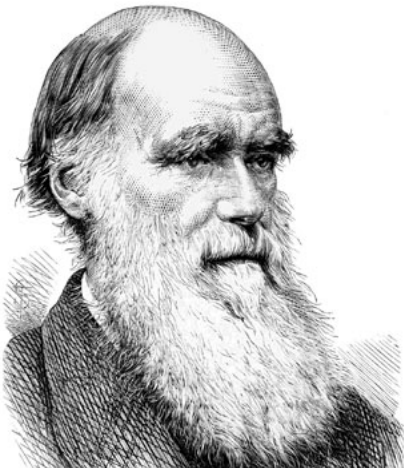


EL4106 INTELIGENCIA COMPUTACIONAL

Cinemática Inversa de un Manipulador con Algoritmos Genéticos

Informe Final



"No es la más fuerte de las especies la que sobrevive y tampoco la más inteligente. Sobrevive aquella que más se adapta al cambio."

Charles Darwin

Profesores: Pablo Estévez
Pablo Huijse

Auxiliar: Pablo Huentelenu

Ayudantes: Rodrigo Carrasco
Ignacio Reyes

Integrantes: Sergio Ehijo
Nicolás Tapia

Fecha: 29 de Diciembre, 2016

Índice General

1. Introducción	1
2. Marco Teórico	2
2.1. Cinemática de un Manipulador	2
2.2. Algoritmos Genéticos (GA)	3
2.3. Algoritmos Genéticos Continuos (CGA)	4
2.4. Correntropía	6
2.4.1. Motivación	6
2.4.2. Definición Correntropía	7
2.5. Energías utilizadas	8
3. Metodología	9
3.1. Planteamiento del Problema	9
3.2. Algoritmos Genéticos a Implementar	10
3.2.1. CGA	10
3.2.2. GA Convencional	10
4. Resultados	13
4.1. Resultados generales	13
4.2. Variación de parámetros	15
5. Análisis	19
6. Conclusiones	23
Referencias	24
Anexos	25
A1. Código de la prueba preliminar	25
A2. Modelo dinámico del brazo robótico	25

Índice de Figuras

2.1. Manipulador robótico de 4 DOF.	2
2.2. Contorno en espacio 2-D con un sigma unitario.	7
4.1. Posición inicial del manipulador robótico. Los ángulos de la configuración de inicio son $(\theta_1, \theta_2, \theta_3, \theta_4) = (90, -30, 30, -20)$	13
4.2. Trayectoria de los cuatro ángulos y posición de los tres brazos planares alcanzando la posición final para el caso de correntropia en ambos fitness.	14
4.3. Trayectoria de los cuatro ángulos y posición de los tres brazos planares alcanzando la posición final para el caso de energía simple.	14
4.4. Trayectoria de los cuatro ángulos y posición de los tres brazos planares alcanzando la posición final para el caso de torque RMS.	15

1. Introducción

Un manipulador robótico es un mecanismo altamente utilizado para poder manejar materiales en la industria de manufactura, automotriz y aplicaciones médicas [1], el cual está conformado por distintos actuadores, que pueden ser rotacionales o telescópicos, que se enlazan entre sí de manera muy similar a un brazo humano. Este manipulador termina en un efector, que se puede ver como una mano, que generalmente está provista de una herramienta para realizar una tarea deseada. Un problema puede variar en los grados de libertad que posea en función de la cantidad de actuadores que se vayan a usar.

Existen dos tipos de problemas interesantes a abordar con respecto a un manipulador robótico. El primero es que dado un arreglo de ángulos o mejor dicho *vector de rotaciones* aplicado a un manipulador, su efector se va a dirigir a cierta posición en el espacio cartesiano; este es el problema de **Cinemática Directa**. Por otro lado, dada una cierta posición del efector en el espacio cartesiano, se desea encontrar un set de ángulos en el espacio de parámetros; este problema se conoce como **Cinemática Inversa** el cual es altamente no lineal. Este último es un problema de optimización complejo dado que los actuadores pueden recorrer múltiples caminos para llegar al mismo punto, por ende se han intentado ocupar distintos criterios de maximización o minimización, entre los que destacan minimizar el torque, minimizar el tiempo de movimiento, minimizar el *jerk*-que es el cambio de la aceleración en el tiempo-, entre otros expuestos por [2].

Así, se llega a la descripción del problema a abordar en este proyecto, el cual posee los siguientes objetivos:

- Se necesita resolver la cinemática inversa de un manipulador de cuatro grados de libertad para mover el efector a una cierta posición final en el espacio.
 - La trayectoria debe **minimizar la energía consumida**.
 - Debido a la complejidad, se utilizarán **Algoritmos Genéticos -GA-** para resolver.

2. Marco Teórico

2.1. Cinemática de un Manipulador

En la Figura 2.1 se puede observar un manipulador robótico de 4 grados de libertad (*degrees of freedom* o simplemente DOF). Este modelo de manipulador es el que se utilizará en el presente proyecto. Posee una primera junta caracterizada por el ángulo θ_1 cuya función es rotar el resto del brazo en forma polar dentro del plano horizontal. Las juntas restantes caracterizadas por θ_2 , θ_3 y θ_4 tienen como función mover el brazo en forma radial y en altura.

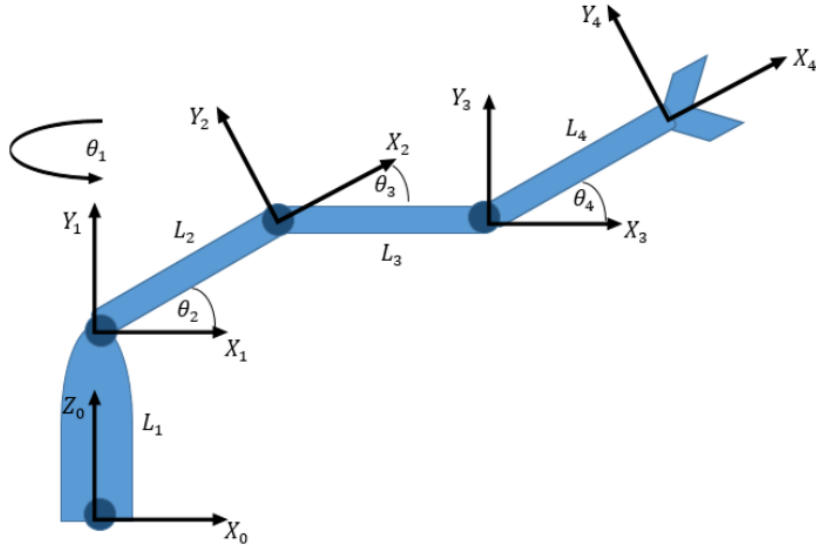


Figura 2.1: Manipulador robótico de 4 DOF.

El origen de θ_1 es arbitrario y de poca importancia para el planteamiento de la cinemática del manipulador; no así el origen de los siguientes tres. Estos últimos tres ángulos son medidos de la siguiente forma:

- θ_2 es el ángulo que forma el segmento de largo L_2 con respecto a la horizontal.
- θ_3 es el ángulo que forma el segmento de largo L_3 con respecto al segmento de largo L_2 .
- θ_4 es el ángulo que forma el segmento de largo L_4 con respecto al segmento de largo L_3 .

De esta forma, los ángulos de cada junta son precisamente los que ven los motores actuadores ubicados en ellas. En todos los ángulos se respeta la convención del sentido positivo de la medición y el origen del sistema cartesiano se ubica en la junta entre el segmento de largo L_1 y el de largo L_2 . Esto último ya que el primer segmento es fijo por ser la base y solo introduce traslación constante en el origen. La configuración se realiza entonces a través del vector $\theta = (\theta_i)_{i=1}^4$ y el efector del manipulador es capaz de alcanzar una posición objetivo $(x(\theta), y(\theta), z(\theta))$ contenida en una esfera de radio $R = L_2 + L_3 + L_4$.

El modelo que permite determinar la posición final del efector (x, y, z) con respecto a la base del manipulador dado un determinado vector θ es el modelo de **cinemática directa**. Para encontrar este modelo conviene utilizar la convención de *Denavit-Hartenberg* para manipuladores robóticos,

la cual es explicada en [3]. Este método determina una matriz de transformación homogénea con la estructura (1), en donde O_4^0 es la posición del origen del sistema de coordenadas solidario al efector y R_4^0 es la matriz de rotación que indica la orientación de tal sistema con respecto a la base. Mandava y Vundavalli [1] determinaron la matriz de transformación homogénea (2) para el caso del manipulador que considera este proyecto, cuyos coeficientes están en (3).

$${}^0T_4 = \begin{pmatrix} R_4^0 & O_4^0 \\ 0 & 1 \end{pmatrix} \quad (1)$$

$${}^0T_4 = \begin{pmatrix} c_1 c_{234} & -c_1 s_{234} & s_1 & c_1(L_2 c_2 + L_3 c_{23} + L_4 c_{234}) \\ s_1 c_{234} & -s_1 s_{234} & -c_1 & s_1(L_2 c_2 + L_3 c_{23} + L_4 c_{234}) \\ s_{234} & c_{234} & 0 & (L_2 s_2 + L_3 s_{23} + L_4 s_{234}) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$\begin{aligned} c_i &= \cos(\theta_i) \\ s_i &= \sin(\theta_i) \\ c_{ij} &= \cos(\theta_i + \theta_j) \\ s_{ij} &= \sin(\theta_i + \theta_j) \\ c_{ijk} &= \cos(\theta_i + \theta_j + \theta_k) \\ s_{ijk} &= \sin(\theta_i + \theta_j + \theta_k) \end{aligned} \quad (3)$$

Por simplicidad el presente trabajo solo considera la posición del efector, sin su orientación. Es decir, solo se considera O_4^0 , lo que equivale a limitar el modelo de cinemática directa al sistema de ecuaciones (4). El problema de invertir el sistema, i.e. de encontrar los ángulos dada una posición cartesiana del efector es el de **cinemática inversa**, y se puede apreciar que es altamente no lineal.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} c_1(L_2 c_2 + L_3 c_{23} + L_4 c_{234}) \\ s_1(L_2 c_2 + L_3 c_{23} + L_4 c_{234}) \\ L_2 s_2 + L_3 s_{23} + L_4 s_{234} \end{pmatrix} \quad (4)$$

2.2. Algoritmos Genéticos (GA)

Todos los seres vivos que habitan actualmente en nuestra tierra son el producto de un largo e histórico proceso evolutivo. A lo largo de distintas generaciones, los individuos que más se adaptan al cambio son capaces de pasar sus genes a la próxima generación. Así, los **Algoritmos Genéticos** son un método de optimización inspirado en la *Selección Natural* propuesta por Darwin, la cual se puede resumir en tres procesos importantes según [4]:

- Herencia: Debe haber un proceso por el cual los hijos reciben las propiedades de sus padres
- Varianza: Debe haber una variedad de rasgos presentes en la población o una manera por la cual introducir varianza.
- Selección: Debe haber un mecanismo por el cual algunos miembros de la población tengan la oportunidad de ser padres y pasar su información genética. Esto es típicamente conocido como "la sobrevivencia del más apto".

A partir de un modelo, se tiene una población de soluciones al problema a resolver la cual se suele llamar **genotipo**, que es generada al azar, y a cada solución se le llama **individuo**. Cada individuo pasa por distintos procesos para evolucionar a un óptimo, según el criterio de una función de **fitness**. Según [5], se procede a explicar paso a paso el GA:

1. Inicialización: Se crea una población al azar tratando de que sea lo más grande y diversa para poder obtener mayor variabilidad en futuras generaciones.
2. Evaluación: La función de fitness hace de cierta manera un juicio de cada individuo de la población viendo cuál se parece más al modelo que se quiere llegar. Cada individuo pasa por la función de fitness y obtiene como resultado un número, conocido como **fitness**.
3. Selección: Existen distintas formas de ver cuáles individuos tienen los mejores fitness, las cuales son explicadas a continuación:
 - Selección por rango: los individuos se ordenan según su fitness y se les asigna una probabilidad de reproducirse según su posición.
 - Selección proporcional: cuando un individuo posea un fitness pequeño, la probabilidad con la que pueda reproducirse será menor a la de un individuo que posea un fitness mayor.
 - Selección por tournament: se eligen varios individuos al azar, usualmente tres o cuatro, y se compara su fitness. El individuo que posea mayor fitness es el que pasa a la siguiente generación. Es muy útil para generar mayor variabilidad genética de los individuos de la siguiente generación.
4. Crossover: Basado en la reproducción biológica, el método de *crossover* se basa en mezclar el material genético de los padres para crear un nuevo individuo. Uno de los métodos más básicos se llama *One-Point Crossover*, el cual consiste en elegir un punto de manera aleatoria de los dos padres e intercambiar la información genética a partir de esa posición. Si el algoritmo se detuviese aquí, habría un grave problema. Si solo se intercambia la información genética dada, existen casos en que nunca se llegaría al modelo pues sólo se está permutando información, nunca **cambiando**. Es por esto que falta un paso importante, la mutación.
5. Mutación: Como se mencionaba, para que exista un cambio en el individuo de una generación a otra, es necesario que exista una etapa de mutación, la cual consiste en añadir pequeñas variaciones al azar del individuo, logrando llegar al modelo.

Por lo general, la población completa es reemplazada por la nueva generación, comenzando nuevamente el proceso de evaluación. Algunas variantes hacen sobrevivir al mejor individuo de la generación (la *elite*) para asegurar que el fitness nunca decrezca.

2.3. Algoritmos Genéticos Continuos (CGA)

Tal como lo demuestran las comparaciones hechas por Abo-Hammour [2], los algoritmos convencionales de GA operan de manera puntual en los genes, y por lo tanto no son adecuados para preservar la suavidad de los genes si es que ellos son utilizados para representar un muestreo de una función suave. A pesar de que son capaces de solucionar el problema en cuanto a seguir la trayectoria deseada del efector, la presencia de redundancias en el manipulador hace que en general las soluciones en el espacio θ sean alternantes entre la gran cantidad de soluciones disponibles,

provocando oscilaciones innecesarias que conllevan a desperdicio de energía y/o soluciones infactibles para los actuadores. Abo-Hammour [6] propone un algoritmo genético denominado **Algoritmo Genético Continuo (CGA)** cuyos operadores actúan de una manera global en las muestras contenidas en el genotipo de forma de evitar estos problemas y preservar la suavidad en todo momento.

Para explicar el algoritmo, en primer lugar se define al genotipo P del problema del manipulador como (5), en donde la fila h representa el muestreo de la trayectoria a seguir por el ángulo θ_h , para $h \in \{1, 2, 3, 4\}$, considerando N_s muestras. Por lo tanto, los operadores actúan sobre una fila completa, de manera independiente entre filas.

$$P_{hi} = \theta_h(i) = \begin{pmatrix} \theta_1(1) & \cdots & \theta_1(i) & \cdots & \theta_1(N_s) \\ \vdots & & \ddots & & \vdots \\ \theta_4(1) & \cdots & \theta_4(i) & \cdots & \theta_4(N_s) \end{pmatrix} \quad (5)$$

La **inicialización** del algoritmo es con una población de individuos suaves generados al azar. Es decir, cada individuo P^j inicializa cada una de sus filas como un muestreo de alguna función suave y se proponen dos alternativas; una función gaussiana modificada (6) y una tangente hiperbólica (7).

$$P_{hi}^j = \theta_h^j(1) + \frac{\theta_h^j(N_s) - \theta_h^j(1)}{N_s - 1}(i - 1) + A \exp\left(-\frac{(i - \mu)^2}{2\sigma^2}\right) \quad (6)$$

$$P_{hi}^j = \theta_h^j(1) + \frac{\theta_h^j(N_s) - \theta_h^j(1)}{2} \left(1 + \tanh\left(\frac{i - \mu}{2\sigma^2}\right)\right) \quad (7)$$

En donde A , μ y σ son números generados al azar para cada fila h con:

- $A \in [-3|\theta_h^j(N_s) - \theta_h^j(1)|, 3|\theta_h^j(N_s) - \theta_h^j(1)|]$.
- $\mu \in [2, N_s]$.
- $\sigma \in [1, N_s/6]$.

Puesto que los ángulos finales $\theta_h(N_s)$ son desconocidos, son generados al azar dentro del rango factible de los actuadores.

Luego de realizar la selección de los padres de la siguiente generación de manera proporcional al *fitness* definido para el problema en cuestión, el **crossover** entre dos padres se realiza a través de una tangente hiperbólica que los combina de forma tal que las filas de los hijos siguen siendo funciones suaves, con una transición completa entre aquella fila de un padre y la del otro. El crossover ocurre entre dos individuos con probabilidad p_{cj} , y entre estos dos individuos cada una de sus filas sufre crossover con probabilidad p_{ch} . En concreto, el crossover entre P^j y P^k produce los hijos C^L y C^{L+1} según (8) y (9), donde W está dado para cada fila h por (10) con μ y σ generados de la misma forma que antes.

$$C_{hi}^L = W_{hi}P_{hi}^j + (1 - W_{hi})P_{hi}^k \quad (8)$$

$$C_{hi}^{L+1} = (1 - W_{hi})P_{hi}^j + W_{hi}P_{hi}^k \quad (9)$$

$$W_{hi} = \frac{1}{2} \left(1 + \tanh\left(\frac{i - \mu}{2\sigma^2}\right)\right) \quad (10)$$

Al finalizar la etapa de crossover, se procede a la etapa de mutación, en donde cada uno de los hijos muta con probabilidad p_{mj} y cada una de sus filas puede mutar con probabilidad p_{mh} . El operador de mutación, de manera similar al caso de crossover, altera una fila agregando una función gaussiana de forma tal que los mutantes siguen representando funciones suaves. En concreto, el hijo C^j muta en \bar{C}^j de acuerdo con (11), donde α es generado aleatoriamente para cada fila según (12) y donde M está dado para cada fila h por (13) con μ y σ generados igual que antes.

$$\bar{C}_{hi}^j = C_{hi}^j + \alpha M_{hi} \quad (11)$$

$$\alpha \in [-\text{range}(h), \text{range}(h)]; \text{range}(h) = \max_{\theta_h^j \in C^j} \{\theta_h^j(i)\} - \min_{\theta_h^j \in C^j} \{\theta_h^j(i)\} \quad (12)$$

$$M_{hi} = \exp\left(-\frac{(i - \mu)^2}{2\sigma^2}\right) \quad (13)$$

Finalmente, la población completa es reemplazada por la nueva generación, comenzando nuevamente el proceso de evaluación. El algoritmo termina cuando se alcanza algún criterio de convergencia definido por el programador, el cual puede ser alcanzar un fitness por sobre un valor objetivo; alcanzar un máximo número de generaciones; o alcanzar un límite del progreso (una mejora del fitness menor a un valor objetivo a lo largo de algún número de generaciones).

2.4. Correntropia

2.4.1. Motivación

Liu et al [7] muestran que esta herramienta está directamente relacionada con la probabilidad de cuan similares son dos variables aleatorias en una vecindad, la cual es controlada por un parámetro σ , el cual funciona como lente, controlando la llamada ventana de observación en donde se asigna esta similitud. Esta ventana ajustable provee de un mecanismo efectivo para eliminar el efecto de los *outliers*, y es distinto al uso de un *threshold* en las técnicas convencionales.

Es normal esperar un significado geométrico de un estadístico, por ejemplo el **MSE** da la norma-2 en el espacio. Se tiene como resultado que la correntropia induce una nueva métrica que va dependiendo de la distancia de los puntos a evaluar; es equivalente a la norma-2 para los puntos cercanos, mientras que se comporta similar a la norma-1 a medida que los puntos se van alejando, hasta que se va aproximando a la norma cero cuando estos puntos se encuentran muy lejos. Todo esto se puede ver reflejado en la figura 2.2 la cual va demostrando como varía la norma a medida que los datos se alejan, usando un tamaño de kernel igual a la unidad.

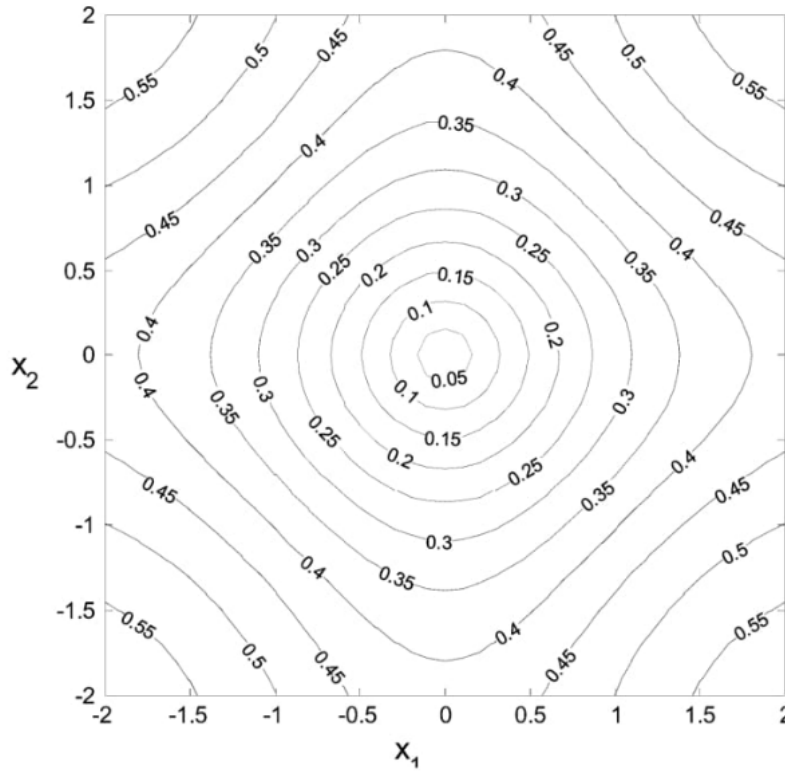


Figura 2.2: Contorno en espacio 2-D con un sigma unitario.

Ya se ha mencionado en un par de ocasiones este nuevo parámetro, y para ello es necesario recordar la definición del *kernel gaussiano*, el cual se ve definido en (14)

$$k_{\sigma}(X - x_i) = \frac{1}{\sqrt{2 * \pi} * \sigma} * \exp\left(-\frac{(X - x_i)^2}{2\sigma^2}\right) \quad (14)$$

en donde σ representa el tamaño del kernel, el cual es un parámetro libre que debe ser elegido de manera rigurosa a través de estimaciones de densidad, como la regla de Silverman o bien máxima verosimilitud. Ya hecha la motivación para esta nueva medida de estadística, se procede a dar su definición formal:

2.4.2. Definición Correntropía

La correntropía es una medida generalizada de similitud entre dos variables escalares aleatorias X e Y definida por 15.

$$V(X, Y)_{\sigma} = \mathbb{E}(\kappa_{\sigma}(X - Y)) \approx \frac{1}{N} \sum_{k=1}^N \kappa_{\sigma}(X_i - Y_i) \quad (15)$$

En la práctica, la función de distribución conjunta es desconocida y hay disponibles solo un número finito de datos, haciendo que se use más el estimador de la correntropía el cual se define en 16

$$V(X, Y)_{N, \sigma} \approx \frac{1}{N} \sum_{k=1}^N \kappa_{\sigma}(X_i - Y_i) \quad (16)$$

2.5. Energías utilizadas

En la ecuación (17) se puede ver el modelo usado de torque con sus distintos componentes, el cual se basa en lo documentado en [8] cuya deducción se basa en la mecánica lagrangiana (mayores detalles de su expresión en el Anexo 2).

$$\tau_i = \sum_{k=1}^4 M_{ik} \ddot{q}_k + \sum_{k=1}^4 \sum_{m=1}^4 h_{ikm} \dot{q}_k \dot{q}_m + g_i ; \quad i = 1, 2, 3, 4 \quad (17)$$

Por otro lado, sigue que para ocupar una relación entre el torque con la energía, se uso el *Torque RMS* mostrado en la ecuación (18).

$$\tau_{RMS} = \sqrt{\sum_{i=1}^4 \frac{1}{T} \int_0^T \tau_i^2(t) dt} \approx \sqrt{\sum_{i=1}^4 \sum_{k=1}^N \frac{1}{N} \tau_i^2(k)} \quad (18)$$

Para empezar un ajuste fino de parámetros, se vio otra manera de usar la energía asumiendo que toda la masa se acumula al final de cada brazo. Esto se puede ver en la ecuación (19)

$$K_i = \frac{1}{2} m v_i^2 \quad (19)$$

Finalmente la energía simple, vista en la ecuación (20) usada para algunos fitness es proporcional a los cambios en el tiempo de la energía de cada brazo. Los supuestos que hay que realizar es que poseen la misma densidad, por lo que se utiliza una expresión proporcional en la que en vez de masa se utiliza el largo de cada brazo. Además, se ignora cualquier otra fuente energética tales como roces y gravedad.

$$f_2 \propto \sum_{k=1}^N |\Delta K_1(k)| + |\Delta K_2(k)| + |\Delta K_3(k)| + |\Delta K_4(k)| \quad (20)$$

3. Metodología

3.1. Planteamiento del Problema

El problema de la planificación de la trayectoria que debe seguir el manipulador para alcanzar la posición objetivo final se plantea como un problema de optimización, en donde se desea minimizar el error entre la posición final deseada y la posición final resultante; además de minimizar la energía utilizada por el manipulador en el movimiento. El problema es de alta dimensionalidad, ya que posee 4 ángulos en donde cada uno debe describir una trayectoria en el espacio θ , por lo que de alguna manera se debe mantener un tracking de la forma de las trayectorias en el problema de optimización. Las trayectorias deben ser suaves para no desperdiciar energía en soluciones alternantes, por lo que esto también debe ser una restricción del problema. Además, si bien el modelo de cinemática directa es sencillo, el modelo inverso no lo es debido a la no-linealidad del sistema, por lo que en principio no se cuenta con derivadas en una forma cerrada.

Por estas razones en este trabajo se propone utilizar algoritmos genéticos para resolver el problema de optimización, en donde se prueban distintas funciones de fitness las cuales deben minimizar el uso de la energía para que el efector llegue a la posición deseada y que además permitan preservar la suavidad de la soluciones en θ .

El modelamiento realizado del brazo robótico, tanto en su cinemática como en su dinámica (torque y energía simplificada) se encapsula a través de métodos de una clase que representa al brazo, llamada *ThePablos* implementada en Python. Así, se puede fijar la trayectoria, solicitar las posiciones cartesianas, el punto final y las expresiones energéticas de forma más limpia dentro de los algoritmos genéticos. Estos algoritmos son implementados utilizando la librería DEAP [9], definiendo operadores cuando sea necesario. Todos los códigos están disponibles para su consulta en el link de GitHub mostrado en el Anexo 1.

Las funciones de fitness, como se mencionó, deben tener en cuenta cuánto se desvía la posición final del efector respecto a la posición que se requiere, y debe tener en cuenta la energía utilizada en el movimiento del efector desde su posición inicial a su posición final. En concreto, un mejor fitness implica una desviación menor y un gasto energético menor. Los distintos fitness que se prueban se diferencian en cómo optimizar el error de posición final y generada por el algoritmo, usando los estadísticos de error cuadrático medio y la correntropía. Además, se define un λ que va a definir cuánta importancia se le da a la energía dentro del mismo fitness, y se hacen variaciones de este parámetro que son mostrados en la sección de Resultados. Así, los fitness que se proponen tienen la estructura de una combinación convexa entre el fitness de error (ecuación que use la posición) y el fitness de la energía como se muestra en (21) para cada individuo P^j .

$$f^j = (1 - \lambda) \cdot f_{\text{error}}^j(\vec{x}^j, \vec{x}_f) + \lambda \cdot f_{\text{energía}}^j; \quad \lambda \in (0, 1) \quad (21)$$

Los diferentes fitness que se proponen son los siguientes. En primer lugar, se utiliza el MSE para la posición final y la expresión de la energía directamente, tal como se muestra en (22), los cuales deben minimizarse. En segundo lugar, se utilizan los inversos de tales expresiones, de manera que se tengan que maximizar y también obtener expresiones que se mueven entre 0 y 1, tal como se muestra en (23). Por último, se utilizan expresiones que siguen la idea de la correntropía, tanto para el error como para la energía, comparando a la energía con 0, tal como se muestra en (24), los cuales deben maximizarse. Aquí se ha utilizado la notación $Energy(\theta)$ para indicar la expresión energética de la trayectoria determinada por el vector θ , la cual puede ser tanto la versión simplificada como la versión extraída con la mecánica lagrangiana.

$$f_1 = \|r_{real} - r_{generado}(\theta)\|^2 \quad f_2 = Energy(\theta) \quad (22)$$

$$f_1 = \frac{1}{1 + \|r_{real} - r_{generado}(\theta)\|^2} \quad f_2 = \frac{1}{1 + Energy(\theta)} \quad (23)$$

$$f_1 = \exp\left(-\frac{\|r_{real} - r_{generado}(\theta)\|^2}{2\sigma_1^2}\right) \quad f_2 = \exp\left(-\frac{Energy(\theta)^2}{2\sigma_2^2}\right) \quad (24)$$

3.2. Algoritmos Genéticos a Implementar

3.2.1. CGA

Según se vio en el marco teórico, con CGA se pueden obtener soluciones suaves para las trayectorias de cada ángulo, introduciendo un genotipo que contenga un muestreo directo de todas ellas como el mostrado en (25), por lo que se propone utilizarlo. El número de muestras N_s por cada trayectoria debe ser suficiente para que la aproximación de la función sea lo suficientemente suave. Se usa una selección por torneo de tres individuos, se propone que el mejor de la población pueda sobrevivir y se propone un método para mantener las curvas luego de realizar las mutaciones dentro de los rangos estipulados, en donde si al mutar un ángulo se sobrepasa del límite superior, entonces este ángulo se satura y queda en el límite; lo mismo para el caso de *underflow* en que si al mutar un ángulo se pasa de la cota inferior, entonces se satura con este límite inferior, generando con esto que las curvas estén dentro de los rangos en los que se desea trabajar. Para poder generar la solución, se otorgan los ángulos iniciales y además el ángulo final de θ_1 es conocido automáticamente luego de saber la posición final, pues es el único ángulo que se mueve en dirección polar y es más fácil conocer su posición para lograr mover el efector a la posición deseada.

$$\theta_i(t) = \theta_i(k\Delta T) \Rightarrow P_j = \begin{pmatrix} \theta_1^j(1) & \dots & \theta_1^j(k) & \dots & \theta_1^j(N_s) \\ \vdots & & \ddots & & \vdots \\ \theta_4^j(1) & \dots & \theta_4^j(k) & \dots & \theta_4^j(N_s) \end{pmatrix} \quad (25)$$

Puesto que en la práctica las trayectorias angulares no puedes tomar valores arbitrarios debido a las limitantes físicas del sistema, es necesario implementar un método que verifique a lo largo de la evolución que tal trayectoria se mantenga dentro de los rangos permitidos. Para el caso del CGA, cuando una porción de la trayectoria se pasa de un límite ya sea θ_{max} o θ_{min} , la corrección se realiza por saturación. Es decir, para la trayectoria angular h -ésima en el instante i -ésimo se cumple:

$$P_j(h, i) = \begin{cases} P_j(h, i) & \theta_{min}(h) \leq P_j(h, i) \leq \theta_{max}(h) \\ \theta_{max}(h) & P_j(h, i) > \theta_{max}(h) \\ \theta_{min}(h) & P_j(h, i) < \theta_{min}(h) \end{cases} \quad (26)$$

3.2.2. GA Convencional

Se propone como grupo además implementar un enfoque alternativo utilizando un GA convencional, en donde en el genotipo en lugar de codificar un muestreo directo de las trayectorias, se codifican coeficientes de su aproximación en serie de Taylor. Es decir, el ángulo θ_i^j sigue la evolución dada por un polinomio en el tiempo de $N_T + 1$ coeficientes (i.e. grado N_T) cuyo primer

coeficiente es conocido e igual al punto de partida de la trayectoria angular, y el genotipo del individuo P^j es el dado por (27) en donde se evoluciona sobre los coeficientes de los cuatro ángulos. El tiempo será el dado por la discretización en base a la cantidad de puntos N_s durante el tiempo dado al movimiento. Al realizar esta codificación, ya no es necesario mantener una suavidad entre genes puesto que la suavidad ya no se encuentra implementada a este nivel. Dado que los coeficientes de la serie de Taylor se pueden ajustar de manera puntual, se puede implementar esta solución con los operadores convencionales de la literatura. Algunos ajustes que se hacen a este algoritmo son realizar una selección por torneo de tres, los coeficientes $a_0 = \theta_i(0)$ son conocidos bajo el mismo argumento dado anteriormente para CGA y también se cumple que el mejor de la población sobrevive entre una generación y otra. Además, el operador de mutación es uno Gaussiano centrado en 0 y con un σ_M que controla su dispersión; y el operador de crossover es el Simulated Binary (SBX) [10], el cual imita el crossover one-point de un GA binario a través de la selección de los hijos dentro de una distribución de probabilidad centrada en los padres. El parámetro η de este crossover se fija en 2 según la literatura.

$$\theta_i(t) = \theta_i(0) + \sum_{m=1}^{N_T} a_{im} t^m \Rightarrow P_j = \begin{pmatrix} a_{11}^j & \cdots & a_{1N_T}^j & a_{21}^j & \cdots & a_{2N_T}^j & \cdots & a_{4N_T}^j \end{pmatrix} \quad (27)$$

Bajo los mismos argumentos anteriores, es necesario cuidar los rangos angulares en que se mueven las trayectorias. Para el caso del GA polinomial no se tiene la trayectoria disponible punto a punto, por lo que no es posible realizar la saturación descrita anteriormente. En su lugar, se cuenta con los coeficientes del polinomio, que en definitiva permiten operar con la función como un todo. Para la regularización se procede como sigue. Como la curva angular está descrita por:

$$\theta_h(i) = \theta_h(0) + \sum_{m=1}^{N_T} a_{hm} t_i^m \quad (28)$$

Y está limitada por arriba por θ_h^{max} y por debajo por θ_h^{min} , se tiene equivalentemente, al restar $\theta_h(0)$ de todas las variables, que:

$$\hat{\theta}_h(i) = \theta_h(i) - \theta_h(0) = \sum_{m=1}^{N_T} a_{hm} t_i^m, \quad \theta_h^{min} - \theta_h(0) \leq \hat{\theta}_h(i) \leq \theta_h^{max} - \theta_h(0) \quad (29)$$

Entonces, si es que algún $\hat{\theta}_h(i)$ se saliera del rango, basta con calcular:

$$\alpha_h(i) = \begin{cases} \frac{\theta_h^{max} - \theta_h(0)}{\hat{\theta}_h(i)} & \hat{\theta}_h(i) > \theta_h^{max} - \theta_h(0) \\ \frac{\theta_h^{min} - \theta_h(0)}{\hat{\theta}_h(i)} & \hat{\theta}_h(i) < \theta_h^{min} - \theta_h(0) \end{cases} \quad (30)$$

Y multiplicar a $\hat{\theta}_h(i)$ por $\alpha_h(i)$ para hacerlo igual al valor extremo del rango en la dirección en que lo supera. Si se revisa iterativamente cada punto de la trayectoria del ángulo h , entonces se tendrá un conjunto de $\alpha_h(i)$. Si se considera el menor de ellos, y se multiplica a toda la trayectoria por este valor, entonces se tendrá que $\hat{\theta}_h$ se encuentra dentro de sus rangos para todos los instantes de tiempo. De esta manera, basta con volver a sumar el ángulo en el tiempo 0 y se tendrá una trayectoria que comienza en el punto deseado y además se mantiene en los rangos definidos. En resumen, y considerando que si el ángulo no presenta problemas es equivalente a multiplicarlo por 1, el algoritmo para regularizar es:

$$\alpha_h(i) = \begin{cases} \frac{\theta_h^{max} - \theta_h(0)}{\theta_h(i) - \theta_h(0)} & \theta_h(i) > \theta_h^{max} \\ \frac{\theta_h^{min} - \theta_h(0)}{\theta_h(i) - \theta_h(0)} & \theta_h(i) < \theta_h^{min} \\ 1 & \theta_h^{min} \leq \theta_h(i) \leq \theta_h^{max} \end{cases} \quad (31)$$

$$\hat{\alpha}_h = \min_i \{\alpha_h(i)\} \Rightarrow a_{hm} \leftarrow \hat{\alpha}_h a_{hm} \quad (32)$$

4. Resultados

En esta sección se procede a mostrar los resultados obtenidos al probar distintos algoritmos genéticos en función de un fitness correspondiente, cuya función fuese lograr resolver el problema de cinemática inversa de un manipulador robótico de cuatro grados de libertad. La primera sección consiste en la exploración de resultados particulares viendo por encima un par de parámetros, mientras que la segunda parte consiste en un enfoque más detenido de parámetros importantes e interesantes.

4.1. Resultados generales

Para comenzar, es necesario dar la posición inicial en que estos resultados particulares inician con los algoritmos correspondientes, la cual se puede ver en la figura 4.1.

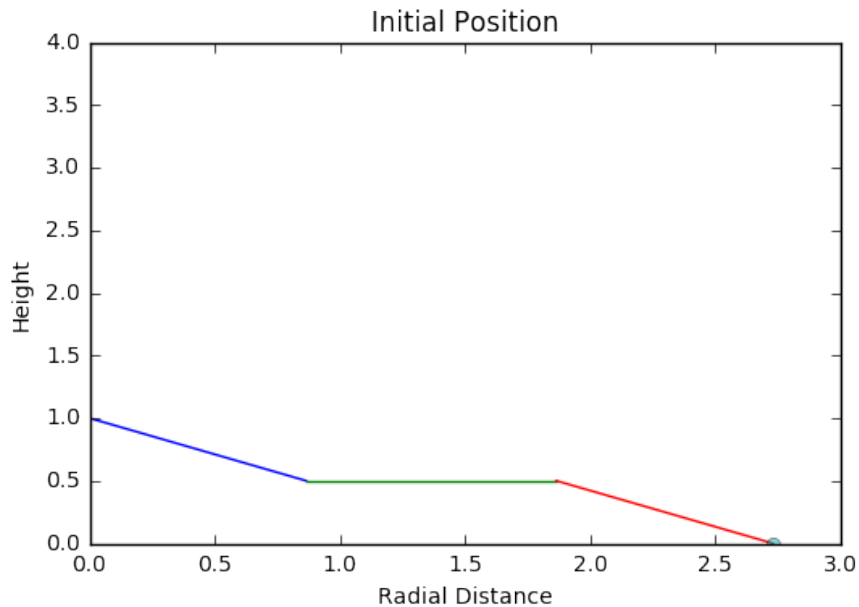


Figura 4.1: Posición inicial del manipulador robótico. Los ángulos de la configuración de inicio son $(\theta_1, \theta_2, \theta_3, \theta_4) = (90, -30, 30, -20)$.

Empezando por el algoritmo genético convencional polinomial, se proceden a mostrar tres resultados importantes. El primero, ocupa la correntropía de la energía simple con sigma igual a 10 y la correntropía del error con sigma igual a 300, en donde los parámetros usados son 20 posiciones en el espacio, cien generaciones, $\lambda = 0,01$, $\sigma_{mut} = 0,1$, $P_{mi} = 0,5$, $P_{mc} = 0,1$, $P_{cc} = P_{ci} = 0,9$, $n_{coef} = 10$, $\eta_{cx} = 2$. Para este caso, la posición esperada es $[1; 1; 2]$ mientras que el generado es $[0.9979; 1.0064; 2.0034]$, lo cual se puede ver en la figura 4.2., con un error de $5,776 * 10^{-5}$ y energía de 120.7729 [s. u.].

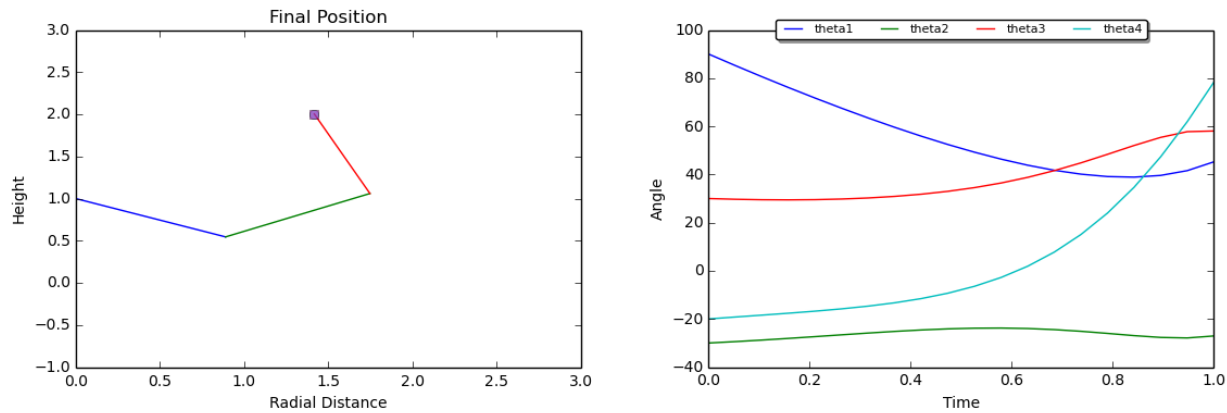


Figura 4.2: Trayectoria de los cuatro ángulos y posición de los tres brazos planares alcanzando la posición final para el caso de correntropia en ambos fitness.

El segundo resultado importante es hacer uso de un fitness con la energía simple citada en marco teórico y metodología y el error cuadrático medio, realizando la minimización del fitness relacionado. En este caso, se tienen los mismos parámetros del caso anterior, excepto por el λ que se cambia a $\lambda = 0,001$. La posición esperada es [1; 1; 2] mientras que el generado es [1.0134; 1.0134; 1.9859], con un error de 0,000557 y energía de 52.28 [s. u.], cuyos resultados se pueden notar en la figura 4.3 ,

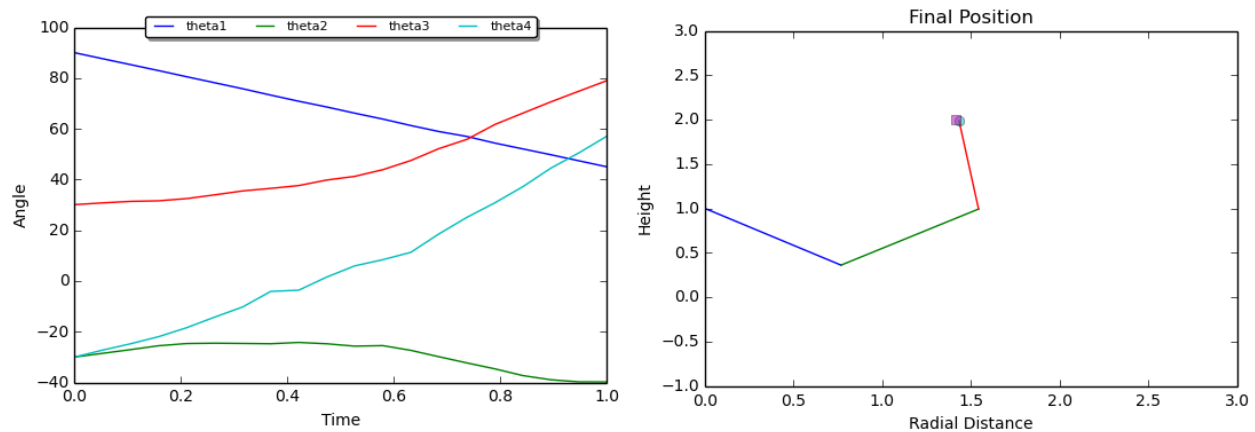


Figura 4.3: Trayectoria de los cuatro ángulos y posición de los tres brazos planares alcanzando la posición final para el caso de energía simple.

El tercer resultado importante, el cual se puede ver en la figura (4.4) es hacer uso del fitness relacionado con el torque RMS y el error cuadrático medio, realizando la minimización del fitness relacionado. En este caso se tienen como parámetros $P_{ci} = P_{cc} = 0,9$, $P_{mi} = 0,5$, $P_{mc} = 0,2$, el número de posiciones es 20, el de individuos es 500 y el lambda usado es $\lambda = 0,01$ con 50 generaciones. La posición esperada es [1.93; 1.93; 1] mientras que el generado es [1.933; 1.933; 0.975], con un error de 0,000615 y energía de 35.0084 [s. u.].

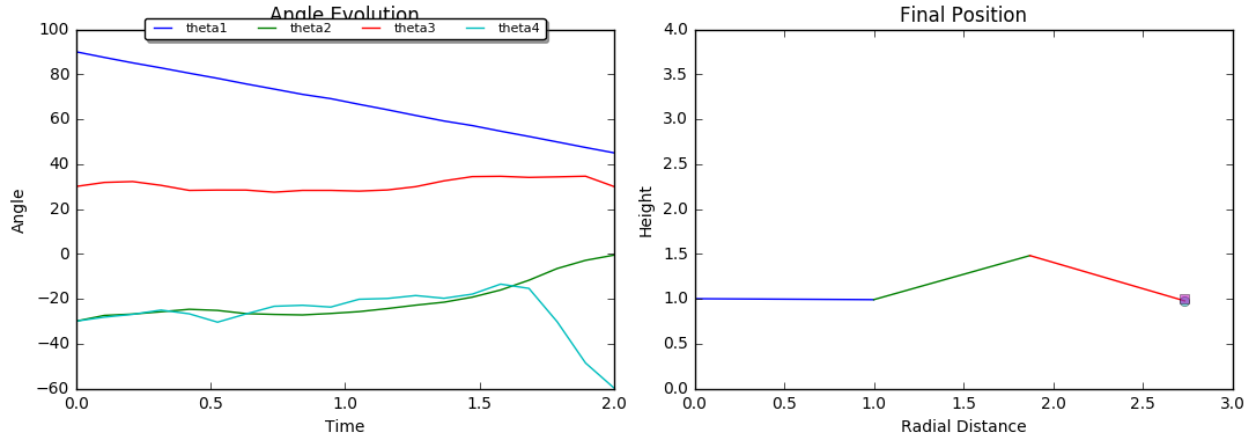


Figura 4.4: Trayectoria de los cuatro ángulos y posición de los tres brazos planares alcanzando la posición final para el caso de torque RMS.

4.2. Variación de parámetros

A continuación, se presentan los desempeños para el algoritmo genético continuo y el polinomial al variar algunos parámetros que experimentalmente se detectaron de mayor sensibilidad en el desempeño. Para el caso de los fitness (22) y (23) se hizo variar λ y P_{MC} (probabilidad de mutar una característica de un individuo seleccionado para mutación). El resto de las probabilidades se fijaron en $P_{cc} = P_{ci} = 0,9$ y $P_{mi} = 0,5$; la población en 300; el número de generaciones en 70 para el caso polinomial y 50 para el caso continuo; 20 puntos en la trayectoria. Los ángulos de la configuración de inicio son $(\theta_1, \theta_2, \theta_3, \theta_4) = (90, -30, 30, -20)$ y la posición final se fija en $(x, y, z) = (1, 1, 2)$. La expresión para la energía utilizada es la energía simplificada. Los resultados para el fitness (22) se muestran en las Tablas 4.1 y 4.2, mientras que los del fitness (23) se muestran en las Tablas 4.3 y 4.4.

Además, se hacen pruebas utilizando el fitness (24) inspirado en la correntropía. Para estos experimentos se hacen variar λ y σ_2 (el ancho de banda asociado a la energía). El resto de los parámetros permanece igual que en los experimentos anteriores. El ancho de banda del error σ_1 se fija en 5 (un valor que tiene un buen desempeño experimentalmente para el error) y la probabilidad P_{MC} se fija en 0.1 en base a los desempeños observados para los fitness anteriores en cada algoritmo. Los resultados se muestran en las Tablas 4.5 y 4.6.

Para el desarrollo de estos experimentos se corre el algoritmo genético 4 veces por cada combinación de parámetros, para así calcular el promedio y la desviación estándar de cada caso. En las tablas se han destacado algunas combinaciones de parámetros en negrita, puesto que, dentro de su propio experimento, son resultados que poseen un buen desempeño.

Tabla 4.1: Variación de λ y P_{MC} para GA polinomial y fitness (22)

λ	P_{MC}	Error	Energía
0.5	0.4	3.79e+0 \pm 2.00e+0	3.95e+0 \pm 2.05e+0
0.5	0.3	3.68e+0 \pm 2.14e+0	3.34e+0 \pm 1.98e+0
0.5	0.2	3.88e+0 \pm 1.95e+0	2.48e+0 \pm 1.49e+0
0.5	0.1	3.73e+0 \pm 2.12e+0	1.92e+0 \pm 1.14e+0
0.1	0.4	8.39e-1 \pm 5.58e-1	1.13e+1 \pm 6.66e+0
0.1	0.3	7.06e-1 \pm 5.38e-1	1.20e+1 \pm 7.07e+0
0.1	0.2	4.82e-1 \pm 2.51e-1	1.28e+1 \pm 6.42e+0
0.1	0.1	7.21e-1 \pm 4.75e-1	8.77e+0 \pm 5.12e+0
0.01	0.4	7.53e-2 \pm 4.96e-2	2.65e+1 \pm 1.39e+1
0.01	0.3	4.81e-2 \pm 3.39e-2	3.11e+1 \pm 1.58e+1
0.01	0.2	4.14e-2 \pm 2.83e-2	3.01e+1 \pm 1.56e+1
0.01	0.1	1.74e-2 \pm 1.72e-2	2.78e+1 \pm 1.39e+1
0.001	0.4	1.19e-2 \pm 1.20e-2	6.57e+1 \pm 3.85e+1
0.001	0.3	3.24e-2 \pm 4.80e-2	1.80e+2 \pm 2.26e+2
0.001	0.2	4.35e-2 \pm 7.26e-2	1.65e+2 \pm 2.21e+2
0.001	0.1	9.88e-2 \pm 1.27e-1	2.53e+2 \pm 2.54e+2

Tabla 4.2: Variación de λ y P_{MC} para GA continuo y fitness (22)

λ	P_{MC}	Error	Energía
0.5	0.4	5.40e+0 \pm 2.86e+0	2.55e+1 \pm 1.37e+1
0.5	0.3	3.75e+0 \pm 2.04e+0	2.67e+1 \pm 1.65e+1
0.5	0.2	3.27e+0 \pm 2.13e+0	1.72e+1 \pm 9.89e+0
0.5	0.1	4.10e+0 \pm 2.53e+0	1.29e+1 \pm 6.84e+0
0.1	0.4	3.17e+0 \pm 1.67e+0	2.85e+1 \pm 1.49e+1
0.1	0.3	2.88e+0 \pm 1.60e+0	2.60e+1 \pm 1.62e+1
0.1	0.2	3.22e+0 \pm 1.74e+0	1.53e+1 \pm 8.14e+0
0.1	0.1	2.59e+0 \pm 1.49e+0	1.59e+1 \pm 8.50e+0
0.01	0.4	4.96e-1 \pm 2.77e-1	7.98e+1 \pm 4.34e+1
0.01	0.3	4.96e-1 \pm 4.00e-1	6.75e+1 \pm 3.97e+1
0.01	0.2	1.09e-1 \pm 8.61e-2	4.85e+1 \pm 2.61e+1
0.01	0.1	2.08e-1 \pm 1.40e-1	4.02e+1 \pm 2.11e+1
0.001	0.4	3.18e-2 \pm 4.69e-2	1.08e+2 \pm 5.76e+1
0.001	0.3	8.85e-3 \pm 1.46e-2	7.46e+1 \pm 4.33e+1
0.001	0.2	8.30e-3 \pm 1.34e-2	6.52e+1 \pm 3.58e+1
0.001	0.1	2.66e-3 \pm 2.22e-3	4.95e+1 \pm 2.63e+1

Tabla 4.3: Variación de λ y P_{MC} para GA polinomial y fitness (23)

λ	P_{MC}	Error	Energía
0.5	0.4	$5.98\text{e-}3 \pm 4.62\text{e-}3$	$2.41\text{e}+3 \pm 1.21\text{e}+3$
0.5	0.3	$8.99\text{e-}3 \pm 6.61\text{e-}3$	$2.52\text{e}+3 \pm 1.32\text{e}+3$
0.5	0.2	$1.24\text{e-}2 \pm 1.3\text{e-}2$	$2.50\text{e}+3 \pm 1.33\text{e}+3$
0.5	0.1	$2.23\text{e-}2 \pm 1.78\text{e-}2$	$2.50\text{e}+3 \pm 1.33\text{e}+3$
0.1	0.4	$8.33\text{e-}3 \pm 4.26\text{e-}3$	$2.09\text{e}+3 \pm 1.10\text{e}+3$
0.1	0.3	$1.32\text{e-}2 \pm 1.33\text{e-}2$	$2.74\text{e}+3 \pm 1.52\text{e}+3$
0.1	0.2	$1.24\text{e-}2 \pm 8.24\text{e-}3$	$2.06\text{e}+3 \pm 1.07\text{e}+3$
0.1	0.1	$9.38\text{e-}3 \pm 1.09\text{e-}2$	$2.09\text{e}+3 \pm 1.04\text{e}+3$
0.01	0.4	$1.00\text{e-}2 \pm 8.42\text{e-}3$	$2.37\text{e}+3 \pm 1.18\text{e}+3$
0.01	0.3	$1.08\text{e-}2 \pm 6.27\text{e-}3$	$2.67\text{e}+3 \pm 1.45\text{e}+3$
0.01	0.2	$9.69\text{e-}3 \pm 5.46\text{e-}3$	$2.02\text{e}+3 \pm 1.03\text{e}+3$
0.01	0.1	$8.56\text{e-}3 \pm 5.51\text{e-}3$	$2.77\text{e}+3 \pm 1.43\text{e}+3$
0.001	0.4	$1.03\text{e-}2 \pm 6.82\text{e-}3$	$2.43\text{e}+3 \pm 1.60\text{e}+3$
0.001	0.3	$8.74\text{e-}3 \pm 6.31\text{e-}3$	$2.26\text{e}+3 \pm 1.24\text{e}+3$
0.001	0.2	$9.51\text{e-}3 \pm 6.06\text{e-}3$	$2.17\text{e}+3 \pm 1.17\text{e}+3$
0.001	0.1	$9.38\text{e-}3 \pm 5.96\text{e-}3$	$2.40\text{e}+3 \pm 1.37\text{e}+3$

Tabla 4.4: Variación de λ y P_{MC} para GA continuo y fitness (23)

λ	P_{MC}	Error	Energía
0.5	0.4	$3.70\text{e-}6 \pm 4.47\text{e-}6$	$1.32\text{e}+3 \pm 8.89\text{e}+2$
0.5	0.3	$1.02\text{e-}5 \pm 1.49\text{e-}5$	$2.28\text{e}+3 \pm 3.15\text{e}+3$
0.5	0.2	$3.64\text{e-}6 \pm 2.62\text{e-}6$	$7.36\text{e}+2 \pm 4.25\text{e}+2$
0.5	0.1	$3.93\text{e-}5 \pm 4.95\text{e-}5$	$1.21\text{e}+3 \pm 1.05\text{e}+3$
0.1	0.4	$1.66\text{e-}7 \pm 1.37\text{e-}7$	$4.26\text{e}+3 \pm 3.38\text{e}+3$
0.1	0.3	$7.62\text{e-}7 \pm 6.16\text{e-}7$	$1.49\text{e}+3 \pm 1.03\text{e}+3$
0.1	0.2	$7.41\text{e-}7 \pm 7.59\text{e-}7$	$1.10\text{e}+3 \pm 6.81\text{e}+2$
0.1	0.1	$5.04\text{e-}5 \pm 9.98\text{e-}5$	$1.74\text{e}+4 \pm 3.39\text{e}+4$
0.01	0.4	$3.90\text{e-}8 \pm 3.59\text{e-}8$	$4.65\text{e}+3 \pm 2.78\text{e}+3$
0.01	0.3	$4.76\text{e-}8 \pm 3.59\text{e-}8$	$2.21\text{e}+3 \pm 1.43\text{e}+3$
0.01	0.2	$3.01\text{e-}8 \pm 2.22\text{e-}8$	$7.23\text{e}+2 \pm 5.01\text{e}+2$
0.01	0.1	$2.36\text{e-}7 \pm 3.77\text{e-}7$	$6.88\text{e}+2 \pm 5.96\text{e}+2$
0.001	0.4	$7.06\text{e-}9 \pm 6.06\text{e-}9$	$5.47\text{e}+3 \pm 3.77\text{e}+3$
0.001	0.3	$2.67\text{e-}9 \pm 1.97\text{e-}9$	$2.60\text{e}+3 \pm 1.60\text{e}+3$
0.001	0.2	$4.96\text{e-}9 \pm 6.98\text{e-}9$	$2.13\text{e}+3 \pm 1.20\text{e}+3$
0.001	0.1	$1.97\text{e-}8 \pm 1.58\text{e-}8$	$1.72\text{e}+3 \pm 2.18\text{e}+3$

Tabla 4.5: Variación de λ y σ_2 para GA polinomial y fitness (24)

λ	σ_2	Error	Energía
0.5	50	$3.28\text{e-}1 \pm 6.49\text{e-}1$	$2.14\text{e+}3 \pm 1.78\text{e+}3$
0.5	100	$4.14\text{e-}1 \pm 4.68\text{e-}1$	$2.05\text{e+}3 \pm 4.08\text{e+}3$
0.5	150	$2.98\text{e-}1 \pm 2.30\text{e-}1$	$1.79\text{e+}1 \pm 9.08\text{e+}0$
0.5	200	$9.22\text{e-}2 \pm 5.40\text{e-}2$	$1.70\text{e+}1 \pm 8.57\text{e+}0$
0.1	50	$9.22\text{e-}2 \pm 1.50\text{e-}1$	$2.86\text{e+}3 \pm 2.90\text{e+}3$
0.1	100	$4.30\text{e-}1 \pm 5.53\text{e-}1$	$5.89\text{e+}2 \pm 1.11\text{e+}3$
0.1	150	$1.96\text{e-}1 \pm 3.19\text{e-}1$	$3.80\text{e+}1 \pm 2.11\text{e+}1$
0.1	200	$2.80\text{e-}2 \pm 2.52\text{e-}2$	$3.53\text{e+}1 \pm 1.79\text{e+}1$
0.01	50	$6.42\text{e-}3 \pm 5.53\text{e-}3$	$2.93\text{e+}3 \pm 1.70\text{e+}3$
0.01	100	$8.68\text{e-}3 \pm 1.24\text{e-}2$	$2.75\text{e+}3 \pm 1.92\text{e+}3$
0.01	150	$1.24\text{e-}3 \pm 1.14\text{e-}3$	$3.90\text{e+}3 \pm 3.55\text{e+}3$
0.01	200	$9.29\text{e-}3 \pm 1.10\text{e-}2$	$1.91\text{e+}3 \pm 1.00\text{e+}3$
0.001	50	$5.42\text{e-}3 \pm 6.58\text{e-}3$	$2.26\text{e+}3 \pm 1.34\text{e+}3$
0.001	100	$6.58\text{e-}3 \pm 6.19\text{e-}3$	$2.15\text{e+}3 \pm 2.20\text{e+}3$
0.001	150	$4.89\text{e-}3 \pm 7.58\text{e-}3$	$4.65\text{e+}2 \pm 7.00\text{e+}2$
0.001	200	$5.28\text{e-}3 \pm 6.36\text{e-}3$	$3.40\text{e+}3 \pm 3.25\text{e+}3$

Tabla 4.6: Variación de λ y σ_2 para GA continuo y fitness (24)

λ	σ_2	Error	Energía
0.5	50	$3.95\text{e-}1 \pm 7.90\text{e-}1$	$2.26\text{e+}4 \pm 1.89\text{e+}4$
0.5	100	$7.42\text{e-}2 \pm 1.48\text{e-}1$	$1.20\text{e+}4 \pm 1.16\text{e+}4$
0.5	150	$3.66\text{e+}0 \pm 4.00\text{e+}0$	$1.85\text{e+}2 \pm 1.28\text{e+}2$
0.5	200	$4.53\text{e+}0 \pm 5.08\text{e+}0$	$9.33\text{e+}1 \pm 5.13\text{e+}1$
0.1	50	$1.24\text{e-}7 \pm 2.49\text{e-}7$	$4.07\text{e+}4 \pm 4.49\text{e+}4$
0.1	100	$1.10\text{e-}10 \pm 2.17\text{e-}10$	$2.90\text{e+}4 \pm 1.77\text{e+}4$
0.1	150	$3.64\text{e-}1 \pm 5.56\text{e-}1$	$2.18\text{e+}4 \pm 3.84\text{e+}4$
0.1	200	$5.88\text{e-}1 \pm 4.68\text{e-}1$	$2.04\text{e+}2 \pm 1.06\text{e+}2$
0.01	50	$2.26\text{e-}11 \pm 4.51\text{e-}11$	$2.12\text{e+}4 \pm 1.14\text{e+}4$
0.01	100	$3.12\text{e-}6 \pm 6.24\text{e-}6$	$8.29\text{e+}4 \pm 6.26\text{e+}4$
0.01	150	$2.02\text{e-}2 \pm 4.05\text{e-}2$	$4.48\text{e+}4 \pm 4.69\text{e+}4$
0.01	200	$4.59\text{e-}4 \pm 9.18\text{e-}4$	$1.30\text{e+}4 \pm 1.55\text{e+}4$
0.001	50	$3.62\text{e-}6 \pm 7.25\text{e-}6$	$4.90\text{e+}4 \pm 4.00\text{e+}4$
0.001	100	$9.83\text{e-}12 \pm 1.42\text{e-}11$	$3.10\text{e+}4 \pm 2.51\text{e+}4$
0.001	150	$4.28\text{e-}11 \pm 5.30\text{e-}11$	$8.06\text{e+}4 \pm 6.82\text{e+}4$
0.001	200	$3.25\text{e-}11 \pm 6.46\text{e-}11$	$3.44\text{e+}4 \pm 2.79\text{e+}4$

5. Análisis

En primer lugar, se procede a analizar los resultados particulares mostrados en la primera parte de la sección anterior, los cuales constituyen buenos casos encontrados a mano dentro del tipo de algoritmo utilizado, antes de las variaciones más sistemáticas de parámetros realizadas en la segunda parte. La experimentación a mano permitió, además de obtener los tres casos ilustrados, tener una idea de las magnitudes de los parámetros a utilizar y reconocer aquellos que son más sensibles a la hora de evaluar el desempeño de la evolución.

El resultado de la Figura 4.2, obtenido utilizando el GA polinomial, muestra un caso en el que se consiguió alcanzar un error del orden de 10^{-5} y una energía relativamente baja (aunque no la mínima del conjunto de experimentos) del orden de 120. El error es bastante pequeño y es difícil apreciar una diferencia de la posición final y la deseada (marcada con un cuadrado morado) en la imagen de la izquierda que describe la orientación final del brazo. Sin embargo, esto es solo una parte del desempeño, puesto que también es necesario que la trayectoria para llegar a tal punto sea una que no posea movimientos que desperdicien energía. Observando la imagen de la derecha, es posible apreciar que las trayectorias se mantienen dentro de los límites establecidos al momento de la experimentación (± 180 para θ_1 y ± 90 para los demás), lo cual verifica que el método propuesto para regularizar las trayectorias es efectivo. Por supuesto, en primera instancia es aceptable también pensar que la regularización nunca se activó durante la evolución, pero esto no es correcto puesto que para la misma posición target se corrió el algoritmo sin regularización y los ángulos tendían a escapar de los rangos, a veces de formas tan notorias como para alcanzar ángulos del orden de los 1000 grados. Esto, aparte de estar fuera de los límites físicos, resulta en un gasto innecesario de la energía al dar vueltas en círculos. Debido a ello es que en un principio se pensó que no era necesario implementar una regularización con la esperanza de que el algoritmo genético optara por aquellos individuos que no se salían de los rangos. Sin embargo, la convergencia resultó ser muy lenta ya que durante los experimentos, para el número de generaciones utilizadas (200 máximo), no se observó este efecto. En definitiva, es difícil dejar que las trayectorias vayan disminuyendo sus valores hasta entrar en el rango de forma autónoma, sobre todo por la inicialización aleatoria de los coeficientes, por lo que la regularización de la trayectoria es un proceso principal y que ayuda a converger más rápido a soluciones adecuadas. Enfocando la atención ahora en la forma de las curvas angulares, se puede apreciar que ellas se acercan bastante a lo que se diría una trayectoria óptima sin oscilaciones ociosas, lo que se demuestra en el bajo valor de su energía. A pesar de ello, sí es posible encontrar movimientos redundantes que a simple vista permiten inferir gasto innecesario de energía como es el caso de la curva azul y verde (θ_1 y θ_2 respectivamente). Estos ángulos presentan durante su evolución ligeros mínimos o máximos locales que no aportan en la trayectoria ya que el brazo debe luego retroceder. Por último, los restantes dos ángulos presentan buenas trayectorias, sin requerir un retroceso por parte de la junta.

Siguiendo con los resultados particulares, en la figura 4.3 se presenta la utilización del algoritmo genético continuo en donde se quiere minimizar el fitness del error y energía, ocupando error cuadrático medio y la energía simple respectivamente. De este se obtiene un error del orden de $5 * 10^{-4}$ y una energía pequeña del orden de 50. Se puede notar que la energía en valor es pequeña y que las trayectorias de los ángulos en el tiempo (parte izquierda de la figura) no realizan movimientos ociosos en el tiempo pues no varían bruscamente, que es lo que se quiere para minimizar la energía. Se obtiene además que la posición generada no es muy cercana a la deseada, teniendo aquí una importante desventaja para este algoritmo el cual si bien logra minimizar la energía y

evitar los movimientos ociosos de los brazos en el tiempo, no llega como se desea a la posición final con un error despreciable.

El tercer resultado particular corresponde al presentado en la figura 4.4 en el cual se tiene un algoritmo genético continuo en donde se quiere minimizar el fitness del error y energía, ocupando de nuevo el error cuadrático medio pero con la diferencia en este caso de que se ocupa el torque RMS para el fitness de la energía. De este fitness se obtiene un error del orden $6 * 10^{-4}$ y una energía pequeña del orden de 30. Se puede notar que para este algoritmo la energía es pequeña en comparación a valores medidos ante una variación de parámetros, y de la parte izquierda de la figura es importante notar la forma de las curvas; para el caso de torque RMS lo óptimo sería que solamente el brazo que contiene al efector se moviera para realizar el menor torque y gastar la menor cantidad de energía, dado un cierto ángulo del ángulo polar y esperando que los otros dos ángulos, es decir del brazo 2 y 3, con respecto a sus horizontales se mantengan en el tiempo o bien no afecten en la minimización de energía. Y este resultado esperado se puede notar que se cumple lo esperado, los ángulos θ_2 y θ_3 se mantienen en una curva suave en el tiempo, generando que el efector se mueva solo por hecho de que se está variando el ángulo del brazo que lo sostiene.

Es necesario poder comparar estos tres resultados particulares, en donde se recalca que fueron elegidos luego de ir cambiando distintos parámetros y fijando cuáles eran los mejores resultados para cada fitness. Por el lado de las curvas que representan las trayectorias de los ángulos, resulta claro que en un algoritmo genético continuo se puede lograr que una trayectoria sea casi plana en el tiempo que en este caso es lo que se quiere para evitar que se gaste la energía para una parte que se sabe que gasta de manera ociosa; por ejemplo en la figura 4.4 se tiene que el tercer ángulo resulta ser casi plano, solamente queda ir cambiando un poco algunos otros parámetros para lograr el óptimo, que queda propuesto como trabajo futuro. Comparando esta solución con el ajuste polinomial, la manera en que las curvas son en función del tiempo tienden a no ser planas, implicando que la única manera en que se obtenga una curva plana es que todos los coeficientes sean constantes o bien todos nulos salvo uno, perdiendo eficacia de aproximación del polinomio. Por otro lado se puede notar que los errores de posición de los algoritmos genéticos continuos particulares mostrados son un orden de magnitud mayor al algoritmo clásico de GA con polinomio, lo cual se condice con el hecho de que este último penaliza menos la energía, la cual resulta mayor de los tres casos, pero ajusta mejor la posición final del efector.

Una vez analizados los ejemplos particulares de la primera parte de los resultados, ahora es momento de revisar aquellos en que se varió de forma sistemática parámetros sensibles en el desempeño final de los algoritmos. Esto se hizo para cada uno de los tres fitness propuestos para encontrar cuál fitness podría ser el más adecuado para el problema, y debido a la condición multiobjetivo del mismo se introdujo en todas las variaciones aquella del parámetro λ que controla el peso de cada uno de los fitness parciales (el del error y el de la energía). En general, es esperable que λ deba ser menor a 0.5, ya que se observó experimentalmente que si la energía pesa más que el error en la posición final el brazo optaría por moverse muy poco, a veces tan poco que el efector queda muy distante a su posición target. Esto justifica el uso del rango de λ solo hasta 0.5. Otro punto particularmente sensible a la hora de correr los algoritmos fue la manera de mutar a los individuos, lo cual es esperable puesto que este proceso introduce aleatoriedad en la evolución. La selección del más apto para sobrevivir en la siguiente generación ayuda a atenuar este efecto si es que se descontrola en alguna generación particular, pero una tasa muy alta de mutación puede acabar con los mejores individuos después de unas pocas generaciones y la evolución se torna más inestable y de convergencia más lenta. Es por esta razón que se elige la tasa de mutación de cada atributo de un individuo como segundo parámetro a variar en los primeros experimentos que

involucran al primer y segundo fitness (es decir, sin correntropía). Al momento de agregar la correntropía a los experimentos, fue necesario fijar la tasa de mutación para poder tener algún ancho de banda disponible para variar. Por supuesto, se podría haber variado más de un par de parámetros, pero esto complejiza el análisis y se optó por lo mencionado. La tasa de mutación se fijó en un valor apropiado según lo que se observó en los experimentos para los dos primeros fitness, que, tal como se esperaba, es un valor bajo. Así, se seleccionó el σ_2 de la energía como segundo parámetro a variar junto con λ . El ancho de banda del error se fijó en 5, un valor que entregó valores buenos de desempeño durante las pruebas preliminares donde se ajustaron parámetros a mano. Esto, por supuesto, no implica que sea el valor correcto, pero dado que la energía fue más problemática en estas primeras instancias se descartó el error como variación en favor de variar la energía.

En primer lugar, se analiza el comportamiento de los algoritmos bajo el uso de la primera opción de fitness, la cual minimiza directamente los valores numéricos del MSE y la energía simplificada, lo cual se encuentra en las Tablas 4.1 y 4.2. Al observar estas tablas, se puede apreciar que los errores alcanzados en promedio no son tan pequeños como se podría esperar a partir de los tres ejemplos particulares mostrados en la primera parte. Sin embargo, hay que recordar que el proceso de inicialización del algoritmo, y de evolución, tiene una componente aleatoria fuerte, y por lo mismo es que ocurren casos en que los desempeños pueden ser particularmente malos. Es directo apreciar de esta tabla que en la medida en que λ es mayor, la energía disminuye, lo cual se corresponde con el hecho de que λ regula la importancia del fitness de la energía dentro de la función fitness global. Sin embargo, se puede observar que para el caso extremo $\lambda = 0,5$ la energía se minimiza tanto que el error es notoriamente grande, indicando que el efector no es capaz de acercarse a la posición target ya que al proceso de evolución le importa más gastar poca energía que llegar a su destino. Por otra parte, si λ es muy pequeño, se observa que el error puede llegar a sus valores más bajos, pero a riesgo de que la energía utilizada en la trayectoria aumente mucho, lo que indica una trayectoria angular con movimientos ociosos u oscilantes. Esto por supuesto no es seguido estrictamente, ya que por ejemplo uno de los mejores casos para el algoritmo CGA fue utilizando el menor λ considerado. Otro efecto que es posible apreciar es que, para un mismo valor de λ , bajar la probabilidad de mutación permite en general disminuir la energía utilizada. Esto se explica puesto que al tener una tasa pequeña de mutación, es posible ir ajustando y manteniendo el ajuste realizado a través de las generaciones en busca de una trayectoria óptima. Una alta tasa de mutación genera cambios en las trayectorias que pueden destruir este camino hacia el óptimo. Se espera, sin embargo, que exista una mínima tasa de mutación luego de la cual el algoritmo se vuelva inestable ya que a partir de ahí dependerá con demasiada fuerza en la inicialización de los genotipos, ya que la información se moverá principalmente a partir de esta inicialización.

Comparando ambos algoritmos es posible apreciar que el GA polinomial es capaz de alcanzar energías particularmente bajas si es que se lo propone. De hecho, entre los ganadores de ambos experimentos, los que consumen menos energía son aquellos del GA polinomial, aunque dentro del mismo orden de magnitud y por lo mismo sujetos a la aleatoriedad. En general, en ambos algoritmos se encuentran combinaciones donde se alcanzan valores que podrían considerarse de baja energía, ya que están por debajo de las 50 unidades. Incluso por debajo de 30 para el caso polinomial.

Cambiando ahora a la segunda opción de fitness, en las Tablas 4.3 y 4.4, se puede apreciar que, si bien los errores son bastante pequeños llegando incluso al orden de 10^{-9} en el caso de CGA, el desempeño es notoriamente malo en la energía. En prácticamente todos los casos la energía se encuentra por lo menos en el tercer orden de magnitud, y todos en la práctica bastante cerca de

este orden, lo que está lejos de los resultados encontrados en las dos primeras tablas, siendo que poseen el mismo objetivo para la planificación. Si bien se han destacado algunas combinaciones dentro de estas dos tablas como los mejores resultados en ellas, ninguno se acerca a los óptimos alcanzados anteriormente. Esto indica que esta opción de fitness es inadecuada para el problema, probablemente debido a que las magnitudes de la energía hacen que la expresión inversa sea demasiado pequeña y entre a la zona de derivada cercana a cero, no permitiendo diferenciar bien entre trayectorias óptimas y subóptimas.

Por último, se analiza el uso de la tercera opción de fitness, en las Tablas 4.5 y 4.6. Al observar la tabla asociada al GA polinomial, es posible apreciar que existe una ventana, en donde se encuentran las mejores dos combinaciones, que consigue desempeños muy similares a los mejores obtenidos en la primera tablas analizadas. Esto indica que la tercera opción de fitness, utilizando correntropía, interactúa bien con el GA polinomial. Teniendo en cuenta que el ancho de banda del error está fijado previamente, y que las variaciones de λ y σ_2 no son tan finas en los casos estudiados, no se descarta la posibilidad que ante unos buenos valores de ancho de banda el desempeño de la correntropía supere al alcanzado por la minimización directa del MSE y la energía. Esto debido a que la correntropía permite obtener fitness con las buenas propiedades de estar acotados por arriba por 1 y con la posibilidad de ajustar su escala a través de los σ . El máximo de 1 se tiene también teóricamente en la segunda opción de fitness, pero los cambios de escala lo hicieron un fitness poco robusto, provocando que en la práctica la energía estuviese muy lejos de esta cota. Al observar la tabla asociada al CGA, es directo apreciar que el desempeño es muy malo, alcanzando energías en su mayoría del cuarto orden de magnitud. Pero también es apreciable que los valores de los errores son notoriamente más pequeños que los casos anteriormente estudiados. Este caso recuerda a lo observado en la Tabla 4.4 asociada a este mismo algoritmo pero con la segunda opción de fitness, donde los problemas de escala hicieron que la energía no fuera bien minimizada y por el contrario el error se llevara a magnitudes notoriamente más pequeñas en promedio. Esta observación sugiere que el problema no es tanto de la elección del fitness sino que de la elección de la escala reflejada en los σ , que, al ser este un algoritmo distinto, no tiene por qué funcionar con los mismos valores de ancho de banda que el caso GA polinomial. Recordando además que el ancho de banda del error fue fijado de antemano, es posible que un buen desempeño se pueda conseguir luego de un estudio exhaustivo de los parámetros que puedan estar fallando. Se puede observar que la combinación destacada en esta tabla obtiene un desempeño que, si bien no es óptimo, no está tan lejano como el resto, ya que posee una energía del orden de los 200 y un error del orden de 10^{-1} . Esto motiva aun más la hipótesis de que es posible ajustar de mejor manera los parámetros para encontrar un desempeño óptimo similar al visto en el GA polinomial.

6. Conclusiones

Se logran los objetivos del proyecto, pues se logra resolver la cinemática inversa de un manipulador robótico de cuatro grados de libertad para mover el efector hacia una cierta posición deseada minimizando energía y usando algoritmos genéticos.

Por otro lado, el diseño del fitness de un algoritmo genético es una pieza clave para lograr los propósitos de este, pues se ha aprendido que con un fitness que no discrimina bien a sus individuos el algoritmo toma mucho más costo computacional en cálculos que se deben optimizar, volviendo la convergencia notoriamente más lenta. También se vio que existe una gran cantidad de parámetros disponibles para el diseño, para los cuales fue necesario consultar en la literatura valores típicos para dejar a solo unos cuantos disponibles para variaciones más específicas.

Se recomienda que para fitness que sean complejos matemáticamente, primero hacer aproximaciones básicas para ir ajustando los distintos parámetros e ir incrementando el nivel de las aproximaciones hasta obtener un estimador decente. En particular, en este trabajo se observó que si bien la expresión compleja del torque permitía buenas trayectorias, estas no se distanciaron tanto de aquellas obtenidas en base a la expresión simplificada de la energía. Así, para aplicaciones sencillas no se justifica el uso de un fitness tan complejo que además aumenta significativamente el costo computacional durante la evolución, siendo que existe una aproximación alternativa al problema que entrega resultados bastante cercanos.

También se recomienda que el algoritmo genético sea corrido varias veces antes de seleccionar la solución. Esto ya que, manteniendo los parámetros, las soluciones entregadas van variando tal como se observó en las tablas generadas debido a la aleatoriedad inserta en los operadores y en la inicialización.

El GA polinomial se observó como una buena alternativa a la literatura para atacar el problema de la planificación de una trayectoria suave. Sin embargo, en general no es óptimo para trayectorias angulares planas debido a la dificultad de hacer que un polinomio tenga esa curva en el tiempo. Para obtener una trayectoria horizontal sería necesario hacer evolucionar todos los coeficientes asociados a cero, y para una trayectoria recta inclinada serían todos menos el primero. De aquí se ve que en general las soluciones tenderán a mostrar ligeras curvaturas. Además, requiere en general de más generaciones para alcanzar un desempeño como el del CGA. En los experimentos se utilizaron 100 generaciones en el GA polinomial para alcanzar desempeños comparables al del CGA con 50 generaciones. Por otro lado, el fitness que se recomienda usar es el de error cuadrático medio y torque rms para minimización directa, pues gracias a las aproximaciones que se hicieron de esta energía se pudo hacer un ajuste de parámetros que permitió obtener resultados muy satisfactorios. No se descarta que ante un mejor ajuste de parámetros la correntropía pueda ser una mejor opción. De todos modos, utilizarla agrega dos nuevos parámetros de diseño que llevan a un planteamiento más complejo de la solución.

El uso de la correntropía en la energía permite ajustar una escala que evita un castigo excesivo a las energías cercanas a valores considerados bajos. Sus buenos resultados se lograron ver parcialmente en el GA polinomial, aunque no en el CGA. Se recomienda como trabajo futuro profundizar más en la determinación correcta de la correntropía y obtener el valor del tamaño del kernel, tanto del error como de la energía, a través de estimaciones de densidad, como la regla de Silverman o bien máxima verosimilitud, y con este valor de *zoom-lens* de similitud empezar a comparar con mse.

Referencias

- [1] R. K. Mandava and P. R. Vundavalli, "Design of pid controllers for 4-dof planar and spatial manipulators," in *Robotics, Automation, Control and Embedded Systems (RACE), 2015 International Conference on*, pp. 1–6, IEEE, 2015.
- [2] S. Momani, Z. S. Abo-Hammour, and O. M. Alsmadi, "Solution of inverse kinematics problem using genetic algorithms," *Applied Mathematics & Information Sciences*, vol. 10, no. 1, p. 225, 2016.
- [3] Universitat Politècnica de Catalunya, "Chapter 3. forward kinematics: The denavit-hartenberg convention." <https://upcommons.upc.edu/bitstream/handle/2099.1/24573/A-Denavit%20Hartenberg.pdf>. En línea; accedido en octubre 2016.
- [4] D. Shiffman, S. Fry, and Z. Marsh, *The Nature of Code*. D. Shiffman, 2012.
- [5] J. McCall, "Genetic algorithms for modelling and optimisation," *Journal of Computational and Applied Mathematics*, vol. 184, no. 1, pp. 205–222, 2005.
- [6] S. Za'er, N. M. Mirza, S. M. Mirza, and M. Arif, "Cartesian path generation of robot manipulators using continuous genetic algorithms," *Robotics and autonomous systems*, vol. 41, no. 4, pp. 179–223, 2002.
- [7] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: properties and applications in non-gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.
- [8] J. Kim, *Dynamics and Motion Planning of Redundant Manipulators Using Optimization, with Applications to Human Motion*. University of Iowa, 2006.
- [9] Distributed Evolutionary Algorithms in Python, "Deap 1.1.0 documentation." <http://deap.readthedocs.io/en/master/index.html>, 2016. En línea; accedido en octubre 2016.
- [10] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 3, pp. 1–15, 1994.
- [11] Wiki de Robótica, "Metodología de denavit-hartenberg." <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/metodologia-de-denavit-hartenberg/>. En línea; accedido en diciembre 2016.
- [12] A. Ghosal, "Kinematics of serial manipulators." http://www.mecheng.iisc.ernet.in/~sitava/-manip_serial_kin.pdf. En línea; accedido en diciembre 2016.

Anexos

A1. Código de la prueba preliminar

Los códigos implementados para este proyecto, para su mejor visualización y para facilitar su ejecución, se encuentran disponibles en:

<https://github.com/nicolasigor/inverseKinematics>

A2. Modelo dinámico del brazo robótico

Según [8], la evolución del torque de la junta i -ésima del brazo robótico está dada por:

$$\tau_i = \sum_{k=1}^4 M_{ik} \ddot{q}_k + \sum_{k=1}^4 \sum_{m=1}^4 h_{ikm} \dot{q}_k \dot{q}_m + g_i ; \quad i = 1, 2, 3, 4 \quad (33)$$

En donde:

$$M_{ik} = \sum_{j=\max\{i,k\}}^4 \text{Tr} \left(d_k^j I_j d_i^{jT} \right) \quad (34)$$

$$h_{ikm} = \sum_{j=\max\{i,k,m\}}^4 \text{Tr} \left(d_{k,m}^j I_j d_i^{jT} \right) \quad (35)$$

$$g_i = - \sum_{j=i}^4 m_j g^T \left(d_i^j {}^j r_j \right) \quad (36)$$

$$d_k^j = \frac{\partial {}^0 T_j(q)}{\partial q_k} \quad (37)$$

$${}^j r_j = (\bar{x}_j, \bar{y}_j, \bar{z}_j, 1)^T \quad (38)$$

$$g = (0, 0, -g, 0)^T \quad (39)$$

$$I_i = \begin{pmatrix} \frac{-{}^i I_{xx} + {}^i I_{yy} + {}^i I_{zz}}{2} & 0 & 0 & m_i \bar{x}_i \\ 0 & \frac{{}^i I_{xx} - {}^i I_{yy} + {}^i I_{zz}}{2} & 0 & m_i \bar{y}_i \\ 0 & 0 & \frac{{}^i I_{xx} + {}^i I_{yy} - {}^i I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{pmatrix} \quad (40)$$

Y las variables ${}^0 T_j$ representan las matrices de transformación homogéneas para el brazo robótico en cuestión en cada junta j . Para obtenerlas es necesario utilizar las matrices ${}^{i-1} T_i$ de forma que:

$${}^0 T_j = \prod_{i=1}^j {}^{i-1} T_i \quad (41)$$

Las últimas matrices mencionadas están dadas por:

$${}^{i-1}T_i = \begin{pmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (42)$$

Las cuales quedan en función de los parámetros de D-H. Para el brazo robótico del presente proyecto, estos parámetros están dados por la Tabla A2.1. Todo este análisis se encuentra respaldado según la información encontrada en [11] y [12].

Tabla A2.1: Parámetros D-H para el brazo robótico de este proyecto.

Articulación	θ_i	d_i	a_i	α_i
1	θ_1	L_1	0	90°
2	θ_2	0	L_2	0°
3	θ_3	0	L_3	0°
4	θ_4	0	L_4	0°