
CAPITULO 3.

Planificación de Caminos Mediante Grafos de Visibilidad.

3.1. Introducción.

La planificación se define como la búsqueda de una ruta libre de obstáculos desde una posición inicial hasta otra final a través del entorno de trabajo del robot móvil. Esta operación se realiza mediante el uso de la información que se posee del entorno actualmente, la descripción de la tarea de navegación y algún tipo de metodología estratégica. Así, el planificador se define por el modelo del entorno y el algoritmo de búsqueda utilizado. El caso más simple consiste en considerar un entorno conocido en su totalidad, estático y modelado de forma geométrica mediante polígonos. Con estas apreciaciones resulta factible la aplicación un algoritmo de búsqueda en grafos, que emplee cierta función de coste para la obtención de la ruta. Sin embargo, la aplicación directa de esta metodología para utilizar la ruta resultante como camino que se debe seguir, implica efectuar ciertas consideraciones acerca del vehículo. De esta manera, se supone un vehículo puntual, omnidireccional, y que recorre los caminos consignados de forma perfecta.

Las simplificaciones mencionadas en el párrafo anterior resultan limitaciones que deberán ser eliminadas por los sucesivos módulos del sistema de navegación. La planificación se constituye como la primera aproximación al problema de la navegación, lo que permite abordar la resolución del mismo de una forma más eficiente. Este hecho se logra por el desacoplo del problema geométrico de encontrar una ruta libre de obstáculos, de la cuestión de admisibilidad cinemática y dinámica del camino. De esta forma, la finalidad de la planificación consiste en la resolución del problema geométrico de localizar una ruta libre de obstáculos, definida como una secuencia ordenada de objetivos, que conduzca al robot desde la posición inicial hasta la final. Esto se efectúa sin considerar que la ruta generada posibilite su seguimiento por parte de un robot móvil que no cumpla alguna de las restricciones mencionadas al final del párrafo anterior.

En este capítulo se aborda el problema de la planificación en el plano, ya que posee entidad propia como para ser estudiado, y sin olvidar que la gran mayoría de las aplicaciones de robots móviles se basan en la resolución de la mencionada cuestión. De este modo, el desarrollo de este capítulo comienza con la formalización de los grafos de visibilidad como herramienta para la búsqueda en el plano de una ruta continua en posición (apartado 3.2.). La elección de esta metodología de planificación se fundamenta en la posibilidad de empleo de algoritmos de bajo coste computacional con respecto al resto de los procedimientos expuestos en el capítulo anterior. Una variación de este concepto de menor complejidad son los subgrafos de visibilidad, los cuales se pueden utilizar en entornos conocidos de forma parcial para encontrar una solución (posiblemente no óptima) al problema planteado (apartado 3.3.). Ambos tipos de grafos poseen el gran inconveniente de considerar robots puntuales para la realización de su cometido. Por ello, a causa de esta consideración poco realista, un paso esencial para la aplicación de estos conceptos en aplicaciones reales se constituye con la eliminación esta restricción (apartado 3.4.). Finalmente se expondrán una serie de conclusiones de los aspectos más relevantes tratados en este capítulo (apartado 3.5.).

3.2. Construcción de la función ruta mediante grafos de visibilidad.

Existen diversos enfoques para la definición de una función ruta () que lleve al robot desde su posición origen hasta la posición final. Todos ellos pretenden la definición de una ruta segura para el vehículo que garantice que: no colisionar con algún obstáculo, y que no viole las restricciones cinemáticas y dinámicas impuestas por la estructura física del robot. No obstante, según el planteamiento expuesto en la introducción de esta tesis, esta cuestión se soluciona en tres etapas: la resolución del problema geométrico, del cinemático y por último del dinámico. El primero de ellos, que se aborda en esta sección, consiste en la búsqueda de una función ruta libre de obstáculos que cumpla la condición de continuidad en posición. Para resolver este problema se emplea una herramienta denominada *grafos de visibilidad*.

Los grafos de visibilidad (Nilsson, 1.969) proporcionan un enfoque geométrico para solventar el problema de la planificación. Este método se encuentra muy extendido debido a que opera con modelos poligonales de entorno, con lo que existen algoritmos que construyen esta clase de grafos con un coste computacional relativamente bajo: $O(n^3)$ (Lozano-Pérez y Wesley;1.979). Este método necesita modelos de entornos definidos con polígonos, y puede trabajar tanto en el plano como en el espacio. En este apartado sólo se formaliza el primer tipo, ya que el segundo queda fuera del ámbito de esta tesis.

Un grafo de visibilidad GV es un grafo no dirigido, que se define por el par (N, E) :

- i) Donde, N resulta un conjunto de nodos formado por la configuración inicial q_a , la configuración final q_f y los vértices de los obstáculos que pertenecen al conjunto B .
- ii) Y la función conectado definida de $N \times N \rightarrow \{0, 1\}$ es no nula si y solo si los dos nodos referenciados se encuentran conectados. Se dice que dos nodos están conectados si y solo si se puede trazar un segmento que los una, de forma que resulte una arista de un obstáculo de B , o bien yazca por completo en el espacio libre del entorno C_f .

De este modo, dos nodos están conectados si y solo si son “*visibles*”, es decir, se puede alcanzar el segundo nodo desde el primero (o viceversa) al seguir la línea recta que los une, sin interceptar algún obstáculo del entorno. También se consideran *visibles* si el segmento que une los dos nodos yace sobre una arista del polígono que modela a un obstáculo.

El algoritmo de planificación basado en grafos de visibilidad constará de dos fases fundamentales: una primera, de construcción del grafo; y una segunda, de búsqueda. Esta última utilizará un algoritmo de búsqueda en grafos (Nilsson; 1.987) para encontrar una ruta desde q_a a q_f siguiendo los arcos del mismo. La ruta consiste en la sucesión de nodos por los cuales deberá pasar el robot al seguir los arcos, para llegar a la configuración final q_f partiendo desde la de inicio q_a . Así, esta se define por un conjunto ordenado de nodos del grafo, como se expresa en la ecuación (3.1).

$$G = \{g_1, \dots, g_p / g_i \in N\} \quad g_1 = q_a; g_p = q_f \quad (3.1)$$

donde el nodo g_i será visitado antes que g_j si y solo si $i < j$.

La formulación de ruta expresada (3.1) proporciona el conjunto de *puntos elegidos* en virtud de los cuales se construye el camino. La especificación de la función ruta $S: [0, p-1]$, inherente a dicha definición, se verifica como una función definida a intervalos:

$$S(i) = \begin{cases} S_1(\cdot) & \text{si } 0 \leq i < 1 \\ \dots & \dots \\ S_{p-1}(\cdot) & \text{si } p-2 \leq i < p-1 \end{cases} \quad (3.2)$$

donde las funciones S_i se definen merced a dos nodos consecutivos del conjunto G . Cada función S_i es un segmento de la ruta, por ello, si el conjunto G consta de p elementos, existirán $p-1$ segmentos. La única restricción que debe cumplir la ruta resulta la continuidad en posición; por ello, un segmento de la misma se define de la siguiente forma:

$$S_i(\mathbf{g}) = (1 - \alpha_i)g_i + \alpha_i g_{i+1} \quad \mathbf{g}_i \in G \quad (3.3)$$

La expresión (3.3) asegura que la función sea continua a lo largo de toda la ruta, y al estar ésta construida en base a los arcos del grafo de visibilidad, se garantiza la pertenencia del conjunto de configuraciones construido a partir de \mathbf{g}_i al espacio libre C_f .

$$q_j = S_i(\mathbf{g}_j) \quad q_j \in C_f \quad (3.4)$$

3.3. Planificación basada en subgrafos de visibilidad.

El algoritmo de planificación descrito en el apartado anterior desempeña su función en entornos conocidos en su totalidad, en el cual los obstáculos se encuentran modelados como polígonos. Otros autores (Rao y otros; 1.985) proponen una variación de la metodología basada en grafos de visibilidad que puede usarse con entornos en parte especificados.

El algoritmo basado en subgrafos de visibilidad tiene como principal premisa el desconocimiento, por parte del robot, del entorno en el cual efectuará su tarea. Por ello, no encuentra una ruta óptima (si existe) sino una solución admisible. La principal ventaja que ofrece consiste en la integración de las dos fases que componen la solución basada en grafos de visibilidad (creación del grafo y búsqueda de la ruta) en una sola, con la consiguiente disminución del coste de la complejidad. Para ciertas aplicaciones en las que se requieran rápidas replanificaciones de la ruta sin importar encontrar el camino óptimo, resulta de interés.

Se caracteriza por no generar el grafo completo, sino el subgrafo necesario para encontrar el objetivo. Suponiendo que el robot se encuentra en el nodo inicial, desde éste elige de la gama de nodos visibles aquel que posea la distancia euclídea menor con respecto al nodo final. El nodo de partida y el nodo elegido se marcan como visitados, y se introducen en la ruta solución. En el caso de que el nodo actual no posea nodos visibles y tampoco sea el nodo meta, se retrocederá al nodo anterior y se seleccionará, entre los visibles, el más cercano al nodo final que no esté visitado.

En la figura 3.1 se muestra como se encuentra el camino desde la configuración inicial q_a hasta la final q_f .

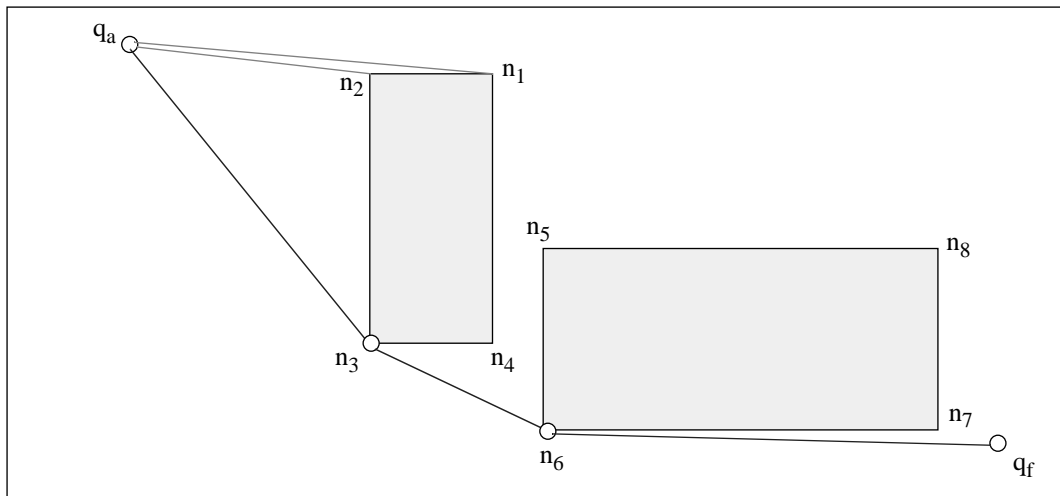


Figura 3.1. Algoritmo de navegación en entornos desconocidos.

Partiendo de q_a , el conjunto de nodos visibles está formado por $\{n_1, n_2, n_3\}$, que coincide con los nodos que pueden ser unidos en un grafo de visibilidad con q_a . Si n_3 es el nodo más cercano a q_f , este será el nuevo nodo que debe expandirse según la metodología anterior. Con ello, se llega a n_6 y desde este se alcanza q_f . Como se puede observar, el algoritmo posee un comportamiento eficaz sobre entornos sencillos. Por ello, puede considerarse de interés en ciertos tipos de aplicaciones en las que se disponga de un robot con telémetro. Dicho sensor debe estar capacitado para medir las distancias a las esquinas de los objetos que se encuentren en el entorno de forma precisa, si se desea aplicar con éxito este método.

Existen fuertes restricciones para llevar este algoritmo a la práctica de la forma descrita, sin embargo resulta posible utilizar una versión modificada como algoritmo de planificación de rutas admisibles. En algunos casos mejora la relación existente entre el cálculo de la ruta óptima y el tiempo empleado en localizarla, ante la utilización conjunta de un algoritmo de generación de grafos de visibilidad y búsqueda heurística tipo A^* . La descripción de una versión del algoritmo que realiza la planificación expuesta en los párrafos anteriores se desarrolla a continuación.

Las estructuras empleadas en el algoritmo forman dos conjuntos: *visitados* y *camino*, los cuales almacenan los nodos evaluados y el camino actual respectivamente. Por otro lado, con ambos conjuntos se asocian las operaciones *introducir* y *sacar*, definidas a continuación.

Supuesto un conjunto L de m elementos formado por una cabecera a y un conjunto cola C :

$$L = \{a|C\} / a = n_1; C = \{n_2, \dots, n_m\} \quad (3.5)$$

entonces

$$\text{introducir}(n, L) \quad L = \{n|L\} \quad n = \text{sacar}(L) \quad L = \{n_2, \dots, n_m\}; n = n_1 \quad (3.6)$$

Por otro lado, se precisa la operación *visibles*, que presenta un nodo como parámetro y devuelve la lista de nodos visibles desde el mismo.

$$\text{visibles}(n) = \{n_i \mid N / (n, n_i) = 1\} \quad (3.7)$$

Y por último la función *mas_cercano* devuelve el nodo más próximo de una lista L , según la distancia euclídea, a otro nodo dado.

$$n = \text{mas_cercano}(L, n_f) \quad n \in L \quad n_i \in L \quad \text{MIN}(\|n, n_f\|) \quad (3.8)$$

```

na=qa;
nf=qf;
visitados={};
camino={};
Función Búsqueda(na,nf)
  L=visibles(na);
  Si (nf ∈ L) entonces
    camino=introducir(nf{na|camino});
    Retorna(camino);
  sino
    Si ( (ni ∈ L) ∧ ni ∉ visitados) entonces
      introducir(na,camino);
      n=mas_cercano(L,nf);
      introducir(na,visitados);
      Búsqueda(n,nf);
    sino
      Si (camino=={}) entonces
        Retorna(no_camino);
      sino
        n=sacar(camino);
        Búsqueda(n,nf);
      Fin Si
    Fin Si;
  Fin Si;
Fin Búsqueda;

```

El algoritmo presentado resulta el esquema general para realizar una planificación desde el nodo n_a hasta el n_f . Efectúa una búsqueda de un camino que, con probabilidad no será el óptimo, a la vez que genera un subgrafo de visibilidad que ayuda a encontrar la solución. Para mejorar la eficacia, se puede proceder a memorizar en un conjunto los nodos visibles desde otro dado, a la vez que se almacena la ruta actual. Así se disminuye la complejidad algorítmica de la función *visibles*. Esto no aparece representado en el algoritmo expuesto por razones de claridad.

3.4. Eliminación de la restricción de un robot puntual.

La metodología empleada, basada en grafos de visibilidad, implica que pueden formar parte de la ruta solución tanto vértices como aristas de los obstáculos que conforman el modelo del entorno. Debido a esto, algunos autores definen la ruta construida mediante la utilización de grafos de visibilidad semi-libre de obstáculos (Latombe, 1.991).

En este apartado se expone un procedimiento que, aplicado al modelo de entorno con anterioridad a realizar la planificación, permite realizar esta última labor con la certeza de que la ruta obtenida se verá libre de obstáculos. El método se basa en la construcción de un entorno sobredimensionado a partir del original, sobre el cual se puede aplicar de la forma descrita la planificación basada en grafos de visibilidad.

La idea fundamental se muestra en la figura 3.2, donde la ruta definida por la secuencia ordenada de nodos $G=\{g_1, g_2, g_3, g_4, g_5\}$, ha sido planificada mediante el uso de grafos de visibilidad y una versión sobredimensionada del modelo del entorno. La obtención de este nuevo modelo implica la utilización de un factor de expansión que garantice que la ruta planificada resulte segura. En la mencionada figura se considera un arquetipo de robot circular, por lo que se ha utilizado un factor de expansión igual al radio de la circunferencia que lo modela. Por tanto, se procede a la construcción de una ruta que impide la colisión del robot con algún obstáculo del entorno.

En este epígrafe se comenta el método propuesto para llevar a cabo las modificaciones oportunas sobre el modelo del entorno para obtener la nueva versión del mismo sobre la que trabaja el algoritmo de planificación. Así, en primer lugar se desarrollan las consideraciones previas; en segundo lugar se describe el método utilizado; y por último se ofrecen una serie de comentarios sobre el modelado circular de robots móviles.

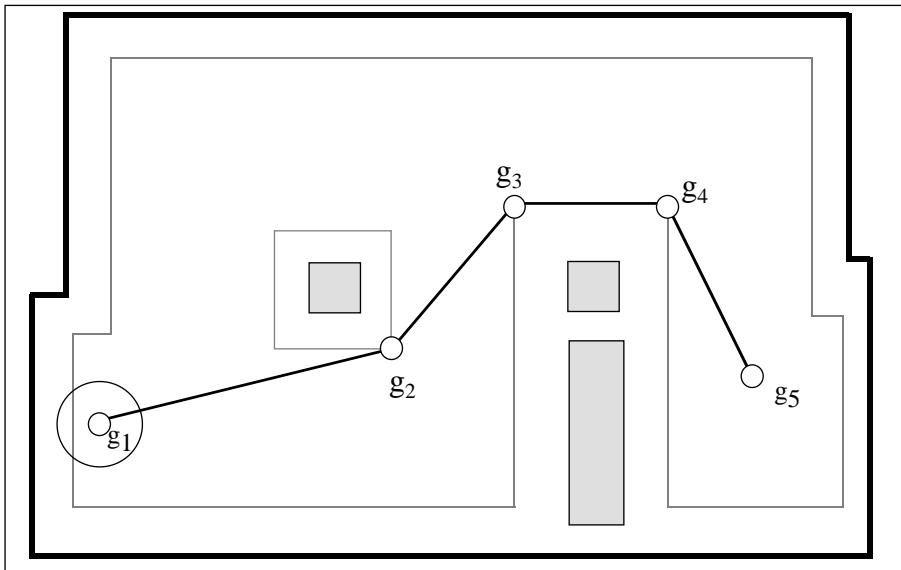


Figura 3.2. Realización de la planificación en un entorno sobredimensionado.

3.4.1. Concepto de entorno expandido.

La restricción impuesta de un robot puntual evita el problema de la construcción de rutas semi-libres de obstáculos de manera que la hace segura. Sin embargo, ya que ésta se presenta como una estimación irreal, el método basado en grafos de visibilidad tal y como se ha expuesto resulta inaplicable. La forma más adecuada de eliminar esta grave dificultad consiste en utilizar una versión *expandida* del entorno, es decir, considerar en la planificación un modelo sobredimensionado de los obstáculos que componen el entorno. De este modo, se añade un parámetro adicional a la hora de realizar la planificación: el *factor de expansión del entorno*. La correcta evaluación de este parámetro resulta de especial importancia, ya que de él depende la capacidad de planificar una ruta segura que evite cualquier contacto del vehículo con un elemento del entorno. En principio, teniendo en cuenta sólo consideraciones geométricas, se puede definir un robot modelado mediante un círculo de radio r . En este caso, el *factor de expansión* se califica como dicho radio. Sin embargo, no sólo confluyen consideraciones geométricas, sino también de incertidumbre en la posición, sobre la que se presta atención en el apéndice A de la presente tesis.

Antes de realizar la expansión del entorno, es preciso considerar que su modelo se compone de dos clases de cuerpos: los *obstáculos* y el *polígono límite* del entorno. El primero de estos conjuntos se refiere a los cuerpos que se encuentran en un determinado entorno de trabajo y definidos según el conjunto B , mientras que el segundo es el polígono P que define los límites físicos del espacio de trabajo del robot, y por tanto, especifica los

confines del espacio de configuraciones C . El tratamiento que se debe efectuar en cada uno de estos elementos es diferente, ya que, mientras que los obstáculos se expanden hacia fuera, el *polígono límite* sufre una contracción (Figura 3.3.).

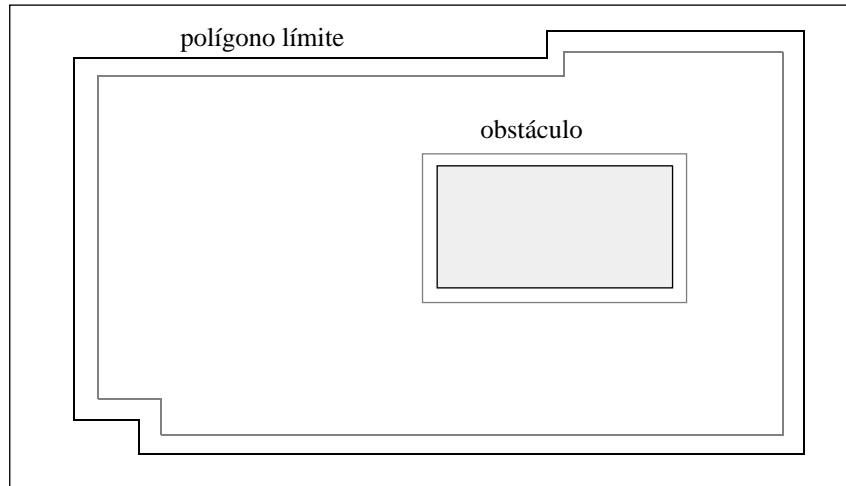


Figura 3.3. Versión expandida de un entorno.

3.4.2. Algoritmo de expansión del entorno.

Al considerar las dos clases de componentes que posee el modelo del entorno, el algoritmo de expansión consiste en aplicar el tratamiento adecuado a cada uno de las partes integrantes, para de forma posterior unirlos en la versión expandida. Esta acción se realizará en tres fases:

- i) Expansión de obstáculos y contracción del polígono límite.
- ii) Cálculo del nuevo polígono límite.
- iii) Cálculo del nuevo conjunto de obstáculos.

Estas etapas se reflejan en la figura 3.4. En ella se plantea cómo con un entorno cualquiera (Figura 3.4.a) se realiza la primera fase del tratamiento expuesto. El resultado obtenido resulta en el entorno dibujado con línea discontinua. A continuación se construye el nuevo polígono límite (Figura 3.4.b) definido como la unión entre el antiguo polígono contraído y los obstáculos que lo interceptan. Por último (Figura 3.4.c) se muestra cómo se construye el nuevo conjunto de obstáculos mediante la unión de aquellos obstáculos que presentan alguna intersección.

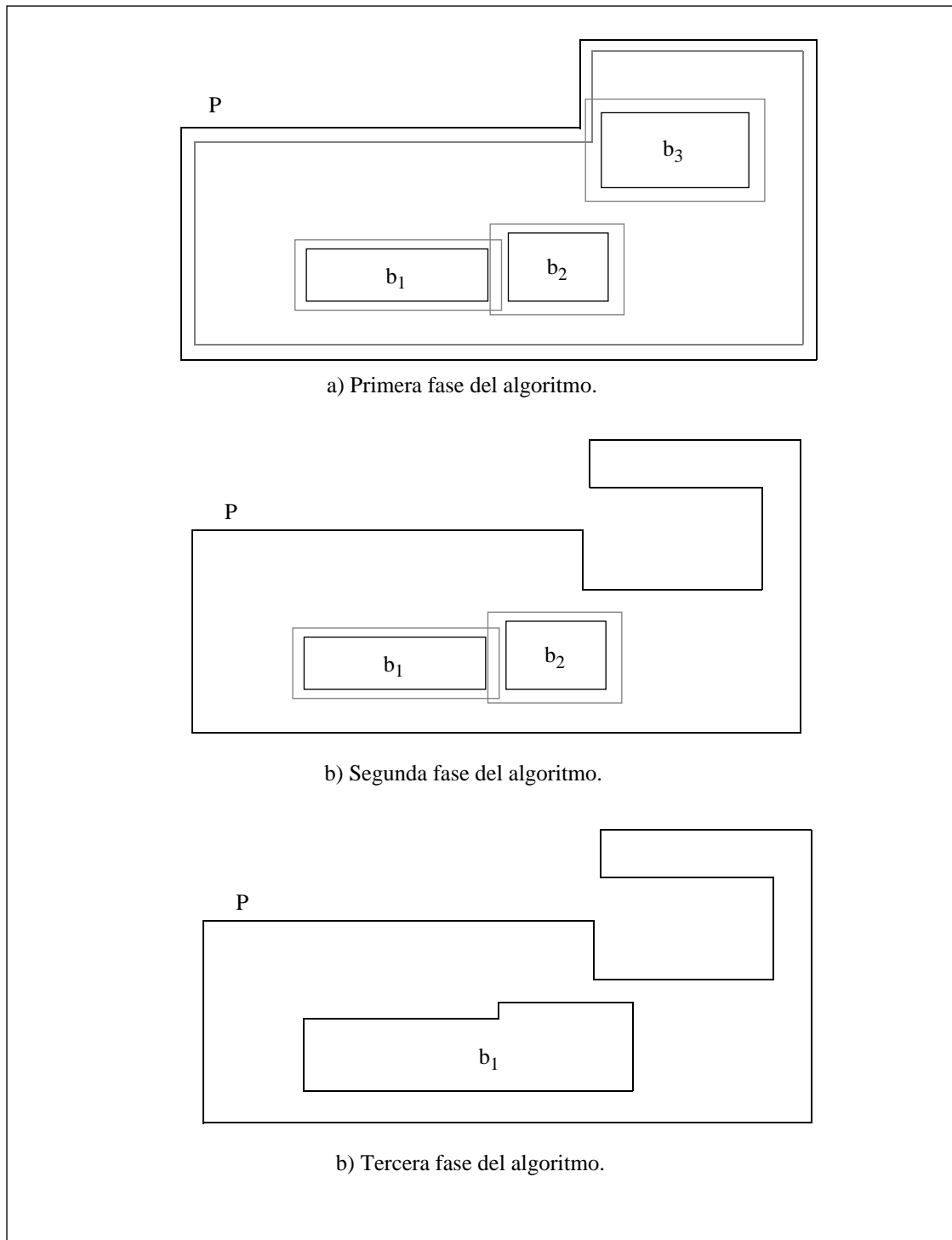


Figura 3.4. Cálculo de la versión expandida de un entorno dado.

De este modo, se verifica que el nuevo entorno consta de un número de obstáculos inferior o igual al original, lo que supone cierta ventaja a la hora de la construcción del grafo de visibilidad.

En el segundo capítulo quedó definido el concepto de obstáculo mediante establecimiento un conjunto de configuraciones prohibidas en C . En el caso de que un obstáculo b_i se modele mediante un polígono bp_i , el conjunto $b_i(q)$ se determinará como el conjunto de configuraciones que están situadas en el interior de bp_i .

Dado un obstáculo b_i , el polígono bp_i que lo modela quedará definido como un conjunto ordenado de vértices.

$$bp_i = \{^i p_1, ^i p_2, \dots, ^i p_{ni}\} \text{ donde } ^i p_j = (x_j, y_j) \quad (3.9)$$

De esta manera las aristas $^i a_j$ que componen el polígono resultan:

$$A_i = \{^i a_1, ^i a_2, \dots, ^i a_{ni}\} \text{ donde } ^i a_j = (^i p_j, ^i p_{j+1}) \text{ para } j=1 \text{ hasta } ni-1 \text{ y } ^i a_{ni} = (^i p_{ni}, ^i p_1) \quad (3.10)$$

Sea $^i v_j$ un vector perpendicular a $^i a_j$ y con sentido hacia el interior del polígono bp_i . Por tanto, se establece que una configuración $q_k = (p_k, c)$ pertenece al conjunto $b_i(q)$ cuando para alguna arista $^i a_j$ se cumple lo siguiente:

$$() \quad p + ^i v_j = p_k \quad 0 \quad c \quad (3.11)$$

en donde p es un punto genérico perteneciente a la arista $^i a_j$, y c que representa el valor del parámetro donde la recta corta a la arista más próxima perteneciente al mismo polígono.

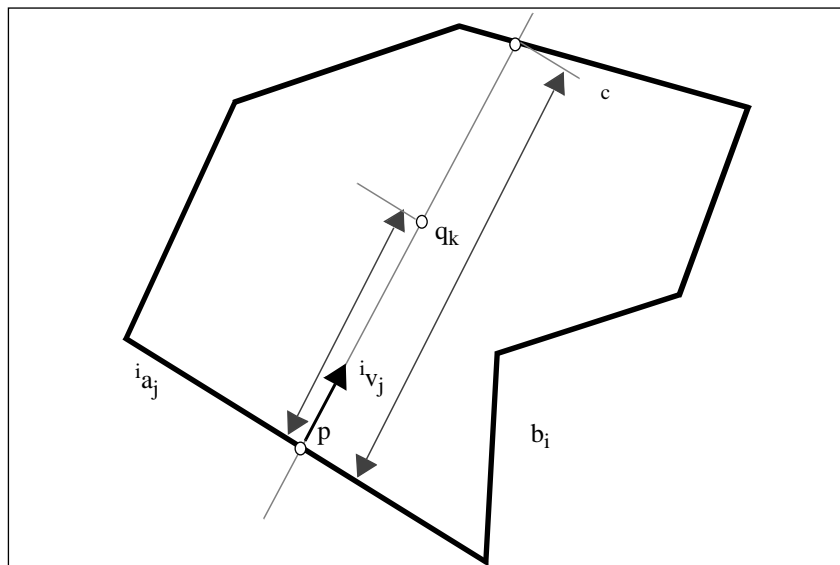


Figura 3.5. Configuración q_k perteneciente a un obstáculo.

3.4.2.1. Expansión de obstáculos y contracción del polígono límite.

La expansión de un obstáculo b_i conlleva añadir un conjunto de configuraciones adicionales $e(b_i)$ a $b_i(q)$, para construir la versión expandida del obstáculo b_i^* .

$$b_i^*(q) = b_i(q) + e(b_i) \quad (3.12)$$

donde $e(b_i)$ es una función que añade las configuraciones inmediatamente próximas a los límites del obstáculo, según refleja la figura 3.3. En el caso de un único obstáculo b_i que esté modelado mediante bp_i , su expansión significa la realización de una serie de modificaciones sobre el polígono que lo modela. Si el polígono es convexo, el algoritmo de expansión resulta fundamentalmente un escalado de sus aristas, proceso que se realiza en dos etapas:

- i) Por cada arista del polígono original construir una recta paralela a cierta distancia hacia el exterior.
- ii) Cálculo del punto de corte entre dos rectas pertenecientes a dos aristas consecutivas del polígono original. El conjunto de puntos de corte entre rectas obtenido serán los vértices del nuevo polígono.

Sin embargo, con polígonos no convexos, el algoritmo anterior resulta incompleto, por lo que se precisa incorporar un paso adicional. La problemática que conlleva realizar la expansión se refleja en la figura 3.6.

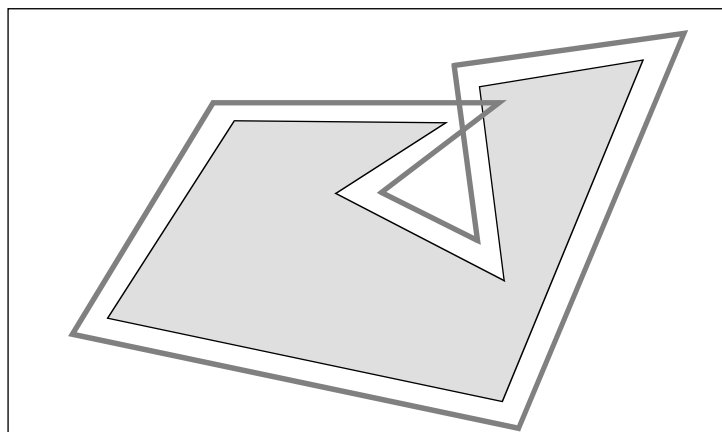


Figura 3.6. expansión de un polígono no convexo.

En dicha figura, se aprecia cómo la expansión significa añadir un conjunto de configuraciones adicionales que no están contempladas por la función $e(b_i)$ y que se muestran dentro de la concavidad del obstáculo.

Para la resolución de este problema, una vez efectuado el paso i), en el paso ii) se añaden los puntos de corte entre dos aristas no consecutivas, de forma que se construye un grafo compuesto por dos tipos de nodos: los vértices originales del polígono expandido, y los puntos de corte entre dos aristas no consecutivas. Las aristas se definen como los segmentos del polígono expandido que unen dos nodos.

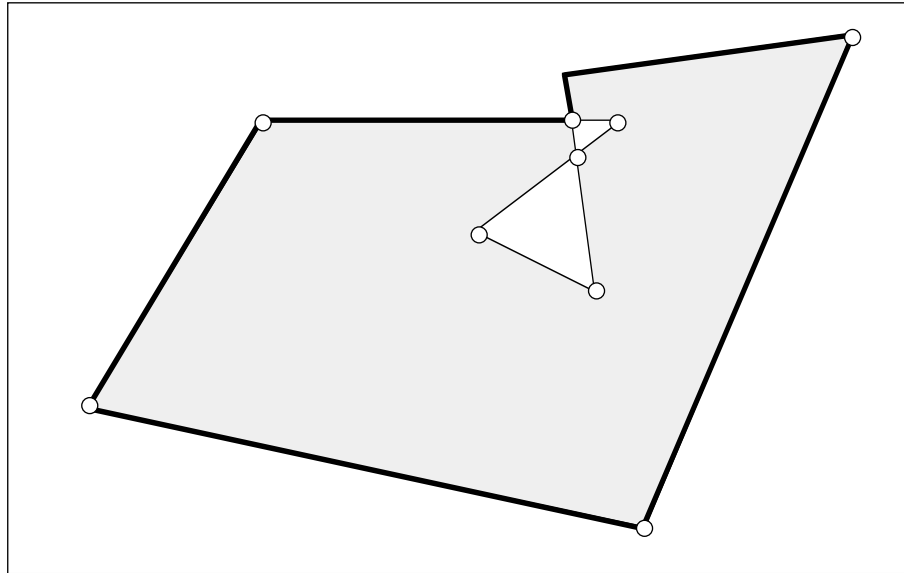


Figura 3.7. Grafo asociado a un obstáculo expandido.

En la figura 3.7, se puede observar cómo el grafo define una serie de polígonos unidos entre sí: uno exterior (punteado), y dos interiores (sin puntear). El área ocupada por el obstáculo expandido se constituye por la unión de las áreas definidas por los distintos polígonos, por tanto, el tercer paso que se incorpora al algoritmo expuesto para polígonos convexos es el cálculo del polígono envolvente que comprende todas las áreas. De este modo:

- iii) Eliminar del grafo del polígono expandido las aristas interiores. Se define una arista interior como aquella en la que al menos uno de los dos vértices que la definen es interior al polígono envolvente.

Este paso se realiza con el criterio (3.11) expuesto para determinar si un punto es interior a un polígono dado. En la figura 3.7 el polígono expandido aparece en trazo grueso.

En la contracción del *polígono límite* se sigue la misma metodología que para la expansión de un obstáculo, sólo que en sentido inverso. Así, en el primer paso la construcción de las rectas paralelas a una determinada arista se realiza a una distancia

hacia el interior del polígono. En este caso generalizado de un entorno modelado mediante un polígono no convexo se calculan los puntos de corte tanto entre dos aristas consecutivas y no consecutivas. En segundo lugar se realiza el tercer paso, que consiste en la elección del polígono que contiene actualmente al robot.

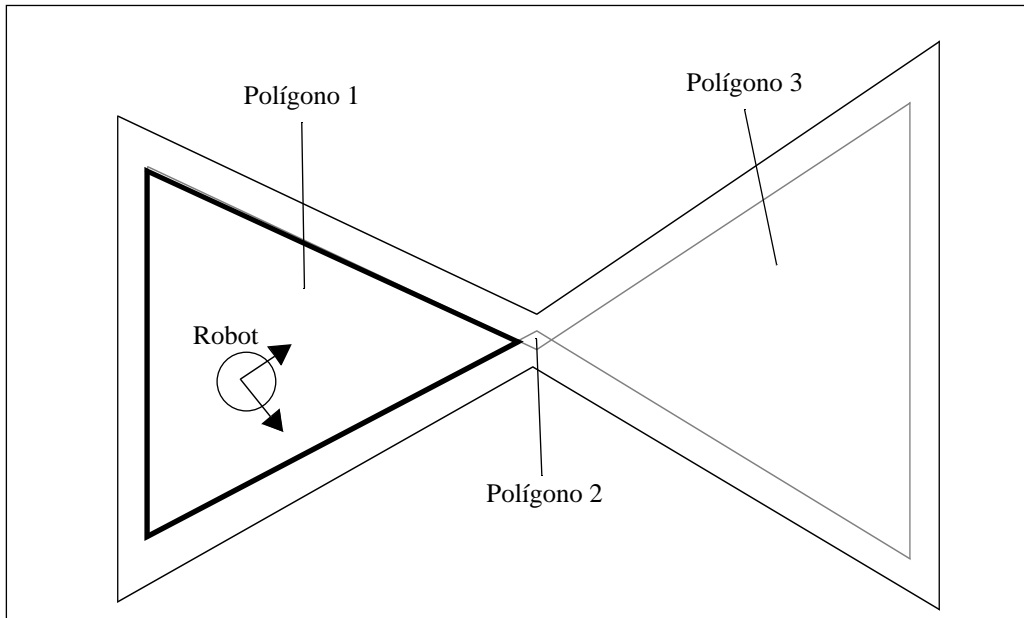


Figura 3.8. Contracción del entorno.

En la figura 3.8 se puede observar cómo la contracción del polígono límite genera tres subpolígonos, numerados del uno al tres. El nuevo entorno resulta del subpolígono en el que se encuentra el robot. Por ello, en la mencionada figura se designa en trazo grueso el *polígono uno* como nuevo polígono límite del entorno.

3.4.2.2. Cálculo del nuevo polígono límite.

Al realizar el primer paso del algoritmo puede ocurrir que algunos de los obstáculos expandidos intercepten el polígono límite contraído. Por ello, se precisa en tal caso la modificación de este último (Figura 3.4.b).

Sea el conjunto $B^* = \{b_1^*, \dots, b_q^*\}$ el conjunto de obstáculos expandidos obtenido a partir de B , se dice que un obstáculo intercepta con el entorno contraído C^* alcanzado a partir de C , cuando se verifica:

$$b_i^*(q) \cap C^* \neq \emptyset \quad (3.13)$$

Ello quiere decir que parte del obstáculo yace en el exterior del espacio de configuraciones contraído. La acción que se debe llevar a cabo cuando se verifique la relación (3.13) por parte de algún obstáculo b_i^* , consiste en eliminar del espacio de configuraciones C^* las ocupadas por el área de dicho obstáculo:

$$C^{**} = C^* - (b_i^*(q) \cap C^*) \quad b_i^*/b_i^*(q) \cap C^* \quad b_i^*(q) \quad (3.14)$$

De este modo, el espacio de configuraciones quedará reducido al conjunto C^{**} . Para la obtención del *polígono límite* asociado a este conjunto se procede de forma análoga a la reflejada en la figura 3.8. Es decir, se localiza el polígono en el cual se encuentra situado el robot, empleando para ello la metodología asociada a la expresión (3.11).

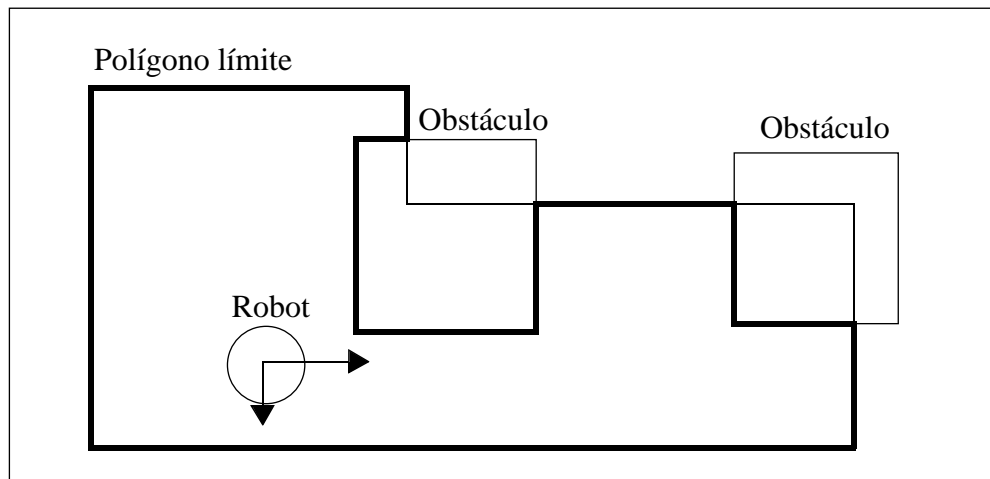


Figura 3.9. Fusión de obstáculos con entorno.

3.4.2.3. Cálculo del nuevo conjunto de obstáculos.

Debido a las acciones verificadas en el paso anterior, algunos de los elementos de B^* se han eliminado al fundirse con el entorno C^* para formar C^{**} . Sin embargo, existe la posibilidad de solapamiento de configuraciones por parte de algunos de los obstáculos expandidos del conjunto B^* . Según el planteamiento del espacio de configuraciones, este hecho carece de importancia, ya que lo esencial para efectuar la planificación consiste en el cálculo del conjunto de configuraciones libres C_f . No obstante, todo ello puede deteriorar la eficiencia del algoritmo de planificación cuando se utiliza una metodología basada en grafos de visibilidad.

Por este motivo, resulta necesario definir un nuevo conjunto de obstáculos B^{**} de tal forma que sus elementos verifiquen la siguiente expresión:

$$(i, j) / i \quad j \quad b_i^{**}(q) \quad b_j^{**}(q) = \quad (3.15)$$

Un elemento de B^{**} se define a partir de los elementos de B^* de la siguiente forma:

$$\begin{aligned} b_i^{**} &= b_j^* \quad b_k^* \quad b_j^*(q) \quad b_k^*(q) \\ b_i^{**} &= b_i^* \quad ((i, j) / i \quad j \quad b_i^*(q) \quad b_j^*(q) =) \end{aligned} \quad (3.16)$$

El algoritmo que se utiliza para la construcción del conjunto B^{**} se fundamenta en detectar en primera instancia los grupos de polígonos que presentan alguna intersección para aplicar, en segundo lugar, el algoritmo de recorte de aristas expuesto en 3.4.2.1 en el caso general de polígonos no convexos, para calcular, de este modo, el polígono envolvente de un grupo.

3.4.3. Consideraciones sobre el modelado circular de robots móviles.

El algoritmo de planificación propuesto consta de dos fases fundamentales: i) una modificación inicial del entorno, para expandirlo según el radio del círculo que modela el robot, y ii) una segunda fase de construcción del grafo de visibilidad para la búsqueda de una ruta óptima según una minimización de cierta función de coste. A partir de esta ruta, se obtiene una función ruta ().

Esta consideración de robot circular para vehículos que pueden modelarse de manera geométrica por la mencionada forma, resulta especialmente eficiente en cuanto al método propuesto. Este hecho se debe a que se calcula una ruta en C_l a un coste computacional bajo, por estar basado en cálculos geométricos. No obstante, cuando el vehículo no puede aproximarse a una forma circular, este planteamiento se muestra deficiente, ya que se descartan ciertas áreas del entorno por donde podría construirse una ruta segura.

La figura 3.10 muestra un vehículo capaz de pasar por un pasillo, al cual le resulta imposible realizar la tarea debido al método de planificación utilizado. La solución se encuentra en emplear un modelado poligonal del robot que aproxime lo más posible la forma del vehículo.

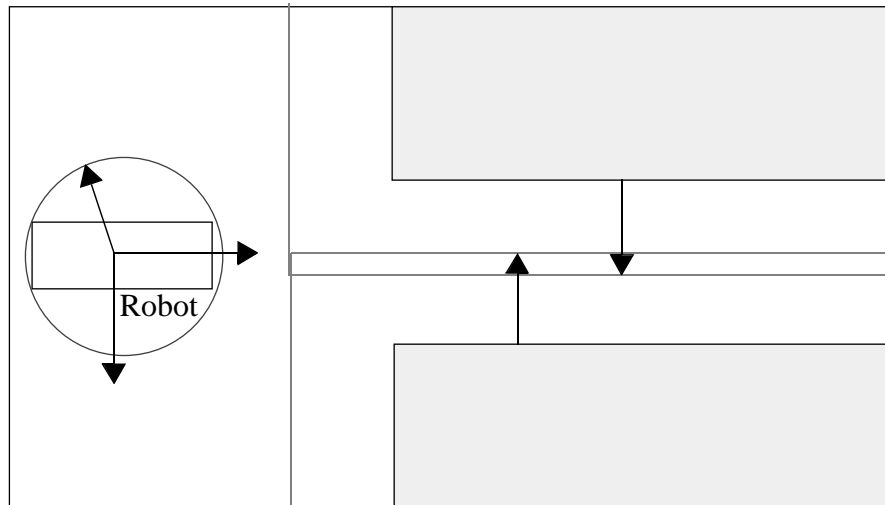


Figura 3.10. Modelado deficiente de un robot mediante un círculo.

Aunque esta nueva aproximación soluciona la dificultad expuesta, por otro lado complica el problema de la planificación. Ahora bien, no se trata de un espacio bidimensional en el cual se puede discernir si una configuración pertenece o no al conjunto de configuraciones libres C_f , mediante el empleo del criterio expuesto en (3.11). El espacio de configuraciones se transforma en tridimensional, por lo que es preciso conocer la orientación del robot para discriminar la pertenencia de una de ellas al conjunto de configuraciones libres u ocupadas. De esta forma, la planificación, sin olvidar este nuevo punto de vista (Kedem y Sharir, 1.990), complica el algoritmo general. Sin embargo, se pueden obtener resultados similares mediante el uso de algoritmos de navegación basado en comportamientos (Mandow y otros, 1.994).

3.5. Conclusiones.

En este capítulo se ha mostrado el primer paso para la construcción de caminos para robots móviles. Esta primera aproximación consiste en la consideración un robot omnidireccional que ejecuta los caminos de modo óptimo sin llegar a tener deslizamientos. De manera que, para este tipo de robots, la única característica que debe presentar el camino es que la continuidad en posición y que una el punto inicial de partida con la posición final deseada. Al camino que cumple esta restricción de continuidad se le ha denominado *ruta*.

La ruta además de la característica de continuidad, debe situarse en el espacio libre de obstáculos del entorno de trabajo del vehículo, y estar formada por un conjunto de Q configuraciones libre del espacio C , definido a partir de una función ruta ().

Los algoritmos basados en grafos de visibilidad para realizar la construcción del grafo muestran una baja complejidad algorítmica en comparación con otros métodos (Rombaut y otros, 1.991; Brooks, 1.983; Kambhampty y Davis, 1.986; Borenstein y Koren, 1.989), aunque presenta el inconveniente de construir rutas semi-libres de obstáculos, al emplear en la construcción de la ruta tanto aristas como vértices de los obstáculos del entorno. Por esta causa, es preciso un tratamiento del entorno previo a la aplicación de este algoritmo, que consiste en la obtención del *entorno expandido* en virtud del cual se construirá el grafo de visibilidad. Esta metodología origina un nuevo parámetro de entrada en la planificación: el *factor de expansión*, para cuya evaluación deben tenerse en consideración ciertas características del entorno y del vehículo.

La elección del método basado en grafos de visibilidad se justifica por el factor de una mayor facilidad en la aplicación del ajuste de una curva en la ruta calculada de forma eficiente, para eliminar, de este modo, y en su totalidad la necesidad de emplear robots omnidireccionales para el seguimiento perfecto del camino. La eficacia de los métodos de ajuste de curva dependerá del tiempo de cómputo preciso, así como de las características que muestre finalmente el camino. De esta manera, algunos métodos de construcción de caminos (Segovia y otros; 1.993) basados en diagramas de Voronoi y ajuste mediante curvas de Bezier se muestran incapaces de relajar la restricción del robot omnidireccional. En los próximos capítulos se expondrá un método de ajuste de curvas (primero para caminos locales y a continuación para globales) que, utilizado en conjunto con la planificación basada en grafos de visibilidad, sí verificará este cometido.