



Universidad
Zaragoza

Trabajo Fin de Máster

Programa Oficial de Posgrado en Ingeniería Informática
Máster en Ingeniería de Sistemas e Informática

Planificación de Movimientos en Ambientes Dinámicos usando Objetos Dinámicos de Velocidad

Autor

Eduardo F. Owen Medina

Director

Luis Montano Gella

Instituto de Investigación en Ingeniería de Aragón
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Septiembre 2012

Agradecimientos

A Luis Montano Gella, catedrático de la Universidad de Zaragoza, por haber dirigido este trabajo y por toda la ayuda prestada a lo largo del mismo.

A Claudia, mi esposa, por su animó en los momentos más difíciles.

A mis Padres, a quienes guardo en mi memoria, por su esfuerzo y motivación para llegar hasta el final.

En último lugar, a todas las personas que han tomado parte en este trabajo. Sin su colaboración este proyecto no se habría podido realizar.

PLANIFICACION DE MOVIMIENTOS EN AMBIENTES DINÁMICOS USANDO OBJETOS DINÁMICOS DE VELOCIDAD (ODV)

Resumen

Los entornos dinámicos siguen representando un segmento importante y de gran crecimiento en la automatización moderna. Aplicaciones tan diversas como el control del tráfico aéreo y marítimo, la interacción de vehículos inteligentes negociando el tráfico, el desarrollo de sillas de ruedas autónomas, la robótica de servicio o la implementación de robots industriales automatizados para el ensamble de piezas requieren que las soluciones consideren la dinamicidad del entorno seleccionando rápidamente movimientos del robot que eviten potenciales colisiones con los objetos dinámicos y estáticos del entorno.

El propósito de este proyecto es desarrollar un método de planificación de movimientos en ambientes dinámicos, evitando los obstáculos estáticos y dinámicos del entorno de navegación del robot mientras este se dirige hacia el objetivo. Para ello, primero se ha llevado a cabo un estudio de los métodos planificación de movimientos relacionados con ambientes dinámicos. Una vez analizado el problema, se ha desarrollado un método para mapear los objetos dinámicos y estáticos en el espacio de velocidades del robot. Gracias a ello, pueden calcularse para cada objeto dinámico los tiempos a colisión y los tiempos de escape así como las velocidades del robot asociadas a estos.

Como resultado final, la información obtenida es tratada para construir los *Objetos Dinámicos de Velocidad (ODV)*, que representan un modelo dinámico de los objetos del entorno en el espacio de velocidades del robot. Esta representación unificada es usada para diseñar un método de planificación de movimientos en entornos dinámicos. El método tiene la capacidad de identificar situaciones en el espacio de velocidades del robot, realizando maniobras que evitan colisiones con los obstáculos del entorno conduciéndolo hasta el objetivo a alcanzar. Finalmente, el problema de encontrar una trayectoria hasta el objetivo es planteado como uno de optimización no lineal restringido, en el cual la trayectoria semilla inicial es generada directamente en el espacio de velocidades usando el modelo construido.

Índice general

Índice de figuras	5
1. Introducción	7
1.1. Objetivos y metodología seguida	7
1.2. Organización de la memoria	8
2. Métodos de navegación: Estado del arte	10
2.1. Aproximaciones Reactivas	10
2.1.1. Campos de potencial	11
2.1.2. Histograma de Campo Vectorial (VFH)	12
2.1.3. Diagrama de Cercanía (ND)	13
2.1.4. Curvatura de Velocidad (CVM)	13
2.1.5. La Ventana Dinámica (DWA)	15
2.1.6. La Ventana Dinámica Variable en el Tiempo (TVDW)	16
2.1.7. El Obstáculo de Velocidad (VO)	16
2.1.8. Métodos en el Espacio de Trayectorias Paramétricas (TPSM)	18
2.2. Aproximaciones Deliberativas	19
2.2.1. Métodos basados en Mapa de Rutas	19
2.2.2. Métodos basados en Descomposición de Celdas	20
2.2.3. Métodos basados en Muestreo	21
2.3. Otros Esquemas Alternativos de Planificación de Movimientos	22
2.3.1. Métodos de Deformación	23
2.3.2. Planificación de Movimientos Parcial (PMP)	23
2.3.3. Métodos de Replanificación	24
3. Planificando movimientos en entornos dinámicos usando ODV	25
3.1. Mapeando los obstáculos móviles del <i>CS</i> al <i>VS</i>	26
3.1.1. Cálculo de los tiempos a colisión a los obstáculos	27
3.1.2. Considerando las Restricciones Dinámicas del Robot	29
3.2. Calculando trayectorias usando el <i>DVS</i>	30
3.3. Resultados experimentales	32
3.4. Conclusiones	34
4. Optimizando la Planificación en el Espacio de Velocidades	37
4.1. Declaración del problema - La aproximación	38
4.2. El planificador de movimientos robocéntrico	39
4.2.1. Las velocidades altas son libres	39

4.2.2. Las velocidades altas no son libres	40
4.2.3. Clases de trayectorias	41
4.3. Optimización de las trayectorias	42
4.4. Resultados experimentales	43
4.5. Conclusiones	44
5. Conclusiones y trabajo futuro	45
A. Características del robot	47
B. Limites del espacio de velocidades	49
C. Ecuaciones de las curvas	51
C.1. Ecuaciones de movimiento del robot	51
C.2. Ecuaciones de una Clotoide	51
C.3. Ecuaciones de una Involuta	52
C.4. Ecuaciones de un arco de círculo con aceleraciones lineal y angular	52
Bibliografía	54

Índice de figuras

2.1.	Campo de potencial del entorno con dos objetos.	11
2.2.	(a) Modelo del mundo basado en una rejilla de certidumbre. (b) Histograma Polar.	12
2.3.	Diagrama de Cercanía.	13
2.4.	Curvatura de Velocidad.	14
2.5.	Ventana Dinámica para un robot holónimo.	15
2.6.	Ventana Dinámica variable en tiempo.	16
2.7.	Obstáculo de Velocidades Lineal.	17
2.8.	Obstáculo de Velocidades Lineal.	17
2.9.	Superficie de muestreo resultante en \mathcal{C} generada a partir de una función de generación en <i>TPS</i>	18
2.10.	Espacio $\mathcal{C} \times \mathcal{T}$	20
2.11.	Representación aproximada por muestreo.	22
2.12.	Modelo de Bandas Elásticas.	23
3.1.	(a) El espacio de trabajo (<i>WS</i>) (b) La banda de colisión, el camino r_j y los puntos de colisión P_{c1j} and P_{c2j} en el espacio de configuración (<i>CS</i>) (c) Extensión a todo el rango de caminos circulares con riesgo de colisión (d) <i>ODV</i> en el espacio de velocidades-tiempo (<i>DVS</i>).	28
3.2.	(a) Proyección del <i>ODV</i> , V_{ODV} , en el plano (v, w)	28
3.3.	(a) \mathbf{v}_o es alcanzable solo dentro de la ventana, (b) \mathbf{v}_o no puede ser alcanzada	30
3.4.	(a) La estrategia elegida es alcanzar $\mathbf{v}_o = \mathbf{v}_G = (v_{max}, 0)$, (b) \mathbf{v}_{MT} conduce al robot a moverse alrededor del obstáculo a máxima velocidad lineal y \mathbf{v}_{SP} genera un movimiento rectilíneo (el camino más corto) a baja velocidad. \mathbf{v}_d es la consigna de velocidad.	31
3.5.	La consigna de velocidad \mathbf{v}_d no conduce a colisión en el tiempo actual y se puede alcanzar \mathbf{v}_o en 4 pasos ($4T$)	32
3.6.	(a) Experimento 1: criterio MT y perfil de velocidades (b) Experimento 2: criterio SP y perfil de velocidades	33
3.7.	Experimento 3: navegación alrededor de cuatro obstáculos móviles y perfiles de velocidad.	34
3.8.	Experimento real en el vestíbulo. El robot maniobra evitando todos los obstáculos móviles, evitando posibles colisiones mientras se dirige hacia el objetivo.	35
3.9.	Vestíbulo del I3A: (a) Perfiles de velocidad lineal y angular.	36
3.10.	Traectoria del robot en verde y entorno percibido por el robot.	36
4.1.	(a) trayectorias y V_{ODV} en el <i>VS</i> de la situación <i>S1</i> , (b) trayectorias y V_{DOV} en el <i>VS</i> de la situación <i>S2</i> . El cuadrado representa las máximas velocidades lineales y angulares que pueden alcanzarse.	41

4.2.	(a) corresponde a la trayectoria en WS de $S1$; el robot acelera para pasar antes que el objeto, (b) corresponde a la trayectoria en WS de $S2$; el robot desacelera para esperar que el objeto pase.	42
4.3.	(a) trayectoria resultante cuando el plan es calculado para cada periodo de tiempo, (b) perfiles de velocidad lineal y angular del primer experimento.	43
4.4.	(a) Trayectoria planeada en otro escenario, con velocidades diferentes (magnitud y dirección) en los objetos, (b) Perfiles de velocidad lineal y angular en el segundo experimento.	44
A.1.	Límites del espacio de velocidades real	48
B.1.	Límites del espacio de velocidades real	50

Capítulo 1

Introducción

Este proyecto se enmarca dentro de la línea de investigación realizada por el grupo de Robótica, Percepción y Tiempo Real de la Universidad de Zaragoza, dentro del campo de la robótica móvil aplicada a entornos dinámicos.

En general puede definirse un robot como un dispositivo mecánico complejo equipado con sensores y actuadores que está capacitado para realizar tareas complejas de forma autónoma o con supervisión humana, llegando a ser estas en general demasiado aburridas, sucias, peligrosas o que requieren precisión. Tareas como extender un brazo robótico, cerrar una pinza o propulsarse para alcanzar un objetivo son el resultado de una secuencia de acciones, donde cada acción es traducida a un movimiento. Por tanto para que el robot consiga llevar a cabo con éxito una tarea él debe ser capaz de *planificarla*, es decir encontrar por sí mismo y por adelantado, la secuencia de movimientos a ejecutar. Este problema en particular es tratado en robótica por la planificación de movimientos.

En los años setenta el concepto de *espacio de configuraciones (CS)* es introducido como una herramienta útil para tratar con el problema básico de la planificación de movimientos. En el *CS*, el robot es representado como un punto y los objetos estáticos como regiones prohibidas. El problema básico es por tanto estrictamente geométrico y es conocido como *planificación de caminos*, su objetivo es encontrar un *camino* libre de colisiones es decir, una secuencia continua de configuraciones del robot en el espacio libre.

Cuando los obstáculos en el ambiente son dinámicos, la planificación de caminos no puede suministrar la información suficiente al robot para alcanzar su objetivo. En tal caso la dimensión del tiempo debe ser tomada en cuenta, siendo necesario introducir una parametrización del tiempo en la secuencia de configuraciones del robot. El problema de la planificación de movimientos consiste entonces en encontrar una *trayectoria* realizable por el robot que evite las colisiones con los obstáculos dinámicos y estáticos mientras se dirige al objetivo. La trayectoria realizada puede llegar a ser complicada de encontrar si se añaden más aspectos a considerar como las restricciones cinemáticas y dinámicas del robot, restricciones de tiempo real, incertidumbre o criterios de optimalidad.

1.1. Objetivos y metodología seguida

Este proyecto se centra en la navegación de robots móviles en ambientes dinámicos. Para ello es necesario llevar a cabo un estudio del estado del arte sobre el tema con el fin de analizar y

tratar el problema apropiadamente.

El objetivo final es desarrollar e implementar un método de planificación de movimientos en ambientes dinámicos usando el espacio de velocidades del robot, que tenga en cuenta sus restricciones cinemáticas y dinámicas. El proyecto cuenta con el apoyo del Laboratorio de robótica del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza para usar las plataformas robóticas y comprobar el buen desempeño y funcionamiento del método desarrollado. Con este proyecto se pretende que el trabajo realizado tenga posibles aplicaciones de implementación en robots de servicio como sillas de ruedas autónomas y manipuladores móviles o en vehículos autónomos sin asistencia de conductor humano.

Para alcanzar el objetivo de este proyecto el trabajo ha sido dividido en las siguientes fases:

En primer lugar se realizará un estudio de los trabajos afines para comprender el problema de planificación a tratar, analizando los distintos aspectos que cada solución aporta y que puedan ser destacados para abordar el problema.

En segundo lugar es necesario implementar una aplicación que mapea los objetos dinámicos y estáticos al espacio de velocidades del robot. El resultado de esta metodología da origen a los Objetos Dinámicos de Velocidad (*ODV*) que posteriormente serán utilizados para diseñar estrategias de planificación. Esta aplicación inicialmente es desarrollada en MatLab con el fin de evaluar su desempeño en simulación.

Conseguido lo anterior, la siguiente fase es desarrollar la estrategia de navegación para el robot usando los *ODV*. La caracterización de los *ODV* en el espacio de velocidades del robot facilita la identificación de estrategias de navegación para la evitación de los objetos dinámicos. Durante esta fase el problema de la planificación de movimientos es tratado desde una perspectiva de optimización no lineal, la cual considera las restricciones cinemáticas y dinámicas en el movimiento del robot y los objetos dinámicos. El diseño de esta parte del método de planificación se implementa usando Matlab para su verificación en simulación.

Finalmente algunos experimentos son considerados, comprobando así la aplicabilidad del método en esta clase de entornos y verificando la capacidad del método para mantener la integridad y seguridad del robot mientras se dirige al objetivo usando la plataformas robóticas del laboratorio.

También hemos de resaltar que el resultado de este trabajo de master ha sido presentado y divulgado a la comunidad científica, siendo evaluado por pares internacionales que investigan en el campo de la robótica móvil. Específicamente son dos las publicaciones presentadas en donde se considera los temas presentados en los capítulos 3 y 4 respectivamente de esta memoria. En [OM05] es tratado el contenido del capítulo 3 y en [OM06] se trata el capítulo 4 de esta memoria de master.

1.2. Organización de la memoria

Esta memoria se encuentra dividida en cuatro capítulos, incluido este a modo de introducción. A continuación se detallan los contenidos de cada uno de ellos.

El segundo capítulo abordará el planteamiento general del problema y una revisión teórica sobre el estado del arte en navegación de robots móviles. Específicamente, se resaltarán aquellos métodos de navegación de robots que incorporan en si mismo la dinámica de los objetos del entorno, permitiendo al planificador considerar los movimientos de los objetos apropiadamente para mantener la integridad del robot mientras alcanza su objetivo.

El capítulo tercero presentará el método que mapea los objetos dinámicos y estáticos al VS del robot. El propósito de este capítulo es mostrar detalladamente la metodología utilizada para transformar los objetos dinámicos del entorno del robot desde el CS al VS del robot. El proceso anterior da origen a los ODV , los cuales son tenidos en cuenta durante el proceso de planificación de movimientos del robot. En este espacio también resulta más simple considerar las restricciones cinemáticas y dinámicas del vehículo fortaleciendo la idea de hacer la planificación en el VS . Un esquema de planificación básico es aplicado como resultado de la interpretación y gestión de la información mapeada en el VS . También serán presentados algunos resultados experimentales obtenidos.

El capítulo cuarto planteará el problema de planificación de movimientos usando el espacio de velocidades del robot como uno de optimización no lineal. La forma de encontrar una trayectoria hasta el goal se presentará como un problema de optimización restringido y no lineal, donde la trayectoria inicial usada como semilla inicial será generada directamente en el VS . El método será aplicado y se simulará su comportamiento sobre un robot con restricciones en su cinemática y su dinámica. En este capítulo algunos resultados experimentales serán discutidos.

En el quinto y último capítulo se exponen las conclusiones extraídas tras la realización de este trabajo. También se hace referencia a la futura línea de investigación que se seguirá, partiendo de la base realizada en este trabajo.

En la sección de anexos se incluye información adicional acerca de los algunos aspectos relacionados con este trabajo: hardware y software utilizado en el proyecto y detalles del método de planificación realizado.

Capítulo 2

Métodos de navegación: Estado del arte

Desde el punto de vista general, este trabajo tiene el propósito de desarrollar un método de navegación para vehículos o robots autónomos en un ambiente con objetos dinámicos. El propósito de un método de navegación consiste en generar y ejecutar los movimientos del robot para alcanzar un objetivo predefinido, evitando las colisiones con los obstáculos presentes en el entorno. Podemos afirmar que el método de navegación está en el corazón de la estrategia de movimientos del robot autónomo.

Las distintas formas en que el problema de la planificación de movimientos ha sido conceptualizado y dividido en partes más pequeñas hace más difícil revisar y clasificar la literatura concerniente a este tema. No obstante, la extensa variedad de aproximaciones puede ser aclarada si su estudio se enfoca desde dos perspectivas distintas: una reactiva y otra deliberativa.

La perspectiva reactiva calcula una sola acción a la vez para ser llevada a cabo el próximo periodo de tiempo. El enfoque deliberativo, al contrario, tiene como objetivo el cálculo de la secuencia completa de acciones para alcanzar el objetivo.

En la siguiente sección, inicialmente nos proponemos revisar las aproximaciones reactivas más importantes. Después de esto nos centramos en las aproximaciones que usan un esquema deliberativo para llevar a cabo la planificación de movimientos. En esta sección nos introduciremos en los diferentes métodos y su capacidad para manejar diferentes limitaciones. Luego, las aproximaciones que no encajan muy bien en la anterior división son examinadas en otro apartado que hemos denominado esquemas alternativos.

2.1. Aproximaciones Reactivas

Los métodos reactivos calculan en el mismo periodo de tiempo una sola acción que debe ser ejecutada inmediatamente. Este nivel trata con los eventos inesperados de un plan concedido previamente, por ejemplo de un planificador deliberativo. La estrategia actualiza la información del ambiente proporcionada por los sensores en cada periodo de tiempo, modificando el plan actual mediante rectificaciones locales para evitar los objetos detectados. Los métodos reactivos son técnicas relativamente simples que pueden ser ejecutados en tiempo real.

2.1.1. Campos de potencial

En este método el robot es considerado como una partícula inmersa en un campo potencial artificial, el cual es atraído hacia el objetivo y rechazado lejos de los obstáculos. El modelo del medio ambiente es especificado con una función de potencial que determina las fuerzas ejercidas sobre el robot. El método elige de forma iterativa una acción de movimiento, seleccionando una dirección que normalmente es señalada por el gradiente negativo de la suma de las fuerzas que actúan. El procedimiento continúa hasta que el robot alcanza la configuración del objetivo de tener éxito. La figura 2.1 representa el campo de potencial calculado para un entorno con dos objetos estáticos representado como una malla de valor P .

Los métodos de campos de potencial habituales [BK89, Kha86] están sujetos a problemas de *mínimos locales*, donde el robot es llevado cuando el método sigue la dirección señalada por el gradiente. Dos técnicas para superar este problema son planteadas. La primera es reemplazar la estrategia del gradiente descendente por una búsqueda guiada (búsqueda primero en profundidad, primero el mejor, A^* o paseo aleatorio), para intentar encontrar una salida cuando el robot se encuentra atrapado en un mínimo local. La segunda técnica es producir campos de potencial que representan funciones de navegación suaves conteniendo un único mínimo local situado en el objetivo a alcanzar como es mostrado en [RK92]. En [KK92], la segunda técnica es implementada generando funciones de navegación no óptimas a partir de funciones armónicas de potencial.

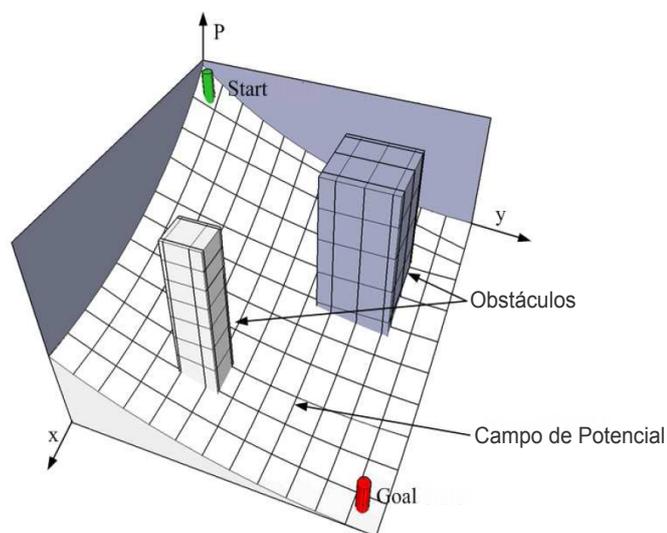


Figura 2.1: Campo de potencial del entorno con dos objetos.

Este método no es adecuado para entornos dinámicos ya que el cálculo de la fuerza no tiene en cuenta explícitamente el movimiento de los objetos porque considera que la distancia a los objetos permanece constante durante el tiempo en el cual es tomada la acción. En [GC02, Hua08] se propone una extensión más apropiada para entornos dinámicos. El movimiento de los objetos dinámicos es representado definiendo una función de potencial que tiene en cuenta la distancia y la velocidad instantánea. Sin embargo, existen problemas de mínimos locales que son resueltos usando heurísticas o búsquedas guiadas sin garantía que una solución sea encontrada.

Otro problema detectado en esta clase de métodos es que el resultado se expresa como una fuerza indicando la dirección donde el robot debe moverse. Esto funciona bien para los robots

holónomos, que pueden moverse libremente en el espacio. No obstante, la gran mayoría de los vehículos robóticos tienen limitaciones cinemáticas y dinámicas que impiden que la dirección indicada por el método pueda ser seguida de inmediato. Como resultado, un controlador de bajo nivel es requerido para lograr la dirección deseada resultando a veces el movimiento alcanzado bastante diferente al esperado.

2.1.2. Histograma de Campo Vectorial (VFH)

El método VFH introducido en [BK91] usa un histograma cartesiano 2D como un modelo del mundo (figura 2.2(a)), el cual puede ser actualizado en tiempo real utilizando la información de los sensores de abordo del robot. Este histograma se basa en una malla de certidumbre [Mor88] y en una malla de ocupación [Elf89], donde la información de cada celda representa la probabilidad de tener un obstáculo en este momento. Luego la malla es mapeada a un histograma polar como se muestra en la figura 2.2(b), donde las posibles decisiones a escoger resultan de evaluar el histograma por debajo de un cierto valor umbral. Una función de coste basada en la dirección del objetivo, la orientación actual del robot y el ángulo de las ruedas es usada para escoger una entre las posibles direcciones autorizadas y definir la orientación del movimiento en el siguiente paso. La velocidad se calcula en una segunda etapa, como una función de la distancia desde el robot móvil y el estado final.

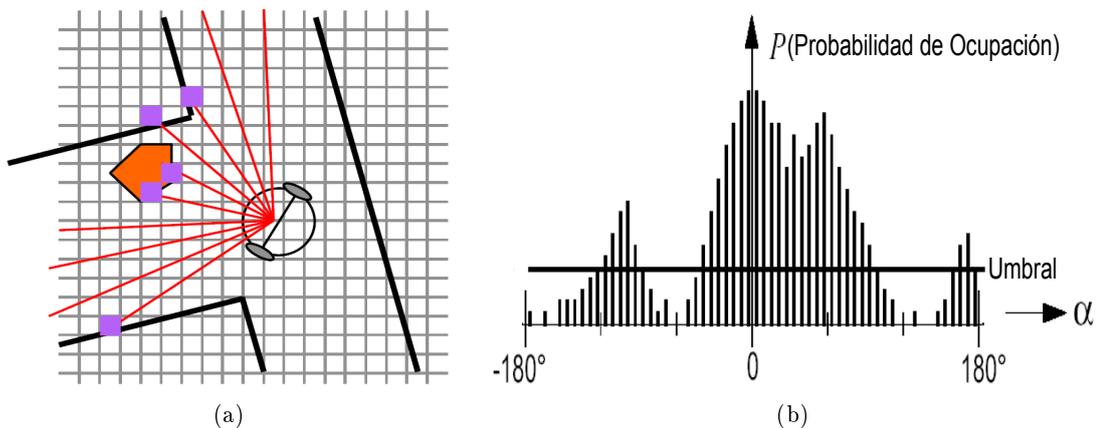


Figura 2.2: (a) Modelo del mundo basado en una rejilla de certidumbre. (b) Histograma Polar.

El método está limitado por el uso de heurísticas que requieren de afinación y que en gran medida influyen en su comportamiento. Además, el uso de este histograma polar reduce el **campo de vista** del robot impidiendo la elección al sistema de la orientación más adecuada. Una extensión llamada VFH*, propuesta en [UB00], mejora el carácter local del esquema original verificando que la dirección elegida por el método pueda guiar el robot alrededor de un obstáculo sin quedar atrapado. Esta verificación es realizada implementando un algoritmo de búsqueda A* con funciones heurísticas apropiadas y ponderadas adecuadamente. Como resultado la estrategia mejora la convergencia del sistema y la capacidad del robot para escapar de la presencia de mínimos locales en el entorno. Este método no considera los entornos dinámicos. Apesar de que las restricciones cinemáticas son tenidas en cuenta en una variante del método presentada en [UB98]. El método no tiene en cuenta las limitaciones dinámicas del sistema y la dinamicidad del ambiente lo cual constituye su principal limitación.

2.1.3. Diagrama de Cercanía (ND)

El método ND, presentado en [MM00], consiste en analizar una situación a partir de dos diagramas polares construidos por medio de la información obtenida por los sensores de abordó del robot. Desde uno de los diagramas se extrae la información característica del entorno y la identificación del valle más próximo al objetivo (ver la figura 2.3(a)). El otro diagrama es usado para definir el nivel de seguridad entre el robot y los obstáculos, identificando el más cercano (ver la figura 2.3(b)). Las restricciones cinemáticas del robot son extendidas en [MMSV02], aproximando sus caminos por segmentos rectos y arcos de círculos. La decisión del siguiente movimiento es tomada eligiendo el control más adecuado en el espacio ego-cinemático, el cual es una representación del mundo en términos de distancia y el radio de curvatura que describe el robot a un obstáculo. En [MMSA01] el ND Global es presentado. El método es una extensión del ND en combinación con una función NF1 cuyo propósito es aumentar el **campo de vista** del robot evitando situaciones en las que el robot quede atrapado. Un espacio ego-dinámico se presenta en [MMK02] para dar cuenta de las limitaciones dinámicas del robot.

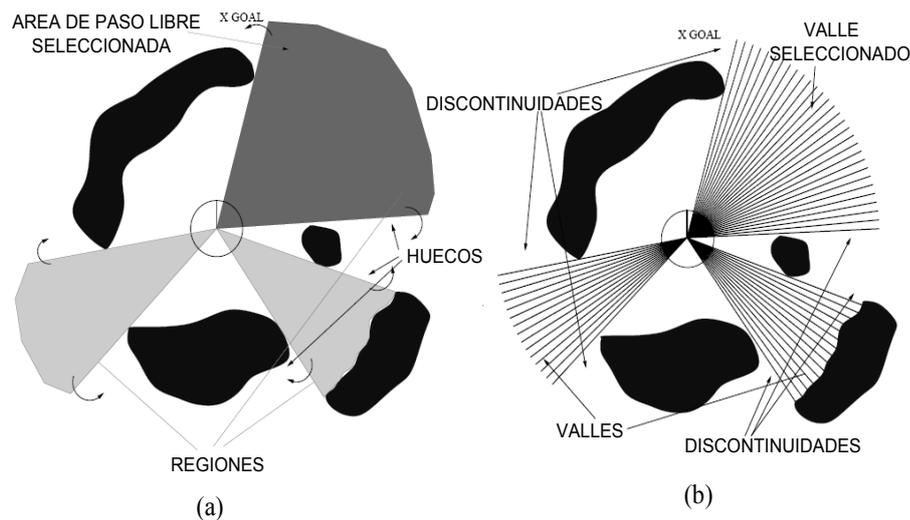


Figura 2.3: Diagrama de Cercanía.

2.1.4. Curvatura de Velocidad (CVM)

El método CVM fue introducido en [Sim96], como un método local de evitación de obstáculos para robots móviles. El método opera en el espacio de velocidades del robot, el cual es conformado por las velocidades lineales y angulares (ν, ω), y toma en cuenta las restricciones cinemáticas (trayectorias circulares) y dinámicas (límites de aceleraciones y velocidad) del robot. Los objetos presentes en el entorno se representan de forma circular, como puede verse en la figura 2.4, para permitir un cálculo más fácil de la distancia que el robot atravesará desde su posición al obstáculo. La distancia a los obstáculos es calculada suponiendo que el robot describe una trayectoria de curvatura constante, cuyo valor es $\kappa = \frac{\omega}{\nu}$. Solo las curvaturas que se encuentran dentro de las capacidades cinemáticas del robot son elegibles por una función objetivo que es la encargada de

seleccionar la dupla de velocidad que será aplicada al robot. La función objetivo está construida de manera que sean preferidas velocidades altas, curvaturas que permitan viajar más tiempo al robot antes de colisionar y orientaciones que tiendan al robot al objetivo. Finalmente la dupla de velocidad que maximiza la función objetivo es elegida como el mejor candidato para ser aplicado en el siguiente paso.

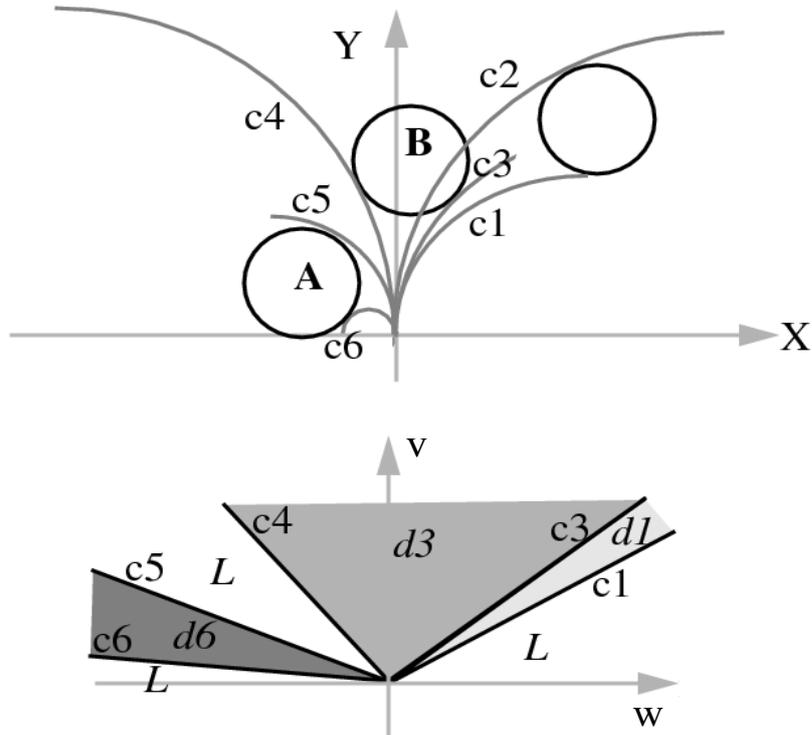


Figura 2.4: Curvatura de Velocidad.

El método carril-curvatura (LCM) es presentado en [KS98], como una extensión del CVM para superar sus principales deficiencias. Principalmente aquellas que surgen de la hipótesis de movimiento en arcos circulares. LCM divide el entorno en carriles que son construidos teniendo en cuenta la máxima distancia, libre de colisión, a los obstáculos y a lo largo de la dirección objetivo deseada. Además, los carriles adyacentes con distancias similares y libres de colisión son fusionados. El método elige el carril más prometedor con la ayuda de una función objetivo, corrigiendo la orientación del robot para cambiar de carril, si el carril seleccionado no se corresponde con el del robot. Esta extensión, tampoco trata el comportamiento futuro de los objetos en un entorno dinámico.

La principal dificultad del método radica en el proceso de mapear los obstáculos del entorno real del robot a su espacio de velocidades. Otra desventaja observada es la dependencia de factores de peso, presentes en la función objetivo, que producen comportamientos no uniformes dependiendo de la situación. Por último el método no considera el comportamiento futuro de los objetos dinámicos en el entorno poniendo en riesgo la integridad del robot.

2.1.5. La Ventana Dinámica (DWA)

La ventana dinámica es presentado en [FBT97]. Es uno de los métodos reactivos más populares usando la representación del espacio de velocidades. La ventana dinámica de cierta manera es muy similar al método de la curvatura de velocidad, pero difiere en que este construye un espacio de velocidades discretizado. Su espacio de búsqueda se compone de todas las duplas posibles de velocidad lineal y angular. Una ventana dinámica reúne todas las velocidades que pueden ser alcanzadas (V_r) alrededor de la velocidad actual del robot (v_c). Esta es construida para un intervalo de tiempo pequeño (Δt) teniendo en cuenta las restricciones del robot en su aceleración/deceleración traslacional y rotacional. Un dupla de velocidad es incluida en el conjunto de velocidades admisibles (V_a) si el robot es capaz de parar antes de colisionar con el objeto detectado en el entorno. En la figura 2.5 se muestra el VS para un robot holónimo con las velocidades admisibles (V_a), las velocidades prohibidas en color gris oscuro (aquellas parejas de velocidad que conducen el robot a colisión) y el conjunto de velocidades alcanzables que pertenecen a la ventana dinámica. La elección del próximo comando de velocidad del robot, es el resultado del proceso de optimización de una función objetivo aplicada a todas las velocidades admisibles en (V_r). La función objetivo favorece tres características: el avance del robot hacia el objetivo, el espacio libre de obstáculos y los movimientos de avance rápido (velocidades lineales altas).

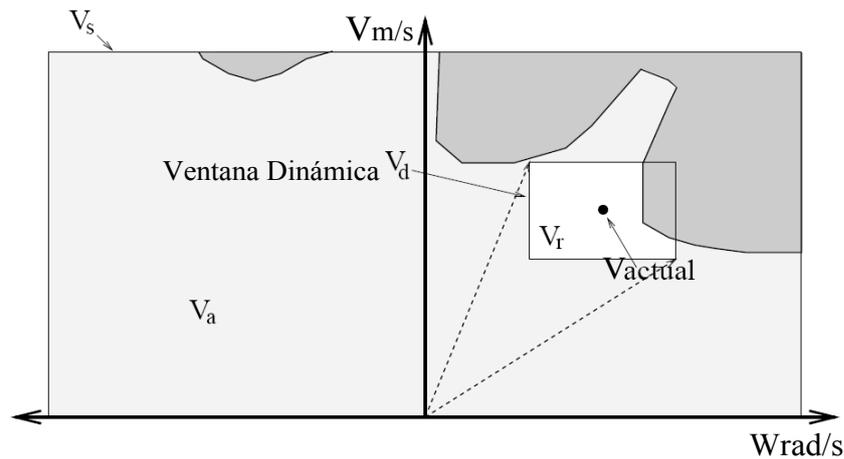


Figura 2.5: Ventana Dinámica para un robot holónimo.

Una extensión al método, la ventana dinámica global (Global DWA), es presentado en [BK99] para superar sus limitaciones. Además de los sensores del robot usados para construir el modelo, el método emplea un mapa para mejorar su campo de vista. De esta manera DWA es combinado con una rejilla basada en una función de navegación global resultando en: un mejor esquema hacia al objetivo y una información más apropiada sobre la conectividad del espacio libre para evitar que el robot quede atrapado en mínimos locales.

Este método está limitado por la construcción del modelo, así como por el ajuste de parámetros de la función objetivo, ya que si un obstáculo es detectado por los sensores del robot pero este queda fuera de la ventana dinámica, entonces este será ignorado. Esto implica que se descarta la valiosa información potencial en el método que podría contribuir a lograr mejorar su rendimiento. DWA también hace la suposición que el entorno permanece estático sin razonar sobre el comportamiento futuro de los objetos en un entorno dinámico.

2.1.6. La Ventana Dinámica Variable en el Tiempo (TVDW)

Este método se introduce en [SP07]. En él se extiende el enfoque clásico de ventana dinámica, calculando para cada instante de tiempo un conjunto de trayectorias futuras para los obstáculos móviles y comprobando sus colisiones a corto plazo. A diferencia de DWA, el método razona sobre el comportamiento futuro de los obstáculos móviles dejando de asumir un entorno estático. El ambiente alrededor del robot es representado por medio de una rejilla de ocupación donde cada celda es clasificada como ocupada o libre. Además, cada celda ocupada es identificada con la etiqueta MC si ella pertenece a un obstáculo móvil. El método asume como conocidas las velocidades lineal y angular (ν_{mc}, ω_{mc}) y la dirección de movimiento θ_{mc} en cada celda móvil MC. Con esta información se predicen el conjunto de trayectorias obstáculo para cada celda móvil MC en el intervalo de tiempo Δt . Una dupla de velocidad (ν, ω) se considera admisible si ninguna colisión ocurre entre las trayectorias MC y y la trayectoria correspondiente a esa dupla. En la figura 2.6 se ilustra el método. Las trayectorias libres de colisión en verde son aquellas producidas por velocidad admisibles en el robot. Como en la DWA, el método usa un función de coste para seleccionar un único control de velocidad del conjunto de velocidades admisibles.

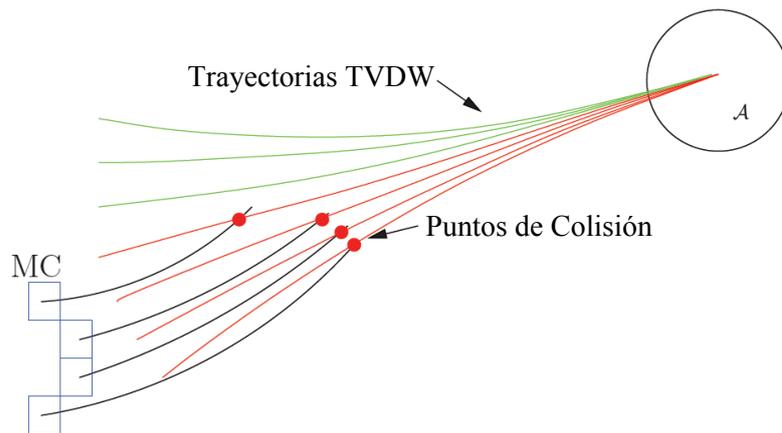


Figura 2.6: Ventana Dinámica variable en tiempo.

2.1.7. El Obstáculo de Velocidad (VO)

El VO y sus extensiones son un conjunto de métodos de navegación simples pero bastante apropiados para tratar entornos con obstáculos móviles. En su forma más simple y original fue presentado en [FS98]. Es un método que opera en el VS del robot compuesto por las velocidades lineales (ν_x, ν_y). Este supone que los objetos dinámicos mantienen una velocidad lineal constante describiendo una trayectoria lineal. Cada objeto del entorno del robot genera un conjunto de velocidades prohibidas con forma de cono, denominado cono de velocidades. La región cónica en color rojo de la figura 2.7 representa las velocidades que conducirían al robot a una posible colisión con el objeto. Si alguna de estas velocidades es seleccionada por el robot este terminaría colisionando con el objeto un tiempo después en el futuro. En la práctica, las velocidades admisibles elegibles pueden ser aquellas que conducirían al robot a una colisión un tiempo después

del horizonte de tiempo dado por t_h .

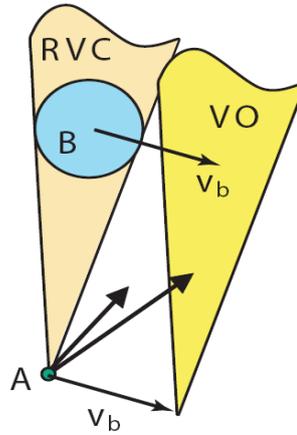


Figura 2.7: Obstáculo de Velocidades Lineal.

EL VO presenta limitaciones en espacios cerrados ya que cada velocidad eventualmente podría conducir a colisión. Esta limitación es resuelta manipulando arbitrariamente el horizonte de tiempo t_h lo que dificulta mayormente fijar un valor apropiado cuando diversas situaciones se presentan simultáneamente. Otra limitación se aprecia en la forma circular de los objetos móviles del entorno del robot y su velocidad lineal constante que puede llegar a ser restrictiva.

El VO fue mejorado en [SLSL01]. El método se conoce como el Obstáculos de velocidades no lineal (NLVO) y permite considerar objetos móviles que describen trayectorias arbitrarias. El NLVO representa todas las velocidades del robot en el tiempo t_0 que pueden conducir al robot a colisión con el obstáculo en cualquier tiempo t entre $t_0 \leq t \leq t_h$ como puede verse en la figura 2.8. El NLVO presenta las mismas limitaciones que el VO mencionadas antes. Finalmente en [WvdBM09] se introduce el concepto de obstáculo de velocidades generalizado (GVO) en cuyo modelo son consideradas las limitaciones cinemáticas del robot.

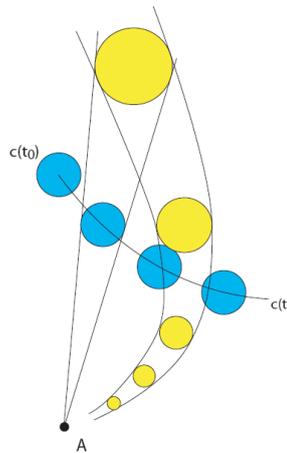


Figura 2.8: Obstáculo de Velocidades Lineal.

2.1.8. Métodos en el Espacio de Trayectorias Paramétricas (TPSM)

Más recientemente otros métodos de navegación reactiva han sido propuestos en el espacio de trayectorias paramétricas (*TPS*) para desacoplar el problema de las restricciones cinemáticas de la evitación de obstáculos permitiendo su aplicación a cualquier forma del robot [MM06, BGF08]. El *TPS* es un espacio de bidimensional donde cada coordenada polar (α, d) se corresponde a una configuración del robot (x, y, θ) en una superficie de muestreo del *CS*. La superficie de muestreo es una representación de lo que el robot es capaz de hacer teniendo en cuenta sus restricciones cinemáticas y una función de control, la misma es el resultado de unir el conjunto de trayectorias (supuestas) dibujadas por el robot cuando se aplica la función de control llamada función de generación. Algunas funciones de generación pueden producir arcos de círculo, segmentos espirales, etc. Una función de generación válida depende de un parámetro de control α , que unido con el valor de distancia d a lo largo de la trayectoria definen un solo punto en el *CS*. La figura 2.9 muestra un ejemplo de una superficie de muestreo en el *CS*.

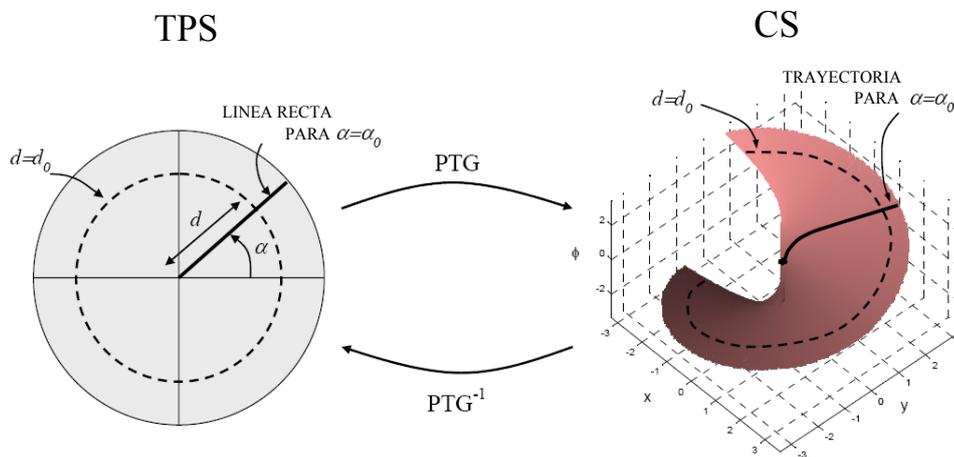


Figura 2.9: Superficie de muestreo resultante en \mathcal{C} generada a partir de una función de generación en *TPS*.

Para enderezar la superficie de muestreo solo es necesario considerar la dupla (α, d) . De esta manera el problema es más fácil de manejar ya que es planteado en un espacio con menos dimensiones. Para que el método funcione necesita incorporar la información de los obstáculos en el *TPS*, es decir que necesita obtener los puntos de intersección entre las trayectorias postura que componen la superficie de muestreo y los obstáculos en el *CS*. Ya que el procedimiento anterior tiene un elevado coste computacional se emplean con frecuencia tablas de búsqueda calculadas previamente. El procedimiento para obtener las tablas de búsqueda empieza haciendo una discretización del espacio alrededor del robot mediante una rejilla rectangular en donde cada celda almacena su obstáculo asociado en el *TPS*. En general, cada celda de la rejilla puede contener varias parejas de (α, d) , distintas parejas de (α, d) pueden generar trayectorias donde la forma del robot toca la misma celda. El uso de esta transformación presenta las siguientes ventajas: En primer lugar, el robot puede ser considerado como un punto volando libremente en el *TPS* al ser consideradas sus limitaciones cinemáticas y de forma. En segundo lugar, cualquier método clásico de evitación de obstáculos, restringido a un punto holónimo o robots circulares, puede ser aplicado a robots de cualquier forma en el *TPS*. Finalmente, un rango más amplio de caminos compatibles con las restricciones cinemáticas (un superconjunto de los arcos circulares típicos) se pueden emplear para evitar obstáculos, aumentando el espacio de soluciones a situaciones que

eran difíciles o imposibles de resolver con otros métodos.

Aunque estos métodos han tenido éxito para resolver problemas difíciles (dado que utilizan una gran diversidad de caminos), ellos no consideran la dinamicidad del entorno afectando la integridad del robot al no poder considerar el comportamiento futuro de los objetos móviles.

2.2. Aproximaciones Deliberativas

El paradigma de las aproximaciones deliberativas es usado con frecuencia para referirse a la planificación de movimientos. El problema puede definirse como la determinación a priori de la estrategia de movimiento basada en un modelo del mundo que conducirá al robot desde su posición actual hasta la posición objetivo. Los métodos deliberativos o globales pretenden producir un conjunto completo de acciones sobre el robot. La solución, cuando se encuentra, se conoce comúnmente como plan global. Estos métodos producen un camino a partir de un objetivo a alcanzar y algún conocimiento general del ambiente, usualmente información a priori en forma de un mapa. El plan global concebido por el método y aplicado al robot le conducirá hasta el objetivo.

El origen de los algoritmos de planificación de movimientos se remonta a la definición del problema clásico de mover el piano presentado en [Rei79]. No obstante, es el trabajo desarrollado en [LP83] sobre el CS , el que sienta las bases en este campo del conocimiento. Una Configuración del robot es la especificación de la posición y la orientación del sistema de referencia del robot con respecto al espacio de trabajo (WS). El espacio de configuraciones es el conjunto de todas las configuraciones posibles del robot. La potencia del CS es que el robot (\mathcal{A}) puede ser representado como un punto independientemente de su forma. Aquí los obstáculos estáticos son asignados a regiones prohibidas que representan el conjunto de configuraciones del robot donde una intersección del robot con el obstáculo ocurre en el WS . De forma que el CS queda dividido en tres subconjuntos separados: el espacio libre (\mathcal{C}_{libre}), el espacio de obstáculos ($\mathcal{C}_{obstáculo}$) y el espacio de contacto ($\mathcal{C}_{contacto}$). El problema de la planificación de movimientos en el CS consiste en encontrar un camino, si existe, que conduzca al robot desde su configuración inicial ($\mathbf{q}_{inicial}$) hasta la configuración objetivo ($\mathbf{q}_{objetivo}$).

Los algoritmos de planificación de movimientos pueden ser clasificados en algoritmos del tipo mapa de rutas, algoritmos basados en descomposición de celdas y algoritmos de muestreo.

2.2.1. Métodos basados en Mapa de Rutas

La idea principal detrás de estos métodos es capturar la conectividad de \mathcal{C}_{libre} mediante la construcción de un grafo llamado mapa de rutas. Esta representación del espacio libre reduce la dimensionalidad del problema. Una vez que el grafo ha sido construido el problema se reduce a conectar primero la configuración inicial y final al grafo y luego usar un algoritmo de búsqueda (por ejemplo, el algoritmo de Dijkstra presentado en [Dij59] o un algoritmo tipo A^* como en [HNR68]) para encontrar un camino que conecte la $\mathbf{q}_{inicial}$ y $\mathbf{q}_{objetivo}$. Estos métodos son completos, es decir, siempre van a encontrar un camino en un tiempo finito, cuando exista, de lo contrario nos hará saber en un tiempo finito, si el camino no existe. La mayor dificultad del método consiste en la construcción del grafo. Las estrategias propuestas que se han presentado

son: los grafos de visibilidad introducidos en [LPW79], los mapas de rutas de voronoi presentados en [CD88, CB00] y el método silueta mostrado en [Can88].

Estos métodos fueron concebidos originalmente para tratar con entornos estáticos. Sin embargo, ellos pueden ser usados para trabajar en entornos con objetos móviles de dos formas principalmente. La primera es representando la dinamicidad del entorno añadiendo la dimensión del tiempo al CS introducido en [ELP87] (ver la figura 2.10), o al espacio de estados (SS) presentado en [Fra98]. De cualquier manera el problema incrementa su complejidad al introducir una nueva dimensión. La segunda Opción es volver a planificar desde cero cada vez que se introduce nueva información para generar un grafo actualizado. Así un nuevo camino puede ser planificado desde la configuración actual del robot hasta la configuración objetivo a alcanzar. Lamentablemente una solución general dada por alguno de estos métodos es inviable por causa de su complejidad, que es demasiado elevada (PSPACE-hard [Can88]) y por las restricciones de tiempo.

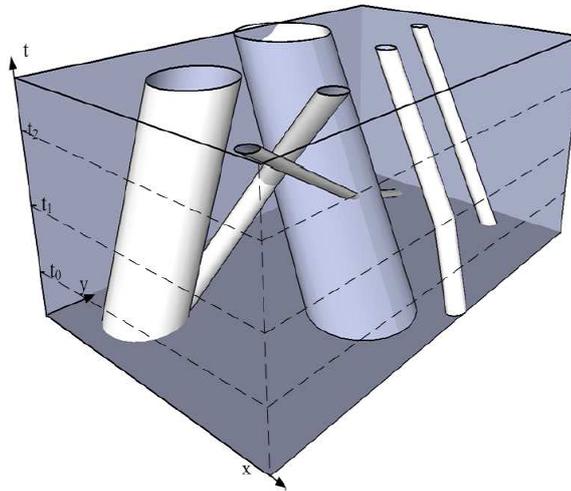


Figura 2.10: Espacio $\mathcal{C} \times \mathcal{T}$.

2.2.2. Métodos basados en Descomposición de Celdas

Estos métodos dividen el espacio libre en un número de conjuntos disjuntos llamados celdas. La relación de adyacencia entre celdas usada para buscar un camino entre dos configuraciones distintas es representada por un grafo de conectividad. Cada celda representa un nodo en el grafo y dos nodos son conectados si y solo si estos son adyacentes. Los métodos basados en descomposición de celdas pueden ser de dos tipos diferentes: exactos y aproximados. Los métodos exactos como su nombre indica generan una descomposición exacta del espacio libre (por ejemplo: trapezoidales [Cha87], curvas críticas [SS83a], algebraicos cilíndricos [SS83b] y bolas conectadas ([BK01, VKA05])). Los métodos aproximados toman la estructura de \mathcal{C} y la aproximan con celdas cuya estructura tiene forma simple como por ejemplo rectanguloides ([Elf83]).

Similar a los enfoques anteriores los métodos basados en descomposición de celdas fueron concebidos para entornos estáticos. Además, la mayoría de estos operan en espacios de dimensión baja ya que la complejidad del algoritmo de subdivisión aumenta exponencialmente con la

dimensión del espacio. De forma que, la representación del espacio de *estados-tiempo*, $\mathcal{S} \times \mathcal{T}$, no es práctica para tratar con ambientes dinámicos. La restricción de tiempo para replanificar desde cero cada vez que se detecta un cambio en el entorno por los sensores del robot lo hace computacionalmente costoso. Una alternativa a este problema consiste en reparar solo la porción del camino que se ve afectada por un cambio detectado en el ambiente. En este caso el algoritmo de subdivisión se ejecuta en la región afectada de \mathcal{C} o de \mathcal{S} , el grafo de adyacencia es reparado en consecuencia a lo anterior y un nuevo camino válido es encontrado en el grafo renovado. Los algoritmos de replanificación tales como [Ste95, KL02, LFG⁺05], corren lo suficientemente rápido como para encontrar un nuevo camino una vez que el grafo de adyacencia ha sido construido. Sin embargo la descomposición del espacio y la reparación del grafo de adyacencia son aplicables en la práctica a espacios de bajas dimensiones. Por esta razón la replanificación es aplicada mejor al espacio de \mathcal{S} y no a $\mathcal{S} \times \mathcal{T}$. Entonces, la representación del medio ambiente evolucionando en el tiempo es por lo general para una porción de tiempo y no una imagen completa del entorno dinámico con su dimensión temporal. En conclusión se puede afirmar que a pesar del esfuerzo realizado para considerar el comportamiento futuro de los objetos parcialmente, estos métodos no garantizan la integridad del robot y la navegación segura.

2.2.3. Métodos basados en Muestreo

A diferencia de los métodos basados en mapa de rutas, estos métodos evitan la construcción de una representación explícita del espacio libre o equivalentemente el espacio de obstáculos (*Obstáculo*). En su lugar, estos llevan a cabo una búsqueda que explora \mathcal{C} con una estrategia de muestreo como en la figura 2.11. Para tal fin un verificador de colisiones esta a cargo comprobar si una muestra pertenece al espacio libre u ocupado. Estos métodos han demostrado ser útiles en espacios de dimensión alta, donde es difícil hacer una discretización del espacio por descomposición de celdas o usar eficientemente un método basado en mapa de rutas. También satisfacen una forma débil de completitud puesto que muchos se basan en una estrategia de muestreo aleatorio resultando ser probabilísticamente completos, es decir que la probabilidad de encontrar una solución existente tiende a uno a medida que el número de muestras aumenta. Algunos ejemplos de este clase de métodos son los siguientes: planificador de caminos aleatorio (RPP) presentado en [BL91], el hilo de Ariadna introducido en [BATM93], los planificadores de mapas de ruta probabilísticos (PRM) expuesto en [KSLO96] y árboles aleatorios de exploración rápida (TSR) realizado en [Ste98, LK01].

Como estos métodos pueden trabajar en espacios de dimension alta también pueden operar en el espacio $\mathcal{S} \times \mathcal{T}$ para representar un entorno dinámico. El método presentado en [HKLR02] es un planificador basado en PRM, el cual considera las limitaciones de movimiento en la cinemática y/o dinámica del robot con un sistema de control, que hace el muestreo del espacio de estados-tiempo del robot seleccionando las entradas de control al azar e integrando las ecuaciones de movimiento que resultan en mapa de rutas probabilístico. El mapa de rutas no se calcula previamente, en su lugar un nuevo mapa de rutas que conecta el estado inicial y el objetivo se construye a partir de cero en cada consulta de planificación. En [vdBO05] un mapa de rutas es calculado previamente para la parte estática del medio ambiente sin tener en cuenta ni los obstáculos dinámicos ni la dimensión del tiempo. En la fase de consulta, el método solo tiene que lidiar con los objetos dinámicos cuando busca una trayectoria entre la configuración inicial y el objetivo a alcanzar. El algoritmo usa un enfoque de dos niveles para encontrar una trayectoria. A nivel local, las trayectorias en los arcos individuales del mapa de rutas son encontradas en

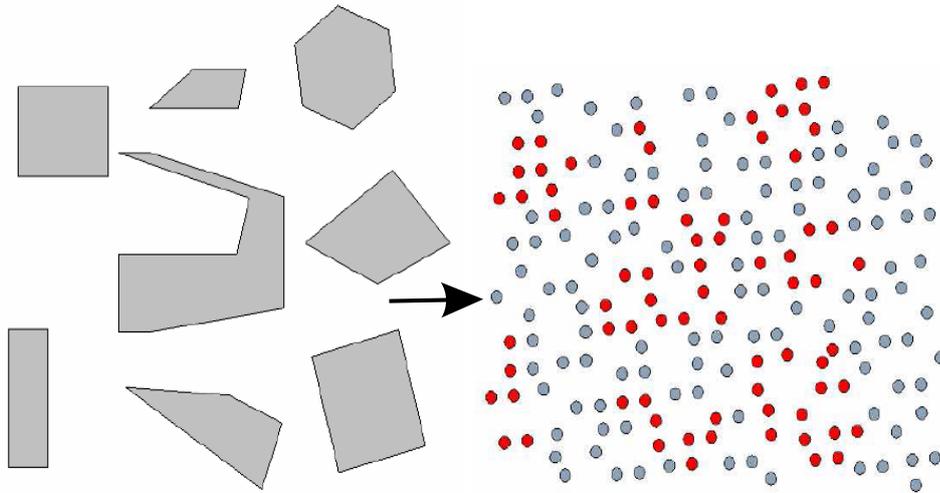


Figura 2.11: Representación aproximada por muestreo.

una rejilla implícita en el espacio de estados-tiempo. El espacio de estados-tiempo se discretiza en una forma similar a [vdBO05], pero utiliza una más eficiente búsqueda primero en profundidad. A nivel global, las trayectorias locales son coordinadas mediante un tipo de búsqueda A^* para encontrar una trayectoria global cercana a aquella óptima en tiempo en el mapa de rutas completo. El trabajo presentado en [BV03], es una extensión de RRT en el cual se intercalan la planificación y la ejecución. Este trabajo introduce dos aspectos adicionales al planificador: la cache tipo "*waypoint*" para replanificación y una búsqueda adaptativa de coste con penalización. La cache sirve para usar un plan que fue encontrado en una iteración anterior como guía para la iteración actual. La búsqueda adaptativa se basa en la idea de tener un plan no muy bueno que es mejor que nada, y una vez que un plan está en la cache, la búsqueda es sesgada con el fin de mejorarlo. Por último, el TSR para cualquier momento presentado en [FS06] tiene por objetivo la reparación de un plan local calculado con el RRT clásico mediante la eliminación de los nodos invalidados, y realizar una nueva búsqueda para añadir nuevos nodos que preservan el camino entre las configuraciones inicial y final.

Aunque estas técnicas de planificación pueden considerar el comportamiento futuro de un entorno dinámico, lo cierto es que ellas no pueden acotar el límite superior del tiempo de ejecución. De manera que parece poco probable que una restricción de tiempo fuerte pueda cumplirse en situaciones reales.

2.3. Otros Esquemas Alternativos de Planificación de Movimientos

Son aquellos métodos de planificación de movimientos que no encajan en ninguna de las dos aproximaciones presentadas anteriormente. Como su nombre sugiere, en ellos se combinan características de ambos para hacer frente al problema de planificación con una perspectiva distinta.

2.3.1. Métodos de Deformación

Los métodos de deformación se remontan al concepto de las bandas elásticas presentado en [QK93]. El método trabaja en una primera etapa haciendo uso de un algoritmo de planificación de movimientos que otorga al robot un camino libre de colisión uniendo las configuraciones inicial y final. El camino esta basado en cualquier clase de información o conocimiento previo del entorno que sea disponible. Luego, durante la etapa de ejecución el camino es deformado en la medida en que se incorpora una nueva información adquirida del ambiente. Dos clases de fuerzas se ejercen en el camino. Las que vienen de parte de los obstáculos que lo apartan del camino, y las que provienen de los restricciones del robot procurando mantener la conectividad del camino, es decir, que siguiendo el robot el camino sigue siendo realizable dada sus limitaciones dinámicas (ver figura 2.12). En los primeros trabajos solo fueron considerados sistemas holónomos, pero los trabajos presentados en [KJCL97, LBL04] extendieron el método a sistemas no holónomos. Una desventaja presentada en estos métodos es que no consideran la dimensión del tiempo. Las situaciones que deforman exageradamente el camino, como por ejemplo un objeto que se esta atravesando, pueden evitarse simplemente deteniendo el robot y permitiendo que el obstáculo pase. Algunas extensiones más prometedoras que operan en $\mathcal{C} \times \mathcal{T}$ y $\mathcal{S} \times \mathcal{T}$ han sido propuestas para tratar estos problemas. En [KF07] para robots holónomos y en [DF08] para no holónomos. No obstante, el mantener la conectividad de la trayectoria en el espacio $\mathcal{S} \times \mathcal{T}$ requiere el uso de algoritmos de generación de trayectorias relativamente complejas, que siguen siendo motivo de investigación pues es un problema abierto. Estos algoritmos deben ser eficientes para dar cuenta de las restricciones impuestas por aquellas limitaciones impuestas por la dinamicidad del entorno en los tiempo de decisión. Además no hay ninguna restricción impuesta sobre el límite superior de tiempo para el cómputo de la trayectoria o camino inicial.

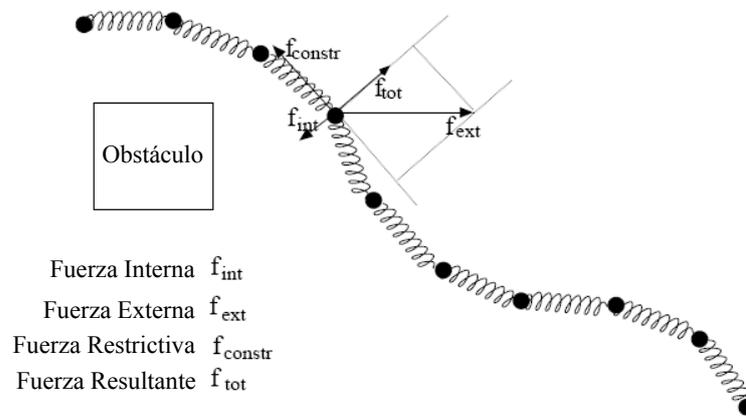


Figura 2.12: Modelo de Bandas Elásticas.

2.3.2. Planificación de Movimientos Parcial (PMP)

La planificación parcial de movimientos introducida en [PF05] es un esquema de planificación que se adapta perfectamente a estas aproximaciones intermedias. En ella no se calcula una secuencia completa de movimientos hasta el objetivo como en los esquemas deliberativos. Tampoco calcula el siguiente comando de movimiento como hacen las aproximaciones reactivas. En su lugar, este método considera la restricción impuesta en el tiempo de decisión para calcular tantos

movimientos como le sean posibles hasta la meta dentro del tiempo disponible. Cuando el tiempo de decisión es superado este devuelve la mejor secuencia de movimientos parcial hasta ahora. El método opera por etapas que se realizan en cada paso de tiempo. La primera es la ejecución del plan del paso anterior. Paralelamente, el modelo sobre el comportamiento futuro se actualiza con base en nuevas observaciones. A continuación, un algoritmo de planificación de movimientos se ejecuta para avanzar y desarrollar el plan hacia el objetivo. Si este no es alcanzado, el camino más próximo al objetivo es puesto en curso.

2.3.3. Métodos de Replanificación

Estos métodos se caracterizan porque el algoritmo de planificación va frecuentemente calculando nuevos planes como respuesta a la aparición de eventos inesperados en el entorno dinámico. El concepto de replanificación es justificado por el hecho de que en ambientes desconocidos y con obstáculos dinámicos, la información conocida a priori y sobre la cual se basa un plan inicial tiene una alta probabilidad de cambiar durante la ejecución del plan. Algoritmos como el presentado en [BH95], el D^* introducido en [Ste93] y su versión más simple el $D-Lite^*$ presentado en [KL02] son capaces de volver a planificar con rapidez sobre la base de las observaciones más recientes. Cada vez que un obstáculo en movimiento invalida parte del plan, el algoritmo busca un camino alternativo que repara el camino solución hacia la meta. De esta manera el plan anterior se descarta y una nueva partida desde el estado actual es creado. A pesar de que estos planificadores respetan la restricción de decisión de tiempo durante su ejecución, asumiendo que un plan inicial se da, sin embargo no tienen ningún límite superior de tiempo establecido que el algoritmo pueda tomar para dar con esta solución inicial.

Capítulo 3

Planificando movimientos en entornos dinámicos usando ODV

En este capítulo presentamos la estrategia usada para llevar a cabo la planificación de movimientos en el robot, la cual se basa en la idea de mapear los obstáculos (dinámicos y estáticos) del entorno del robot desde el espacio de trabajo (WS) al espacio de velocidades del robot (VS). La información mapeada en este espacio es calculada a partir de dos conceptos: los tiempos a colisión con el objeto y los tiempos para escapar de la colisión. Luego de mapear el ambiente del robot en el VS, el mejor comando de velocidad del robot puede ser calculada aprovechando que la información sobre la dinamicidad del entorno queda directamente reflejada en el modelo. Este modelado del entorno en el VS facilita la elección del tipo de maniobras llevadas a cabo por el robot, aplicando diversos criterios de evitación de colisiones como por ejemplo: pasar por delante o dejar pasar permitiendo que los objetos pasen delante del robot.

Esta aproximación difiere de otras en varias características:

- *Las velocidades del robot son calculadas directamente en el espacio de velocidades VS, no en el espacio de configuraciones, CS. Esto es una ventaja cuando se desean calcular trayectorias óptimas, el razonamiento en términos de velocidad contra el razonamiento en términos de caminos.*
- *El método toma en cuenta las restricciones que limitan las capacidades de movimiento del robot (cinemáticas y dinámicas), y ellas son reflejadas directamente en el VS.*
- *El mapeo del ambiente al espacio de velocidades es obtenido mediante expresiones analíticas. De manera que no es necesario el uso de rejillas como en otras aproximaciones. Además, las velocidades elegibles se extienden a todo el espacio completo de velocidades, dentro de los límites físicos, y ninguna discretización es realizada en este sentido, como en otros métodos antes mencionados.*
- *El espacio de velocidades incluye implícitamente la información de tiempo sobre las potenciales colisiones y las posibilidades de escapar de una posible colisión, con los objetos del entorno. El modelo permite razonar sobre los movimientos futuros del robot dentro de un horizonte de visibilidad más allá del correspondiente a un periodo de muestreo, facilitando la búsqueda sobre el mejor comando de velocidad.*

- *Sobre este espacio pueden ser aplicadas técnicas de optimización, usando optimización global o aplicando heurísticas para encontrar la velocidad óptima en tiempo real. Esta última aproximación será presentada en este capítulo.*

El método es aplicado iterativamente, de forma que en cada periodo de muestreo el espacio de configuraciones del robot es calculado y mapeado al espacio de velocidades, donde el mejor comando de velocidad es buscado razonando en términos de velocidad. Durante el siguiente periodo de muestreo el procedimiento es repetido. Así, la trayectoria completa es calculada como una secuencia de pequeños tramos de trayectoria que convergen al objetivo a alcanzar. En este trabajo un planificador global, como el presentado en [Ste93], puede ser usado ya que toma en cuenta los cambios en el ambiente, y se ajusta bastante bien con el objetivo de la navegación en ambientes parcialmente conocidos que cambian. La aproximación considera que la localización de los objetos es proporcionada por un sensor laser de distancia (por ejemplo un laser del tipo range-finder) y las velocidades de los objetos son calculadas a partir de dicha información.

En este capítulo primero presentaremos la técnica usada para mapear los obstáculos dinámicos desde el espacio de configuraciones al espacio de velocidades del robot. Después explicaremos como nuestro planificador básico de movimientos obtiene las velocidades que se aplican al robot para alcanzar el objetivo propuesto.

3.1. Mapeando los obstáculos móviles del *CS* al *VS*

El método propuesto para mapear los obstáculos móviles esta basado en calcular las trayectorias del robot que podrían provocar colisiones con los obstáculos que contiene el entorno del robot. Los tiempos de colisión y escape con estos obstáculos corresponde a información contenida de manera implícita en el espacio mapeado, que es muy relevante para mantener la integridad y seguridad del robot conforme a los criterios de seguridad presentados en [Fra07] en cuanto al comportamiento futuro de los obstáculos móviles y su propia dinámica.

Para reducir la complejidad del método se han impuesto algunas restricciones a saber:

- El robot se mueve siguiendo caminos circulares o rectos. Esta es una restricción común impuesta para movimientos en robots no holónomos. Asumiendo que las velocidades lineales y angulares son constantes durante un periodo de muestreo y el proceso es repetido para el siguiente periodo, esta es una restricción coherente y razonable. De esta forma el método toma en cuenta las restricciones cinemáticas.
- Los objetos se mueven con velocidad constante siguiendo caminos de segmentos de línea recta. Aunque esto parece una restricción fuerte, la verdad es que al ser repetido el proceso para cada periodo de muestreo, el método puede considerar el movimiento de los objetos como una secuencia de caminos constituidos por pequeños segmentos rectilíneos. En cualquier caso el método presentado puede ser fácilmente extendido a otra clase de caminos sin pérdida de generalidad. Con el fin de dar claridad al método y disminuir su tiempo de ejecución se implementa la aproximación usando esta restricción.
- Los objetos son representados como polígonos adquiriendo formas cuadradas o rectangulares. Esta aproximación hace más fácil su cálculo y proporciona un margen de seguridad al calcular los tiempos de colisión y escape manteniendo la integridad del robot. Además, el robot adquiere forma circular simplificando los cálculos en el *CS*.

3.1.1. Cálculo de los tiempos a colisión a los obstáculos

El método es implementado usando un sistema de referencia atado al robot de manera que todos los cálculos realizados son hechos en base a este sistema coordinado local de referencia. La figura 3.1a representa el espacio de trabajo (WS) con un robot y un obstáculo móvil a velocidad constante siguiendo un camino rectilíneo. El método mapea la información del entorno al espacio de configuración (CS) como se ve en la figura 3.1b. En ella se muestra la *Banda de Colisión* (zona barrida por el obstáculo móvil a lo largo de su trayectoria rectilínea) y el obstáculo en dos localizaciones \mathbf{x}_o^1 and \mathbf{x}_o^2 que representan las localizaciones en las cuales el robot, siguiendo la trayectoria circular r_j , llega a el punto P_{c1j} en el tiempo t_{1j} justo cuando el obstáculo ha pasado o escapa de la colisión cruzando justo antes que el obstáculo arribe al punto P_{c2j} en el tiempo t_{2j} . El cómputo de esta pareja de puntos y sus tiempos asociados, que pueden llevar a una colisión o escapar de esta, constituyen la base del modelo propuesto.

A partir de la localización del obstáculo $\mathbf{x}_o = (x_o, y_o, \phi_o)$ en el sistema de referencia del robot y su velocidad \mathbf{v}_{ob} , los puntos de colisión $P_{c1j}(x_{1j}, y_{1j})$ y $P_{c2j}(x_{2j}, y_{2j})$ con sus correspondientes tiempos t_{1j} y t_{2j} son calculados resolviendo las ecuaciones para cada trayectoria circular, caracterizada por su radio de curvatura r_j y su centro $(0, y_{cj})$ en el sistema de referencia del robot R . Entonces para cada r_j, t_{ij} ($i=1,2$) es calculado,

$$\begin{aligned} x_{ij} &= x_0 + v_0 \cos(\phi_0) t_{ij} \\ y_{ij} &= y_0 + v_0 \sin(\phi_0) t_{ij} \\ r_j^2 &= x_{ij}^2 + (y_{ij} - y_{cj})^2 \end{aligned}$$

obteniendo dos soluciones para cada punto de colisión P_{cij} ,

$$\begin{aligned} t_{ij} &= \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \\ A &= v_0^2 \\ B &= 2v_0(x_0 \cos(\phi) + y_0 \sin(\phi)) - 2v_0 \sin(\phi) y_{cj} \\ C &= x_0^2 + y_0^2 + y_{cj}^2 - 2y_0 y_{cj} - r_j^2 \end{aligned}$$

Al observar la situación del robot en relación al obstáculo móvil de la figura 3.1b dos casos pueden aparecer. Cuando el robot está fuera de banda de colisión el método selecciona t_i (a continuación eliminamos el subíndice j para mayor claridad) como la solución correspondiente al primer punto de intersección (el valor más pequeño de t_i) para ambos P_{c1} y P_{c2} . A partir de los tiempos t_1 y t_2 , las velocidades angulares del robot son calculadas como $w_i = \theta_i/t_i$ donde θ_i es el desplazamiento angular sobre el camino r para alcanzar P_{ci} . Luego las velocidades lineales son calculadas como $v_i = r w_i$. De esta forma se calculan las velocidades $\mathbf{v}_1, \mathbf{v}_2$ y sus correspondientes tiempos t_1 and t_2 . Cuando el robot está dentro de la banda de colisión, la estrategia seguida por el robot es escapar de la banda de colisión antes que el objeto llegue, evitando así la colisión. En este caso solo es necesario calcular el punto de escape P_{c2} y la velocidad \mathbf{v}_2 usando las anteriores ecuaciones. Una velocidad baja podría terminar en colisión. De manera que $\mathbf{v}_1 = \mathbf{0}$ and $t_1 = 0$

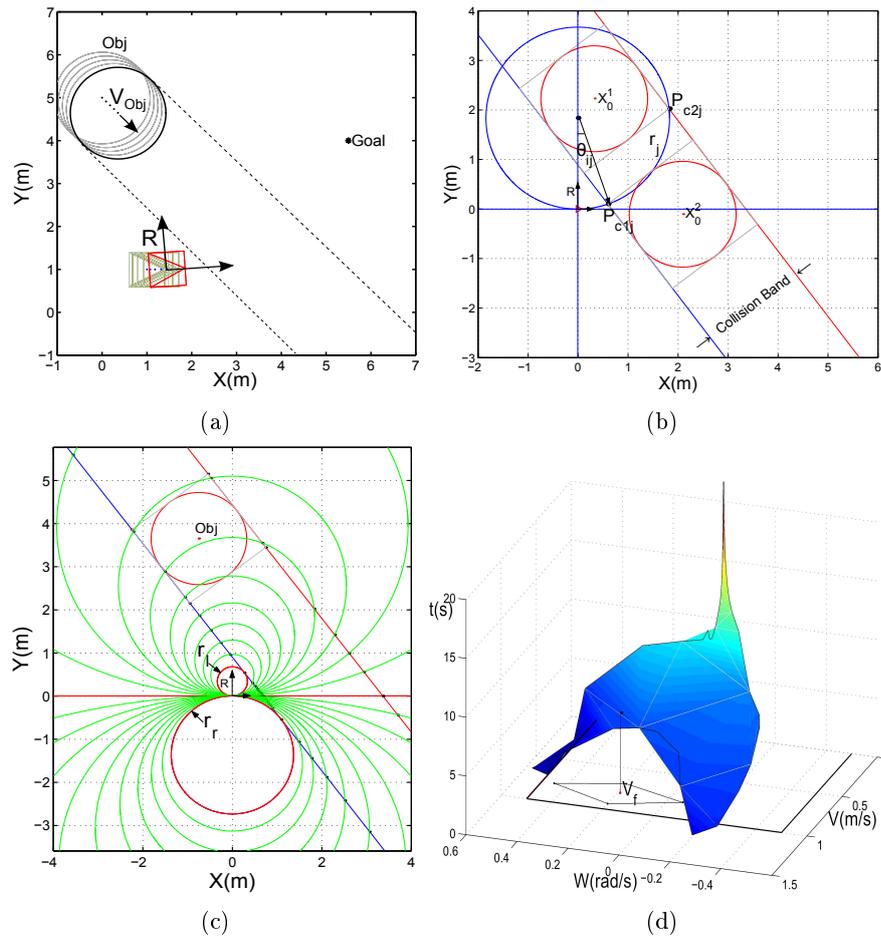


Figura 3.1: (a) El espacio de trabajo (WS) (b) La banda de colisión, el camino r_j y los puntos de colisión P_{c1j} and P_{c2j} en el espacio de configuración (CS) (c) Extensión a todo el rango de caminos circulares con riesgo de colisión (d) ODV en el espacio de velocidades-tiempo (DVS).

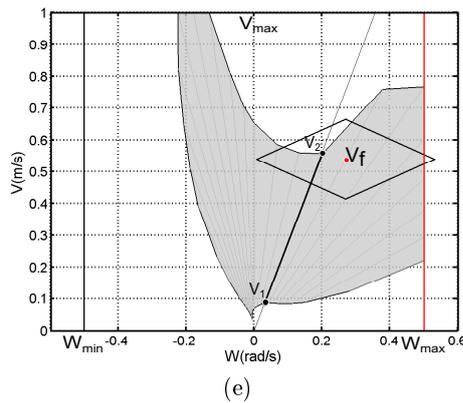


Figura 3.2: (a) Proyección del ODV, V_{ODV} , en el plano (v, w)

son escogidos.

Los cálculos son extendidos a todo el espacio completo, considerando un rango de radios de

curvatura entre r_l y r_r (ver la figura 3.1c). La figura 3.1d representa el espacio de *velocidades-tiempo* obtenido a partir de los cálculos anteriores. La superficie es construida a partir de las parejas (v_1, w_1, t_1) (v_2, w_2, t_2) pertenecientes al contorno de la superficie, obtenida de cada trayectoria r_j . Note como los caminos circulares son transformados en líneas rectas ($r = v/w$). En esta figura, \mathbf{v}_1 y \mathbf{v}_2 son las velocidades extremo calculadas para el camino circular r_j . En adelante llamaremos a esta superficie *Objeto Dinámico de Velocidades (ODV)*. La figura 3.2 muestra la proyección del ODV en el plano (v, w) , V_{ODV} . Formalmente,

$$ODV = \{(v, w, t) | (v, w) \in [INT(V_{ODV}) \cup \delta V_{ODV}]\}$$

Donde INT denota el conjunto de velocidades que pertenecen a la zona V_{ODV} y δ las velocidades del contorno de V_{ODV} . Las velocidades abajo del ODV o fuera de su proyección son velocidades elegibles por el robot, porque ellas corresponden a velocidades que aplicadas al robot no conducen a colisión con el obstáculo.

El contorno de V_{ODV} representa los límites de velocidad \mathbf{v}_1 and \mathbf{v}_2 para cada camino circular evaluado como fue explicado arriba. Note que las velocidades en el ODV no corresponden a velocidades prohibidas (no deben ser tratadas como obstáculos en el sentido de la navegación reactiva). En otras palabras, la velocidad \mathbf{v}_f en las figuras 3.1d y 3.2 podría ser escogida si el tiempo t_f en el cual es aplicada fuese menor que el tiempo de colisión sobre la superficie ODV (esto es, si el está por debajo de la superficie). Esta es una muy importante consideración para calcular la mejor trayectoria en este espacio. Los objetos estáticos son calculados de manera similar, pero en este caso todas las velocidades pertenecientes al V_{ODV} están prohibidas. Esta circunstancia es tomada en cuenta por el método para calcular el comando de velocidad del robot.

La extensión del proceso para múltiples objetos es directa. Los cálculos son realizados para cada objeto. La unión de todas las superficies de cada ODV constituye el *Conjunto de Objetos Dinámicos de Velocidades (ODVS)*. El ODVS representa todas las velocidades para las cuales habría colisión si ellas fueran mantenidas en el tiempo, V_{ODVS} es su proyección en VS y V_{free} es el conjunto de velocidades sin riesgo a colisión,

$$ODVS = \cup_{i=1}^m (ODV_i)$$

$$V_{free} = \{(v, w) | (v, w) \in [V_{adm} \ominus V_{ODVS}]\}$$

Donde V_{adm} es el conjunto de velocidades admisibles limitadas por v_{max} y w_{max} , y \ominus es la operación del conjunto diferencia.

En adelante llamaremos *Espacio de Velocidades Dinámico (DVS)* al espacio de *velocidades-tiempo* incluyendo el ODVS y la información V_{free} . En el siguiente apartado usaremos diferentes estrategias para calcular el mejor movimiento. Note que las estrategias a implementar pueden elegir velocidades en ambas zonas, V_{free} y V_{ODV} .

3.1.2. Considerando las Restricciones Dinámicas del Robot

En el apartado anterior, fue asumido que cualquier velocidad \mathbf{v}_1 , \mathbf{v}_2 puede ser alcanzada en un solo paso (un periodo de muestreo T), lo cual no es real. La velocidades máximas que pueden

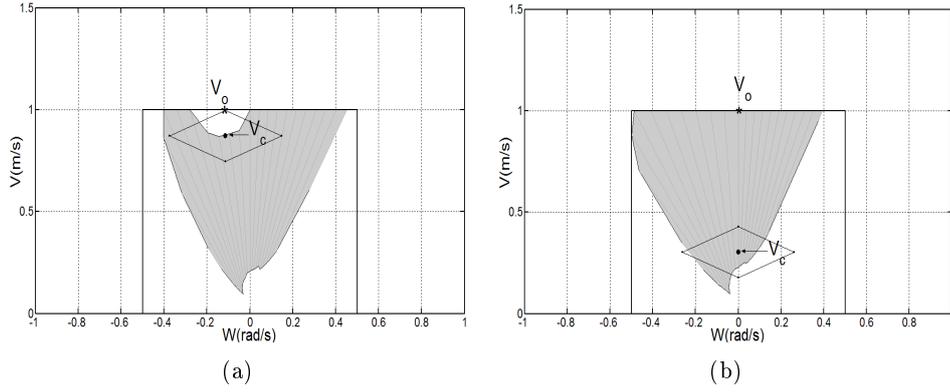


Figura 3.3: (a) \mathbf{v}_o es alcanzable solo dentro de la ventana, (b) \mathbf{v}_o no puede ser alcanzada

ser alcanzadas son solo aquellas que están dentro de la ventana de velocidades cuyo centro está en la velocidad actual, \mathbf{v}_c , estas son $(v_c \pm \Delta v_{max}, w_c \pm \Delta w_{max})$, y representan las restricciones dinámicas es decir las restricciones en la aceleración del robot. La figura 3.3a muestra que \mathbf{v}_o puede ser alcanzada en un solo paso desde las velocidades dentro de la ventana, debido a las restricciones dinámicas. En la figura 3.3b puede verse como \mathbf{v}_o no puede ser alcanzada en un solo paso, porque está fuera de la ventana. Para tratar con este problema, son calculados los tiempos necesitados para alcanzar los límites \mathbf{v}_1 y \mathbf{v}_2 desde \mathbf{v}_c en DVS , t'_1 y t'_2 respectivamente. Estos tiempos son usados para volver a calcular los límites de velocidad ($\mathbf{v}'_1, \mathbf{v}'_2$). El computo de t'_1 y t'_2 es una forma conservativa de considerar las restricciones dinámicas en DVS . Más formalmente el procedimiento puede ser expresado así:

1. Calcular cuántos periodos de muestreo son necesarios para alcanzar las velocidades \mathbf{v}_1 y \mathbf{v}_2 desde \mathbf{v}_c . Estos son t'_1 and t'_2 .
2. Calcular los nuevos límites de velocidad ($\mathbf{v}'_1, \mathbf{v}'_2$) a partir de t'_1 y t'_2 , respectivamente, usando las mismas ecuaciones presentadas anteriormente. De esta manera las restricciones dinámicas son consideradas para tener límites seguros en las velocidades de colisión.

La figura 3.3b representa el DVS cuando las restricciones dinámicas son tenidas en cuenta, En este caso \mathbf{v}_o conduce a una colisión, de manera que no puede ser elegida. Este procedimiento permite aumentar el horizonte de visibilidad en DVS para velocidades elegibles en más de un periodo de muestreo.

3.2. Calculando trayectorias usando el DVS

Diferentes técnicas para encontrar las velocidades óptimas del robot pueden ser aplicadas usando el modelo basado sobre el DVS . Nosotros desarrollamos en esta sección un método heurístico que explota información contenida en el DVS . El método es aplicado en tres pasos: primero, el objetivo suministrado por el planificador global es mapeado en el DVS , \mathbf{v}_G . Segundo, un comando de velocidad, \mathbf{v}_o , es planificado en el V_{free} como una función de \mathbf{v}_G para conducir el robot hacia el objetivo. Tercero, la consigna de velocidad en el periodo de muestreo es calculada, considerando las restricciones en la dinámica del robot.

■ Paso 1: Mapeando el Objetivo

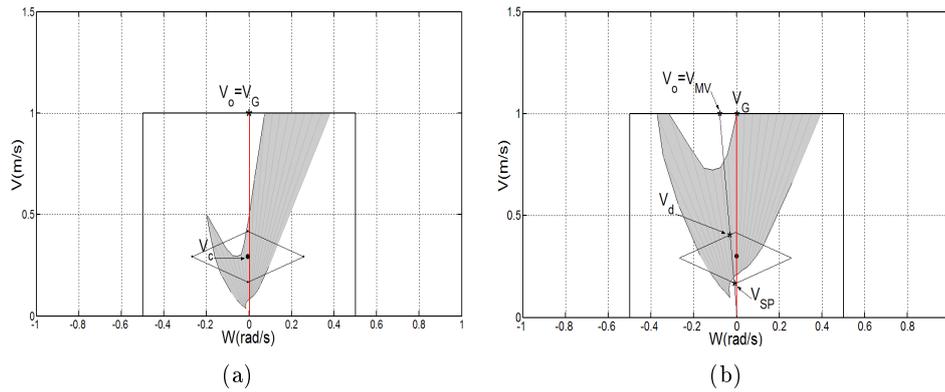


Figura 3.4: (a) La estrategia elegida es alcanzar $\mathbf{v}_o = \mathbf{v}_G = (v_{max}, 0)$, (b) \mathbf{v}_{MT} conduce al robot a moverse alrededor del obstáculo a máxima velocidad lineal y \mathbf{v}_{SP} genera un movimiento rectilíneo (el camino más corto) a baja velocidad. \mathbf{v}_d es la consigna de velocidad.

Para mapear el objetivo a alcanzar en el *DVS*, se calcula el ángulo entre la orientación actual del robot y la dirección del objetivo (ψ) en el sistema de referencia local. A partir de este ángulo se obtiene la velocidad angular, $w = \psi/T$, para alinear el robot y el objetivo, siendo T el periodo de muestreo. Bajo los criterios de máxima velocidad lineal (v_{max}) y velocidad máxima de giro, si $w \geq w_{max}$, entonces $w = w_{max}$. La pareja (v_{max}, w) define la velocidad, \mathbf{v}_G . Esta política trata de alinear el robot con el objetivo tan rápido como sea posible en ausencia de obstáculos. Si hay obstáculos, \mathbf{v}_G es solo una guía inicial para seleccionar la mejor velocidad para el robot. Obviamente solo aquellas velocidades compatibles con la particular cinemática y dinámica del vehículo (conducción diferencial, conducción tipo coche) pueden ser seleccionadas.

■ Paso 2: Planificando la velocidad

La velocidad del robot \mathbf{v}_o es calculada considerando que \mathbf{v}_G ha sido previamente mapeado. La política será alcanzar la velocidad tan pronto sea posible, de forma que la mejor velocidad a escoger, \mathbf{v}_o , será la más cercana a \mathbf{v}_G en V_{free} . En la figura 3.4a una situación es representada. Una clara estrategia óptima es escoger $\mathbf{v}_o = \mathbf{v}_G = (v_{max}, 0)$. Esta estrategia permite que el robot pase antes que los objetos móviles lleguen evitando una colisión. Otra situación es mostrada en la figura 3.4b. En el *DVS* se observa que $\mathbf{v}_o = \mathbf{v}_G = (v_{max}, 0)$, no puede ser seleccionada porque esta conduce a colisión. Así que, la velocidad $\mathbf{v}_{MT} = (v_{max}, w_{MT})$, que es la más cercana a \mathbf{v}_G , o $\mathbf{v}_{SP} = (v_{SP}, 0)$ pueden ser elegidas, siendo que ambas velocidades $\in V_{free}$ y $v_{SP} < v_{max}$, $w_{MT} \neq 0$. La velocidad \mathbf{v}_{MT} genera una trayectoria circular alrededor del obstáculo móvil, conduciendo el robot a máxima velocidad lineal haciendo prevalecer un tiempo mínimo. Por otro lado, la velocidad \mathbf{v}_{SP} genera un camino rectilíneo, usando una velocidad lineal menor que la v_{max} . Esta produce un camino más corto (rectilíneo, esperando que el obstáculo pase). Los tiempos asociados t_{MT} and t_{SP} , incluidos implícitamente en *DVS*, también pueden usarse para establecer algún criterio que permita tomar la mejor decisión sobre la elección del mejor comando de velocidad que puede ser aplicado al robot.

■ Paso 3: Calculando la consigna \mathbf{v}_d

Como fue explicado arriba, las restricciones dinámicas limitan las velocidades del robot que pueden ser alcanzadas en un solo periodo de muestreo. Por tanto la consigna de ve-

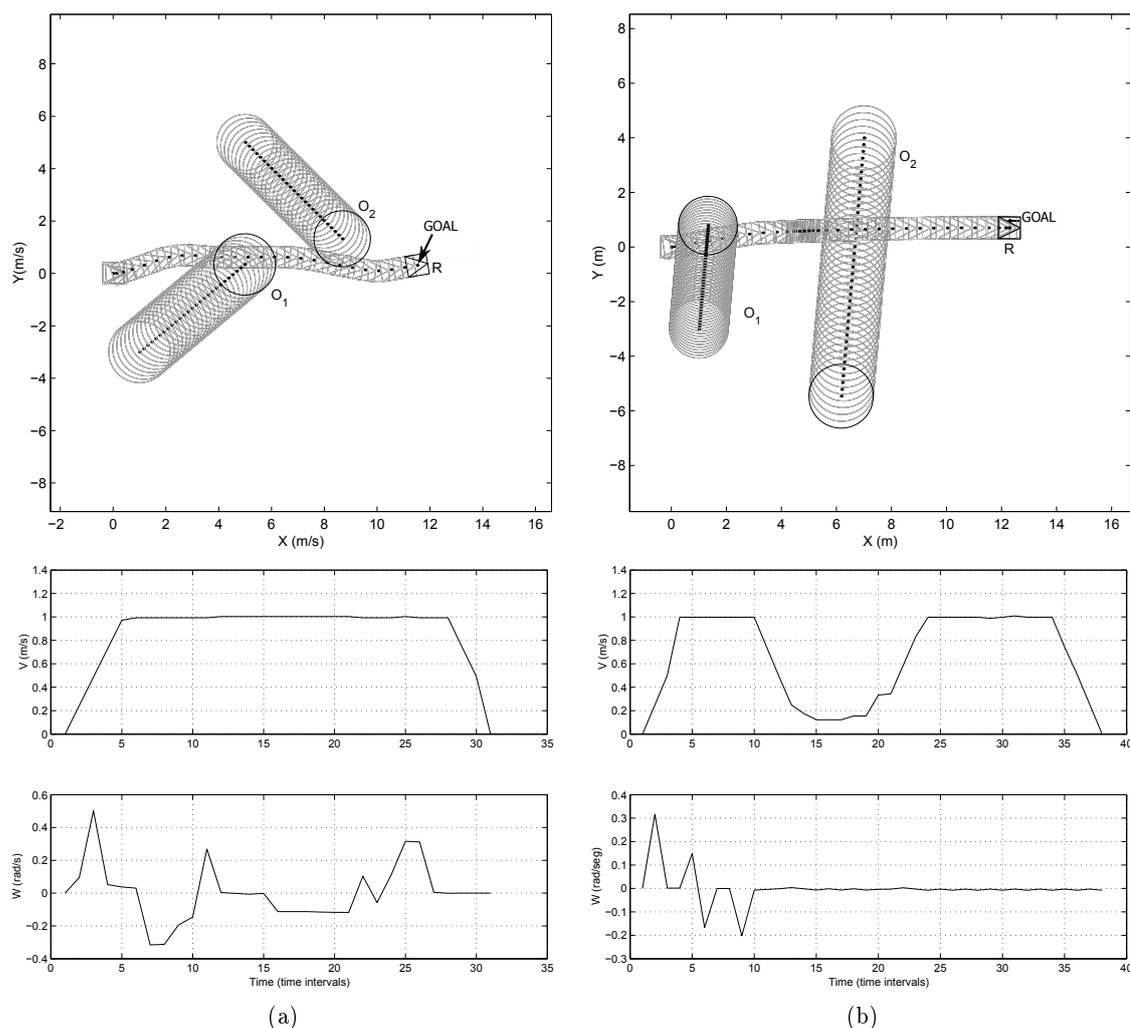


Figura 3.6: (a) Experimento 1: criterio MT y perfil de velocidades (b) Experimento 2: criterio SP y perfil de velocidades

cual el robot sigue un camino más recto, más corto que en el experimento 1 pero con un tiempo total mayor y una velocidad lineal más baja.

En el tercer experimento se muestra el comportamiento del robot en un escenario más complejo, con varios objetos moviéndose a su alrededor (ver figura 3.7). Cuatro objetos móviles cruzan enfrente de él. El tiene que maniobrar incrementando su velocidad lineal (cerca de los objetos O_1 y O_3) y reduciendo su velocidad (cerca de los objetos O_2 y O_4) cuando es necesario, para alcanzar el objetivo sin colisionar con los objetos y manteniendo movimientos sin comportamientos oscilatorios. Esta situación puede ser problemática para los clásicos métodos de evitación de obstáculos, produciendo movimientos oscilatorios cerca de los objetos y por lo tanto trayectorias que no son óptimas. Puede verse como el robot manteniendo máximas velocidades, gira más suavemente para pasar antes que el objeto llegue.

En el cuarto y último experimento, la estrategia de navegación fue probada sobre tres robots Pioneer 3-AT (ver características en el apéndice A y B) en el vestíbulo del edificio I3A. Los tres robots móviles se mueven restringiendo la navegación de nuestro robot. En la la figura 3.8

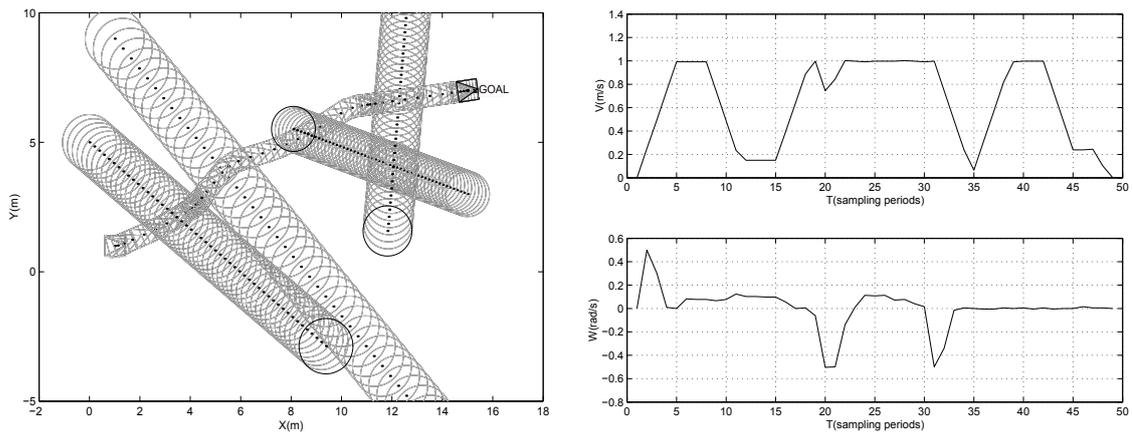


Figura 3.7: Experimento 3: navegación alrededor de cuatro obstáculos móviles y perfiles de velocidad.

se observan varios pasos del experimento. Las maniobras seleccionadas mantienen la máxima velocidad lineal durante todo el experimento mientras el robot evita los obstáculos dinámicos y estáticos mientras avanza con la mínima desviación de la dirección del objetivo a alcanzar. Los perfiles de velocidad son ilustrados en la figura 3.9a. Primero, el planificador selecciona la velocidad \mathbf{v}_{MT} para pasar secuencialmente los objetos 1 y 2. Luego el robot gira un poco para evitar el objeto 3 usando la velocidad \mathbf{v}_{MT} pasando a máxima velocidad lineal para alcanzar el objetivo en el menor tiempo posible. Los obstáculos son debidamente detectados y seguidos, siendo el movimiento del robot libre de oscilaciones (ver en la figura 3.9a). La trayectoria completa del robot es dibujada en color verde en la figura 3.10.

3.4. Conclusiones

En este capítulo se ha presentado un método para modelar el dinamismo del ambiente junto con las restricciones del robot, construyendo el espacio de velocidades dinámico (*DVS*). Este es obtenido mapeando el espacio de configuraciones al espacio de velocidades del robot. El modelo permite aplicar estrategias de navegación para calcular los comandos de velocidad del robot directamente en su espacio de velocidades, mientras los obstáculos del entorno son evitados. Para lograr ese objetivo se ha presentado una estrategia heurística, y algunos experimentos muestran la aplicación del método. Los resultados de distintas estrategias han sido discutidos también. Una característica importante de nuestro planteamiento heurístico es la habilidad del método para controlar la forma de la solución mediante el uso de distintas estrategias. El diseño de estrategias más complejas en las cuales se puedan aplicar otros criterios en la toma de decisiones es un objetivo para ser desarrollado en un trabajo futuro.

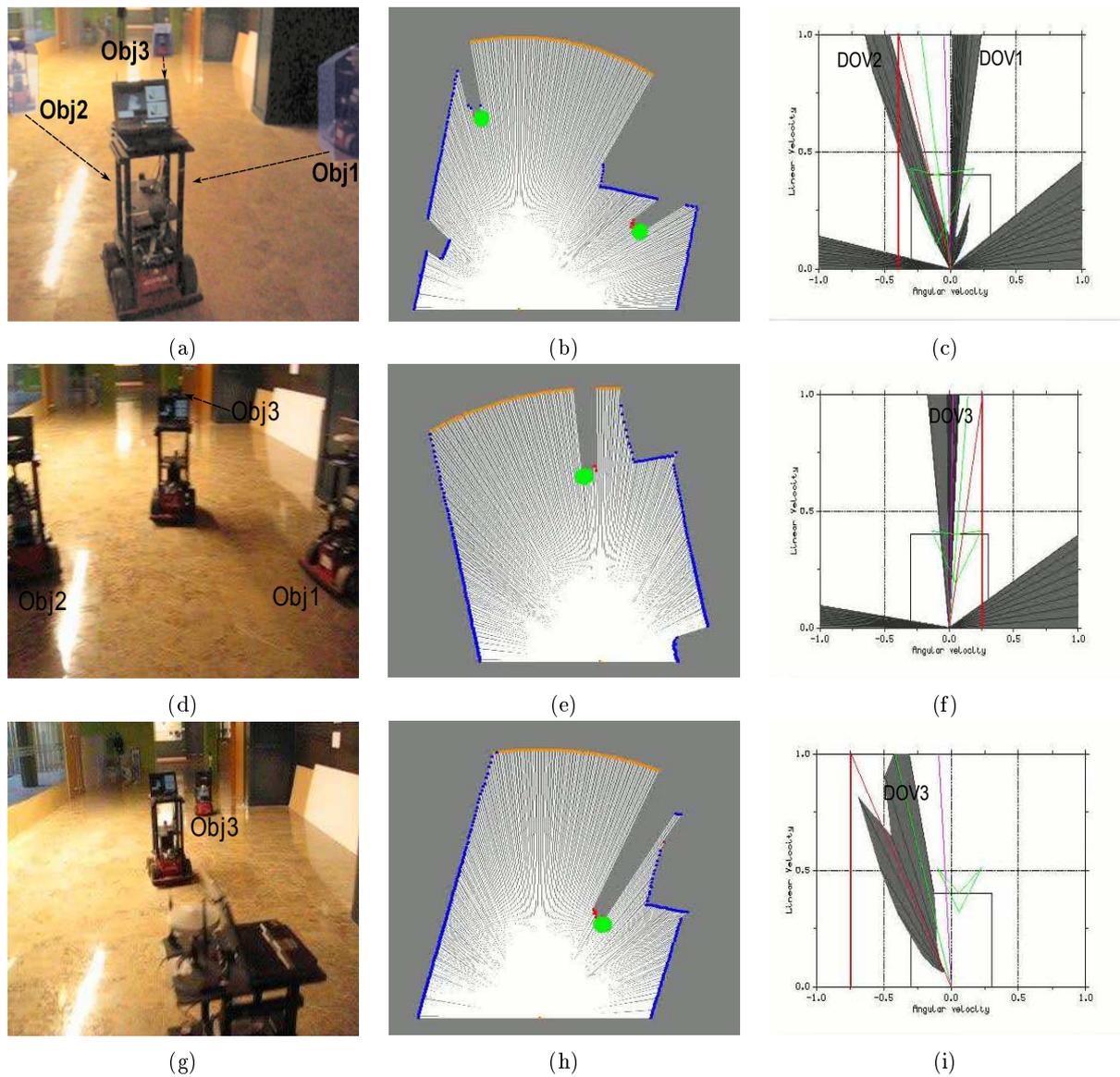


Figura 3.8: Experimento real en el vestíbulo. El robot maniobra evitando todos los obstáculos móviles, evitando posibles colisiones mientras se dirige hacia el objetivo.

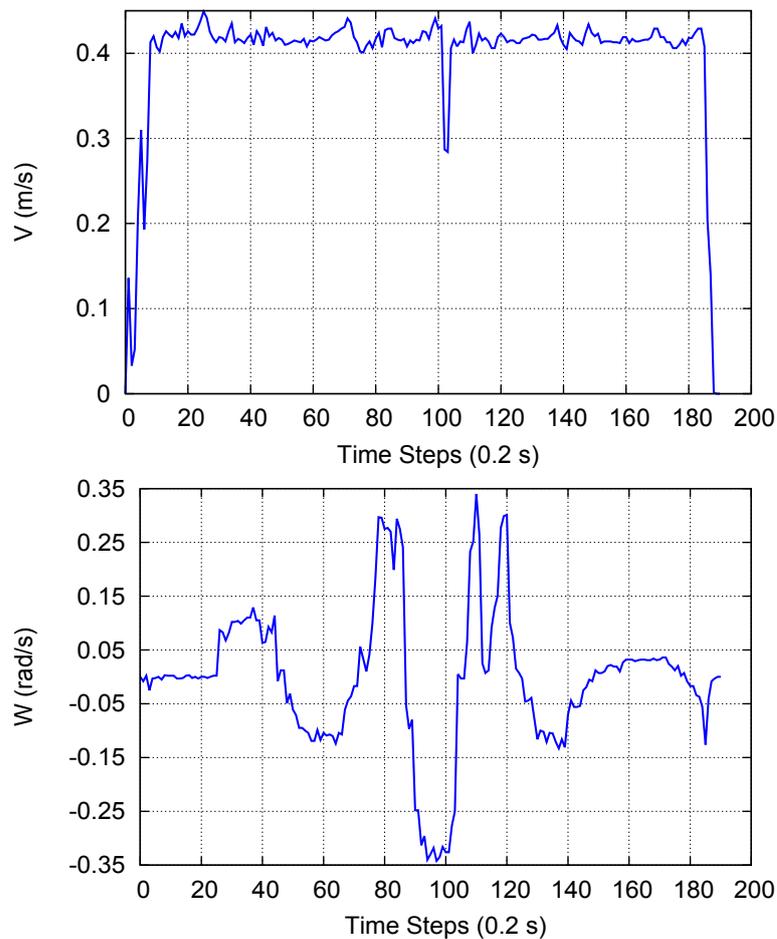


Figura 3.9: Vestíbulo del I3A: (a) Perfiles de velocidad lineal y angular.

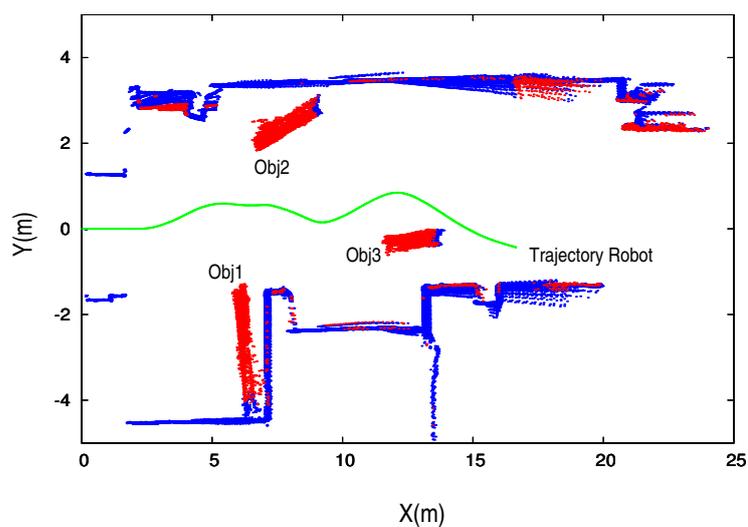


Figura 3.10: Trayectoria del robot en verde y entorno percibido por el robot.

Capítulo 4

Optimizando la Planificación en el Espacio de Velocidades

En este capítulo tratamos un método para planificar los movimientos del robot. Es competencia del método encontrar una trayectoria desde el punto de partida del robot hasta el objetivo a alcanzar que satisfaga: i. evitar los obstáculos en el entorno, ii. generar trayectorias realizables por el robot (deben considerar las restricciones cinemáticas y dinámicas) y iii. minimizar un criterio (tiempo mínimo). El problema de la evitación de colisiones en la planificación de movimientos ha sido tratado extensamente en la literatura sobre robótica (capítulo2), y los métodos más recientes basados en evitación reactiva consideran algunas de estas restricciones. Pero cuando la información de los objetos dinámicos es tomada en cuenta, el sistema de navegación puede calcular trayectorias más estables mejorando el desempeño de los movimientos del robot comparado con otros métodos clásicos de planificación de movimientos. La información obtenida a partir de los sensores del robot, la cual refleja el dinamismo del ambiente, debe ser tratada e incorporada para planear trayectorias óptimas o próximas a ellas.

De esta manera usamos una técnica para planificar los movimientos del robot que transforma el problema del espacio de configuraciones al espacio de velocidades, para luego tomar decisiones sobre la mejor estrategia de movimiento en este espacio. El método mapea, a partir de la información captada por los sensores de abordo del robot, las posiciones de los obstáculos y sus trayectorias propias o estimadas en el espacio de velocidades del robot considerando sus restricciones cinemáticas y dinámicas. Este proporciona información sobre colisiones adicionales, suministrando un mapa de todas las velocidades disponibles libres de colisión (capítulo3). Una trayectoria inicial libre de colisión es calculada en el modelo, la cual es usada como una semilla en el proceso de optimización para encontrar una trayectoria que converge al objetivo. Este proceso es repetido en cada periodo de muestreo.

El problema de la caracterización de las trayectorias de mínimo tiempo uniendo cualquier par de configuraciones, y donde el robot parte del reposo ha sido estudiado por Jacobs y otros en [JLR91]. Los autores han demostrado que las trayectorias de mínimo tiempo corresponden a trayectorias obtenidas mediante control tipo bang-bang, resultan en piezas de clotoides e involutas de círculo. Este problema es tratado en espacio libre. [RF97] se encarga del control de tiempo óptimo para robots móviles calculando controles extremos como un problema de control óptimo. En su trabajo declaran que es un problema que continúa abierto. Fleury y otros en [FSLC95], abordan el problema de suavizar los movimientos del robot, proponiendo varias estrategias subóptimas para

suavizar las trayectorias de líneas quebradas en ambientes abarrotados pero no dinámicos. Por otro lado [FS97] presenta un método para calcular la trayectoria de tiempo óptimo de un robot manipulador moviéndose en un entorno dinámico sencillo y usando el concepto del obstáculo de velocidad. La trayectoria es calculada utilizando un algoritmo de pasos descendientes (*steepest descent algorithm*). Se calculan los tiempos de interrupción para un control óptimo de tipo "bang-bang". En [LSSL02] un sistema compuesto por un método local de evitación de obstáculos de goal orientado es desarrollado usando el concepto del obstáculo de velocidad no lineal y un planificador incremental. El método utiliza un criterio que pesa las velocidades, el cambio de orientación, el riesgo a colisión y el tiempo a colisión. Pero su convergencia al objetivo es difícil de conseguir por causa de la sintonización de los parámetros de peso.

En [SF97], los caminos suaves compuestos por curvas que mantienen la continuidad de la curvatura (clotoides y arcos de círculo) son propuestos para obtener trayectorias realizables. Pero ninguna consideración temporal es hecha. [KS05] calcula trayectorias casi óptimas bajo las restricciones cinemáticas y dinámicas del robot usando trayectorias por tramos que mantienen la continuidad de la curvatura pero no en un entorno dinámico. Estas trayectorias están basadas en el trabajo de [SF97], y el costo métrico está basado en la duración del control aplicado.

4.1. Declaración del problema - La aproximación

La aproximación presentada en este capítulo se centra en la planificación de movimientos en entornos dinámicos, asumiendo que es un planificador global el que proporciona al sistema una secuencia de localizaciones (subobjetivos) para alcanzar el objetivo final (por ejemplo un algoritmo tipo D^* puede ser usado). No se trata en la aproximación el problema de la planificación global. El trabajo está centrado en calcular trayectorias para alcanzar subobjetivos minimizando un criterio de tiempo.

El problema por solucionar es planteado como un problema de optimización no lineal restringido, en el cual las restricciones surgen por la forma del robot (restricciones cinemáticas y dinámicas), pero también de la clase de caminos seleccionados para que sean seguidos por el robot y de los obstáculos móviles en el entorno. El sistema puede sacar provecho de la información obtenida del dinamismo del entorno para calcular el mejor movimiento, en términos de caminos y velocidades (que son trayectorias). El movimiento es calculado por medio de un planificador de movimientos robocéntrico en el espacio de velocidades, que permite planear las trayectorias en cada periodo de muestreo. El planificador es descrito en la sección 4.2. Este usa una técnica de optimización para calcular trayectorias que convergen al objetivo planteado, tomando en cuenta todas las restricciones involucradas en el problema. El método planea una trayectoria que minimiza para alcanzar el objetivo, pero restringido a algunos caminos para asegurar la continuidad de la curvatura. Sólo el comando óptimo para el próximo periodo de muestreo es aplicado, y una nueva optimización hasta el objetivo es hecha en cada periodo. De forma que el movimiento pueda adaptarse a los cambios del ambiente, y el sistema pueda planear de nuevo habiendo observado las nuevas condiciones.

Se imponen condiciones a las trayectorias para que el robot pueda seguirlas. Es decir que cualquier camino geométrico no es permitido. Las clases de caminos admitidos son: clotoides, involutas, caminos circulares y líneas rectas. Obsérvese que las clotoides y las involutas a máxima aceleración corresponde a controles extremos. No obstante esta clase de secuencias no correspon-

de a trayectorias de mínimo tiempo (no son secuencias de controles extremos). Por medio de ellas se consigue una curvatura continua en los caminos seguidos por el robot, dando como resultado trayectorias realizables y compatibles con las restricciones de aceleración/deceleración del robot.

En la sección 3.1 la técnica para mapear el dinamismo del entorno fue presentada. Este modelo es usado en nuestra propuesta de planificación para el cálculo de las trayectorias del robot, y está construido sobre la base de los caminos y velocidades (trayectorias) que pudieran provocar colisiones con los objetos. El tiempo a colisión es también una información relevante en el espacio mapeado. El método opera todos los cálculos en el sistema de referencia local del robot, obteniendo así una representación "robocéntrica" del entorno dinámico alrededor del robot.

4.2. El planificador de movimientos robocéntrico

En esta sección se describe el planificador robocéntrico (*RP*), el cual se desarrolla siguiendo estos pasos:

1. El planificador calcula el modelo del ambiente, mapeando las trayectorias de los objetos móviles en el espacio de velocidades del robot (*VS*) (Sección 3.1).
2. Calcula una trayectoria circular desde de la localización actual del robot hasta el siguiente subobjetivo, para seleccionar una trayectoria realizable usada como heurística con el fin de poner la semilla inicial al proceso de optimización.
3. Luego calcula una trayectoria hacia el subobjetivo resolviendo el problema de optimización no lineal, en el cual todas las restricciones son tomadas en cuenta (sección 4.3).
4. Aplica el siguiente comando de movimiento para seguir la trayectoria solución. El proceso se repite cíclicamente para cada periodo de muestreo, hasta que converge al objetivo.

La razón para calcular una trayectoria circular en el segundo paso es que el modelo refleja las velocidades libres que pueden ser elegidas por el robot sin colisión, cuando él sigue esta clase de caminos (línea recta es un caso particular). Pero esta trayectoria es solo una semilla para el siguiente paso, en la cual ella optimizada.

La trayectoria final estará compuesta a partir de una secuencia de caminos como el presentado en la sección 4.1 (por ejemplo clotoide-circular-clotoide-recta, *Cl-C-Cl-R*). Esta clase de caminos han sido usados en trabajos anteriores (como en [KS05], [SF97]) pero en nuestro caso ellas serán parametrizadas en tiempo, ya que es necesario para el proceso de optimización de tiempo.

Dos situaciones pueden aparecer: *i.* que las velocidades altas sean libres de colisión (*S1*); *ii.* o que no lo sean (*S2*). Usando la representación en el *VS* se puede dar una explicación intuitiva para cada caso así como las estrategias de movimiento seleccionadas por cada uno.

4.2.1. Las velocidades altas son libres

La figura 4.1a representa la situación *S1*. Esta corresponde al caso en que las velocidades altas (en la parte alta del *VS*) pueden ser elegidas, es decir que el robot puede pasar antes que

Cuadro 4.1: Algoritmo: Planificador de movimientos robocéntrico

Input: $goal, num_{obj}, obj, V_o$

Output: $traj_{opt}$

```

begin
  repeat
     $[DOVs] = map_{obj}(obj);$ 
     $[k, DOVk] = fusionar_{obj}(num_{obj}, ODVs);$ 
     $[situacin] = detectar_{obj}(k, ODVk);$ 
    if situación = 1
       $[R_{goal}] = goal_{map}(goal);$ 
       $[traj] = generar_{traj}(R_{goal}, k, ODVk, V_o);$ 
    else
       $[R_{new-goal}] = new_{goal}(goal, k, ODVk);$ 
       $[traj] = generar_{traj}(R_{new-goal}, k, ODVk, V_o);$ 
       $[traj_{opt}] = optimizar(traj, V_o);$ 
    end
  until alcanzado(goal)
end

```

el objeto móvil llegue. El robot está en su velocidad actual $V1$. RP elige una de las velocidades altas libres de colisión, $V3$, usando el criterio de proximidad al objetivo mapeado (R_{goal} , en este caso RP ajusta la línea $V2 - V3$) dentro de la zona libre de colisiones. Como el robot tiene restricciones dinámicas, es posible que no pueda alcanzar instantáneamente la nueva trayectoria circular, la línea ($V2 - V3$). Para conseguir esto, el sistema calcula desde la velocidad actual $V1$ una trayectoria clotoide ($V1 - V2$), uniendo los caminos circulares inicial y final, manteniendo la continuidad en la curvatura de la trayectoria total. Observe que esta trayectoria circular tiene dos piezas: la primera parte que es seguida con aceleración (lineal y angular) hasta alcanzar $V3$, y la segunda parte que es seguida a velocidad lineal (máxima) y angular constante, $V3$. Los tiempos asociados a cada tramo son también calculados siendo más usados en el proceso de optimización. En cada tiempo RP verifica que la trayectoria compuesta es libre de colisiones. La figura 4.2a muestra la trayectoria correspondiente en el espacio de trabajo.

4.2.2. Las velocidades altas no son libres

La figura 4.1b representa la situación $S2$. Este es el caso en el cual la velocidad actual conduce a colisión si es mantenida y las velocidades altas son prohibidas. Por lo tanto una solución segura es tomada seleccionando una trayectoria circular, la cual está fuera de la zona de velocidades peligrosas ($R_{new-goal}$). Varias soluciones pueden ser consideradas, por ejemplo, reducir la velocidad a $V2$. Este camino es una involuta, que corresponde a una línea vertical en el VS . Para unir la nueva trayectoria circular, una clotoide $V2 - V3$, luego un tramo circular con aceleración sobre el $R_{new-goal}$ ($V3 - V4$), y finalmente una trayectoria circular con velocidad constante son calculados. Esto permite evitar los obstáculos móviles, reduciendo la velocidad hasta que el objeto pase. Después que el objeto pasa, la situación $S1$ es repetida. La figura 4.2b muestra la

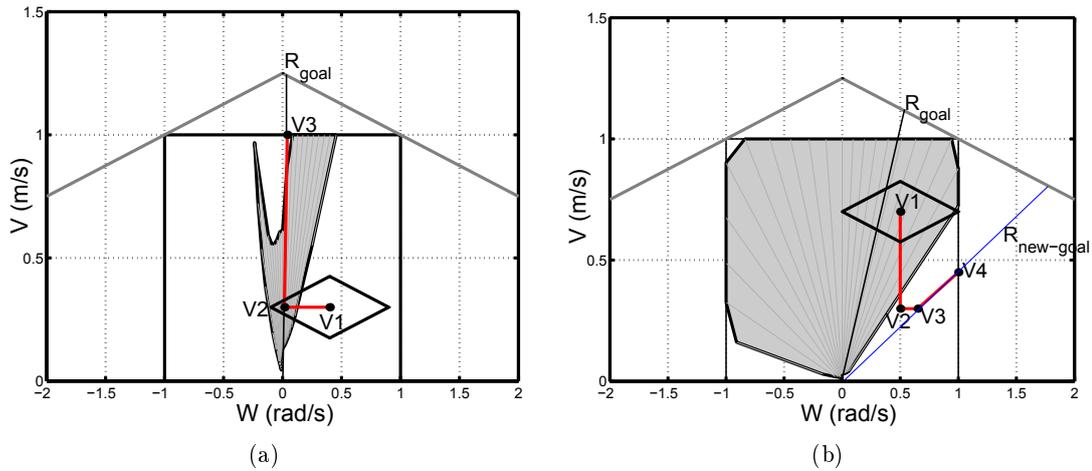


Figura 4.1: (a) trayectorias y V_{ODV} en el VS de la situación $S1$, (b) trayectorias y V_{DOV} en el VS de la situación $S2$. El cuadrado representa las máximas velocidades lineales y angulares que pueden alcanzarse.

trayectoria correspondiente en WS .

La tabla 4.1 muestra el algoritmo principal que ejecuta RP . Las entradas son el objetivo, el número de objetos, los objetos, la velocidad actual del robot, y la salida es una trayectoria realizable por el robot. La función $mapear_{obj}$ mapea los obstáculos en el VS , $fusionar_{obj}$ ordena los objetos en función del tiempo a colisión, $detectar_{obj}$ selecciona las situaciones $S1$ o $S2$, $goal_{map}$ mapea la trayectoria hasta el objetivo dentro del VS , new_{goal} calcula un nuevo $R_{new-goal}$ para $S2$, $generar_{traj}$ genera una trayectoria semilla para el proceso de optimización. En la siguiente sección se presenta de cómo esta trayectoria óptima es generada.

4.2.3. Clases de trayectorias

La trayectoria calculada en la sección anterior es utilizada como una semilla del algoritmo de optimización asegurando una convergencia al objetivo a alcanzar. Se presentan a continuación las clases de trayectorias usadas, la función que será optimizada y las restricciones impuestas en la optimización.

Como se dijo más arriba, se imponen trayectorias compuestas (una trayectoria es el resultado de unir varios tramos de trayectoria como: clotoide, involuta, circular o rectilíneo), que son seguidas por el robot, manteniendo la continuidad de la curvatura. Dependiendo de si el sistema robot se encuentra en $S1$ o $S2$, las trayectorias resultantes son diferentes:

- $S1$: $Cl-C_{ac}-C_{un}-Cl-S$
- $S2$: $ACl-Cl-C_{ac}-C_{un}$

donde Cl significa Clotoide, ACl Involuta, C_{ac} trayectoria circular con aceleración, C_{un} trayectoria circular con velocidad constante (máxima velocidad), and S trayectoria rectilínea.

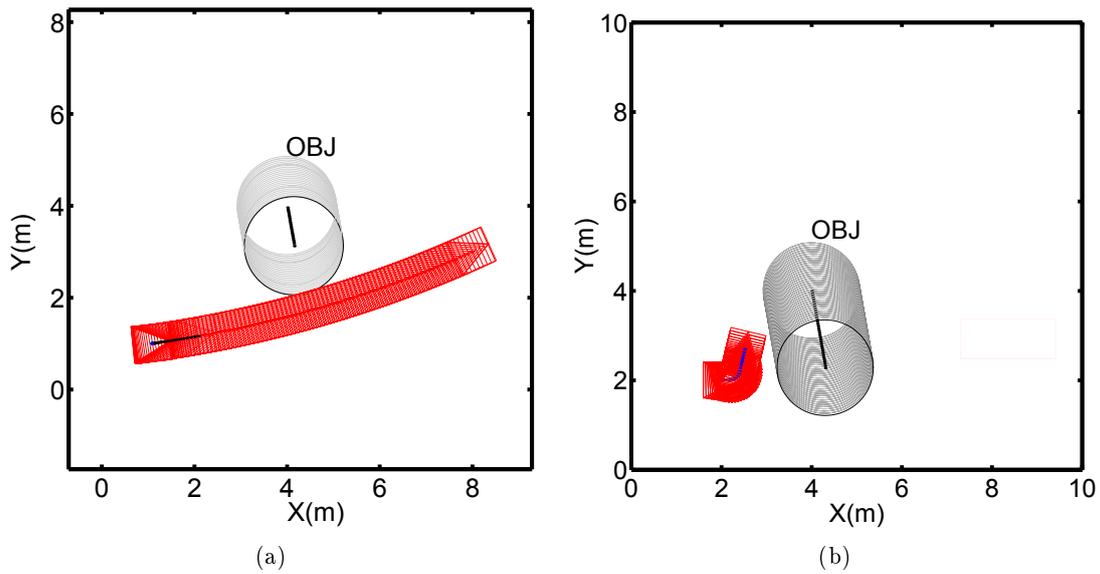


Figura 4.2: (a) corresponde a la trayectoria en WS de $S1$; el robot acelera para pasar antes que el objeto, (b) corresponde a la trayectoria en WS de $S2$; el robot desacelera para esperar que el objeto pase.

4.3. Optimización de las trayectorias

El problema de optimización puede ser formulado como sigue:

$$t^* = \operatorname{argmin}_{(t)} F(\mathbf{x}(t))$$

donde $\mathbf{t} = [\sum_{j=1}^k t_j]$ es un vector de tiempos cuyas componentes se corresponden con el tiempo de cada tramo de la trayectoria completa, y $F(\mathbf{x}(t))$ es una función que representa la secuencia de subtrayectorias parametrizadas en el tiempo (ver apéndice C). Esta función depende de varios parámetros como: máximas aceleraciones, número de subtrayectorias, la localización y la velocidad iniciales, y los parámetros geométricos del robot.

Las restricciones del problema de optimización son:

- restricciones no lineales de igualdad:

$$x_k - x_{goal} = 0 \quad (1)$$

$$y_k - y_{goal} = 0 \quad (2)$$

$$\sum_{j=1}^k L_k(w_o) = 0 \quad (3)$$

$$\sum_{j=1}^k L_k(v_o) = 0 \quad (4)$$

donde k es el número de tramos, (1) y (2) conducen la búsqueda hacia el objetivo, (3) y (4) consideran la continuidad en las curvas.

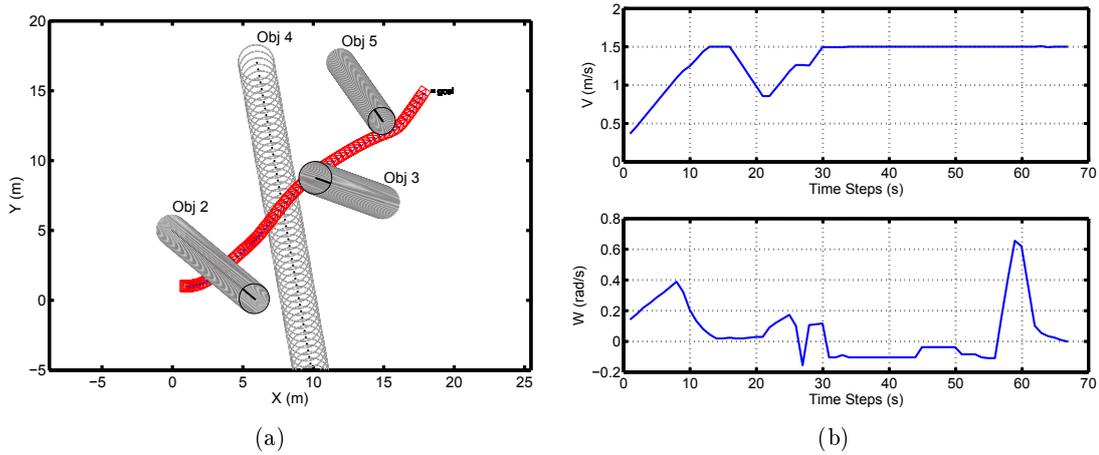


Figura 4.3: (a) trayectoria resultante cuando el plan es calculado para cada periodo de tiempo, (b) perfiles de velocidad lineal y angular del primer experimento.

- restricciones no lineales de desigualdad:

$$\sum_{j=1}^k f(w_j, v_j) \geq 0 \quad (5)$$

$$V_k \geq 0 \quad (6)$$

$$\sum_{j=1}^k t_j \geq 0 \quad (7)$$

$$\sum_{q=1}^{num} (x - f_{xq})^2 + (y - f_{yq})^2 - r_q^2 \geq 0 \quad (8)$$

donde num es el número de objetos móviles, (5) considera las restricciones en las velocidades lineal y angular en el VS (en nuestro caso para un robot tipo "*differential-drive*", ver el apéndice A), (6) impone velocidades lineales positivas, (7) impone soluciones para tiempos positivos, y (8) refleja las restricciones que vienen de los objetos móviles en el CS modelados como círculos. f_{xq} y f_{yq} son funciones de tiempo, y r_q es el radio del objeto aumentado con el radio del robot.

4.4. Resultados experimentales

El objetivo de los experimentos es mostrar como el planificador trabaja en diferentes condiciones. El tiene que adaptar la decisión a la dinámica de los objetos y del robot para evitar las colisiones. En el primer experimento (ver la figura 4.3a), el planificador calcula la trayectoria óptima en cada periodo de tiempo, produciendo una trayectoria suave (continua) hasta el objetivo. Observe como el planificador toma la decisión de esperar que el objeto con la etiqueta $O4$ pase y acelera incrementando su velocidad para pasar por delante de los objetos $O2$, $O3$, $O5$. La figura 4.3b muestra los perfiles de velocidad lineal y angular del Robot. Observe como el robot reduce la velocidad lineal mientras el objeto $O4$ se mueve hacia él, permitiendo así que el objeto pase. Después, el robot se mueve a máxima velocidad lineal para pasar antes que los objetos $O2$, $O3$ y $O5$ lleguen a colisionar con él. Note que para hacerlo el robot maniobra evitando las colisiones con los objetos.

En el segundo experimento la dirección y velocidad de los objetos es ligeramente cambiada, de forma que el plan anterior no es válido y el nuevo difiere del experimento anterior. Ahora el robot

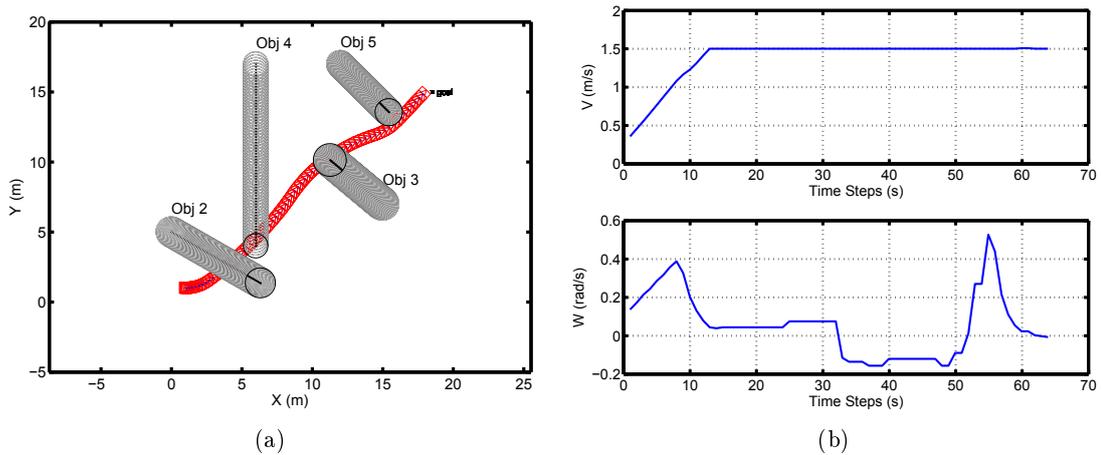


Figura 4.4: (a) Trayectoria planeada en otro escenario, con velocidades diferentes (magnitud y dirección) en los objetos, (b) Perfiles de velocidad lineal y angular en el segundo experimento.

pasa antes que todos los objetos colisionen con él, como puede verse en la figura 4.4a. El robot adapta el plan a la nueva situación. En la figura 4.4b los perfiles de velocidad son presentados. Puede apreciarse como durante el periodo inicial de aceleración, los tramos de trayectoria circular y clotoide (ambas con velocidad lineal constante) son alternados, debido a la planificación realizada en cada periodo de tiempo. Se observa como el robot se dirige al objetivo a máxima velocidad lineal, rotando cuando necesita evitar los objetos que le vienen.

4.5. Conclusiones

Un planificador robocéntrico para entornos dinámicos ha sido presentado. El planificador minimiza un criterio de tiempo al objetivo a alcanzar, restringido a: *i.* algunas clases de trayectorias con el fin de asegurar la suavidad de la curvatura (curvatura continua), *ii.* a las restricciones cinemáticas y de aceleración del robot y *iii.* a las velocidades que evitan colisión con los objetos.

El método calcula las trayectorias desde el estado actual del robot (localización y velocidades) hasta el objetivo a alcanzar, las cuales son evaluadas en cada periodo de tiempo adaptándose a los cambios en el entorno y a las nuevas localizaciones y velocidades del robot. Un modelo representando a los objetos móviles alrededor del robot en el espacio de velocidades permite seleccionar velocidades libres de colisión para inicializar el proceso de optimización local, calculando el próximo comando de velocidad del robot.

Como trabajo futuro, puede ser planteado un planificador global que explote la información de velocidad y tiempo incluida en el modelo, para ser usada juntamente con el planificador aquí presentado. También, otras secuencias de trayectorias pueden ser analizadas para observar comportamientos y calcular movimientos de tiempo cercanos al óptimo.

Capítulo 5

Conclusiones y trabajo futuro

El objetivo principal de este proyecto se centraba en desarrollo de un método de planificación de movimientos en ambientes dinámicos, evitando los obstáculos dinámicos y estáticos del entorno de navegación del robot mientras este se dirige hacia el objetivo. Para ello se ha realizado una extensa revisión bibliográfica acerca de los métodos de planificación de movimientos relacionados con ambientes dinámicos.

En este trabajo se ha propuesto un método para modelar el dinamismo del ambiente junto con las restricciones del robot, construido sobre el espacio de velocidades dinámico (*DVS*) del robot. El modelo conseguido permite aplicar estrategias de navegación para calcular los comandos de velocidad del robot directamente en su espacio de velocidades, mientras los obstáculos del entorno son evitados manteniendo así la integridad del robot. Este objetivo es alcanzado aplicando estrategias heurísticas para controlar la forma de la solución. El siguiente paso es analizar en profundidad diversas estrategias con otros criterios en la toma de decisiones logrando diversos comportamientos sobre el robot.

Se ha presentado un planificador robocéntrico para entornos dinámicos, usando un proceso de optimización no lineal restringido que minimiza el criterio de tiempo para alcanzar el objetivo. El planificador propuesto trabaja restringiendo las trayectorias a algunas clases de estas, asegurando la suavidad de la trayectoria seguida por el robot.

De nuevo el modelo usado para representar la dinamicidad del entorno del robot es utilizado para generar la trayectoria semilla inicial requerida para iniciar el proceso de optimización no lineal. El método así puede calcular las trayectorias desde su estado actual hasta el objetivo adaptándose en cada periodo de tiempo a los cambios del entorno y a los nuevos estados del robot.

Como trabajo futuro, se plantea el diseño de un planificador global que explote la información de velocidad y tiempo incluida en el modelo, para ser usada juntamente con el planificador aquí presentado. También, otras secuencias de trayectorias pueden ser analizadas para observar comportamientos y calcular movimientos de tiempo cercanos al óptimo.

Planteamos también trabajar en el diseño de un planificador de movimientos para entornos dinámicos basado en situaciones del espacio de velocidades del robot. En este trabajo solo se han presentado dos tipos de situaciones que se traducen en dos tipos de comportamientos bien definidos en el robot. Hemos observado e identificado más situaciones en el espacio de velocidades que

pueden traducirse en nuevos comportamientos, generando nuevas maniobras que el robot podría aplicar. Con relación al trabajo de optimización aplicado a la planificación de movimientos el trabajo presentado nos abre a propuestas de planificación óptima de trayectorias en el *VS* usando funciones de coste y dirigiendo la búsqueda mediante las situaciones encontradas en el *DVS*.

Apéndice A

Características del robot

Este robot, que es del tipo "differential-drive", está equipado con un sensor laser de distancia 2-D (laser rangefinder Sick LMS-200), y cuenta con un ordenador a bordo que lleva montado un procesador Intel Centrino duo a 1.6 GHz. El tiempo de cómputo de la estrategia de navegación fue de alrededor 200 ms, tiempo para el cual una nueva medida del laser está disponible. El campo de vista del sensor laser es de 180 grados, con una resolución angular de 0,5 grados y un rango máximo de 8 m. La velocidad traslacional fue puesta a 0.3 m./sec. y una velocidad rotacional a 0.4 rad./sec. Hay dos razones para esto: primero, la seguridad en un escenario real; segundo, estas velocidades pueden ser simultáneamente alcanzadas por esta clase de robot, ya que las velocidades máximas reales son mas altas que los valores ajustados, ajustando la ventana cuadrada de velocidades en el método de navegación (ver detalles en el Apéndice B).

El método usado para seguir los objetos móviles fue basado en un filtro de Kalman extendido (EKF), en el cual el vector de estado incluye la localización y las velocidades de los objetos seguidos. Los mapas de los objetos móviles y estáticos fueron obtenidos usando un método desarrollado en [MMM08] para modela un ambiente dinámico. En la figuraA.1 se muestra una imagen del robot.



Figura A.1: Límites del espacio de velocidades real

Apéndice B

Limites del espacio de velocidades

En simulación todas las velocidades pueden ser alcanzadas, se consideraban viables todos los pares de velocidades incluidos entre los rangos máximos admisibles predefinidos. Sin embargo al llegar a las pruebas reales las restricciones aumentan. El robot Pioneer 3-AT consta de cuatro ruedas que giran independientemente, permitiéndole realizar giros sobre sí mismo.

El fabricante proporciona unos valores de velocidad lineal y angular máximos que el robot puede alcanzar. Sin embargo no todos los pares de velocidad incluidos dentro del rectángulo que definen las velocidades máximas son alcanzables ni estables.

No considerar estas limitaciones conlleva a errores en la actuación. De manera que si el algoritmo devuelve un par de velocidades que no es posible alcanzar, se producen situaciones no deseadas. Por ejemplo en algunas pruebas realizadas antes de observar el fenómeno resultaba que el robot seguía en línea prácticamente recta hacia una pared, cuando la consigna del programa le mandaba girar, pero al ser una velocidad no alcanzable, el controlador interno del robot le daba prioridad a la velocidad lineal, despreciando la angular. Ante este hecho había dos alternativas. La primera sería trabajar en un rango de velocidades inferior, para el que el espacio de velocidades alcanzables estuviera perfectamente acotado en un rectángulo por una velocidad lineal y una angular. La segunda alternativa correspondía a definir adecuadamente los límites de velocidades alcanzables e introducirlos en el programa para emplear el máximo rango de velocidades posible.

Para definir los límites de velocidad aceptables se realizó un ensayo con el robot en el que se comprobaron las duplas de velocidad lineal y angular que se mantenían estables y con error de posición cero ante la consigna. En la figura B.1 se muestran los límites medidos.

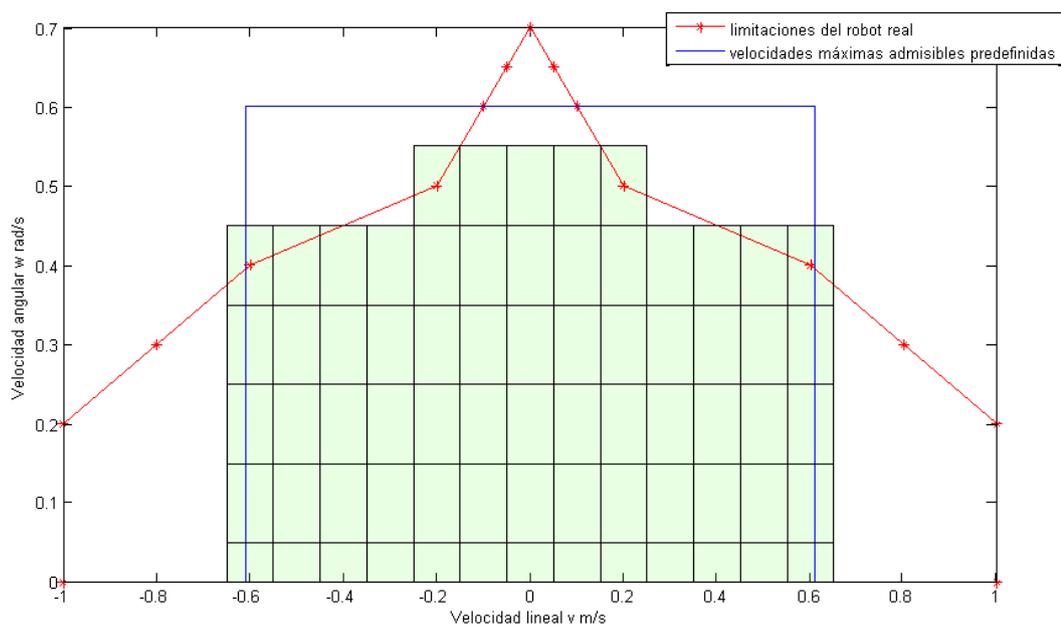


Figura B.1: Límites del espacio de velocidades real

Apéndice C

Ecuaciones de las curvas

En este apéndice se presentan las ecuaciones de las curvas para el caso general a partir de una configuración inicial $C_0 = (x_0, y_0, \theta_0)$ y las velocidades iniciales v_0 y w_0 .

C.1. Ecuaciones de movimiento del robot

Las ecuaciones para un robot del tipo diferencial ("*differential-drive*", como es nuestro caso) son:

$$x(t) = x_0 + \int_0^t v(u) \cos \theta(u) du \quad (1)$$

$$y(t) = y_0 + \int_0^t v(u) \sin \theta(u) du \quad (2)$$

$$\theta(t) = \theta_0 + \int_0^t w(u) du \quad (3)$$

C.2. Ecuaciones de una Clotoide

Se considera el caso en el cual $a_r = -a_l$. a_r y a_l representan las aceleraciones de las ruedas. Ahora, $a_r = \pm a$, mientras $a_r = \mp a$. En este caso $\dot{v}(t) = \frac{1}{2}(a_r + a_l) = 0$. Entonces $v(t) = v_0$. Por integración $v_r(t) = \pm at + v_{r_0}$, $v_l(t) = \mp at + v_{l_0}$ y $w(t) = \dot{\theta}(t) = \pm \frac{2a}{d}t + w_0$. Las expresiones para $v(t)$ y $w(t)$ son reemplazadas en (1), (2) and (3). Las coordenadas del robot son dadas por las integrales de Fresnel (SF y CF representan el Seno y Coseno de Fresnel), respectivamente (véase [JJ92] por ejemplo):

$$\begin{aligned} x(t) = & x_0 - \frac{1}{2}v_0\sqrt{2}\sqrt{\pi}\{-CF(*4)(*1) \cos \theta_0 \\ & -CF(*4)(*3) \sin \theta_0 - SF(*4)(*3) \cos \theta_0 \\ & +SF(*4)(*1) \sin \theta_0 + CF(*2)(*1) \sin \theta_0 \\ & +CF(*2)(*3) \sin \theta_0 + SF(*2)(*3) \cos \theta_0 \\ & -SF(*2)(*3) \sin \theta_0 + SF(*2)(*3) \cos \theta_0 \\ & -SF(*2)(*1) \sin \theta_0\} / \sqrt{\frac{a}{d}} \end{aligned}$$

$$y(t) = y_0 - \frac{1}{2}v_0\sqrt{2}\sqrt{\pi}\{SF(*4)(*1) \cos \theta_0$$

$$\begin{aligned}
& +SF(*4)(*3) \sin \theta_0 - CF(*4)(*3) \cos \theta_0 \\
& +CF(*4)(*1) \sin \theta_0 - SF(*2)(*1) \cos \theta_0 \\
& -SF(*2)(*3) \sin \theta_0 + CF(*2)(*3) \cos \theta_0 \\
& -SF(*2)(*3) \sin \theta_0 \} / \sqrt{\frac{a}{d}}
\end{aligned}$$

$$\theta(t) = \theta_0 + \left(\frac{at^2}{d} + w_0t\right)$$

donde (*1) = $\cos\left(\frac{1}{4}\frac{dw_0^2}{a}\right)$, (*2) = $\frac{1}{2}\frac{\sqrt{2}w_0}{\sqrt{\pi}\sqrt{\frac{a}{d}}}$, (*3) = $\sin\left(\frac{1}{4}\frac{w_0^2d}{a}\right)$, y (*4) = $\frac{1}{2}\frac{\sqrt{2}(2at+w_0d)}{\sqrt{\pi}d\sqrt{(a/d)}}$

C.3. Ecuaciones de una Involuta

Se considera el caso en donde $a_r = a_l$. Entonces $\dot{w}(t) = \frac{1}{d}(a_r - a_l) = 0$ y $w(t)$ es una constante e igual a w_0 . La aceleración $a(t)$ es igual a $sgn(a_r)at$. Así $v(t) = sgn(a_r)at + v_0$.

$$\begin{aligned}
x(t) &= x_0 + \{a \cos(w_0t + \theta_0) + a \sin(w_0t + \theta_0)w_0t \\
&\quad + v_0 \sin(w_0t + \theta_0)w_0 - a \cos \theta_0 \\
&\quad - v_0 \sin(\theta_0)w_0\} / w_0^2. \\
y(t) &= y_0 - \{-a \sin(w_0t + \theta_0) + a \cos(w_0t + \theta_0)w_0t \\
&\quad + \cos(w_0t + \theta_0)w_0v_0 + a \sin \theta_0 \\
&\quad - \cos(\theta_0)v_0w_0\} / w_0^2. \\
\theta(t) &= \theta_0 + w_0t;
\end{aligned}$$

C.4. Ecuaciones de un arco de círculo con aceleraciones lineal y angular

La velocidad $v(t)$ depende de la velocidad lineal inicial v_0 y de la aceleración lineal \dot{v} . La orientación $\theta(t)$ es una función de la orientación inicial θ_0 , la velocidad angular w_0 y la aceleración angular \dot{w} . Sustituyendo $v(t)$ y $\theta(t)$ como una función de la configuración cinemática y dinámica inicial v_0, θ_0, w_0 y de las aceleraciones \dot{v} y \dot{w} se tienen las siguientes expresiones:

$$\begin{aligned}
x(t) &= x_0 + \{A[(*)4)(BD + CE) + (*)3)(CD - BE)] \\
&\quad + F[E(GH - IJ) + D(IH + GJ)] \\
&\quad + K[(*)3)(BE - CD) - (*)4)(BD + CE)] \\
&\quad + A[-(*)2)(BD + CE) + (*)1)(CD + BE)] - L \\
&\quad + K[(*)2)(BD + CE) + (*)1)(CD + BE)]\} / M
\end{aligned}$$

$$\begin{aligned}
y(t) = & y_0 + \{ A[(\ast 4)(BD + CE) + (\ast 3)(BE - CD)] \\
& + F[E(IJ - GH) + D(IH + GJ)] \\
& + K[(\ast 3)(CD - BE) - (\ast 4)(BD + CE)] \\
& + A[-(\ast 2)(BD + CE) + (\ast 1)(CD - BE)] - L \\
& + K[(\ast 2)(BD + CE) + (\ast 1)(BE - CD)] \} / M \\
\theta(t) = & \theta_0 + w_0 t + \left(\frac{\dot{w} t^2}{d} \right)
\end{aligned}$$

donde ($\ast 1$), ($\ast 2$), ($\ast 3$), ($\ast 4$) son dadas por las integrales de Fresnel como: ($\ast 1$) = $FS\left(\frac{ta_r+w_0}{\sqrt{\pi}\sqrt{a_r}}\right)$, ($\ast 2$) = $FC\left(\frac{ta_r+w_0}{\sqrt{\pi}\sqrt{a_r}}\right)$, ($\ast 3$) = $FS\left(\frac{w_0}{\sqrt{\pi}\sqrt{a_r}}\right)$, ($\ast 4$) = $FC\left(\frac{w_0}{\sqrt{\pi}\sqrt{a_r}}\right)$. $A = v_0\pi\dot{w}$, $B = \cos\left(\frac{w_0^2}{2\dot{w}}\right)$, $C = \sin\left(\frac{w_0^2}{2\dot{w}}\right)$, $D = \cos\theta_0$, $E = \sin\theta_0$, $F = \dot{w}\sqrt{\dot{w}^3}$, $G = \cos(w_0 t)$, $H = \cos\left(\frac{\dot{w}t^2}{2}\right)$, $I = \sin(w_0 t)$, $J = \sin\left(\frac{\dot{w}t^2}{2}\right)$, $K = \dot{w}w_0\pi$, $L = \dot{w}\sin\theta_0\sqrt{\dot{w}^3}$, $M = \sqrt{\dot{w}^5}$.

Bibliografía

- [BATM93] P. Bessiere, J.-M. Ahuactzin, E.-G. Talbi, and E. Mazer. The ariadne's clew algorithm: global planning with local methods. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1373–1380, 1993.
- [BGF08] J.L. Blanco, J. González, and J.A. Fernández. Extending obstacle avoidance methods through multiple parameter-space transformations. *Autonomous Robots*, 24(1), 2008.
- [BH95] M. Barbehenn and S. Hutchinson. Efficient search and hierarchical motion planning by dynamically maintaining singlsource shortest path trees. *IEEE Transactions on Robotics and Automation*, 11(2):198–214, 1995.
- [BK89] J. Borenstein and Y. KorenD. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5), 1989.
- [BK91] J. Borenstein and Y. Koren. The vector field Histogram-fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7:278–288, 1991.
- [BK99] O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings of International Conference on Robotics and Automation*, Detroit, Usa, 1999.
- [BK01] O. Brock and L.E. Kavraki. Decomposition-based motion planning: a framework for real-time motion planning in high-dimensional configuration spaces. In *IEEE International Conference on Robotics and Automation*, pages 1469–1474, 2001.
- [BL91] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *Int. Journal of Robotics Research*, 10(6), 1991.
- [BV03] J. Bruce and M. Veloso. Real-time randomized path planning for robot navigation. *RoboCup 2002: Robot Soccer World Cup VI*, pages 288–295, 2003.
- [Can88] J. F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA, 1988.
- [CB00] H. Choset and J. Burdick. Sensor-based exploration: The hierarchical generalized voronoi graph. *Discrete and Computational Geometry*, 19(2), 2000.
- [CD88] J. Canny and B. R. Donald. Simplified voronoi diagrams. *Discrete and Computational Geometry*, 3(1), 1988.

-
- [Cha87] B. Chazelle. *Approximation and decomposition of shapes in Algorithmic and Geometric Aspects of Robotics*. J. T. Schwartz and C. K. Yap, editors, Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.
- [DF08] V. Delsart and T. Fraichard. Navigating dynamic environments using trajectory deformation. In *IEEE International Conference on Intelligent Robots and Systems*, Nice, France, 2008.
- [Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 1959.
- [Elf83] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(5), 1983.
- [Elf89] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6), 1989.
- [ELP87] M. Erdmann and T. Lozano-Pérez. On multiple moving objects. *Algorithmica*, 2(4), 1987.
- [FBT97] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics and Automation Magazine*, 4(1), 1997.
- [Fra98] T. Fraichard. Trajectory planning in a dynamic workspace: a 'state-time space' approach. *Algorithmica*, 13(1), 1998.
- [Fra07] T. Fraichard. A short paper about motion safety. In *IEEE International Conference on Robotics and Automation*, pages 1140–1145, 2007.
- [FS97] P. Fiorini and Z. Shiller. Time optimal motion planning in dynamic environments. *International Journal of Applied Mathematics and Computer Science*, 7(4):101–126, 1997.
- [FS98] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *Int. Journal of Robotics Research*, 17(7), 1998.
- [FS06] D. Ferguson and A. Stentz. Anytime rrts. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5369–5375, 2006.
- [FSLC95] S. Fleury, P. Soueres, J.-P. Laumond, and R. Chatila. Primitive for smoothing mobile robot trajectories. *IEEE Transactions on Robotics and Automation*, 11(3), 1995.
- [GC02] S.S. Ge1 and Y.J. Cui. Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3), 2002.
- [HKLR02] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. *Int. Journal of Robotics Research*, 21(3):233–256, 2002.
- [HNR68] E.P. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Numerische Mathematik*, SSC-4(2), 1968.
-

- [Hua08] L Huang. Velocity planning for a mobile robot to track a moving target - a potential field approach. *Robotics and Autonomous Systems*, 57(1), 2008.
- [JJ92] James and James. *Mathematics Dictionary*. Van Nost. Reinhold, 1992.
- [JLR91] P. Jacobs, J.-P. Laumond, and A. Rege. Non-holonomic motion planning for hilare-like mobile robots. In *Proceedings of the Intenational Symposium on Intelligent Robots*, pages 338–347, 1991.
- [KF07] H. Kurniawati and Th. Fraichard. From path to trajectory deformation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 159–164, 2007.
- [Kha86] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1), 1986.
- [KJCL97] M. Khatib, H. Jaouni, R. Chatila, and J.-P. Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2920–2925, 1997.
- [KK92] J.-O. Kim and P.K. Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3), 1992.
- [KL02] S. Koenig and M. Likhachev. Improved fast replanning for robot navigation in unknown terrain. In *IEEE International Conference on Robotics and Automation*, pages 968–975, 2002.
- [KS98] N.Y. Ko and R. Simmons. The lane-curvature method for local obstacle avoidance. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3375–3382, Victoria, Canada, 1998.
- [KS05] M.B. Kobilarov and G.S. Sukhatme. Near-optimal constrained trajectory planning on outdoor terrain. In *Int. Conference on Robotics and Automation*, pages 1833–1840, 2005.
- [KSLO96] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 1996.
- [LBL04] F. Lamiroux, D. Bonnafous, and O. Lefebvre. Reactive path deformation for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 20(6):967–977, 2004.
- [LFG⁺05] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun. Anytime dynamic A*: An anytime replanning algorithm. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2005.
- [LK01] S.M. LaValle and J.J. Kuffner. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and Computational Robotics New Directions*, pages 293–308, 2001.
- [LP83] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Transactions on Computing*, C-32(2), 1983.

- [LPW79] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10), 1979.
- [LSSL02] F. Large, S. Sekhavat, Z. Shiller, and C. Laugier. Towards real-time global motion planning in a dynamic environment using the nlvo concept. In *Int. Conf. on Intelligent Robots and Systems*, 2002.
- [MM00] J. Minguez and L. Montano. Nearness Diagram Navigation (ND): A New Real-Time Collision Avoidance Approach. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2094–2100, Takamatsu, Japan, 2000.
- [MM06] J. Minguez and L. Montano. Abstracting vehicle shape and kinematic constraints from obstacle avoidance methods. *Autonomous Robots*, 20(1), 2006.
- [MMK02] J. Minguez, L. Montano, and O. Khatib. The Ego-Dynamic Space (EDS): Dynamics of the vehicle. In *Proceedings of the Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002.
- [MMM08] L. Montesano, J. Minguez, and L. Montano. Modeling dynamic scenarios for local sensor-based motion planning. *Autonomous Robots*, 25(3):231–251, 2008.
- [MMSA01] J. Minguez, L. Montano, Th. Simeon, and R. Alami. Global Nearness Diagram Navigation (GND). In *IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001.
- [MMSV02] J. Minguez, L. Montano, and J. Santos-Victor. Reactive navigation for nonholonomic robots using the ego kinematic space. In *IEEE International Conference on Robotics and Automation*, Washington, US, 2002.
- [Mor88] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2), 1988.
- [OM05] E. Owen and L. Montano. Motion planning in dynamic environments using the velocity space. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 997–1002, 2005.
- [OM06] E. Owen and L. Montano. A robocentric motion planner for dynamic environments usin the velocity space. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4368–4374, 2006.
- [PF05] S. Petti and Th. Fraichard. Partial motion planning framework for reactive planning within dynamic environments. In *Proc. of the IFAC/AAAI Int. Conf. on Informatics in Control, Automation and Robotics*, Barcelona (SP), 2005.
- [QK93] S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. In *IEEE International Conference on Robotics and Automation*, pages 802–807, 1993.
- [Rei79] J. H. Reif. Complexity of the mover’s problem and generalizations. In *Proceedings IEEE Symposium on Foundations of Computer Science*, pages 421–427, 1979.
- [RF97] M. Renaud and J-Y. Fourquet. Minimum time motion of a mobile robot with two independent, acceleration-driven wheels. In *IEEE Int. Conference on Robotics and Automation*, pages 2608–2612, Albuquerque, New Mexico, 1997.

- [RK92] E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5), 1992.
- [SF97] A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 997–1003, 1997.
- [Sim96] R. Simmons. The Curvature-Velocity Method for Local Obstacle Avoidance. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 3375–3382, Minneapolis, USA, 1996.
- [SLSL01] Z. Shiller, F. Large, S. Sekhavat, and C. Laugier. Motion planning in dynamic environment: Obstacles moving along arbitrary trajectories. In *Int. Conference on Robotics and Automation*, 2001.
- [SP07] M. Sender and I. Petrovic. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In *IEEE Int. Conf. on Robotics and Automation*, 2007.
- [SS83a] J. T. Schwartz and M. Sharir. On the piano movers' problem: I. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics*, 36, 1983.
- [SS83b] J. T. Schwartz and M. Sharir. On the piano movers' problem: II. general techniques for computing topological properties of algebraic manifolds. *Advances in Applied Mathematics*, 12, 1983.
- [Ste93] A. Stentz. Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation*, 10:89–100, 1993.
- [Ste95] A. Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1652–1659, 1995.
- [Ste98] LaValle Steven. Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Science Computer Dpt., Iowa State University, October 1998.
- [UB98] I. Ulrich and J. Borenstein. VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. In *In Proceedings IEEE International Conference on Robotics and Automation*, pages 1572–1577, 1998.
- [UB00] I. Ulric and J. Borenstein. VFH*: Local Obstacle Avoidance with Look-Ahead Verification. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2505–2511, 2000.
- [vdBO05] J.P. van den Berg and M.H. Overmars. Roadmap-based motion planning in dynamic environments. *IEEE Transactions on Robotics*, 21(5):885–897, 2005.
- [VKA05] N. Vandapel, J. Kuffner, and O. Amidi. Planning 3-d path networks in unstructured environments. In *IEEE International Conference on Robotics and Automation*, pages 4624–4629, 2005.

-
- [WvdBM09] D. Wilkie, J. van den Berg, and D. Manocha. Generalized velocity obstacles. In *Proceedings IEEE International Conference on Intelligent Robots and Systems*, pages 5573–5578, 2009.