

Jornadas de Automática

Modelado cinemático inverso con control de forma de un robot blando mediante algoritmos genéticos

García-Samartín, J.F.^{a,*}, del Cerro, J.^a, Barrientos, A.^a

^aCentro de Automática y Robótica (UPM-CSIC), C/ José Gutiérrez Abascal, n° 2, 28006, Madrid, España.

To cite this article: García-Samartín, J.F., del Cerro, J., Barrientos, A. 2024. Inverse kinematic modelling with shape control of a soft robot using genetic algorithms.

Jornadas de Automática, 45. <https://doi.org/10.17979/ja-cea.2024.45.10968>

Resumen

Uno de los principales problemas que está encontrando la robótica blanda y, en parte, frenando su auge, es la dificultad para modelar con precisión la cinemática inversa de estos manipuladores. Su carácter redundante hace compleja esta tarea y, en multitud de ocasiones, las técnicas de aprendizaje automático precisan de un número de muestras difícilmente alcanzable. Se presenta aquí un algoritmo genético que, a partir del modelo cinemático directo, fácilmente obtenible, logra notables resultados, tanto en términos de precisión como de tiempo. En concreto se ha conseguido, al aplicarlo sobre un robot neumático modular, un error de 0,9 mm con tiempos de ejecución de 12 s. La metodología desarrollada permite, además, gestionar las redundancias y elegir la pose que se desea que el robot adopte, pudiendo recibir como entrada, además de las coordenadas del extremo, la posición deseada de cuantos módulos intermedios se precise. Esto abre la puerta a posibles aplicaciones de interés, como la teleoperación de manipuladores blandos mediante realidad virtual.

Palabras clave: Robótica blanda, Cinemática robótica, Control basado en datos, Algoritmos evolutivos, Control basado en el conocimiento, Planificación de trayectorias y caminos, Métodos de IA para robótica, Tecnología robótica

Inverse kinematic modelling with shape control of a soft robot using genetic algorithms

Abstract

One of the primary challenges facing soft robotics is the difficulty in accurately modelling the inverse kinematics of these manipulators. Their redundant nature makes this task complex, and on numerous occasions, machine learning techniques require a number of samples that is difficult to achieve. In this paper, we present a genetic algorithm that, based on the readily accessible direct kinematic model, achieves remarkable results in terms of accuracy and time. In particular, when applied to a modular pneumatic robot, an error of 0,9 mm has been achieved with execution times of 12 s. The methodology developed also allows redundancies to be managed and the pose that the robot is to adopt to be chosen. Furthermore, it is capable of receiving as input, in addition to the coordinates of the end, the desired position of as many intermediate modules as required. This opens the door to potential applications of interest, such as the teleoperation of soft manipulators by means of virtual reality.

Keywords: Soft Robotics, Robot kinematics, Data-based control, Evolutionary algorithms, Knowledge-based control, Trajectory and Path Planning, AI methods for Robotics, Robotics technology

1. Introducción

La robótica blanda es una disciplina en auge, que está encontrando aplicaciones en campos como la asistencia perso-

nal (Manti et al., 2016), la cirugía (Nguyen et al., 2023), la rehabilitación (Cianchetti et al., 2018), la inspección y exploración (Li et al., 2023) o la manipulación, tanto a nivel humano (Terrile et al., 2021; Montenegro-Bravo et al., 2023),

como de precisión (Chi et al., 2024). Y es que estos dispositivos, inspirados en la flexibilidad y adaptabilidad inherentes a los organismos naturales, ofrecen ventajas como su bajo coste, su adaptabilidad a superficies irregulares, su capacidad de soportar impactos o una interacción segura con los seres humanos (Wang et al., 2022). No obstante, estas ventajas no están exentas de retos, especialmente en lo que a modelado y control se refiere.

En concreto, uno de los problemas aún abiertos es el de resolver su modelo cinemático inverso, que relaciona la posición alcanzada por el elemento terminal del robot con sus variables de control –presiones de hinchado, longitudes de los cables, tensiones de mando...–. Más allá de la determinación de la relación física entre la posición y dichas variables, la dificultad reside en que estos manipuladores son habitualmente estructuras continuas de infinitos grados de libertad (Trivedi et al., 2008) o, en el mejor de los casos, una sucesión de módulos que, en conjunto, forman un mecanismo redundante (Cerrillo et al., 2022), con varias soluciones posibles.

Una primera forma de abordar el problema es el recurrir a técnicas de aprendizaje, como redes neuronales (Alkhodary and Gur, 2022) o aprendizaje por refuerzo (Centurelli et al., 2022). En ellos, que el estado del robot no viene dado únicamente por su posición final, sino también por las posiciones de sus distintos elementos intermedios. Con el fin de reducir la cantidad de datos de entrenamiento necesarios, algunos autores han optado por eliminar redundancias (Wang et al., 2024) o aprender, en lugar del modelo completo, la matriz jacobiana (Fang et al., 2022) u otros parámetros característicos de su cinemática, como se hace en (Yang et al., 2022). Si bien esta solución tiende, además, a ser más interpretable, su escalabilidad sigue siendo un aspecto comprometido: el tamaño de la jacobiana, por ejemplo, crece cuadráticamente al hacerlo el número de actuadores del manipulador. En tareas complejas, como la manipulación (Bhagat et al., 2019), da muy buen resultado el aprendizaje por imitación (Nazeer et al., 2024). El número de muestras necesarias es reducido, pero estas se toman sobre el mundo real, con el coste en tiempo que esto implica.

Frente a esto, el empleo de técnicas basadas en optimización precisan únicamente de disponer de un modelo cinemático directo del robot. Dos son aquí los enfoques posibles. Por un lado, se han adaptado a robots blandos algoritmos propios de manipuladores hiperredundantes discretos, como CCD (Zhang et al., 2021) o FABRIK (Chiang et al., 2021). Su principal ventaja es su velocidad pero no permiten fijar de manera sencilla la posición de los elementos intermedios del sistema..

Por otro lado, un amplio abanico de autores ha empleado técnicas de minimización convencionales sobre modelos más o menos precisos del robot. Algunos ejemplos son (Keyvanara et al., 2023), donde se utiliza programación cuadrática sobre un modelo geométrico, (Liu et al., 2023), en el que se utiliza un filtro de partículas sobre un robot caracterizado con PCC, (Bern et al., 2020), que recurre a una red neuronal y a un algoritmo BFGS o (Bern and Rus, 2021) donde se usa el método de Newton sobre un modelo de elementos finitos. De estos, si bien alguno permite trabajar imponiendo restricciones en el espacio cartesiano, ninguno da la posibilidad de realizar un control de forma del robot, es decir, de elegir una de entre las

múltiples soluciones del modelo cinemático inverso.

Este es el enfoque seguido en este trabajo: generar un algoritmo de optimización en base a un modelo cinemático directo previamente obtenido. Sus contribuciones, por ende, pueden resumirse en los siguientes puntos:

- Por un lado, se ha desarrollado un modelo cinemático inverso, basado en algoritmos genéticos, válido para robots blandos redundantes y capaz de dar soluciones precisas a una velocidad adecuada para el movimiento del robot. El algoritmo se valida sobre el gemelo digital de un robot concreto, aunque su uso es válido para cualquier robot blando de múltiples módulos.
- Por otro lado, se extiende el algoritmo para permitirle al operador del sistema robótico fijar no únicamente la posición del extremo final del robot sino también la de alguno o todos sus módulos intermedios, controlando así la pose que se quiere que adopte el manipulador.

El artículo se ha estructurado de la siguiente manera: en primer lugar, en la Sección 2, se presenta el manipulador sobre el que se ha implementado el algoritmo, que se detalla en el Epígrafe 3. Los experimentos llevados a cabo y los resultados obtenidos se detallan en el Apartado 4. Las conclusiones y líneas futuras del trabajo se han colocado en la Sección 5.

2. Herramientas utilizadas

2.1. El robot neumático PAUL

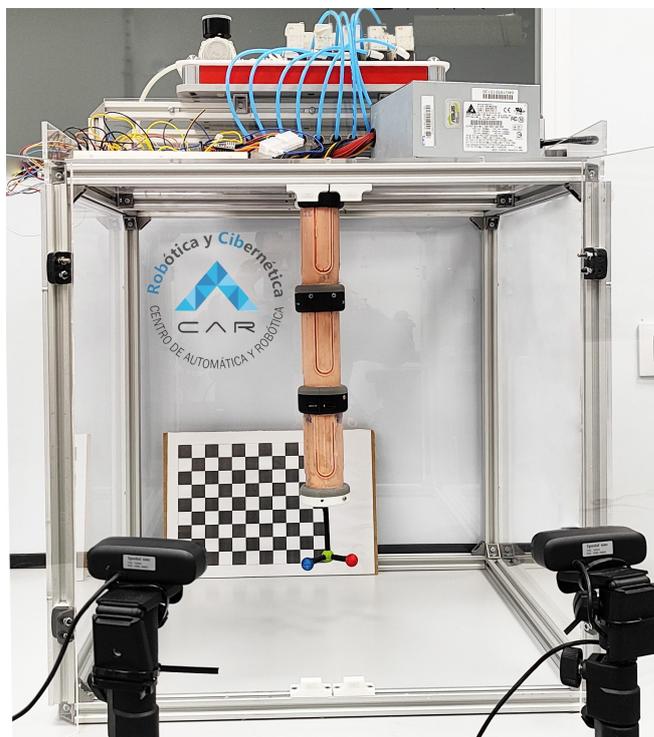


Figura 1: Robot neumático PAUL

Si bien la metodología aquí presentada es válida para cualquier manipulador blando redundante, su desarrollo se ha hecho sobre PAUL (acrónimo de *Pneumatic Articulated Ultra-soft Limb*, un robot neumático desarrollado en el seno del

grupo de Robótica y Cibernética del Centro de Automática y Robótica de la Universidad Politécnica de Madrid (García-Samartín et al., 2024b) que puede observarse en la Figura 1.

PAUL está formado por módulos –o segmentos– de silicona, cada uno de ellos con tres vejigas o cavidades que, al hincharse, producen el movimiento del robot. Cada módulo, por tanto, posee 3 entradas de aire que le dotan de 3 grados de libertad. Se ha trabajado en una configuración de 3 módulos, pues en esta se tiene el número máximo de grados de libertad posibles en el espacio cartesiano –5, pues no se puede lograr torsión de ninguna forma– con 9 entradas, existiendo, por ende, una notable redundancia.

2.2. Modelo cinemático directo

El modelado cinemático del robot se realizó en un trabajo previo (García-Samartín et al., 2024a) haciendo uso de una red neuronal *feedforward* de 25 neuronas entrenada con unas 500 muestras por segmento. Este número relativamente reducido –para la cinemática inversa harían faltan miles de muestras– hace la tarea asequible y segura frente a posibles pinchazos. Esta solución es más precisa que el empleo de simplificaciones como PCC, a la vez que menos compleja que el modelado por elementos finitos, donde la determinación de parámetros de materiales no lineales, como la silicona, sigue siendo un problema.

El modelo recibe como entrada los tiempos de hinchado de las 9 vejigas del robot y devuelve la posición alcanzada por el extremo de cada uno de los segmentos. Con el fin de facilitar el proceso de pruebas, se desarrolló un simulador en MATLAB a partir de dicha cinemática directa, que puede verse en la Figura 2.

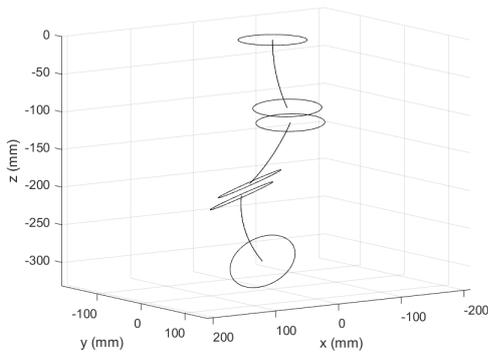


Figura 2: Simulador desarrollado

3. Algoritmo Genético desarrollado

3.1. Introducción

Los algoritmos genéticos son una metodología de optimización bioinspirada en el que las distintas soluciones propuestas, inicialmente de forma aleatoria, se van combinando entre sí hasta alcanzar un resultado suficientemente bueno.

En concreto, tras una generación al azar de individuos o cromosomas, se ejecuta en un bucle una sucesión de cuatro etapas: evaluación en base a una función objetivo de los distintos individuos existentes en ese momento, selección de los

mejores como "padres" de los individuos de la siguiente generación, cruce y combinación de estos para formar nuevos cromosomas y mutación de los mismos. Se han aplicado a un amplio número de campos, estando el modelado cinemático de robots rígidos entre ellos (Lee and Chang, 2021; García-Samartín and Barrientos, 2023).

Una de las grandes desventajas ligadas al uso de algoritmos genéticos es su baja velocidad de convergencia, motivada principalmente por la aleatoriedad inherente a los mismos. Debido a que el algoritmo de cinemática inversa debe ser capaz de funcionar en un bucle de control, el primer objetivo de su diseño fue conseguir una velocidad lo más alta posible sacrificando, en cierto modo, la capacidad exploradora del mismo. Es decir, se primó conseguir una solución suficientemente buena lo antes posible sobre una solución excelente un poco más tarde.

3.2. Características del algoritmo

Se optó por una población de 50 individuos, lo más reducida posible para aumentar la velocidad, pero suficiente para, al recombinarlos entre ellos, lograr alcanzar la solución. Los detalles de este y otros hiperparámetros, como porcentaje de elitismo y ratio de mutación, se encuentran detallados en el repositorio enlazado en el Apéndice A Cada individuo i se codifica con un vector $\mathbf{q}_i = [q_{i1}, \dots, q_{i3N}] \in \mathbb{R}^{3N}$ (siendo N el número de módulos de PAUL, que se mantiene en 3 todo el artículo) en el que cada una de sus componentes q_{ij} representa el tiempo de hinchado de la j -ésima vejiga de PAUL. Cada uno de esos valores es un número real, obtenido originalmente al azar de una distribución uniforme entre 0 y 1000 ms.

Dentro del bucle, en primer lugar, los individuos son evaluados mediante la expresión de la Ecuación 1:

$$f_i = \|\mathbf{x}_i - \mathbf{x}^d\| \quad (1)$$

en la que \mathbf{x}^d es la posición que se desea que alcance el elemento terminal y \mathbf{x}_i representa el extremo final del robot cuando se le hincha unos tiempos \mathbf{q}_i . Esta se obtiene introduciendo dichos tiempos en la red neuronal que juega el papel de modelo cinemático directo.

Con el fin de conseguir la mayor presión selectiva posible, el proceso de selección de individuos se hace ordenándolos en base a los valores f_i de su función objetivo y tomando como padres a los 10 mejores. Dichos individuos, además, pasan automáticamente a la siguiente generación, con el fin de asegurar que ninguna generación conduce a peores resultados que la anterior.

El cruce se realiza tomando al azar a dos padres distintos entre sí y asignando a cada componente del elemento hijo la media entre los correspondientes tiempos de hinchado de sus padres. Con el fin de compensar el alto elitismo del proceso de selección, se estableció que una probabilidad de cruce de 1: todo par de padre escogido engendra un hijo distinto a ellos.

Finalmente, en la etapa de mutación se multiplica cada tiempo de hinchado de cada uno de los individuos por un valor extraído al azar de una distribución uniforme en el intervalo (0,9; 1,1). La probabilidad de mutación se fijó en un 70%. Si, como consecuencia de este proceso, algún individuo presenta alguna componente con un valor superior a 1000 ms, se satura a este valor.

El algoritmo termina pasadas 20 iteraciones o, si tras la evaluación, algún individuo presenta un valor de función objetivo inferior a 1 mm. Cuando no ocurre, el algoritmo devuelve la mejor solución encontrada, así como el error asociado, lo que permite decidir si es preferible guardarla –es decir, fijar tolerancia a la salida del algoritmo algo mayor, solución habitual– o volver a ejecutar el algoritmo. El algoritmo fue codificado en MATLAB, dado que todos los programas del robot habían sido desarrollados en este entorno.

3.3. Control de forma

El segundo objetivo del trabajo era que el algoritmo desarrollado permitiese, además, una gestión de las redundancias del manipulador, permitiendo desde fuera –ya sea a un operador o a un sistema de control a más alto nivel– fijar la forma que el brazo adopte. Esta gestión debía ser en todo momento voluntaria, es decir, que, en caso de que no se especificase ninguna instrucción sobre las posiciones intermedias del robot, este debería poder adoptar alguna de ellas.

Para poder lograr esta cinemática inversa con control de forma incorporado manteniendo la compatibilidad con lo desarrollado anteriormente, cuyos buenos resultados se presentan en la Sección 4, se buscó mantener la misma estructura de algoritmo e introducir sobre él el menor número de cambios.

En concreto, en esta nueva versión del algoritmo, la posición deseada \mathbf{x}^d no contiene únicamente las coordenadas (x, y, z) del elemento terminal sino también la de todos los puntos intermedios que se desea fijar. Del mismo modo, si bien cada individuo sigue codificándose con $\mathbf{q} \in \mathbb{R}^{3N}$, al calcular su modelo cinemático directo no se almacena solo el punto final sino también las posiciones de las articulaciones intermedias a las que se han fijado en \mathbf{x}^d . Pueden fijarse tantas posiciones como se desee y estas no tienen por qué ser consecutivas (puede establecerse posición del final del primer segmento, dejar libre el segundo y fijarse el punto final).

Así, si se desean fijar, además de la posición final, n posiciones intermedias (con $n < N$, evidentemente), se tiene que $\mathbf{x}^d = [\mathbf{x}_{int}^d | \mathbf{x}_{fin}^d]$ y $\mathbf{x}_i = [\mathbf{x}_{i,int} | \mathbf{x}_{i,fin}]$ con $\mathbf{x}_{int}^d, \mathbf{x}_{i,int} \in \mathbb{R}^{3n}$ y $\mathbf{x}_{fin}^d, \mathbf{x}_{i,fin} \in \mathbb{R}^3$.

La función objetivo se define entonces siguiendo la Ecuación 2:

$$f_i = \|\mathbf{x}_{i,fin} - \mathbf{x}_{fin}^d\| + \alpha \|\mathbf{x}_{i,int} - \mathbf{x}_{int}^d\| \quad (2)$$

siendo $\alpha \in (0, 1)$ un parámetro encargado de matizar la importancia de alcanzar las posiciones intermedias, ya que el objetivo, más que alcanzar una serie de punto es que el robot alcance la posición final con una forma dada. De manera experimental se fijó $\alpha = 0,3$, que se vio adecuado fijando tanto 1 como 2 posiciones intermedias. Se estima, no obstante, que este valor tenga que ser rebajado si se trabaja con robots de un mayor número de segmentos.

4. Resultados

El objetivo de los distintos experimentos llevados a cabo fue conocer la precisión alcanzada por el modelo cinemático aquí planteado, considerando la situación de ausencia de control de forma, así como la fijación de la posición de uno y de dos de los segmentos intermedios. Al no tenerse en cuenta efectos dinámicos, todos los experimentos fueron llevados a

cabo sobre el simulador presentado en la Sección 2, cuyo buen funcionamiento ha quedado demostrado en trabajos previos.

Las pruebas consistieron en calcular la cinemática inversa de 300 puntos situados elegidos al azar dentro del espacio de trabajo del robot. Para asegurar esta condición, se generaron aleatoriamente 300 combinaciones de $\mathbf{q}_i \in \mathbb{R}^9$ y se calculó su modelo cinemático directo. El resultado de esta operación fue dado como entrada \mathbf{x}^d al algoritmo genético. Se midió a la salida el error entre la solución obtenida y la deseada, el número de iteraciones necesario para alcanzarla, el porcentaje de convergencia temprana así como el tiempo total de ejecución.

Este mismo experimento se realizó tanto para el caso en el que se fija únicamente la posición del elemento terminal del robot, como para aquellos en los que se indica la posición de uno o de los dos segmentos restantes. Todas las pruebas se llevaron a cabo sobre un i7 de 12ª generación y 2,1 GHz.

La Tabla 1 resume los distintos resultados obtenidos. Se indican para el tiempo y el error la mediana ya que, al tratarse de una distribución fuertemente no simétrica, es un valor mucho más representativo. Estos pasarán a comentarse y discutirse a lo largo de la sección.

Tabla 1: Resumen de los resultados más relevantes de los experimentos

	0 seg. fijos	1 seg. fijo	2 seg. fijos
Error mediano	0,90 mm	1,72 mm	1,76 mm
Conv. temprana	67.33 %	30.00 %	27.67 %
Media iteraciones	12.44	19.75	19.96
Tiempo mediano	13,02 s	20,99 s	18,31 s

4.1. Error

En las tres situaciones se ha medido el error como la diferencia entre el punto final alcanzado por el robot y el realmente deseado. Esto se ha hecho así porque el objetivo, como se ha comentado, ha sido desarrollar un algoritmo de cinemática inversa que permita además un control de forma, mas el primer deseo al trabajar con un manipulador es que su elemento terminal llegue al lugar deseado.

Puede decirse en un primero momento que las tres situaciones estudiadas se consiguen, a la vista de la Tabla 1, resultados excelentes para lo que a robótica blanda se refiere, no solo a nivel absoluto sino también relativo: dado que se está trabajando con un robot de 36 cm de largo, estos errores suponen en todos los casos una precisión superior al 99.5 % de su longitud, valores mucho más altos a los encontrados en la literatura, como se discute en (García-Samartín et al., 2024a).

La tabla muestra, sin embargo, una clara diferencia de precisión en función de si se establece o no control de forma. Este comportamiento entra dentro de lo esperable, ya que el hecho de imponer condiciones secundarias hace que el algoritmo se centre en cubrirlas todas. Si bien reducir el valor de α puede contribuir a paliar este problema, se observó cómo, en algunas ocasiones, esto no lograba que se obtuviese la forma del robot deseada.

La diferencia es también latente cuando se analiza la Figura 3: mientras que la situación sin control de forma tiene más de dos tercios de sus muestras con un error inferior al milímetro, los otros dos casos arrojan una distribución mucho más concentrada. Sin embargo, sí se puede decir, desde el punto de vista cualitativo –y a falta de comprobaciones de

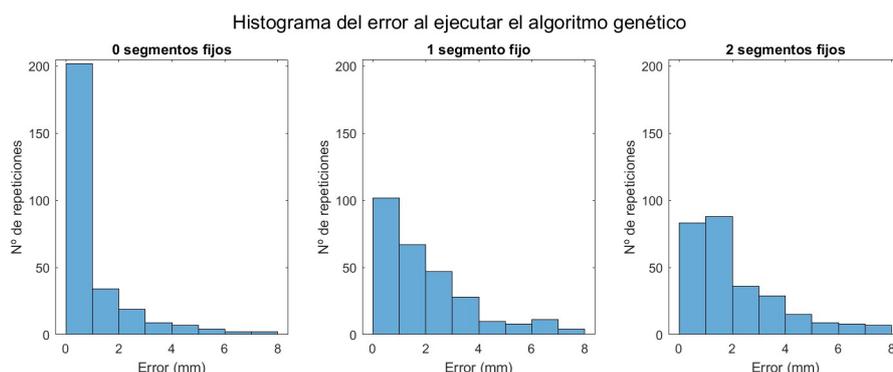


Figura 3: Histograma del error al ejecutar el algoritmo en las tres situaciones: (a) sin control de forma, (b) fijando uno de los segmentos y (c) con los dos segmentos fijos. En el eje vertical se muestran el número de repeticiones y en el horizontal, el error, en milímetros.

lo que pasaría al aumentar considerablemente el número de segmentos— que no es excesivamente significativo fijar más o menos módulos una vez se ha impuesto una primera restricción.

4.2. Convergencia temprana y número de iteraciones

La diferencia presentada en el apartado anterior se hace aquí aún más palpable. Mientras que un 67 % de las veces el algoritmo genético converge antes de alcanzar el tope de iteraciones cuando no hay control de forma, lo hace apenas un 30 % de las veces cuando se impone este. Del mismo modo, puede verse como se pasa de un número medio de iteraciones de 12 a prácticamente 20, lo que indica que fijar módulos intermedios hace muy difícil al algoritmo lograr los objetivos de *fitness* propuestos. Debe tenerse en cuenta que el valor de tolerancia para salir del algoritmo no se relaja al implementar control de forma, lo que explica en parte este comportamiento.

Experimentalmente se ha comprobado que, cuando existe control de forma, el algoritmo tiende a encontrar muy rápido soluciones cercanas a los 5 mm de error pero le cuesta luego un elevado número de iteraciones terminar de afinar. Esto concuerda con los errores observados en la Figura 3.

4.3. Tiempo de ejecución

Como cabe esperar, número de iteraciones y tiempo de ejecución están ampliamente relacionados. Así, cuando no se controla la forma el algoritmo da una respuesta en 13 s, mientras que, cuando no es así, tarda prácticamente 20 s en hacerlo. La Figura 4 refleja gráficamente este aspecto.

Para discutir si este tiempo es o no adecuado, no debe olvidarse que, por las propias características de diseño y materiales, un robot blando tarda mucho más en alcanzar un punto que uno rígido. En algunas referencias de la literatura (Almanzor et al., 2023), se dan por válidos tiempos de algoritmo de entre 30 y 60 segundos. Y es que, en un espacio de trabajo de este tamaño, con la sobreoscilación además inherente a PAUL, es perfectamente normal que tarde alrededor de 5 o 6 segundos en estabilizarse en su destino.

En tareas de planificación de trayectorias, además, este tiempo puede mejorarse suponiendo que la siguiente solución alcanzada va a ser cercana al punto actual del robot. Para ello una alternativa es generar, en la etapa inicial, valores de q_i no uniformemente distribuidos a lo largo de todo el espacio de trabajo, sino centrados en torno a dicha posición actual.

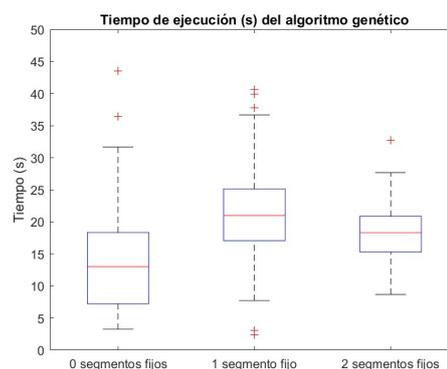


Figura 4: Diagrama de caja que compara el tiempo de ejecución algoritmo genético al fijar la posición de 1, 2 o de ningún segmento intermedio.

5. Conclusiones y líneas futuras

En este trabajo se ha empleado un algoritmo genético para conseguir el modelado cinemático inverso con control de forma de un robot blando. Aunque la metodología es válida para cualquier tipo de robot continuo, se ha aplicado sobre un manipulador neumático modular previamente desarrollado. Frente a las técnicas basadas en aprendizaje, el uso de algoritmos de optimización, como el aquí presentado, pueden lograr hacer frente a las redundancias existentes sin la necesidad de trabajar con una cantidad de datos en muchas ocasiones prácticamente imposible de asumir. De entre todos los modelos estudiados en la literatura, ninguno de ellos consigue un control de forma como el aquí presentado.

Los parámetros del algoritmo se han seleccionado con el fin de provocar una convergencia lo más temprana posible para poder operar lo más cerca posible de unas condiciones de tiempo real. Ello no ha implicado sacrificar la precisión: al contrario, se obtienen errores de entre 0,9 y 1,76 mm, que suponen entre un 0,25 y un 0,48 % de la longitud del robot. Si bien el tiempo de ejecución es mejorable, se adapta a lo esperable en robótica blanda y puede ser fácilmente optimizado en situaciones de planificación de trayectorias continuas.

Los excelentes resultados obtenidos abren la puerta a varias aplicaciones de este método. La más clara de todas ellas es la teleoperación de estos robots para inspección entornos complejos, como pueden ser tuberías o, con los materiales de diseño adecuados, operaciones quirúrgicas, situaciones ambas

en las que un control de forma es necesario. Dicha teleoperación puede realizarse incluso, como se ha hecho en otro tipo de robots, mediante realidad virtual, donde de manera natural se fijan el punto de destino y cuantos puntos intermedios se deseen, así como la importancia de que se acerque más o menos a uno u otro, situaciones, cubiertas por este método.

Agradecimientos

Este trabajo ha sido realizado gracias a las “Ayudas para contratos predoctorales para la realización del doctorado con mención internacional en sus escuelas, facultad, centros e institutos de I+D+i”, financiadas por el “Programa Propio I+D+i 2022 de la Universidad Politécnica de Madrid” y es parte del proyecto “CollaborativE Search And Rescue robots (CESAR)” (PID2022-142129OB-I00) financiado por MCIN/AEI/10.13039/501100011033 y “FEDER, UE”

Apéndice A. Material adicional

Repositorio con el código del proyecto: <https://github.com/Robcib-GIT/PAUL>.

En el *Readme*, se incluye una sección “Genetic Algorithms” en la que una tabla justifica los hiperparámetros escogidos.

Vídeo del sistema en funcionamiento: <https://www.youtube.com/watch?v=Fus6LwPv0w4>.

Referencias

- Alkhodary, A., Gur, B., 2022. Kinematics Transformer : Solving The Inverse Modeling Problem of Soft Robots using Transformers. unpublished.
- Almanzor, E., Ye, F., Shi, J., Thuruthel, T. G., Wurdemann, H. A., Iida, F., 2023. Static Shape Control of Soft Continuum Robots Using Deep Visual Inverse Kinematic Models. *IEEE Transactions on Robotics* 39 (4), 2973–2988.
DOI: 10.1109/TR0.2023.3275375
- Bern, J. M., Rus, D., 2021. Soft IK with stiffness control. 2021 IEEE 4th International Conference on Soft Robotics, *RoboSoft 2021*, 465–471.
DOI: 10.1109/RoboSoft51838.2021.9479195
- Bern, J. M., Schneider, Y., Banzet, P., Kumar, N., Coros, S., 2020. Soft Robot Control with a Learned Differentiable Model. 2020 3rd IEEE International Conference on Soft Robotics, *RoboSoft 2020*, 417–423.
DOI: 10.1109/RoboSoft48309.2020.9116011
- Bhagat, S., Banerjee, H., Tse, Z. T. H., Ren, H., 2019. Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges. *Robotics* 8 (1), 1–36.
DOI: 10.3390/robotics8010004
- Centurelli, A., Arleo, L., Rizzo, A., Tolu, S., Laschi, C., Falotico, E., 2022. Closed-Loop Dynamic Control of a Soft Manipulator Using Deep Reinforcement Learning. *IEEE Robotics and Automation Letters* 7 (2), 4741–4748.
DOI: 10.1109/LRA.2022.3146903
- Cerrillo, D., Barrientos, A., Del Cerro, J., 2022. Kinematic Modelling for Hyper-Redundant Robots—A Structured Guide. *Mathematics* 10 (16).
DOI: 10.3390/math10162891
- Chi, Y., Zhao, Y., Hong, Y., Li, Y., Yin, J., 2024. A Perspective on Miniature Soft Robotics: Actuation, Fabrication, Control, and Applications. *Advanced Intelligent Systems* 6 (2).
DOI: 10.1002/aisy.202300063
- Chiang, S. S., Yang, H., Skorina, E., Onal, C. D., 2021. SLInKi: State Lattice based Inverse Kinematics - A Fast, Accurate, and Flexible IK Solver for Soft Continuum Robot Manipulators. In: *IEEE International Conference on Automation Science and Engineering*. IEEE, pp. 1871–1877.
DOI: 10.1109/CASE49439.2021.9551686
- Cianchetti, M., Laschi, C., Menciassi, A., Dario, P., 2018. Biomedical applications of soft robotics. *Nature Reviews Materials* 3 (6), 143–153.
DOI: 10.1038/s41578-018-0022-y
- Fang, G., Tian, Y., Yang, Z. X., Geraedts, J. M., Wang, C. C., 2022. Efficient Jacobian-Based Inverse Kinematics With Sim-to-Real Transfer of Soft Robots by Learning. *IEEE/ASME Transactions on Mechatronics*, 1–11.
DOI: 10.1109/TMECH.2022.3178303
- García-Samartín, J. F., Barrientos, A., 2023. Kinematic Modelling of a 3RRR Planar Parallel Robot Using Genetic Algorithms and Neural Networks. *Machines* 11 (10), 1–26.
DOI: 10.3390/machines11100952
- García-Samartín, J. F., Molina-Gómez, R., Barrientos, A., 2024a. Model-Free Control of a Soft Pneumatic Segment. *Biomimetics* 9 (127).
DOI: <https://doi.org/10.3390/biomimetics9030127>
- García-Samartín, J. F., Rieker, A., Barrientos, A., 2024b. Design, Manufacturing, and Open-Loop Control of a Soft Pneumatic Arm. *Actuators* 13 (1).
DOI: 10.3390/act13010036
- Keyvanara, M., Goshtasbi, A., Kuling, I. A., 2023. A Geometric Approach towards Inverse Kinematics of Soft Extensible Pneumatic Actuators Intended for Trajectory Tracking. *Sensors* 23 (15), 1–16.
DOI: 10.3390/s23156882
- Lee, C.-T., Chang, J.-Y. J., 2021. A Workspace-Analysis-Based Genetic Algorithm for Solving Inverse Kinematics of a Multi-Fingered Anthropomorphic Hand. *Applied Sciences* 11 (6).
DOI: 10.3390/app11062668
- Li, G., Wong, T. W., Shih, B., Guo, C., Wang, L., Liu, J., Wang, T., Liu, X., Yan, J., Wu, B., Yu, F., Chen, Y., Liang, Y., Xue, Y., Wang, C., He, S., Wen, L., Tolley, M. T., Zhang, A. M., Laschi, C., Li, T., 2023. Bioinspired soft robots for deep-sea exploration. *Nature Communications* 14 (1), 1–10.
DOI: 10.1038/s41467-023-42882-3
- Liu, H., Liu, M., Jiang, Y., Zhang, X., 2023. Research on obstacle avoidance planning of soft robotic arm based on the idea of cutting-edge growth. In: *2023 2nd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC)*. IEEE, pp. 35–40.
DOI: 10.1109/RAIIC59453.2023.10281195
- Manti, M., Pratesi, A., Falotico, E., Cianchetti, M., Laschi, C., 2016. Soft assistive robot for personal care of elderly people. *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics 2016-July*, 833–838.
DOI: 10.1109/BIOROB.2016.7523731
- Montenegro-Bravo, J. S., Ruiz-Flórez, J. D., Romero-Ante, J. D., Manrique-Córdoba, J., Vivas Albán, O. A., Sabater-Navarro, J. M., jul. 2023. Generador 3d de trayectorias libres de colisiones para un manipulador ur3e con pinza blanda. *Revista Iberoamericana de Automática e Informática industrial* 21 (1), 52–61.
DOI: 10.4995/riai.2023.19332
- Nazeer, M. S., Laschi, C., Falotico, E., 2024. RL-based Adaptive Controller for High Precision Reaching in a Soft Robot Arm. *IEEE Transactions on Robotics* 40, 2498–2512.
DOI: 10.1109/TR0.2024.3381558
- Nguyen, C. C., Thai, M. T., Hoang, T. T., Davies, J., Phan, P. T., Zhu, K., Wu, L., Brodie, M. A., Tsai, D., Ha, Q. P., Phan, H. P., Lovell, N. H., Nho Do, T., 2023. Development of a soft robotic catheter for vascular intervention surgery. *Sensors and Actuators A: Physical* 357 (November 2022), 114380.
DOI: 10.1016/j.sna.2023.114380
- Terrile, S., Argüelles, M., Barrientos, A., 2021. Comparison of Different Technologies for Soft Robotics Grippers. *Sensors* 21 (9).
DOI: 10.3390/s21093253
- Trivedi, D., Rahn, C. D., Kier, W. M., Walker, I. D., 2008. Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics* 5 (3), 99–117.
DOI: 10.1080/11762320802557865
- Wang, J., Zhao, Y., Zhang, X., Li, Z., Yao, W., 2024. Model-Free Intelligent Control for Space Soft Robotic Manipulators. *Space: Science & Technology* 4, 1–13.
DOI: 10.34133/space.0120
- Wang, P., Tang, Z., Xin, W., Xie, Z., Guo, S., Laschi, C., 2022. Design and Experimental Characterization of a Push-Pull Flexible Rod-Driven Soft-Bodied Robot. *IEEE Robotics and Automation Letters* 7 (4), 1–8.
- Yang, C., Xu, H., Li, X., Yu, F., 2022. Kinematic modeling and solution of rigid-flexible and variable-diameter underwater continuous manipulator with load. *Robotica* 10, 1020–1035.
DOI: 10.1017/S0263574721000989
- Zhang, Z., Wang, S., Meng, D., Wang, X., Liang, B., 2021. Soft-CCD Algorithm for Inverse Kinematics of Soft Continuum Manipulators. *IEEE International Conference on Intelligent Robots and Systems*, 639–644.
DOI: 10.1109/IR0551168.2021.9635921