una referencia fija.

Con lo explicado hasta ahora, el problema de síntesis se podría escribir formalmente según la Ecuación (4.3)

$$\min_{\mathbf{w}} E = \min_{\mathbf{w}} \left[\frac{1}{2} \sum_{i=1}^p \left[(\mathbf{g}^i - \mathbf{d}^i)^T (\mathbf{g}^i - \mathbf{d}^i) \right] \right] \quad (4.3)$$

Sujeto a $\{\Phi(\mathbf{w}, \mathbf{q}) = 0\}$

4.3 Resolución del problema de síntesis

Las ecuaciones de restricción a la que se ve sujeta la optimización para buscar el mínimo de la función de error elegida, se puede eliminar mediante la sustitución en los puntos generados por las ecuaciones mediante las que se obtienen dichos puntos. Estas ecuaciones dependen de las coordenadas generalizadas, $\mathbf{q}$. A su vez, las coordenadas generalizadas $\mathbf{q}$, dependen de las variables de diseño $\mathbf{w}$, por lo que la Ecuación (4.3) se podría reescribir como se presenta en la Ecuación (4.4).

$$\min_{\mathbf{w}} \left[\frac{1}{2} \sum_{i=1}^p \left[(\mathbf{g}^i(\mathbf{q}(\mathbf{w})) - \mathbf{d}^i)^T (\mathbf{g}^i(\mathbf{q}(\mathbf{w})) - \mathbf{d}^i) \right] \right] \quad (4.4)$$

Este tipo de problemas de optimización de funciones de forma cuadrática pueden resolverse según varios métodos que aprovechan esa forma cuadrática para acelerar la convergencia del método. De los métodos existentes el más simple es el de Gauss-Newton, que es monótonamente convergente, produce puntos estacionarios en mínimos, máximos o puntos de inflexión, y en las proximidades de la solución converge de forma cuadrática. Este tipo de propiedades son interesantes para la implementación del método de solución, siendo este el método elegido para este trabajo, aunque lo expuesto en este trabajo podría ser ampliable a métodos más elaborados como el de Levenberg-Marquardt.

4.3.1 Cálculo de matriz Jacobiana

El método de Gauss-Newton linealiza el error de síntesis, cumpliéndose en el punto óptimo la condición de punto estacionario, siendo el valor del gradiente de la función objetivo en ese punto nulo, por lo que a partir de la Ecuación (4.4), se podría escribir lo expuesto en este párrafo mediante la Ecuación (4.5)

$$\begin{aligned} \min_{\mathbf{w}} \left[\frac{1}{2} \left[\nabla(\boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w})))^T \cdot \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w})) + (\boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w})))^T \cdot \nabla \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w})) \right] \right] = \\ = \min_{\mathbf{w}} \left[\nabla(\boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w})))^T \cdot \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w})) \right] \Rightarrow \\ \mathbf{J}^T \cdot \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w})) = 0 \end{aligned} \quad (4.5)$$

Donde $\boldsymbol{\varepsilon}$ estaría definido por la Ecuación (4.6).

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \mathbf{g}^1 - \mathbf{d}^1 \\ \mathbf{g}^2 - \mathbf{d}^2 \\ \dots \\ \mathbf{g}^P - \mathbf{d}^P \end{bmatrix} \quad (4.6)$$

En esta ecuación y sucesivas, se ha omitido la dependencia de los términos respecto de las variables de diseño y de las coordenadas dependientes para mayor claridad. Siendo $\mathbf{J}$ en la Ecuación (4.5) la matriz jacobiana del error respecto a las variables de diseño, tal y como se muestra en la Ecuación (4.7), con k variables de diseño y p puntos de precisión.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \varepsilon_1}{\partial \mathbf{w}_1} & \frac{\partial \varepsilon_1}{\partial \mathbf{w}_2} & \dots & \frac{\partial \varepsilon_1}{\partial \mathbf{w}_k} \\ \frac{\partial \varepsilon_2}{\partial \mathbf{w}_1} & \frac{\partial \varepsilon_2}{\partial \mathbf{w}_2} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varepsilon_p}{\partial \mathbf{w}_1} & \dots & \dots & \frac{\partial \varepsilon_p}{\partial \mathbf{w}_k} \end{bmatrix} \quad (4.7)$$

El cálculo de $\mathbf{J}$ se puede facilitar teniendo en cuenta que la derivación del error respecto a las variables de diseño se puede hacer como una derivación por partes como se muestra en la Ecuación (4.8).

$$\begin{aligned} \mathbf{J} &= \frac{d\varepsilon}{d\mathbf{w}} = \frac{\partial \varepsilon}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial \mathbf{w}} + \frac{\partial \varepsilon}{\partial \mathbf{w}} \\ \Phi(\mathbf{q}(\mathbf{w}), \mathbf{w}) &= 0 \\ \frac{\partial \Phi}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial \mathbf{w}} + \frac{\partial \Phi}{\partial \mathbf{w}} &= 0 \Rightarrow \frac{\partial \mathbf{q}}{\partial \mathbf{w}} = -\frac{\partial \Phi}{\partial \mathbf{q}}^{-1} \cdot \frac{\partial \Phi}{\partial \mathbf{w}} \\ \Rightarrow \mathbf{J} &= -\frac{\partial \varepsilon}{\partial \mathbf{q}} \cdot \frac{\partial \Phi}{\partial \mathbf{q}}^{-1} \cdot \frac{\partial \Phi}{\partial \mathbf{w}} \end{aligned} \quad (4.8)$$

Obteniéndose de esta forma el jacobiano exacto con un coste computacional reducido.

4.3.2 Cálculo de tamaño de paso / módulo de la dirección de descenso

Debido a la no linealidad del problema es necesario el uso de un proceso iterativo. En este proceso iterativo se aproxima la función de error estimado por su desarrollo en serie de Taylor hasta su derivada de primer orden, tal y como se muestra en la Ecuación (4.9)

$$\varepsilon(\mathbf{q}(\mathbf{w})) = \varepsilon(\mathbf{q}(\mathbf{w}))_i + \nabla \varepsilon(\mathbf{q}(\mathbf{w}))_i \cdot \Delta \mathbf{w} \quad (4.9)$$

Donde el subíndice i indica los valores iniciales de las variables de diseño o de la iteración de diseño actual. Si se sustituye la Ecuación (4.7) en la (4.9), se obtiene la Ecuación (4.10).

$$\mathbf{J}^T [\boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w}))_i + \nabla \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w}))_i \cdot \Delta \mathbf{w}] = 0$$

$$\Delta \mathbf{w} = -[\mathbf{J}_i^T \cdot \mathbf{J}_i]^{-1} \cdot \mathbf{J}_i^T \cdot \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w}))_i \quad (4.10)$$

La Ecuación (4.10) permite calcular el vector $\Delta \mathbf{w}$. Este vector representa la variación de las variables de diseño respecto a la iteración actual para minimizar la función objetivo definida en la Ecuación (4.3); es decir, representa la dirección de variación de las variables de diseño. Sin embargo, esta variación de las variables de diseño no es única para obtener el mínimo de la función, ya que se ha obtenido a partir de un desarrollo de Taylor de primer orden de la función, por lo que el mínimo se obtendrá tras un proceso iterativo.

Esta dirección de variación de las variables de diseño representa la dirección en la cual se encuentra la máxima pendiente de descenso de la función objetivo. Esta dirección se expresará tal y como se presenta en la (4.11)(4.10).

$$\mathbf{d} = -[\mathbf{J}_i^T \cdot \mathbf{J}_i]^{-1} \cdot \mathbf{J}_i^T \cdot \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w}))_i \quad (4.11)$$

Por otro lado la Ecuación (4.11) presenta una inversa de matriz. Se debe tener en cuenta que la inversión de una matriz proporciona buenos resultados siempre que dicha matriz esté bien condicionada. Debido a esto, la bondad de la variación de la dirección de descenso de la función objetivo expresada por la Ecuación (4.11) dependerán de si dicha matriz está bien condicionada. Para evitar esta dependencia a lo largo de este trabajo se ha utilizado una matriz pseudoinversa de Moore-Penrose, en la que se mejora el condicionamiento la matriz a invertir sumándole una matriz identidad de la misma dimensión multiplicada por un factor tan pequeño como sea posible para conseguir la inversión de matriz. Esto se traduce matemáticamente en lo expresado en la Ecuación (4.12).

$$[\mathbf{J}_i^T \cdot \mathbf{J}_i]^{-1} \approx \lim_{\zeta \rightarrow 0} [\mathbf{J}_i^T \cdot \mathbf{J}_i + \zeta \cdot \mathbf{I}]^{-1} \quad (4.12)$$

Por lo que si se sustituye en la Ecuación (4.11) la propuesta formulada en la Ecuación (4.12), se obtiene la expresión mostrada en la Ecuación (4.13).

$$\mathbf{d} = -\left(\lim_{\zeta \rightarrow 0} [\mathbf{J}_i^T \cdot \mathbf{J}_i + \zeta \cdot \mathbf{I}]\right)^{-1} \cdot \mathbf{J}_i^T \cdot \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w}))_i \quad (4.13)$$

Para explicar el cálculo de la pseudoinversa, se puede asumir que en una descomposición de una matriz en valores singulares, de una matriz $\mathbf{A}$, se obtendría la Ecuación (4.14)

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T \quad (4.14)$$

Donde $\mathbf{U}$ y $\mathbf{V}$ son matrices cuyas columnas son vectores ortonormales y $\mathbf{S}$ representa la matriz de valores singulares de la matriz $\mathbf{A}$. Si se considera ahora $\mathbf{A}$ invertible, se tendría la Ecuación (4.15)

$$\mathbf{A}^{-1} = \tilde{\mathbf{U}} \cdot \tilde{\mathbf{S}} \cdot \tilde{\mathbf{V}}^T \quad (4.15)$$

Con $\tilde{\mathbf{V}} = \mathbf{V} \cdot \mathbf{P}$, y $\tilde{\mathbf{U}} = \mathbf{U} \cdot \mathbf{P}$, siendo $\mathbf{P}$ una matriz de permutación. Al ser $\mathbf{S}$ una matriz diagonal que contiene los valores singulares de $\mathbf{A}$, $\tilde{\mathbf{S}}$ se puede invertir fácilmente, como una matriz diagonal de valores inversos a los valores singulares.

Así, se puede definir la matriz pseudoinversa de $\mathbf{A}$, $\mathbf{A}^\dagger$, mediante el cálculo de la única matriz que cumpla las condiciones mostradas en la (4.16)

$$\begin{aligned} (i) \quad & \mathbf{A} \cdot \mathbf{A}^\dagger \cdot \mathbf{A} = \mathbf{A} \\ (ii) \quad & \mathbf{A}^\dagger \cdot \mathbf{A} \cdot \mathbf{A}^\dagger = \mathbf{A}^\dagger \\ (iii) \quad & \mathbf{A}^\dagger \cdot \mathbf{A} \text{ es hermítica} \\ (iv) \quad & \mathbf{A} \cdot \mathbf{A}^\dagger \text{ es hermítica} \end{aligned} \quad (4.16)$$

A pesar de que la definición de la matriz $\mathbf{A}^\dagger$ depende de la elección de las matrices $\mathbf{U}$ y $\mathbf{V}$ en la descomposición de $\mathbf{A}$ en valores singulares, se puede demostrar que a pesar de la arbitrariedad en la elección de los vectores singulares por la izquierda y por la derecha, $\mathbf{U}$ y $\mathbf{V}$, la inversa de Moore-Penrose es única, y es una generalización de la matriz inversa.

Tras esta breve explicación sobre el cálculo de la matriz pseudoinversa, y aunque la Ecuación (4.13) proporcione la dirección de variación de las variables de diseño para la obtención del error mínimo (ecuación (4.4)), el módulo de la variación se debe reducir, ya que no se puede obtener el mínimo en una única iteración debido a que $\mathbf{d}$ se ha obtenido de forma aproximada. Esta reducción puede expresarse matemáticamente según la Ecuación (4.17).

$$\Delta \mathbf{w} = -\alpha \left(\lim_{\zeta \rightarrow 0} [\mathbf{J}_i^T \cdot \mathbf{J}_i + \zeta \cdot \mathbf{I}]^{-1} \right) \cdot \mathbf{J}_i^T \cdot \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w}))_i = -\alpha \cdot \mathbf{d} \quad (4.17)$$

Donde α representa la modulación de la variación de las variables de diseño, es decir, el tamaño de paso.

Así, para operar de forma iterativa, teniendo en cuenta que el subíndice indica la iteración de diseño, se puede representar la variación de las variables de diseño de una iteración a la siguiente como se muestra en la Ecuación (4.18).

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \alpha_i \left(\lim_{\zeta \rightarrow 0} [\mathbf{J}_i^T \cdot \mathbf{J}_i + \zeta \cdot \mathbf{I}]^{-1} \right) \cdot \mathbf{J}_i^T \cdot \boldsymbol{\varepsilon}(\mathbf{q}(\mathbf{w}))_i = \mathbf{w}_i - \alpha_i \cdot \mathbf{d} \quad (4.18)$$

El valor de α ya ha sido estudiado en otros trabajos como (De Juan, 2011), y en este caso se ha utilizado un valor que varía a lo largo de la optimización dependiendo de la convergencia de la solución. Así el valor inicial debe tener una magnitud reducida, α_0 (de forma orientativa y como ejemplo, en este trabajo se ha utilizado un valor de 0.001). Este valor es el utilizado en la primera iteración. A partir de esta iteración siempre que la solución converja el valor de α se duplica en cada iteración en un factor $\lambda_{M\alpha}$, hasta un valor máximo, α_{lim} ; reduciéndose en un factor $\lambda_{N\alpha}$ en el caso en que la solución no converja. El proceso se muestra en el esquema de la Fig. 4.1.

Los valores de α_{lim} varían dependiendo de la fase de la optimización que se esté abordando. Estos valores se han fijado tras diversas pruebas en 0.2 para las fases de traslación, traslación con giro y traslación con giro y escalado, y 0.05 para la fase final en la que se optimizan todas las variables de diseño. De igual forma el valor de $\lambda_{N\alpha}$ utilizado es de 0.1 y el de $\lambda_{M\alpha}$ es de 2 para las fases de traslación, rotación y escalado y 1.2 para la última fase de optimización.

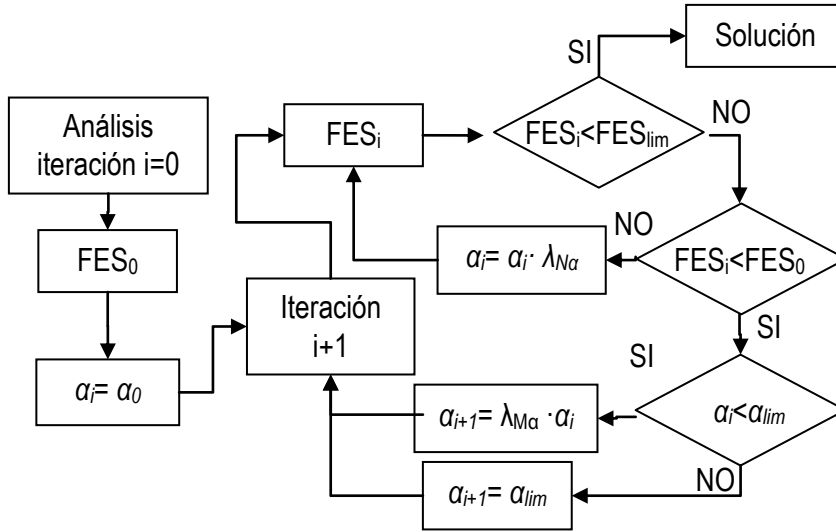


Fig. 4.1 Esquema de variación del tamaño de paso

4.3.3 Verificación del ensamblaje

Con lo explicado sobre el tamaño de paso y utilizando la Ecuación (4.18) se obtienen nuevos valores de las variables de diseño en cada iteración del proceso de optimización.

Tras la obtención de las variables en la iteración de optimización, se deben resolver las ecuaciones de restricción del mecanismo (4.2), de forma que el sistema de ecuaciones no lineales que las forman se cumpla para cada posición de síntesis. Sin embargo, si el tamaño de paso fuese excesivamente grande, no se puede garantizar que las ecuaciones de restricción se cumplan. Para poder valorar si las ecuaciones de restricción se cumplen o no, se debe verificar la (4.19).

$$\|\Phi(\mathbf{q}_i \mathbf{w})\| \leq \chi_c \quad (4.19)$$

En la Ecuación (4.19) χ_c representa un vector de tolerancia cercano a cero. Si la Ecuación (4.19) no se cumple, se debe reducir el tamaño de paso hasta que se cumpla la condición, pudiéndose ensamblar el mecanismo en todas las posiciones de síntesis.

4.3.4 Criterios de convergencia-parada

El proceso global de optimización tiene dos criterios de parada, que determinan cuándo debe de finalizar la optimización. Estos criterios son un valor de error objetivo o un número de iteraciones máximo.

Mediante el valor de error objetivo se limita el valor del error a obtener, es decir, un valor de error aceptable en el mecanismo objeto de síntesis. Este valor parece suficiente para marcar un criterio de parada del proceso de síntesis, pero sin embargo no tiene por qué alcanzarse, ya que dependiendo del individuo inicial y de la función objetivo no se puede garantizar que cierto valor de error máximo pueda obtenerse. Debido a esto es necesario utilizar un criterio de parada adicional, que en este trabajo se utiliza como criterio el número límite de iteraciones. Lo explicado en este párrafo se presenta en forma de diagrama de flujo en la Fig. 4.2, revisándose estos criterios en todas las iteraciones del proceso de síntesis.

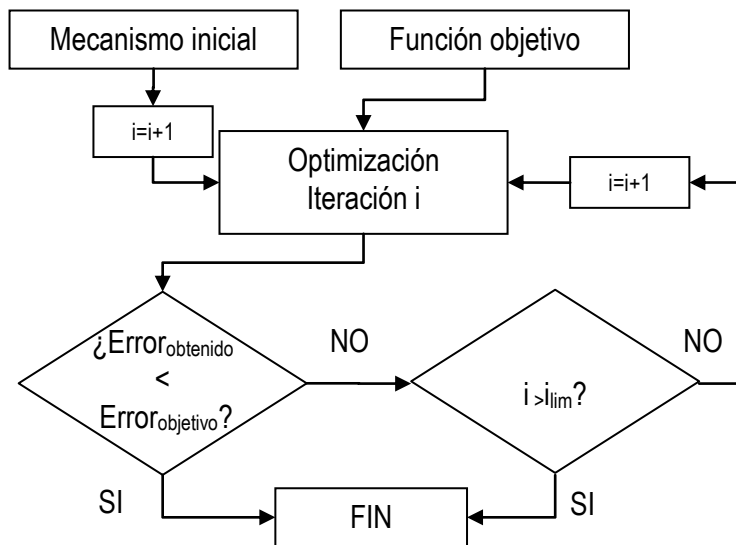


Fig. 4.2 Diagrama de flujo de criterio general de parada

Adicionalmente al criterio general de parada se han implementado también criterios de cambio de fase. Así, para cambiar de la fase de optimización de las variables de diseño que permiten traslación del individuo, w_t , a la siguiente fase, rotación con traslación, w_r , se debe fijar un criterio. El

criterio utilizado en este trabajo es la variación del error entre la última iteración y la anterior. Así, si no se consigue reducir el error entre una iteración y la anterior en un porcentaje mayor a criterio establecido, la siguiente iteración cambiará a una fase posterior. De esta forma tras llegar a la última fase y si no se han alcanzado los criterios de convergencia-parada globales, se vuelve de nuevo a comenzar con la primera fase de optimización, hasta alcanzar los criterios de convergencia-parada globales, tal y como se muestra en la Fig. 4.3.

Con esta estrategia de cambio de fase se intenta evitar que el método de optimización caiga en mínimos locales de la función objetivo. Al modificar el vector de variables de diseño en cada una de las fases, se modifica el rango de direcciones de la dirección de descenso, en función de las variables de diseño consideradas.

Esta modificación del rango de direcciones de descenso con las variables de diseño permite que si en alguna de las fases de optimización el vector de variables de diseño representa un mínimo local para esa fase, se puedan explorar nuevas direcciones de descenso, mediante un cambio de fase. Este cambio de fase puede permitir obtener una nueva dirección de descenso en la función objetivo, ya que con el cambio de fase, el vector de diseño no tiene por qué representar un mínimo local de la función objetivo; lo que sucedía en la fase anterior, por lo que nuevamente se modificará el vector de variables de diseño hasta incurrir en un mínimo local de la fase, y así nuevamente cambiar de fase.

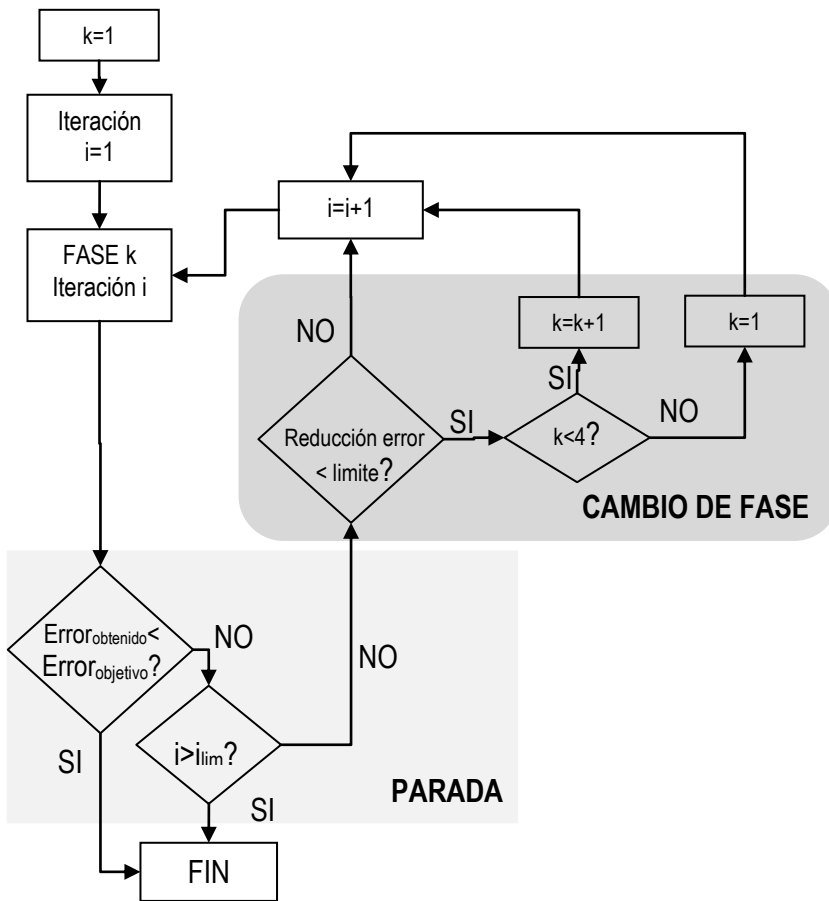


Fig. 4.3 Diagrama de flujo de cambio de fase a lo largo de la optimización

De forma gráfica supongamos un espacio de trabajo de dos variables de diseño, y supongamos una función objetivo como la presentada en la Fig. 4.4.

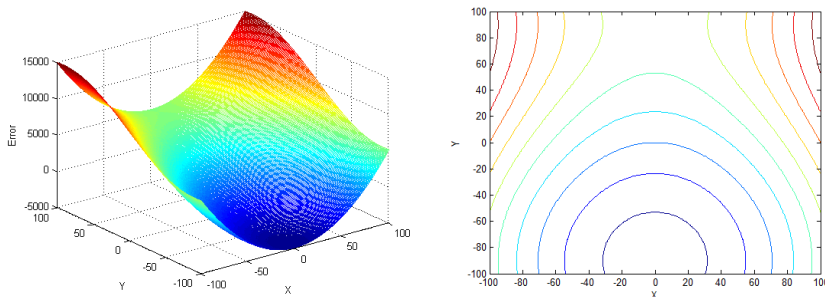


Fig. 4.4 Función objetivo para 2 variables de diseño.

Si para esta función objetivo se realiza una optimización en X, considerando un individuo inicial $X=-100$, $Y=100$, se obtendrá un mínimo para $X=0$, $Y=100$. Sin embargo, este mínimo no es un mínimo de la función objetivo. Así, si en una fase posterior se realiza una optimización en X e Y, se pasará a un nuevo mínimo en $X=0$, $Y=0$.

Lo explicado en un espacio de 2 variables es sencillo, pero en una síntesis de mecanismos, el número de variables de diseño es muy superior. Por lo que la función objetivo es mucho más compleja, y con posibilidad de incurrir en un número de mínimos locales muy superior.

4.3.5 Diagrama de flujo

Lo explicado hasta este punto se puede condensar en el diagrama de flujo presentado en la Fig. 4.5.

Partiendo de un individuo inicial, y mediante la función objetivo, se evalúa la aptitud del mecanismo calculando el error de este individuo. A partir de aquí se inicializan las variables de control de iteraciones y de fase de optimización. Se calcula el jacobiano de la función objetivo, que permite el cálculo de variación de las variables de diseño para reducir el error de la función objetivo. En un principio, las variables de diseño que se consideran permiten la traslación del mecanismo.

Tras la definición del nuevo individuo se chequea si se puede ensamblar y si el error que se obtiene con el nuevo individuo es un error menor al error obtenido con el individuo de partida. En función de los chequeos se propone un nuevo tamaño de paso. Si los chequeos son positivos se aumenta el tamaño de paso para la siguiente iteración de la optimización. Este aumento de tamaño de paso tiene un límite, que puede variar en función de la fase de optimización en la que se encuentre el proceso. En este trabajo, y tras realizar diferentes pruebas con límites de tamaño de paso, se ha determinado que para las fase de traslación el límite de tamaño de paso utilizado $\alpha_{lim}=0.2$; para las fases de rotación y escalado $\alpha_{lim}=0.1$, y para la fase en la que se consideran todas las variables de diseño el límite del tamaño de paso considerado es de $\alpha_{lim}=0.04$. De igual forma, el aumento del tamaño de paso varía en función de la fase de optimización que se esté abordando. Así en la primera fase el aumento del tamaño de paso entre iteraciones puede tomar un valor de $\lambda_{Ma}=2$, siendo el valor de aumento de tamaño de fase para el resto de fases de $\lambda_{Ma}=1.2$.

En caso de que alguno de estos chequeos sea negativo, se reduce el tamaño de paso, aplicando un factor de $\lambda_{Nd}=10$ en todas las fases. Sin embargo, este tamaño de paso no se aplica a la siguiente iteración, sino que se recalcula con el nuevo tamaño de paso el individuo de la iteración actual.

Tras esto, se verifica mediante el criterio de parada y convergencia si se debe detener la optimización o debe continuar. En este trabajo el valor de error de la función objetivo exigido para la parada es de $Error_{objetivo}=1.e-4$. Si no se llega al valor de error fijado por el $Error_{objetivo}$, tras realizar un número de iteraciones limitado por $i_{lim}=400$, el algoritmo se detiene, a pesar de no haber obtenido el error objetivo marcado. Si no se cumple ninguno de los criterios de parada, el proceso de optimización continúa, pasando a evaluar la fase en la que se abordará la siguiente iteración en el módulo de control de fase.

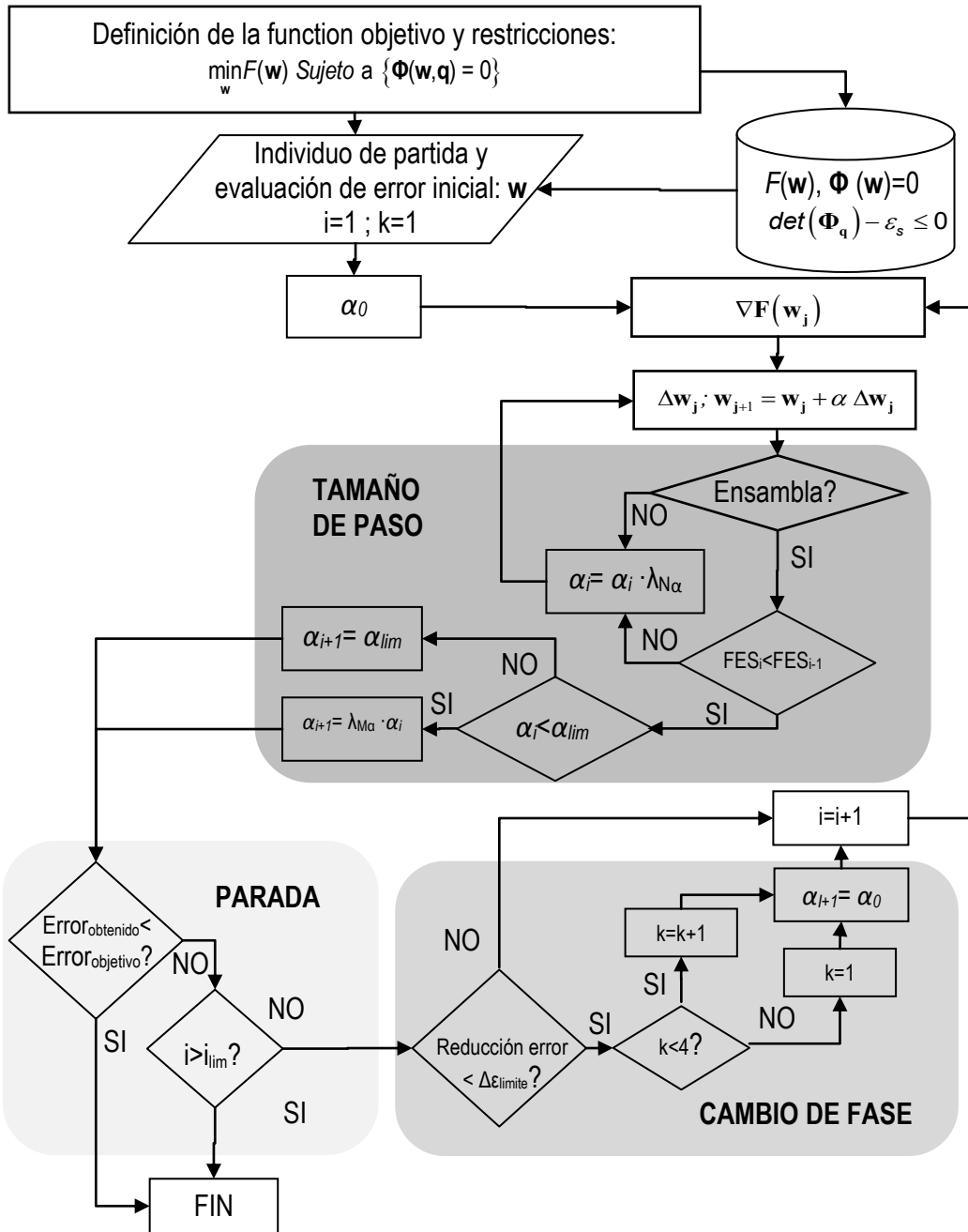


Fig. 4.5 Diagrama de flujo para la implementación de la optimización local propuesta

En el módulo de control de fase, se verifica si la diferencia de error entre la iteración actual y la anterior es suficiente, según un límite fijado por $\Delta\epsilon_{\text{limite}}$. Si la reducción del error es mayor de $\Delta\epsilon_{\text{limite}}$ (en el cuadro de cambio de fase de la Fig. 4.5), se mantiene la fase de optimización y se pasa a la siguiente iteración. En caso de que este criterio no sea satisfecho, se modifica la fase de optimización. En este trabajo se han considerado las 4 fases de optimización descritas, pasando progresivamente desde la fase 1 hasta la 4. Tras llegar a la fase 4, se vuelve a la fase 1. Este proceso se repite hasta que se cumple alguno de los criterios de parada. En cuanto al límite de variación de error entre iteraciones, en este trabajo se ha decidido usar un límite $\Delta\epsilon_{\text{limite}}=0.1\%$ para todas las fases, tras valorar mediante pruebas diferentes límites. Cada una de las fases de diseño tiene unas variables de diseño asociadas, tal y como se describe en la Tabla 4.1

Fase de optimización	Variables de diseño
1	$\mathbf{w}_T$
2	$\mathbf{w}_R$
3	$\mathbf{w}_S$
4	$\mathbf{w}$

Tabla 4.1 Variables de diseño a considerar según la fase de optimización

Definido el nuevo individuo, su error, el tamaño de paso de la fase siguiente, y la fase, se procede de nuevo a evaluar el jacobiano de la función objetivo para la nueva iteración, y así poder obtener un nuevo vector de variables de diseño, hasta cumplir los criterios de parada propuestos.

4.4 Ejemplos de síntesis local

En este punto se van a presentar ejemplos de síntesis local, continuando con las trayectorias objetivo utilizadas en el capítulo anterior, arco de elipse y gota de agua Tabla 3.1, y añadiendo una nueva: un ángulo recto que se presenta en la Tabla 4.2 y Fig. 4.6.

El primer objetivo de estos ejemplos es mostrar los resultados obtenidos por el algoritmo de síntesis local propuesto. Otro objetivo de estos ejemplos es poder comparar los resultados obtenidos mediante el estimador de error TRS, en el que se desacoplan las fases de traslación, rotación y escalado, con un método de optimización local en el que estas

fases se abordan de forma acumulativa y conjunta, sometiendo al mecanismo a una optimización morfológicas tras estas fases.

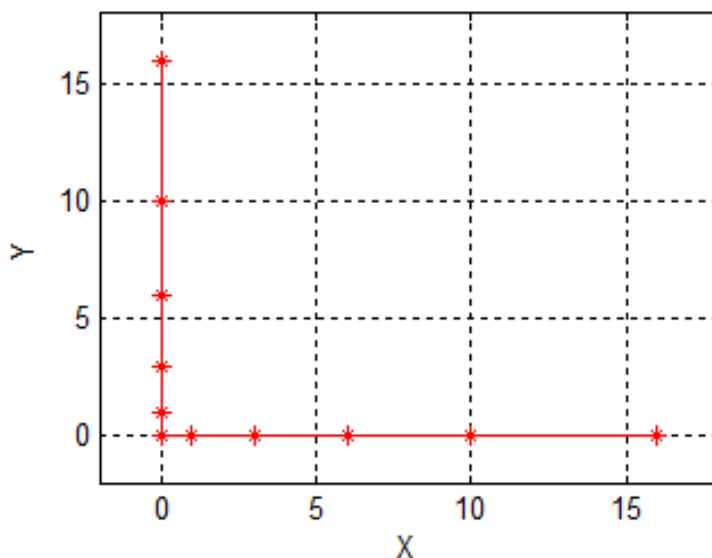


Fig. 4.6 Trayectoria de ángulo recto

	Ángulo Recto	
<i>Punto n°</i>	<i>X1</i>	<i>Y1</i>
1	0	16
2	0	10
3	0	6
4	0	3
5	0	1
6	0	0
7	1	0
8	3	0
9	6	0
10	10	0
11	16	0

Tabla 4.2 Trayectoria de ángulo recto

Al igual que en el capítulo anterior, se parte de mecanismos generados aleatoriamente. Los mecanismos de partida para las tres trayectorias propuestas se presentan en Tabla 4.3, Tabla 4.4 y Tabla 4.5.

Original	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
A	9.14	5.19	0.74	-2.42	0.57	9.56	13.48	19.39	3.63	-5.95
B	-2.91	9.77	2.64	-1.32	4.57	14.34	6.56	13.50	11.73	-0.46
C	-7.56	-3.04	2.26	-1.00	5.39	15.25	15.87	6.58	16.33	0.13
D	4.40	8.45	1.22	-1.56	2.45	10.75	7.84	8.07	13.53	4.14
E	6.02	0.98	1.73	1.08	4.06	6.70	12.42	10.60	5.72	1.27
F	-7.81	-5.43	-2.07	1.14	13.35	16.86	17.16	14.50	16.58	-0.66
G	2.51	5.23	0.04	3.13	1.64	4.27	4.58	6.76	2.48	0.17
H	5.27	9.88	0.87	-2.54	4.96	7.80	8.46	8.25	9.06	-4.52
I	6.28	8.04	0.34	-1.17	4.34	11.41	7.40	10.16	7.86	4.98
J	3.83	8.19	0.91	-0.80	8.29	15.07	14.44	18.15	6.54	-0.06

Tabla 4.3 Mecanismos originales de arco abierto

Original	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
K	0.26	-6.32	0.46	2.68	2.49	7.42	8.05	6.09	14.38	-0.67
L	8.12	3.59	-0.17	-1.42	8.59	18.70	11.53	17.74	12.45	0.33
M	2.51	-0.26	2.61	-0.32	3.26	13.98	17.20	7.07	1.46	-1.49
N	6.35	0.66	-0.66	-1.70	3.67	12.14	5.46	11.14	4.00	-0.33
O	4.25	-9.15	-0.08	3.01	4.28	9.02	11.83	13.11	15.37	-0.30
P	-4.90	5.61	2.85	-1.86	2.19	16.05	12.42	12.32	6.25	-0.19
Q	-1.71	-6.44	-2.44	1.30	3.77	8.51	9.37	10.60	7.31	-2.28
R	0.94	-3.19	-1.06	-0.50	8.41	10.71	10.16	9.46	8.53	-0.68
S	5.23	8.10	0.12	-1.91	7.87	13.76	14.33	13.70	4.88	1.55
T	1.13	2.16	2.01	2.45	4.17	14.38	14.38	10.59	4.01	0.19

Tabla 4.4 Mecanismos originales de gota de agua

Original	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
U	0.04	-2.04	-0.87	-0.13	10.03	13.05	15.42	14.10	17.20	5.70
V	-3.42	-4.19	-2.24	2.33	3.08	9.36	7.35	8.26	16.90	-0.61
W	6.38	-4.74	-1.03	2.76	14.28	15.93	17.80	16.16	14.71	-0.34
X	6.54	1.99	-0.56	2.22	5.03	16.32	15.03	8.31	5.59	5.12
Y	-1.40	-2.22	-2.61	0.28	2.95	10.85	7.71	7.25	9.19	-0.18
Z	6.10	-0.87	-0.14	0.85	3.86	10.78	14.68	12.89	13.27	4.79
AA	2.45	0.43	-0.59	-1.65	8.66	17.93	17.56	12.95	7.19	0.68
AB	-6.82	9.45	1.93	-1.34	6.75	10.26	11.26	12.20	14.74	-0.45
AC	-0.05	9.83	0.84	-0.63	2.11	7.74	7.59	5.23	5.20	-1.49
AD	8.03	2.36	-0.16	-0.50	2.14	10.45	11.72	16.06	19.28	0.76

Tabla 4.5 Mecanismos originales de ángulo recto

Aplicando el estimador de error propuesto, TRS, en el capítulo anterior, los mecanismos se modifican en traslación, rotación y escalado, obteniendo para cada trayectoria los mecanismos mostrados en la Tabla 4.6, Tabla 4.7 y Tabla 4.8 respectivamente.

TRS	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
A	33.95	27.77	0.96	-1.70	5.65	95.57	134.76	193.93	36.34	0.34
B	-116.8	-35.76	-2.23	0.09	36.19	113.59	51.95	106.93	92.90	-0.46
C	-2.90	-2.03	2.98	0.06	2.33	6.59	6.85	2.84	7.05	0.13
D	16.01	-27.80	0.31	-2.01	7.10	31.19	22.75	23.40	39.27	4.14
E	-16.09	15.02	1.80	1.16	36.05	59.47	110.26	94.09	50.76	1.27
F	14.85	-7.97	-3.02	0.19	26.44	33.39	34.00	28.73	32.84	-0.66
G	26.58	-26.75	-1.68	1.41	16.44	42.68	45.80	67.62	24.80	0.17
H	13.79	43.97	1.19	-1.30	15.54	24.47	26.54	25.87	28.43	-4.52
I	2.37	-9.96	-0.61	-1.66	15.78	41.49	26.89	36.93	28.59	-1.30
J	-1.69	-2.83	2.68	0.97	37.12	67.48	64.68	81.30	29.28	-0.06

Tabla 4.6 Mecanismos de arco abierto tras estimador de error propuesto, TRS

TRS	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
K	9.55	4.44	1.88	0.83	0.84	2.49	2.70	2.04	4.82	5.61
L	4.85	6.85	0.77	2.29	1.65	3.60	2.22	3.41	2.40	0.33
M	5.31	4.57	-3.07	1.80	0.77	3.32	4.09	1.68	0.35	4.79
N	4.07	3.83	-1.70	1.79	1.15	3.80	1.71	3.49	1.25	-0.33
O	-1.53	2.64	-1.51	0.57	2.03	4.27	5.59	6.20	7.27	-0.30
P	8.19	6.73	1.64	1.71	1.35	9.91	7.67	7.60	3.86	-0.19
Q	6.50	4.33	3.06	1.77	0.83	1.88	2.07	2.34	1.62	4.00
R	3.89	3.30	-2.43	1.66	2.31	2.94	2.79	2.60	2.34	-0.68
S	4.74	5.01	-0.42	2.08	0.86	1.51	1.57	1.50	0.54	1.55
T	5.84	5.45	1.77	1.71	1.29	4.44	4.44	3.27	1.24	0.19

Tabla 4.7 Mecanismos de gota de agua tras estimador de error propuesto, TRS

Finalmente en la Tabla 4.9, Tabla 4.10 y Tabla 4.11 se presentan los mecanismos tras el proceso de optimización local explicado a lo largo de este capítulo. El número de iteraciones máximas permitidas durante el proceso de optimización local ha sido de 400.

TRS	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
U	5.41	2.34	-1.13	2.18	1.80	2.34	2.77	2.53	3.09	5.70
V	-5.00	7.80	2.52	3.08	1.97	5.99	4.70	5.28	10.81	-0.61
W	4.24	9.41	-2.30	1.49	14.96	16.69	18.64	16.93	15.41	-0.34
X	13.39	-0.69	-0.49	2.29	8.38	27.19	25.03	13.85	9.32	5.12
Y	28.29	9.11	-0.22	2.67	7.27	26.70	18.98	17.84	22.62	-0.18
Z	13.72	23.20	1.11	2.11	5.84	16.29	22.17	19.47	20.04	-1.49
AA	0.19	6.77	-2.91	2.32	11.48	23.79	23.29	17.18	9.54	0.68
AB	7.99	10.30	0.52	-0.18	3.56	5.41	5.94	6.43	7.77	-0.45
AC	-24.0	12.39	-2.94	1.87	15.84	58.19	57.10	39.36	39.09	-1.49
AD	21.73	-17.5	-0.85	1.09	3.20	15.65	17.55	24.06	28.88	0.76

Tabla 4.8 Mecanismos de ángulo recto tras estimador de error propuesto, TRS

Opt. Local	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
A	0.61	11.56	9.47	17.58	4.24	-6.08	-2.03	0.40	136.99	26.80
B	0.65	15.78	0.49	15.95	18.43	-1.33	-0.11	3.33	-128.0	277.10
C	5.58	4.32	17.37	22.12	7.19	0.01	-1.19	2.15	-36.73	5.53
D	1.92	14.09	11.34	6.74	8.79	4.09	-3.09	0.29	31.73	-38.46
E	11.44	2.72	12.27	3.50	4.76	0.71	0.90	1.15	-2.17	-8.44
F	22.99	6.77	9.11	25.57	2.83	-0.83	0.59	-1.90	-5.81	-11.28
G	0.70	16.67	4.83	20.58	5.85	0.43	4.08	-1.43	109.47	-114.4
H	8.72	3.36	12.03	5.38	5.68	-4.68	-5.39	0.88	-4.19	-3.15
I	2.15	15.21	12.84	5.44	3.73	4.79	-2.81	0.49	48.38	3.46
J	1.27	21.77	20.83	3.52	3.83	-0.41	-2.76	0.38	12.38	-16.77

Tabla 4.9 Mecanismos de arco abierto tras proceso de optimización local propuesto

Opt. Local	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
K	1.53	5.38	9.54	12.90	9.44	-0.88	3.52	1.86	-6.25	1.13
L	3.86	11.20	11.18	17.95	21.45	0.03	-2.81	-0.34	-4.75	1.85
M	0.12	0.14	20.72	20.71	0.05	-1.26	0.39	3.31	5.58	4.83
N	0.11	14.65	0.22	14.56	7.90	-0.50	-1.74	-2.14	96.21	52.83
O	2.01	7.04	12.58	16.96	12.35	-0.87	0.38	-1.28	-6.21	1.13
P	0.37	18.21	16.86	2.64	1.85	-0.26	-1.79	1.33	-0.42	4.02
Q	0.19	0.29	14.69	14.65	0.20	-1.58	0.26	-2.86	6.61	4.75
R	9.78	10.20	9.43	9.79	8.19	-0.88	-0.15	-0.77	5.26	4.35
S	2.22	8.08	11.85	17.24	15.73	0.92	0.48	0.75	12.46	-4.32
T	1.98	10.56	7.20	15.36	13.20	-0.53	1.93	2.20	10.03	21.10

Tabla 4.10 Mecanismos de gota de agua tras proceso de optimización local propuesto

Opt. Local	x0	y0	θ_0	θ_{20}	L ₁	L ₂	L ₃	L ₄	L ₅	β
U	10.57	11.50	13.89	15.80	17.84	6.03	-1.27	-1.89	7.06	-5.23
V	1.78	16.85	16.19	3.23	6.01	-1.53	2.18	-2.12	-33.91	-25.63
W	21.63	7.25	25.50	4.70	10.47	-0.76	2.57	-1.26	7.73	4.39
X	1.49	18.24	17.96	2.45	1.60	4.97	2.66	-1.12	8.64	-2.03
Y	0.62	16.64	16.59	0.89	2.29	-1.51	2.36	-2.33	-16.59	-16.57
Z	0.34	1.04	22.86	23.52	2.47	7.06	-1.93	-3.04	17.29	-41.42
AA	2.99	12.98	7.55	17.56	21.24	-0.86	2.05	1.79	62.63	46.29
AB	3.01	9.13	7.51	13.64	17.79	0.95	-3.48	-0.32	30.78	45.47
AC	2.11	7.74	7.59	5.23	5.20	-1.49	-0.63	0.84	-0.05	9.83
AD	0.36	16.11	2.32	17.87	26.06	-0.38	3.28	1.84	138.36	622.23

Tabla 4.11 Mecanismos de ángulo recto tras proceso de optimización local propuesto

A continuación se presenta una comparativa entre el error DMC de la trayectoria generada por el mecanismo original, el error tras aplicar el estimador TRS y tras el proceso de síntesis local con la optimización explicada durante este capítulo. Estos resultados se presentan en tablas según la trayectoria evaluada, así la Tabla 4.12 hace referencia a la comparación en con la trayectoria de arco abierto, la Tabla 4.13 con la trayectoria gota de agua, y la Tabla 4.14 con la trayectoria ángulo recto.

Errores en Trayectoria de arco abierto				
	DMC inicial	TRS	Opt. Local	Comparación Opt. Local vs DMC
A	664.72	21.60	38.37	94.23%
B	1319.63	34.14	6.07	99.54%
C	3340.41	49.37	0.07	100.00%
D	2444.05	45.71	10.60	99.57%
E	1012.78	6.61	0.13	99.99%
F	6085.18	40.41	13.92	99.77%
G	595.84	40.31	27.64	95.36%
H	968.40	29.41	1.35	99.86%
I	1148.85	46.92	13.69	98.81%
J	825.00	43.44	15.95	98.07%

Tabla 4.12 Errores tras optimización local, TRS, DMC y comparación DMC con optimización local

Errores en Trayectoria gota de agua				
	DMC inicial	TRS	Opt. Local	Comparación Opt. Local vs DMC
K	983.23	21.6	0.95	99.90%
L	2174.36	34.14	0.57	99.97%
M	598.62	49.37	2.56	99.57%
N	233.09	45.71	36.55	84.32%
O	255.21	6.61	0.95	99.63%
P	1340.69	40.41	37.36	97.21%
Q	5106.79	40.31	1.73	99.97%
R	1349.04	29.41	34.72	97.43%
S	1398.18	46.92	0.79	99.94%
T	979.07	43.44	0.79	99.92%

Tabla 4.13 Errores tras optimización local, TRS, DMC y comparación DMC con optimización local

Errores en Trayectoria ángulo recto				
	DMC inicial	TRS	Opt. Local	Comparación Opt. Local vs DMC
U	2326.13	273.28	38.24	98.36%
V	496.68	277.69	1	99.80%
W	1379.55	44.18	39.78	97.12%
X	188.79	109.99	39.06	79.31%
Y	459.18	136.48	45.83	90.02%
Z	1075.95	162.17	20.8	98.07%
AA	679.59	36.72	6.13	99.10%
AB	721.39	247.36	5.23	99.28%
AC	759.55	70.83	729.2	4.00%
AD	1471.74	221.21	34.95	97.63%

Tabla 4.14 Errores tras optimización local, TRS, DMC y comparación DMC con optimización local en trayectoria de ángulo recto

En las Fig. 4.7, la Fig. 4.8 y la Fig. 4.9 se muestran los mecanismos obtenidos tras la optimización local y la evolución del error en cada uno de los 30 procesos de síntesis local estudiados en estos ejemplos.

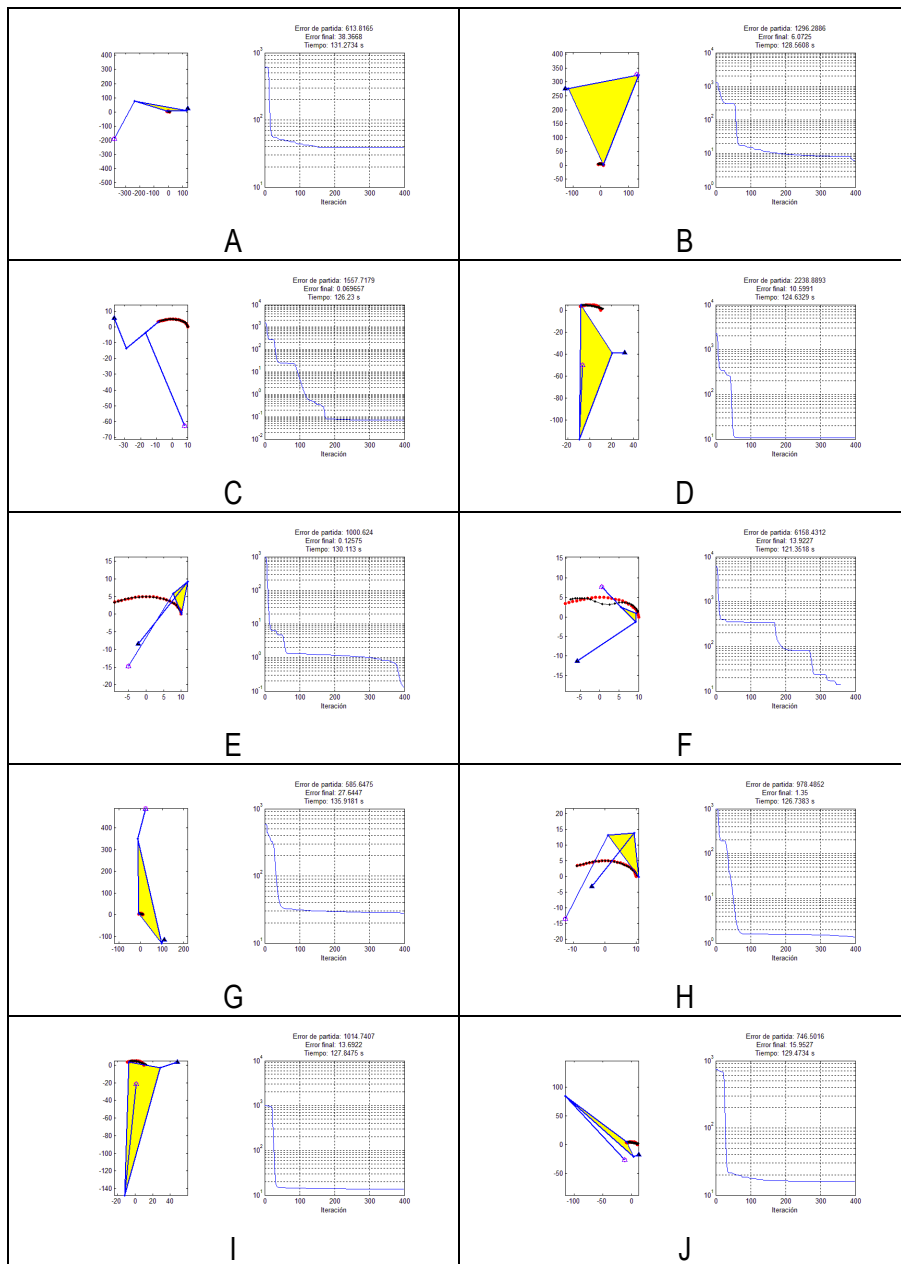


Fig. 4.7 Gráficas de la optimización local para mecanismos de arco abierto

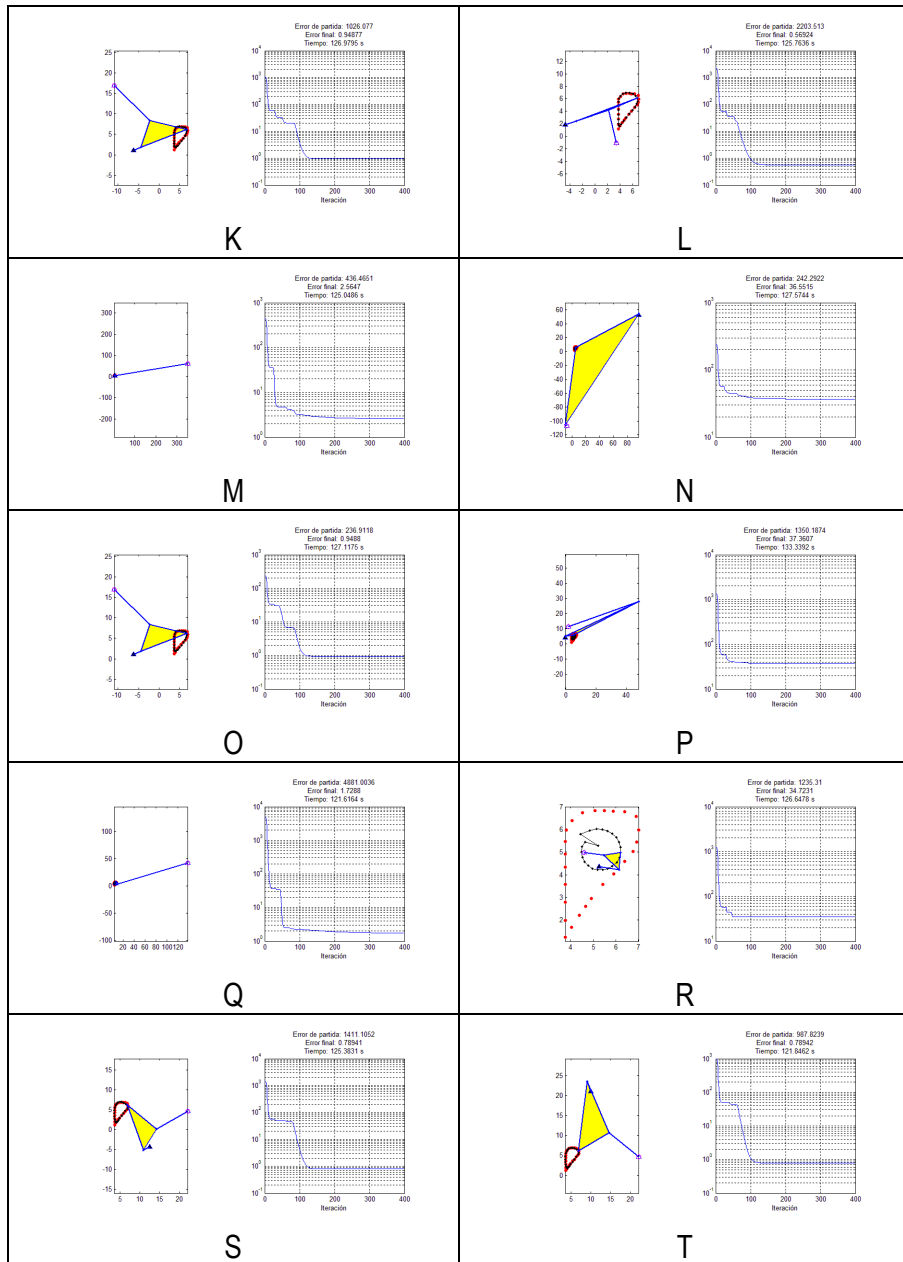


Fig. 4.8 Gráficas de la optimización local para mecanismos de gota de agua

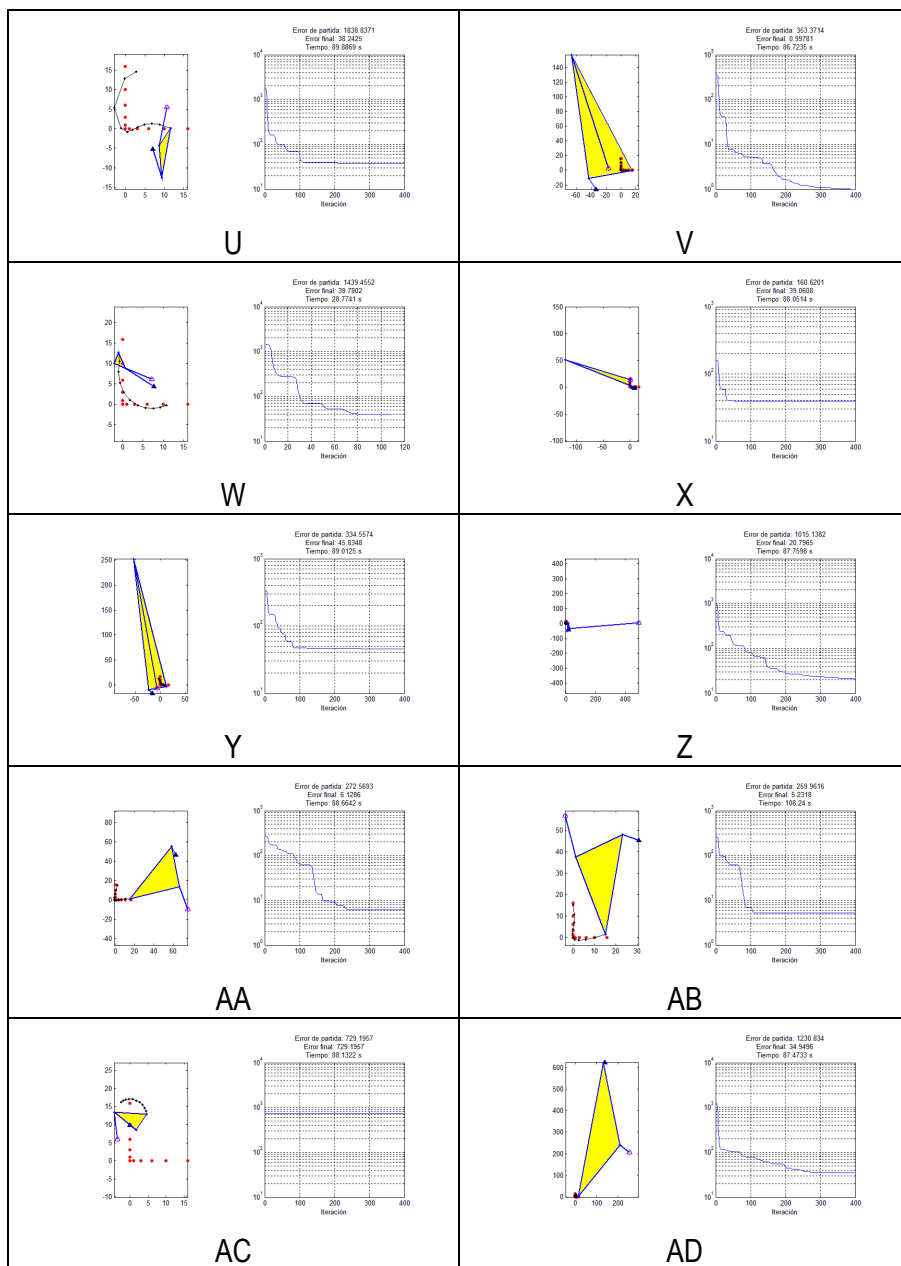


Fig. 4.9 Gráficas de la optimización local para mecanismos de ángulo recto

Como se puede observar de estos ejemplos, el método de optimización propuesto consigue reducciones del error superiores al 95%, y hasta del 100% como para el mecanismo C. Para todas las trayectorias se consiguen mecanismos con un error muy reducido, como muestran los mecanismos C, L y V.

4.5 Conclusiones

La rutina de síntesis local propuesta tiene las siguientes características a destacar. Como primer aspecto, la síntesis local se aborda mediante un proceso de optimización local estructurado en varias fases. Unas primeras fases en las que se modifica la posición, orientación y escala del mecanismo, para posteriormente abordar una cuarta fase en la que se optimizan todas las variables de diseño, y por tanto la morfología del mecanismo. Esta división de la optimización en fases obtiene buenos resultados debido a la rápida convergencia de las primeras fases, que permiten además una adecuación del individuo de partida a la trayectoria objetivo antes de comenzar el proceso de optimización morfológica del mecanismo.

En cuanto a la búsqueda de la dirección de descenso en el proceso de optimización, se define la dirección de descenso según la Ecuación (4.13), la inversión de la matriz jacobiana para la obtención de la dirección de descenso ha modificado su formulación clásica, realizando una inversión de dicha matriz mediante la inversa generalizada o pseudoinversa de Moore-Penrose. Este cambio permite obtener mejores resultados aunque el proceso de optimización provoque pasar por individuos que estén cerca de una posición singular, y por lo tanto la matriz a invertir no esté bien condicionada.

Tras calcular la dirección de descenso, se debe modular el paso de optimización. El módulo del vector de cambio de variables de diseño se modifica en este trabajo mediante un algoritmo de paso variable expuesto en la Fig. 4.3 y la Fig. 4.5. Este algoritmo permite acelerar la convergencia del proceso de optimización respecto a otros métodos.

Adicionalmente se ha implementado una restricción durante el proceso de optimización Fig. 4.5, mostrada en la Ecuación (4.19), que permite garantizar el ensamblaje del mecanismo a lo largo del proceso de optimización.

Finalmente se ha mostrado el algoritmo completo en la Fig. 4.5, que describe el algoritmo de síntesis local, y se ha demostrado su validez mediante los ejemplos presentados.

En estos ejemplos se muestra que el método de síntesis local propuesto consigue resultados que reducen el error en más de un 95% respecto al

error inicial medido como distancia media cuadrática, con diferentes trayectorias, dos abiertas y una cerrada, dos de ellas con puntos singulares.

Por otro lado, el coste computacional no es elevado, ya que teniendo en cuenta una limitación del proceso de optimización a 400 iteraciones, el tiempo consumido en el proceso no supera los 140 segundos en un procesador Intel Core i5 (M560 2.53GHz) y por lo general oscila alrededor de los 90 segundos.

Con lo expuesto en este punto, el proceso de síntesis local, podría ser utilizado intensivamente en métodos de búsqueda global, debido a una elevada reducción de error obtenida (>95%), y un coste computacional razonable.

Capítulo 5

“Estoy casi convencido (de forma diametralmente opuesta a mi primera opinión) de que las especies (esto es como confesar un crimen) no son inmutables”

Charles Darwin

5 Síntesis global: síntesis de mecanismos basada en algoritmos evolutivos.

5.1 Introducción

Tras el estudio de estimadores de error y la propuesta del estimador de error TRS, se ha desarrollado un método de síntesis local, que recoge en parte, los aspectos observados en el desarrollo del estimador de error. Además, el método de síntesis local propuesto ha resultado mostrar una gran eficiencia, consiguiendo reducir el error de partida en un número limitado de iteraciones y con un coste computacional reducido.

Sin embargo, y como en todos los métodos locales, existe una gran dependencia entre el individuo de partida y el resultado obtenido tras el proceso de síntesis. En funciones objetivo fácilmente obtenibles por el tipo de mecanismo propuesto, por ejemplo un arco de elipse por un cuadrilátero articulado, el individuo de partida tiene menor importancia que en otras funciones objetivo. Así, si la función objetivo considerada fuese una trayectoria en ángulo recto, la elección de un buen individuo de partida como individuo inicial, o la elección de un individuo con peores características conlleva que los resultados pueden mostrar unas significativas diferencias de error tras ambos procesos de síntesis local. Llevando este razonamiento al extremo, sólo si se selecciona un individuo de partida suficientemente adecuado se podría obtener un mecanismo que cumpliera con la función objetivo incurriendo en un error mínimo. Es decir, si se busca una optimización global del error para cumplir con la función objetivo, no es suficiente el uso de métodos de síntesis local en el diseño de mecanismos.

Para la obtención de mínimos globales, una de las técnicas con mayor difusión en diferentes campos de aplicación son los algoritmos evolutivos, y debido a ello, este capítulo trata sobre los mismos.

Como un primer paso para conocer este tipo de algoritmos, y ver la viabilidad de los mismos para obtener buenos resultados, se ha optado por estudiar el comportamiento del algoritmo evolutivo generalista propuesto por MATLAB®. Tras observar la viabilidad de este tipo de métodos con el algoritmo propuesto por MATLAB®, se ha procedido a desarrollar un

algoritmo propio, con un menor número de opciones, pero más adecuado para la síntesis global de mecanismos.

5.2 Revisión de la herramienta de MATLAB® de métodos evolutivos

La revisión de la herramienta de algoritmos evolutivos propuesta por MATLAB®, se ha realizado mediante pruebas de síntesis de mecanismos, en particular cuadriláteros articulados. Mediante estas pruebas se ha pretendido demostrar la viabilidad de este tipo de métodos para la síntesis de mecanismos, así como estudiar diferentes opciones que permite el algoritmo.

Para entender qué parámetros permite variar el algoritmo de MATLAB®, se presenta en la Fig. 5.1 El esquema básico de un posible algoritmo evolutivo.

Según este esquema, a partir de una población inicial, generada de forma aleatoria, se clasifican los individuos según su aptitud (adaptación a la función objetivo), y en función de esta clasificación se agrupan para generar nuevos individuos según operadores como el operador de cruce o mutación. Así de forma secuencial se van generando individuos cada vez mejor adaptados al entorno propuesto (que se ajusten mejor a la función objetivo impuesta, así como a las posibles restricciones).

Se van a describir a continuación los diferentes operadores básicos. Así, el **operador de selección** clasifica los distintos individuos de una población y les asigna una probabilidad de selección en función de su aptitud, teniendo un valor de probabilidad asignado mayor cuanto mayor sea la aptitud del individuo. Se debe tener en cuenta que el valor de probabilidad asignado debe normalizarse, para que el sumatorio de probabilidades de los individuos de una generación no supere la unidad.

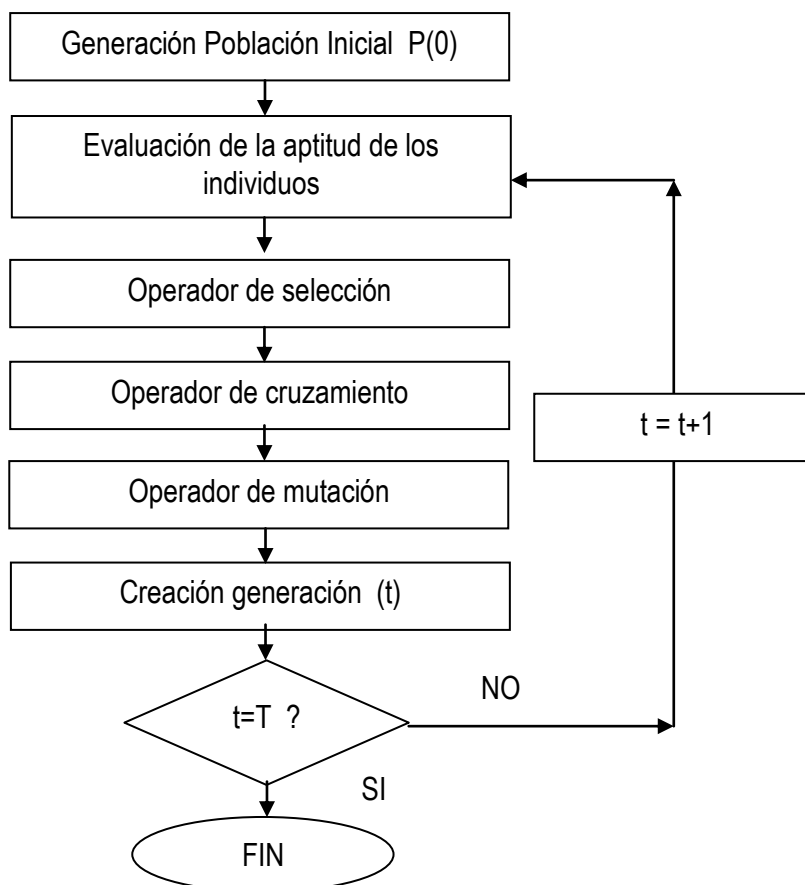


Fig. 5.1 Esquema básico de algoritmo evolutivo

Para el **operador de cruzamiento o cruce**, y con los individuos de la generación y sus valores de aptitud correspondientes, se seleccionan parejas de individuos según una función de probabilidad, por ejemplo una función de Gauss u otro (se utiliza también el método Montecarlo). En esta selección pueden aparecer individuos varias veces, sobre todo los individuos más aptos, que tienen un mayor valor de probabilidad asignado, y que por lo tanto tienen más probabilidades de participar en el operador de cruce. El número de individuos seleccionados para participar en el operador de cruce depende del número de individuos a generar por este operador, y debe ser, como regla general el doble de los individuos a generar.

En el operador de cruzamiento, se toman de forma aleatoria dos individuos de los seleccionados como padres, y se “cruzan” sus cromosomas. Existen varias formas de cruzamiento, desde sustituir parte del cromosoma de uno de los progenitores por los genes del otro progenitor, conocido como cruzamiento de punto único; cruzar los genes uno a uno como media aritmética ponderada; cruzar los genes uno a uno como media geométrica ponderada; etc.

Finalmente y cerrando el primer ciclo de la secuencia, se aplica el **operador de mutación**. Este operador toma individuos de la población inicial y varía de forma aleatoria uno o varios genes. Este operador tiene una probabilidad de actuación, que debe garantizar la diversidad de individuos, para evitar que se converja a una solución o grupo de individuos en una zona del espacio de búsqueda de forma pronta, y que no se evalúen individuos en otras zonas del espacio de búsqueda.

Así la tasa de cruzamiento y mutación, quitando la tasa de élite si es utilizada, deben generar los p (o $p - p_{elite}$) individuos de cada generación, influyendo el valor de estas tasas en la convergencia del problema de forma directa. En algunos métodos estas tasas varían a lo largo de las generaciones, aumentando la tasa de mutación a medida que aumentan las generaciones, para garantizar la diversidad de la población (dentro del espacio de búsqueda).

Para realizar las diferentes pruebas, la primera decisión adoptada, ha sido la codificación el mecanismo mediante genes. Esta codificación podría estar ligada con el tipo de coordenadas a utilizar durante el proceso de optimización. En el caso de algoritmos evolutivos, para realizar pruebas con la herramienta propuesta por MATLAB®, la generación aleatoria de individuos, así como los operadores de selección, cruce y mutación se ha decidido que trabajen con coordenadas cartesianas, aunque para la evaluación de aptitud y análisis de los mecanismos se utilizan coordenadas naturales utilizadas en los métodos de síntesis local, y que se presenta en la Ecuación (5.1) según el esquema de la Fig 3.6.

$$\mathbf{w}^T = [x_0 \ y_0 \ \theta_1 \ \theta_{20} \ L_1 \ L_2 \ L_3 \ L_4 \ L_5 \ \beta] \quad (5.1)$$

A continuación se realizará un breve repaso de las opciones que presenta la herramienta de MATLAB®, agrupando estas opciones en 4 grandes

grupos: parámetros generales, operador de selección, operador de cruce y operador de mutación.

5.2.1 Parámetros generales disponibles en la herramienta de algoritmos evolutivos de MATLAB®

5.2.1.1 Población

Se debe introducir el número de individuos de una población y el rango inicial de cada variable dentro del cual se genera aleatoriamente la población inicial. Generar una diversidad de individuos adecuada mediante el rango inicial es un factor muy importante en el comportamiento del algoritmo genético. Encontrar la cantidad de diversidad adecuada es una cuestión del número de individuos a considerar en cada población, así como de rango de variación de cada una de las variables (genes) que identifican a esos individuos. Para conseguir un ratio de tamaño de población adecuada, se deben realizar pruebas representativas del problema a analizar.

5.2.1.2 Creación de una nueva generación

Se deben fijar diferentes constantes para determinar cómo el algoritmo va a crear la siguiente población. Es decir, se debe proveer al método de un porcentaje de élite, cruzamiento y mutación, que permitan al algoritmo obtener resultados adecuados.

En el caso de querer trabajar con un método elitista, en el que los mejores individuos de una generación pasan a formar parte de la siguiente generación directamente, se debe definir también un parámetro que permita determinar el porcentaje de individuos que se consideren élite ("Elite count").

El porcentaje de individuos a generar con el operador de cruce se define mediante otra variable ("Crossover fraction"). De esta forma la población de la nueva generación contendrá un porcentaje de individuos de élite, otro de cruce, y el resto se generarán mediante mutación.

5.2.1.3 Criterios de parada

El algoritmo proporciona diversos métodos de parada, y es recomendable utilizar varios simultáneamente. Los criterios de parada disponibles son un valor de la función de aptitud, número máximo de generaciones, tiempo máximo de ejecución y número de generaciones sin obtener mejoras en la aptitud de sus individuos.

5.2.2 Parámetros del operador de selección

5.2.2.1 Escalado de aptitud

La función de escalado convierte los valores de aptitud obtenidos de la función objetivo para que sean apropiados para la función de selección de los progenitores que darán la siguiente población, es decir, asigna una probabilidad a cada individuo, que depende de su aptitud, para mediante esta probabilidad determinar las posibilidades de dicho individuo para tomar parte en los operadores de elitismo, cruce o mutación.

Las funciones de escalado disponibles son las siguientes:

- **Rank:** se le asigna un valor de aptitud 1 al mejor individuo, un 2 al segundo, y así sucesivamente. Mediante esta función se consigue eliminar la diseminación de los valores de aptitud dentro de una población.
- **Proporcional:** generando probabilidades de selección proporcionales a los valores de aptitud.
- **Top:** selecciona un número de individuos n , siendo estos n individuos los de mejor aptitud, y le asigna a todos la misma probabilidad $(1/n)$. El resto de individuos no tiene ninguna probabilidad de participar en los operadores de élite, cruce o mutación.
- **Shift linear:** escala los valores de aptitud de forma que la expectativa del mejor individuo es igual a una constante multiplicada por la aptitud media.

5.2.2.2 Selección

La función de selección escoge los progenitores para la siguiente generación basándose en los valores escalados de la aptitud. Se puede introducir una función de selección propia o usar una de las predeterminadas: estocástica, uniforme, remanente (remainder), uniforme, ruleta y torneo.

5.2.3 Parámetros del operador de cruce

Este operador define cómo se deben cruzar dos individuos para crear uno nuevo para la siguiente generación. El algoritmo ofrecido por MATLAB®, presenta las siguientes posibilidades de operador de cruce:

- **Disperso (Scattered):** genera un vector binario con el mismo número valores que el gen que define a un individuo. Así, si se tienen n cromosomas genera un vector de dimensión n que contiene valores 1 o 0, o vector de cruce. Posteriormente toma dos progenitores en función de su aptitud y el operador de selección, y forma un nuevo individuo tomando los cromosomas de uno de los progenitores que tienen asignado un valor 0 en el vector de cruce, y los del otro progenitor los que tienen asignado un valor 1 en el vector de cruce.
- **Pivotamiento de punto fijo:** En este caso se genera un número aleatoriamente entre 1 y el número máximo de genes que definen un individuo. En base a este número se dividen los genes de ambos progenitores por el cromosoma determinado mediante el número aleatorio, en dos partes. El nuevo individuo se generará con una primera parte de uno de los progenitores y una segunda del otro, encadenando los fragmentos generados.
- **Pivotamiento de dos puntos:** Este caso es similar al anterior pero se fragmenta el cromosoma de cada progenitor en 3 partes,

generando el nuevo individuo mediante intercambio de las partes de los progenitores.

- **Media ponderada:** Para definir un gen del nuevo individuo se calcula la media ponderada del mismo gen de los progenitores como se muestra en la Ecuación (5.2). En el parámetro λ (*ratio*) se introduce el peso de cada uno de los progenitores, y se afecta por un número aleatorio entre 0 y 1, κ , tal y como se muestra en la Fig. 5.2.

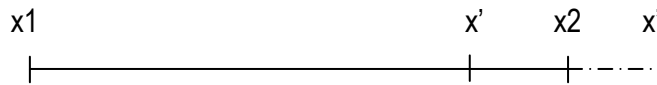


Fig. 5.2 Operador Cruce de media ponderada

El superíndice de los individuos marca el número de generación y el subíndice marca el número de individuo dentro de la generación.

$$\mathbf{w}_j^{i+1} = \mathbf{w}_n^i + \kappa \cdot \lambda \cdot (\mathbf{w}_m^i - \mathbf{w}_n^i) \quad (5.2)$$

Si $\lambda = 1 \rightarrow$ el nuevo individuo estará en algún lugar entre ambos progenitores. Es igual que la opción aritmético.

Si $\lambda > 1 \rightarrow$ el nuevo individuo puede caer fuera del espacio entre los progenitores en la dirección al segundo.

Si $\lambda < 1 \rightarrow$ el nuevo individuo estará más cerca del primer progenitor y nunca será el segundo.

- **Heurístico:** el nuevo individuo se genera operando gen a gen con los progenitores. Los genes del nuevo individuo tomarán valores cercanos a los presentados por los genes del progenitor con mejor aptitud según un ratio. No dependiendo esta cercanía o lejanía del progenitor de ninguna variable aleatoria, tal y como se muestra en la Fig. 5.3.

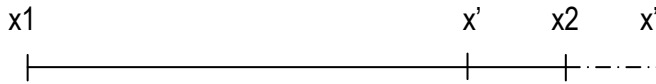


Fig. 5.3 Operador Cruce heurístico

Si $\lambda = 1 \rightarrow$ el nuevo individuo es igual al mejor progenitor (*Progenitor x2*).

Si $1 > \lambda > 0.5 \rightarrow$ el nuevo individuo queda entre los dos progenitores y más cerca del mejor

Si $\lambda > 1 \rightarrow$ el nuevo individuo se sale del espacio entre los dos progenitores en la dirección del mejor

$$\mathbf{w}_j^{i+1} = \mathbf{w}_n^i + \lambda \cdot (\mathbf{w}_m^i - \mathbf{w}_n^i) \quad (5.3)$$

- **Aritmético:** Mediante este método cada gen del nuevo individuo se define mediante la media aritmética ponderada con un número aleatorio de los valores de dicho gen en sus progenitores.

$$\mathbf{w}_j^{i+1} = \lambda \cdot \mathbf{w}_n^i + (1 - \lambda) \cdot \mathbf{w}_m^i \quad (5.4)$$

- **A medida (Custom):** Se puede generar una función de cruce que genere un nuevo individuo como requiera el usuario.

5.2.4 Parámetros del operador de mutación

El operador de mutación trabaja generando pequeños cambios aleatorios en los genes de los individuos seleccionados para mutar. La mutación proporciona diversidad genética y posibilita que el algoritmo busque en un espacio más amplio.

La opción de distribución **Gaussiana** es la opción por defecto. Añade un número aleatorio escogido de una distribución normal para cada gen del individuo. La cantidad de mutación es proporcional a la desviación estándar, controlada por el parámetro 'Scale' (desviación estándar en la primera generación). Típicamente, la cantidad de mutación disminuye en cada nueva generación. Esto se controla mediante el parámetro 'Shrink'. Cuando es 0, la varianza se mantiene constante. Cuando es 1 disminuye linealmente hasta 0 cuando se alcanza la última generación.

Otras opciones de distribución son uniforme, adaptativa factible, y custom.

5.3 Definición de pruebas y resultados con la herramienta de algoritmos evolutivos de MATLAB®

Como primer paso, para comprobar el funcionamiento de estos métodos, y la posibilidad de encontrar soluciones cercanas a un óptimo global con un coste computacional razonable, se ha propuesto elegir la trayectoria objetivo como la trayectoria descrita por un mecanismo conocido. En estas primeras pruebas se ha usado el método evolutivo para determinar únicamente las variables de diseño que marcan la orientación del mecanismo, θ_0 , y la posición en el espacio del mecanismo, (x_0, y_0) . Es decir, conocido el mecanismo óptimo para generar una trayectoria se trata de posicionar ese mecanismo espacialmente mediante traslación y giro para obtener así la trayectoria buscada.

5.3.1 Prueba 1: Convergencia de los algoritmos evolutivos

Esta primera prueba trata de constatar si los algoritmos evolutivos son capaces de converger hacia una solución que sea un óptimo global de la función objetivo.

Con tal fin se parte de un mecanismo conocido de dimensiones y posición mostradas en la Tabla 5.1:

L1	L2	L3	L4	L5	alfa	θ_0	x_0	y_0
10	20	30	25	15	$\pi/6$	π	0	0

Tabla 5.1 Mecanismo seleccionado para generar trayectoria objetivo

Se evalúa la trayectoria en 9 puntos. Para probar el algoritmo genético se introducen como objetivo los puntos calculados de la trayectoria y se trata que se encuentre únicamente x_0 e y_0 , fijando los demás parámetros mediante los límites inferior y superior, tal y como se muestra en la Tabla 5.2.

Rango población inicial	[10 20 30 25 15 $\pi/6$ π -
-------------------------	---------------------------------

	100 -100; 10 20 30 25 15 pi/6 pi 200 200]
Límites inferior y superior	[10 20 30 25 15 pi/6 pi - 100 -100; 10 20 30 25 15 pi/6 pi 200 200]

Tabla 5.2 Rango de variación de las variables que definen el mecanismo

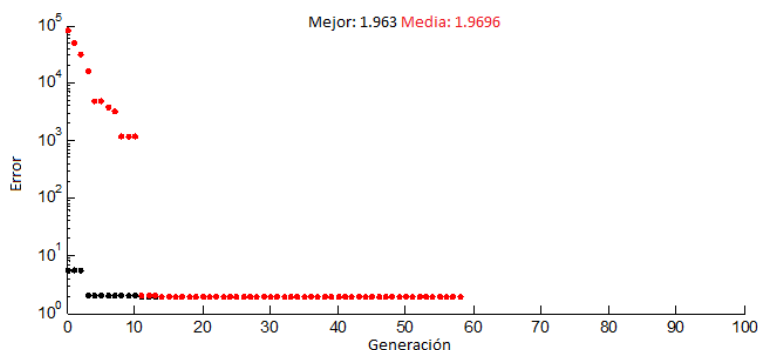


Fig. 5.4 Evaluación del error en Prueba 1

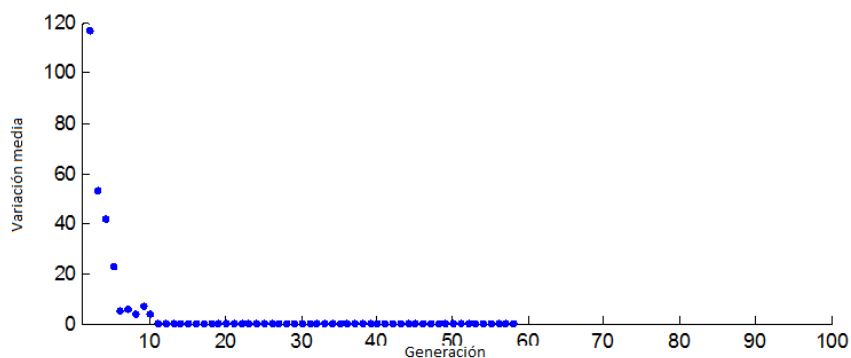


Fig. 5.5 Variación de mejor individuo entre generaciones

El algoritmo finaliza a las 58 generaciones porque no consigue reducir el error en sucesivas generaciones, tal y como se muestra en la Fig. 5.4 y Fig. 5.5.

Se llega al punto ($x_0=-0.314$, $y_0=0.582$) con un error de 1.969. Un error reducido y una solución muy similar al mecanismo óptimo y conocido.

En torno a 10 generaciones la media de aptitud se equipara mucho a la del mejor individuo. También se ve que la distancia media entre individuos se hace muy pequeña rápidamente.

En este caso, parámetros por defecto como el tipo de cruce (disperso), tipo de mutación, tasa de cruce y mutación, etc., son los propuestos por defecto en la herramienta, que no tienen por qué ser los óptimos para el tipo de problema abordado.

5.3.2 Prueba 2: Estudio de proporción de cruce y mutación

Se ha estudiado a continuación el efecto de cambiar la proporción cruce-mutación.

Los parámetros del programa seleccionados son los siguientes:

```
options.Generations = 3000;  
options.TimeLimit = 10;
```

A pesar del alto número de generaciones establecido, sólo se alcanzan 8 generaciones debido al límite de tiempo, influye en la cantidad de mutación. Por defecto, se disminuye linealmente la varianza de la distribución normal a medida que pasan las generaciones, hasta 0 para la última. Por lo tanto la cantidad de mutación apenas disminuye en este caso.

En el siguiente gráfico se muestran los resultados de 10 algoritmos diferentes para cada uno de los 6 niveles de cruce probados (en abscisas, 0, 0.2, 0.4, 0.6, 0.8 y 1). El color del gráfico indica el valor de la aptitud en cada caso.

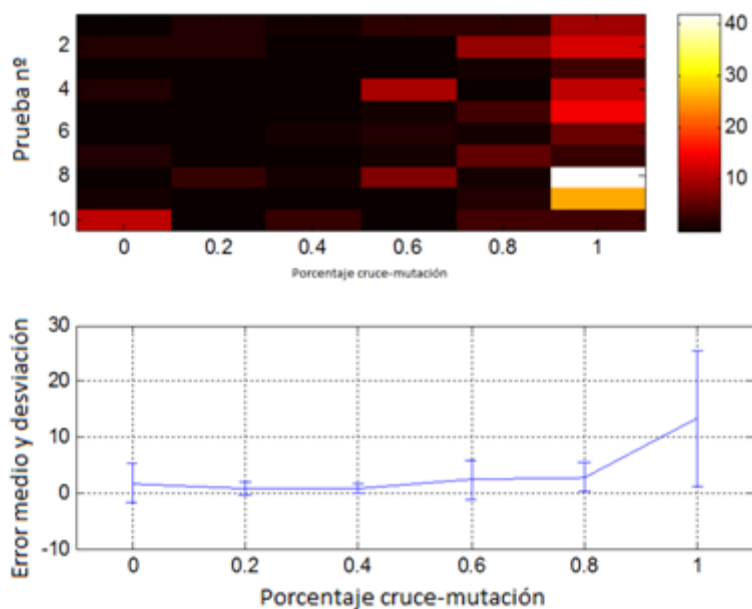


Fig. 5.6 Pruebas de tasa cruce-mutación (abcisas variación de la mutación, 0 no varía la tasa, 1 se elimina la mutación en la última generación; en ordenadas número de prueba)

En los 60 casos el algoritmo termina porque se excede el tiempo límite fijado (10 s por cada algoritmo).

En estas condiciones, parece que los resultados empeoran drásticamente cuando no hay mutación (CrossoverFraction=1). Aún así y viendo que el número de generaciones es reducido se repiten las 60 optimizaciones aumentando el tiempo límite de optimización a 20s.

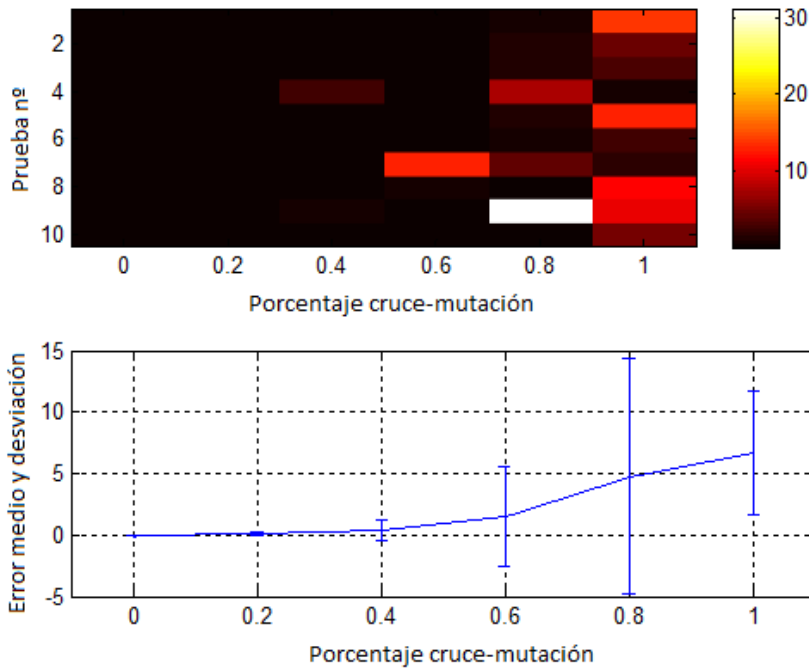


Fig. 5.7 Pruebas de tasa cruce-mutación (abcisas variación de la mutación, 0 no varía la tasa, 1 se elimina la mutación en la última generación; en ordenadas número de prueba), con mayor tiempo límite por prueba

Con mayor tiempo para ejecutar los algoritmos se consiguen resultados con menor error, pero se observa la misma tendencia, un efecto negativo con la reducción de tasa de mutación y utilizando cruce Disperso (*Scattered*) (opción por defecto). Puede que este tipo de cruce no sea adecuado para este problema, como muestra la Fig. 5.7.

Otra prueba más se ha realizado eliminando el número límite de generaciones, obtenido los resultados mostrados en la Fig. 5.8.

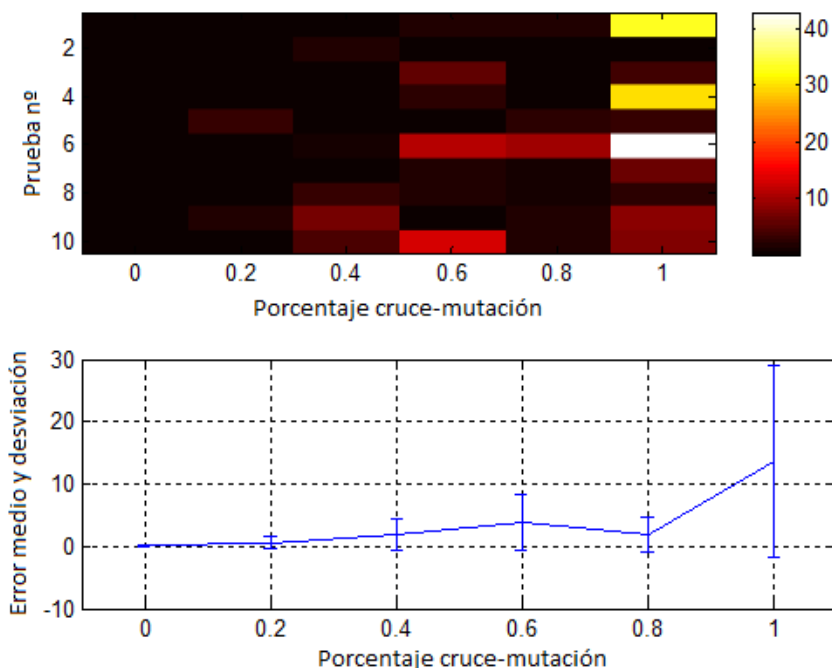


Fig. 5.8 Pruebas de tasa cruce-mutación (abcisas variación de la mutación, 0 no varía la tasa, 1 se elimina la mutación en la última generación; en ordenadas número de prueba), eliminando el límite de generaciones

Tras estas pruebas se puede observar que mantener una tasa de mutación a lo largo de las generaciones proporciona mejores resultados que ir reduciendo la tasa de mutación según aumentan las generaciones. Esto podría tener su origen en que el cruce conlleva a una rápida convergencia hacia una solución como “mezcla” de los individuos más aptos de la generación inicial, pero con una pobre exploración del espacio de trabajo. Pudiéndose ampliar esta exploración mediante el mantenimiento de la tasa de mutación.

5.3.3 Prueba 3: Influencia del tipo de cruce

Realizando la misma prueba que la prueba 2, pero modificando el operador de Cruce por la opción de cruce aritmético, se pretende estudiar cómo afecta el operador de cruce a la tasa de mutación.

Con las mismas condiciones que en el primer estudio de la prueba 2, se sustituye el cruce Scattered por el aritmético, obteniéndose los resultados mostrados en la Fig 5.9.

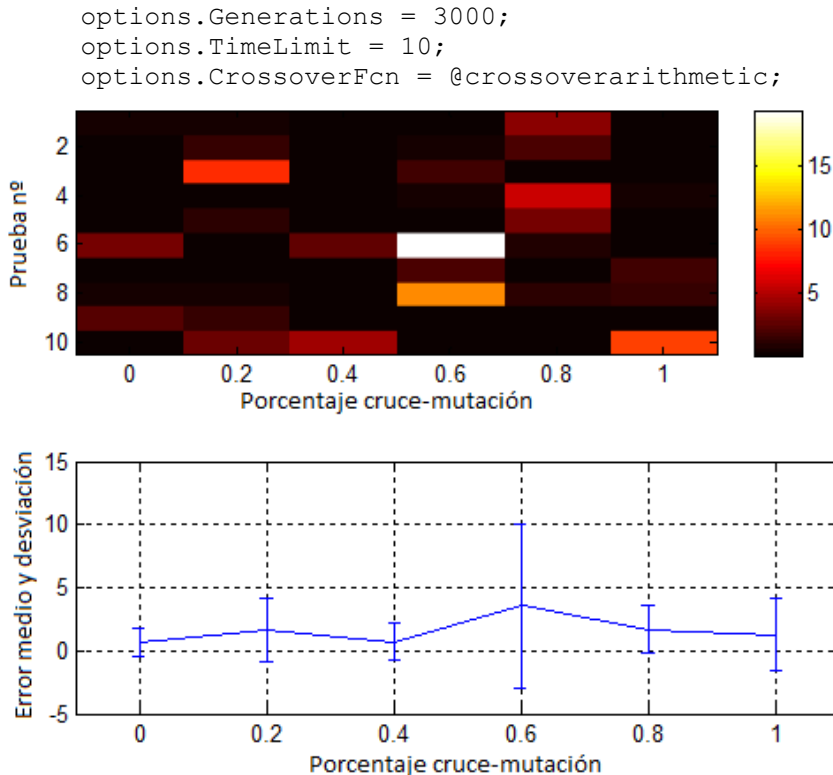


Fig. 5.9 Estudio de la influencia de la tasa de cruce-mutación con operador de cruce aritmético.

El algoritmo también termina por el límite de tiempo. En la Fig. 5.9 no se observa una tendencia clara en cuanto a la proporción cruce/mutación con el tipo de cruce aritmético. Se mejoran los resultados de aptitud, ligeramente menores. Parece que este cruce da mejores resultados que el cruce disperso (*scattered*).

5.3.4 Prueba 4: Influencia del cruce comparado con el tamaño de la población

Se pretende analizar el efecto del tamaño de la población inicial. Se espera que con mayor número de individuos se encuentre una mejor solución. Se

ha establecido un límite de generaciones de 20 y se han hecho 10 pruebas por cada número de población de las que se muestran los resultados en la Fig. 5.10, Fig. 5.11 y Fig. 5.12.

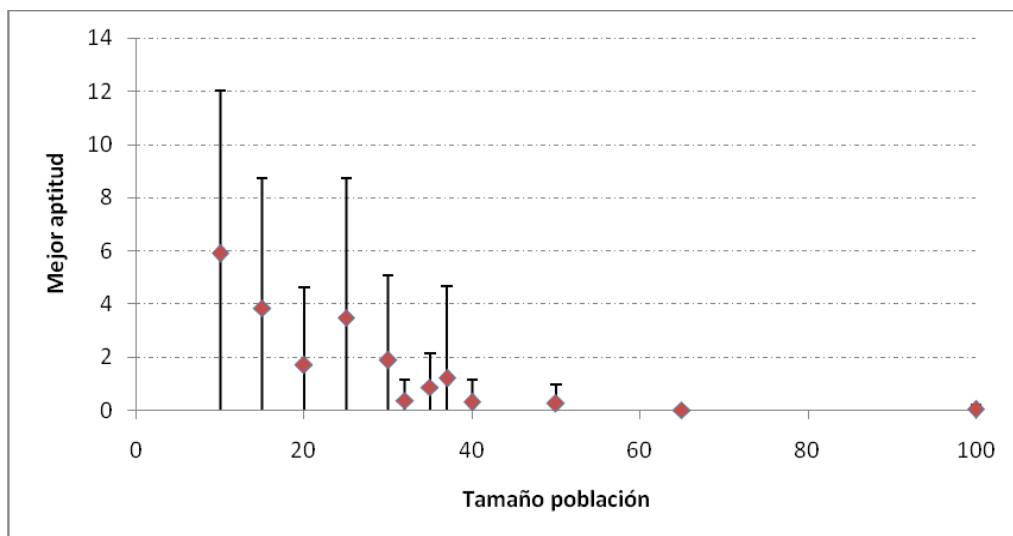


Fig. 5.10 Error según el tamaño de la población con Cruce "Scattered"

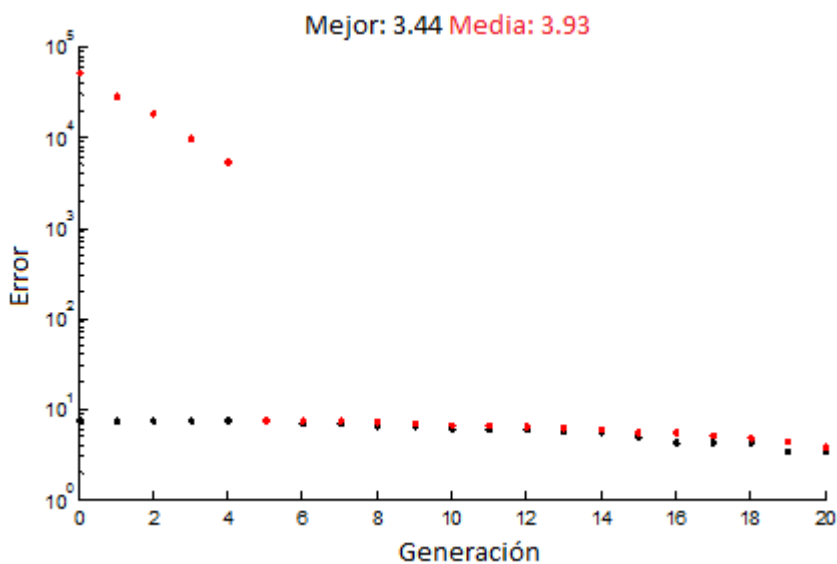


Fig. 5.11 Evolución del error medio y mínimo para población de 10 individuos por generación

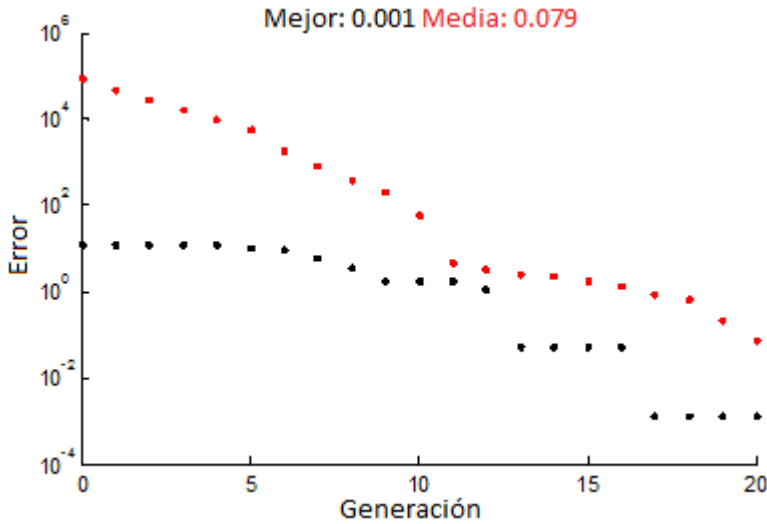


Fig. 5.12 Evolución del error medio y mínimo para población de 50 individuos por generación

Como se esperaba, al aumentar la población se mejora la aptitud y se reduce la desviación de los resultados, sobre todo a partir de 40 individuos. En los dos ejemplos mostrados se ven las diferencias en la evolución de la mejor aptitud y aptitud media. Con menor población la evolución del mejor individuo se queda estancada antes (ver Fig. 5.12).

En la Fig. 5.13 se muestra un análisis similar al anterior pero utilizando el cruce aritmético.

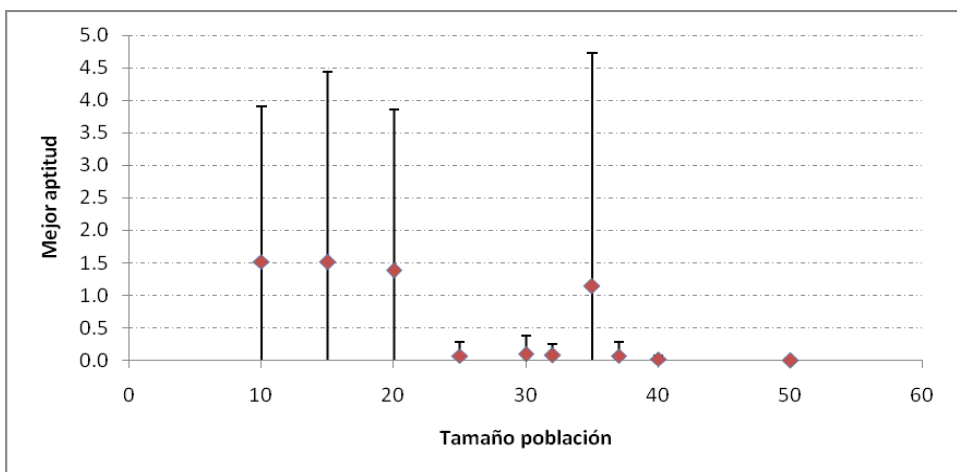


Fig. 5.13 Efecto del tamaño de la población. Cruce aritmético

Los resultados obtenidos son mucho mejores que los obtenidos con cruce disperso ("Scattered"). Como se veía anteriormente el cruce aritmético es más eficaz en el caso estudiado que el cruce disperso. Además parece que hace falta menos población para obtener buenos resultados.

En el caso de 35 individuos se han obtenido malos resultados. Esto es debido a que uno de los 10 algoritmos ha obtenido una aptitud mucho peor que el resto, lo que ha elevado la media y la desviación estándar. Estos casos no son frecuentes pero se dan de vez en cuando.

5.3.5 Prueba 5: Influencia de la tasa cruce mutación con tamaños de población mayores, cruce aritmético y 20 generaciones como máximo

Visto el efecto del tamaño de población y del tipo de cruce, se va a repetir ahora el análisis del efecto de la proporción cruce/mutación con población de 35 individuos, cruce aritmético y limitando el algoritmo a 20 generaciones. Se espera ver más clara la influencia de este parámetro para este tipo de cruce, mostrándose los resultados en la Fig. 5.14 y Fig. 5.15.

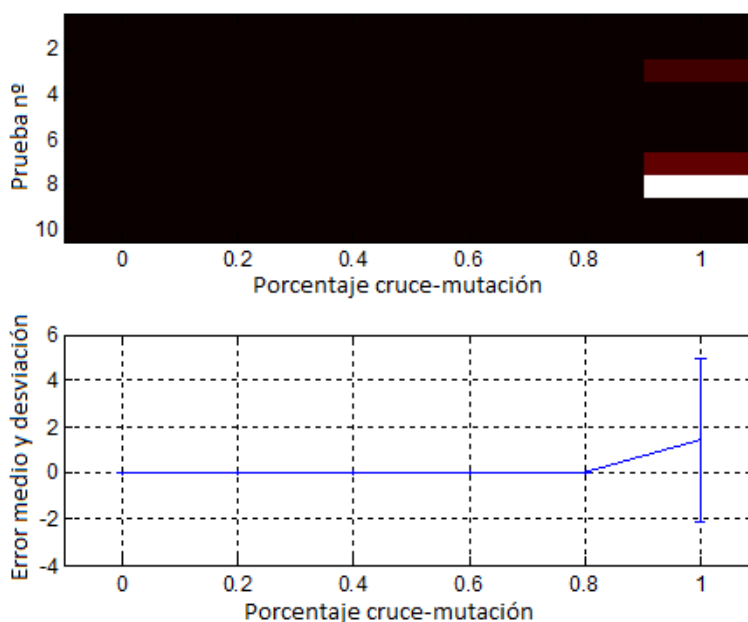


Fig. 5.14

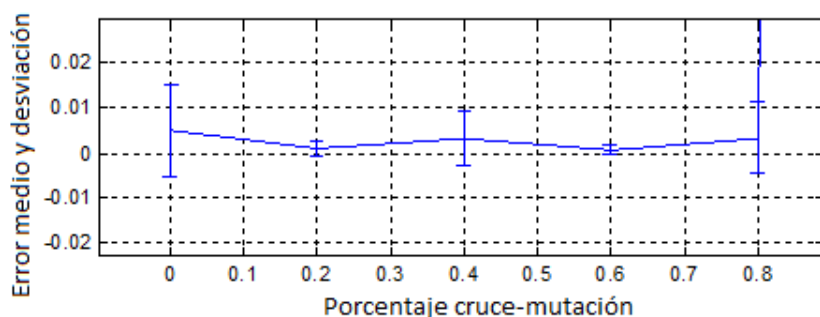


Fig. 5.15 Zoom de la figura anterior

Se obtienen los peores resultados cuando no existe mutación alguna. Para todas las proporciones, incluso sin cruce, el error es bajo. No se ve ninguna tendencia clara en cuanto a la proporción de 0 a 0.8. La única conclusión que se puede extraer es que parece que es necesario que haya algo de mutación para no obtener malos resultados.

Se ven similares resultados probando la mutación uniforme como se muestra en la Fig. 5.16 y Fig. 5.17.

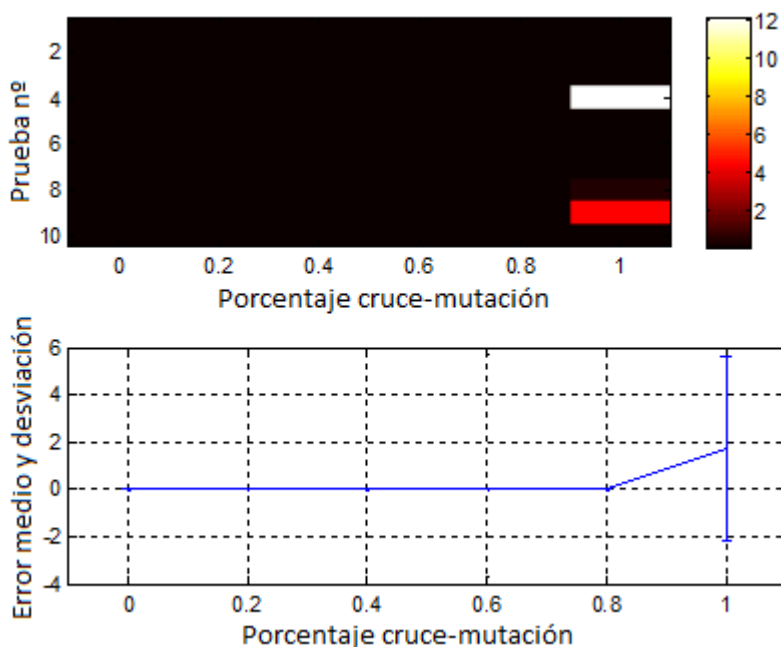


Fig. 5.16

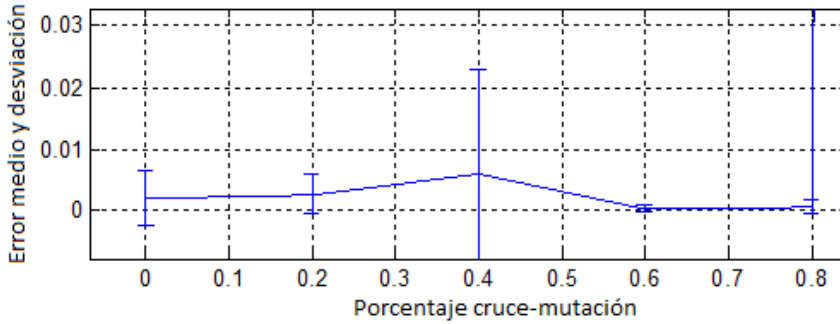


Fig. 5.17

5.3.6 Prueba 6: Media geométrica vs aritmética

Se propone usar la media geométrica en lugar de la aritmética para el cruce. Al no estar incluida se pretende modificar la función `@crossoverarithmetic` incluida en MATLAB®. Se observa que la media aritmética es ponderada y se hace con números aleatorios como se muestra en la Ecuación (5.4).

Se ha creado la función `@crucegeometrico`. La media geométrica de cada variable a cruzar se hace como se muestra en la Ecuación (5.5):

$$\mathbf{w}_j^{i+1} = \sqrt{\mathbf{w}_n^i \cdot \mathbf{w}_m^i} \quad (5.5)$$

Este tipo de media valdría para las longitudes de barra pero no para posiciones, ya que no se puede usar con valores negativos o nulos.

Se introduce como variable también el ángulo φ_4 . Con el objetivo de fijar un tamaño de población adecuado para realizar las pruebas con los demás parámetros, se hacen 10 pruebas para diferentes tamaños de población. Los algoritmos se limitan a 20 generaciones y se usa el cruce aritmético. Los resultados se muestran en la Fig. 5.18.

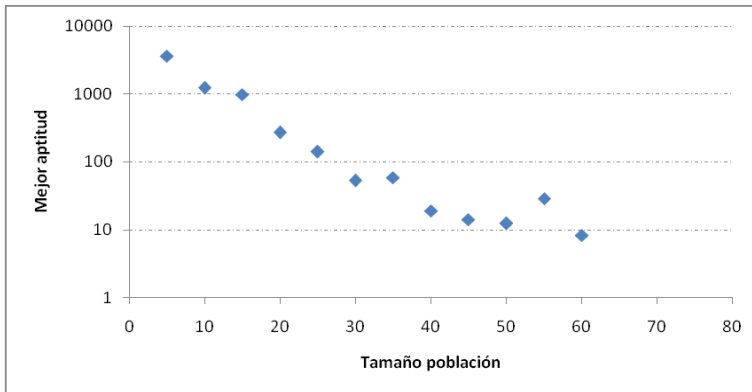


Fig. 5.18

Como es lógico el error disminuye aumentando la población. Se determina como tamaño suficiente de población 50 individuos para hacer las siguientes pruebas.

En la Fig. 5.19 y la Fig. 5.20 se muestra un ejemplo con 50 individuos dejando trabajar el algoritmo hasta 100 generaciones.

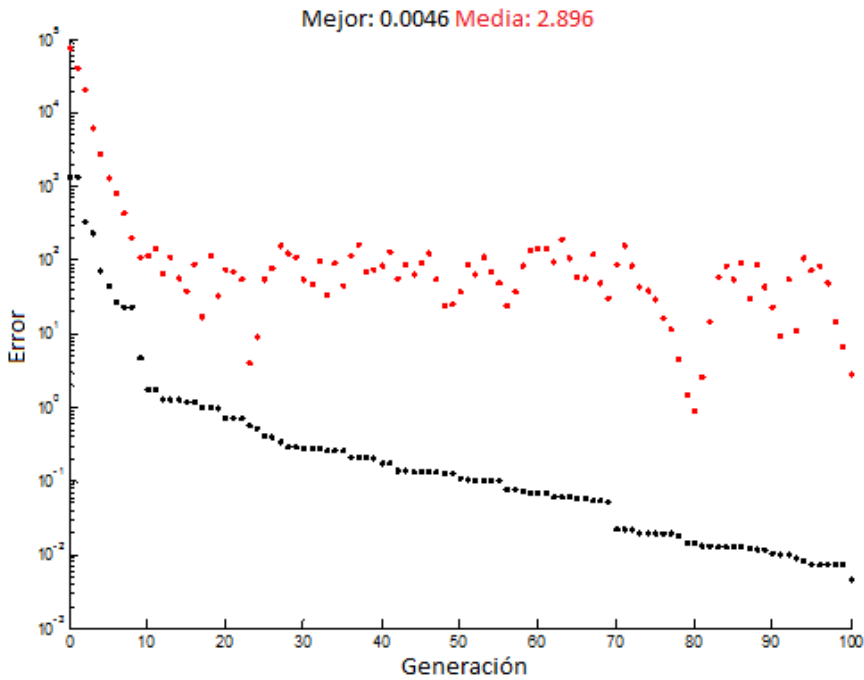


Fig. 5.19

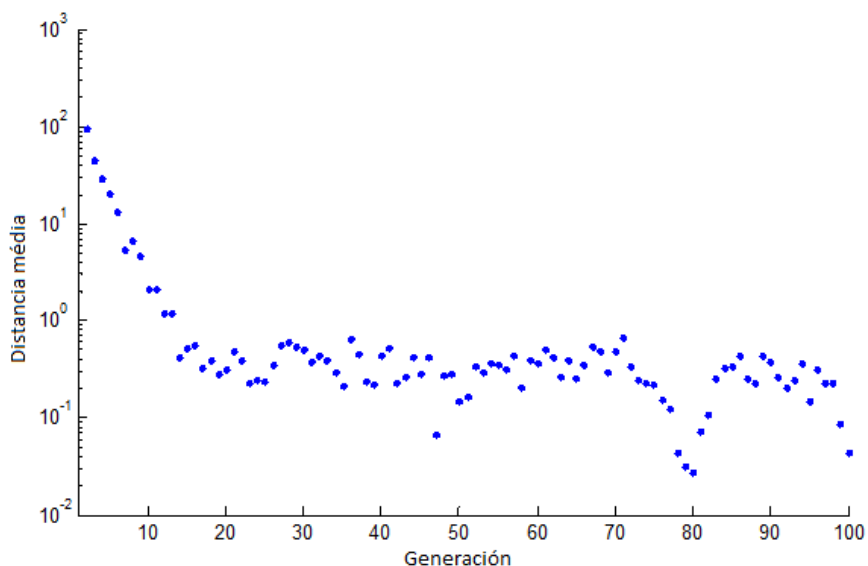


Fig. 5.20

Para limitar el tiempo de las pruebas se harán las siguientes comparaciones de parámetros con 20 generaciones, ya que tal y como se observa en la Fig. 5.19 y Fig. 5.20, la variación de la aptitud a partir de la generación 20 es escasa.

5.3.7 Prueba 7: Tipos de cruce

Con el tamaño de población y generaciones elegidos anteriormente, 50 y 20 respectivamente, se van a probar diferentes tipos de cruce. Se han probado los siguientes:

- Disperso (*Scattered*)
- 1 punto
- 2 puntos
- intermedio, rate=0,7
- intermedio, rate=1,3
- heuristic, rate=0,8
- heuristic, rate=1,2
- arithmetic (media ponderada aleatoriamente)
- aritmético 0,5 (sin ponderar)

En la Fig. 5.21 se muestran los resultados de 10 pruebas con cada tipo de cruce. Los cruces 1 punto y 2 puntos dan muy malos resultados y no se muestran en la gráfica. La amplia dispersión de los resultados se debe a que en ciertos casos aislados se obtiene un valor de error muy elevado.

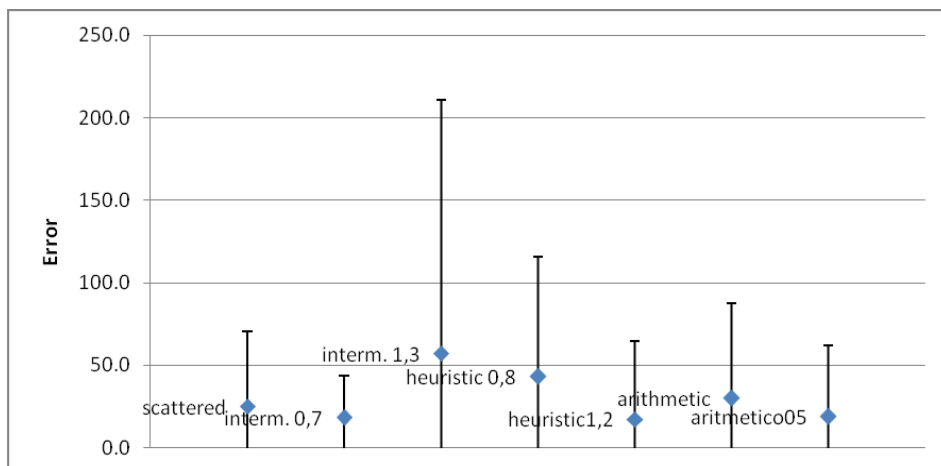


Fig. 5.21 Estudio de dependencia del tipo de operador de cruzamiento.

Las pruebas realizadas permiten vislumbrar que este tipo de métodos puede obtener resultados óptimos desde un punto de vista global, sin necesidad de fijar un individuo de partida. Debido a esto se ha procedido a desarrollar un método propio, tal y como se describe de aquí en adelante.

5.4 Método evolutivo desarrollado para la síntesis dimensional de mecanismos

5.4.1 Programación

Con las pruebas realizadas en la toolbox de algoritmos genéticos de MATLAB®, se ha planteado una programación específica enfocada a la resolución del problema de síntesis de mecanismos.

La rutina propuesta estudia distintos individuos dentro de un mismo tipo de mecanismos. Así, ante una trayectoria objetivo, por ejemplo, el programa estudiará distintos individuos del tipo cuadrilátero articulado para conseguir un individuo que obtenga el objetivo propuesto. El problema de tipo se puede solventar haciendo lo mismo para un número determinado de tipos de mecanismos, y eligiendo el de menor error de síntesis como óptimo.

El programa que se expone a continuación, es el desarrollado para abordar el problema de síntesis con objetivo de trayectoria.

5.4.1.1 Introducción de la trayectoria objetivo

La primera tarea a realizar es la introducción de la trayectoria objetivo. Esta trayectoria se introduce mediante puntos en coordenadas cartesianas mediante tabulación. La trayectoria tendrá tantos puntos de precisión como número de coordenadas se recoja en su definición.

Se debe tener en cuenta que un alto número de puntos de precisión implica un coste computacional mayor, por lo que exceder de una treintena de puntos puede ralentizar el programa. En las pruebas realizadas se han utilizado entre 10 y 25 puntos, obteniéndose costes computacionales razonables.

5.4.2 Definición de parámetros generales del programa

Tras la introducción de la trayectoria se deben determinar los diferentes parámetros del programa. En este caso se debe determinar:

1. Número de individuos por generación
2. Número máximo de generaciones
3. Error objetivo del criterio de parada general de la optimización
4. Porcentaje de élite, mutación y cruce
5. Variación del porcentaje de mutación a lo largo de las generaciones.

Estos valores se ha decidido introducirlos mediante una hoja Excel en forma de fila. De esta forma se pueden programar varias pruebas variando alguno de los parámetros y pudiendo comparar así el comportamiento del algoritmo en función de los parámetros.

5.4.3 Generación de la población inicial

Una vez definidos los distintos parámetros se pasa a generar la población inicial. Esta población se define mediante coordenadas cartesianas. Aunque posteriormente para el análisis cinemático del mecanismo y el

cálculo de su aptitud se traduzcan estas coordenadas cartesianas en coordenadas naturales.

En la rutina de generación de la población se hace un chequeo de validez del individuo para determinar si es o no válido y en función del resultado del mismo se desecha o pasa a formar parte de esta población. El criterio de validez aplicado está relacionado con estudiar cual es el rango de funcionamiento del mecanismo generado, en un cuadrilátero articulado y para una trayectoria cerrada podría evaluarse si se cumplen las condiciones de Grasso. Sin embargo para una trayectoria abierta en un cuadrilátero articulado, o en mecanismos más complejos se estudia el rango de movimiento del mecanismo.

5.4.4 Cálculo de la aptitud de individuos

Con la población inicial definida, se procede a la evaluación de cada uno de los individuos. Para dicha evaluación se debe utilizar un estimador de error. En las primeras programaciones del algoritmo se introdujeron dos tipos de estimador de error: el DMC, distancia media cuadrática entre los puntos de precisión de la trayectoria objetivo y la trayectoria generada por el mecanismo. Y el estimador de error TRS, en el que el mecanismo propuesto se traslada, gira y escala para generar una trayectoria con mayor similitud a la trayectoria objetivo.

Finalmente, se ha utilizado el estimador TRS en la implementación final del programa, ya que los resultados obtenidos con este tipo de estimador han sido mucho mejores que los obtenidos con el estimador DMC como se mostrará en adelante.

5.4.5 Escalado de la aptitud: probabilidad de un individuo de participar en operadores de cruce o mutación

Tras evaluar la aptitud de todos los individuos, estos se clasifican en función de sus valores de aptitud y la función de escalado. En una primera programación se han utilizado dos tipos de funciones de escalado posibles, normalizado de la aptitud en función del error obtenido por cada individuo, o mediante un escalado normal geométrico.

Tras ordenar los individuos según su valor de error (aptitud), obtienen un identificador en dicha clasificación desde 1 hasta n. Con este valor se define la expectativa de que el individuo participe en un proceso de cruce o mutación para generar la nueva generación tal y como se muestra en la Ecuación (5.6).

$$expectativa = \frac{1}{\sqrt{\text{identificador}_{\text{clasificación}}}} \quad (5.6)$$

Tras definir este valor de expectativa, se normalizan los diferentes valores de expectativa según se muestra en la Ecuación (5.7).

$$expectativa_{\text{normalizada}_i} = \frac{expectativa_i}{\sum_1^n expectativa_i} \quad (5.7)$$

La función de expectativa normalizada descrita proporcionaría, para una población de 10 individuos, una distribución de probabilidad de participar en un cruce o mutación como el mostrado en la Fig. 5.22, donde en abscisas se muestra la clasificación de cada individuo, y en ordenadas su probabilidad de participar en un proceso de cruce o mutación.

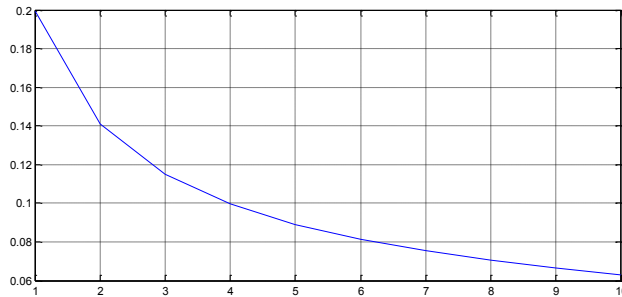


Fig. 5.22 Función expectativa para una población de 10 individuos

5.4.6 Selección de padres

En este punto se debe seleccionar qué individuos de la generación de padres van a participar en la función de cruce o mutación. Para tal fin se utiliza una función que elige los índices de la población que van a participar. En esta función se genera una variable ruleta. Esta variable se genera a partir del vector de expectativas, sumando para la fila i los valores del vector de expectativas desde 1 hasta i, teniendo i un valor máximo igual al del número de individuos de la población cero. Tras esto se define la probabilidad lineal de ser elegido, y se multiplica por un

número aleatorio entre 0 y 1, en la variable posición. A continuación se entra en un doble bucle que recorre dos índices, el primero indica el número de padres a generar desde 1 hasta el número de padres totales, y el segundo recorre los índices de la variable ruleta, desde el valor inferior, variable *lowest* (siendo este valor 1 en la primera iteración), hasta el número de individuos por población, buscando un valor en la misma superior al de la variable posición. Si se cumple esta condición, el índice de la generación cero que marque será elegido como padre de la futura generación, y la variable *lowest* adquirirá el mismo valor del índice elegido. Finalmente, mediante el método descrito (Fig. 5.23), los padres salen ordenados por índice, y se desordenan aleatoriamente.

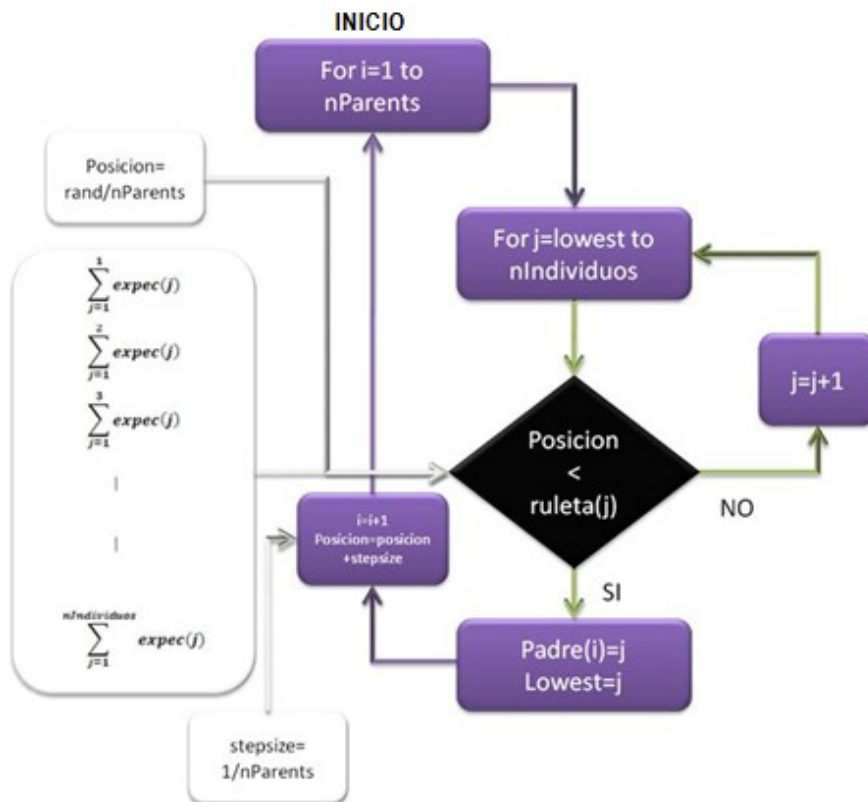


Fig. 5.23 Proceso selección de padres

5.4.7 Operadores para la creación de una nueva generación de individuos

Para obtener la siguiente generación, los individuos más aptos de la generación anterior pasan como élite (operador lamarkiano), estando el resto de la generación formada por individuos procedentes de mutación (operador de mutación) o cruce (operador de cruce). Los individuos procedentes de mutación o cruce toman como referencia individuos procedentes de la generación anterior según la función de escalado.

La nueva generación tendrá un porcentaje de individuos que procede directamente de la generación anterior. Estos individuos son la élite de la generación anterior, los que presentan mejor aptitud. De esta forma se garantiza que a lo largo de las generaciones el valor de la aptitud siempre decrece o permanece con el mismo valor que en la generación anterior, pero no puede aumentar.

Otro porcentaje de la población estará formado por individuos procedentes del operador de mutación. Este operador toma un individuo de la generación anterior, y modifica uno o varios de sus genes de forma aleatoria. Para este operador se programaron en primer lugar varios tipos de mutación, mutación aleatoria de todos los genes, mutación de varios genes seleccionados aleatoriamente, o variación de un solo gen seleccionado aleatoriamente. Tras una serie de pruebas se decidió implementar en el algoritmo final la variación de un solo gen aleatorio del individuo seleccionado.

El resto de la población se genera mediante el operador de cruce. Este operador toma dos individuos de la generación anterior según su aptitud, y genera un nuevo individuo para la nueva generación. Los operadores de cruce son diversos. En la programación inicial del método se han planteado dos operadores de cruce. Un operador de cruce en base a la media aritmética ponderada, otro con la media aritmética ponderada con un rango que puede exceder valores entre los progenitores.

Tras esto se vuelve a evaluar el error de la misma forma, y se sigue iterando de la misma forma hasta agotar el número de generaciones o conseguir un individuo con un error menor o igual al objetivo.

Con el programa descrito se han llevado a cabo distintas pruebas que han permitido fijar los operadores a utilizar, así como las variables globales

para un buen funcionamiento del algoritmo. Las pruebas realizadas se definen en el siguiente punto.

5.4.8 Pruebas realizadas

Se han realizado pruebas con trayectorias abiertas y cerradas y diferentes tipos de cruce o mutación. Cada una de las pruebas se ha repetido para distinto número de individuos en cada generación y varias veces para cada número de individuos. El objetivo de la repetición de las pruebas es poder generar datos suficientes sobre el método para poder tratar los métodos mediante parámetros estadísticos. A continuación se presentan las pruebas realizadas.

5.4.8.1 Prueba 1: operador de mutación con variación de todos los genes.

Esta prueba se ha realizado con una trayectoria objetivo generada por un mecanismo conocido. El número máximo de generaciones es 30, con pruebas de 50, 100, 200 y 500 individuos por generación. El operador de cruce es de media aritmética ponderada, la fracción de élite es del 2%. Se han realizado 10 pruebas para cada número de individuos por generación (40 ejercicios de optimización). El estimador de error utilizado es el de DMC.

La trayectoria objetivo y el mecanismo conocido que la genera se muestran en la siguiente Fig. 5.24.

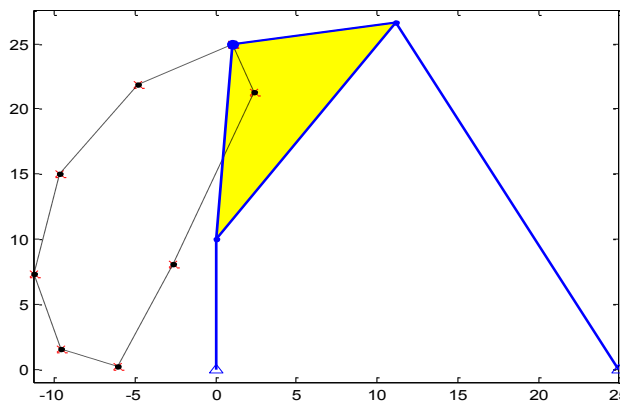


Fig. 5.24 Trayectoria objetivo utilizada y mecanismo que genera dicha trayectoria

En la Fig. 5.25, se muestran los resultados de error conseguido para diferente número de individuos por generación. Se presenta en línea azul la media de error, en rojo el error máximo y en verde el error mínimo para cada serie de ejercicios de síntesis (10 por generación).

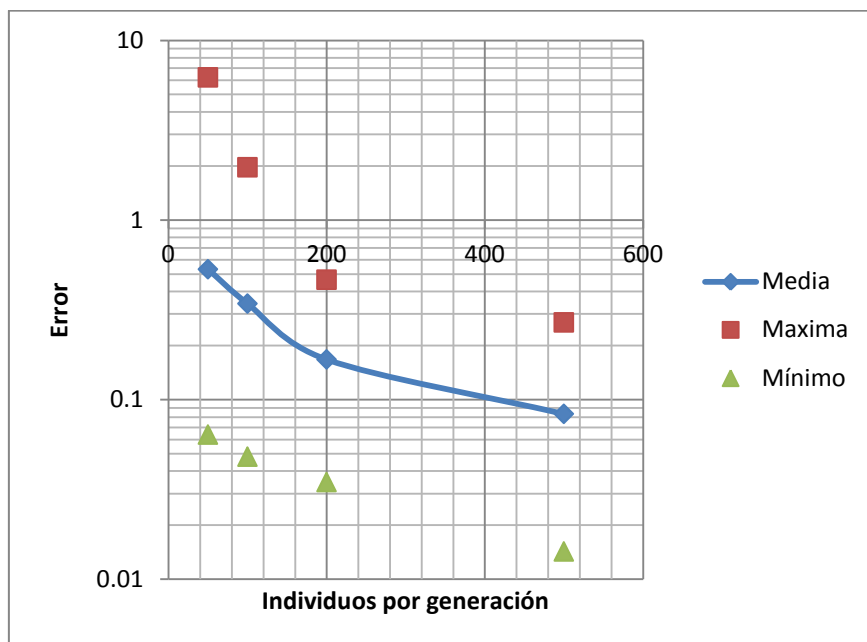


Fig. 5.25 Error medio máximo y mínimo para cada grupo de ejercicio de síntesis, agrupados por número de individuos por generación

En mecanismo resultante de menor error, así como la evolución del error durante el ejercicio de síntesis se presentan en la Fig. 5.26.

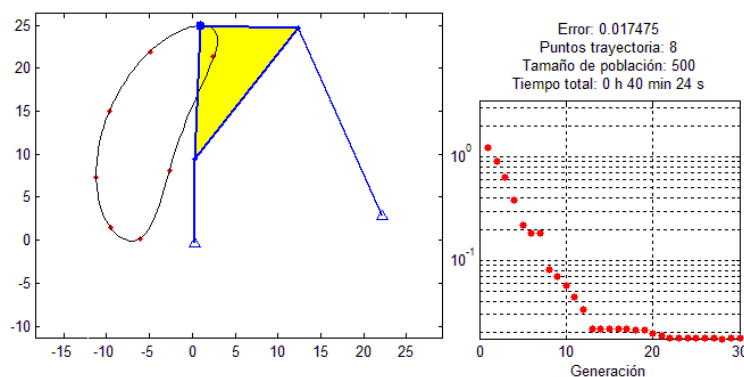


Fig. 5.26 Error mínimo obtenido en la Prueba 1

Los mecanismos y trayectorias obtenidas se muestran en el Anexo A.

5.4.8.2 Prueba 2: operador de mutación con variación de uno de los genes.

Esta prueba se ha realizado con una trayectoria objetivo generada por un mecanismo conocido (mismo que en la prueba 1). El número máximo de generaciones es 30, con pruebas de 50, 100, 200 y 500 individuos por generación. El operador de cruce es de media aritmética ponderada, la fracción de élite es del 2%. Se han realizado 10 pruebas para cada número de individuos por generación (40 ejercicios de optimización). El estimador de error utilizado es el de DMC.

En la Fig. 5.27 se muestran los errores medios, máximos y mínimos de los ejercicios de síntesis realizados en esta prueba.

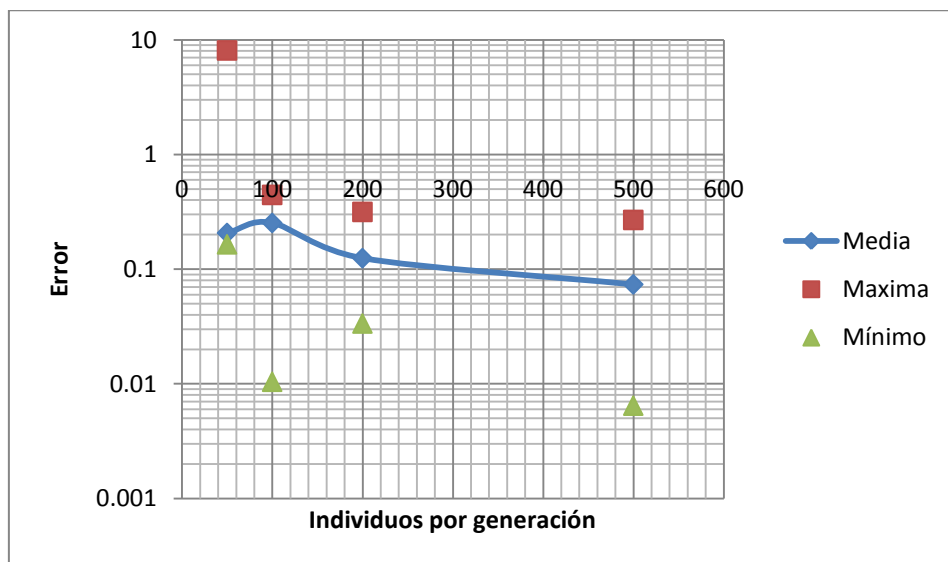


Fig. 5.27 Error medio máximo y mínimo para cada grupo de ejercicio de síntesis, agrupados por número de individuos por generación

El error mínimo obtenido, es de 0.03. El mecanismo que obtiene este error mínimo se muestra en la Fig. 5.28.

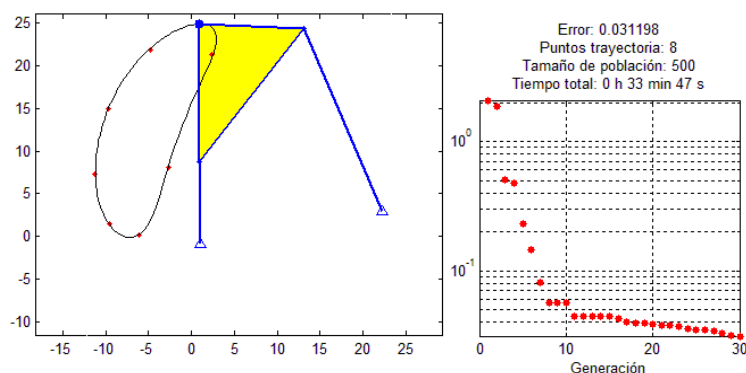


Fig. 5.28 Error mínimo obtenido en la Prueba 2

Los mecanismos y trayectorias obtenidas en la prueba 2 se presentan en el Anexo B.

5.4.8.3 Prueba 3: operador de mutación con variación de dos de los genes.

Esta prueba se ha realizado con una trayectoria objetivo generada por un mecanismo conocido (mismo que en la prueba 1). El número máximo de generaciones es 30, con pruebas de 50, 100, 200 y 500 individuos por generación. El operador de cruce es de media aritmética ponderada, la fracción de élite es del 2%. Se han realizado 10 pruebas para cada número de individuos por generación (40 ejercicios de optimización). El estimador de error utilizado es el de DMC.

En la Fig. 5.29 se muestran los errores medios, máximos y mínimos de los ejercicios de síntesis realizados en esta prueba.

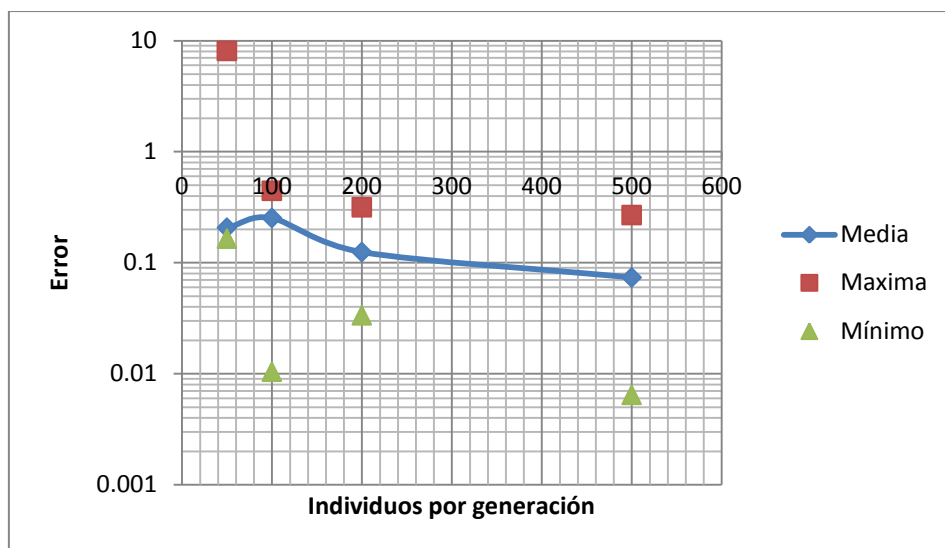


Fig. 5.29 Error medio máximo y mínimo para cada grupo de ejercicio de síntesis, agrupados por número de individuos por generación

El error mínimo obtenido en esta prueba es de 0.006, y se obtiene con el mecanismo mostrado en la Fig. 5.30

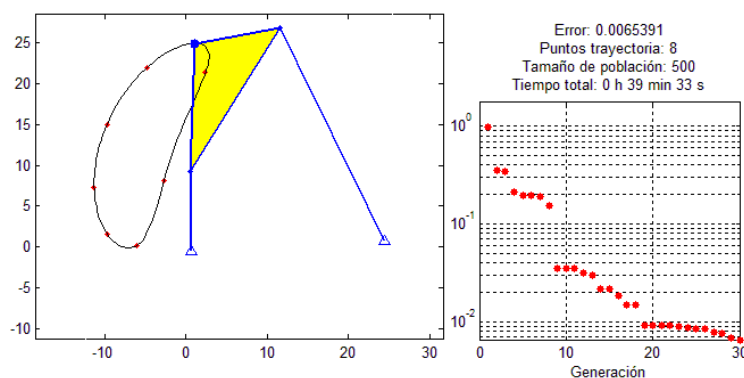


Fig. 5.30 Error mínimo obtenido en la Prueba 3

El resultado de las diferentes pruebas de síntesis en la prueba 3 se presentan en el Anexo C.

5.4.8.4 Prueba 4: operador de cruce de media aritmética ponderada con un rango superior al espacio definido por los progenitores.

Esta prueba se ha realizado con una trayectoria objetivo generada por un mecanismo conocido (mismo que en la prueba 1). El número máximo de generaciones es 30, con pruebas de 50, 100, 200 y 500 individuos por generación. El operador de cruce es de media aritmética ponderada, la fracción de élite es del 2%. Se han realizado 10 pruebas para cada número de individuos por generación (40 ejercicios de optimización). El estimador de error utilizado es el de DMC.

En la Fig. 5.31 se muestran los errores medios, máximos y mínimos de los ejercicios de síntesis realizados en esta prueba.

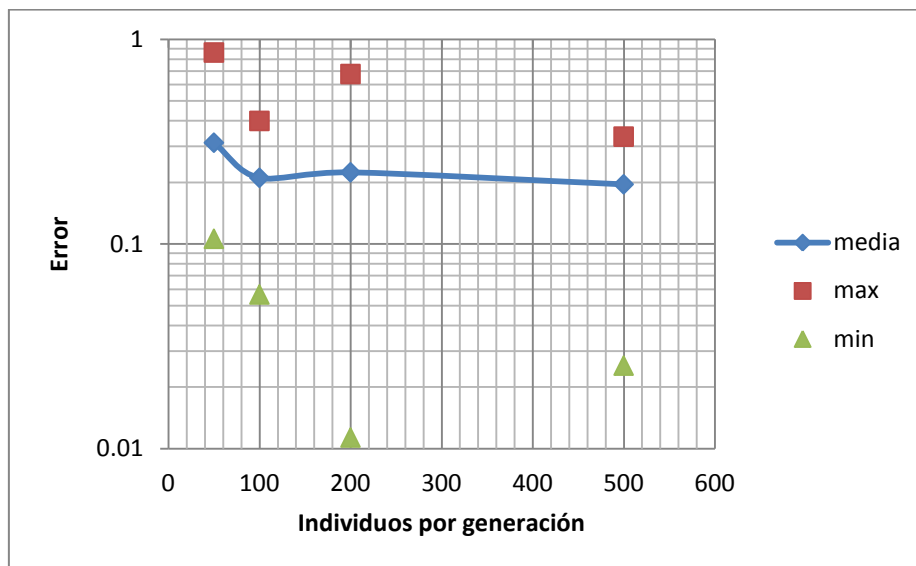


Fig. 5.31 Error medio máximo y mínimo para cada grupo de ejercicio de síntesis, agrupados por número de individuos por generación

El error mínimo obtenido es de 0.01, pero en este caso se presenta para 200 individuos por generación en lugar de para 500. El mecanismo que obtenido con este error y la evolución del error se muestran en la Fig. 5.32.

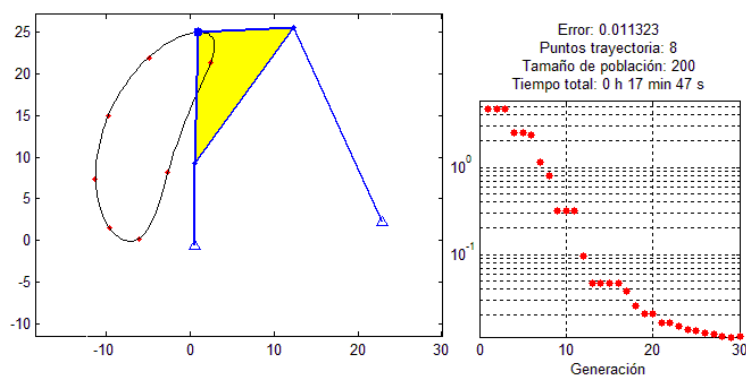


Fig. 5.32 Error mínimo obtenido en la Prueba 4

Los resultados de la prueba 4 se presentan en el Anexo D.

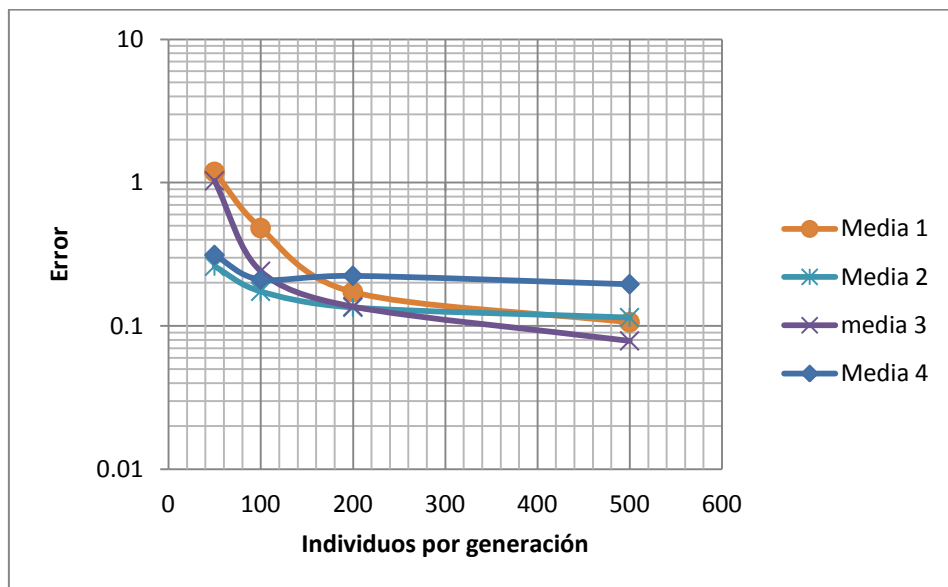


Fig. 5.33 Grafica resumen de pruebas realizadas con operador de cruce y mutación.
Media 1 es el error medio en la prueba 1 y así sucesivamente

5.4.8.5 Resultados Obtenidos

Como se puede ver en la Fig. 5.33, grafica resumen de las anteriores pruebas realizadas, la mejor estrategia, sobre todo utilizando un número de individuos reducido por generación, debería utilizar un sistema de mutación que únicamente varíe uno de los genes del mecanismo. Por lo que el operador de mutación implementado en la programación final será

con mutación de un solo gen por individuo. En cuanto al operador de cruce, la media aritmética ponderada da buenos resultados que justifican el no utilizar esta técnica con rango ampliado. Así, finalmente se ha implementado un operador de cruce con media aritmética ponderada.

Con la programación final se ha procedido a evaluar una serie de ejemplos, en base a diferentes trayectorias que se presenta a continuación.

5.5 Ejemplos del método programado con 4 trayectorias diferentes.

Los ejemplos se presentan para dos posibles alternativas del algoritmo, con estimador de error DMC (Método 0) y con estimador de error TRS (Método 1 o también AH1 en el próximo capítulo, algoritmo híbrido propuesto 1) que mejora la aptitud del mecanismo optimizando su posición, giro y tamaño

En cuanto al número de individuos por generación en cada uno de los métodos, se resume en la Tabla 5.3.

Método	0	1
50 ind/gen	X	X
100 ind/gen	X	X
200 ind/gen	X	X
500 ind/gen	X	X

Tabla 5.3

Desde un punto de vista de trayectorias probadas, se pueden clasificar en trayectorias abiertas, generando un ángulo recto como trayectoria objetivo, trayectoria ya evaluada en otras publicaciones (Sedano, et al., 2012), y trayectorias cerradas, elipse con punto singular, gota de agua y limason, extraídas de diferentes referencias bibliográficas.

La presentación de las pruebas realizadas se ha organizado por trayectoria objetivo, presentando para cada una de las trayectorias datos estadísticos de las pruebas y el mejor resultado obtenido.

5.5.1 Trayectoria L

La trayectoria L utilizada, está compuesta por 11 puntos que se presentan en la Tabla 5.4 en coordenadas cartesianas:

Punto n°	Coordenada <i>x</i>	Coordenada <i>y</i>
1	0	16
2	0	10
3	0	6
4	0	3
5	0	1
6	0	0
7	1	0
8	3	0
9	6	0
10	10	0
11	16	0

Tabla 5.4 Coordenadas de la trayectoria de ángulo recto

Los resultados para los diferentes métodos se presentan en la Tabla 5.5. En primer lugar se presenta una tabla con datos estadísticos del error. Para cada método y un número de individuos por generación se muestran datos de error máximo, mínimo, medio y desviación estándar del error.

Individuos/Generación	Error	Método 0	Método 1
50	max	79.56	20.13
	min	30.55	5.29
	med	47.06	11.08
	std	16.76	5.20
100	max	80.88	15.30
	min	34.26	1.50
	med	56.52	8.08
	std	16.52	4.74
200	max	61.95	7.88
	min	11.31	1.02
	med	31.65	4.00
	std	15.61	2.31
500	max	37.44	1.72
	min	18.28	0.04
	med	25.27	0.96
	std	7.56	0.59

Tabla 5.5 Error máximo, mínimo, medio y desviación estándar para trayectoria L

Junto con los datos de error, se presenta otra tabla en la que se recoge el tiempo medio para cada uno de los ejercicios de síntesis por método e individuos por generación (para un procesador Intel Core i5 (M560 2.53GHz)). Este dato de tiempo pretende mostrar diferencias en el coste computacional de cada uno de los métodos. En la Tabla 5.6 se muestra la media y desviación típica de los datos temporales en minutos.

Individuos/Generación	Tiempo	Método 0	Método 1
50	med	2.62	1.20
	std	0.18	0.10
100	med	5.10	2.21
	std	0.25	0.18
200	med	10.00	4.49
	std	0.80	0.22
500	med	24.79	11.32
	std	1.98	0.68

Tabla 5.6 Tiempo medio y desviación estándar para trayectoria L

Como resumen de ambas tablas se presenta la Fig. 5.35 en la que se ha representado el error medio (máximo y mínimo) para cada uno de los métodos y número de individuos por generación, y en área de marcador el tiempo medio de cada una de las pruebas.

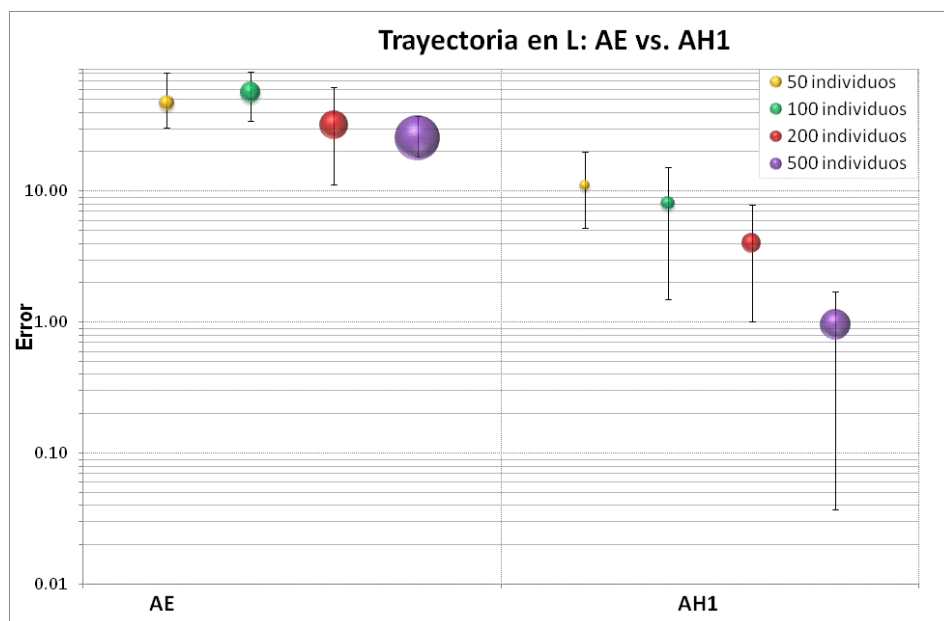


Fig. 5.34 Resultados obtenidos para trayectoria en L

Finalmente se presenta la Fig. 5.35 con el mejor resultado obtenido para esta trayectoria.

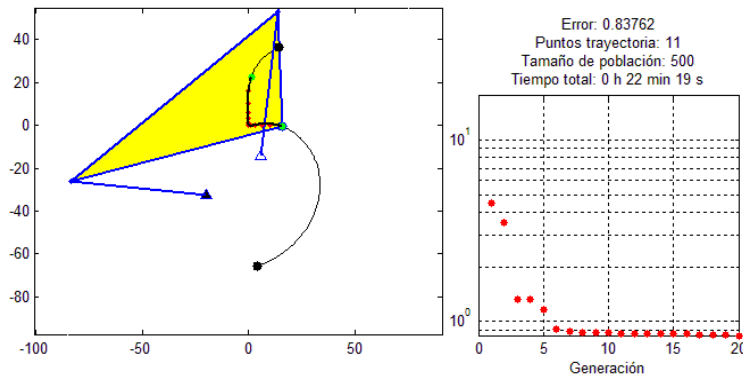


Fig. 5.35 Mejor mecanismo para trayectoria L

Los resultados de esta prueba se muestran en el anexo E en las dos primeras columnas (AE, algoritmos evolutivos, o método 0 propuesto; y AH1 que se corresponde con el método 1 propuesto).

5.5.2 Elipse con punto anguloso

La trayectoria utilizada, está compuesta por 18 puntos que se presentan en la Tabla 5.7 en coordenadas cartesianas:

<i>Punto n°</i>	<i>Coordenada x</i>	<i>Coordenada y</i>
1	0.50	1.10
2	0.40	1.10
3	0.30	1.10
4	0.20	1.00
5	0.10	0.90
6	0.05	0.75
7	0.02	0.60
8	0.00	0.50
9	0.00	0.40
10	0.03	0.30
11	0.10	0.25
12	0.15	0.20
13	0.20	0.30
14	0.30	0.40
15	0.40	0.50
16	0.50	0.70
17	0.60	0.90
18	0.60	1.00

Tabla 5.7 Puntos de trayectoria elipse con punto anguloso

Esta trayectoria ha sido utilizada en otras referencia bibliográficas como (Kunjur & Krishnamurty, 1997) y (Laribi, et al., 2004).

Los resultados para los diferentes métodos se presentan en la Tabla 5.8. En primer lugar se presenta una tabla con datos estadísticos del error. Para cada método y un número de individuos por generación se muestran datos de error máximo, mínimo, medio y desviación estándar del error.

Individuos/Generación	Error	Método 0	Método 1
50	max	46.62	1.26
	min	4.22	0.53
	med	21.39	0.85
	std	14.80	0.28
100	max	41.36	0.88
	min	5.25	0.47
	med	21.54	0.60
	std	15.30	0.12
200	max	27.79	0.70
	min	6.21	0.48
	med	14.08	0.54
	std	7.84	0.06
500	max	32.35	0.54
	min	2.29	0.46
	med	14.88	0.49
	std	10.23	0.02

Tabla 5.8 Error máximo, mínimo, medio y desviación estándar para trayectoria elipse con punto angularo

Junto con los datos de error, se presenta la Tabla 5.9 recoge el tiempo medio para cada uno de los ejercicios de síntesis por método e individuos por generación (para un procesador Intel Core i5 (M560 2.53GHz)). Este dato de tiempo pretende mostrar diferencias en el coste computacional de cada uno de los métodos. En la Tabla 5.9 se muestra la media y desviación típica de los datos temporales en minutos.

Individuos/Generación	Tiempo	Método 0	Método 1
50	med	9.29	9.81
	std	0.74	0.69
100	med	17.80	20.01
	std	1.07	1.60
200	med	35.72	37.85
	std	2.50	3.03
500	med	89.06	105.66
	std	7.12	8.45

Tabla 5.9 Tiempo medio y desviación estándar para trayectoria elipse con punto angularo

Como resumen de ambas tablas se presenta la Fig 5.36 en la que se ha representado el error medio (máximo y mínimo) para cada uno de los métodos y número de individuos por generación, y en área de marcador el tiempo medio de cada una de las pruebas.

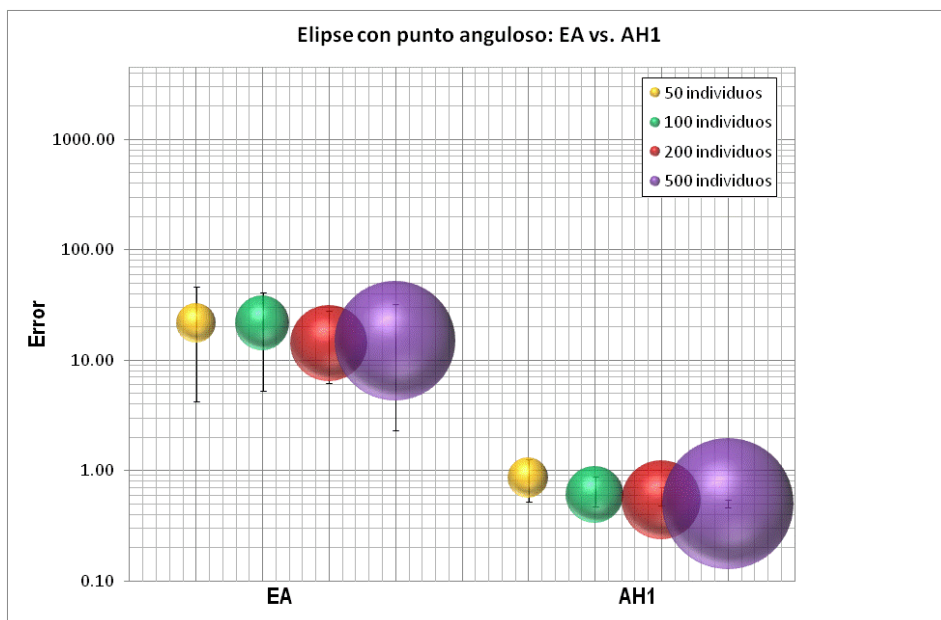


Fig. 5.36 Resultados de ejercicio de síntesis para elipse

Finalmente se presenta la Fig. 5.37 con el mejor resultado obtenido para esta trayectoria:

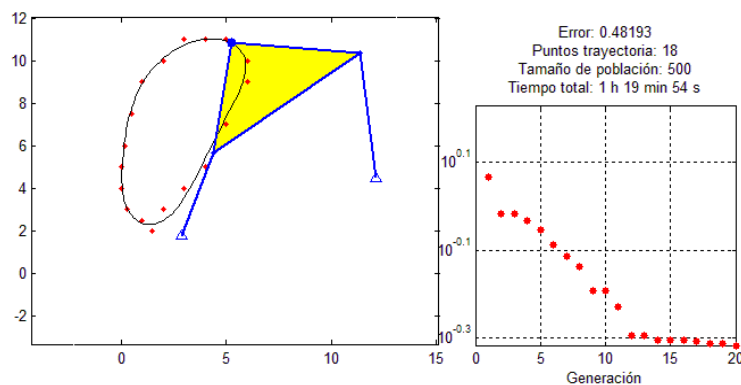


Fig. 5.37 Mejor mecanismo obtenido para trayectoria de elipse con punto angular

Los resultados de esta prueba se muestran en el anexo F en las dos primeras columnas (AE, algoritmos evolutivos, o método 0 propuesto; y AH1 que se corresponde con el método 1 propuesto).

5.5.3 Gota de agua

La trayectoria utilizada, está compuesta por 25 puntos que se presentan en la Tabla 5.10 en coordenadas cartesianas:

<i>Punto n°</i>	<i>Coordenada x</i>	<i>Coordenada y</i>
1	7.03	5.99
2	6.95	5.45
3	6.77	5.03
4	6.40	4.60
5	5.91	4.03
6	5.43	3.56
7	4.93	2.94
8	4.67	2.60
9	4.38	2.20
10	4.04	1.67
11	3.76	1.22
12	3.76	1.97
13	3.76	2.78
14	3.76	3.56
15	3.76	4.34
16	3.76	4.91
17	3.76	5.47
18	3.80	5.98
19	4.07	6.40
20	4.53	6.75
21	5.07	6.85
22	5.50	6.84
23	5.89	6.83
24	6.41	6.80
25	6.92	6.58

Tabla 5.10 Puntos trayectoria gota de agua

Esta trayectoria ha sido utilizada en otras referencia bibliográficas como (Mc Garva & al., 1994) y (Laribi, et al., 2004).

Los resultados para los diferentes métodos se presentan en la Tabla 5.11. En primer lugar se presenta una tabla con datos estadísticos del error. Para cada método y un número de individuos por generación se muestran datos de error máximo, mínimo, medio y desviación estándar del error.

Individuos/Generación	Error	Método 0	Método 1
50	max	55.87	1.73
	min	16.64	0.99
	med	40.19	1.29
	std	15.10	0.29
100	max	30.97	1.27
	min	5.87	0.55
	med	18.41	0.86
	std	10.36	0.28
200	max	37.29	1.26
	min	3.84	0.42
	med	15.62	0.85
	std	11.86	0.27
500	max	17.39	0.78
	min	4.30	0.43
	med	7.99	0.60
	std	4.41	0.14

Tabla 5.11 Error máximo, mínimo, medio y desviación estándar para trayectoria gota de agua

Junto con los datos de error, se presenta la Tabla 5.12 que recoge el tiempo medio para cada uno de los ejercicios de síntesis por método e individuos por generación (para un procesador Intel Core i5 (M560 2.53GHz)). Este dato de tiempo pretende mostrar diferencias en el coste computacional de cada uno de los métodos. En la Tabla 5.10 se muestra la media y desviación típica de los datos temporales en minutos.

Individuos/Generación	Tiempo	Método 0	Método 2
50	med	8.64	5.33
	std	0.43	0.43
100	med	14.27	10.17
	std	1.00	0.81
200	med	27.60	19.73
	std	1.38	1.18
500	med	69.41	49.02
	std	5.55	2.94

Tabla 5.12 Tiempo medio y desviación estándar para trayectoria gota de agua

Como resumen de ambas tablas se presenta la Fig. 5.38 en la que se ha representado el error medio (máximo y mínimo) para cada uno de los métodos y número de individuos por generación, y en área de marcador el tiempo medio de cada una de las pruebas.

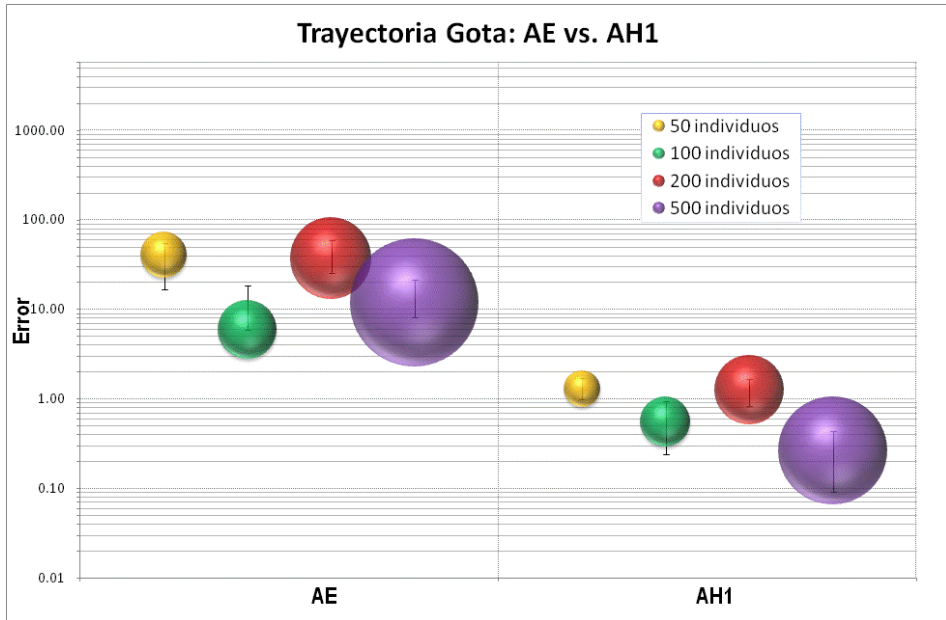


Fig. 5.38 Resultados para síntesis de gota

Finalmente se presenta la Fig. 5.39 con el mejor resultado obtenido para esta trayectoria:

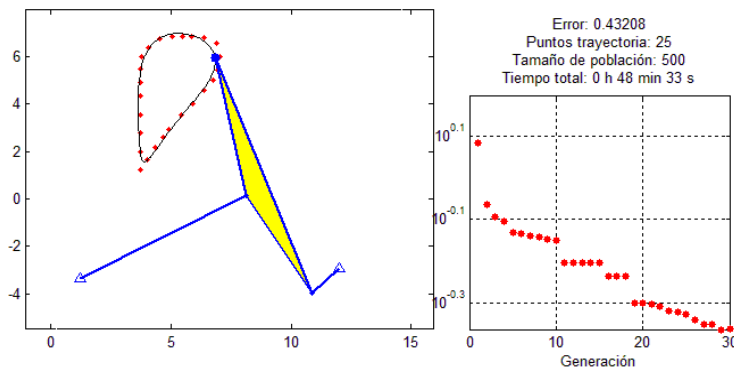


Fig. 5.39 Mejor mecanismo obtenido para trayectoria de gota

Los resultados de esta prueba se muestran en el anexo G en las dos primeras columnas (AE, algoritmos evolutivos, o método 0 propuesto; y AH1 que se corresponde con el método 1 propuesto).

5.5.4 Limason

La trayectoria utilizada, está compuesta por 25 puntos que se presentan en la Tabla 5.13 en coordenadas cartesianas, y gráficamente en la Fig. 5.40.

Punto n°	Coordenada x	Coordenada y
1	-15.00	0.00
2	-14.22	-3.65
3	-12.06	-6.63
4	-8.958	-8.41
5	-5.550	-8.74
6	-2.500	-7.69
7	-0.353	-5.61
8	0.5857	-3.07
9	0.3160	-0.67
10	-0.875	1.058
11	-2.500	1.816
12	-3.995	1.582
13	-4.882	0.616
14	-4.882	-0.61
15	-3.995	-1.58
16	-2.500	-1.81
17	-0.875	-1.05
18	0.3160	0.671
19	0.5857	3.070
20	-0.353	5.616
21	-2.500	7.694
22	-5.550	8.745
23	-8.958	8.412
24	-12.06	6.630
25	-14.22	3.652

Tabla 5.13 Puntos de precisión de trayectoria Limason

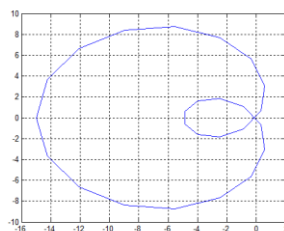


Fig. 5.40 Trayectoria Limason

Los resultados para los diferentes métodos se presentan a continuación. En primer lugar se presenta la Tabla 5.14 con datos estadísticos del error. Para cada método y un número de individuos por generación se muestran datos de error máximo, mínimo, medio y desviación estándar del error.

Individuos/Generación	Error	Método 0	Método 1
50	max	4.33E+02	2.00E+01
	min	1.30E+02	1.71E+00
	med	3.03E+02	8.94E+00
	std	1.19E+02	6.70E+00
100	max	4.45E+02	1.49E+01
	min	8.20E+01	1.06E+00
	med	3.26E+02	7.90E+00
	std	1.23E+02	6.05E+00
200	max	3.32E+02	9.82E+00
	min	9.00E+01	1.27E+00
	med	1.71E+02	3.68E+00
	std	1.05E+02	2.89E+00
500	max	3.34E+02	4.48E+00
	min	1.99E+01	4.45E-01
	med	9.11E+01	2.02E+00
	std	1.09E+02	1.52E+00

Tabla 5.14 Error máximo, mínimo, medio y desviación estándar para trayectoria Limason

Junto con los datos de error, se presenta la Tabla 5.15 en la que se recoge el tiempo medio para cada uno de los ejercicios de síntesis por método e individuos por generación (para un procesador Intel Core i5 (M560 2.53GHz)). Este dato de tiempo pretende mostrar diferencias en el coste computacional de cada uno de los métodos. En la Tabla 5.15 se muestra la media y desviación típica de los datos temporales en minutos.

Individuos/Generación	Tiempo	Método 0	Método 1
50	med	9.42	4.04
	std	0.57	0.24
100	med	17.81	7.77
	std	1.42	0.39
200	med	36.14	15.24
	std	2.17	0.76
500	med	90.47	38.65
	std	4.52	2.71

Tabla 5.15 Tiempo medio y desviación estándar para trayectoria Limason

Como resumen de ambas tablas se presenta la Fig. 5.41 en la que se ha representado el error medio (máximo y mínimo) para cada uno de los métodos y número de individuos por generación, y en área de marcador el tiempo medio de cada una de las pruebas.

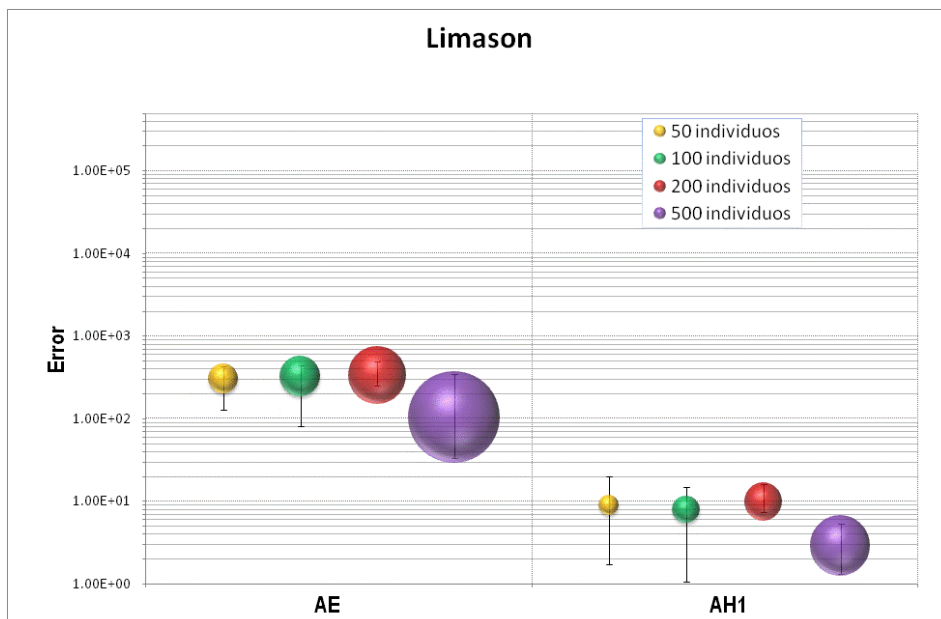


Fig. 5.41 Resultados estadísticos de los ejercicios de síntesis para trayectoria limason

Finalmente se presenta la Fig. 5.42 con un resultado obtenido para esta trayectoria.

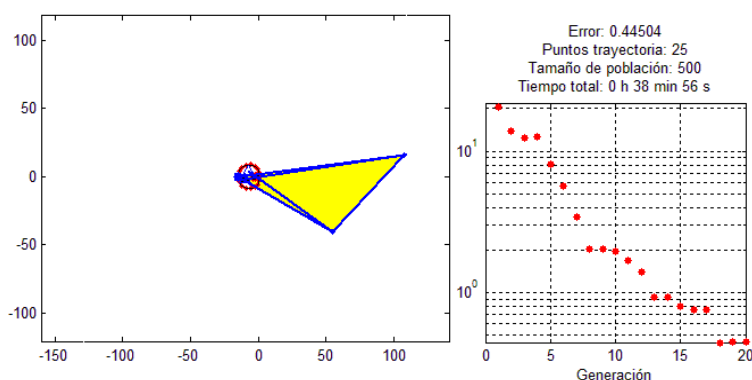


Fig. 5.42 Mecanismo con menor error obtenido para la trayectoria Limason

Los resultados de esta prueba se muestran en el anexo H en las dos primeras columnas (AE, algoritmos evolutivos, o método 0 propuesto; y AH1 que se corresponde con el método 1 propuesto).

5.6 Conclusiones

Se han estudiado los algoritmos evolutivos como posible solución a los problemas de síntesis. En un principio utilizando la toolbox de MATLAB® de algoritmos evolutivos con el fin de evaluar alguna de las opciones que presentan estos mecanismos.

Se ha generado una rutina propia para solución del problema de síntesis mediante algoritmos evolutivos. En esta rutina se han aplicado dos estimadores de error, DMC y TRS. La rutina tiene también los siguientes operadores:

- Élite, para garantizar que generaciones posteriores tengan al menos una aptitud como la de la generación actual
- Cruce, el operador de cruce utilizado en la programación propia es el de media aritmética ponderada
- Mutación, finalmente se ha implementado un operador de mutación que selecciona un gen del individuo a mutar, para posteriormente modificar el valor de este gen de forma aleatoria.

En los ejemplos anteriores se han utilizado 2 métodos evolutivos diferentes para generar 4 trayectorias diferentes y para un tipo de mecanismo, el cuadrilátero articulado.

Con las 4 trayectorias utilizadas, se ha querido garantizar la validez del método para diferentes tipos de trayectorias, así, se han utilizado trayectorias abiertas y cerradas, con y sin puntos singulares.

Para poder hacer una comparativa entre todas las trayectorias y para todos los métodos se ha generado la Fig. 5.43 de error similar a las mostradas en los ejemplos, en la que se ha normalizado el error para todas las trayectorias, tomando como referencia error 100 para todas las trayectorias, para poblaciones de 50 individuos en el método 1 (AH1).

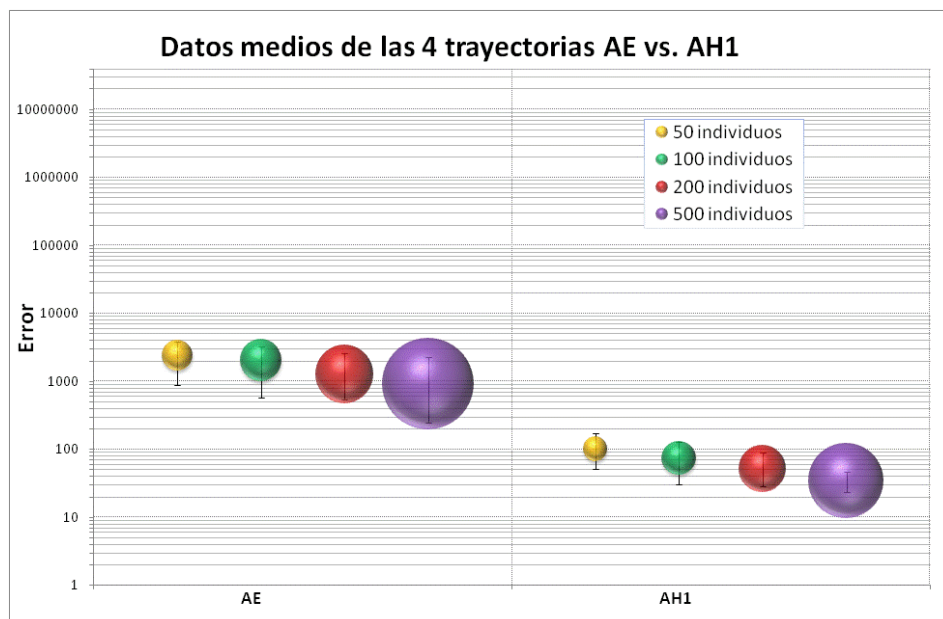


Fig. 5.43 Comparativa de los 2 métodos y las 4 trayectorias

Los métodos han mostrado resultados aceptables, incluso el método 0 (AE), un método evolutivo sin modificación del estimador de error, DMC. Sin embargo, para este primer método, los resultados no son muy precisos, aunque el coste computacional no es elevado.

En el método 1 (AH1), que es un método evolutivo convencional, en el que el estimador de error, TRS, tiene en cuenta la posible traslación, rotación y escalado de la trayectoria, los resultados mejoran sensiblemente, reduciéndose en más de un orden de magnitud el error, y sin incrementar el coste computacional.

El error obtenido en el método 1 (AH1) se podría reducir más si se aplicase un método local sobre el individuo obtenido con el algoritmo evolutivo. Esto representaría una variable de hibridación del método. Así, y debido al interés que pueden presentar los métodos híbridos, se pasará a desarrollarlos en el siguiente capítulo.

Capítulo 6

“Un aspecto curioso de la teoría de la evolución es que todo el mundo cree que la comprende.”

Jacques-Lucien Monod

6 Métodos híbridos para síntesis dimensional de mecanismos.

6.1 Introducción

Como una conclusión del capítulo anterior, se vislumbró que los métodos de síntesis basados en algoritmos evolutivos, son métodos eficientes para buscar individuos con un gran potencial de obtener una buena solución al problema de síntesis, sin necesidad de plantear un individuo de partida para dicho proceso.

De la misma forma, y en capítulos anteriores, se ha mostrado, que los métodos de síntesis basados en optimización local son rápidos y eficientes, y que si el individuo de partida tiene suficiente aptitud, se pueden conseguir muy buenos resultados con un coste computacional muy reducido.

Con las observaciones hechas hasta ahora, se van a plantear en este capítulo diferentes estrategias para generar métodos híbridos. En concreto se van a plantear 3 nuevas posibles estrategias de hibridación. El estimador de error TRS, implica un método de optimización local, no multivariable, sino de variable única y abordando secuencialmente una optimización local para cada una de las variables; por lo tanto su aplicación dentro de un algoritmo evolutivo es el primer método de hibridación, llamado en este trabajo Método 1 o AH1.

Una segunda estrategia y posiblemente la más evidente, es realizar un proceso de optimización local de todos los individuos de la última generación obtenida mediante un método evolutivo.

En una tercera estrategia, se va a plantear el modificar el estimador de error TRS, o DMC, por un proceso de optimización local, dentro del método evolutivo. Utilizar un proceso de optimización local como estimador de error pretende valorar la aptitud del individuo evaluado, evolucionándolo tanto como sea posible para alcanzar la función objetivo, en la teoría Lamarkiana representaría el aprendizaje del individuo a lo largo de su vida. De esta forma, y aunque el proceso de evaluación de las generaciones tenga u

n coste computacional más elevado, se deben conseguir buenos resultados en un número de generaciones mucho más reducido.

Finalmente se plantea un cuarto método de hibridación en el que se aplica una optimización local como estimador de error sólo a los elementos de élite de cada una de las generaciones.

6.2 Algoritmo Híbrido 1 (AH1): Hibridación en el estimador de error

En este primer método se parte de un algoritmo evolutivo ordinario y se introduce una optimización local para la estimación del error y la evaluación de la aptitud.

La Fig 6.1 muestra el diagrama de flujo de dicho algoritmo híbrido que incluye el procedimiento estocástico y la optimización determinista. En el lado izquierdo de la Fig. 6.1, se muestra el esquema de la técnica evolutiva. En el lado derecho en la misma figura se muestra la parte determinista del algoritmo híbrido.

El algoritmo comienza con la definición por parte del diseñador de la función deseada basada en el movimiento requerido por el diseño. El diseñador también establece los parámetros del algoritmo evolutivo que serán utilizados (por ejemplo, los operadores de selección, cruce, etc.), así como el criterio de convergencia del proceso, es decir, hasta qué punto es admisible que el mecanismo no cumpla exactamente la función objetivo.

El proceso de optimización se inicia con la generación de la primera población. La evaluación de la aptitud de esta primera generación requiere de la estimación del error mediante el uso de la optimización determinista, para como se ha explicado en puntos anteriores, poder evaluar la capacidad del mecanismo para obtener la función objetivo, sin que esto dependa de la traslación, el ángulo de orientación y factor de escala, s , del mecanismo generado (aleatoriamente) respecto a la función objetivo. En este caso, el estimador de error utilizado es el estimador de error TRS (TRS: Translation-Rotation-Scaling).

Si en esta primera generación el valor de aptitud está por encima de un determinado umbral de aptitud, criterio de convergencia definido por el diseñador, se dará por terminado el proceso. Entonces se selecciona la

familia de mecanismos, que cumplan con las restricciones y cuya aptitud supere la aptitud umbral. Pudiendo elegir dentro de esta familia de soluciones el individuo más interesante para el diseñador, normalmente el individuo con una mayor aptitud como solución óptima. Esto rara vez ocurre en la primera generación y son necesarias varias generaciones para alcanzar la solución.

El proceso evolutivo se repite para cada generación haciendo una llamada al método local, estimador TRS, cada vez que se trata de valorar la aptitud de los individuos. Cuando se alcanza el umbral de aptitud el proceso se detiene obteniendo una familia de mecanismos todos ellos capaces de cumplir con los requisitos establecidos.

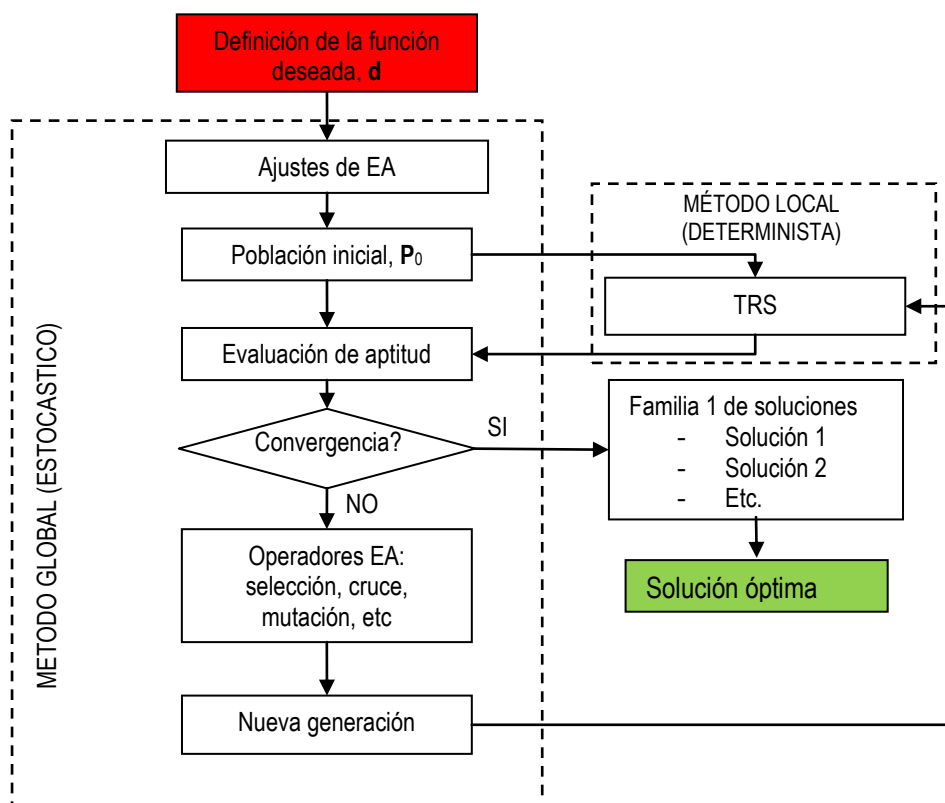


Fig. 6.1 Diagrama de flujo del AH1

6.3 Algoritmo Híbrido 2 (AH2): Hibridación en el estimador de error y en la solución final.

Este caso es similar al anterior pero se añade una nueva optimización local cuando el proceso evolutivo ha alcanzado la convergencia en el método estocástico. Así, en este método, se deben definir dos criterios de convergencia, uno para el método estocástico, y otro para el método determinista.

En la Fig. 6.2 se muestra el esquema de dicho proceso. Se observa cómo una vez alcanzada la convergencia en el lado izquierdo, comienza un nuevo proceso de optimización local. Para ello se seleccionan los mejores individuos obtenidos tras la convergencia del método estocástico y se les aplica un proceso de optimización local completo. Como resultado se obtiene una segunda familia de individuos que han mejorado sustancialmente su aptitud. De entre estos individuos se selecciona el mejor como solución óptima.

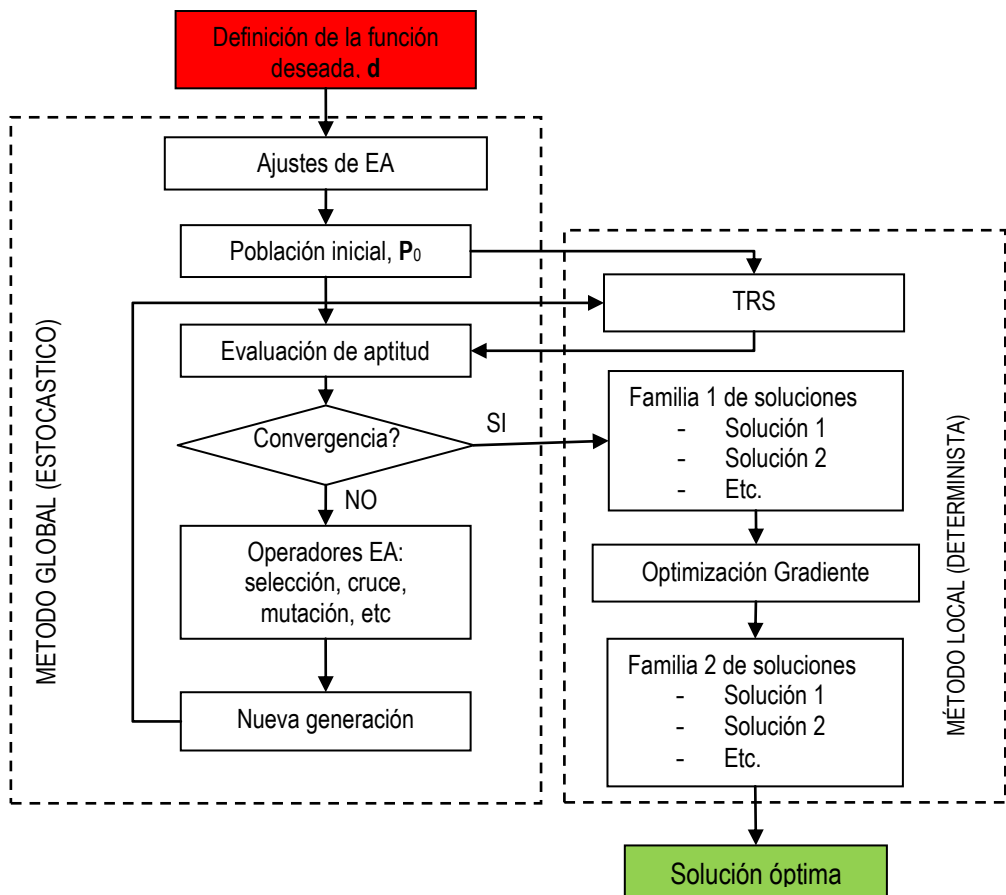


Fig. 6.2 Diagrama de flujo del AH2

6.4 Algoritmo Híbrido 3 (AH3): Optimización local de todos los individuos

El tercer método es similar al primero, pero modificando el estimador de error que ahora pasa a denominarse estimador de Error Optimizado mediante métodos Locales (EOL), basado en el estimador TRS y el algoritmo de optimización local propuesto en el capítulo 4. El EOL se nutre de los mecanismos generados por el algoritmo evolutivo y realiza una optimización local en todos los individuos generados. Esta optimización local se divide en dos pasos:

1. Se realiza el traslado, rotado y escalado de cada individuo mediante el estimador de error TRS.
2. En segundo lugar se optimizan todas las variables de diseño empleando el método de optimización local basado en el gradiente, propuesto en el capítulo 4.

Una vez realizada la optimización local se procede a evaluar la aptitud de todos los individuos activando de nuevo el algoritmo evolutivo. Tras esto, si se cumple el criterio de convergencia impuesto, se obtiene una familia de individuos, pudiéndose seleccionar de entre esta familia el individuo óptimo, el de mejor aptitud. Tal y como ya se ha comentado, esto no sucede habitualmente, por lo que tras evaluar la aptitud, se chequea la convergencia de la solución, y tras esto se aplican los operadores evolutivos y se genera una nueva generación para volver a evaluar el error en cada uno de los individuos, repitiendo este proceso hasta obtener la convergencia.

Por todo ello en el AH3 la optimización local final del AH2 se suprime, al realizarse durante la estimación de error.

Evidentemente la ventaja que se busca con esta nueva implementación es obtener una mejor evaluación de la aptitud de los individuos lo que debe redundar un menor error en los individuos obtenidos como resultados. Por el contrario, la principal desventaja es el tiempo necesario para realizar la optimización local de todos los individuos. Este tiempo se suma al necesario para la exploración del espacio de las variables de diseño que se requiere en algoritmo evolutivo. Por tanto los ensayos deben subrayar la eficiencia del algoritmo en comparación con otras soluciones.

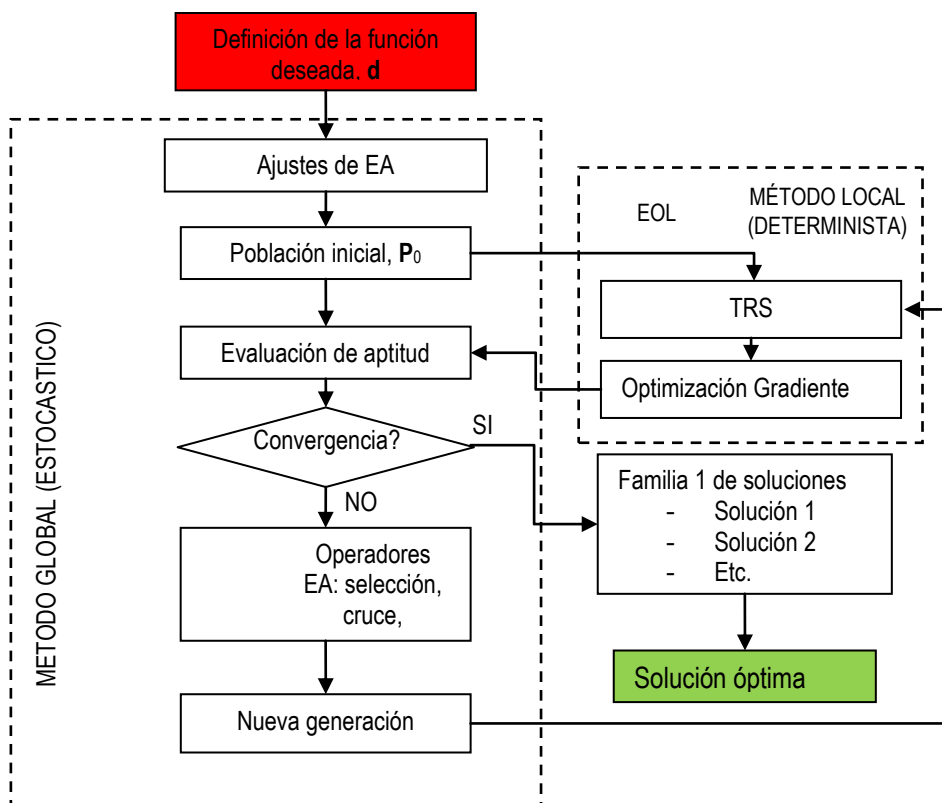


Fig. 6.3 Diagrama de flujo del AH3

6.5 Algoritmo Híbrido 4 (AH4): Optimización local centrada en la élite.

El cuarto método de hibridación propuesto combina los dos estimadores de error descritos: TRS y EOL. Un esquema general del método se presenta en la Fig. 6.4. En este método, y tras definir la función objetivo, parámetros del método evolutivo y el criterio de convergencia, se genera la población inicial.

Para evaluar la aptitud de esta población inicial (generación 0), se utiliza el estimador de error 1 (TRS). Tras evaluar la aptitud, y al igual que en el resto de métodos, si se alcanza el criterio de convergencia, el método proporciona una familia de soluciones que contiene la solución óptima.

Normalmente no se consigue la convergencia con la población inicial, por lo que tras evaluar la aptitud de la generación 0, se aplican los operadores del método evolutivo y se obtiene la generación 1.

En la generación 1 y sucesivas existen dos grupos de individuos:

1. Grupo Elite: Son los mejores individuos de la generación anterior.
2. Resto de la población: compuesto por los individuos generados mediante operadores evolutivos de cruce y mutación.

Así, a partir de este punto, para estimar el error del Grupo de Élite de la generación analizada se utiliza el EOL, mientras que para el resto de la población se utiliza el estimador TRS. Tras estimar el error de los individuos de la generación, se evalúa su aptitud, para posteriormente chequear el criterio de convergencia. En caso de no cumplirse, se vuelven a aplicar los operadores evolutivos para generar la siguiente generación, con un grupo de élite y otro proveniente de los operadores de cruce y mutación, y así volver a estimar el error.

Este proceso se repite hasta obtener una generación que cumpla el criterio de convergencia. Una vez cumplido el criterio, se obtiene una familia de individuos, de la que el individuo con mejor aptitud representa la solución óptima.

Con este algoritmo se persigue aumentar la eficiencia reduciendo el número de optimizaciones locales en la estimación del error mediante el estimador EOL, y como consecuencia reduciendo el coste computacional. Como se menciona en el AH3 el coste computacional es el principal inconveniente del método híbrido. El método híbrido así obtenido dedica un mayor esfuerzo computacional a los individuos mejor adaptados (estimador EOL), mientras que los peor adaptados utilizan un estimador de error con menor coste computacional, el TRS.

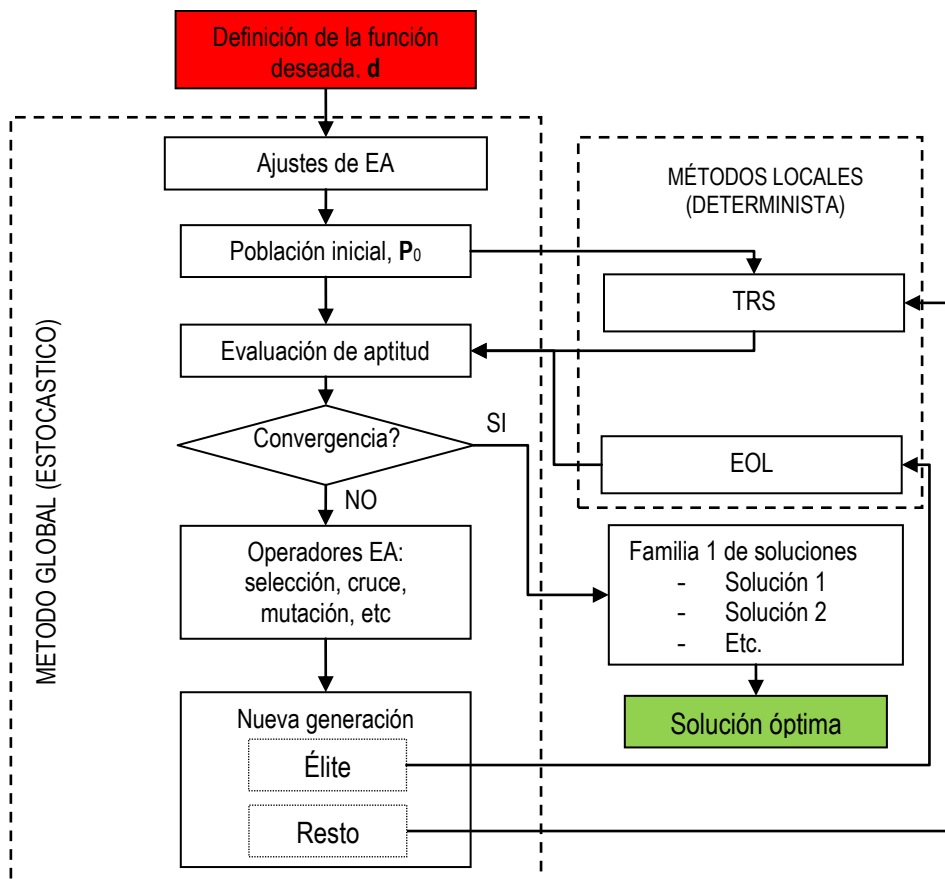


Fig. 6.4 Diagrama de flujo del AH4

6.6 Ejemplos

Tomando como referencia diferentes trayectorias de interés, se han realizado pruebas con los diferentes métodos híbridos descritos, y con un método evolutivo convencional, con el estimador de error DMC. Así, los métodos están representados por las columnas de las tablas, siendo el identificador:

Método 0: el que representa el método evolutivo convencional con estimador de error DMC

Método 1: AH1: primer método híbrido propuesto, que aplica el estimador de error TRS

Método 2: AH2: segundo método híbrido propuesto con un proceso de optimización mediante gradiente propuesto en el capítulo 4.

Método 3: AH3: tercer método híbrido propuesto que utiliza el estimador de error EOL.

Método 4: AH4: cuarto método híbrido propuesto que combina la utilización del estimador de error EOL en la élite y el estimador de error TRS en el resto de individuos.

En cuanto al número de individuos por generación en cada uno de los métodos, se resume en la Tabla 6.1.

Método	0	1	2	3	4
50 ind/gen	X	X	X	X	X
100 ind/gen	X	X	X	X	X
200 ind/gen	X	X	X		
500 ind/gen	X	X	X		

Tabla 6.1 Pruebas realizadas con los métodos híbridos propuestos

Con los métodos AH3 y AH4 no se han realizado pruebas con poblaciones grandes, ya que involucran procesos de mejora como parte del estimador de error. En el método AH3 esta mejora se da para cada uno de los individuos de la población y en el AH4 para la élite. Debido a esta mejora en la aptitud del individuo (optimización local), la aptitud de cada individuo analizado (en el caso AH3 en todos y en el AH4 en la élite), es cuantificada teniendo en cuenta la mejora posible de aptitud por cambios morfológicos en el individuo, por lo que no es necesario tener poblaciones grandes por generación para obtener buenos resultados.

Desde un punto de vista de trayectorias probadas, se pueden clasificar en trayectorias abiertas, generando un ángulo recto como trayectoria objetivo

y trayectorias cerradas, elipse con punto singular, gota de agua y limason, todas ellas utilizadas en el Capítulo 5.

La presentación de las pruebas realizadas se ha organizado por trayectoria objetivo, presentando para cada una de las trayectorias datos estadísticos de las pruebas y el mejor resultado obtenido.

6.6.1 Trayectoria L

La trayectoria L utilizada, está compuesta por 11 puntos que se presentan en la Tabla 5.4 en coordenadas cartesianas:

Los resultados para los diferentes métodos se presentan a continuación. En primer lugar se presenta la Tabla 6.2 con datos estadísticos del error. Para cada método y un número de individuos por generación se muestran datos de error máximo, mínimo, medio y desviación estándar del error.

Individuos/Generación	Error	Método 0	Método 1	Método 2	Método 3	Método 4
50	max	79.56	20.13	7.51	0.20	6.91
	min	30.55	5.29	0.04	0.02	0.13
	med	47.06	11.08	2.81	0.15	3.14
	std	16.76	5.20	2.85	0.09	2.95
100	max	80.88	15.30	14.78	0.05	5.32
	min	34.26	1.50	0.67	0.01	0.55
	med	56.52	8.08	4.23	0.02	1.94
	std	16.52	4.74	5.08	0.01	1.94
200	max	61.95	7.88	3.28		
	min	11.31	1.02	0.61		
	med	31.65	4.00	1.99		
	std	15.61	2.31	1.06		
500	max	37.44	1.72	0.87		
	min	18.28	0.04	0.58		
	med	25.27	0.96	0.72		
	std	7.56	0.59	0.10		

Tabla 6.2 Error máximo, mínimo, medio y desviación estándar para trayectoria L

Junto con los datos de error, se presenta la Tabla 6.3 en la que se recoge el tiempo medio en minutos para cada uno de los ejercicios de síntesis por método e individuos por generación (para un procesador Intel Core i5 M560 2.53GHz). Este dato de tiempo pretende mostrar diferencias en el coste computacional de cada uno de los métodos. Además del tiempo medio, se dan también valores de desviación típica de los datos temporales en minutos.

Individuos/Generación	Tiempo	Método 0	Método 1	Método 2	Método 3	Método 4
50	med	2.62	1.20	2.03	105.00	1.77
	std	0.18	0.10	0.12	5.25	0.14
100	med	5.10	2.21	3.11	159.10	3.03
	std	0.25	0.18	0.25	8.47	0.21
200	med	10.00	4.49	5.41		
	std	0.80	0.22	0.43		
500	med	24.79	11.32	12.22		
	std	1.98	0.68	0.98		

Tabla 6.3 Tiempo medio y desviación estándar para trayectoria L

Como resumen de la Tabla 6.2 y la Tabla 6.3 se presenta la Fig. 6.5 en la que se ha representado el error medio (máximo y mínimo) para cada uno de los métodos y número de individuos por generación, y el área de marcador el tiempo medio de cada una de las pruebas.

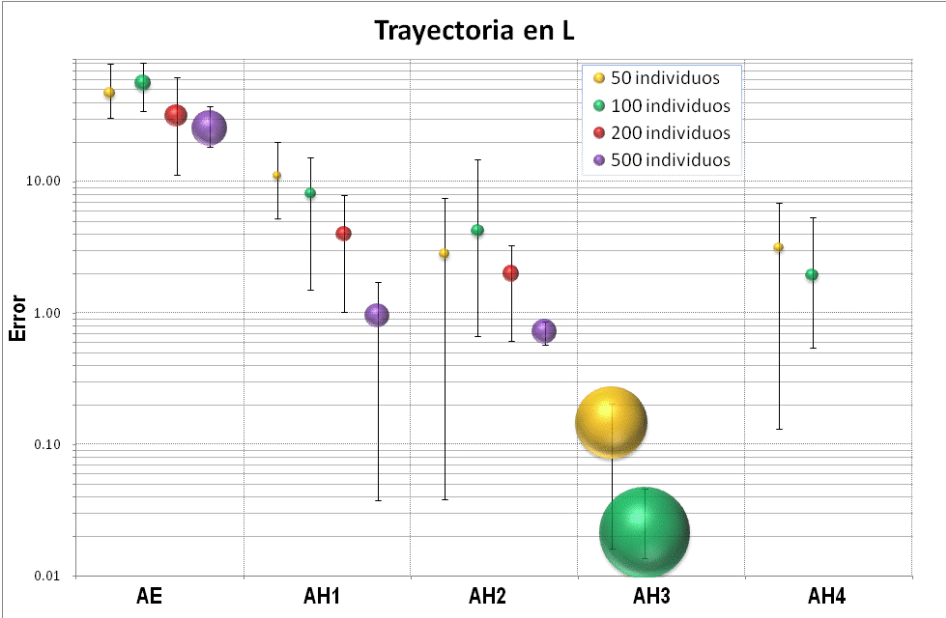


Fig. 6.5 Gráfica comparativa de los distintos métodos

Finalmente se presenta la Fig. 6.6 con el mejor resultado obtenido para esta trayectoria:

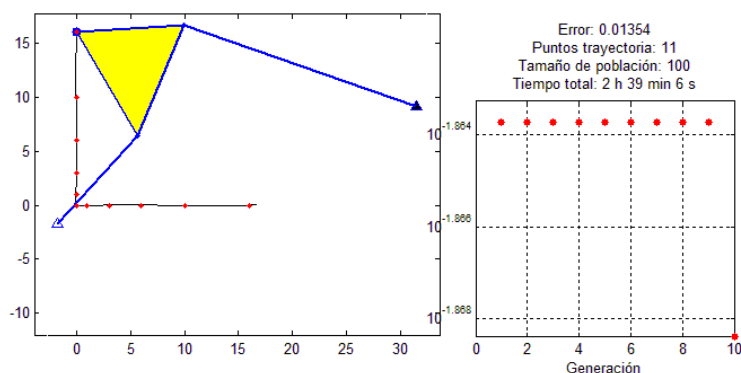


Fig. 6.6 Individuo óptimo obtenido para la trayectoria en ángulo recto

Los resultados de esta prueba se muestran en el anexo E.

6.6.2 Elipse con punto anguloso

La trayectoria utilizada, está compuesta por 18 puntos que se presentan en la Tabla 5.7 en coordenadas cartesianas.

Los resultados para los diferentes métodos se presentan a continuación. En primer lugar se presenta la Tabla 6.4 con datos estadísticos del error. Para cada método y un número de individuos por generación se muestran datos de error máximo, mínimo, medio y desviación estándar del error.

Junto con los datos de error, se presenta la Tabla 6.5 en la que se recoge el tiempo medio en minutos para cada uno de los ejercicios de síntesis por método e individuos por generación (para un procesador Intel Core i5 M560 2.53GHz). Este dato de tiempo pretende mostrar diferencias en el coste computacional de cada uno de los métodos. Además del tiempo medio, se dan también valores de desviación típica de los datos temporales en minutos.

Como resumen de ambas la Tabla 6.4 y la Tabla 6.5 se presenta la Fig. 6.7 en la que se ha representado el error medio (máximo y mínimo) para cada uno de los métodos y número de individuos por generación, y en el área del marcador el tiempo medio de cada una de las pruebas.

Individuos/Generación	Error	Método 0	Método 1	Método 2	Método 3	Método 4
50	max	46.62	1.26	0.70	0.45	0.70
	min	4.22	0.53	0.45	0.45	0.45
	med	21.39	0.85	0.49	0.45	0.52
	std	14.80	0.28	0.08	0.01	0.10
100	max	41.36	0.88	0.49	0.45	0.52
	min	5.25	0.47	0.45	0.45	0.45
	med	21.54	0.60	0.46	0.45	0.47
	std	15.30	0.12	0.02	0.01	0.03
200	max	27.79	0.70	0.51		
	min	6.21	0.48	0.45		
	med	14.08	0.54	0.46		
	std	7.84	0.06	0.02		
500	max	32.35	0.54	0.49		
	min	2.29	0.46	0.45		
	med	14.88	0.49	0.46		
	std	10.23	0.02	0.01		

Tabla 6.4 Error máximo, mínimo, medio y desviación estándar para trayectoria elipse con punto anguloso

Individuos/Generación	Tiempo	Método 0	Método 2	Método 2	Método 3	Método 4
50	med	9.29	9.81	12.14	380.16	5.93
	std	0.74	0.69	0.85	30.41	0.36
100	med	17.80	20.01	22.34	497.75	11.76
	std	1.07	1.60	1.12	39.82	0.71
200	med	35.72	37.85	40.15		
	std	2.50	3.03	2.81		
500	med	89.06	105.66	107.95		
	std	7.12	8.45	8.64		

Tabla 6.5 Tiempo medio y desviación estándar para trayectoria elipse con punto anguloso

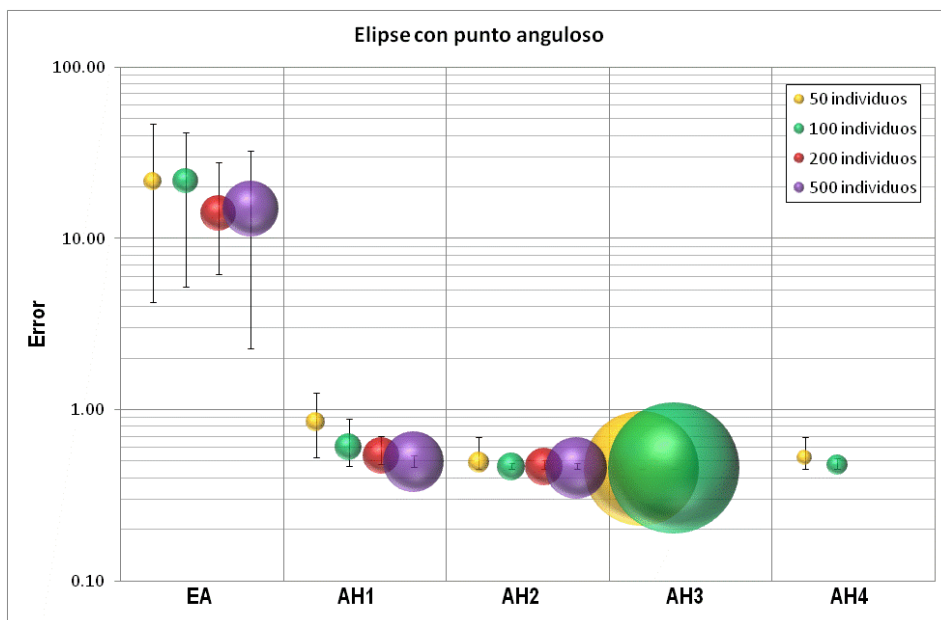


Fig. 6.7 Gráfica comparativa de los distintos métodos

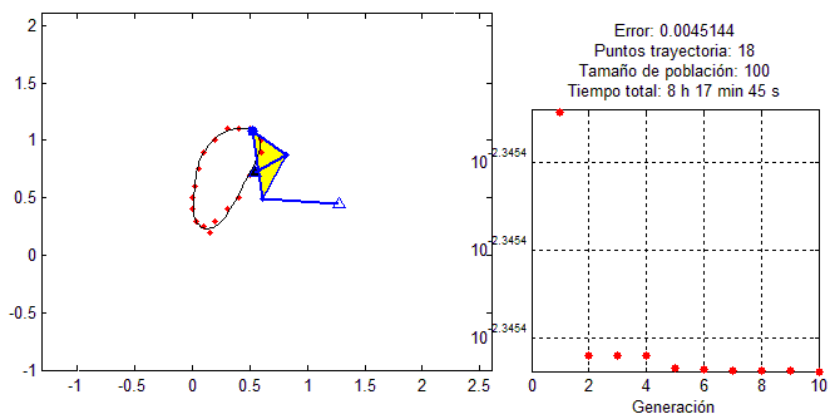


Fig. 6.8 Individuo óptimo obtenido para la trayectoria de elipse con punto anguloso

Finalmente se presenta la Fig. 6.8 con el mejor resultado obtenido para esta trayectoria.

Los resultados de esta prueba se muestran en el Anexo F.

6.6.3 Gota de agua

La trayectoria utilizada, está compuesta por 25 puntos que se presentan en la Tabla 5.10 en coordenadas cartesianas.

Los resultados para los diferentes métodos se presentan a continuación. En primer lugar se presenta la Tabla 6.6 con datos estadísticos del error. Para cada método y un número de individuos por generación se muestran datos de error máximo, mínimo, medio y desviación estándar del error.

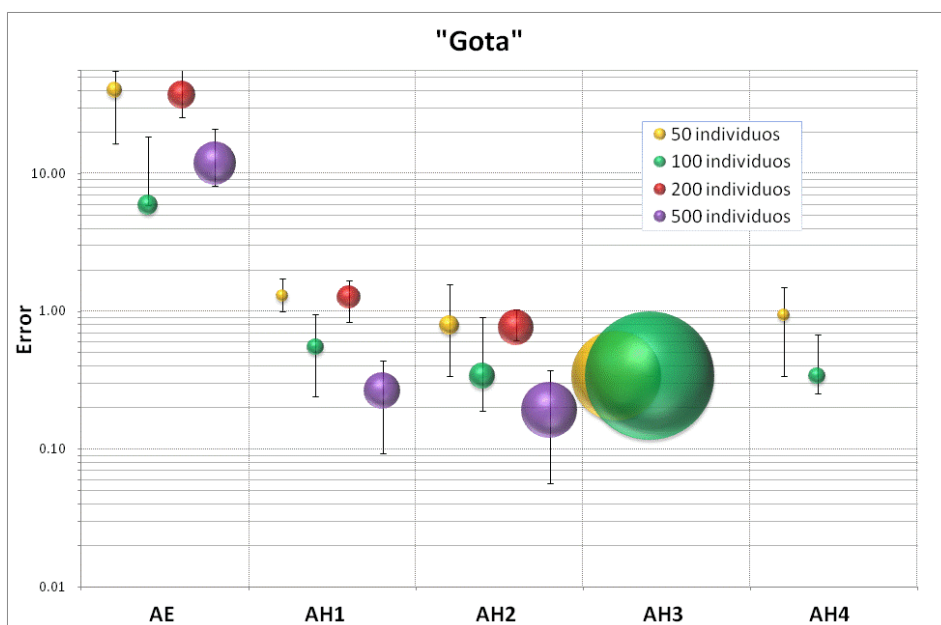
Individuos/Generación	Error	Método 0	Método 1	Método 2	Método 3	Método 4
50	max	55.87	1.73	1.58	0.34	1.50
	min	16.64	0.99	0.34	0.34	0.34
	med	40.19	1.29	0.79	0.34	0.93
	std	15.10	0.29	0.48	0.00	0.50
100	max	30.97	1.27	1.06	0.34	0.77
	min	5.87	0.55	0.34	0.34	0.34
	med	18.41	0.86	0.49	0.34	0.42
	std	10.36	0.28	0.28	0.00	0.19
200	max	37.29	1.26	0.77		
	min	3.84	0.42	0.34		
	med	15.62	0.85	0.49		
	std	11.86	0.27	0.19		
500	max	17.39	0.78	0.65		
	min	4.30	0.43	0.34		
	med	7.99	0.60	0.47		
	std	4.41	0.14	0.16		

Tabla 6.6 Error máximo, mínimo, medio y desviación estándar para trayectoria gota de agua

Junto con los datos de error, se presenta la Tabla 6.7 en la que se recoge el tiempo medio en minutos para cada uno de los ejercicios de síntesis por método e individuos por generación (para un procesador Intel Core i5 M560 2.53GHz). Este dato de tiempo pretende mostrar diferencias en el coste computacional de cada uno de los métodos. Además del tiempo medio, se dan también valores de desviación típica de los datos temporales en minutos.

Individuos/Generación	Tiempo	Método 0	Método 2	Método 2	Método 3	Método 4
50	med	8.64	5.33	13.97	310.49	5.44
	std	0.43	0.43	0.84	18.63	0.43
100	med	14.27	10.17	24.44	642.12	10.48
	std	1.00	0.81	1.47	44.95	0.73
200	med	27.60	19.73	47.33		
	std	1.38	1.18	2.37		
500	med	69.41	49.02	118.43		
	std	5.55	2.94	7.11		

Tabla 6.7 Tiempo medio y desviación estándar para trayectoria gota de agua



Como resumen de ambas tablas se presenta la Fig. 6.9 en la que se ha representado el error medio (máximo y mínimo) para cada uno de los métodos y número de individuos por generación, y en el área del marcador el tiempo medio de cada una de las pruebas.

Finalmente se presenta la Fig. 6.10 con el mejor resultado obtenido para esta trayectoria.

Los resultados de esta prueba se muestran en el Anexo G.

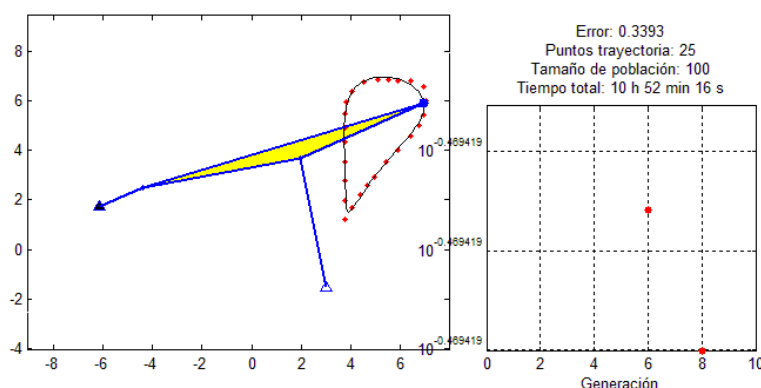


Fig. 6.10 Mejor individuo obtenido para la trayectoria gota

6.6.4 Limason

La trayectoria utilizada, está compuesta por 25 puntos que se presentan en la Tabla 5.13 en coordenadas cartesianas. La Fig. 5.40 muestra la trayectoria Limason gráficamente.

Los resultados para los diferentes métodos se presentan a continuación. En primer lugar se presenta la Tabla 6.8 con datos estadísticos del error. Para cada método y un número de individuos por generación se muestran datos de error máximo, mínimo, medio y desviación estándar del error.

Junto con los datos de error, se presenta la Tabla 6.9 en la que se recoge el tiempo medio en minutos para cada uno de los ejercicios de síntesis por método e individuos por generación (para un procesador Intel Core i5 M560 2.53GHz). Este dato de tiempo pretende mostrar diferencias en el coste computacional de cada uno de los métodos. Además del tiempo medio, se dan también valores de desviación típica de los datos temporales en minutos.

Como resumen de ambas tablas se presenta la Fig. 6.11 en la que se ha representado el error medio (máximo y mínimo) para cada uno de los métodos y número de individuos por generación, y en el área del marcador el tiempo medio de cada una de las pruebas.

Finalmente se presenta la Fig. 6.12 con un resultado obtenido para esta trayectoria.

Individuos/Generación	Error	Método 0	Método 1	Método 2	Método 3	Método 4
50	max	4.33E+02	2.00E+01	8.42E-02	1.05E-05	1.07E-03
	min	1.30E+02	1.71E+00	2.35E-02	6.71E-06	3.43E-05
	med	3.03E+02	8.94E+00	3.36E-02	8.67E-06	2.76E-04
	std	1.19E+02	6.70E+00	2.23E-02	1.86E-06	4.48E-04
100	max	4.45E+02	1.49E+01	3.08E-02	6.01E-06	1.85E-04
	min	8.20E+01	1.06E+00	7.09E-04	2.93E-06	3.53E-05
	med	3.26E+02	7.90E+00	1.86E-02	4.92E-06	1.09E-04
	std	1.23E+02	6.05E+00	1.18E-02	1.72E-06	6.02E-05
200	max	3.32E+02	9.82E+00	3.86E-02		
	min	9.00E+01	1.27E+00	6.85E-04		
	med	1.71E+02	3.68E+00	2.76E-02		
	std	1.05E+02	2.89E+00	1.23E-02		
500	max	3.34E+02	4.48E+00	2.99E-02		
	min	1.99E+01	4.45E-01	1.86E-02		
	med	9.11E+01	2.02E+00	2.56E-02		
	std	1.09E+02	1.52E+00	3.12E-03		

Tabla 6.8 Error máximo, mínimo, medio y desviación estándar para trayectoria Limason

Individuos/Generación	Tiempo	Método 0	Método 2	Método 2	Método 3	Método 4
50	med	9.42	4.04	6.19	269.89	6.11
	std	0.57	0.24	0.31	21.59	0.31
100	med	17.81	7.77	9.90	530.57	12.23
	std	1.42	0.39	0.59	26.53	0.98
200	med	36.14	15.24	17.32		
	std	2.17	0.76	1.04		
500	med	90.47	38.65	40.72		
	std	4.52	2.71	2.44		

Tabla 6.9 Tiempo medio y desviación estándar para trayectoria Limason

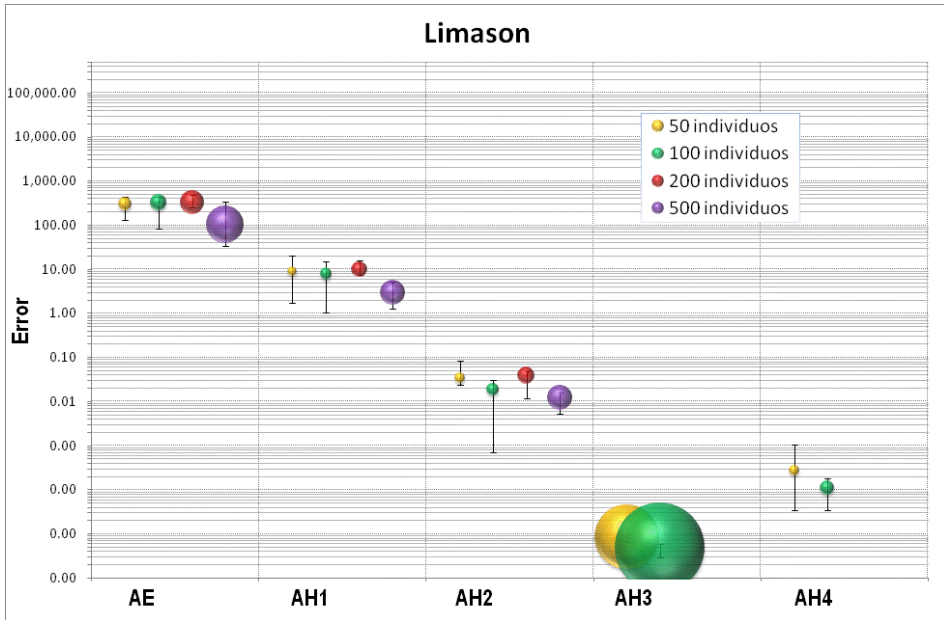


Fig. 6.11 Gráfica comparativa de los distintos métodos

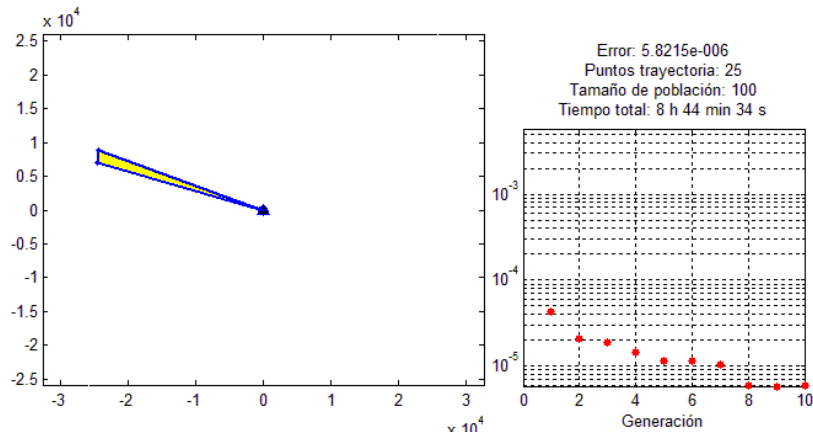


Fig. 6.12 Mejor individuo obtenido para la trayectoria Limason

El resultado mostrado en la Fig. 6.13 es un mecanismo con unas longitudes de barra muy superiores a las dimensiones de la trayectoria. Se debe recordar, que en estos procesos de optimización no se han añadido restricciones, por lo que los mecanismos pueden variar su tamaño tanto como sea necesario para conseguir cumplir con la función objetivo.

Los resultados de esta prueba se muestran en el Anexo H.

6.7 Ejemplo con restricciones de tamaño

Con este ejemplo se quiere evaluar la aptitud de uno de los métodos de hibridación presentados, en concreto el método 4, en el que se aplica el estimador de error EOL en la élite de las diferentes generaciones, y el TRS en el resto de individuos.

La prueba se ha realizado modificando la función objetivo, y teniendo en cuenta en la misma una penalización en función del tamaño del mecanismo. En concreto al error medio cuadrático de los diferentes puntos de precisión, se le suma el sumatorio de las diferentes longitudes de barra con un factor de escala, tal y como se muestra en la Ecuación (6.1).

$$FES = \frac{1}{2} \cdot \varepsilon \cdot \varepsilon^T + Factor_{escala} \cdot \sum L_i \quad (6.1)$$

Utilizado este término adicional en el estimador de error, se han probado 3 valores diferentes para el factor de escala del tamaño, en concreto 1, 10 y 100.

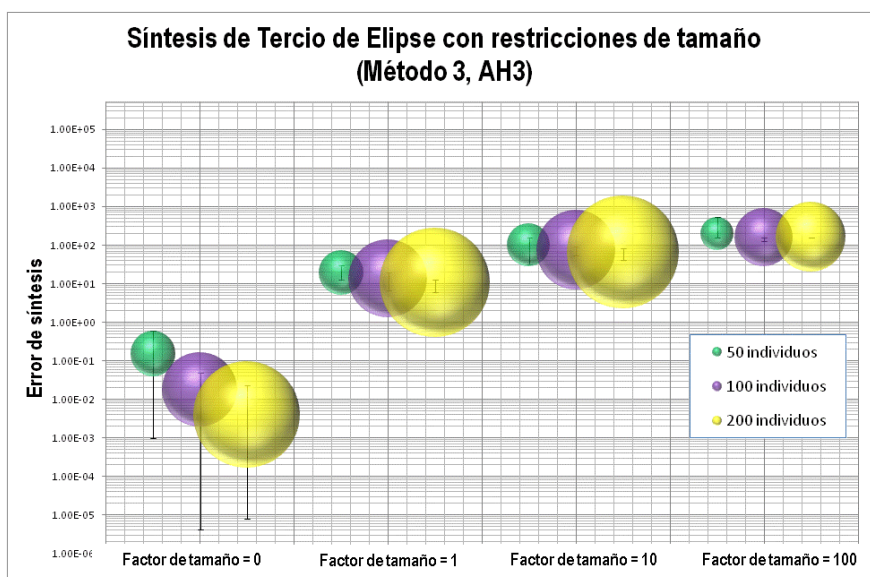


Fig. 6.13 Aplicación de una factor de escala para restringir el tamaño del mecanismo generado para una trayectoria objetivo de arco de elipse, y con el método AH3 (Método 3 propuesto)

Lo anterior se ha aplicado a dos trayectorias objetivo ya utilizadas: un ángulo recto y una limason; pero antes de estas trayectorias, se ha

probado con y sin factor de tamaño para un tercio de elipse. Para esta trayectoria, relativamente sencilla, se han obtenido los siguientes resultados presentados de forma gráfica en la Fig. 6.13.

En valores, los resultados se presentan en la Tabla 6.10.

	Factor_{escala}	Error_{Medio}	Mínimo	Máximo	Des_{Estandar}
50 individuos	0	0.15	0.00	0.61	0.23
	1	19.28	13.01	29.79	4.72
	10	99.44	32.15	158.35	42.06
	100	195.04	158.35	525.27	116.03
100 individuos	0	0.02	0.00	0.05	0.02
	1	13.43	6.56	19.10	4.27
	10	72.84	52.63	93.49	12.36
	100	154.49	126.58	158.35	10.04
200 individuos	0	0.00	0.00	0.02	0.01
	1	10.65	5.94	12.76	2.40
	10	64.45	41.36	81.79	11.80
	100	158.35	158.35	158.35	0.00

Tabla 6.10 Errores para diferentes factores de escala

Así, sin aplicar el factor de tamaño, síntesis sin restricciones, los mecanismos obtenidos tiene un error en la trayectoria muy reducido, y una morfología como la que se presenta en la Fig. 6.14.

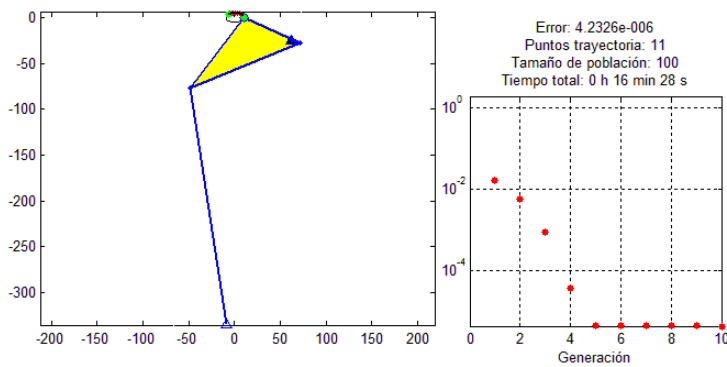


Fig. 6.14 Mejor mecanismo obtenido sin restricción de tamaño

Como se ve, no hay error de síntesis, pero el tamaño del mecanismo comparado con la trayectoria es bastante grande. Si se le aplica una restricción de tamaño penalizando el error en función del tamaño del mecanismo, tal y como se muestra en la Ecuación (6.1), y utilizando un factor de escala 1, se obtiene un mecanismo con la morfología presentada en la Fig. 6.16.

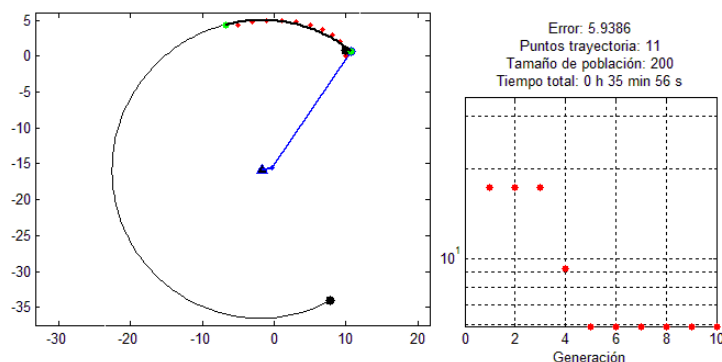


Fig. 6.15 Mejor individuo obtenido con una restricción de tamaño mediante factor de escala 1

Como se puede ver, el error en la generación de trayectoria es algo mayor, aunque podría ser aceptable, pero a cambio se obtiene con un mecanismo de tamaño muy inferior al anterior. Si se aplica un factor de tamaño 10, se obtiene el mecanismo mostrado en la Fig. 6.16.

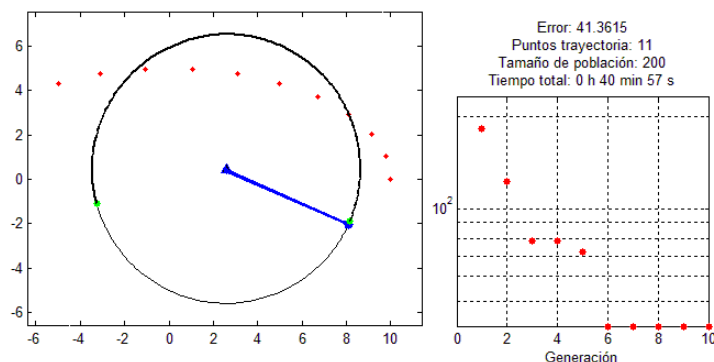


Fig. 6.16 Mejor individuo obtenido con una restricción de tamaño mediante factor de escala 10

En este caso, el error en la generación de trayectoria es probablemente no asumible, pero se genera con un mecanismo muy pequeño. Si finalmente

se aplica un factor de escala 100, se obtiene el mecanismo mostrado en la Fig. 6.17.

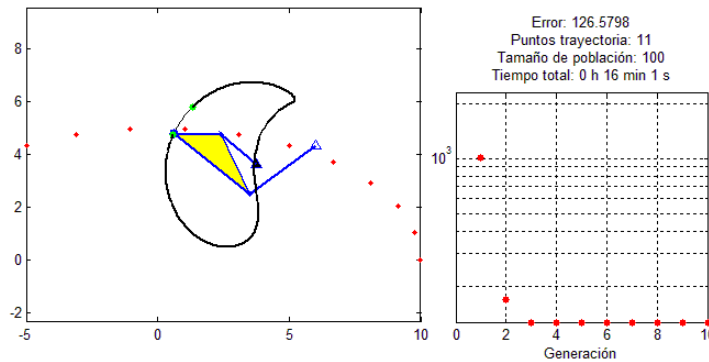


Fig. 6.17 Mejor individuo obtenido con una restricción de tamaño mediante factor de escala 100

Como se puede ver, el tamaño del mecanismo es muy reducido, pero la trayectoria obtenida difiere demasiado de la trayectoria objetivo.

Las trayectorias utilizadas tras este primer ejemplo han sido una trayectoria abierta, el ángulo recto, y una trayectoria cerrada, la limason, ya utilizadas en este trabajo.

Las pruebas se han realizado, como ya se ha comentado con el método 3, y para diferentes tamaños de población. Las poblaciones utilizadas han sido 50 individuos, 100 individuos y 200 individuos.

El número de generaciones utilizado en todos los casos ha sido de 10 generaciones.

A continuación se presentan los resultados obtenidos para la trayectoria de ángulo recto en la Tabla 6.11.

	Factor_{escala}	Error_{Medio}	Mínimo	Máximo	Des_{Estandar}
50 individuos	1	47.40	35.14	55.91	6.33
	10	101.16	61.96	219.45	47.53
	100	284.18	284.18	284.18	0.00
100 individuos	1	29.90	20.67	48.55	9.31
	10	76.47	38.05	134.39	33.65
	100	284.18	121.88	284.18	54.10
200 individuos	1	30.60	16.07	39.24	8.35
	10	78.90	23.58	100.24	25.84
	100	284.18	230.81	284.18	17.79

Tabla 6.11 Errores según factores de escala para trayectoria L

Esta tabla, se puede graficar tal y como se ha realizado en los ejemplos anteriores, obteniendo la Fig. 6.18.

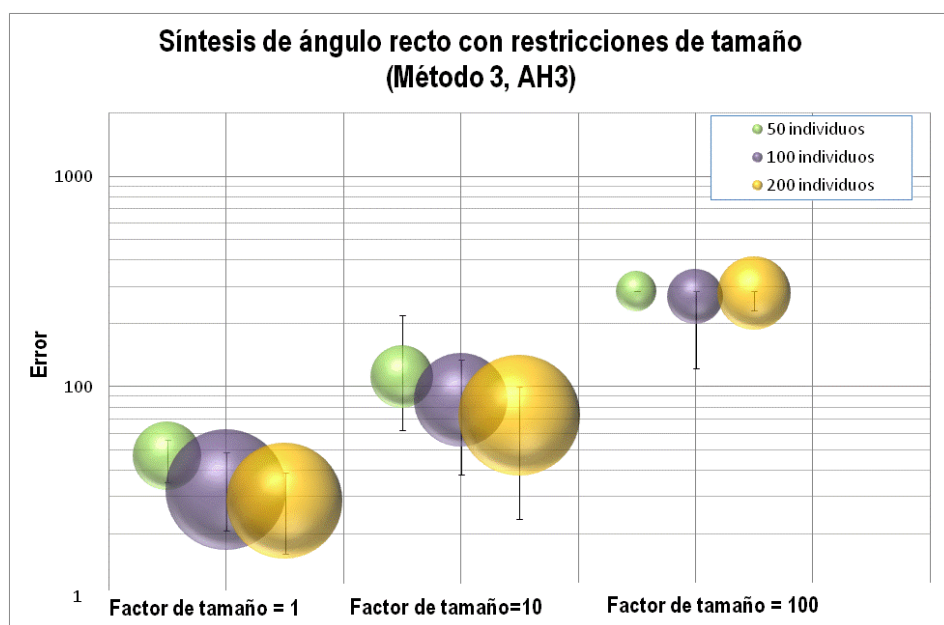


Fig. 6.18 Comparación de resultados obtenidos para el AH3 con restricciones de tamaño (diferentes factores de escala) para la trayectoria objetivo ángulo recto

En cuanto a coste computacional en segundos, se muestra valor medio y desviación estándar en la Tabla 6.12 (para un procesador Intel Core i5 M560 2.53GHz).

	Factor _{escala}	Tiempo _{Medio}	Des _{Estandar}
50 individuos	1	815.80	496.66
	10	601.82	122.36
	100	220.98	75.16
100 individuos	1	2277.47	143.82
	10	1253.78	288.50
	100	438.21	154.51
200 individuos	1	2185.81	315.34
	10	2314.65	89.26
	100	794.07	213.88

Tabla 6.12 Tiempos de procesado de pruebas

El mejor individuo, con mínimo error asociado, se ha obtenido para un factor de escala de tamaño de 1, y se presenta en la Fig. 6.19.

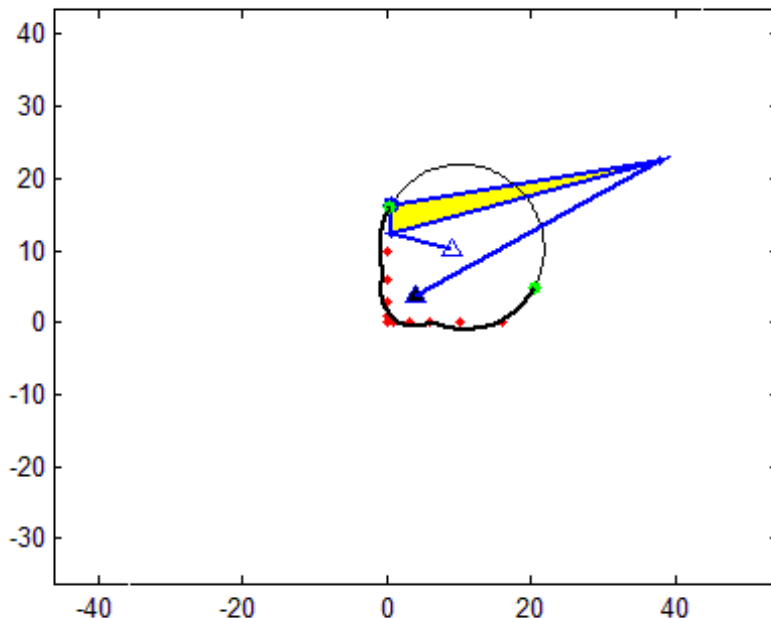


Fig. 6.19 Mejor individuo con restricciones de tamaño para la trayectoria ángulo recto

De la misma forma, los resultados obtenidos para la trayectoria limason se presentan en la Tabla 6.13.

	Factor_{escala}	Error_{Medio}	Mínimo	Máximo	Des_{Estandar}
50 individuos	1	50.42	39.45	69.68	11.34
	10	149.58	15.02	208.35	69.85
	100	438.75	291.74	625.00	116.22
100 individuos	1	50.67	42.62	71.23	8.61
	10	142.73	30.42	210.80	49.29
	100	337.23	56.15	467.43	116.93
200 individuos	1	45.91	31.04	83.28	16.01
	10	95.26	19.74	145.36	46.91
	100	311.64	201.43	489.37	84.62

Tabla 6.13 Errores según factores de escala para trayectoria Limason

En modo gráfico, similar a la anterior los resultados se pueden mostrar según la Fig. 6.20.

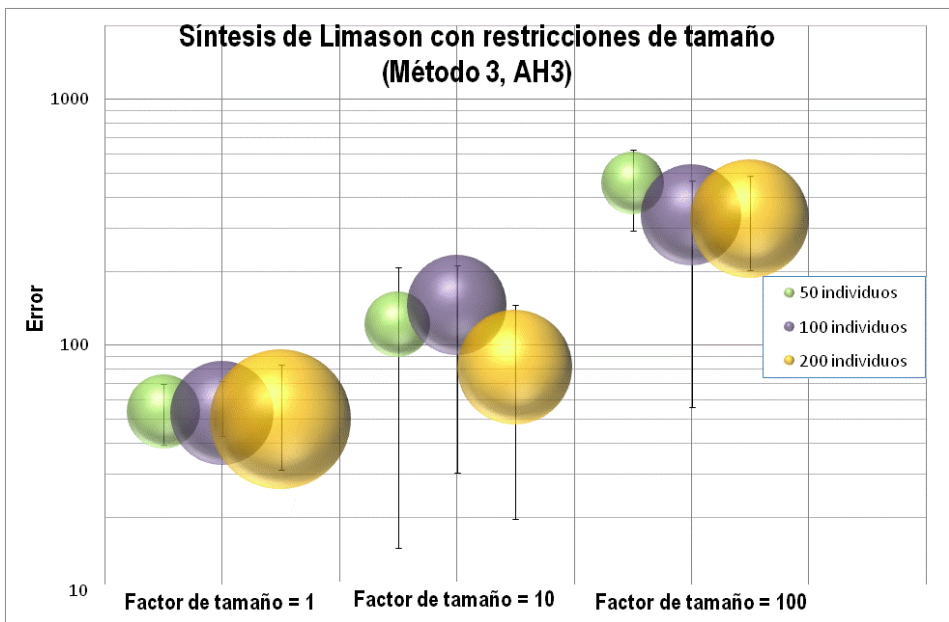


Fig. 6.20 Comparación de resultados obtenidos para el AH3 con restricciones de tamaño (diferentes factores de escala) para la trayectoria objetivo limason

En términos de coste computacional, los tiempos medios y desviaciones estándar en segundos para cada uno de los casos se presentan en la Tabla 6.14 (para un procesador Intel Core i5 M560 2.53GHz), presentando tiempos medios por generación.

	Factor _{escala}	Tiempo _{Medio}	Des _{Estandar}
50 individuos	1	442.09	10.72
	10	434.14	153.72
	100	300.83	64.03
100 individuos	1	877.86	20.59
	10	869.33	218.53
	100	801.70	135.51
200 individuos	1	1619.74	28.85
	10	1144.05	732.47
	100	1247.00	623.87

Tabla 6.14 Tiempos de ejecución para pruebas con trayectoria Limason

El mejor individuo en este caso también para un factor de escala en tamaño de 1, generando la trayectoria limason que se presenta en la Fig. 6.21.

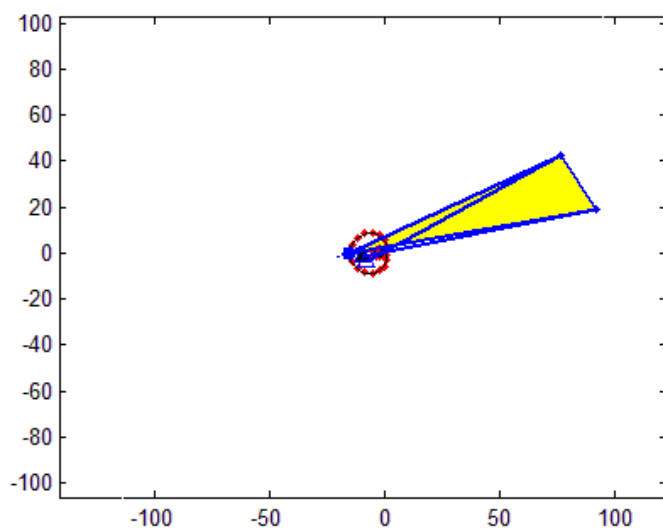


Fig. 6.21 Mejor individuo con restricciones de tamaño para la trayectoria limason

6.8 Aplicación del método 1 y 3 a un mecanismo de inyección

En este apartado se va a aplicar el método 1 y 2 (AH1 y AH2 respectivamente), estimador de error TRS, con traslación giro y escalado, y esto con una optimización local final, a un mecanismo de una máquina de inyección de plástico. El esquema del mecanismo se presenta en la Fig. 6.22. El mecanismo seleccionado para esta aplicación es una combinación de un cuadrilátero articulado con un biela-manivela, a través del acoplador.

Para obtener una buena calidad en la fabricación, el proceso necesita un inicio de inyección rápido, seguido de un avance suave, para finalmente presentar un retroceso rápido. Este requerimiento se puede plasmar en una trayectoria compuesta por 18 puntos, que relacionan el giro de la barra de entrada (con velocidad de giro constante), y la posición de la deslizadera.

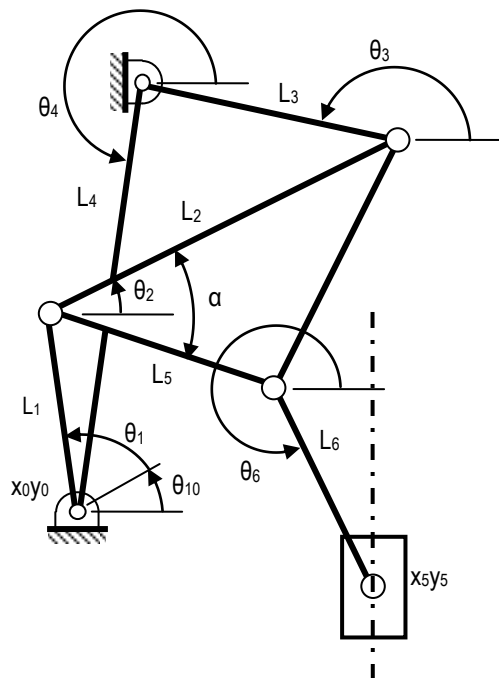


Fig. 6.22 Esquema del mecanismo para inyección de plástico

A continuación se presenta la Tabla 6.15 de puntos de precisión requeridos, y los obtenidos:

	Deseado	Solución método 1	Solución metodo 3
	yd (mm)	yg (mm)	ys (mm)
1	900	892.33	908.6
2	800	757.53	776.8
3	600	588.30	587.4
4	400	417.67	402.7
5	200	274.19	255.0
6	190	178.37	165.4
7	180	137.35	143.2
8	170	140.40	161.5
9	160	161.43	161.6
10	150	173.98	146.8
11	140	166.79	133.0
12	130	143.37	123.9
13	120	113.29	117.4
14	110	88.23	110.3
15	100	83.79	104.2
16	150	125.08	118.4
17	200	250.88	219.1
18	500	491.63	496.4
19	800	777.75	785.5
20	900	937.63	908.4

Tabla 6.15 Puntos de precisión de inyectora

Esto se puede presentar en forma gráfica a través de la Fig. 6.23.

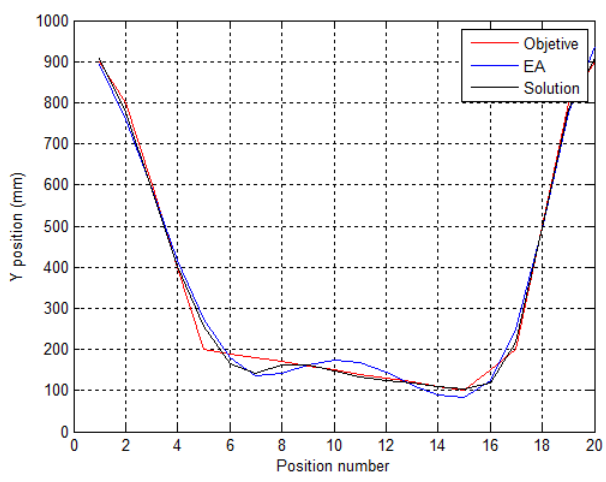


Fig. 6.23 Trayectoria deseada, y obtenida por método 1 (EA) y método 2 (Solution)

Las variables de diseño obtenidas tras el proceso de síntesis se muestran en la Tabla 6.20.

Variables de diseño	Método 1	Método 2
x_0	810.92	537.39
y_0	311.51	393.31
$\theta_{10} \text{ (rad)}$	2.745	2.785
$\theta_4 \text{ (rad)}$	-1.23	-1.4
L_1	517.03	416.52
L_2	992.82	603.75
L_3	991.66	554.85
L_4	1227.4	724.41
L_5	1161.7	611.08
L_6	722.41	372.27
$\alpha \text{ (rad)}$	-0.286	0.2437
x_5	100	100

Tabla 6.16 Mecanismos obtenidos con el Método 1 y el Método 2

En ambos casos, la solución se ajusta con bastante precisión con la función objetivo. Se comprueba así, que los métodos propuestos consiguen buenos resultados ante funciones objetivo con elevados

requerimientos, función no continua con dos puntos singulares; y realizando ejercicio de síntesis con mecanismos diferentes del cuadrilátero articulado.

6.9 Conclusiones de los ejemplos expuestos.

En los ejemplos de métodos híbridos propuestos se han utilizado 5 métodos evolutivos diferentes, 1 método evolutivo y 4 híbridos, para generar 4 trayectorias diferentes y para un tipo de mecanismo, el cuadrilátero articulado (punto 6.6).

Con las 4 trayectorias utilizadas, se ha querido garantizar la validez del método para diferentes tipos de trayectorias, así, se han utilizado trayectorias abiertas y cerradas, y con y sin puntos singulares.

Para poder hacer una comparativa entre todas las trayectorias y para todos los métodos se ha generado la Fig. 6.25 de error similar a las mostradas en los ejemplos, en la que se ha normalizado el error para todas las trayectorias, tomando como referencia error medio 100 para poblaciones de 50 individuos en el método 1, y escalando el resto de errores en la misma proporción.

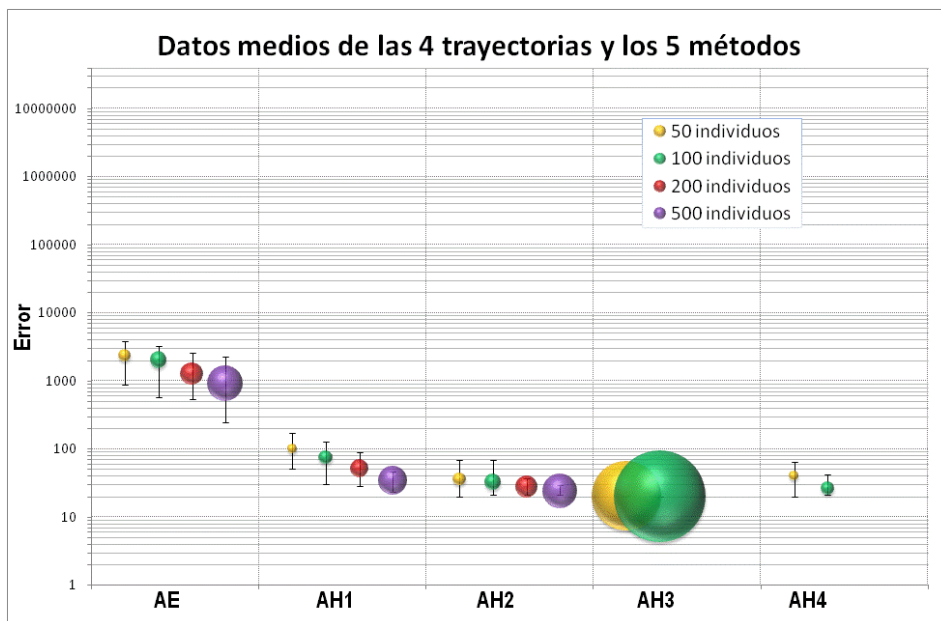


Fig. 6.24 Comparativa de los 5 métodos y las 4 trayectorias

En todos los casos todos los métodos han mostrado resultados aceptables, incluso el método 0, un método evolutivo sin modificación del estimador de error, DMC. Sin embargo, para este primer método, los resultados no son muy precisos, aunque el coste computacional no es elevado.

En el método 1 (AH1), que es el primer método híbrido propuesto, en el que el estimador de error tiene en cuenta la posible traslación, rotación y escalado de la trayectoria, TRS, obteniendo estos parámetros mediante optimizaciones simples, los resultados mejoran sensiblemente, reduciéndose en más de un orden de magnitud el error, y sin incrementar el coste computacional.

Para el método 2 (AH2), segundo de los métodos híbridos propuestos, en el que tras el método evolutivo se aplica una optimización local a los individuos, presenta unos resultados aún mejores que el método 1, aunque un ligero incremento del coste computacional, que se justifica por la reducción de error obtenida.

Desde un punto de vista de dispersión de resultados, los 3 primeros métodos, obtienen dispersiones bajas sólo cuando se aplica el nuevo estimador de error y para poblaciones con alto número de individuos, 500 individuos por generación, que como es lógico conlleva un mayor coste computacional.

El método 3 (AH3), presenta los mejores resultados, tanto en valor de error, como en dispersión del mismo. Pero al utilizar un método híbrido, con el estimador de error propuesto, EOL, y con una optimización local para todos los individuos, el coste computacional es el mayor de todos los métodos.

Finalmente, para el método 4 (AH4), en el que se utiliza un estimador de error con traslación, giro y escalado, TRS en casi todos los individuos, y el estimador de error EOL, con optimización local sólo en la élite, los resultados en error también son buenos, con baja dispersión para 100 individuos por generación, y con un coste computacional reducido.

Con lo expuesto hasta ahora, se puede concluir que desde un punto de vista de consecución de mínimo error el método 3 (AH3), es el mejor método, pero con un elevado coste computacional. Sin embargo, desde un punto de vista práctico, y a nivel de aplicación industrial, los métodos 3 (AH2) y 4 (AH4) ofrecen buenos resultados, con un coste computacional mucho más comedido.

Para analizar estos métodos con restricciones de tamaño se han contrastado frente a varias trayectorias objetivo en el punto 6.7 de este trabajo. Como resumen de este punto, se presenta la Fig. 6.25 en la que se han combinado los resultados de las pruebas realizadas con las diferentes trayectorias, tomando como referencia el valor de error medio para factor de escala 10 con 100 individuos, y escalando el resto de valores de error considerando error medio 100 para este caso.

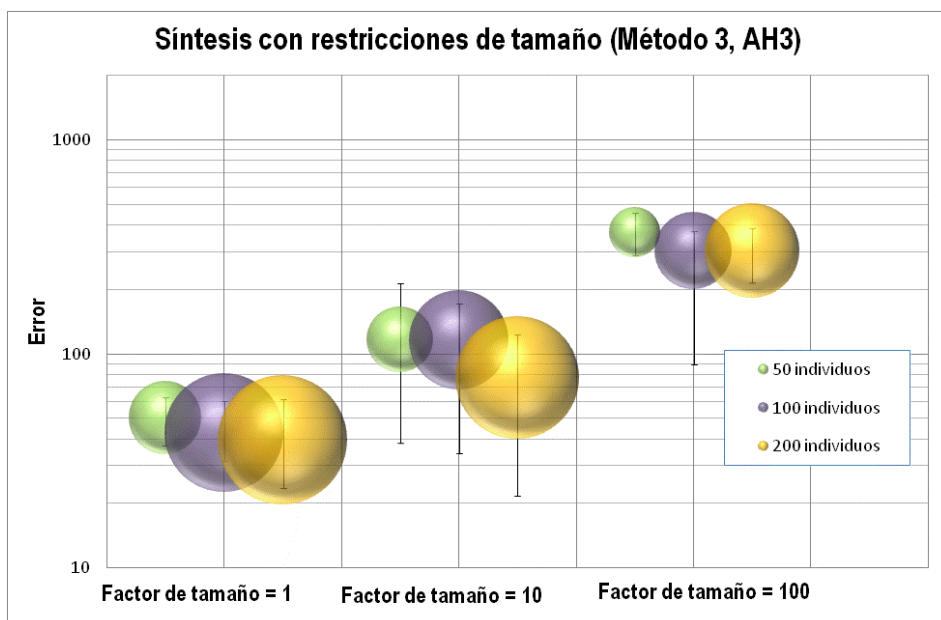


Fig. 6.25 Comparación de resultados obtenidos para el AH3 con restricciones de tamaño (diferentes factores de escala) para las trayectorias objetivo evaluadas

De la Fig. 6.25 se puede concluir que:

- La utilización de una penalización por tamaño en la función objetivo es una herramienta adecuada para obtener mecanismos de tamaño reducido.
- Penalizar en exceso la función objetivo por el tamaño del mecanismo provoca mayor dispersión de resultados, y mecanismos que priman el tener un tamaño pequeño y no obtener una trayectoria adecuada.
- En cuanto al tamaño de población, los resultados mejoran aumentando la misma, pero con un mayor coste computacional asociado.

Como comentario adicional, se puede reseñar que los métodos presentados a lo largo de este capítulo tienen una de sus bases en poder escalar el mecanismo para aprovechar toda o parte de la trayectoria generada por el mismo, ajustando su tamaño al requerido por la función objetivo. Esto lleva en algunos casos, como en la trayectoria limason, a la obtención de mecanismos de gran tamaño, por lo que limitar el tamaño del mecanismo a costa de reducir la precisión en la trayectoria parece adecuado. Sin embargo, en otras trayectorias, como el ángulo recto, el tamaño del mecanismo es más reducido, de forma que si se penaliza la función objetivo con mayores tamaños, se obtienen resultados en trayectoria que en algunos casos pueden carecer de la precisión necesaria. Debido a esto, parece adecuado no utilizar penalización por tamaño en este tipo de trayectoria, o si se emplea, hacerlo con un factor de escala por tamaño reducido.

Finalmente, aplicando los métodos propuestos a un mecanismo diferente, tomando como ejemplo un mecanismo de una máquina inyectora, como se ha mostrado en el punto 6.8. Si se trata de resolver el problema de síntesis planteado en este punto por los métodos 1, 2 y 3 (AH1, AH2 y AH3 respectivamente) con 50, 100, 200 y 500 individuos se obtiene la Fig. 6.26 con datos estadísticos.

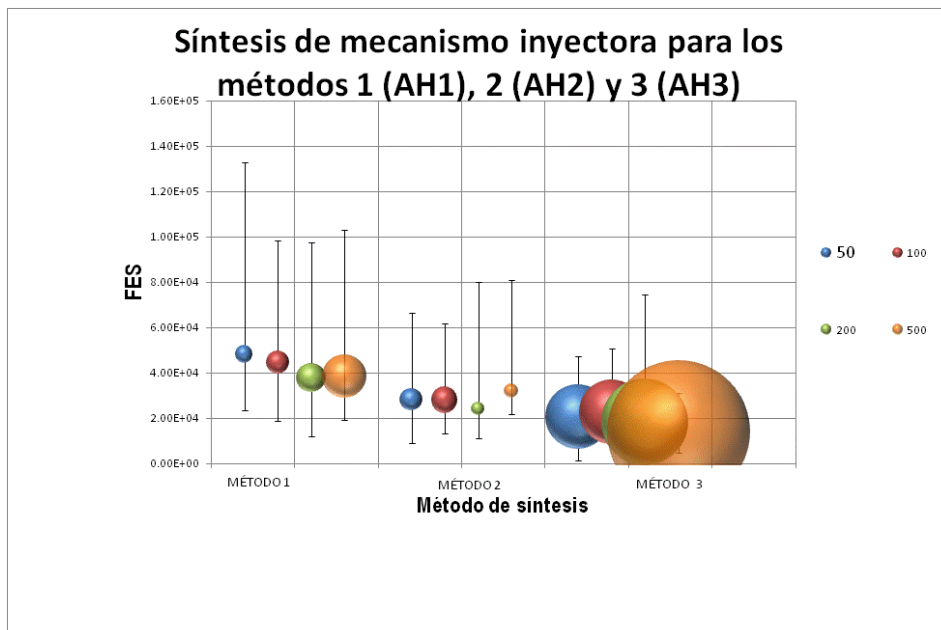


Fig. 6.26 Comparativa de AH1, AH2 y AH3 para obtener el objetivo de la máquina inyectora

Los valores de la gráfica se presentan tabulados en la Tabla 6.17.

	<i>Individuos</i>	<i>Error</i>				<i>Tiempo</i>	
		<i>Medio</i>	<i>Máximo</i>	<i>Mínimo</i>	<i>Std</i>	<i>Medio</i>	<i>Std</i>
Método 1	50	4.81E+04	8.47E+04	2.45E+04	1.93E+04	2.45E+00	3.64E-01
	100	4.45E+04	5.40E+04	2.58E+04	9.92E+03	4.22E+00	1.19E+00
	200	3.78E+04	5.96E+04	2.56E+04	1.05E+04	6.94E+00	2.18E+00
	500	3.85E+04	6.43E+04	1.92E+04	1.36E+04	1.59E+01	7.06E+00
Método 2	50	2.83E+04	3.80E+04	1.96E+04	7.24E+03	4.00E+00	4.50E-01
	100	2.79E+04	3.37E+04	1.47E+04	6.28E+03	5.71E+00	1.19E+00
	200	2.42E+04	5.61E+04	1.29E+04	1.32E+04	1.36E+00	4.22E-01
	500	3.22E+04	4.89E+04	1.03E+04	1.57E+04	1.57E+00	1.39E-01
Método 3	50	2.06E+04	2.69E+04	1.34E+04	4.30E+03	3.50E+01	1.21E+00
	100	2.25E+04	2.82E+04	8.69E+03	6.50E+03	3.50E+01	1.21E+00
	200	1.84E+04	5.61E+04	6.09E+03	1.54E+04	6.33E+01	1.89E+01
	500	1.40E+04	1.71E+04	9.32E+03	2.68E+03	1.74E+02	1.20E+01

Tabla 6.17 Valores de error y tiempo de ejecución para diferentes métodos

De estos valores, así como de la Fig. 6.23, se puede concluir que los métodos propuestos en este trabajo no sólo obtienen resultados de síntesis con un error reducido para diferentes trayectorias objetivo, sino que además también son válidos para otros mecanismos diferentes al cuadrilátero articulado, como el analizado en el punto 6.8 de este trabajo, obteniendo también errores de síntesis reducidos, y costes computacionales aceptables.

Capítulo 7

“La vida es como andar en bicicleta. Debes seguir moviéndote para mantener el equilibrio.”

Albert Einstein

7 Conclusiones y trabajos futuros

7.1 Conclusiones

Este trabajo se ha centrado en la mejora de los métodos de síntesis dimensional en mecanismos planos para obtener un método más eficiente, robusto y preciso. Para ello se han planteado diversas aportaciones que van desde la mejora de la evaluación del error hasta la hibridación de los algoritmos de síntesis.

Una de las aportaciones más novedosas que se ha propuesto y desarrollado en este trabajo es un nuevo estimador de error, denominado TRS, con el que se consigue comparar curvas de forma eficiente, tanto abiertas como cerradas, y sin tener en cuenta la posible traslación, rotación u homotecia existente entre ellas. Este estimador de error es independiente del número de puntos de precisión utilizados en la definición de la curva generada y deseada. Resuelve además el problema de punto de inicio de cada curva haciendo que la evaluación del error sea independiente de la numeración arbitraria de los puntos de precisión. El estimador de error propuesto es eficiente y conlleva un coste computacional bajo, por lo que su uso es viable en métodos de optimización locales y en métodos de optimización estocásticos.

Otro aspecto importante aportado por este trabajo ha sido la mejora de algunos de los procesos de los algoritmos de síntesis óptima empleando métodos deterministas. Como primer aspecto a destacar se puede mencionar que la optimización local se aborda mediante un proceso de optimización estructurado en varias fases. Esta división de la optimización en fases obtiene buenos resultados debido a la rápida convergencia en los primeros estadios de la optimización, que permiten además una adecuación del individuo de partida a la trayectoria objetivo antes de comenzar el proceso de optimización morfológica del mecanismo. Esta aportación proporciona importantes mejoras en la eficiencia y precisión del algoritmo.

Otro de los problemas abordados ha sido la singularidad de la matriz jacobiana en la búsqueda de la dirección de descenso en el proceso de optimización. La inversión de la matriz jacobiana para la obtención de la

dirección de descenso ha modificado su formulación previa, realizando una inversión de dicha matriz mediante la inversa generalizada o pseudoinversa de Moore-Penrose, obteniéndose mejores resultados incluso considerando posiciones muy cercanas a posiciones singulares del mecanismo.

Otra aportación propuesta para la mejora de la síntesis óptima local ha sido el desarrollo de un algoritmo eficiente en la determinación del módulo del vector de descenso. Este módulo se ha formulado como variable, cambio de paso variable, que permite acelerar la convergencia del proceso de síntesis.

En general, se puede decir que las aportaciones en el campo de la síntesis óptima local mediante métodos deterministas proporcionan una mejora en la robustez, precisión y eficiencia del métodos consiguiéndose un coste computacional razonable para la resolución de problemas complejos. Por tanto, el método obtenido puede utilizarse de forma intensiva (como se ha mostrado en los ejemplos) en cualquier tipo de mecanismo plano.

En cuanto a métodos de síntesis dimensional empleando algoritmos evolutivos se ha generado una rutina en base a dos estimadores de error, DMC y TRS. Dentro de la rutina se han desarrollado operadores de cruce, mutación y elitismo propios adaptadas a las condiciones particulares de los métodos de diseño de mecanismos. Esta rutina se ha aplicado a diferentes trayectorias constatando su validez para todas ellas. La validez del método se ha mostrado para trayectorias abiertas y cerradas, con y sin puntos singulares.

Se han analizado también estos métodos con restricciones de tamaño, mediante penalización en la función objetivo, mostrando su validez para este tipo de problemas. En este caso el factor de penalización en función del tamaño debe estudiarse en cada caso según la dificultad para obtener la trayectoria objetivo.

Un aspecto fundamental de las aportaciones realizadas en este trabajo de Tesis Doctoral ha sido la hibridación de los métodos de síntesis óptima. Es decir, se han combinado las ventajas de los métodos de optimización deterministas (optimización local) con las ventajas de los métodos de optimización estocásticos (optimización global) para obtener un algoritmo altamente eficiente e independiente del punto de partida.

Se ha demostrado que la hibridación puede realizarse de diversas maneras obteniendo resultados que varían en cuanto a su eficiencia y precisión. Los cuatro métodos híbridos propuestos han mostrado su validez para trayectorias abiertas y cerradas, con posibilidad de puntos singulares en las mismas.

El algoritmo híbrido 1, denominado algoritmo AH1 (método evolutivo con estimador de error TRS), muestra resultados sensiblemente mejores a los mostrados por el método evolutivo convencional (sin hibridación), reduciendo en más de un orden de magnitud el error sin incrementar el coste computacional.

El algoritmo híbrido AH2, con una optimización local una vez finalizado el algoritmo AH1, ha mostrado mejores resultados que el método AH1, aunque conlleva un ligero aumento del coste computacional justificado por la reducción de error obtenida.

El tercer algoritmo híbrido propuesto, AH3, presenta los mejores resultados desde un punto de vista de la reducción del error como en dispersión del mismo. Este método conlleva la utilización como estimador de error el algoritmo de síntesis local EOL, por lo que al realizarse una optimización local en todos los individuos el aumento de precisión se ve mermado por el aumento considerablemente del coste computacional.

Finalmente el algoritmo híbrido AH4, utilizando el estimador de error EOL sólo en la élite de cada generación, muestra resultados muy similares en cuanto a precisión al método AH3, pero con un coste computacional significativamente inferior.

Como conclusión sobre los métodos híbridos evaluados en este trabajo, y considerando la precisión como factor determinante, el algoritmo que mejor resultados proporciona es el AH3. Sin embargo, desde un punto de vista práctico en ingeniería de diseño, y considerando la precisión y eficiencia como factores determinantes, la aplicación más interesante a nivel industrial son los algoritmos AH2 y AH4, ya que ofrecen buenos resultados con un coste computacional reducido.

7.2 Trabajos futuros

Tomando como base este trabajo, se podrían seguir desarrollando otros trabajos, entre los que se podrían resaltar:

Desarrollo de métodos de **síntesis dimensional óptima multiobjetivo**, pudiéndose considerar objetivos mixtos de cinemática y dinámica de los mecanismos a diseñar.

Síntesis de tipo, desarrollando métodos que sean capaces de evaluar y comparar diferentes topologías de mecanismos, así como otros aspectos tales como complejidad de la solución, procesos de fabricación, mantenimiento, etc.

Desarrollo de un entorno gráfico integrado que permita utilizar los desarrollos generados de forma fácil e intuitiva.

Mecanismos espaciales. Los métodos desarrollados en este trabajo son susceptibles de ser empleados en problemas 3D sin gran dificultad.

Desarrollo de métodos que aborden problemas dinámicos, no trabajando únicamente con la cinemática de los diferentes mecanismos, y pudiendo incluir como objetivo diferentes perfiles de fuerza o relación de transmisión de fuerza o par

Desarrollo de métodos que incluyan la flexibilidad de los elementos, ya que ciertos ejercicios de diseño implican elementos con alta flexibilidad no despreciable en la evaluación de la solución.

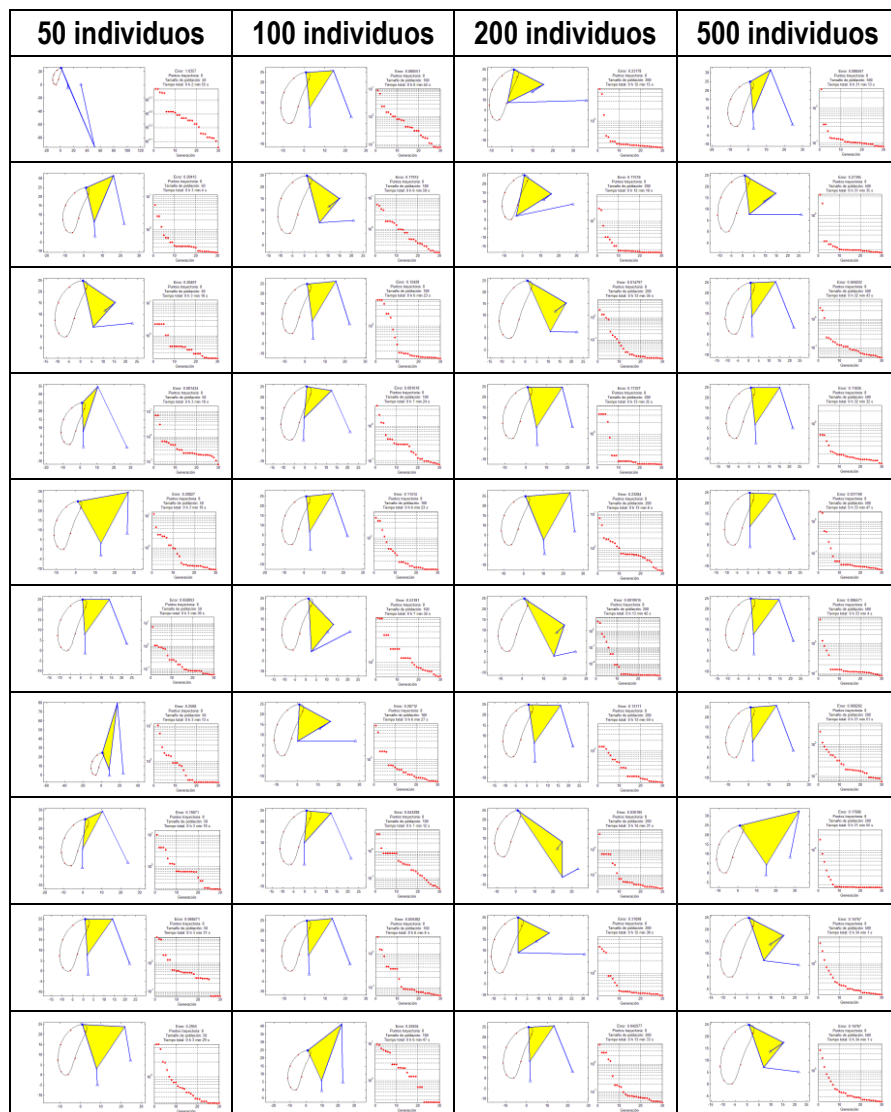
Anexo A

Resultados de la Prueba 1: Los mecanismos y trayectorias obtenidas se muestran a continuación



Anexo B

Los mecanismos y trayectorias obtenidas en la prueba 2 se presentan a continuación:



Anexo C

El resultado de las diferentes pruebas de síntesis en la prueba 3 se presentan en la siguiente tabla:

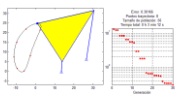
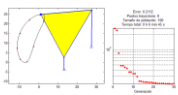
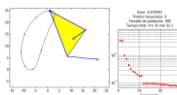
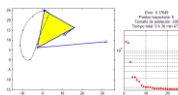
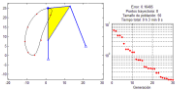
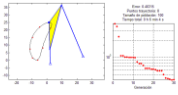
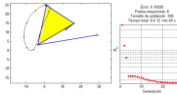
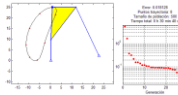
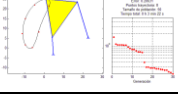
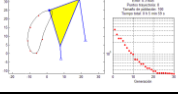
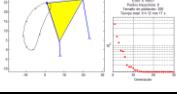
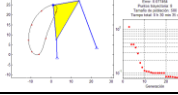
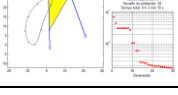
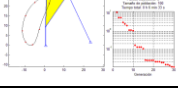
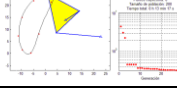
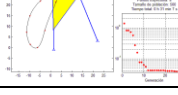
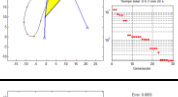
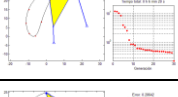
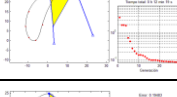
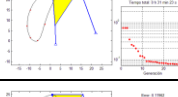
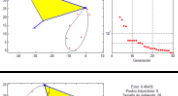
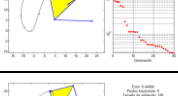
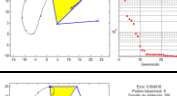
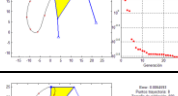
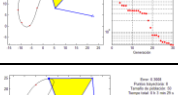
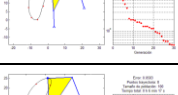
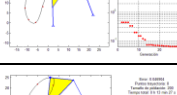
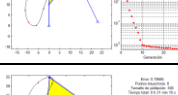
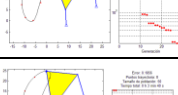
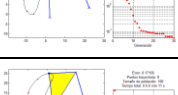
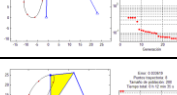
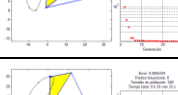
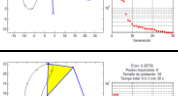
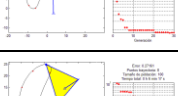
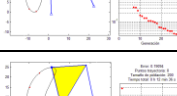
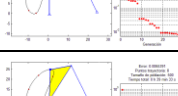
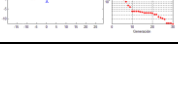
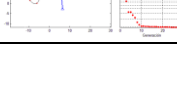
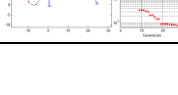
50 individuos	100 individuos	200 individuos	500 individuos
			
			
			
			
			
			
			
			
			
			

Figure 1 displays the effect of sample size on the estimation of the probability density function (PDF) and the cumulative distribution function (CDF) for a bivariate distribution. The figure is organized into a grid with 4 columns representing sample sizes (50, 100, 200, 500 individuals) and 10 rows representing different parameter values (alpha = 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9). Each plot shows the PDF (yellow shaded area) and the CDF (red dashed line) for a given sample size. The x-axis represents the first variable (ranging from -10 to 10) and the y-axis represents the second variable (ranging from -10 to 10). The plots are arranged in a grid where rows represent different parameter values and columns represent sample sizes. As the sample size increases, the estimated PDF and CDF converge to the true distribution, which is shown as a black line and a solid red line, respectively.

Anexo E

Función objetivo: Trayectoria en L

Métodos evaluados:

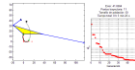
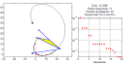
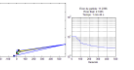
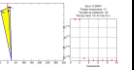
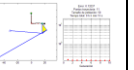
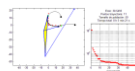
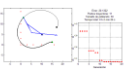
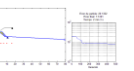
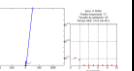
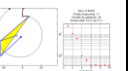
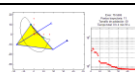
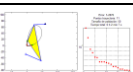
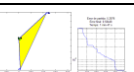
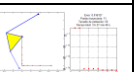
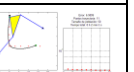
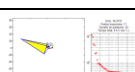
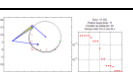
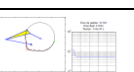
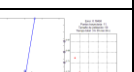
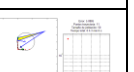
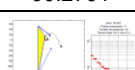
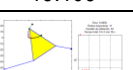
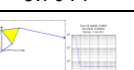
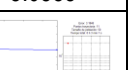
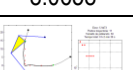
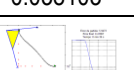
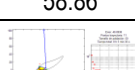
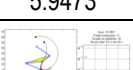
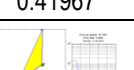
AE: Algoritmo evolutivo con estimador de error DMC

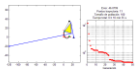
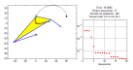
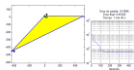
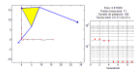
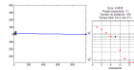
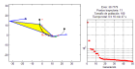
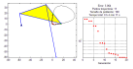
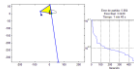
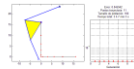
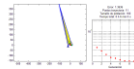
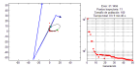
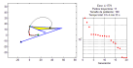
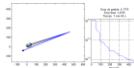
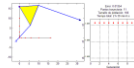
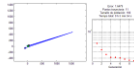
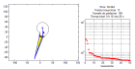
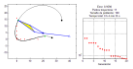
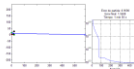
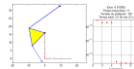
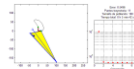
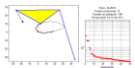
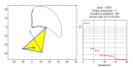
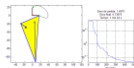
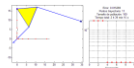
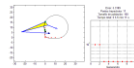
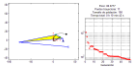
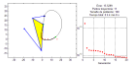
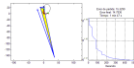
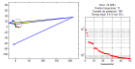
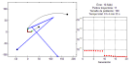
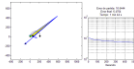
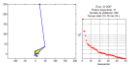
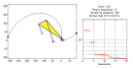
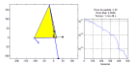
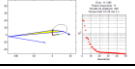
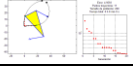
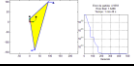
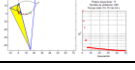
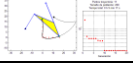
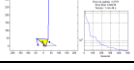
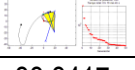
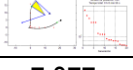
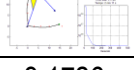
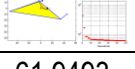
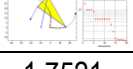
AH1: Algoritmo evolutivo con estimador de error TRS

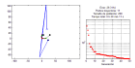
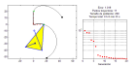
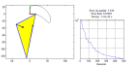
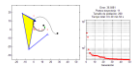
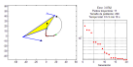
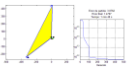
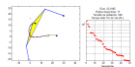
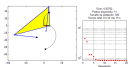
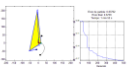
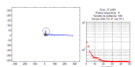
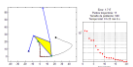
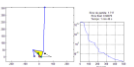
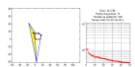
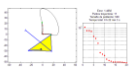
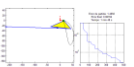
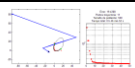
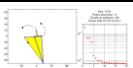
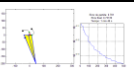
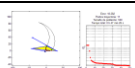
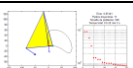
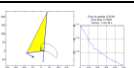
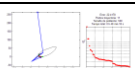
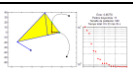
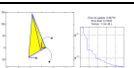
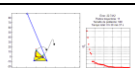
AH2: Algoritmo híbrido 2 como AH1 con optimización local final en los individuos de la última generación

AH3: Algoritmo híbrido con estimador de error EOL (optimización local en estimador de error en todos los individuos)

AH4: Algoritmo híbrido con estimador EOL en la élite y estimador TRS en el resto de individuos

Método	AE	AH1	AH2	AH3	AH4
50 individuos por generación	 41.0584	 11.2396	 2.7455	 0.20411	 0.13237
	 30.5493	 20.1322	 7.5101	 0.18142	 0.36763
	 79.5605	 5.2876	 0.58448	 0.016127	 6.9099
	 36.2731	 13.103	 5.7014	 0.18458	 5.0885
	 43.021	 8.0658	 0.038103		 3.1848
	 58.86	 5.9473	 0.41967		
	 40.0938	 13.7807	 2.6695		

Método	AE	AH1	AH2	AH3	AH4
100 individuos por generación					
	46.0799	10.8885	0.66643	0.017059	0.8038
					
	50.7179	5.064	3.5439	0.045942	1.3636
					
	51.1456	4.1779	1.4785	0.01354	1.6479
					
	56.642	8.9598	1.9508	0.013883	0.5496
					
	34.2619	1.4979	0.73079	0.01539	5.3189
					
	80.877	15.299	14.7836		
					
	75.9261	10.6444	6.4704		
200 individuos por generación					
	21.8597	5.25	2.6936		
					
	11.3105	4.9593	1.6498		
					
	28.2677	3.2779	3.2779		
					
	33.3417	7.877	3.1726		
					
	61.9493	1.7521	1.0628		

Método	AE	AH1	AH2	AH3	AH4
					
	29.3114	1.019	0.61051		
					
500 individuos por generación	35.5061	3.8762	1.4797		
					
	22.0165	0.03762	0.6729		
					
	37.4359	1.717	0.66079		
					
	34.5785	1.4892	0.86764		
					
	19.4308	0.781	0.73139		
					
	18.282	0.85141	0.79495		
					
	22.4176	0.86778	0.57835		
					
	22.7412				

Anexo F

Función objetivo: Trayectoria elipse con punto angularo

Métodos evaluados:

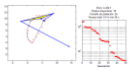
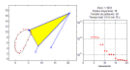
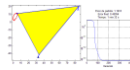
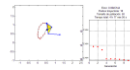
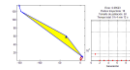
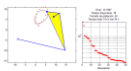
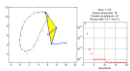
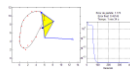
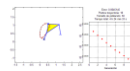
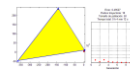
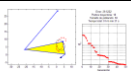
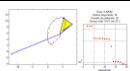
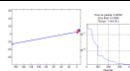
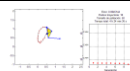
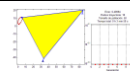
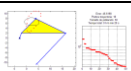
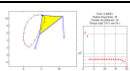
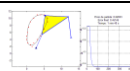
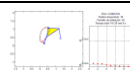
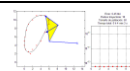
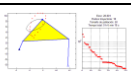
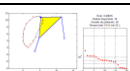
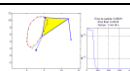
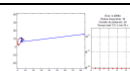
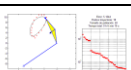
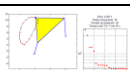
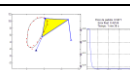
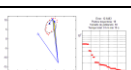
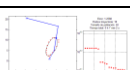
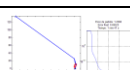
AE: Algoritmo evolutivo con estimador de error DMC

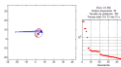
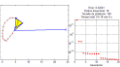
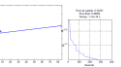
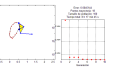
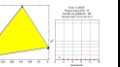
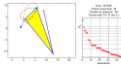
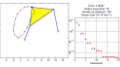
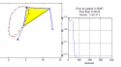
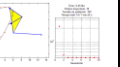
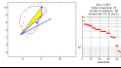
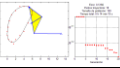
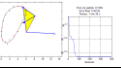
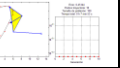
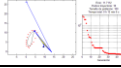
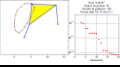
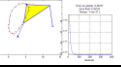
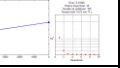
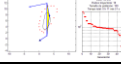
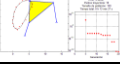
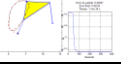
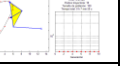
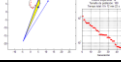
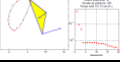
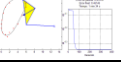
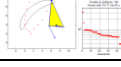
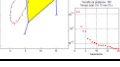
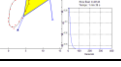
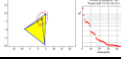
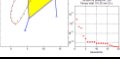
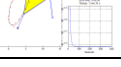
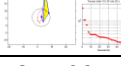
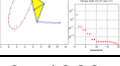
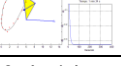
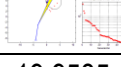
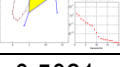
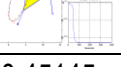
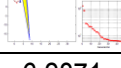
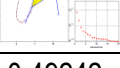
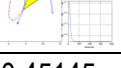
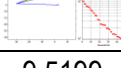
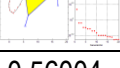
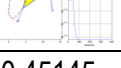
AH1: Algoritmo evolutivo con estimador de error TRS

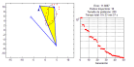
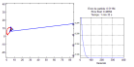
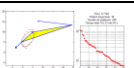
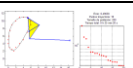
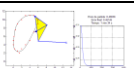
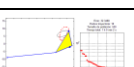
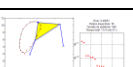
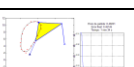
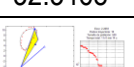
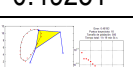
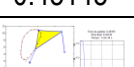
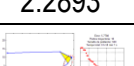
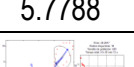
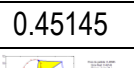
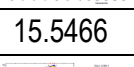
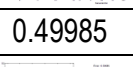
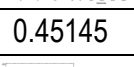
AH2: Algoritmo híbrido 2 como AH1 con optimización local final en los individuos de la última generación

AH3: Algoritmo híbrido con estimador de error EOL (optimización local en estimador de error en todos los individuos)

AH4: Algoritmo híbrido con estimador EOL en la élite y estimador TRS en el resto de individuos

Método	AE	AH1	AH2	AH3	AH4
50 individuos por generación					
	4.2241	1.1633	0.48994	0.0045144	0.69523
					
	13.7807	1.175	0.45145	0.0045145	0.49527
					
	29.522	0.92582	0.92592	0.0045144	0.48934
					
	46.6198	0.62063	0.45145	0.0045145	0.45144
					
	28.801	0.68616	0.45145		0.48954
					
	5.1244	0.5671	0.45145		
					
	12.6463	1.2588	0.69523		

Método	AE	AH1	AH2	AH3	AH4
100 individuos por generación					
	41.356	0.52221	0.48994	0.0045144	0.49526
					
	18.9668	0.56467	0.45145		0.45144
					
	5.4077	0.5166	0.45145		0.45144
					
	11.7112	0.46727	0.45145		0.51826
					
	29.935	0.62047	0.45145		0.45144
					
	5.2533	0.87876	0.45145		
					
	28.1814	0.5307	0.45145		
200 individuos por generación					
	16.3904	0.5728	0.45145		
					
	27.7935	0.51323	0.45145		
					
	19.9585	0.5021	0.45145		
					
	6.2871	0.49242	0.45145		
					
	9.5199	0.56004	0.45145		

Método	AE	AH1	AH2	AH3	AH4
					
	11.8867	0.51194	0.48994		
					
500 individuos por generación	6.7748	0.49038	0.49038		
					
	32.3468	0.49251	0.45145		
					
	2.2893	0.48193	0.45145		
					
	5.7788	0.45371	0.45145		
					
	20.2017	0.52515	0.45145		
					
	19.0466	0.54008	0.45145		
					
	15.5466	0.49985	0.45145		
					
	8.9611	0.50686	0.48994		

Anexo G

Función objetivo: Trayectoria gota de Agua

Métodos evaluados:

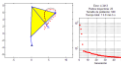
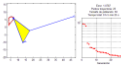
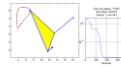
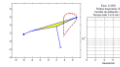
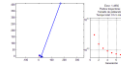
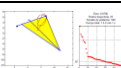
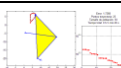
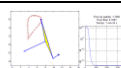
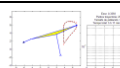
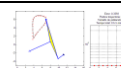
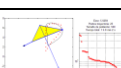
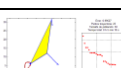
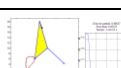
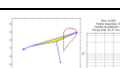
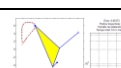
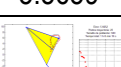
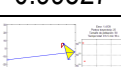
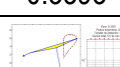
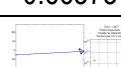
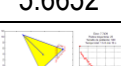
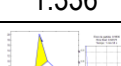
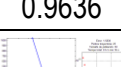
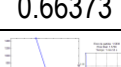
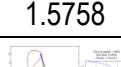
AE: Algoritmo evolutivo con estimador de error DMC

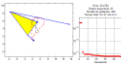
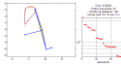
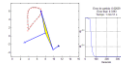
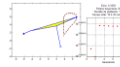
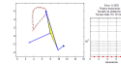
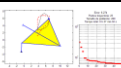
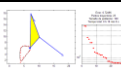
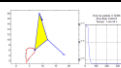
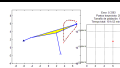
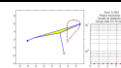
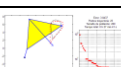
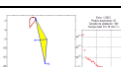
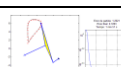
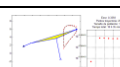
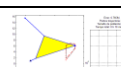
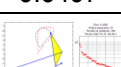
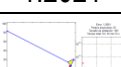
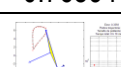
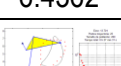
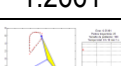
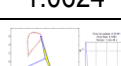
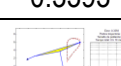
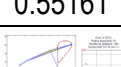
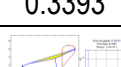
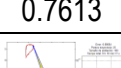
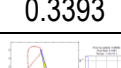
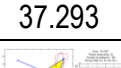
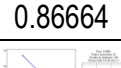
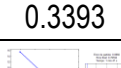
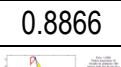
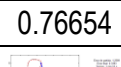
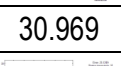
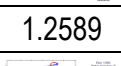
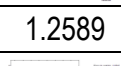
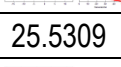
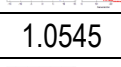
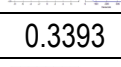
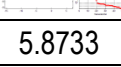
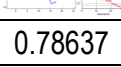
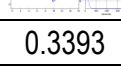
AH1: Algoritmo evolutivo con estimador de error TRS

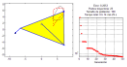
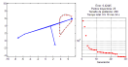
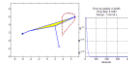
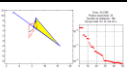
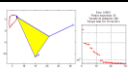
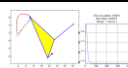
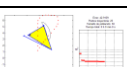
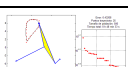
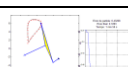
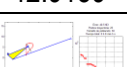
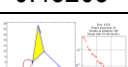
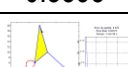
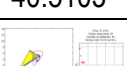
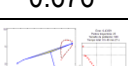
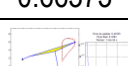
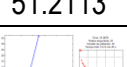
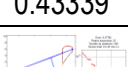
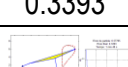
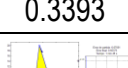
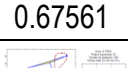
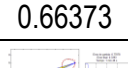
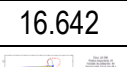
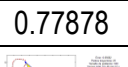
AH2: Algoritmo híbrido 2 como AH1 con optimización local final en los individuos de la última generación

AH3: Algoritmo híbrido con estimador de error EOL (optimización local en estimador de error en todos los individuos)

AH4: Algoritmo híbrido con estimador EOL en la élite y estimador TRS en el resto de individuos

Método	AE	AH1	AH2	AH3	AH4
50 individuos por generación					
	4.3413	1.0787	0.66373	0.3393	1.4956
					
	8.8706	1.7268	0.3393	0.3393	0.3393
					
	5.5659	0.99527	0.65373	0.3393	0.66373
					
	5.6652	1.4129	1.336	0.3393	1.4077
					
	7.7439	0.9636	0.66373		0.76594
					
	17.3047	1.5838	1.5758		
					
	6.6434	1.2582	0.33931		

Método	AE	AH1	AH2	AH3	AH4
100 individuos por generación					
	22.4764	0.62629	0.3393	0.3393	0.3393
					
	6.274	0.72496	0.66373	0.3303	0.3393
					
	3.8437	1.2021	0.3393	0.3393	0.76594
					
	6.4362	1.2661	1.0624		0.3393
					
	13.701	0.55161	0.3393		0.3393
					
	19.2993	0.7613	0.3393		
					
	37.293	0.86664	0.3393		
200 individuos por generación					
	10.7837	0.8866	0.76654		
					
	30.969	1.2589	1.2589		
					
	25.5309	1.0545	0.3393		
					
	5.8733	0.78637	0.3393		
					
	18.2378	0.68889	0.66373		

Método	AE	AH1	AH2	AH3	AH4
					
	8.2653	0.42445	0.3393		
					
500 individuos por generación	29.2306	0.8615	0.66373		
					
	42.9139	0.43208	0.3393		
					
	40.5103	0.676	0.66373		
					
	51.2113	0.43339	0.3393		
					
	55.8676	0.57785	0.3393		
					
	51.5875	0.67561	0.66373		
					
	16.642	0.77878	0.3393		
					
	22.588	0.65662	0.66373		

Anexo H

Función objetivo: Trayectoria Limason

Métodos evaluados:

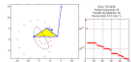
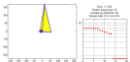
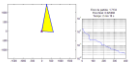
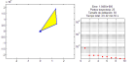
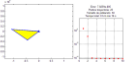
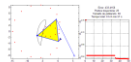
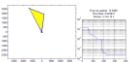
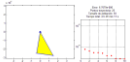
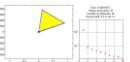
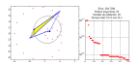
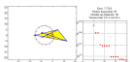
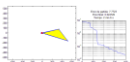
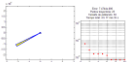
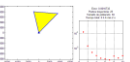
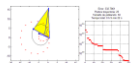
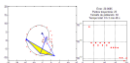
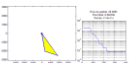
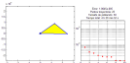
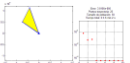
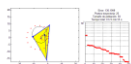
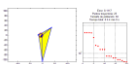
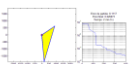
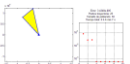
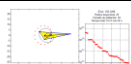
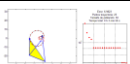
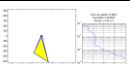
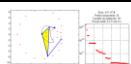
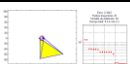
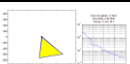
AE: Algoritmo evolutivo con estimador de error DMC

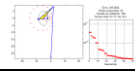
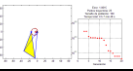
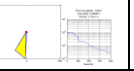
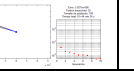
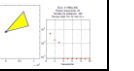
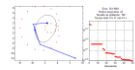
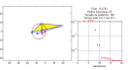
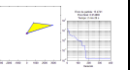
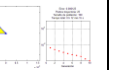
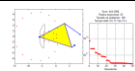
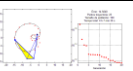
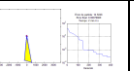
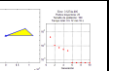
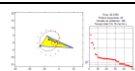
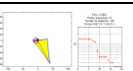
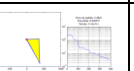
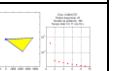
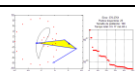
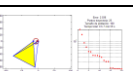
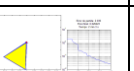
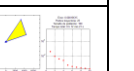
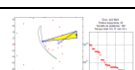
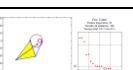
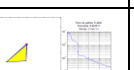
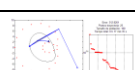
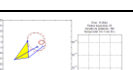
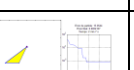
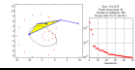
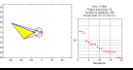
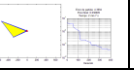
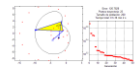
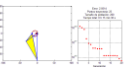
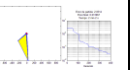
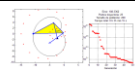
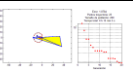
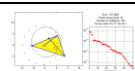
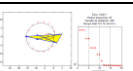
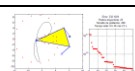
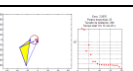
AH1: Algoritmo evolutivo con estimador de error TRS

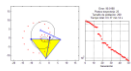
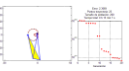
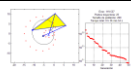
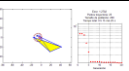
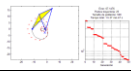
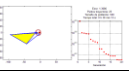
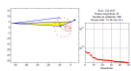
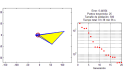
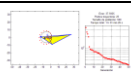
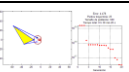
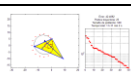
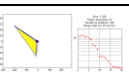
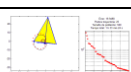
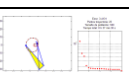
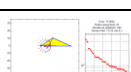
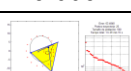
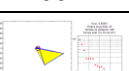
AH2: Algoritmo híbrido 2 como AH1 con optimización local final en los individuos de la última generación

AH3: Algoritmo híbrido con estimador de error EOL (optimización local en estimador de error en todos los individuos)

AH4: Algoritmo híbrido con estimador EOL en la élite y estimador TRS en el resto de individuos

Método	AE	AH1	AH2	AH3	AH4
50 individuos por generación					
	355.6438	1.7134	0.025088	1.0455e-5	7.8269e-5
					
	433.4156	15.6489	0.023831	6.7079e-6	0.00015617
					
	294.7294	7.7723	0.024726	7.4744e-6	0.0010732
					
	332.7861	20.0085	0.084166	1.0045e-5	3.8105e-5
					
	130.199	8.1117	0.026073		3.4348e-5
					
	158.1296	6.9023	0.023546		
					
	417.1778	2.3243	0.027636		

Método	AE	AH1	AH2	AH3	AH4
100 individuos por generación					
	326.6824	1.0615	0.028675	5.8215e-6	6.1986e-5
					
	301.9861	13.4741	0.014885		0.000125
					
	444.5984	14.9248	0.00070869		3.5274e-5
					
	82.0356	2.2843	0.030757		0.00013737
					
	370.2761	2.036	0.026223		0.00018515
					
	443.9821	8.4446	0.023311		
					
	312.8201	13.0544	0.0054127		
200 individuos por generación					
	314.2375	4.3894	0.038609		
					
	126.7828	2.8514	0.031067		
					
	108.3362	1.8784			
					
	107.8801	9.8217			
					

Método	AE	AH1	AH2	AH3	AH4
	332.1071	3.2676			
					
	90.0168	2.3069			
					
	119.537	1.2702			
500 individuos por generación					
	67.1476	1.3086			
					
	334.1017	0.44504			
					
	37.5935	4.476			
					
	42.4892	1.228			
					
	19.9498	3.4131			
					
	73.0034	2.3812			
					
	63.4848	0.85861			

Referencias

- Abadie, J. & Carpenter, J., 1969. Generalization of the Wolfe Reduced Gradient. In: *Optimisation*. s.l.:Academic Press, pp. 37-47.
- Acharyya, S. & Mandal, M., 2009. Performance of EAs for four-bar linkage synthesis. *Mechanism and Machine Theory*.
- Ahmad, A. & Waldron, K. J., 1979. Synthesis of adjustable planar 4-bar. *Mechanism and Machine Theory*, 14 (6), p. 405 – 411.
- Akçali, I. & Dittrich, G., 1989. Path generation by subdomain method. *Mechanism and Machine Theory*, 24 (1), p. 45 – 52.
- Akhras, R. & Angeles, J., 1990. Unconstrained nonlinear least-square optimization. *Mechanism and Machine Theory*, 25 (1), p. 97 – 118.
- Al-Dwairi, A., Dweiri, F. & Ashour, O., 2010. A novice-centered decision-support system for type synthesis of function-generation mechanisms. *Mechanism and Machine Theory* 45, pp. 1252-1268.
- Arora, J. S., 2004. *Introduction to optimum design*. San Diego, California: Elsevier Academic Press.
- Bagci, C., 1977. Synthesis of double-crank (drag-link) driven mechanisms with adjustable motion and dwell time ratios. *Mechanism and Theory*, 12 (6), p. 619 – 638.
- Belegundu, A. D. & Chandrupatla, T. R., 1999. *Optimization concepts and applications in engineering*. Upper Saddle River New Jersey: Prentice-Hall.
- Bulatovic, R. & Dordevic, S., 2009. On the optimum synthesis of a four-bar linkage using differential evolution and method of variable controlled deviations. *Mechanism and Machine Theory* 44, pp. 235-246.
- Burmester, L., 1888. *Textbook of Kinematics, First Volume, Planar Motion*. s.l.:s.n.
- Burns, S., 1989. Graphical representation of design optimization processes. *Computer-Aided Design*, 21 (1), p. 21–24.

- Cabrera, J. A., Simon, A. & Prado, M., 2002. Optimal synthesis of mechanisms with genetic algorithms. *Mechanism and Machine Theory*, 37 (10), p. 1165 – 1177.
- Cabrera, J., Nadal, F., Muñoz, J. & Simon, A., 2006. Multiobjective constrained optimal synthesis of planar mechanisms using a new evolutionary algorithm. *Mechanism and Machine Theory*.
- Cerny, V., 1985. A Thermodynamical Approach to the Travelling Salesman Problem: an Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications* 45, pp. 41-51.
- Chuang, S. & Chiang, C., 1987. Fifth order synthesis of plane four-bar function generators optimized by varying scale factors. *Mechanism and Machine Theory*, 22 (1), p. 55 – 63.
- Darwin, C., 1859. *Origin of Species*. Londres: John Murray.
- De Juárez, A., 2011. *Metodología de Síntesis Óptima Dimensional de Mecanismos Incluyendo Objetivos de Posición y Velocidad*. Santander: Tesis Doctoral, Universidad de Cantabria, Departamento de Ingeniería Estructural y Mecánica.
- Dorigo, M., 1992. *Optimization, Learning and Natural Algorithms*. Milán: Tesis Doctoral, Politécnico di Milano.
- Dranga, M. M. & Manolescu, N. I., 1977. Matrix equation for the motion of planar mechanisms with one degree of mobility using transmission functions. *Mechanism and Machine Theory*, 12 (2), p. 165 – 172.
- Erkaya, S. & Uzmay, I., 2008. A neural-genetic (NN-GA) approach for optimizing mechanisms having joints with clearance. *Multibody Systems Dynamics*.
- Erkaya, S. & Uzmay, I., 2008. Determining link parameters using genetic algorithm in mechanisms with joint clearance. *Mechanism and Machine Theory*.
- Fang, W., 1994. Simultaneous type and dimensional synthesis of mechanisms by genetic algorithms-DE. *Mechanics Synthesis Analysis* 70.
- Farhang, K., Midha, A. & Bajaj, A. K., 1988. Synthesis of harmonic motion generating linkages - part I: function generation. *Journal of mechanisms, transmissions, and automation in design*, 110 (1), p. 16–21.

- Farhang, K., Midha, A. & Hall Jr., A. S., 1988. Synthesis of harmonic motion generating linkages - part II: path and motion generation. *Journal of mechanisms, transmissions, and automation in design*, 110 (1), p. 22–27.
- Fernández de Bustos, I., 2003. *Síntesis cinemática dimensional de mecanismos mediante algoritmos genéticos*. Bilbao: Tesis Doctoral, Universidad del País Vasco.
- Frandsen, P. E., Jonasson, K. & Nielsen, H. B. y. T. O., 1999. *Unconstrained Optimization*. s.l.:s.n.
- Freudenstein, F., 1955. Approximate Synthesis of Four-Bar Linkages. *Transactions of ASME*, 77 (Aug.), p. 853–861.
- Gatti, G. & Mundo, D., 2006. Optimal Synthesis of six-bar cammed-linkages for exact rigid-body guidance. *Mechanism and Machine Theory*.
- Glover, F., 1986. Future paths for integer programming and links to artificial intelligences. *Computer and Operation Research* 13, pp. 533-549.
- Goldberg, D., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Massachusetts: Addison Wesley.
- Habibi, H., Shirazi, K. & Shishesaz, M., 2008. Roll steer minimization of McPherson-strut suspension system using genetic algorithm method. *Mechanism and Machine Theory* 43, pp. 57-67.
- Holland, J., 1973. Genetic Algorithms and the Optimal Allocations of trials. *SIAM Journal of Computing* 2(2).
- Holland, J., 1975. *Adaptation in Natural and Artificial Systems*. Michigan: The University of Michigan Press.
- Kennedy, J., 1997. The Particle Swarm: Social Adaptation of Knowledge. *Proceedings of IEEE International Conferenceo on Evolutionary Computation*, pp. 303-308.
- Khorshidi, M. y otros, 2011. Optimal design of four bar mechanisms using a hybrid multi-objective GA with adaptive local search. *Mechanism and Machine Theory* 46, pp. 1453-1465.
- Kimbrell, J. T., 1984. Graphical synthesis of a 4-bar mechanism. *Mechanism and Machine Theory*, 19 (1), p. 45 – 49.

- Kirkpatrick, S., Gelatt, C. & Vecchi, M., 1983. Optimization by Simulated Annealing. *Science* 220, pp. 671-680.
- Kramer, S. & Sandor, G., 1970. Finite Kinematic synthesis of a cycloidal crank mechanism for function generation. *Journal of Engineering and Industry*, p. 531–536.
- Kunjur, A. & Krishnamurty, S., 1997. Genetic algorithms in mechanical synthesis. *Journal of Applied Mechanisms and Robotics* 4 (2).
- Laribi, M., Mlika, A., Romdhane, L. & Zeghloul, S., 2004. A combined genetic algorithm-fuzzy logic method (GA-FL) in mechanisms synthesis. *Mechanisms and Machine Theory*.
- Lin, W.-Y., 2010. A GA-DE hybrid evolutionary algorithm for path synthesis of four-bar linkage. *Mechanism and Machine Theory* 45, pp. 1096-1107.
- Litvin, F. L., 1980. Application of theorem of implicit function system existence for analysis and synthesis of linkages. *Mechanism and Machine Theory*, 15 (2), p. 115 – 125.
- Liu, Y. & McPhee, J., 2005. Automated Type Synthesis of Planar Mechanisms Using Numeric Optimization With Genetic Algorithms. *Journal of Mechanical Design*.
- Mc Garva, J. & al., 1994. Rapid search and selection of path generation mechanisms from a library. *Mechanism and Machine Theory* 29 (2), pp. 223-235.
- McDougall, R., 2008. *Optimization-Based Mechanism Synthesis using Multi-Objective Parallel Asynchronous Particle Swarm Optimization*. Ontario: Tesis Doctoral, The University of Ontario Institute of Technology.
- Mitchiner, R. G. & Mabie, H. H., 1977. The synthesis of 4-bar linkage coupler curves using derivatives of the radius of curvature—I. Straight path procedure. *Mechanism and Machine Theory*, 12 (2), p. 133 – 146.
- Mitchiner, R. G. & Mabie, H. H., 1977. The synthesis of 4-bar linkage coupler curves using derivatives of the radius of curvature—II. Circular path procedure. *Mechanism and Machine Theory*, 12 (2), pp. 147-163.
- Nelder, J. A. & Mead, R., 1965. A simplex method for function minimization. *Computer Journal*, 7 (4), p. 308–313.
- Nieto, J., 1978. *Síntesis de mecanismos*. Madrid: Editorial AC.

- Noriega, A., 2008. *Síntesis dimensional óptima de mecanismos mediante estrategias evolutivas*. Gijón: Universidad de Oviedo, PhD.
- Oblonsky, J. G., 1980. Eloge: Antonin Svoboda, 1907-1980. *Annals of the History of Computing*, 2 (4), p. 284–298.
- Oren, C. & Kaufman, R. E., 1984. Three position precessing mechanism synthesis. *Mechanism and Machine Theory*, 19 (1), p. 139 – 148.
- Peñuñuri, F., Peón-Escalante, R., Villanueva, C. & Pech-Oy, D., 2011. Synthesis of mechanisms for single and hybrid tasks using differential evolution. *Mechanism and Machine Theory*.
- Rao, S. & Pan, T., 1991. Optimal Placement of Actuators in Actively Controlled Structures using Genetic Algorithms. *AIAA Journal* n°6, pp. 942-943.
- Reklaitis, G. V., Ravindran, A. & Ragsdell, K. M., 1983. *Engineering optimization: methods and applications*. New York: John Wiley & Sons.
- Rudolph, G., 1994. Convergence Analysis of Canonical Genetic Algorithms. *IEEE Transactions on Neural Networks*, Vol. 5, n°1.
- Sánchez Marín, F. T., 2000. *Contribución a la síntesis dimensional de mecanismos planos para generación de trayectoria*. Castellón: Universitat Jaume I, PhD.
- Sancibrián, R., García, P., Viadero, F. & Fernández, A., 2006. A general procedure based on exact gradient determination in dimensional synthesis of planar mechanisms. *Mechanism and Machine Theory*, 41 (2), p. 212–229.
- Sancibrián, R., Viadero, F., García, P. & Fernández, A., 2004. Gradient based optimization of path synthesis problems in planar mechanisms. *Mechanism and Machine Theory*, 39 (8), p. 839–856.
- Sandor, N. E. A. G. G., 1984. *Advanced mechanism design: Analysis and Synthesis*. Vol. 2. New Jersey: Prentice-Hall.
- Sedano, A. y otros, 2012. Hybrid Optimization Approach for the Design of Mechanisms Using a New Error Estimator. *Mathematical Problems in Engineering*.
- Starosta, R., 2008. Application of Genetic Algorithm and Fourier Coefficients (GA-FC) in Mechanism Synthesis. *Journal of Theoretical and Applied Mechanics*.

- Sun, J. W. H. & Waldron, K. J., 1981. Graphical transmission angle control in planar linkage synthesis. *Mechanism and Machine Theory*, 16 (4), p. 385 – 397.
- Sutherland, G. & Karwa, N., 1978. Ten-design-parameter 4-bar synthesis with tolerance considerations. *Mechanism and Machine Theory*, 13 (3), p. 311 – 327.
- Svoboda, A., 1965. *Computing mechanisms and linkages*. New York: Dover Publ..
- Szydlowski, W. & Nelson, C., 2007. Synthesis of Adaptive Pliers Mechanism using Genetic Algorithms. *Proceedings of the ASME 2007, International Design Engineering Technical Conferences & Computers and Information Engineering Conference*.
- Unruh, V. & Krishnaswami, P., 1995. A computer-aided design technique for semi-automated infinite point coupler curve synthesis of four-bar linkages. *Journal of Mechanical Design Vol.117*, p. 143.
- Vallejo, J., 1993. *Optimización de mecanismos planos con cualquier configuración y tipo de síntesis*. Bilbao: Universidad del País Vasco, PhD.
- Watanabe, K., 1992. Application of natural equations to the synthesis of curve generating mechanisms. *Mechanism and Machine Theory Vol. 27*, pp. 261-273.