

Cinemática Inversa de un Brazo Robot Utilizando Algoritmos Genéticos

Luis Felipe Giraldo, Edilson Delgado y Germán Castellanos

UNIVERSIDAD NACIONAL DE COLOMBIA.

luspiper16@yahoo.com ; edelgadot@unal.edu.co ; gcastell@telesat.com.co

Recibido para revisión May-2006, aceptado Jun-2006, versión final recibida Jun-2006

Resumen: Se describe el procedimiento para resolver el problema de la cinemática inversa de un brazo robot utilizando algoritmos genéticos a partir de la solución del problema cinemático directo. Se demuestra que este método tiene varias ventajas sobre otros métodos: el mismo algoritmo aplica para dar solución a la cinemática inversa de cualquier brazo robot, independientemente del tipo de articulaciones y del número de grados de libertad; se obtiene una solución factible en forma rápida, resaltando la sencillez en la implementación; se utiliza la representación de elementos de Denavit-Hartenberg, caracterizada por su universalidad algorítmica. Al algoritmo genético utilizado se le realiza un control en sus parámetros y en la función de evaluación con el fin de mejorar su rendimiento ante el problema propuesto.

1 INTRODUCCIÓN

Un manipulador mecánico, se puede modelar como una cadena articulada en lazo abierto con unos elementos conectados en serie por una articulación de revolución o prismática movida por actuadores. El movimiento relativo en las articulaciones resulta en el movimiento de los elementos que posicionan la mano en una orientación deseada.

En la mayoría de las aplicaciones de robótica, se está interesado en la descripción espacial del efector final del manipulador con respecto a un sistema de coordenadas de referencia fija, para lo cual necesariamente se debe resolver el problema de la cinemática directa y la cinemática inversa. Para un manipulador determinado, la cinemática directa consiste en hallar la orientación y posición del efector final a partir del vector de ángulos de las articulaciones y los parámetros geométricos del elemento.

En Denavit y Hartenberg (1955) se propone un método matricial para resolver en forma sistemática y generalizada este problema. La cinemática inversa consiste en hallar el vector de ángulos de las articulaciones a partir de la orientación y posición del efector final, el cual es un problema de difícil solución debido a que incluye ecuaciones no lineales y múltiples soluciones. Este problema ha sido abordado de muchas formas planteándose diversos métodos para resolverlo. En Pieper (1968) se presenta la solución cinemática para manipuladores con seis grados de libertad en donde tres ejes consecutivos se interceptan en un punto.

En Lee y Ziegler (1984) se descompone la geometría

espacial del brazo en varios problemas de geometría planar. En Paul, Shimano y Mayer (1981) se propone la técnica de la transformada inversa, en donde se intenta, a través de diversas aproximaciones resolver los ángulos de Euler. Estos métodos por aproximaciones geométricas presentan muchas condiciones en cuanto a la estructura física del manipulador y el tipo de articulaciones, además de la dificultad en encontrar la solución.

Carvalho y Gaspar (1991), Oyama, Chong, Agah, Maeda y Tachi (2001) proponen varios métodos utilizando redes neuronales; las redes implementadas necesitan muchos ejemplos para que puedan ser entrenadas, además de que no se alcanza una precisión adecuada. Varios autores han reportado el uso de algoritmos evolutivos para solucionar el problema de la cinemática inversa de un manipulador para puntos individuales [Khoogar, Parker y Goldberg (1989), Gibbs (1996), Khawaja, Rahman y Wagner (1998)], requiriéndose muchas generaciones para que el algoritmo llegue al óptimo ó al cercano-óptimo.

En este trabajo se propone realizar la solución al problema cinemático inverso de un brazo robot utilizando algoritmos genéticos a partir de la representación Denavit-Hartenberg, independiente del tipo de articulaciones y del número de grados de libertad, resaltándose la rápida convergencia del algoritmo mediante el control de sus parámetros y la función de evaluación. El brazo robot tomado como referencia para realizar las pruebas es el Scorbot VR plus, con articulaciones rotacionales y cinco grados de libertad, adquirido por la Universidad Nacional de Colombia sede Manizales.

2 CINEMÁTICA DIRECTA DEL BRAZO ROBOT

Una herramienta indispensable para describir la geometría espacial de un manipulador es la representación en coordenadas homogéneas. El concepto de una representación en coordenadas homogéneas en un espacio euclídeo tridimensional es útil para desarrollar transformaciones matriciales que incluyan rotación, traslación, escalado y transformación de perspectiva.

En general, una matriz de transformación homogénea es una matriz que transforma un vector de posición expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas. Una matriz de transformación homogénea se puede considerar que consiste en cuatro submatrices:

$$T = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \mathbf{f}_{1 \times 3} & \kappa_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{matriz de rotación} & \text{vector posición} \\ \text{transformación de perspectiva} & \text{escalado} \end{bmatrix}$$

La submatriz superior izquierda representa la matriz de rotación; la submatriz superior derecha representa el vector de posición del origen del sistema de coordenadas rotado con respecto al sistema de referencia; la submatriz inferior izquierda representa la transformación de perspectiva; y el cuarto elemento diagonal es el factor de escala global. En aplicaciones de robótica, este factor de escala será siempre igual a 1.

Para analizar la cinemática del brazo robot se utiliza la representación de Denavit-Hartenberg, la cual establece en forma sistemática un sistema de coordenadas para cada elemento de la cadena articulada. Dicha representación resulta en una matriz de transformación homogénea que representa cada uno de los sistemas de coordenadas de los elementos en la articulación con respecto al sistema de coordenadas del elemento previo, denominada matriz de transformación D-H. Para la aplicación particular, se establece el sistema de coordenadas como se ilustra en la Figura 1.

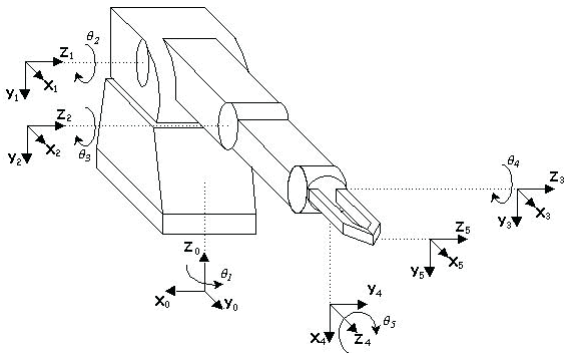


Figura 1: Establecimiento del sistema de coordenadas para el Scorbot VR plus

A partir del sistema de coordenadas elegido y las medidas del brazo robot, se hallan los parámetros de la representación de Denavit-Hartenberg (ver Tabla I) para ser reemplazados en la matriz de transformación D-H en sistemas de coordenadas adyacentes i e $i-1$, notados en la expresión (1). Estos cuatro parámetros describen totalmente cualquier articulación prismática o de revolución.

Para calcular los parámetros de la representación Denavit-Hartenberg se tienen las siguientes consideraciones:

θ_i : Es el ángulo de la articulación del eje x_{i-1} al eje x_i respecto del eje z_{i-1} (utilizando la regla de la mano derecha).

d_i : Es la distancia desde el origen del sistema de coordenadas $i-1$ hasta la intersección del eje z_{i-1} con el eje x_i a lo largo del eje z_{i-1} .

a_i : Es la distancia de separación desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo a lo largo del eje x_i (ó la distancia más corta entre los ejes z_{i-1} y z_i).

α_i : Es el ángulo de separación del eje z_{i-1} al eje z_i respecto del eje x_i (utilizando la regla de la mano derecha).

Tabla 1: Parámetros de coordenadas del brazo del robot

Articulación	α_i	a_i (mm)	d_i (mm)
1	-90^0	0	364
2	0^0	220	0
3	0^0	220	0
4	90^0	0	0
5	0^0	0	170

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i - \cos \alpha_i \sin \theta_i \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i \cos \alpha_i \cos \theta_i - \sin \alpha_i \cos \theta_i a_i \sin \theta_i & \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

A partir de la matriz de transformación descrita en la expresión (1), se halla la matriz homogénea 0T_i , la cual especifica la localización del sistema de coordenadas i -ésimo con respecto al sistema de coordenadas base. Esta matriz está dada por:

$${}^0T_i = \prod_{j=1}^{i-1} A_j \quad (2)$$

El problema cinemático directo consiste en hallar la posición y orientación final del brazo robot a partir de los ángulos de articulación $\hat{\theta} = \{\theta_1, \theta_2, \dots, \theta_i\}$. Utilizando la matriz de transformación 0T_i se relaciona un punto p_i en reposo en el elemento i expresado en coordenadas homogéneas con respecto al sistema de coordenadas i en el

sistema de coordenadas base establecido en el elemento base por la expresión:

$$p_0 = {}^0T_i p_i \quad (3)$$

Teniendo en cuenta que el brazo robot es de cinco grados de libertad, y que el punto del espacio que se quiere obtener es el origen del sistema de coordenadas de la herramienta, se reemplaza en las expresiones (2) y (3), obteniéndose:

$$p_0 = {}^0T_5 [0 \ 0 \ 0 \ 1]^t \quad (4)$$

donde,

$${}^0T_5 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5$$

Se resuelve la multiplicación de la expresión (4) y se obtiene:

$$p_0 = [x \ y \ z \ 1]^t \quad (5)$$

donde,

$$x = {}^0T_5(1,4), \quad y = {}^0T_5(2,4), \quad z = {}^0T_5(3,4)$$

siendo (x, y, z) la posición espacial del origen del sistema de coordenadas del efector final respecto al sistema de coordenadas base.

3 ESTRUCTURA GENERAL DE LOS ALGORITMOS GENÉTICOS

Los algoritmos genéticos (AGs) son procedimientos de búsqueda basados en los principios de la selección natural, genética y evolución. Se asume que la evolución de los seres vivos es un proceso que se opera en los cromosomas - aparatos orgánicos codificadores de la estructura de los seres vivos; de esta forma, los AGs resuelven problemas de encontrar buenas cromosomas sin algún conocimiento del tipo de problema que se está resolviendo. Dado un método de codificar soluciones de un problema en la forma de n cromosomas de longitud L y dada una función de evaluación que proporcione una medida del costo γ de algún cromosoma en el contexto del problema. En el Algoritmo 1 se describe el procedimiento que generalmente siguen los algoritmos genéticos.

Algoritmo 1. Pseudocódigo de los AGs

Se requiere: p_m y p_c

$t := 0;$
 inicializar $(\vec{x}(0) = \{\vec{x}_1(0), \vec{x}_2(0), \dots, \vec{x}_n(0)\})$ // Población
 // inicial

repetir

$\vec{\gamma} := \text{evaluar}(\vec{x}(t))$ // Evaluar cada miembro de la población
 // y obtener medida de costo

$\vec{x}_m(t) := \text{selección}(\vec{x}(t), \vec{\gamma})$ // Seleccionar mejores
 // cromosomas con base en γ

$\vec{x}_c(t) := \text{cruzamiento}(\vec{x}_m(t))$ // Realizar cruzamiento y
 // obtener población de crías

$\vec{x}(t) := \vec{x}(t) - \vec{x}_{rand}(t) + \vec{x}_c(t)$; // Eliminar miembros de la
 // población aleatoriamente e
 // ingresar crías

$\vec{x}(t) := \text{mutación}(\vec{x}(t))$ // Aplicar mutación

$t := t+1;$

hasta que se satisfaga el criterio de parada

4 MARCO EXPERIMENTAL

El esquema del algoritmo implementado para dar solución al problema cinemático inverso se ilustra en la Figura 2. Se puede observar el control realizado a los Parámetros del algoritmo genético y la función de evaluación utilizada.

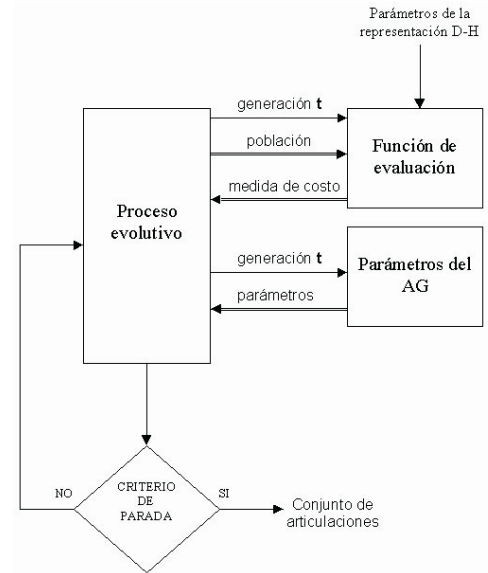


Figura 2: Diagrama de bloques del algoritmo implementado

Para resolver el problema de la cinemática inversa se utiliza un AG simple con las siguientes especificaciones:
Codificación: Los ángulos de rotación $\hat{\theta} =$

$\{\theta_1, \theta_2, \dots, \theta_i\}$ se codifican en forma de cromosomas con cadenas binarias de 15 bits por cada parámetro, en un código concatenado, mapeado, multiparámetro de punto fijo.

$$\left| \begin{array}{c|c|c|c|c} 01 \dots 01 & 00 \dots 10 & \dots & 11 \dots 00 & \\ \hline \theta_1 & \theta_2 & \dots & \theta_5 & \end{array} \right|$$

Función de evaluación: Cada cromosoma se decodifica, obteniendo así los valores de los cinco ángulos de rotación, los cuales están dentro de un rango dado por las especificaciones del brazo robot. Estos valores se reemplazan en la expresión (5) para calcular el punto en el espacio resultante después de posicionar el brazo robot con estos ángulos.

Por lo tanto, la medida de costo γ del cromosoma debe ser inversamente proporcional a la distancia entre el punto obtenido a partir del cromosoma y el punto para el cual la herramienta del brazo robot debe posicionarse respecto al sistema de coordenadas base. La función utilizada se describe en la expresión (6).

$$\gamma = \exp \left[- \frac{d^2(p_r, p_c)}{2\sigma^2} \right] \quad (6)$$

donde $d^2(p_r, p_c)$ es la distancia entre el punto p_r para el cual la herramienta del brazo robot debe posicionarse en el punto p_c obtenido a partir del cromosoma, y σ es la desviación estándar de la función exponencial, cuyo valor depende de las características físicas del problema. Para el cromosoma óptimo $\gamma = 1$.

Para puntos alejados del punto objetivo ($d > 3\sigma$), γ es casi nulo. De acuerdo a esto, si la población inicial contiene cromosomas que generan puntos de este tipo, la medida de costo va a proveer muy poca información y el algoritmo no va a poder direccionar el proceso evolutivo hacia el óptimo ó cercano-óptimo. Con el fin de evitar este inconveniente, la función de evaluación dada por la expresión (6), se plantea de la siguiente forma: para las generaciones iniciales, γ toma un valor grande, ampliando la función de evaluación, de tal forma, que los cromosomas que generan puntos alejados del punto objetivo, tengan γ un considerable, lográndose que el algoritmo pueda direccionar la evolución de los mismos. A medida que el número de generaciones aumenta, γ se debe hacer más pequeño con el fin de obtener una mayor precisión en los cromosomas. Así, γ queda definido por la expresión (7), siguiendo un esquema determinístico lineal.

$$\sigma(t) = 50 - 49 \frac{t}{T} \quad (7)$$

donde, t es la generación actual y T es el número máximo de generaciones. Se toma $\sigma(0) = 50$, y $\sigma(T) = 1$. Reemplazando la expresión (7) en la expresión (6) se obtiene la nueva función de evaluación, dada por la expresión (8) e ilustrada en la Figura 3.

$$\gamma(d, t) = \exp \left[- \frac{d^2(p_r, p_c)}{2(50 - 49 \frac{t}{T})^2} \right] \quad (8)$$

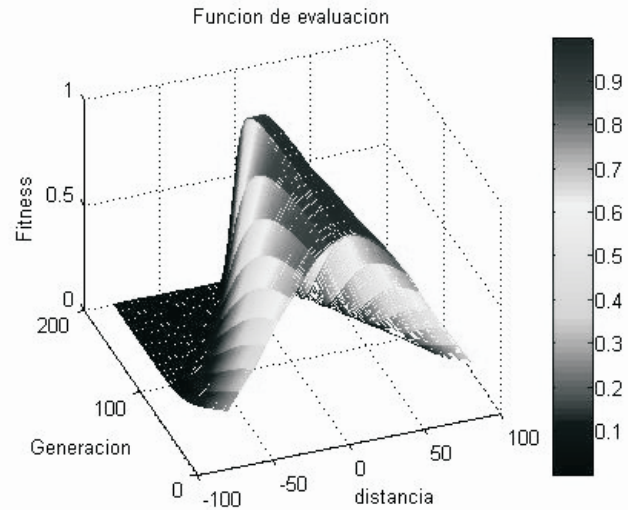


Figura 3: Gráfica de $\gamma(d, t)$

Mutación: Debido a que el espacio de búsqueda es bastante amplio, se necesita que el algoritmo sea eficiente y que encuentre una solución lo más cercana posible al óptimo. Para obtener lo anterior, se elige realizar un control en la tasa de mutación, la cual es el operador decisivo en la búsqueda de un algoritmo evolutivo [Beyer (2001)]. En Back (1993) se demuestra que es un buen método la probabilidad de mutación, siendo p_m la probabilidad de mutación y L la longitud del cromosoma; sin embargo, se ha encontrado que variar la tasa de mutación durante la ejecución del AG provee mejores resultados [Eiben, Hinterding y Michlewicz (1999)]. Por lo tanto, se utiliza un esquema de control determinístico de la probabilidad de mutación en función de la generación, dada por la expresión (9), obteniendo mejores resultados.

$$p_m(t) = \left[1 + \frac{L-1}{T} t \right]^{-1} \quad (9)$$

donde, T es el número total de generaciones y t es la generación actual, siendo $0 \leq t \leq T$. Esta función decreciente p_m tal que $p_m(0) = 1$ y $p_m(T) = \frac{1}{L}$.

Otros parámetros: En De Jong (1975) y Grefenstette (1986), se determina experimentalmente valores adecuados para los parámetros de un algoritmo genético, de los cuales se eligieron los siguientes: tasa de cruzamiento: 0.95; tamaño de la población: 50; número de bits: 15; criterio de parada: el AG termina cuando se ejecuten el número total de generaciones, o cuando se supere un umbral de la función de evaluación.

5 RESULTADOS Y DISCUSIÓN

Como se observa en la Figura 4, después de ejecutar el algoritmo 100 veces con $\sigma = 1.7$ y 300 generaciones, hay 23 ejecuciones que no alcanzaron a superar el umbral establecido de $d(p_r, p_c) = 0.14mm$. Después de ejecutar el algoritmo con la nueva función de evaluación (ecuación 8) 100 veces, con un número máximo de 300 generaciones y el mismo umbral, se obtienen los resultados mostrados en la Figura 5. Se observa que el óptimo (ó cercano-óptimo) es alcanzado entre 50 y 175 generaciones aproximadamente, con una efectividad del 99%.

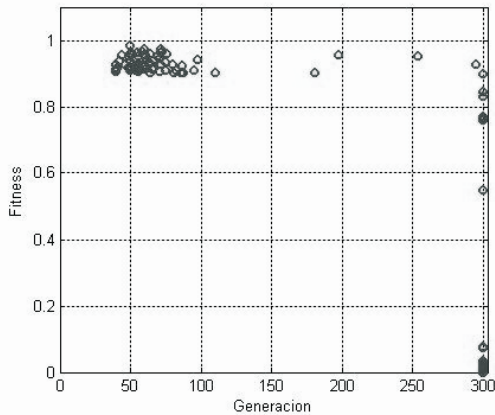


Figura 4: Resultados para $\sigma = 1.7$

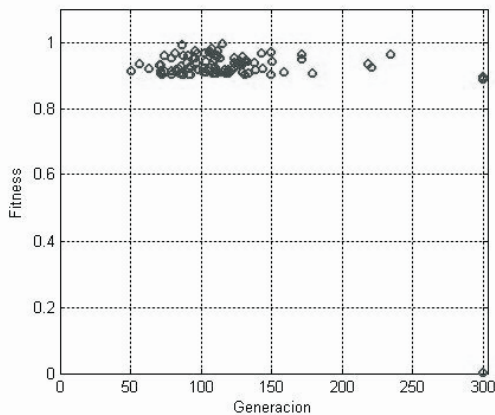
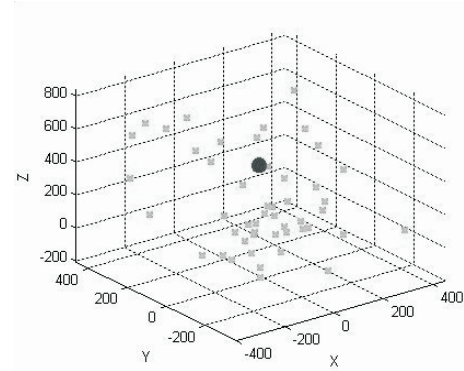
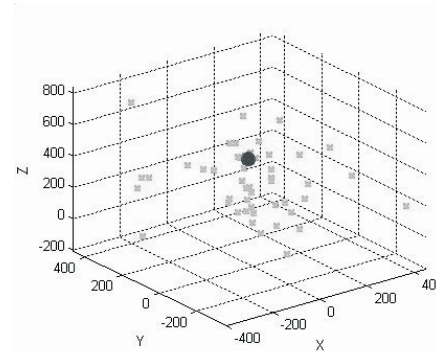


Figura 5: Resultados para σ variable

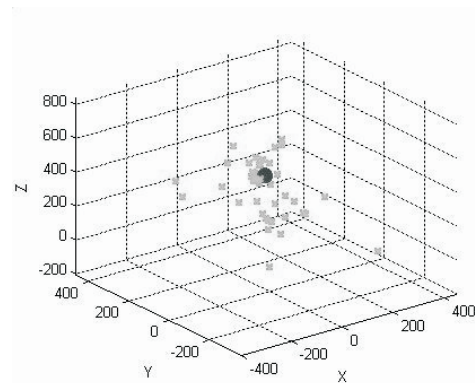
En la Figura 6 se observa el progreso de los cromosomas. Después de 40 generaciones se alcanza el criterio de parada especificado anteriormente. La población de cromosomas se representa en su valor equivalente en el espacio tridimensional mediante las 'x', y el punto que se desea alcanzar se representa por medio de un círculo de gran tamaño. Como se observa, a medida que el número de generaciones aumenta, la población se acerca cada vez más al óptimo.



(a) Estado de la población en la generación 5



(b) Estado de la población en la generación 15



(c) Estado de la población en la generación 40

Figura 6: Proceso evolutivo

6 CONCLUSIONES Y TRABAJO FUTURO

Se demuestra que el algoritmo es efectivo y eficiente al hallar la cinemática de un brazo robot, a pesar de que el espacio de búsqueda es bastante amplio.

Otros métodos antes propuestos, en la solución del problema de la cinemática inversa de un brazo robot, son complejos de desarrollar, y es por esto que, o sólo quedan aplicables a casos específicos, o si encuentran la solución es en un tiempo muy prolongado. El método que se propone en este trabajo es de fácil implementación y rápida convergencia, dejando el camino abierto para que

se mejore conforme al desarrollo de nuevas técnicas en el área de la computación evolutiva.

Se debe tener en cuenta que un brazo robot puede posicionarse en un punto del espacio de muchas maneras, proporcionalmente al número de grados de libertad que tenga, siempre y cuando esté dentro de su espacio de trabajo y las características físicas lo permitan; por lo tanto, pueden existir muchas soluciones al problema de la cinemática inversa.

El algoritmo implementado direcciona el proceso evolutivo hacia una de las soluciones factibles en forma aleatoria. Queda planteado como trabajo futuro la aplicación de restricciones al algoritmo para que encuentre tan sólo la solución de interés, utilizando acciones de penalización en la función de evaluación, las cuales también pueden ser controladas durante la ejecución del algoritmo [Eiben et al. (1999)].

REFERENCIAS

- Back, T. (1993), Optimal mutatin rates in genetic search, in S. Forrest, ed., 'Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann'.
- Beyer, H. (2001), *The Theory of Evolution Strategies*, Leiden Center of Natural Computing. Springer-Verlag.
- Carvalho, L. y Gaspar, E. (1991), The solution of the inverse kinematic problem of robot arms with neural networks, in 'XI Brazilian Congress on Mechanical Engineering COBEM - Brasil'.
- De Jong, K. (1975), The Analysis of the Behaviour of a Class of Genetic Adaptive Systems, PhD thesis, University of Michigan, Ann Arbor, Michigan.
- Denavit, J. y Hartenberg, R. (1955), 'A kinematic notation for lower-pair mechanisms on matrices', *ASME Journal of Applied Mechanics* pp. 215–221.
- Eiben, A., Hinterding, R. y Michlewicz, Z. (1999), 'Parameter control in evolutionary algorithms', *IEEE Transactions on Evolutionary Computation* **3**(2), 124–141.
- Gibbs, J. (1996), Easy inverse kinematics using genetic programming, in 'Proceedings of the GP-96 Conference', p. 28.
- Grefenstette, J. (1986), 'Optimization of control parameters for genetic algorithms', *IEEE Transactions on Systems, Man and Cybernetics* **16**(1), 122–128.
- Khawaja, A., Rahman, M. y Wagner, M. (1998), *Inverse Kinematics of Arbitrary Robotic Manipulators using Genetic Algorithms*, Kluwer Academic Publishers. In J. Lenarcic y M.L. Justy, editores. *Advances in Robot Kinematics: Analysis and Control*.
- Khoogar, A., Parker, J. y Goldberg, D. (1989), Inverse kinematics of redundant robots using genetic algorithms, in 'International Conference on Robotics and Automation', p. 271.
- Lee, C. y Ziegler, M. (1984), 'Geometric approach in solving inverse kinematics of puma robots', *IEEE Transactions on Aerospace and Electronic Systems* **20**(5).
- Oyama, E., Chong, N., Agah, A., Maeda, T. y Tachi, S. (2001), Inverse kinematics learning by modular architecture neural networks with performance prediction networks, in 'IEEE International Conference on Robotics and Automation', p. 1006.
- Paul, R., Shimano, B. y Mayer, G. (1981), 'Kinematic control equations for simple manipulators', *IEEE Transactions on Systems, Man and Cybernetics* **11**(6).
- Pieper, D. (1968), The Kinematics of Manipulators Under Computer Control, PhD thesis, Stanford University.