



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

“PLANIFICACIÓN Y SEGUIMIENTO DE CAMINOS DE MANERA AUTÓNOMA PARA ROBOTS MÓVILES TIPO UNICICLO”

LEONARDO MANUEL ASQUI PAGUAY

Trabajo de Titulación modalidad: Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

**MAGISTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN
INDUSTRIAL**

RIOBAMBA-ECUADOR

Noviembre 2017

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, denominado: "PLANIFICACIÓN Y SEGUIMIENTO DE CAMINOS DE MANERA AUTÓNOMA PARA ROBOTS MÓVILES TIPO UNICICLO", de responsabilidad del señor Leonardo Manuel Asqui Paguay, ha sido minuciosamente revisado y se autoriza su presentación.

Ing. Wilson Armando Zuñiga Vinueza; M. Sc.

PRESIDENTE

Ing. Víctor Hugo Andaluz Ortiz; Ph.D.

DIRECTOR DE TESIS

Ing. Jorge Saúl Sánchez Mosquera; M. Sc.

MIEMBRO DEL TRIBUNAL

Ing. Julio Francisco Acosta Núñez; M. Sc.

MIEMBRO DEL TRIBUNAL

Riobamba, Noviembre de 2017

DERECHOS INTELECTUALES

Yo, Leonardo Manuel Asqui Paguay soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica de Chimborazo.

LEONARDO MANUEL ASQUI PAGUAY

N° Cédula: 060407864-2

DERECHOS DE AUTENTICIDAD

Yo, Leonardo Manuel Asqui Paguay, declaro que el presente proyecto de investigación, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

LEONARDO MANUEL ASQUI PAGUAY

N° Cédula: 060407864-2

DEDICATORIA

A mis padres, hermanos, a mi novia, profesores y a todos mis amigos quienes siempre estuvieron presentes, durante todo este tiempo; contribuyendo y apoyándome incondicionalmente.

AGRADECIMIENTO

A Dios; a mis padres, principalmente a mi madre, a mis hermanos, a mi querida novia, a mi tutor de tesis, y a todos mis amigos quienes siempre me motivan a alcanzar nuevas metas.

Leonardo

CONTENIDO

RESUMEN.....	xi
SUMMARY	xii

CAPÍTULO I

1.	INTRODUCCIÓN	1
1.1.	Planteamiento del problema.....	1
1.2.	Formulación del problema	2
1.3.	Preguntas directrices o específicas	2
1.4.	Justificación de la investigación.....	3
1.5.	Objetivos	3
1.5.1.	<i>Objetivo general</i>	3
1.5.2.	<i>Objetivos específicos</i>	4
1.6.	Hipótesis.....	4
1.6.1.	<i>Hipótesis general</i>	4
1.6.2.	<i>Hipótesis específicas</i>	4

CAPÍTULO II

2.	MARCO DE REFERENCIA	5
2.1.	Robótica móvil.....	5
2.1.1.	<i>Tipos de sistemas de locomoción</i>	6
2.1.2.	<i>Locomoción con ruedas</i>	6
2.1.3.	<i>Locomoción con patas</i>	10
2.2.	Planificación de caminos	10
2.2.1.	<i>Grafos de visibilidad</i>	11
2.2.2.	<i>Diagramas de Voronoi</i>	12
2.2.3.	<i>Descomposición de celdas</i>	13
2.2.4.	<i>Mapas probabilísticos (PRMs)</i>	16
2.2.5.	<i>Exploración rápida de árboles al azar (Rapidly Exploring Random tree)</i>	17
2.3.	Post-procesado.....	20

CAPÍTULO III

3.	PLANIFICACIÓN Y SEGUIMIENTO DE CAMINOS	22
3.1.	Modelo Cinemático Robot uniclo.....	22
3.2.	Modelo dinámico	25

3.2.1.	<i>Propiedades de los modelos Dinámicos</i>	28
3.3.	Validación del modelo Cinemático y Dinámico	30
3.4.	Algoritmo de Planificación de caminos	33
3.4.1.	<i>Construcción del ambiente estructurado</i>	33
3.4.2.	<i>Algoritmo de Planificación de caminos</i>	35
3.4.3.	<i>Consideraciones del tamaño del robot unicycle</i>	40
3.4.4.	<i>Post-Procesado de caminos</i>	42
3.5.	Control del robot móvil tipo unicycle	45
3.5.1.	<i>Diseño del sistema de control de robot móvil</i>	47
3.5.2.	<i>Simulación caso 1: Velocidad deseada sobre camino</i>	51
3.5.3.	<i>Simulación caso 2: Velocidad deseada en todo instante</i>	53
3.5.4.	<i>Simulación caso 3: Velocidad deseada en función del error ρ</i>	55
 CAPÍTULO IV		
4.	RESULTADOS Y DISCUSIÓN	58
4.1.	Resultados de Simulación	59
4.1.1.	<i>Simulación 1</i>	59
4.1.2.	<i>Simulación 2</i>	62
4.2.	Resultados experimentales	65
 CONCLUSIONES		69
RECOMENDACIONES		70
BIBLIOGRAFÍA		
ANEXOS		

LISTA DE FIGURAS

Figura 1-2:	Disposición de ruedas en un Robot Móvil	6
Figura 2-2:	Robot móvil Uranus	7
Figura 3-2:	Robot unicycle Pioneer P3-DX.....	7
Figura 4-2:	Robot móvil Neptune.	8
Figura 5-2:	Sistema Ackerman.....	9
Figura 6-2:	Locomoción por pistas de deslizamiento.	9
Figura 7-2:	Robot cuadrúpedo.....	10
Figura 8-2:	Grafo de visibilidad.....	11
Figura 9-2:	Diagrama de Voronoi generalizado.....	12
Figura 10-2:	Descomposición vertical (Celdas exactas).	14
Figura 11-2:	Descomposición de Delaunay.	15
Figura 12-2:	Descomposición quadtree.....	16
Figura 13-2:	PRM de múltiple consulta.	17
Figura 14-2:	RRT básico.....	18
Figura 15-2:	Algoritmo RRT básico.	19
Figura 16-2:	Función Extiende.....	19
Figura 17-2:	Esquema de expansión del RRT.....	20
Figura 18-2:	Post-procesado.....	21
Figura 1-3:	Geometría uno del robot móvil unicycle.....	22
Figura 2-3:	Geometría dos del robot móvil unicycle	24
Figura 3-3:	Parámetros del robot móvil tipo unicycle.	25
Figura 4-3:	Robot móvil tipo unicycle.....	29
Figura 5-3:	Validación 1: Camino del modelo cinemático y dinámico.....	31
Figura 6-3:	Validación 1: Velocidad lineal del modelo cinemático y dinámico.	31
Figura 7-3:	Validación 2: Camino del modelo cinemático y dinámico.....	32
Figura 8-3:	Validación 2: Velocidad lineal del modelo cinemático y dinámico.	32
Figura 9-3:	Ambiente con obstáculos del tipo círculos.....	34
Figura 10-3:	Ambiente con obstáculos cuadrados	34
Figura 11-3:	Algoritmo RRT.....	35
Figura 12-3:	Función Extiende.....	36
Figura 13-3:	Punto randómico dentro de obstáculo.	36
Figura 14-3:	Punto randómico fuera de obstáculos.....	37
Figura 15-3:	Trayectoria qnew-gradn en colisión.	38
Figura 16-3:	qrand igual a qnew	38
Figura 17-3:	Función conecta.....	39
Figura 18-3:	qfin cerca de qnew.....	40
Figura 19-3:	Camino cerca de obstáculos.	41
Figura 20-3:	Camino lejos de obstáculos.	41
Figura 21-3:	Algoritmo de post-procesado.	42
Figura 22-3:	Distancias del camino planificado.....	43
Figura 23-3:	Puntos g1 y g2.	44
Figura 24-3:	Cuerda valida.....	44
Figura 25-3:	Camino suavizado.	45
Figura 26-3:	Variables para el seguimiento de caminos.	46
Figura 27-3:	Controlador caso 1: Movimiento estroboscópico.....	51
Figura 28-3:	Controlador Caso 1: Error de control.	52
Figura 29-3:	Controlador Caso1: Velocidades de referencia.	52
Figura 30-3:	Controlador Caso 1: Velocidad real y de referencia.....	53
Figura 31-3:	Controlador caso 2: Movimiento estroboscópico.....	53

Figura 32-3:	Controlador Caso 2: Error de control	54
Figura 33-3:	Controlador Caso 2: Velocidades de referencia	54
Figura 34-3:	Controlador Caso 2: Velocidad real y de referencia.....	55
Figura 35-3:	Controlador caso 3: Movimiento estroboscópico.....	56
Figura 36-3:	Controlador Caso 3: Error de control	56
Figura 37-3:	Controlador Caso 3: Velocidades de referencia	57
Figura 38-3:	Velocidad real y de referencia: Controlador Caso 3.....	57
Figura 1-4:	Robot móvil tipo unicycle.....	58
Figura 2-4:	Simulación 1: Resultado del algoritmo RRT.....	59
Figura 3-4:	Simulación 1: Numero de nodos del algoritmo RRT	60
Figura 4-4:	Simulación 1: Camino realizado por el robot móvil.....	60
Figura 5-4:	Simulación 1: Error de control	61
Figura 6-4:	Simulación 1: Velocidades de referencia (acciones de control).....	61
Figura 7-4:	Simulación 1: Velocidad lineal deseada vs velocidad real.....	62
Figura 8-4:	Simulación 2: Resultado del algoritmo RRT.....	62
Figura 9-4:	Simulación 2: Numero de nodos del algoritmo RRT.	63
Figura 10-4:	Simulación 2: Camino realizado por el robot móvil.....	63
Figura 11-4:	Simulación 2: Error de control	64
Figura 12-4:	Simulación 2: Velocidades de referencia (acciones de control).....	64
Figura 13-4:	Simulación 2: Velocidad lineal deseada vs velocidad real.....	65
Figura 14-4:	Resultado del algoritmo RRT.....	66
Figura 15-4:	Numero de nodos del algoritmo RRT.	66
Figura 16-4:	Camino realizado por el robot móvil.....	66
Figura 17-4:	Error de control.	67
Figura 18-4:	Velocidades de referencia (acciones de control)	68
Figura 19-4:	Velocidad lineal de referencia versus velocidad real	68

RESUMEN

El objetivo fue la simulación, implementación de la planificación y seguimiento de caminos de manera autónoma para robots móviles tipo unicycle, considerando un ambiente de trabajo estructurado; para la simulación se usó la herramienta de software matemático Matlab, mientras que las pruebas experimentales se realizó con el robot móvil construido en el proyecto de investigación “Tele-operación bilateral cooperativo de múltiples manipuladores móviles”, ejecutado por la Escuela Superior Politécnica de Chimborazo, Universidad Técnica de Ambato, Escuela Politécnica Nacional y la Universidad de las Fuerzas Armadas. Como punto de partida se especifica las características cinemáticas y dinámicas de un robot móvil tipo unicycle, luego se proponer un algoritmo de planificación de caminos basado en algoritmos de exploración rápida de árboles randomicos (RRT) para un ambiente estructurado, luego de realizar la planificación de caminos se implementa un esquema de control para el seguimiento de manera autónoma. Como punto final se obtiene caminos sin colisiones desde una configuración inicial hasta una configuración final, con errores de seguimiento menores a 0,02 [m]; con esto se concluye que con el uso del algoritmo RRT se crea un árbol de configuraciones, explorando de forma randomica todo el ambiente de trabajo hasta encontrar una solución o hasta un número determinado de iteraciones; es importante considerar el radio del robot móvil en la exploración para obtener una solución lo suficientemente alejadas de los obstáculos y evitar colisiones. Se recomiendo estudiar los resultados y material obtenido en este trabajo de titulación para su uso en otros proyectos futuros en las que el ambiente de trabajo sea parcialmente estructurado, y por medio de sensores de proximidad posicionar los obstáculos en el ambiente de estudio, creando una búsqueda reactiva del camino.

Palabras Claves:

<TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <ROBÓTICA>, <ROBOT MÓVIL>, <UNICICLO>, <AMBIENTE ESTRUCTURADO>, <PLANIFICACIÓN DE CAMINOS>, <EXPLORACIÓN RÁPIDA DE ÁRBOLES RANDOMICOS (RRT)> <SEGUIMIENTO DE CAMINOS>.

SUMMARY

The objective was the simulation, implementation of planning and path tracking in an autonomous way for unicycle-type mobile robots, considering a structured work environment. For the simulation the Matlab mathematical software tool was used, while the experimental tests were performed with the mobile robot built in the research project "Cooperative bilateral tele-operation of multiple mobile manipulators", implemented by the Escuela Superior Politécnica de Chimborazo, Universidad Técnica de Ambato, Escuela Politécnica Nacional and the Universidad de las Fuerzas Armadas. The kinematic and dynamic characteristics of a unicycle-type mobile robot, are specified as a starting point, then it is proposed a path planning algorithm based on fast scan algorithms of random trees(RRT) for a structured environment, after carrying out the path planning, a control scheme is implemented to monitor it autonomously. As a final point paths without collisions are obtained from an initial configuration to a final configuration, with tracking errors less than 0.02 [m]. It is concluded that with the use of the RRT algorithm a configurations tree is created, randomly exploring the entire work environment until finding a solution or even a certain number of iterations; it is important to consider the mobile robot's radius in the exploration to obtain a solution far enough away from the obstacles and avoid collisions. It is recommended to study the results and material obtained in this degree work for use in other future projects in which the work environment is partially structured, and by means of the proximity sensors position the obstacles in the study environment, creating a reactive search of the path.

Key words:

<TECHNOLOGY AND ENGINEERING SCIENCES, <ROBOTICS>, <MOBILE ROBOT>, <UNICYCLE>, <STRUCTURED ENVIRONMENT>, <PATH PLANNING>, <RAPID EXPLORATION OF RANDOM TREES (RRT) >, <PATH TRACKING>.

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Planteamiento del problema.

Los robots móviles a menudo se encuentran en situaciones en las que tienen que trasladarse de un punto a otro enfrentando los inconvenientes del entorno como son los obstáculos y las limitaciones de su capacidad. (Bruce & Veloso, 2002)

A menudo las técnicas de control de movimiento requieren altos costes computacionales y múltiples errores numéricos que los hacen inadecuados para aplicaciones en tiempo real. (Frazzoli, Dahleh, & Feron, 2005). Cuando se encuentran en ambientes con obstáculos en movimiento una selección reactiva de la trayectoria es una solución, para que el robot evite los obstáculos y consiga llegar a su meta. (Bruce & Veloso, 2002)

Los métodos de planificación de caminos continuamente se están desarrollando para problemas cada vez más desafiantes y con menores costes computacionales, apuntando todas las investigaciones a lograr robots móviles autónomos, este objetivo no se ha alcanzado solamente con el uso de métodos de planificación de caminos si no también con el uso de métodos reactivos que permiten al robot interactuar con el medio de trabajo. (Nieto, Slawinski, Mut, & Wagner, 2010).

El básico problema en la planificación de caminos de un robot móvil con ruedas está enfocado en encontrar el camino de un punto inicial a un punto final evadiendo los obstáculos que se presenten en el ambiente de trabajo, el enfoque está dado para un sistema de control en lazo cerrado para ser utilizado con el algoritmo RRT (rapidly-exploring random trees), este método puede ser una solución que resuelve planificación de caminos en un ambiente con una dificultad alta y con un tiempo computacional razonable (Palmieri & Arras, 2014).

Tomando en consideración lo mencionado en los párrafos anteriores y teniendo en cuenta el plan nacional de Buen Vivir vigente objetivo cuarto “*Fortalecer las capacidades y potencialidades de la ciudadanía*” (Secretaría Nacional de Planificación y Desarrollo, 2013), que nos indica como punto de interés para el buen cumplimiento de este objetivo la inversión en la ciencia, tecnología e innovación; bajo este contexto se plantea el desarrollo de un algoritmo de planificación de caminos en ambientes estructurados y un esquema de control para el seguimiento del camino planificado con un robot móvil tipo unicycle.

La propuesta contempla un algoritmo para la planificación de caminos desde un punto de partida hacia un punto de llegada en un ambiente estructurado; además de un esquema de control autónomo para el seguimiento del camino.

Como puntos principales el proyecto contempla: 1) definir las características cinemáticas y dinámicas del robot unicycle; 2) desarrollar el algoritmo para la planificación de caminos en ambientes estructurados; 3) definir el esquema de control para el movimiento del robot tipo unicycle tomando en cuenta las características cinemáticas del robot.

1.2. Formulación del problema

¿Cómo desarrollar la planificación de caminos en ambientes estructurados y controlar su seguimiento con un robot móvil tipo unicycle?

1.3. Preguntas directrices o específicas

¿Cómo especificar el modelo cinemático y dinámico de un robot móvil tipo unicycle?

¿Cómo resolver el problema de la planificación de caminos en ambientes estructurados para un robot móvil tipo unicycle?

¿Cómo controlar el seguimiento de caminos a través de la velocidad lineal y velocidad angular del robot móvil tipo unicycle?

1.4. Justificación de la investigación

El desarrollo de las fuerzas productivas se centra en la formación de talento humano para la creación del conocimiento. La formación de conocimiento usando nuevas técnicas de control ayuda a mejorar la tecnología para un desempeño óptimo de los procesos de la vida cotidiana y de la industria (Secretaría Nacional de Planificación y Desarrollo, 2013).

Una buena estrategia para cubrir las necesidades del país, es la gestión del conocimiento común como un bien público, para generar nuevas ideas fomentando la creatividad y el desarrollo de nuevas técnicas, a fin de mejorar la tecnología y usarlas para incrementar valor agregado en los productos elaborados (Secretaría Nacional de Planificación y Desarrollo, 2013)

Enmarcándose según la planificación estratégica del Plan Nacional del Buen Vivir en su objetivo 10 *“Impulsar la transformación de la matriz productiva”*, (Secretaría Nacional de Planificación y Desarrollo, 2013), en referencia al lineamiento 10.2 que expresa *“Promover la intensidad tecnológica en la producción primaria, de bienes intermedios y finales”*, y específicamente en el literal a) que expresa *“Articular la investigación científica, tecnológica y la educación superior con el sector productivo, para una mejora constante de la productividad y competitividad sistémica, en el marco de las necesidades actuales y futuras del sector productivo y el desarrollo de nuevos conocimientos”*. tomando en consideración este objetivo se plantea un algoritmo para la planificación de caminos en ambientes estructurados y un esquema de control para el seguimiento del camino planificado; considerando las características cinemáticas de un robot móvil tipo unicycle, fomentando la creación de nuevas tecnologías con ideas innovadoras y creativas que se podrán usar para la solución de varios problemas industriales, problemas de la vida cotidiana, generando valor agregado a los productos finales de la industria, y dando mayor comodidad de vida a las personas mediante el uso de estas nuevas tecnologías.

1.5. Objetivos

1.5.1. Objetivo general

Implementar un algoritmo para la planificación de caminos en ambientes estructurados y un esquema de control autónomo para el seguimiento del camino planificado, considerando las restricciones de los movimientos de un robot móvil tipo unicycle.

1.5.2. Objetivos específicos

- Especificar las características cinemáticas y dinámicas de un robot móvil tipo unicycle, afín de proponer un algoritmo de planificación y seguimiento de caminos.
- Proponer un algoritmo para la planificación de caminos basado en algoritmos de exploración rápida de árboles randomicos, RRT, para ambientes de trabajo estructurados, afín de ser seguido por un robot móvil tipo unicycle.
- Proponer un esquema de control para seguimiento de caminos de manera autónoma. Asimismo, garantizar analíticamente la estabilidad del algoritmo de control propuesto.
- Evaluar experimentalmente el esquema de planificación y seguimiento de caminos propuesto en un ambiente estructurado a través de un robot móvil tipo unicycle

Nota: Las pruebas experimentales se realizarán con el robot móvil construido en el proyecto de investigación “Tele-operación bilateral cooperativo de múltiples manipuladores móviles”, proyecto ganador de la IX Convocatoria CEDIA-CEPRA 2015. El proyecto es ejecutado por la Escuela Superior Politécnica de Chimborazo, Universidad Técnica de Ambato, Escuela Politécnica Nacional y la Universidad de las Fuerzas Armadas.

1.6. Hipótesis

1.6.1. Hipótesis general.

Se puede planificar y controlar el seguimiento de caminos en ambientes estructurados para un robot móvil tipo unicycle.

1.6.2. Hipótesis específicas.

Se puede definir las características cinemáticas y dinámicas de un robot móvil tipo unicycle

Se puede proponer un algoritmo para la planificación de caminos libre de colisiones desde un punto de partida hasta un punto de llegada en ambientes estructurados.

Utilizando el modelo del robot tipo unicycle y el camino planificado se puede plantear un algoritmo de control de seguimiento de caminos de manera autónoma.

CAPÍTULO II

2. MARCO DE REFERENCIA

2.1. Robótica móvil

El desarrollo de los robots móviles se basa en un principio en la necesidad de extender el campo de aplicación de la robótica en la industria, ya que en un principio se limitaba a un robot fijo a una base (Ollero, 2001), este problema surgió con la nueva concepción de fabricación flexible que involucra el traslado de los materiales entre puntos distintos de la fábrica (Domingo Esteve, 2001), las soluciones a este problema son varios, como situar a las maquinas dentro de un área, cerca una a las otras para que un brazo robótico traslade los materiales entre ellas, esta configuración es apropiada para un número limitado de máquinas, y resulta imposible a medida que la célula sufra ampliaciones significativas, una solución a este problema es la utilización de vehículos sobre rieles (Secchi, 2008) o la utilización de cables enterrados que sirven como guías creando un campo magnético, este último dando un ahorro significativo a la fábrica.

La gran inversión de las industrias y la posibilidad de estructurar su entorno para la navegación de los robots, hacen que estos dispongan de una capacidad sensorial y razonamiento mínimo, esta alta inversión realizada por la industria en una determinada aplicación es imposible de realizar en aplicaciones fuera de ella y la imposibilidad de estructurar el ambiente de trabajo dan como resultado la búsqueda de un robot móvil para actuar dentro de un entorno no estructurado mediante información suministrada por sensores, con la ayuda de la comunidad científica, el desarrollo de la mecánica, sistemas de control, sensores y algoritmos de planificación de caminos, dan paso a un nuevo concepto de robot móvil, capaces de trasladarse a cualquier punto dentro del espacio de trabajo por medio de una planificación reactiva del camino y un sistema de control estable.

2.1.1. Tipos de sistemas de locomoción

Un robot móvil tiene varias formas para moverse sobre suelos sólidos, esta es una de las primeras características que está en dependencia de su entorno, entre los sistemas de locomoción más comunes tenemos: sobre ruedas, patas y cadenas. Siendo el primero uno de los más utilizados debido a su fácil implementación y gran capacidad de carga.

2.1.2. Locomoción con ruedas.

Este tipo de robot móvil es uno de los más simples, en donde su campo de aplicación es extenso, dando lugar a un gran número de variantes, con distintas características de movilidad. Entre las limitaciones de este tipo de robot son los deslizamientos y vibraciones, provocados por terrenos blandos e irregulares, entre este tipo de robot tenemos:

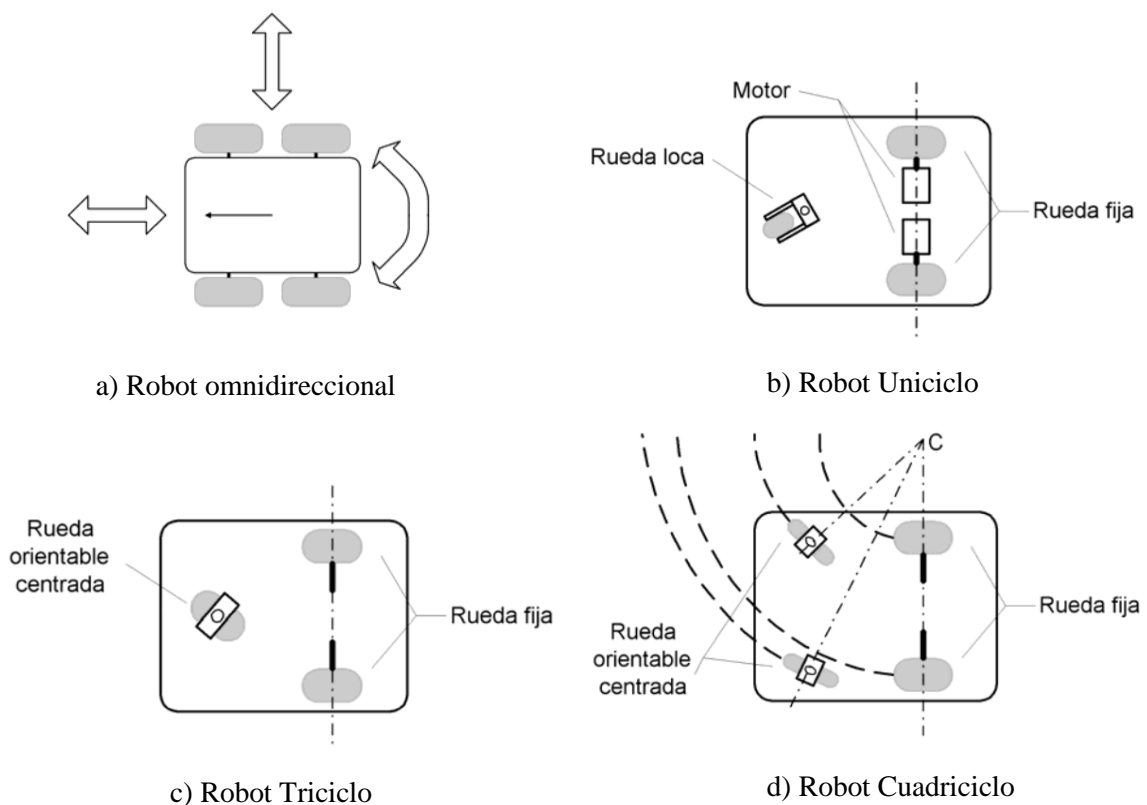


Figura 1-2: Disposición de ruedas en un Robot Móvil

Fuente: Secchi, H. A. Una introducción a los robots móviles, 2008.

2.1.2.1. Robot omnidireccional

Este tipo de robot posee máxima capacidad de movilidad sin la necesidad de reorientarse, a diferencia de otros tipos de robot que tiene una maniobrabilidad restringida, ya que si desean cambiar de dirección también tienen que cambiar su orientación.

La Figura 2-2 muestra el robot omnidireccional Uranus desarrollado en la Universidad de Michigan, este robot tiene la capacidad de desplazarse lateralmente de acuerdo a la rotación de cada una de sus ruedas.



Figura 2-2: Robot móvil Uranus

Fuente: <https://www.cs.cmu.edu/~gwp/robots/Uranus.html>

2.1.2.2. Robot móvil unicycle.

Este tipo de robot móvil está compuesto por dos ruedas fijas convencionales sobre el mismo eje, controladas de manera independiente y una rueda loca para dar estabilidad, este tipo de robot es muy usado en pruebas de laboratorio gracias a su cinemática simple, estructura mecánica y electrónica de control.

La Figura 3-2 muestra el robot Pioneer P3-DX muy usado en ambientes de laboratorio.



Figura 3-2: Robot unicycle Pioneer P3-DX

Fuente: <https://www.generationrobots.com/en/402395-robot-mobile-pioneer-3-dx.html#>

2.1.2.3. Triciclo clásico

El robot tipo triciclo está formado por tres ruedas, una delantera que sirve para la tracción como para la dirección, dos ruedas traseras que se mueven libremente, este sistema al igual que el anterior tiene una mecánica y electrónica de control simples, más sencillo que el sistema de cuatro ruedas, un gran problema de este tipo de robot es que su centro de gravedad tiende a moverse a los límites de la superficie de equilibrio definido por sus tres ruedas, perdiendo tracción y ocasionando errores significativos al momento de estimar su posición.

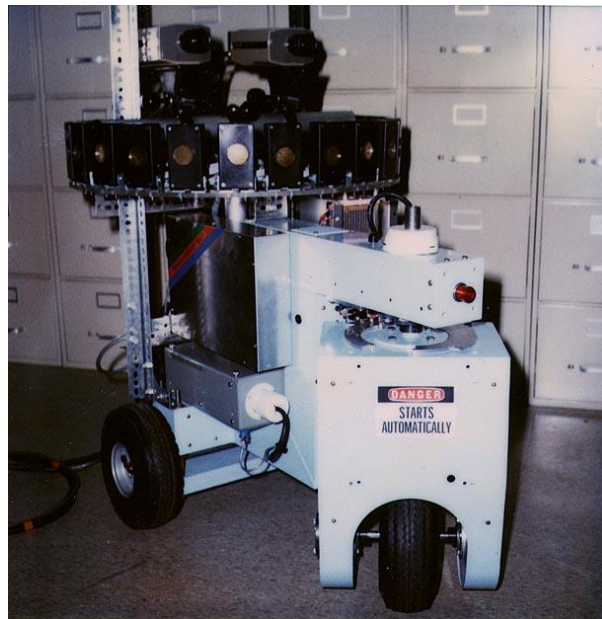
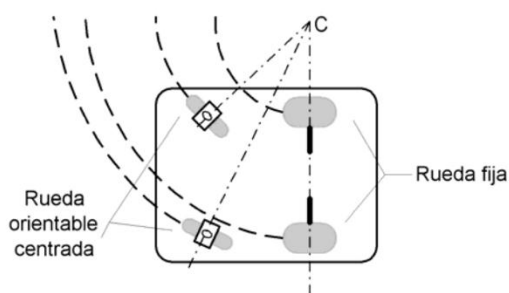


Figura 4-2: Robot móvil Neptune.

Fuente: <http://www.frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1985/al2.html>

2.1.2.4. Ackerman o cuadríciclo

Este tipo de robot móvil posee una estabilidad mucho mejor que el robot triciclo, siendo apto para aplicaciones todo terreno, la Figura 5-2a muestra la estructura de este robot, en donde los ejes frontales de las ruedas delanteras se interceptan en el punto C de la prolongación del eje de las ruedas traseras, provocando que la rueda interior gire un ángulo ligeramente superior a la rueda exterior, esto ayuda a más de dar una mejor estabilidad evita deslizamiento en las ruedas, dando menores errores al momento de estimar su posición



a) Sistema de tracción



b) Vehículo NavLab

Figura 5-2: Sistema Ackerman

Fuente: Secchi, H. A. Una introducción a los robots móviles, 2008.

En la Figura 5-2b se muestra el NavLab 1, un vehículo autónomo que ha venido empleándose en el Robotics Institute de Carnegie Mellon University desde 1986 para experimentos de navegación autónoma en exteriores (Ollero, 2001)

2.1.2.5. Pistas de deslizamiento.

Estos robots móviles son tipo oruga, en donde la tracción como el direccionamiento se consiguen mediante las pistas de deslizamiento, este tipo de locomoción es análogo al Skid steer, la pista de deslizamiento actúa de forma análoga a una rueda de gran diámetro, es ideal para terrenos difíciles e irregulares

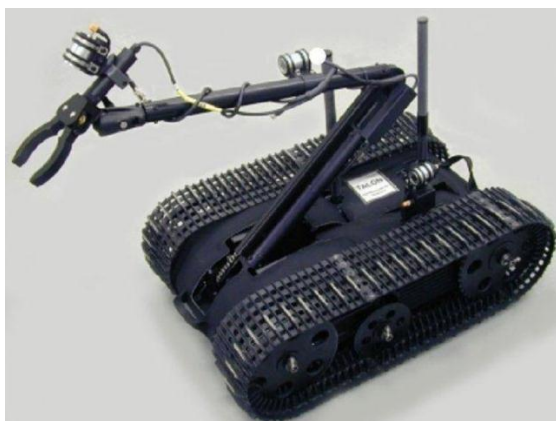


Figura 6-2: Locomoción por pistas de deslizamiento.

Fuente: <https://www.xatakaciencia.com/robotica/robots-moviles-i>

2.1.3. Locomoción con patas

En este tipo de locomoción los mecanismos, cinemática, control y estabilidad son más complejo que en los sistemas de locomoción con ruedas, en donde la estabilidad del robot es de gran interés ya que para su movilidad necesariamente tiene que levantar una o algunas de sus patas manteniendo su estabilidad en las restantes, para mejorar este inconveniente se incrementa el número de patas consiguiendo en cada paso más número de patas fijas en que apoyarse. En este tipo de locomoción, el cuerpo del robot móvil está separado del terreno por medio de patas dando mejores propiedades al momento de ejecutar caminos con obstáculos ya que el robot puede pasar sobre ellos (Rodríguez Díaz, 2014); adicionalmente por medio de las patas es posible conseguir la omnidireccionalidad y reducir el deslizamiento en la locomoción.



Figura 7-2: Robot cuadrúpedo.

Fuente: <http://noticiasdelaciencia.com/not/9452/robot-cuadrupedo>

2.2. Planificación de caminos

El problema de la planificación de caminos se centra en encontrar un camino óptimo desde una configuración inicial hasta una configuración final o meta, esta planificación es fundamental en aplicaciones donde se requiere de autonomía total o parcial del robot móvil; dentro del ambiente de trabajo las dimensiones y posición de los obstáculos pueden ser conocidas completamente, parcialmente o completamente desconocidas, la planificación de caminos en estos dos últimos requieren de un sistema de sensores, para ir recopilando información sobre el ambiente a medida que el robot avanza hacia la meta, la re-planificación de su trayectoria se vuelve inevitable por la aparición de nuevos obstáculos conforme se mueve el robot. Dentro de los algoritmos de planificación más conocidos tenemos:

2.2.1. Grafos de visibilidad.

Los grafos de visibilidad dan un enfoque geométrico al problema de generación de caminos en ambientes con obstáculos (Guzmán Luna, 2012), este método tiene la característica de generar líneas rectas que unen los vértices de los mismos, además consideran la configuración inicial y final como solución del grafo, otra característica importante de este método es que las líneas que unen dos vértices deben tener una línea de vista directa, es decir las líneas rectas no intersecan a los obstáculos.

La Figura 8-2 muestra como los grafos de visibilidad se crean, estos parten desde la configuración inicial q_{ini} y final q_{fin} , trazando tantas líneas rectas como le sea posible hacia los vértices de los obstáculos con línea de vista, sin intersecar a ninguno de ellos, de igual manera todos los vértices de los obstáculos con línea de vista directa se unen entre ellos, sin intersecarse. De este modo un camino que une el punto de salida y meta es encontrado.

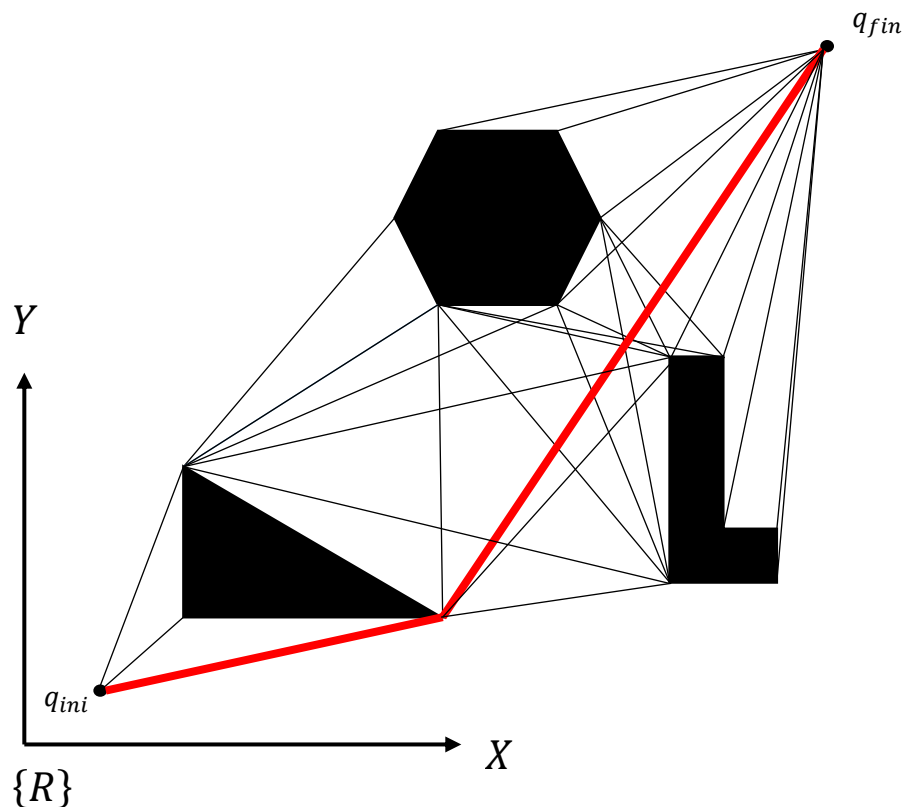


Figura 8-2: Grafo de visibilidad.

Fuente: López G. Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos, 2012

Un aspecto importante a considerar en este tipo de planificador son las dimensiones del robot móvil, dejarlo de considerar como un punto en el espacio, al tomar esta consideración los

puntos que unen los vértices dejan de ser solución de la planificación (Guzmán Luna, 2012), puesto que se incrementa las dimensiones de cada uno de los obstáculos el radio del robot, debido a este incremento las posibles soluciones disminuyen quedando como solución una trayectoria válida para la navegación del robot sin enfrentar colisión.

2.2.2. Diagramas de Voronoi

El diagrama de Voronoi puede definirse como el lugar geométrico de los puntos del plano que equidistan en todo momento de los obstáculos más cercanos, maximizando en todo momento la distancia entre el robot y el obstáculo (Rodríguez Díaz, 2014), esto se logra usando las mediatrices entre los segmento que unen los obstáculos y verificando su intersección.

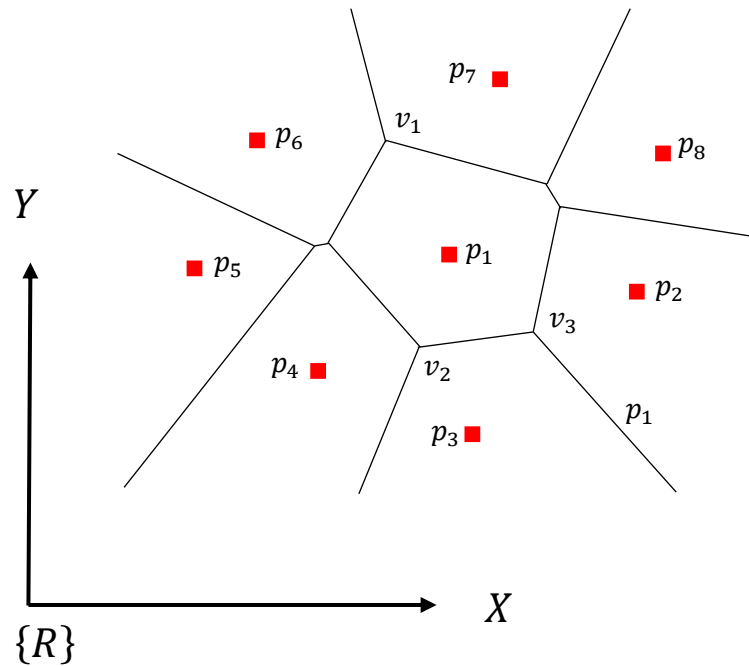


Figura 9-2: Diagrama de Voronoi generalizado.

Fuente: Muñoz C. Modelo Dinámico y Controlador de Seguimiento para Robots Móviles Tipo Uniciclo, 2014.

La Figura 9-2 muestra como el diagrama de Voronoi divide al plano en regiones, llamadas regiones de Voronoi, estas están compuestas por los punto cercanos al punto p_i que a ningún otro punto (Muñoz Cueva, 2014). Considerando el conjunto de puntos $\{p_1, p_2, \dots, p_n\}$, como la posición de los obstáculos en el plano, las líneas del diagrama de Voronoi son las trayectorias más seguras a seguir por el robot móvil ya que esta equidista de todos los obstáculos

Otra componente de los diagramas de Voronoi son los vértices $\{v_1, v_2, \dots, v_n\}$, estos equidistan de dos o más puntos p_i , en la planificación de caminos estos vértices representan los puntos

intermedios del camino, un aspecto a notar es que la configuración inicial y final no es parte del diagrama de Voronoi, para anexar estos puntos existen varios métodos, uno de los más simples es anexar estos puntos a los vértices más cercanos con líneas rectas, siempre y cuando en esta trayectoria no exista colisión alguna.

2.2.3. Descomposición de celdas

Este método consiste en dividir el espacio de configuraciones en varias regiones, dando a cada una de ellas una etiqueta para indicar si está o no dentro de un obstáculo, o está libre de colisión, la idea general de estos métodos consiste en resolver dos problemas, un problema general que resuelve las trayectorias por zonas y un problema más simple o local (López García, 2012), que encuentra una secuencia de celdas vecinas libres de colisión, estas han de unir dos puntos cualesquiera de su interior para aseguran un camino realizable. La descomposición de celdas puede dividirse en dos tipos: descomposición exacta de celdas y descomposición aproximada de celdas.

2.2.3.1. Descomposición exacta

Este método de descomposición también conocido como descomposición trapezoidal o descomposición vertical consiste en la división del espacio libre de colisiones, trazando divisiones verticales en los vértices de los polígonos que conforman los obstáculos (Rodríguez Díaz, 2014), de esta división en celdas se halla el grafo de conectividad cuyos nodos estarán en los puntos medios de los segmentos verticales generados, encontrando una secuencia de celdas en donde va estar la trayectoria, esto se puede ver en la Figura 10-2.

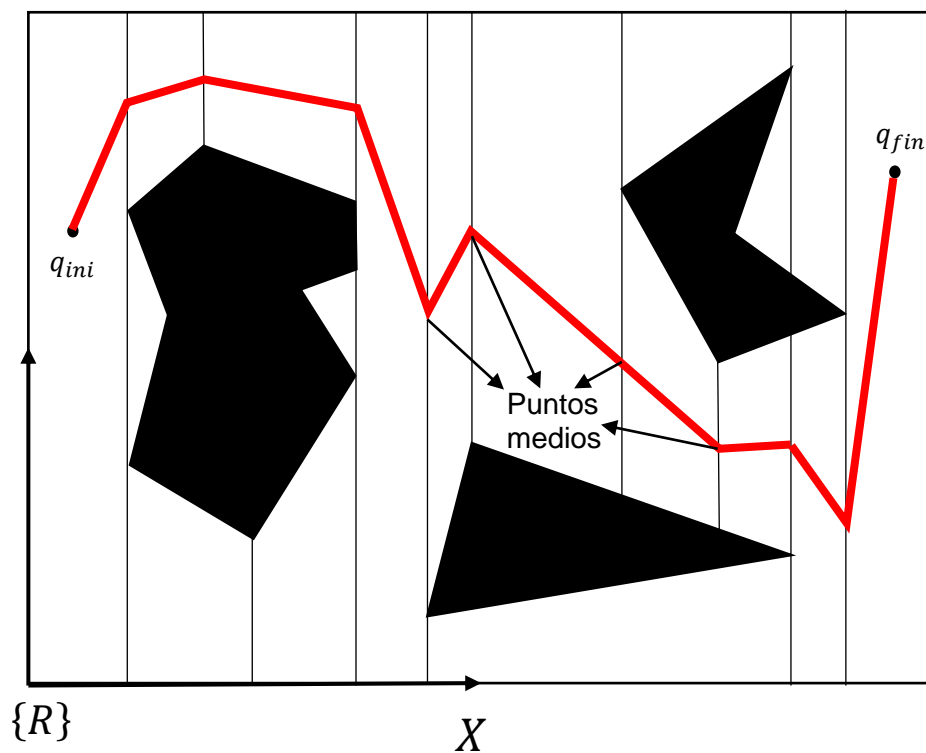


Figura 10-2: Descomposición vertical (Celdas exactas).
Fuente: LaValle S. Planning Algorithms, 2006.

Otro ejemplo de descomposición en celdas exactas es la triangulación de Delaunay, inicialmente este método considera al conjunto de puntos formado por los vértices de los obstáculos, este método consiste en una descomposición en celdas triangulares delimitadas por segmentos definidos. Sobre estas celdas se construye un grafo de adyacencia y a partir de este punto se resuelve de igual forma que en el apartado anterior (La Valle , Planning Algorithms, 2006).

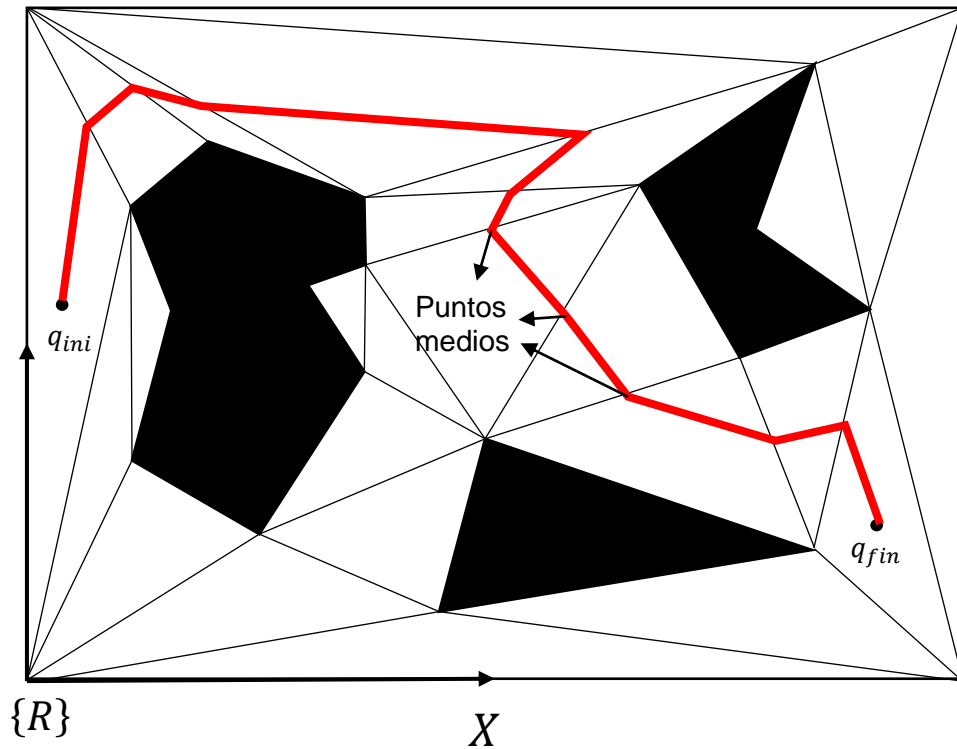
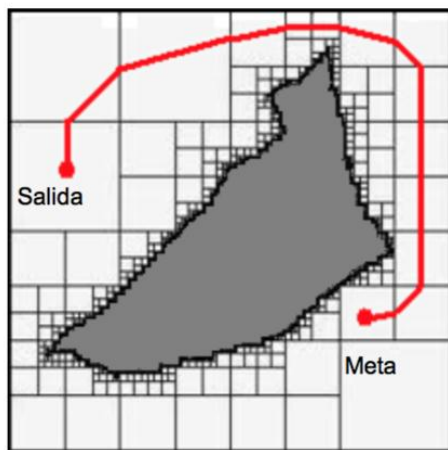


Figura 11-2: Descomposición de Delaunay.
Fuente: LaValle S. Planning Algorithms, 2006.

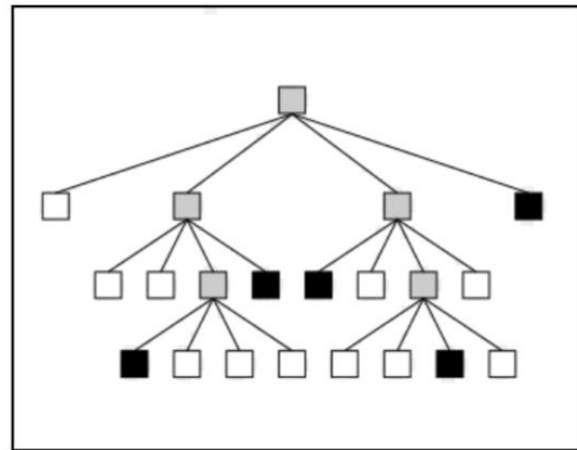
2.2.3.2. Descomposición aproximada

Este tipo de descomposición en celdas consiste principalmente en dividir el espacio en varias celdas, la resolución de estas dependerá de la densidad de los obstáculos por celda (Rodríguez Díaz, 2014). A cada celda se las clasifica en tres tipos, según su área este contenido dentro de un obstáculo, fuera o su vez en ambos.

Un ejemplo de descomposición aproximada es el denominado “*quadtree*”, este método parte de un escenario rectangular, a este rectángulo se lo divide en cuatro rectángulos semejantes a cada uno de estos se les clasifica en función de si su área esta contenido, no contenido en un obstáculo o a su vez su área este en ambos, a este último se le aplica el mismo proceso de segmentación y clasificación anterior hasta encontrar una secuencia de celdas que una la configuración inicial y final. En cada una de estas iteraciones se actualiza un árbol de grado cuatro, es decir de cada nodo del árbol descienden cuatro. El nodo raíz es representado por el rectángulo inicial que contiene todo el escenario, de este descienden 4 nodos (rectángulos) de la primera subdivisión, del mismo modo están relacionados el resto de nodos, tomando en cuenta que los nodos etiquetados como llenos o vacíos no tienen nodos descendientes, solo los mixtos (López García, 2012).



a) Partición del escenario



b) Estructura en árbol de las particiones

Figura 12-2: Descomposición *quadtree*.

Fuente: López G. Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos, 2012.

La Figura 12-2a muestra cómo se realiza la descomposición en celdas, la resolución de estas dependen de la densidad de los obstáculos, la descomposición se realiza hasta que el algoritmo encuentre una secuencia de celdas desde la salida hasta la meta, la Figura 12-2b muestra un ejemplo del árbol de grado 4 en donde se representa a las celdas vacías en color blanco, las celdas llenas color negro y las celdas mixtas en color gris, siendo estas últimas las únicas que tiene nodos descendientes

2.2.4. Mapas probabilísticos (PRMs)

La idea de este planificador consiste en que durante la construcción de un mapa probabilístico cree también una estructura de datos para ser usado en la planificación de caminos, comúnmente este tipo de planificador se usa para robots móviles de muchos grados de libertad, en donde se distingue dos tipos, los de consulta única y los de consulta múltiple.

En los PRM de consulta múltiple puede existir varios grafos, este método inicia la planificación generando varios puntos aleatorios en el escenario, para luego ejecutar un algoritmo de planificación local entre pares de puntos próximos entre si (López García, 2012), si en la planificación local existe alguna colisión entre la trayectoria de dos puntos, esta es descartada, sino un nuevo grafo es incrementado con los puntos en donde se ha logrado conectividad, esto se realiza hasta que finalmente el grafo contenga una solución que une los puntos inicial y final.

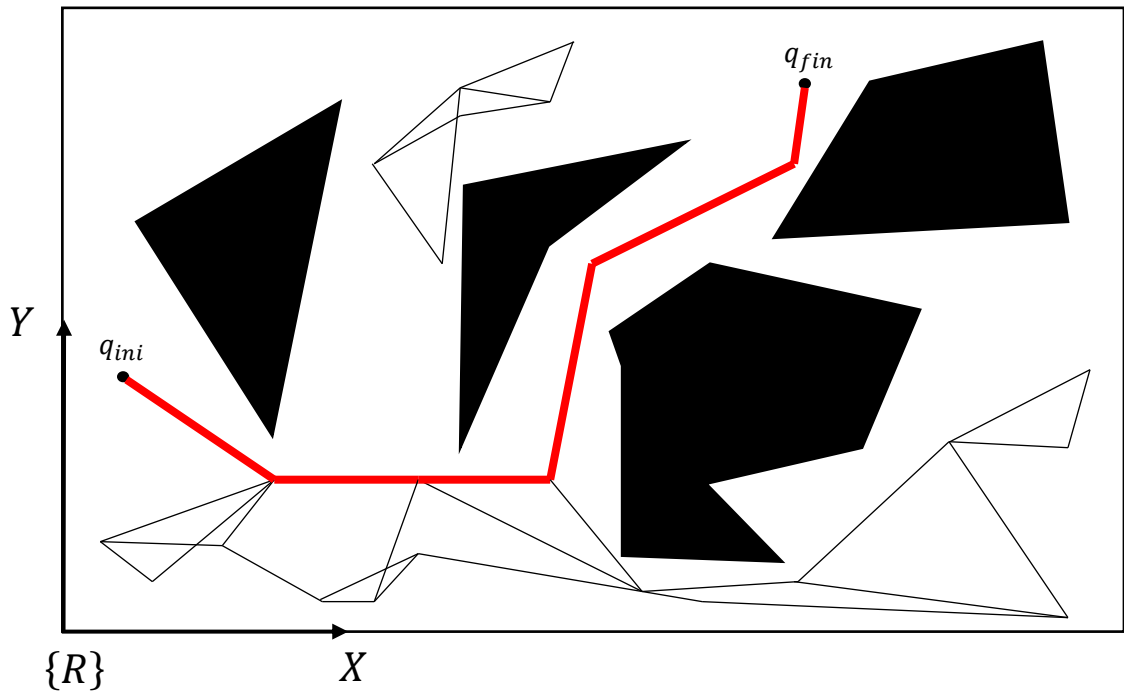


Figura 13-2: PRM de múltiple consulta.

Fuente: López G. Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos, 2012.

Una de las principales ventajas de este método consiste en que el algoritmo no tiene que determinar los límites de los obstáculos, sino determinar si existe o no colisión con alguno de ellos, ya que el costo computacional para determinar los límites de todos los obstáculos crece con el incremento de los mismos, otra principal ventaja consiste en la completitud probabilista, debido a que la probabilidad de encontrar una solución que una los puntos inicial y final tiende a uno con el incremento de nodos.

2.2.5. Exploración rápida de árboles al azar (*Rapidly Exploring Random tree*)

Desde su aparición en 1998, en una publicación de Steven M. LaValle, este algoritmo de planificación ha sido de gran interés por la comunidad científica dando como resultado varios trabajos de investigación en donde se resaltan sus características, ventajas y nuevas posibles mejoras, este algoritmo crea un árbol de configuraciones al azar, este árbol de configuraciones se puede usar para la planificación de trayectorias como una herramienta de exploración aleatoria en un espacio de búsqueda en donde se desea encontrar un camino entre dos configuraciones o nodos (Nieto, Slawinski, Mut, & Wagner, 2010), si los nodos almacenan las configuraciones visitadas en el espacio, la búsqueda puede empezar por agregar el nodo de inicio como en nodo raíz del árbol y hacerlo crecer por todo el espacio de configuraciones libre

de colisiones, hasta un número determinado de iteraciones o hasta alcanzar la meta, si existe un camino y el árbol cubre todo el espacio libre de colisiones la solución se encontrará.

2.2.5.1. Algoritmo RRT básico.

Este algoritmo ha sido de gran interés por la comunidad científica, dando como resultado varias investigaciones en las cuales se realiza posibles mejoras como en (Nieto, Slawinski, Mut, & Wagner, 2010) donde el algoritmo básico RRT incluye dos eurísticas para mejorar el rendimiento y generar un conjunto de trayectorias libres de colisiones, de las cuales se elige la trayectoria que tiene mayor similitud con un camino de referencia dado por un operador humano. Como principal objetivo del algoritmo RRT básico es encontrar un árbol de configuraciones al azar que cubra todo el espacio libre de colisiones como se muestra en la Figura 14-2; y con la ayuda de este árbol de configuraciones se encuentra un camino entre dos puntos.

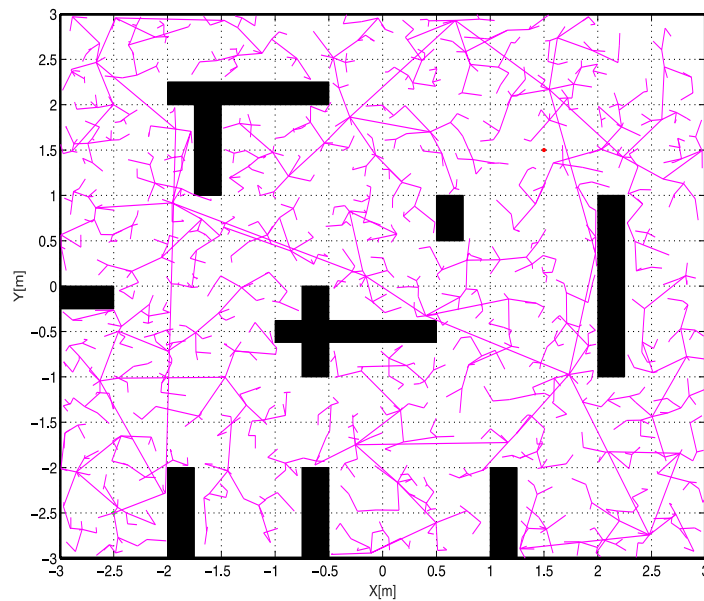


Figura 14-2: RRT básico.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017

La Figura 14-2 muestra en color magenta el árbol de configuraciones creado por el algoritmo RRT, este árbol explora todo el espacio de configuraciones libre de colisiones.

```

RRT( $q_{ini}$ )
1  $Arbol[0] = q_{ini}$ ;
2 while  $k = 1$  hasta  $K_{max}$ 
3    $q_{rand} =$ 
CONFIGURACION_ALEATORIA();
4   EXTIENDE( $Arbol, q_{rand}$ );
5 devuelve  $Arbol$ 

```

Figura 15-2: Algoritmo RRT básico.

Fuente: LaValle &Kuffner. Randomized Kinodynamic Planning, 2001

La Figura 15-2 muestra el algoritmo RRT básico, este algoritmo en la línea 1 anexa la configuración inicial q_{ini} al árbol de configuraciones, en la línea 2 empieza el bucle **while** limitado por K_{max} ; K_{max} determina el número máximo de iteraciones a ejecutar el algoritmo, en caso de que una solución no sea encontrada, Dentro del bucle **while** en la línea 3 la función CONFIGURACION_ALEATORIA() genera un punto al azar dentro del espacio de configuraciones; mientras que en la línea 4 la función EXTIENDE() hace crecer al árbol en dirección a q_{rand} . La función EXTIENDE () se muestra en la Figura 16-2

```

EXTIENDE( $Arbol, q_{rand}$ )
1  $q_{near} =$ VECINO_PROXIMO;
2 if NUEVA_CONFIGURACION( $q_{rand}, q_{near}, q_{new}$ ) then
3   AÑADE_VERTICE( $Arbol, q_{new}$ )
4   if  $q_{new} = q_{rand}$  then
5     devuelve "alcanzado"
6   else
7     devuelve "avanzado"
8 else
9   devuelve "rechazado"

```

Figura 16-2: Función Extiende.

Fuente: LaValle &Kuffner. Randomized Kinodynamic Planning, 2001.

En la línea 1 de la función EXTIENDE(), q_{near} es encontrado con la función VECINO_PROXIMO(), este punto es el más cercano a q_{rand} del árbol de configuraciones, en la línea 2 q_{new} es encontrado con la función NUEVA_CONFIGURACION(), esta realiza un desplazamiento de tamaño ε partiendo de q_{near} en dirección a q_{rand} , para la obtención de q_{new} se tiene en cuenta si en dicho desplazamiento existe o no colisión, en la línea 3 q_{new} es añadido al árbol de configuraciones con la función AÑADE_VERTICE(), este nuevo punto es agregado al árbol de configuraciones distinguiéndose dos posibilidades, si el punto aleatorio q_{rand} esta dentro de la circunferencia con centro en q_{near} y radio ε , q_{new} coincide con q_{rand} , en este caso el algoritmo notifica que el punto ha sido "alcanzado", si por el contrario no se ha podido

alcanzar q_{rand} el algoritmo devuelve el valor de "avanzado". De no existir una nueva configuración q_{new} el algoritmo devuelve el valor de "rechazado" indicando que no se ha incrementado nuevas ramas al árbol de configuraciones.

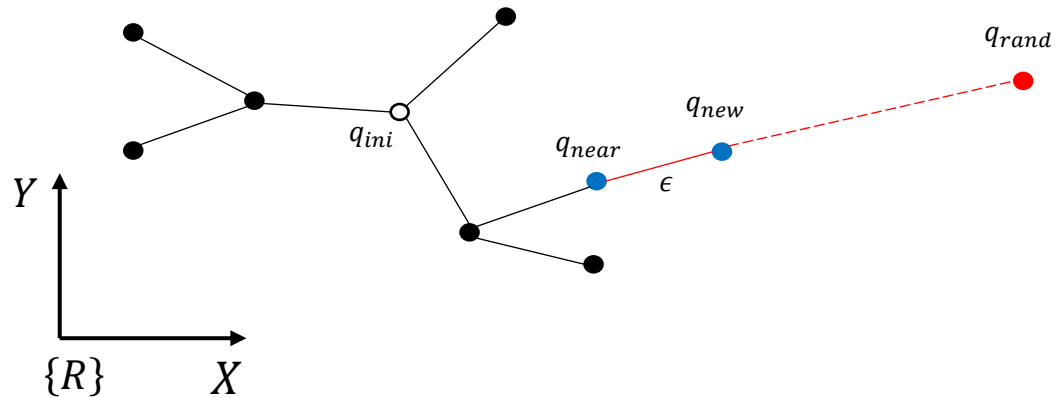


Figura 17-2: Esquema de expansión del RRT.

Fuente: LaValle & Kuffner. Randomized Kinodynamic Planning, 2001.

La Figura 17-2 muestra el proceso de la función `EXTIENDE()`, en donde se observa como q_{new} es encontrado.

2.3. Post-procesado

La mayoría de caminos obtenidos a través del algoritmo RRT son muy irregulares, dejando mucho que desear por los sistemas de control, en la mayoría de casos el robot tiene que realizar giros innecesarios (López García, 2012), generando altos costes computacionales innecesarios, por esto es de mucha importancia realizar un post-procesado al camino inicial encontrado. Este algoritmo permite obtener un camino simplificado sustituyendo el camino inicial por tramos rectos.

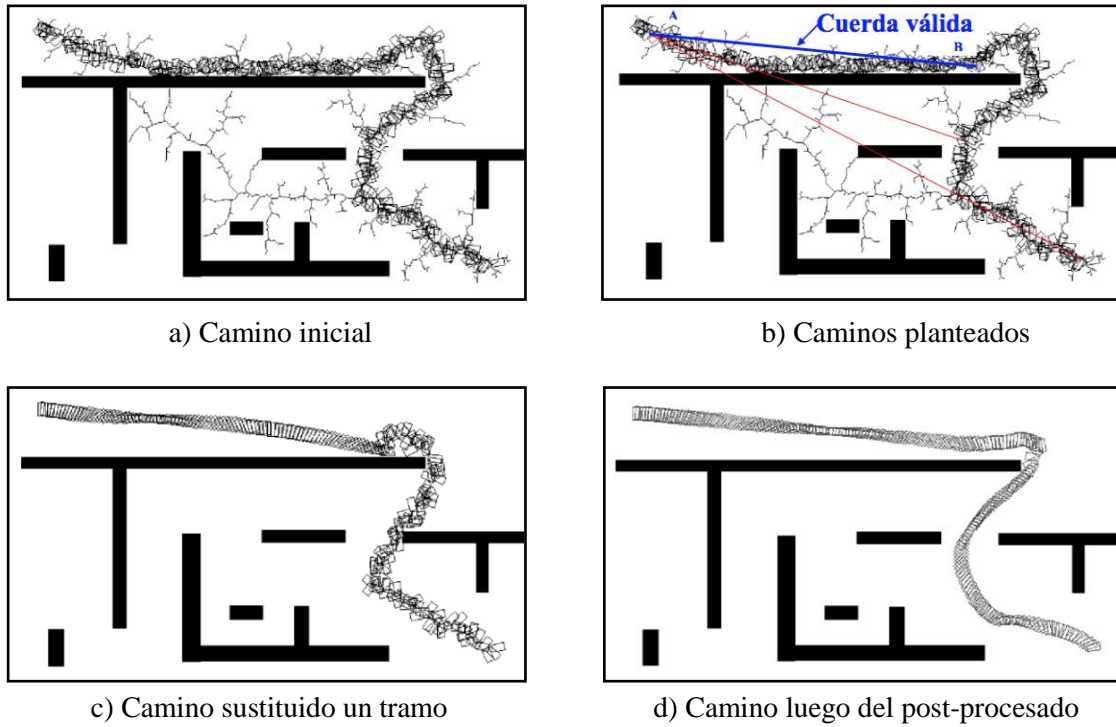


Figura 18-2: Post-procesado.

Fuente: López G. Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos, 2012.

La Figura 18-2 muestra las acciones de un algoritmo de post-procesado, en a) muestra el camino inicial encontrado por el algoritmo RRT, en primera instancia este algoritmo intenta reemplazar todo el camino por una trayectoria recta como se muestra en b), si esto no es posible encontrándose con algún obstáculo, disminuye el camino a reemplazar, normalmente toma la mitad del tramo inicial para realizar otra prueba, de esta manera el algoritmo va disminuyendo el tramo a reemplazar hasta encontrar una ruta sin colisiones, en cada nuevo reemplazo el camino queda configurado con un nuevo vértice inicial y el camino final con un nuevo tramo recto, en c) se observa como el tramo recto reemplaza una sección del camino original, y finalmente en d) se observa el camino resultante luego de aplicar un post-procesado.

CAPITULO III

3. PLANIFICACION Y SEGUIMIENTO DE CAMINOS

3.1. Modelo Cinemático Robot unicycle.

Un robot con ruedas tal como el de tipo unicycle es uno de los más usados para investigación y desarrollo debido a su alta tracción con ruedas y ventajas de movilidad, estos robots en su mayoría están conformados por dos ruedas controladas de manera autónoma (De La Cruz & Carelli, 2006). Para este caso la posición del robot móvil se define en el punto $\mathbf{P}(x, y)$, como se muestra en la Figura 1-3, además se considera una masa puntual por lo que esta no ejerce efecto alguno sobre la estructura, anulando los momentos de inercia y rozamientos.

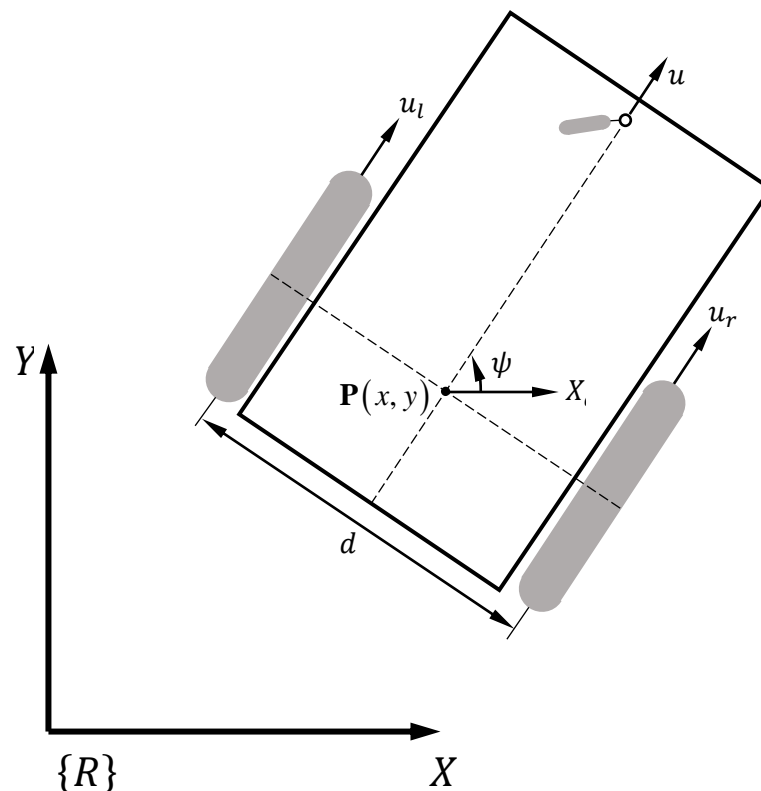


Figura 1-3: Geometría uno del robot móvil unicycle

Fuente: De La Cruz & Carelli. Dynamic Modeling and Centralized Formation Control of Mobile Robots 2006.

La Figura 1-3 muestra la geometría y parámetros del robot móvil tipo unicycle considerados para hallar la cinemática del robot móvil, en donde $\{R\}$, representa el sistema de coordenadas global, $\mathbf{P}(x, y)$ representa la posición del robot móvil, ψ la orientación, todo esto con respecto al sistema de coordenadas global $\{R\}$. La velocidad lineal del robot móvil está determinada por:

$$u = \frac{u_r + u_l}{2} \quad (3.1)$$

donde, u es la velocidad lineal del robot móvil; u_r y u_l representan las velocidades de desplazamiento de la rueda derecha e izquierda respectivamente.

La velocidad angular del robot móvil está determinado por

$$\omega = \frac{u_r - u_l}{d} \quad (3.2)$$

donde ω es la velocidad angular del robot móvil; y d es la distancia ente la rueda derecha e izquierda.

Tomando en consideración (3.1) y (3.2) la cinemática del robot móvil con el punto de control en $\mathbf{P}(x, y)$ es:

$$\begin{aligned} \dot{x} &= u \cos \psi \\ \dot{y} &= u \sin \psi \\ \dot{\psi} &= \omega \end{aligned} \quad (3.3)$$

En donde la restricción no holonómica está dada por:

$$\dot{y} \cos \psi - \dot{x} \sin \psi = 0 \quad (3.4)$$

Esta restricción indica que el robot móvil solo se puede desplazar en dirección a las ruedas de tracción, es decir el robot no puede moverse hacia sus costados.

Otra consideración al momento de determinar la cinemática del robot móvil tipo unicycle es cuando el punto de control no se encuentra en el centro del eje las ruedas como se ve en la Figura 3-2.

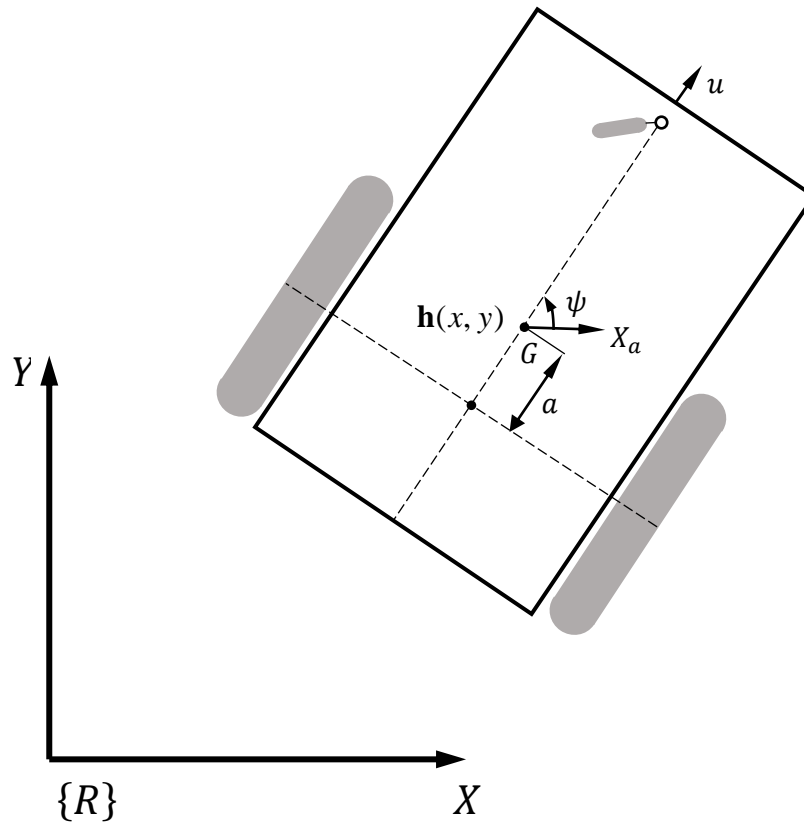


Figura 2-3: Geometría dos del robot móvil unicycle

Fuente: De La Cruz & Carelli. Dynamic Modeling and Centralized Formation Control of Mobile Robots 2006.

En la Figura 2-3, $\{R\}$ es el sistema de coordenadas global, la posición del robot móvil está definida por $\mathbf{h}(x, y)$, en este mismo punto se encuentra el punto G , este último es el centro de masa y se encuentra a una distancia a del centro del eje de las ruedas, ψ representa la orientación del robot móvil, todos estos puntos toman como referencia el Sistema de coordenadas global $\{R\}$

El modelo cinemático en donde la posición del robot está determinado por el punto $\mathbf{h}(x, y)$, se lo determina de acuerdo a (Rossomando, Soria, & Carelli, 2010) en donde indica el modelo cinemático del robot móvil tipo unicycle con la siguiente ecuación.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -a\sin\psi \\ \sin\psi & a\cos\psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (3.5)$$

En este modelo no se ha considerado ningún tipo de incertidumbre provocadas por la velocidad de deslizamiento y de la orientación del robot.

La restricción no holonómica de este modelo es:

$$\dot{x} \sin \psi - \dot{y} \cos \psi + a \psi = 0 \quad (3.6)$$

en donde indica como en el modelo anterior que el robot móvil solo puede desplazarse en dirección a las ruedas de tracción, si se desea cambiar al robot móvil de dirección también tiene que cambiar su orientación.

3.2. Modelo dinámico

La obtención del modelo dinámico del robot móvil tipo unicycle se basa en los modelos datos por (Andaluz, Roberti, Toibero, Carelli, & Wagner, 2011). La Figura 3-3 muestra los parámetros a considerar en la obtención del modelo dinámico:

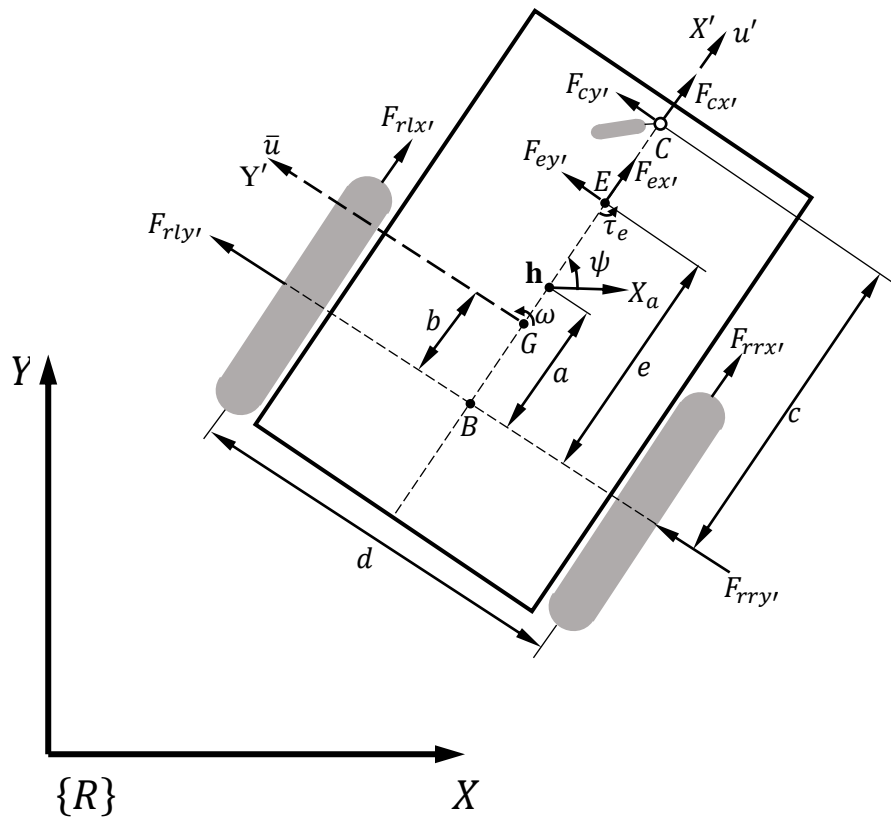


Figura 3-3: Parámetros del robot móvil tipo unicycle.

Fuente: De La Cruz & Carelli. Dynamic Modeling and Centralized Formation Control of Mobile Robots 2006.

A continuación se describen los parámetros del robot unicycle a considerar en la obtención del modelo.

- G Centro de masa
- B Centro de la línea que une las dos ruedas

$\mathbf{h}(x, y)^T$	Es el punto de control
u y \bar{u}	Velocidades longitudinal y lateral del centro de masa
ω y ψ	Velocidad angular y orientación del robot móvil
a, b, c, d y e	Son las distancias
F_{rrx} y F_{rry}	Fuerza longitudinal y lateral en el neumático de la rueda derecha
F_{rlx} y F_{rly}	Fuerza longitudinal y lateral en el neumático de la rueda izquierda
F_{cx} y F_{cy}	Fuerzas longitudinal y lateral sobre C por la rueda libre
F_{ex} y e_{cy}	Fuerzas longitudinal y lateral ejercidas sobre E por la herramienta
τ_e	Momento ejercido por la herramienta

La obtención del modelo dinámico completo se muestra en (Andaluz, Roberti, Toibero, Carelli, & Wagner, 2011), en donde toma en consideración que para la mayoría de aplicaciones con robots móviles, estos dispone de controladores tipo PD para los motores, es por esto que se muestra el modelo incluyendo este tipo de controlador y teniendo como variables de control las velocidades y no los voltajes; de este modo el controlador PD con referencia de velocidad constante, es descrito a continuación:

$$\begin{bmatrix} \dot{v}_u \\ \dot{v}_\omega \end{bmatrix} = \begin{bmatrix} k_{PT}(u_{ref} - u_{me}) + k_{DT}(\dot{u}_{ref} - \dot{u}_{me}) \\ k_{PR}(\omega_{ref} - \omega_{me}) + k_{RT}(\dot{\omega}_{ref} - \dot{\omega}_{me}) \end{bmatrix} \quad (3.7)$$

donde,

$$u_{me} = \frac{1}{2}[r(\omega_r)], \quad \omega_{me} = \frac{1}{d}[r(\omega_r - \omega_l)] \quad (3.8)$$

Las variables \dot{u}_{ref} y $\dot{\omega}_{ref}$ son despreciadas en (3.7) para simplificar el modelo

Tomando en consideración lo anterior el modelo dinámico del robot móvil tipo unicycle se muestra a continuación:

$$\begin{bmatrix} \dot{u} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{\bar{\theta}_3^0}{\bar{\theta}_1^0} \omega^2 - \frac{\bar{\theta}_4^0}{\bar{\theta}_1^0} u \\ -\frac{\bar{\theta}_5^0}{\bar{\theta}_2^0} u \omega - \frac{\bar{\theta}_6^0}{\bar{\theta}_2^0} \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{\bar{\theta}_1^0} & 0 \\ 0 & \frac{1}{\bar{\theta}_2^0} \end{bmatrix} \begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix} + \begin{bmatrix} \bar{\delta}_u \\ \bar{\delta}_\omega \end{bmatrix} \quad (3.9)$$

Los parámetros del modelo dinámico son:

$$\begin{aligned}
\bar{\theta}_1^0 &= \frac{\frac{R_a}{k_a} (mR_t r + 2I_e) + 2rk_{DT}}{2rk_{PT}} \\
\bar{\theta}_2^0 &= \frac{\frac{R_a}{k_a} (i_e d^2 + 2R_t r (I_z + mb^2)) + 2rdk_{DR}}{2rdk_{PR}} \\
\bar{\theta}_3^0 &= \frac{\frac{R_a}{k_a} mbR_t}{2k_{PT}}, \quad \bar{\theta}_4^0 = \frac{\frac{R_a}{k_a} \left(\frac{k_a k_b}{R_a} + B_e \right)}{rk_{PT} + 1} \\
\bar{\theta}_5^0 &= \frac{\frac{R_a}{k_a} mbR_t}{dk_{PR}}, \quad \bar{\theta}_6^0 = \frac{\frac{R_a}{k_a} \left(\frac{k_a k_b}{R_a} + B_e \right) d}{2rk_{PR} + 1}
\end{aligned} \tag{3.10}$$

Los elementos del vector incertidumbre $[\bar{\delta}_u \quad \bar{\delta}_\omega]^T$ son:

$$\begin{aligned}
\bar{\delta}_u &= \frac{R_1}{2\bar{\theta}_1^0} \frac{R_a}{K_{PT} K_a} (m\omega \bar{u}^s + F_{ex} + F_{ex'}) + \frac{\bar{\theta}_4^0}{2\bar{\theta}_1^0} (u_r^s + u_l^s) \\
&\quad + \left(\frac{1_e R_a + rk_a k_{DR}}{2\bar{\theta}_1^0 rk_{PT} k_a} \right) (\dot{u}_r^s + \dot{u}_l^s) \\
\bar{\delta}_\omega &= \frac{\bar{\theta}_6^0}{\bar{\theta}_6^0 d} (u_l^s - u_r^s) + \left(\frac{1_e R_a d + 2rk_a k_{DR}}{2\bar{\theta}_2^0 rk_{PR} K_a d} \right) (\dot{u}_r^s - \dot{u}_l^s) - \frac{\bar{\theta}_5^0}{\bar{\theta}_2^0} \dot{u}^5 \\
&\quad + \frac{R_1 R_a}{\bar{\theta}_2^0 dk_{PR} K_a} (eF_{ey'} + cF_{cy'} + \tau_e)
\end{aligned} \tag{3.11}$$

Reagrupando los términos y despreciando δ_u y $\delta_{\omega'}$, (3.9) se puede escribir como:

$$\begin{bmatrix} \bar{\theta}_1^0 & 0 \\ 0 & \bar{\theta}_2^0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \bar{\theta}_4^0 & -\bar{\theta}_3^0 \omega \\ \bar{\theta}_5^0 \omega & \bar{\theta}_6^0 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix} \tag{3.12}$$

o en una forma compacta como:

$$\mathbf{H}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} = \mathbf{v}_{ref} \tag{3.13}$$

donde $\mathbf{v}_{ref} = [u_{ref} \quad \omega_{ref}]^T$ es el vector de referencias de velocidades; $\mathbf{v} = [u \quad \omega]^T$ es el vector que contiene las velocidades reales de robot; y las matrices \mathbf{H} y $\mathbf{C}(\mathbf{v})$ son:

$$\mathbf{H} = \begin{bmatrix} \bar{\theta}_1^0 & 0 \\ 0 & \bar{\theta}_2^0 \end{bmatrix} \quad y \quad \mathbf{C}(\mathbf{v}) = \begin{bmatrix} \bar{\theta}_4^0 & -\bar{\theta}_3^0 \\ \bar{\theta}_5^0 \omega & \bar{\theta}_6^0 \end{bmatrix} \tag{3.14}$$

La matriz $\mathbf{C}(\mathbf{v})$ puede ser dividida en dos matrices $\mathbf{C}(\mathbf{v}) = \mathbf{C}'(\mathbf{v}) + \mathbf{F}'$, donde

$$\mathbf{C}'(\mathbf{v}) = \begin{bmatrix} 0 & -\bar{\theta}_3^0 \omega \\ \bar{\theta}_5^0 \omega & 0 \end{bmatrix} \quad y \quad \mathbf{F}' = \begin{bmatrix} \bar{\theta}_4^0 & 0 \\ 0 & \bar{\theta}_6^0 \end{bmatrix} \quad (3.15)$$

Tomando esto en consideración el modelo dinámico puede ser escrito como:

$$\mathbf{H}\dot{\mathbf{v}} + \mathbf{C}'(\mathbf{v})\mathbf{v} + \mathbf{F}'\mathbf{v} = \mathbf{v}_{\text{ref}} \quad (3.16)$$

De (3.16) se aclara que los productos de las matrices \mathbf{H} por $\dot{\mathbf{v}}$, \mathbf{C}' por \mathbf{v} y \mathbf{F}' por \mathbf{v} resulta en velocidades y no en pares como los modelos clásicos.

3.2.1. Propiedades de los modelos Dinámicos

Par analizar las propiedades del modelo el término $\mathbf{C}(\mathbf{v})\mathbf{v}$ de (3.12), será escrito como:

$$\begin{bmatrix} \bar{\theta}_4^0 & -\bar{\theta}_3^0 \omega \\ \bar{\theta}_5^0 \omega & \bar{\theta}_6^0 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \bar{\theta}_4^0 & -\bar{\theta}_3^0 \omega \\ I\bar{\theta}_3^0 \omega & \bar{\theta}_6^0 + (\bar{\theta}_5^0 - I\bar{\theta}_3^0)u \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (3.17)$$

donde $I = 1rad^2/m^2$ es una constante para estabilizar las unidades. Considerando que $\bar{\theta}_6^0 \gg |(\bar{\theta}_5^0 - I\bar{\theta}_3^0)u|$, el término $\mathbf{C}(\mathbf{v})\mathbf{v}$ puede ser escrito como:

$$\begin{bmatrix} 0 & -\bar{\theta}_3^0 \omega \\ I\bar{\theta}_3^0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \bar{\theta}_4^0 & -\bar{\theta}_3^0 \omega \\ 0 & \bar{\theta}_6^0 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (3.18)$$

y el modelo dinámico puede ser escrito como:

$$\mathbf{H}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{F}\mathbf{v} = \mathbf{v}_{\text{ref}} \quad (3.19)$$

Una importante característica del modelo presentado es que las entradas son las velocidades lineal y angular, no pares ni voltajes como el resto de los modelos típicos. El modelo presentado en (3.19) posee importantes características que pueden ser usadas para diseño de controladores en donde se tome en cuenta la dinámica del robot, esas características son:

1. La matriz \mathbf{H} es simétrica y definida positiva, o sea $\mathbf{H} = \mathbf{H}^T > \mathbf{0}$;
2. La inversa de \mathbf{H} existe y también es definida positiva, o sea $\exists \mathbf{H}^{-1} > \mathbf{0}$
3. La matriz \mathbf{F} es simétrica y definida positiva, o sea $\mathbf{F} = \mathbf{F}^T > \mathbf{0}$
4. Las matrices \mathbf{H} y \mathbf{F} son constantes

5. La matriz $\mathbf{C}(\mathbf{v})$ es anti-simétrica.
6. El mapeo $\mathbf{V}_{\text{ref}} \rightarrow \mathbf{V}$, es simétricamente positivo de salida.

La propiedad 1 y 2 puede ser probadas observándose que \mathbf{H} es una matriz diagonal y que sus términos son todos positivos, la propiedad 3 es válida si $\bar{\theta}_6^0 > -(\bar{\theta}_5^0 - I\bar{\theta}_3^0)u$, o cuando se considera $\bar{\theta}_6^0 \gg |(\bar{\theta}_5^0 - I\bar{\theta}_3^0)u|$. La propiedad 4 es verdadera bajo la hipótesis de que no hay cambio en los valores de los parámetros como, la estructura del robot, su masa, su momento de inercia, etc. Es importante mencionar que \mathbf{F} es constante cuando se considera $\bar{\theta}_6^0 \gg |(\bar{\theta}_5^0 - I\bar{\theta}_3^0)u|$, y que \mathbf{H} , no depende de la posición del robot, cuando se desplace en el plano horizontal, ya que un desplazamiento en un plano inclinado provoca cambios en su momento de inercia.

Como se mencionó en el capítulo 1, el robot utilizado para las pruebas experimentales es el construido en el proyecto de investigación “Tele-operación bilateral cooperativo de múltiples manipuladores móviles”, proyecto ganador de la IX Convocatoria CEDIA-CEPRA 2015. Este robot es un Pioneer 3-AT, como se muestra en la Figura 4-3.



Figura 4-3: Robot móvil tipo unicycle
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En donde los parámetros identificados son:

$$\bar{\theta}_1^0 = 0.4072$$

$$\bar{\theta}_2^0 = 0.2937$$

$$\bar{\theta}_3^0 = -0.287$$

$$\bar{\theta}_4^0 = 0.9979$$

$$\bar{\theta}_5^0 = 0.0004$$

$$\bar{\theta}_6^0 = 0.9865$$

Con una velocidad de u reducida menor igual a $0,8 \text{ m/s}$ la hipótesis de $\bar{\theta}_6^0 \gg |(\bar{\theta}_5^0 - I\bar{\theta}_3^0)u|$ es verdadera, por lo tanto el modelo dinámico del robot Pioneer 3-AT, puede ser representado por (3.16), Es posible observar que $(\dot{\mathbf{H}} - 2\mathbf{C})$ es anti-simétrica, ya que la matriz \mathbf{H} es constante y por lo tanto, $\dot{\mathbf{H}} = \mathbf{0}$. Para mostrar que el mapeo $\mathbf{v}_{\text{ref}} \rightarrow \mathbf{v}$ es estrictamente pasivo de salida, considerando el modelo (3.19), considérese la función positiva $V = \frac{1}{2} \mathbf{v}^T \mathbf{H} \mathbf{v}$ y su primera derivada temporal es $\dot{V} = \mathbf{v}^T \mathbf{H} \dot{\mathbf{v}}$ Usando (3.16) y aplicando la propiedad anti-simétrica de $\mathbf{C}(\mathbf{v})$, \dot{V} puede ser escrito como.

$$\dot{V} = \mathbf{v}^T (\mathbf{v}_{\text{ref}} - \mathbf{C}\mathbf{v} - \mathbf{F}\mathbf{v}) = \mathbf{v}^T \mathbf{v}_{\text{ref}} - \mathbf{v}^T \mathbf{F}\mathbf{v}. \quad (3.20)$$

Integrando (3.20),

$$V(T) - V(0) = \int_0^T \mathbf{v}^T \mathbf{v}_{\text{ref}} dt - \int_0^T \mathbf{v}^T \mathbf{F}\mathbf{v} dt \quad (3.21)$$

Que resulta, después de despreciar $V(T)$, en la desigualdad

$$\int_0^T \mathbf{v}^T \mathbf{v}_{\text{ref}} dt \geq -V(0) + \lambda_{\min}(\mathbf{F}) \|\mathbf{v}\|_{2,T}^2 \quad (3.22)$$

Donde

$\lambda_{\min}(\cdot)$ Representa el menor auto valor de una matriz, por lo tanto, basado en (3.22) se puede concluir que el mapeo de $\mathbf{v}_{\text{ref}} \rightarrow \mathbf{v}$ de (3.19) es estrictamente pasivo de salida.

3.3. Validación del modelo Cinemático y Dinámico.

Para mostrar el comportamiento de los modelo cinemático y dinámico se plantea realizar una simulación en donde se introduce en ambos modelos la misma referencia de velocidades $[u_{\text{ref}}, \omega_{\text{ref}}]^T$, esto se muestra en la Figura 3-5.

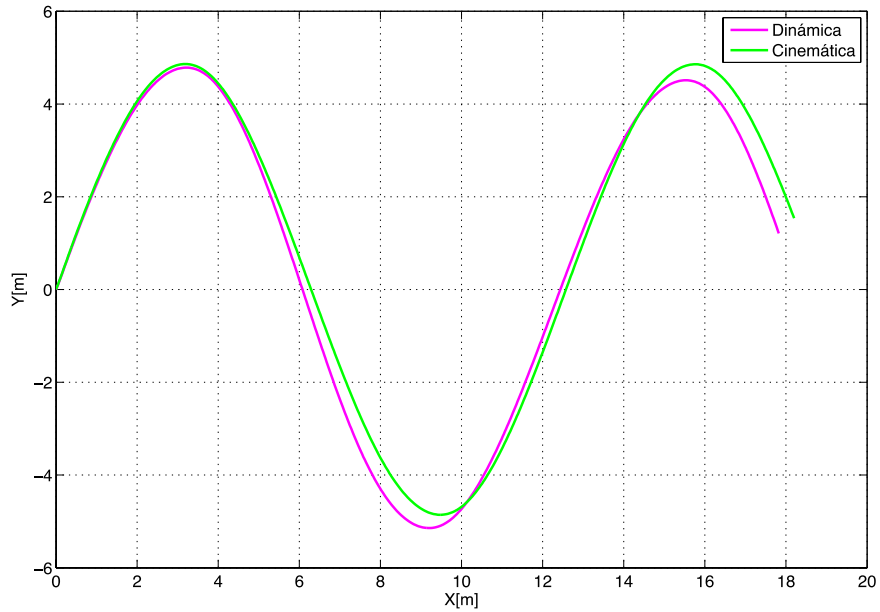


Figura 5-3: Validación 1: Camino del modelo cinemático y dinámico.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 5-3 muestra en color magenta el camino ejecutado considerando el modelo dinámico y en color verde el camino ejecutado por el modelo cinemático, en donde se observa una pequeña diferencia en el comportamiento de ambos modelos, el modelo cinemático describe un comportamiento ideal, mientras que el modelo dinámico describe un comportamiento más real, producto de la inercia del robot móvil dando como resultado caminos similares pero diferentes.

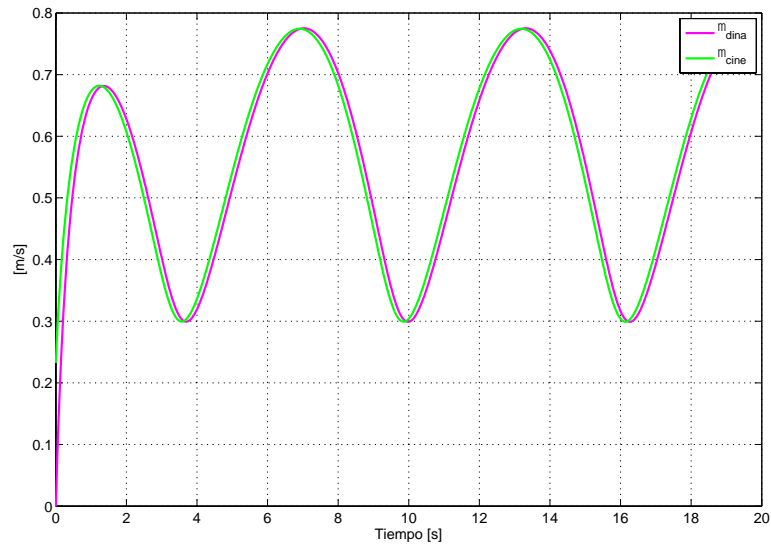


Figura 6-3: Validación 1: Velocidad lineal del modelo cinemático y dinámico.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 6-3 muestra el comportamiento de la velocidad en ambos modelos, en donde se observa comportamientos similares.

Se realiza una segunda simulación incrementando la masa del robot móvil en un 15%, para distinguir de mejor forma la acción de esta en el comportamiento de los modelos.

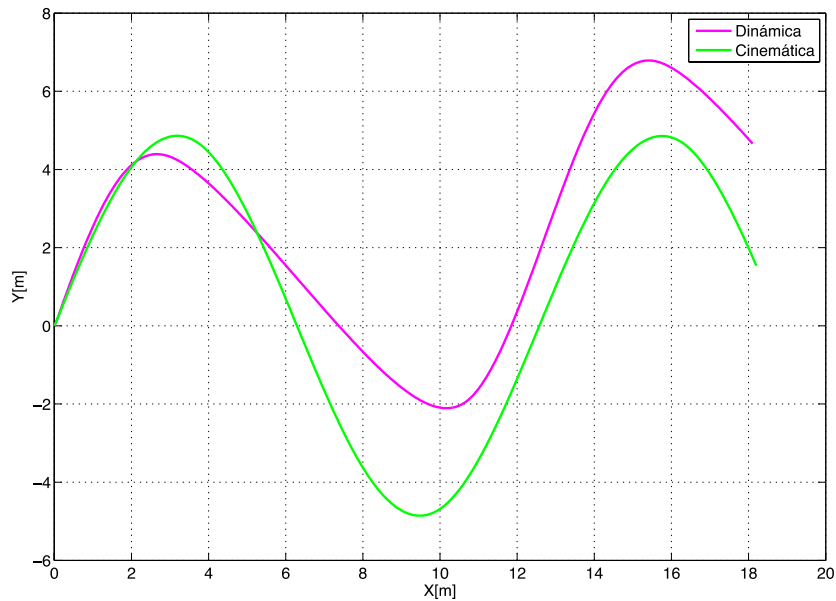


Figura 7-3: Validación 2: Camino del modelo cinemático y dinámico.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 7-3 muestra en color magenta el camino realizado considerando el modelo dinámico y en color verde considerando el modelo cinemático; con una masa mayor del robot móvil se puede observar claramente la diferencia en el comportamiento de ambos modelos ya que el modelo cinemático no toma en cuenta los efectos de la masa e inercia como lo hace el modelo dinámico.

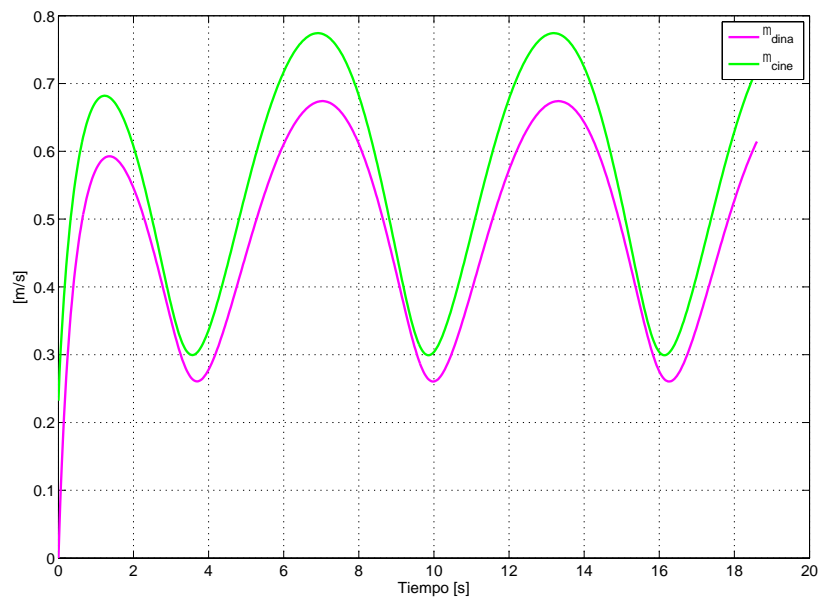


Figura 8-3: Validación 2: Velocidad lineal del modelo cinemático y dinámico.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 8-3 muestra la velocidad lineal de ambos modelos, en donde es evidente la diferencia producto del incremento de la masa del robot móvil.

3.4. Algoritmo de Planificación de caminos.

Como solución al problema de planificación de caminos se ha seleccionado el algoritmo RRT, este algoritmo consiste en la construcción de un árbol de configuraciones que crece de forma randómica explorando diferentes puntos del ambiente de estudio (López García, 2012), como resultado de este algoritmo se tiene una trayectoria desde una configuración inicial q_{ini} hasta una configuración final q_{fin} evadiendo todos los obstáculos del ambiente de trabajo, una vez obtenida la solución al problema se procede a ejecutar un post-procesado que elimina las curvas innecesarias de la trayectoria, obteniendo un camino suavizado a seguir por el robot tipo unicycle.

Para la planificación de caminos y suavizado del mismo el algoritmo completo consta de las siguientes secciones:

- Construcción del ambiente estructurado
- Algoritmo de planificación de caminos
- Consideración de las dimensiones del robot en la planificación de caminos
- Post-procesado o suavizado de caminos

Dentro del algoritmo de planificación se usarán los siguientes conceptos:

q_{ini}	Es la configuración inicial del robot móvil, coordenadas (x, y, ψ)
q_{fin}	La configuración que se desea alcanzar con el robot móvil
q_{rand}	En una configuración aleatoria dentro del espacio de configuraciones
q_{near}	Es la configuración más próxima a q_{rand} de entre las existentes en el árbol
q_{new}	Es la nueva configuración a añadirse al árbol
ε	Longitud de segmento de crecimiento

3.4.1. Construcción del ambiente estructurado

Esta sección consiste en estructurar las dimensiones del espacio de trabajo así como las posiciones, tipos y tamaños de los obstáculos que estarán dentro del ambiente de estudio; se ha

considerado dos tipos de obstáculos, círculos y polígonos de 4 lados, pudiendo crear ambientes con un solo tipo de obstáculo o ambientes mixtos (círculos y polígonos).

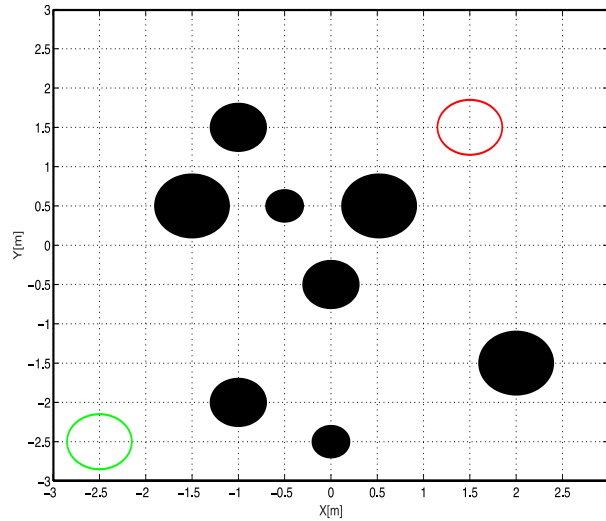
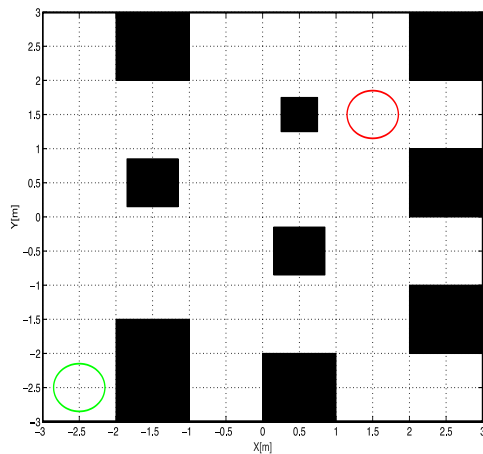
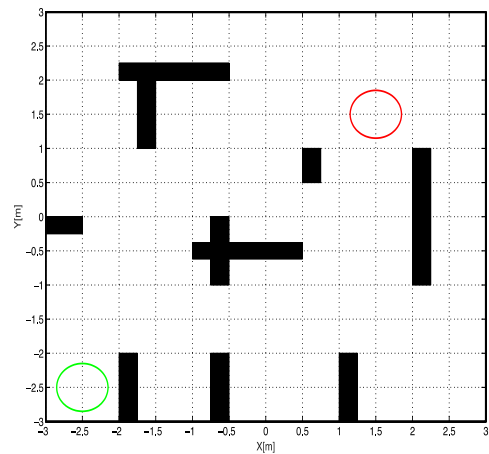


Figura 9-3: Ambiente con obstáculos del tipo círculos
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017

En la Figura 9-3 se observa un ambiente estructurado limitado desde -3 hasta 3 en el eje X y eje Y , dando un total de 6×6 [m], los obstáculos están representados por círculos de color negro, estos pueden localizarse en cualquier punto dentro del espacio de configuraciones y tener un radio variable, los círculos de color verde y rojo representan la configuración inicial q_{ini} y final q_{fin} respectivamente del robot móvil tipo unicyclo.



a) Ambiente con polígonos 1



b) Ambiente con polígonos 2

Figura 10-3: Ambiente con obstáculos cuadrados
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

Las dimensiones de los ambiente de las Figuras 10-3a y 10-3b son las mismas que en la Figura 3-9, 6×6 [m], y los tipos de obstáculos son del tipo polígono de 4 lados, los obstáculos no

necesariamente pueden ser polígonos regulares si no también polígonos irregulares, y estos pueden ubicarse en cualquier punto del espacio de configuraciones; dentro de este ambiente se tiene como en el caso anterior dos círculos, uno de color verde y rojo que representan la configuración inicial q_{ini} y final q_{fin} respectivamente.

3.4.2. Algoritmo de Planificación de caminos

En esta sección se describe los pasos que ejecuta el algoritmo RRT, este tiene como objetivo principal encontrar un camino desde una configuración inicial q_{ini} hasta una configuración final q_{fin} , usando un árbol de configuraciones que explora el espacio de trabajo de forma randómica, el algoritmo usado es el siguiente:

```

FUNCIÓN_RRT( $q_{ini}$ ,  $q_{fin}$ )
1   $Arbol[0] = q_{ini}$ ;
2  while  $k = 1$  hasta  $K_{max}$ 
3     $q_{rand} =$ 
    CONFIGURACION_ALEATORIA();
4    EXTIENDE( $Arbol$ ,  $q_{rand}$ ,  $obst$ );
5    if CONECTA( $q_{new}$ ,  $q_{fin}$ ,  $obst$ ) then
5      devuelve  $Camino$ ;
6    else
7      nueva iteración;
8    end
9  end

```

Figura 11-3: Algoritmo RRT.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En el paso 1 del algoritmo RRT mostrado en la Figura 11-3 agrega al árbol la configuraciones q_{ini} , en el paso 2 empieza el bucle **while**, este tiene un máximo número de iteraciones dado por K_{max} , K_{max} ayuda a detener el algoritmo por si la configuración final es inalcanzable provocando un bucle infinito; dentro del bucle **while** en el paso 3 una configuración aleatoria q_{rand} es generada dentro del espacio de configuraciones con la función CONFIGURACION_ALEATORIA(), en el paso 4 con la función EXTIENDE() el árbol de configuraciones crece de ser posible en dirección a q_{rand} , con la función CONECTA() en el paso 5, el árbol de configuraciones trata intensivamente de llegar hasta la configuración final q_{fin} , de ser alcanzado esta configuración, un camino es devuelto como solución, de no ser alcanzado q_{fin} una nueva iteración del algoritmo RRT se ejecuta

3.4.2.1. Función extiende

La función `EXTIENDE()` es la encargada de hacer crecer el árbol de configuraciones hacia q_{rand} de ser posible, para ello q_{new} es encontrado, este punto es el que se agrega al árbol de configuraciones, la función verifica que no haya colisión en las trayectorias propuestas; la función `EXTIENDE()` se muestra a continuación:

```
EXTIENDE (Arbol,  $q_{rand}$ , obst)
1  if COLISION_OBSTACULO ( $q_{rand}$ , obst) then
2    nueva iteración;
3  end
4   $q_{near}$  = VECINO_PROXIMO(Arbol,  $q_{rand}$ );
5  if NUEVA_CONFIGURACION (obst,  $q_{near}$ ,  $q_{rand}$ )
then
6    AÑADE_CONFIGURACION (Arbol,  $q_{new}$ );
7  end
```

Figura 12-3: Función Extiende.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la línea 1 de la función `EXTIENDE()` mostrado en la Figura 12-3, verifica que el punto aleatorio q_{rand} este fuera del área de los obstáculos, esto se lo hace con la función `COLISION_OBSTACULO()`, de estar el punto aleatorio q_{rand} dentro del área de algún obstáculo este es descartado dando paso a una nueva iteración de la `FUNCIÓN_RRT()`, de no estar q_{rand} dentro del área de ningún obstáculo es aceptado y se procede a encontrar q_{near} que es la configuración más próxima a q_{rand} , esto se hace con la función `VECINO_PROXIMO()`, paso 4 de la función `EXTIENDE()`.

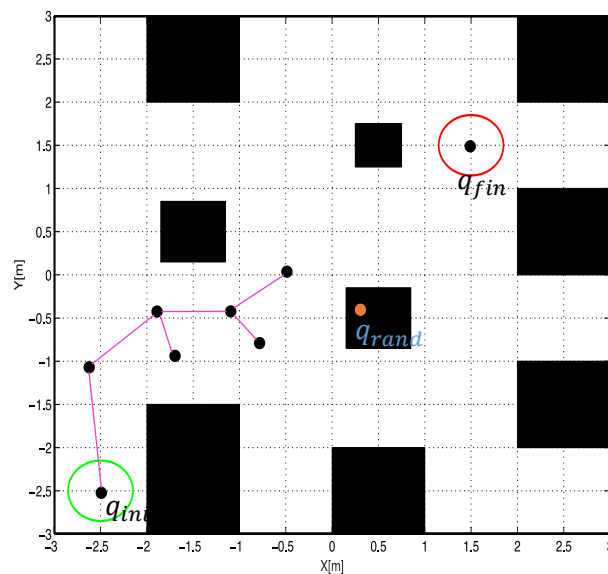


Figura 13-3: Punto aleatorio dentro de obstáculo.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 13-3 se observa como q_{rand} cae dentro del área de un obstáculo, en este caso el punto es descartado dando paso a una nueva iteración de la FUNCIÓN_RRT(), para de esta forma obtener un nuevo punto q_{rand} .

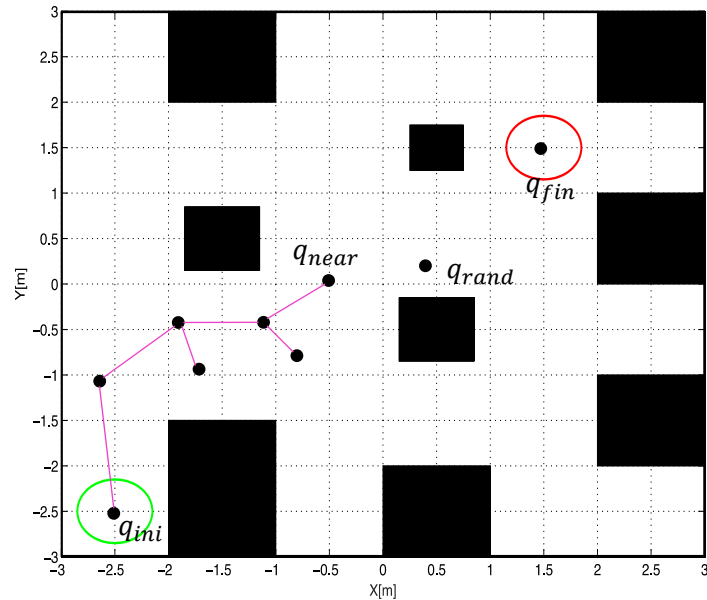


Figura 14-3: Punto aleatorio fuera de obstáculos
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 14-3 muestra como el punto aleatorio cae fuera del área de los obstáculos por lo que el punto es válido, dando paso a la búsqueda de la configuración más próxima con la función VECINO_PROXIMO(), esta función consiste principalmente en buscar en todo el árbol de configuraciones el punto más cercano a q_{rand} , este punto cercano es denominado q_{near}

Encontrado q_{near} se procede a encontrar una nueva configuración q_{new} para agregarse al árbol de configuraciones, esto se hace en el paso 5 con la función NUEVA_CONFIGURACION(), esta función consiste principalmente en trazar una trayectoria recta entre q_{near} y q_{rand} , y verificar que no exista colisión en esta trayectoria, para esto se procede a ir avanzando una distancia ϵ a lo largo de esta e ir comparando en cada avance la existencia o no de alguna colisión.

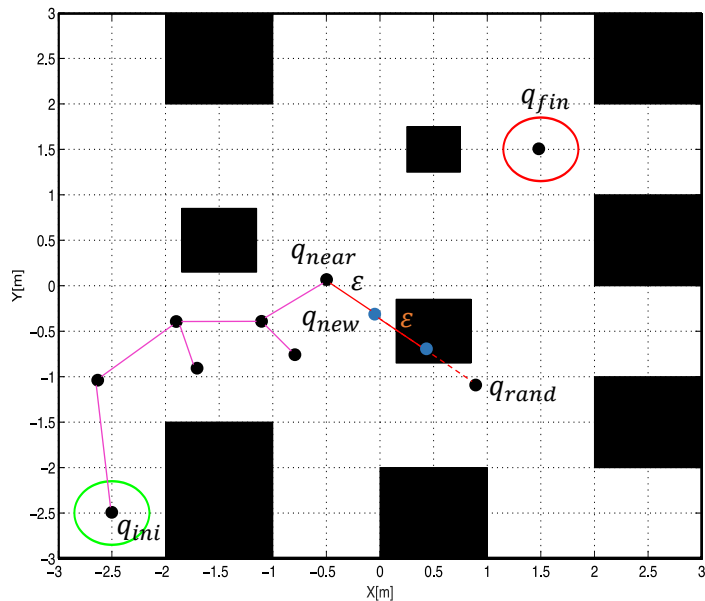


Figura 15-3: Trayectoria $q_{new} - q_{rand}$ en colisión.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 15-3 se observa como la función NUEVA_CONFIGURACION(), va avanzando en pasos de tamaño ϵ , a lo largo de la trayectoria recta entre q_{near} hasta q_{rand} por cada avance se verifica la existencia o no de colisión, en caso de existir colisión en uno de sus avances se escoge como q_{new} al punto anterior a la colisión, de no existir colisión a lo largo de toda la trayectoria q_{rand} pasa a ser q_{new} .

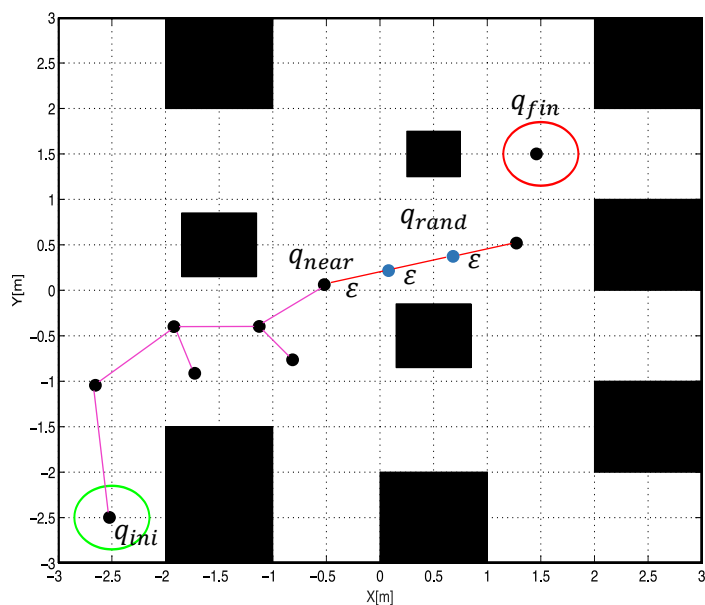


Figura 16-3: q_{rand} igual a q_{new}
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 16-3 se observa como la trayectoria desde q_{near} hasta q_{rand} está libre de colisiones, dando lugar a que el punto q_{rand} sea el nuevo q_{new} y este a su vez sea agregado al árbol de configuraciones.

3.4.2.2. Función conecta

La función CONECTA() mostrado en la Figura 17-3, trata intensivamente de llegar a la meta si la distancia desde q_{new} hasta q_{fin} es menor a $dist$.

```
CONECTA ( $q_{new}, q_{fin}, Obst$ )
1  if (norma ( $q_{new}, q_{fin}$ ) <  $dist$ ) then
2    if NUEVA_CONFIGURACION ( $Obst, q_{new}, q_{fin}$ ) then
3      AÑADE_CONFIGURACION ( $Arbol, q_{fin}$ );
4      devuelve alcanzado;
5    else
6      devuelve rechazado;
7  end
```

Figura 17-3: Función conecta.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La función CONECTA() en el paso 1 empieza por determinar la distancia entre q_{new} hasta q_{fin} y compararla con $dist$, si la distancia encontrada es inferior a esta, la función procede a determina la existencia de colisión a lo largo de la trayectoria recta entre estos dos puntos, esto se hace en el paso 2 con la función NUEVA_CONFIGURACION(), de existir colisión la función devuelve el valor de *rechazado* indicando que la configuración final no se ha alcanzado, de no existir colisión la configuración final es anexada al árbol de configuraciones con la función AÑADE_CONFIGURACION(), y la función devuelve el valor de *alcanzado*.

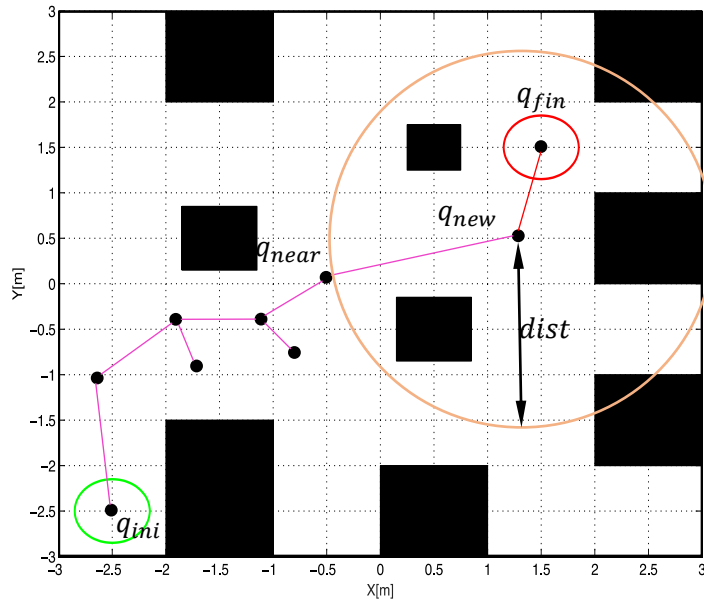


Figura 18-3: q_{fin} cerca de q_{new} .
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 18-3 se puede ver como la función conecta agrega q_{fin} al árbol de configuraciones al estar a una distancia menor a $dist$ de q_{new} , esta función verifica que no exista colisión en la trayectoria recta entre estos dos puntos, esto lo hace de igual forma que en la función `EXTIENDE()`, avanzando una distancia ϵ desde q_{new} hasta q_{fin} y verificando en cada avance la existencia o no de colisión, si en dicha trayectoria no existe colisión q_{fin} es agregado al árbol de configuraciones devolviendo la función el valor de *alcanzado*.

3.4.3. Consideraciones del tamaño del robot unicycle

Es de gran importancia considerar las dimensiones del robot para determinar el espacio de configuraciones libre de colisión, estos puntos del árbol de configuraciones no pueden estar muy cercanas a los obstáculos, de estar estos puntos muy cercanos a los obstáculos el camino encontrado por el algoritmo RRT puede estar muy cercano a ellos provocando una colisión, por esto se considera el radio del robot móvil para la planificación del caminos.

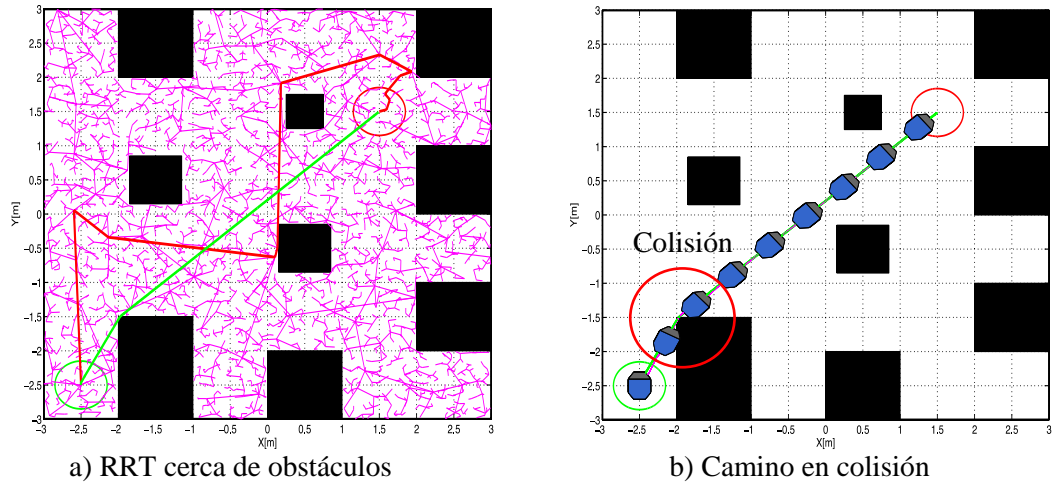


Figura 19-3: Camino cerca de obstáculos.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 19-3a muestra como el algoritmo RRT genera los puntos aleatorios muy cercanos a los obstáculos dando como resultado un camino que los evade pero muy cercano a ellos. En la Figura 19-3b se observa el movimiento estroboscópico del robot móvil tipo unicycle, en el cual se evidencia una colisión.

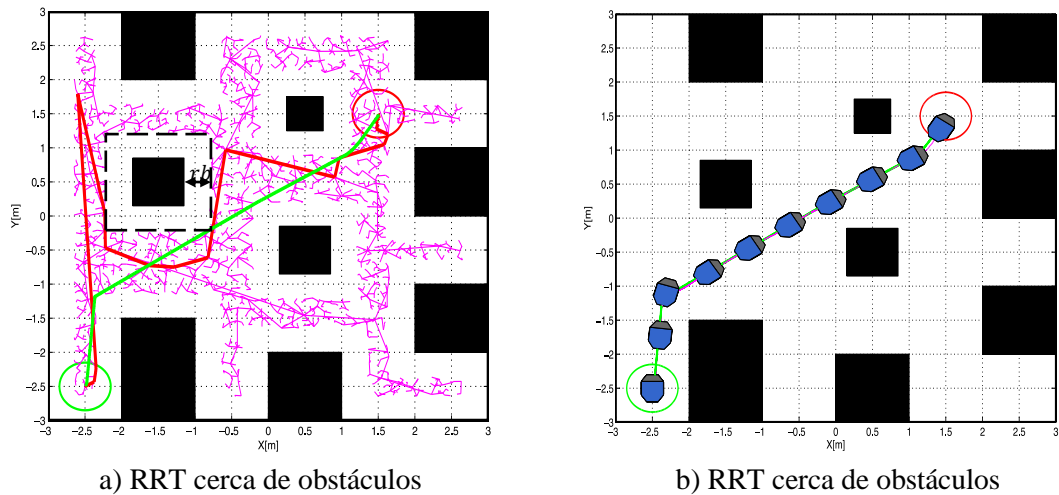


Figura 20-3: Camino lejos de obstáculos.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 20-3a muestra como los nodos del árbol de configuraciones se alejan de los obstáculos, dando como resultado un camino alejado de ellos para evitar colisiones, esta distancia depende del radio del robot unicycle rb , y para efectos de planificación el algoritmo RRT incrementa el área de todos los obstáculos tipo cuadrados, haciéndolos crecer la distancia de sus lados más el radio rb del robot, y si los obstáculos fuesen de tipo círculo, se le suma al radio de los obstáculos el radio del robot rb . La Figura 20-3b muestra como el robot unicycle

recorre el camino planificado sin ninguna colisión ya que los nodos encontrados en el algoritmo RRT están a una distancia prudencial de los obstáculos.

3.4.4. Post-Procesado de caminos.

El camino encontrado por el algoritmo RRT da mucho que desear a la parte de control y en muchos de los casos se vuelve imposibles de ejecutar, de este modo se debe reducir en lo posible las curvas innecesarias y caminos irregulares, para esto se usa el siguiente función mostrado en la Figura 21-3

```
SUAVISADO (obst, P)
1  for k = 2, size (P)
2    L(k) = norma(P(k) - P(k - 1)) + L(k - 1);
3  end
4  m = size (P);
5  while k = 1 hasta maxiter
6    s1 = CONFIGURACION_ALEATORIA()*L(m);
7    s2 = CONFIGURACION_ALEATORIA()*L(m);
8    if CUERDA_VALIDA(s1, s2, obst) then
9      newP = ELIMINA_ARCO(P, g1, g2)
10   else
11     nueva iteración;
12   end
```

Figura 21-3: Algoritmo de post-procesado.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la función de SUAVISADO(), línea 1, con el bucle “**for**” se determina una matriz de distancias con las sumas de todos los segmentos del camino encontrado, este proceso se lo puede ver en la Figura 22-3.

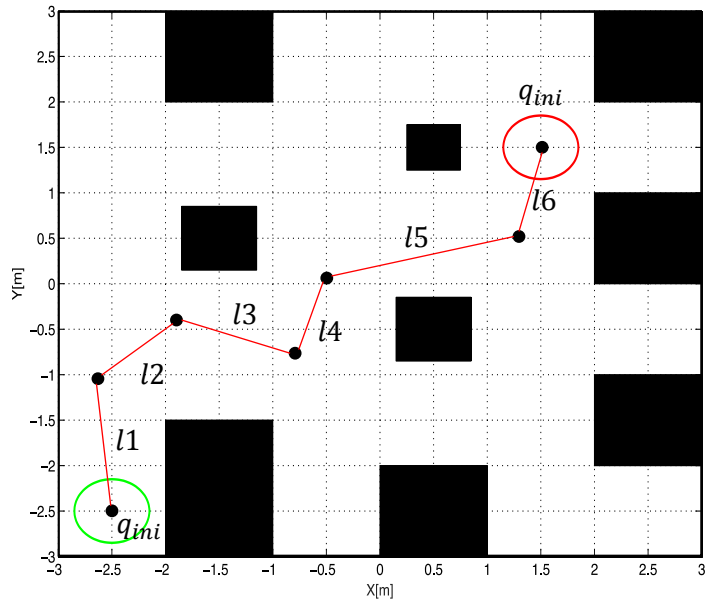


Figura 22-3: Distancias del camino planificado.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura se observa cómo se clasifica las distancias del camino planificado, estas sirven para crear una matriz L , esta matriz para el camino mostrado en la Figura 22-3 queda de la siguiente manera:

$$L = [0, l1, l1 + l2, l1 + l2 + l3, l1 + l2 + l3 + l4, l1 + l2 + l3 + l4 + l5, l1 + l2 + l3 + l4 + l5 + l6] \quad (3.23)$$

En el paso 4 se determina el tamaño de la matriz de puntos que corresponden al camino, en el paso 5 empieza el bucle **while** donde se ejecuta hasta *maxiter*, dentro de este bucle en el paso 6 y 7 se crea dos distancias aleatorias ($s1, s2$), dando como restricción que el valor máximo no sobrepase a la distancia registrada en la última posición del vector L (distancia de la trayectoria desde q_{ini} hasta q_{fin}), adicionalmente en esta sección se verifica y se intercambia de ser el caso para que $s2$ sea mayor que $s1$.

En el paso 8 con la función `CUERDA_VALIDA()`, usando $s1, s2$ se encuentra dos puntos $g1, g2$ para con estos puntos trazar una trayectoria recta entre ellos y reemplazar el camino inicial por esta nueva ruta, adicionalmente en esta función se verifica la existencia o no de colisión para la nueva ruta, de existir colisión se descarta esta y se procede a una nueva iteración de la función de `SUAVISADO()`, de no existir colisión, en el paso 9 con la función `ELIMINA_ARCO` se reemplaza la trayectoria original por el segmento de puntos $g1, g2$

En la Figura 23-3 se extiende el camino inicial como si fuera una línea recta, para mostrar de mejor manera la forma de encontrar $g1$ y $g2$

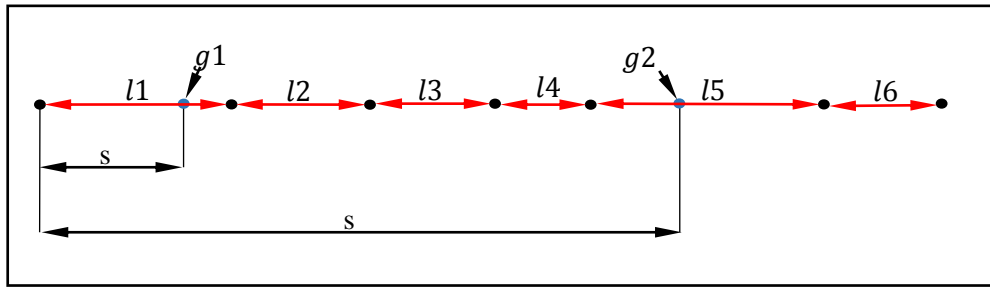


Figura 23-3: Puntos $g1$ y $g2$.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En esta Figura se muestra como $s1, s2$ ayuda a encontrar los puntos $g1, g2$; considerando que $s1, s2$ son distancias aleatorias, y usando los segmentos del camino inicial se encuentra una razón para encontrar los puntos $g1, g2$, con la siguiente ecuación.

$$\begin{bmatrix} g_x \\ g_y \end{bmatrix} = (1 - r_1) \begin{bmatrix} p1_x \\ p1_y \end{bmatrix} + r_1 \begin{bmatrix} p2_x \\ p2_y \end{bmatrix} \quad (3.24)$$

Encontrados los puntos $g1, g2$ se procede a verificar si existe colisión en la trayectoria recta entre estos dos puntos, de no existir colisión esta nueva ruta reemplaza la original, de existir colisión se busca una nueva posible ruta descartando la anterior.

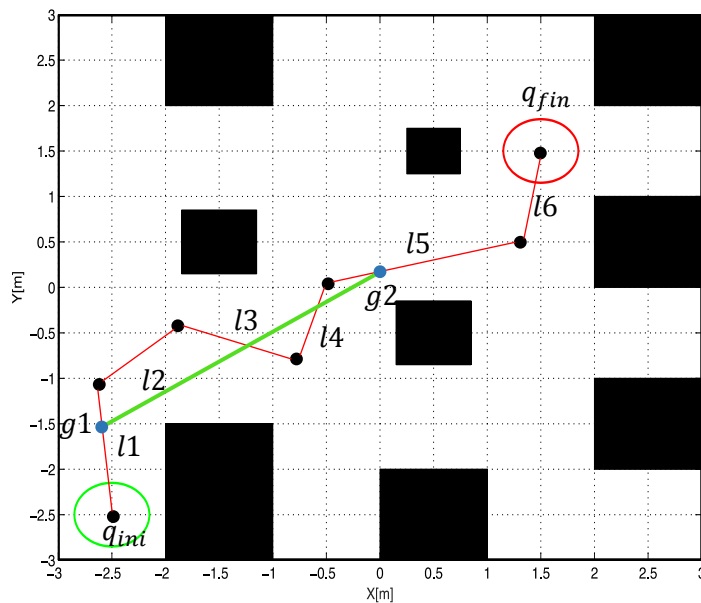


Figura 24-3: Cuerda válida.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 24-3 muestra como la función de suavizado crea un nuevo camino más directo que el original (segmento color verde), cabe recalcar que este segmento es válido siempre y cuando no exista ninguna colisión.

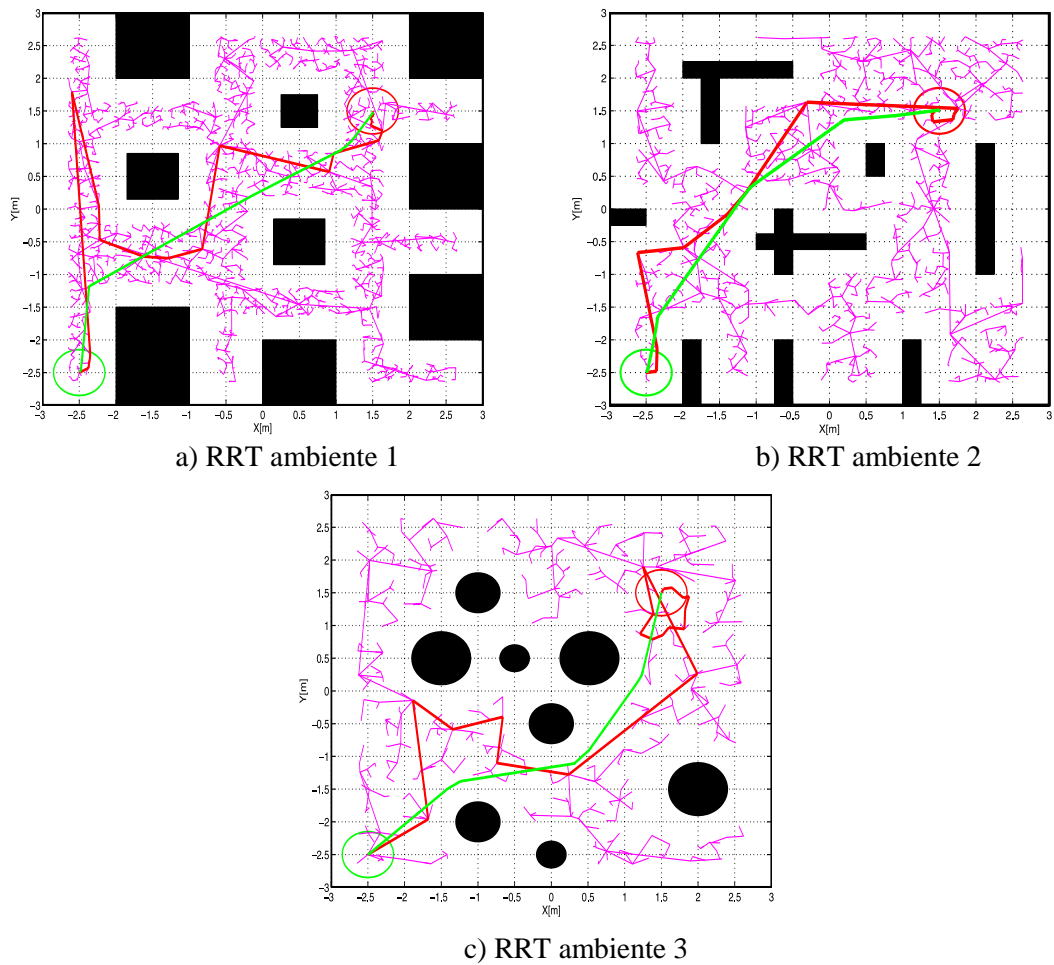


Figura 25-3: Camino suavizado.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 25-3 se puede observar el resultado de la planificación de caminos para 3 escenarios diferentes, en todos los casos el color magenta representa el árbol de configuraciones, color rojo el camino sin post-procesado y en color verde el camino aplicado un post-procesado.

3.5. Control del robot móvil tipo unicycle

Para resolver el problema de seguimiento de caminos se toma en consideración los elementos mostrados en la Figura 26-3, esta ley de control debe reducir a cero la distancia ρ entre la posición del robot móvil y el camino deseado, además debe reducir a cero el ángulo formado entre la velocidad del robot móvil y la tangente al camino.

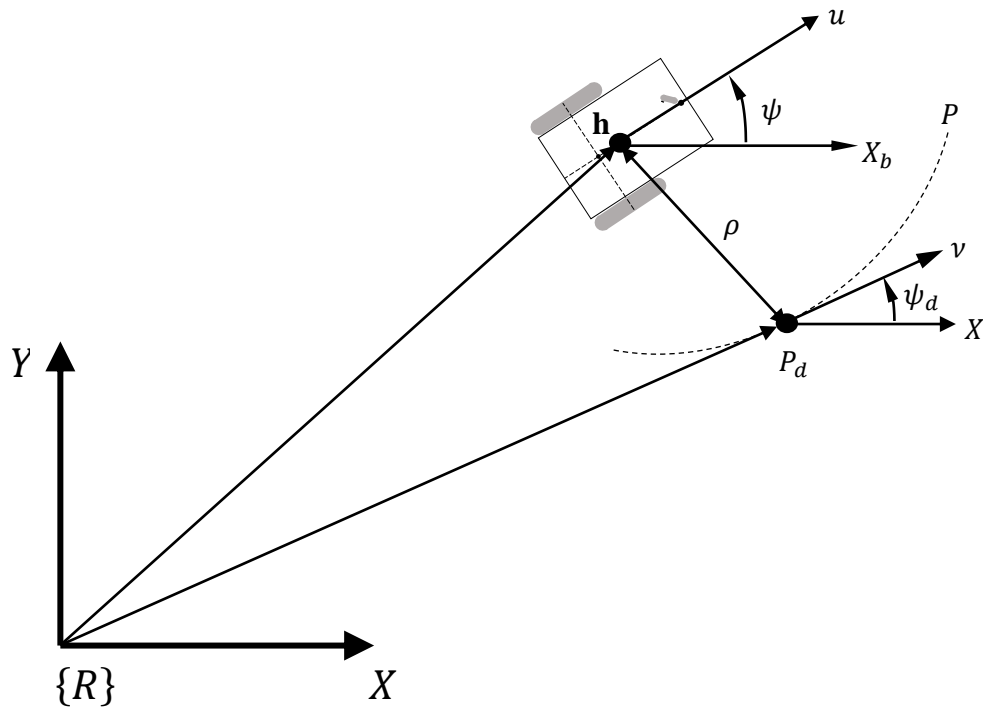


Figura 26-3: Variables para el seguimiento de caminos.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 26-3 se puede ver las variables de interés que intervienen el control de seguimiento de caminos, donde:

- P Representa el camino a ser seguido por el robot móvil
- P_d Es un punto en el camino \mathcal{P}
- $\mathbf{h} = [x \ y]^T$ Es la posición del robot móvil respecto al sistema de coordenadas global
- ρ Distancia entre el robot móvil y el camino
- v Es el vector de velocidad deseada sobre el camino
- ψ_d Angulo formado entre el eje X y la tangente del camino P

El error de orientación podemos obtener de la resta $\tilde{\psi} = \psi_d - \psi$, el control para el seguimiento de caminos de robots móviles consiste en encontrar una ley de control de tipo:

$$v(t) = k(s, \rho, \tilde{\psi}, v(s)) \quad (3.25)$$

En donde la distancia ρ puede ser representada como la norma euclidiana del error $\tilde{\mathbf{h}}$, ósea $\rho = d(P_d, \mathbf{h}) = \tilde{\mathbf{h}}$, donde:

$$\tilde{\mathbf{h}} = P_d - \mathbf{h} \quad (3.26)$$

También se puede determinar las componentes ortogonales del vector de velocidad deseado en el punto P_d :

$$\begin{aligned} v_{xpd} &= v \cos(\psi_d) \\ v_{ypd} &= v \operatorname{sen}(\psi_d) \end{aligned} \quad (3.27)$$

El vector de la velocidad deseado en el punto P_d está definido por $v_{pd} = [v_{xpd}, v_{ypd}]^T$, por consiguiente la posición y velocidad desea del robot móvil en el punto sobre el camino P son determinados como P_d y v_{pd} respectivamente. Como controlador en esta sección se propone un esquema de control basado en el modelo cinemático del robo móvil tipo unicycle, adicional a esto se considera la saturación de los comandos de velocidad del robot móvil.

Se considera a P_d como punto más cercano entre la posición del robot móvil \mathbf{h} y el camino P , por lo que se tiene que $P_d = \mathbf{h}_d$ donde \mathbf{h}_d es la posición deseada del robot móvil sobre el punto P_d . Consecuentemente si $\lim_{t \rightarrow \infty} \tilde{\mathbf{h}}(t) = 0$ entonces $\lim_{t \rightarrow \infty} \rho(t) = 0$. Consecuentemente esta condición implica que $\lim_{t \rightarrow \infty} \tilde{\psi}(t) = 0$ debido a la no holonomía del robo móvil.

3.5.1. Diseño del sistema de control de robot móvil

En esta sección se propone un esquema de control cinemático para el seguimiento de caminos del robot móvil tipo unicycle.

Para el diseño del controlador cinemático consideramos dos casos:

- 1) Una vez que el robot móvil este sobre el camino, este debe seguir una velocidad deseada
- 2) En todo instante el robot móvil debe mantenerse a una velocidad deseada.

3.5.1.1. Control cinemático caso 1

Se considera el modelo cinemático del robot móvil tipo unicycle con $a \neq 0$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\psi & -a\text{sen}\psi \\ \text{sen}\psi & a\cos\psi \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}$$

$$\dot{\mathbf{h}}(t) = \mathbf{J}(\psi)\mathbf{v}(t) \quad (3.28)$$

Considerado la cinemática inversa de la ecuación. (3.28) se tiene:

$$\mathbf{v}(t) = \begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\psi & \text{sen}\psi \\ -\frac{1}{a}\text{sen}\psi & \frac{1}{a}\cos\psi \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (3.29)$$

Reescribiendo (3.29) en forma compacta, se tiene

$$\mathbf{v}(t) = \mathbf{J}^{-1}(\psi)\dot{\mathbf{h}}(t) \quad (3.30)$$

Donde $\mathbf{J}^{-1}(\psi)$ es la inversa de la matriz Jacobiana definida en (3.28) y $\mathbf{v}(t)$ es el vector de velocidad del robot móvil.

De las ecuaciones obtenidas anteriormente se propone la siguiente ley de control

$$\mathbf{v}_c = \begin{bmatrix} u_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} \cos\psi & \text{sen}\psi \\ -\frac{1}{a}\text{sen}\psi & \frac{1}{a}\cos\psi \end{bmatrix} \begin{bmatrix} v\cos\psi_d + l_x \tanh\left(\frac{k_x}{l_x}\bar{x}\right) \\ v\text{sen}\psi_d + l_y \tanh\left(\frac{k_y}{l_y}\bar{y}\right) \end{bmatrix} \quad (3.31)$$

Donde el error de la posición respecto al camino está definida por robot $\tilde{\mathbf{h}} = [\bar{x} \quad \bar{y}]^T$, en donde $\bar{x} = x_d - x$ y $\bar{y} = y_d - y$, ψ_d es el ángulo deseado formando entre el eje X y la tangente a P en el punto P_d , mientras que l_x, l_y, k_x y k_y son constantes positivas de ganancia y v es el modulo de la velocidad deseada sobre el camino, $\tanh(\cdot)$ ha sido incorporado para limitar las velocidades de referencia evitando la saturación de las velocidades del vehículo.

Escribiendo la ecuación (3.31) en forma compacta quedaría:

$$\mathbf{v}_c = \mathbf{J}^{-1}\left(\mathbf{v}_d + \mathbf{L}\tanh(\mathbf{L}^{-1}\mathbf{K}\tilde{\mathbf{h}})\right) \quad (3.32)$$

donde, \mathbf{v}_d representa la velocidad deseada sobre el camino; $\mathbf{L} = \text{diag}(l_x, l_y)$, $\mathbf{K} = \text{diag}(k_x, k_y)$ y $\tanh(\mathbf{L}^{-1}\mathbf{K}\tilde{\mathbf{h}})$ es una operación componente a componente.

Para el análisis de estabilidad se asume momentáneamente seguimiento perfecto de velocidad $\mathbf{v}_c \equiv \mathbf{v}$, o sea, se considera que $u \equiv u_c$ y $\omega \equiv \omega_c$, esto implica que los efectos dinámicos del robot móvil son ignorados. De este modo sustituyendo (3.32) en (3.30) se tiene:

$$(\mathbf{v}_d - \dot{\mathbf{h}}) + \mathbf{L} \tanh(\mathbf{L}^{-1} \mathbf{K} \tilde{\mathbf{h}}) = 0 \quad (3.33)$$

Ahora, definiendo a γ como la diferencia entre $\dot{\mathbf{h}}$ y \mathbf{v}_d se tiene:

$$\gamma = \dot{\mathbf{h}}_d - \mathbf{v}_d \quad (3.34)$$

Y recordemos que $\dot{\tilde{\mathbf{h}}} = \dot{\mathbf{h}}_d - \dot{\mathbf{h}}$, la ecuación de lazo cerrado (3.30) puede ser escrita como:

$$\dot{\tilde{\mathbf{h}}} + \mathbf{L} \tanh(\mathbf{L}^{-1} \mathbf{K} \tilde{\mathbf{h}}) = \gamma \quad (3.35)$$

Nota 1.- \mathbf{v}_p Es colineal a $\dot{\mathbf{h}}_d$, por lo tanto γ , \mathbf{v}_p y $\dot{\mathbf{h}}_d$ son vectores colineales.

Se considera la siguiente función candidata de Lyapunov para analizar la estabilidad del sistema

$$V(\tilde{\mathbf{h}}) = \frac{1}{2} \tilde{\mathbf{h}}^T \tilde{\mathbf{h}} \quad (3.36)$$

Donde su derivada temporal en las trayectorias del sistema es:

$$\begin{aligned} \dot{V}(\tilde{\mathbf{h}}) &= \tilde{\mathbf{h}}^T \dot{\tilde{\mathbf{h}}} \\ \dot{V}(\tilde{\mathbf{h}}) &= \tilde{\mathbf{h}}^T \gamma - \tilde{\mathbf{h}}^T \mathbf{L} \tanh(\mathbf{L}^{-1} \mathbf{K} \tilde{\mathbf{h}}) \end{aligned} \quad (3.36)$$

Una condición suficiente para que $\dot{V}(\tilde{\mathbf{h}})$ sea definida negativa es:

$$|\tilde{\mathbf{h}}^T \mathbf{L} \tanh(\mathbf{L}^{-1} \mathbf{K} \tilde{\mathbf{h}})| > |\tilde{\mathbf{h}}^T \gamma| \quad (3.38)$$

Para valores grandes de $\tilde{\mathbf{h}}$, se puede considerar que $\mathbf{L} \tanh(\mathbf{L}^{-1} \mathbf{K} \tilde{\mathbf{h}}) = \mathbf{L}$. La \dot{V} es definida negativa solo si:

$$\|\mathbf{L}\| > \|\gamma\| \quad (3.39)$$

Por lo tanto, disminuye los errores $\tilde{\mathbf{h}}$.

Ahora, para valores pequeños de $\tilde{\mathbf{h}}$, se puede considerar que $L \tanh(L^{-1}K\tilde{\mathbf{h}}) \approx K\tilde{\mathbf{h}}$ donde (3.36) se puede escribir como:

$$\|\tilde{\mathbf{h}}\| > \frac{\|\gamma\|}{\lambda_{\min}(\mathbf{K})} \quad (3.40)$$

Nota 2. Como se indicó anteriormente para pequeños errores de $\tilde{\mathbf{h}}$, $L \tanh(L^{-1}K\tilde{\mathbf{h}}) \approx K\tilde{\mathbf{h}}$. Por lo tanto, la ecuación de lazo cerrado (3.32) se escribe como:

$$\dot{\tilde{\mathbf{h}}} + \mathbf{K}\tilde{\mathbf{h}} = \gamma \quad (3.41)$$

Donde su transformada de Laplace es:

$$\hat{\tilde{\mathbf{h}}}(s) = (s\mathbf{I} + \mathbf{K})^{-1} = \gamma(s) \quad (3.42)$$

Por lo tanto, la dirección del vector de errores de control $\tilde{\mathbf{h}}$ tiende a la dirección del vector de error de velocidad γ . Entonces, para valores finitos de $\tilde{\mathbf{h}}$, su dirección es normal a ν_{Pd} (por el criterio de mínima distancia entre \mathbf{h} y el camino P) y por la definición de γ (ver Nota 1); $\tilde{\mathbf{h}}$ tiene que ser cero. Ahora se puede concluir $\tilde{\mathbf{h}}(t) \rightarrow 0$ asintóticamente, entonces $\rho(t) \rightarrow 0$ con $t \rightarrow \infty$. Por otra parte, debido a la restricción no holonómica, la orientación del robot móvil una vez alcanzada el camino estará definida por el ángulo formado entre el eje x y la tangente a P en P_d , o sea, por el ángulo ψ_d . Por lo que se puede concluir, que como $\tilde{\mathbf{h}} \rightarrow 0$ cuando $t \rightarrow \infty$, entonces $\tilde{\psi} \rightarrow 0$ con $t \rightarrow \infty$. Con lo que se cumpliría el objetivo de control para el seguimiento de caminos de un robot móvil no holonómico.

3.5.1.2. Control cinemático caso 2

Para este ítem en el que se desea mantener en todo instante la velocidad deseada sobre el camino, se realiza un análisis similar al Caso 1. La ley de control propuesta para el Caso 2, queda definida como:

$$v_c = \begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix} = \begin{bmatrix} v \\ \frac{v}{a} \left[\frac{l_x}{a} \sin(\tilde{\psi}) - \frac{l_x}{a} \sin(\psi) \tanh\left(\frac{k_x}{l_x} \tilde{x}\right) + \frac{l_y}{a} \cos(\psi) \tanh\left(\frac{k_y}{l_y} \tilde{y}\right) \right] \end{bmatrix} \quad (3.43)$$

Cabe recalcar que la única diferencia entre (3.31) y (3.43), es la velocidad lineal de referencia, en (3.43) es siempre igual a la velocidad deseada sobre el camino, o sea, $u_c = v$, mientras que la ω_c de (3.31) y (3.43) son iguales.

Para diferenciar entre estos dos casos se propone simulaciones, en donde se ha implementado las leyes de control antes mencionadas.

3.5.2. Simulación caso 1: Velocidad deseada sobre camino

En la Figura 27-3 se muestra el movimiento estroboscópico y la evolución del seguimiento del robot móvil, este parte con un error de 2 [m] respecto al camino a seguir.

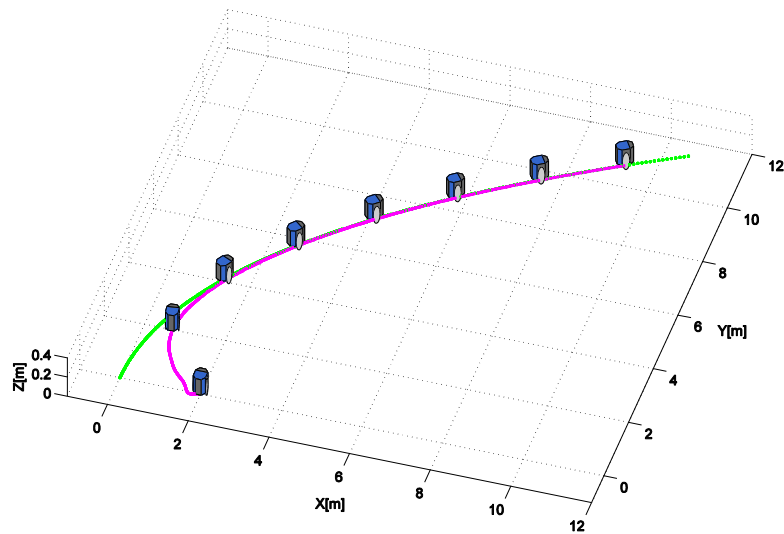


Figura 27-3: Controlador caso 1: Movimiento estroboscópico.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 28-3 muestra el error del robot móvil durante todo el camino, este parte con un error de 2 [m] para luego tender asintóticamente a cero.

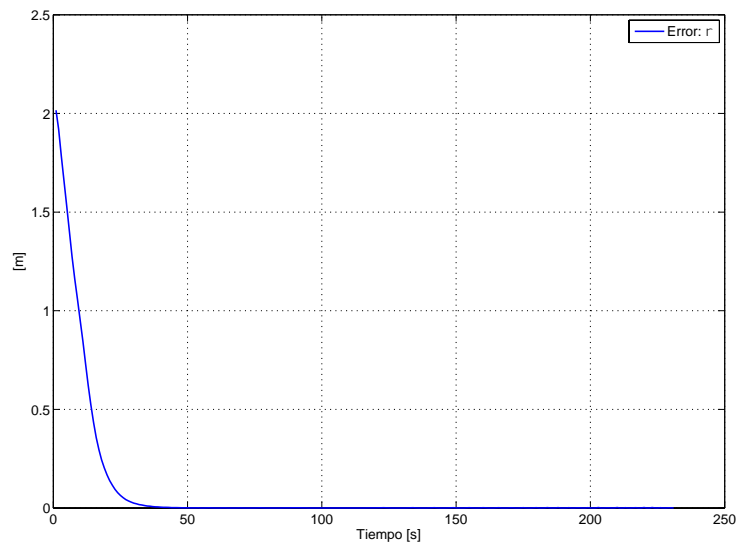


Figura 28-3: Controlador Caso 1: Error de control.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 29-3 representa las velocidades de control dado por (3.31) $[u_{ref} \ \omega_{ref}]^T$.

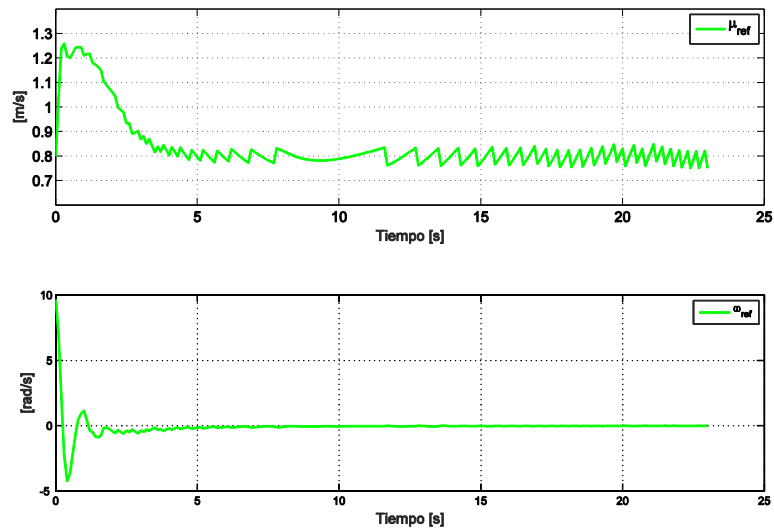


Figura 29-3: Controlador Caso1: Velocidades de referencia.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 30-3 muestra la velocidad de referencia u_{ref} y la velocidad real del robot movil a lo largo del camino.

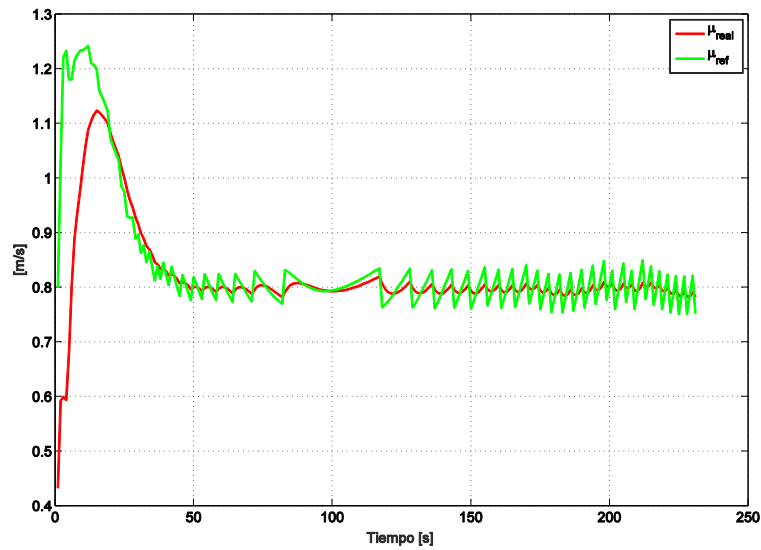


Figura 30-3: Controlador Caso 1: Velocidad real y de referencia
 Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

3.5.3. Simulación caso 2: Velocidad deseada en todo instante

La Figura 31-3 muestra el movimiento estroboscópico del robot móvil y la evolución del seguimiento implementado el controlador propuesto en (3.43), el robot móvil parte con 2 [m] de error respecto camino a seguir.

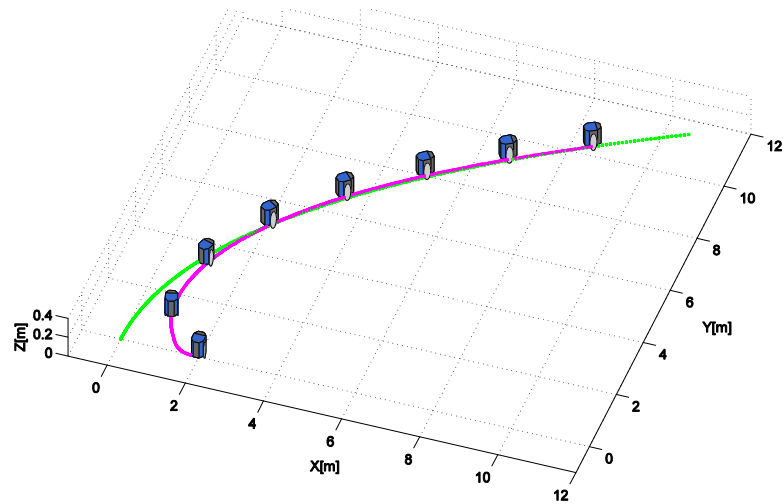


Figura 31-3: Controlador caso 2: Movimiento estroboscópico
 Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

El error de control a lo largo del camino se muestra en la Figura 32-3, este parte con un error de 2 [m]

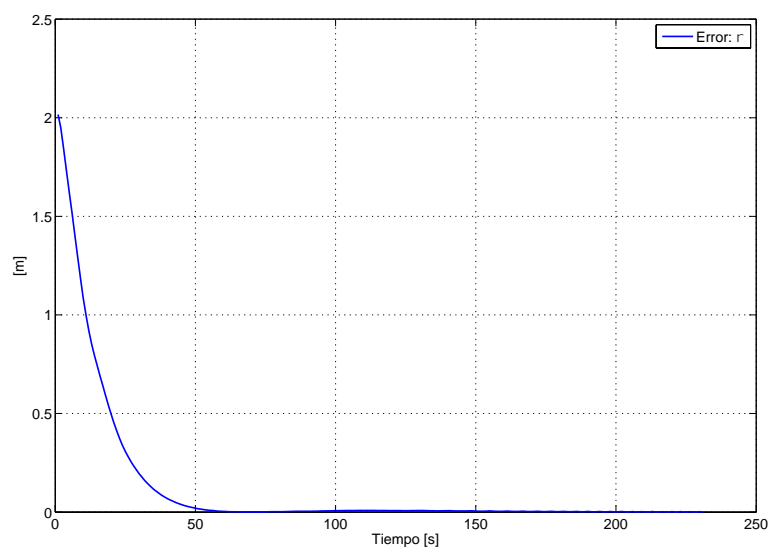


Figura 32-3: Controlador Caso 2: Error de control
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 33-3 representa las velocidades de control dado por (3.43) $[u_{ref} \omega_{ref}]^T$, en el que se puede evidenciar como la velocidad de referencia es igual a la velocidad deseada en todo instante.

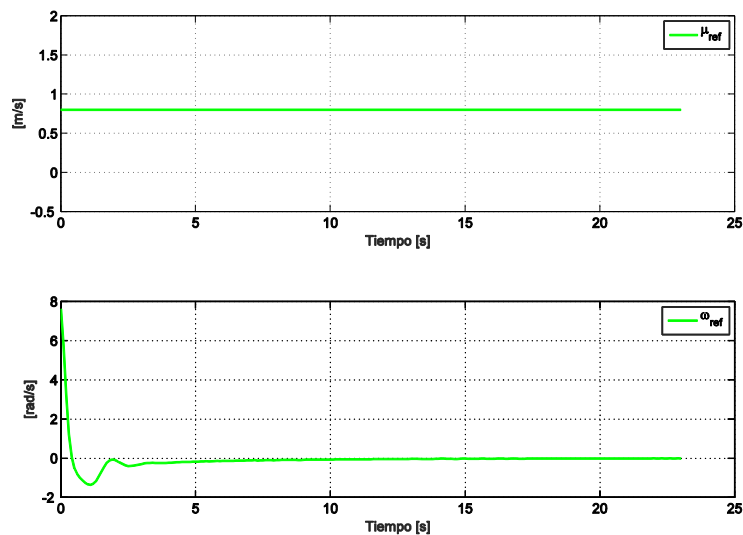


Figura 33-3: Controlador Caso 2: Velocidades de referencia
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 34-3 muestra la velocidad de referencia u_{ref} y la velocidad real del robot móvil a lo largo del camino. u_{real} .

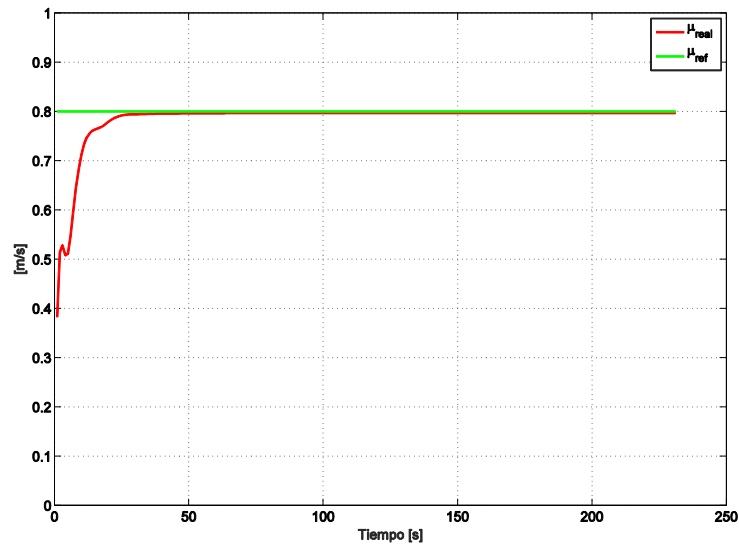


Figura 34-3: Controlador Caso 2: Velocidad real y de referencia
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

Ambos controladores siguen el camino de referencia con errores bajos, en el caso 1 la velocidad u_{ref} es variante, y solo es igual a la velocidad deseada cuando el robot móvil está sobre el camino, mientras que en el caso 2 la velocidad de referencia siempre es igual a la velocidad deseada, este o no el robot móvil sobre el camino.

Asemejándose a vehículos reales en movimiento en donde la velocidad deseada varía conforme el error, es decir a medida que el vehículo sale del camino deseado, este frena levemente para volver hacia él de manera eficiente evitando deslizamientos y oscilaciones, se plantea lo siguiente:

$$v_d(t) = \frac{v_{max}}{1 + k_v \rho} \text{ [m/s]} \tag{3.44}$$

En donde v_{max} representa la velocidad máxima del robot móvil, k_v es una constante de ganancia y ρ es el error en [m] a lo largo del camino, como se puede observar de (3.41) cuando el error es cero la velocidad deseada del robot móvil es v_{max} y a medida que este error crezca la velocidad deseada se reduce en proporción a k_v . Para mostrar de mejor manera los efectos de esta modificación se plantea el caso 3:

3.5.4. Simulación caso 3: Velocidad deseada en función del error ρ

La Figura 35-3 muestra el movimiento estroboscópico considerando el controlador mostrado en (3.31) y con una velocidad deseada por (3.44).

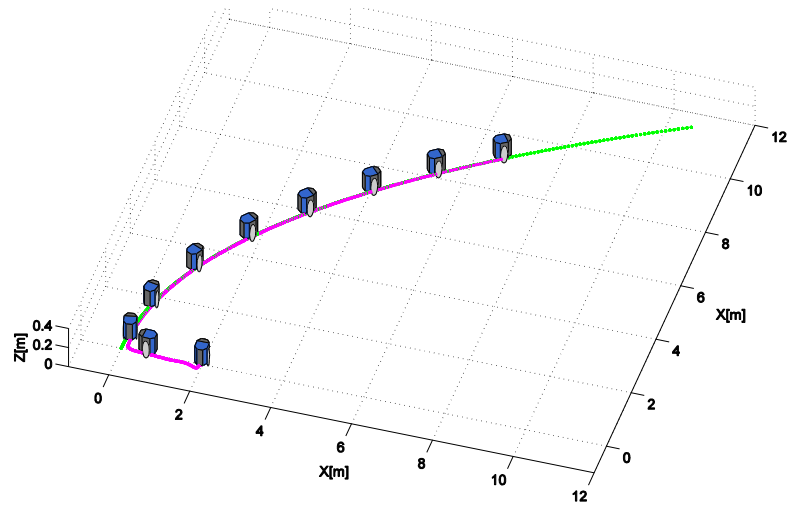


Figura 35-3: Controlador caso 3: Movimiento estroboscópico
 Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En las Figuras 36-3 hasta 38-3 se muestra los resultados del controlador en donde la velocidad deseada está en función del error. En la Figura 36-3 se muestra el error de seguimiento en [m], la Figura 37-3 muestra las velocidades de referencia resultados de las acciones de control y en la Figura 38-3 se muestra la velocidad real del robot móvil frente a la velocidad de referencia.

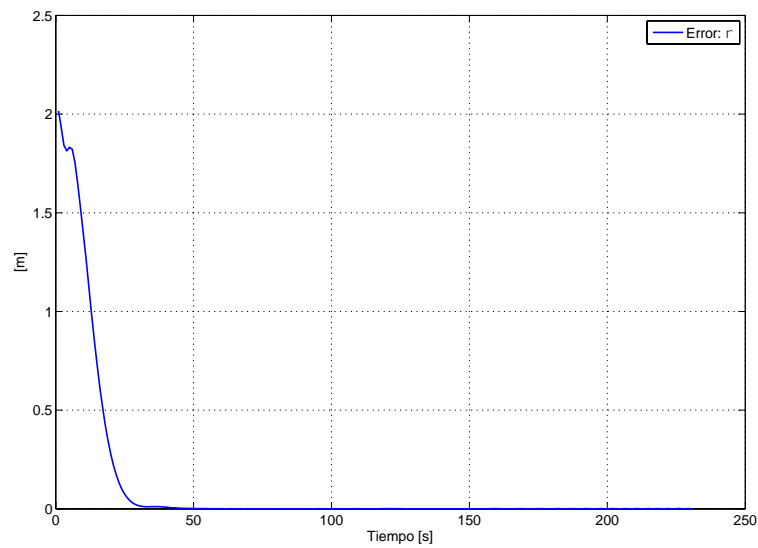


Figura 36-3: Controlador Caso 3: Error de control
 Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

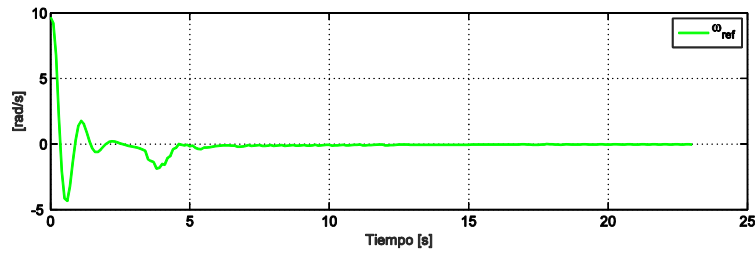
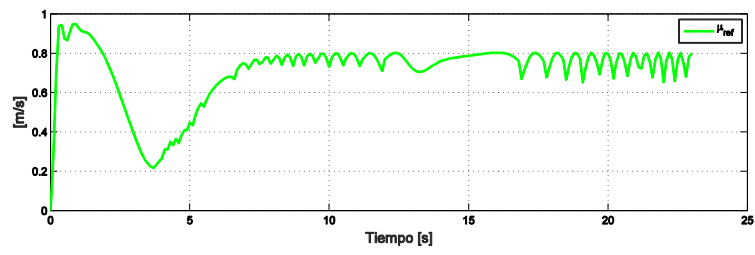


Figura 37-3: Controlador Caso 3: Velocidades de referencia
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

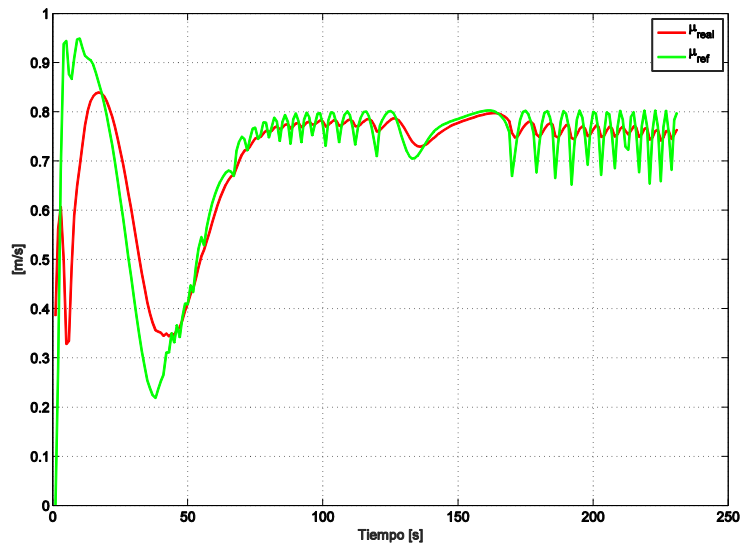


Figura 38-3: Velocidad real y de referencia: Controlador Caso 3.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

En esta sección revisaremos y analizaremos los resultados del algoritmo de planificación de caminos y los resultados del esquema de control implementados, caso 3 de la sección 3.5.4; este por ser el que más se asemeja a un vehículo real en movimiento; cabe mencionar que los resultados de simulación y experimentales difieren en la solución del camino encontrado ya que este es determinado de forma randómica, y aunque parezcan iguales serán distintos en cada caso.

Las pruebas experimentales serán realizadas con el robot móvil construido en el proyecto de investigación “Tele-operación bilateral cooperativo de múltiples manipuladores móviles”, ejecutado por la Escuela Superior Politécnica de Chimborazo, Universidad Técnica de Ambato, Escuela Politécnica Nacional y la Universidad de las Fuerzas Armadas, este robot móvil consta con las siguientes características:

- Plataforma unicycle.
- Transmisión Polea-Banda
- Tracción 4 ruedas con dos motores Dinamixel Pro
- Dimensionamiento 600 X 360 X 200 mm
- Estructura rígida, liviana y estable.

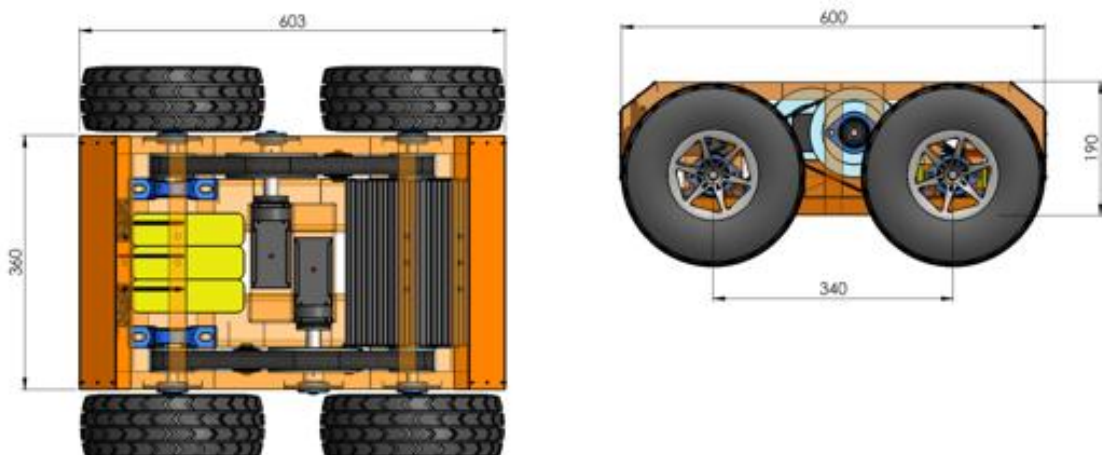


Figura 1-4: Robot móvil tipo unicycle.

Fuente: Proyecto de investigación “Tele-operación bilateral cooperativo de múltiples manipuladores móviles

La parte electrónica del robot móvil consta de varios elementos como son: protecciones, alimentación de los motores, medición de corriente y presencia de las baterías, reductor de voltaje para la alimentación de varios dispositivos, visualización de la información y controlador del sistema electrónico.

4.1. Resultados de Simulación.

4.1.1. Simulación 1

En esta simulación se revisarán los resultados del algoritmo de planificación y esquema de control dado en (3.31) y velocidad deseada dada por (3.44).

La configuración inicial del robot móvil está en $x = -2.5$, $y = -2.5$, y $\psi = 90$, la configuración final en: $x = 1.5$, $y = 1.5$.

La Figura 2-4 muestra el resultado del algoritmo RRT, en color magenta se puede ver los nodos del árbol de configuraciones, en donde se observa como el algoritmo RRT realiza una búsqueda por todo el espacio de configuraciones libre de colisiones, en color rojo se observa el camino inicial encontrado por el algoritmo, este posee demasiadas curvas pronunciadas e innecesarias difíciles de ser ejecutadas por el robot móvil, en color verde se observa el camino luego de haber aplicado un post-procesado.

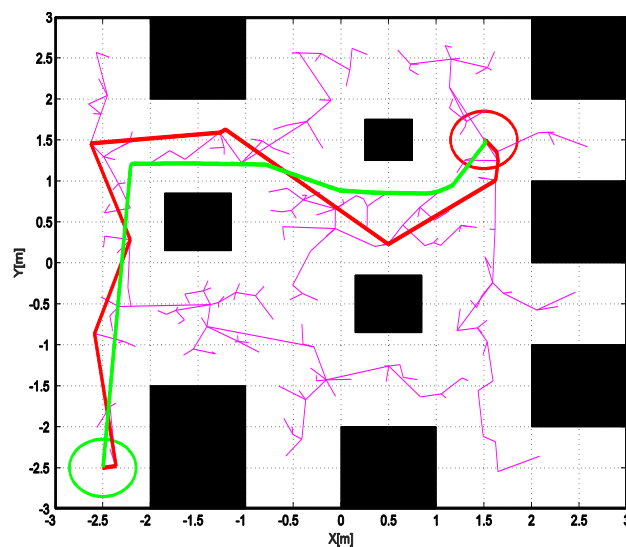


Figura 2-4: Simulación 1: Resultado del algoritmo RRT.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 3-4 se observa en la primera línea de comandos de Matlab, los nodos que componen el árbol de configuraciones, este está compuesto por 248 nodos, la distancia del penúltimo nodo antes de alcanzar la configuración final es 0 [m], el algoritmo realizó 634 iteraciones hasta alcanzar la configuración final; en la segunda fila se observa la cantidad de nodos que componen el camino inicial obtenido por el algoritmo RRT, este está compuesto por 13 nodos, en la tercera fila se observa el número de nodos del camino luego de aplicar un post-procesado, este está compuesto de 482 nodos, y el tiempo total de ejecución del algoritmo RRT es de 15.4 segundos.

```
Nodos Arbol: 248, Distancia: 0.0, Iteraciones: 634/15000
Nodos Camino RRT: 13, Iteraciones RRT: 634/15000
Nodos Camino Suavizado: 482, tiempo Total RRT: 15.4
```

Figura 3-4: Simulación 1: Numero de nodos del algoritmo RRT.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la siguiente figura se observa el movimiento estroboscópico del robot móvil y camino deseado versus camino ejecutado.

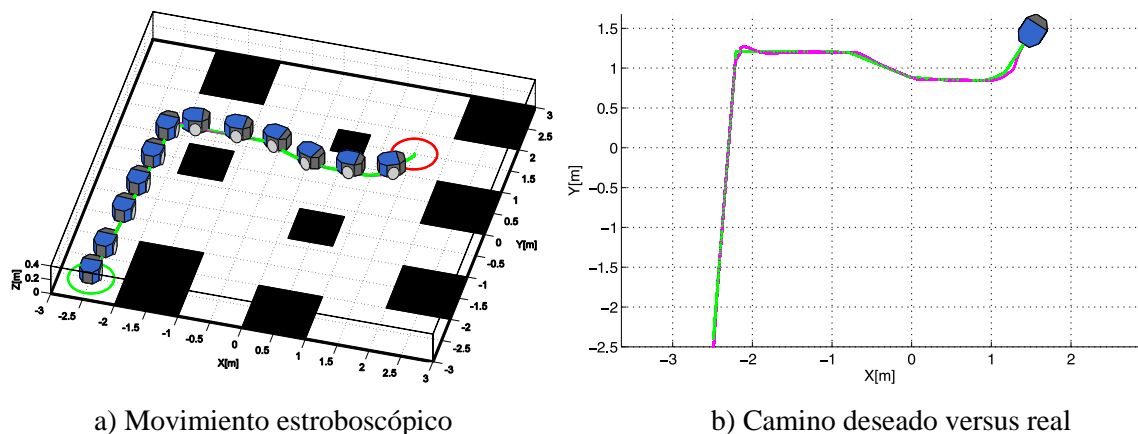


Figura 4-4: Simulación 1: Camino realizado por el robot móvil.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 4-4a se observa el movimiento estroboscópico del robot móvil realizado con valores reales, en donde se evidencia como el robot móvil llega exitosamente a la configuración final sin efectuar ninguna colisión, en la Figura 4-4b se observa en color verde el camino deseado resultado del algoritmo RRT, en color magenta el camino realizado por el robot móvil, en esta grafica se evidencia como el robot sigue fielmente al camino deseado con un error bajo.

En la Figura 5-4 se observa el error $\rho(t)$, a lo largo del camino, este error es el resultado de la resta en cada punto del camino deseado menos el camino ejecutado por el robot móvil; como se

observa este error no sobrepasa los 0.02 [m], manteniéndose en su mayoría en 0.005 [m], de este modo comprobamos la robustez del algoritmo de control.

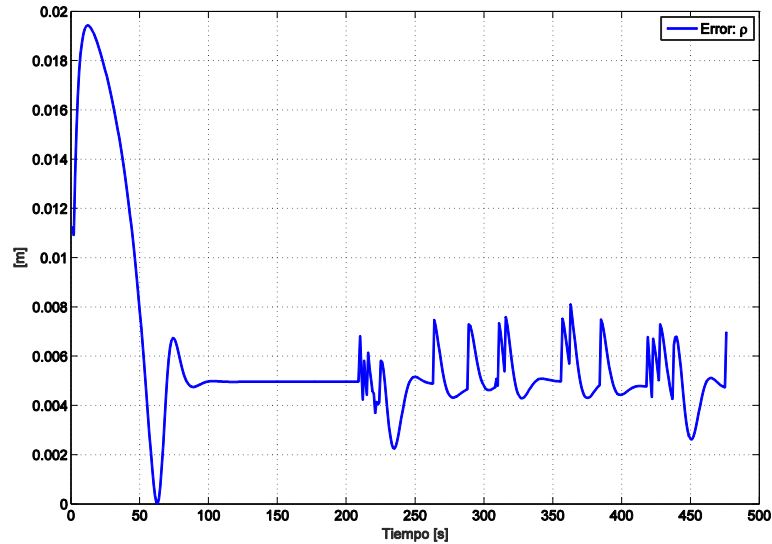


Figura 5-4: Simulación 1: Error de control
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 6-4 se observa las acciones del esquema de control, estas acciones son representadas como velocidades, lineales u_{ref} y angulares ω_{ref} que serán enviadas al robot móvil.

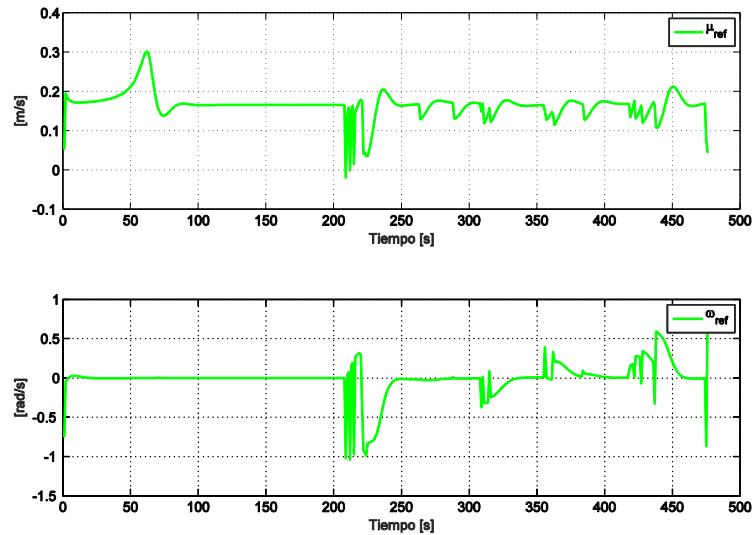


Figura 6-4: Simulación 1: Velocidades de referencia (acciones de control).
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 7-4 se observa en color verde la velocidad deseada $v_d(t)$ y en color rojo la velocidad realizada por el robot móvil $u(t)$

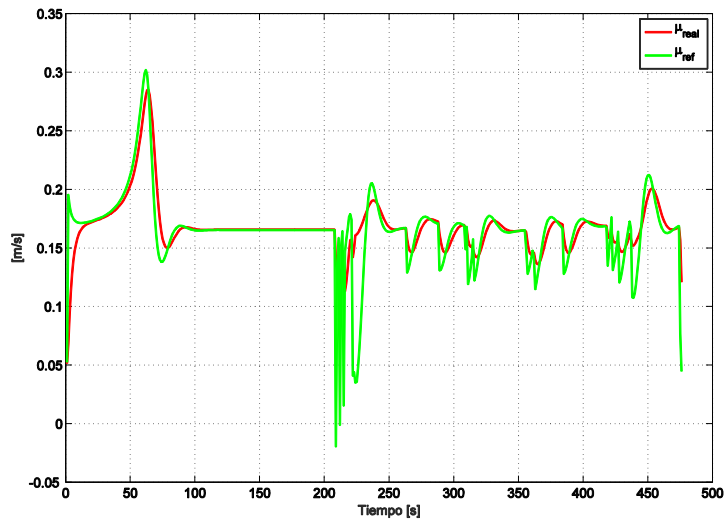


Figura 7-4: Simulación 1: Velocidad lineal deseada vs velocidad real.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

4.1.2. Simulación 2

La configuración inicial del robot móvil está en $x = -2.5$, $y = 2.5$, y $\psi = -90$, la configuración final en: $x = 2.5$, $y = -2.5$.

En la Figura 8-4 y 9-4, se muestran los resultados del algoritmo RRT, en donde se observa 274 nodos del árbol de configuraciones en color magenta, en color rojo se ve el camino inicial que consta de 15 nodos, en color verde se observa el camino luego de realizar un post-procesado, este está compuesto de 500 nodos

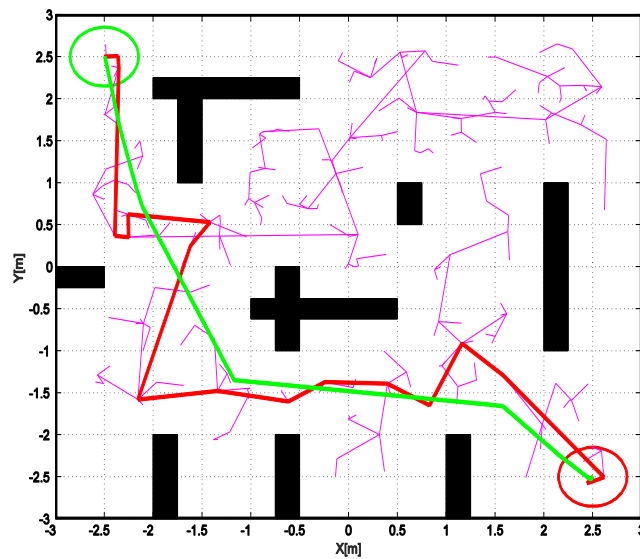


Figura 8-4: Simulación 2: Resultado del algoritmo RRT
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

Nodos Arbol: 274, Distancia: 0.1, Iteraciones: 589/15000
Nodos Camino RRT: 15, Iteraciones RRT: 589/15000
Nodos Camino Suavizado: 500, tiempo Total RRT: 21.4

Figura 9-4: Simulación 2: Numero de nodos del algoritmo RRT.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En las Figuras 10-4 hasta la 13-4 se observan los resultados del esquema de control implementado, en donde la Figura 10-4a muestra el movimiento estroboscópico del robot móvil, en la Figura 10-4b se observa el camino deseado versus el camino realizado por el robot; la Figura 11-4 muestra el error de control a lo largo del camino, en la Figura 12-4 se observa las acciones de control que están dadas por la velocidad lineal y angular de referencia, la Figura 13-4 muestra la velocidad de referencia versus la velocidad realizada por el robot móvil

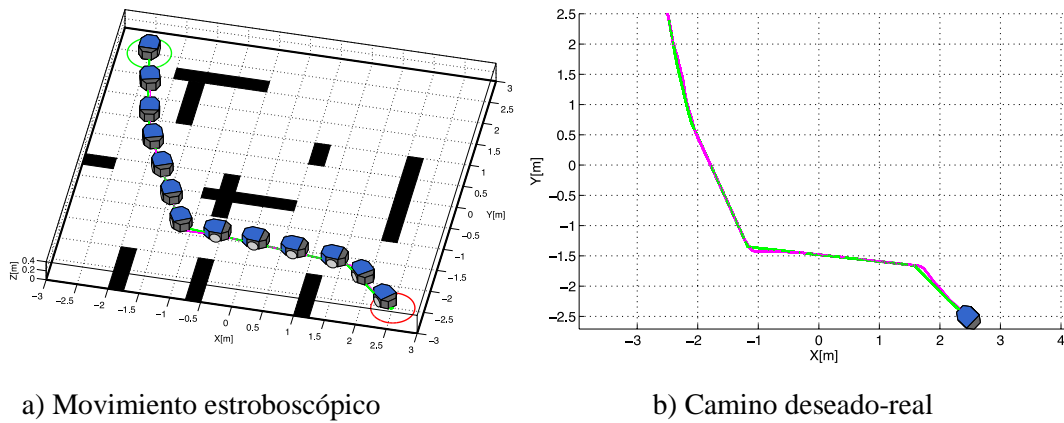


Figura 10-4: Simulación 2: Camino realizado por el robot móvil.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

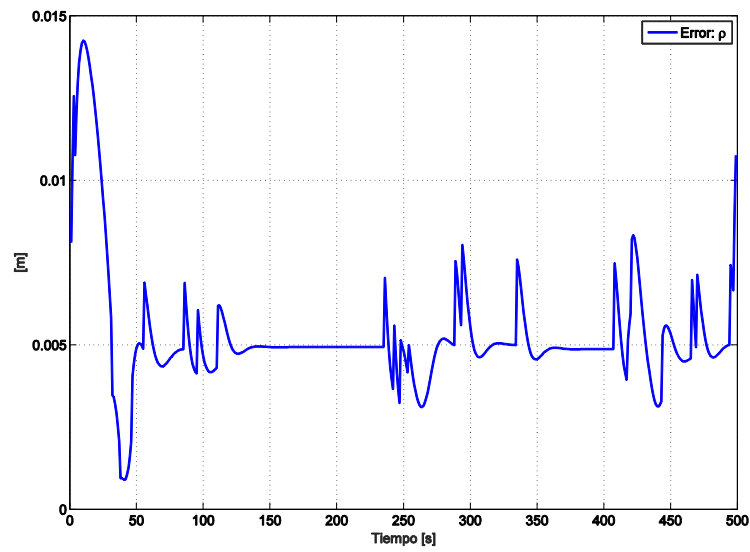


Figura 11-4: Simulación 2: Error de control
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

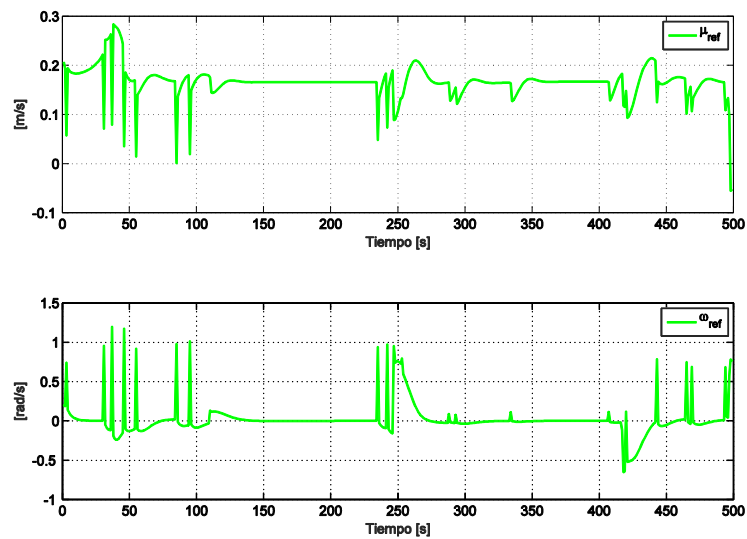


Figura 12-4: Simulación 2: Velocidades de referencia (acciones de control).
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

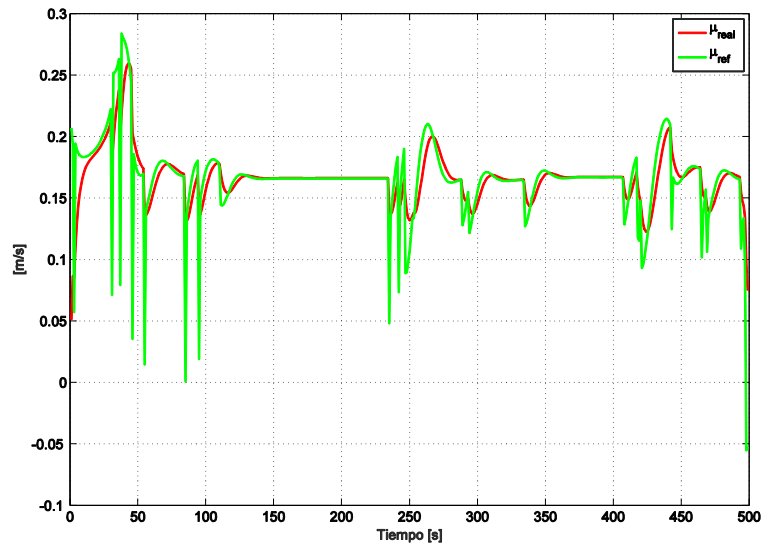


Figura 13-4: Simulación 2: Velocidad lineal deseada vs velocidad real.
 Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

4.2. Resultados experimentales

Los parámetros considerados en las pruebas experimentales son: configuración inicial del robot, $x = -2.5$, $y = -2.5$, y $\psi = 90$, la configuración final $x = 1.5$, $y = 1.5$.

En la Figura 14-4 tenemos los resultados del planificador RRT, en donde se evidencia tres elementos: el árbol de configuraciones en color magenta, camino inicial encontrado por el planificador en color rojo, el camino luego del post-procesado en color verde, notamos en esta figura como el algoritmo RRT explora todo el espacio de configuraciones, encontrando un camino inicial complicado con muchas irregularidades difíciles de ser seguidas por el robot móvil, mientras que el camino aplicado un post-procesado es más sencillo y realizable para el robot.

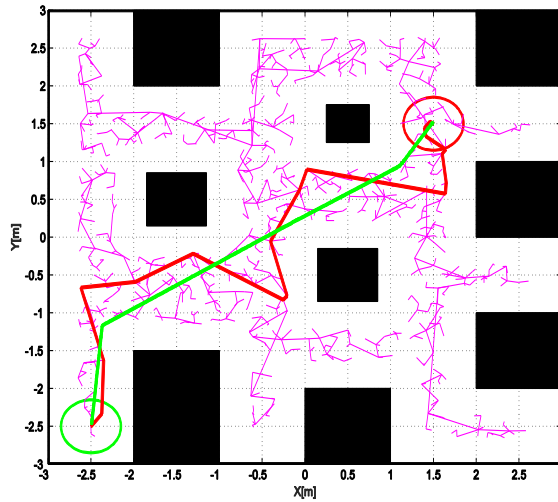


Figura 14-4: Resultado del algoritmo RRT.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 15-4 muestra que el árbol de configuraciones está compuesto por 1014 nodos, la distancia del penúltimo nodo del árbol de configuraciones hacia la meta es de 0 m, en número de iteraciones ejecutadas por el algoritmo es de 2438, el camino inicial encontrado está compuesto por 20 nodos, el camino suavizado lo componen 375 nodos, el tiempo total de ejecución del algoritmo de planificación es de 24.7 segundos.

```
Nodos Arbol: 1014, Distancia: 0.0, Iteraciones: 2438/15000
Nodos Camino RRT: 20, Iteraciones RRT: 2438/15000
Nodos Camino Suavizado: 375, tiempo Total RRT: 24.7
```

Figura 15-4: Numero de nodos del algoritmo RRT.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La Figura 16-4 muestra el camino ejecutado por el robot móvil.

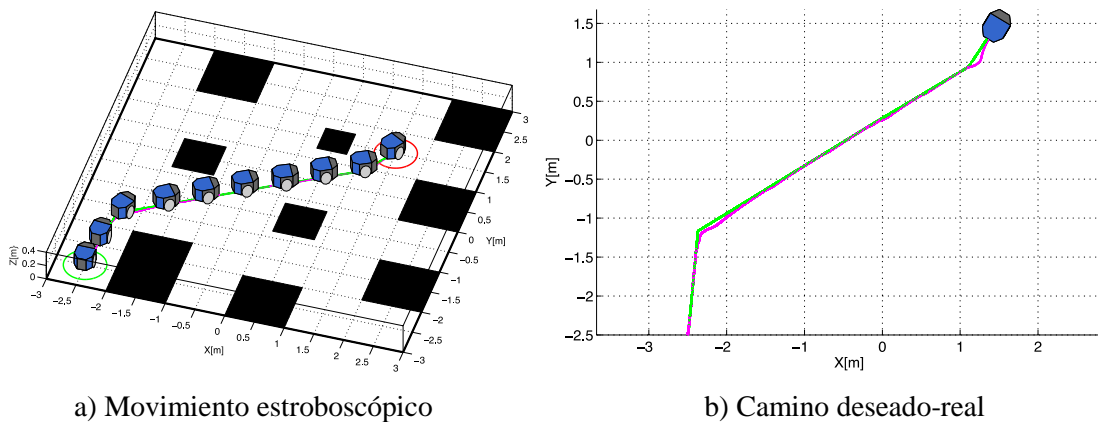


Figura 16-4: Camino realizado por el robot móvil.
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

En la Figura 16-4a se observa el movimiento estroboscópico del robot móvil, este atraviesa todo el camino sin colisiones, La Figura 16-4b muestra en color verde el camino deseado y en color magenta el camino ejecutado por el robot móvil.

La Figura 17-4 muestra el error de control $\rho(t)$ de todo el camino realizado por el robot móvil, como se puede ver este es inferior a 0.025 [m]

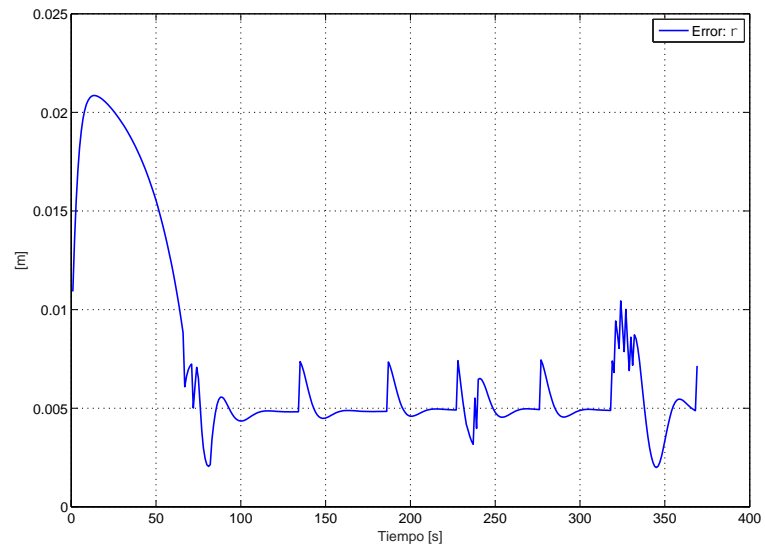


Figura 17-4: Error de control.

Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

Se muestra en la Figura 18-4 las velocidades de referencia $[u_{ref} \ \omega_{ref}]^T$ resultado de las acciones de control.

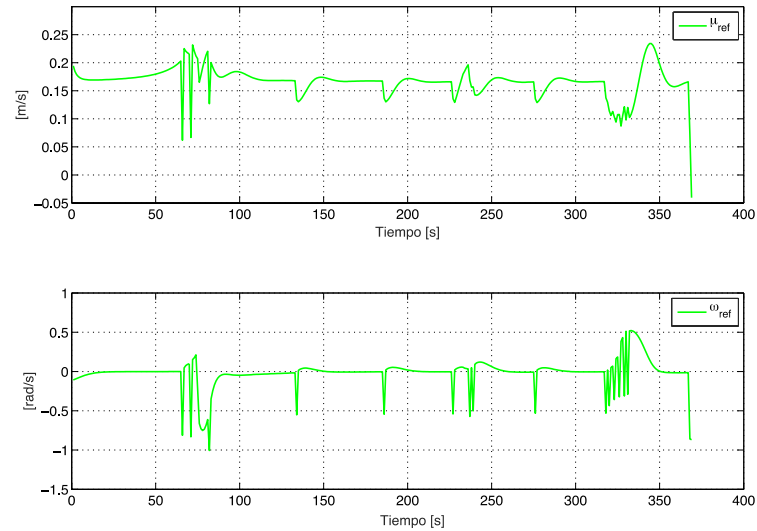


Figura 18-4: Velocidades de referencia (acciones de control)
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017.

La velocidad lineal de referencia y real se muestra en la Figura 19-4.

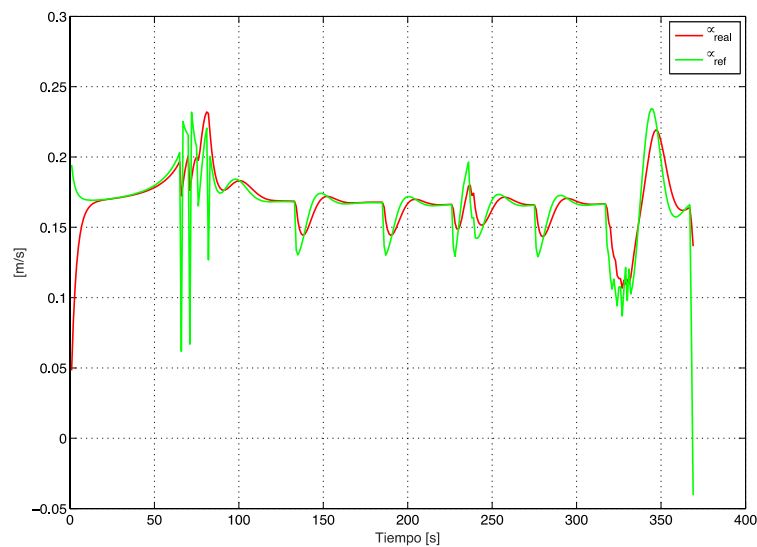


Figura 19-4: Velocidad lineal de referencia versus velocidad real
Fuente: ASQUI PAGUAY, Leonardo Manuel, 2017

CONCLUSIONES

Con el uso del algoritmo RRT se crea un árbol de configuraciones, explorando de forma randomica todo el ambiente de trabajo hasta encontrar una solución o hasta un número determinado de iteraciones; es importante considerar el radio del robot móvil en la exploración para obtener una solución lo suficientemente alejadas de los obstáculos para evitar colisiones.

Luego de encontrar una camino por el algoritmo RRT este está lleno de irregularidades, curvas innecesarias, difíciles de ser realizadas por el robot móvil, es por esto que el uso de un algoritmo de post-procesado para suavizar el camino inicial es indispensable, este algoritmo es de bajo coste computacional para no incrementar en lo posible el tiempo de obtención del camino buscado.

El modelo cinemático considera las velocidades, posición del robot móvil, sin tomar en cuenta los efectos de las fuerzas e inercias productos de la masa del robot, este modelo cinemático sirve para la implementación de un esquema de control de seguimiento de caminos. Las fuerzas e inercias son tomadas en cuenta en el modelo dinámico, es por esto que el robot no alcanza las velocidades de referencia dando como resultados caminos similares pero diferentes entre el modelo cinemático y dinámico.

El controlador cinemático propuesto para el seguimiento de caminos tiene como variable de control la velocidad lineal y angular del robot móvil, este controlador garantiza la estabilidad mediante el criterio de Lyapunov, demostrando que los errores de posición convergen a cero cuando el tiempo tiende al infinito.

Los resultados experimentales y de simulación obtenidos demuestran la versatilidad del algoritmo de planificación de caminos y esquema de control implementado, encontrando caminos suavizados con bajos costes computacionales y errores de seguimiento muy bajos

RECOMENDACIONES

Recomiendo estudiar los resultados y material obtenido en este trabajo de titulación para su uso en otros proyectos futuros en las que el ambiente de trabajo sea parcialmente estructurado, y por medio de sensores de proximidad se detecten los obstáculos en el ambiente, creando una búsqueda reactiva del camino.

Se recomienda el tomar en consideración los espacios entre los obstáculos y radio del robot móvil para que exista solución al camino buscado, y de no existir solución se recomienda el uso de una constante que detenga el algoritmo RRT.

Se recomienda el posicionamiento físico de forma correcta del robot móvil de acuerdo a la posición inicial ingresada en el programa y sistema global de coordenadas, caso contrario el robot realizara el camino de acuerdo a su posición inicial.

BIBLIOGRAFÍA

- Andaluz, V., Roberti, F., Toibero, J. M., Carelli, R., & Wagner, B.** (2011). Adaptive Dynamic Path Following Control of an Unicycle-Like Mobile Robot. *Intelligent Robotics and Applications* , 563-574.
- Bruce, J., & Veloso, M.** (2002). Real-Time Randomized Path Planning for Robot Navigation. *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference* , 3, 2383-2388.
- CAZORLA.** (2017). *Robots móviles (I)*. Recuperado el 18 de Febrero de 2017, de Xatakaciencia.com: <https://www.xatakaciencia.com/robotica/robots-moviles-i>
- De La Cruz, C., & Carelli, R.** (2006). Dynamic Modeling and Centralized Formation Control of Mobile Robots. *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics* .
- Domingo Esteve, J.** (2001). *Robótica - Apuntes para la asignatura*. Valencia.
- Ecuador. Secretaria Nacional de Planificación y Desarrollo.** (2013). *Plan Nacional de Desarrollo / Plan Nacional para el Buen Vivir* (1 ed.). Quito, Pichincha, Ecuador.
- ElectronicsTeacher.com.** (2014). *Robotics*. Recuperado el 18 de Enero de 2017, de Robotics - Types of Robots: <http://www.electronicsteacher.com/robotics/type-of-robots.php>
- Eusebio, C., & Bustos, A.** (2014). Control visual para la formación de robots móviles tipo unicycle bajo el esquema líder-seguidor. *Ingeniería Investigación y Tecnología* , 14 (4), 593-602.
- Ferguson, D., Likhachev, M., & Stentz, A.** (2005). A Guide to Heuristic-based Path Planning. *American Association for Artificial Intelligence* .
- Frazzoli, E., Dahleh, M., & Feron, E.** (2005). Maneuver-Based Motion Planning for Nonlinear Systems With Symmetries. *IEEE TRANSACTIONS ON ROBOTICS* , 21 (6), 1077-1091.
- Guzmán Luna, J. A.** (2012). Búsqueda de la ruta óptima mediante los algoritmos: genético y dijkstra utilizando mapas de visibilidad. *Scientia et Technica* (51), 107-112.

- Karaman, S., & Frazzoli, E.** (2013). Sampling-based Optimal Motion Planning for Non-holonomic Dynamical Systems. *Robotics and Automation (ICRA)* , 5041-5047.
- Karaman, S., & Frazzoli, E.** (2011). Incremental Sampling-based Algorithms for Optimal Motion Planning. *International Journal of Robotics Research* , 30, 846-894.
- Kuffner, J., & La Valle, S.** (2000). RRT-Connect: An Efficient Approach to Single-Query Path Planning. *IEEE int'l Conf. on robotics and Automation (ICRA)* , 1-7.
- La Valle, S.** (2006). *Planning Algorithms*. Illinois.
- La Valle, S., & Kuffner, J.** (2001). Randomized Kinodynamic Planning. *The International Journal of Robotics Research* , 20 (5), 378-400.
- Lopez, D., Gómez Bravo, F., Cuesta, F., & Ollero, A.** (2006). Planificación de trayectorias con el algoritmo rrt. Aplicación a robots no holónomos. *Iberoamerica de Automática e Informática Industrial* , 3 (3), 56-67.
- López García, D. A.** (Junio de 2012). Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos. *Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos* . Huelva, España.
- Martins, F. N., Carelli, R., Sarcinelli-Filho, M., & Bastos, T. F.** (2008). Modelo Dinámico y Controlador de Seguimiento para Robots Móviles Tipo Uniciclo. *Jornadas Argentinas de Robótica (JAR'08)* , V, 1-6.
- Muñoz Cueva, A.** (2014). Generación global de trayectorias para robots móviles, basada en curvas betaspline. 1-87. Sevilla, España.
- Nieto, J., Slawinski, E., Mut, V., & Wagner, B.** (Abril de 2010). Online Path Planning based on Rapidly-Exploring Random Trees. *ResearchGate* , 1430-1436.
- Ollero, A.** (2001). *ROBÓTICA Manipuladores y robots móviles*. (S. Marcombo, Ed.) Barcelona, Barcelona, España.
- Palmieri, L., & Arras, K.** (14-18 de septiembre de 2014). A Novel RRT función Extender para eficiente y suave movimiento del robot móvil Planificación. *IEEE / RSJ Conferencia Internacional de Robótica Inteligente y Sistemas (IROS)* , 205-211.

Robótica, A. E., & Producción, A. T. (2014). *AER-ATP*. Recuperado el 6 de 1 de 2017, de Robótica Industrial y de Servicio: <http://www.aeratp.com/aer-atp/robotica-industrial-y-de-servicio/>

Rodríguez Díaz, L. (2014). Planificación local eficiente de caminos para UAVs en entornos desconocidos. Sevilla.

Rossomando, F., Soria, C., & Carelli, R. (2010). Control de Robots Móviles con Incertidumbres Dinámicas usando Redes de Base Radial. *Iberoamericana de Automática e informática Industrial* , 7 (4), 28-35.

Secchi, H. A. (2008). *Una Introducción a los Robots Móviles*. Buenos Aires.

Silva, O. (2007). *UNA PANORÁMICA DE LOS ROBOTS MÓVILES*. Recuperado el 15 de 2 de 2017, de TELEMATIQUE: <http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/833/2037>

Zhang, Y., Hong, D., Chung, J., & Velinsky, S. (june de 1998). Dynamic Model Based Robust Tracking Control of a Differentially Steered Wheeled Mobile Robot. *Proceedings of the American Control Conference* , 850-855.

ANEXOS

Anexo A. Especificaciones de los motores Dynamixel Pro.

2 Reference

2.1 Default values by model

2.1.1 H Series

i. H54-200-S500-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	251000
Min Position Limit	-251000
Velocity Limit	16600
Current Limit	620
Velocity I Gain	14
Velocity P Gain	399
Position P Gain	32

ii. H54-100-S500-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	251000
Min Position Limit	-251000
Velocity Limit	17000
Current Limit	310
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

iii. H42-20-S300-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	151875
Min Position Limit	-151875
Velocity Limit	10300
Current Limit	465
Velocity I Gain	40
Velocity P Gain	440
Position P Gain	32

2.1.2 M Series

i. M54-60-S250-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)

Max Position Limit	125700
Min Position Limit	-125700
Velocity Limit	8000
Current Limit	180
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

ii. M54-40-S250-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	125700
Min Position Limit	-125700
Velocity Limit	8000
Current Limit	180
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

.1.3 L Series

i. L54-50-S290-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	103860
Min Position Limit	-103860
Velocity Limit	8000
Current Limit	180
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

ii. L54-30-S400-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	144180
Min Position Limit	-144180
Velocity Limit	13000
Current Limit	180
Velocity I Gain	16
Velocity P Gain	256
Position P Gain	32

iii. L42-10-S300-R

Name	Default Value
ID	1
BaudRate	1 (57600bps)
Max Position Limit	151875
Min Position Limit	-151875
Velocity Limit	
Current Limit	
Velocity I Gain	
Velocity P Gain	
Position P Gain	

2.2 Control Table of Dynamixel Pro

(R : Read, RW : Read and Write)

AREA	address	size (byte)	feature	description	type	Default value
EEPROM	0	2	Model Number	Model number	R	-
	6	1	Version of Firmware	Firmware version info	R	-
	7	1	ID	ID of Dynamixel PRO	RW	1
	8	1	Baud Rate	Baud rate	RW	1
	9	1	Return Delay Time	Return delay time	RW	250
	11	1	Operating mode	Operating mode	RW	3
	13	4	Homing offset	Homing value	RW	0
	17	4	Moving threshold	Moving threshold	RW	50
	21	1	Temperature limit	Internal temperature limit	RW	80
	22	2	Max Voltage Limit	Operating upper limit voltage	RW	400
	24	2	Min Voltage Limit	Operating lower limit voltage	RW	150
	26	4	acceleration Limit	Acceleration limit	RW	-
	30	2	Torque limit	Torque limit	RW	-
	32	4	Velocity Limit	Velocity limit	RW	-
	36	4	Max Position Limit	Position upper limit	RW	-
	40	4	Min Position Limit	Position lower limit	RW	-
	44	1	External Port Mode 1	External Port Mode 1	RW	0
	45	1	External Port Mode 2	External Port Mode 2	RW	0
	46	1	External Port Mode 3	External Port Mode 3	RW	0
	47	1	External Port Mode 4	External Port Mode 4	RW	0
	48	1	Shutdown	Shutdown	RW	26
	49	2	Indirect Address 1	Indirect Address 1	RW	634
	51	2	Indirect Address 2	Indirect Address 2	RW	635
	53	2	Indirect Address 3	Indirect Address 3	RW	636
	Indirect Address N	RW	-
569	2	Indirect Address 256	Indirect Address 256	RW	889	

RAM	562	1	Torque Enable	torque On/Off	RW	0
	563	1	LED RED	RED LED intensity value	RW	0
	564	1	LED GREEN	GREEN LED intensity value	RW	0
	565	1	LED BLUE	BLUE LED intensity value	RW	0
	586	2	Velocity I Gain	Velocity I Gain	RW	-
	588	2	Velocity P Gain	Velocity P Gain	RW	-
	594	2	Position P Gain	Velocity P Gain	RW	-
	596	4	Goal position	Target position value	RW	-
	600	4	Goal velocity	Target velocity value	RW	0
	604	2	Goal Torque	Target torque value	RW	0
	606	4	Goal acceleration	Target acceleration value	RW	0
	610	1	Moving	moving	R	-
	611	4	Present position	Current position value	R	-
	615	4	Present velocity	Current velocity	R	-
	621	2	Present Current	Present current value	R	-
	623	2	Present input voltage	Current input voltage	R	-
	625	1	Present temperature	Current internal temperature	R	-
	626	2	External Port Data 1	External Port Data 1	R / RW	0
	628	2	External Port Data 2	External Port Data 2	R / RW	0
	630	2	External Port Data 3	External Port Data 3	R / RW	0
	632	2	External Port Data 4	External Port Data 4	R / RW	0
	634	1	Indirect Data 1	Indirect Data 1	RW	0
	635	1	Indirect Data 2	Indirect Data 2	RW	0
	636	1	Indirect Data 3	Indirect Data 3	RW	0
	Indirect Data N	Indirect Data N ...	RW	0
	889	1	Indirect Data 256	Indirect Data 256	RW	0
	890	1	Registered Instruction	Registered Instruction value	R	0
	891	1	Status Return Level	Status Return Level value	RW	2
892	2	Hardware error status	Hardware error status value	R	0	