



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Proyecto Fin de Carrera
Ingeniería Industrial
Curso 2011-2012

Navegación de robots autónomos en entornos dinámicos

Cristina Menéndez Romero

Director: Luis Montano Gella

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Junio 2012

NAVEGACIÓN DE ROBOTS AUTÓNOMOS EN ENTORNOS DINÁMICOS

Resumen

Uno de los principales objetivos de la robótica es desarrollar métodos robustos de planificación autónoma de trayectorias. En la mayoría de las situaciones pueden aparecer otros obstáculos móviles que dificultan la navegación y con los que es necesario tratar en tiempo real para poder asegurar en todo momento la integridad del robot mientras realiza sus tareas.

Este proyecto se enmarca en los proyectos de robótica desarrollados por el Grupo de Robótica, Percepción y Tiempo Real de la Universidad de Zaragoza.

El objetivo general es desarrollar una técnica de navegación para un vehículo autónomo en un entorno dinámico y no estructurado. El vehículo debe ser capaz de maniobrar con rapidez y seguridad. Todo esto se logra planificando trayectorias en el espacio dinámico de velocidades y tiempo (DVTS) del robot.

En trabajos anteriores, se desarrollaron las técnicas para representar la información de los obstáculos tanto estáticos como dinámicos en el espacio de velocidades del robot, considerando los pares de velocidades para los cuales podría aparecer colisión a partir de un tiempo dado. Además, se desarrolló una técnica de planificación mediante un árbol de decisiones, que trabajaba con la información proyectada en el plano de velocidades (v,w) . Se trata de una técnica heurística basada en el reconocimiento de situaciones con las que se encuentra el robot y a las que asocia un conjunto de acciones.

Este proyecto da un paso más, desarrollando una técnica que minimiza el tiempo de alcance de los subobjetivos de la trayectoria a la vez que genera movimientos seguros. Para ello emplea además de la información del espacio de velocidades, la información de tiempo para obtener trayectorias más ajustadas y eficientes. La búsqueda de la trayectoria óptima está dirigida por una función de coste, en la que se incluyen las restricciones cinemáticas y dinámicas del robot (no todos los movimientos son posibles, ni todas las velocidades alcanzables de forma instantánea), criterios de seguridad del robot, criterios de selección de estrategias de evitación y criterios de distancia hasta el objetivo.

La obtención de la información del entorno se realiza mediante un escáner láser de 180 ° de rango y 0.5° de precisión. Este dispositivo limita el tiempo de refresco del programa a 250 milisegundos. Para el tratamiento de la información obtenida a través del láser y de la odometría se emplean algoritmos desarrollados previamente en el laboratorio.

El proyecto se desarrolla en el lenguaje de programación C++. La interfaz proporcionada por la plataforma de software para robótica Player se emplea para interactuar con los robots en los diferentes experimentos realizados. Estos experimentos constan tanto de simulaciones llevadas a cabo en la plataforma Stage como de experimentos reales empleando los robots Pioneer del laboratorio.

Índice general

| | |
|---|-----------|
| 1. Introducción | 1 |
| 1.1. Motivación | 1 |
| 1.2. Estado del arte | 2 |
| 1.3. Estructura de la memoria | 3 |
| 2. El espacio de control y la búsqueda en el mismo | 5 |
| 2.1. El espacio de velocidades-tiempo | 5 |
| 2.2. Cálculo de trayectorias en el espacio DVTS | 8 |
| 2.3. Discretización del DVTS | 10 |
| 2.3.1. Resolución de la malla: tamaño de celda | 10 |
| 2.3.2. Horizonte Temporal | 11 |
| 2.3.3. Obtención de la superficie completa | 11 |
| 3. Búsqueda de trayectorias en DVTS | 13 |
| 3.1. Introducción al Algoritmo A* | 13 |
| 3.2. El operador de expansión | 15 |
| 3.3. Costes y seguridad | 17 |
| 4. Estrategias de navegación | 23 |
| 4.1. Heurísticas para la selección de subobjetivos | 24 |
| 4.1.1. Integración de estrategias | 25 |
| 4.2. Maniobras de adelantamiento | 27 |
| 5. Módulos del programa | 31 |
| 5.1. Estructura del programa | 31 |
| 6. Simulaciones | 33 |
| 6.1. Cruce | 34 |
| 6.2. Escenario sin obstáculos estáticos | 35 |
| 6.3. Escenario de interiores con obstáculos estáticos y móviles | 36 |
| 6.4. Adelantamiento | 38 |
| 7. Experimentos | 41 |
| 7.1. Hall | 42 |
| 7.2. Adelantamiento | 44 |

| | |
|---|-----------|
| 8. Conclusiones y trabajo futuro | 47 |
| 8.1. Conclusiones | 47 |
| 8.2. Limitaciones y Trabajo Futuro | 48 |
| A. Selección de estrategias | 51 |
| A.1. El árbol de Decisión | 51 |
| A.2. Adelantamiento | 52 |
| A.2.1. Clotoides en el cambio de carril | 52 |
| A.2.2. Maniobra de adelantamiento | 53 |
| B. Filtros y estimadores | 55 |
| B.1. Filtro de Kalman extendido | 55 |
| B.2. Estimadores | 56 |
| B.2.1. Clasificación de puntos | 56 |
| B.2.2. Autolocalización | 57 |
| B.2.3. Obstáculos estáticos | 57 |
| B.2.4. Obstáculos móviles | 58 |
| C. Simulaciones | 61 |
| C.1. Cruce | 61 |
| C.2. Caos | 64 |
| C.3. Interiores | 67 |
| C.4. Adelantamiento | 72 |
| D. Experimentos | 77 |
| D.1. Hall | 77 |
| D.2. Adelantamiento | 81 |
| E. Consideraciones | 87 |
| E.1. Límites del espacio de velocidades | 87 |
| E.2. Pruebas de coeficientes | 89 |
| Índice de figuras | 93 |
| Bibliografía | 95 |

Capítulo 1

Introducción

1.1. Motivación

El objetivo principal de este proyecto es dotar al robot de un planificador local capaz de integrar toda la información sensorial disponible y con ella seleccionar la trayectoria óptima para alcanzar el objetivo. Se supone la existencia de un planificador global que señala los objetivos intermedios o puntos vía. El planificador local desarrollado es el encargado de resolver la trayectoria óptima entre los puntos vía.

Concretamente, las etapas necesarias para alcanzar el objetivo de este proyecto son las siguientes:

- Establecer una discretización del espacio tridimensional de velocidades-tiempo que permita calcular la trayectoria y tratar la información de forma rápida y eficiente.
- Desarrollar un método de planificación de trayectorias basado en un algoritmo de búsqueda A^* . Dicho algoritmo incluye unos términos de ponderación que dirigen la búsqueda hacia trayectorias seguras y rápidas y trabaja sobre el espacio de velocidades-tiempo.
- Integrar el algoritmo en un programa que pueda ejecutarse en los robots del laboratorio. La integración implica trabajar con algoritmos de tratamiento de datos sensoriales desarrollados en el laboratorio en trabajos previos. Estos algoritmos son los de detección, seguimiento y modelado de los objetos móviles.
- Validar el planificador desarrollado mediante los resultados obtenidos en experimentos simulados y reales.

1.2. Estado del arte

Parece evidente que si un robot es un sistema capaz de realizar las tareas de forma autónoma, debe ser capaz de decidir los movimientos que le conducen a realizar esas tareas.

La resolución de trayectorias en entornos conocidos y estáticos se considera ya resuelta; [Latombe(1991)] recopila los diferentes métodos empleados. Sin embargo, incluso en un entorno perfectamente estructurado no es realista anticipar todos los posibles movimientos. En el mundo real la mayor parte de los entornos son solamente parcialmente conocidos y variantes en el tiempo ante la aparición de obstáculos imprevistos u obstáculos móviles imprevistos.

La planificación de trayectorias en robots cuenta clásicamente con dos aproximaciones: la planificadora y la reactiva. Desacoplar planificación de ejecución es una práctica común. En primer lugar se planifica una trayectoria con un mapa conocido a priori, y se desvía de su planificación para volver a ella cuando le aparecen obstáculos imprevistos.

En los casos en los que la información no se conoce a priori y tiene que ser estimada en tiempo real es muy probable que las predicciones no sean completas y que sea necesario replanificar, lo que conlleva unos costes de tiempo demasiado elevados. Como consecuencia, en entornos no conocidos y con presencia de obstáculos móviles la planificación de la trayectoria no puede ser resuelta simplemente construyendo un camino geométrico ya que peligran la integridad del robot.

En [Fox and Thrun(1997)] se introduce un método de planificación conocido como la ventana dinámica, donde se incluyen las velocidades alcanzables por el robot en la siguiente iteración y mediante un objetivo de dirección va seleccionando las velocidades para llegar al objetivo. Este método no incluía las trayectorias de los obstáculos móviles para anticipar el espacio libre por lo que las trayectorias resultantes pueden cambiar rápidamente y llevar a situaciones demasiado peligrosas. [Fiorini and Shillert(1998)] tiene en cuenta de forma explícita las velocidades de los obstáculos, representandolos directamente en el espacio de velocidades del robot, sin embargo no considera las restricciones cinemáticas de los robots.

Otros métodos [Brock and Khatib(1999)], [J. Mínguez and Alami(2001)], [Stachniss and Burgard(2002)] calculan varios pasos según el tiempo disponible. La trayectoria se calcula incrementalmente y el proceso puede ser interrumpido en cualquier momento para mantener la reactividad del vehículo. Esta aproximación es la más prometedora, sin embargo el entorno y su dinámica deben ser representados de un modo más eficiente.

El concepto de seguridad en navegación en entornos dinámicos se presenta en [LaValle and Kuffner(2001)]. Una navegación segura y estable hace necesario razonar sobre el comportamiento de los obstáculos móviles en el futuro además de definir cuánto tiempo se tiene en cuenta (horizonte temporal). En [Fraichard(1999)] se define el Estado de Inevitable Colisión, que corresponde a estados en los que no existe ninguna

trayectoria en el que el sistema pueda evitar la colisión, este método tiene un coste computacional muy elevado. [Gal and Shiller(2009)] soluciona el problema de seguridad eligiendo un horizonte temporal apropiado para el obstáculo de velocidad.

En [Owen and Montano(2005)] se presentaba un modelo para representar el entorno y las restricciones dinámicas y no-holonómicas del robot en el espacio de velocidades continuo del robot. Todas las órdenes que conducen a espacio libre pueden ser seleccionadas. En [Owen and Montano.(2006)] se presenta un planificador de trayectorias sobre la proyección bidimensional del espacio de velocidades dinámico (DVS). En [Lorente(2011)] se mejora el planificador, dotándolo de un Árbol de Situaciones que determina la estrategia más adecuada para aplicar en la siguiente iteración. Este Árbol de Situaciones razona sobre la proyección bidimensional del DVS, fusionando la información obtenida de los múltiples obstáculos detectados e identifica un caso de entre un conjunto finito de situaciones definidas en el árbol. Se establece una jerarquía de condiciones relacionadas con estas situaciones para obtener la estrategia de navegación.

No obstante, mientras [Lorente(2011)] trabaja en la proyección bidimensional del espacio de velocidades-tiempo, en este trabajo se planifican las trayectorias en el espacio tridimensional. Trabajar en la proyección bidimensional restringe la actuación y la capacidad de movimientos ya que se tiende al espacio libre demasiado rápido, cuando en muchas ocasiones todavía es posible tomar pares de velocidad que aparecen por debajo de las superficies prohibidas. La información y las conclusiones desarrolladas en [Lorente(2011)] se integran en un algoritmo de búsqueda A*, que considera tanto las estrategias de evitación más adecuadas para cada objeto como otros factores de decisión como el tiempo hasta alcanzar el objetivo o la seguridad. De este modo se pretende dotar al algoritmo de unas herramientas que le conduzcan a trayectorias seguras y óptimas en tiempo.

1.3. Estructura de la memoria

En los apartados anteriores se han introducido los antecedentes de la planificación de trayectorias en entornos dinámicos. Además se ha descrito el proyecto y se han definido los principales objetivos que se deseaba alcanzar.

En el capítulo siguiente se presentan los principales espacios relacionados con la planificación de trayectorias, más concretamente se desarrolla el modelo de espacio de velocidades-tiempo (DVTS) [Owen and Montano(2005)] ya que es el principal espacio sobre el que se realiza la planificación de trayectorias. En el capítulo 3 se expone el algoritmo de búsqueda A*, la función de coste que guía dicha búsqueda y la imposición de las restricciones cinemáticas y dinámicas a través del operador de expansión. Seguidamente el capítulo 4 introduce un método heurístico basado en [Lorente(2011)] para la selección de los objetivos de velocidad y su inserción en el espacio DVTS, se complementa con una estrategia especial para el caso de navegación en carriles y adelantamientos. Los principales módulos del programa se recogen en el capítulo 5. En los dos capítulos siguientes se presentan los escenarios de trabajo en los que se comprueba

la correcta ejecución del programa: en el capítulo 6 los escenarios del simulador Stage y en el capítulo 7 los experimentos llevados a cabo con los robots reales Pioneer del laboratorio. Finalmente en el capítulo 8 se presentan las conclusiones del trabajo, posibles aplicaciones y algunas líneas de desarrollo.

Los anexos están estructurados para completar la información presentada en el cuerpo del proyecto. El anexo A amplía los métodos de selección de estrategias empleados. El anexo B resume los principales filtros de información proporcionados por el departamento que fueron integrados en el programa. Los anexos C y D desarrollan en mayor profundidad los escenarios descritos en los capítulos 6 y 7. El anexo E presenta algunos resultados adicionales sobre el A* así como otro tipo de pruebas que se llevaron a cabo para la realización del proyecto.

Finalmente se incluye un índice de figuras y la bibliografía.

Capítulo 2

El espacio de control y la búsqueda en el mismo

2.1. El espacio de velocidades-tiempo

En este proyecto se ha desarrollado un método de cálculo de trayectorias para la navegación de robots autónomos. Para comprender un poco mejor los conceptos empleados, a continuación se definen los principales espacios involucrados.

- **El espacio de trabajo.**

Es el espacio euclídeo en el que se desplaza el robot y aparecen los diversos obstáculos fijos y móviles. Se trabaja en referencias absolutas. Es el presentado en la figura 2.1a.

- **El espacio de configuraciones.**

El objetivo del espacio de configuraciones es reducir el robot a un punto para el que se planifican las trayectorias. Por una parte, los obstáculos móviles se inscriben en una circunferencia cuyo radio se amplía incluyendo el radio del robot. Esta representación es conservadora ya que no considera los detalles de la forma del robot, pero simplifica considerablemente el modelo.

Por otra parte, para engordar los objetos estáticos de tipo paredes, se calcula la recta paralela situada a una distancia del radio del robot. En el anexo B.2.3 se amplía este método.

En este modelo el espacio de configuraciones es un espacio cartesiano robocéntrico, por lo que todos los objetos son referenciados al robot, figura 2.1b.

- **El espacio de velocidades-tiempo: DVTS.**

El espacio de configuraciones se emplea como intermedio para modelar los obstáculos en el espacio de velocidades-tiempo, denominado DVTS (*Dynamic Velocity Time Space*). El DVTS es espacio principal en el que se realiza la planificación.

Se trata de un espacio tridimensional con las velocidades lineales, angulares y el tiempo en sus ejes. En este espacio se representan los objetos prohibidos como una superficie de velocidades prohibidas que conducen a colisión.

Para generar las superficies prohibidas, se considera que los obstáculos móviles se desplazan a velocidad constante siguiendo una trayectoria rectilínea. Esta simplificación puede parecer bastante estricta, pero debido a la elevada frecuencia de muestreo del sistema de control resulta suficiente.

La zona recorrida por el obstáculo se denomina Banda de Colisión. Desde el robot y tangente a su orientación (nula en este espacio) se genera un haz de circunferencias y se calculan los puntos de intersección con la Banda de Colisión y los tiempos asociados. Así finalmente para cada radio del haz se obtienen las velocidades y tiempos (v_1, w_1, t_1) y (v_2, w_2, t_2) límites que conducen a colisión. Para generar las superficies prohibidas se parte de los pares de velocidades lineales, angulares y tiempos obtenidos con la Banda de Colisión y se interpola la superficie total. Este proceso se detalla más adelante en el apartado 2.3.3.

- **El espacio de velocidades: DVS.**

Es la proyección bidimensional del DVTS. Sobre este espacio se razona en trabajos anteriores, [Lorente(2011)], para la selección de las estrategias de evitación identificando las situaciones a partir de las proyecciones de las superficies prohibidas.

En la figura 2.1b se muestra el espacio de configuraciones, en el que el robot ha sido reducido a un punto que es el origen del sistema de referencia. El objeto dinámico ha sido engordado y se puede ver el concepto de la Banda de Colisión. Para el radio r_1 aparecen dos puntos de corte con la Banda. Estos puntos generan los pares de velocidades $V_1 (v_1, w_1, t_1)$ y $V_2 (v_2, w_2, t_2)$ que se marcan en el DVTS (figura 2.1c) y en su proyección DVS (figura 2.1d).

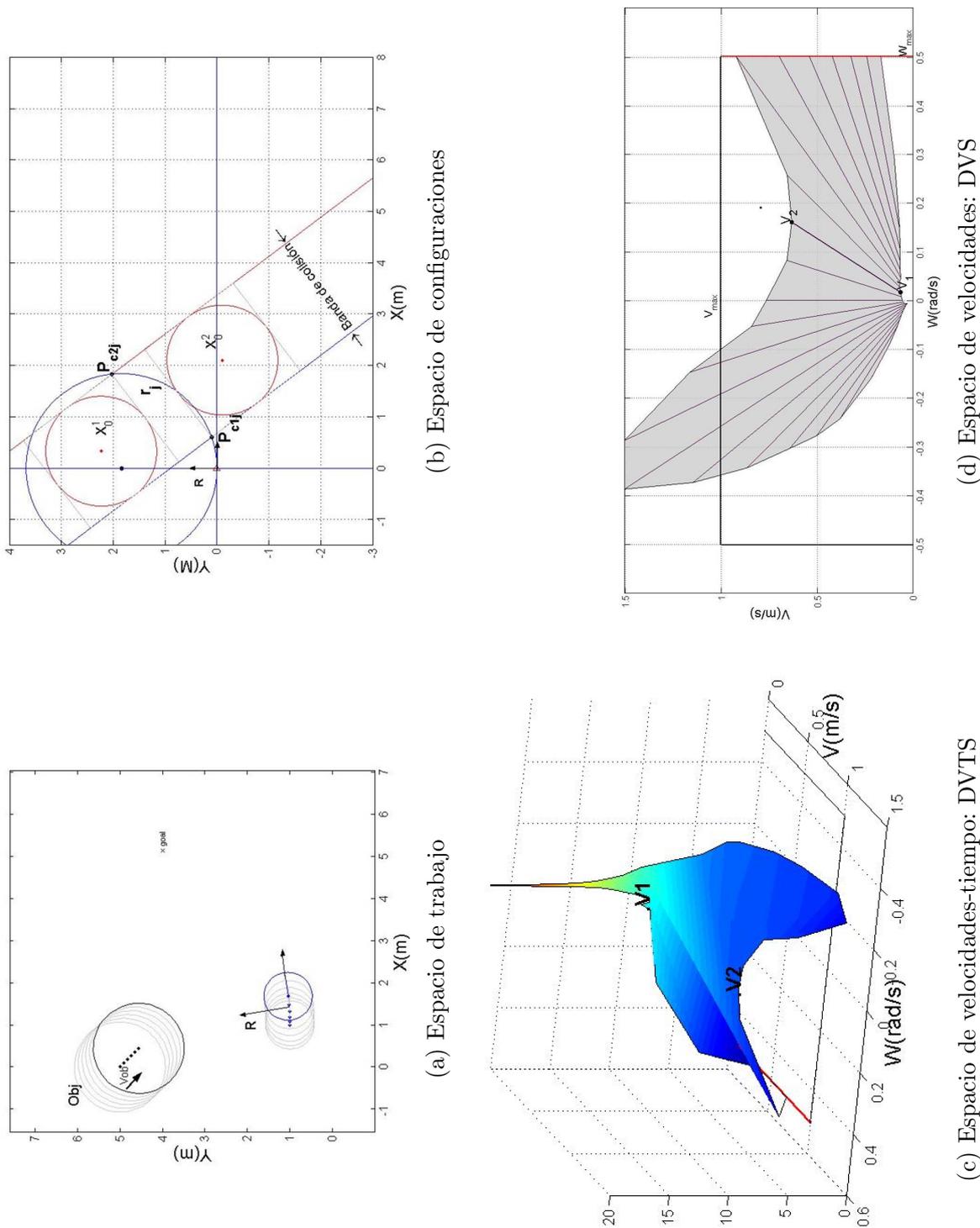


Figura 2.1: Diferentes espacios

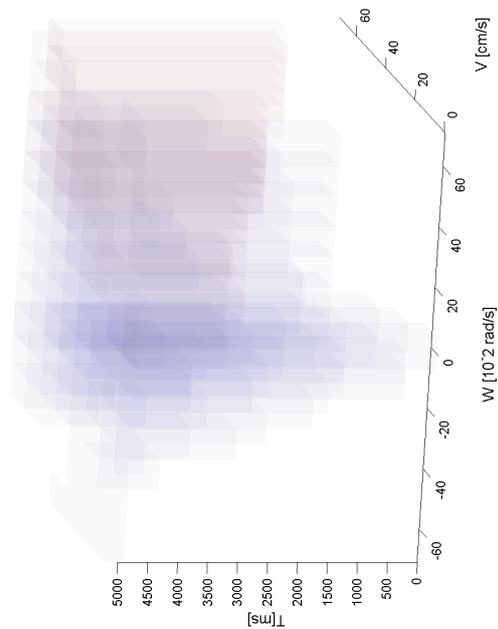
2.2. Cálculo de trayectorias en el espacio DVTS

En trabajos previos relativos al cálculo de trayectorias en entornos dinámicos, el espacio habitual de trabajo es el de configuraciones extendido en el tiempo. En el presente trabajo el espacio de configuraciones se representa en el espacio de velocidades-tiempo. De esta forma la planificación de trayectorias consiste en la búsqueda en este espacio de la trayectoria que minimiza el tiempo de alcance de los objetivos, lo que tiene algunas ventajas sobre los métodos clásicos al trabajar sobre el espacio de control del robot. Nótese que al considerar el DVTS está implícita la información de aceleraciones, es decir, las restricciones dinámicas se pueden representar de manera natural en este espacio tal y como se explicará más adelante con los operadores de expansión.

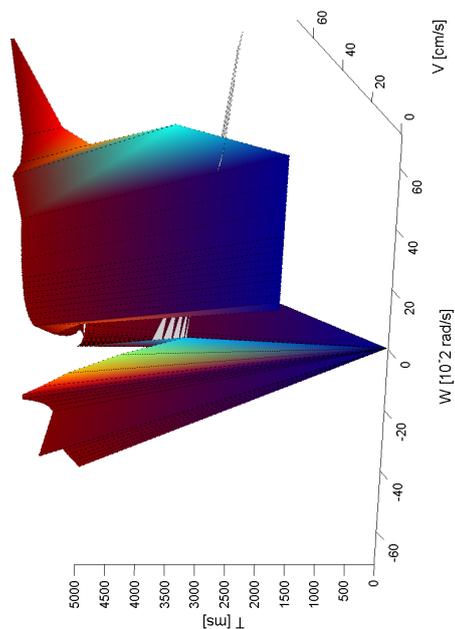
El espacio es discretizado en prismas rectangulares, y el espacio delimitado por las superficies prohibidas calculadas se convierte en celdas ocupadas, quedando el resto del espacio como celdas libres en las que se realizará la búsqueda. En la sección 2.3.3 se amplia.

Para realizar la búsqueda se trabaja con un algoritmo A*(capítulo 3) que va expandiendo un árbol de celdas guiado por una función de costes que pondera términos de seguridad, estrategias de evitación y cercanía al objetivo.

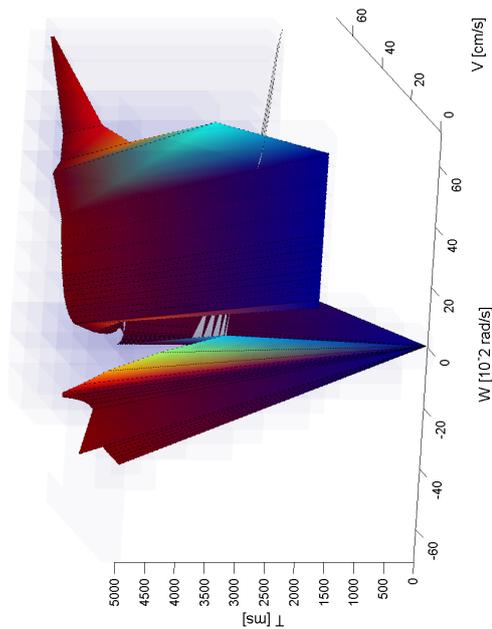
Las imágenes de la figura 2.2 muestran el proceso de trabajo en el DVTS. En primer lugar se calculan las superficies prohibidas para los obstáculos móviles y estáticos. En la figura 2.2a se observan dos obstáculos móviles y uno estático (la superficie gris). A continuación las celdas correspondientes a las superficies de velocidades prohibidas y superiores se marcan como ocupadas en la malla discretizada, figura 2.2b. En la figura 2.2c se muestra la búsqueda realizada por el A* entre las casillas no ocupadas. En la última figura, 2.2d, se aprecia como la planificación discurre por debajo de superficies prohibidas, algo que en el trabajo anterior era evitado ya que se intentaba siempre mantener la velocidad en zonas libres.



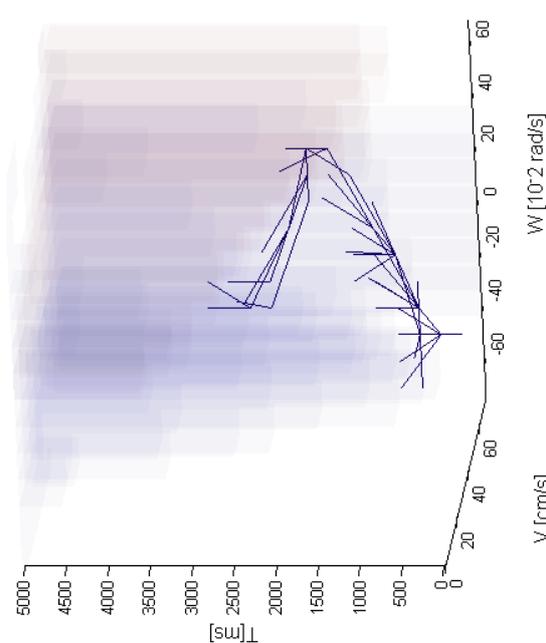
(b) Discretización del DVTS



(a) Superficies prohibidas



(d) Vista general



(c) Búsqueda en el DVTS

Figura 2.2: Espacio de velocidades tiempo

2.3. Discretización del DVTS

El espacio de trabajo es el espacio de velocidades lineales, velocidades angulares y tiempo. Es un espacio tridimensional semiacotado por las velocidades máximas alcanzables y un horizonte de tiempo. Para trabajar en este espacio, se discretiza con prismas rectangulares de longitudes constantes, δv , $\delta \omega$ y δt .

Cada celda se identifica mediante sus coordenadas centrales en v y w , y su menor tiempo. Además la celda contiene información sobre su estado (correspondiente a estar ocupada o al valor que adquiere en el algoritmo de búsqueda). Todas las celdas se almacenan en una malla, para la que se definen las operaciones de cálculo de superficies estáticas y dinámicas así como otras operaciones necesarias en el programa.

2.3.1. Resolución de la malla: tamaño de celda

En toda discretización, el tamaño de malla juega un rol importante. La discretización de una malla puede ser constante o variable. Aunque una discretización variable permitiría una mayor resolución en zonas conflictivas y una menor resolución en zonas menos interesantes, se optó por una discretización constante para homogeneizar todo el programa, así como los métodos de cálculo. Un razonamiento similar se empleó para descartar los métodos de descomposición aproximada de celdas [Latombe(1991)], que además de introducir una discretización variable aumentaban el tiempo de cómputo en cada iteración al hacer necesario rehacer el mallado para cada conjunto de superficies. La estructura de celdas permanece constante a lo largo de todo el programa, con la misma discretización predefinida al lanzarlo.

Para definir el tamaño de las celdas, era necesario llegar a un compromiso entre la precisión alcanzada y el coste computacional que conlleva un aumento de la resolución. Con la variación de tiempo δt , fue sencillo ya que posee un significado físico que corresponde a la duración de cada iteración o actualización del laser además del periodo entre lanzamientos de órdenes. δv y $\delta \omega$ se definieron buscando un equilibrio entre la resolución de la malla y la velocidad de cálculo.

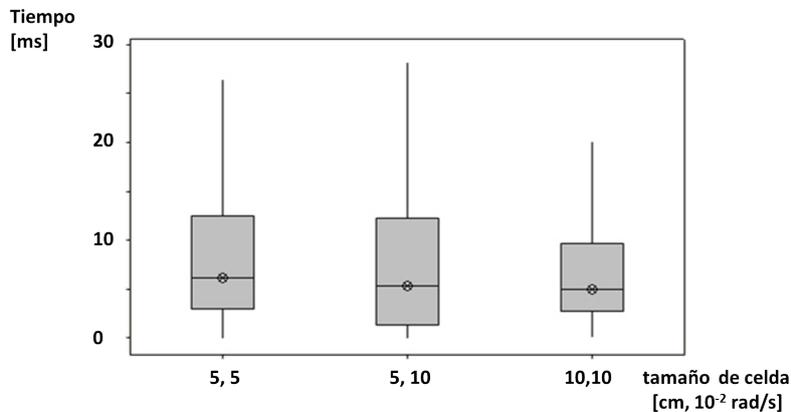


Figura 2.3: Tiempos de cálculo de una superficie según el tamaño de celda

En la figura 2.3 aparecen las estadísticas del tiempo de cómputo de la superficie tridimensional para diferentes discretizaciones de la malla. Se observa como emplear mallas más grandes favorece considerablemente el tiempo de cómputo. Para tamaños de celda mayores a los que aparecen en la figura 2.3 el espacio de búsqueda resultaba demasiado restrictivo, lo que conducía a problemas de búsquedas no finalizadas. Finalmente la discretización empleada es de $\delta v = 0.1$ m/s y $\delta \omega = 0.1$ rad/s.

2.3.2. Horizonte Temporal

Una variable muy importante en este algoritmo es el horizonte temporal hasta el cual se considera la planificación. Planificar a un tiempo demasiado corto deja obstáculos importantes sin tratar, mientras que horizontes demasiado tardíos aumentan el tiempo de cómputo. Una de las premisas de este programa es su continua actualización, lo que favorece emplear horizontes de planificación no muy largos.

El horizonte temporal máximo con el que se trabaja finalmente es de 5 s. No obstante, para cada iteración se calcula el tiempo estimado de alcance del objetivo cartesiano a la velocidad actual y se elige como horizonte de planificación temporal el menor de los dos. Por encima del horizonte temporal no se buscan trayectorias, y en caso de que las superficies de los obstáculos superen este nivel, se proyecta sobre el último plano de casillas la superficie que quedaría por encima.

2.3.3. Obtención de la superficie completa

A partir de los pares de velocidades calculados con la Banda de Colisión [Owen and Montano(2005)], se interpola la superficie completa. En una primera aproximación, en este trabajo para dos pares de comandos consecutivos se calculaba la superficie comprendida entre los cuatro puntos mediante una factorización SVD con la librería ALGLIB [Alglib()], calculando para cada celda su posición relativa respecto a la superficie. Sin embargo este método tenía problemas de convergencia en demasiadas ocasiones, además de resultar muy lento.

Por ello se modificó la estrategia y se pasó a calcular una serie de puntos intermedios que cubriesen todas las celdas superficie intermedia tal y como se muestra en la figura siguiente.

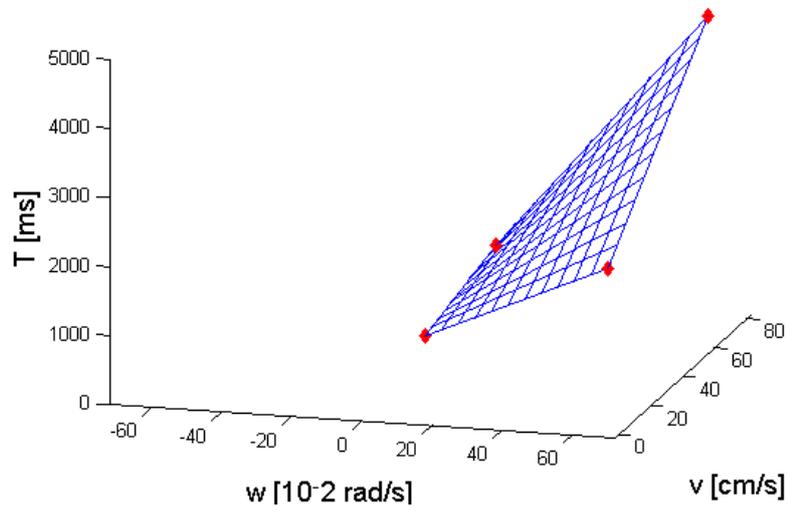


Figura 2.4: Mallado generado entre dos pares de velocidades consecutivos

Dado que la superficie prohibida define el límite de los comandos prohibidos, en la malla se prohíben las celdas que contienen la superficie y las de tiempos superiores.

Capítulo 3

Búsqueda de trayectorias en DVTS

3.1. Introducción al Algoritmo A*

El algoritmo de búsqueda A* pertenece a los algoritmos de búsqueda en grafos. Bajo unas determinadas condiciones encuentra el camino de menor coste entre un nodo origen y un nodo objetivo.

El algoritmo A* se emplea para encontrar un camino de celdas libres desde las velocidades actuales $N_{inicial}$ hasta el objetivo indicado N_{goal} , [Latombe(1991)]. Este algoritmo explora iterativamente las celdas o nodos, abriendo diferentes caminos desde el nodo inicial. Para cada nodo visitado N , el algoritmo ha producido uno o varios caminos conectándolo con $N_{inicial}$ pero solo memoriza la representación del camino cuyo coste asociado es mínimo. En cada iteración de este algoritmo, el conjunto de nodos expandidos forma un árbol de expansión con origen en $N_{inicial}$. Cada nodo visitado nace de su predecesor, por lo que para obtener la trayectoria desde un nodo N hasta el inicial, basta con recorrer el árbol hacia su origen.

El algoritmo A* asigna a cada nodo explorado una función de coste. Los nodos explorados se insertan en una lista denominada *Abiertos*, en la que se ordenan de menor a mayor coste. En cada iteración se analiza uno de los nodos de *Abiertos*, el de menor coste. Mediante el operador de expansión, introducido en el apartado 3.2 se van abriendo sus casillas vecinas. Para cada celda abierta se comprueba que está libre de colisión, que es alcanzable en ese nivel temporal y se calcula el coste asociado.

Si está libre de colisión la celda aparecerá marcada en la malla como vacía, en cuyo caso se estima el coste asociado mediante una función de coste. Básicamente, se trata de una estimación heurística del coste asociado. La función consta de dos términos,

$$f(N) = g(N) + h(N) \tag{3.1}$$

donde $g(N)$ representa el coste asociado al camino desde el nodo inicial N_{inicio} hasta el nodo actual N , $h(N)$ es una estimación heurística del coste del camino de menor coste entre N y $N_{objetivo}$ que incluye el coste asociado a la seguridad. Los detalles del método de cálculo de estos términos se encuentran en la sección 3.3.

Una vez se han visitado todos los vecinos, se marca la celda como ya expandida y se añade al árbol de expansión.

A continuación, en la figura 3.1 muestra un esquema del proceso de expansión. Desde la celda a se abren las celdas adyacentes. b, d, e y g están marcadas como prohibidas por los que no se expanden. De las celdas abiertas y vacías se elige la de menor coste, la celda f y se expande. A continuación se expandiría la de menos coste, en este ejemplo la celda c. Progresivamente se va creando un grafo de expansión que conecta las celdas tridimensionales.

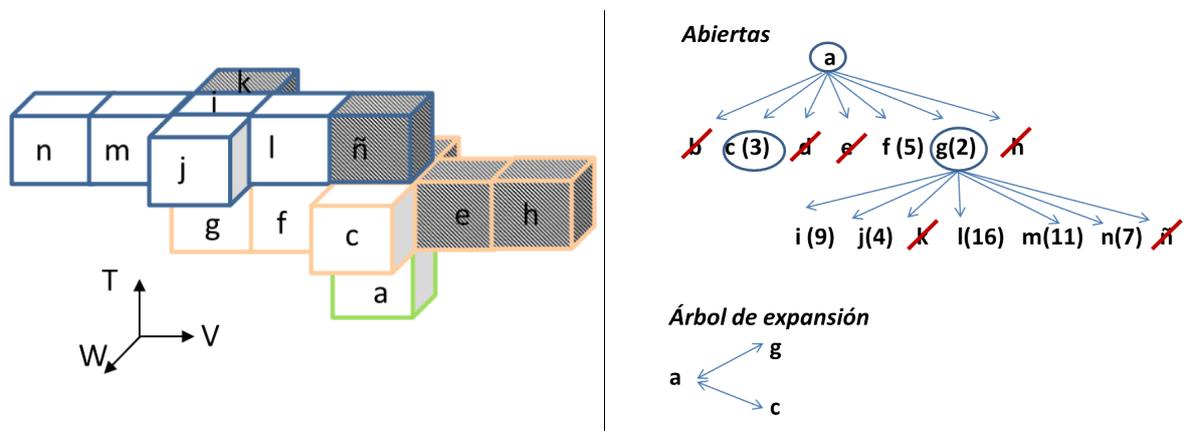


Figura 3.1: A*: expansión y generación de *Abiertas* y del árbol de expansión

El pseudocódigo correspondiente a estas acciones, se encuentra en la figura 3.2.

El tiempo de búsqueda de un camino satisfactorio podía sobrepasar el tiempo adecuado para esta parte del programa e impedir que la ejecución del programa principal se realizara en el periodo definido. Por ello, al lanzar el algoritmo se inicializa un temporizador. En cada iteración se comprueba que no se ha sobrepasado el tiempo de búsqueda máximo permitido, si se ha sobrepasado se termina la búsqueda y se devuelve la trayectoria correspondiente al nodo de menor coste del árbol de expansión.

Al lanzar el algoritmo, se introduce la lista de los subobjetivos de velocidad calculados. El cálculo de los subobjetivos corresponde con las heurísticas de evitación y se desarrolla en la sección 4.

Una vez finalizado el proceso de búsqueda, si se ha conseguido alcanzar al menos un subobjetivo se reconstruye la trayectoria. Esto se realiza recorriendo en el árbol de expansión el camino desde el último objetivo alcanzado hasta la celda inicial (la posición actual). Se guarda la trayectoria completa calculada permitiendo, en caso de que una búsqueda no pueda finalizar, emplear las velocidades calculadas en los periodos anteriores. Al programa principal se le devuelve el par de velocidades (v,w) de la trayectoria en el primer nivel de tiempo, que será la orden de velocidad del robot para la próxima iteración.

```

Inicialización del temporizador;
Inserción de los subobjetivos;
Inserción del par de velocidades actual (v,w) en Abiertos ( $N_{init}$ );
while (No se ha alcanzado el último subobjetivo) or (Queda tiempo de cálculo)
  { Se procesa la casilla de menor coste de Abiertos;
    for (Todos los vecinos)
      { if (No está ocupada) and (Es alcanzable)
        { Cálculo del coste asociado;
          if (Ya había sido visitada) and (Los predecesores coinciden)
            Se mantiene en Abiertos la de menor coste;
          else Se inserta en abiertas;
        }
      }
    else Se descarta la celda vecina;
  }
  Se guarda el nodo expandido en el árbol de expansión.
}

```

Figura 3.2: Estructura del algoritmo A*

3.2. El operador de expansión

Los robots reales presentan restricciones dinámicas, esto es, las velocidades no son alcanzables instantáneamente sino que se rigen por unas aceleraciones máximas. Para representar esta restricción, en [Fox and Thrun(1997)] se presentó la *Ventana Dinámica* que delimita todas las velocidades alcanzables según la aceleración del robot en un periodo.

Además, para robots no holonómicos es común restringir el movimiento a caminos rectos o circulares para cada periodo de muestreo. Esto permite respetar las restricciones cinemáticas así como mantener una curvatura continua. El tipo de trayectorias descritas, curvas clotoideas y anticlotoideas obtenidas manteniendo la velocidad lineal o angular constante respectivamente, se presenta en [Owen and Montano.(2006)]. Así, durante la búsqueda para cada nivel temporal solo se permite el desplazamiento o en velocidad lineal o en velocidad angular dentro de la *Ventana Dinámica*.

El cumplimiento de estas restricciones se introdujo en el método de expansión de las celdas del A*. El algoritmo de búsqueda emplea un árbol de expansión (*spanning tree*) en el que los nodos hijos se definen mediante un operador de expansión.

Para determinar las celdas que se consideraban expandibles desde otra se plantearon dos alternativas:

- La primera opción consistía en expandir las cinco celdas adyacentes, esto es las cuatro ortogonales del mismo nivel de tiempo y la inmediatamente superior. Esto permitía una exploración más genérica de todas las alternativas. Se conservaba la información del nodo que alcanzaba el nivel de tiempo y en cada expansión se

comprobaba que la celda fuera accesible en ese nivel de tiempo (que estuviera en el interior de la ventana dinámica), así como que sólo realizara aceleraciones en velocidad lineal o angular para cumplir las condiciones de curvatura continua. Este método permitía una expansión genérica de todas las celdas contiguas, independientemente de la discretización empleada. Las comprobaciones de cumplimiento de las restricciones cinemáticas y dinámicas se realizaban al abrir la celda. La figura 3.3a presenta un esquema de este operador de expansión.

- La segunda alternativa se tomó para el caso concreto de la discretización empleada. Una vez elegido el tamaño de celda, se comprobó que era posible predefinir las celdas que conformaban la ventana dinámica como vecinas directamente en el nivel de tiempo superior, evitando comprobaciones adicionales al realizar la expansión.

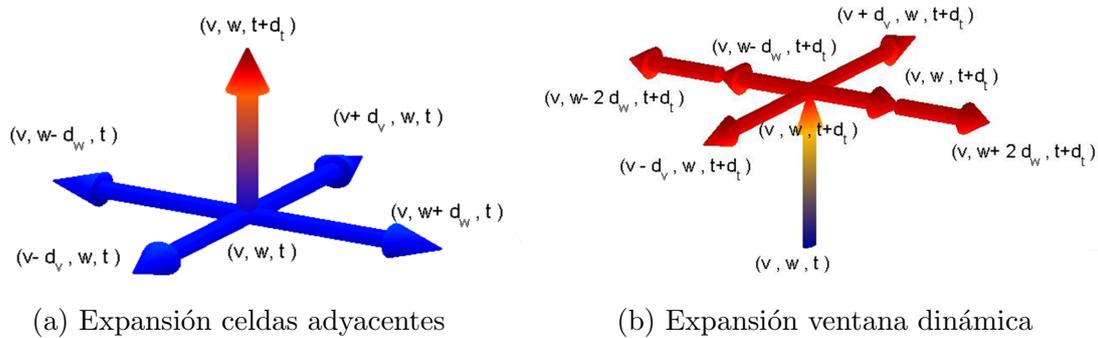


Figura 3.3: operadores de expansión

Con la primera metodología la búsqueda en el nivel temporal de la casilla a expandir está permitida, ya que las comprobaciones se realizan a posteriori, mientras que con la segunda metodología se expanden todas las celdas accesibles de la ventana dinámica para la iteración siguiente.

3.3. Costes y seguridad

Como ya se ha mencionado, la casilla a expandir en cada iteración se determina seleccionando en la lista de abiertas el nodo de menor coste. Para conseguir un resultado óptimo se debe ponderar tanto la seguridad de la trayectoria como que el objetivo se alcance lo más rápido posible. La función de coste presentada en la 3.3 hace referencia a dos tipos de costes diferentes: el coste de llegar desde el nodo inicial al nodo que se está estudiando $g(N)$, el coste estimado de llegar desde el nodo actual hasta el objetivo $h(N)$.

$$f(N) = g(N) + h(N) \quad (3.2)$$

A continuación se desarrolla la composición y método de cálculo de estos sumandos.

- **Coste hasta la casilla actual $g(N)$**

Indica el coste de llegar desde el nodo inicial hasta el nodo que se está estudiando. El tiempo transcurrido desde el nodo inicial hasta el nodo actual aparece directamente representado con la componente de tiempo de la celda. Este tiempo se normaliza al número de iteraciones equivalentes, ya que la discretización de las celdas en tiempo corresponde a la duración de cada iteración,

$$g(N) = numCasillas \quad (3.3)$$

- **Coste heurístico hasta el objetivo $h(N)$**

La función de costes del algoritmo A* consta de un término $h(N)$ que es una estimación heurística del coste $h^*(N)$ del camino de menor coste entre N y $N_{objetivo}$. Este coste se dice que es admisible si y sólo si para cada nodo N del grafo G se satisface :

$$\forall N \in G : 0 \leq h(N) \leq h^*(N) \quad (3.4)$$

Si $h(N)$ es admisible, se garantiza que el algoritmo A* devuelve un camino de mínimo coste entre el nodo inicial y el final. Por ello, los factores de coste heurísticos deben ser optimistas.

El término de coste heurístico hasta el objetivo consta de tres componentes ponderadas: la primera hace referencia al coste de alcanzar la velocidad más idónea para ese instante $h_v(N)$, la segunda introduce el coste asociado en el espacio de configuraciones hasta alcanzar el objetivo cartesiano h_{dist} la tercera componente h_s aporta un coste relativo a la seguridad.

$$h(N) = \alpha_v h_v(N) + \alpha_{dist} h_{dist}(N) + \alpha_s h_s \quad (3.5)$$

A continuación se amplía el método de cálculo de estos tres términos. La evaluación de los factores de ponderación se encuentra en el anexo E.2.

- ***Términos de velocidad $h_v(N)$***

Las estrategias de evitación para cada obstáculo móvil constituyen los subobjetivos de velocidad que busca el A*. En la sección 4 se presentan las consideraciones realizadas para introducir las heurísticas de evitación más adecuadas para los obstáculos móviles más inminentes, así como su ordenación según determinadas prioridades para ser considerados.

Dado que esta búsqueda incluye diferentes subobjetivos, una vez se identifica el más prioritario se expanden todas las celdas coincidentes y se les asigna el coste correspondiente (en el que este componente será nulo) para tiempos superiores. Una vez expandidas todas las casillas superiores de un subobjetivo se marca éste como alcanzado. De esta manera las casillas expandidas a partir de entonces priorizan la búsqueda del siguiente subobjetivo.

Cada subobjetivo va dirigiendo sucesivamente la búsqueda, por lo que alcanzar rápidamente el subobjetivo siguiente es de gran relevancia. En la figura 3.4 se desglosa el cálculo de este término, que a su vez está formado por dos sumandos:

- *Número de pasos mínimo: numSteps*

Es el indicador del número de iteraciones necesarias hasta alcanzar el objetivo de velocidad o lo que es lo mismo, el tiempo mínimo hasta alcanzarlo. Está definido por la dinámica del robot, en particular por sus aceleraciones máximas. Se trata de un valor acotado por su propia naturaleza y se calcula como la suma del número de iteraciones mínimas necesarias en v y en w para alcanzar el objetivo.

- *Cercanía relativa: relSteps*

Sirve para que dado el mismo número de pasos hasta el objetivo, priorizar una casilla que está más cerca. En trabajos anteriores se ha demostrado que trabajar con velocidades extremas conduce al objetivo en tiempo mínimo. Por otro lado en este proyecto se trabaja también con velocidades no extremas de la ventana dinámica ya que en ocasiones ayuda a mejorar la precisión de los movimientos y evita sobreoscilaciones.

Este término permite priorizar las velocidades máximas alcanzables en cada iteración (los extremos de la ventana dinámica, sección 3.2) frente a otros comandos alcanzables pero no máximos en los casos en los que el objetivo de velocidad esté alejado mientras que también permite no estar obligado a trabajar siempre con los extremos de la ventana.

```

//número de pasos (niveles temporales) en cada dirección

distV = abs(v_candidato - v_objetivo)
stepsV = ceil( distV / maxAcLinealPeriodo )
distW = abs(w_candidato - w_objetivo)
stepsW = ceil( distW / maxAcAngularPeriodo )

//número de iteraciones mínimas (tiempo) hasta alcanzar la velocidad objetivo
numSteps = stepsV + stepsW

//distancia relativa dentro de la ventana dinámica a la velocidad objetivo

relSteps = parteDecimal( distV / maxAcLinealPeriodo ) + parteDecimal( distW / maxAcAngularPeriodo )

//término de velocidad
h_v = numSteps + relSteps

```

Figura 3.4: Término heurístico de velocidad h_v

- **Ponderación del coste en el espacio de trabajo h_{dist}**

Todos los términos introducidos hasta el momento hacen referencia a aspectos de velocidades o tiempos, y aunque éstos dependen en última instancia de la distancia al objetivo en el espacio de trabajo, esta información no aparece de forma directa. Por ello se introduce este término.

- *Tiempo hasta alcanzar el objetivo*

Se trata de una estimación del tiempo que necesita el robot hasta alcanzar su destino. Se calcula como el tiempo que le cuesta alcanzar la velocidad máxima ideal que le conduce al objetivo (*freemotion*, capítulo 4) más el tiempo que le costaría recorrer la distancia que le falta hasta el destino a la velocidad máxima.

Como se explicará más adelante, el par de velocidades *freemotion* no lleva directamente al objetivo sino que al recalcularse en cada iteración converge en él. Para estimar de forma optimista el tiempo que le costaría alcanzar el destino, se calcula el tiempo que tardaría a velocidad lineal nula en girar para orientarse más el tiempo que le costaría recorrer la distancia al destino.

```
//cálculo de la distancia euclídea y la variación angular hasta el objetivo
```

$$dist = \sqrt{(x_{candidato} - x_{objetivo})^2 + (y_{candidato} - y_{objetivo})^2}$$

$$ang = \arctan \frac{y_{candidato} - y_{objetivo}}{x_{candidato} - x_{objetivo}} - tita_{candidato}$$

```
//tiempo en alcanzar la velocidad ideal que orienta y lleva al objetivo
```

$$t_{ac} = abs\left(\frac{v_{ideal} - v_{candidato}}{a_v}\right) + abs\left(\frac{w_{ideal} - w_{candidato}}{a_w}\right)$$

```
//tiempo a máxima velocidad hasta alcanzar el objetivo
```

$$t_v = abs(dist/v_{ideal})$$

$$t_w = abs(ang/w_{ideal})$$

$$tiempo = t_{ac} + t_v + t_w$$

```
//Paso de tiempo en segundos a pasos temporales
```

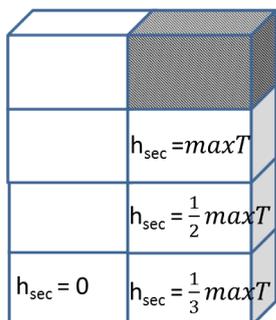
$$h_{dist} = ceil(tiempo/delta_t)$$

Figura 3.5: Término heurístico de cercanía en el espacio de trabajo h_{dist}

- **Término de seguridad h_s**

Además de buscar una trayectoria libre hasta alcanzar el comando de velocidad elegido, es necesario plantear la seguridad de dicha trayectoria. Una búsqueda en el tiempo permite considerar velocidades que se encuentran por debajo de la superficie prohibida, Por ello, se añade un término de seguridad.

Si la celda se encuentra al principio de la planificación, dentro de un margen de seguridad predefinido comprueba la cercanía de la superficie prohibida. Este margen de seguridad corresponde al número de iteraciones que le costaría frenar a la máxima velocidad. Si la superficie prohibida aparece dentro del margen se considera que la velocidad no es segura, por lo que se penaliza el coste de la celda abierta de forma proporcional a la distancia a la que aparecen las velocidades prohibidas. En la figura siguiente se representa de forma esquemática el cálculo de este término.



//se comprueban todas las celdas superiores dentro del margen de seguridad. Si alguna de ellas está ocupada, se penaliza el coste de forma proporcional a la distancia de la primera celda superior prohibida

$$h_s = \frac{1}{\text{celdasLibresHastaProhibido}} * \text{HorizonteTemporalMax}$$

Figura 3.6: Término heurístico de seguridad h_s

Coste Total

Cada uno de estos términos recibe una ponderación determinada. Los coeficientes de ponderación α_v , α_{dist} y α_s se pueden ajustar para favorecer más la seguridad, la anticipación mediante la selección de estrategias o el tiempo de alcance del objetivo. En el anexo E.2 se presentan diferentes resultados comparativos.

El coste total de cada celda presentado en la ecuación 3.3 quedaría así:

$$\begin{aligned} \text{coste}_{total} &= \text{coste}_{N_{inicio} \rightarrow N} + \text{coste}_{estimado}_{N \rightarrow N_{objetivo}} \\ &= g(N) + h(N) \\ &= g(N) + \alpha_v h_v + \alpha_{dist} h_{dist} + \alpha_s h_s \end{aligned}$$

Capítulo 4

Estrategias de navegación

La planificación de trayectorias se realiza en este proyecto sobre el espacio de velocidades del robot, por lo que los subobjetivos que se introducen para el guiado heurístico de la búsqueda de trayectorias mediante el A* son pares de velocidades.

Al plantear la orden ideal de velocidad del robot se consideran principalmente dos criterios: la continuidad de la curvatura y la rapidez en alcanzar el objetivo. Por un lado, para una situación en la que el robot circulase libre de obstáculos y sin restricciones cinemáticas ni dinámicas, la trayectoria más rápida se logra realizando en primer lugar un giro para alinearse con el objetivo y avanzando después hacia él lo más rápido posible en dirección lineal. Esta trayectoria resulta muy brusca y poco continua. Por otra parte una trayectoria que lleva directamente al objetivo respetando la continuidad de la curvatura, consiste en recorrer a las máximas velocidades posibles el arco de la circunferencia que une la posición del robot con el objetivo. Este método resulta demasiado lento.

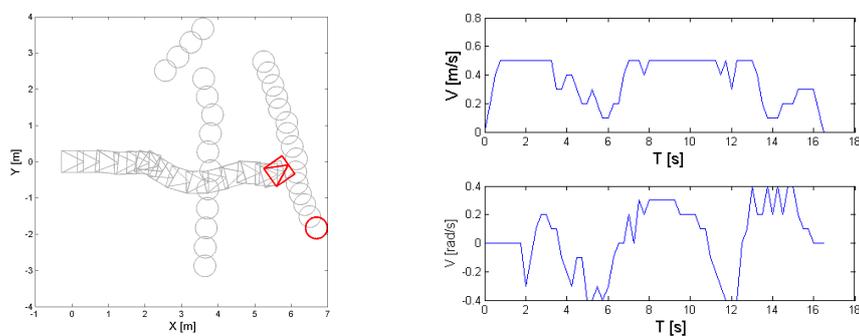
En trabajos anteriores se comprobó que elegir una orden de velocidad intermedia, denominada *freemotion*, entre estas dos opciones permite al robot alcanzar su objetivo de forma óptima en situaciones libres de obstáculos. Sin embargo generalmente aparecen diversos obstáculos que obligan a modificar las trayectorias, ya que el objetivo de *freemotion* no siempre es libre y/o alcanzable.

Al comienzo de este proyecto se trabajaba simplemente introduciendo el objetivo de *freemotion*. Esto en ocasiones conducía a maniobras mejorables. Si se trabajaba con coeficientes de seguridad altos, este método era demasiado conservador, ya que aunque podía buscar por debajo de superficies prohibidas tendía siempre a velocidades completamente libres, perdiendo gran parte de la potencia del método. Trabajando con coeficientes de seguridad bajos, el robot reaccionaba más tarde ante los obstáculos, conduciendo a muchas maniobras de frenada o incluso resultando peligroso.

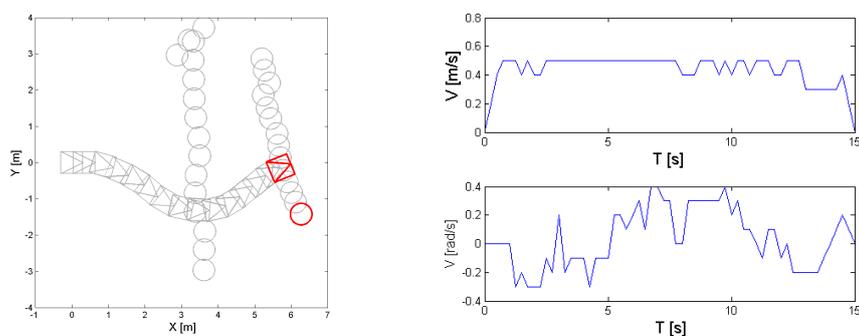
Para mejorar este comportamiento se decidió utilizar una heurística de actuación más adecuada aprovechando la información de trabajos anteriores.

En el ejemplo siguiente se compara el método inicial con el método ya completo que integra las heurísticas de evitación de obstáculos. Al emplear las heurísticas de

selección de objetivos 4.1b la velocidad lineal se mantiene más continua se obtiene una trayectoria más suave, logrando además alcanzar el objetivo más rápidamente.



(a) Empleo del *freemotion*



(b) Integración de heurísticas de evitación

Figura 4.1: Escenario comparativo de métodos de selección de subobjetivos

4.1. Heurísticas para la selección de subobjetivos

En [Lorente(2011)] se desarrolló una estrategia para la selección de la estrategia de evitación óptima mediante un Árbol de Decisiones. En el anexo A.1 se amplía la información de este trabajo.

Para este proyecto, se trata cada obstáculo de forma individual. Según la proyección de la silueta de las velocidades prohibidas (la forma del obstáculo en el DVS) , aparecen las siguientes estrategias:

- Pasar por delante
- Evitar por detrás
- Alineación
- Dejar pasar

4.1.1. Integración de estrategias

De esta manera, se calculan las órdenes de movimiento para los obstáculos que son relevantes. Aquí conviene hacer un inciso sobre cuándo se considera un obstáculo como relevante.

Relevancia de obstáculos

- No se consideran los obstáculos que se encuentran a una distancia mayor de $2/3$ el alcance del láser. Con esta condición se permite que obstáculos demasiados lejanos sean procesados durante más tiempo por los algoritmos de seguimiento (mejorando sus estimaciones) pero no afecten a la planificación de trayectorias.
- Para obstáculos más próximos considerados, se calculan las velocidades límites que definen su superficie prohibida del modo indicado en el apartado 2. Una segunda criba se realiza no calculando la orden de evitación para obstáculos que se encuentran a una distancia mayor que el doble de la máxima distancia alcanzable en el horizonte temporal considerado por el A*.

$$distancia \leq 2 * velocidad\ máxima\ del\ robot * horizonte\ temporal\ máximo$$

Para los obstáculos que cumplen ambas restricciones, se calcula la estrategia asociada de evitación.

Inserción de los objetivos de velocidad

Así, para cada objeto dinámico considerado, se ha definido la estrategia más adecuada de evitación según la silueta de las velocidades prohibidas. Finalmente es preciso integrar el conjunto de todas las estrategias para introducir en el A* subobjetivos que ayuden a dirigir la búsqueda.

- Según el tamaño angular en el DVS:
Este tamaño angular es representativo tanto de la distancia a la que se encuentra el obstáculo, como de la velocidad y de su dirección y sentido relativos al robot. A mayor tamaño angular, proximidad o velocidad del objeto, es decir menor seguridad y mayor prioridad en su evitación. En la figura siguiente se muestra un tamaño angular grande, 4.2a, frente a otro menor, 4.2b.
- Según el tiempo que es posible mantener esa velocidad libre de colisión:
Si bien trabajar en el espacio velocidades-tiempo permite emplear velocidades que en el plano eran evitados, es necesario tener en cuenta cuánto tiempo pueden ser mantenidos estos comandos. Por ello, cada vez que un subobjetivo definido por un obstáculo se incluye en el planificador A*, se comprueba para cuál es la primera casilla prohibida en la que aparece y se le asocia un tiempo durante el cual ese par de velocidades es posible y seguro (disminuyendo a la primera casilla prohibida un margen de frenado).

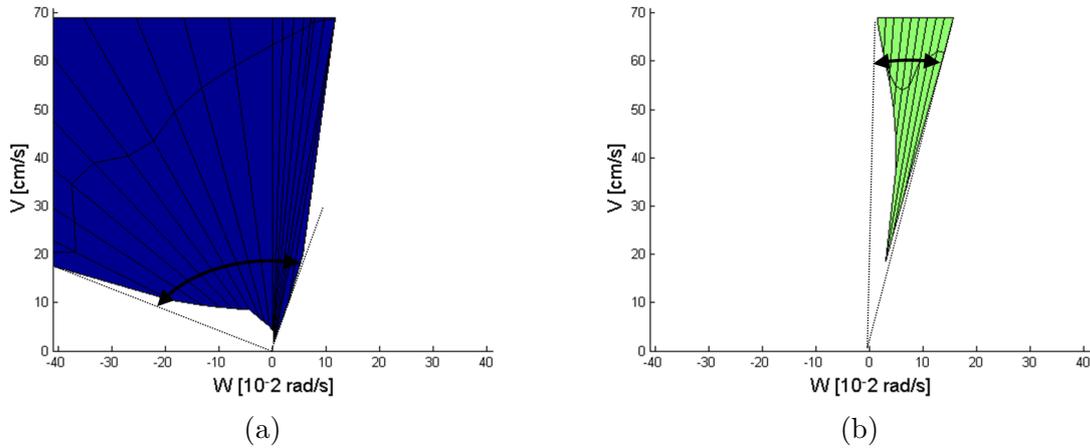


Figura 4.2: Figura comparativa de dos tamaños angulares

En esta aplicación el margen de tiempo corresponde a un margen de seguridad definido como el máximo tiempo de frenado del robot. Si se desea ampliar la seguridad del sistema, bastaría con modificar este margen. En el caso más extremo, en el que todos los subobjetivos correspondientes a obstáculos fueran eliminados, el A^* funcionaría con el único objetivo del *freemotion*.

Este parámetro permite descartar como subobjetivo las velocidades que no son alcanzables por las restricciones cinemáticas y dinámicas del robot en el tiempo asociado. Al comienzo de cada iteración, el algoritmo comprueba que cada subobjetivo sea alcanzable antes de su tiempo máximo desde el par de velocidades que lleva el robot en ese instante. Si no es así, lo descarta.

Como se ha indicado en la sección 2.3.2, para limitar el tiempo de búsqueda se define en cada iteración un techo de búsqueda temporal u horizonte temporal. Este tiempo corresponde al tiempo que le costaría llegar al objetivo en el espacio de trabajo si mantuviera la velocidad del *freemotion*. Está acotado superiormente por el tamaño máximo de la malla para evitar búsquedas infinitas o muy largas.

Una vez todos los subobjetivos correspondientes con obstáculos han sido ordenados, es necesario incluir el *freemotion*. Para decidir su prioridad de inserción, se sigue el siguiente criterio:

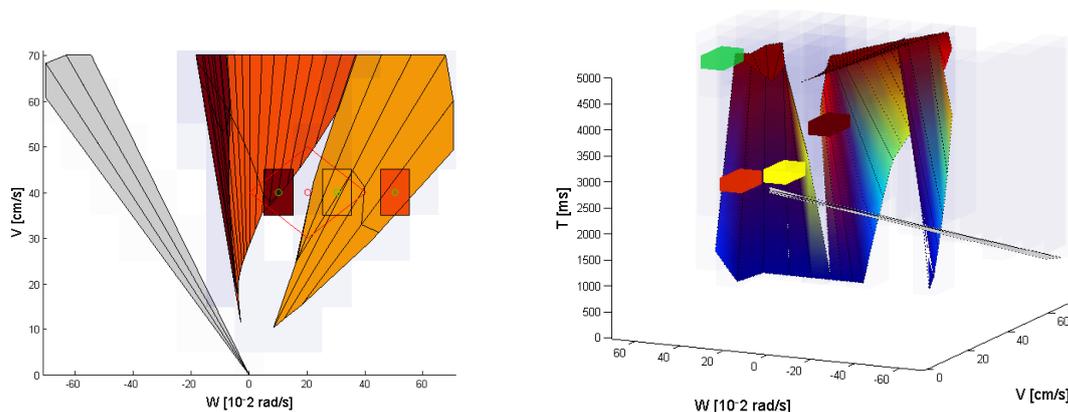
- Si las velocidades de *freemotion* son libres en el nivel del horizonte temporal, se insertará en su posición correspondiente de tiempo y se dejará de incluir subobjetivos detrás. Esto es especialmente importante en casos en los que el robot se halla muy cerca de su objetivo cartesiano, ya que la cercanía de otros robots puede indicar trayectorias de evitación, que lo alejarían del objetivo.
- En caso contrario se añade al final de la lista. Si ninguno de los subobjetivos está libre en el horizonte temporal, se añade el más cercano al de *freemotion*.

Este método busca trayectorias en las que una vez evitado el obstáculo más próximo fuera posible alcanzar la orden de velocidad que evitaba al siguiente.

En la figura 4.3a se aprecian tres superficies prohibidas correspondientes a obstáculos móviles y una (gris) correspondiente a una pared. Con rectángulos del mismo color aparecen las estrategias asociadas a cada uno de los obstáculos móviles que se ordenan priorizando el tamaño angular: naranja (evitar por detrás), amarillo (libre), granate (evitar por detrás).

La figura 4.3b muestra los objetivos finales que se introducen al A*. Se puede observar cómo todos a los subobjetivos se les ha asignado un tiempo máximo de alcance que los sitúa por debajo de cualquier superficie prohibida.

El subobjetivo amarillo coincide con el de *freemotion*, pero ninguno de los subobjetivos calculados está libre en horizonte temporal. Por ello se añade un último subobjetivo cercano al *freemotion*, que está libre en el horizonte temporal (celda verde).



(a) Subobjetivos calculados en el DVS para cada obstáculo

(b) Subobjetivos ordenados e introducidos en el DVTS

Figura 4.3: Ordenación de subobjetivos

En este apartado se ha explicado la técnica general, que se aplica a escenarios poco estructurados con obstáculos moviéndose en direcciones aleatorias como la navegación en interiores, una plaza o el cruce de una calle. Adicionalmente se consideró una escena de adelantamiento, en la que la selección de estrategias difiere.

4.2. Maniobras de adelantamiento

A diferencia de los escenarios presentados hasta ahora, la circulación de un vehículo en carriles se trata de un entorno muy estructurado. Por ello, la selección de las heurísticas de velocidad que se introducen al algoritmo de búsqueda, se realiza de un modo diferente. Básicamente se pretende avanzar a máxima velocidad siguiendo el vial mientras sea posible. Si aparece un vehículo en el mismo carril más lento se analiza la

opción de realizar un adelantamiento, reduciendo la velocidad si este no se puede llevar a cabo de forma segura.

Un adelantamiento consta básicamente de tres etapas. En primer lugar hay un cambio al carril izquierdo, luego se avanza por dicho carril hasta superar el vehículo a adelantar y finalmente se realiza un cambio al carril original. El anexo A.2 explica la generación de maniobras de cambio de carril, basada en la combinación de pares de clotoides simétricas [Montés and Tornero(2004)], figura 4.4 que permite un cambio de carril suave, manteniendo la continuidad de la curvatura de la trayectoria.

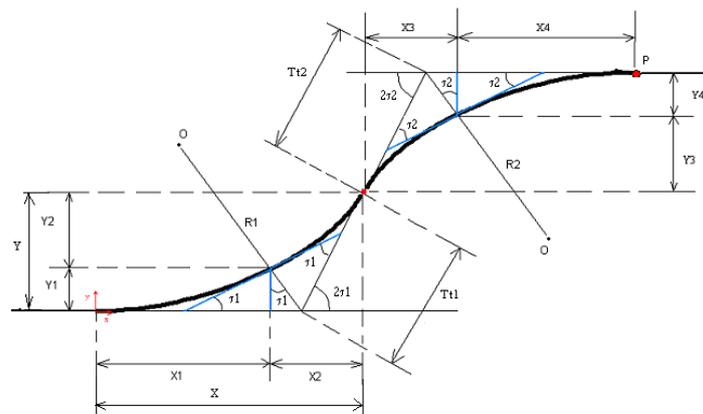


Figura 4.4: Figura obtenida del artículo [Montés and Tornero(2004)] en la que se muestra la combinación de cuatro clotoides simétricas para el cambio de carriles

Al inicializar el programa se calcula la maniobra aproximada de adelantamiento. Una vez conocido el tramo y el tiempo estimado de cambio de carril se obtienen las condiciones de las distancias mínimas entre los vehículos a adelantar y las distancias de seguridad respecto a los vehículos que vienen de frente. Durante la ejecución del programa, se identifica la situación actual según el esquema presentado en la figura 4.5.

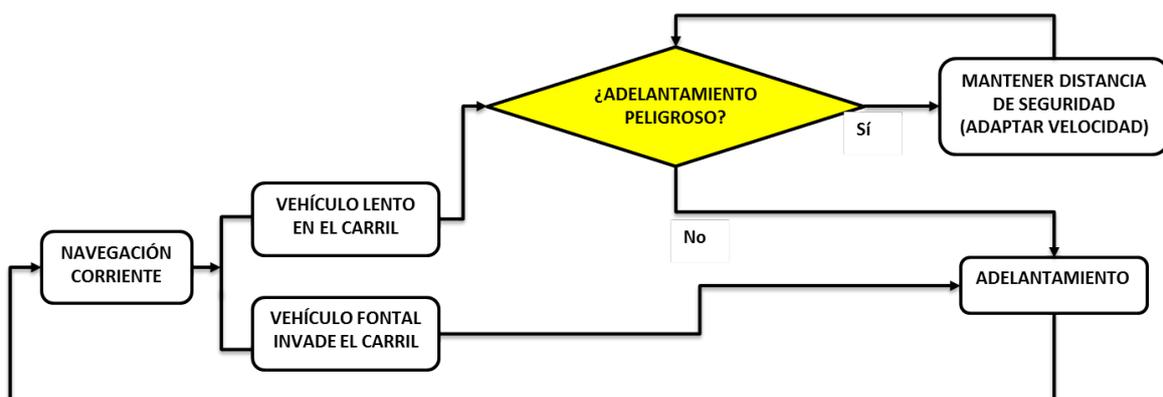


Figura 4.5: Esquema adelantamiento

Este método calcula para cada iteración un objetivo de velocidad que introduce en el A* en función de la situación en el espacio de trabajo. Esto lo diferencia del método general que seleccionaba los subobjetivos para cada obstáculo basándose en sus siluetas en el DVS. Sin embargo, en lo referente a la búsqueda del A* son similares. Una vez seleccionadas las heurísticas de velocidad, la búsqueda se lanza. Esto permite en el adelantamiento tender a las velocidades que marca el adelantamiento teórico, pero considerando que en todo momento las acciones son seguras.

La figura 4.6a representa un caso de necesario adelantar pero no posible, por lo que el robot debe adaptarse a la velocidad del vehículo lento. Así la velocidad se mantiene lineal y baja por debajo de las superficies prohibidas (el caso general hubiera desviado al robot a la derecha, generando comportamientos excesivamente oscilatorios). En la figura 4.6b el robot se mantiene paralelo al vehículo a adelantar mientras lo sobrepasa, en lugar de desviarse a la izquierda.

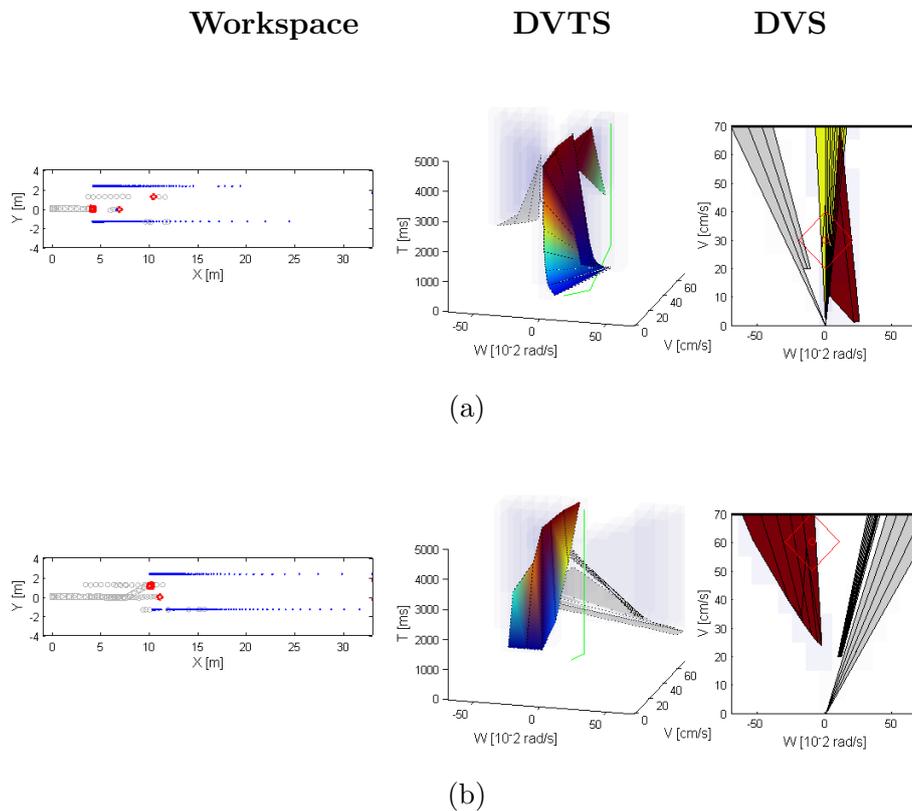


Figura 4.6: Ejemplos de la selección de estrategias en el adelantamiento

En los apartados 6.4 y 7.2 se presentan las pruebas realizadas en este tipo de situaciones, tanto en la simulación como con los robots reales. En el anexo A.2 se desarrollan las pruebas más en detalle, y se aclaran las modificaciones que fueron necesarias para realizar la prueba real.

Capítulo 5

Módulos del programa

5.1. Estructura del programa

El programa es un ejecutivo cíclico de 250 milisegundos de duración, durante los cuales se realizan todas las operaciones necesarias para planificar la trayectoria. El tiempo de refresco del láser es aproximadamente 250 milisegundos, por esta razón se ha fijado esta duración de periodo. En la siguiente figura se muestran los módulos y dependencia de las acciones realizadas tras las inicializaciones y durante cada iteración hasta la finalización del programa.

Mediante el color granate se indica que partes fueron desarrolladas en este proyecto. En color azul se representan módulos se integraron de trabajos anteriores. Los algoritmos de estimación y filtrado de la información fueron proporcionados por el departamento, al igual que el método de cálculo de los límites de velocidad que conducen a colisión. En el anexo B se amplía el funcionamiento de estos filtros. En el caso de los obstáculos móviles (naranja), se modificó la selección de estrategias del árbol de decisiones A para adaptarla y se desarrolló el método de selección de estrategias en el adelantamiento a partir del cambio de carril mediante combinaciones de clotoides [Montés and Tornero(2004)]. Los módulos en verde representan las interacciones del programa con el robot, a través de Player [PlayerStage()].

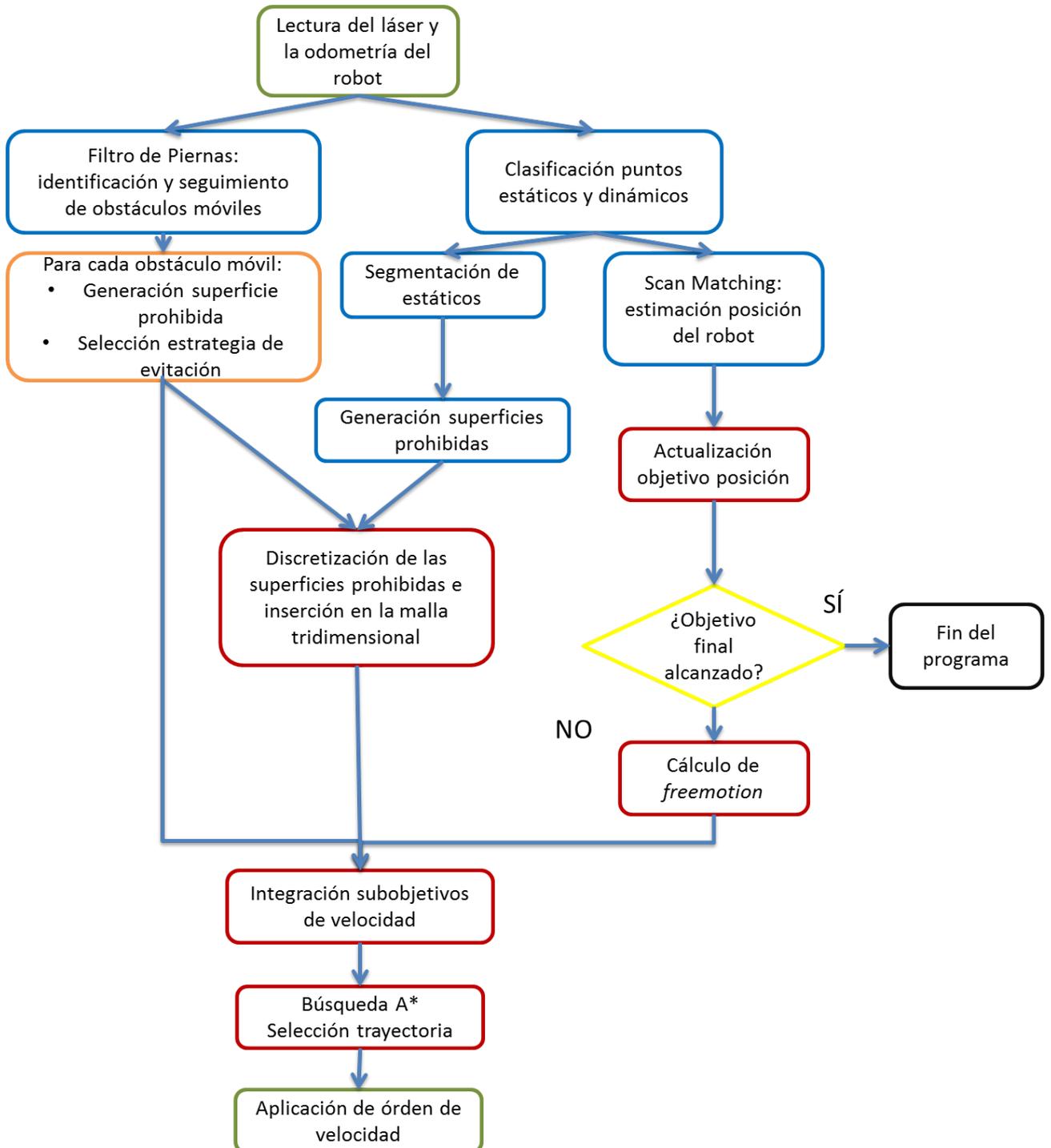


Figura 5.1: Módulos ejecutados en cada iteración del programa

Capítulo 6

Simulaciones

En este capítulo se presentan los experimentos llevados a cabo sobre la plataforma de simulación Stage. A pesar de que este programa proporciona los datos exactos de posición y velocidad de todos los robots, todas las simulaciones toman los datos a través de la percepción del láser de la simulación. A los datos del láser se les aplican los mismos tipos de filtros que se emplean en las pruebas reales para ser lo más similares posible.

Se diseñaron varios escenarios para probar la eficacia del método e intentar reproducir diferentes situaciones comunes en la vida real. En simulación se realizaron cuatro experimentos:

- Un cruce de carretera, 6.1 .
- Una situación caótica en espacio abierto, 6.2 .
- El interior de un edificio en el que un planificador global le ha indicado varios subobjetivos 6.3 .
- Un adelantamiento en una carretera, 6.4 .

Se presenta para cada experimento una descripción general del escenario y una breve descripción de las trayectorias que han resultado, junto con unas capturas del escenario para mejorar su comprensión. En el anexo C se amplían estas explicaciones con el razonamiento sobre el espacio de velocidades-tiempo.

En todas las simulaciones, el robot que está ejecutando el programa es el robot verde. Los demás se mueven con velocidades constantes.

6.1. Cruce

En este escenario, el robot debe alcanzar un objetivo con el que no se encuentra alineado inicialmente, mientras tandas de obstáculos se le cruzan todas en la misma dirección y sentidos contrarios.

Siempre y cuando es posible, pasa por delante del obstáculo móvil corrigiendo ligeramente su orientación. Al vehículo rojo central lo rodea por detrás, al igual que a los dos últimos naranjas.

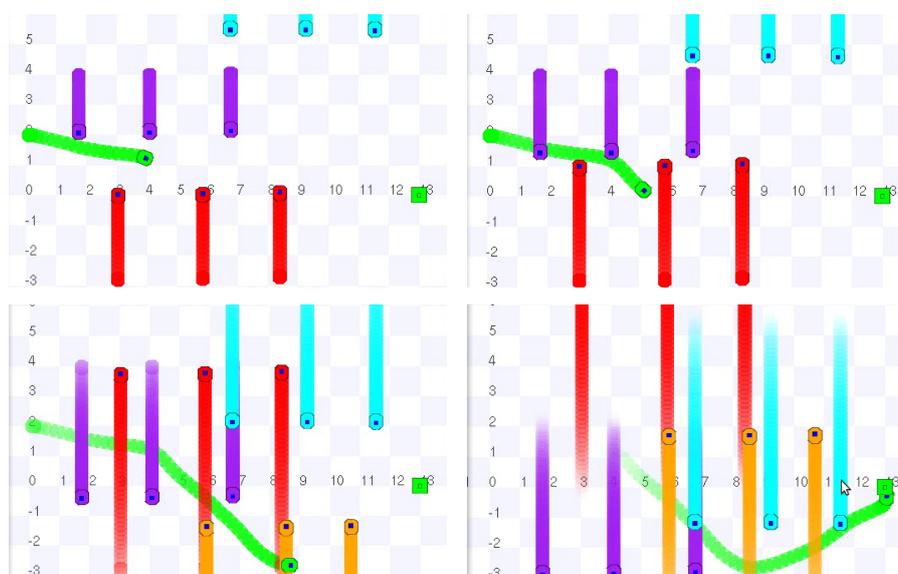


Figura 6.1: Escenas de la simulación *Cruce*

En la figura 6.2 aparece la evolución de la velocidad que ha llevado el robot, manteniéndose a altas velocidades lineales hasta alcanzar el objetivo.

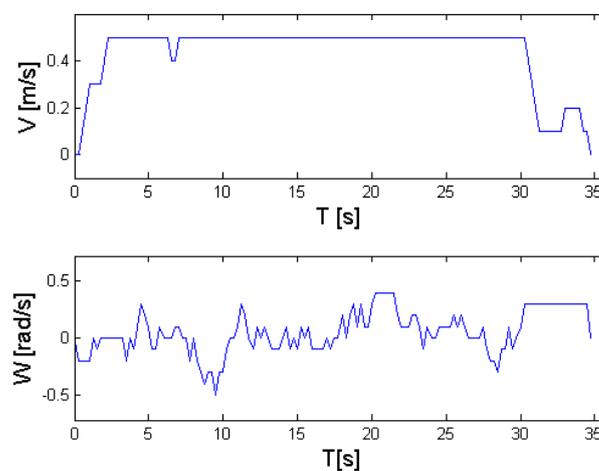


Figura 6.2: *Carriles*: Perfil de Velocidades

6.2. Escenario sin obstáculos estáticos

En la simulación denominada como *Caos* múltiples objetos móviles obstaculizan el camino hasta el objetivo. El robot (verde) debe evitarlos de la forma más adecuada posible respetando su seguridad. Los obstáculos, a diferencia del caso anterior, no llevan trayectorias puramente lineales. La opción de esperar hasta que pasen no es viable ya que todo el entorno está ocupado. En las imágenes de la figura 6.3 se muestra la evolución del escenario.



Figura 6.3: Escenas de la simulación *Caos*

El robot comienza alineándose con el obstáculo gris para evitarlo por delante. Continúa sin recuperar su orientación hacia el objetivo para también evitar por delante al obstáculo amarillo. Para evitar al obstáculo azul de nuevo realiza una maniobra de alineación. Pasa por detrás del obstáculo gris se mantiene por detrás del morado hasta tener camino libre para alcanzar su objetivo.

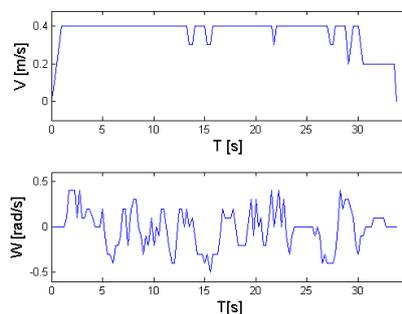


Figura 6.4: *Chaos*: Perfil de Velocidades

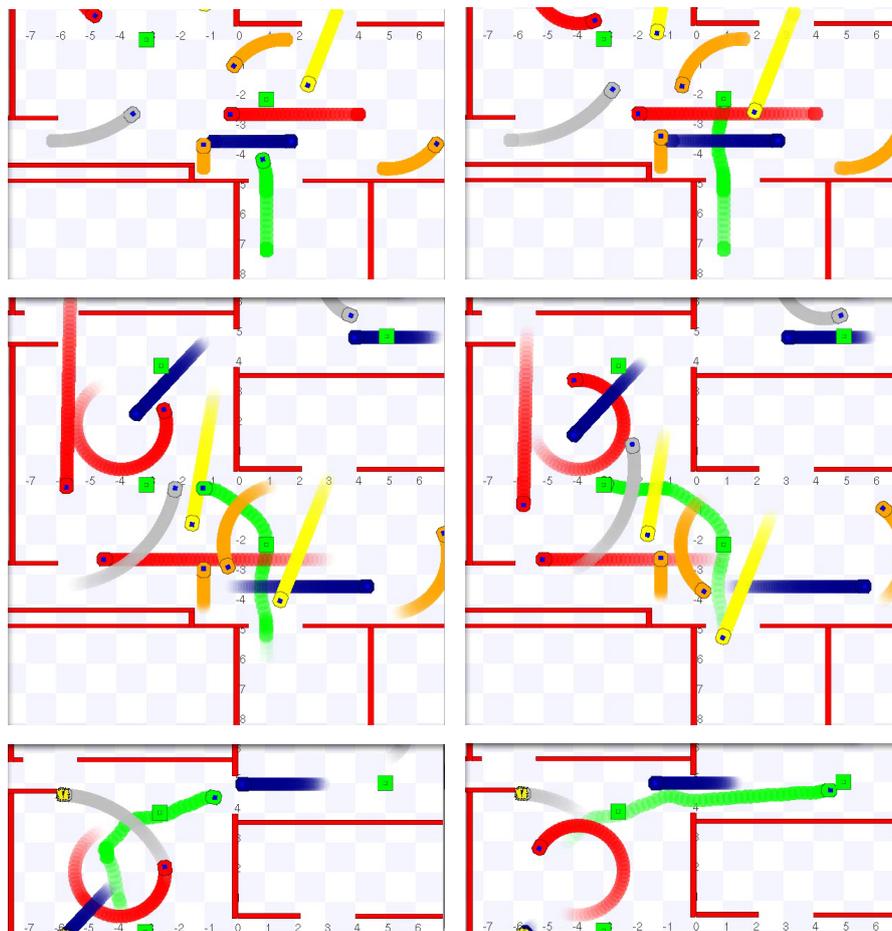
En la figura 6.4 se observa cómo la velocidad lineal se mantiene lo más alta posible, realizando giros para evitar a los obstáculos. A pesar de que alguno de los obstáculos no lleva trayectorias rectilíneas, que es una de las premisas sobre las que trabaja el modelado de los obstáculos en el DVTS, gracias a la alta frecuencia de actualización del programa no constituye un problema.

6.3. Escenario de interiores con obstáculos estáticos y móviles

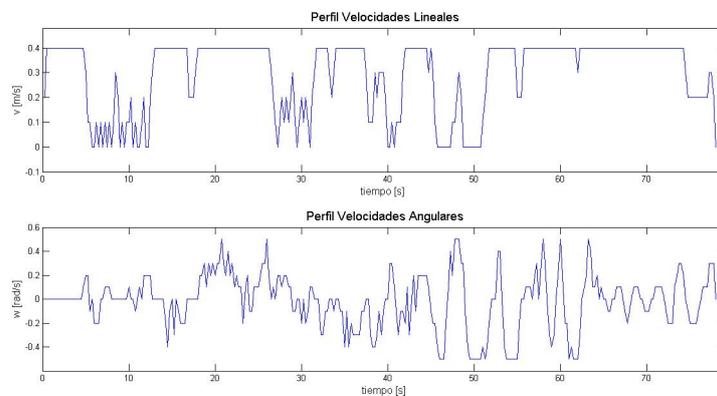
En este escenario, denominado *Interiores*, se representa la actuación del sistema en una navegación de interiores en la que un planificador global ha marcado una serie de subobjetivos de posiciones. El robot(verde) debe ir superando cuatro objetivos para llegar al destino final. En su recorrido aparecen otros objetos móviles que le dificultan la realización de la tarea. Además el escenario es un entorno interior, en el que aparecen paredes y puertas abiertas que restringen los movimientos posibles.

El robot está inicialmente situado en una sala, tras atravesar la puerta navega por un pasillo lleno de obstáculos para, finalmente, entrar en otra sala.

Tal y como muestra la evolución del escenario en la figura 6.5, el robot comienza saliendo de la primera habitación, dejando pasar al obstáculo azul. Luego realiza un leve giro a la izquierda para salir de la trayectoria del obstáculo amarillo. Tras alcanzar su primer objetivo se dirige hacia el siguiente. Pasa por detrás del naranja y del amarillo, y debe reducir su velocidad para dejar pasar al gris. Al encaminarse hacia el tercer objetivo, se encuentra con el obstáculo azul y debe frenar al no tener tiempo de realizar otro tipo de estrategia. Con el obstáculo gris intenta una maniobra de alineación, que no consigue realizar, lo que le obliga a dejar pasar y orientarse hacia el objetivo de nuevo. Para terminar su recorrido se encuentra con el robot azul que le restringe el paso por la puerta. Se mantiene entre el obstáculo móvil y las paredes para pasar por la puerta y finalizar su recorrido.

Figura 6.5: Escenas de la simulación *Interiores*

En este escenario las velocidades no se han mantenido máximas ya que han aparecido varios casos en los que ha sido inevitable la estrategia de reducir velocidad para dejar pasar al obstáculo.

Figura 6.6: *Interiores*: Perfil de Velocidades

6.4. Adelantamiento

Como ya se ha comentado en el apartado 4.2, la navegación en carreteras y carriles llevaba consigo un sistema diferente para elegir las estrategias.

El escenario que aparece a continuación consta del robot controlado, un obstáculo lento en el carril del robot y tres obstáculos rápidos que vienen de frente. El robot espera a que los dos primeros terminen de pasar, manteniéndose a una velocidad baja mientras espera el momento adecuado para realizar el adelantamiento, figuras 6.8a. En la figura 6.7 se puede observar esta reducción de velocidad, (1).

Hay otro robot que viene de frente pero está lo suficientemente lejos para permitir el adelantamiento. Antes de comenzar el adelantamiento el robot acelera hasta su velocidad máxima. A continuación cambia al carril izquierdo, en la velocidad angular de la figura 6.7 se producen dos picos altos de velocidad angular (2) que corresponden a los puntos de concatenación entre clotoides. Luego el robot se mantiene a velocidades mas o menos constante hasta rebasar al vehículo lento y una vez lo rebasa vuelve al carril inicial(3). La figura 6.8b muestra varias tomas del adelantamiento.

Una vez completada la maniobra sigue circulando normalmente hasta su destino.

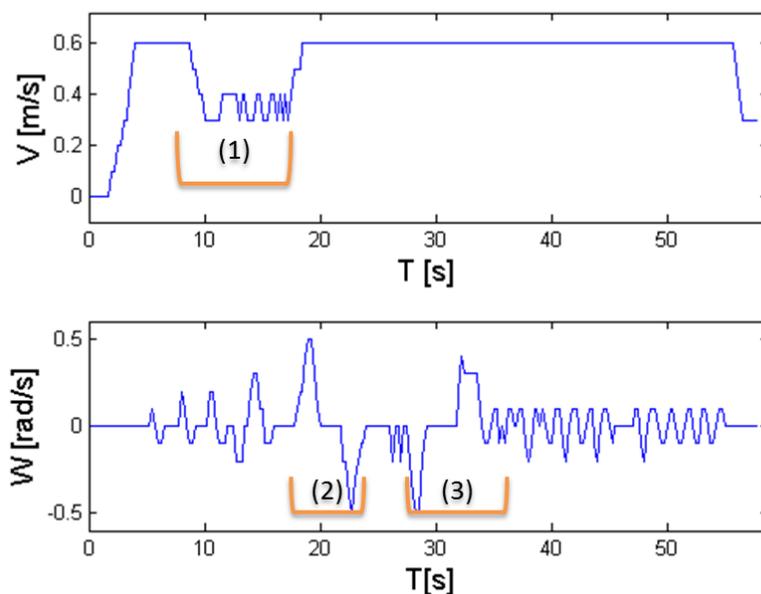
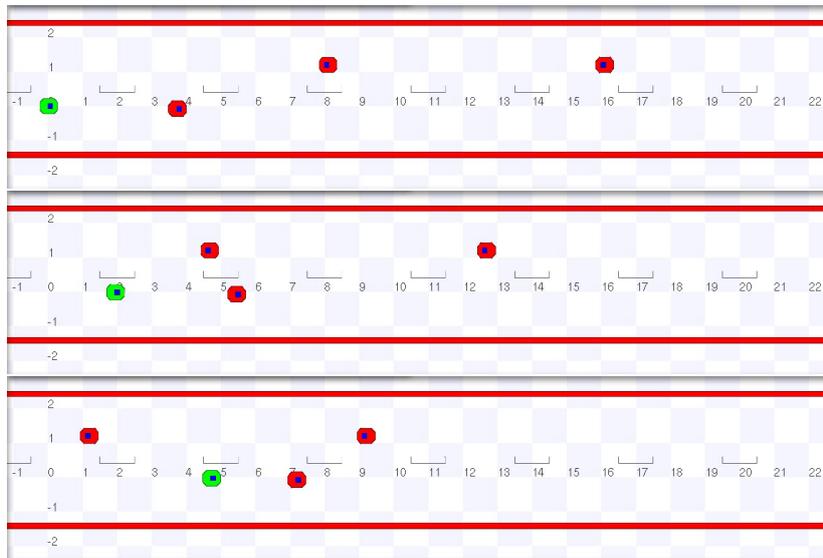
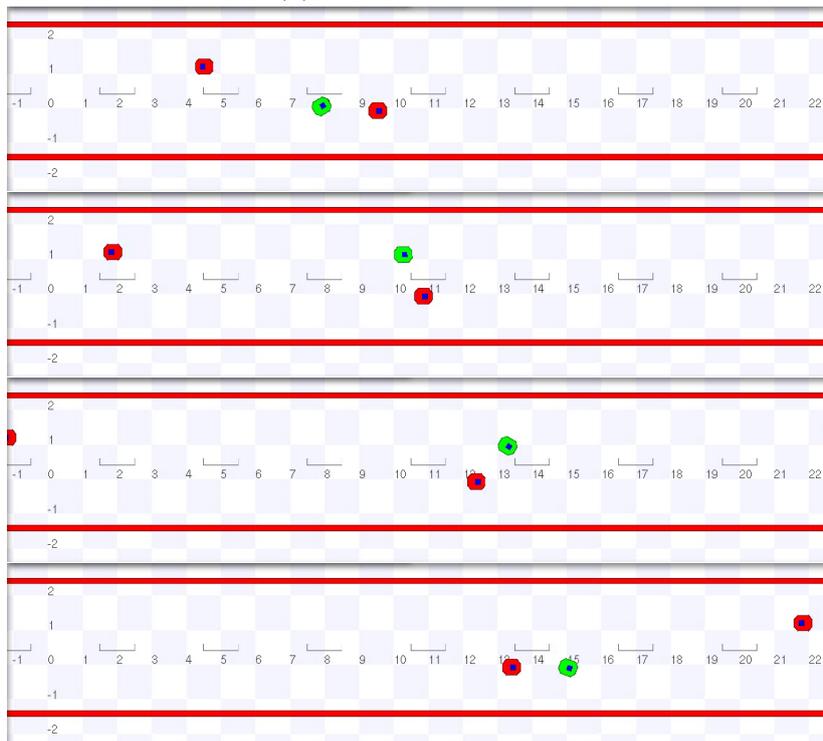


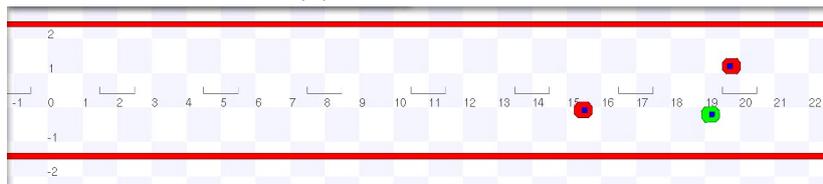
Figura 6.7: *Adelantamiento*: Perfil de velocidades



(a) Espera a adelantar



(b) Adelantamiento



(c) Maniobra finalizada

Figura 6.8: *Adelantamiento*: Evolución

Capítulo 7

Experimentos

Una vez realizados los experimentos en simulación, se procedió a montar los experimentos con los robots Pioneer reales. Para realizar estos experimentos fue necesario incluir un filtro para los obstáculos dinámicos, el filtro de piernas (veáse anexo B), que fue facilitado por el departamento.

Se decidió llevar a cabo dos experimentos, cada uno de ellos presenta alguna variación respecto a sus homólogos de simulación:

- *Hall*

Se desarrolla en un entorno interior. Aparecen otros robots y personas.

- *Adelantamiento*

Además de un vehículo lento al que debe adelantar junto con otro vehículo en el carril izquierdo que le obliga a retrasar el adelantamiento, en este experimento aparece un tercer vehículo que invade su carril y le obliga a realizar una maniobra de emergencia.

7.1. Hall

Este experimento se llevó a cabo en la entrada del edificio de I+D. Aparecen seis obstáculos móviles de los cuales dos son robots y cuatro personas. Se ha simulado un planificador global que le indica dos puntos vía hasta el objetivo.

El robot parte de su posición inicial y le aparece el primer obstáculo, un robot, al que comienza a evitar por detrás y luego deja pasar. Reduce su velocidad al llegar al primer punto vía y se dirige hacia el objetivo siguiente.

Al comienzo del segundo tramo, se encuentra con un peatón que se dirige en diagonal hacia él, realiza un giro a izquierdas para dejarle pasar y recupera su orientación hacia el objetivo. Otro peatón entra por la puerta del edificio y se cruza por delante en perpendicular, realiza un leve giro a la derecha para proseguir su camino. Un tercer peatón le aparece por su lateral. Este peatón describe una trayectoria curva y se le cruza de derecha a izquierda, el robot realiza una maniobra de giro la derecha para pasar por detrás de él de forma segura y alcanza el segundo punto vía.

Un quinto obstáculo móvil, esta vez otro robot, avanza en línea recta obstaculizando el camino directo que conduce a su objetivo final. Mediante un giro a izquierdas se aparta de la trayectoria del otro móvil, recuperando su posición. Aparece un sexto obstáculo, un peatón. Su presencia retrasa la recuperación de la orientación directa al objetivo. Finalmente concluye su recorrido.

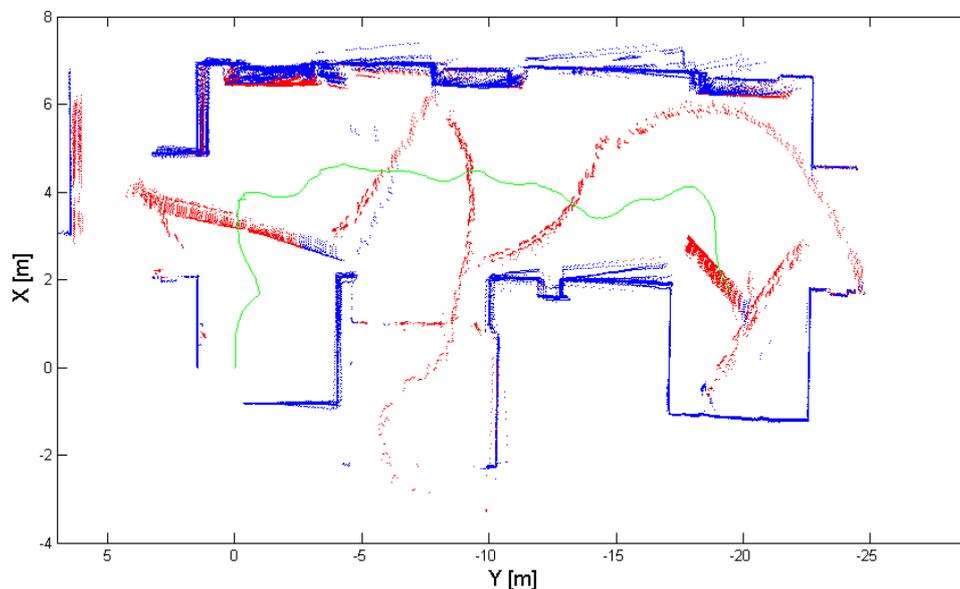
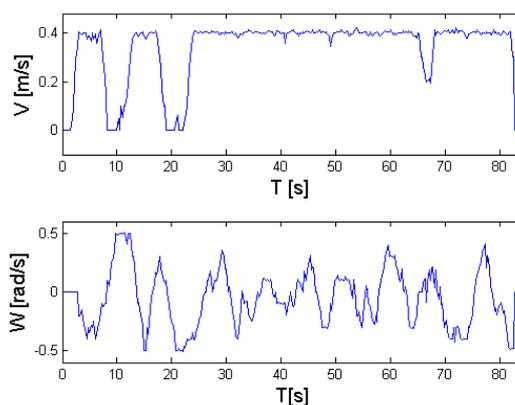


Figura 7.1: *Hall*: Escenario

La figura anterior reproduce la identificación de puntos estáticos (azules) y dinámicos (rojos) del robot a lo largo del experimento y la trayectoria que ha descrito (línea verde).

Figura 7.2: *Hall*: Perfil de Velocidades

En la figura 7.4 se observa cómo el robot consigue mantener la velocidad lineal alta en general. El primer descenso de velocidad corresponde con dejar pasar al primer obstáculo, los otros dos se realizan al pasar por los puntos vía.

Figura 7.3: Tomas de la prueba del *Hall*

Con este experimento, se ha probado el método completo en una situación realista. Mediante el filtro de piernas se ha logrado identificar de forma adecuada a las personas. El robot ha sido capaz de tratar con el entorno completo y desarrollar una trayectoria adecuada y segura.

7.2. Adelantamiento

En este experimento se lleva a cabo un adelantamiento y una maniobra de evitación ante una invasión frontal del carril propio. En la figura 7.4 aparece la evolución de las velocidades del robot. En la figura 7.6 se muestran algunas capturas de pantalla del experimento.

Comienza identificando un vehículo más lento en el carril derecho al que debe adelantar, pero como por el carril izquierdo viene otro de frente y está bastante cerca debe esperar a realizar su maniobra reduciendo la velocidad. Esto se aprecia claramente en el momento en el que la velocidad lineal se reduce, en la gráfica 7.4 marca (1).

Cuando el vehículo del carril izquierdo ha pasado comienza el adelantamiento, (2) primero un giro a izquierdas y luego a derechas. Una vez sobrepasa al vehículo lento y estima que ya lo ha superado suficientemente vuelve a su carril, (3) giro a derechas y luego a izquierdas, y corrige su trayectoria para terminar bien alineado en su carril.

Al final del tramo aparece un vehículo que le viene de frente en su propio carril, lo que le obliga a pasar al carril izquierdo y volver una vez el vehículo hostil ha pasado (4).

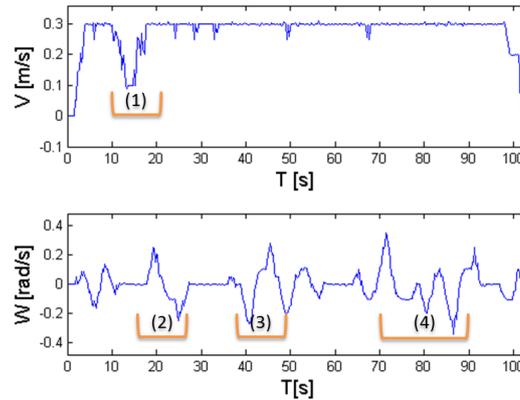


Figura 7.4: *Adelantamiento real*: Perfil de Velocidades

La figura 7.5 muestra el escenario completo, siendo los puntos azules puntos clasificados como estáticos los círculos rojos los obstáculos identificados y la línea verde la trayectoria seguida por el robot a lo largo del experimento.

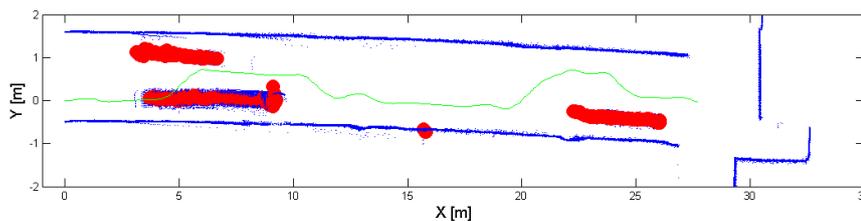


Figura 7.5: *Adelantamiento real*: Escenario

Se puede apreciar una ligera deriva en la autolocalización de la posición en Y. Por otra parte se observa que los puntos que forman el obstáculo lento aparecen clasificados en muchas ocasiones como estáticos. Gracias al empleo del filtro de piernas en lugar de un clasificador a partir de los puntos dinámicos (anexo B), esto no represente un grave problema.

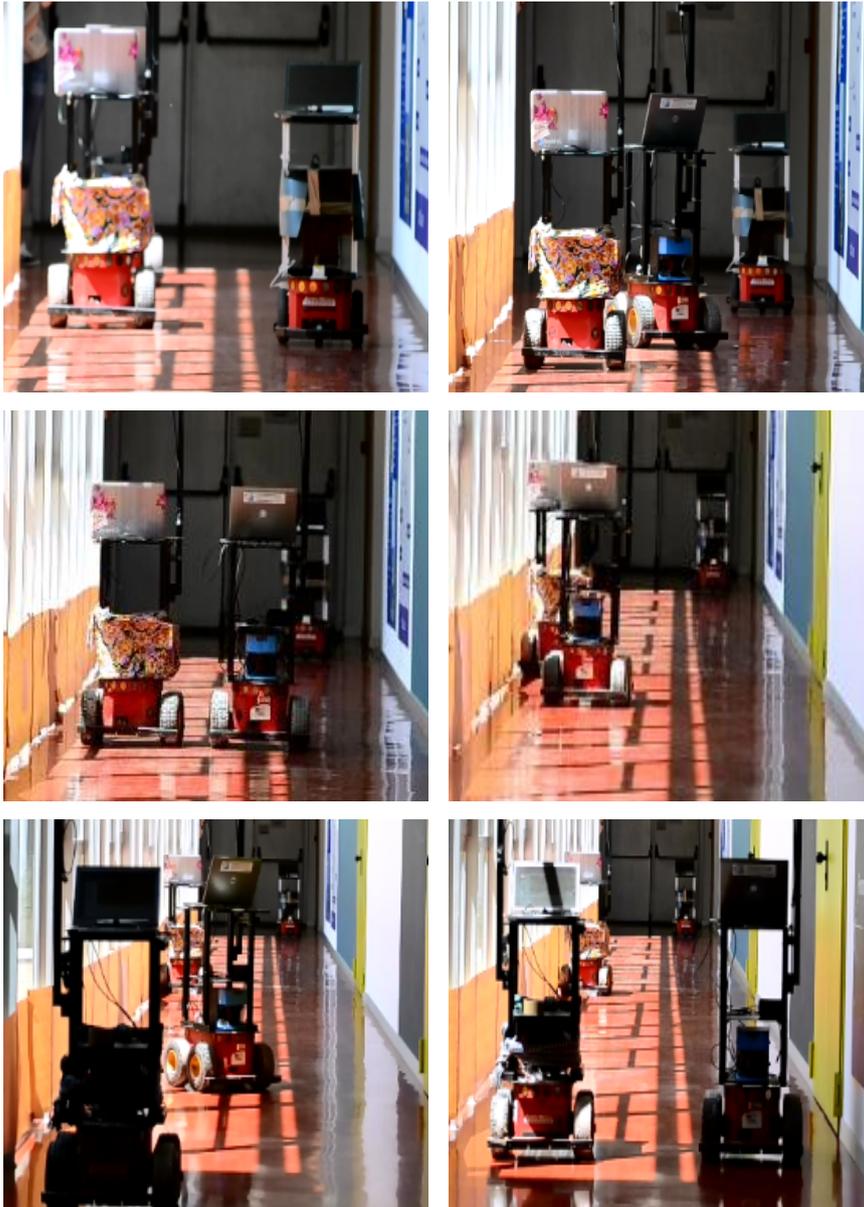


Figura 7.6: Tomas de la prueba de *Adelantamiento real*

Capítulo 8

Conclusiones y trabajo futuro

8.1. Conclusiones

Este proyecto presenta un método robusto de planificación de trayectorias óptimas en tiempo y seguridad. Para ello se trabaja sobre un espacio discreto de velocidades-tiempo, que permite representar de forma explícita las restricciones dinámicas del robot y el tiempo.

Las principales aportaciones de este trabajo son dos:

- Trabajar con la componente de tiempo de forma explícita para realizar la planificación permite realizar maniobras que no eran permitidas en trabajos anteriores que consideraban la proyección bidimensional.
- Se han integrado de un modo natural la seguridad, la rapidez y la selección de estrategias.

A pesar del alto coste computacional que supone trabajar con la componente tiempo, se ha alcanzado un compromiso adecuado entre el nivel de discretización, el horizonte temporal y el tiempo de cómputo requerido que permite ejecutar el programa en tiempo real.

La trayectoria calculada es una trayectoria óptima según la función de costes desarrollada en este trabajo, la cual minimiza el tiempo de alcance de los objetivos a la vez que asegura la integridad del robot. Esto se logra gracias a la integración en la función de costes de heurísticas de evitación de obstáculos, de cercanía al objetivo y de seguridad, además de considerar de forma explícita el tiempo. Realizar una planificación a varios niveles de tiempo de la trayectoria también aumenta la robustez del método, ya que permite aplicar órdenes de velocidad calculados en iteraciones anteriores en caso de no poder completar una búsqueda.

Adicionalmente, seleccionar de forma independiente las estrategias de evitación de cada obstáculo e integrarlas al final permite tratar un elevado número de obstáculos móviles con menos restricciones que con el método anterior que fusionaba las proyecciones de los obstáculos y los trataba como uno solo.

El método fue satisfactoriamente validado en varios escenarios diferentes, tanto reales como de simulación. Los resultados muestran que se ha conseguido implementar un método general y seguro.

8.2. Limitaciones y Trabajo Futuro

La realización de los experimentos reales supuso la aparición de varios problemas:

- La clasificación de los obstáculos móviles a través de las agrupaciones de puntos dinámicos no permitía identificar algunas situaciones concretas, como las de un obstáculo más lento que se desplaza en la misma dirección y sentido.

Para solventarlo se empleó el filtro de identificación y seguimiento de obstáculos móviles a través de patrones tipo piernas, que en lugar de emplear la clasificación de puntos estáticos y dinámicos buscaba directamente el patrón de discontinuidad. El empleo de este filtro además permitió realizar un experimento con personas reales, lo que de otro modo no hubiera sido posible.

- En los experimentos reales, especialmente en el caso del adelantamiento, se observó que al realizar giros bruscos el algoritmo de autolocalización de Scan Matching perdía parte de su eficacia, lo que causaba derivas en la estimación de la posición del robot respecto a la posición real.
- Otro posible punto de mejora es el tratamiento de obstáculos estáticos. La segmentación de los puntos observados en cada iteración por el láser puede no ser suficiente para representar el conjunto de los obstáculos estáticos que dificultan la navegación.

Una futura línea de trabajo podría completar este método de planificación con un método de autolocalización y generación de mapas tipo SLAM (*Simultaneous Localization And Mapping*, [Thrun and Fox(2005)]). Mientras los métodos SLAM permiten un mejor tratamiento de los obstáculos estáticos, el DVTS permite trabajar mejor con obstáculos móviles al plantear en el tiempo las estrategias de evitación. La planificación de caminos sobre el espacio de trabajo es una línea bastante tratada en la robótica móvil. Emplear un método SLAM podría mejorar además el problema de autolocalización.

Una posible solución para integrar ambos métodos y aprovechar las principales ventajas de cada uno de ellos, sería a través de la función de costes. En la función de costes presentada en este proyecto se ha introducido ya un término relativo al espacio de trabajo (h_{dist}), en el que se realiza una estimación de la posición para la celda a expandir, pero esta estimación solo era empleada para realizar una estimación del tiempo desde esa posición al objetivo. Este término se podría ampliar o redefinir comparando la posición estimada con el coste equivalente en el espacio cartesiano empleando por ejemplo el método de potencial [Latombe(1991)].

Apéndice A

Selección de estrategias

A.1. El árbol de Decisión

[Lorente(2011)] trabaja sobre una proyección bidimensional del espacio de velocidades-tiempo. En este trabajo se realizó la identificación de la siluetas de la superficie prohibida para un objeto y su estrategia de evitación más adecuada correspondiente. El árbol de decisiones se inserta en la figura A.1.

Para trabajar con varios obstáculos se extrapoló el método fusionando las siluetas y tomando la decisión sobre la silueta conjunta resultante. El trabajo presentado en esta memoria permite mantener la independencia de las estrategias de evitación para cada uno de los obstáculos y realizar búsquedas por debajo de las superficies prohibidas lejanas.

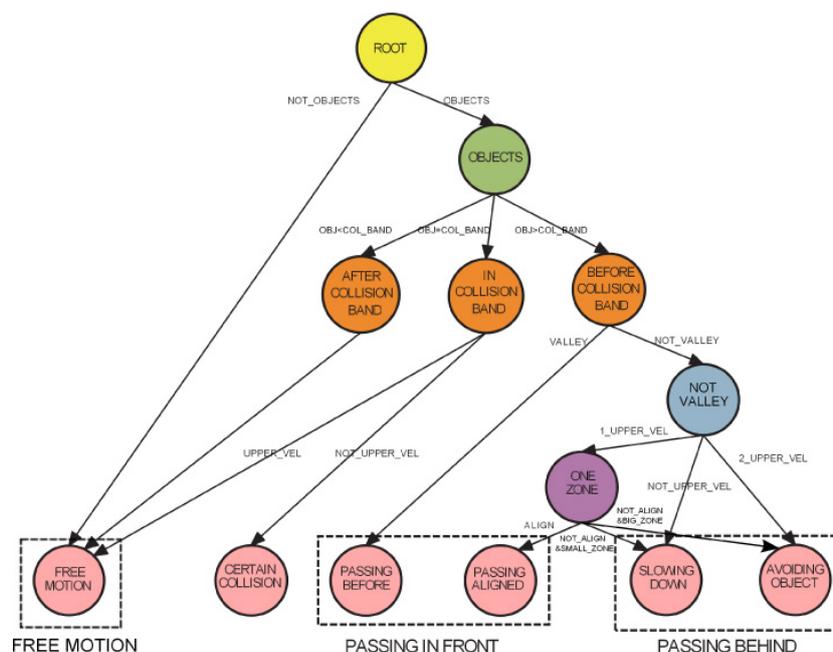


Figura A.1: Esquema del Árbol de Toma de Decisiones

A.2. Adelantamiento

A.2.1. Clotoides en el cambio de carril

En [Montés and Tornero(2004)] se presentaba la combinación de pares de clotoides simétricas para la generación de maniobras en el cambio de carriles de un vehículo. Este artículo presentaba un método para generar una serie de puntos de paso para completar el cambio de carril mediante cuatro clotoides simétricas.

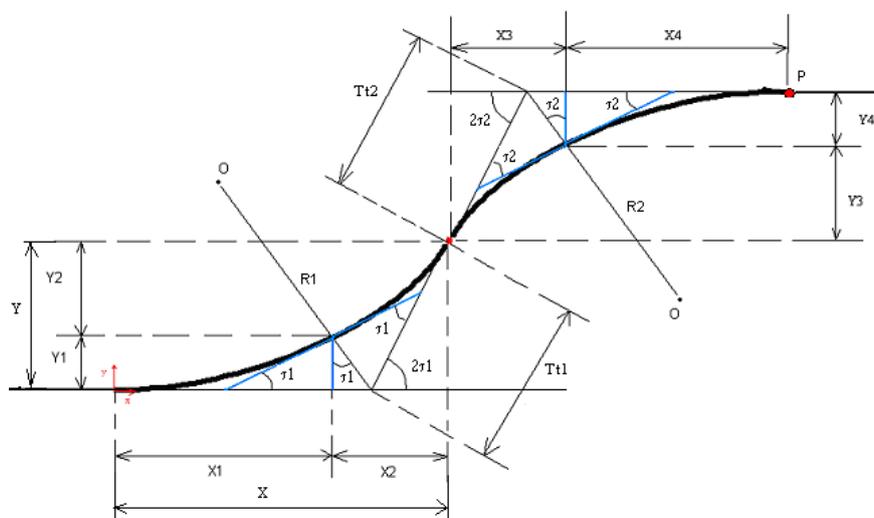


Figura A.2: Figura obtenida del artículo [Montés and Tornero(2004)] en la que se muestra la combinación de cuatro clotoides simétricas para el cambio de carriles

En la figura A.2 se observa el cambio al carril izquierdo. La primera clotoide comienza con radio infinito (trayectoria rectilínea) realizando la curvatura en sentido levógiro va disminuyendo su radio hasta alcanzar el radio mínimo en el punto (x_1, y_1) . A partir de ahí comienza la segunda clotoide que va disminuyendo su curvatura hasta alcanzar de nuevo un radio infinito, punto (x_2, y_2) . Aquí comenzaría la tercera clotoide, que disminuye el radio realizando una curva en sentido destrógiro hasta alcanzar el radio mínimo para concatenar con la cuarta y última clotoide que va aumentando el radio de curvatura hasta el infinito, quedando dejando el móvil en la misma dirección y sentido que al comenzar la maniobra. Para realizar un cambio al carril derecho se realiza una maniobra similar comenzando con sentido destrógiro y después levógiro.

Sin embargo al introducir las restricciones de aceleraciones y velocidades máximas y al trabajar con variaciones de velocidad concretas, resulta que no es siempre posible realizar el cambio de carril completo respetando la anchura de carril deseada. No es posible definir la trayectoria a velocidades máximas de forma genérica solamente combinando cuatro clotoides simétricas.

Por ello se adapta en método de cambio de carril con cuatro clotoides añadiendo, si es necesario, a cada cambio de carril un tramo rectilíneo. Este tramo se añade entre

los dos dos pares de clotoides simétricas en el punto (x2,y2) de la figura A.2. Para conseguir alcanzar los radios de curvatura deseados la velocidad debe ser puramente lineal en este punto. El móvil se encuentra con velocidad angular nula y una inclinación τ por lo que simplemente se prolonga el recorrido a esa velocidad lineal para cubrir la anchura de carril necesaria.

Para la generación de estos tramos se comprueba el cumplimiento de una serie de condiciones como respetar el radio mínimo (definido por la máxima velocidad angular y la máxima velocidad lineal) y los ángulos de concatenación de clotoides máximos, que son función de la velocidad máxima lineal y la variación de velocidad angular por unidad de tiempo. Para calcular la trayectoria se divide cada clotoide en varios pasos definidos por la aceleración del robot. Se aproxima cada una de las clotoides por tramos de circunferencias que van variando su radio con aceleración constante.

A.2.2. Maniobra de adelantamiento

El cálculo aproximado del cambio de carril aproximando las clotoides por circunferencias da una primera estimación de la distancia necesaria para cambiar de carril \hat{d}_{carril} y del tiempo empleado en ello \hat{t}_{carril} . Con estas variables y las percepciones de los obstáculos móviles se establecen las condiciones de comenzar o no un adelantamiento. Las principales variables de decisión aparecen explicadas en la figura siguiente: A.3

//distancia de aceleración. vMax es la máxima velocidad permitida en el vial y coincide con la máxima velocidad permitida al robot

$$d_{ac} = \text{ceil}\left(\frac{vMax - v_{vehículo}}{\text{maxAcLinealPeriodo}}\right) * \text{periodo} * vMax$$

//parámetro de que indica que es necesario adelantar a un vehículo que está en el carril derecho circulando en el mismo sentido

$$d_{necesario} = \hat{d}_{carril} + d_{ac}$$

//tiempo estimado de la maniobra completa de adelantamiento

$$\hat{t}_{adelantamiento} = 2 * \hat{t}_{carril} + 2 * \phi_{robot} * vRel_{robot-vehículo}$$

//distancia de seguridad de los vehículos que vienen de frente en el carril izquierdo, si es menor el adelantamiento no puede comenzar incluye un factor de seguridad f_s de 1.5

$$d_{frontal} = \hat{d}_{adelantamiento} + f_s * vMax * \hat{t}_{adelantamiento}$$

//parámetro de que indica que es necesario adelantar a un vehículo que circula en sentido contrario y ha invadido el carril derecho

$$d_{necesarioFrontal} = \hat{d}_{carril} + vRel_{robot-vehículo} * \hat{t}_{carril}$$

Figura A.3: Distancias y condiciones de seguridad durante la maniobra adelantamiento

Si se identifica un obstáculo más lento en el carril de la derecha circulando en el mismo sentido, el robot se encuentra a una distancia inferior a la distancia de adelantamiento $d_{necesario}$ y no hay ningún vehículo en el carril izquierdo que inclumpla

la distancia frontal $d_{frontal}$ (da tiempo a realizar la maniobra completa de adelantamiento), se comienza a adelantar. Si por el contrario aparece un vehículo en el carril izquierdo que no permite realizar el adelantamiento, el robot se adaptará a la velocidad del vehículo que tiene delante manteniendo en todo momento una distancia de seguridad hasta que sea posible adelantar.

Si se identifica un vehículo en el carril derecho mientras el robot está circulando por él, que continúa acercándose peligrosamente a él, el robot iniciará automáticamente una maniobra de cambio al carril izquierdo para evitar la colisión inminente. Esta maniobra puede parecer peligrosa e innecesaria en el caso de conducción estándar, pero se realiza en última instancia (una vez se llega al límite para realizarla $d_{necesarioFrontal}$). Se introdujo ya que aunque el propio espacio de búsqueda enviaba al vehículo al otro carril para evitar la colisión frontal, esta maniobra era demasiado tardía y brusca, por lo que le enviaba contra la pared izquierda no dándole tiempo a recuperar la orientación adecuada.

Para hacer el cambio de carril se partía de la combinación de cuatro clotoides y un tramo rectilíneo. El algoritmo identifica la subetapa del cambio del carril en el que se encuentra y envía la estrategia de velocidad al algoritmo A*. Así durante la primera etapa busca velocidad angular máxima a izquierdas, pasa a velocidad angular nula que mantiene hasta finalizar el tramo recto de concatenación después velocidad angular máxima a derechas y de nuevo velocidad angular nula, manteniendo la velocidad lineal máxima durante todo el proceso. En este punto, el robot se encontraría en el carril contrario y perfectamente alineado. Mantiene la velocidad lineal máxima hasta sobrepasar al obstáculo y comienza el cambio al carril derecho. Las semillas son procesadas por el A*, por lo que si ha ocurrido algún imprevisto (la rueda desliza, hay cambios de velocidad del robot al que se adelanta y riesgo de alcanzarlo, se acerca demasiado a los límites de los carriles), el término de seguridad que penalizaba velocidades actúa.

Para asegurar el seguimiento de la maniobra ideal, en las pruebas reales se añaden condiciones para asegurar alcanzar y no sobrepasar los ángulos de giros necesarios y corregir la alineación durante los tramos que deberían ser rectilíneos.

Apéndice B

Filtros y estimadores

El robot recibe información del entorno a través de dos tipos de sensores, los de odometría propios y el láser. La odometría informa de la estimación a partir del número de revoluciones relativas de las ruedas de la posición y velocidad propias del móvil.

Mediante el láser el robot recibe además la información de su entorno. Un barrido de 180° con resolución de 0.5° le devuelve un valor de distancia a la que se encuentra un obstáculo. Estos valores se procesan mediante diferentes filtros y se convierten en la información de obstáculos dinámicos y estáticos que se vuelcan en el espacio de búsqueda DVTS. El principal estimador empleado es el filtro de Kalman, cuyo funcionamiento general se describe brevemente en la siguiente sección, además se hace un breve resumen de los tres filtros que se emplean en el proyecto y utilizan el filtro de Kalman.

El otro estimador que se emplea para mejorar los datos es el Scan Matching, que a través de la información de la odometría y de la generación de un mapa con los puntos clasificados como estáticos devuelve una mejor estimación de la posición del robot que la simplemente proporcionada por la odometría.

B.1. Filtro de Kalman extendido

En esta sección se presenta el concepto general del filtro de Kalman extendido, que es el empleado por alguno de los filtros que se explican en la sección siguiente.

El filtro de Kalman extendido presenta un conjunto de ecuaciones, que linealizando el sistema en torno a la media y la covarianza a actuales, proporciona una estimación del sistema.

El sistema tiene un vector de estados $x(t)$ tal que

$$x(t+1) = f(x(t), u(t)) + \omega(t)$$

del que se puede obtener una medición $z(t)$

$$z(t) = h(x(t)) + \nu(t)$$

Donde $\omega(t)$ y $\nu(t)$ representan un ruido gaussiano de media cero del proceso y la medida, con matrices de covarianza $Q(t)$ y $R(t)$. Las funciones $f(x(t), u(t))$ y $h(x(t))$ son funciones no lineales que relacionan el estado en el instante t con el estado en el instante $t+1$ y la medición en el instante t con el estado en ese mismo instante.

La predicción *a priori* del estado en el instante actual $\hat{x}(t)$ queda:

$$\hat{x}(t)^- = f(\hat{x}(t-1), u(t-1))$$

con una covarianza *a priori* del error $P(t)$:

$$P(t)^- = A(t-1) * P(t-1) * A(t-1)^T + W(t-1) * Q(t-1) * W(t-1)^T$$

siendo $A(t)$ el jacobiano de f respecto al estado y $W(t)$ la matriz de derivadas parciales de f respecto al ruido ω .

La matriz de ganancia de Kalman $K(t)$ se define como:

$$K(t) = P(t)^- * H(t)^T * (H(t) * P(t)^- * H(t)^T + V(t) * R(t) * V(t)^T)^{-1}$$

donde $H(t)$ es el jacobiano de h respecto al estado y $V(t)$ es la matriz de derivadas parciales de h respecto al ruido de medida ν

La estimación del estado *a posteriori* y la covarianza del error finalmente quedan:

$$\hat{x}(t) = \hat{x}(t)^- + K(t) * (d(t) - h(\hat{x}(t)^-))$$

$$P(t) = (I - K(t) * H(t)) * P(t)^-$$

Si $R(t)$ se aproxima a cero, se puede confiar en la medida, mientras si lo hace $P(t)^-$ se confía más en la estimación.

B.2. Estimadores

A continuación se describen los principales filtros y estimadores empleados en el programa y proporcionados por el grupo de Robótica.

B.2.1. Clasificación de puntos

Conforme el robot avanza se genera un mapa robocéntrico que se emplea para mejorar la autolocalización y para realizar la clasificación en estáticos, dinámicos o desconocidos de los puntos percibidos por el láser. [Montesano and Montano(2005b)]

Mediante una malla de probabilidad centrada en el robot, se calcula comparando la percepción y la predicción sobre el estado anterior, la probabilidad de que ese punto sea estático. Si esa probabilidad es alta se clasifica como estático, si es baja como dinámico y si no está dentro de ninguno de los dos umbrales, como desconocido.

Además, los puntos dinámicos se introducen como entradas en un filtro de Kalman, que realiza su seguimiento individual.

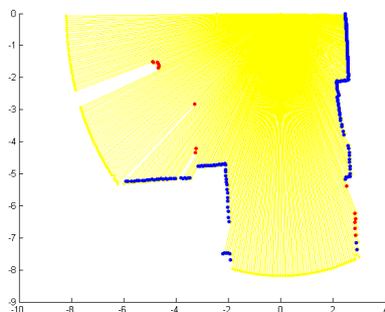


Figura B.1: Láser: clasificación de puntos estáticos (rojos) y dinámicos (azules)

B.2.2. Autolocalización

■ Scan Matching

[Montesano and Montano(2005a)], con los puntos clasificados como estáticos se realiza una mejora en la estimación de la posición del robot, comparando los desplazamientos de los puntos estáticos entre dos medidas consecutivas minimizando el error cuadrático medio.

B.2.3. Obstáculos estáticos

■ Segmentación

Los puntos estáticos se agrupan generando los segmentos que se emplearán para definir los límites de velocidad de los obstáculos estáticos. La agrupación de los puntos estáticos se basa en la distancia mínima entre dos puntos consecutivos, [D. Castro and Ruano(2002)]. Si los segmentos generados constan de más de cuatro puntos se tratan, eliminando espúreos. Posteriormente los segmentos se engordan con el radio del robot.

En la figura siguiente se observa a la izquierda el espacio de trabajo y cómo los segmentos han sido engordados y a la derecha su equivalente en la proyección de la superficie del espacio de velocidades.

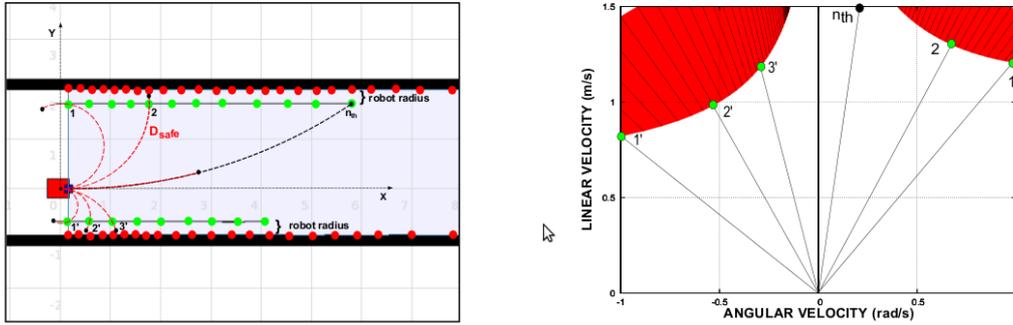


Figura B.2: Figura obtenida de [Lorente(2011)] en la que se muestra el engorde de los segmentos para el cálculo de las superficies estáticas

B.2.4. Obstáculos móviles

- **Seguimiento de puntos dinámicos**

En primer lugar sobre los puntos dinámicos cercanos se agrupan en una misma nube, que se emplea como medida en un filtro de Kalman Extendido, que agrupa los puntos dinámicos cercanos en una que los almacena para realizar su seguimiento. Así las nubes de puntos dinámicos son agrupadas en un único obstáculo dinámico que pasará a ser mapeado en el espacio de velocidades tiempo.

- **Clasificador de piernas**

En las pruebas reales, la estimación de la localización del robot empeoró radicalmente, lo que obligó a realizar una clasificación de estáticos más rigurosa para mejorarla. Como consecuencia la clasificación de puntos dinámicos se vio afectada y no era posible realizar el seguimiento de los obstáculos dinámicos por este método.

Por ello fue necesario emplear otro filtro para los obstáculos dinámicos y se adaptó el desarrollado en [Urcola and Montano(2011)]. Este filtro no realiza la clasificación sobre puntos estáticos y dinámicos sino que busca el patrón 'piernas'. Se clasifica como pierna una agrupación de puntos consecutiva de un tamaño acotado, cuyos puntos limítrofes presentan un salto mínimo con los puntos siguientes.

Una vez identificado el patrón pierna se compara con las predicciones de los identificados anteriormente, se agrupa con el vecino más cercano y se introduce en el filtro como observación.

En la figura B.3a se observan dos patrones de clasificación de piernas. El más pequeño corresponde a una persona y el grande a un robot, ambos fueron correctamente identificados. La figura B.3b presenta dos piernas de la misma persona que también son correctamente clasificadas.

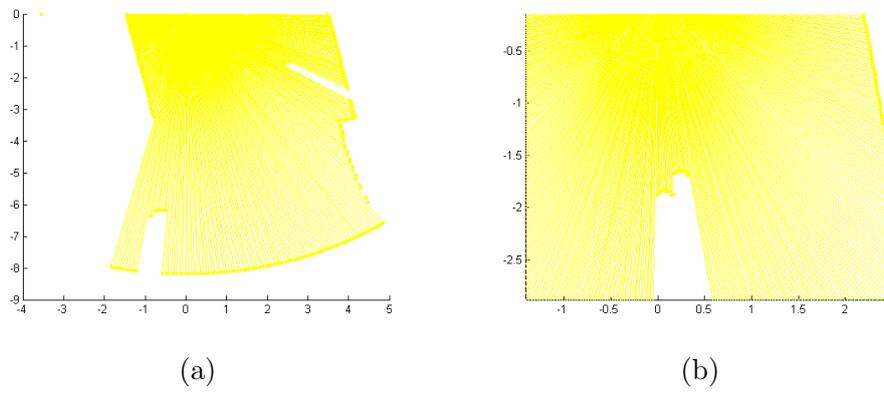


Figura B.3: Láser: Identificación del patrón pierna

Apéndice C

Simulaciones

El propósito de este anexo es detallar las pruebas introducidas en el capítulo 6.

C.1. Cruce

A continuación se muestran imágenes del espacio de trabajo, del DVTS y de su proyección el DVS del ejemplo presentado en 6.1.

El robot comienza avanzando hacia el objetivo, pasando por delante de los obstáculos mientras le es posible. En las figuras C.1a se observa que aunque el espacio se esté cerrando al aparecer varios obstáculos enfrentados, la búsqueda por debajo de las superficies prohibidas permite pasar por delante. Cuando ésto ya no es posible, debe evitar a un obstáculo que viene desde su derecha por detras, por lo que gira a la derecha, figura C.1b. Una vez este obstáculo ha sido evitado, sin embargo no puede recuperar su alineación al objetivo. Aquí hay otro obstáculo que le viene por la derecha y al que no le da tiempo de pasar por delante, figura C.2b, por lo que mantiene su dirección hasta pasarlo por detrás, C.2b. Todavía le quedan dos obstáculos, pero uno ya se está alejando, C.2c, y al último lo pasa por delante (figura C.2d) para finalmente alcanzar su objetivo.

Este escenario es especialmente interesante porque muestra cómo, a pesar de que gran parte del DVTS está prohibido para tiempos lejanos, la planificación se realiza por debajo de las superficies prohibidas permitiendo realizar una trayectoria segura a pesar de la aparición de varios obstáculos de forma simultánea.

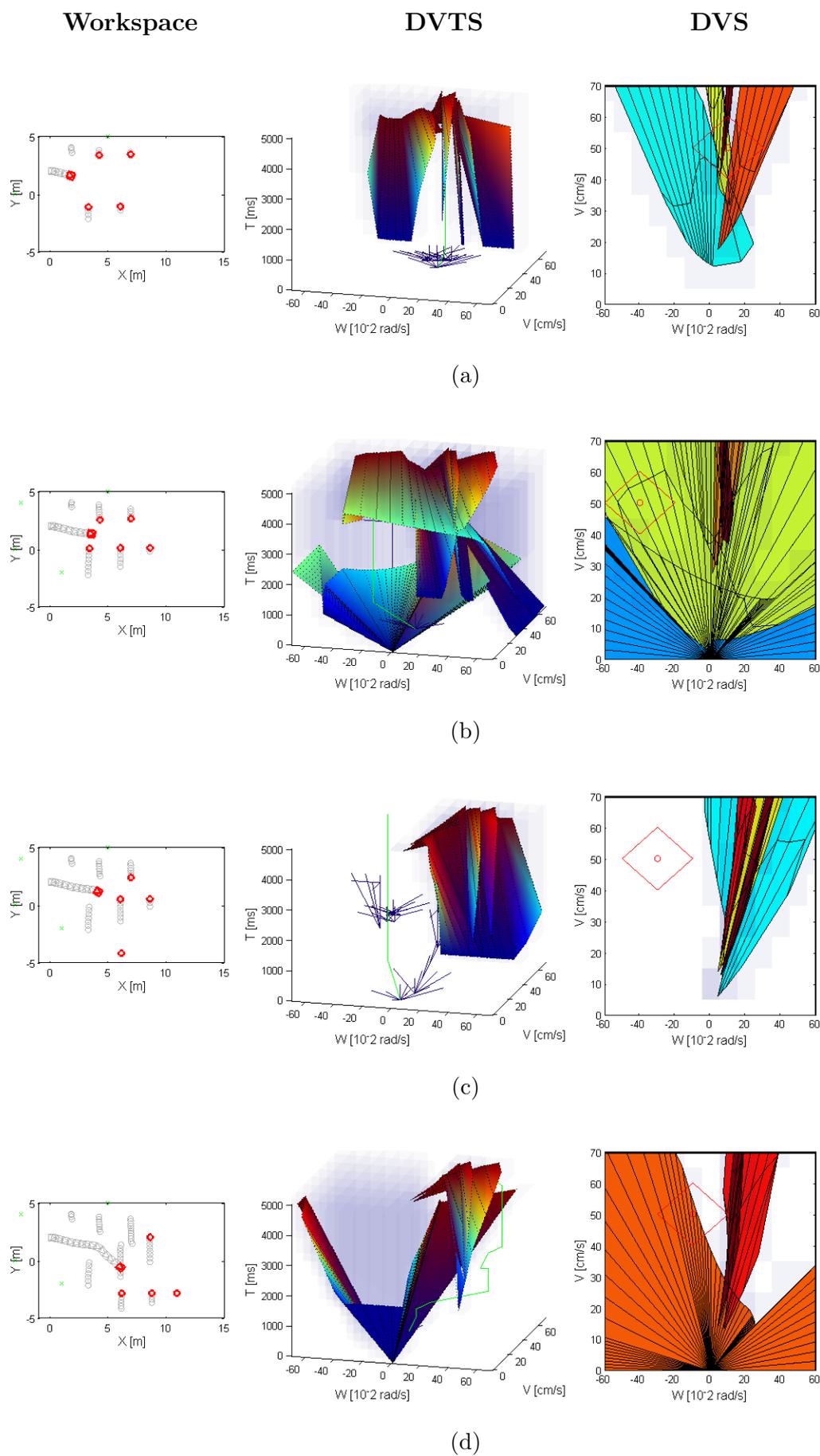


Figura C.1: Carriles I

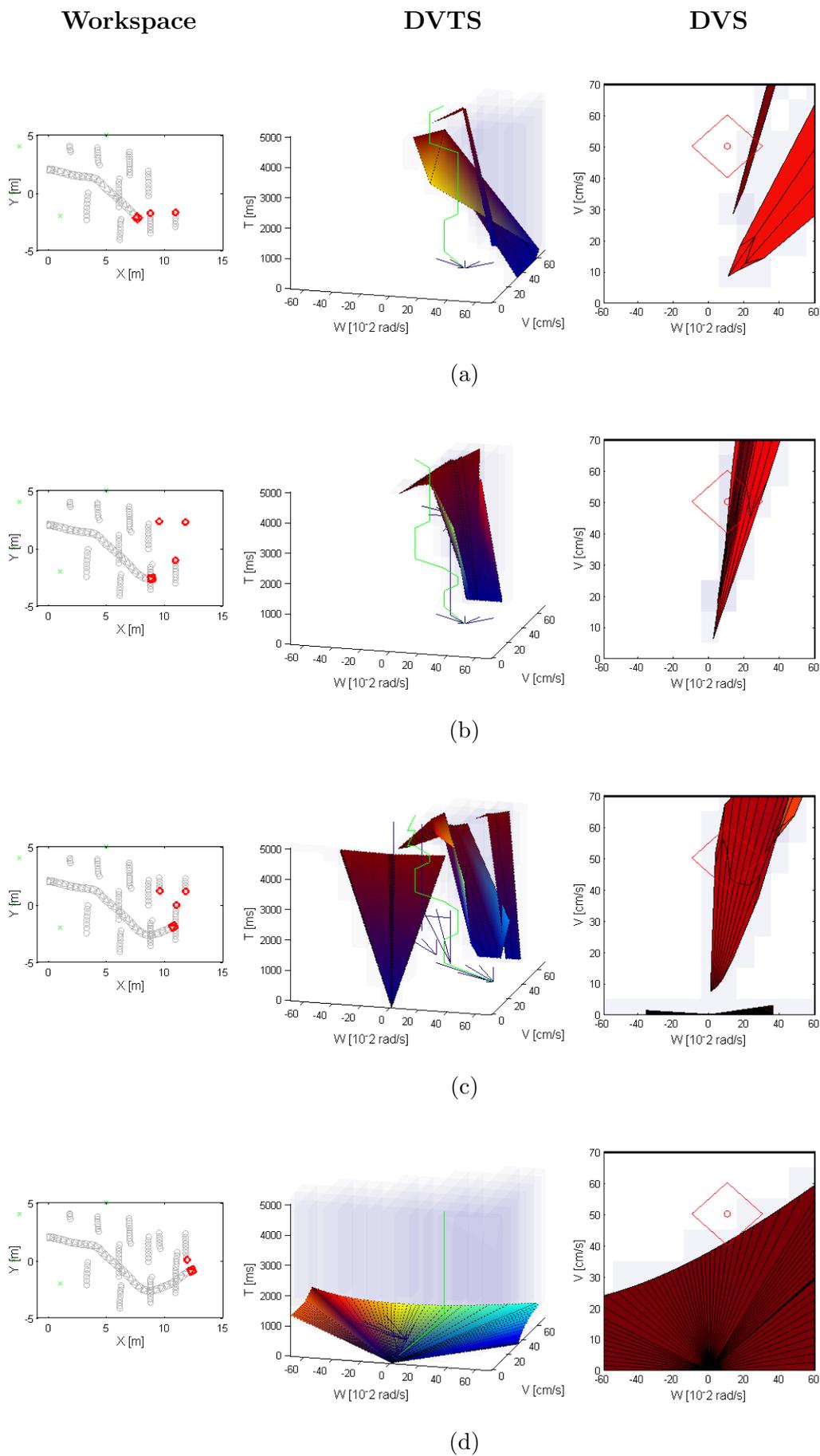


Figura C.2: Carriles II

C.2. Caos

El escenario *Caos* se desarrolla en un espacio abierto con un gran número de obstáculos circulando en todas direcciones, alguno de ellos con trayectorias no lineales.

El robot comienza realizando una maniobra de alineación con el primer obstáculo al que pasa por delante C.3a. Lo pasa por delante y mantiene su dirección para pasar por delante de otro obstáculo que le viene por su derecha C.3b. En C.3c planea pasar entre un vehículo que se le acerca por la derecha y el que se le acerca por la izquierda. Sin embargo el vehículo de su izquierda describe una trayectoria curva y se le acerca obligándole a realizar una alineación más drástica de alineación con él C.3d, una vez alineado y hasta superarlo las velocidades de giro a izquierdas están altamente penalizadas. Una vez lo sobrepasa el peligro más inmediato es otro vehículo que se le cruza en su trayectoria C.4a. Lo rodea por detrás, dirigiéndose a su objetivo C.4b. El último obstáculo lo intenta evitar mediante un giro a izquierdas, pero no es suficiente, se le cierra el espacio de búsqueda y debe reducir la velocidad C.4c. Termina alcanzando su objetivo.

Una de las hipótesis de las que parte el cálculo del DVTS a través de la Banda de Colisión es que los obstáculos móviles se desplazan con trayectorias lineales en la que se basa, dada la frecuencia de actualización del programa se trata de una hipótesis aceptable. Este escenario presenta varios obstáculos con velocidades angulares que han sido correctamente evitados en un escenario bastante poblado.

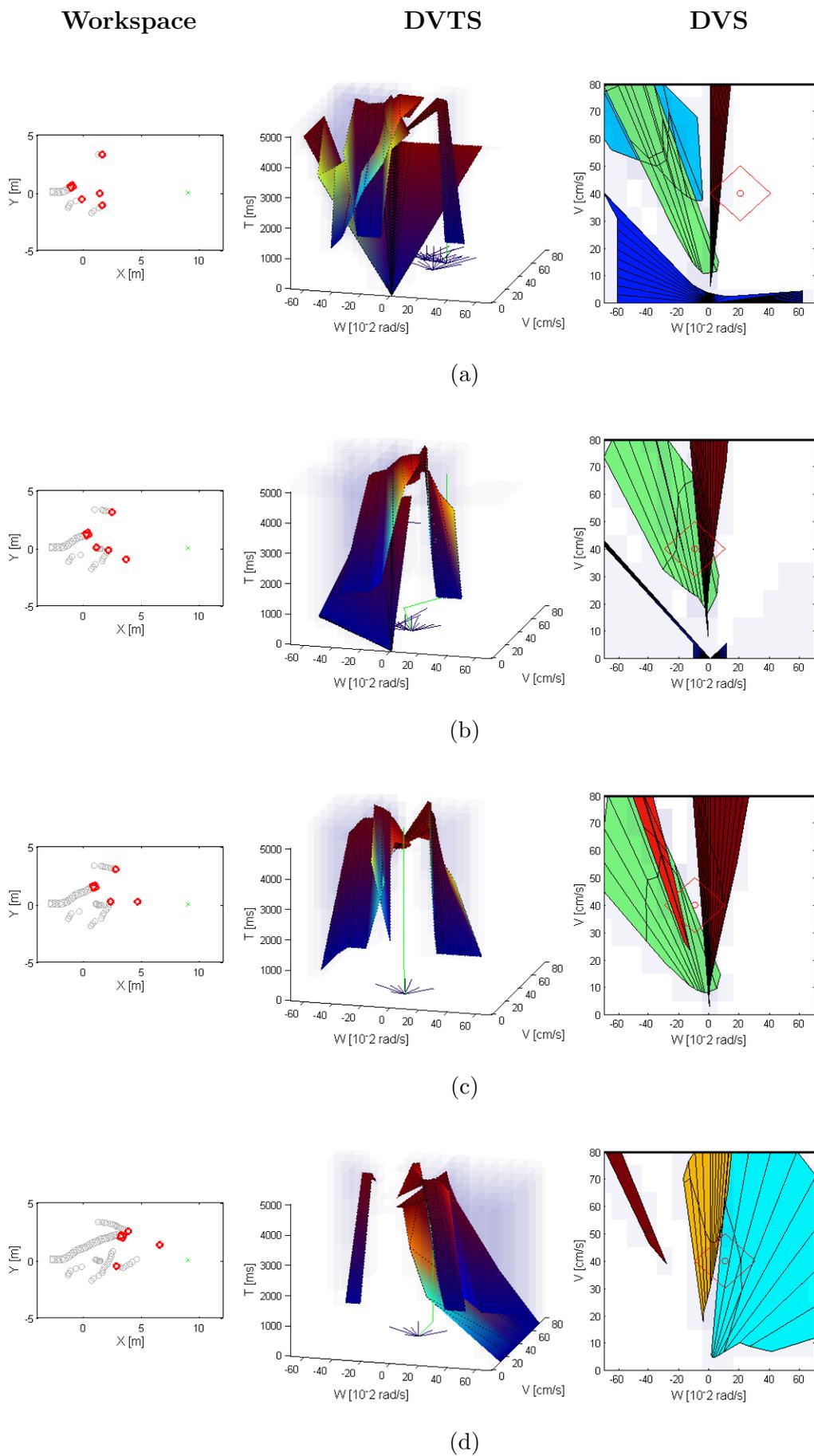


Figura C.3: *Caos I*

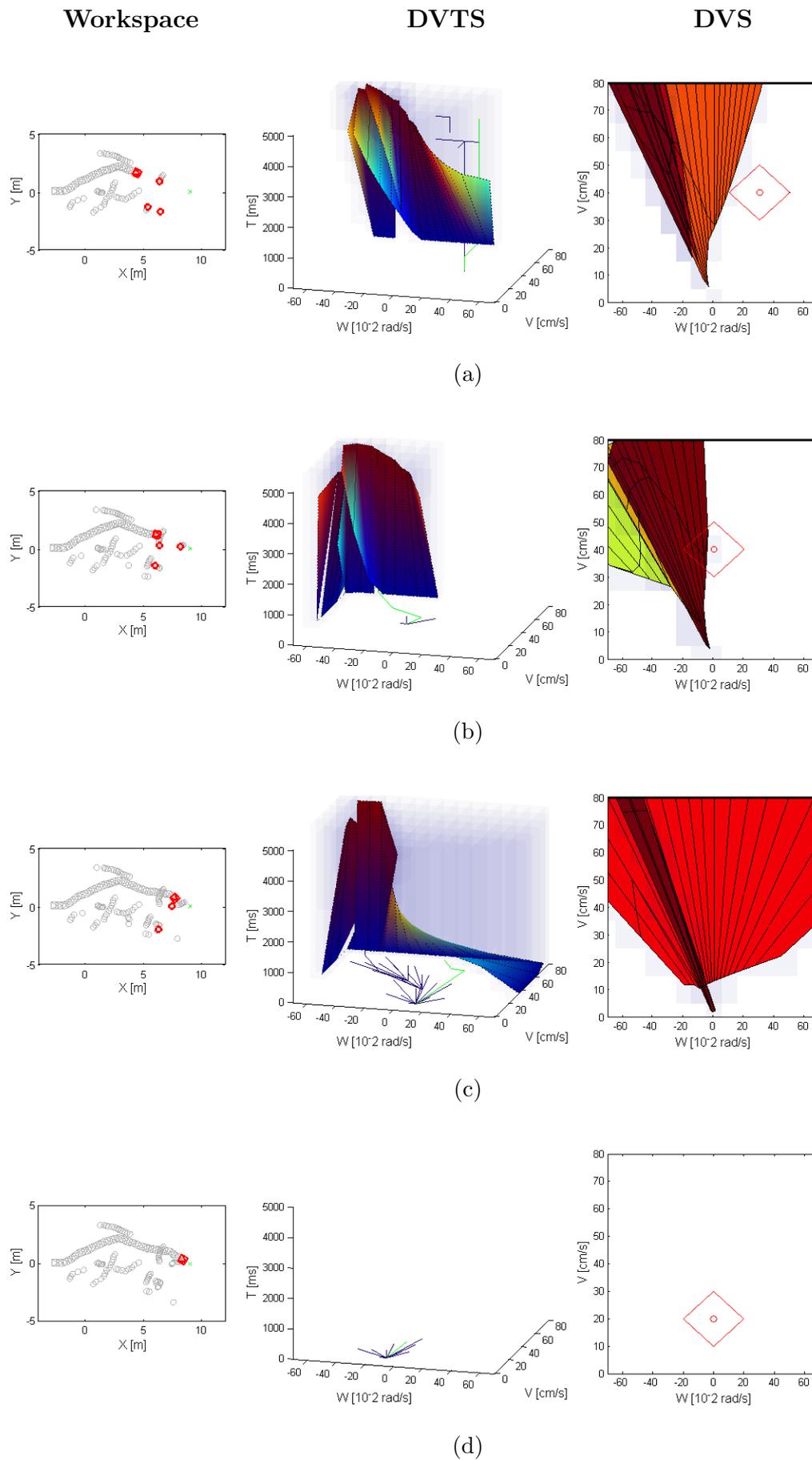


Figure C.4: *Caos II*

C.3. Interiores

En este apartado se desarrolla más en detalle el caso presentado en 6.3.

En el primer tramo, C.5, el robot comienza avanzando en línea recta para salir de la sala. Percibe los dos primeros obstáculos que se cruzan en dirección perpendicular a la de su primer objetivo. Inicialmente intenta evitarlos pasando por detrás del primero C.5a, pero no es posible por lo que debe reducir la velocidad. Se mantiene a bajas velocidades en espera de que sea posible volver a avanzar de forma segura, C.5b. Una vez el obstáculo 2 le deja espacio suficiente, comienza a evitar al obstáculo 1 por detrás C.5c evitándolo por la izquierda. Retrasa recuperar la posición para no encontrarse de frente con el obstáculo 3, una vez es seguro la recupera y avanza hasta su primer objetivo C.5d.

Una vez alcanza el primer subobjetivo debe girar a izquierdas para orientarse con el siguiente. Además a su izquierda hay tres obstáculos sin embargo, la superficie prohibida dinámica es elevada (los obstáculos 5 y 6 están lejos y el obstáculo 4 ya ha pasado) C.6a . La estrategia de evitación C.6b sería pasar por detrás para ambos, pero no es posible ya que vienen en sentidos contrarios. pasa por detrás del obstáculo 5 y un poco más adelante C.6c debe frenar para dejar pasar al obstáculo 6. Aprovecha ese tiempo parado para intentar alinearse con una dirección de escape o con el objetivo. Alcanza su siguiente subobjetivo C.6d .

Comienza a girar a derechas para alinearse con el objetivo siguiente. C.7a. Tiene que volver a reducir su velocidad debido al obstáculo 7, ya que no le ha sido posible evitarlo por detrás porque no le ha dado tiempo de girar lo suficiente. Cuando le es posible lo evita por evitarlo por detrás C.7b, además intenta evitar al obstáculo 6 y al obstáculo 9. Intenta una estrategia de alineación con el obstáculo 6, C.7c, pero dado que éste no sigue una trayectoria recta, termina reduciendo velocidad para rectificar su dirección, dejarlo pasar C.7d y no alejarse de su objetivo. Alcanza el tercer objetivo y se encamina hacia el objetivo final.

Para ello debe atravesar una estrecha puerta hacia la sala final mientras el obstáculo 10 se dirige en sentido contrario hasta el pasillo. Mientras es posible, intenta alinearse con el objetivo C.8a para girar después. Se le van cerrando las posibilidades, forzando un giro a radical a derechas, para abrirse un espacio que le permite pasar por la puerta paralelo al obstáculo C.8b. En la figura C.8c se observa que se le ha cerrado completamente el espacio de velocidades, pero empleando órdenes de las trayectorias calculadas en iteraciones anteriores avanza hasta volver a planificar nuevas trayectorias.

Se ha mostrado un entorno con obstáculos móviles y estáticos, en el que a través de un planificador global que le ha introducido tres puntos de vía intermedios, el robot se desplaza por un entorno de interiores. Es el escenario más completo de los presentados en este proyecto, permite hacerse una idea de la potencia de este método.

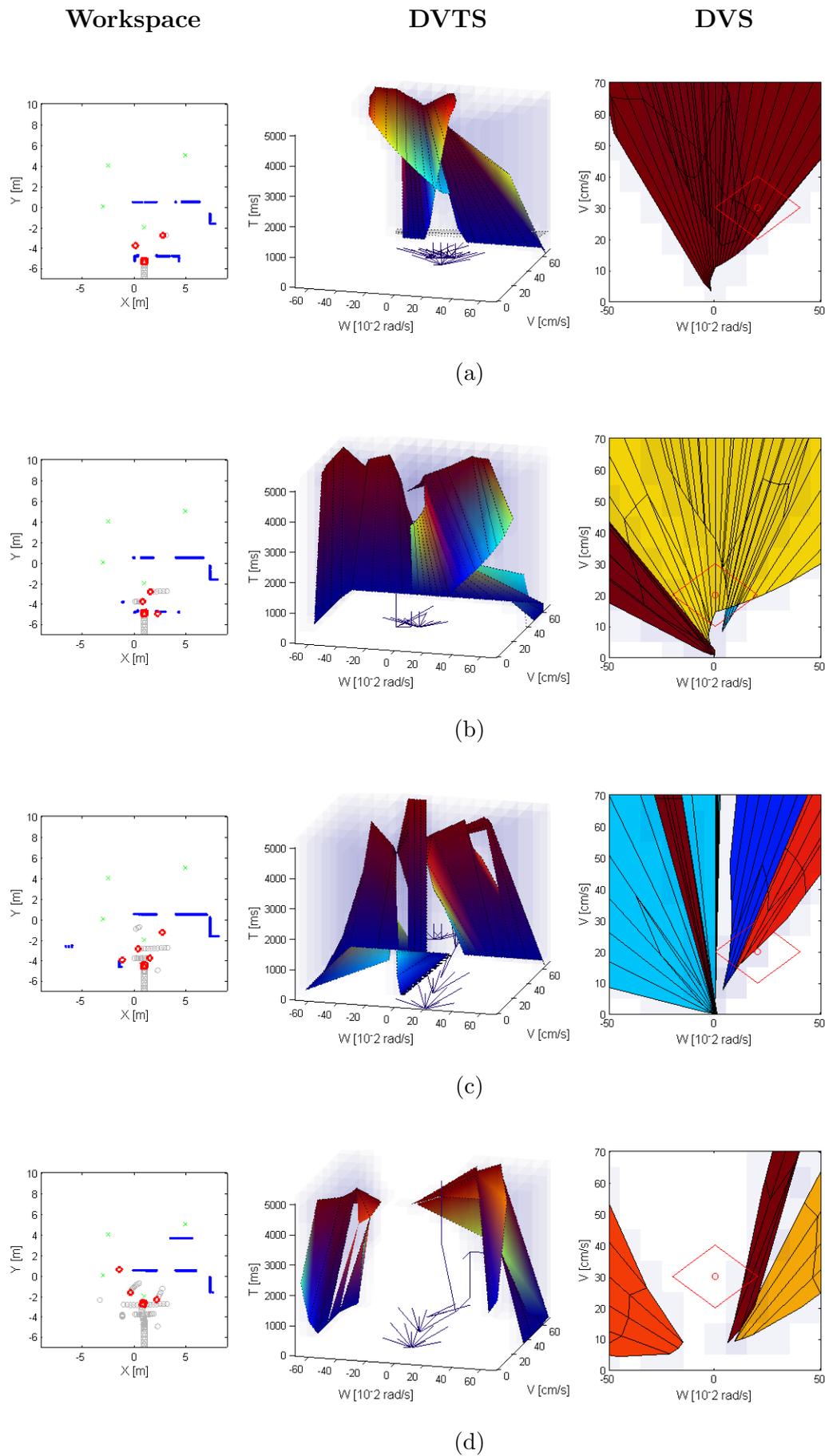


Figura C.5: *Interiores*: primer tramo

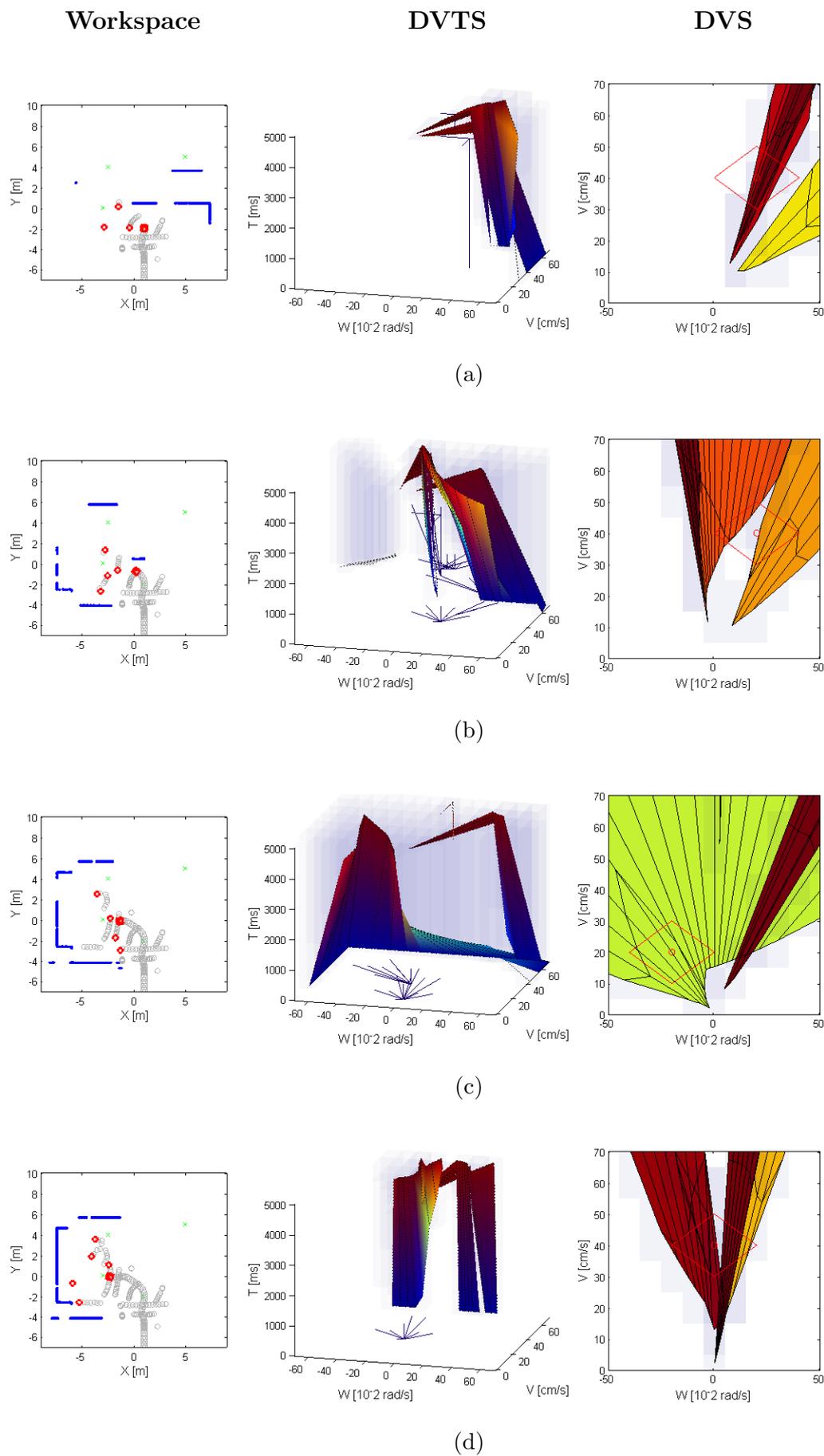


Figura C.6: *Interiores: segundo tramo*

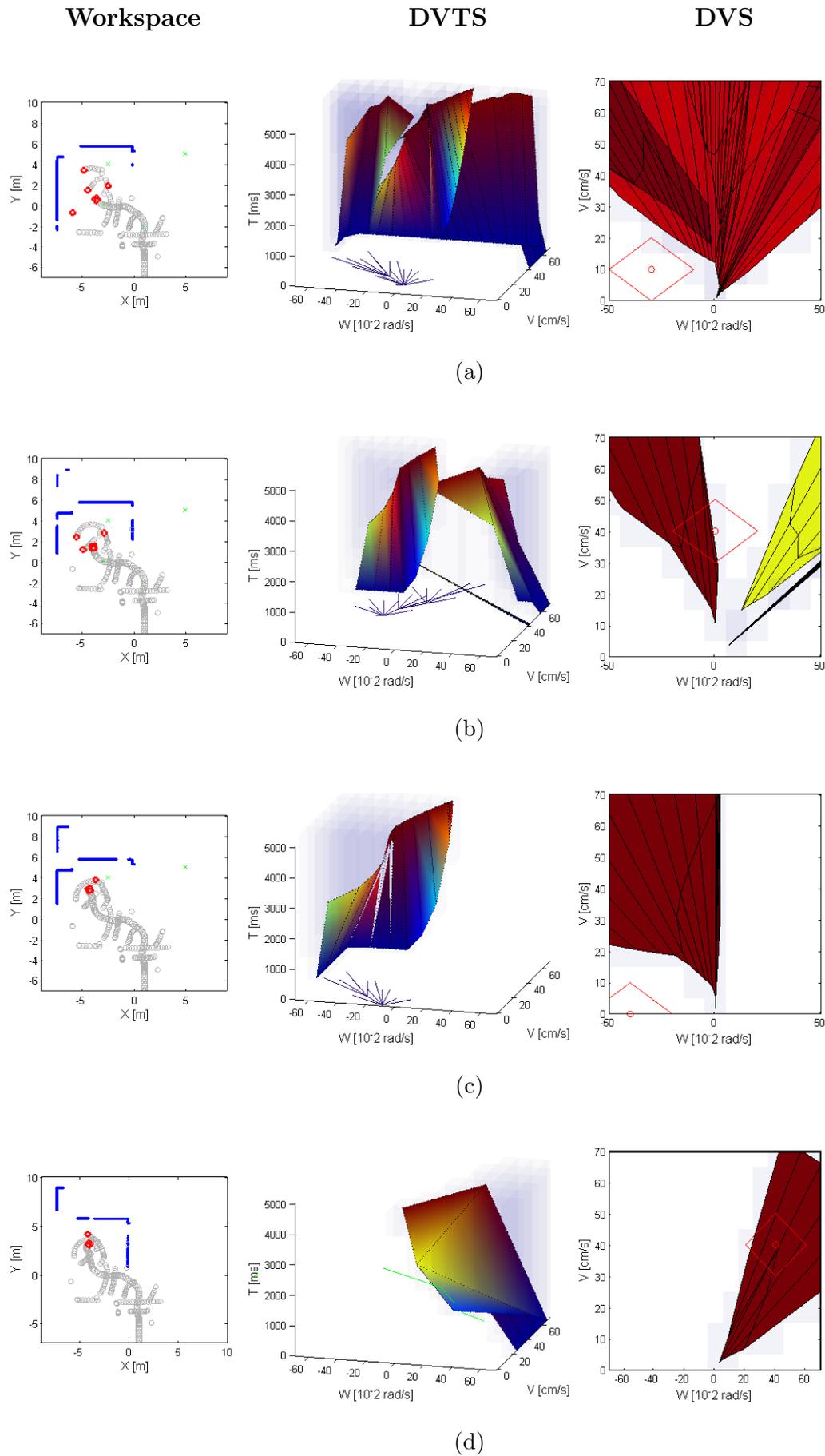


Figura C.7: *Interiores*: tercer tramo

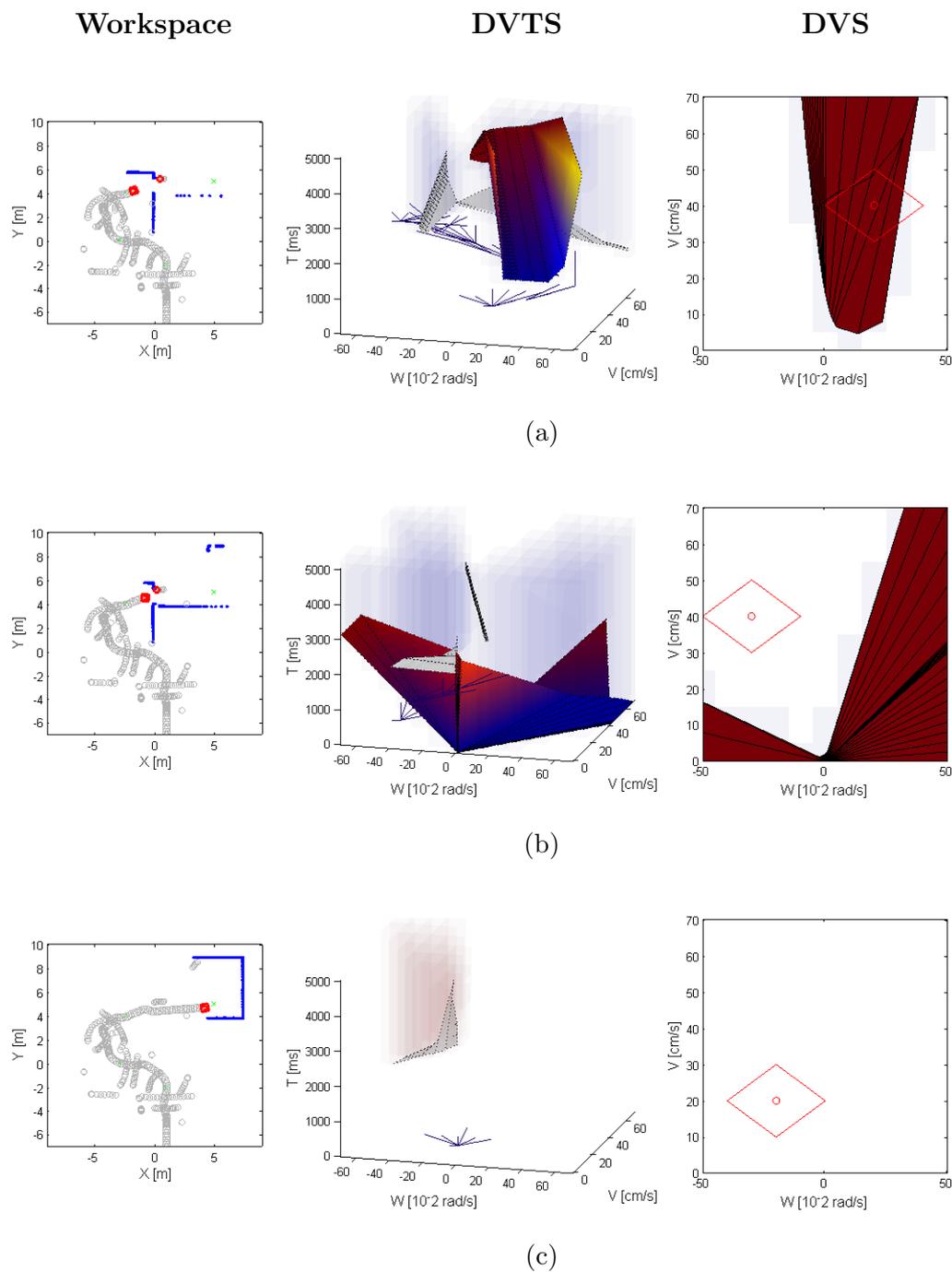


Figura C.8: *Interiores*: cuarto tramo

C.4. Adelantamiento

En el anexo A.2 se ha presentado más en detalle el método de cálculo de las estrategias para el caso del adelantamiento. A continuación se desarrolla el escenario correspondiente presentado en el capítulo 6.

Consiste en una carretera en la que el robot se encuentra un vehículo más lento en su carril al que no puede adelantar de forma inmediata por dos vehículos en el carril contrario que se lo impiden.

En la figura C.9a se muestra el momento en el que se identifica el vehículo más lento. Al no poder adelantar se ve obligado a reducir la velocidad tal y como aparece en la figura C.9b. Al decidir la estrategia que se le introduce al A^* en su búsqueda según la situación en el espacio de velocidades, se consigue que el robot se adapte al vehículo lento, en lugar de estar oscilando hacia izquierda y derecha para intentar rodearlo.

Una vez el carril frontal está despejado, se acelera hasta la velocidad máxima y se comienza la maniobra de cambio de carril, figura C.9c. El primer tramo coincide con la primera clotoide, que va aumentando su radio de forma progresiva. Esto se traduce en una consigna de giro a izquierdas a la velocidad angular máxima como muestra la figura C.9d. A continuación se disminuye la velocidad angular (segunda clotoide) y se mantiene a velocidad angular nula durante el tramo de concatenación, figura C.10a. El cuarto y quinto tramo del cambio de carril consisten en un giro a derechas con consigna de velocidad angular máxima a derechas y velocidad angular nula respectivamente. La figura C.10b muestra el momento en el que la consigna cambia. Una vez terminado el cambio de carril, el vehículo se encuentra alineado.

Se mantiene paralelo al carril mientras sobrepasa al vehículo lento, figura C.10c. Una vez lo deja atrás, vuelve a su carril con una maniobra similar. Comienza con un giro a derechas, figura C.10d. Reduce la velocidad angular a cero C.11a. A continuación realiza un giro a izquierdas, figura C.11b, y termina alineado en su carril, figura C.11c.

Mientras estaba realizando la maniobra de vuelta, se acercaba otro obstáculo por el carril izquierdo. En la figura C.11d se ve cómo un vehículo en el carril contrario no le afecta para avanzar por su carril.

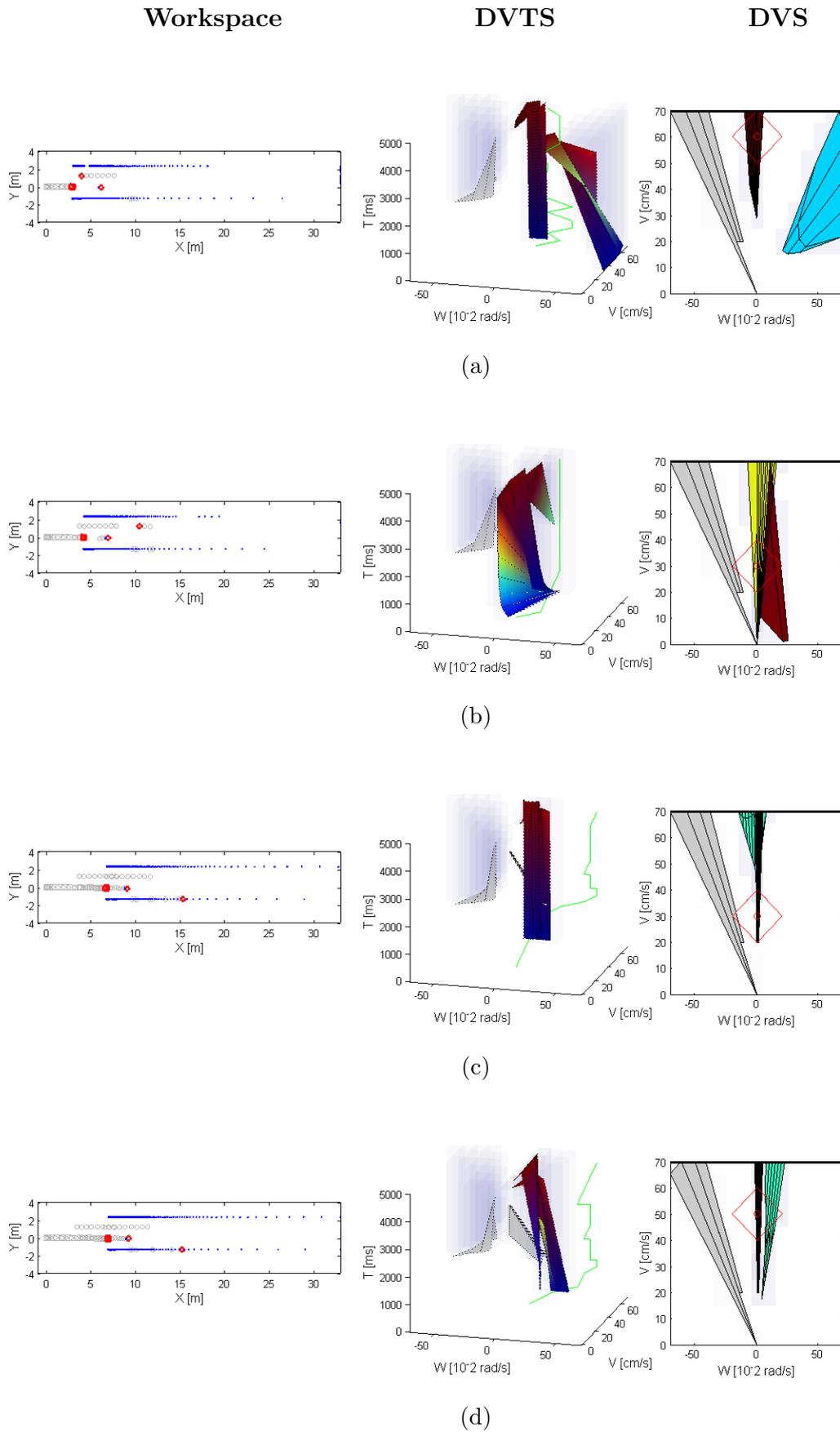


Figura C.9: Adelantamiento: I

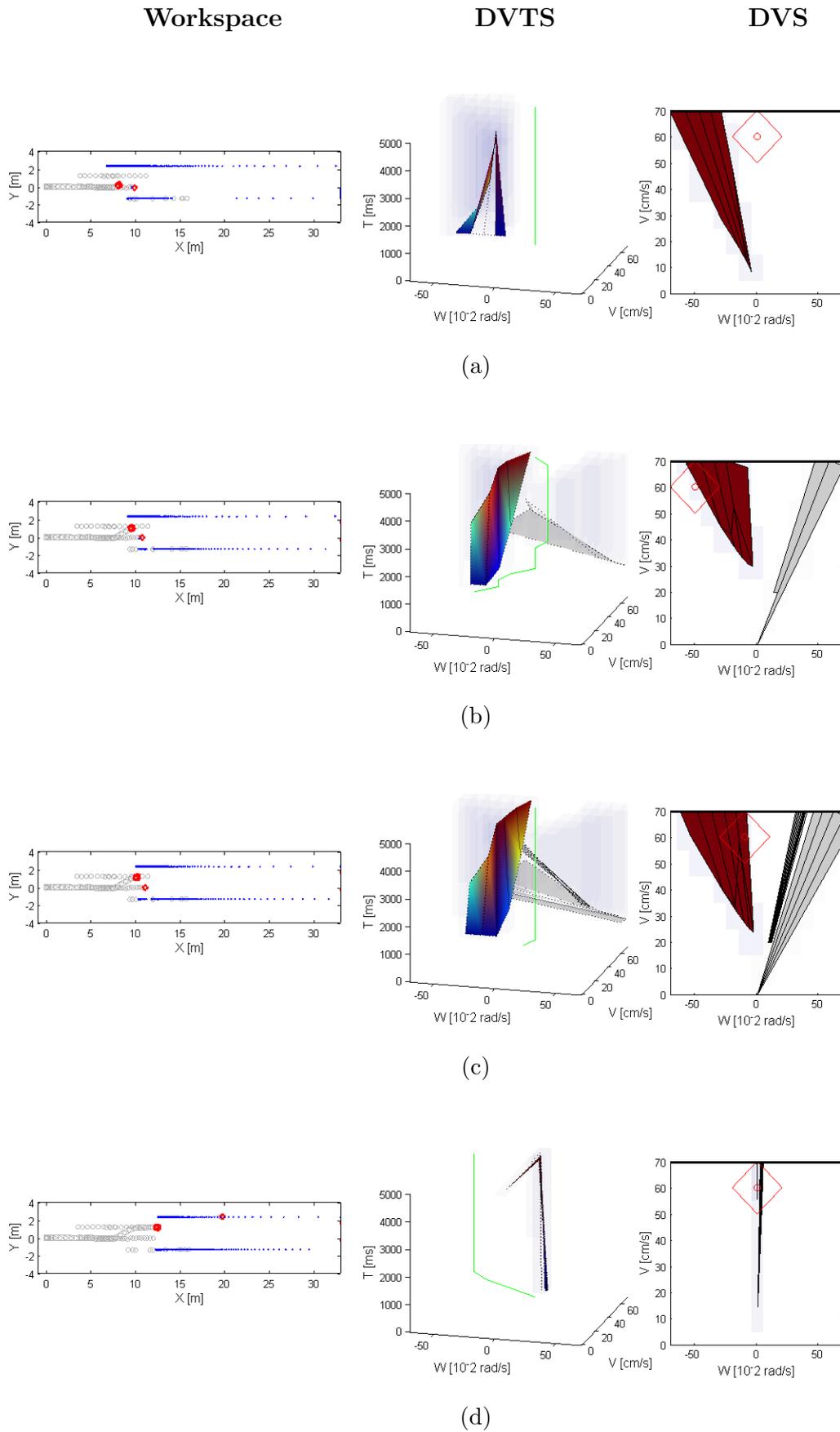


Figura C.10: Adelantamiento: II

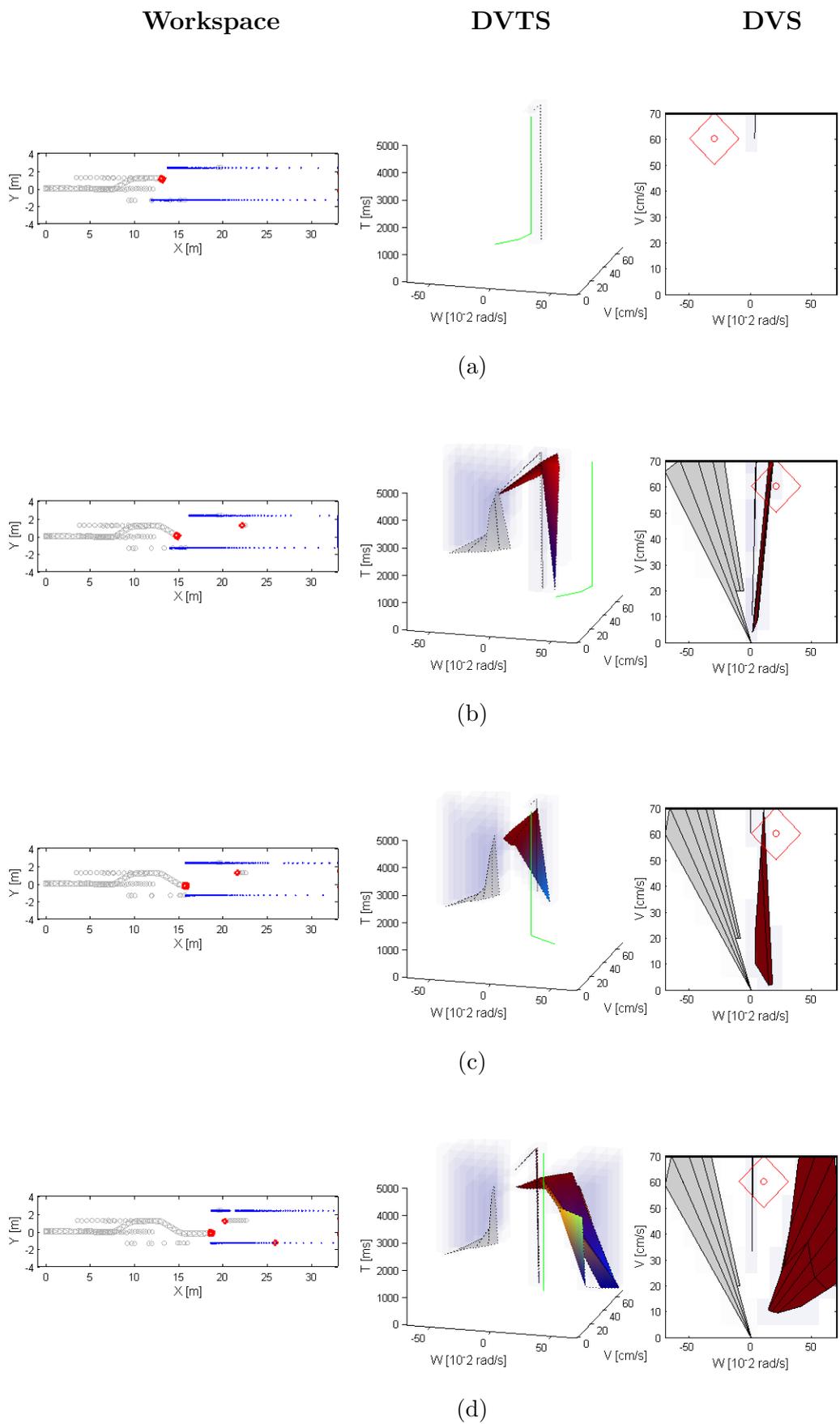


Figura C.11: Adelantamiento: III

Apéndice D

Experimentos

D.1. Hall

El robot comienza su recorrido cuando le aparece un obstáculo que se le acerca de derecha a izquierda, figura D.1a que comienza a evitar por detrás girando hacia la derecha. Cuando ha girado suficiente (la superficie prohibida del obstáculo, la azul oscura D.1b recupera su alineación girando a la izquierda. Sin embargo al evitar al obstáculo se ha alejado demasiado en orientación del primer vía de su trayectoria, por lo que debe reducir la velocidad, figura D.1c. Una vez recupera la orientación vuelve a aumentar su velocidad para llegar al primer punto vía. En la figura D.1d se observan además del primer obstáculo dos falsos positivos devueltos por el filtro que corresponden con las patas de un banco. El filtro de piernas tiene un parámetro para distinguir las piernas de las patas de sillas o mesas a través del tamaño, pero la anchura de las patas de este banco es comparable al tamaño de una pierna.

Superado el primer punto vía, el robot continúa la trayectoria por la parte más larga de la entrada. De izquierda a derecha se le acerca otro obstáculo móvil, granate en el DVS de la figura D.2a. Lo rodea mediante un giro primero a izquierdas y luego a derechas, figura D.2b. Otro obstáculo le viene de derechas a izquierdas, lo evita con un giro a derechas y luego recuperando su alineación, figura D.2c. Los dos obstáculos que aparecen en la figura D.2c corresponden a sendas piernas.

Aparece otro obstáculo móvil que comienza avanzando paralelo al robot, y luego se le cruza por delante. Lo deja pasar por delante desviándose a la derecha para no reducir velocidad, figura D.2d. Cerca de alcanzar su segundo punto vía, otro obstáculo se dirige de frente hacia él, figura D.3a. Al venir de frente, le cierra el espacio de velocidades, obligando al robot a desviarse de su trayectoria con un giro a izquierdas. Una vez se ha salido de la trayectoria del obstáculo, lo rodea dejándolo a su derecha, figura D.3b. Detecta otro obstáculo que se aleja por la izquierda, figura D.3c. Se mantiene a su izquierda hasta que alcanza el objetivo final, figura D.3d.

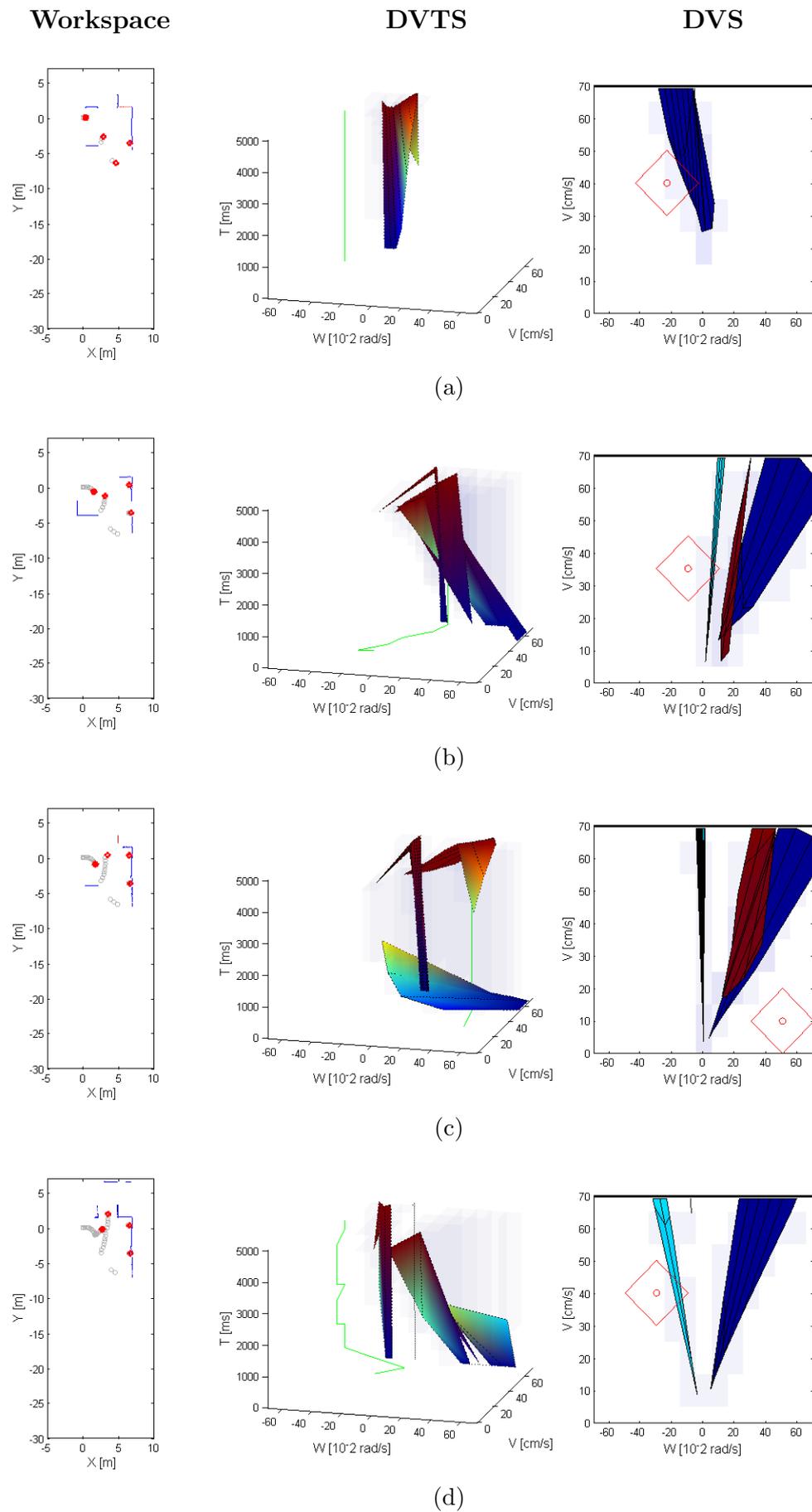


Figura D.1: *Hall: I*

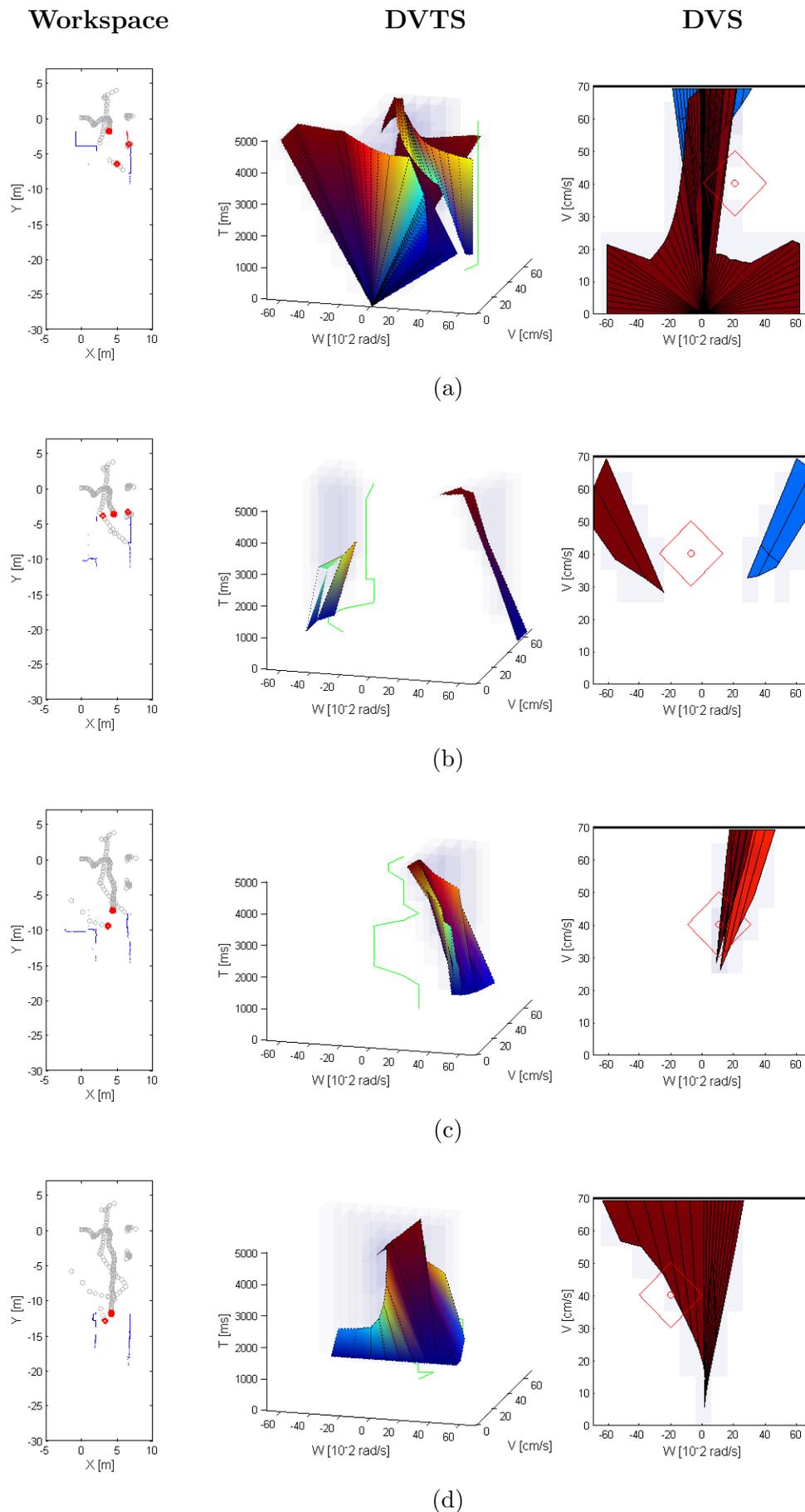


Figura D.2: *Hall: II*

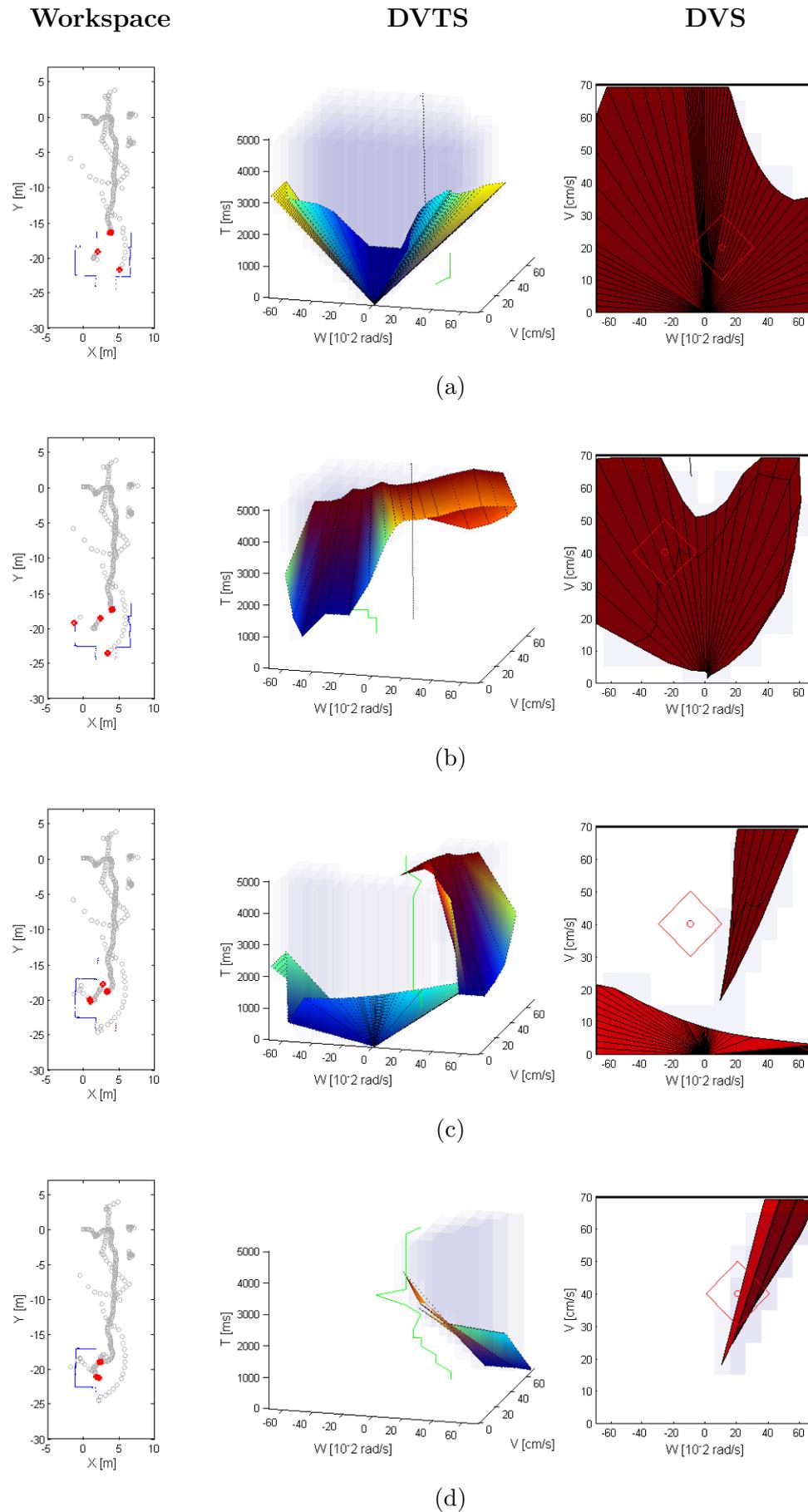


Figure D.3: *Hall*: III

D.2. Adelantamiento

El robot comienza avanzando con su carril, pero se encuentra con un vehículo más lento al que no puede adelantar porque le viene otro de frente. En la figura D.4b se observa cómo se mantiene a velocidades bajas detrás del vehículo lento.

Una vez desaparece el vehículo del otro carril comienza el adelantamiento. En primer lugar la consigna de velocidad es girar a izquierdas, D.4c, que corresponde con el primer tramo de los cinco que componen el cambio de carril mediante clotoides. En el segundo tramo reduce la velocidad angular hasta cero y la mantiene D.4d hasta completar el tercer tramo, el tramo lineal entre los dos pares de clotoides simétricas. El cuarto tramo consiste en girar a derechas, figura D.5a, para finalizar el cambio de carril reduciendo la velocidad angular a cero y terminando alineado, D.5b.

Avanza paralelo al vehículo lento (figura D.5c). Una vez deja de verlo y estima que ya lo ha sobrepasado comienza la vuelta al carril derecho. Primero realiza un giro a derechas (figura D.5d), reduce velocidad angular (figura D.6a), gira a izquierdas (figura D.6b) y enlaza con el carril por el que circula normalmente D.6c.

Detecta un obstáculo móvil que invade su carril en sentido contrario. Cuando la distancia es demasiado cercana realiza la maniobra de emergencia, figura D.6d. Al igual que en el adelantamiento realiza el cambio de carril girando a izquierdas, luego avanzando (figura D.7a) y finalmente girando a derechas para enlazar con el carril izquierdo (figura D.7b). Una vez el peligro ha desaparecido vuelve a su carril girando a derechas, y a izquierdas para enlazar (figura D.7c) y terminar el recorrido (figura D.7d).

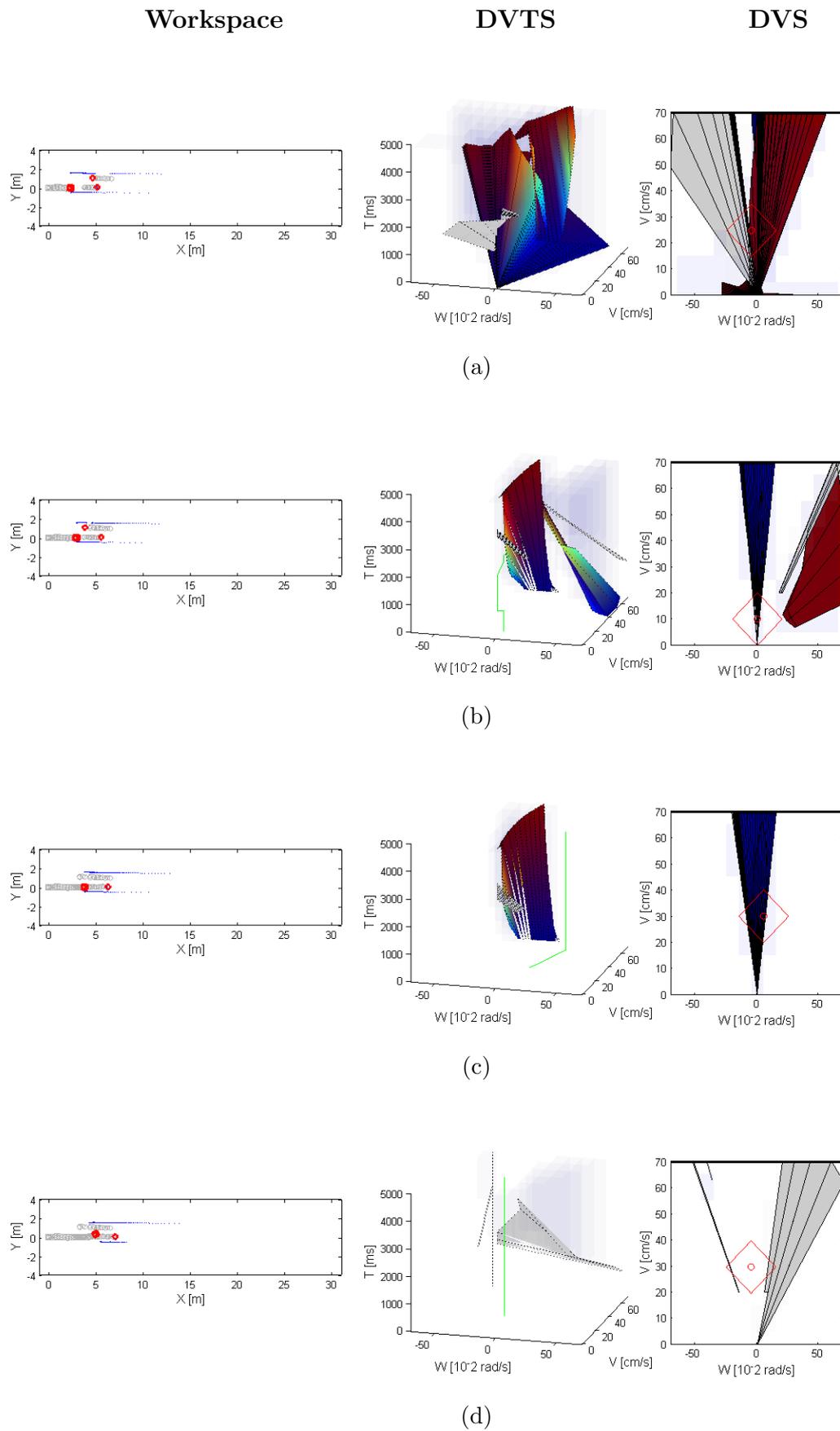


Figura D.4: Adelantamiento real: I

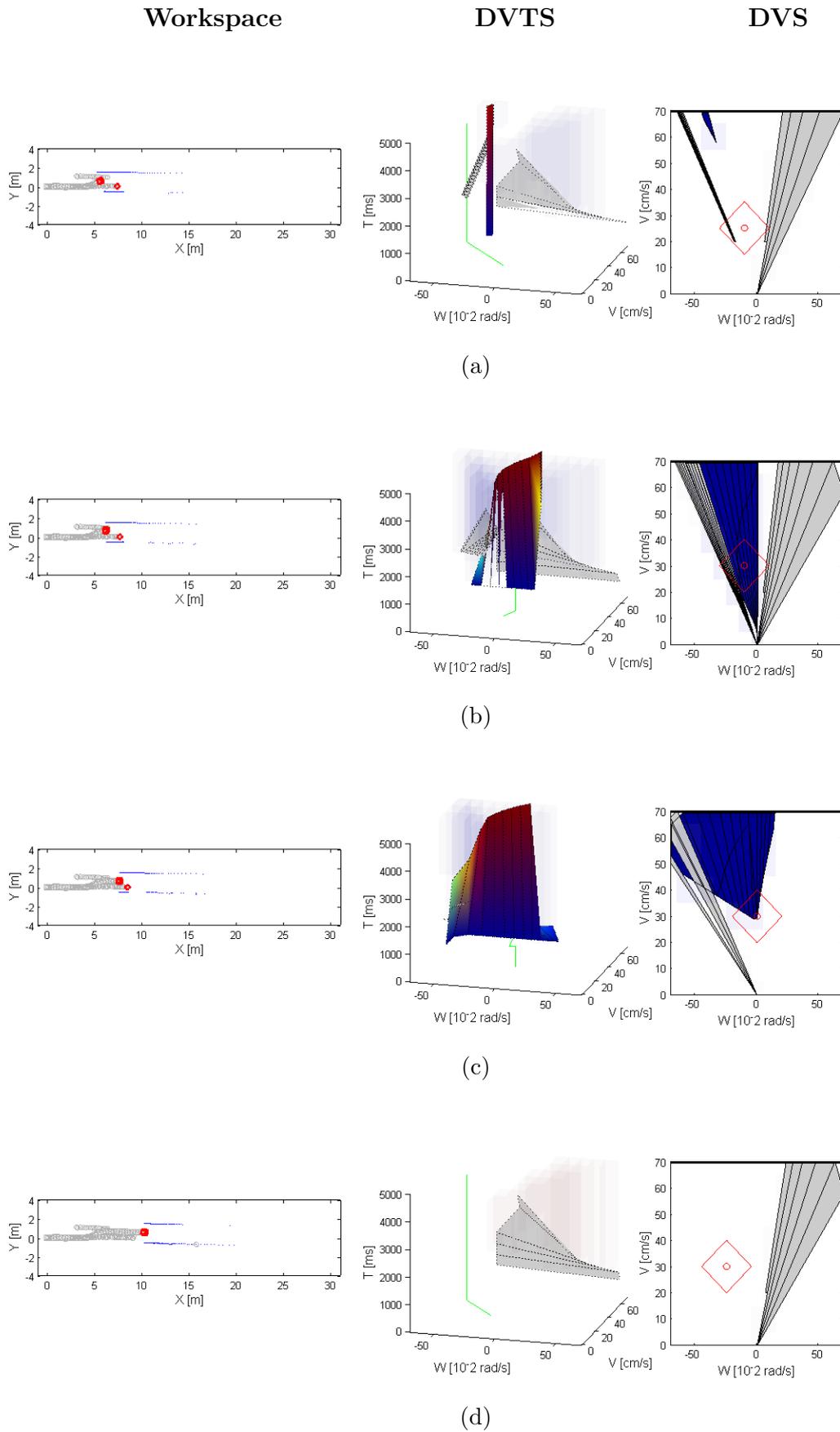


Figura D.5: Adelantamiento real: II

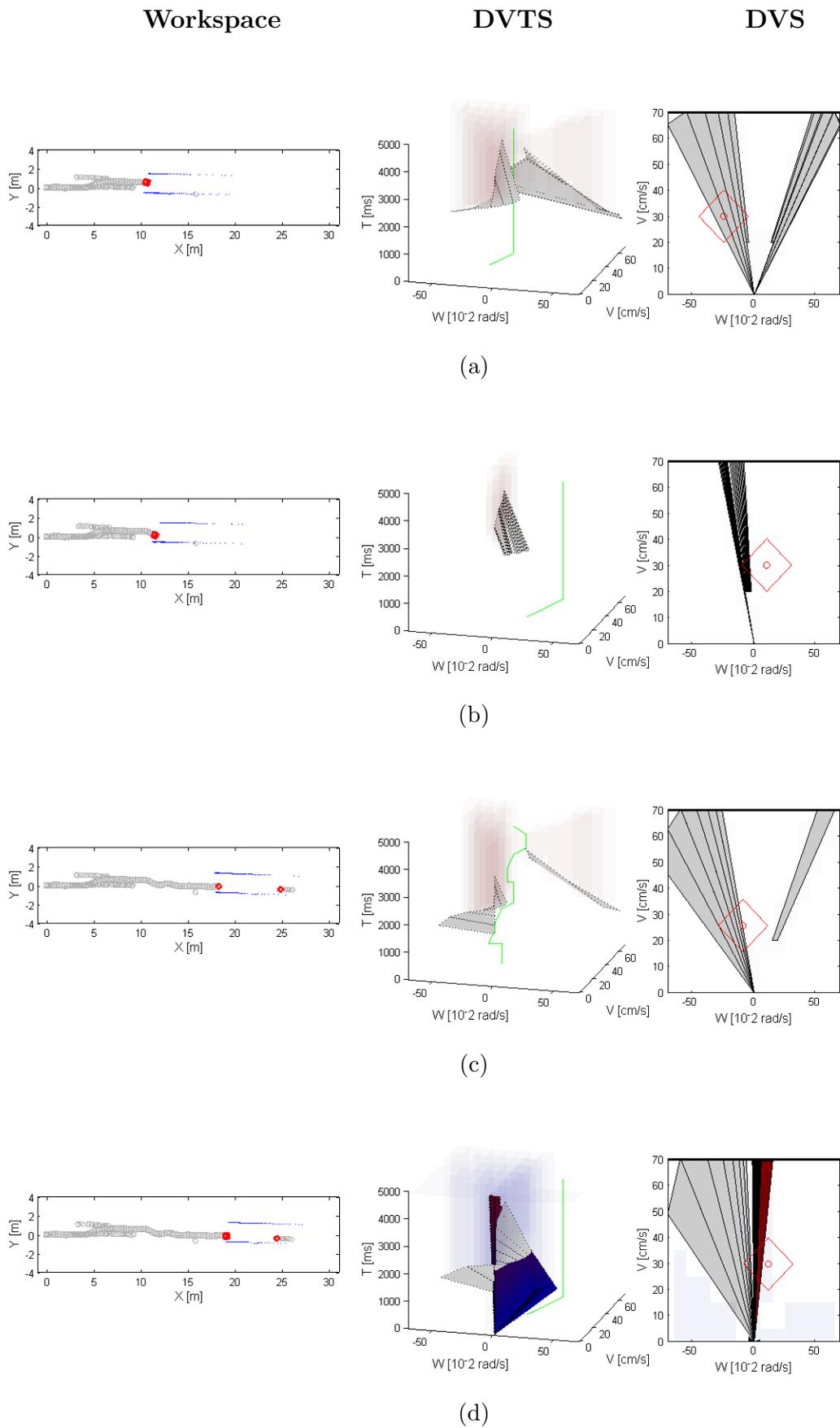


Figura D.6: Adelantamiento real: III

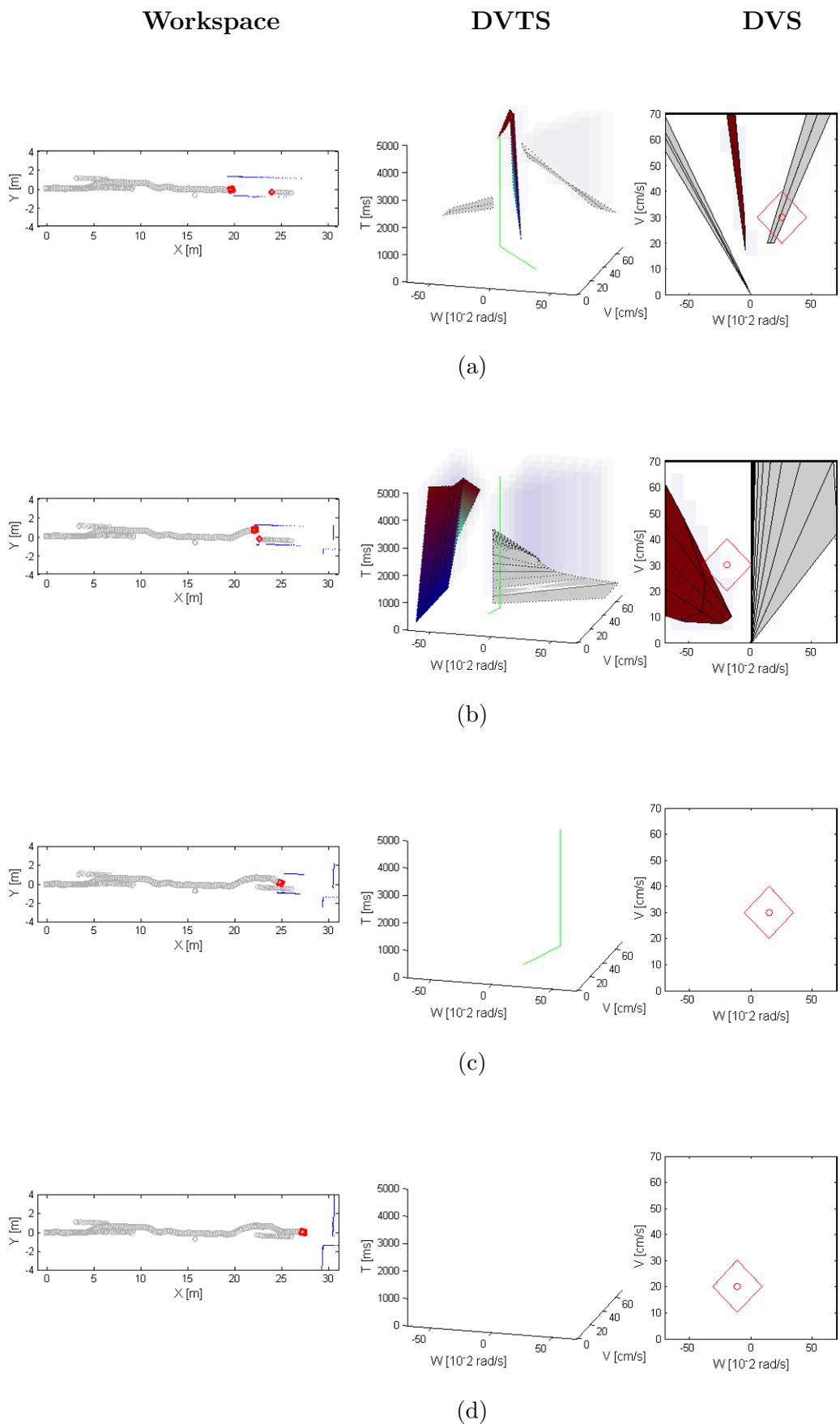


Figura D.7: Adelantamiento real: IV

Apéndice E

Consideraciones

E.1. Límites del espacio de velocidades

En simulación todas las velocidades son alcanzables, 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 consta de cuatro ruedas que giran independientemente, permitiéndole realizar giros sobre sí mismo. En la figura E.1 se muestra una imagen del robot.

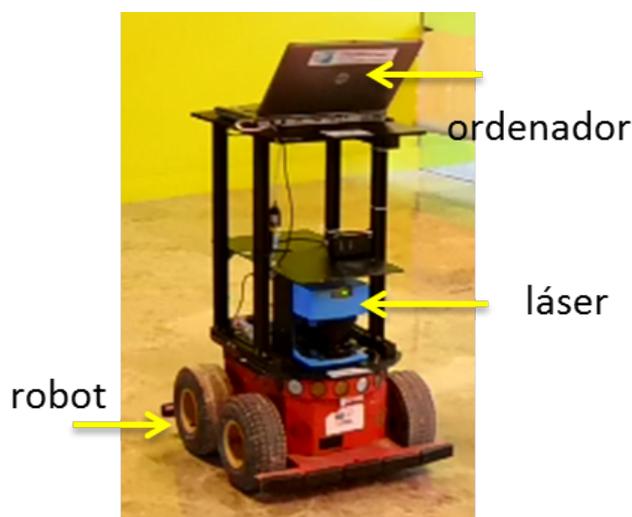


Figura E.1: Robot Pioneer 3AT

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 llevaba a errores en la actuación. Si el algoritmo devuelve un par de velocidades que no es posible, se producen situaciones no deseadas. Por ejemplo en algunas pruebas realizadas antes de observar este fenómeno resultaba que el robot seguía en línea prácticamente recta hacia una pared. Se comprobó que

la consigna del programa le mandaba girar, pero al ser una velocidad no alcanzable el controlador interno del robot priorizaba la velocidad lineal y despreciaba 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 los pares de velocidad lineal y angular que se mantenían estables y con error de posición cero ante la consigna. En la figura E.2 se muestran los límites medidos.

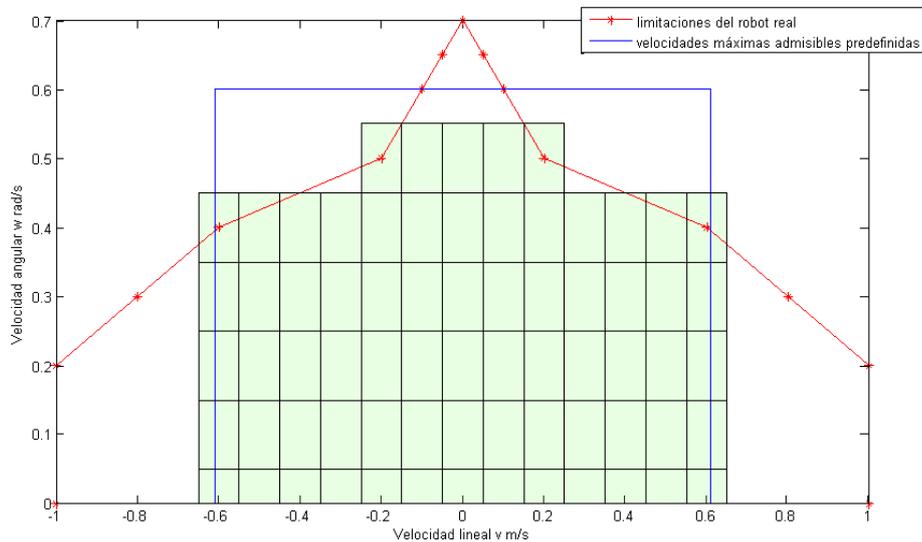


Figura E.2: Límites del espacio de velocidades real

E.2. Pruebas de coeficientes

En el capítulo 3.3 se presentaban los distintos términos que componen la función de costes. Para decidir la ponderación de cada uno se llevaron a cabo diferentes ensayos. En esta sección se presentan los resultados más relevantes y se analiza un poco más en profundidad la importancia de cada término.

Importancia del término de selección de velocidad

Algunos métodos de búsqueda de trayectorias ya emplean la información de seguridad y distancia al objetivo. Una de las contribuciones de este proyecto es la integración de información del espacio de velocidades y definición de estrategias determinadas para realizar la búsqueda de la trayectoria. A continuación se muestra una serie de ensayos en los que se observa cómo el método mejora considerablemente.

Si el término de velocidades se deja con coeficiente cero, es decir que no se guía al robot para elegir la forma de evitación más adecuada ni se le guía hacia las velocidades más favorecedoras en ausencia de obstáculos, los resultados obtenidos son bastante negativos. Para conseguir evitar la colisión se debe restringir bastante el margen de actuación. Es necesario aumentar tanto el margen de seguridad como el peso del término de seguridad y penalizar en mayor medida hallarse debajo de la superficie prohibida ya que en caso contrario, el robot no evita a tiempo trayectorias que le llevan a colisión. Este caso se presenta en la figura E.3.

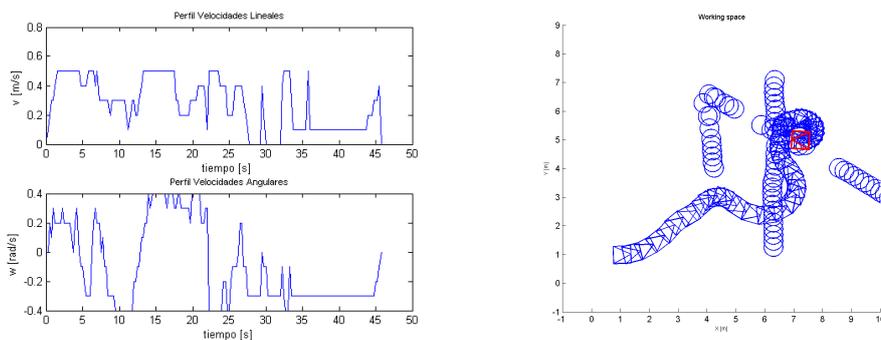


Figura E.3: Búsqueda sin guía de velocidades, $\alpha_{dist} = 1$, $\alpha_v = 0$, $\alpha_s = 4$

Simplemente incluyendo el *freemotion* la trayectoria converge de forma más adecuada. En la primera figura se observa como el robot debe reducir su velocidad para dejar evitar la colisión. Todavía no tiene la selección de estrategias de evitación, por lo que el movimiento resulta algo brusco. Además emplear el término de velocidades con el *freemotion* pero sin selección de estrategias resulta o bien demasiado arriesgado por cruzar con poco margen por delante o demasiado conservador. En la figura E.4 se observa como el robot espera a tener despejado el espacio para dirigirse a su objetivo.

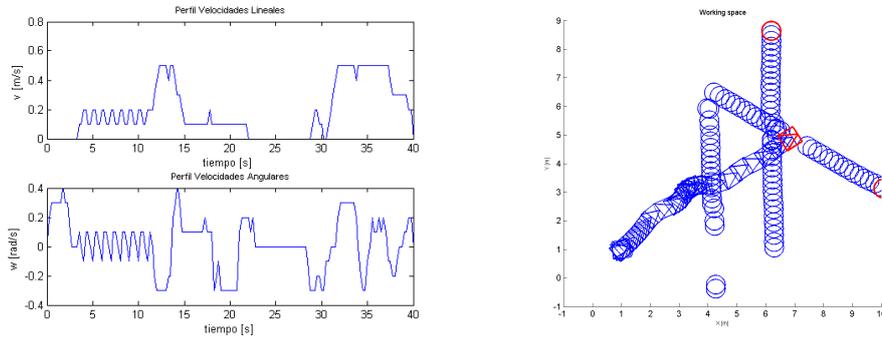


Figura E.4: Búsqueda con *freemotion* $\alpha_{dist} = 0$, $\alpha_v = 1$, $\alpha_s = 1$

Ambos tipos de actuaciones son demasiado pasivas, el robot frena en lugar de realizar maniobras activas de evitación, lo que le puede llevar a situaciones de atrapamiento. La figura muestra todas las maniobras activas. Realiza una maniobra que en este caso resulta más lenta que la de *freemotion*, pero es válida de forma más general, como se demuestra en los ejemplos siguientes.

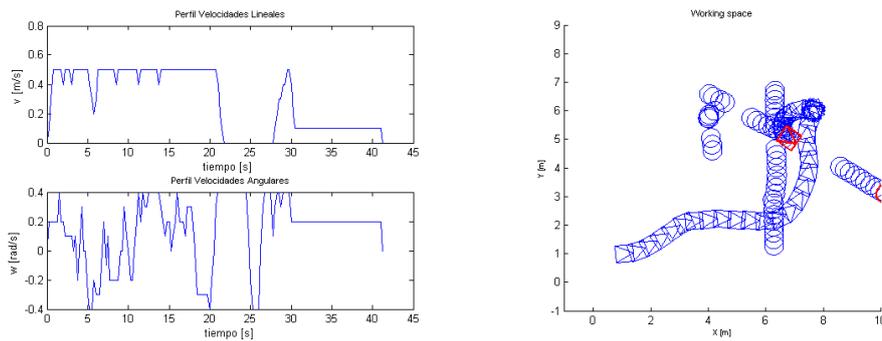
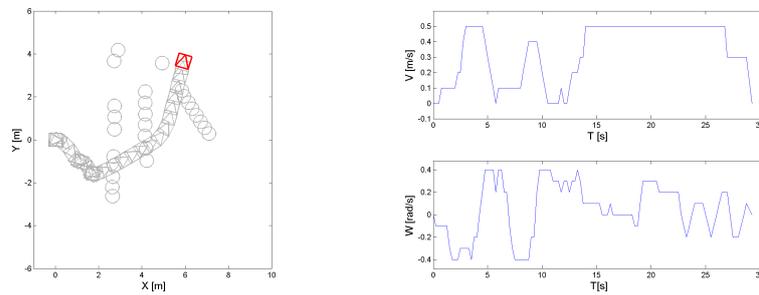
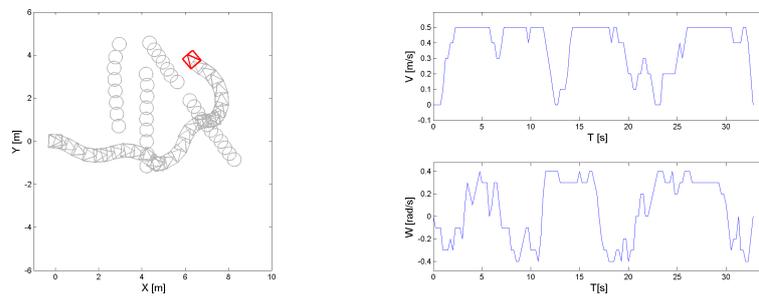


Figura E.5: Búsqueda con todos los parámetros activos, $\alpha_{dist} = 0.5$, $\alpha_v = 1$, $\alpha_s = 1$

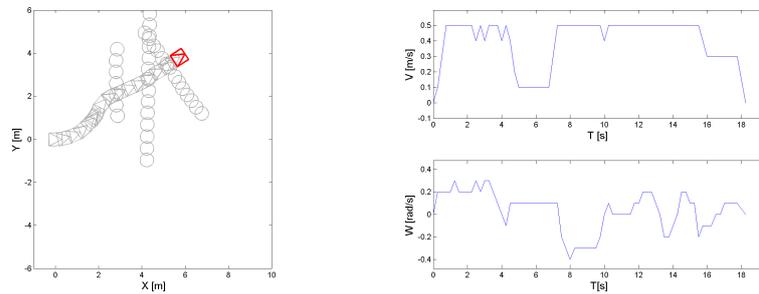
A continuación en las figuras E.6 y E.7 se muestran dos baterías de experimentos sobre escenarios similares en los que se comparan diferentes ponderaciones de los coeficientes de coste. Se muestra cómo aunque la estrategia de *freemotion* en ocasiones es más rápida, no garantiza trayectorias seguras en diferentes escenarios para los mismos coeficientes de seguridad. En la figura E.7b el coeficiente de seguridad es bajo y conduce a una colisión frontal, para unos valores que el caso E.6c habían dado un buen resultado.



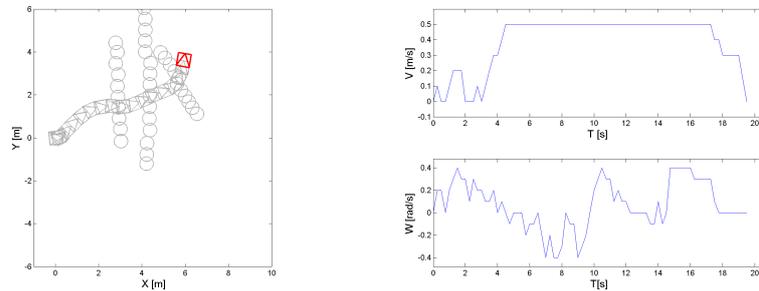
(a) Búsqueda sin guía de velocidades, $\alpha_{dist} = 1$, $\alpha_v = 0$, $\alpha_s = 0.5$



(b) Búsqueda sin guía de velocidades, $\alpha_{dist} = 1$, $\alpha_v = 0$, $\alpha_s = 1$

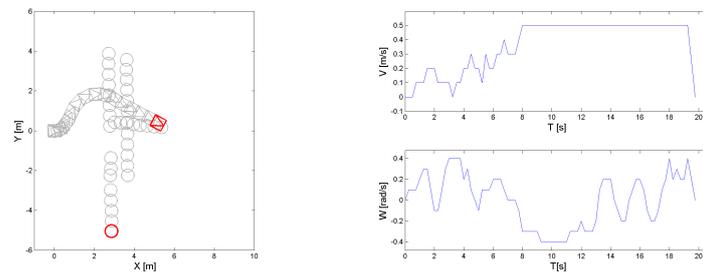


(c) Búsqueda con *freemotion*, $\alpha_{dist} = 0$, $\alpha_v = 1$, $\alpha_s = 0.5$

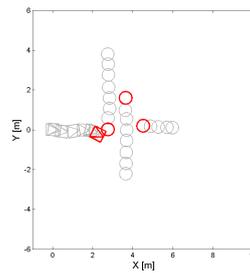


(d) Búsqueda con todo activo, $\alpha_{dist} = 0.5$, $\alpha_v = 1$, $\alpha_s = 1$

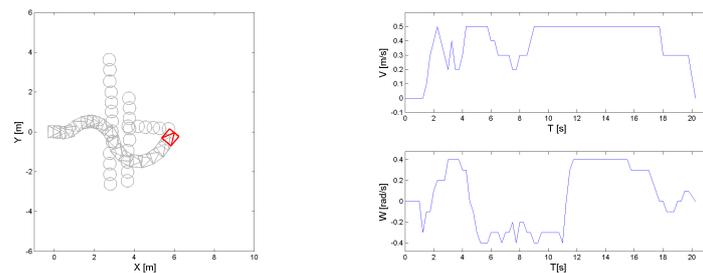
Figura E.6: Comparaciones coeficientes I



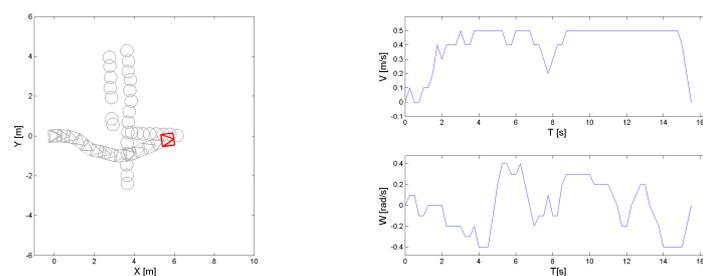
(a) Búsqueda sin guía de velocidades, $\alpha_{dist} = 1$, $\alpha_v = 0$, $\alpha_s = 1$



(b) Búsqueda con *freemotion*, $\alpha_{dist} = 0$, $\alpha_v = 1$, $\alpha_s = 0.5$



(c) Búsqueda con *freemotion*, $\alpha_{dist} = 0$, $\alpha_v = 1$, $\alpha_s = 4$



(d) Búsqueda con todo activo, $\alpha_{dist} = 0.5$, $\alpha_v = 1$, $\alpha_s = 1$

Figura E.7: Comparaciones coeficientes II

Índice de figuras

| | |
|--|----|
| 2.1. Diferentes espacios | 7 |
| 2.2. Espacio de velocidades tiempo | 9 |
| 2.3. Tiempos de cálculo de una superficie según el tamaño de celda | 10 |
| 2.4. Mallado generado entre dos pares de velocidades consecutivos | 12 |
| 3.1. A*: expansión y generación de <i>Abiertas</i> y del árbol de expansión | 14 |
| 3.2. Estructura del algoritmo A* | 15 |
| 3.3. operadores de expansión | 16 |
| 3.4. Término heurístico de velocidad h_v | 19 |
| 3.5. Término heurístico de cercanía en el espacio de trabajo h_{dist} | 20 |
| 3.6. Término heurístico de seguridad h_s | 21 |
| 4.1. Escenario comparativo de métodos de selección de subobjetivos | 24 |
| 4.2. Figura comparativa de dos tamaños angulares | 26 |
| 4.3. Ordenación de subobjetivos | 27 |
| 4.4. Figura obtenida del artículo [Montés and Tornero(2004)] en la que se muestra la combinación de cuatro clotoides simétricas para el cambio de carriles | 28 |
| 4.5. Esquema adelantamiento | 28 |
| 4.6. Ejemplos de la selección de estrategias en el adelantamiento | 29 |
| 5.1. Módulos ejecutados en cada iteración del programa | 32 |
| 6.1. Escenas de la simulación <i>Cruce</i> | 34 |
| 6.2. <i>Carriles</i> : Perfil de Velocidades | 34 |
| 6.3. Escenas de la simulación <i>Caos</i> | 35 |
| 6.4. <i>Caos</i> : Perfil de Velocidades | 36 |
| 6.5. Escenas de la simulación <i>Interiores</i> | 37 |
| 6.6. <i>Interiores</i> : Perfil de Velocidades | 37 |
| 6.7. <i>Adelantamiento</i> : Perfil de velocidades | 38 |
| 6.8. <i>Adelantamiento</i> : Evolución | 39 |
| 7.1. <i>Hall</i> : Escenario | 42 |
| 7.2. <i>Hall</i> : Perfil de Velocidades | 43 |
| 7.3. Tomas de la prueba del <i>Hall</i> | 43 |

| | | |
|-------|---|----|
| 7.4. | <i>Adelantamiento real</i> : Perfil de Velocidades | 44 |
| 7.5. | <i>Adelantamiento real</i> : Escenario | 44 |
| 7.6. | Tomas de la prueba de <i>Adelantamiento real</i> | 45 |
| A.1. | Esquema del Árbol de Toma de Decisiones | 51 |
| A.2. | Figura obtenida del artículo [Montés and Tornero(2004)] en la que se muestra la combinación de cuatro clotoides simétricas para el cambio de carriles | 52 |
| A.3. | Distancias y condiciones de seguridad durante la maniobra adelantamiento | 53 |
| B.1. | Láser: clasificación de puntos estáticos (rojos) y dinámicos (azules) . . . | 57 |
| B.2. | Figura obtenida de [Lorente(2011)] en la que se muestra el engorde de los segmentos para el cálculo de las superficies estáticas | 58 |
| B.3. | Láser: Identificación del patrón pierna | 59 |
| C.1. | <i>Carriles</i> I | 62 |
| C.2. | <i>Carriles</i> II | 63 |
| C.3. | <i>Caos</i> I | 65 |
| C.4. | <i>Caos</i> II | 66 |
| C.5. | <i>Interiores</i> : primer tramo | 68 |
| C.6. | <i>Interiores</i> : segundo tramo | 69 |
| C.7. | <i>Interiores</i> : tercer tramo | 70 |
| C.8. | <i>Interiores</i> : cuarto tramo | 71 |
| C.9. | Adelantamiento: I | 73 |
| C.10. | Adelantamiento: II | 74 |
| C.11. | Adelantamiento: III | 75 |
| D.1. | <i>Hall</i> : I | 78 |
| D.2. | <i>Hall</i> : II | 79 |
| D.3. | <i>Hall</i> : III | 80 |
| D.4. | Adelantamiento real: I | 82 |
| D.5. | Adelantamiento real: II | 83 |
| D.6. | Adelantamiento real: III | 84 |
| D.7. | Adelantamiento real: IV | 85 |
| E.1. | Robot Pioneer 3AT | 87 |
| E.2. | Límites del espacio de velocidades real | 88 |
| E.3. | Búsqueda sin guía de velocidades, $\alpha_{dist} = 1$, $\alpha_v = 0$, $\alpha_s = 4$ | 89 |
| E.4. | Búsqueda con <i>freemotion</i> $\alpha_{dist} = 0$, $\alpha_v = 1$, $\alpha_s = 1$ | 90 |
| E.5. | Búsqueda con todos los parámetros activos, $\alpha_{dist} = 0.5$, $\alpha_v = 1$, $\alpha_s = 1$ | 90 |
| E.6. | Comparaciones coeficientes I | 91 |
| E.7. | Comparaciones coeficientes II | 92 |

Bibliografía

- [Alglib()] Alglib. <http://www.alglib.net/>.
- [Brock and Khatib(1999)] Oliver Brock and Oussama Khatib. High-speed navigation using the global dynamic window approach. In *In IEEE International Conference on Robotics and Automation*, pages 341–346, 1999.
- [D. Castro and Ruano(2002)] U. Nunes D. Castro and A. Ruano. Reactive local navigation. In *IEEE 28th Annual Conference on Industrial Electronics Society*, pages 2427–2432, 2002.
- [Fiorini and Shillert(1998)] Paolo Fiorini and Zvi Shillert. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17:760–772, 1998.
- [Fox and Thrun(1997)] Burgard Fox and Thrun. The dynamic window approach to collision avoidance. In *Robotics & Automation Magazine, IEEE*, volume 4 (1), pages 13–23, 1997.
- [Fraichard(1999)] Th. Fraichard. Trajectory planning in a dynamic workspace: a ‘state-time space’ approach. In *Advanced Robotics, 13 (1)*, pages 75–94, 1999.
- [Gal and Shiller(2009)] Oren Gal and Zvi Shiller. Mapping obstacles to collision states for on-line motion planning in dynamic environments. In *Workshop on Safe navigation of Autonomous vehicles, ICRA, IEEE, Japan*, 2009.
- [J. Mínguez and Alami(2001)] N. Simeon J. Mínguez, L. Montano and R. Alami. ”global nearness diagram navigation (gnd)”. In *IEEE International Conference on Robotics and Automation (ICRA ’01)*, 2001.
- [Latombe(1991)] Jean-Claude Latombe. *Robot Motion Planning*. 1991.
- [LaValle and Kuffner(2001)] Steven M. LaValle and James J. Kuffner. Randomized kinodynamic planning. In *The International Journal of Robotics Research*, volume 20 (5), pages 378–400, 2001.
- [Lorente(2011)] M^aTeresa Lorente. Planificación de la navegación de robots en entornos dinámicos. In *PFC Universidad de Zaragoza*, 2011.

- [Montés and Tornero(2004)] N. Montés and J. Tornero. Lane changing using s-series clothoidal approximation and dual-rate based on bezier points to controlling vehicle. In *Proceedings of the 4th WSEAS International Conference on Systems Theory and Scientific Computation*, 2004.
- [Montesano and Montano(2005a)] J. Montesano, L. Minguez and L. Montano. Probabilistic scan matching for motion estimation in unstructured environments. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, 2005a.
- [Montesano and Montano(2005b)] J. Montesano, L. Minguez and L. Montano. Modeling the static and the dynamic parts of the environment to improve sensor-based navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 2005b.
- [Owen and Montano(2005)] E. Owen and L. Montano. Motion planning in dynamic environments using the velocity space. In *RSJ International Conference on Intelligent Robots and Systems (IROS'2005)*, pages 997–1002, Edmonton, Alberta, Canada, 2005.
- [Owen and Montano.(2006)] E. Owen and L. Montano. A robocentric motion planner for dynamic environments using the velocity space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.
- [PlayerStage()] PlayerStage. <http://playerstage.sourceforge.net/>.
- [Stachniss and Burgard(2002)] Cyrill Stachniss and Wolfram Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *IN IEEE-RSJ International Conference on Intelligent Robots and Systems*, pages 508–513, 2002.
- [Thrun and Fox(2005)] Burgard Thrun and Fox. *Probabilistic Robotics*. 2005.
- [Urcola and Montano(2011)] P. Urcola and L. Montano. Adapting robot team behavior from interaction with a group of people. In *Intelligent Robots and Systems, 2011. IROS 2011. IEEE/RSJ International Conference on*, 2011.