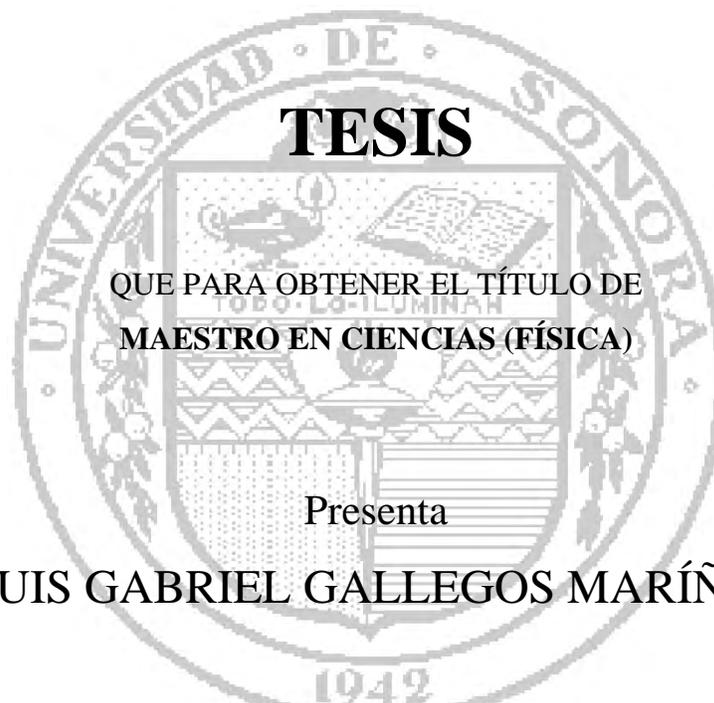


UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO DE INVESTIGACION EN FÍSICA
PROGRAMA DE POSGRADO EN CIENCIAS (FÍSICA)

MODELADO, GENERACIÓN DE RUTA Y CONTROL DE
UN VEHÍCULO TERRESTRE TIPO DIFERENCIAL



TESIS

QUE PARA OBTENER EL TÍTULO DE
MAESTRO EN CIENCIAS (FÍSICA)

Presenta

LUIS GABRIEL GALLEGOS MARÍÑEZ

Bajo la dirección de

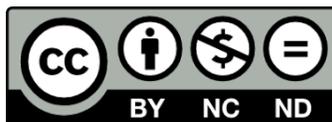
DR. LUIS ARTURO GARCÍA DELGADO

Universidad de Sonora

Repositorio Institucional UNISON



"El saber de mis hijos
hará mi grandeza"



Excepto si se señala otra cosa, la licencia del ítem se describe como openAccess

Modelado, generación de ruta y control de un vehículo terrestre tipo diferencial

Luis Gabriel Gallegos Maríñez
Universidad de Sonora
Director de tesis:
Dr. Luis Arturo García Delgado

Contenidos

Agradecimientos	iii
Resumen	v
Abstract	vii
Prefacio	ix
Introducción	xi
1 Marco teórico	1
1.1 Movimiento de un cuerpo rígido	1
1.1.1 Sistemas de referencia	1
1.1.2 Ecuaciones de Newton-Euler	3
1.2 Teoría de control lineal en espacio de estado	4
1.3 Linealización de sistemas	9
1.4 Potenciales artificiales	10
1.5 Algoritmos de búsqueda	12
1.5.1 Algoritmo de búsqueda de Dijkstra	14
1.5.2 Algoritmo de búsqueda A*	15
2 Metodología	17
2.1 Análisis cinemático	17
2.2 Análisis dinámico	20
2.3 Modelos matemáticos	23
2.4 Seguimiento de trayectoria	25
2.5 Controlador de potencial artificial	26
2.6 Controlador de DeLuca	27
2.7 Búsqueda de ruta	28
3 Montaje Experimental	33
3.1 Hardware	33
3.2 Software	34
3.3 Esquema experimental	35

4 Resultados	37
4.1 Resultados experimentales	37
4.2 Prueba de controlador de DeLuca	37
4.3 Prueba de controlador de potencial artificial	39
4.4 Búsqueda de ruta con seguimiento de trayectoria	41
Conclusiones	45
A Matrices ortogonales y ángulos de Euler	47
B Cuaterniones	51
B.1 Introducción	51
B.2 Rotaciones con cuaterniones	53
B.3 Ecuaciones de movimiento con cuaterniones	54
C Interpolación	57
C.1 Interpolación polinomial	58
C.2 Splines	58
D Sistemas reactivos por medio de esquemas	61
E Código fuente	63
Lista de símbolos	67

Agradecimientos

Quiero agradecer la oportunidad que me ha brindado CONACYT de estudiar una maestría y realizar este trabajo, ampliando mi conocimiento científico y elevar mi formación como miembro de la sociedad.

También quiero agradecer al departamento de posgrado en Física con sus profesores tan dedicados en la enseñanza.

Por ultimo agradezco a mis padres que me apoyaron en todo momento para conseguir mis metas y superarme.

Resumen

En este trabajo se presenta un problema de la robótica móvil, que es el encontrar un camino óptimo en tiempo real desde una posición inicial de un vehículo hasta un destino seleccionado. Se desarrollará un modelado dinámico mecánico y uno cinemático para el caso de un vehículo terrestre sobre ruedas. Se muestran algunos algoritmos necesarios para el guiado del vehículo terrestre en un entorno bidimensional con obstáculos estáticos formados por polígonos que el vehículo terrestre evitará tanto en simulación como experimentalmente.

Palabras Clave

Vehículo autónomo, algoritmo de Dijkstra, robótica móvil, búsqueda de ruta, inteligencia artificial, potenciales artificiales.

Abstract

In this work it is presented a subject in the mobile robotics, which is to find a optimal path in real-time for a vehicle from its initial position to its desired goal. A dinamical mechanical and a kinematical model is developed for the case of a wheeled ground vehicle. Some required algorithms are presented for the guidance of the ground vehicle in a two-dimensional environtment with static obstacles aproximated by polygons which must be evaded by the vehicle in simulations and experimentally.

Keywords

Autonomous vehicle, Dijkstra algorithm, mobile robotics, pathfinding, artificial intelligence, artificial potential fields.

Prefacio

La realización de un vehículo autónomo es una tarea que requiere el uso de teorías y técnicas de varias disciplinas. Se requieren suficientes conocimientos mecánicos para realizar un modelo dinámico o cinemático del vehículo para poder caracterizar su respuesta al entorno y a sus actuadores. Esta respuesta es usada para elaborar simulaciones computacionales y también obtener información cualitativa del vehículo, información que es usada para desarrollar estrategias de control como las usadas en teoría de control que aseguren un comportamiento deseable. Una vez que se tenga esto se implementan algoritmos en tiempo real con estrategias de control utilizando la infraestructura necesaria. Para un principiante en el tema es difícil encontrar los conocimientos necesarios para su realización experimental ya que éstos se encuentran en disciplinas muy variadas.

Este trabajo está dirigido a personas que empiezan a estudiar sobre la autonomía en robótica móvil y requieren algún ejemplo concreto, completo y sencillo sobre los pasos que hay que seguir para la realización de un sistema autónomo. Se muestran múltiples formas para solucionar los diferentes problemas que se presentan en la realización de propuestas para guiado autónomo de un vehículo terrestre.

Se hace hincapié en los conceptos físicos y matemáticos requeridos para obtener un modelo mecánico lo suficientemente preciso para su uso experimental. También se muestran algunos algoritmos conocidos en ciencias computacionales necesarios para obtener una autonomía inteligente pero sencilla en los vehículos. Así mismo se hace una introducción a la teoría de control moderna necesaria para el control de posición y postura del vehículo en algunos enfoques.

Introducción

Los agentes autonomos

La *autonomía* (del griego auto, "uno mismo", y nomos, "norma") es un concepto moderno, procedente de la filosofía y, más recientemente, de la psicología, que, en términos generales, expresa la capacidad para darse reglas a uno mismo o tomar decisiones sin intervención ni influencia externa. Esto en el concepto de la robótica y automatización lleva a sistemas realizados por el hombre que tienen cierto grado de esta autonomía, siendo capaces de mantenerse en funcionamiento y realizar sus funciones por largos periodos de tiempo con poca o hasta sin intervención humana, esto implica un grado de inteligencia artificial. Estos sistemas autónomos, mejor llamados robots autónomos son capaces de realizar gran cantidad de tareas recibiendo órdenes de alto nivel, como indicarles destinos, patrones de movimiento o acciones complejas de forma que éste decida por sí mismo la mejor forma de realizarlas. Un vehículo autónomo, o también conocido como AGV por sus siglas en inglés (*Autonomous Guided Vehicle*), es un vehículo con un control automático capaz de imitar las capacidades humanas de manejo y control. Este tipo de vehículos posee medios sensoriales y de navegación tanto de sistemas externos como GPS, estaciones de transmisión y circuitos de cámaras o mediante sistemas abordo como sistemas de visión artificial, mapas, acelerómetros, giroscopios, tacómetros, radares, LIDAR, detectores de proximidad infrarrojos, entre otros, de forma que el controlador dentro del vehículo procesa en conjunto todos los medios disponibles para dirigir el vehículo. Para que el controlador interno del vehículo logre su objetivo, éste debe ser capaz de manejar todos sus grados de libertad mediante sus actuadores, sean turbinas, ruedas, hélices, etc. [1]. Los vehículos remotamente controlados son otro tipo de vehículo no tripulado, donde su control se relega a operadores remotos humanos por lo que no son considerados autónomos. En muchos casos el control del vehículo es preferible que se realice dentro del mismo ya que las capacidades de comunicación poseen retrasos inevitables que pueden llevar a la inestabilidad, además de requerir un gran volumen de datos y atención continua.

El primer vehículo autónomo fue mostrado en la exposición universal de 1939 por General Motors, un vehículo eléctrico controlado por un circuito eléctrico en el pavimento de la carretera. En 1980 el fabricante de automóviles Mercedes-Benz con un equipo de la universidad de Munich presentó una furgoneta guiada por visión que alcanzó 100 km/h en calles sin tráfico. En 1980 también DARPA (*Defense Advanced Research Projects Agency*) realizó su primer vehículo autónomo con radar laser y visión por computadora. En 1987 los laboratorios HRL (*Hughes Research Laboratories*) construyeron su vehículo que podía generar su propia ruta con un mapa. Pudo moverse 600 metros a través de terreno complejo con pendientes, rocas y vegetación. En 1994, dos vehículos VaMP y Vita-2 de Daimler-Benz y

Ernst Dickmans condujeron solos más de 1000 km en una autopista en París con 3 carriles en día de tráfico intenso hasta una velocidad de 130 km/h. En 1995 este mismo equipo modificó un Mercedes-Benz clase S para un viaje entre las ciudades de Munich y Copenhague de ida y vuelta, utilizando visión computarizada con movimientos sacádicos en tiempo real, los movimientos sacádicos en el caso de robots consisten en rápidos movimientos en la cámara de forma continua. El robot alcanzó 175 km/h en la Autobahn alemana, solo interviniendo los conductores en promedio cada 9 km, esto es 95% de la conducción fue autónoma. En 2014 Audi anuncia su modelo RS7 autónomo, alcanzó los 240 km/h en el circuito de Hockenheim en Alemania. En octubre de 2014 la empresa Tesla Motors integró en su automóvil eléctrico Tesla Model S hardware que permite una función de autopiloto al vehículo, éste incluye un radar frontal y un arreglo de sensores ultrasónicos, así mismo se asiste de GPS y mapas digitales para su guiado, este sistema permite mantenerse dentro de un carril, mantener velocidad y distancia a los demás vehículos en la carretera, entre otros. En verano del 2015 la Universidad de Michigan puso en marcha un proyecto en el que se construyó y puso en funcionamiento un pueblo de utilidad llamado MCity en cuyas calles se pueden probar vehículos autónomos, cuenta con una calle de más de 1.5km de largo, curvas, semáforos, etc. Uno de otros proyectos complementarios es la puesta en marcha de 9000 vehículos interconectados en la ciudad de Ann Arbor, en Michigan. Así como otros 20000 en carreteras en el suroeste del estado [1]. Los vehículos guiados autónomamente han existido en la industria desde 1953. Éstos al principio se basaban en vehículos remolcadores industriales que se guiaban con campos magnéticos en el suelo, siendo generados por imanes permanentes o cables eléctricos previamente instalados en el entorno que se planea que el vehículo circule [2]. Los AGV son usados en aplicaciones tales como redes de transporte dentro de las fábricas en forma de vehículos de carga completamente automatizados que realizan sus tareas en conjunto de forma ordenada, como los que se muestran en la Figura 1. Éstos cuentan con sistemas de guiado dentro de entornos controlados como las fábricas, por medio de sistemas de guiado por láser con reflectores en donde dichos vehículos calculan su posición en todo momento dentro del lugar usando reflectores posicionados en las paredes o vigas a lo largo de su recorrido, usándolos de referencia por medio de un sensor láser abordo. También existen otros que son guiados por medio de pequeños imanes permanentes colocados en el suelo, detectándolos con sensores magnéticos, o incluso líneas pintadas en el suelo detectadas con sensores infrarrojos. El AGV entonces utiliza éstos u otros métodos a la vez para guiarse de una forma segura y robusta dentro del espacio de trabajo [3].



Figura 1: AGV de transporte de material industrial

Estado del arte

Una de las formas más recientes para resolver el problema del guiado autónomo se basa principalmente en subdividir el problema en subproblemas, dos de ellos son la búsqueda de camino y el seguimiento de trayectoria. El seguimiento de trayectoria consta de la creación de un algoritmo de control que se comunica directamente con los motores del robot de forma que éste siga una trayectoria trazada. Debido a que el seguimiento de trayectoria suele ser un problema no lineal para vehículos terrestres con ruedas, es común utilizar controladores especializados para esta tarea. Distintas propuestas actuales para resolver el problema de seguimiento son, entre otras, basadas en *modos deslizantes* [4] [5], *backstepping* [6] [7], *lógica difusa* [8] [9], *redes neuronales* [10] [11], y *potenciales artificiales* [12] [13] [14]. Los controladores basados en potenciales artificiales utilizados para el seguimiento de trayectoria consisten en la creación de potenciales ficticios sobre el área donde se desplazará el vehículo y hacer que éste se comporte como si fuera afectado por éste potencial artificial. El campo potencial artificial genera una fuerza que guía el vehículo hacia su destino, tal como ocurre con una carga eléctrica negativa (el vehículo) atraída hacia un lugar donde hay carga positiva (el destino). Este método tiene cierta similitud que el método de guiado con campos magnéticos en los robots industriales, pero en el caso de los potenciales artificiales éstos son completamente simulados (no existen campos físicos reales en el lugar) y libres de tomar las formas que se quiera.

El uso de potenciales artificiales se publicó por primera vez en [15], enfocado principalmente para robots manipuladores. Se planteó como una forma resolver el problema de evitar colisiones en tiempo real, evitando la utilización de técnicas de planeación de alto nivel que requieren generalmente un tiempo de procesamiento mucho mayor que el tiempo de respuesta de los vehículos. Después en [16] se dio una mayor libertad en la definición de los potenciales artificiales, en los llamados sistemas reactivos basados en esquemas, que al considerarlos desde el punto de vista de la inteligencia artificial se lograba extender sus capacidades. Los campos aquí podrían cambiar en el tiempo según la situación y adoptar formas más arbitrarias, una breve introducción a este tema se da en el Anexo D sobre sistemas reactivos por medio de esquemas. Publicaciones recientes con el uso de potenciales artificiales presentan resultados de simulaciones computacionales sobre potenciales artificiales específicos, como son [17] y [18].

Para el caso de resolver el problema de búsqueda de ruta se ha optado por utilizar algoritmos de alto nivel que suelen discretizar el área de trabajo en celdas cuadrículadas. El algoritmo busca un camino entre dos celdas dadas, utilizando un árbol de búsqueda. Los árboles de búsqueda son estructuras de datos en teoría de la computación, hechas por un conjunto de nodos interconectados desde un nodo *raíz* hasta los nodos *hoja* como si fuera un árbol, donde cada nodo puede poseer un conjunto de datos asociados. Entonces el algoritmo de búsqueda de ruta va buscando celda por celda, desde la celda inicial, el mejor camino de celdas para llegar a una celda destino, formando un árbol de búsqueda donde sus nodos corresponden con celdas y donde cada camino desde la raíz a cada hoja se ve como un potencial camino hacia el destino. Al final se decide cuál es el mejor camino, al elegir el camino entre la raíz y una hoja destino más corto. No sólo se ha probado con áreas de trabajo constituidas por celdas cuadrículadas, también se puede subdividir el área de formas arbitrarias, por ejemplo formando grafos. En [18] se sientan las bases del primer algoritmo formal para resolver el problema de planeación de ruta esto de forma óptima, llamado el algoritmo de Dijkstra, que tiempo después en [19], se logra mejorar su rendimiento en base a métodos heurísticos que utilizan conocimientos sabidos de antemano para reducir el tiempo de búsqueda. Aunque estas publicaciones no hacen énfasis en una aplicación particular, han sido ampliamente adoptadas en el ámbito de la robótica móvil y a partir de ahí se generaron una multitud de variaciones del algoritmo de Dijkstra, como son: el Algoritmo D* [20], theta* [21], LPA* [22] entre otros. Siendo por ejemplo el algoritmo D* enfocado en la robótica móvil. En [23] se hace una discusión de la familia de algoritmos más recientes en su aplicaciones para robótica móvil.

Objetivos de la tesis

En este trabajo se presenta un problema muy común en la robótica móvil: encontrar un camino óptimo en tiempo real desde una posición inicial de un AGV terrestre con ruedas hasta el destino seleccionado. Se supondrá que el entorno es una superficie bidimensional conocida que contiene obstáculos, que se definen como lugares impasables para el vehículo. El vehículo, después de recibir la ubicación de destino instruida por el operador, debe ser capaz de poder evadir por sí mismo los obstáculos y llegar a su destino sin ninguna intervención humana. El AGV debe obtener por su cuenta un camino realizable por él mismo y determinar las entradas de los actuadores para seguir dicho camino. Para realizar el guiado autónomo se requiere desarrollar el modelado matemático-mecánico del vehículo autónomo terrestre y diseñar controladores realimentados para su guiado autónomo. En este trabajo se propone utilizar un guiado autónomo híbrido, donde se diseñaran potenciales artificiales en conjunto con el algoritmo de búsqueda de caminos de Dijkstra, de forma que las limitaciones de los potenciales artificiales, los mínimos locales y los caminos subóptimos, sean superadas al utilizar el algoritmo de Dijkstra, que asegura además caminos óptimos. Para resolver este problema se propuso subdividirlo en una serie de objetivos específicos:

- Obtener las ecuaciones de movimiento para un vehículo terrestre con ruedas utilizando la formulación de Newton-Euler.
- Utilizar estrategias de control basadas en potenciales artificiales como medio para el seguimiento de trayectoria de un vehículo.

- Realizar programas de computadora que implementen algoritmos de búsqueda de caminos basados en árboles de búsqueda.
- Desarrollar programas para simulación computacional, que incluyan el modelo mecánico y las estrategias de control.
- Realizar experimentos que evalúen el desempeño de las estrategias de control .

Organización de la tesis

El Capítulo 1 da una introducción a la teoría necesaria para resolver el problema planteado en su conjunto, empezando por las ecuaciones de movimiento del vehículo, pasando por la teoría de control y finalizando con los algoritmos computacionales necesarios para su ejecución. El Capítulo 2 muestra los desarrollos matemáticos para el caso específico del vehículo que se controlará. Además se muestran las soluciones encontradas a los problemas presentados para la implementación de las estrategias de control utilizando la teoría del Capítulo 1. El Capítulo 3 explica el arreglo experimental realizado. En el Capítulo 4 se exponen y se discuten los resultados de las simulaciones y los experimentos. También se añaden cinco anexos necesarios para explicar ciertos aspectos teóricos utilizados en el desarrollo de este trabajo.

Capítulo 1

Marco teórico

En esta sección se presentan los aspectos teóricos que se necesitan para modelar y controlar un robot móvil rodado. Se comienza estudiando los principios de las ecuaciones de movimiento para cuerpos rígidos. En seguida se presentan las bases del control lineal por realimentación de estado. Después se presenta una sección dedicada a la linealización de sistemas. También se muestran las bases de la teoría de campo potencial, para finalizar el capítulo hablando sobre algoritmos de búsqueda.

1.1 Movimiento de un cuerpo rígido

Un cuerpo rígido consta de un sistema de partículas que tienen la restricción de mantener la distancia fija entre cada una de ellas, debido a esto el movimiento de un cuerpo rígido posee 6 grados de libertad en el espacio, mientras no posea alguna restricción de movimiento. Estos 6 grados de libertad se pueden escoger de forma que 3 de ellos sean las coordenadas de posición de algún punto del cuerpo y los otros 3 sean un conjunto de parámetros que expresen una rotación alrededor del punto elegido. Entonces, cualquier movimiento de un cuerpo rígido se puede expresar como la combinación de una translación y una rotación alrededor de algún punto [24].

1.1.1 Sistemas de referencia

Antes de hablar del comportamiento de un cuerpo rígido es necesario establecer un sistema de referencia inercial, desde el cual se realizarán las mediciones. Se define un sistema de referencia inercial ortogonal de tres dimensiones, \mathcal{I} , expresado por los tres vectores base:

$$\hat{x}, \hat{y}, \hat{z} \tag{1.1}$$

También se define otro sistema de coordenadas ortogonal, \mathcal{B} , fijo al cuerpo rígido con su origen dentro del mismo, ver Figura 1.1, conformado por los vectores base. Este sistema de referencia sirve para caracterizar por completo la orientación del cuerpo con respecto al marco \mathcal{I} . El marco de referencia del cuerpo, \mathcal{B} , es un sistema generalmente no-inercial ya que se ve afectado por aceleraciones debido a los actuadores. Los vectores que conforman el marco inercial se denotan mediante:

$$\hat{x}', \hat{y}', \hat{z}' \tag{1.2}$$

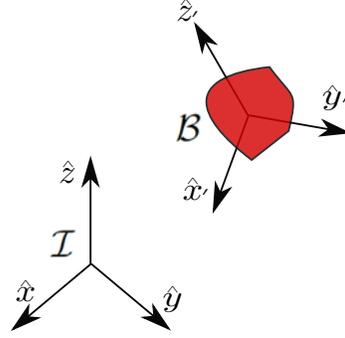


Figura 1.1: Marcos de referencia

El marco \mathcal{B} tiene entonces una rotación igual que la del cuerpo y presenta tanto una velocidad en su origen como una rotación en sus vectores base que siguen al cuerpo. Existe una relación entre los vectores del marco \mathcal{B} y los del marco \mathcal{I} dada por una matriz de rotación \mathbf{R} que transforma los componentes de un vector en \mathcal{B} a sus componentes correspondientes en el marco \mathcal{I} , el Anexo A desarrolla un resumen sobre el tema de las rotaciones y cómo se caracterizan por medio de los ángulos de Euler. Así mismo el Anexo B desarrolla una forma alternativa para tratar las rotaciones, utilizando cuaterniones y desarrollando las ecuaciones de movimiento en base a éstos.

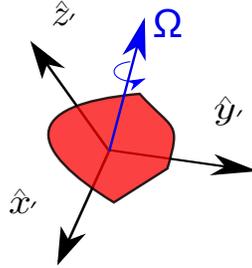


Figura 1.2: Sistema de referencia en rotación

Denotamos con $\boldsymbol{\Omega}$ la velocidad angular vectorial del sistema de referencia \mathcal{B} con respecto al marco inercial \mathcal{I} como se muestra en la Figura 1.2. Cuando se tiene un sistema de referencia en movimiento general como el sistema \mathcal{B} , es conveniente poder expresar la razón de cambio de cualquier vector \mathbf{Q} que se tiene expresado en el sistema \mathcal{B} , pero visto desde el sistema inercial \mathcal{I} [24]. Esto se puede realizar mediante la ecuación

$$\frac{d\mathbf{Q}}{dt} = \left(\frac{d\mathbf{Q}}{dt} \right)_{\mathcal{B}} + \boldsymbol{\Omega} \times \mathbf{Q} \quad (1.3)$$

donde $\left(\frac{d\mathbf{Q}}{dt} \right)_{\mathcal{B}}$ es la velocidad del vector medida en el marco \mathcal{B} considerando sus vectores base como constantes, $\boldsymbol{\Omega}$ es el vector velocidad angular del marco \mathcal{B} y donde \times denota el producto cruz. Para expresar el vector de velocidad \mathbf{V}_P de algún punto P en el marco \mathcal{B} , se utiliza la ecuación

$$\mathbf{V}_P = \mathbf{V}_o + \boldsymbol{\Omega} \times \mathbf{r}_{P/o} + \mathbf{V}_{P/o} \quad (1.4)$$

donde \mathbf{V}_o es la velocidad del origen del marco \mathcal{B} , $\mathbf{V}_{P/o}$ la velocidad del punto P medida desde el marco \mathcal{B} y $\mathbf{r}_{P/o}$ denota el vector posición que va desde el origen del marco \mathcal{B} hasta P [24].

1.1.2 Ecuaciones de Newton-Euler

Un conjunto de fuerzas sobre un sistema de partículas tiene el efecto de mover su centro de masa como si éste fuera una sola partícula con toda la masa del sistema concentrada en ella, de acuerdo a la segunda ley de Newton, además la sumatoria de torques alrededor del centro de masa de un sistema de partículas tiene la propiedad de ser igual al cambio en su momento angular con respecto al centro de masa [24]. Las ecuaciones que describen el efecto de la aplicación de fuerzas y torques alrededor del centro de masa de un cuerpo son:

$$\dot{\mathbf{P}} = \frac{d(m\mathbf{v}_G)}{dt} = \sum \mathbf{F}_i \quad (1.5)$$

$$\dot{\mathbf{L}} = \frac{d(\mathbf{I} \cdot \boldsymbol{\Omega})}{dt} = \sum \mathbf{N}_i \quad (1.6)$$

donde \mathbf{P} denota el momento lineal del cuerpo, \mathbf{v}_G denota la velocidad de su centro de masa, m es la masa del cuerpo, $\boldsymbol{\Omega}$ su velocidad angular, \mathbf{L} su momento angular, e \mathbf{I} es el tensor de inercia del cuerpo. \mathbf{F}_i y \mathbf{N}_i denotan las fuerzas y torques aplicados al cuerpo, respectivamente. Cuando los ejes del marco de referencia corresponden con los ejes principales de inercia del cuerpo rígido, el tensor de inercia adopta una forma diagonal[25], como se muestra en la siguiente ecuación

$$[\mathbf{I}]' = \begin{bmatrix} I'_{11} & 0 & 0 \\ 0 & I'_{22} & 0 \\ 0 & 0 & I'_{33} \end{bmatrix}$$

Si fijamos un marco de referencia \mathcal{B} al centro de masa del cuerpo rígido de forma que los ejes de este marco correspondan con los ejes principales de inercia, podemos utilizar la Ecuación (1.3), con el cambio del vector momento angular del cuerpo, de la siguiente manera

$$\dot{\mathbf{L}} = \left(\frac{d\mathbf{L}}{dt} \right)_{\mathcal{B}} + \boldsymbol{\Omega} \times \mathbf{L}$$

Esto forma un sistema de ecuaciones diferenciales conocidas como ecuaciones de Euler para un sólido rígido [24]. Escribiendo estas ecuaciones con componentes de $\boldsymbol{\Omega}$ expresadas en base a los ejes del marco \mathcal{B} se tiene:

$$N'_x = I'_{11}\dot{\Omega}'_x + (I'_{33} - I'_{22})\Omega'_y\Omega'_z \quad (1.7)$$

$$N'_y = I'_{22}\dot{\Omega}'_y + (I'_{11} - I'_{33})\Omega'_x\Omega'_z \quad (1.8)$$

$$N'_z = I'_{33}\dot{\Omega}'_z + (I'_{22} - I'_{11})\Omega'_x\Omega'_y \quad (1.9)$$

estas tres ecuaciones junto con las ecuaciones:

$$F_x = ma_x \quad (1.10)$$

$$F_y = ma_y \quad (1.11)$$

$$F_z = ma_z \quad (1.12)$$

donde a_x, a_y, a_z son las aceleraciones del origen del marco \mathcal{B} , definen el comportamiento dinámico del cuerpo rígido.

1.2 Teoría de control lineal en espacio de estado

Conociendo el sistema de ecuaciones dinámico del cuerpo rígido de las Ecuaciones (1.9)-(1.12), se desea llevar las variables de este sistema a un estado deseado, o en otras palabras, se requiere que algunas de sus variables describan una función específica con respecto al tiempo. Por ejemplo se puede requerir que el centro de masa del cuerpo rígido siga una trayectoria específica en el espacio. Esto se puede expresar como la necesidad de que $x_G = x_G(t), y_G = y_G(t), z_G = z_G(t)$, que son las coordenadas del centro de masa del cuerpo, teniendo la relación

$$F_x(t) = m \frac{d^2 x_G(t)}{d^2 t} \quad (1.13)$$

$$F_y(t) = m \frac{d^2 y_G(t)}{d^2 t} \quad (1.14)$$

$$F_z(t) = m \frac{d^2 z_G(t)}{d^2 t} \quad (1.15)$$

que se obtiene de las Ecuaciones (1.10)-(1.12). Entonces si se tiene control directo de las fuerzas aplicadas al centro de masa, se puede llevar el cuerpo rígido por la trayectoria deseada al manipular la fuerza F_x, F_y, F_z según las Ecuaciones (1.13) - (1.15). La teoría que trata esta clase de problemas se le llama teoría de control. La teoría de control es una rama interdisciplinaria de las matemáticas y la ingeniería que trata sobre hacer que sistemas dinámicos dados se comporten de una forma deseable. Los sistemas dinámicos son sistemas que producen un conjunto de variables de salida que dependen de los estados internos actuales y anteriores del sistema y de un conjunto de variables de entrada proporcionados a éste. Los sistemas dinámicos pueden ser sistemas físicos, químicos, sociales, etc, normalmente descritos por ecuaciones diferenciales o ecuaciones en diferencias.



Figura 1.3: Diagrama de bloques de un sistema dinámico de 3 entradas y 2 salidas

En la Figura 1.3 se muestra un ejemplo de un sistema dinámico representado en diagrama de bloques. En el caso de que un sistema dinámico cumpla con ciertas características de linealidad entre sus entradas y sus salidas, como la proporcionalidad y superposición, se dice que es un sistema dinámico lineal. El control lineal es un tema ampliamente investigado en el siglo XX, en especial la teoría de control clásica basada en el análisis de sistemas lineales en la frecuencia compleja por medio de la transformada de Laplace [26]. En la teoría de control moderna, más específicamente en la representación en el espacio de estado, se representa un

sistema de ecuaciones diferenciales de primer orden por medio de matrices y vectores. Un vector de estado \mathbf{x} es un vector columna con un conjunto variables de un sistema tales que su conocimiento presente, así como el conocimiento de las entradas, nos permite predecir la evolución futura del mismo. La forma estándar de representar un sistema lineal con variables de estado es por medio de un sistema de ecuaciones diferenciales de primer orden como el siguiente [27]

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (1.16)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (1.17)$$

donde \mathbf{x} , \mathbf{y} y \mathbf{u} son vectores columna correspondientes al vector de estado, vector de salida y vector de entrada, respectivamente, y donde $\dot{\mathbf{x}}$ es la derivada del vector de estado con respecto al tiempo. El vector de salida está formado por las variables de salida de interés, una por renglón, y de igual forma el vector de entrada esta formado por las variables de entrada. Las matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} son matrices de coeficientes constantes, llamadas matriz de estado, matriz de entrada, matriz de salida y matriz de transmisión directa. Estas matrices definen el comportamiento dinámico del sistema. Un ejemplo sencillo de un sistema dinámico que toma esta forma seria la del oscilador armónico simple, éste puede ser una masa m sujeta a un resorte con el otro lado del resorte sujeto a un punto fijo. Además se supone que una fuerza externa F afecta a la masa. La ecuación diferencial de segundo orden que lo define es

$$m\ddot{x} = -kx + F \quad (1.18)$$

donde k es la constante del resorte y x representa la posición de la masa con respecto a su posición de equilibrio. Para escribir esta ecuación en la misma forma de las Ecuaciones (1.16) y (1.17) definimos una variable $x_2 = \dot{x}$ que representa la velocidad de la masa y definimos $x_1 = x = x_1$. La variable que nos interesa puede ser x por lo que definimos la salida como $y = x$. Con esta definición de variables podemos escribir la Ecuación (1.18) como

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{k}{m} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F \quad (1.19)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (1.20)$$

aquí podemos notar, comparando con las Ecuaciones (1.16) y (1.17) que

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ \frac{k}{m} & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{D} = 0, \quad \mathbf{u} = F$$

Nótese que la matriz \mathbf{A} tiene un tamaño $n \times n$, donde n es igual al orden del sistema. Esta representación en espacio de estado es una representación mínima del sistema ya que un sistema de orden n necesita al menos n variables para representarlo con este método, además dejando claro que la representación de espacio estado para un sistema no es única. La solución general con una condición inicial $\mathbf{x}(0) = \mathbf{x}_0$ para un sistema en espacio de estado es [26]

$$\begin{aligned} \mathbf{x}(t) &= \exp(\mathbf{A}t)\mathbf{x}_0 + \int_0^t \exp(\mathbf{A}(t-\tau))\mathbf{B}\mathbf{u}(\tau)d\tau \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \quad (1.21)$$

donde la operación $\exp(\mathbf{A}t)$ se define como

$$\exp(\mathbf{A}t) = \sum_{n=0}^{\infty} \frac{(\mathbf{A}t)^n}{n!} \quad (1.22)$$

La Ecuación (1.21) es similar a la conocida solución

$$x(t) = e^{kt}x_0 + \int_0^{\infty} e^{k(t-\tau)}u(\tau)d\tau$$

para la ecuación diferencial escalar $\dot{x} = kx$. A la Ecuación (1.22) también se le llama matriz de transición de estado, también expresada como $\exp(\mathbf{A}t) = \Phi(t)$ que transforma el estado $\mathbf{x}(t_0)$ en un tiempo t_0 al estado $\mathbf{x}(t_1) = \Phi(t_1 - t_0)\mathbf{x}(t_0)$ en el tiempo t_1 cuando las entradas son cero. Los eigenvalores $p_i, i = 1, 2, 3, \dots$ con eigenvectores correspondientes \mathbf{x}_{p_i} de \mathbf{A} determinan el comportamiento transitorio y la estabilidad del sistema de ecuaciones, a estos eigenvalores se les denomina *polos*. Para simplificar la discusión se presupone que no existe multiplicidad en los eigenvalores, es decir, todos son diferentes. Siendo $\exp(\mathbf{A}t)$ una función polinomial de \mathbf{A} , ésta conserva los mismos eigenvectores \mathbf{x}_{p_i} pero con eigenvalores correspondientes por $e^{p_i t}$. Esto se puede comprobar al sustituir \mathbf{x}_{p_i} en la Ecuación (1.22). Suponiendo que $\mathbf{u} = 0$ y tomando en cuenta lo anterior, podemos obtener que para una condición inicial igual a uno de los eigenvectores $\mathbf{x}_0 = \mathbf{x}_{p_i}$, que la Ecuación (1.21) resulta en

$$\begin{aligned} \mathbf{x}(t) &= e^{p_i t} \mathbf{x}_{p_i} \\ \mathbf{y}(t) &= e^{p_i t} \mathbf{C} \mathbf{x}_{p_i} \end{aligned}$$

Por lo tanto, si se descompone una condición inicial \mathbf{x}_0 arbitraria en la base de los eigenvectores de la matriz de transición, de forma que

$$\mathbf{x}_0 = \sum_{i=1}^n a_i \mathbf{x}_{p_i}, i = 1, 2, \dots, n \quad (1.23)$$

donde a_i son constantes y n denota la cantidad de polos del sistema, entonces, debido a que la multiplicación de matrices es una operación lineal, se puede llegar sustituyendo (1.23) en (1.21) a que la solución para $\mathbf{x}(t)$ es

$$\mathbf{x}(t) = \sum_{i=1}^n e^{p_i t} \mathbf{x}_{p_i}, i = 1, 2, 3, \dots, n \quad (1.24)$$

$$\mathbf{y}(t) = \sum_{i=1}^n e^{p_i t} \mathbf{C} \mathbf{x}_{p_i} \quad (1.25)$$

En el caso donde $\mathbf{u} \neq 0$, los polos también afectan de forma similar la salida $\mathbf{y}(t)$ ante condiciones iniciales. Haciendo las condiciones iniciales cero $\mathbf{x}_0 = 0$ y con una entrada \mathbf{u} tal que $\mathbf{B}\mathbf{u} = \delta(t)\mathbf{x}_{p_i}$ (o si es posible, $\mathbf{u} = \delta(t)\mathbf{B}^{-1}\mathbf{x}_{p_i}$) se consigue que el segundo termino en la Ecuación (1.21) sea

$$\begin{aligned} \int_0^{\infty} \exp(\mathbf{A}(t - \tau))\mathbf{B}\mathbf{u}(\tau)d\tau &= \int_0^{\infty} \exp(\mathbf{A}(t - \tau))\mathbf{x}_{p_i}\delta(\tau)d\tau \\ &= \exp(\mathbf{A}(t))\mathbf{x}_{p_i} = e^{p_i t}\mathbf{x}_{p_i} \end{aligned}$$

que es una solución idéntica a la de una condición inicial \mathbf{x}_{p_i} . Entonces al hablar de los efectos que tienen los polos en la condición inicial también se obtiene información sobre la solución ante entradas con forma de un impulso $\delta(\tau)$. En el caso de entradas más generales, por ejemplo $\mathbf{B}\mathbf{u} = f_i(t)\mathbf{x}_{p_i}$ siendo f_i funciones arbitrarias continuas, la solución es la convolución de $\mathbf{B}\mathbf{u}(t)$ y la solución al impulso del sistema el polo p_i

$$\begin{aligned} \int_0^\infty \exp(\mathbf{A}(t-\tau))\mathbf{x}_{p_i}f_i(\tau)d\tau &= \left(\int_0^\infty e^{p_i(t-\tau)}f_i(\tau)d\tau\right)\mathbf{x}_{p_i} \\ &= (e^{p_it} * f_i(t))\mathbf{x}_{p_i} \end{aligned}$$

Los polos son números complejos, definimos su parte real como σ_i y su parte imaginaria como ω_i , por lo que $p_i = \sigma_i + \omega_i j$ (donde la unidad imaginaria es j). Dependiendo de las partes real e imaginaria de los polos se puede tener comportamientos completamente diferentes para las Ecuaciones (1.24) y (1.25). Por ejemplo, dado un polo p_i y su factor correspondiente

$$e^{p_it} = e^{(\sigma_i + \omega_i j)t} = e^{\sigma_it} e^{\omega_ijt}$$

se puede ver que e^{ω_ijt} es una función oscilatoria con una frecuencia ω_i de oscilación, mientras que e^{σ_it} es una función que crece o decrece exponencialmente en el tiempo según σ_i , creciendo para $\sigma_i > 0$ y decreciendo para $\sigma_i < 0$. Si alguno de los polos del sistema tiene una parte real $\sigma_i > 0$ entonces el sistema es inestable, ya que para una entrada que contenga el eigenvector \mathbf{x}_{p_i} , la solución $\mathbf{x}(t) \rightarrow \infty$ cuando $t \rightarrow \infty$ [26]. En cambio, si se cumple que para todos los polos $\sigma_i < 0$, la solución tenderá asintóticamente al vector nulo $\mathbf{0}$, se dice en ese caso que es *asintóticamente estable*.

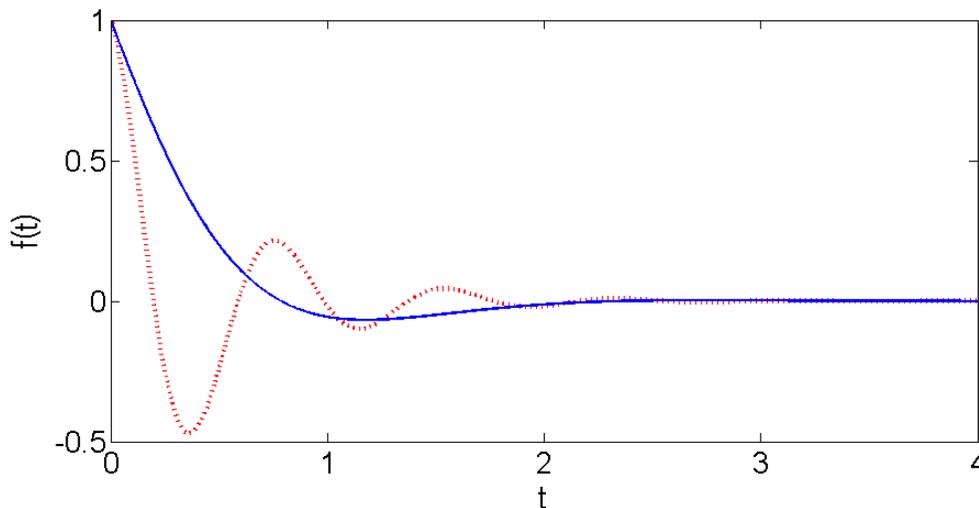


Figura 1.4: $f(t) = \text{Re}(e^{(\sigma+\omega j)t})$, con $\sigma = -2$ y $\omega = 2$ para la línea sólida y para la línea punteada $\sigma = -2$ y $\omega = 8$

Para casos con $\sigma_i < 0$ y $\omega_i \neq 0$ la parte real de la función $e^{(\sigma_i + \omega_i j)t}$ toma formas de oscilaciones amortiguadas como las de la Figura 1.4, haciendo el valor de $\|\sigma_i\|$ mayor se consiguen salidas que decaen más rápido con el tiempo. Por tanto los polos con menor $\|\sigma_i\|$

dominan el tiempo de decaída a cero de la salida del sistema, ya que la contribución de los términos correspondientes a polos más rápidos en la Ecuación (1.24) desaparecen más rápidamente con el tiempo. Además debe notarse que las oscilaciones nunca se detienen, sólo decrecen en magnitud. Siguiendo con el ejemplo del oscilador armónico simple de las Ecuaciones (1.19) y (1.20), tenemos que los eigenvalores de su matriz de estado son $p_1 = \sqrt{\frac{k}{m}j}$, $p_2 = -\sqrt{\frac{k}{m}j}$, siendo éstos sus polos. Al ser su $\sigma = 0$ para ambos polos, la amplitud de las oscilaciones no crece ni decrece con el tiempo, manteniéndose con oscilaciones estables. Sus eigenvectores son:

$$\mathbf{x}_{p1} = \begin{bmatrix} 1 \\ \sqrt{\frac{k}{m}j} \end{bmatrix} \quad (1.26)$$

$$\mathbf{x}_{p2} = \begin{bmatrix} 1 \\ -\sqrt{\frac{k}{m}j} \end{bmatrix} \quad (1.27)$$

La evolución de $\mathbf{x}(t)$ queda dada por la Ecuaciones (1.24) y (1.25); cuando $F = 0$ queda dada por:

$$\mathbf{x}(t) = a_1 e^{\sqrt{\frac{k}{m}j}t} \begin{bmatrix} 1 \\ \sqrt{\frac{k}{m}j} \end{bmatrix} + a_2 e^{-\sqrt{\frac{k}{m}j}t} \begin{bmatrix} 1 \\ -\sqrt{\frac{k}{m}j} \end{bmatrix} \quad (1.28)$$

$$\mathbf{y}(t) = a_1 e^{\sqrt{\frac{k}{m}j}t} + a_2 e^{-\sqrt{\frac{k}{m}j}t} \quad (1.29)$$

donde a_1 y a_2 son constantes que dependen del estado inicial según la Ecuación (1.23), siendo los componentes del estado inicial del sistema en la base de los eigenvectores de la matriz de estados \mathbf{A} .

Conociendo las características dinámicas de los sistemas la teoría de control busca crear un sistema de control que manipule las entradas de un sistema para producir una salida $\mathbf{y}(t)$ con características requeridas. Una forma de resolver el problema de control utilizando el llamado control en lazo abierto, que es crear un sistema de control con un vector de entrada $\mathbf{r}(t)$ que representa el estado de salida $\mathbf{y}(t)$ deseado y que cuyas salidas se conectan directamente a las entradas $\mathbf{u}(t)$ del sistema a controlar. El problema más común es lograr que $\mathbf{y}(t)$ sea lo más cercano posible a $\mathbf{r}(t)$ en todo momento, a este problema se le llama *seguimiento de referencia* (en inglés *reference tracking*). Conociendo las características dinámicas del sistema a controlar de antemano, se programa el controlador para generar el vector $\mathbf{u}(t)$ de señales de control necesarias para *seguir* la entrada de referencia $\mathbf{r}(t)$, siendo $\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{r}(t)$ el *error de seguimiento*.

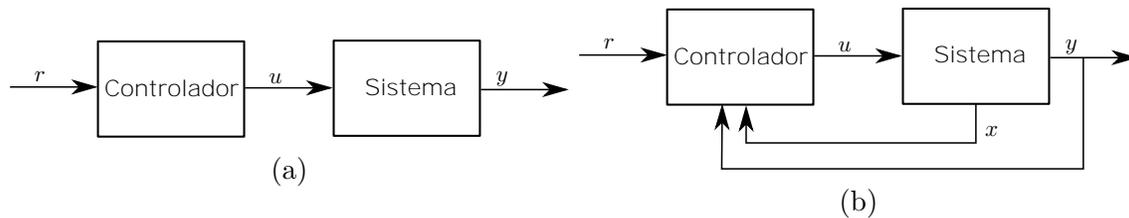


Figura 1.5: (a) Sistema controlado en lazo abierto; (b) sistema controlado en lazo cerrado.

En sistemas relativamente sencillos no sujetos a perturbaciones impredecibles, el sistema de control en lazo abierto es un buen método para lograr obtener la salida $\mathbf{y}(t)$ deseada. En el caso de que el comportamiento dinámico del sistema se conozca de forma imprecisa y/o esté sujeto a perturbaciones externas impredecibles, este método de resolución puede dar resultados erróneos. El método de control en lazo cerrado permite corregir este tipo de errores, ya que basa su estrategia en utilizar también la salida \mathbf{y} y en ocasiones las variables internas \mathbf{x} medibles del sistema para corregir el error del sistema a niveles aceptables. En la Figura 1.5 se muestra en diagramas de bloques un esquema de sistema de control en lazo abierto (a), y en lazo cerrado (b), para el caso de una entrada una salida.

El control en lazo cerrado mediante realimentación de estado se puede resumir en encontrar un vector de entrada \mathbf{u} de la forma

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (1.30)$$

donde la matriz \mathbf{K} se selecciona de una manera adecuada tal que asigne los polos del sistema resultante en lazo cerrado

$$\begin{aligned} \dot{\mathbf{x}} &= (\mathbf{A} - \mathbf{BK})\mathbf{x} \\ \mathbf{y} &= (\mathbf{C} - \mathbf{DK})\mathbf{x} \end{aligned}$$

con parte real negativa, velocidades de respuesta rápidas, errores de seguimiento acotados y reducidos. Estos objetivos son conflictivos y se debe llegar a un equilibrio aceptable entre ellos. A esta forma de control se le pueden agregar también términos adicionales relacionados con las entradas en el caso de lazo abierto, para reducir aún más el error.

1.3 Linealización de sistemas

La mayoría de los sistemas físicos presentan una naturaleza no-lineal, que al idealizarlos y simplificarlos, toman una forma lineal que se aproxima lo suficiente a resultados experimentales. Una representación para sistemas de ecuaciones diferenciales no lineales es

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (1.31)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (1.32)$$

donde $\mathbf{f}(\mathbf{x}, \mathbf{u}, t) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$ es una función no lineal con n variables de estado y m variables de entrada. Una forma de analizar el comportamiento de un sistema como este es por medio de linealización, de forma que se pueda utilizar la teoría de control lineal. Si se desea que un sistema como el anterior siga una trayectoria de referencia $\mathbf{x}_r(t)$ se puede obtener una aproximación de primer orden del comportamiento del sistema (1.31) con $\mathbf{x} = \Delta\mathbf{x}(t) + \mathbf{x}_r$ para el caso de pequeñas desviaciones $\Delta\mathbf{x}(t)$ y $\Delta\mathbf{u}(t)$ con respecto a $\mathbf{x}_r(t)$ y $\mathbf{u}_r(t)$ respectivamente, por medio del gradiente de $\mathbf{f}(\mathbf{x}, \mathbf{u}, t)$ como sigue

$$\mathbf{f}(\mathbf{x}_r + \Delta\mathbf{x}, \mathbf{u}_r + \Delta\mathbf{u}, t) \approx \mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t) + \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t)(\Delta\mathbf{x}) + \nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t)(\Delta\mathbf{u}) \quad (1.33)$$

donde $\nabla_{\mathbf{x}}$ y $\nabla_{\mathbf{u}}$ son los operadores gradiente con respecto a \mathbf{x} y \mathbf{u} respectivamente. Aquí debe notarse que $\mathbf{u}_r(t)$ son las entradas exactas necesarias para obtener $\mathbf{x}(t) = \mathbf{x}_r(t)$ en

todo momento y se pueden obtener despejando (si es posible) \mathbf{u} en la Ecuación (1.31) con $\mathbf{x}(t) = \mathbf{x}_r(t)$. Al sustituir (1.33) en (1.31) y reordenando, se obtiene

$$\dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t) = \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t)(\Delta\mathbf{x}) + \nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t)(\Delta\mathbf{u})$$

Teniendo en cuenta que $\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t) = \dot{\mathbf{x}}_r(t)$ se puede obtener

$$(\Delta\dot{\mathbf{x}})(t) = \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t)(\Delta\mathbf{x}) + \nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t)(\Delta\mathbf{u}) \quad (1.34)$$

un sistema completamente lineal y cuyo vector de estado es $\Delta\mathbf{x}$ y con entrada $\Delta\mathbf{u}$. Este método se conoce como linealización *aproximada*. La matriz de estado de este nuevo sistema es obtenida por $\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t)$ y la matriz de entrada $\nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x}_r, \mathbf{u}_r, t)$. El vector $\Delta\mathbf{x}$ representa la diferencia entre la trayectoria de referencia y el valor real de la variable \mathbf{x} . Este tipo de sistemas linealizados son una buena aproximación para $\Delta\mathbf{x}$ pequeños, en especial porque si el estado inicial del sistema \mathbf{x}_0 se encuentra lo suficientemente cerca de \mathbf{x}_r y los polos tienen parte real negativa, el sistema tiende a minimizar el error $\Delta\mathbf{x}$ mejorando cada vez más la aproximación. Si no existen perturbaciones grandes, $\Delta\mathbf{x}$ convergerá a 0 y por tanto \mathbf{x} converge a \mathbf{x}_r , este es un hecho conocido en el estudio de ecuaciones diferenciales no lineales [28].

Otra manera de resolver el problema es buscar una matriz \mathbf{T} y una función $\mathbf{g}(\mathbf{x}, t)$, tal que al aplicarse una función de control de la forma [29]

$$\mathbf{u}(t) = \mathbf{T}\mathbf{w}(t) + \mathbf{g}(\mathbf{x}, t) \quad (1.35)$$

se obtenga un sistema lineal de ecuaciones en el nuevo vector de entrada \mathbf{w} . A este método se le llama linealización *exacta* por realimentación de estado. Este sistema presenta un comportamiento lineal entre las entradas y salidas, es exacto (no es una aproximación como en el caso anterior) y puede resolverse utilizando los métodos de teoría de control lineal.

1.4 Potenciales artificiales

En este trabajo se presenta una forma de utilización de los potenciales artificiales para el guiado autónomo del vehículo, en esta técnica se ve al agente inteligente como una especie de partícula cargada expuesta a un potencial artificial $V_a(x, y)$ generado debido al entorno y otros agentes inteligentes en el espacio de trabajo, donde (x, y) representa la posición del agente. El agente entonces emulará, utilizando sus actuadores, el comportamiento de una partícula cargada afectada por fuerzas debidas a estos potenciales, análogamente al caso de las fuerzas electromagnéticas o gravitacionales.

Comúnmente el potencial artificial es generado debido a la suma de varios potenciales individuales. Cada uno de estos potenciales puede tener formas arbitrarias y son diseñados para atraer al agente a los lugares donde se desea o prefiere tener y alejarlo de lugares con peligro de colisión u otros inconvenientes [17]. Los potenciales artificiales entonces, pueden resolver simultáneamente tanto el problema de búsqueda de ruta de una forma subóptima, como el problema de determinar la señal de control de los actuadores para seguir la ruta deseada.

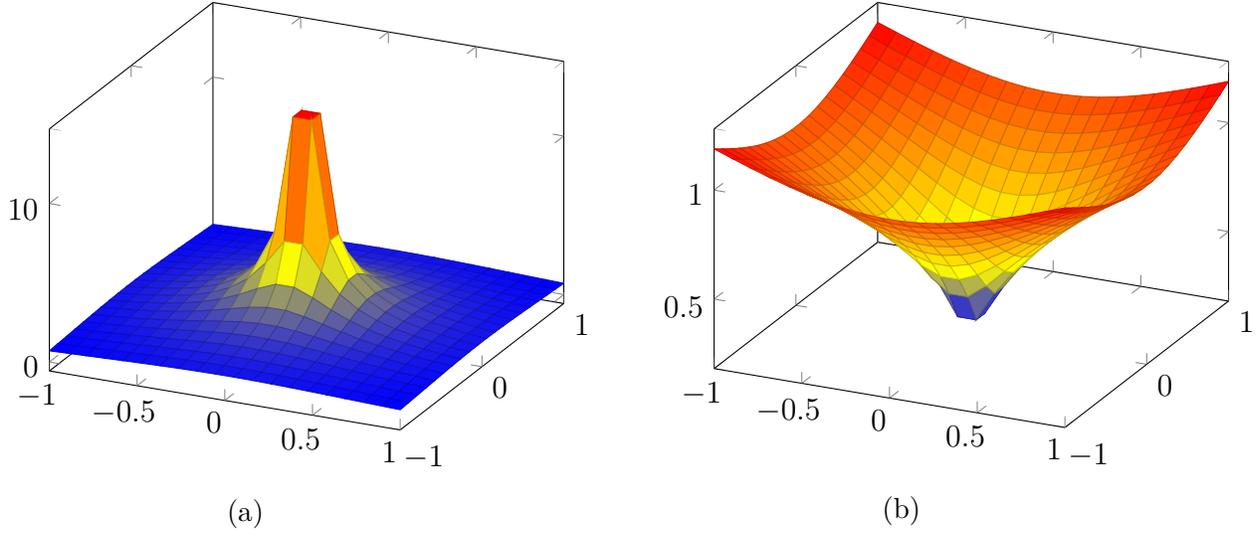


Figura 1.6: (a) Función de potencial artificial repulsiva $V_a = \frac{1}{r^2}$; (b) Función de potencial artificial atractiva $V_a = \sqrt{r}$

El enfoque general para el uso de potenciales artificiales es crear un potencial alto alrededor de cada objeto de interés, el potencial es una función de la distancia definida como

$$r = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$

desde el objeto i hasta el agente. Para el caso de obstáculos se generan potenciales repulsivos y para el lugar destino un potencial atractivo. Para simplificar se considera al agente como una masa puntual unitaria. Debido a que las velocidades del agente en el caso de un vehículo están limitadas a las capacidades de sus actuadores, el vehículo puede no ser capaz de mantener estos potenciales artificiales válidos, además, en el caso de ser capaz de lograrlos, puede llevar a velocidades innecesariamente grandes y a sus correspondientes peligros. Por esto se agrega una fuerza de amortiguación al vehículo, que depende de la velocidad [15]. Esta fuerza tenderá a limitar la velocidad del vehículo dentro de rangos aceptables. La fuerza debida a un potencial i se puede escribir como

$$\mathbf{F}_i = -\nabla V_i(x, y)$$

mientras que la fuerza de disipación se puede expresar como alguna de las siguientes funciones

$$\mathbf{F}_v = -k_v \mathbf{v} \quad (1.36)$$

$$\mathbf{F}_v = -k_v \|\mathbf{v}\| \mathbf{v} \quad (1.37)$$

donde k_v es una constante. Las funciones de las Ecuaciones (1.36) y (1.37) están inspiradas en las ecuaciones de fuerza de arrastre aerodinámica. La fuerza total que actúa sobre el vehículo se define como

$$\mathbf{F} = \sum_{i=1}^N \mathbf{F}_i + \mathbf{F}_v$$

A continuación se listan algunos ejemplos de funciones potenciales:

$$V_a = k\frac{1}{r} \quad (1.38)$$

$$V_a = \begin{cases} \frac{1}{2}k(\frac{1}{r} - \frac{1}{r_1})^2 & r \leq r_1 \\ 0 & r > r_1 \end{cases} \quad (1.39)$$

$$V_a = \begin{cases} c - kr & r \leq r_1 \\ 0 & r > r_1 \end{cases} \quad (1.40)$$

$$V_a = \frac{1}{2}kr^2 \quad (1.41)$$

$$V_a = kr \quad (1.42)$$

$$V_a = kr^{\frac{1}{2}} \quad (1.43)$$

donde c , k y r_1 son constantes que se seleccionan para cada problema. La función de la Ecuación (1.38) representa un potencial electromagnético para cargas unitarias. En este trabajo se trata al vehículo como una carga positiva, por lo que potenciales ascendentes repelen al agente (véase también Ecuaciones (1.39) y (1.40)) [18]. Para el caso de un lugar destino se tiene un potencial descendente en dirección del destino, tal es el caso de los potenciales dados en las Ecuaciones (1.41), (1.42) y (1.43).

Además de obstáculos y puntos destino, existen más utilidades para los potenciales artificiales. Existe una corriente de investigación sobre búsqueda de ruta, control reactivo y potenciales artificiales en robots, íntimamente relacionada con la psicología y la neurología [16], esta corriente se basa en el concepto de *esquemas*. Para una introducción al control basado en esquemas vease el Anexo D.

Debido al problema de la generación de mínimos locales, muy conocidos en optimización de funciones, el agente puede quedar atrapado en uno de estos mínimos locales sin haber llegado al mínimo global (el destino). Esta es una falla que se debe al carácter local de este método, por tal razón este método no se suele utilizar solo. Una forma conocida para resolver este problema es que el agente dé pequeños pasos aleatorios para que éste pueda salir de los mínimos locales. Una técnica más eficiente es hacer que el agente genere un rastro de potenciales repulsivos, de forma que evite localizaciones recientes, empujando hacia adelante al agente cuando está lejos del mínimo global. Este rastro es similar a los de feromonas en las hormigas [30].

1.5 Algoritmos de búsqueda

Debido al problema de los mínimos locales y al hecho que no se asegura que el camino hacia el destino sea el más corto, los potenciales artificiales se suelen ayudar de algoritmos de alto nivel para asegurar un buen desempeño. Los algoritmos de búsqueda son aptos para esta tarea ya que aseguran un camino óptimo (teóricamente) y sin interrupciones. Entonces la tarea para tales algoritmos es la de planear el movimiento de un vehículo para encontrar un camino posible en un entorno con obstáculos. Este problema corresponde al problema de *búsqueda de rutas* y consiste en que dado un agente inteligente (robot móvil) en una

localización inicial, se requiere encontrar un camino factible para éste, a través del entorno para llegar a un punto destino. El entorno puede ser tanto uno físico como uno virtual y lo llamaremos *espacio de trabajo*. En la Figura 1.7 se muestra un vehículo en un espacio de trabajo obstruido por obstáculos. Otros ejemplos de búsqueda de rutas pueden ser encontrar el camino más rápido para un paquete de datos en una red de computadoras, encontrar el movimiento que debe realizar un personaje de videojuego para llegar a una localización en su entorno virtual de forma segura o planear un viaje de una ciudad a otra con el menor coste posible.

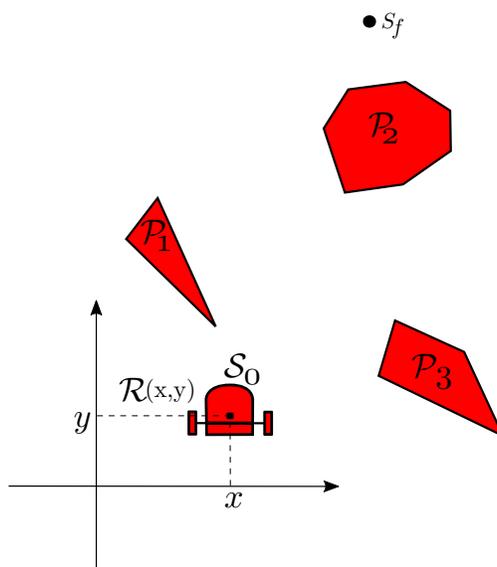


Figura 1.7: Espacio de trabajo con vehículo y obstáculos, \mathcal{P}_i representa el conjunto de puntos que comprenden el obstáculo i y $\mathcal{R}(x, y)$ el área que ocupa el vehículo.

Existen varias formas de representar este problema, unas consisten en discretizar el espacio de trabajo en puntos de interés, entre los cuales el agente se moverá, otros toman el espacio de trabajo como un continuo. Una forma de discretizar el espacio de trabajo es mediante un grafo ponderado donde sus nodos corresponden a los lugares de interés y los enlaces corresponden a caminos posibles entre estos lugares, cada enlace además tiene asociado un peso no negativo que representa distancia, costo o penalización asociado a viajar a través del enlace. Entonces, un algoritmo de búsqueda de ruta busca una secuencia o camino de nodos (lugares) que conecten un nodo inicial con un nodo destino y cuya suma de pesos en los enlaces recorridos es mínima [23]. La Figura 1.8 muestra un ejemplo de grafo ponderado y su camino más corto.

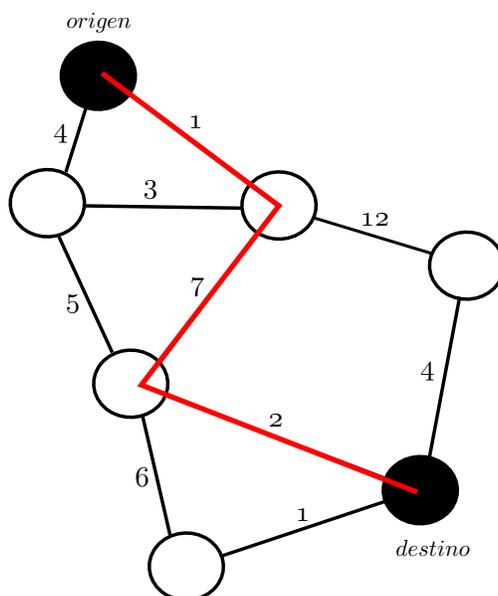


Figura 1.8: Ejemplo de grafo ponderado y su camino más corto.

En el ejemplo de viajar de una ciudad a otra, una red de nodos se puede formar con los caminos y las ciudades de forma que los nodos corresponden a ciudades y los enlaces corresponden a carreteras y con su función de costo que puede representar para un enlace tanto la longitud de éste, el costo monetario o el tiempo necesario en recorrerlo, dependiendo de qué se requiera optimizar, de forma que se busca ya sea, el camino más corto, el más barato o el más rápido respectivamente.

1.5.1 Algoritmo de búsqueda de Dijkstra

Un tipo de enfoque para resolver el problema de búsqueda de rutas se basa en el algoritmo de Dijkstra [31]. El algoritmo de Dijkstra puede escribirse en pseudocódigo como en el Algoritmo 1. El algoritmo requiere un nodo inicial S_0 , un nodo destino S_f , un conjunto de nodos S en dónde buscar, donde cada nodo posee un estimado de la distancia más corta a S_0 , denotado por $g(S_a)$ para un nodo a , y un nodo predecesor denotado como $predecesor(S_a)$, que es el nodo estimado por el que se llega a S_a con menor costo desde S_0 . También se tiene la función $c(S_a, S_b)$ que determina el peso del enlace entre el nodo S_a y el nodo S_b , y la función $VECINOS(S_a)$ que obtiene el conjunto de los nodos con enlace hacia S_a , también llamados vecinos.

Como ejemplo de aplicación del algoritmo, considere el problema mostrado en la Figura 1.9, donde se debe llegar desde el nodo origen hasta el nodo destino. El algoritmo comienza inicializando los valores g y con el nodo inicial en el conjunto de nodos abiertos llamado *ABIERTOS*. En cada iteración toma el nodo S_i en *ABIERTOS* con el menor el estimado de distancia $g(S_i)$ para después inspeccionar sus vecinos, actualizando los valores g y $predecesor$, en caso de haber encontrado una mejor estimación, e introduciéndolos en el conjunto de *ABIERTOS*. En ese caso, a este proceso se le llama expansión de nodos. El algoritmo termina una vez que no existen más nodos en *ABIERTOS*. El camino de menor

Algoritmo 1 Algoritmo de Dijkstra

```

1: entrada:  $S, S_0, S_f$ 
2:  $g(S_0) = 0$  ,  $g(S_f) = \infty$ 
3: for  $S_i \in S$  do
4:    $g(S_i) = \infty$ 
5: end for
6:  $ABIERTOS = S_0$ 
7: for all  $S_i \in ABIERTOS \mid g(S_i) \leq g(S_j) : \forall S_j \in ABIERTOS$  do
8:   for all  $S' \in VECINOS(S_i)$  do
9:     if  $g(S') > g(S_i) + c(S_i, S')$  then
10:       $g(S') = g(S_i) + c(S_i, S')$ 
11:       $predecesor(S') = S_i$ 
12:       $ABIERTOS = ABIERTOS \cup S'$ 
13:     end if
14:   end for
15:    $ABIERTOS = ABIERTOS - S_i$ 
16: end for

```

costo se obtiene siguiendo desde S_f el camino hecho por los predecesores $predecesor(S_a)$ hasta llegar a S_0 .

1.5.2 Algoritmo de búsqueda A*

El algoritmo de Dijkstra asegura encontrar el camino menos costoso para llegar del nodo S_0 al nodo S_f , pero para esto requiere expandir por lo menos una vez cada nodo, en redes de nodos grandes, esto puede representar una gran carga computacional. Este aspecto puede ser mejorado con ayuda de una función heurística denotada como $h(S_a)$ que ayudará a expandir primero los nodos más prometedores y desechando los que de antemano se saben no conducirán a un mejor resultado. Un algoritmo que utiliza esta técnica es el algoritmo de búsqueda A* [19], que se muestra en el Algoritmo 2.

La función heurística $h(S_a)$ es una función que estima el menor costo posible para llegar a S_f , partiendo de S_a , por tanto $g(S_a) + h(S_a)$ representa la mejor estimación optimista del costo para un camino que pasa por S_a , de S_0 a S_f . Si cumple con ser el menor costo posible se le llama admisible y asegura que siempre se obtendrá el camino óptimo si éste existe. La función $h(S_a)$ se obtiene al considerar el conocimiento general que se tiene sobre la red de nodos en la que se trabaja. Considerando el problema del viaje entre ciudades, podemos utilizar como función heurística la distancia euclidiana entre la ciudad (nodo) S_a a S_f en línea recta, ya que sabemos que esta es la distancia más corta posible entre esas ciudades.

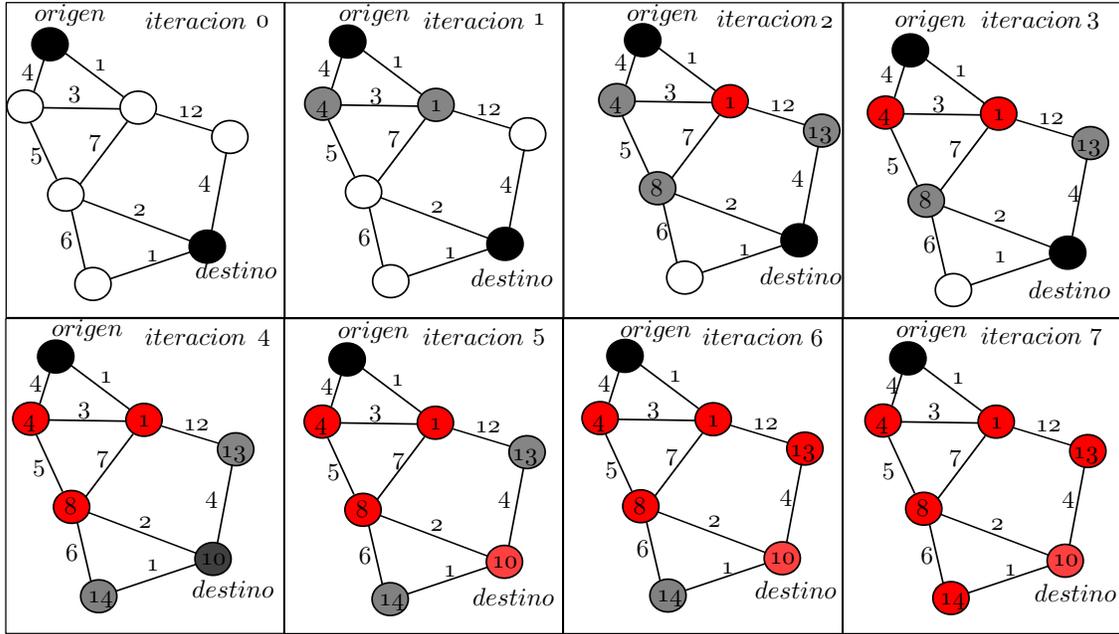


Figura 1.9: Iteraciones del algoritmo de Dijkstra, en rojo se muestran los nodos en *CERRADOS* y en gris los nodos en *ABIERTOS*. El número en el interior de los nodos representa su valor g asignado

Algoritmo 2 Algoritmo de búsqueda A*

```

1: entrada:  $S, S_0, S_f$ 
2:  $g(S_0) = 0, g(S_f) = \infty$ 
3: for  $S_i \in S$  do
4:    $g(S_i) = \infty$ 
5: end for
6:  $ABIERTOS = \emptyset$ 
7:  $ABIERTOS = ABIERTOS \cup S_0$ 
8: for all  $S_i \in ABIERTOS \mid g(S_i) + h(S_i) \leq g(S_j) + h(S_j) : \forall S_j \in ABIERTOS$  do
9:   if  $S_i = S_f$  then
10:    FIN
11:   end if
12:   for all  $S' \in VECINOS(S_i)$  do
13:     if  $g(S') > g(S_i) + c(S_i, S')$  then
14:        $g(S') = g(S_i) + c(S_i, S')$ 
15:        $predecesor(S') = S_i$ 
16:        $ABIERTOS = ABIERTOS \cup S'$ 
17:        $CERRADOS = CERRADOS - S_i$ 
18:     end if
19:   end for
20:    $ABIERTOS = ABIERTOS - S_i$ 
21:    $CERRADOS = CERRADOS \cup S_i$ 
22: end for

```

Capítulo 2

Metodología

En este capítulo se presenta el desarrollo del análisis cinemático y dinámico, basado en las ecuaciones presentadas en el capítulo previo, para obtener un modelo matemático completo que exprese el comportamiento dinámico del robot móvil utilizado en este trabajo de tesis. Después de obtener el modelo se presentan las ecuaciones para seguimiento de trayectoria y el controlador potencial artificial, concluyendo con el algoritmo utilizado para búsqueda de ruta.

2.1 Análisis cinemático



Figura 2.1: Robot Móvil Rodado de tipo diferencial

El vehículo que se pretende guiar es un vehículo terrestre o Robot Móvil Rodado (RMR) de tipo diferencial, esto es, un vehículo que posee dos ruedas para impulsarse con un motor independiente cada una, como el que se muestra en la Figura 3.1. Además cuenta con una rueda no actuada que sirve de apoyo.

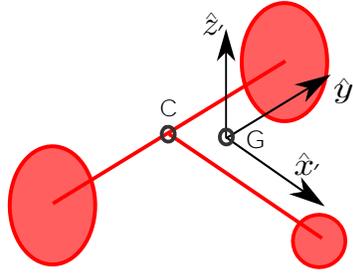


Figura 2.2: Marco de referencia para el vehículo terrestre

El vehículo está compuesto por cuatro cuerpos rígidos, siendo éstos las dos ruedas activas, la rueda de apoyo y el cuerpo principal del vehículo, cada uno de estos cuerpos posee una velocidad angular diferente. Definimos el marco de referencia \mathcal{B} fijo al cuerpo principal del vehículo y con origen en su centro de gravedad y otro marco de referencia \mathcal{I} inercial con su origen sobre la superficie de la tierra y con el eje \hat{z}' apuntando hacia arriba como se muestra en la Figura 1.1. La Figura 2.2 muestra la disposición del marco de referencia \mathcal{B} fijo en el vehículo. Definimos los puntos C y G como el centro del eje de las ruedas y el centro de gravedad, respectivamente, y además 4 vectores, como se muestra en la Figura 2.3.

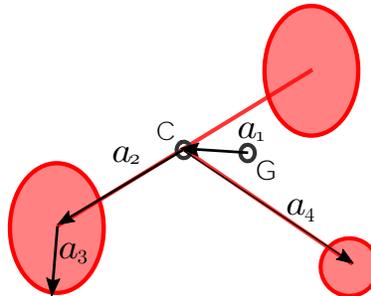


Figura 2.3: Vectores definidos sobre el vehículo terrestre

Estos vectores son:

- \mathbf{a}_1 . Es el vector que va desde el centro de gravedad hasta el centro del eje de las ruedas del vehículo. Este vector no tiene componente en dirección de \hat{y}' ya que se considera al vehículo simétrico en el plano $x'z'$.
- \mathbf{a}_2 . Es el vector que va desde el centro del eje de las ruedas hacia el centro de la llanta derecha
- \mathbf{a}_3 . Es el vector que va desde el centro de la rueda derecha hacia el punto de contacto de la rueda con el suelo.
- \mathbf{a}_4 . Es el vector que va desde el centro del eje de las ruedas hasta la rueda de apoyo.

El vehículo posee una serie de restricciones en su movimiento, una de ellas se obtiene de suponer que todas las ruedas se mantienen siempre en contacto con la superficie del suelo, al ser tres puntos de contacto con la superficie esto restringe a cualquier punto en el vehículo

incluyendo al punto C , a moverse en planos paralelos al de la superficie, este resultado se puede escribir como

$$(\mathbf{r}_c - \mathbf{r}_o) \cdot \hat{\mathbf{n}} = l \quad (2.1)$$

donde \mathbf{r}_o es algún punto sobre la superficie, $\hat{\mathbf{n}}$ es un vector normal a la superficie, \mathbf{r}_c es el vector posición del punto C y l es una constante igual a la distancia entre la superficie y el punto C . Entonces cualquier punto fijo en el cuerpo del vehículo mantiene una distancia fija a la superficie del suelo. Un resultado de esto también es que

$$\hat{\mathbf{z}}' - \hat{\mathbf{n}} = 0 \quad (2.2)$$

esto es, el eje $\hat{\mathbf{z}}'$ del marco \mathcal{B} se mantiene siempre paralelo a la normal de la superficie. Otra restricción viene de la *condición de no-deslizamiento* en las ruedas, esto es, las ruedas están sujetas a una fuerza de fricción con la superficie que no les permite deslizarse, solo rodar sobre ella. Utilizando las Ecuaciones (1.3) y (1.4), la condición dada por la Ecuación (2.2) y algo de álgebra vectorial, se llega una ecuación que describe esta restricción:

$$\mathbf{v}_c \cdot \hat{\mathbf{y}} = 0 \quad (2.3)$$

Esta ecuación expresa que el movimiento del punto C en el vehículo no puede ser lateral, ya que se encuentra en el eje de las ruedas, esto es intuitivo ya que un movimiento como éste llevaría al deslizamiento de las dos ruedas. Además debido a la Ecuación (2.2), se llega a la conclusión de que el movimiento del punto C sólo puede darse en dirección de $\hat{\mathbf{x}}'$, por lo tanto si el vehículo está en rotación, sería en un eje que pasa por C , ya que una rotación en otro eje resultaría en un componente de velocidad no nulo en $\hat{\mathbf{y}}'$. Cualquier punto sobre el eje de las ruedas tiene esta restricción, siendo el punto C el de mayor interés ya que se encuentra justo en el punto medio, proveyendo simetría y facilitando los cálculos. Debido a la propiedad presentada por la Ecuación (2.3), se opta por utilizar el punto C como el punto de referencia del vehículo. Cuando una restricción no puede expresarse de la forma:

$$f(\mathbf{q}) = 0$$

donde \mathbf{q} es una matriz columna con las N coordenadas, se dice que es una restricción no-holónoma o también llamada restricción diferencial no integrable. Este tipo de restricción presenta una dependencia en el camino, en la que se llega a una configuración tal que no existe una dependencia funcional entre las coordenadas, de forma que se pueda eliminar alguna de éstas hasta que se haya resuelto el problema para un caso específico [32]. Utilizando la Ecuación (2.3) en conjunto con las Ecuación (1.3) y mas álgebra vectorial obtenemos una relación entre las velocidades angulares de las ruedas y la velocidad \mathbf{v}_c del centro del eje de las ruedas y se llega a que las velocidades lineal y angular del vehículo son:

$$\begin{aligned} \mathbf{v}_c &= -\frac{1}{2}(\boldsymbol{\omega}_1 + \boldsymbol{\omega}_2) \times \mathbf{a}_3 \\ \boldsymbol{\Omega} \times \mathbf{a}_2 &= -\frac{1}{2}(\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2) \times \mathbf{a}_3 \end{aligned}$$

En el caso de movimiento en un plano, aplicando los productos cruz se obtiene

$$\mathbf{v}_c = v_c \hat{\mathbf{x}}' = \frac{|\mathbf{a}_3|}{2} (\omega_1 + \omega_2) \hat{\mathbf{x}}' \quad (2.4)$$

$$\boldsymbol{\Omega} = \Omega' \hat{\mathbf{z}}' = \frac{|\mathbf{a}_3|}{2|\mathbf{a}_2|} (\omega_2 - \omega_1) \hat{\mathbf{z}}' \quad (2.5)$$

El modelo cinemático del vehículo sólo posee 3 grados de libertad, con coordenadas generalizadas $\{x_c, y_c, \phi\}$, siendo dos coordenadas sobre el plano x_c, y_c y un ángulo para la dirección del vehículo sobre la superficie ϕ . Debido a que el sistema de referencia \mathcal{B} está fijo al vehículo e interesa estudiar su comportamiento desde el marco de referencia global \mathcal{I} de un observador, se buscan las ecuaciones que describen el comportamiento cinemático del vehículo terrestre en componentes del marco \mathcal{I} , que se pueden obtener al utilizar la Ecuación (A.2) con (2.4) y utilizando los ángulos de Euler para las rotaciones intrínsecas $x - y - z$ para obtener

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{z}_c \end{bmatrix} = v_c \begin{bmatrix} c_\theta c_\psi \\ c_\phi s_\psi + s_\phi s_\theta c_\psi \\ s_\phi s_\psi - c_\phi s_\theta c_\psi \end{bmatrix} \quad (2.6)$$

donde v_c es el componente en dirección $\hat{\mathbf{x}}'$ de la velocidad lineal del punto C . Debido a que $\boldsymbol{\Omega}'$ sólo tiene un componente Ω' en dirección de $\hat{\mathbf{z}}'$, utilizando la Ecuación (A.5) obtenemos

$$\begin{bmatrix} 0 \\ 0 \\ \Omega' \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\psi & 0 \\ -c_\theta s_\psi & c_\psi & 0 \\ s_\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

que al resolverlo obtenemos que $\dot{\psi} = 0$, $\dot{\theta} = 0$ y $\Omega' = \dot{\phi}$. Sabiendo esto y utilizando (A.4) con (2.5) se obtiene

$$\begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} s_\theta \\ -s_\phi c_\theta \\ c_\phi c_\theta \end{bmatrix} \dot{\phi} \quad (2.7)$$

Las Ecuaciones (2.6) y (2.7) definen entonces la forma que debe tener el movimiento del vehículo en todo momento, esto mientras sus restricciones cinemáticas de las Ecuaciones (2.3), (2.2) y (2.1) se mantengan válidas.

2.2 Análisis dinámico

Sobre el cuerpo del vehículo existe un sistema de fuerzas y torques como se muestra en la Figura 2.4. Las fuerzas \mathbf{F}_i son debidas a las reacciones con las ruedas de forma que éstas se mantengan unidas al cuerpo en su centro. También existe una fuerza de gravedad \mathbf{g} que no es necesariamente perpendicular al plano. Los torques $\boldsymbol{\tau}_i$ aplicados sobre el eje son los necesarios para mantener las ruedas en posición recta y perpendicular al plano, además incluye el torque producido por los motores.

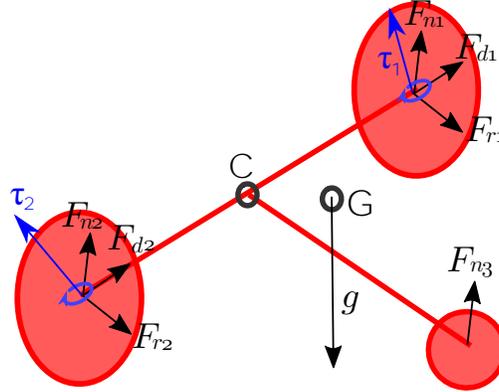


Figura 2.4: Diagrama de fuerzas sobre el vehículo terrestre.

En este análisis despreciamos las masas y momentos de inercia de las ruedas, de forma que $\frac{m_r}{m} \rightarrow 0$, donde m_r es la masa de una rueda y m es la masa del cuerpo del vehículo. Debido a esto, las fuerzas y torques aplicadas por la superficie sobre las ruedas se transmiten directamente al cuerpo del vehículo. Las fuerzas \mathbf{F}_i se pueden descomponer en tres componentes ortogonales que corresponden con [33] :

- \mathbf{F}_n son las fuerzas normales de contacto que mantiene a las ruedas sobre la superficie, en dirección $\hat{\mathbf{z}}'$. Corresponden a las restricciones de movimiento presentadas en las Ecuaciones (2.1) y (2.2). Éstas son mayores a cero en todo momento ya que de lo contrario el vehículo perdería contacto con el plano.
- \mathbf{F}_d son las fuerzas de no-deslizamiento lateral, las cuales fuerzan la restricción no-holónoma de la Ecuación (2.3) sobre las ruedas, en dirección $\hat{\mathbf{y}}'$. No deben superar la fuerza de fricción estática máxima con la superficie, dada como $\mu_e \|\mathbf{F}_{ni}\| > \|\mathbf{F}_{di} + \mathbf{F}_{ri}\|$, con $i = 1, 2$, donde μ_e es aproximadamente 1.0 para el caucho en contacto con el concreto seco y 0.3 en el caso de estar mojado.
- \mathbf{F}_r son las fuerzas de fricción sobre la superficie que fuerzan el rodamiento puro de las ruedas y corresponden a las fuerzas que recibe el vehículo para producir el movimiento en dirección $\hat{\mathbf{x}}'$.

Los torques τ_i se pueden descomponer en:

- τ_m debido al torque del motor aplicado sobre el eje de las ruedas, está en dirección $\hat{\mathbf{y}}'$.
- τ_n debido al torque que aplican los soportes del eje montados en el cuerpo del vehículo, para mantener las ruedas en un plano perpendicular a la superficie, en alguna dirección perpendicular a $\hat{\mathbf{y}}'$. Es también un resultado de las restricciones de las Ecuaciones (2.1) y (2.2).

Debido a que el cuerpo principal del vehículo se comporta como un cuerpo rígido, podemos obtener sus ecuaciones de movimiento al utilizar la Ecuaciones (1.5) y (1.6) para el formalismo

de Newton-Euler. Entonces la sumatoria de fuerzas sobre el centro de masa G y los torques con respecto a G para el cuerpo del vehículo quedan expresados:

$$\begin{aligned} m\dot{\mathbf{v}}_G &= \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3 + \mathbf{g} & (2.8) \\ \frac{d(\mathbf{I} \cdot \boldsymbol{\Omega})}{dt} &= (\mathbf{a}_1 + \mathbf{a}_2) \times \mathbf{F}_1 + (\mathbf{a}_1 - \mathbf{a}_2) \times \mathbf{F}_2 + (\mathbf{a}_1 + \mathbf{a}_4) \times \mathbf{F}_3 - \boldsymbol{\tau}_1 - \boldsymbol{\tau}_2 & (2.9) \end{aligned}$$

Los torques $\boldsymbol{\tau}_{m1}$ y $\boldsymbol{\tau}_{m2}$, debidos a los motores, producen las fuerzas \mathbf{F}_r sobre el cuerpo principal del vehículo en dirección de $\hat{\mathbf{x}}'$ dadas por

$$\mathbf{F}_{r1} = \frac{\boldsymbol{\tau}_{m1}}{\|\mathbf{a}_3\|}, \quad \mathbf{F}_{r2} = \frac{\boldsymbol{\tau}_{m2}}{\|\mathbf{a}_3\|}$$

estas ecuaciones se obtiene al aplicar la segunda ley de Newton sobre cada rueda y haciendo la aproximación de masa despreciable $\frac{m_r}{m} \rightarrow 0$. Debido a que la formulación cinemática obtenida en la Ecuación (2.6) está dada para la velocidad del punto C , expresamos $\dot{\mathbf{v}}_G$ en base a $\dot{\mathbf{v}}_C$, sabiendo que $\mathbf{r}_G = \mathbf{r}_c - \mathbf{a}_1$ (ver Figura 2.3) y derivandolos dos veces se obtiene $\ddot{\mathbf{r}}_G = \dot{\mathbf{v}}_C - \ddot{\mathbf{a}}_1$. Debido a que el cambio de \mathbf{a}_1 es debido a la rotación del marco de referencia en que se encuentra, podemos obtener $\ddot{\mathbf{a}}_1$ empleando la Ecuación (1.3) y considerando que el vector \mathbf{a}_1 en el marco \mathcal{B} es constante, obteniendo

$$\ddot{\mathbf{a}}_1 = \frac{d}{dt}(\boldsymbol{\Omega} \times \mathbf{a}_1) = \dot{\boldsymbol{\Omega}} \times \mathbf{a}_1 + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{a}_1)$$

Con esto, la Ecuación (2.8) toma la forma

$$m\dot{\mathbf{v}}_C - \dot{\boldsymbol{\Omega}} \times \mathbf{a}_1 - \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{a}_1) = \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3 + \mathbf{g} \quad (2.10)$$

Debido a que la velocidad del vehículo se encuentra restringida a permanecer en dirección de $\hat{\mathbf{x}}'$ según la Ecuación (2.4), basta con obtener la proyección escalar de la Ecuación (2.10) en dirección de $\hat{\mathbf{x}}'$ utilizando el producto punto, para entonces obtener una ecuación para la variable \dot{v}_c que es la velocidad lineal del vehículo, reordenando esto resulta en la ecuación

$$m\dot{v}_c = \frac{1}{\|\mathbf{a}_3\|}(\tau'_{m2} + \tau'_{m1}) - mg(s_\phi s_\psi - c_\phi s_\theta c_\psi) - ma'_{1x}(\Omega')^2 \quad (2.11)$$

donde τ'_{m1} y τ' son la magnitud de los torques de los motores y a'_{1x} denota el componente en x del vector \mathbf{a}_1 dentro del marco \mathcal{B} . El término que contiene mg corresponde a la fuerza de gravedad y $ma'_{1x}(\Omega')^2$ es la fuerza centrífuga debida a que el centro de masa se encuentra alejado una distancia $\|a'_{1x}\|$ del eje de rotación en el punto C , en el caso de que el centro de masa se encuentre en el punto C , este termino desaparecería. De forma análoga, debido a la restricción presentada por la Ecuación (2.5) en el vector $\boldsymbol{\Omega}$, se obtiene una ecuación para la evolución de $\Omega' = \|\boldsymbol{\Omega}\|$ al proyectar (2.9) en dirección de $\hat{\mathbf{z}}'$, obteniendo

$$(I'_{33} + m(a'_{1x})^2)\dot{\Omega}' = ma'_{1x}v_c\Omega' - mga'_{1x}(s_\phi c_\psi + c_\phi s_\theta s_\psi) + \frac{\|\mathbf{a}_2\|}{\|\mathbf{a}_3\|}(\tau'_{m2} - \tau'_{m1}) \quad (2.12)$$

Las Ecuaciones (2.11), (2.12), (2.6) y la expresión $\dot{\phi} = \Omega'$ forman un sistema de ecuaciones diferenciales y al ser todas de primer orden, se pueden expresar directamente en la formulación de espacio de estado del capítulo anterior. No se utiliza la Ecuación (2.7) en el sistema de ecuaciones diferenciales ya que $\Omega' = \|\boldsymbol{\Omega}'\| = \|\boldsymbol{\Omega}\|$ expresa que la magnitud de la velocidad angular $\boldsymbol{\Omega}$ es Ω' y no es de interés la dirección precisa de la velocidad angular, ya que se encuentra en una dirección fija a la normal de la superficie además de que no aparecerá directamente en la salida del sistema de espacio de estado.

2.3 Modelos matemáticos

Una vez obtenidas las ecuaciones de movimiento se procede a formar el modelo matemático en ecuaciones en espacio de estado. Para facilitar los cálculos, se elige un sistema de coordenadas fijo y alineado al plano de movimiento del vehículo terrestre. Se considera un marco de referencia $\hat{\mathbf{x}}''', \hat{\mathbf{y}}''', \hat{\mathbf{z}}'''$ generado según los ángulos de Euler utilizando una matriz de rotación $\mathbf{R}(\phi, \theta, 0)$. Las coordenadas de un vector \mathbf{q} en la base $\hat{\mathbf{x}}''', \hat{\mathbf{y}}''', \hat{\mathbf{z}}'''$ se relacionan con las demás bases utilizando la transformación de coordenadas de la Ecuación (A.2), obteniendo las relaciones:

$$\begin{aligned} [\mathbf{q}]' &= \mathbf{R}_z^{-1}(\psi) [\mathbf{q}]''' \\ [\mathbf{q}]''' &= \mathbf{R}^{-1}(\phi, \theta, 0) [\mathbf{q}] \end{aligned}$$

Con esta transformación, las Ecuaciones (2.11), (2.12), (2.6) resultan en:

$$\begin{aligned} m\dot{v}_c &= g_x''' \cos \psi + g_y''' \sin \psi - a'_{1x}(\Omega')^2 + \frac{1}{\|\mathbf{a}_3\|}(\tau'_{m2} + \tau'_{m1}) \\ (I_{33} + m(a'_{1x})^2)\dot{\Omega}' &= a'_{1x}(g_x''' \sin \psi - g_y''' \cos \psi) + ma'_{1x}v_c\Omega' + \frac{\|\mathbf{a}_2\|}{\|\mathbf{a}_3\|}(\tau'_{m2} - \tau'_{m1}) \\ \begin{bmatrix} \dot{x}_c''' \\ \dot{y}_c''' \\ \dot{z}_c''' \end{bmatrix} &= v_c \begin{bmatrix} \cos \psi \\ \sin \psi \\ 0 \end{bmatrix} \end{aligned}$$

Entonces, esta transformación tiene como efecto presentar un nuevo marco de referencia fijo a la superficie, siguiendo cualquier inclinación que pueda tener y con una fuerza de gravedad que no se encuentra sólo en el eje $\hat{\mathbf{z}}'''$, además se tiene que $\hat{\mathbf{x}}' = \cos \phi \hat{\mathbf{x}}''' + \sin \phi \hat{\mathbf{y}}'''$. Las entradas del modelo dinámico son los torques producidos por los motores, que se supone son completamente controlables. La salida del sistema son las coordenadas x_c''', y_c''' sobre la superficie, siendo z_c''' siempre cero. Con las ecuaciones anteriores se puede formar un sistema en forma de espacio de estado como

$$\begin{aligned} \begin{bmatrix} \dot{v}_c \\ \dot{\Omega}' \\ \dot{x}_c''' \\ \dot{y}_c''' \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} \frac{1}{m}(g_x''' \cos \psi + g_y''' \sin \psi - a'_{1x}(\Omega')^2) \\ \frac{1}{I_T}(a'_{1x}(g_x''' \sin \psi - g_y''' \cos \psi) + ma'_{1x}v_c\Omega') \\ v_c \cos \psi \\ v_c \sin \psi \\ \Omega' \end{bmatrix} + \begin{bmatrix} -\frac{1}{ma''_{3z}} & -\frac{1}{ma''_{3z}} \\ \frac{a''_{2y}}{I_T a''_{3z}} & -\frac{a''_{2y}}{I_T a''_{3z}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tau'_{m1} \\ \tau'_{m2} \end{bmatrix} \quad (2.13) \\ \mathbf{y} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_c \\ \Omega' \\ x_c''' \\ y_c''' \\ \psi \end{bmatrix} \quad (2.14) \end{aligned}$$

donde $g_x''' = g \cos \phi \sin \theta$ y $g_y''' = -g \sin \phi$ son constantes que representan los componentes del vector fuerza de gravedad en la base anteriormente mencionada y donde definimos $I_T =$

$I_{33} + m(a'_{1x})^2$ que es el momento inercial con respecto al punto C . Este modelo es obviamente no lineal, debido a los términos trigonométricos en ϕ y al factor $(\Omega')^2$. Este sistema de ecuaciones tiene la forma de las Ecuaciones (1.31) y (1.32) y puede ser linealizado. El vector de estados \mathbf{x} y el vector de entrada \mathbf{u} son

$$\mathbf{x} = \begin{bmatrix} v_c \\ \Omega' \\ x_c''' \\ y_c''' \\ \psi \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \tau'_{m1} \\ \tau'_{m2} \end{bmatrix}$$

Para el caso de un plano completamente horizontal, esto es, con la fuerza de gravedad alineada a la normal $\hat{\mathbf{z}}$ del plano, el sistema de Ecuaciones (2.13) y (2.14) se reduce a

$$\begin{bmatrix} \dot{v}_c \\ \dot{\Omega}' \\ \dot{x}_c''' \\ \dot{y}_c''' \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{1}{m} a'_{1x} (\Omega')^2 \\ \frac{m}{I_T} a'_{1x} v_c \Omega' \\ v_c \cos \psi \\ v_c \sin \psi \\ \Omega' \end{bmatrix} + \begin{bmatrix} -\frac{1}{ma''_{3z}} & -\frac{1}{ma''_{3z}} \\ \frac{a''_{2y}}{I_T a''_{3z}} & -\frac{a''_{2y}}{I_T a''_{3z}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tau'_{m1} \\ \tau'_{m2} \end{bmatrix} \quad (2.15)$$

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_c \\ \Omega' \\ x_c''' \\ y_c''' \\ \psi \end{bmatrix} \quad (2.16)$$

El tratamiento cinemático del vehículo autónomo, en el que no se consideran fuerzas y torques, es muy recurrido en la literatura [34], en este enfoque solamente cinemático se presupone que las ruedas pueden ser llevadas a cualquier velocidad requerida, siendo las velocidades las entradas del sistema, esto resulta en una buena aproximación que simplifica enormemente la solución, sin embargo, cuando vehículos pesados sujetos a fuerzas inerciales considerables son tratados, el enfoque puramente cinemático puede llevar a comportamientos claramente erróneos e inestables. El modelo cinemático puede obtenerse de las Ecuación (2.6) y la expresión $\dot{\phi} = \Omega'$ o se puede extraer directamente de los últimos 3 renglones de la Ecuación (2.15), resultando en

$$\begin{bmatrix} \dot{x}_c''' \\ \dot{y}_c''' \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & 0 \\ \sin \psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \Omega' \end{bmatrix} \quad (2.17)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c''' \\ y_c''' \\ \psi \end{bmatrix} \quad (2.18)$$

Aquí el vector de estados \mathbf{x} y el vector de entrada \mathbf{u} son dados por

$$\mathbf{x} = \begin{bmatrix} x_c''' \\ y_c''' \\ \psi \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} v_c \\ \Omega' \end{bmatrix}$$

Este sistema entonces no considera los aspectos dinámicos del problema y se basa en la premisa de que es posible realizar un controlador para la velocidad de los motores lo suficientemente rápido como para despreciar los efectos transitorios en su velocidad.

2.4 Seguimiento de trayectoria

En este trabajo se estudiará principalmente el sistema sobre una superficie completamente horizontal obtenido en las Ecuaciones (2.15) y (2.16). Si se busca obtener un controlador de lazo abierto en el modelo cinemático para obtener una salida dada por la curva $\mathbf{y}_r(t) = [x_r'''(t) \ y_r'''(t)]^T$, se puede obtener de las siguientes ecuaciones [35]

$$v_r(t) = \sqrt{(\dot{x}_r''')^2 + (\dot{y}_r''')^2} \quad (2.19)$$

$$\Omega_r'(t) = \frac{\ddot{y}_r''' \dot{x}_r''' - \dot{y}_r''' \ddot{x}_r'''}{(\dot{x}_r''')^2 + (\dot{y}_r''')^2} \quad (2.20)$$

Se debe tener en cuenta que el vehículo debe tener un estado inicial tal que

$$x'''(0) = x_r'''(0), \quad y'''(0) = y_r'''(0), \quad \phi(0) = \tan\left(\frac{y_r'''(0)}{x_r'''(0)}\right), \quad v_c(0) = v_r(0), \quad \Omega'(0) = \Omega_r'(0)$$

Estas ecuaciones se obtienen fácilmente del vector tangencial y de la curvatura de la curva $[x_r'''(t) \ y_r'''(t)]^T$ parametrizada en t [35]. Esto se debe a que el vector tangencial de una curva continua que siga el vehículo es siempre paralelo a $\hat{\mathbf{x}}'$ debido a las restricciones de no-deslizamiento lateral. Siguiendo este razonamiento, para el caso dinámico se puede hacer uso de los primeros 2 renglones de la Ecuación (2.15) donde sustituimos las Ecuaciones (2.19) y (2.20), reordenando se obtiene:

$$\begin{aligned} \frac{1}{\|\mathbf{a}_3\|} (\tau'_{m2} + \tau'_{m1}) &= m \frac{d}{dt} (\sqrt{(\dot{x}_r''')^2 + (\dot{y}_r''')^2}) + a'_{1x} \left(\frac{\ddot{y}_r''' \dot{x}_r''' - \dot{y}_r''' \ddot{x}_r'''}{(\dot{x}_r''')^2 + (\dot{y}_r''')^2} \right)^2 \\ \frac{\|\mathbf{a}_2\|}{\|\mathbf{a}_3\|} (\tau'_{m2} - \tau'_{m1}) &= I_T \frac{d}{dt} \left(\frac{\ddot{y}_r''' \dot{x}_r''' - \dot{y}_r''' \ddot{x}_r'''}{(\dot{x}_r''')^2 + (\dot{y}_r''')^2} \right) - m a'_{1x} \left(\frac{\ddot{y}_r''' \dot{x}_r''' - \dot{y}_r''' \ddot{x}_r'''}{\sqrt{(\dot{x}_r''')^2 + (\dot{y}_r''')^2}} \right) \end{aligned} \quad (2.21)$$

que puede resolverse para τ'_{m1} y τ'_{m2} , que son las entradas necesarias para el control en lazo abierto. Si se aplican estas ecuaciones para una situación ideal, se resuelve por completo el problema, sin embargo este enfoque en lazo abierto no es efectivo en casos prácticos. La incertidumbre en la medición de la masa y el momento inercial, perturbaciones externas, el pequeño deslizamiento presente las ruedas y el ruido presente en los sensores, debido a la integración en las variables, aumentará con el tiempo el error total en la posición. Para resolver este problema se utilizó un control en lazo cerrado que reduce el error producido por estos factores.

2.5 Controlador de potencial artificial

Para la implementación del seguimiento de trayectoria en lazo cerrado, se utilizó una propuesta con un controlador de potencial artificial V de la forma:

$$V = \frac{1}{3}k_{1p}r^3 \quad (2.22)$$

donde k_{1p} es una constante y r es la distancia hacia el destino. La fuerza debida a este potencial se calcula como el gradiente negativo del campo potencial V y está aplicada a una distancia delante del punto C sobre el robot. Si el robot se encontrara sujeto a una fuerza debida a un potencial como el de la Ecuación (2.22), entonces la fuerza y el torque artificial que afectan al robot móvil están dados por:

$$\mathbf{F} = k_{1p}r^2\hat{\mathbf{r}} - k_{1d}\mathbf{v}_c \quad (2.23)$$

$$\boldsymbol{\tau} = k_{2p}\mathbf{F} \times \hat{\mathbf{x}}' - k_{2d}\Omega'\hat{\mathbf{z}}' \quad (2.24)$$

donde $\hat{\mathbf{r}}$ es un vector unitario que apunta desde el robot móvil hacia el destino, \mathbf{v} es el vector velocidad del centro del punto C , y donde k_{1p} , k_{2p} , k_{1d} y k_{2d} son constantes ajustables para lograr una respuesta del sistema adecuada. Aquí además se incluyeron los términos $-k_{1d}\mathbf{v}$ y $-k_{2d}\Omega'\hat{\mathbf{z}}'$, correspondientes a un potencial dependiente de la velocidad para ambas ecuaciones, de la forma (1.36) para agregar *amortiguamiento* en el movimiento del robot, esto es, asegurar que los polos se encuentren del lado izquierdo. Experimentalmente se ajustaron las constantes como se indica en la Tabla 2.1.

Tabla 2.1: Constantes ajustadas para el controlador de campos potenciales de las Ecuaciones (2.23) y (2.24)

Constante	Valor
k_{1p}	20.0
k_{1d}	16.0
k_{2p}	2.25
k_{2d}	4.5

Debido a que este control se aplica directamente como una fuerza y un torque, se puede hacer referencia a las Ecuaciones (2.8) y (2.9) que están en términos directos a las fuerzas y torques. Desarrollando se llega a que \dot{v}_c se ve afectado por la proyección $\mathbf{F} \bullet \hat{\mathbf{x}}'$ y $\dot{\Omega}'$ por $\boldsymbol{\tau} \bullet \hat{\mathbf{z}}'$ de las Ecuaciones (2.23) y (2.24) siendo éstas:

$$\mathbf{F} \bullet \hat{\mathbf{x}}' = F = k_{1p}r \left(\cos \phi(x_r''' - x_c''') + \sin \phi(y_r''' - y_c''') \right) - k_{1d}v_c \quad (2.25)$$

$$\boldsymbol{\tau} \bullet \hat{\mathbf{z}}' = \tau = k_{1p}k_{2p}r \left(\sin \phi(x_r''' - x_c''') - \cos \phi(y_r''' - y_c''') \right) - k_{2d}\Omega' \quad (2.26)$$

$$r = \sqrt{(x_r''' - x_c''')^2 + (y_r''' - y_c''')^2} \quad (2.27)$$

Con estas entradas, el modelo del sistema es

$$\begin{aligned} \begin{bmatrix} \dot{v}_c \\ \dot{\Omega}' \\ \ddot{x}_c''' \\ \ddot{y}_c''' \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} -\frac{1}{m}a'_{1x}(\Omega')^2 \\ \frac{m}{I_T}a'_{1x}v_c\Omega' \\ v_c \cos \psi \\ v_c \sin \psi \\ \Omega' \end{bmatrix} + \begin{bmatrix} F \\ \tau \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{m}a'_{1x}(\Omega')^2 + k_{1p}r \left(\cos \phi(x_r''' - x_c''') + \sin \phi(y_r''' - y_c''') \right) - k_{1d}v_c \\ \frac{m}{I_T}a'_{1x}v_c\Omega' + k_{1p}k_{2p}r \left(\sin \phi(x_r''' - x_c''') - \cos \phi(y_r''' - y_c''') \right) - k_{2d}\Omega' \\ v_c \cos \psi \\ v_c \sin \psi \\ \Omega' \end{bmatrix} \end{aligned} \quad (2.28)$$

En este sistema se puede apreciar que la diferencia entre las variables de salida x_c''', y_c''' y las de referencia x_r''', y_r''' aparecen en la entrada y corresponden a su realimentación para la entrada.

Los torques necesarios en los motores para producir la nueva entrada se pueden obtener comparando la Ecuación (2.28) con la Ecuación (2.15) llegando a que

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} -\frac{1}{ma''_{3z}} & -\frac{1}{ma''_{3z}} \\ \frac{a''_{2y}}{I_T a''_{3z}} & -\frac{a''_{2y}}{I_T a''_{3z}} \end{bmatrix} \begin{bmatrix} \tau'_{m1} \\ \tau'_{m2} \end{bmatrix} \quad (2.29)$$

2.6 Controlador de DeLuca

Para tener una referencia y poder evaluar la efectividad de los sistemas de control propuestos se presenta un controlador encontrado en la literatura, se eligió el de DeLuca [34], donde se propone un controlador cinemático para el sistema de la Ecuación (2.7), basado en linealización exacta por realimentación de estado. El controlador de DeLuca se obtiene al alimentar con una nueva entrada de control al sistema de forma similar a la Ecuación (1.35) según:

$$\dot{\eta} = u_1 \cos \psi + u_2 \sin \psi \quad (2.30)$$

$$v_c = \eta \quad (2.31)$$

$$\Omega' = \frac{u_2 \cos \psi - u_1 \sin \psi}{\eta} \quad (2.32)$$

Se introduce la variable η como una variable dinámica interna del controlador y u_1 y u_2 son las nuevas entradas. Un control como éste transforma al sistema de la Ecuación (2.7) en un sistema de dos ecuaciones de segundo orden:

$$\begin{bmatrix} \ddot{x}_c''' \\ \ddot{y}_c''' \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

que escrito en la forma de la Ecuación (1.16) queda

$$\begin{bmatrix} \dot{x}_c''' \\ \dot{y}_c''' \\ \ddot{x}_c''' \\ \ddot{y}_c''' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_c''' \\ y_c''' \\ \dot{x}_c''' \\ \dot{y}_c''' \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

donde definimos \mathbf{x} como el vector de estado, \mathbf{u} como el vector de entradas y \mathbf{A} como la matriz de estado. Este sistema lineal de ecuaciones puede controlarse utilizando la teoría de control lineal con realimentación. Se define el vector de entradas utilizando la Ecuación (1.30) como

$$\mathbf{u} = \mathbf{K}(\mathbf{x} - \mathbf{x}_r) + \mathbf{u}_r = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} k_1 & 0 & k_2 & 0 \\ 0 & k_3 & 0 & k_4 \end{bmatrix} \begin{bmatrix} x_c''' - x_r''' \\ y_c''' - y_r''' \\ \dot{x}_c''' - \dot{x}_r''' \\ \dot{y}_c''' - \dot{y}_r''' \end{bmatrix} + \begin{bmatrix} \ddot{x}_r''' \\ \ddot{y}_r''' \end{bmatrix}$$

donde \mathbf{u}_r es la entrada de prealimentación y \mathbf{x}_r el vector de estado correspondiente al estado de referencia. Escogiendo los valores k_1 , k_2 , k_3 y k_4 correctamente se lleva a un sistema de lazo cerrado estable y que cumple con los objetivos, se eligieron experimentalmente las constantes mostradas en la Tabla 2.2.

Tabla 2.2: Constantes ajustadas para el controlador de DeLuca

Constante	Valor
k_1	3.0
k_2	1.5
k_3	3.0
k_4	1.5

2.7 Búsqueda de ruta

El problema de llevar un robot móvil de un lugar a otro de forma autónoma se suele subdividir en tres etapas: la búsqueda de ruta, la generación de trayectoria y el seguimiento de trayectoria. Cada uno de estos subproblemas es resuelto secuencialmente en una jerarquía donde la entrada de un sistema se obtiene de la salida de otro [16]. La Figura 2.5 muestra un diagrama de bloques de la jerarquía de sistemas.

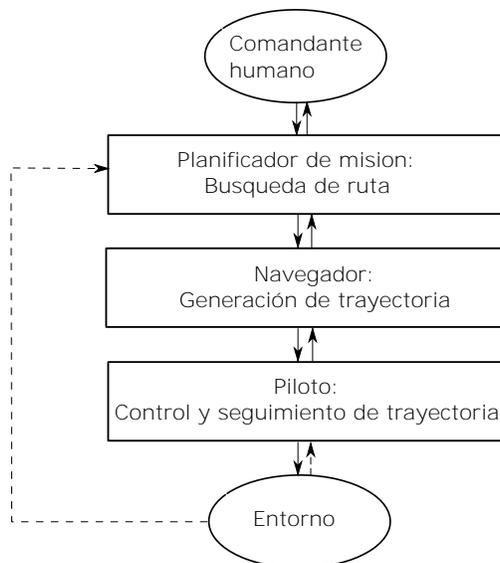


Figura 2.5: Jerarquía de sistemas

En el nivel más alto de la jerarquía se encuentra el *planificador de misión*, que corresponde al problema de búsqueda de ruta y que recibe los comandos de alto nivel provistos por un comandante humano. El planificador de misión se encarga de encontrar una estrategia para resolver el problema definido por el comandante y seguir las condiciones que estableció. El planificador de misión generalmente implica la generación de una representación interna del mundo exterior al robot, en la que se trabaja de forma más abstracta y de alto nivel. El planificador suele tener tiempos de respuesta largos en comparación a los otros dos niveles, debido a que éste resuelve un problema computacionalmente costoso. Así mismo, una vez planificada la ruta, no es necesario más trabajo por parte del planificador de ruta, a menos hasta que algún evento imprevisto deje inviable la ruta anteriormente calculada. En este trabajo se seleccionó al algoritmo de Dijkstra, presentado en el Algoritmo 1, como el núcleo del planificador de misión. Para el navegador se utilizó un enfoque sencillo basado en simplemente crear splines cúbicos que pasen por el camino de nodos generado por el planificador de misión, en el Anexo C se presenta brevemente el método de los splines cúbicos. En el caso del piloto se utilizó el control basado en el potencial artificial de la Ecuación (2.22), con el sistema dinámico resultante de la Ecuación (2.28). El piloto seguirá la trayectoria $[x_r'''(t) \ y_r'''(t)]^T$ generada por el navegador.

Entre las formas de representar el problema que no implica la discretización del área de trabajo, está una forma alternativa utilizar los campos potenciales [36], siendo esta la primera forma en la que se concibió y es ingresar directamente el destino como único punto de referencia, sin utilizar la jerarquía anterior, pero incluyendo potenciales repulsivos en los obstáculos para poder evadirlos.

Una vez determinados los obstáculos y el vehículo que se desea trasladar, se requiere el espacio de configuración permisible para el vehículo con el que se formara el grafo necesario para el algoritmo de Dijkstra en este entorno. Si consideramos los obstáculos como conjuntos abiertos de puntos con forma poligonal, el camino más corto que debe seguir una partícula, evadiendo los obstáculos, para llegar a un destino, es a través de los vértices de los obstáculos,

excepto en el caso en que exista una línea entre la partícula y el destino [37]. Con esto podemos formar una red de nodos con los vértices de los obstáculos, incluyendo el punto inicial y final del trayecto. Debido a que esto aplica para cualquier partícula moviéndose entre cualquier par de vértices, podemos decir que dos vértices están conectados directamente (sin pasar por otros vértices) si existe una línea recta libre o línea de visión entre ellos. Entonces existe un enlace entre cada par de nodos que tengan línea de visión uno con el otro y por tanto, un camino libre de obstáculos entre ellos en línea recta. A este tipo de gráfica se le llama *gráfica de visibilidad* (ver Figura 2.6).

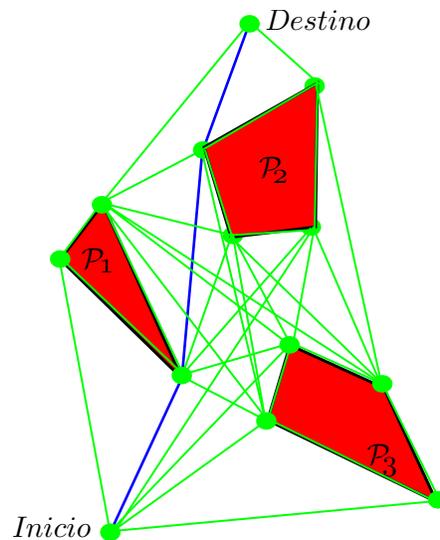


Figura 2.6: Gráfica de visibilidad y camino más corto entre dos puntos

El vehículo representado como una partícula puntual puede entonces recorrer el grafo pasando por los nodos, para recorrer el camino más corto a su destino que está incluido entre los nodos, agregando a cada enlace su longitud como peso/costo y buscando el camino con la menor suma de pesos posible. Tomando en cuenta que el vehículo no es una partícula puntual sino un objeto que ocupa un área y un lugar en el espacio de trabajo, se puede aproximar este espacio con un polígono, además este polígono puede incluir un margen de seguridad, de tal forma que sea ligeramente más grande que el vehículo para evitar el riesgo de colisiones. Por simplicidad, podemos utilizar un cuadrado que encierre completamente al vehículo en todos sus ángulos de rotación.

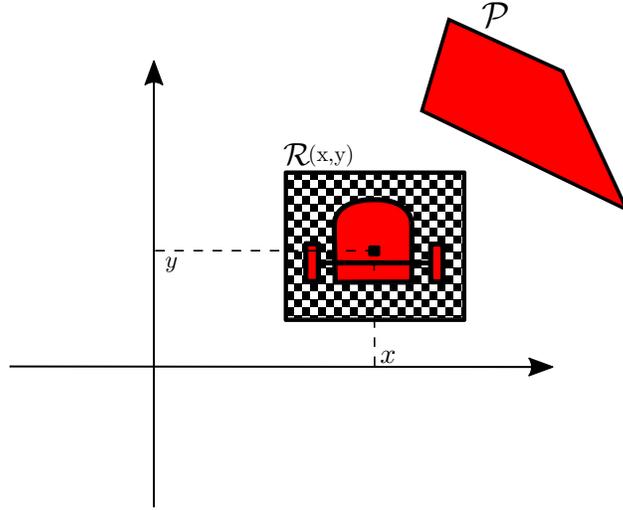


Figura 2.7: Vehículo con su conjunto de puntos $\mathcal{R}(x, y)$ y obstáculo \mathcal{P} .

En la Figura 2.7 se muestra el conjunto $\mathcal{R}(x, y)$ en forma de cuadro punteado, el cual enmarca al vehículo como un área segura que debe evitar traslaparse con los obstáculos, y donde $(x, y) = (x_c, x_c)$ son las coordenadas del punto C fijo al vehículo, en este caso el origen de su sistema de coordenadas. El conjunto $\mathcal{R}(0, 0)$ es el mismo polígono pero centrado en el origen del espacio de trabajo. Además \mathcal{P} denota el conjunto abierto de puntos que representan un obstáculo conformado por un polígono convexo. Para evitar colisiones se debe asegurar que no existan intersecciones entre $\mathcal{R}(x, y)$ y \mathcal{P} , para todo (x, y) en una trayectoria del vehículo.

El problema se puede volver a reducir al de una partícula puntual si aumentamos el tamaño de cada obstáculo \mathcal{P} a un polígono \mathcal{CP} . El problema de aumentar figuras poligonales está relacionado con el llamado *desplazamiento del polígono*, éste se utiliza en el área de maquinado CNC para encontrar trayectorias para las herramientas [38]. Entonces \mathcal{CP} es un polígono tal que $\mathcal{R}(x, y)$ y \mathcal{P} nunca se intersecan mientras (x, y) esté fuera de \mathcal{P} . Las coordenadas (x, y) que cumplan con ello formarán el espacio de configuración permisible para el vehículo y formarán una nueva red de nodos por los que el vehículo puede pasar de forma segura, utilizando una gráfica de visibilidad. El conjunto \mathcal{CP} se puede obtener de la suma de Minkowski de \mathcal{P} y $-\mathcal{R}(0, 0)$, donde definimos $-\mathcal{R} = \{-q = (-q_x, -q_y) : q \in \mathcal{R}\}$ [37]. La ecuación que define la suma de dos conjuntos \mathcal{A} y \mathcal{B} se expresa mediante

$$\mathcal{C} = \mathcal{A} \oplus \mathcal{B} = \{c = a + b = (a_x + b_x, a_y + b_y) : a \in \mathcal{A}, b \in \mathcal{B}\} \quad (2.33)$$

La suma de Minkowski se define como un conjunto formado por la unión de todas las sumas de elementos de \mathcal{A} con los elementos de \mathcal{B} como en (2.33). Para el caso de conjuntos de puntos (figuras) en dos dimensiones, esto se puede visualizar como si tomáramos \mathcal{A} llena de pintura y recorriéramos todo el área de \mathcal{B} con el punto de referencia de \mathcal{A} pero sin rotarlo. La suma de Minkowski de dos polígonos convexos es un polígono convexo. Un algoritmo para la la suma de Minkowski para polígonos convexos se muestra en el Algoritmo 3, en él se construye la suma de Minkowski de un polígono \mathcal{A} en tiempo $\mathcal{O}(n)$ lineal y un polígono \mathcal{B}

con vértices a_i y b_i , respectivamente, la suma es un polígono de vértices c_i . Los vértices c_i se obtienen haciendo sumas de coordenadas entre los vértices más extremos en cada dirección. Esta suma de Minkowski también se puede obtener haciendo la suma de Minkowski del conjunto de los vértices de ambos polígonos y construyendo el *envolvente convexo* (*convex hull* en inglés) del conjunto, aunque esto requiere más esfuerzo computacional. La función $angulo(a_i, a_{i+1})$ obtiene el ángulo antihorario entre el segmento formado por los vértices a_i, a_{i+1} y el eje de las x .

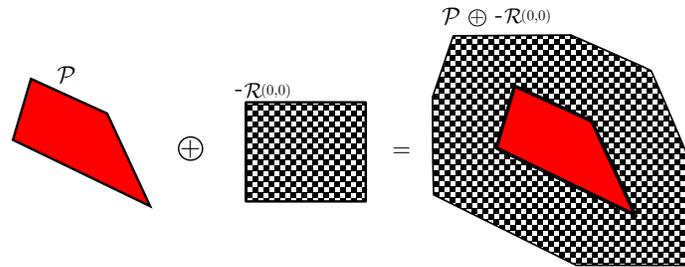


Figura 2.8: Visualización de la suma de Minkowski

Algoritmo 3 Suma de Minkowski

- 1: entrada: conjunto de n vértices a_i del polígono \mathcal{A} y b_i de \mathcal{B} en orden antihorario y con a_1 y b_1 siendo los vértices con menor coordenada y en su polígono.
 - 2: salida: \mathcal{C} polígono con vértices c_i que representa $\mathcal{A} \oplus \mathcal{B}$
 - 3: $i = 1, j = 1, k = 1$
 - 4: $a_{n+1} = a_1, b_{n+1} = b_1, a_{n+2} = a_2, b_{n+2} = b_2$
 - 5: **while** $i \neq n + 1$ **and** $j \neq n + 1$ **do**
 - 6: $c_k = a_i + b_j$
 - 7: $k = k + 1$
 - 8: **if** $angulo(a_i, a_{i+1}) < angulo(b_j, b_{j+1})$ **then**
 - 9: $i = i + 1$
 - 10: **else**
 - 11: **if** $angulo(a_i, a_{i+1}) > angulo(b_j, b_{j+1})$ **then**
 - 12: $j = j + 1$
 - 13: **else**
 - 14: $i = i + 1$
 - 15: $j = j + 1$
 - 16: **end if**
 - 17: **end if**
 - 18: **end while**
-

Capítulo 3

Montaje Experimental

En este capítulo se presenta el montaje experimental utilizado, incluyendo una descripción de los dispositivos y programas usados, así como los códigos fuente escritos más relevantes. Los experimentos fueron realizados en un laboratorio de la Universidad de Sonora, perteneciente al departamento de Ingeniería en Tecnología Electrónica, del Departamento de Investigación en Física, de la Universidad de Sonora.

3.1 Hardware



Figura 3.1: Vehículo Amigobot, de Adept Mobile Robots

Se utilizó como vehículo autónomo un robot móvil llamado *Amigobot*, mostrado en la Figura 3.1 y desarrollado por la compañía *Adept Mobile Robots*. Éste es un vehículo de control remoto que cuenta con dos ruedas impulsadas cada una con un motor de corriente continua de forma independiente, además de una rueda loca para soporte y estabilidad. El robot cuenta con un controlador interno PID que, por medio de un encoder y una señal de modulación de ancho de pulso, regula la velocidad de las ruedas para alcanzar una velocidad programada. El robot cuenta con un receptor *WiFi* que le permite unirse a una red local de computadoras. Una computadora en la misma red envía por medio de una conexión *TCP/IP* órdenes al robot continuamente.



Figura 3.2: Cámara de infrarrojos Flex 13, de Optitrack

El control de la posición y velocidad del vehículo en el entorno de trabajo requiere la medición precisa de la posición y orientación del robot. De estas mediciones se obtiene el estado actual del sistema de ecuaciones diferenciales. Aunque los encoders instalados en el robot permiten esta funcionalidad, ésta presenta un error de integración en su estimación que va aumentando con el tiempo y distancia recorridas, por lo que no resulta fiable para las largas maniobras que se necesitan. Se utilizó un sistema de visión con seis cámaras *Flex 13* de la compañía *Optitrack*, como la que se muestra en la Figura 3.2. Éstas son cámaras de infrarrojos capaces de detectar la posición de objetos reflejantes de luz infrarroja por medio de triangulación, a una velocidad máxima de 120 cuadros por segundo y hasta una resolución milimétrica. Se fijaron 3 marcadores reflejantes de luz infrarroja en el robot, suficientes para representar la posición y orientación del mismo. Las cámaras envían continuamente datos a una computadora que combina las mediciones de las diferentes cámaras para reconstruir la posición de los marcadores y por lo tanto del robot. Seis de estas cámaras fueron posicionadas circularmente alrededor de un área de trabajo de 3 metros por 3 metros.

3.2 Software

La programación necesaria para el trabajo fue realizada en el software matemático *MATLAB* de la compañía MathWorks. *MATLAB* cuenta con una interfaz de programación de alto nivel con librerías de software preinstaladas para el manejo de operaciones matemáticas básicas. Se utilizó una librería de software para *MATLAB* llamada *Aria* que proporciona la compañía Mobile Robots para facilitar la comunicación con el robot Amigobot. Esta librería de software incluye funciones para el envío de comandos de movimiento al robot, así como para recibir información del estado de éste. Para la recepción y procesamiento la información proveniente de las cámaras Flex 13, se utilizó el software proporcionado por Optitrack, llamado *Motive*. Este software es capaz de enviar a la red, por medio de TCP/IP, paquetes de datos que contienen la posición de los marcadores detectados por las cámaras. Así mismo *Motive* es capaz de agrupar conjuntos de marcadores y reconocerlos como objetos sólidos, enviando también su orientación en forma de cuaterniones hacia la red. Se utilizó una librería para *MATLAB*, proporcionada en conjunto con *Motive*, para la recepción de los paquetes enviados por *Motive*. En el Anexo E se muestran ciertas partes del código fuente realizado en *MATLAB*, para resolver problemas fundamentales del aspecto experimental.

3.3 Esquema experimental

Una vez reconocidos los dispositivos necesarios para los experimentos, se muestra un esquema de los dispositivos en conjunto y la interacción que existe entre ellos. La Figura 3.3 muestra dicho esquema.

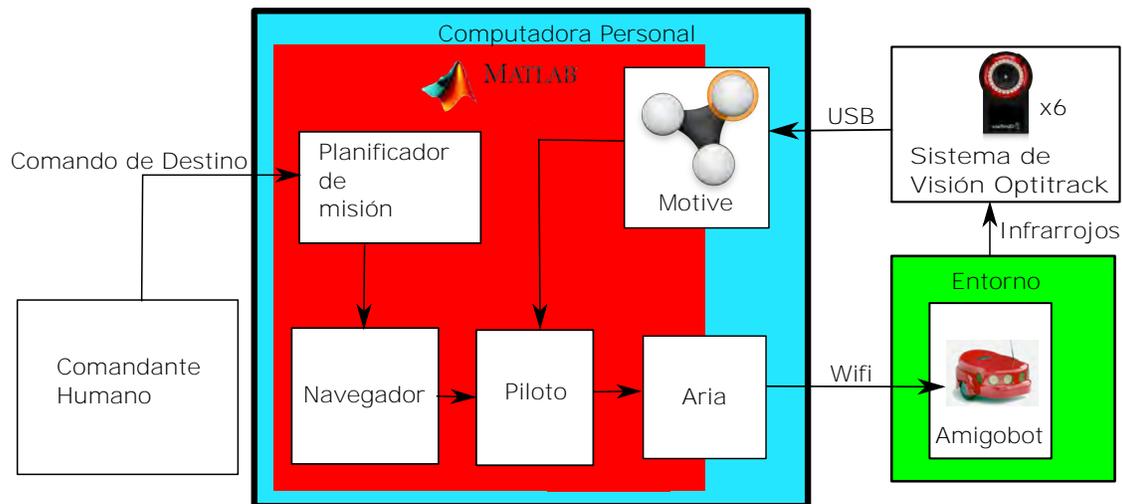


Figura 3.3: Sistema experimental

En el diagrama de la Figura 3.3, las flechas indican la dirección del flujo de información. Todo lo encerrado en azul claro corresponde a software en ejecución en una computadora personal, así mismo el color rojo representa todo lo que se ejecuta en el entorno MATLAB. El comandante humano introduce el destino y la localización de los obstáculos a evadir. En seguida inicia un programa escrito en MATLAB. Después de este paso no se requiere intervención humana alguna. El programa iniciado se conecta con Aria y Motive para la recepción de datos. A continuación se ejecuta la rutina del planificador de misión, el cual forma la gráfica de visibilidad y encuentra el camino más corto posible hacia el objetivo por medio de el Algoritmo de Dijkstra. El camino más corto posible consiste en un conjunto de puntos por los que el robot debe pasar, éstos son mostrados en pantalla junto con la gráfica de visibilidad y los obstáculos. Tales datos son utilizados por el navegador, que genera un conjunto de splines cúbicos, que son mostrados con el fin de utilizar técnicas de análisis numérico para determinar datos como la velocidad y aceleración del robot. Después se muestra en pantalla el camino continuo generado por el navegador. El piloto consiste en una rutina en MATLAB que se ejecuta continuamente, a razón de 10 veces por segundo. En cada ejecución la rutina recibe los datos capturados por el software Motive, procesándolos con los datos generados por el navegador y envía las señales de control necesarias al robot mediante el programa Aria. En pantalla se muestran continuamente gráficas que informan las variables de estado del robot y las señales de control enviadas. Una vez que el robot ha llegado a su destino, se detiene y la rutina finaliza.

Capítulo 4

Resultados

En este capítulo se muestran los resultados obtenidos por mediciones en tiempo real en los experimentos, utilizando el montaje experimental descrito en el Capítulo 3. Los experimentos se realizaron para el controlador basado en el potencial artificial propuesto y para el método de DeLuca, también se discuten los resultados obtenidos.

4.1 Resultados experimentales

Primero se realizaron experimentos para comprobar la eficacia de los controladores de seguimiento, sin utilizar el navegador ni el planificador de misión, esto es debido a que el controlador presentado por deLuca es un controlador de seguimiento de trayectoria y no un sistema completo de guiado autónomo. Se eligió una trayectoria de referencia continua de la forma

$$x_r = A_1 \cos(\omega_1 t + \phi_1) + A_2 \cos(\omega_2 t + \phi_2) \quad (4.1)$$

$$y_r = A_1 \sin(\omega_1 t + \phi_1) + A_2 \sin(\omega_2 t + \phi_2) \quad (4.2)$$

donde ω_m y ϕ_m , con índices $m = 1, 2$ son constantes arbitrarias y donde se tiene que $\omega_1 < \omega_2$.

4.2 Prueba de controlador de DeLuca

Se realizó el experimento para el controlador de DeLuca, donde al robot se le dio la tarea de seguir la trayectoria de referencia de las Ecuaciones (4.1) y (4.2), por medio del controlador de las Ecuaciones (2.30) - (2.32). El robot partió de una posición cercana a la trayectoria, las gráficas fueron obtenidas en tiempo real al ejecutar el experimento. Las gráficas obtenidas se muestran en las Figuras 4.1-4.3.

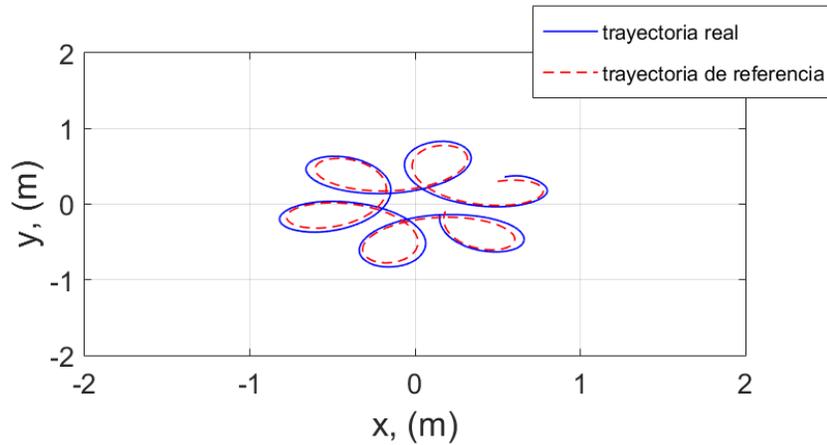


Figura 4.1: Prueba de controlador DeLuca -Trayectorias

En la gráfica de la la Figura 4.1 se muestra la representación del área de trabajo con las distancias medidas en metros, la línea punteada representa la trayectoria de referencia generada y la línea sólida la trayectoria medida del robot con el controlador de DeLuca. De la gráfica se observa que el robot sigue de cerca la trayectoria deseada. No sigue exactamente la trayectoria debido a que el controlador es de regulación y no de seguimiento.

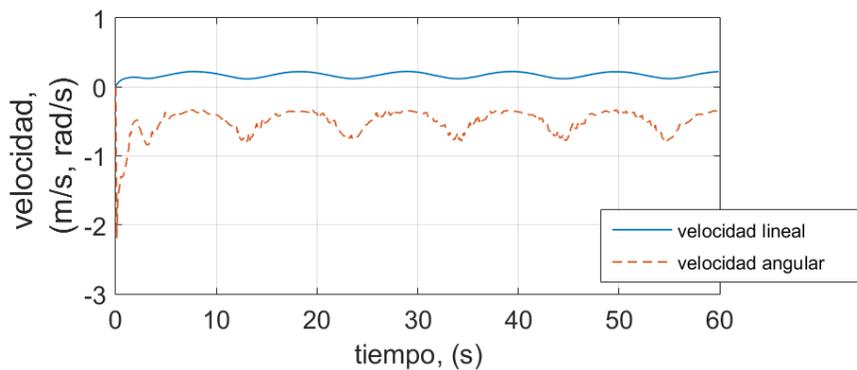


Figura 4.2: Prueba de controlador DeLuca -Señales de control.

La Figura 4.2 muestra las entradas generadas por el controlador de DeLuca, éstas corresponden con la velocidad lineal y angular del robot. Se observa que las entradas de control se mantienen acotadas.

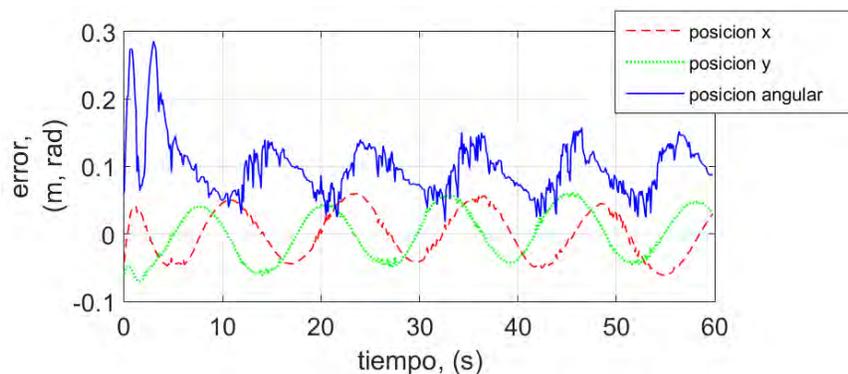


Figura 4.3: Prueba de controlador DeLuca -Errores

Los errores en el tiempo para la posición lineal (x_c, y_c) y angular ϕ se muestran en la Figura 4.3. El error se define como la diferencia entre la posición deseada y la medida, para cada variable. Como se aprecia en la figura, en estado estable los errores de seguimiento tienen un error absoluto máximo de 0.05 m para las posiciones lineales y menor a 0.15 radianes en la posición angular u orientación del robot. El error presenta oscilaciones no amortiguadas correspondientes a la frecuencia angular ω_2 en las Ecuaciones (4.1) y (4.2) de la trayectoria.

4.3 Prueba de controlador de potencial artificial

Se realizó el mismo experimento pero esta vez utilizando el controlador de potencial artificial propuesto anteriormente en la Ecuación (2.22). Las Figuras 4.4 - 4.7 muestran los resultados obtenidos mediante este controlador.

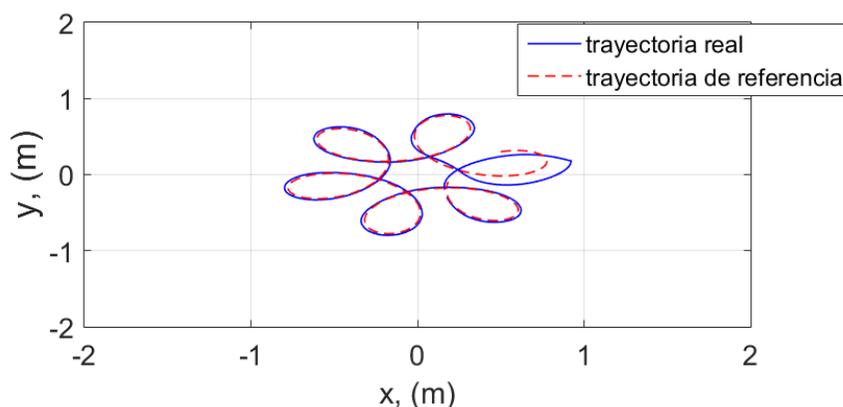


Figura 4.4: Prueba de controlador potencial artificial - Trayectorias

La Figura 4.4 muestra la trayectoria deseada y la realizada por el vehículo. Se observa que se sigue muy de cerca la referencia después de un gran error transitorio inicial.

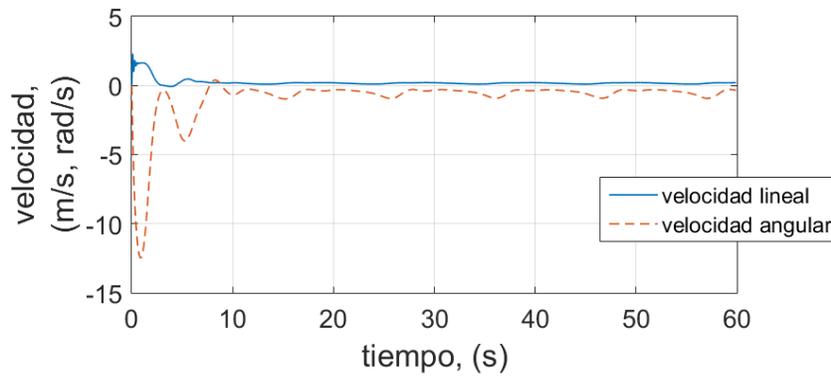


Figura 4.5: Prueba de controlador potencial artificial - Velocidades

La Figura 4.5 muestra las velocidades lineal y angular del vehículo. Después de un tiempo de asentamiento de aproximadamente 4 segundos las velocidades se volvieron periódicas como se esperaba.

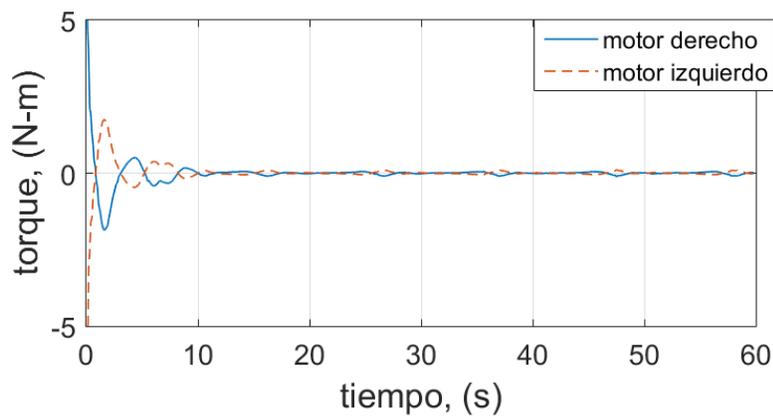


Figura 4.6: Prueba de controlador potencial artificial - Entradas

En la Figura 4.6 se puede ver la gráfica de los torques generados por los motores izquierdo y derecho. Los torques iniciales fueron grandes, debido al error transitorio inicial. Los grandes errores iniciales corresponden con grandes fuerzas producidas por el potencial artificial, al ser el error igual a la distancia hacia el punto de referencia.

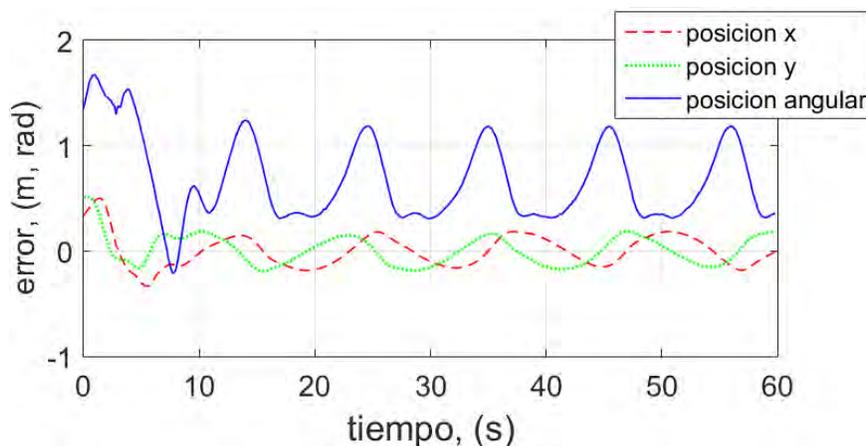


Figura 4.7: Prueba de controlador potencial artificial - Errores

La Figura 4.7 muestra el error en la postura del robot, los errores muestran al igual que el controlador de deLuca un comportamiento oscilatorio estacionario, correspondiente a la frecuencia de oscilación en la trayectoria de referencia. En este caso se muestra un error absoluto máximo en estado estacionario de aproximadamente 0.15 m en posición y 1.23 radianes en el ángulo de orientación. Cabe mencionar que este error, aparentemente más grande que el controlador de deLuca se debe a un retraso entre la posición de la referencia y la posición del robot. Este retraso se produce porque el controlador potencial se basa en obtener un fuerza que aumenta con la distancia, por lo que para continuar produciendo una fuerza que impulse al vehículo se requiere mantener cierta distancia al punto de referencia. Cuando el vehículo se encuentra demasiado cerca al punto de referencia el potencial produce una fuerza muy pequeña y el potencial dependiente de la velocidad domina sobre esa fuerza.

4.4 Búsqueda de ruta con seguimiento de trayectoria

Se realizó un experimento completo, que incluye no sólo el controlador de potencial artificial, sino también el navegador y el planificador de misión. Esto se realizó en la misma área de trabajo cuadrada de $4\text{ m} \times 4\text{ m}$, donde se colocaron dos obstáculos con forma triangular. Estando el vehículo en una posición inicial con coordenadas $(1.2, 0.0)$ es comandado a un destino en coordenadas $(-1.0, 0.0)$. Una vez colocado el robot en su posición inicial, ingresado la localización de los obstáculos y la localización destino, todo esto en el software MATLAB, se inicia la prueba.

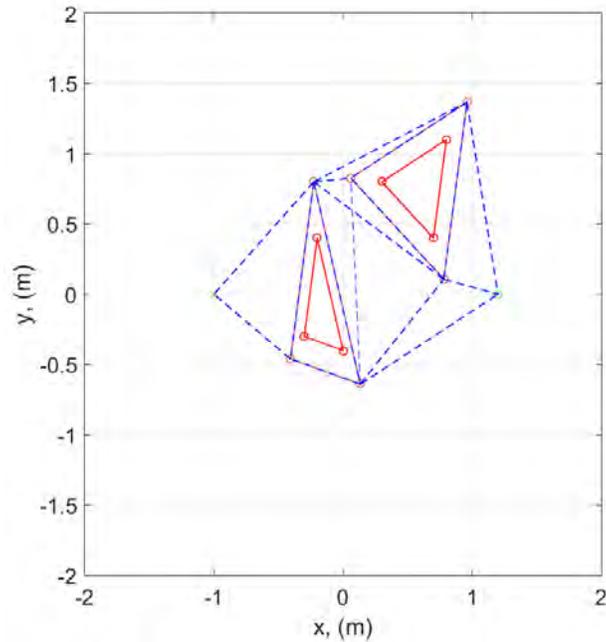


Figura 4.8: Gráfica de visibilidad

En la Figura 4.8 se muestra con líneas punteadas los enlaces generados entre los nodos que forman la gráfica de visibilidad. El triángulo interior formado por líneas sólidas corresponde al obstáculo y el triángulo exterior al obstáculo aumentado del espacio libre de configuración. Los vértices de este triángulo corresponden a los nodos de la gráfica de visibilidad.

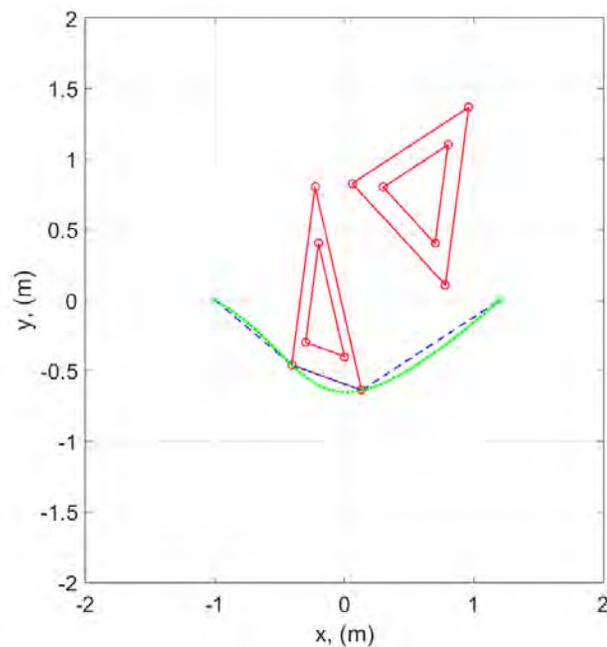


Figura 4.9: Camino óptimo

La trayectoria generada por el navegador es la curva de puntos que se ve en la Figura 4.9,

esta trayectoria está formada por splines cúbicos. Las líneas rectas punteadas corresponden a la ruta nodo a nodo generado por el planificador de misión, que utiliza el algoritmo de Dijkstra, la trayectoria generada pasa por dos nodos además del inicio y el destino. Así mismo con líneas rojas se muestran las aristas de los obstáculos.

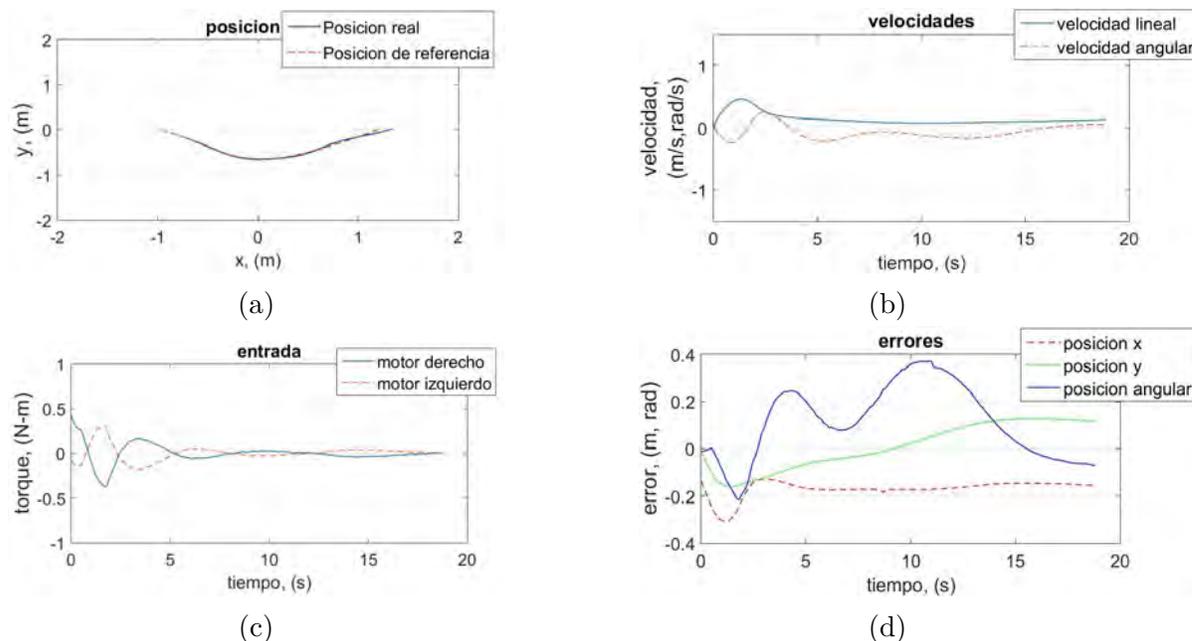


Figura 4.10: Gráficas del experimento

Las mismas gráficas que en las pruebas de los controladores son mostradas para el experimento completo en la Figura 4.10, la duración del experimento fue de 20 segundos. El punto inicial de referencia (x_r, y_r) se define como la posición inicial del robot, por lo que el error inicial es cero y los transitorios iniciales pequeños, esto se puede apreciar en la gráfica de la Figura 4.10 (c) donde los torques de los motores no superan los 0.5 N-m en ningún momento. Además, en la misma gráfica se muestra un comportamiento de oscilaciones amortiguadas, este es el periodo donde el robot ajusta su orientación acercándose cada vez mas a la trayectoria. Los errores, Figura 4.10 (d), muestran el mismo comportamiento del retraso en la posición, se aprecian errores considerables que se mantienen, sin embargo el vehículo mantiene una distancia promedio de 0.18 m al punto actual de la trayectoria de referencia. Finalmente, se puede decir que el vehículo completó correctamente el trayecto de la forma esperada. Una vez obtenidos estos resultados experimentales, se concluye que la estrategia de generación de trayectoria y control autónomo son exitosas.

Conclusiones

La investigación inicial reveló una gran cantidad teorías aplicables a la robótica móvil, en especial a robots sobre ruedas. La diversidad de conocimientos en diferentes ramas que son necesarios para el funcionamiento de un robot móvil lo convierten en un tema multidisciplinario.

Esto se vio reflejado en que la aplicación de conocimientos de algunas disciplinas además de la física, para la generación de modelos mecánicos, como la electrónica para el control de los actuadores, programación para la implementación de los algoritmos e integración de los diferentes sistemas, la ingeniería de control que presenta un enfoque clásico, inteligencia artificial para armar una estrategia para resolver el problema, la geometría diferencial para estudiar de forma más profunda el comportamiento de las ecuaciones diferenciales, incluso se relacionó con temas que parecen a simple vista tan distantes como la psicología con los patrones de comportamiento. El uso de técnicas de diferentes disciplinas es en realidad necesario para lograr un comportamiento aceptable de forma práctica, y no sólo eso, también se requiere de una estrategia que integre de forma correcta estas técnicas, como la estrategia por jerarquías usada.

Se realizaron dos análisis mecánicos para el robot móvil terrestre, uno cinemático y otro dinámico. En el análisis cinemático se aplica el conocimiento geométrico y de restricciones para llegar a un modelo. En cambio, el análisis dinámico requirió del uso de la mecánica newtoniana por lo que requiere mayor trabajo y consideraciones para llegar al modelo. Estos análisis se pusieron a prueba experimentalmente en conjunto con las estrategias de control para el robot de forma satisfactoria.

Las pruebas mostraron que los potenciales artificiales pueden obtener resultados similares a los obtenidos por métodos clásicos exactos. El comportamiento del robot se vio sensiblemente influido por los valores de las constantes de sintonización del controlador y por el orden de la función potencial. Los valores requeridos para la estabilidad en cada orden son significativamente diferentes y su ajuste manual es complicado. Como se esperaba, el incremento en el valor de las constantes reduce el error estacionario. El aumento en el orden de la función potencial lleva a generar rápidas y cortas aceleraciones cuando se tienen grandes errores y a su vez obtiene un comportamiento muy suave una vez reducidos los errores. Se llegó a la conclusión que en la configuración experimental usada, un potencial de tercer orden como el de la Ecuación (2.22) tiene mucho mejor comportamiento que uno de primer o segundo orden. Se probó brevemente con funciones de cuarto orden pero no se notó una diferencia significativa. Se utilizó el controlador de potencial artificial como piloto en conjunto con el navegador y el planeador de misión. También podría ser usado en solitario por lo que no se caracterizaron por completo sus formas de utilización en esta investigación,

aunque en la bibliografía se pueden encontrar ejemplos de uso en solitario como en [18], [17], [15].

Un punto que se tiene que destacar es el uso de los potenciales de velocidad para amortiguar las velocidades del vehículo. Sin tales términos de potencial de velocidad es difícil lograr un comportamiento aceptable en el robot, obteniendo grandes oscilaciones en las mediciones de error. Este comportamiento corresponde a un sistema dinámico sub-amortiguado.

Una diferencia considerable que se presentó es el mayor tiempo de repuesta del controlador de potencial artificial antes de llegar a errores lo suficientemente pequeños, comparado con el método de DeLuca. Este tiempo puede ser reducido al aumentar el orden de la función potencial. Pero aumentar el orden de esta ecuación introduce un retraso considerable en la posición obtenida por el robot, esto es, el robot siempre se encuentra en una localización de referencia anterior y no en la actual. El retraso es del orden de décimas de segundo. Este comportamiento es similar al de un remolque jalado por un vehículo, siendo el vehículo el robot virtual de referencia y el remolque el robot real.

Anexo A

Matrices ortogonales y ángulos de Euler

Teniendo un marco de referencia \mathcal{I} , podemos obtener una transformación lineal que relacione los vectores base ortonormales $\hat{x}, \hat{y}, \hat{z}$ con los vectores base $\hat{x}', \hat{y}', \hat{z}'$ de otro marco ortonormal \mathcal{B} , utilizando los ángulos de Euler $\{\phi, \theta, \psi\}$ y matrices de rotación sobre los ejes base, mediante la ecuación

$$\begin{aligned} \mathbf{R}(\phi, \theta, \psi) &= \mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\psi & s_\theta \\ c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi - s_\phi s_\theta s_\psi & -s_\phi c_\theta \\ s_\phi s_\psi - c_\phi s_\theta c_\psi & s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix} \end{aligned} \quad (\text{A.1})$$

donde se utiliza la notación $c_\varphi = \cos \varphi$, $s_\varphi = \sin \varphi$ y $t_\varphi = \tan \varphi$ [39].

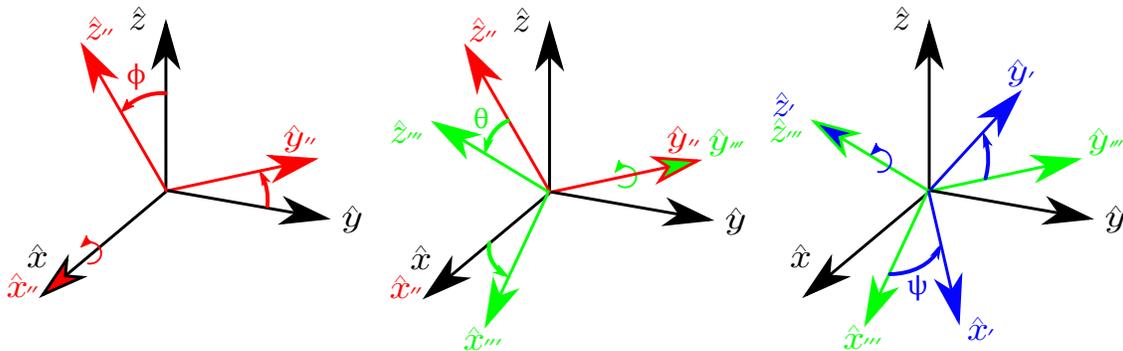


Figura A.1: 3 rotaciones sucesivas el marco sobre \mathcal{B}

La Ecuación (A.1) define una matriz de rotación \mathbf{R} obtenida por rotaciones intrínsecas, teniendo inicialmente coincidentes los vectores del marco \mathcal{B} con los del marco \mathcal{I} y aplicando rotaciones sucesivas a los vectores $\hat{x}', \hat{y}', \hat{z}'$ en los ejes $x - y - z$ actuales, en ese orden y en sentido contrario a las agujas del reloj, como se observa en la Figura A.1. Por tanto \mathbf{R} es una matriz cuyas columnas son los componentes de los vectores (1.2) expresadas en la base

de \mathcal{I} (1.1). Esta matriz de rotación sirve para transformar los componentes de un vector \mathbf{q} en la base de \mathcal{B} a sus componentes en \mathcal{I} , de acuerdo a la siguiente ecuación

$$[\mathbf{q}] = \mathbf{R}(\phi, \theta, \psi) [\mathbf{q}]' \quad (\text{A.2})$$

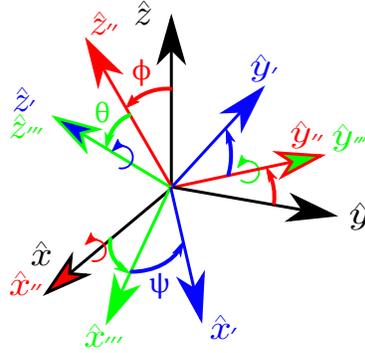


Figura A.2: Ejes de rotación instantáneos

El vector velocidad angular $\boldsymbol{\Omega}$ del marco \mathcal{B} puede expresarse tanto por los vectores de \mathcal{I} como los vectores de \mathcal{B} . La rotación debido al cambio en los ángulos de Euler $\{\phi, \theta, \psi\}$ sucede en 3 ejes diferentes, tal como se muestra en la Figura A.2, éstos son $\hat{\mathbf{x}} = \hat{\mathbf{x}}''$, $\hat{\mathbf{y}}' = \hat{\mathbf{y}}'''$, $\hat{\mathbf{z}}''' = \hat{\mathbf{z}}'$, respectivamente [24]. Las ecuaciones para expresar la velocidad angular se dan a continuación

$$\boldsymbol{\Omega} = \dot{\phi} \hat{\mathbf{x}} + \dot{\theta} \hat{\mathbf{y}}'' + \dot{\psi} \hat{\mathbf{z}}' \quad (\text{A.3})$$

$$[\boldsymbol{\Omega}] = \mathbf{S} \dot{\boldsymbol{\eta}} = \begin{bmatrix} 1 & 0 & s_{\theta} \\ 0 & c_{\phi} & -s_{\phi} c_{\theta} \\ 0 & s_{\phi} & c_{\phi} c_{\theta} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (\text{A.4})$$

$$[\boldsymbol{\Omega}]' = \mathbf{S}' \dot{\boldsymbol{\eta}} = \begin{bmatrix} c_{\theta} c_{\psi} & s_{\psi} & 0 \\ -c_{\theta} s_{\psi} & c_{\psi} & 0 \\ s_{\theta} & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (\text{A.5})$$

donde la Ecuación (A.3) representa la velocidad angular como una suma vectorial de velocidades angulares y las Ecuaciones (A.4) y (A.5) relacionan los componentes del vector velocidad angular y los ángulos de Euler en las bases $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ y $\hat{\mathbf{x}}', \hat{\mathbf{y}}', \hat{\mathbf{z}}'$, respectivamente. Para la postura de los vehículos aéreos se suelen usar los llamados ángulos de navegación, también llamados guiñada (*yaw*), cabeceo (*pitch*) y alabeo (*roll*). Estos ángulos corresponden a los de Tait-Bryan para 3 rotaciones intrínsecas en los ejes z, y, x actuales [40]. En este

caso se tienen las transformaciones para orientación y velocidad angular:

$$\begin{aligned}
 \mathbf{R}(\phi, \theta, \psi) &= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix} \\
 &= \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix} \quad (\text{A.6})
 \end{aligned}$$

$$\begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} 0 & -s_\phi & c_\phi c_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 1 & 0 & -s_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (\text{A.7})$$

$$\begin{bmatrix} \Omega'_x \\ \Omega'_y \\ \Omega'_z \end{bmatrix} = \begin{bmatrix} -s_\theta & 0 & 1 \\ c_\theta s_\psi & c_\psi & 0 \\ c_\theta c_\psi & -s_\psi & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (\text{A.8})$$

donde la matriz de rotación de la Ecuación (A.6) es diferente a la de la Ecuación (A.3), y las transformaciones de las Ecuaciones (A.7) y (A.8) son diferentes a las de las Ecuaciones (A.4) y (A.5) debido a que las rotaciones sucesivas se consideran sobre distinto orden en los ejes.

Anexo B

Cuaterniones

B.1 Introducción

Un formalismo adicional que se puede utilizar para representar rotaciones en un espacio tridimensional es el de los cuaterniones. El conjunto de los cuaterniones es denotado por \mathbb{H} y son una generalización de los números complejos que se obtienen agregando dos unidades imaginarias más: \mathbb{j} y \mathbb{k} . Además, estos números cumplen con las siguientes ecuaciones [41]:

$$\begin{aligned}i^2 = \mathbb{j}^2 = \mathbb{k}^2 = i\mathbb{j}\mathbb{k} &= -1 \\i\mathbb{j} = -\mathbb{j}i &= \mathbb{k} \\ \mathbb{j}\mathbb{k} = -\mathbb{k}\mathbb{j} &= i \\ \mathbb{k}i = -i\mathbb{k} &= \mathbb{j}\end{aligned}$$

Un número cuaternión puede ser representado de diferentes maneras y forman un espacio vectorial sobre \mathbb{R}^4 , ya que se representa con 4 parámetros. Así mismo, los cuaterniones forman un grupo con respecto a la operación de multiplicación [42]. Los números cuaterniones se pueden representar de las siguientes formas:

$$\begin{aligned}h &= h_0 + h_1i + h_2\mathbb{j} + h_3\mathbb{k} \\ &= h_w + h_xi + h_y\mathbb{j} + h_z\mathbb{k} = h_w + \vec{h} \\ [h] &= \begin{bmatrix} h_w \\ h_x \\ h_y \\ h_z \end{bmatrix} = \begin{bmatrix} h_w \\ \vec{h} \end{bmatrix}\end{aligned}\tag{B.1}$$

en (B.1) h_w es la parte escalar en \mathbb{R} del cuaternión y $\vec{h} = h_xi + h_y\mathbb{j} + h_z\mathbb{k}$ representa la parte imaginaria del cuaternión, también llamada parte vectorial, ya que corresponde a un elemento en \mathbb{R}^3 con componentes $[h_x, h_y, h_z]$ y por lo tanto, también los componentes de un vector \mathbf{q} , dada una base, como se expresa en

$$\mathbf{q} = q_xi + q_y\mathbb{j} + q_z\mathbb{k} \leftrightarrow [\mathbf{q}] = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}$$

Se dice que un cuaternión es imaginario puro cuando su parte escalar es igual a cero [39]. El conjugado de un cuaternión se obtiene negando su parte imaginaria. Las siguientes ecuaciones expresan algunas propiedades de los cuaterniones:

$$\begin{aligned}
\mathfrak{q}(\mathfrak{h}_1 + \mathfrak{h}_2) &= \mathfrak{q}\mathfrak{h}_1 + \mathfrak{q}\mathfrak{h}_2 \\
\mathfrak{q}\mathfrak{h}\mathfrak{u} &= (\mathfrak{q}\mathfrak{h})\mathfrak{u} = \mathfrak{q}(\mathfrak{h}\mathfrak{u}) \\
\mathfrak{h}^* &= h_w - \vec{\mathfrak{h}} \\
(\mathfrak{h}\mathfrak{q})^* &= \mathfrak{q}^*\mathfrak{h}^* \\
\mathfrak{q}\mathfrak{h} &\neq \mathfrak{h}\mathfrak{q} \\
\|\mathfrak{q}\| &= \sqrt{\mathfrak{q}\mathfrak{q}^*} = \sqrt{h_w^2 + h_x^2 + h_y^2 + h_z^2} \\
\|\mathfrak{q}\mathfrak{h}\| &= \|\mathfrak{q}\| \|\mathfrak{h}\| \\
\mathfrak{q}^{-1} &= \frac{\mathfrak{q}^*}{\|\mathfrak{q}\|^2}
\end{aligned} \tag{B.2}$$

La suma de cuaterniones es una suma componente a componente, como la suma común en \mathbb{R}^4 , expresado como

$$\mathfrak{h} + \mathfrak{q} = h_w + q_w + (h_x + q_x)\mathfrak{i} + (h_y + q_y)\mathfrak{j} + (h_z + q_z)\mathfrak{k} = h_w + q_w + \vec{\mathfrak{h}} + \vec{\mathfrak{q}}$$

La multiplicación de cuaterniones es una operación asociativa y distributiva, pero no conmutativa, además existe un inverso[43]. En la multiplicación de cuaterniones se puede hacer uso del producto punto euclidiano y el producto cruz considerando las partes imaginarias como vectores \mathbb{R}^3 utilizando la ecuación

$$\mathfrak{h}\mathfrak{q} = (h_w q_w - \vec{\mathfrak{h}} \cdot \vec{\mathfrak{q}}) + h_w \vec{\mathfrak{q}} + q_w \vec{\mathfrak{h}} + \vec{\mathfrak{h}} \times \vec{\mathfrak{q}}$$

En notación matricial se pueden representar las ecuaciones anteriores para la suma y multiplicación de cuaterniones como :

$$\begin{aligned}
[\mathfrak{h} + \mathfrak{q}] &= \begin{bmatrix} h_w + q_w \\ h_x + q_x \\ h_y + q_y \\ h_z + q_z \end{bmatrix} \\
[\mathfrak{h}\mathfrak{q}] &= Q_L(\mathfrak{h})[\mathfrak{q}] = Q_R(\mathfrak{q})[\mathfrak{h}] \\
Q_L(\mathfrak{h}) &= \begin{bmatrix} h_w & -[\vec{\mathfrak{h}}]^T \\ [\vec{\mathfrak{h}}] & h_w I_{3 \times 3} + \widehat{[\vec{\mathfrak{h}}]} \end{bmatrix} = \begin{bmatrix} h_w & -h_x & -h_y & -h_z \\ h_x & h_w & -h_z & h_y \\ h_y & h_z & h_w & -h_x \\ h_z & -h_y & h_x & h_w \end{bmatrix} \\
Q_R(\mathfrak{q}) &= \begin{bmatrix} q_w & -[\vec{\mathfrak{q}}]^T \\ [\vec{\mathfrak{q}}] & q_w I_{3 \times 3} - \widehat{[\vec{\mathfrak{q}}]} \end{bmatrix} = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix}
\end{aligned}$$

El operador presente en $\widehat{[\vec{q}]}$ mapea $[\vec{q}]$ a una matriz que aplica el producto cruz $[\vec{q}] \times$ a vectores columna multiplicando la matriz por la izquierda $\widehat{[\vec{q}]}[h] = [\vec{q}] \times [h]$. Las matrices $Q_R(\mathfrak{h})$ y $Q_L(\mathfrak{q})$ efectúan la multiplicación del cuaternión columna por la derecha y por la izquierda, respectivamente, y son matrices ortogonales cuando su argumento es un cuaternión unitario, ya que conservan la norma de los cuaterniones [44] [45].

El producto punto euclidiano en tres dimensiones y el producto vectorial pueden representarse en cuaterniones imaginarios puros [46], por medio de

$$\begin{aligned} \mathfrak{q} \cdot \mathfrak{h} &= -\frac{1}{2}(\mathfrak{q}\mathfrak{h} + \mathfrak{h}\mathfrak{q}) \\ \mathfrak{q} \times \mathfrak{h} &= \frac{1}{2}(\mathfrak{q}\mathfrak{h} - \mathfrak{h}\mathfrak{q}) \end{aligned} \tag{B.3}$$

Se dice que un cuaternión es unitario cuando $\|\mathfrak{q}\| = 1$ y corresponde con un vector unitario en \mathbb{R}^4 , para un cuaternión unitario mediante la Ecuación (B.2) obtenemos que su inverso corresponde a su conjugado $\mathfrak{q}^{-1} = \mathfrak{q}^*$.

B.2 Rotaciones con cuaterniones

Para obtener los componentes $[\mathfrak{q}]$ en la base de \mathcal{I} de un vector \mathbf{q} , se puede utilizar un cuaternión unitario $\mathfrak{h} = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} \mathfrak{u}$ y donde el marco del cuerpo, \mathcal{B} , se obtiene al rotar el marco \mathcal{B} empezando desde \mathcal{I} un ángulo θ , alrededor del eje representado por \mathfrak{u} , que es un cuaternión imaginario puro, también unitario.

Para hacer corresponder un cuaternión imaginario puro \mathfrak{q}' con los componentes $[\mathfrak{q}]'$ del vector \mathbf{q} y obtener el cuaternión imaginario puro \mathfrak{q} que corresponde con los componentes $[\mathfrak{q}]$ en la base de \mathcal{I} [39], se utilizan las siguientes ecuaciones:

$$\begin{aligned} \mathfrak{q} &= \mathfrak{h}\mathfrak{q}'\mathfrak{h}^* = (h_w^2 - \vec{\mathfrak{h}} \cdot \vec{\mathfrak{h}})\vec{\mathfrak{q}} + 2(\vec{\mathfrak{h}} \cdot \vec{\mathfrak{q}})\vec{\mathfrak{h}} + 2h_w(\vec{\mathfrak{h}} \times \vec{\mathfrak{q}}) \\ \mathfrak{q}' &= \mathfrak{h}^*\mathfrak{q}\mathfrak{h} \end{aligned} \tag{B.4}$$

$$[\mathfrak{q}] = Q_R(\mathfrak{h}^*)Q_L(\mathfrak{h})[\mathfrak{q}'] = R(\mathfrak{h})[\mathfrak{q}'] = \begin{bmatrix} 0 \\ q_x \\ q_y \\ q_z \end{bmatrix}$$

Una secuencia de rotaciones se puede formar a través de la multiplicación de cuaterniones de la misma forma en la que se hace con matrices de rotación, por ejemplo si se quiere obtener un cuaternión \mathfrak{h}_t que efectúe una rotación dada por \mathfrak{h}_1 , seguida de una rotación dada por \mathfrak{h}_2 , podemos premultiplicar \mathfrak{h}_2 a \mathfrak{h}_1 :

$$\begin{aligned} \mathfrak{h}_t &= \mathfrak{h}_2\mathfrak{h}_1 \\ \mathfrak{q} &= \mathfrak{h}_t\mathfrak{q}'\mathfrak{h}_t^* \\ [\mathfrak{q}] &= R(\mathfrak{h}_t)[\mathfrak{q}'] \end{aligned}$$

En el caso en que se requiera hacer una rotación intrínseca según el marco \mathcal{B} , se post-multiplica el cuaternión $h_t = h_1 h_2$.

La derivada de la Ecuación (B.4) se obtiene [47] mediante

$$\begin{aligned}
 \dot{q} &= \dot{h}q'h^* + h\dot{q}'h^* + hq'h^* \\
 &= \dot{h}h^*qh^* + hh^*q\dot{h}^* + h\dot{q}'h^* \\
 &= \dot{h}h^*q + q\dot{h}h^* + h\dot{q}'h^* \\
 &= (\dot{h}h^*)q - q(\dot{h}h^*) + h\dot{q}'h^* \\
 &= 2(\dot{h}h^*) \times q + h\dot{q}'h^* \\
 &= \omega \times q + h\dot{q}'h^*
 \end{aligned} \tag{B.5}$$

La ecuación para la velocidad angular ω , en términos de cuaterniones es

$$\omega = 2\dot{h}h^*$$

de la cual, despejando se puede obtener una relación para la derivada \dot{h} en términos de ω :

$$\dot{h} = \frac{1}{2}\omega h$$

En la Ecuación (B.5) se reconoce la Ecuación (1.3) ya que ω representa los componentes de la velocidad angular del marco \mathcal{B} en la base de \mathcal{I} $[\omega] = [\Omega]$.

B.3 Ecuaciones de movimiento con cuaterniones

En términos de cuaterniones y utilizando el formalismo de Lagrange, podemos obtener el lagrangiano L , el cual está definido como la energía cinética menos la energía potencial, de la siguiente manera:

$$L = T - V = L_{rot}(\Omega) + L_{trans}(\mathbf{v}_G)$$

donde la parte de energía translacional del lagrangiano esta dada por:

$$L_{trans}(\mathbf{v}_G) = \frac{1}{2}[v]^T M [v] - mgz$$

y la parte de energía rotacional se expresa mediante:

$$L_{rot}(\Omega) = \frac{1}{2}([\Omega]')^T [I]' [\Omega]'$$

De la parte translacional se obtienen las ecuaciones ya conocidas para una partícula puntual sometida a una fuerza [32].

Los cuaterniones son una representación redundante para las rotaciones, ya que contienen cuatro parámetros cuando sólo hay 3 grados de libertad rotacionales para un cuerpo rígido. La condición de cuaternión unitario puede utilizarse con el metodo de multiplicadores de

lagrange para reducir un grado de libertad y resolver el sistema de ecuaciones. Esta condición se expresa mediante las siguientes ecuaciones.

$$\begin{aligned} g(\mathfrak{h}) &= [\mathfrak{h}]^T [\mathfrak{h}] - 1 = 0 \\ \frac{\partial g(\mathfrak{h})}{\partial [\mathfrak{h}]} &= 2[\mathfrak{h}] \end{aligned}$$

La parte rotacional en términos de cuaterniones se puede escribir como

$$\begin{aligned} L_{rot}(\Omega) &= \frac{1}{2}([\Omega]')^T [I]' [\Omega]' \\ &= 2([\mathfrak{h}\mathfrak{h}^*])^T [I]'([\mathfrak{h}\mathfrak{h}^*]) \\ \frac{d}{dt} \frac{\partial L_{rot}}{\partial [\mathfrak{h}^*]} - \frac{\partial L_{rot}}{\partial [\mathfrak{h}]} - [F_{\mathfrak{h}}] &= 2\lambda \left[\frac{\partial g(\mathfrak{h})}{\partial [\mathfrak{h}]} \right] \end{aligned}$$

aquí $F_{\mathfrak{h}} \in \mathbb{R}^4$ es la fuerza generalizada y por tanto puede ser representado como un cuaternión. Estas fuerzas generalizadas pueden obtenerse utilizando desplazamientos virtuales. Para un desplazamiento virtual de \mathfrak{h} , utilizando el mismo método con que se obtuvo ω , se puede llegar a una expresión del trabajo realizado mediante:

$$\delta W = F \cdot ([\delta \mathfrak{q}])$$

con la variación de \mathfrak{q} que se obtiene como:

$$(\delta \mathfrak{q}) = (\delta \mathfrak{h}\mathfrak{h}^*)\mathfrak{q} - \mathfrak{q}(\delta \mathfrak{h}\mathfrak{h}^*)$$

se puede llegar a

$$\begin{aligned} \delta W &= [F]^T (Q_R(\mathfrak{q})Q_R(\mathfrak{h}^*) - Q_L(\mathfrak{q})Q_R(\mathfrak{h}^*)) [\delta \mathfrak{h}] \\ &= [F]^T (Q_R(\mathfrak{q}) - Q_L(\mathfrak{q})) Q_R(\mathfrak{h}^*) [\delta \mathfrak{h}] \end{aligned}$$

Utilizando la Ecuación (B.3) para el producto cruz en la expresión anterior, aparece el término $[F] \times [\mathfrak{q}]$, que es el torque que aplica la fuerza.

$$\begin{aligned} \delta W &= [T]^T Q_R(\mathfrak{h}^*) [\delta \mathfrak{h}] = [F_{\mathfrak{h}}] [\delta \mathfrak{h}] \\ [T] &= -(Q_R(\mathfrak{q}) - Q_L(\mathfrak{q})) [F] = [F] \times [\mathfrak{q}] \\ [F_{\mathfrak{h}}] &= [T]^T Q_R(\mathfrak{h}^*) \end{aligned}$$

Las ecuaciones de movimiento pueden expresarse tanto en cuaterniones como en forma matricial y se muestran en las siguientes ecuaciones:

$$\begin{aligned}
4\mathfrak{h}I_Q(\mathfrak{h}^*\ddot{\mathfrak{h}} + \dot{\mathfrak{h}}^*\dot{\mathfrak{h}}) + 8\dot{\mathfrak{h}}I_Q\mathfrak{h}^*\dot{\mathfrak{h}} - 2\lambda\mathfrak{h} - F_{\mathfrak{h}} &= 0 \\
4Q_L(\mathfrak{h})[I'_Q](Q_L(\mathfrak{h}^*)[\ddot{\mathfrak{h}}] + Q_L(\dot{\mathfrak{h}}^*)[\dot{\mathfrak{h}}]) \\
+ 8Q_L(\dot{\mathfrak{h}})[I'_Q]Q_L(\mathfrak{h}^*)[\dot{\mathfrak{h}}] - 2\lambda[\mathfrak{h}] - [F_{\mathfrak{h}}] &= 0 \\
[I'_Q] &= \begin{bmatrix} I'_{11} + I'_{22} + I'_{33} & 0 \\ 0 & [I'] \end{bmatrix}
\end{aligned}$$

donde $[I'_Q]$ es una matriz de inercia extendida a \mathbb{R}^4 [48], expresada en notación de cuaterniones. Ésta representa una operación de mapeo $\mathbb{H} \rightarrow \mathbb{H}$ para cuaterniones imaginarios puros. Por lo tanto se tiene que

$$\mathfrak{q} = q_x\mathfrak{i} + q_y\mathfrak{j} + q_z\mathfrak{k} \rightarrow I'_Q\mathfrak{h} = q_xI'_{11}\mathfrak{i} + q_yI'_{22}\mathfrak{j} + q_zI'_{33}\mathfrak{k}$$

Anexo C

Interpolación

La interpolación consiste en la estimación de valores $y \in Y$ de una función desconocida $y = f(x)$ en un subconjunto convexo de su dominio $I \subset X$, donde X es el dominio de $f(x)$, conociendo sólo un conjunto finito $\{y_i : 0 < i < n, y_i = f(x_i)\}$ de valores en I [49]. La interpolación entonces encuentra una aproximación para la función desconocida y debe al menos coincidir en los valores conocidos. En el caso de que se estimen valores fuera de I se le llama extrapolación.

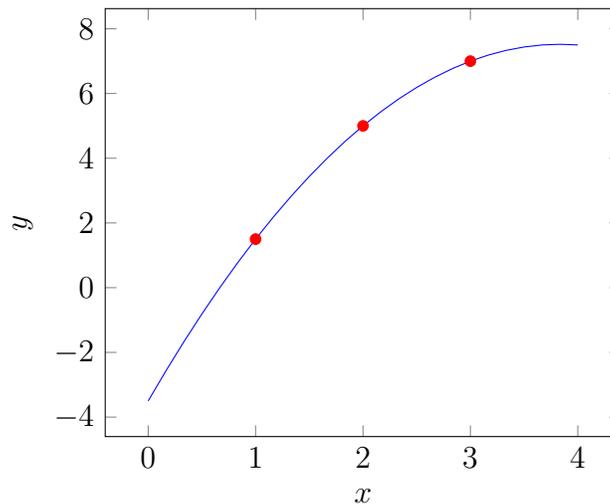


Figura C.1: Ejemplo de interpolación polinomial $f(x) = -0.75x^2 + 5.75x - 3.5$

Tabla C.1: Puntos a interpolar

i	x_i	y_i
0	1	1.5
1	2	5
2	3	7

C.1 Interpolación polinomial

En el caso de funciones reales $f : \mathbb{R} \rightarrow \mathbb{R}$ de variable real, el problema se puede representar gráficamente como encontrar una curva que pase por todo el conjunto de puntos (x_i, y_i) en la gráfica. A la función $f : \mathbb{R} \rightarrow \mathbb{R}$, donde

$$f = \sum_{i=0}^n a_i x^i \quad (\text{C.1})$$

se le llama interpolación polinomial de grado n . Como ejemplo considere los puntos, en coordenadas (x, y) , que se muestran en la Tabla C.1. Un polinomio interpolador que atraviesa esos puntos es $f(x) = -0.75x^2 + 5.75x - 3.5$, como se muestra en la Figura C.1. Los $n + 1$ coeficientes del polinomio a_i caracterizan por completo a éste. La forma más común de abordar el problema de generación de trayectoria es por medio de interpolación polinomial, ya que los polinomios siempre son continuos $C^\infty(\mathbb{R})$ y son bien comportados para todo \mathbb{R} . Dados $n + 1$ puntos (x_i, y_i) se requiere, en general, un polinomio de grado n para asegurar que éste pueda pasar por todos los puntos. La función del polinomio de interpolación se forma mediante la siguiente ecuación

$$f(x) = \sum_{i=0}^n L_i(x) y_i$$

donde cada una de las funciones $L_i(x)$ se definen mediante

$$L_i(x) = \prod_{j=0; j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)} \quad (\text{C.2})$$

Dada una secuencia de puntos (x_i, y_i) , existe una forma sencilla de encontrar el polinomio interpolador, esto es por medio de los polinomios interpolantes de Lagrange [49], de la Ecuación (C.2). La expresión para el polinomio no se encuentra tan conveniente como la de la Ecuación (C.1), pero es fácil de utilizar en base a los puntos por los que debe pasar. En la Ecuación (C.2) se puede ver que la función $L_i(x)$ se hace cero en todos los puntos $x = x_j$ para $j = 1, 2, 3, \dots$, excepto en $x = x_i$ en el cual $L_i(x_i) = 1$. Entonces para todo x_i $L(x_i) = y_i$, que resulta en $f(x) = y_i$. Para el caso mostrado en el ejemplo de la Figura C.1 se tiene el polinomio expresado según la Ecuación (C.2) como

$$f(x) = y_0 L_0 + y_1 L_1 + y_2 L_2 = 1.5 \frac{(x-2)(x-3)}{(1-2)(1-3)} + 5 \frac{(x-1)(x-3)}{(2-1)(2-3)} + 7 \frac{(x-1)(x-2)}{(3-1)(3-2)}$$

C.2 Splines

Un problema con la interpolación polinomial es que, para polinomios de alto grado, suelen generar curvas con grandes oscilaciones, no aptas para su uso en generación de trayectorias.

Tabla C.2: Valores a interpolar

i	x_i	y_i
0	1	3
1	2	4
2	3	4.9
3	4	5
4	5	8
5	6	7.8
6	7	9

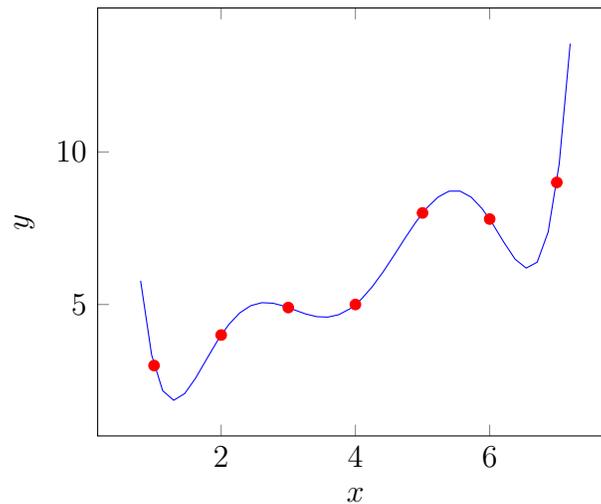


Figura C.2: Interpolación polinomial $f(x) = 55.9 - 126.738x + 111.959x^2 - 47.4313x^3 + 10.3924x^4 - 1.13042x^5 + 0.0481944x^6$

En la Tabla C.2 se muestran coordenadas de puntos a los que se le aplicará una interpolación polinomial. En la Figura C.2 se muestran los puntos dados y la curva de interpolación. Este es un caso donde se tienen considerables oscilaciones al inicio y final del intervalo. Si se tienen $n + 1$ puntos, no siempre se puede encontrar un polinomio de grado menor a n que interpole todos los valores. Una solución para esto es utilizar n polinomios $f_i(x)$, de grado $m < n$, que interpolen los valores, de forma que el polinomio $f_i(x)$ interpola los valores entre x_i y x_{i+1} . Cada polinomio requiere de $m + 1$ coeficientes, en total $n(m + 1)$ coeficientes deben determinarse de $n + 1$ valores a interpolar, por lo que se deben determinar restricciones para eliminar los $nm - 1$ grados de libertad adicionales. Las condiciones

$$f_i(x_{i+1}) = f_{i+1}(x_{i+1}), \quad 0 < i < n - 1 \quad (\text{C.3})$$

$$f_0(x_0) = y_0 \quad (\text{C.4})$$

$$f_{n-1}(x_n) = y_n \quad (\text{C.5})$$

$$f_i(x_i) = y_i \quad (\text{C.6})$$

aseguran la continuidad de $f(x)$ de un intervalo a otro. Condiciones adicionales pueden obtenerse al considerar la continuidad de las $n - 1$ derivadas de $f(x)$. Una vez que se tiene

el conjunto de condiciones y se han resuelto todas las ambigüedades, se pueden reordenar para obtener un sistema lineal de ecuaciones simultáneas para los $n(m+1)$ coeficientes. Se abordará el caso donde $m=3$, y se requiere continuidad en la primera y segunda derivada. A la función generada en este caso se le llama un spline cúbico. Se abordarán los polinomios cúbicos debido a que son los de mínimo grado necesario para conseguir el movimiento de un punto a otro, además de permitir elegir la velocidad en estos dos puntos arbitrariamente. Así se permite mantener la primera derivada continua y diferenciable que a su vez permite una segunda derivada continua [29]. La ecuación para la segunda derivada se determina mediante

$$f_i''(x) = f_i''(x_i) \frac{(x - x_{i+1})}{x_i - x_{i+1}} + f_i''(x_{i+1}) \frac{(x - x_i)}{x_{i+1} - x_i} \quad (\text{C.7})$$

Aunque se pueden determinar los spline cúbicos de acuerdo a las Ecuaciones (C.3)-(C.6), existe una forma computacionalmente más eficiente para realizarlo. Esta técnica se basa en notar que la segunda derivada de un spline cúbico se puede representar como una interpolación lineal de puntos, como en la Ecuación (C.7), esto es debido a que la segunda derivada de un polinomio de tercer grado es una línea recta y a que además se requiere la continuidad de la segunda derivada entre los polinomios interpolantes. De la Ecuación (C.7) se puede obtener $f(x)$ integrando dos veces, con lo cual se llega a la siguiente expresión

$$\begin{aligned} f_i(x) &= f_i''(x_i) \frac{(x_{i+1} - x)^3}{6(x_{i+1} - x_i)} + f_i''(x_{i+1}) \frac{(x - x_i)^3}{6(x_{i+1} - x_i)} \\ &+ \left[\frac{f(x_i)}{x_{i+1} - x_i} - f_i''(x_i) \frac{(x_{i+1} - x_i)}{6} \right] (x_{i+1} - x) \\ &+ \left[\frac{f(x_{i+1})}{x_{i+1} - x_i} - f_i''(x_{i+1}) \frac{(x_{i+1} - x_i)}{6} \right] (x - x_i) \end{aligned} \quad (\text{C.8})$$

que representa un polinomio cúbico para $f_i(x)$ y en donde sólo se desconocen los valores $f_i''(x_i)$ y $f_i''(x_{i+1})$. Para encontrarlos se puede utilizar la condición de continuidad en la primera derivada que establece que

$$f_i'(x_{i+1}) = f_{i+1}'(x_{i+1})$$

y al derivar (C.9) y sustituir en la Ecuación (C.7) para f_{i+1} y f_i se obtiene

$$\begin{aligned} (x_{i+1} - x_i) f_i''(x_i) + 2(x_{i+2} - x_i) f_i''(x_{i+1}) \\ + (x_{i+2} - x_{i+1}) f_i''(x_{i+2}) &= \frac{6}{x_{i+2} - x_{i+1}} (f(x_{i+2}) - f(x_{i+1})) \\ &+ \frac{6}{x_{i+1} - x_i} (f(x_i) - f(x_{i+1})) \end{aligned}$$

el cual es un sistema de $n-1$ ecuaciones lineales con $n+1$ coeficientes $f_i''(x_i)$. Si definimos $f_i''(x_0) = 0$ y $f_i''(x_n) = 0$ correspondientes a las segundas derivadas de f_i en los extremos, tenemos un sistema de $n-1$ ecuaciones lineales tridiagonal con $n-1$ incógnitas, muchas menos que $n(m+1)$ incógnitas del sistema de ecuaciones anteriormente descrito.

Anexo D

Sistemas reactivos por medio de esquemas

Existe una corriente de investigación sobre búsqueda de ruta, control reactivo y potenciales artificiales en robots, íntimamente relacionada con la psicología y la neurología [16], ésta se basa en el concepto de *esquemas*. Un esquema representa características genéricas de comportamiento reactivo en un agente. Por medio de un esquema específico el agente adquiere un comportamiento reactivo basado directamente en la información de los sensores y del estado interno reducido del agente. Ejemplos de estos esquemas son: *Evade-Obstáculo*, *Mantener-Camino*, *Ir-a-Destino*.

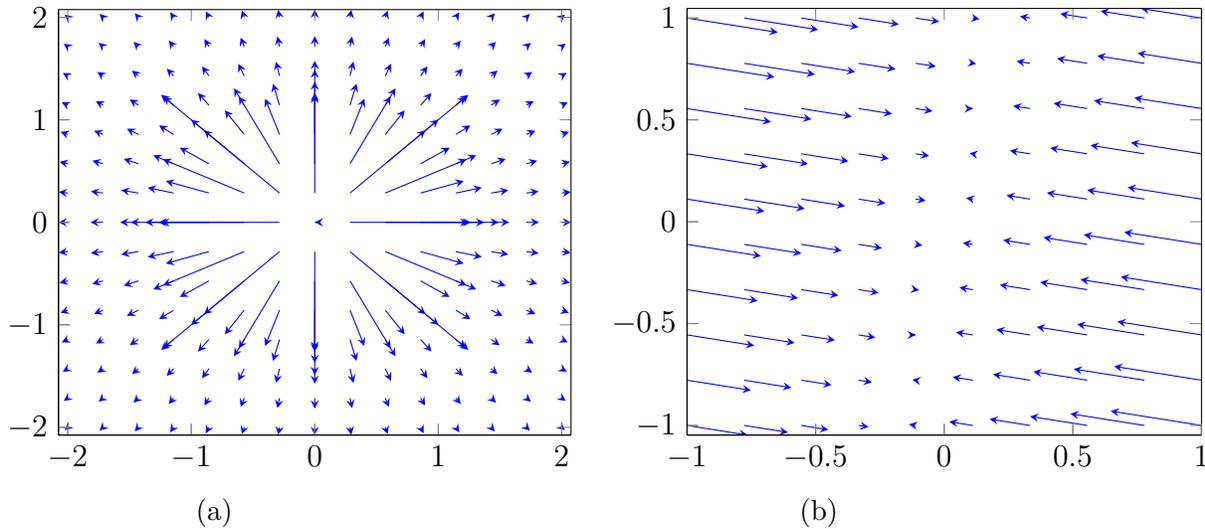


Figura D.1: (a) Fuerza debido al potencial $V_a = \frac{1}{r}$, Esquema: *Evade-Obstáculo*; (b) Fuerza debido al potencial $V_a = -\nabla \frac{(5x-y)^2}{\sqrt{26}}$, Esquema: *Mantener-Camino*

Estos esquemas genéricos poseen una serie de parámetros como son ganancias y localizaciones, que se especifican al momento de ser instanciados. Un agente puede instanciar y desinstanciar esquemas al momento que sean necesarios y su comportamiento es entonces afectado por éste, pudiendo tener más de un esquema instanciado simultáneamente, además

de poder tener más de una instancia de un mismo esquema.

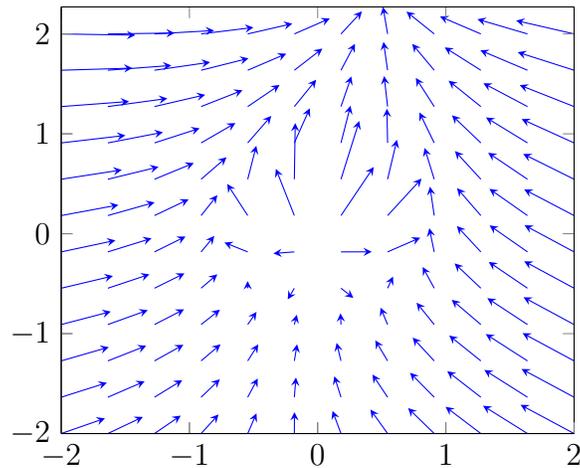


Figura D.2: Fuerza debido a los potenciales de los esquemas: *Mantener-Camino*, *Evade-Obstáculo* e *Ir-a-Destino*

El uso de esquemas es una forma del método divide y vencerás, donde se subdivide el problema en un conjunto de esquemas (comportamientos) sencillos que pueden ejecutarse rápidamente en tiempo real [50], necesarios en el caso de espacios de trabajo dinámicos o con información incompleta, donde no siempre se dispone de suficiente tiempo de replanear completamente la ruta por medio de algoritmos de búsqueda de ruta de alto nivel. Los agentes reactivos por medio de esquemas pueden lidiar con pequeñas variaciones imprevistas en la ruta sin necesidad de replanear la ruta, y en caso, contrario reportarlo al planeador de ruta mientras se evaden colisiones en ese momento. El efecto de algunos esquemas se puede representar con un potencial artificial como en la Figura D.2 donde se presenta un potencial que lleva al vehículo a seguir un camino asociado a *Mantener-Camino*, un potencial repulsivo debido a un obstáculo localizado en $(0, 0)$, asociado a *Evade-Obstáculo*, y un potencial atractivo hacia un destino en dirección del camino en $(0.5, 2.5)$, asociado a *Ir-a-Destino*. La combinación de los efectos de varios esquemas es obtenida mediante suma vectorial en este caso. Al ser instanciado un esquema representado con un potencial artificial, este potencial se parametriza con valores específicos en función del estado interno del agente en ese momento y según criterios pre-establecidos, entrando en efecto sobre el vehículo.

Anexo E

Código fuente

A continuación se presentan los códigos desarrollados junto con este trabajo para la realización de simulaciones y experimentos, se utilizó la jerarquía mostrada en la Figura 2.5. El Código fuente E.1 muestra la implementación del Algoritmo 1 utilizado en el planificador de misión, de Dijkstra, como una función de MATLAB que acepta como argumento una *Matriz de adyacencia*. En una matriz de adyacencia M se tiene como entradas $M_{i,j}$ el peso entre el nodo i y el nodo j (habiendo numerado todos los nodos con anterioridad), el peso es tomado como la distancia euclidiana entre nodos. En esta implementación, si dos nodos no se encuentran conectados, el peso se define como cero. La matriz de adyacencia se calcula previamente formando la gráfica de visibilidad del problema para obstáculos triangulares, que se genera con un algoritmo desarrollado en este trabajo que implementa detección de intersecciones entre los lados de los obstáculos y las líneas que unen nodos. La función regresa como resultado un arreglo `pred` y un arreglo `g` que contienen el predecesor de cada nodo y el costo mínimo para llegar a cada nodo, respectivamente.

Código fuente E.1: Implementación del algoritmo de Dijkstra

```
1 function [ g,pred ] = DijkstraPathfinding( adjM )
2
3     vertexnum=size(adjM,1);
4
5     %inicializar valores g
6     g=zeros(vertexnum,1);
7     for i=2:vertexnum
8         g(i)=1000;
9     end
10    %inicilizar predecesores
11    pred=zeros(vertexnum,1);
12
13    %inicializar arreglo de indices en ABIERTOS
14    openset=zeros(vertexnum,1);
15    openset(1)=1;%abrir primer nodo
16
17    %inicio de bucle
18    for iteration=1:100
19        %encontrar vertice minimo
```

```

20     minVertex=0;
21     %definir primer minimo
22     for j=1:vertexnum
23         if openset(j)==1
24             minVertex=j;
25             break;
26         end
27     end
28     if minVertex==0
29         break; %no hay minimo, ni mas nodos
30     end
31     %
32     for j=1:vertexnum
33         if openset(j)==1
34             if g(j)<g(minVertex)
35                 minVertex=j;%nuevo minimo encontrado
36             end
37         end
38     end
39
40     %actualizar vecinos
41     for j=1:vertexnum
42         if adjM(j,minVertex)>0 %es vecino?
43             if g(j)>g(minVertex)+adjM(j,minVertex)
44                 g(j)=g(minVertex)+adjM(j,minVertex);
45                 pred(j)=minVertex;
46                 openset(j)=1;%agregar a ABIERTOS
47             end
48         end
49     end
50     end
51     openset(minVertex)=0;
52 end
53
54 end

```

Para el piloto con el controlador basado en el potencial artificial se programó una subrutina mostrada en el Código fuente E.2, éste se ejecuta a una razón de 10 veces por segundo en una computadora portátil. Note que las líneas de código 21 y 22 representan las Ecuaciones (2.25) y (2.26). El controlador recibe como argumentos: el valor actual x del vector de estado, la referencia ref de entrada y la iteración actual i . Como salida da las velocidades de los motores que serán enviadas al controlador abordado del robot o hacia el solucionador de ecuaciones diferenciales para el caso de simulación. Debido a que el controlador abordado del robot sólo acepta referencias de velocidad y no torque, se optó por simular en la computadora la parte dinámica de las ecuaciones de estado, correspondientes al renglón 1 y 2 de la Ecuación (2.28) y enviar las velocidades obtenidas de esta simulación al robot. Como solucionador de ecuaciones diferenciales se utilizó la rutina `ode45` integrada en MATLAB, que utiliza el método de integración numérica Dormand-Prince de la familia de Runge-Kutta 4 explícita.

Código fuente E.2: Implementación del piloto

```

1 function [ in1, in2 ] = ControllerPotential2( x ,ref,i)
2
3     %seno y coseno de phi se guardan para evitar recalcularlos
4     cost=cos(x(1,5));
5     sint=sin(x(1,5));
6
7     %evitar acceder a datos mas halla del arreglo de referencia
8     if i+10<length(ref{1})
9         xr=[ref{1}(i+10),ref{2}(i+10)];
10        end
11
12        %obtener vector r
13        r=xr-x(1,3:4);
14
15        %vectores en el marco de referencia B
16        vecx=[cost,sint];
17        vecy=[-sint,cost];
18
19        %calculo de fuerza y torque
20        F=(r)*norm(r);
21        Flin=20*dot(F,vecx)-16*x(1,1);
22        Torque=45*dot(F,vecy)-4.5*x(1,2);
23
24        %limitar el valor absoluto de las fuerzas
25        %para evitar accidentes
26        limit(Flin,1.4);
27        limit(Torque,4);
28
29        %realizar la conversion de torque a velocidad
30        [in11,in21]=Acc2mTorque(ref{9}(i),ref{10}(i));
31        [in1, in2]=motorTorque(Flin,Torque);
32        in1=in1+in11;
33        in2=in2+in21;
34
35 end

```

Además del código anterior, se programaron 7 rutinas más basadas en diferentes potenciales artificiales, siendo la presentada la más satisfactoria. También se implementó el controlador de DeLuca, presentado en la Ecuación (2.32).

Para generar la trayectoria de referencia, se utilizó la función de MATLAB `spline` que obtiene los coeficientes de los polinomios, así mismo se realizó derivación numérica para obtener las velocidades de las Ecuaciones (2.19) y (2.20). En Código fuente E.3 se muestra el navegador implementado.

Código fuente E.3: Implementación del navegador

```

1 t=0:testtime/(length(path)-1):testtime;
2 for i=path
3     waypoints=[transpose(vertex(i,1));transpose(vertex(i,2));t];
4 end
5

```

```
6 tm=0:frametime:testtime;
7
8 %generar los splines para (x,y)
9 splinex=spline(waypoints(3,:),waypoints(1,:));
10 spliney=spline(waypoints(3,:),waypoints(2,:));
11
12 %obtener el arreglo con la trayectoria
13 trayx=ppval(splinex,tm);
14 trayy=ppval(spliney,tm);
15
16 %obtener primeras derivadas de la trayectoria
17 trayx1=zeros(1,length(tm));
18 trayy1=zeros(1,length(tm));
19 trayx1(1,1:length(tm)-1)=diff(trayx)/frametime;
20 trayy1(1,1:length(tm)-1)=diff(trayy)/frametime;
21
22 %obtener segundas derivadas de la trayectoria
23 trayx2=zeros(1,length(tm));
24 trayy2=zeros(1,length(tm));
25 trayx2(1,1:length(tm)-1)=diff(trayx1)/frametime;
26 trayy2(1,1:length(tm)-1)=diff(trayy1)/frametime;
27
28 %obtener velocidades
29 trayv=sqrt(trayx1.^2+trayy1.^2);
30 trayw=(trayy2.*trayx1-trayx2.*trayy1)./trayv;
31
32 %obtener derivadas de las velocidades
33 trayv1=zeros(1,length(tm));
34 trayw1=zeros(1,length(tm));
35 trayv1(1,1:length(tm)-1)=diff(trayv)/frametime;
36 trayw1(1,1:length(tm)-1)=diff(trayw)/frametime;
37
38 trayth=atan2(trayy1,trayx1);
39
40 %estructurar la referencia
41 ref={trayx,trayy,trayx1,trayy1,trayv,trayw,
42     trayx2,trayy2,trayv1,trayw1,trayth};
43
44 graphs={vertex,triangles};
45 hold on
46 plot(trayx,trayy,'-');
```


Lista de símbolos

\mathcal{I}	marco de referencia inercial
\mathcal{B}	marco de referencia en el cuerpo
$\hat{x}, \hat{y}, \hat{z}$	vectores unitarios en dirección de los ejes del marco inercial
$\hat{x}', \hat{y}', \hat{z}'$	vectores unitarios en dirección de los ejes del marco de referencia en el cuerpo
Ω	vector velocidad angular del marco de referencia en el cuerpo
\mathbf{Q}	vector arbitrario
\mathbf{V}_P	vector velocidad de un punto P
\mathbf{V}_O	vector velocidad del origen del marco de referencia
$\mathbf{V}_{P/O}$	vector velocidad del punto P con respecto al origen
$\mathbf{r}_{P/O}$	vector posición del punto P con respecto al origen
$\mathbf{R}(\phi, \theta, \psi)$	matriz ortogonal de rotación
$\mathbf{R}_x(\phi), \mathbf{R}_y(\phi), \mathbf{R}_z(\phi)$	matrices de rotación elementales en los ejes x, y y z respectivamente
$c., s., t.$	funciones trigonométricas $\cos(\cdot), \sin(\cdot)$ y $\tan(\cdot)$ respectivamente
\mathbf{q}	vector arbitrario
$\begin{bmatrix} \mathbf{q} \end{bmatrix}$	matriz columna con componentes de \mathbf{q} en el marco inercial
$\begin{bmatrix} \mathbf{q} \end{bmatrix}'$	matriz columna con componentes de \mathbf{q} en el marco de referencia en el cuerpo
ϕ, θ, ψ	ángulos de Euler, ángulos de Tayt bryan
$\boldsymbol{\eta}$	matriz columna con los ángulos de Euler o Tayt bryan como componentes
$\mathfrak{i}, \mathfrak{j}, \mathfrak{k}$	cuaterniones basicos unitarios
$\mathfrak{h}, \mathfrak{q}, \mathfrak{u}$	cuaterniones arbitrarios
q_x, q_y, q_z	componentes x, y y z de un vector \mathbf{q} o un cuaternión \mathfrak{q}
q_w	componente escalar de un cuaternión \mathfrak{q}
ω	cuaternión puro con componentes del vector velocidad angular $\boldsymbol{\omega}$
$\begin{bmatrix} \hat{\mathbf{q}} \end{bmatrix}$	matriz de producto cruz de la matriz columna $\begin{bmatrix} \mathbf{q} \end{bmatrix}$
$Q_R(\mathfrak{q})$	representación matricial para la multiplicación del cuaternión \mathfrak{q} por la derecha
$Q_L(\mathfrak{q})$	representación matricial para la multiplicación del cuaternión \mathfrak{q} por la izquierda
\mathbf{P}	momento lineal de un cuerpo
\mathbf{L}	momento angular de un cuerpo
\mathbf{v}_G	vector velocidad del centro de masa G
v_G	velocidad escalar del centro de masa G
\mathbf{F}	fuerza aplicada

\mathbf{N}	torque aplicado
$\hat{\mathbf{n}}$	vector unitario normal a un plano
\mathbf{I}	tensor momento inercial
L, L_i	Lagrangiano, polinomio interpolador de Lagrange
T	energía cinética, empuje total
V, V_i	energía potencial, potencial artificial i
g, \mathbf{g}	magnitud de la fuerza de gravedad y vector fuerza de gravedad, respectivamente
L_{rot}	parte rotacional del Lagrangiano
L_{trans}	parte translacional del Lagrangiano
$g(\cdot)$	ecuación de restricción, función de peso entre nodos
λ	multiplicador de Lagrange
W	Trabajo efectuado
δW	variación de W
ω_i	velocidad angular de la rueda o rotor numero i
\mathcal{P}_i	conjunto de puntos del obstáculo numero i
$\mathcal{R}(x, y)$	conjunto de puntos que ocupa el robot, dependiente de x y y
$\mathcal{S}_o, \mathcal{S}_f$	puntos de inicio y de destino para el robot
$h(\cdot)$	función heurística para el algoritmo de búsqueda
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$	matrices de estado, de entrada, de salida y de transmisión directa, respectivamente, para el método de espacio de estados
$\mathbf{x}, \mathbf{u}, \mathbf{y}$	matriz columna de estados, de entrada y de salida, respectivamente
\mathbf{K}	matriz de realimentación

Bibliografía

- [1] “Vehículo autónomo,” Jul. 2016, page Version ID: 92399382. [Online]. Available: https://es.wikipedia.org/w/index.php?title=Veh%C3%ADculo_aut%C3%B3nomo&oldid=92399382
- [2] “History of AGVS,” Aug. 2012. [Online]. Available: <http://www.agvsystems.com/history-agvs/>
- [3] “Vehículo de guiado automático,” Jun. 2016, page Version ID: 91966755. [Online]. Available: https://es.wikipedia.org/w/index.php?title=Veh%C3%ADculo_de_guiado_autom%C3%A1tico&oldid=91966755
- [4] J.-M. Yang and J.-H. Kim, “Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots,” *IEEE Transactions on robotics and automation*, vol. 15, no. 3, pp. 578–587, 1999.
- [5] D. Chwa, “Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates,” *IEEE transactions on control systems technology*, vol. 12, no. 4, pp. 637–644, 2004.
- [6] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot: backstepping kinematics into dynamics,” in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 4. IEEE, 1995, pp. 3805–3810.
- [7] D. Chwa, “Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 6, pp. 1285–1295, 2010.
- [8] Z.-G. Hou, A.-M. Zou, L. Cheng, and M. Tan, “Adaptive control of an electrically driven nonholonomic mobile robot via backstepping and fuzzy approach,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 803–815, 2009.
- [9] L. Astudillo, O. Castillo, P. Melin, A. Alanis, J. Soria, and L. T. Aguilar, “Intelligent control of an autonomous mobile robot using type-2 fuzzy logic.” *Engineering Letters*, vol. 13, no. 3, 2006.
- [10] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot using neural networks,” *IEEE transactions on neural networks*, vol. 9, no. 4, pp. 589–600, 1998.

- [11] J. Ye, “Tracking control for nonholonomic mobile robots: Integrating the analog neural network into the backstepping technique,” *Neurocomputing*, vol. 71, no. 16, pp. 3373–3378, 2008.
- [12] S. S. Ge and Y. J. Cui, “New potential functions for mobile robot path planning,” *IEEE Transactions on robotics and automation*, vol. 16, no. 5, pp. 615–620, 2000.
- [13] F. Lamiroux, D. Bonnafous, and O. Lefebvre, “Reactive path deformation for nonholonomic mobile robots,” *IEEE transactions on robotics*, vol. 20, no. 6, pp. 967–977, 2004.
- [14] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on robotics and automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [15] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.
- [16] R. C. Arkin, “Motor schema—based mobile robot navigation,” *The International journal of robotics research*, vol. 8, no. 4, pp. 92–112, 1989.
- [17] J. Hagelbäck and S. J. Johansson, “The rise of potential fields in real time strategy bots.” *AIIDE*, vol. 8, pp. 42–47, 2008.
- [18] —, “A multiagent potential field-based bot for real-time strategy games,” *International Journal of Computer Games Technology*, vol. 2009, p. 4, 2009.
- [19] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [20] A. Stentz, “Optimal and efficient path planning for partially-known environments,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*. IEEE, 1994, pp. 3310–3317.
- [21] K. Daniel, A. Nash, S. Koenig, and A. Felner, “Theta*: Any-angle path planning on grids,” *Journal of Artificial Intelligence Research*, vol. 39, pp. 533–579, 2010.
- [22] S. Koenig and M. Likhachev, “Improved fast replanning for robot navigation in unknown terrain,” in *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 968–975.
- [23] D. Ferguson, M. Likhachev, and A. Stentz, “A guide to heuristic-based path planning,” in *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS)*, 2005, pp. 9–18.
- [24] P. J. C. Ferdinand P. Beer, E. Russell Johnston Jr, *Mecanica Vectorial para ingenieros Dinamica*, 10th ed. Mc Graw Hill, 2013, vol. 2.
- [25] J. B. M. Stephen T. Thornton, *Classical Dynamics of Particles and Systems*, 1st ed. Brooks/Cole, 2003.

- [26] R. Dorf, R. Bishop, S. Canto, and R. Canto, *Sistemas de control moderno*, ser. Fuera de colección Out of series. Pearson Educación, 2005. [Online]. Available: <https://books.google.com.mx/books?id=NPRGAAAACAAJ>
- [27] B. Kuo, *Sistemas de control automático*. Prentice Hall Hispanoamericana, 1996. [Online]. Available: <https://books.google.com.mx/books?id=GyWr6cT8SEsC>
- [28] G. F. Simmons and S. G. Krantz, *Ecuaciones diferenciales: teoría, técnica y práctica*. McGraw-Hill, 2007.
- [29] J. Craig, *Robótica*. Pearson Educación de México, 2006. [Online]. Available: https://books.google.com.mx/books?id=hRzOp_qdxG8C
- [30] T. Balch, “Avoiding the past: A simple but effective strategy for reactive navigation,” in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. IEEE, 1993, pp. 678–685.
- [31] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [32] J. S. Herbert Goldstein, Charles Poole, *Classical Mechanics*, 3rd ed. Addison Wesley, 1992.
- [33] E. Gutiérrez-Arias, J. Flores-Mena, M. Morin-Castillo, and H. Suárez-Ramírez, “Design of an optimal control for an autonomous mobile robot,” *Revista mexicana de física*, vol. 57, no. 1, pp. 75–83, 2011.
- [34] G. Oriolo, A. De Luca, and M. Vendittelli, “Wmr control via dynamic feedback linearization: design, implementation, and experimental validation,” *IEEE Transactions on control systems technology*, vol. 10, no. 6, pp. 835–852, 2002.
- [35] E. Kreyszig, *Differential Geometry*, 1st ed. Dover Publications, 1991.
- [36] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.
- [37] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer Berlin Heidelberg, 2008. [Online]. Available: <https://books.google.com.mx/books?id=9nhHUZpKzeEC>
- [38] X. Chen and S. McMains, “Polygon offsetting by computing winding numbers,” in *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2005, pp. 565–575.
- [39] S. L. Altmann, *Rotations, quaternions, and double groups*. Courier Corporation, 2005.
- [40] G. Cai, B. Chen, and T. Lee, *Unmanned Rotorcraft Systems*, ser. Advances in Industrial Control. Springer London, 2011. [Online]. Available: <https://books.google.com.mx/books?id=RJYtO51KZ6kC>

- [41] J. Kuipers, “Quaternions and rotation sequences,” *Geometry, Integrability & Quantization*, vol. 1, pp. 127–143, 2000.
- [42] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [43] M. Agoston, *Computer Graphics and Geometric Modelling: Mathematics*, ser. Computer Graphics and Geometric Modeling. Springer, 2005. [Online]. Available: <https://books.google.com.mx/books?id=LPsAM-xuGG8C>
- [44] J. Gallier, *Geometric methods and applications: for computer science and engineering*. Springer Science & Business Media, 2011, vol. 38.
- [45] P. Betsch and R. Siebert, “Rigid body dynamics in terms of quaternions: Hamiltonian formulation and conserving numerical integration,” *International Journal for Numerical Methods in Engineering*, vol. 79, no. 4, pp. 444–473, 2009. [Online]. Available: <http://dx.doi.org/10.1002/nme.2586>
- [46] J. P. Morais, S. Georgiev, and W. Sprößig, *Real quaternionic calculus handbook*. Springer, 2014.
- [47] W. B. Heard, *Rigid body mechanics: mathematics, physics and applications*. John Wiley & Sons, 2008.
- [48] J. C. Chou, “Quaternion kinematic and dynamic differential equations,” *IEEE Transactions on robotics and automation*, vol. 8, no. 1, pp. 53–64, 1992.
- [49] S. Chapra and R. Canale, *Numerical Methods for Engineers*. McGraw-Hill Education, 2009. [Online]. Available: <https://books.google.com.mx/books?id=SA1LPgAACAAJ>
- [50] R. J. Clark, R. C. Arkin, and A. Ram, “Learning momentum: online performance enhancement for reactive systems,” in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE, 1992, pp. 111–116.