

Optimización multiobjetivo de trayectorias en manipuladores industriales mediante el algoritmo MOPSO

Peñacoba Yagüe, Mario^{a,*}, Sierra García, Jesús Enrique^a, Santos Peñas, Matilde^b

^a Departamento de Digitalización, Universidad de Burgos, Campus Río Vena, Avda. Cantabria, s/n, 09006 Burgos, España.

^b Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid, Madrid, España.

To cite this article: Peñacoba Yagüe, Mario, Sierra García, Jesús Enrique, Santos Peñas, Matilde, 2025. Multi-objective Trajectory Optimization in Industrial Manipulators Using the MOPSO Algorithm. XX Simposio CEA de Control Inteligente, Huelva (Spain), 2025.

Resumen

Este trabajo presenta un primer enfoque hacia una metodología multiobjetivo para la optimización de trayectorias en manipuladores industriales, basada en el algoritmo *Multi-Objective Particle Swarm Optimization* (MOPSO). La propuesta se implementa en un entorno de simulación desarrollado en MATLAB que incorpora un modelo dinámico del robot colaborativo ABB CRB15000-95, teniendo en cuenta parámetros físicos como la masa, el volumen y la inercia de sus eslabones. Además, se consideran las restricciones geométricas impuestas por los obstáculos del caso de uso, permitiendo detectar colisiones tanto con el entorno como con el propio robot. La estrategia de optimización se basa en dos funciones de coste para optimizar simultáneamente métricas físicas relevantes como el tiempo total de ejecución y el esfuerzo articular, estimado a partir de los pares medios y máximos aplicados. Cada función de coste tiene una estructura escalonada en dos fases: en la primera, se descartan las trayectorias que presentan colisiones, garantizando la viabilidad geométrica; en la segunda, se optimizan las métricas físicas. A través de la ejecución del algoritmo, se obtiene un frente de Pareto que recoge las soluciones más eficientes en términos de tiempo y esfuerzo, permitiendo identificar trayectorias óptimas desde una perspectiva multiobjetivo. Los resultados evidencian que el enfoque basado en MOPSO permite generar trayectorias suaves, físicamente eficientes y libres de colisiones, adaptadas a los requisitos operativos de entornos industriales exigentes.

Palabras clave: Optimización Multiobjetivo, Generación de Trayectorias, Robótica Industrial, Detección de Colisiones, Frente de Pareto, Optimización Multiobjetivo por Enjambre de Partículas.

Multi-objective Trajectory Optimization in Industrial Manipulators Using the MOPSO Algorithm

Abstract

This work presents a first approach toward a multi-objective methodology for trajectory optimization in industrial manipulators, based on the Multi-Objective Particle Swarm Optimization (MOPSO) algorithm. The proposed strategy is implemented in a MATLAB-based simulation environment that incorporates a dynamic model of the ABB CRB15000-95 collaborative robot, accounting for physical parameters such as the mass, volume, and inertia of its links. In addition, geometric constraints imposed by obstacles in the use case are considered, enabling the detection of collisions both with the environment and with the robot itself. The optimization strategy relies on two cost functions to simultaneously optimize relevant physical metrics, namely total execution time and joint effort, estimated through average and peak torques. Each cost function follows a two-phase hierarchical structure: in the first phase, trajectories with collisions are discarded, ensuring geometric feasibility; in the second phase, the physical metrics are optimized. By executing the algorithm, a Pareto front is obtained, representing the most efficient solutions in terms of time and effort, and enabling the identification of optimal trajectories from a multi-objective perspective. The results demonstrate that the MOPSO-based approach is capable of generating smooth, physically efficient, and collision-free trajectories tailored to the operational requirements of demanding industrial environments.

Keywords: Multi-objective Optimization, Trajectory Generation, Industrial Robotics, Collision Detection, Pareto Front, Multi-Objective Particle Swarm Optimization.

1. Introducción

La planificación eficiente de trayectorias es un elemento clave en la programación de manipuladores industriales, especialmente en escenarios donde es necesario garantizar

tanto la viabilidad geométrica como la eficiencia física del movimiento. Aunque los métodos clásicos de planificación han permitido resolver problemas cinemáticos de forma robusta, su principal limitación radica en la falta de consideración de variables dinámicas como el esfuerzo

*Autor para correspondencia: mpenacoba@ubu.es

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

articular o el tiempo de ejecución real, fundamentales en entornos industriales con altas exigencias operativas.

En los últimos años, los algoritmos de optimización inspirados en la inteligencia de enjambre han ganado popularidad como alternativa eficaz para abordar problemas complejos de planificación. En particular, el algoritmo Multi-Objective Particle Swarm Optimization (MOPSO) se ha consolidado como una técnica versátil que permite encontrar soluciones óptimas desde una perspectiva multiobjetivo, generando frentes de Pareto que ofrecen un equilibrio entre distintos criterios de evaluación, sin requerir su ponderación previa (Reyes et al, 2025).

Este trabajo propone una metodología de optimización de trayectorias basada en MOPSO, centrada en minimizar el tiempo de ejecución y el esfuerzo articular, mientras se garantiza la ausencia de colisiones con el entorno o entre eslabones del manipulador (Wang et al, 2024). La elección de MOPSO frente a otros enfoques como NSGA-II o MOEA/D se justifica por su bajo número de parámetros, su buena capacidad de exploración del espacio de soluciones y su eficiencia computacional (Duan et al, 2024), (Zhou et al, 2024). Para ello, se emplea un entorno de simulación desarrollado en MATLAB que integra el modelo dinámico del robot colaborativo ABB CRB15000-95, incluyendo masa, volumen e inercia de sus eslabones. Además, se consideran restricciones geométricas derivadas de obstáculos presentes en el caso de uso (Hao et al, 2024). La evaluación de cada trayectoria se realiza mediante una función de coste escalonada en dos fases: una primera en la que se penalizan fuertemente las trayectorias que generan colisiones, y una segunda que optimiza las soluciones viables según las métricas físicas seleccionadas, si cumplido previamente las restricciones de la primera fase (Peñacoba et al, 2023).

Durante la ejecución del algoritmo, se registra la evolución del proceso, incluyendo todas las evaluaciones realizadas, las mejores soluciones encontradas y el frente de Pareto en tiempo real. Esto permite analizar el comportamiento del sistema, garantizar la reproducibilidad de los resultados y reanudar la optimización en caso de interrupción (Dalai et al, 2024).

Este trabajo se enmarca dentro del proyecto europeo MANiBOT, cuyo objetivo es desarrollar soluciones avanzadas de manipulación bimanual colaborativa para entornos industriales y logísticos altamente exigentes. En particular, la metodología propuesta se ha validado en uno de los casos de uso definidos en el proyecto: un escenario de manipulación en estanterías de supermercado, donde el robot debe generar trayectorias seguras, eficientes y adaptadas a restricciones físicas y espaciales propias del entorno (MANiBOT, 2023).

El resto del artículo se organiza como sigue: en la Sección 2 se describe el funcionamiento del algoritmo MOPSO y su configuración para el problema tratado. La Sección 3 introduce el problema de optimización. En la Sección 4 se presenta el modelado del sistema. La Sección 5 muestra los resultados experimentales y su análisis. Finalmente, en la Sección 6 se discuten las conclusiones extraídas y se proponen líneas de trabajo futuro.

2. Multi-Objective Particle Swarm Optimization

En este trabajo, la optimización de trayectorias del manipulador industrial se aborda mediante el algoritmo Multi-

Objective Particle Swarm Optimization (MOPSO), una extensión del algoritmo PSO clásico orientada a resolver problemas con múltiples objetivos simultáneos (Ghafour, K, 2024), (Yang et al, 2025). Inspirado en el comportamiento colectivo de organismos naturales como bandadas de aves o cardúmenes de peces, el MOPSO mantiene una población de partículas (trayectorias candidatas) que se actualizan iterativamente en función de su experiencia individual y de un conjunto de soluciones no dominadas que conforman el frente de Pareto (Agarwal et al., 2024).

La evolución de cada partícula se define por dos ecuaciones fundamentales. En primer lugar, su nueva posición se actualiza según la Ecuación 1 (Peñacoba et al, 2024).

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (1)$$

Donde x_i^t y v_i^t representan respectivamente la posición y la velocidad de la partícula i en la iteración t . La velocidad, por su parte, se ajusta mediante la Ecuación 2.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_i^t - x_i^t) + c_2 r_2 (g_i^t - x_i^t) \quad (2)$$

Donde:

- I. ω es el factor de inercia que regula el impacto del movimiento anterior.
- II. c_1 y c_2 son los coeficientes de aprendizaje cognitivo y social.
- III. r_1 y r_2 son números aleatorios en el intervalo $[0,1]$.
- IV. p_i^t es la mejor posición alcanzada por la partícula i .
- V. g_i^t se corresponde con una solución del frente de Pareto elegida aleatoriamente como líder global.

El conjunto de soluciones no dominadas se construye dinámicamente a lo largo del proceso de optimización. En cada iteración, las partículas que no son superadas simultáneamente en todos los objetivos por ninguna otra de la población se incorporan al repositorio de élite, que representa el frente de Pareto. Este repositorio se actualiza constantemente, descartando soluciones dominadas conforme evoluciona la población (Kwilinski & Kardas, 2024).

3. Definición del problema de Optimización

El presente estudio se centra en la optimización de trayectorias para un brazo robótico industrial operando en un entorno tridimensional con obstáculos estáticos. El objetivo consiste en generar trayectorias que permitan al efector final completar una tarea determinada, en este caso, el movimiento se corresponde con recoger un objeto de una balda de una estantería de un supermercado y depositarlo en otra de forma segura y eficiente, evitando colisiones con el entorno y reduciendo al mínimo el esfuerzo articular y el tiempo de ejecución.

3.1. Variables de decisión

La trayectoria del efector final se modela mediante un conjunto de puntos de paso distribuidos en el espacio

tridimensional, cada uno con coordenadas y orientación $[x_i, y_i, z_i, \phi_i, \theta_i, \psi_i]$. De este conjunto de puntos, el primero y el último están predefinidos según los requisitos de la tarea (recogida y colocación), mientras que los puntos intermedios son los que varía el algoritmo, y constituyen las variables de decisión del problema. En esta configuración inicial se han definido 4 puntos intermedios optimizables, lo que da lugar a un total de 4 puntos $\times$ 6 variables = 24 variables reales de decisión. Esta cantidad permite un equilibrio adecuado entre flexibilidad en la trayectoria y coste computacional.

En este trabajo, al ser una primera aproximación a esta metodología, los ángulos $[\phi_i, \theta_i, \psi_i]$ se han fijado en $[0, \pi, 0]$, con el objetivo de tener menos variables de decisión, y favorecer la velocidad de cómputo del algoritmo. Esta posición se corresponde con la posición de recogida y colocación del objeto.

Cada partícula del algoritmo MOPSO representa una solución candidata, es decir, una secuencia concreta de puntos de paso intermedios que configura una trayectoria completa.

3.2. Control del manipulador

Para evaluar la viabilidad de las trayectorias generadas, es necesario convertir los puntos de paso definidos en el espacio cartesiano en configuraciones articulares (Peñacoba et al, 2024). Este proceso se resuelve mediante la cinemática inversa, que permite determinar los valores articulares (Abbes and Poisson, 2024). Véase la Ecuación 3.

$$q_i = g_i(x, y, z, \phi, \theta, \psi) \quad (3)$$

Donde i se corresponde con el número de articulaciones del brazo manipulador.

En este trabajo, la cinemática inversa se resuelve punto a punto a lo largo de la trayectoria interpolada, utilizando el solucionador `inverseKinematics()` de la `Robotics System Toolbox`.

Para mejorar la continuidad entre puntos consecutivos y evitar soluciones articulares no deseadas (como cambios abruptos o configuraciones cercanas a singularidades), se implementa un enfoque multi-semillado. Para cada punto objetivo, se generan cinco configuraciones iniciales ($k = 5$) cercanas a la configuración previa, siguiendo una perturbación gaussiana según la Ecuación 4.

$$q_k^{seed} = q_{prev} + \mathcal{N}(0, \sigma^2) \quad (4)$$

Para cada semilla, se calcula una solución articular q_k , y se selecciona aquella que minimiza la distancia euclídea con respecto a la configuración anterior (Ecuación 5).

$$q_i = \min(q_k - q_{prev}) \quad (5)$$

Este enfoque incremental garantiza la continuidad articular y reduce el riesgo de trayectorias no suaves o no factibles desde el punto de vista dinámico, mejorando la estabilidad del movimiento del robot a lo largo de toda la trayectoria.

Cabe destacar que, si bien se ha utilizado la norma euclídea estándar como criterio de continuidad, en configuraciones con articulaciones de diferente rango o sensibilidad podría ser

recomendable emplear una distancia ponderada o normalizada. Esta mejora se plantea como una posible línea futura para aumentar la equidad entre variables articulares con distintas magnitudes.

3.3. Restricciones geométricas y físicas

Para una optimización efectiva de los puntos de paso de la trayectoria, se han incorporado restricciones directamente asociadas a las variables de decisión que parametrizan la trayectoria en el espacio cartesiano. Cada solución candidata se representa mediante un conjunto de puntos intermedios (puntos de paso), definidos por sus coordenadas $(x, y, z, \phi, \theta, \psi)$.

Con el fin de reducir la complejidad del espacio de búsqueda en esta primera aproximación metodológica, se ha optado por fijar las componentes angulares (ϕ, θ, ψ) a los valores $(0, \pi, 0)$. Esta simplificación permite focalizar el esfuerzo computacional del algoritmo en las posiciones espaciales (x, y, z) , disminuyendo así el número de variables libres de 12 a 6 y acelerando significativamente el proceso de exploración.

Las restricciones impuestas sobre estas variables de decisión son las siguientes:

- I. Espacio de trabajo: se imponen cotas superior e inferior sobre las coordenadas (x, y, z) para garantizar que los puntos generados se mantengan dentro del volumen operativo del robot.
- II. Coherencia geométrica: los puntos inicial y final de la trayectoria permanecen fijos, al estar asociados a tareas específicas del caso de uso. Solo los puntos intermedios son optimizables dentro del dominio definido.

Este planteamiento busca un equilibrio entre realismo operativo y eficiencia computacional, permitiendo validar la metodología propuesta sin incurrir en un coste excesivo de exploración ni comprometer la validez geométrica de las soluciones generadas.

3.4. Funciones de coste

El proceso de optimización multiobjetivo requiere definir dos funciones de coste independientes, correspondientes a los criterios físicos a minimizar: el tiempo total de ejecución y el esfuerzo articular. No obstante, ambas funciones comparten una primera fase común de validación geométrica, diseñada para garantizar la viabilidad espacial de las trayectorias generadas. Las funciones de coste se formalizan con las Ecuaciones (6-7).

$$f_{c_1} = \begin{cases} 1 + t_{colisión}, & t_{colisión} > 0 \\ \frac{t_{trayectoria}}{t_{Máx}}, & t_{colisión} = 0 \end{cases} \quad (6)$$

$$f_{c_2} = \begin{cases} 1 + t_{colisión}, & t_{colisión} > 0 \\ \alpha * \frac{\bar{T}}{T_{Lim}} + \beta * \frac{T_{Máx}}{T_{Lim}}, & t_{colisión} = 0 \end{cases} \quad (7)$$

Ambas funciones de coste tienen una estructura secuencial con dos fases.

Fase 1: Evasión de obstáculos:

Antes de evaluar los criterios físicos, se descartan todas aquellas trayectorias que presentan colisiones, ya sea con el entorno o consigo mismas. Para ello, se aplica una penalización uniforme y severa a ambas funciones objetivo cuando se detecta un tiempo en colisión $t_{colisión}$, es decir, la duración total de la trayectoria durante la cual el robot intersecta algún obstáculo. Esta penalización se implementa asignando a ambas funciones de coste un valor artificialmente elevado, (primera fila en Ecuaciones 6-7).

De este modo, las soluciones no viables quedan desplazadas fuera del frente de Pareto durante la ejecución del algoritmo MOPSO, favoreciendo exclusivamente trayectorias libres de colisiones.

Fase 2: Optimización de métricas físicas:

Superada la evasión de obstáculos la optimización se centra en optimizar el tiempo y el esfuerzo articular.

- I. Función de coste 1 (f_{c1}): Minimiza el tiempo total de ejecución de la trayectoria, normalizado respecto a un valor de referencia $t_{Máx}$ definido a partir de una trayectoria de referencia (Ecuación 6, fila 2).
- II. Función de coste 2 (f_{c2}): Minimiza el esfuerzo articular, calculado como una media ponderada del par articular medio ($\bar{T}$) y el par máximo ($T_{Máx}$), ambos normalizados respecto a un límite estructural de seguridad T_{Lim} (Ecuación 7, fila 2).

En este estudio se ha adoptado una ponderación equitativa, con ($\alpha = 0,5$ y $\beta = 0,5$) lo que permite equilibrar la suavidad global del movimiento con la reducción de picos de esfuerzo.

4. Modelado del sistema

El entorno de evaluación ha sido implementado íntegramente en MATLAB R2024b, utilizando las funcionalidades avanzadas de la Robotics System Toolbox para llevar a cabo simulaciones físicas realistas del comportamiento del manipulador.

El robot simulado es un ABB CRB15000-95 (GOFA), importado mediante un fichero URDF enriquecido con propiedades dinámicas (masa, volumen e inercia de los eslabones). En la Figura 1 se representa gráficamente el modelo y su entorno de trabajo.

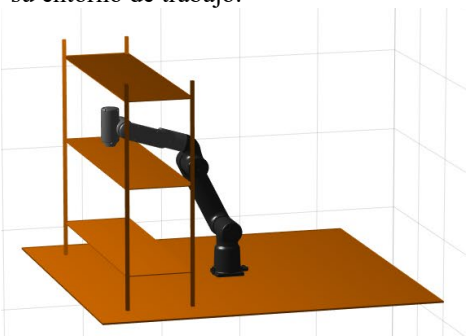


Figura 1: Entorno de simulación.

4.1. Generación e Interpolación de trayectorias

Las trayectorias candidatas generadas por el algoritmo se definen mediante un conjunto de puntos intermedios en el espacio cartesiano. Para evaluar su ejecución realista, estas trayectorias se interpolan utilizando polinomios de orden cinco, lo que garantiza la continuidad en posición, velocidad y aceleración. La expresión general del interpolador es la descrita en la Ecuación 8.

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (8)$$

Los coeficientes $a_0, \dots, a_5$ se obtienen a partir de las condiciones iniciales y finales de cada segmento de trayectoria, de modo que se respete la suavidad cinemática. Esta propiedad es crucial para asegurar que el robot pueda seguir las consignas sin generar comportamientos abruptos o no deseados en los actuadores.

4.2. Evaluación de colisiones

Durante el seguimiento de la trayectoria, se emplea la función `checkCollision()` para verificar si alguna configuración articular del robot provoca colisiones externas con el entorno simulado o colisiones internas entre sus propios eslabones. Esta comprobación se realiza paso a paso a lo largo de toda la trayectoria, garantizando una detección exhaustiva de cualquier invasión del espacio ocupado por obstáculos.

Los obstáculos fijos se definen previamente en función del escenario del caso de uso, mientras que la geometría del robot se gestiona automáticamente a partir del modelo URDF.

4.3. Cálculo del esfuerzo articular

La estimación de la carga dinámica que cada trayectoria impone al manipulador se realiza empleando una evaluación paso a paso del perfil de par articular necesario para ejecutarla. Este análisis se lleva a cabo utilizando el modelo dinámico del robot, sobre el que se aplica la función `inverseDynamics()` disponible en la Robotics System Toolbox.

Para calcular el par requerido en cada instante, se estiman primero las velocidades y aceleraciones articulares mediante un esquema de diferencias finitas entre las configuraciones obtenidas en puntos consecutivos de la trayectoria. Esta aproximación permite una integración directa con las configuraciones calculadas durante el proceso de interpolación y cinemática inversa.

A lo largo de toda la trayectoria, se registra:

- I. El valor promedio del par absoluto aplicado en todas las articulaciones, que representa el esfuerzo mecánico global sostenido;
- II. El valor máximo alcanzado por cualquiera de los ejes del robot, que refleja el pico de exigencia dinámica al que estaría sometido el sistema.

Ambos indicadores se utilizan posteriormente como criterios cuantitativos dentro de la función objetivo, permitiendo identificar trayectorias que, además de ser viables y eficientes en tiempo, eviten operar cerca de los límites estructurales del manipulador.

5. Resultados

El algoritmo MOPSO ha permitido generar un conjunto de trayectorias físicamente viables y libres de colisiones, optimizando simultáneamente distintas métricas relevantes. A partir del frente de Pareto obtenido, se han seleccionado tres soluciones representativas: una que prioriza el tiempo de ejecución, otra que minimiza el esfuerzo articular y una tercera de carácter intermedio.

La Tabla 1 resume las métricas más relevantes de cada trayectoria: el tiempo total de ejecución, el par medio, el par máximo medio y la existencia de colisiones. Se incluye también la trayectoria inicial (no optimizada) como referencia.

Tabla 1: Métricas

Trayectoria	Tiempo	Par Medio	Par Máximo Medio	Colisión
Inicial	5,00	-	-	Sí
Mín. Tiempo	11,48	26,73	77,27	No
Mín. Par	17,01	22.17	59,17	No
Intermedia	13,77	22,64	64,29	No

En la Figura 2 se representan las trayectorias seleccionadas del frente de Pareto, junto con la trayectoria inicial. Esta visualización permite apreciar cómo cada solución adapta su geometría para evitar el obstáculo presente en el entorno.

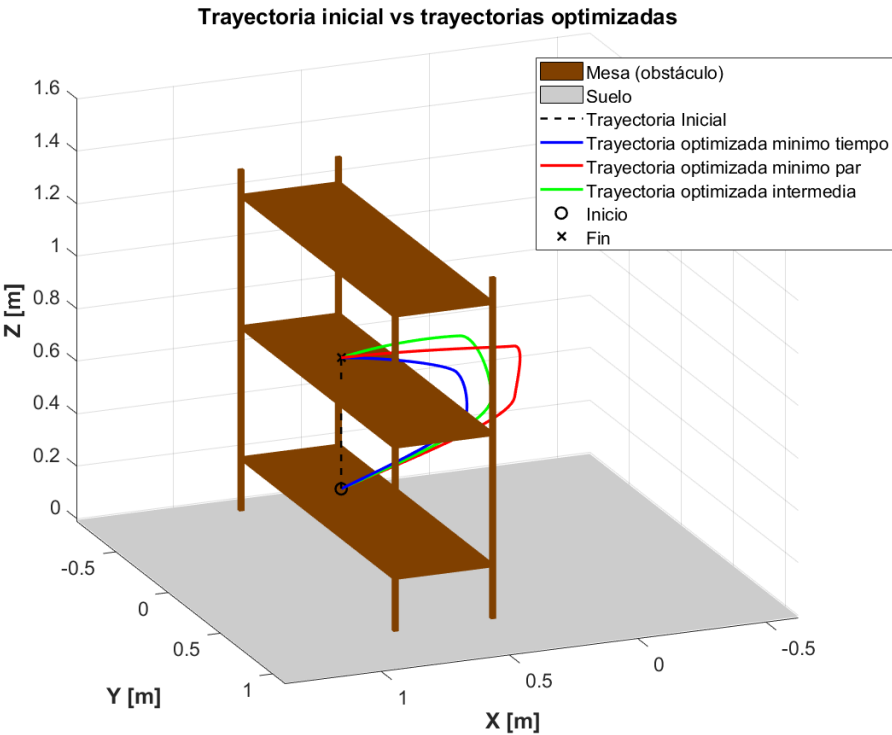


Figura 2: Título de una sola línea al centro.

En la Figura 3 se presenta la evolución del par máximo articular a lo largo del tiempo para las tres trayectorias seleccionadas del frente de Pareto.

La trayectoria optimizada en función del par articular (línea roja) muestra un perfil muy contenido, con valores de par máximo consistentemente bajos en toda la trayectoria.

Por el contrario, la trayectoria optimizada con respecto al tiempo de ejecución (línea azul) presenta varios picos de par significativamente más elevados.

La tercera trayectoria (línea verde), correspondiente a una solución intermedia, presenta valores de par máximo entre los dos casos anteriores, reflejando un compromiso adecuado entre eficiencia temporal y esfuerzo físico.

Aunque el análisis se centra en estas tres trayectorias representativas, el algoritmo ha sido probado con distintos escenarios y posiciones inicial/final, observándose un comportamiento coherente del método en todos los casos, lo que respalda la generalización de los resultados obtenidos.

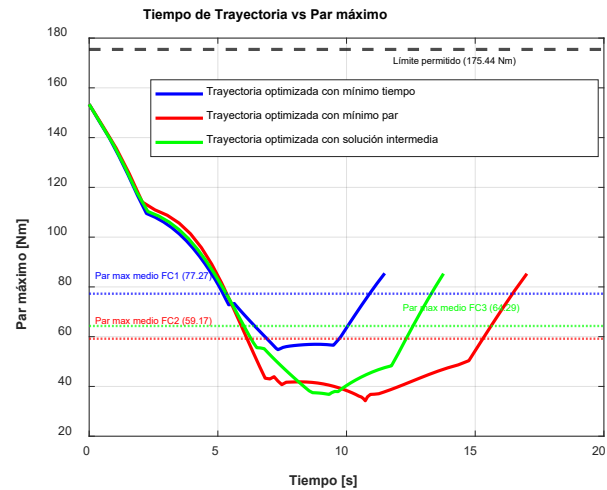


Figura 3: Par máximo medio de cada trayectoria con respecto al tiempo.

Finalmente, en la Figura 4 se muestra el frente de Pareto generado por MOPSO en el espacio de objetivos (tiempo de trayectoria vs esfuerzo articular). Este frente recoge un conjunto diverso de soluciones no dominadas, que representan distintos grados de compromiso entre eficiencia temporal y esfuerzo mecánico. Las tres soluciones seleccionadas aparecen reflejadas como puntos característicos de dicho equilibrio.

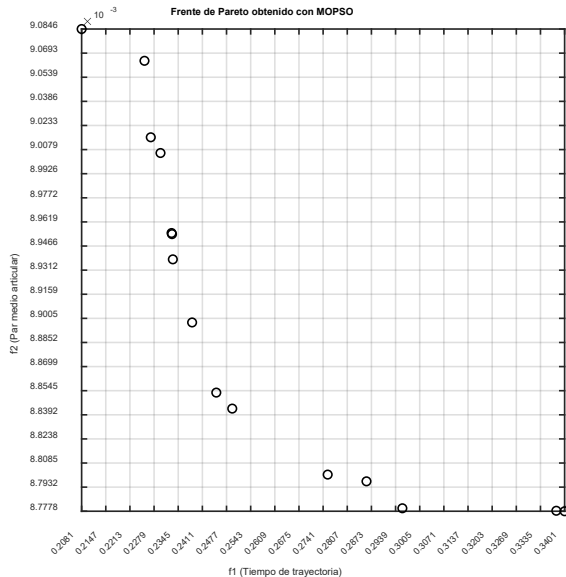


Figura 4: Par máximo medio de cada trayectoria con respecto al tiempo.

6. Conclusiones y trabajos futuros

Este trabajo ha presentado una metodología de optimización multiobjetivo de trayectorias para manipuladores industriales basada en el algoritmo MOPSO, aplicada sobre un entorno simulado que incorpora tanto restricciones geométricas como métricas físicas derivadas del modelo dinámico del robot. A través de una función de coste escalonada en dos fases, el sistema garantiza primero la viabilidad espacial descartando trayectorias con colisiones, y posteriormente optimiza criterios de rendimiento como el tiempo total de ejecución y el esfuerzo articular. Los resultados obtenidos muestran que el enfoque propuesto es capaz de generar trayectorias físicamente realistas, suaves y libres de colisiones, adaptadas a entornos industriales exigentes como los planteados en el proyecto europeo MANiBOT.

Entre las principales ventajas observadas se encuentra la capacidad del algoritmo para explorar soluciones diversas en el espacio de Pareto, permitiendo seleccionar trayectorias según distintos compromisos entre rapidez y eficiencia mecánica.

Como líneas futuras, se plantea extender la metodología actual en varias direcciones. En primer lugar, se evaluará la posibilidad de realizar una optimización independiente por articulación, con el objetivo de alejar cada eje del par máximo permitido, lo cual podría aumentar la vida útil de los actuadores. Además, se explorará la incorporación de nuevas restricciones dinámicas como límites de velocidad, aceleración y potencia, así como la introducción de trayectorias temporizadas con perfil de velocidad adaptable. Por último, se pretende validar la metodología en un entorno físico real,

empleando el robot ABB CRB 15000 con sensores externos para la reconstrucción de trayectorias ejecutadas y la validación experimental de las métricas propuestas. Aunque la simulación utilizada es realista y completa, se espera que la transición al entorno físico presente desafíos adicionales relacionados con la precisión de los modelos, la respuesta dinámica del sistema y la robustez frente a incertidumbres.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo de la Comisión Europea en el marco del proyecto europeo MANiBOT, número de referencia 101120823.

Referencias

- Abbes, M., & Poisson, G. (2024). Geometric Approach for Inverse Kinematics of the FANUC CRX Collaborative Robot. *Robotics*, 13(6), 91.
- Agarwal, A. K., Chauhan, S. S., Sharma, K., & Sethi, K. C. (2024). Development of time-cost trade-off optimization model for construction projects with MOPSO technique. *Asian Journal of Civil Engineering*, 25(6), 4529-4539.
- Dalai, R., Swain, S. C., & Giri, N. C. (2024, November). Microgrid Pareto Dispatch Optimization with Multiobjective Particle Swarm Optimization. In *2024 4th International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET)* (pp. 1-6). IEEE.
- Duan, P., Yu, Z., Gao, K., Meng, L., Han, Y., & Ye, F. (2024). Solving the multi-objective path planning problem for mobile robot using an improved NSGA-II algorithm. *Swarm and Evolutionary Computation*, 87, 101576.
- Ghafari, K. (2024). Multi-objective continuous review inventory policy using MOPSO and TOPSIS methods. *Computers & Operations Research*, 163, 106512.
- Hao, F., Jing, X., Liu, T., Da, Y., & He, G. (2024, December). Parameter Identification Of Robotic Arm Dynamics Model Based On Multi-Objective Particle Swarm Algorithm. In *2024 7th International Conference on Data Science and Information Technology (DSIT)* (pp. 1-6). IEEE.
- Kwilinski, A., & Kardas, M. (2024). The role of the Pareto principle in quality management within Industry 4.0: A comprehensive bibliometric analysis. *Virtual Economics*, 7(3), 7-24.
- MANiBOT. (2023). *MANiBOT project*. <https://manibot-project.eu/>
- Peñacoba, M., Bayona, E., Sierra-García, J. E., & Santos, M. (2024). Route Optimization for UVC Disinfection Robot Using Bio-Inspired Metaheuristic Techniques. *Biomimetics*, 9(12), 744.
- Peñacoba, M., Sierra-García, J. E., Santos, M., & Mariolis, I. (2023). Path Optimization Using Metaheuristic Techniques for a Surveillance Robot. *Applied Sciences*, 13(20), 11182.
- Peñacoba-Yagüe, M., Sierra-García, J. E., Santos-Peñas, M., Ruano, A. (2025). Enhancing Robotic Control Efficiency with MLP-Based Inverse Kinematics: First Approach. In: Aguiar, A.P., Rocha Malonek, P., Pinto, V.H., Fontes, F.A.C.C., Chertovskih, R. (eds) *CONTROL 2024. Lecture Notes in Electrical Engineering*, vol 1325. Springer, Cham. https://doi.org/10.1007/978-3-031-81724-3_51
- Reyes, C., Ramos-Fernández, J. C., Espinoza, E. S., & Lozano, R. (2025). Modelado difuso y control PDC de un sistema de generación de energía basado en un motor de combustión interna con un generador eléctrico. *Revista Iberoamericana de Automática e Informática industrial*, 22(1), 13-20.
- Wang, J., Zhang, Y., Zhu, S., & Wang, J. (2024). A Novel Multi-Objective Trajectory Planning Method for Robots Based on the Multi-Objective Particle Swarm Optimization Algorithm. *Sensors (Basel, Switzerland)*, 24(23), 7663.
- Yang, N. C., Su, C. H., & Yang, H. (2025). Optimal parameter tuning for voltage control thresholds in distribution networks using MOPSO and Pareto fronts. *Electric Power Systems Research*, 244, 111552.
- Zhou, J., Chieng, D., Lee, B. G., Ji, J., & Li, J. (2024, June). Many Objectives Autonomous Robot Path Planning with Improved MOEA/D. In *2024 IEEE Congress on Evolutionary Computation (CEC)* (pp. 01-08). IEEE.