

Generación de trayectorias para vehículos autónomos en entornos urbanos

Castellanos, C.^{a,*}, Yagüe, D.^a, Marín, P.^a, Armingol, J.^a

^aLaboratorio de Sistemas Inteligentes, Universidad Carlos III de Madrid
Avda de la Universidad, 30, 28911, Leganés, Madrid

To cite this article: Castellanos, C., Yagüe, D., Marín, P., Armingol, J. 2023. Trajectory generation for autonomous vehicles in urban environments. XLIV Jornadas de Automática, 557-562. <https://doi.org/10.17979/spudc.9788497498609.557>

Resumen

Este artículo plantea la implementación de un módulo de generación de trayectorias local dentro de una estructura de control ya desarrollada, perfeccionando un prototipo experimental para la conducción autónoma. En concreto, el planificador debe permitir al vehículo adaptarse a influencias externas, al mismo tiempo que considera las órdenes de los usuarios y su comodidad. En este trabajo se especifican las técnicas implicadas, desde los sistemas de referencia espacial que facilitan el posicionamiento del vehículo y los obstáculos de manera coherente en el espacio de la carretera, hasta los modelos de planificación de trayectorias. Asimismo, se muestran sus posibilidades de diseño y se discute su desempeño real. A nivel del software, se propone la utilización de la plataforma estructural ROS2, funcionando conjuntamente con el simulador CARLA, enfocado a la experimentación de vehículos autónomos. Por otra parte, el programa propone la optimización de la generación de trayectorias y una interpolación posterior, asegurando la continuidad del camino seguido por el vehículo.

Palabras clave: Vehículo autónomo, generación de trayectorias, planificación, optimización, sistemas de posicionamiento.

Trajectory generation for autonomous vehicles in urban environments

Abstract

The present article proposes the implementation of a local trajectory generation module within an already developed control structure, refining an experimental prototype for autonomous driving. In particular, the planner should allow the adaptation of the vehicle to external influences, while considering user commands and comfort. This paper specifies the techniques involved, from spatial reference systems which facilitate coherent positioning of the vehicle and obstacles in the road space, to trajectory planning models. Furthermore, its design possibilities are shown, as well as its actual performance is discussed. At the software level, the use of the ROS2 structural platform is proposed, working in conjunction with the CARLA simulator, focused on the experimentation of autonomous vehicles. On the other hand, the program proposes the optimization of the generation of trajectories and a subsequent interpolation, ensuring the continuity of the path followed by the vehicle.

Keywords: Autonomous driving, trajectory generation, path planning, optimization, positioning systems.

1. Introducción

El campo de la conducción autónoma plantea una serie de ventajas que justifican su auge actual indiscutible. En primer lugar, representa una alternativa más segura en los entornos viales, puesto que los algoritmos empleados se basan en las normas

de circulación, priorizando la protección de los pasajeros. Por otra parte, se favorece la accesibilidad a personas impedidas para la conducción, dotándoles de una mayor autonomía. Por último, los vehículos autónomos propician una optimización de recursos a nivel interno, como puede ser el combustible, mientras que a gran escala suponen la agilización de tráfico vial.

*Autor para correspondencia: cacastel@pa.uc3m.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

Contrariamente, se renuncia gradualmente al factor humano, dependiendo del rango de autonomía alcanzado. En concreto, se parte del nivel 1, donde se sitúan las ayudas a la conducción como el sistema de control de crucero adaptativo, hasta el nivel 5, que define la autonomía completa en todos los supuestos posibles en la carretera. En la Figura 1 se muestran las características de cada nivel de autonomía.

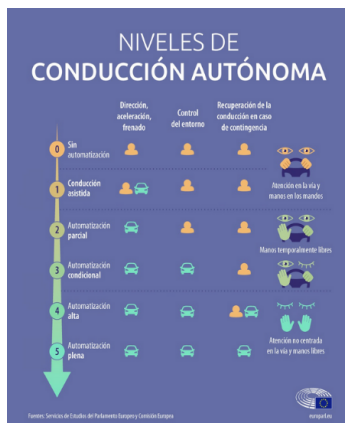


Figura 1: Niveles de autonomía por (Duch, 2019)

Las líneas de investigación dentro de este campo se centran en el desarrollo de sistemas de percepción capaces de interpretar escenarios dinámicos, arquitecturas de comunicación eficientes e interactivas con el usuario y el diseño de un control eficaz que permita al vehículo actuar de manera natural. En referencia al presente trabajo, parte de la plataforma ARES de (Marin-Plaza et al., 2021), destinada a la creación de un vehículo autónomo. Por lo tanto, la estructura de control existente interaccionará con el módulo de planificación propuesto.

Concretamente, se deben analizar las variables propioceptivas del vehículo y estimar su posición respecto a una trayectoria predefinida. Para este fin, es importante establecer un sistema de referencia eficiente para la tarea, por lo que en contraposición del sistema cartesiano, utilizado habitualmente en la robótica móvil, muchas investigaciones, como (Peng et al., 2022), (Xu et al., 2011) o (Werling et al., 2010), plantean definir el entorno de la carretera a través de un sistema de coordenadas centrado en el camino a seguir. Este planteamiento será una de las estrategias estudiadas para optimizar la conducción.

Del mismo modo, se considera el entorno urbano para posicionar los obstáculos y percibir las restricciones viales. Posteriormente, se calculan los puntos objetivos para la futuras trayectorias. En particular, el método exhaustivo se plantea como la medida principal en proyectos como los de (Werling et al., 2010) o (Xu et al., 2011), aunque conlleva un coste computacional alto. Por consiguiente, (Peng et al., 2022) plantea un algoritmo de optimización, denominado algoritmo SAA (*Simulated Annealing Algorithm*) que será objeto de estudio.

Finalmente, se procede a la generación de rutas alternativas para el vehículo autónomo. En el estudio de (Gonzalez Bautista et al., 2015) se encuentran diversas técnicas para abordar la planificación. Entre ellas, se destacan dos opciones comunes en las tareas de conducción autónoma, que son la generación por polinomios y por curvas de Bézier.

Sin embargo, a pesar de ser opciones recomendables se encuentran deficiencias para asegurar la continuidad y, por tanto, la seguridad de las trayectorias. Por este motivo, una de las propuestas de este artículo es la interpolación posterior de las rutas calculadas por ambas estrategias a través de *splines*, evitando esta problemática.

Aunque este planteamiento parezca asumible, la conducción autónoma representa un problema abierto, puesto que el control combinado de las dinámicas longitudinal y lateral en entornos dinámicos o a velocidades muy altas son un desafío de complejidad elevada. Por este motivo, este proyecto pretende establecer una base eficaz para las tareas asumibles de la conducción autónoma con vistas, en un futuro, de abordar las complejidades mencionadas.

Con este fin, el presente artículo establece, en primer lugar, la metodología teórica. A continuación, se detalla la arquitectura desarrollada junto con las técnicas aplicadas. Posteriormente, se especifican las simulaciones experimentales para analizar su desempeño y, finalmente, se presentan las conclusiones.

2. Metodología

Esta sección establece una base fundamental para la planificación, en particular, precisa los aspectos principales del estudio de los marcos de referencia y los modelos de trayectoria.

2.1. Sistemas de referencia

En la Figura 2 se describe el escenario inicial, donde la tarea es seguir la ruta establecida, minimizando la distancia lateral siempre que sea posible. Este planteamiento utiliza el sistema cartesiano, sin embargo, esta perspectiva complica obtener la desviación lateral respecto a la ruta central, al necesitar un cálculo continuo de la distancia. Además, presenta poca adaptabilidad a caminos con numerosos cambios de orientación.

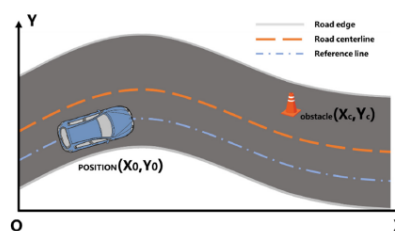
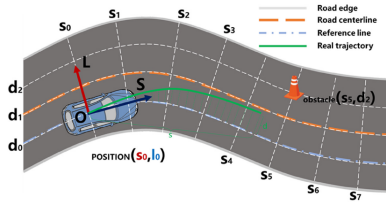


Figura 2: Representación en el sistema cartesiano por (Peng et al., 2022).

Como alternativa, se plantea una transformación de coordenadas, presente en trabajos como (Peng et al., 2022) o (Werling et al., 2010). El nuevo sistema de coordenadas se conoce como *Frenet Frame* y permite una interpretación más intuitiva de la posición del vehículo respecto a la carretera, ya que generaliza el camino a seguir como el eje central. La coordenada *s* representa la posición longitudinal en el camino global y la coordenada *d* se define como la desviación lateral.

La nueva definición del entorno se muestra en la Figura 3, donde el sistema se adapta a la curva del camino y la posición de los objetos se define por sus desviaciones longitudinal y lateral. De este modo, se facilita el diseño de nuevas trayectorias, puesto que sólo alteran la coordenada lateral.


 Figura 3: Diagrama de posiciones en *Frenet Frame* por (Peng et al., 2022)

Adicionalmente, el nuevo sistema asume que el entorno y la trayectoria de referencia han sido suavizados, como menciona (Werling et al., 2010). Este paso es imprescindible, ya que el empleo de rutas no suavizadas podría generar trayectorias imposibles y, por tanto, no se asegura el correcto funcionamiento.

2.2. Planificación de trayectorias

La planificación local parte de una trayectoria global como referencia, sin embargo, debe considerar aspectos descartados a nivel global, como los obstáculos, para modificar la ruta. Simultáneamente, se calculan los perfiles de velocidad y aceleración por si la maniobra implica un cambio de estos para ser completada con seguridad. Por lo tanto, se requiere un sistema eficiente que se adapte a la actualización de información.

El estudio de (Gonzalez Bautista et al., 2015) presenta diversas técnicas que abordan la planificación, recogidas en la Figura 4. Este proyecto se centrará en un primer generador de trayectorias de polinomios de quinto orden, mientras, el siguiente se fundamentará en las curvas de Bézier. Posteriormente, las trayectorias generadas se interpolarán con el método B-splines.

2.2.1. Generación de trayectorias

Los polinomios son un recurso común que permite ajustar $n+1$ restricciones, siendo n el grado del polinomio. Partiendo de (Werling et al., 2010) y (Peng et al., 2022), las trayectorias en un espacio bidimensional y variable en el tiempo están restringidas lateral y longitudinalmente en el *Frenet Frame*. Por consiguiente, los movimientos deben ser descritos como estados del vehículo en relación al tiempo. En otros términos, las restricciones de la trayectoria se agrupan en estados de inicio y final. En consecuencia, la planificación se transforma en un cálculo de optimización con restricciones representado en (1).

$$\text{Encontrar } f_{mov} \text{ que minimice } f_{cost} \\ \text{considerando } \left\{ \begin{array}{l} f_{mov}(t_0) = x_{t_0}, \quad f_{mov}(t_1) = x_{t_1} \\ f_{mov}'(t_0) = \dot{x}_{t_0}, \quad f_{mov}'(t_1) = \dot{x}_{t_1} \\ f_{mov}''(t_0) = \ddot{x}_{t_0}, \quad f_{mov}''(t_1) = \ddot{x}_{t_1} \end{array} \right\} \quad (1)$$

Siendo f_{mov} la ruta definida y x una restricción al principio o final del intervalo en términos de posición, velocidad o aceleración. El objetivo es minimizar la función de coste, establecida para asegurar factores de comodidad, seguridad y eficiencia.

Este método a priori tiene un coste computacional bajo, aunque aumentado por el grado del polinomio, lo que dificulta la ejecución en tiempo real y justifica los métodos de optimización mencionados. Asimismo, se asegura como mínimo la continuidad en posición, requisito esencial para la interpolación.

La segunda técnica utiliza el desarrollo matemático conocido como curvas de Bézier. Estas funciones polinómicas se basan en una misma ecuación binomial para calcular dos coordenadas especificadas, en este caso, s y d . En la Ecuación (2), el parámetro n representa el grado del polinomio, mientras que la variable τ representa un valor entre 0 y 1 que modela los puntos a lo largo de la curva. Por este motivo, variará en función del número de puntos especificado para componer la trayectoria. Igualmente, para definir la ruta a seguir se emplea $n + 1$ puntos de control, a través de un peso w_i , como se detalla en (2).

$$\text{Bezier}(n, \tau) = \sum_{i=0}^n \binom{n}{i} * (1 - \tau)^{n-i} * \tau^i * w_i \quad (2)$$

Entre las opciones estudiadas se utiliza las curvas de Bézier de cuarto orden, consultadas en (Han et al., 2010), ya que suponen un coste computacional generalmente bajo y la posibilidad de diseñar rutas cómodas, al disponer de un número suficiente de puntos de control. Además, establecen la continuidad en posición, requisito mínimo para realizar la interpolación.

2.2.2. Interpolación

Este proceso se basa en los *splines*, es decir, una metodología que utiliza funciones polinómicas separadas para definir cada segmento entre sus límites. Dentro de estas soluciones, se decide aplicar el método *B-splines*. Como menciona (Li et al., 2021), este modelo matemático preserva la continuidad de la trayectoria, en concreto, depende de la multiplicidad de los nudos k fijados en la ruta y, por otro lado, del grado de los polinomios p , resultando en una continuidad C^{p-k} .

Su funcionamiento parte de la Ecuación (3), consultada en (Sakai et al., 2018). En concreto, la trayectoria será el sumatorio de los puntos de control definidos y el componente *Basic Function* $N_{i,p}$, basado en la fórmula de recursión *Cox-de-Boor* como menciona (Li et al., 2021).

$$s(u) = \sum_{i=0}^{n-1} N_{i,p}(u) * P_i \quad (3)$$

La variable de control u tendrá un valor incremental que especifica el intervalo actual, mientras que n indica el número de puntos de control. Asimismo, se utilizan *B-splines* de orden tres para asegurar la continuidad de la trayectoria final generada.

3. Arquitectura Software

Este apartado detalla la estructura software y las técnicas diseñadas en el programa, a partir de los fundamentos expuestos.

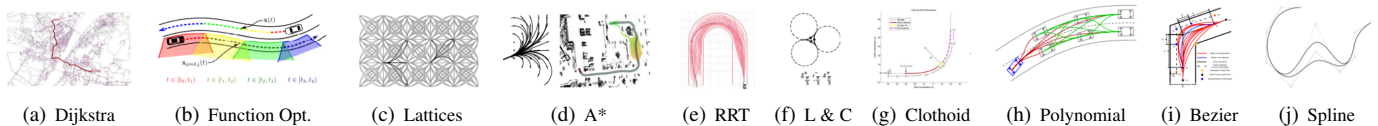


Figura 4: Métodos de planificación de trayectorias recogidos por (Gonzalez Bautista et al., 2015)

3.1. Estructura software

El programa propuesto debe funcionar dentro de una estructura de control diseñada para un vehículo autónomo y dirigida por la plataforma modular ROS2. Por lo tanto, el primer módulo planteado diseñará y publicará un conjunto de trayectorias alternativas para que un segundo módulo elija la ruta de menor coste, ajustada a las condiciones de la carretera.

La parte experimental se ha desarrollado en el simulador CARLA (*Car Learning to Act*) (Dosovitskiy et al., 2017). Esta aplicación destaca por su capacidad de personalización de entornos urbanos, como presenta la Figura 5, y su infraestructura para simular agentes externos. Por otra parte, existe una herramienta que permite la conexión eficaz con la plataforma ROS2, ya que permite convertir mensajes de los módulos en órdenes para los componentes del simulador, por medio de *topics* específicos que componen el *ROS-bridge*.



Figura 5: Ejemplos simulación CARLA

3.2. Diseño de la solución práctica

El mapa existente está dividido en secciones que identifican cada carril, por tanto, la planificación recoge una serie de tramos adyacentes, como ilustra la Figura 6. Cada segmento incluye información geométrica, límites de velocidad y una ruta central de puntos de referencia. Asimismo, los vecinos contienen un dato adicional acerca de la conexión con el tramo actual.

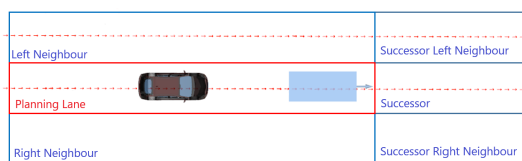


Figura 6: Estructura de la información del entorno

Partiendo de esta base, se establece el marco local de planificación, creado por un rango de puntos de referencia de la ruta central en proporción de la velocidad actual. Dentro de este marco, se definen las transiciones entre el sistema cartesiano, utilizado en el sistema de control, y el *Frenet Frame*, establecido para el entorno de planificación.

La transformación *Cartesian-Frenet* guarda el punto de referencia más cercano, además de su anterior y su siguiente. A continuación, se procede al cálculo geométrico descrito en la Figura 7, estimando las coordenadas frenet como las proyecciones de la recta entre el punto actual y el punto de interés. Es preciso que α sea inferior a 90° , por tanto, en caso contrario se repite la operación centrado el cálculo en el punto anterior.

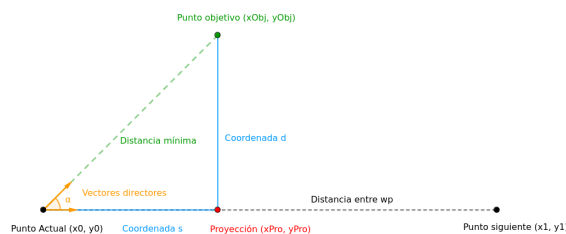


Figura 7: Transformación *Cartesian-Frenet*

Asimismo, las coordenadas frenet de los puntos de referencia están normalizadas en función de los segmentos, por lo que la coordenada s se sitúa según el tramo actual y su sucesor, es decir, en un rango de 0 a 2. Por lo tanto, se procede a un cálculo proporcional según distancias geométricas, fijando el marco local de planificación. En cuanto a la coordenada d , se estima la distancia real que separa al punto de interés de la referencia.

Por otra parte, antes de proceder con la transformación *Frenet-Cartesian*, se efectúa el proceso inverso al anterior para obtener las coordenadas normalizadas. A partir de estas, se establece el rango de puntos que comprenden la coordenada s del punto de interés. Seguidamente, se calcula geoméricamente su posición cartesiana, como muestra la Figura 8.

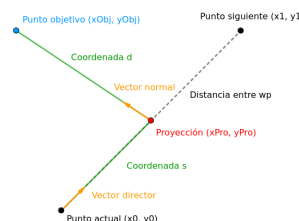


Figura 8: Transformación *Frenet-Cartesian*

El siguiente reto es el análisis de los eventos de percepción, establecidos en un conjunto de *bounding boxes*. Cada una se define como un vector de hipótesis que detallan su centro, tamaño y orientación. Con esta base, se estima el ángulo de las diagonales con el eje de simetría vertical para definir correctamente el ángulo de giro. De este modo, se deducen los vértices para las bibliotecas de análisis de colisiones. Por último, como muestra la Figura 9, se aplica una dilatación de $1,5\text{ m}$ como método de seguridad que penaliza las rutas en contacto con la forma final.

Por otra parte, la detección de obstáculos supone una modificación temporal en el marco local, para considerarlos hasta que sean adelantados. El límite inferior se fija en la posición del vehículo en el instante de la detección y, superiormente, según la coordenada s mayor del objeto.

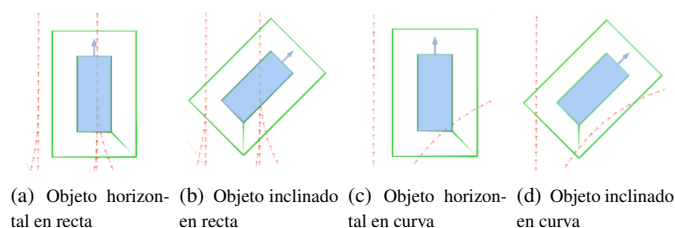


Figura 9: Ejemplos de representación de obstáculos

Con la culminación del marco local de planificación, se calculan los puntos objetivos. Previamente, su referencia se establece según el límite del marco local y es dinámica, ya que se ajusta a la velocidad del vehículo. Cuando se detecta un obstáculo, el programa fija la referencia al objeto, favoreciendo el desplazamiento lateral.

En base a la referencia, se procede al cálculo de las posiciones objetivo. La primera alternativa es un cómputo exhaustivo que establece los puntos equidistantes en el espacio lateral disponible y, longitudinalmente, se dispondrán según la líneas de planificación, como expone la Figura 10.

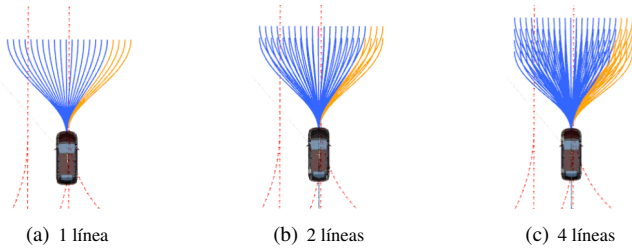


Figura 10: Configuraciones del método exhaustivo

La segunda estrategia utiliza el algoritmo SAA (*Simulated Annealing Algorithm*), recogido en la Figura 11. Es un generador de perturbaciones, donde la variación escogida se encuentra en la vecindad del espacio de soluciones. Si la alternativa resulta en una trayectoria con un menor coste, se acepta como punto central para continuar la búsqueda. En el caso contrario, se consulta el criterio de aceptación de Metropolis, definido por (Peng et al., 2022). En conclusión, este algoritmo representa una optimización del proceso, al mismo tiempo que evita mínimos locales, al favorecer la aceptación en las primeras iteraciones.

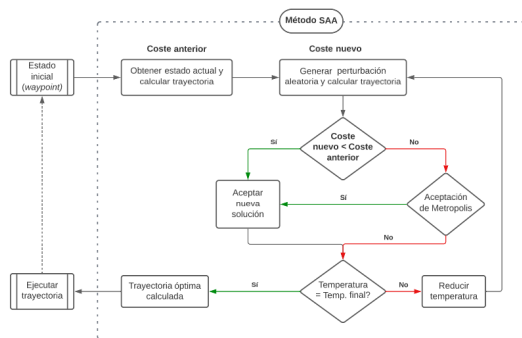


Figura 11: Algoritmo SAA

Por último, se procede a la generación de trayectorias con una de dos técnicas descritas, creando una ruta preliminar al interpolador. Asimismo, previamente se calcula el tiempo de ejecución, dependiendo de la velocidad y la distancia al objetivo. El resultado de la interpolación es una trayectoria en el espacio cartesiano definida en posición, velocidad y aceleración.

Al mismo tiempo, se estiman los costes asociados a la ruta para deducir el coste total de la misma. Las métricas diseñadas se basan en conceptos de la propia definición de la trayectoria, como la posición del objetivo respecto a la ruta central (C_{ref}), la presencia de obstáculos (C_{obs}) o su curvatura (C_k).

Por otra parte, la influencia de la carretera se determina según las conexiones del carril actual (C_{con}) y su acción de dirección para llegar al objetivo del usuario (C_{acc}). Cada coste tiene vinculado un peso que permite ajustar su influencia en el cómputo final, como se observa en (4).

$$C_{total} = W_{ref} * C_{ref} + W_{obs} * C_{obs} + W_k * C_k + W_{con} * C_{con} + W_{acc} * C_{acc} \quad (4)$$

4. Resultados experimentales

Esta sección detalla las pruebas realizadas para analizar el desempeño del módulo diseñado. Las primeras experimentaciones representan maniobras sencillas, como es el caso del cambio de carril y el adelantamiento. El primer grupo de factores evaluados serán los pesos de las influencias.

En el cambio de carril (Tabla 1) se consideran las influencias de la proximidad a la referencia, la acción y la curvatura del camino. El desempeño se evalúa según la distancia lateral conseguida respecto a la ruta de referencia del carril final.

Tabla 1: Experimentaciones con los pesos asociados al cambio de carril

Experimentación	W_{ref}	W_{acc}	W_k	Dist. referencia (m)
1	5.0	2.5	2.5	0.53
2	2.5	2.5	5.0	1.02
3	2.5	5.0	2.5	1.27
4	5.0	2.5	0.5	0.02

De esta prueba se deduce que la influencia de referencia es clave para aproximarse al objetivo y no interfiere en la separación inicial, siempre que la acción sea considerara. Mientras, el peso W_k se establece menor, al suponer un inconveniente.

Por otra parte, en el adelantamiento (Tabla 2) interviene la influencia añadida de los obstáculos. En este proceso, se debe analizar el distanciamiento respecto al obstáculo que debería superar el mínimo de seguridad de 1,5 m.

Tabla 2: Experimentaciones con los pesos asociados al adelantamiento

Experimentación	W_{ref}	W_{obs}	W_k	Dist. obstáculo (m)
1	10.0	5.0	5.0	colisión
2	5.0	10.0	5.0	colisión
3	5.0	5.0	10.0	colisión
4	5.0	15.0	0.5	2.23

En este proceso se estima que la mayor importancia debe recaer en la influencia de los obstáculos, mientras que los otros pesos deben quedar acorde con la anterior prueba.

Continuando con las opciones del algoritmo, se analiza el desempeño de cada generador de trayectorias en las pruebas descritas, además de considerar su tiempo de ejecución, resultando en las gráficas de la Figura 12. En primer lugar, las rutas efectuadas para el cambio de carril se representan por la línea verde en las Figuras 12(a) y 12(b). Se comprueba que el generador quíntico es más conservativo a la hora de aproximarse a la referencia del nuevo carril, representada por la línea naranja.

En cuanto al adelantamiento, en la Figura 12(c) se observa como ambos generadores superan el límite de seguridad, sin embargo, el generador quíntico obtiene menos separación, pudiendo favorecer alcances con la parte trasera del obstáculo.

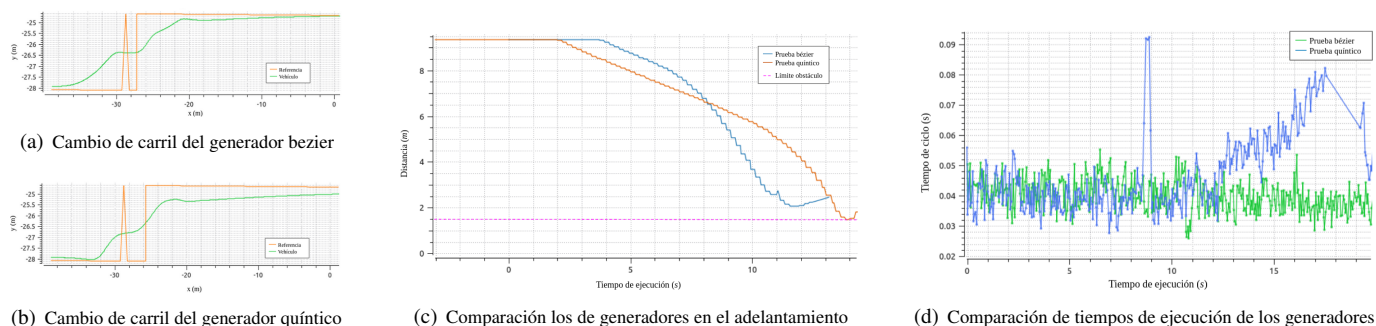


Figura 12: Resultados obtenidos de los generadores quintico y bezier

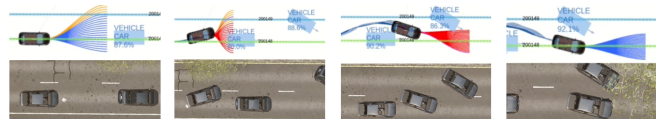
Finalmente, la Figura 12(d) muestra que el tiempo de ejecución del generador bezier es más estable, sin picos que supongan retardos. En suma, se establece como la opción predeterminada, por su mejor posicionamiento y eficiencia.

El estudio final (Tabla 3) examina el rendimiento en las pruebas y el tiempo medio de ejecución de la generación de puntos objetivo. Este análisis demuestra que agregar líneas de planificación al método exhaustivo no supone una mejora sustancial. Conjuntamente, se observó que el algoritmo SAA presenta cierta inestabilidad en el camino, ya que, en un rango cercano, los pesos de las rutas son similares y, por tanto, se producen redirecciones constantes, perjudicando la comodidad.

Tabla 3: Experimentaciones con la generación de puntos objetivo

Método	Dist. ref (m)	Dist. obs (m)	Tiempo (s)
Exh. 1 línea	0.02	2.23	0.04
Exh. 3 líneas	0.41	2.42	0.11
Exh. 5 líneas	0.47	2.20	0.19
Algoritmo SAA	0.18	2.08	0.03

Por último, la Figura 13 muestra la ejecución de la configuración final ante un entorno estático complejo. El vehículo debe evitar los objetos presentes, permaneciendo lo más alejado posible, y dirigirse al destino indicado por la línea verde.



(a) Primer instante (b) Segundo instante (c) Tercer instante (d) Cuarto instante

Figura 13: Maniobra de evasión con dos obstáculos

5. Conclusión

En este trabajo se plantea el desarrollo un generador de trayectorias local que debía integrarse en la estructura de control existente, con un eficiencia adecuada para la ejecución en tiempo real. Además, las pruebas efectuadas permiten afirmar que el sistema es capaz de navegar en entornos estáticos complejos, siguiendo las peticiones del usuario, en tanto que verifica la seguridad y comodidad de la ruta.

En cuanto a la configuración diseñada, se destaca el generador bezier por su precisión y eficiencia. Asimismo, el análisis de colisiones de la biblioteca *OpenCV* y la optimización del método exhaustivo han contribuido al rendimiento expuesto.

Como trabajos futuros, se debe concretar la definición de los pesos y ahondar en la configuración del algoritmo SAA al ser una opción prometedora. Además, sería recomendable optimizar la estructura de mensajes a fin de precisar funcionamientos dentro de la tarea permitiendo, por ejemplo, la obtención directa de la posición de los obstáculos en *Frenet Frame*. Por último, continuando con las experimentaciones, las simulaciones dinámicas son esenciales antes de proceder con las pruebas en escenarios reales.

Agradecimientos

Subvención PID2019-104793RB-C31, PDC2021-121517-C31, PDC2022-133684- C31 y PID2021-124335OB-C21 financiados por MCIN/AEI/ 10.13039/501100011033 y por la Unión Europea "NextGenerationEU/PRTR".

Referencias

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. pp. 1–16.

Duch, J., 2019. Coches autónomos en la ue: de la ciencia ficción a la realidad. Parlamento Europeo.

Gonzalez Bautista, D., Pérez, J., Milanes, V., Nashashibi, F., 2015. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems* 17, 1–11.

Han, L., Yashiro, H., Tehrani Nik Nejad, H., Do, Q. H., Mita, S., 2010. Bézier curve based path planning for autonomous vehicle in urban environment. *IEEE Intelligent Vehicles Symposium*, 1036–1042.

Li, Z., Xiong, L., Zeng, D., Fu, Z., Leng, B., Shan, F., 2021. Real-time local path planning for intelligent vehicle combining tentacle algorithm and b-spline curve. *6th IFAC Conference on Engine Powertrain Control, Simulation and Modeling E-COSM 2021* 54 (10), 51–58.

Marín-Plaza, P., Yagüe, D., Royo, F., de Miguel, M. A., Moreno, F. M., Ruiz-de-la Cuadra, A., Viadero-Monasterio, F., García, J., San Roman, J. L., Armingol, J. M., 2021. Project ares: Driverless transportation system. challenges and approaches in an unstructured road. *Electronics* 10 (15). DOI: <https://doi.org/10.3390/electronics10151753>

Peng, B., Yu, D., Zhou, H., Xiao, X., Xie, C., 2022. A motion planning method for automated vehicles in dynamic traffic scenarios. *Symmetry* 14 (2). DOI: <https://doi.org/10.3390/sym14020208>

Sakai, A., Ingram, D., Dinius, J., Chawla, K., Raffin, A., Paques, A., 2018. Pythonrobotics: a python code collection of robotics algorithms.

Werling, M., Ziegler, J., Kammel, S., Thrun, S., 2010. Optimal trajectory generation for dynamic street scenarios in a frenet frame. *Proceedings - IEEE International Conference on Robotics and Automation*, 987 – 993. DOI: 10.1109/ROBOT.2010.5509799

Xu, W., Yao, W., Zhao, H., Zha, H., 2011. A vehicle model for micro-traffic simulation in dynamic urban scenarios. *Proceedings - IEEE International Conference on Robotics and Automation*, 2267 – 2274.