



UNIVERSIDAD DE CHILE

UNIVERSIDAD DE CHILE

EL7012

CONTROL INTELIGENTE DE SISTEMAS, OTOÑO

Ejercicio N°1

Autor:

Elias Obrequé
Gustavo Ceballo
Maibeth Sánchez

22 de abril de 2020

Índice

1. Introducción	1
2. Marco Teórico	1
2.1. Denavit Hartenberg	1
2.2. Algoritmos Genéticos	3
2.2.1. Creación de los individuos de la Población Original	4
2.2.2. Fitness	5
2.2.3. Selección de los individuos	6
2.2.4. Crossover y mutación	7
3. Metodología	8
3.1. Selección de los hiperparámetros de fitness	9
4. Resultados	11
5. Conclusión	13

1. Introducción

Las recientes décadas han visto un crecimiento cada vez mayor en la implementación de tecnología y robotización en industrias de diversas índoles, un reflejo de esto es la implementación de brazos robóticos en líneas de montaje, un problema bastante interesante en esta área es la trayectoria óptima de un brazo robotico entre un origen y destino dados, siendo este óptimo un parámetro medible mediante diversos factores como: un menor consumo energético, tiempo de traslado, torque involucrado, entre otros.

Intuitivamente se puede deducir que este tipo de problemas conlleva una búsqueda y optimización exhaustiva, es por esto que se hace uso de una rama de la inteligencia artificial conocida como algoritmos genéticos la cual permite que conjunto de individuos (en este caso soluciones) sean sometidos a un proceso de selección, reproducción y mutación selectiva produciendo de esta manera individuos con mejores características.

Esto es una prueba para el trabajo offline.

2. Marco Teórico

El problema a abordar es el siguiente: dada una posición original del brazo robotico que se observa en la figura 1 es necesario calcular la trayectoria angular que deben seguir las diferentes articulaciones para alcanzar una posición objetivo mediante la aplicación de algoritmos genéticos, para estos se debe diseñar una función de fitness que además de tomar en cuenta la posición objetivo, considere una aproximación de la energía total gastada, de tal forma que las trayectorias sean cercanas a un movimiento óptimo

La implementación y funcionamiento de algoritmos genéticos junto con el método de Denavit Hartenberg (que permite convertir el movimiento angular en trayectorias) se explican a continuación:

2.1. Denavit Hartenberg

El movimiento de cada eslabón de un brazo robótico es llamado grado de libertad, así un brazo robótico con n eslabones tiene n grados de libertad, si se toma cada eslabón como un elemento aislado del resto del brazo se puede apreciar que el movimiento que este hace únicamente depende de la variación angular respecto de su origen y de su longitud o sea que cualquier trayectoria del brazo robotico estará compuestas de la variación de los ángulos y longitudes de cada eslabón además de la relación entre ellos. La complejidad de este problema viene dada en que el calculo de las trayectorias

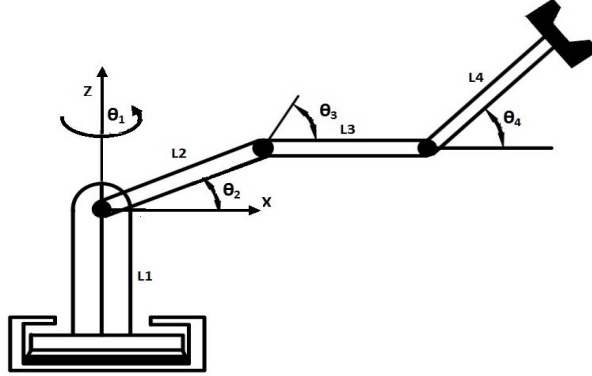


Figura 1: Brazo robotico con 4 grados de libertad.

Tabla 1: Parametros de DH

Eslabon(i)	a_i	α_i	d_i	θ_i
1	0	90°	0	θ_1
2	L_2	0	0	θ_2
3	L_3	0	0	θ_3
4	L_4	0	0	θ_4

posibles de un brazo compuesto de una cantidad relativamente pequeña de eslabones (3 por ejemplo) [1] da paso a sistemas de ecuaciones de orden 32 con al menos 16 soluciones complejas. Es en este tipo de problemas que son difícilmente abordables por su complejidad matemática en donde el método de Denavit Hartenberg (DH) sale a relucir, ya que este permite establecer la relación entre los eslabones adyacentes así como determinar la relación entre el un sistema de coordenadas origen y uno final mediante la relación entre cada par de eslabones de manera sistemática.

El primer paso es tabular los parámetros del brazo robotico, para el caso de un brazo robótico como el presentado en la sección 1 el resultado seria el de la tabla 1, en donde a_i es la longitud de cada eslabón, α_i es el ángulo entre los ejes X e Y , d_i representa la traslación de los eslabones respecto del eje X , si la coordenadas iniciales del primer eslabón están en el origen cartesiano (0,0,0) y no hay traslación en el eje X entonces sera 0 para todos los eslabones y finalmente θ_i es el ángulo conformado entre X y Z .

El segundo paso es el calculo de la matriz A por cada uno de los eslabones, su notación es ${}^i_{i-1}A$ y representa la relación de cada articulación con la anterior, ecuación (1), en donde c y s son coseno y seno respectivamente y los índices se reemplazan por los componentes de la Tabla 1:

$${}^i_{i-1}A = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Finalmente y para obtener las coordenadas cartesianas fruto de la relación entre el eslabón de origen con cualquier otro es necesario obtener la matriz homogénea T , para esto se multiplican la matrices A como se observa en la ecuación (2) en donde las coordenadas cartesianas en 3D son d_x , d_y y d_z , los otros parámetros d_i junto con r_i están asociados a la rotación y el resto a la perspectiva.

$${}^i_0T = \prod_{j=1}^i ({}^j_{j-1}A) = \begin{bmatrix} r_{11} & d_{12} & d_{13} & d_x \\ r_{21} & d_{22} & d_{23} & d_y \\ r_{31} & d_{32} & d_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

2.2. Algoritmos Genéticos

Los algoritmos genéticos han nacido como herramienta para la optimización y solución de problemas en los que sus parámetros son difícil de calcular de forma analítica. Un ejemplo claro de esto son los problemas no lineales de fenómenos físicos, problemas para el cual ya se han desarrollado herramientas resolutivas [2]. El trasfondo del algoritmo genético se basa en crear un conjunto inicial de posibles soluciones (que puede ser expresada en un valor real o binario según corresponda el problema) y luego es evaluada en la función característica del problema.

Dependiendo del tipo de problema se pueden presentar dos escenarios: cuando se define un conjunto de valores de referencias a la que la función característica debe alcanzar como posición, velocidad, temperatura, presión, etc; o cuando simplemente se desea escoger el conjunto de solución que maximiza (minimiza) un término, por ejemplo, el empuje de motor. Este conjunto de solución posibles se evalúa por medio una función fitness (a definir por el usuario) que permite decidir cuantitativamente que valores de la solución fue más acertada.

Los mejores conjuntos solución irán evolucionando mediante avanzan en el tiempo (o número de generaciones) y crearan nuevos conjuntos que tendrán las propiedades de las generaciones pasados más favorables. Dependiendo de la rapidez con la que evoluciona la solución, se pueden aplicar técnicas de algoritmos genéticos para solucionar problemas de control robusto de alto grado de dificultad como sistemas

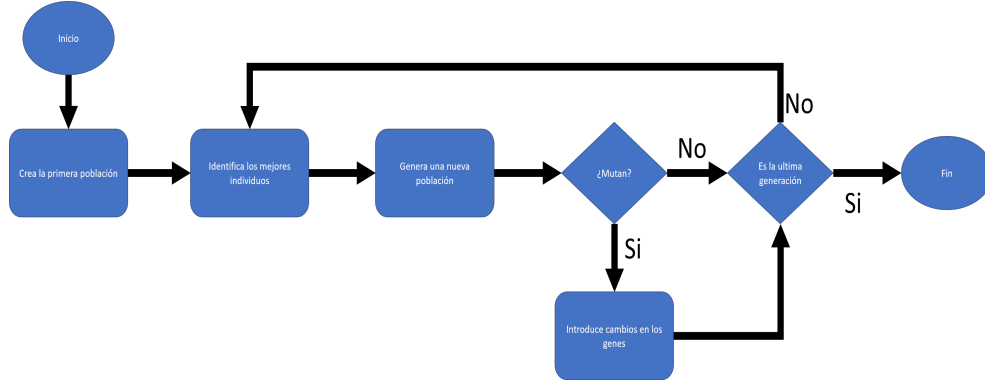


Figura 2: Diagrama de algoritmo genético.

no lineales [3]. Otro ejemplo cercano al problema mencionado en este proyecto se encuentra en [4] y [5] donde se busca solucionar la dinámica de un brazo robótico con algoritmo genético y explicando el uso de la dinámica inversa. El proceso evolutivo se presenta en el diagrama general de la Figura 2.

Para representar las soluciones en algoritmo genético se presentan las siguientes definiciones las cuales estan de acorde al problema presentado:

- Generación: Conjunto de trayectorias de G brazos independientes.
- Individuo: Trayectoria de cada brazo producto del movimiento de los eslabones que lo componen, denotado como I .
- Gen: Trayectoria individual de un eslabón denotado como g .
- Cromosoma: Cada uno de los k puntos cuyo conjunto representa la trayectoria de cada eslabón, denotado como i .

2.2.1. Creación de los individuos de la Población Original

El gen asociado a una articulación se crea según lo visto en ecuación (3), en donde $\theta_{inicial}$ y θ_{final} representan los ángulos máximos y mínimos en los cuales el brazo realiza su movimiento, permite que la población inicial presente una trayectoria cercana a la solución deseada, si bien no óptima. Por otra parte la ecuación (4) es una función tangente hiperbolica con limites entre $[0,1]$ que en fondo establece el comportamiento de cada articulación, en donde:

- i representa cada cromosoma siendo sus limites $2 \leq i \leq i_k - 1$ con i_k siendo la cantidad de cromosomas.

Tabla 2: Representación tabular de un individuo.

Gen/Eslabón	Cromosomas [Grados]					
	1	2	3	4	...	i_k
1	0	1,42-11	3,38e-11	6,10e-11	...	87.15
2	0	1,87	4,03	6,50	...	90
3	0	0	0	0	...	79.87
4	0	2,57e-05	5,41e-05	9,79e-05	...	89.99

- μ variable aleatoria con limites $1 \leq \mu \leq i_k$.
- σ variable aleatoria con limites $1 \leq \sigma \leq i_k/6$.

$$Gen(i) = \theta_{inicial} + (\theta_{final} - \theta_{inicial}) * W(i) \quad (3)$$

$$W(i) = \frac{1}{2} \left[1 + \tanh \left(\frac{i - \mu}{\sigma} \right) \right] \quad (4)$$

Ya que la ecuación (4) tiene una componente aleatoria, al utilizar esta en la ecuación (3) para la creación de los genes de cada individuos, hace mas probable tener distintas trayectorias. El como esto se represan tabularmente se puede observar en la tabla 2 en donde se tiene como los i_k cromosomas asociados a cada uno de los genes varían dentro del rango de un $\theta_{inicial}$ de 0° hasta un θ_{final} de 90° .

Una vez se tienen todos los individuos de la población original, cada uno con sus genes y cromosomas correspondientes es necesario traducir este movimiento angular en coordenadas cartesianas, por lo cual se hace uso del método DH explicado en la sección 2.1, a partir de esto se obtiene la trayectoria de los k individuos de la generación en 3 dimensiones.

2.2.2. Fitness

El fitness escogido relaciona el punto geométrico objetivo con el punto en el cual se encuentra la punta del brazo y la velocidad lineal del brazo en cada paso. Utilizando el método de Denavit Hartenberg se obtiene la posición cartesiana del brazo y se calcula la norma del vector diferencia denominado como E . De esta forma se puede penalizar el cumplimiento de la posición objetivo cambiando su peso al multiplicar E por un factor de ganancia β .

Para penalizar la distancia que hay de una muestra a otra (distancia entre cromosomas) que llamaremos *smoothness* se penaliza la velocidad de traslación de las

Figura 3: Trayectoria de un brazo mediante los puntos de muestreo.

muestras. Para ello, se calcula la velocidad de cada individuo en sus respectivos pasos, luego, se escoge la máxima velocidad de traslación de cada engranaje. Finalmente, se define V como la media de las velocidades máximas de cada engranaje. Este término también es multiplicado por una ganancia α de tal forma que la penalización de V tenga una correspondencia con E , es decir, se impone que β es el complemento de α ($\alpha + \beta = 1$).

Un caso que refleja un *smoothness* (V) demasiado alto se observa en la figura 3, en donde se tiene que no existe una distancia consistente entre los diferentes cromosomas que conforman un gen en particular, concentrándose estos en los bordes.

$$Fitness = \frac{1}{1 + \beta E + \alpha V} \quad (5)$$

2.2.3. Selección de los individuos

Es necesario obtener una cantidad de pares de individuos igual a (cantidad de individuos de la generación)/2 ya que estos son los que darán paso a la nueva generación en el crossover, para realizar esto se sigue el siguiente procedimiento:

Como una forma de asegurar que un % de los individuos con mejor fitness sean parte de los padres de la nueva generación es que se selecciona el 10 % de la población con el mejor fitness, o sea la elite de la generación, y se ordenan de a pares.

Con el 90 % restante se realiza un torneo, eso significa que se toman de a pares y se selecciona a quien tenga mejor fitness descartando al otro, luego se toman los individuos ganadores del torneo y se ordenan de a pares, ya que con esto faltaran parejas estas se obtienen a partir de las permutaciones faltantes de quienes ganen el torneo.

Finalmente los pares de padres de la elite de la generación y los pares obtenidos mediante el torneo se unen para posteriormente ser utilizados en el proceso del crossover.

2.2.4. Crossover y mutación

En este paso se toma cada pares de padres obtenidos en el paso anterior y se identifican como *Padre1* y un *Padre2*, la creación de cada par de hijos se hace según la ecuación (6), en donde la cantidad de genes de cada individuo se representa como h y W corresponde a la ecuación (4), en el fondo la ecuación (4) funciona como un peso que se multiplica por los cromosomas de un padre y el complemento del peso por el padre restante, consiguiendo de esta manera un hijo que aúna parte de los cromosomas de cada uno de los progenitores.

$$\begin{aligned} Hijo1(i, h) &= Padre1(i, h) * W(i, h) + Padre2(i, h) * [1 - W(i, h)] \\ Hijo2(i, h) &= Padre2(i, h) * W(i, h) + Padre1(i, h) * [1 - W(i, h)] \end{aligned} \quad (6)$$

Es aquí en donde es relevante la introducción de las mutaciones ya que permite introducir cambios a corto, mediano y largo plazo en las poblaciones, para esto se utiliza la ecuación (7), en donde d esta en el rango entre $[\min(Hijo(h, i)) - \max(Hijo(h, i)), \max(Hijo(h, i)) - \min(Hijo(h, i))]$, M es una función de mutación gaussiana que varía de 0 a 1 definida en la ecuación (8) cuyos μ , σ e i son los mismos que se utilizan en el proceso del crossover (ecuación (6)).

$$HijoMutado(h, i) = Hijo(h, i) + d * M(h, i) \quad (7)$$

$$M(h, i) = \exp \left(\frac{-(i - \mu)^2}{2\sigma^2} \right) \quad (8)$$

Algunas mutaciones pueden dar paso a situaciones en las cuales los genes muestran comportamientos que se salen demasiado de lo esperado, en este caso es necesario limitar estos comportamientos acotándolos inferior y superiormente llamándose esto método *Fixed Limit*.

Un ejemplo de todo lo anteriormente nombrado se puede observar en la figura 4, en donde en 4(a) se tiene el proceso de crossover y en 4(b) como un hijo cualquiera sufre una mutación y luego se le aplica el método *Fixed Limit*,

- (a) Cruza entre 2 individuos. (b) Regulación de la mutación mediante fixed limit.

Figura 4: Crossover y mutación.

3. Metodología

El desarrollo de la solución se crea en ambiente Matlab debido a la familiaridad con el lenguaje y la ductilidad que este entrega en la implementación de soluciones, donde uno de los subconjunto es implementado para conocer la ubicación en coordenadas cartesianas de cada muestra por medio de la matriz de Denavid Hartenberg. El uso de algoritmo genético, la creación de los primeros individuos, la ebulición, selección, crossover y mutación se presentaran con más detalles en el mismo capítulo.

Para visualizar los resultados se presentan 3 link dispuestos desde el origen que presentan las movimientos de 3 de los 4 engranajes. El último engranaje se encuentra en el origen y permite la rotación en el eje Z.

No se utiliza una base de datos ya que la población inicial se genera mediante la ecuación (4) en donde es particularmente importante la ecuación (3) la cual es una función aleatoria uniformemente distribuida entre 0 y 1, siendo esta misma la que interviene en el proceso de crossover.

El código (creado en su totalidad por los autores del informe) y las instrucciones para su ejecución esta disponible en <https://github.com/EliasObreque/EL4106> [6], los parámetros utilizados en la simulación están tabulados en la tabla 3 y el tiempo de simulación en un notebook con las características descritas en [7] es de aproximadamente 20 minutos. El código esta seteado para entregar los resultados de una semilla en particular la cual puede ser configurada mediante el parámetro rng de Matlab.

Tabla 3: Parámetros de la simulación.

Parámetro	Valor
<i>Generaciones</i>	80
<i>Individuos</i>	200
<i>Cromosomas</i>	100
<i>Genes</i>	4
<i>Probabilidad de mutación</i>	0.1
α	0.5
β	0.5

3.1. Selección de los hiperparámetros de fitness

Como se ha mencionado en la sección 2.2.2 es necesario definir el peso de α y β de tal manera de que sean lo suficientemente representativos del real estado de la generación, a modo de recordatorio lo que representa cada uno de ellos:

- α regula el *smoothness* de las curvas o sea la velocidad de traslación de las muestras o cromosomas.
- β regula el error de posición o sea la distancia cartesiana entre las coordenadas deseadas y las coordenadas a las cuales se llevo

El resultado de variar α y β con los mismos parámetros de la tabla 3 se puede observar en la figura 5, en este caso se considera una población con mutaciones. En 5(a) se tiene el error de posición para cada una de estas variaciones, queda claro que para alfas cada vez mas grandes el error de posición aumenta, en 5(b) refleja como el promedio del fitness evoluciona con cada generación, siendo particularmente alto para un β y un α cada vez mayor y menor respectivamente, en 5(c) se tiene que el mejor fitness de la generación sigue la misma tendencia.

No obstante y si se comparan estos resultados con la trayectoria que cada uno de estos casos tiene, obtenido mediante el método DH, y que se observa en la figura 6 se tiene que las combinatorias de α y β que en la figura 5 obtuvieron los mejores resultados así como los que entregaron los peores, son los que muestran trayectorias mas erráticas.

Para evitar estos casos extremos es que se decidió que los hiperparametros α y β fuesen de 0.5 cada uno, esperando que de esta manera los resultados mostrasen un comportamiento mas homogéneo.

(a) Error de posición

(b) Fitness promedio.

(c) Mejor fitness.

Figura 5: Variación de los hiperparametros por generación.

(a) α : 0.1 y β : 0.9

(b) α : 0.2 y β : 0.8

(c) α : 0.3 y β : 0.7

(d) α : 0.4 y β : 0.6

(e) α : 0.5 y β : 0.5

(f) α : 0.6 y β : 0.4

(g) α : 0.7 y β : 0.3

(h) α : 0.8 y β : 0.2

(i) α : 0.9 y β : 0.1

Figura 6: Trayectoria del brazo con el mejor fitness para combinaciones de α y β .

(a) Fitness promedio.

(b) Mejor fitness.

(c) Error de posición.

Figura 7: Evolución de los resultados con y sin mutaciones para un α : 0.5 y β : 0.5.

4. Resultados

Cuando se realizaron experimentos preliminares en donde solo se penalizaba la distancia del brazo al punto objetivo, los resultados mostraban un comportamiento similar al de la figura 3 en donde la ubicación geométrica de cada muestra se encontraba muy separada, es decir, su velocidad era muy grande. Esto se soluciona al incorporar una penalización en la velocidad y creando un nuevo fitness como se muestra en la ecuación (5). La selección de los hiperparámetros α y β del fitness, se ajustan experimentalmente logrando los resultados de las figuras 5 y 6. Se puede notar que si el valor de α es pequeño, el fitness es cada vez mejor y la distancia al punto objetivo es cada vez más pequeña. Sin embargo, dejar con muy poca penalización la velocidad conlleva problemas relacionados a la separación de las muestras, que al ser en exceso requerirá de mayor torque en los motores actuadores, por otro lado, si α es pequeño el movimiento no es tan suave y nacen curvas bruscas que cambian de una muestra a otra como se ve en la figura 6(a), (b) y (c). Por otro lado, si β es pequeño (penalizando de menor manera la posición final), se puede ver en la figura 6(i) que la punta del brazo no es capaz de llegar al objetivo y el fitness promedio es menor. De esta manera se selecciono valores medio igual a 0.5, para evitar los problemas de ambos extremos.

Luego de haber seleccionados los mejores individuos y generar el crossover, el sistema realiza la pregunta lógica de si crear o no la mutación. Cuando el sistema no realiza mutación se presenta el conocido problema de estancamiento en los resultados como muestra la figura 7. Esto cambia cuando el sistema tiene la opción de aplicar muta-

(a) Mejor trayectoria con mutación.

(b) Peor trayectoria con mutación.

(c) Mejor trayectoria sin mutación.

(d) Peor trayectoria sin mutación.

Figura 8: Comparación entre mejores y peores trayectorias para simulaciones con y sin mutación con un α : 0.5 y β : 0.5.

ción, donde, a pesar que el mejor fitness de las generaciones no varia mucho (figura 7(b)) en comparación al no usar mutación, si se notaron cambios importantes en la distancia que tiene la punta del brazo con el objetivo. Sin embargo, cabe mencionar que si el valor del porcentaje de mutación es muy alto, el sistema se comporta de manera errática, por lo tanto se recomienda utilizar valores pequeños de mutación no mayores al 10 %.

Finalmente, respecto a los resultados finales se lograron los objetivos mencionados, logrando que el brazo llegara a su objetivo con bajas velocidades. Esto se puede ver en la Figura 8 donde se muestran los cambios generados durante las generaciones. En la figura 8(b) y (d) se puede ver el contraste en usar o no la mutación, ya que cuando se analiza el peor caso sin mutación, el resultado no es tan malo ni tan alejado del mejor. Por otro lado, si se usa mutación, cambia completamente el peor escenario, desfavoreciendo incluso la distancia final del brazo hacia el objetivo. Por el contrario, si se analiza el mejor caso de las generaciones, se ve claramente la ventaja de usar mutación mostrando en la figura 8(a) que llega casi completamente al objetivo mientras que en la figura 8(c) no logra llegar completamente.

5. Conclusión

Como repuesta general del sistema ante las curvas definidas en la evolución de la trayectoria, se logro sintetizar un resultado en el que cada individuo iba mejorando su alcance geométrico hasta el punto objetivo evitando que las muestras se separen demasiado. Por otro lado, se tiene que:

- El método para crear los individuos (en este caso: tanh) es crítico para posicionar las muestras de manera consistente. Se debe estudiar en detalle como los crea y como genera la dispersión de los puntos antes de aplicar directamente. Uno de los inconvenientes era que al dejar todo la creación de puntos al azar, el punto inicial se reiniciaba y se convertía en un valor aleatorio, fenómeno que no debiera ocurrir. Esto se soluciono desplazando el punto inicial creado en la primera generación hasta su valor inicial definido por el problema.
- Los resultados van mejorando a medida que el fitness va considerando más variables estado en la penalización, por ejemplo, la posición y velocidad. Esto hace que la curva mejor llegue al objetivo y con la menor velocidad posible.
- Cuando la evolución de los dato no posee mutación, como es de esperar, se estabiliza en un punto del fitness y de la posición objetivo. Aplicar un porcentaje de mutación minimiza el error de posición final. Sin embargo, para valores muy alto de porcentaje de mutación la solución tiende a oscilar y/o error.
- Si bien la solución final muestra curvas bruscas en la trayectoria, la solución cumple con el objetivo y con el fitness propuesto, llegando al punto final y evitando que la separación entre cada punto sea excesiva, es decir, se disminuye la velocidad.
- Cuando se analiza el comportamiento del brazo para diferentes valores de α y β se pueden notar leves cambios en la trayectoria impactando principalmente en la separación de las muestras. La llegada del brazo al punto final tiene ciertos márgenes de error aún para altos valores de α .

Referencias

- [1] J. Denavit and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” *Trans. ASME E, Journal of Applied Mechanics*, vol. 22, pp. 215–221, June 1955.
- [2] A. Ghodousian and A. Babalhavaeji, “An efficient genetic algorithm for solving nonlinear optimization problems defined with fuzzy relational equations and max-Lukasiewicz composition,” *Applied Soft Computing*, vol. 69, pp. 475 – 492, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494618302205>
- [3] “Robust Control Systems with Genetic Algorithms.” [Online]. Available: <https://www.crcpress.com/Robust-Control-Systems-with-Genetic-Algorithms/Jamshidi-Krohling-Coelho-Fleming/p/book/9780367395728>
- [4] J. J. V. González, C. A. P. Caro, and H. V. González, “Cinemática inversa de robot serial utilizando algoritmo genético basado en MCDS,” *Tecnura*, vol. 19, no. 44, pp. 33–46, Apr. 2015. [Online]. Available: <https://revistas.udistrital.edu.co/index.php/Tecnura/article/view/8351>
- [5] S. Momani, Z. Abo-Hammour, and P.-O. Alsmadi, “Solution of inverse kinematics problem using genetic algorithms,” *Applied Mathematics and Information Sciences*, vol. 10, pp. 1–9, 01 2015.
- [6] E. Obreque and E. Peña, “El4106,” Dec 2019. [Online]. Available: <https://github.com/EliasObreque/EL4106>
- [7] “Gp62 7rd leopard: Laptops - the best gaming laptop provider: Msi españa,” *GP62 7RD Leopard — Laptops - The best gaming laptop provider — MSI España*, Dec 2019. [Online]. Available: <https://es.msi.com/Laptop/GP62-7RD-Leopard>