

# Un enfoque genético cooperativo para el cálculo de trayectorias de los dedos de una mano robótica

A cooperative genetic approach for calculating trajectories of robotic hand fingers

Rafael Rivera-López <sup>1\*</sup>, Fernando Ramos-Quintana <sup>2</sup>, Alberto Muñoz <sup>3</sup>

<sup>1</sup> Departamento de Sistemas y Computación, Instituto Tecnológico de Veracruz.  
Calzada Miguel Ángel de Quevedo 2779. Veracruz, México

<sup>2</sup> Instituto Tecnológico y de Estudios Superiores de Monterrey, campus Cuernavaca. Autopista del Sol km 104, col. Real del Puente.  
Xochitepec, Morelos, México. CP 62790

<sup>3</sup> The Robotics Institute of Yucatán. Calle 60 452B, col. Centro. Mérida, Yucatán, México  
\* Correo-e: rrivera@itver.edu.mx

## PALABRAS CLAVE:

metaheurísticas, algoritmo genético,  
sujeción de objetos, robótica

## RESUMEN

En este artículo se presenta la aplicación de un algoritmo genético cooperativo para el cálculo de trayectorias que los dedos de una mano robótica configurable deben seguir para sujetar un objeto en dos dimensiones. A partir de un objetivo común, se establece el conjunto de movimientos coordinados para alcanzar la meta. Se describen las características de la sujeción de objetos, que involucra tanto la determinación de los puntos de contacto como la definición de las trayectorias de los dedos para alcanzar dichos puntos. Además de los algoritmos para la sujeción y para las trayectorias, se muestran los resultados experimentales.

## KEYWORDS:

metaheuristic, genetic algorithm,  
grasped object, robotic

## ABSTRACT

This paper describes the implementation of a cooperative genetic algorithm for computing fingers trajectories of a configurable robotic hand in order to grasping 2D objects. From a common goal, a set of coordinated movements are established, in order to achieve the goal. The characteristics of the grasped object, which involves both the determination of the contact points, and the definition of the trajectories of the fingers to achieve those points are described. Algorithms for grasping and construction of trajectories are presented and experimental results are shown.

## 1 INTRODUCCIÓN

La manipulación diestra es un tópico de interés dentro del ensamblado industrial, el diseño de prótesis y el estudio del movimiento humano. Dicha área comprende muchos tópicos como el diseño de herramientas, la sujeción, el control, el sensado táctil y la planificación de tareas [1]. Los tres problemas fundamentales de la manipulación diestra son: la determinación de cuán fuerte sostener un objeto con los dedos para asegurar una prensión estable; la definición de los movimientos de los dedos para producir un desplazamiento sobre el objeto y el establecimiento de los rangos de movimiento del objeto sujetado por la mano [2]. Adicionalmente, se agregan a los problemas de sujeción y manipulación los de planeación de alto nivel y el diseño de las manos robóticas y herramientas para manipulación. El estudio de estos problemas ha producido diferentes subcampos de investigación como la estática de prensión, la cinemática, la planeación y el control de la manipulación [3].

En el trabajo de Mishha y Silver [1] se indica que una mano robótica tiene

[...] múltiples dedos articulados (dedos con dos o más articulaciones), cada uno de los cuales es una cadena cinemática abierta, y posiblemente una superficie palmar. La mano contiene un gran número de grados de libertad, y pueden lograr resolución fina con alta calidad y alta velocidad. Cada uno de los dedos podría contener sensores de fuerza y posición y contener adicionalmente sensores táctiles [...]

Michelman indica que es posible utilizar una mano robótica en el diseño de prótesis para sustituir miembros del ser humano, en ambientes peligrosos, donde el ser humano se enfrenta a un serio daño y, en manufactura flexible, en líneas de producción adaptables.

Una tarea de manipulación y ensamblado puede ser dividida en cuatro pasos fundamentales [5]:

1. *La preconfiguración.* Se requiere de un camino libre de colisiones para el sistema en el cual el objeto es alcanzable sólo con movimientos de los dedos.
2. *La sujeción.* Consiste en tomar el objeto y evitar el movimiento relativo de dicho objeto respecto a la mano que lo sujeta [6].
3. *La traslación.* Es el desplazamiento del objeto sujetado por la mano de forma estable hacia el lugar donde se llevará a cabo la operación de

ensamblado. La posibilidad de colisiones con obstáculos externos también debe ser tomada en cuenta.

4. *La operación de ensamblado.* La tarea de ensamblado requiere estrategias de control de fuerza y posición reactivas y robustas.

Cada dedo de una mano robótica debe seguir una trayectoria que le permita alcanzar un punto de contacto con la superficie del objeto. La planificación de movimientos robóticos se refiere al problema de encontrar un camino libre de colisiones por un robot con ciertos grados de libertad en un ambiente con obstáculos [7].

Los enfoques utilizados en la producción del desplazamiento de un dedo sobre una trayectoria determinada [8, 4] incluyen el análisis de cinemática inversa para encontrar la configuración de cada articulación en los dedos que permita seguir dicha trayectoria hasta ponerse en contacto con el objeto. La complejidad inherente al cálculo de una matriz jacobiana inversa sugiere el uso de técnicas alternativas para el cálculo de trayectorias.

De acuerdo con Michalewicz y Michalewicz [9], los algoritmos genéticos son métodos probabilísticos de búsqueda que ayudan a optimizar una función determinada y se basan en los principios de evolución y selección natural. Los algoritmos genéticos tienen los siguientes componentes: una población de cadenas binarias, parámetros de control, una función de aptitud y un conjunto de operadores genéticos. En otro trabajo se indica que un algoritmo genético cooperativo puede verse como una clase especial de algoritmo genético paralelo; cuenta con una serie de subpoblaciones pero, a diferencia de un algoritmo genético distribuido, estas poblaciones son de individuos de distinta especie [10].

En la siguiente sección de este trabajo se hace una descripción del modelo de sujeción de objetos utilizado para después describir el algoritmo genético cooperativo aplicado a la búsqueda de trayectorias. Finalmente se describen los resultados y las conclusiones de este trabajo.

## 2 MODELO DE SUJECIÓN DE OBJETOS

El punto inicial de este trabajo es la representación del contorno del objeto que se va a sujetar. Este contorno se puede representar como un polígono o como una curva cerrada, ya que es construido mediante puntos

de control. A partir de esta representación se aplica una búsqueda de los puntos de contacto que garanticen el equilibrio de la sujeción.

### 2.1 Condiciones de equilibrio

De acuerdo con la definición presentada por varios autores [1, 11-13], una sujeción tiene *form-closure* si los dedos de la mano pueden ejercer fuerzas que contrarresten la acción de una torsión externa (una fuerza externa que puede producir una traslación y rotación sobre el objeto); si cualquier torsión externa  $w_{ext}$  aplicada al objeto puede ser balanceada por fuerzas de los dedos  $f_i$ , se dice que la sujeción alcanza *form-closure*.

Lo anterior significa que la torsión resultante de las fuerzas de los dedos,  $w_{int}$  en (1), debe ser igual a la aplicada por un agente externo,  $w_{ext}$  en (2), como se define en (3).  $F_e$  es la fuerza externa y  $R_e \times F_e$  es el momento producido.

$$w_{int} = \sum_{i=1}^n \begin{pmatrix} f_i \\ r_i \times f_i \end{pmatrix} \dots\dots\dots(1)$$

$$w_{ext} = \begin{pmatrix} F_e \\ R_e \times F_e \end{pmatrix} \dots\dots\dots(2)$$

$$w_{ext} - w_{int} = 0 \dots\dots\dots(3)$$

Las fórmulas anteriores indican que para probar *form-closure* se requiere la presencia de un agente externo tratando de romper la estabilidad del objeto sujetado bajo alguna configuración; los resultados mostrados con el enfoque presentado por Liu [13] se basan en la definición de una torsión externa aplicada sobre un objeto para el cual se conocen previamente los puntos de contacto.

$$w_{int} = \sum_{i=1}^n \begin{pmatrix} f_i \\ r_i \times f_i \end{pmatrix} = 0 \dots\dots\dots(4)$$

De (4) se observa que para alcanzar el equilibrio de un objeto, la suma de fuerzas y la suma de momentos debe ser igual a cero, y que en ella se utilizan tres valores, por lo que se puede decir que dichas sumatorias están en función a:

1. La localización del punto de contacto (con base

en una referencia global, dada por el vector  $r_i$ ).

2. La magnitud de la fuerza aplicada por el dedo en el punto ( $f_i$ ).

3. La orientación de la fuerza aplicada por el dedo respecto a la dirección normal y al contorno del objeto en el punto de contacto, o si el tipo de contacto está definido considerando la fricción.

Por lo anterior, se observan tres parámetros que definen el equilibrio de un objeto. Entonces se puede utilizar un enfoque de búsqueda que, con base en la prueba de valores sobre dichas variables, determine si existe o no el equilibrio; esto es, la condición de equilibrio será evaluada repetidas veces, cambiando estos tres parámetros hasta encontrar los que garanticen una sumatoria igual a cero.

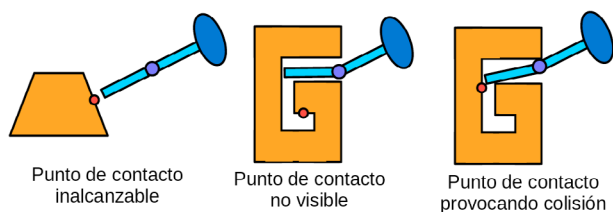
Para determinar la orientación de la fuerza aplicada en un punto de contacto, es preciso conocer la dirección normal a la superficie del objeto en ese punto. En el caso de un contorno representado por líneas, la normal se calcula mediante el concepto de pendiente de líneas perpendiculares.

En el caso de un contorno representado por curvas, el segmento de curva se discretiza en varios puntos calculando la normal a la superficie como una normal a la recta que une dos puntos generados por la discretización. En función del grado de discretización, las rectas se acercan más al segmento de curva discretizado, por lo cual no se pierde generalidad en el algoritmo.

### 2.2 Criterios previos a la búsqueda de puntos de contacto

Los puntos de contacto encontrados por el método deben ser alcanzados por el dedo de la mano robótica para completar el modelo de sujeción. La única limitante para una sujeción de objetos convexos es que el punto de contacto esté dentro del espacio de trabajo del dedo asignado para alcanzarlo. En el caso de objetos no convexos se debe considerar lo siguiente:

1. *Alcanzabilidad*. Que el punto de contacto se encuentre en el espacio de trabajo del dedo que lo tocará.
2. *Visibilidad*. Que el punto de contacto sea visible desde el exterior para poder llegar a él.
3. *Presencia de colisiones*. Debido a que se requiere una orientación específica para las fuerzas aplicadas por los dedos, la orientación del último



**Figura 1.** Puntos de contacto que no pueden ser alcanzados por el dedo de una mano robótica

segmento debe ser igual a la de la fuerza calculada. Esto produce que, aun cuando el punto sea visible y alcanzable, la postura del último segmento puede tener alguna colisión con el objeto o con los otros dedos.

La figura 1 muestra ejemplos de puntos de contacto que no respetan estas consideraciones.

### 2.3 Búsqueda del equilibrio

Se puede hacer asignando aleatoriamente valores a los parámetros del equilibrio y modificándolos en forma iterativa, tratando de reducir la sumatoria de fuerzas y de momentos en cada paso, hasta lograr el equilibrio.

Al aprovechar la representación del objeto usando puntos de control, en el cual para  $N$  puntos de control, se tienen  $N$  segmentos de contorno, se puede aplicar una selección inicial de  $N$  segmentos (igual al número de contactos deseados) y aplicar una búsqueda dentro de ellos, seleccionando puntos y probando la condiciones de equilibrio. Para cada punto en los segmentos de búsqueda, se deben seleccionar los valores de la fuerza y de orientación y probar si existe equilibrio; la orientación de la fuerza debe respetar las restricciones de fricción definidas para el problema.

Se puede decir que la búsqueda debe enfocarse a mejorar el valor de la sumatoria de fuerza y momentos para llegar al equilibrio; para esto se debe utilizar un enfoque de ascenso de colina (*hill-climbing*). Es evidente que esta búsqueda puede no encontrar un valor que satisfaga el equilibrio.

Al recordar la selección inicial de segmentos de contorno, el proceso se puede reiniciar seleccionando otro subconjunto de  $N$  segmentos para aplicar nuevamente la búsqueda, lo que produce un enfoque de ascenso de colinas con reinicio aleatorio. La figura 2 muestra el algoritmo que define los puntos de contacto sobre el contorno del objeto.

## 3 PLANIFICACIÓN DE TRAYECTORIAS

Como indican Mishha y Silver [1], cada dedo de una mano robótica puede considerarse como un manipulador independiente, por lo que el proceso de planificación de trayectorias de estos dedos puede ser manejado como un proceso de planificación para manipuladores cooperativos que trabajan en un ambiente común, donde los dedos utilizan el mismo espacio de trabajo, realizan una tarea dependiente y manipulan un mismo objeto.

Este trabajo considera la planificación de movimientos de los dedos de una mano robótica para alcanzar los puntos de contacto previamente definidos.

De la Cueva y Ramos resumen las desventajas de utilizar enfoques tradicionales:

1. La construcción de restricciones en el método jacobiano extendido.
2. La presencia de configuraciones singulares del robot en el control cinemático al usar la matriz pseudoinversa.
3. La definición del vector y del control cinemático al usar la matriz pseudoinversa.
4. El costo computacional para el cálculo del espacio de configuraciones.
5. El incremento en la complejidad del cálculo del espacio de configuraciones para ambientes dinámicos.

Para evitar estas desventajas se han propuesto métodos llamados no tradicionales, como el algoritmo genético cooperante propuesto en ese trabajo [10], donde se trabaja en el espacio del manipulador y se utiliza cinemática directa en un ambiente de dos o más manipuladores.

### 3.1 Enfoque genético cooperante para el cálculo de trayectorias

Los algoritmos genéticos (AG) han probado ser una herramienta efectiva para resolver el problema de cinemática inversa. La principal ventaja es que se labora con el modelo de cinemática directa, lo que evita el cálculo de la matriz jacobiana inversa, y se trabaja directamente en el espacio de la tarea, con lo que se deja de usar una nueva representación del espacio como un espacio de configuraciones [14, 15].

El algoritmo genético cooperante (AGC) tiene la característica de utilizar varias poblaciones que cooperan

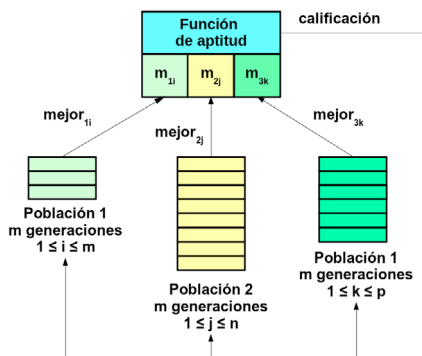
```

Entrada: Conjunto de puntos de control, REINICIOS, REINTENTOS
Salida: Puntos de contacto P y orientación de las fuerza, si lo encuentra
BusquedaPuntosdeContacto() {
  repite hasta REINICIOS {
    C = Selección aleatoria de segmentos de contorno
    G = Selección inicial aleatoria de puntos de contacto y orientación
    de fuerzas dentro de C
    repite hasta REINTENTOS o condición de equilibrio (4) satisfecha {
      Evaluación de condición con G
      Si se satisface condición de equilibrio (4) {
        P = Puntos de contacto y orientación de fuerzas
      }
      de otra forma {
        G = Selección aleatoria de puntos de contacto y orientación
        de fuerzas dentro de C
      }
    } // Fin de reintentos
  } // Fin de reinicios
} // Fin de algoritmo
    
```

**Figura 2.** Algoritmo de búsqueda de puntos de contacto

coordinadamente para llegar al objetivo planteado; la coordinación se realiza mediante un mecanismo llamado de intercambio de información entre las poblaciones. Las poblaciones se forman de individuos con longitud diferente ya que cada individuo representa una posible trayectoria del manipulador.

Un AGC puede verse como una serie de algoritmos genéticos simples que operan independientemente, pero donde la evaluación de los individuos de su población depende del comportamiento de las otras poblaciones.



**Figura 3.** Algoritmo genético cooperante de tres poblaciones. Fuente: De la Cueva y Ramos [10]

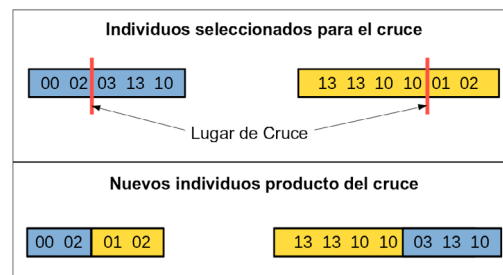
La selección, cruce y mutación de un AG independiente se realiza de la misma forma que en un AG tradicional; la diferencia radica en la evaluación, donde se toma en cuenta la cooperación con el resto de los algoritmos genéticos simples. La figura 3 muestra un esquema de un AGC de tres poblaciones.

Al final de cada generación, cada población coloca su mejor elemento en una plantilla que sirve para determinar el valor de la función de aptitud. Las otras poblaciones utilizan los mejores elementos re-

colectados en la plantilla para evaluar cada uno de los elementos de su población.

*Codificación de las soluciones.* Una trayectoria robótica está formada por un conjunto de configuraciones indexadas en el tiempo, que le permiten mover el órgano terminal de una posición original a otra final. Para determinar la codificación de una trayectoria en una cadena de bits se utiliza el concepto de configuración vecina, que indica que dos configuraciones son vecinas si todos los valores articulares son iguales, excepto para uno de ellos. Por lo anterior, para representar la siguiente configuración (conociendo la inicial) sólo se necesita una tupla de valores (s,a), donde s es el sentido del giro y a la articulación que se movió. Se asocia a s un valor 0 ó 1 que indica el sentido de giro, ya que la magnitud del giro se considera constante.

Por lo anterior, una trayectoria completa forma una cadena de longitud variable, por ejemplo,  $\{(s_1, a_1), (s_2, a_2), \dots, (s_m, a_m)\}$ , considerando el tiempo constante de una configuración a otra.



**Figura 4.** Ejemplo de cruce de dos individuos de longitud variable. Fuente: De la Cueva y Ramos [10]

Esta codificación tiene la ventaja de no ser altamente dependiente del número de articulaciones

```

Entrada: Configuración inicial de los dedos y conjunto de puntos de contacto
Salida: Conjunto de trayectorias libres de colisiones para los dedos
AlgoritmoGeneticoCooperante() {
  repite para cada dedo {
    G = Generación inicial de individuos (posibles trayectorias)
    repite hasta satisfacer condición de terminación {
      Evaluación de la población de individuos
      si se satisface la condición de terminación {
        T = Conjunto de trayectorias libres de colisiones
      }
      de otra forma {
        Selección de soluciones para la siguiente generación
        G = Cruce y mutación para crear una nueva población
      }
    } // fin del ciclo de evaluaciones
  } // fin del ciclo para los dedos
} // fin del algoritmo
    
```

**Figura 5.** Algoritmo genético cooperante para determinar las trayectorias libres de colisiones de los dedos de una mano robótica

que forman un manipulador, lo que permite aplicarse a robots redundantes y no redundantes.

*Selección, cruce y mutación.* La selección de un individuo para realizar el cruce se realiza por medio de un mecanismo de torneo, donde los ganadores son aquellos dos individuos que tengan la mayor aptitud de los seleccionados aleatoriamente.

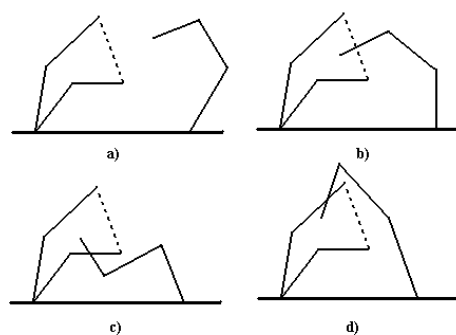
El cruce debe respetar las tuplas formadas para cada configuración; se obtiene el punto de cruce en cada uno de los individuos seleccionados y se realiza el intercambio. Este intercambio produce dos nuevos individuos que pueden tener distinta longitud. La figura 4 muestra un ejemplo de cruce. La mutación se realiza en dos pasos, uno para cada parte de la tupla  $(s,a)$ .

*Función de aptitud.* Está formada por una combinación de funciones que intentan describir una trayectoria óptima; las tres características son:

1. Ir del punto inicial al final; mientras más cerca esté el órgano terminal del punto final, mejor será la trayectoria.
2. Evitar completamente los obstáculos.
3. Ser lo más corta posible.

La cercanía se mide utilizando la fórmula de distancia euclidiana entre dos puntos; la longitud de una cadena se mide contando el número de configuraciones de la cadena. Finalmente, la función de aptitud tiene incluida una penalización para las colisiones encontradas.

La figura 5 muestra el algoritmo que determina las trayectorias libres de los dedos de una mano robótica usando un algoritmo genético cooperante.



**Figura 6.** Ejemplos de detección de colisiones: a) no existe colisión; b) colisión con la trayectoria; c) colisión con la configuración final; d) colisión con la configuración inicial. Fuente: De la Cueva y Ramos [10]

### 3.2 Detección de colisiones e intercambio de información

La figura 6 muestra los diferentes tipos de colisiones que se deben detectar para un sistema robótico múltiple; el robot de la izquierda muestra su posiciones inicial y final y la trayectoria (línea punteada) que sigue el órgano terminal; lo anterior forma un polígono, así que para encontrar colisiones se debe verificar que no exista cruce entre ninguno de los lados de dicho polígono y los segmentos del robot de la derecha; esto se verifica mediante el concepto de cruce de segmentos de recta.

El mecanismo de intercambio de información se hace usando la plantilla que contiene los mejores elementos de cada población. Para la evaluación de una población, se utiliza la plantilla para colocar cada uno de sus elementos que, junto con el mejor de la otra población, son utilizados para establecer su aptitud.

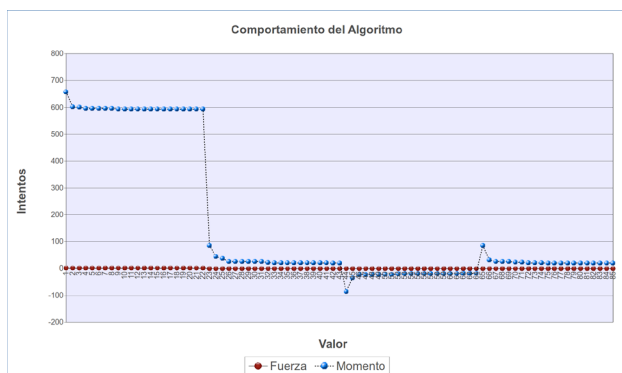


Figura 7. Comportamiento de las variantes del equilibrio, para un contorno cuadrado, con dos contactos sin fricción

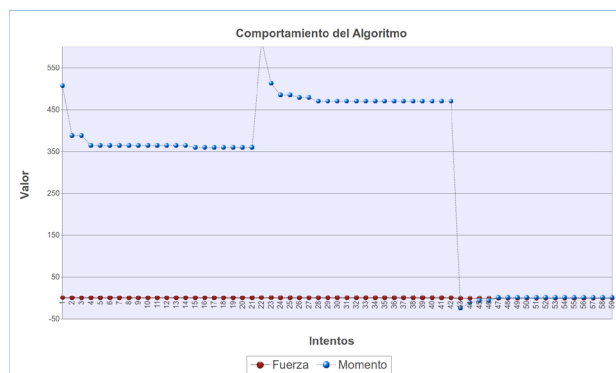


Figura 9. Comportamiento de las variantes del equilibrio, para un contorno no convexo, con dos contactos y un coeficiente de fricción de 0.2

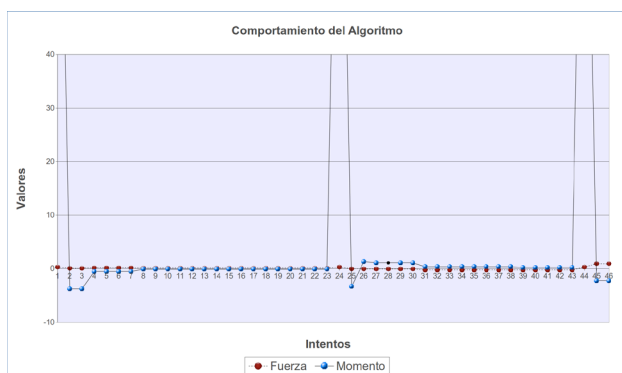


Figura 8. Comportamiento de las variantes del equilibrio, para un contorno pentagonal, con un coeficiente de fricción de 0.4 y cinco contactos

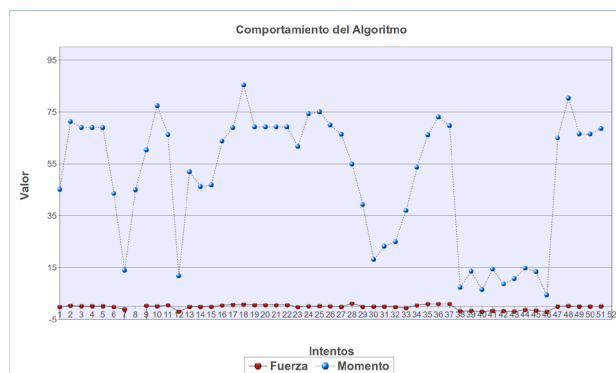


Figura 10. Comportamiento de las variantes del equilibrio, para un contorno curvo, para cuatro contactos y un coeficiente de fricción de 0.4

#### 4 PRUEBAS Y ANÁLISIS DE RESULTADOS

Se realizaron varias pruebas con las siguientes modificaciones: el tipo de objeto para sujetar, las condiciones de fricción y el número de dedos que se utilizan para la sujeción.

##### 4.1 Pruebas de sujeción

*Primer grupo de prueba.* En este caso se probó el funcionamiento del algoritmo con un polígono regular (un cuadrado). En los experimentos se observó que aun en presencia de fricción, el algoritmo no puede encontrar una solución para sujetar el objeto con tres o cinco dedos. La figura 7 muestra el comportamiento que tuvieron tanto la suma de fuerza como la de momentos para la prueba número uno de este experimento.

*Segundo grupo de prueba.* El segundo grupo de prueba también incluye un polígono regular (un

pentágono). La figura 8 muestra el comportamiento que tuvieron tanto la suma de fuerza como la de momentos para la prueba 39 de este experimento, que es el caso de cinco dedos con un coeficiente de fricción de 0.4.

*Tercer grupo de prueba.* El tercer grupo de prueba también incluye un polígono (cóncavo en dos vértices). La figura 9 muestra el comportamiento que tuvieron tanto la suma de fuerza como la de momentos para la prueba 13 de este experimento, que es el caso de dos dedos con un coeficiente de fricción igual a 0.2.

*Cuarto grupo de prueba.* El cuarto grupo de prueba se hace con un contorno representado por un *spline* (el cual también tiene concavidades). La figura 10 muestra el comportamiento que tuvieron tanto la suma de fuerza como la de momentos para la prueba 45 de este experimento, que es el caso de cuatro dedos con un coeficiente de fricción igual a 0.4.

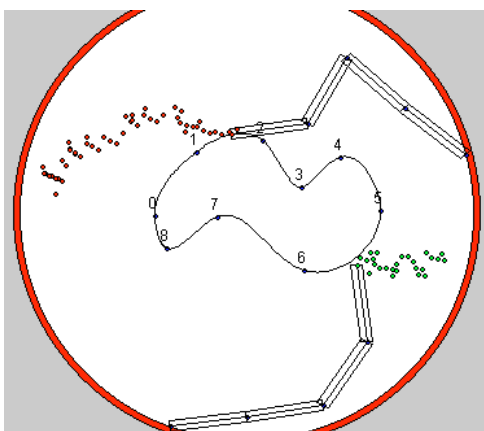


Figura 11. Trayectoria generada para dos dedos

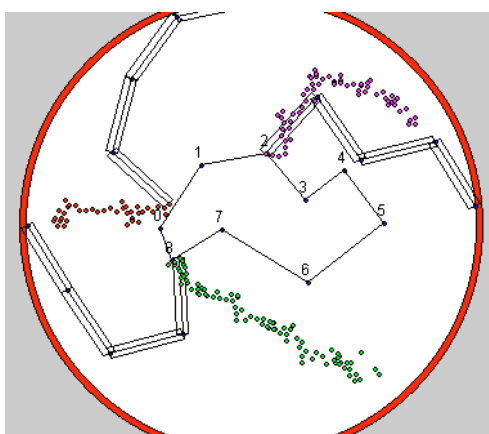


Figura 12. Trayectoria generada para tres dedos

#### 4.2 Pruebas de generación de trayectorias

Al aplicar el AGC en este modelo, se utilizaron parámetros iguales a los señalados por De la Cueva y Ramos [10], es decir, las poblaciones son de 30 individuos, el número de elementos para el torneo de selección es 6, la probabilidad de cruce es 0.6 y la probabilidad de mutación es 0.01. Se realizaron pruebas para obtener trayectorias para dos y tres dedos.

*Trayectorias de dos dedos.* La figura 11 muestra las trayectorias generadas por el AGC para dos dedos. En esta prueba se crearon doce generaciones, se calcularon 36 colisiones con obstáculos fijos, sin reportar colisiones entre dedos.

*Trayectorias para tres dedos.* La figura 12 muestra las trayectorias generadas por el AGC para tres dedos. En esta prueba se crearon diecinueve generaciones, se calcularon 2 039 colisiones con obstáculos fijos y 84 colisiones entre dedos.

De lo anterior podemos establecer que el AGC encuentra buenas soluciones para las trayectorias de los dedos, pero debido a que el espacio de trabajo que comparten es más reducido (debido a la palma de la mano), el tiempo de cálculo aumenta considerablemente.

#### 5 CONCLUSIONES

El algoritmo de sujeción propuesto encuentra, en la mayoría de los casos, los puntos de contacto que aseguran equilibrio en la sujeción. Se probó para varios objetos simples y amorfos, y con diferentes configuraciones de la mano.

Las ventajas de este algoritmo son: no está restringido al número de puntos de contacto ni a la forma del contorno del objeto, y el algoritmo puede encontrar puntos de contacto en cualquier tipo de objeto; por lo cual, una aportación importante de este trabajo es precisamente que no existen restricciones de convexidad para los objetos analizados.

Una desventaja del modelo es que, aunque es confiable en el sentido de producir soluciones correctas, es incompleto, debido a que puede no encontrar una configuración de sujeción que asegure equilibrio, aunque ésta exista.

Otro punto en contra es el tiempo de ejecución, que está en función del número de segmentos de curva y el número de reinicios y reintentos, por lo cual, no podría ser aplicado en línea para la ejecución de una tarea de sujeción.

Para el cálculo de trayectorias, al utilizar el AGC en este dominio se observa que es aplicable en la determinación de trayectorias para dedos de una mano robótica, con el inconveniente del tiempo que se invierte en el cálculo. Este tiempo es mayor al reportado [10] debido a que el espacio de trabajo en la mano bidimensional está restringido al interior de una circunferencia. Sin embargo, este método es interesante porque no requiere modificarse para aplicarlo a diferentes configuraciones como el número de dedos, el número de segmentos, la longitud de los segmentos, etcétera, y trabaja en el espacio de los manipuladores. Lo último es importante, ya que los enfoques actuales para el cálculo de trayectorias en manipulación diestra siempre utilizan el modelo de cinemática inversa para una configuración de mano específica, aunque presenten los inconvenientes ya comentados.

## REFERENCIAS

1. Mishha, B., Silver, N. Some discussion of static gripping and its stability. *IEEE Transactions on Systems, Man and Cybernetics*. 1989, 19 (1), 783-796.
2. Kerr, J., Roth, B. Analysis of multifingered hand. *International Journal of Robotics Research*. 1989, 4 (4), 3-17.
3. Han, L., Trinkle, J. C., Li, Z. X. Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics and Automation*. 2000, 16 (6), 663-674.
4. Michelman, P. Precision object manipulation with a multifingered robot hand. *IEEE Transactions on Robotics and Automation*. 1998, 14 (1), 105-113.
5. Muñoz, L. A., Bard, C., Nájera, J. A. Dexterous manipulation: A geometrical reasoning point of view. *Proceedings of the 1995 International Conference on Robotics and Automation (ICRA)*. 1995, (1), 458-463.
6. Bicchi, A. Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*. 2000, 16 (6), 652-662.
7. Overgaard, L., Nelson, B. J., Khosla, P. K. A multi-agent framework for grasping using visual servoing and collision avoidance. *Proceedings of the 1996 International Conference on Robotics and Automation (ICRA)*. 1996, (3), 2456-2461.
8. Muñoz, L.A., Doersam, Th. Simple Object Manipulation with dexterous hands. *IFAC – International Conference of Industrial Systems*. Belfort, France, 20-22 May 1997.
9. Michalewicz, Z., Michalewicz, M. Evolutionary Computation, techniques and their Applications. *Proceedings of the 1997 IEEE International Conference on Intelligent Processing Systems (ICIPS'97)*. 1997, (1), 14-25.
10. De la Cueva, V., Ramos, F. Cooperative genetic algorithms: a new approach to solve the path planning problem for cooperative robotic manipulators sharing the same work space. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1998, (1), 267-272.
11. Ponce, J., Sullivan, S., Sudsang, A., Boissonnat, J.D., Merlet, J.P. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *International Journal of Robotics Research*. 1997, 16 (1), 11-35.
12. Teichmann, M. *Grasping and fixturing*. Tesis doctoral. New York: New York University, 1995.
13. Liu, Y. H. Qualitative test and force optimization of 3-D frictional form-closure grasps using linear programming. *IEEE Transactions on Robotics and Automation*. 1999, 15 (1), 163-173.
14. Gill, M.A., Zomaya, A.Y. *Obstacle avoidance in multi-robot systems: Experiments in parallel genetic algorithms*. World Scientific Publishing, 1998.
15. Mazer, E., Ahuactzin, J.M., Bessiere, P. The Ariadne's clew algorithm. *Journal of Artificial Intelligence Research (JAIR)*. 1998, 9, 295-316.

*Acerca de los autores*



Rafael Rivera López es Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Veracruz y Maestro en Ciencias de la Computación por el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM). Actualmente es profesor-investigador en el Departamento de Computación y Sistemas del Instituto

Tecnológico de Veracruz. Sus áreas de interés incluyen la programación orientada a objetos y la optimización.



Luis Alberto Muñoz Ubando es Técnico en Computación por el Instituto Politécnico Nacional; Ingeniero en Sistemas por el ITESM; Maestro en Ciencias con especialidad en Matemáticas Aplicadas y Doctor en Informática con especialidad en Robótica y Procesamiento de Imágenes, ambas por el Inria de Francia, experto en robótica.

Actualmente es director del Instituto de Robótica de Yucatán y director de Innovación y Desarrollo Tecnológico de Grupo Plenum; además participa en proyectos de robótica submarina y de robótica satelital en diversas redes de investigación a nivel nacional e internacional. Ha trabajado en las universidades de Pisa en Italia, de Tokyo en Japón, de Karlsruhe en Alemania, de Stanford y de Massachusetts en Estados Unidos. Ha publicado más de 50 artículos entre revistas, libros y conferencias tanto nacionales como internacionales.



Fernando Ramos Quintana es profesor-investigador del ITESM, Campus Cuernavaca y miembro del Sistema Nacional de Investigadores; es director de investigación y del posgrado en computación en el mismo campus, y coordinador del Proyecto de Fomento a la Investigación en Licenciatura (Profil), donde se encuentra el laboratorio de robots humanoides. En el año 2004, dos de los equipos que dirige ganaron al medalla de oro y la de bronce en robots bípedos danzantes dentro del marco de la 6a Olimpiada Internacional de Robótica, celebrada en el KAIST, de Corea del Sur. Su área principal de investigación son los robots que cooperan en la realización de una tarea. Tiene bajo su cargo el proyecto de desarrollo de plataformas de aprendizaje en robótica para niños de 5 a 12 años.