




# Sistema de planeación y control de navegación para un vehículo autónomo en un entorno urbano

## Planning system and navigation control for an autonomous vehicle in an urban environment

C. Barrera-Ramírez <sup>a,\*</sup>, O. González-Miranda <sup>a</sup>, J. M. Ibarra-Zannatha <sup>b</sup>

<sup>a</sup> Departamento de Control Automático del Centro de Investigación y de Estudios Avanzados del IPN, 07360 Ciudad de México, México.

<sup>b</sup> Laboratorio de Robótica y Visión Artificial del Departamento de Control Automático del Centro de Investigación y de Estudios Avanzados del IPN, 07360 Ciudad de México, México.

### Resumen

En este trabajo se reporta el desarrollo de un sistema que permite la planificación de caminos en un entorno urbano para un vehículo autónomo dada la posición actual del vehículo así como la posición del destino deseado. Por otro lado, este sistema produce la especificación de la trayectoria correspondiente y que será utilizada como consigna en el controlador que genera la velocidad y la dirección (*steering*) del vehículo. Se presenta, además, un controlador clásico para asegurar la correcta navegación, es decir, hacer el seguimiento de esta trayectoria con base exclusivamente en la odometría y de un controlador visual lateral que aporta las correcciones necesarias para mantenerse en el centro del carril, en particular después de realizar las vueltas en las esquinas, ya sea a la izquierda o a la derecha, las cuales suelen perturbar un poco la dirección de avance del vehículo. Finalmente se presentan resultados en simulación utilizando ROS, Gazebo y el modelo del vehículo AutoMINY de la U. Libre de Berlín.

**Palabras Clave:** Vehículos autónomos, planificación de caminos, generación de trayectorias.

### Abstract

In this work, the development of a planning system of paths for an autonomous vehicle is reported. It works in an urban environment and uses the current and goal positions. This system gets the corresponding trajectory that will be the reference in a steering and speed controller. Also, a classic controller is presented which ensures the appropriate navigation. It does the trajectory tracking using odometry and a visual-feedback controller to get the necessary corrections to lane-keeping; especially in turns where perturbations in the steering appear. Finally, simulation results are presented using the Gazebo ROS simulator and the AutoMiny model from the Free University of Berlin.

**Keywords:** Autonomous vehicles, road planning, trajectory generation.

## 1. Introducción

Actualmente, la industria automotriz es una de las más grandes del mundo; nuestro ritmo de vida y la sociedad actual no podrían existir sin el poder y el impacto de las capacidades de movilidad vehicular. Todos los espacios urbanos, aún aquellos de las más pequeñas localidades, así como las carreteras de todo el mundo están dominados por todo tipo de automotores. Además, este tipo de transporte está en constante progreso y se beneficia de innovaciones tecnológicas cada vez más importantes.

Así, gracias a las investigaciones y desarrollos de los últimos 40 años, los automóviles dependen cada vez más de subsistemas electromecánicos (mecatrónicos) basados en sensores, actuadores y, más recientemente, en tecnologías informáticas y en sistemas de control. De este modo, los vehículos autónomos están a un paso de dejar su etapa de prototipo y formar parte de nuestra vida cotidiana (World Motor Vehicle Production: 2021 Production Statistics, 2021).

En este sentido, el mayor desafío en la actualidad es el desarrollo de sistemas eficientes que puedan implementarse en un vehículo con el principal objetivo de que éste se conduzca de manera autónoma (Otero, et al., 2017). En este contexto, el

\*Autor para la correspondencia: cruz.barrera@cinvestav.mx

**Correo electrónico:** cruzbaram@hotmail.com (Cruz Barrera-Ramírez), ogonzalez31416@gmail.com (Óscar González-Miranda), jibarra@cinvestav.mx (Juan Manuel Ibarra-Zannatha)

presente trabajo se centra en la planificación de caminos, en la generación de trayectorias y en el control de dichas trayectorias a fin de garantizar la conducción autónoma de vehículos en entornos urbanos.

### 1.1. La conducción autónoma de vehículos

La conducción de un vehículo puede parecer simple y sencilla para una persona con experiencia en dicha acción, sin embargo, si se analiza la tarea con detenimiento es sorprendente cuantas evaluaciones y decisiones realmente ejecuta una persona al momento de conducir eficazmente. Esto se puede tornar complejo al intentar hacerlo con una máquina o, en este caso en particular, con un vehículo autónomo.

De acuerdo con la *Society of Automobile Engineers*, SAE por sus siglas en inglés, la acción de conducción autónoma está constituida por el control del vehículo, el cual a su vez se divide en control lateral y control longitudinal. El *control lateral* se refiere al control de la dirección (*steering*) es la tarea de dirigir y navegar por la carretera manteniéndose al centro de su carril, en particular durante las curvas. A la tarea de controlar la posición y/o velocidad del coche a lo largo de la carretera por medio de acciones como la aceleración y el frenado corresponde al *control longitudinal*. Por otra parte, también es necesario considerar las tareas de planificación tanto de la ruta como de la trayectoria correspondiente, aspectos muy importantes de la conducción autónoma, así como el respeto de la señalización de tránsito, tanto la horizontal como la vertical (Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2014).

### 1.2. Problema a resolver y solución propuesta

En este trabajo se propone un sistema de planificación y control de la navegación de un vehículo autónomo a escala 1:10 que le permita ir de su posición actual a una posición deseada en un entorno urbano conduciéndose de manera autónoma, sin considerar por el momento la señalización ni la presencia de otros vehículos. Como primer paso se aplica un algoritmo de búsqueda en grafos de amplitud y profundidad ( $A^*$ , *Dijkstra*, etc.), para encontrar el camino entre las esquinas de origen y de destino como una secuencia de nodos a visitar. El segundo paso consiste en convertir este camino en una trayectoria que sirva de consigna al sistema de control del vehículo. Las calles del entorno urbano considerado (figura 7) cuentan con dos carriles, uno para cada sentido, por lo que la trayectoria a calcular debe incluir las coordenadas de alguna de las ocho posiciones que cada esquina tiene, es decir, alguna de las posiciones detrás del paso de peatones en la que un auto está esperando la luz verde del semáforo para circular, como se muestra en la figura 1, así como las trayectorias curvas correspondientes a los giros a la izquierda y a la derecha. El tercer paso es el diseño de un sistema de control lateral y longitudinal que permita hacer navegar al vehículo a lo largo de esta trayectoria.

### 1.3. Organización del artículo

En la Sección 2 se presenta la metodología de planificación desarrollada, la cual incluye la generación del camino entre la esquina actual y la esquina deseada mediante la heurística

clásica de *Dijkstra* (La Valle, 2006), así como una metodología para generar la trayectoria correspondiente. En la Sección 3 se presenta el sistema de control de navegación desarrollado, comenzando por el modelado cinemático del vehículo, seguido de la máquina de estados que comanda la secuencia de movimientos correspondientes a la trayectoria deseada y, finalmente, el sistema de control lateral y longitudinal implementado sobre el vehículo autónomo. La Sección 4 se dedica a la implementación en simulación del sistema desarrollado, así como al análisis de resultados. Finalmente, la Sección 5 presenta los comentarios finales y trabajo a futuro dentro de este proyecto.

## 2. Planificación del movimiento

El propósito principal de la conducción autónoma de un vehículo es navegar desde su posición actual *punto A* hasta su destino final *punto B* de manera eficiente (Rajamani, 2011). Para ello primero es necesario obtener el camino más corto entre ambos puntos y luego generar la trayectoria correspondiente a fin de garantizar que el vehículo circulará por el centro de su carril aún en el caso de tener que dar la vuelta en alguna esquina. Al primer problema se le conoce como *Planificación de la Ruta* y al segundo *Generación de Trayectorias* (Bruce A. & Manfredi, 2016). A continuación, se presenta cada uno de ellos.

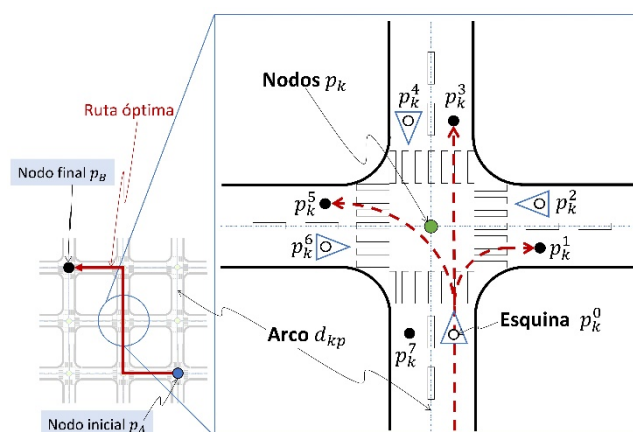


Figura 1: Elementos de un cruce de una ciudad urbana.

### 2.1. Planificación de la Ruta

La Planificación de la Ruta (*Path Planning*) es un procedimiento, imprescindible en la navegación de vehículos autónomos, que debe encontrar eficientemente el camino óptimo hacia un destino dado basándose en la Topología del mapa, haciendo abstracción de otros detalles como las reglas de tránsito o los agentes presentes en el escenario de conducción, entre otros (La Valle, 2006). Para capturar la Topología del mapa de calles del entorno urbano de interés se utilizará su representación en un grafo orientado y ponderado (Godsil & Royle, 2001), sobre el cual se aplicará un algoritmo de búsqueda que permita obtener la ruta deseada. Los nodos del grafo en el mapa se definen como la intersección de las medianas de las calles, se denotan como  $p_k$  y se caracterizará por sus coordenadas  $(x_k, y_k)$  en el referencial de dicho mapa.

### 1) Representación del mapa con un grafo

El grafo representado matemáticamente como  $G = (P, D)$  es una función real positiva sobre el conjunto de arcos orientados, es decir, una función  $d : D \rightarrow \mathbb{R}^+$  (La Valle, 2006) (Russell & Norvig, 2016), el grafo captura la Topología del mapa y está formado por un conjunto de nodos  $p_k = (x_k, y_k)$  representando a todos los cruceros del mapa y un conjunto de arcos  $d(p_i, p_j) = d_{ij}$  que las conectan como se ilustra en la figura 1. La orientación de cada arco estará representando los sentidos que tenga cada calle. Además, cada arco tendrá un peso correspondiente a la longitud de la calle que representa, la cual se calcula como la distancia euclidiana entre los cruceros (nodos) que los conecta, recordando que cada crucero está representado por sus coordenadas en el mapa. Así, el arco  $d_{ij}$  es aquel que conecta los nodos  $p_i$  y  $p_j$  cuyo peso se define como:

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (1)$$

Se llama matriz de pesos del grafo ponderado a la matriz de adyacencias  $\Omega = (d_{ij})_{n \times n}$ , cuyos elementos son los arcos del grafo y se representan por su peso; de modo que  $d_{ii} = 0$  para todo  $i$  y, si no hay un arco desde el nodo  $d_i$  al nodo  $d_j$ , se considera como  $d_{ij} = \infty$ . Además, se define como peso mínimo de  $p_i$  a  $p_j$  al mínimo de los pesos de los posibles caminos entre  $p_i$  y  $p_j$ , este peso mínimo se denota como  $d_{ij}^*$ . Al camino (de pesos) mínimo de  $p_i$  a  $p_j$  es cualquier camino  $C_{ij}$  entre los que se tengan cuyo peso es mínimo, es decir,  $d(C_{ij}) = d_{ij}^*$  (La Valle, 2006) (Russell & Norvig, 2016) (Takeyas, 2007).

### 2) Algoritmo de búsqueda Dijkstra

A partir del grafo orientado obtenido de la manera anteriormente explicada, ahora debe determinarse la ruta óptima, la de menor longitud, entre la posición actual del vehículo *nodo A* y la de destino *nodo B*. Esta ruta es la secuencia de nodos o cruceros que debe seguir el vehículo para llegar a su destino. Para ello se aplica un algoritmo de búsqueda aplicado a grafos. Con base en la estructura del grafo de la ciudad propuesta figura 7 y en la cantidad de datos a procesar, es decir, del número de intersecciones o cruceros que contiene la ciudad, se opta por utilizar el algoritmo de búsqueda *Dijkstra*.

El objetivo esencial del algoritmo *Dijkstra* es proporcionar los pesos mínimos y los arcos asociados para llegar al nodo objetivo desde el nodo inicial dado, teniendo como base un conjunto de nodos  $P = \{p_1, p_2, \dots, p_n\}$  en el grafo, su matriz de adyacencias o pesos, los nodos inicial y final denotados como  $p_A$  y  $p_B$  respectivamente (La Valle, 2006) (Russell & Norvig, 2016) (Takeyas, 2007). Para esto se usa un conjunto  $L$ , que contendrá los nodos que ya han sido usados para construir el camino mínimo, denominado *conjunto cerrado* o *conjunto de nodos ya visitados*, y un conjunto  $H$  denominado *conjunto mínimo abierto* que almacena los nodos y los arcos con sus pesos asociados, ordenados de menor a mayor peso y un *vector de pesos D* que contendrá al final los pesos mínimos.

Inicialmente  $L = \{p_A\}$  y  $D = \Omega(p, :)$ , la  $p$ -ésima fila de la matriz de pesos corresponde al vértice inicial.

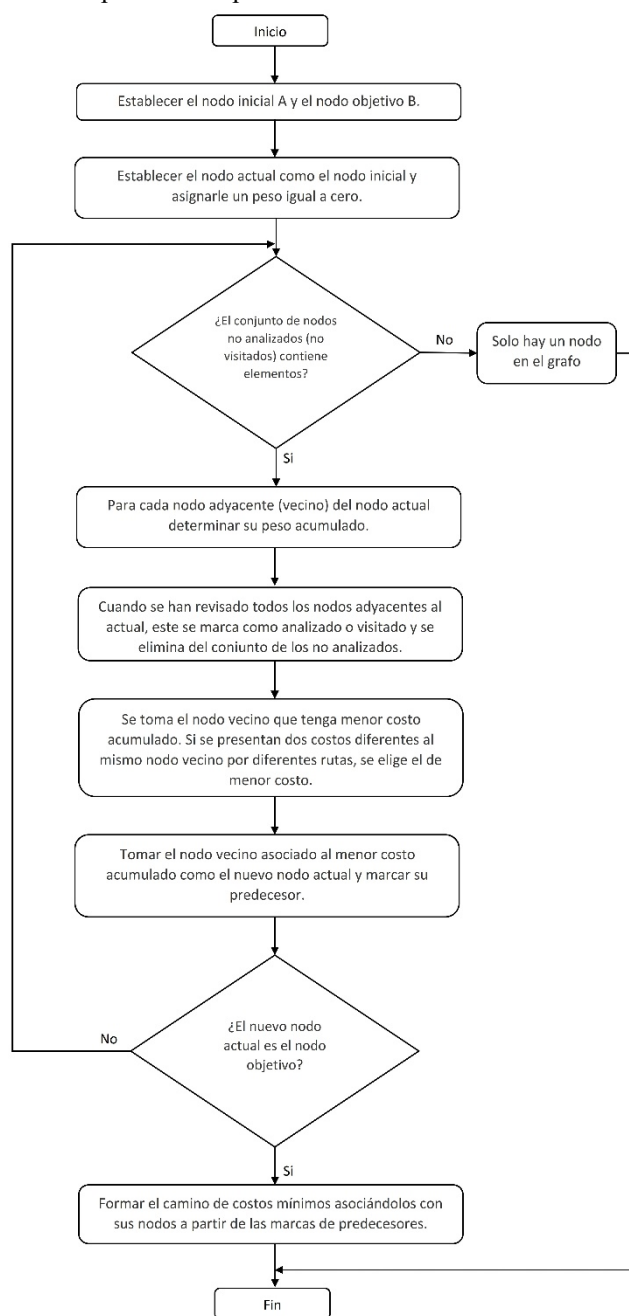


Figura 2: Diagrama de flujo para el algoritmo *Dijkstra*.

En la figura 2 se presenta el diagrama de flujo del algoritmo *Dijkstra*. Se comienza por establecer un punto de partida inicial  $p_A$  y un punto objetivo  $p_B$ , luego se considera el nodo de partida como el nodo actual, es decir,  $p_A = p_k$  para iniciar el análisis y se le pone un peso igual a cero  $d_{AA} = 0$  puesto que es el nodo de partida, dicho nodo  $p_k$  se extrae del conjunto  $P$  y se inserta en el conjunto  $H$ . Después se analiza el conjunto de nodos  $P$  que está conformado por todos los nodos del grafo considerado, denominado *conjunto de nodos no analizados (no visitados)*.

Enseguida se toma una decisión sobre el conjunto  $P$ , si no contiene elementos quiere decir que el grafo considerado sólo tiene un nodo y no se puede generar una ruta como tal; en el caso contrario, si  $P$  contiene elementos se procede a analizar

cada uno de los nodos adyacentes (vecinos) al nodo actual, cabe mencionar que esta información se extrae de la matriz de pesos o adyacencias, cada nodo vecino se extrae del conjunto  $P$ , se añade al conjunto  $H$  y se calcula el peso acumulado hasta ese nodo vecino  $D(j)$ , de la siguiente forma:

$$D(j) = D(k) + d_{kj} \quad (2)$$

Donde  $d_{kj}$  es el peso correspondiente al arco que conecta al nodo actual con el vecino y, a su vez,  $D(k)$  es el peso acumulado al nodo actual. Una vez que se han revisado todos los nodos adyacentes al nodo actual, éste se marca como analizado o visitado, es decir, se extrae del conjunto  $H$  y se añade al conjunto  $L$ :

$$L = L \cup \{p_k\} \quad (3)$$

Un caso particular es cuando varias posiciones de  $D$  tienen diferente costo acumulado, es decir, cuando hay dos o más rutas al mismo nodo, pero con diferentes costos; aquí se considera siempre el menor, lo cual se analiza siempre en el vector  $H$ . Después, se toma el menor costo acumulado  $\min(D(j)) \in H$  de los obtenidos entre todos los nodos vecinos y se considera su nodo asociado como el nuevo nodo actual:

$$p_j = p_k \quad (4)$$

Ahora se realiza una decisión sobre el nuevo nodo actual, si no es el nodo objetivo se repite el proceso a partir de la primera decisión que considera si hay o no elementos en  $P$ , en caso contrario se llega al nodo objetivo y se reconstruye el camino a partir del vector de pesos  $D$  y de sus nodos asociados, de esta manera se obtienen los cruceos que forman el camino más corto  $D = d_{AB}^* = d(C_{A,B})$  para ir de  $p_A$  a  $p_B$ , expresado también como:

$$d_{AB}^* = \{p_A, \dots, p_B\} = \{(x_A, y_A), \dots, (x_B, y_B)\} \quad (5)$$

## 2.2. Generador de trayectorias

La ruta que produce el planificador es un concepto topológico representado por la secuencia de nodos (cruceos) por los que debe pasar el vehículo autónomo para llegar a su destino. Ahora se debe obtener la trayectoria geométrica correspondiente formada por segmentos de recta y curvas. Así, se deben calcular las coordenadas de inicio y terminación de estos elementos de trayectoria y, en el caso de las curvas, el radio y el sentido de giro. Cabe recordar que el cruceo o nodo  $p_k$  está definido por las coordenadas  $(x_k, y_k)$  del punto de intersección entre las medianas de las calles que ahí se cruzan, el cual nunca forma parte de la trayectoria.

En realidad los puntos que definen la trayectoria son las ocho esquinas  $p_k^j = (x_k^j, y_k^j)$  que tiene cada cruceo o nodo  $p_k$ : 4 puntos de llegada (llegada al nodo), correspondientes a los puntos detrás del paso de peatones en donde un vehículo estaría detenido esperando la luz verde del semáforo (numeradas 0, 2, 4 y 6, comenzando con la que se encuentra más al sur y avanzando en el sentido opuesto a las manecillas del reloj) marcadas en la figura 1 con un triangulito y un círculo dentro; y 4 más de salida que se encuentran a la izquierda de estos puntos sobre el carril en sentido opuesto pero de la misma calle

(numeradas como 1, 3, 5 y 7), marcadas con un pequeño círculo negro como se muestra en dicha figura 1. Entonces, si tenemos el vehículo autónomo en la esquina de llegada al cruceo  $p_k$  denotada como  $p_k^0$ , sólo hay tres posibles trayectorias (ilustradas en la figura 3):

- i) Continuar recto hacia la esquina de llegada al siguiente nodo  $p_{k+1}^0$ .
- ii) Dar vuelta a la derecha con un radio  $R_{der}$  hacia la esquina de salida de ese mismo nodo  $p_k^1$  y luego hasta la esquina de llegada del nodo siguiente  $p_{k+1}^6$ .
- iii) Dar vuelta a la izquierda con radio  $R_{izq}$  hacia la esquina de salida de ese mismo nodo  $p_k^5$  para continuar hacia la esquina de llegada del nodo siguiente  $p_{k+1}^2$ .

Con base en lo anterior cada nodo  $p_k$  proporcionado en la ruta óptima debe ser transformado a sus ocho esquinas  $p_k^j$  con  $j = 0, 1, 2, \dots, 7$ , es decir:

$$p_k = \{p_k^0, p_k^j\} \Leftrightarrow (x_k, y_k) = \{(x_k^0, y_k^0), (x_k^j, y_k^j)\} \quad (6)$$

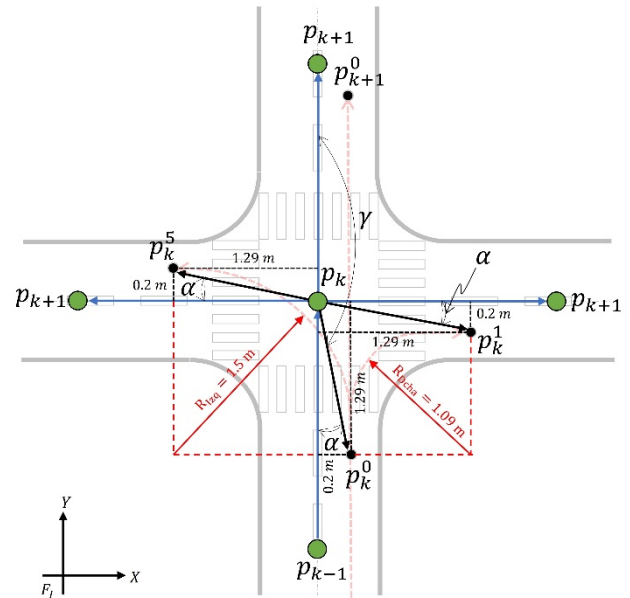


Figura 3: Esquinas correspondientes al nodo  $p_k$ .

Se debe tener en cuenta también que todos los cruceos de la ciudad considerada para este trabajo son iguales en cuanto a su geometría, por lo que todos los cruces están exactamente a  $90^\circ$ , lo cual se puede apreciar en la figura 7, por lo tanto, los tres segmentos de trayectoria antes referidos son iguales para todos los cruceos variando solamente su orientación. De esta manera la trayectoria completa de un nodo inicial  $p_A$  a un nodo objetivo  $p_B$  será la unión de cada uno de los segmentos de trayectoria determinado para cada nodo que indique el algoritmo *Dijkstra*, consecuentemente, la trayectoria  $T_{AB}$  que debe seguir el vehículo se define como el conjunto de esquinas específicas para cada nodo de la ruta óptima, es decir:

$$T_{AB} = \{(x_A^0, y_A^0), (x_A^j, y_A^j), \dots, (x_B^0, y_B^0), (x_B^j, y_B^j)\} \quad (7)$$

Ahora bien, para determinar el segmento de trayectoria que debe seguir el vehículo a través de cada cruceo, se hace uso

de la geometría del mismo y de los vectores de posición de los nodos y esquinas. Considerando dos nodos en la secuencia de ruta óptima del algoritmo *Dijkstra*  $p_k$  y  $p_{k+1}$  las dos esquinas de  $p_k$  se obtienen a partir de la siguiente ecuación vectorial:

$$p_k^j = \overrightarrow{F_I p_k} + \left[ \left| \overrightarrow{p_k p_k^j} \right| \cos(\rho), \left| \overrightarrow{p_k p_k^j} \right| \sin(\rho) \right] \quad (8)$$

Donde  $\overrightarrow{F_I p_k}$  es el vector de posición con componentes  $[x_k, y_k]$  que ubica el nodo  $p_k$  con respecto al sistema global  $F_I$ , el segundo término de la ecuación 8 corresponde al vector de posición que va del nodo  $p_k$  a cualquiera de las tres subesquinas que definen los segmentos de trayectoria en el cruce, los módulos  $\left| \overrightarrow{p_k p_k^j} \right| = \sqrt{(0.2)^2 + (1.29)^2} = 1.3$  para  $j = 0, 1, 2, \dots, 7$  son todos iguales, consecuentemente  $\rho$  se define como:

$$\rho = \tan^{-1} \left( \frac{\Delta y_{p_k p_{k+1}}}{\Delta x_{p_k p_{k+1}}} \right) + \lambda \quad (9)$$

Donde  $\Delta y_{p_k p_{k+1}} = y_k - y_{k+1}$  y  $\Delta x_{p_k p_{k+1}} = x_k - x_{k+1}$ .

Por otra parte, dependiendo del valor de  $\lambda$  y de los nodos  $p_k$  y  $p_{k+1}$  la ecuación 8 proporciona las coordenadas de la esquina correspondiente a la trayectoria que se busca. Ahora sabiendo que todos los segmentos de las trayectorias en los cruces comienzan siempre en  $p_k^0$  el cual se calcula por medio de la ecuación 8 considerando a  $\lambda$  como:

$$\lambda = \pi - \gamma = \pi - \tan^{-1} \left( \frac{0.2}{0.29} \right) \quad (10)$$

Para percatarse si el segmento de trayectoria es recto o hay una vuelta en el cruce se utiliza la siguiente sentencia, que considera tres nodos consecutivos  $p_{k-1}$ ,  $p_k$  y  $p_{k+1}$ :

$$\tan^{-1} \left( \frac{\Delta y_{p_{k-1} p_k}}{\Delta x_{p_{k-1} p_k}} \right) = \tan^{-1} \left( \frac{\Delta y_{p_k p_{k+1}}}{\Delta x_{p_k p_{k+1}}} \right) \quad (11)$$

Si la igualdad de la ecuación 11 se cumple la trayectoria es recta, por lo tanto, se debe calcular  $p_{k+1}^0$  considerando a  $\lambda$  igual a la expresión 12.

No obstante, si la igualdad 11 no se cumple, se presenta una vuelta, por lo tanto, se calculan las esquinas  $p_k^1$  o  $p_k^5$  dependiendo si es una vuelta a la derecha o a la izquierda.

Para esto usamos la misma expresión 8 con una  $\lambda$  igual:

$$\lambda = \alpha = \tan^{-1} \left( \frac{0.2}{0.29} \right) \quad (12)$$

Cabe mencionar que en el anterior cálculo el resultado será  $p_k^1$  si el nodo  $p_{k+1}$  está a la derecha de  $p_k$  y a su vez será  $p_k^5$  si  $p_{k+1}$  está a la izquierda de  $p_k$ , esta decisión se ejecutará por el elemento  $\tan^{-1} \left( \frac{\Delta y_{p_k p_{k+1}}}{\Delta x_{p_k p_{k+1}}} \right)$  de la expresión 9, por lo tanto, no hay necesidad de hacer otro cálculo.

Realizando el anterior procedimiento para cada uno de los nodos en  $d_{AB}^*$  que es la secuencia de nodos proporcionados por *Dijkstra* para el camino óptimo, obtenemos así finalmente todos los elementos de la trayectoria óptima  $T_{AB}$ .

### 3. Control de la navegación

Una de las principales preocupaciones en los vehículos autónomos es garantizar que el vehículo pueda seguir una trayectoria predefinida, la cual, es una interfaz fundamental entre el sistema de generación de trayectorias y el controlador, se puede definir entonces la trayectoria como una secuencia de segmentos de líneas rectas y curvas para el caso de vueltas que a su vez proporcionan una secuencia de puntos finales (Rajamani, 2011) (Jiang & Astolfi, 2018).

Contar con un buen modelo matemático que describa el movimiento del vehículo con los comandos de dirección, aceleración y freno, es esencial para las tareas de simulación y las de su automatización, particularmente para el desarrollo de la algorítmica de control, así como de la heurística de planificación. En el contexto de este proyecto el modelo fundamental es el que describe la cinemática del vehículo y, para ello, se ha utilizado el llamado *Modelo Cinemático de Bicicleta* que a continuación se describe.

#### 3.1. Modelado del vehículo autónomo

El *Modelo Cinemático de Bicicleta MCB* es un modelo clásico adecuado para representar el movimiento del vehículo en condiciones normales de conducción, además de considerar las restricciones no holonómicas propias del vehículo, es decir, el movimiento del coche está restringido a moverse tan solo hacia adelante o hacia atrás sin poder hacerlo de manera lateral; por tanto, no puede cambiar de orientación libremente sin cambiar de posición, consecuencia de que la orientación de la rueda delantera se puede controlar en relación con la orientación del vehículo (Rajamani, 2011) (Takeyas, 2007) (González Miranda, 2019).

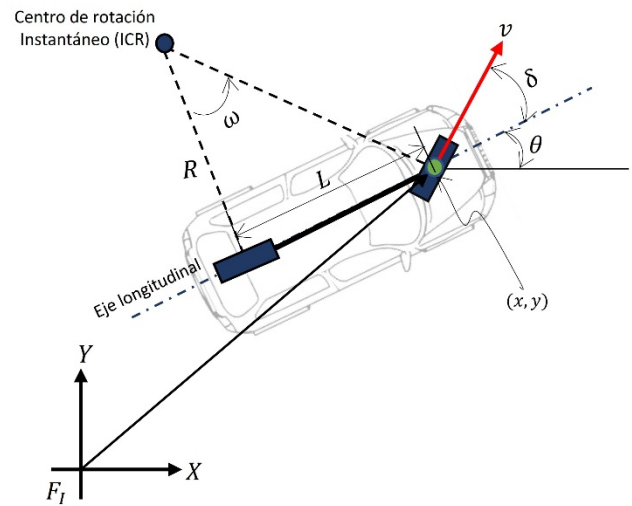


Figura 4: Modelo Cinemático de Bicicleta MCB.

Ahora bien, el uso de este modelo supone que el vehículo se mueve en un plano 2D denotado por el sistema inercial  $F_I$ , así mismo se propone que la dirección Ackerman y las dos ruedas delanteras del coche queden representadas por la rueda delantera de la bicicleta, mientras que la rueda trasera de la bicicleta representa las dos ruedas traseras del coche (Takeyas, 2007) (Corke & Khatib, 2011). Para analizar la cinemática de este modelo se toma como punto de referencia el centro del eje delantero del vehículo, como se muestra en la figura 4.

Se establece  $(x,y)$  para denotar la ubicación del punto de referencia con respecto al sistema inercial y la orientación del automóvil como  $\theta$ . Así,  $[x, y, \theta]$  son los principales estados del sistema, por último se define  $L$  como la longitud de la bicicleta medida entre los ejes de las dos ruedas (Rajamani, 2011) (Ardema, 2004).

Por otra parte, para determinar las entradas del modelo se define el un ángulo de dirección variable para la rueda delantera representado por  $\delta$  y se mide en relación con la dirección hacia adelante de la bicicleta. La velocidad de desplazamiento del vehículo se denota como  $v$  y apunta en la misma dirección de cada rueda, por lo tanto,  $[v, \delta]$  se toman como entradas del sistema (Rajamani, 2011) (Ardema, 2004). También se establece la velocidad de rotación de la bicicleta  $\omega$  alrededor del *Centro de Rotación Instantáneo ICR* por sus siglas en inglés, que es el centro del círculo que se genera cuando el vehículo se mueve en un camino curvo, lo anterior se puede expresar como:

$$\dot{\theta} = \omega = \frac{v}{R} \quad (13)$$

En la ecuación 13,  $R$  es el radio al ICR. Asimismo, con base también en la figura 4, se deduce, que a partir de los triángulos semejantes formados por  $L$  y  $R$ , y  $v$  y  $\delta$ , se obtiene la siguiente expresión:

$$\tan(\delta) = \frac{L}{R} \quad (14)$$

De esta manera, combinando la ecuación 13 y 14, se obtiene:

$$\dot{\theta} = \omega = \frac{v \tan(\delta)}{L} \quad (15)$$

Con las expresiones anteriores y tomando en cuenta que la velocidad, en el eje delantero, apunta en la dirección de desplazamiento de la rueda delantera, dicha orientación se define por la suma de  $\delta$  y  $\theta$ , se puede formar ahora el MCB. Así, los componentes de velocidad del punto de referencia en la dirección  $x$  e  $y$ , con respecto al sistema global y la ecuación para la velocidad de rotación forman el modelo de bicicleta en el punto de referencia establecido, lo anterior se expresa como:

$$\begin{aligned} \dot{x} &= v \cos(\theta + \delta) \\ \dot{y} &= v \sin(\theta + \delta) \\ \dot{\theta} &= \frac{v \tan(\delta)}{L} \end{aligned} \quad (16)$$

### 3.2. Control lateral

Una de las partes fundamentales del control lateral como ya se mencionó anteriormente es adoptar una estrategia de diseño de control adecuada de acuerdo a las características del vehículo autónomo disponible y acorde a los requerimientos necesarios

#### a) Estrategia de diseño de control lateral

La estrategia utilizada se basa en un *Controlador Geométrico Stanley*, el cual consiste esencialmente en un cambio del punto de referencia en el vehículo, es decir, se

pretende seguir una ruta de referencia utilizando únicamente la geometría de la cinemática del vehículo basada en el modelo de bicicleta y en las coordenadas de la ruta de referencia para hacer odometría y así determinar las medidas de los errores (Corke & Khatib, 2011) (Snider & others, 2009) (Hoffmann, Tomlin, Montemerlo, & Thrun, 2007).

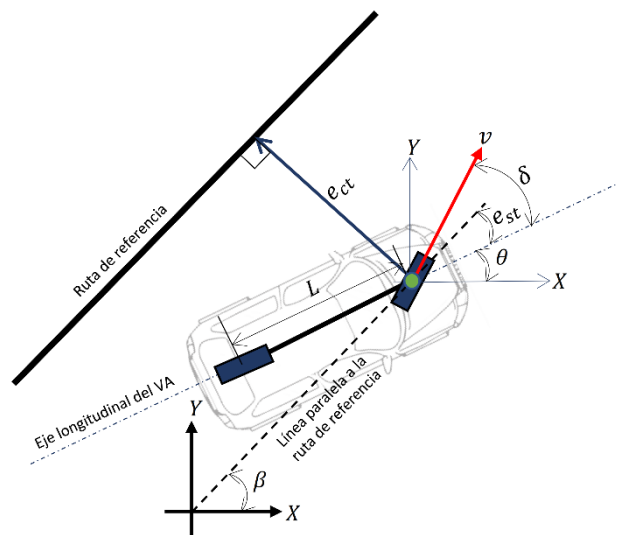


Figura 5: Elementos del control lateral.

En la figura 5 se presentan gráficamente los elementos indispensables para definir los errores necesarios para la implementación del controlador *Stanley*, considerando primeramente un segmento de línea como la ruta de referencia, que se muestra como una línea negra sólida. Una línea negra discontinua que es paralela a la ruta de referencia, pero atraviesa el centro del eje delantero del vehículo, se define también una dirección relativa al segmento de línea de la ruta de referencia  $\beta$ , a esta orientación también se le puede denominar como la dirección deseada a la que tiende el vehículo. La velocidad de la rueda delantera  $v$  y el ángulo de dirección variable  $\delta$  con respecto al rumbo del automóvil, también se muestran.

Los dos tipos de error que se emplearán son:

- *Error de dirección ( $e_{st}$ )*: Es una medida principal de lo bien que el vehículo está alineado y moviéndose en la dirección de la ruta deseada. Se define como la diferencia entre la orientación de la trayectoria deseada  $\beta$  y la dirección del vehículo en el punto de referencia  $\theta$  a lo largo de la ruta, es decir:

$$e_{st} = \beta - \theta \quad (17)$$

- *Error de desviación ( $e_{ct}$ )*: Es la principal medida de cuán cerca está la posición del vehículo  $P$ , con coordenadas  $(x, y)$  de la posición deseada a lo largo de la ruta.

El error de desviación, representado como una flecha roja en la figura 6, que va desde el punto de referencia del vehículo al punto de referencia de la ruta debe ser perpendicular a la misma. Para calcular este error primero se obtienen los módulos de los vectores  $\overline{P_0P}$ ,  $\overline{PP_1}$  y  $\overline{P_0P_1}$  a partir de la resta vectorial de las

magnitudes que los conforman con respecto al sistema inercial  $F_I$ .

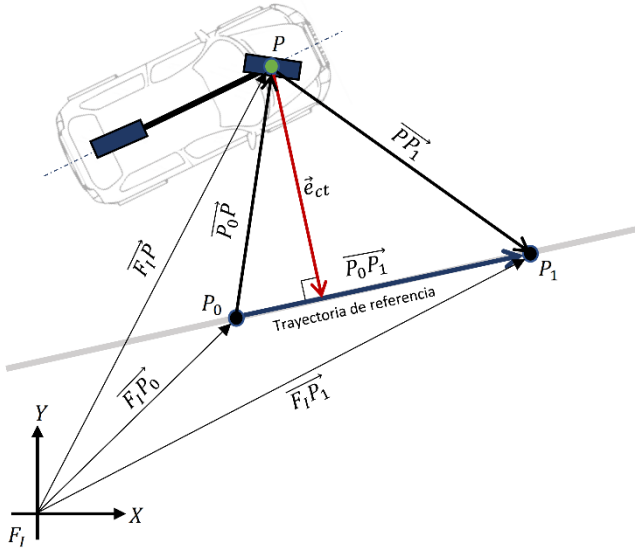


Figura 6: Error de desviación.

En segundo lugar, se determina el  $|\vec{e}_{ct}|$  como la altura del triángulo escaleno con vértices  $P_0$ ,  $P$  y  $P_1$  que va desde la ruta de referencia al punto  $P$ , para esto, se utiliza la *Ecuación de Herón*:

$$|\vec{e}_{ct}| = \frac{2\sqrt{S(S-|\overline{P_0P}|)(S-|\overline{PP_1}|)(S-|\overline{P_0P_1}|)}}{|\overline{P_0P_1}|} \quad (18)$$

Donde  $S$  es el semiperímetro del triángulo, expresado como:

$$S = \frac{|\overline{P_0P}| + |\overline{PP_1}| + |\overline{P_0P_1}|}{2} \quad (19)$$

Tanto el error de dirección como el error de desviación deben converger a cero para que el vehículo esté siguiendo correctamente la ruta deseada.

Para lograr eliminar el error de orientación en relación con la trayectoria, el ángulo de dirección se ajusta directamente al error de dirección, es decir:

$$\delta_1(t) = e_{st}(t) \quad (20)$$

A su vez, para eliminar el error de desviación, se establece un control proporcional, cuya ganancia se multiplica por la inversa de la velocidad de avance. El control se pasa entonces a través de una función arcotangente para mapear la señal de control proporcional con el rango angular de  $-\frac{\pi}{2}$  a  $\frac{\pi}{2}$ , definiendo así la siguiente expresión:

$$\delta_2(t) = \arctan\left(\frac{k_p e_{ct}(t)}{v_f(t)}\right) \quad (21)$$

Por último, el comando de ángulo de dirección se establece para que caiga dentro de los ángulos de dirección mínimo y máximo, que suelen ser simétricos alrededor de  $90^\circ$  es decir  $\delta_2(t) \in [\delta_{\min}, \delta_{\max}]$ .

La ley de control final se obtiene simplemente juntando  $\delta_1(t)$  y  $\delta_2(t)$ , es decir, sumando la ecuación 20 y 21 para establecer el ángulo de dirección del coche de la siguiente manera:

$$\delta(t) = e_{st} + \arctan\left(\frac{k_p e_{ct}(t)}{v(t)}\right), \delta(t) \in [\delta_{\min}, \delta_{\max}] \quad (22)$$

### 3.3. Control longitudinal

Para el control longitudinal del vehículo que particularmente se refiere al control de velocidad, se considera que el auto debe de seguir la trayectoria deseada básicamente a dos diferentes velocidades, una velocidad cruceo  $v_c$  para los segmentos rectos del camino y una velocidad menor, denotada  $v_g$ , para los segmentos curvos de la trayectoria. Para esto se implementa el clásico *Perfil Trapezoidal de Velocidad*, es decir, se establece un control proporcional con saturación en  $v_c$  para las rectas y en  $v_g$  para los giros, de manera general la expresión de este control lateral es la siguiente:

$$v = (v_c + \text{sat}(k_c e_p))SW + (v_g + \text{sat}(k_g e_p))SW \quad (23)$$

Donde  $k_c$  y  $k_g$  son constantes de proporcionalidad y  $e_p$  es igual al valor de  $|\overline{PP_1}|$  referente al error de desviación y mostrado en la figura 6. Considerando la posición del vehículo  $P = (x, y)$  y un punto en trayectoria  $p_k^j$ ,  $e_p$  se expresa como:

$$e_p = \sqrt{(x - x_k^j)^2 + (y - y_k^j)^2} \quad (24)$$

Por otra parte  $SW$  es un switch que puede tomar valores de 1 o 0 si el condicional de la expresión 11 se cumple o no respectivamente, pero ahora la igualdad se aplica a los elementos de la trayectoria.

## 4. Resultados y simulación

En este trabajo hemos considerado que todas las calles están orientadas verticalmente (norte-sur) o bien horizontalmente (este-oeste), es decir, no hay calles diagonales. Se asociará el referencial del mundo a la esquina ubicada más al sur y más al oeste del mapa, cada nodo está representado con un círculo color verde y los arcos se representan como flechas que unen cada nodo, para identificar cada nodo se le asigna un ID formado por un número y una letra correspondiente a la columna y a la fila en donde se encuentre. En la figura 7 se muestra la ciudad considerada y su grafo asociado.

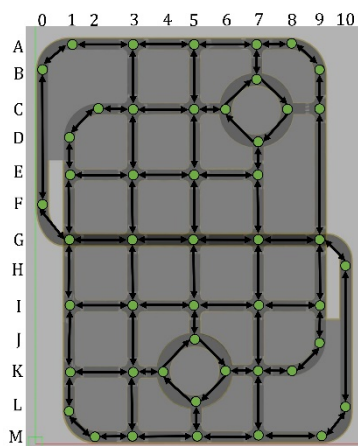
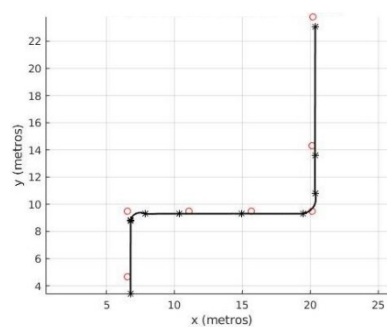
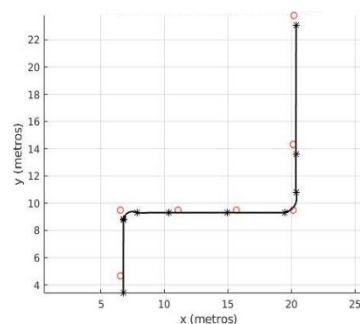


Figura 7: Entorno urbano considerado.



a)



b)

Figura 8: Trayectoria descrita por el vehículo: a) A bajas velocidades  $v_c = 0.337 \text{ m/s}$  y  $v_g = 0.084 \text{ m/s}$ , b) A la máxima velocidad posible:  $v_c = 0.9 \text{ m/s}$  y  $v_g = 0.337 \text{ m/s}$ .

El controlador longitudinal 22 y lateral 23 expuestos anteriormente se implementaron en ROS, con el simulador 3D Gazebo el cual cuenta con muchos modelos de robots, actuadores y sensores entre ellos el AutoMINY desarrollado por la Universidad Libre de Berlín (Berlin, Freie Universität, 2021).

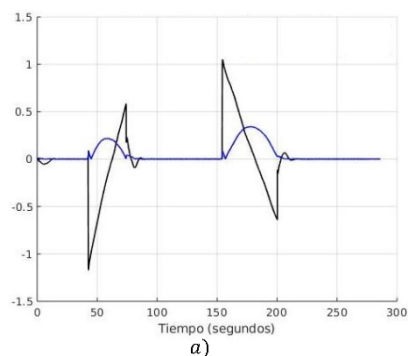
El ancho de cada carril es de  $0.40 \text{ m}$  por lo que el ancho de la carretera es de  $0.80 \text{ m}$ . Los giros a la derecha tienen un radio de  $1.09 \text{ m}$  y las vueltas a la izquierda tiene un radio de  $1.5 \text{ m}$ . Las constantes de proporcionalidad utilizadas fueron  $k_p = 8$  y  $300$  para  $k_c$  y  $k_g$ .

Las pruebas en el simulador del vehículo se hicieron a diferentes velocidades de cruce y de giro, siendo  $v_c = 0.9 \text{ m/s}$ , correspondiente en el simulador a  $800$ , la máxima velocidad alcanzada y  $v_g = 0.337 \text{ m/s}$ , que corresponde a  $300$  en el simulador, para la velocidad máxima de giro. Para fines de comparación también se presentan los resultados de una prueba a bajas velocidades siendo  $v_c = 0.337 \text{ m/s}$ , correspondiente en el simulador a  $300$ , y  $v_g = 0.084 \text{ m/s}$ , que corresponde a  $75$  en el simulador, para la velocidad máxima de giro.

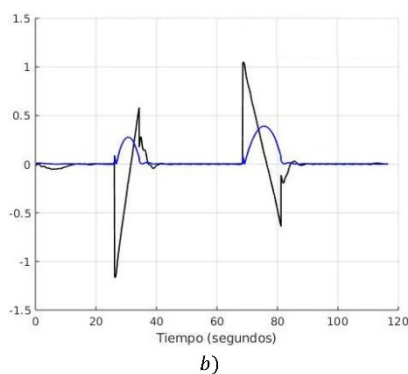
El seguimiento de la trayectoria se presenta en la figura 8, donde los círculos pequeños representan los nodos obtenidos a partir del algoritmo *Dijkstra* para ir del nodo  $K3$  al nodo  $C9$  de acuerdo a la identificación de los nodos expuesta en la figura 7, los \* representan la trayectoria a seguir resultado del planificador implementado, por último, la línea negra representa la trayectoria descrita por el vehículo.

Si bien a la máxima velocidad alcanzada, tanto de giro como de cruce, se pasa por todos los puntos de la trayectoria planeada, sin embargo, a velocidades más altas el auto comienza a pisar las medianas, o a pisar las líneas amarillas de las banquetas, esto sucede particularmente en los giros a la derecha los cuales son más cerrados, por lo tanto, los errores máximos de dirección y desviación se presentan en estas circunstancias, como se observa en la figura 9, pese a esto siempre convergen a cero.

Por otra parte, la señal de control tiene un valor máximo de  $90^\circ$  correspondiente al giro a la derecha y el valor inferior constante de  $-53^\circ$  es el de la dirección que denota un giro a la izquierda, la ley de control rápidamente converge a cero en los trayectos rectos, esto sucede sobre todo cuándo se presentan grandes errores de desviación, como se muestra en la figura 10.



a)



b)

Figura 9: Error de dirección  $e_{st}$  línea negra y error de desviación  $e_{ct}$  línea roja: a) A bajas velocidades  $v_c = 0.337 \text{ m/s}$  y  $v_g = 0.084 \text{ m/s}$ , b) A la máxima velocidad posible:  $v_c = 0.9 \text{ m/s}$  y  $v_g = 0.337 \text{ m/s}$ .

Finalmente, en la figura 11 se puede observar el perfil de velocidad trapezoidal obtenido a partir del controlador longitudinal propuesto en la ecuación 23.



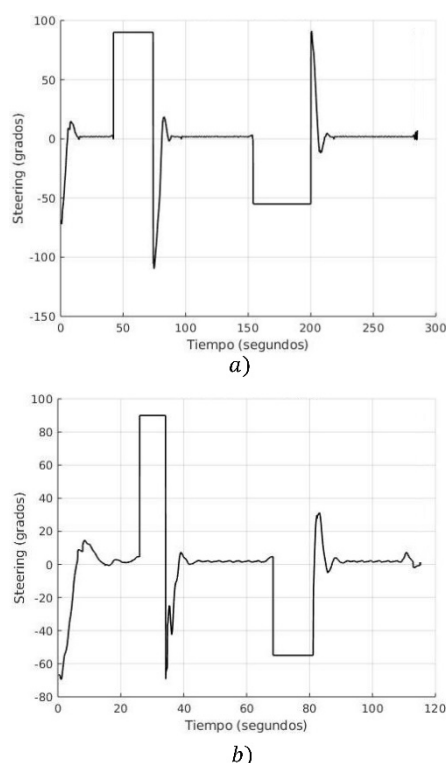


Figura 10: Señal de control  $\delta(t)$ : a) A bajas velocidades  $v_c = 0.337 \text{ m/s}$  y  $v_g = 0.084 \text{ m/s}$ , b) A la máxima velocidad posible:  $v_c = 0.9 \text{ m/s}$  y  $v_g = 0.337 \text{ m/s}$ .

Los resultados fueron bastante buenos como puede apreciarse en el video que hemos subido a Youtube (Ibarra Zannatha, 2022).

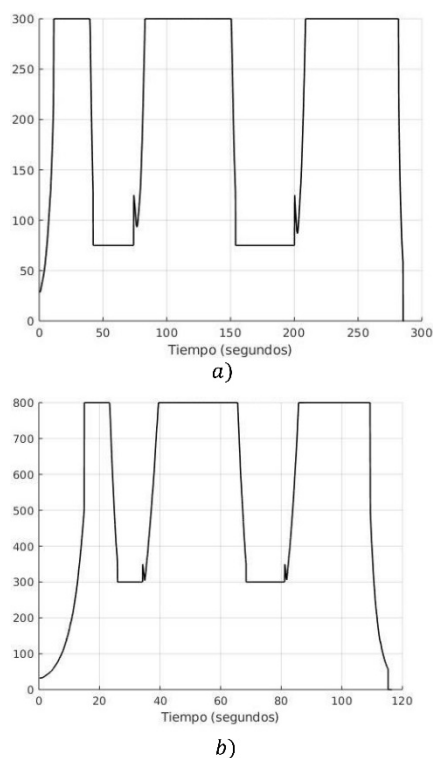


Figura 11: Perfil de velocidad a) A bajas velocidades de entrada  $v_c = 0.337 \text{ m/s}$  y  $v_g = 0.084 \text{ m/s}$  correspondientes a 300 y 75 en el simulador. b) A máximas velocidades de entrada  $v_c = 0.9 \text{ m/s}$  y  $v_g = 0.337 \text{ m/s}$  correspondientes a 800 y 300 en el simulador.

## 5. Comentarios finales y trabajo a futuro

Los resultados obtenidos hasta el momento son bastante buenos, aunque sólo hemos considerado calles rectas y giros a  $90^\circ$ . Sin embargo, ya estamos trabajando el problema de circular por las glorietas de nuestra ciudad virtual y en las curvas de las esquinas de dicha ciudad o bien en diagonales que, por el momento, no existen en nuestra ciudad virtual. Además, se trabaja en la implementación de un controlador lateral visual, con lo cual será más eficiente mantenerse en el carril sin necesidad de depender de la Odometría interna del vehículo autónomo. Además, estamos tratando de simplificar la generación de trayectorias utilizando un algoritmo más apropiado. Finalmente, dentro de los planes del proyecto tenemos el desarrollo de controladores conexionistas, los cuales podrían aprender tanto de un operador humano telecomandando el vehículo como del sistema que hemos reportado en este trabajo.

## Referencias

- Ardema, M. D. (2004). *Newton-Euler Dynamics*. Springer Science & Business Media.
- Berlin, *Freie Universität*. (2021). Obtenido de Autonomos model: <https://github.com/AutoModelCar/AutoModelCarWiki/wiki>
- Bruce A., F., & Manfredi, M. (2016). Flocking and Rendezvous in Distributed Robotics. En S. Cham (Ed.).
- Corke, P. I., & Khatib, O. (2011). *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer.
- Godsil, C., & Royle, G. F. (2001). *Algebraic graph theory* (Vol. 207). (S. S. Media, Ed.)
- González Miranda, O. (2019). Modelado y control de un vehículo autónomo. México: Centro de Investigación y de Estudios Avanzados del IPN, Departamento de Control Automático.
- Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., & Thrun, S. (2007). Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. En *2007 American control conference* (págs. 2296-2301). IEEE.
- Ibarra Zannatha, J. M. (2022). Obtenido de <https://www.youtube.com/channel/UCm3PWkzF363Dw2m32RmBOBw>
- Jiang, J., & Astolfi, A. (2018). Lateral Control of an Autonomous Vehicle. *IEEE Transactions on Intelligent Vehicles*, 3(2), 228-237. DOI:10.1109/TIV.2018.2804173
- LaValle, S. M. (2006). *Planning algorithms*. (C. u. press, Ed.)
- Otero, C., Paz, E., López, J., Barea, R., Romera, E., Molinos, E., others. (2017). Simulación de vehículos autónomos usando v-rep bajo ros. (S. d. Oviedo, Ed.) *Actas de las XXXVIII Jornadas de Automática*.
- Rajamani, R. (2011). *Vehicle dynamics and control*. (S. S. Media, Ed.)
- Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: A modern approach* (3rd ed.). (P. Norvig, Ed.) Pearson.
- Snider, J. M., & others. (2009). Automatic steering methods for autonomous automobile path tracking. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*.
- Takeyas, B. L. (2007). *Introducción a la inteligencia artificial*. (I. T. Laredo, Ed.) Recuperado el 3 de Septiembre de 2022, de <http://www.itnuevolaredo.edu.mx/takeyas>
- Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Practice, Society of Automobile Engineers SAE. Recuperado el 7 de Septiembre de 2022, de [www.sae.org/standards/content/j3016\\_201806/](http://www.sae.org/standards/content/j3016_201806/)
- World Motor Vehicle Production: 2021 Production Statistics*. Report, International Organization of Motor Vehicle Manufacturers OICA. Recuperado el 10 de Agosto de 2022, de <https://www.oica.net/category/production-statistics/2021-statistics/>