Síntesis posicional de mecanismos doble manivela usando algoritmos evolutivos

César Guerra Torres, Luis Torres Treviño, Juan Ángel Rodríguez Liñán
Laboratorio de Mecatrónica, CIIDIT , FIME-UANL
cguerratorres@gmail.com, luis.torres.ciidit@gmail.com, angel.rodriguezln@uanl.edu.mx

RESUMEN

La síntesis posicional de mecanismos de tipo doble manivela (RRRR) usando métodos algebraicos, generalmente es representada por un sistema de ecuaciones, cuyas n incógnitas determinan las n condiciones de diseño. Cuando se desea satisfacer una mayor cantidad de condiciones, no hay garantía de obtener una solución algebraica que cumpla con todas las ecuaciones de posición. En este trabajo se presenta un técnica basada en algoritmos evolutivos, para obtener una solución aproximada del sistema de ecuaciones de posición, que a su vez contemple una cantidad mayor de condiciones de diseño.

PALABRAS CLAVE

Síntesis, mecanismos, algoritmos evolutivos, Freudenstein.

ABSTRACT

The positional synthesis of mechanisms of type double crank (RRRR) using algebraic methods, is represented generally by a system of equations, which n incognitos determines the n conditions of design. When it is desired to satisfy a greater amount with conditions, there is guarantee no to obtain an algebraic solution that fulfills all the equations of position. In this work, a technique based on evolutionary algorithms appears, to obtain an approximated solution of the position equations system, that as well contemplate a greater amount of conditions of design.

KEYWORD

Synthesis, mechanisms, evolutionary algorithms, Freudenstein.

INTRODUCCIÓN

La síntesis cinemática es de gran importancia en muchos diseños o rediseños de maquinaria, ya que requieren de mecanismos que satisfagan con ciertas condiciones de movimiento.[1] La idea de la síntesis cinemática de mecanismos consiste en determinar el tipo de mecanismo, así como las longitudes de sus eslabones para satisfacer ciertas condiciones de diseño. Diversas técnicas han sido exploradas destacando aquellas que se basan en métodos gráficos y algebraicos.

Los métodos gráficos son altamente laboriosos y de poca precisión. Por otro lado, en los métodos algebraicos se requiere resolver un sistema de *N* ecuaciones con *n* incógnitas, lo que limita obtener soluciones para un máximo de *n* condiciones de diseño, en los mejores de los casos se puede tener solución única si $n=N$ o soluciones múltiples si $N<n$.[2] Por ejemplo, la ecuación de Freudenstein la cual determina la movilidad de un mecanismo plano RRRR, presenta tres constantes que relacionan las longitudes de los eslabones, para el caso de síntesis posicional estas constantes son incógnitas, por lo tanto sólo es posible encontrar soluciones algebraicas al establecer máximo tres condiciones de diseño (pares de posición, entrada, salida).[3]

Considerando algunas limitaciones de los métodos anteriores, existen nuevas metodologías basadas en algoritmos computacionales para la síntesis cinemática de mecanismos planos,[4] la generación de trayectorias,[5] así como la síntesis topológica del mecanismo.[6]

Una de estas técnicas es la computación evolutiva, que es una rama de la inteligencia artificial inspirada en la teoría de la Selección Natural propuesta por Charles Darwin. Los algoritmos evolutivos se pueden utilizar para resolver problemas de optimización si se tienen modelos específicos y criterios de evaluación adecuados,[7] los cuales pueden utilizarse para resolver problemas donde la solución algebraica no existe o es muy complicada. Por lo tanto, considerando *n* condiciones de diseño deseadas, mediante estos algoritmos es posible encontrar una solución aproximada de un sistema de *N* ecuaciones con *n* incógnitas, donde $N>n$.

Particularmente, en este trabajo se presenta un algoritmo evolutivo que permite calcular las constantes de la ecuación de Freudenstein, a fin de determinar un único mecanismo RRRR que satisfaga un mayor número de condiciones de diseño. Además, dado un mecanismo con condiciones de diseño (pares entrada, salida), es posible determinar mecanismos afines que se aproximen a dichas condiciones de diseño.

El resto del trabajo se organiza de la siguiente manera: Primero se presenta la ecuación de Freudenstein, y luego se describe el algoritmo evolutivo. Posteriormente se ilustran tanto la solución algebraica como la solución propuesta basada en algoritmos evolutivos, y con el fin de demostrar la eficiencia de la metodología se presentan los resultados de simulaciones para dos condiciones de diseño. Por último, se exponen la discusión y conclusión.

ANÁLISIS POSICIONAL DEL MECANISMO DOBLE MANIVELA

Un mecanismo es una cadena de eslabones de uno o varios grados de libertad. El mecanismo doble manivela o RRRR (figura 1) es de un grado de libertad (GDL) y tiene la función de convertir un movimiento rotacional a otro rotacional pero con diferentes características en su movilidad.

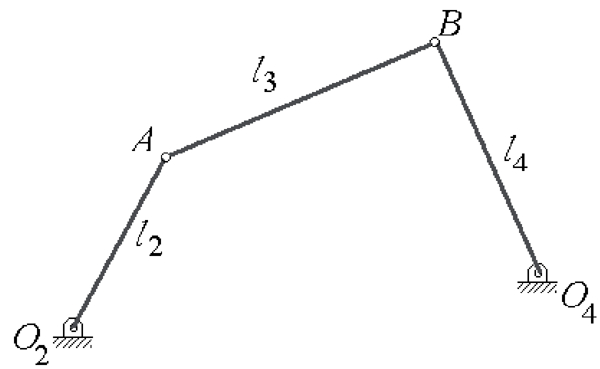

Fig. 1. Mecanismo plano RRRR.

De la figura 1, los eslabones de longitud l_2 y l_4 se conocen como manivela de entrada y salida, respectivamente; mientras que el eslabón de longitud l_3 se conoce como barra de acoplamiento.

Este tipo de mecanismo se utiliza en diversos mecanismos de maquinaria, tales como máquinas transportadoras, mecanismos de limpiadores de parabrisas, máquinas de ejercicio, entre otras. En la figura 2, se muestra cómo un mecanismo doble manivela es utilizado para generar una trayectoria específica en una excavadora móvil.

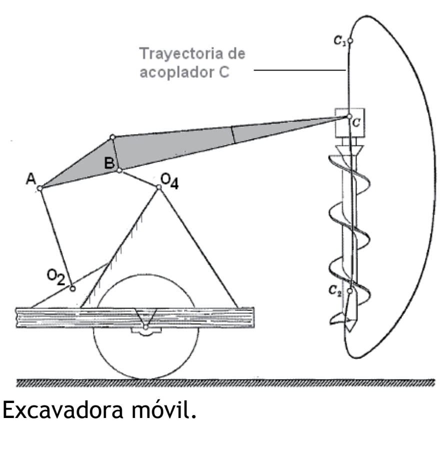

Fig. 2. Excavadora móvil.

Ecuación de Freudenstein

Para determinar la ecuación algebraica que describe la movilidad del mecanismo de la figura 1, considere un análisis vectorial que se muestra en la figura 3. En la cual, $\mathbf{r_k} \in \Re^2$ se representa en su forma Euler como $\mathbf{r_k} = l_k e^{i\theta k}$, donde $l_k, \theta_k \in \Re$ son la longitud del eslabón y su posición angular, respectivamente.

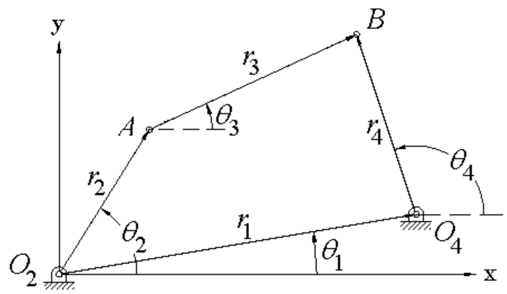

Fig. 3. Mecanismo RRRR plano.

De la figura 3 se puede observar que la ecuación vectorial queda expresada por

$r_3 = r_1 + r_4 - r_2$

Aplicando el producto escalar por si mismo $r_3 \cdot r_3 = (r_1 + r_4 - r_2) \cdot (r_1 + r_4 - r_2)$, se obtiene ahora una ecuación escalar de la forma,

$$l_3^2 - l_1^2 - l_2^2 - l_4^2 = 2l_1l_4cos(\theta_4 - \theta_1) - 2l_1l_2cos(\theta_2 - \theta_1) + 2l_1l_4cos(\theta_4 - \theta_2)$$

La cual puede reescribirse como,

$$k_1 - k_2cos(\theta_4 - \theta_1) + k_3cos(\theta_2 - \theta_1) + cos(\theta_4 - \theta_2) = 0 \quad (1)$$

Donde las constantes k_i quedan expresadas como,

$$k_1 = \frac{l_3^2 - l_1^2 - l_2^2 - l_4^2}{2l_2l_4}, \quad k_2 = l_1/l_2, \quad k_3 = l_1/l_4 \quad (2)$$

La ecuación (1) establece la relación cinemática de la posición del eslabón de salida y la posición del eslabón de entrada θ_2. El problema de la cinemática directa consiste en lo siguiente: Conociendo las longitudes de los eslabones del mecanismo de la figura 1, determinar los valores de las constantes k_i de la ecuación (2), tal que para cada ángulo de entrada θ_2, determinar su correspondiente ángulo θ_4 de salida. Para lo anterior es necesario utilizar algún método numérico que resuelva la ecuación (1).

Por otro lado, para la cinemática directa es necesario comprobar que el mecanismo RRRR cumpla con la condición de mobilidad llamada Ley de Grashof1 que establece:

$$l_{maj} + l_{min} \leq \sum l_{res} \quad (3)$$

donde l_{maj} y l_{min} se refieren a la mayor y menor longitud del mecanismo respectivamente, l_{res} se refiere a las dos longitudes restantes. Si se cumple con la condición de Grashof (3), entonces el eslabón de menor longitud podrá tener revoluciones completas.

ALGORITMOS EVOLUTIVOS

Existen muchos paradigmas para aplicar computación evolutiva, desde algoritmos genéticos y estrategias evolutivas hasta los algoritmos basados en estimación de distribuciones. En este paradigma se representan las soluciones potenciales en una estructura base conocida como cromosoma. Esta estructura base o cromosoma pueden ser vectores binarios, vectores reales, símbolos e inclusive estructuras más complejas como matrices, listas y árboles. Para respetar la analogía con la teoría de la evolución se trabaja con poblaciones de individuos, es decir, un conjunto de soluciones potenciales todas con una misma representación común (mismo cromosoma). Es importante considerar que en todos los algoritmos evolutivos siempre se aplican los mismos procesos: Evaluación de todos los individuos de una población, selección de los individuos con los valores de evaluación más alta y generación de nuevos individuos considerando solamente los individuos seleccionados. Estos tres procesos se aplican sucesivamente hasta que un criterio de finalización sea satisfecho.

Evonorm es un algoritmo evolutivo basado en la estimación de una distribución normal univariada marginal en donde se utilizan los individuos seleccionados para calcular los parámetros para generar una variable aleatoria con distribución normal, específicamente la media y la desviación estándar.[8,9]

El algoritmo de Evonorm establece:

1. Generación de una población aleatoria.
2. Evaluación de la población.
3. Selección de los mejores individuos.
4. Generación de una nueva población.
5. Ir al paso (2) o terminar si un criterio específico se satisface.

La diferencia que tienen los algoritmos basados en estimación de distribuciones es sustituir los mecanismos de cruce y mutación, por un mecanismo basado en la generación de nuevos individuos utilizando una variable aleatoria con distribución normal, cuyos parámetros son estimados y utiliza una variable aleatoria para cada variable de decisión. Es importante indicar que Evonorm utiliza una heurística adicional para preservar el mejor encontrado e involucrarlo en el proceso de búsqueda. Es muy similar al mecanismo de elitismo que es ampliamente utilizado en la computación evolutiva porque explota con mejor eficiencia el espacio de búsqueda, aunque a veces puede comprometer el proceso de exploración provocando convergencia prematura en algunos casos. En esta heurística, en el momento de generar nuevas soluciones se utiliza como parámetro de la media al mejor encontrado en el 50% de las veces y en el otro 50% se utiliza el valor medio calculado. El siguiente algoritmo ilustra lo aquí expuesto:

Algoritmo para generar una nueva población:

1. Calcular media μ y desviación estándar σ por variable de decisión *pr* a partir de la matriz V que representa a la población de individuos seleccionados;

$$\mu_{pr} = \sum_{k=1}^{I_s} (V_{pr,k}) / I_s$$

$$\sigma_{pr} = \sqrt{(\sum_{k=1}^{I_s} (V_{pr,k} - \mu_{pr}))} / I_s$$

2. Generar una nueva población P de tamaño I_p. P es una matriz e I_s es el mejor individuo que representa a la mejor solución encontrada en el momento. En el 50% de las veces se elige la media calculada previamente y en el otro 50% de las veces se elige como media el valor propuesto por el mejor individuo encontrado. Para k = 1 hasta I_p, si $U(*) > 0.5$ entonces se evalúa $P(k, N(\mu_{pr}, \sigma_{pr}))$; en caso contrario $P(k, N(Is_{pr}, \sigma_{pr}))$.

SÍNTESIS POSICIONAL

La síntesis cinemática es el problema opuesto a la cinemática directa y consiste en determinar las dimensiones de los eslabones de un mecanismo, para satisfacer ciertas condiciones de posición en el plano. Este tipo de síntesis se refiere a la posición angular o lineal de un eslabón o en sí, a la posición misma de todo el eslabón en el plano.

El tipo de síntesis abordado en este trabajo es el siguiente: Dado un mecanismo RRRR y las posiciones para diversos pares de diseño $[\theta_{2,i}, \theta_{4,i}]$, determinar las longitudes de los eslabones, así como el ángulo del eslabón fijo θ_1, para que el mecanismo cumpla con las posiciones deseadas.

Solución algebraica

Usando la ecuación de Freudenstein (1), es posible obtener un mecanismo que cumpla las condiciones de diseño. Fijando θ_1, algebraicamente sólo es posible establecer un máximo de $N = 3$ condiciones de pares de diseño:

$$[\theta_{21},\theta_{41}],[\theta_{22},\theta_{42}],[\theta_{23},\theta_{43}] \tag{4}$$

A fin de calcular las $n = 3$ incógnitas k_1, k_2 y k_3. Por lo tanto se generan ecuaciones de posición,

$$\begin{aligned} k_1 - a_1 k_2 + b_1 k_3 + c_1 &= 0 \\ k_1 - a_2 k_2 + b_2 k_3 + c_2 &= 0 \\ k_1 - a_3 k_2 + b_3 k_3 + c_3 &= 0 \end{aligned} \tag{5}$$

donde $a_i = cos(\theta_{4i} - \theta_1)$, $b_i = cos(\theta_{2i} - \theta_1)$ y $c_i = cos(\theta_{4i} - \theta_{2i})$. El sistema de ecuaciones (5) puede representarse en forma matricial por:

$$BK = C \tag{6}$$

donde

$$B = \begin{pmatrix} 1 & -a_1 & b_1 \\ 1 & -a_2 & b_2 \\ 1 & -a_3 & b_3 \end{pmatrix}, K = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix}, C = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \tag{7}$$

Algebraicamente sólo es posible encontrar una solución única si $N = n$ y existe B^{-1}. Para las demás condiciones, no existencia de B^{-1} ó $N \neq n$, podrían existir soluciones múltiples o bien, no existir una solución exacta.

Solución propuesta

A fin de resolver las limitaciones de los métodos algebraicos respecto a la ecuación de Freudenstein, se propone utilizar algoritmos de computación evolutiva para satisfacer las siguientes condiciones de diseño:

1. Dado un número de pares de diseño $N > 3$, obtener un mecanismo que se aproxime a todos esos pares de diseño.

2. Obtener mecanismos afines que cumplan con la relación θ_4/θ_2, para N pares de interés dentro de una región de operación.

Considerando las condiciones de diseño expresadas en un número de pares usualmente mayor que tres, se define una función de evaluación para calificar a cada individuo generado por el algoritmo evolutivo, luego seleccionar los que tengan el valor más alto y que en este caso consiste en minimizar la función dada por

$$Ft_i=\sum_{p=1}^{N}\left|k_{1,i}-k_{2,i}\cos(\theta_{4,p}-\theta_{1,i})+k_{3,i}\cos(\theta_{2,p}-\theta_{1,i})+\cos(\theta_{4,p}-\theta_{1,i})\right| \qquad (8)$$

donde k_1, k_2 y k_3 son las variables de decisión que están representadas en cada individuo i de la población. La variable p determina un único par de diseño de un total de pares de diseño N. La variable θ_1 puede ser fija o ser considerada como una variable de decisión más. En nuestro caso la consideramos como una variable de decisión más.

RESULTADOS EN SIMULACIÓN

El algoritmo evolutivo utiliza una población de 100 individuos de los cuales 10 son seleccionados y con ellos se genera la siguiente población, por lo que es un algoritmo evolutivo generacional en donde los padres más fuertes dejan descendencia pero no sobreviven a excepción de uno (elitismo) que representa a la mejor solución encontrada.

Para acotar el desempeño del algoritmo y que éste genere soluciones en un espacio factible, se establece arbitrariamente $k_1\in[-3,3]$, $k_2\in[0.3,5]$, $\boldsymbol{k}_3\in[0.3,5\ ]$ y $\theta_1\in[-\pi/2,\ \pi/2]$. Cada algoritmo se ejecuta durante 25 generaciones y por diez corridas, guardando la mejor solución encontrada en cada caso.

DISEÑO 1. Se desea sintetizar un mecanismo doble manivela cuya relación θ_4/θ_2 se eleve gradualmente hasta θ_2 = 170°, posteriormente permanezca constante hasta θ_2 = 300° y por último, decaiga rápidamente.

Los pares de diseño deseados se describen en la tabla I.

Tabla I. Pares del diseño 1.

θ_2	θ_4
10	62.44
40	69.14
90	96.73
150	128.44
170	131
180	131
210	131
260	131
300	131

Después de efectuar 10 corridas, el algoritmo evolutivo arrojó el mejor mecanismo con l_1 = 8, l_2 = 4.77, l_3 = 4.8, l_4 = 8.07 y θ_1 = -48.14.

Para comprobar la proximidad de los resultados, se elaboró un programa en Visual Basic ® para simular un mecanismo doble manivela. La figura 4 muestra la relación θ_4 vs θ_2, cumpliendo con las condiciones de diseño.

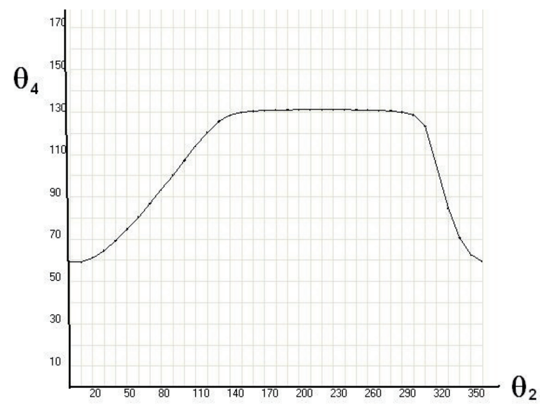

Fig. 4: Relación salida/entrada del diseño 1.

La figura 5 muestra la pantalla del programa de simulación.

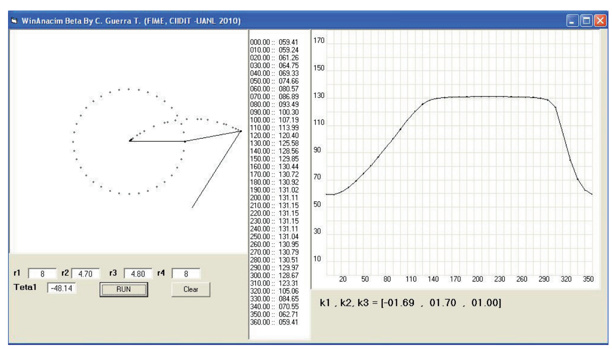

Fig. 5: Programa de simulación.

DISEÑO 2. Obtener una familia de mecanismos afines para satisfacer las condiciones de pares de diseño de la tabla II.

Para este diseño, se ejecutaron 10 corridas y así obtener el mejor mecanismo. Este proceso se repitió

Tabla II. Pares del diseño 2.

θ_2	θ_4
10	62.44
40	67.61
90	96.73
150	128.44
170	133.79

cuatro veces para obtener cuatro mecanismos afines al movimiento deseado, obteniendo los resultados mostrados en la tabla III.

Tabla III. Soluciones del Diseño 2.

Mec.	k_1	k_2	k_3	θ_2 (DEG)
1	-1.63	2.67	1.60	-22.00
2	-1.45	1.32	0.81	-43.06
3	-1.60	2.43	1.43	-22.00
4	-1.53	1.74	1.03	-32.11

Fijando $l_1 = 8$, las longitudes de los otros eslabones se pueden obtener a partir de las ecuaciones (2). La figura 6 muestra los resultados en simulación.

Se puede observar en la figura 6, como los cuatro mecanismos obtenidos cumplen con las condiciones de diseño dentro de la región de operación. Mientras los mecanismos 1, 3 y 4 cumplen con la condición de Grashof (3), el mecanismo 2 no cumple con dicha condición por lo que se agarrota en alguna posición.

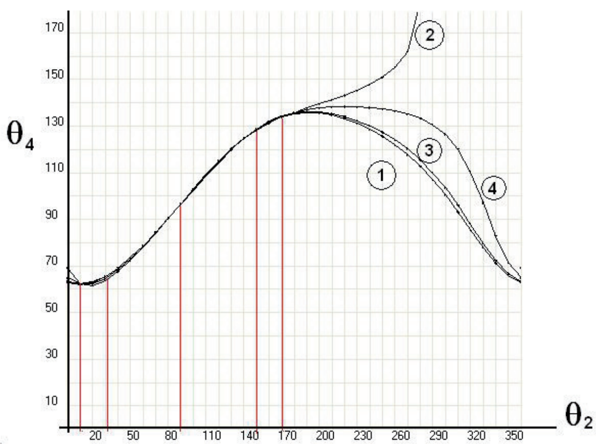

Fig. 6: Resultados en simulación del diseño 2.

CONCLUSIONES Y DISCUSIONES

Los mecanismos del tipo doble manivela son altamente utilizados en maquinaria, para que estos cumplan con ciertas condiciones de movilidad es necesario utilizar metodologías algebraicas, gráficas computacionales para sintetizar el movimiento del mismo.

En este trabajo se presentó una metodología basada en algoritmos evolutivos para sintetizar las posiciones de un mecanismo doble manivela, de modo que cumpla con ciertas condiciones de pares de diseño [entrada, salida].

A diferencia del método algebraico donde sólo se pueden tener tres condiciones de diseño, con la metodología propuesta es posible sintetizar un mecanismo doble manivela para satisfacer un mayor número de condiciones de diseño, obteniendo un único mecanismo que se aproxime a todas las condiciones deseadas.

Como demostración de la metodología se presentaron dos diseños. El primer diseño, exigía una movilidad con cierto grado de dificultad y difícil de sintetizar por métodos algebraicos. El segundo diseño requería una familia de mecanismos afines para cumplir con ciertos pares de diseño. En ambos casos la metodología propuesta permitió obtener las dimensiones de los mecanismos deseados.

REFERENCIAS

1. Robert L. Norton. Design of Machinery. Mc. Graw Hill, 1999.
2. G. Williams. Álgebra lineal con aplicaciones. Mc Graw Hill, 2001.
3. F. Feudenstein and G. Sandor. Synthesis of path-generating mechanism by means of programmed digital computer. ASME J. Eng. Ibd, Serie B, 81(2), 1959.
4. Ahmad Smaili and Nadim Diaba. Optimum synthesis of hybrid-task mechanisms using ant-gradient search method. Mechanism and Machine Theory, 42(1):115-130, 2007.
5. Yi Liu and John McPhee. Automated Type Synthesis of Planar Mechanisms Using Numeric Optimization With Genetic Algorithms. J. Mech. Des., 127(5):910-917, 2005.
6. D. Mundo and J. Y. Liu and H. S. Yan. Optimal Synthesis of Cam-Linkage Mechanisms for Precise Path Generation. J. Mech. Des., 128(6):1253 - 1261, 2006.
7. Daniel Ashlock. Evolutionary Computation for Modeling and Optimization. Springer, 2009.
8. Luis Torres-T. EvoNorm, A New Evolutionary Algorithm to Continuous Optimization. Workshop on Optimization by Building and Using Probabilistic Models, 2006.
9. Luis Torres-T. Evonorm: Easy and effective implementation of estimation of distribution algorithms. Journal of Research in Computing Science, 23:75-83, 2006.