Implementacion de un planificador exacto del movimiento de dos robots.(Matemáticas)

Ba�on, Jos� Mar�a, and Olmedo Arcila. "Implementacion de un planificador exacto del movimiento de dos robots." *Matem�ticas: Ense�anza Universitaria* 8.1-2 (2000): 45+. *Informe Acad�mico*. Web. 29 Sept. 2010.

Document URL
http://find.galegroup.com/gtx/infomark.do?&contentSet=IAC-Documents&type=retrieve&tabID=T002&prodId=IFME&docId=A236334565&source=gale&srcprod=IFME&userGroupName=univalle&version=1.0

Resumen:

Se implementa un algoritmo exacto de planificación del movimiento de dos robots circulares e independientes [1].

Texto Completo:COPYRIGHT 2000 Universidad del Valle

1 Introducción

A pesar del avance en el desarrollo de algoritmos exactos y eficientes de planificación del movimiento de robots con pocos grados de libertad [2, 3, 4, 5, 6] han habido muy pocas implementaciones prácticas en la literatura [7, 8, 9, 10]. La razón principal es que estas soluciones eficientes involucran métodos sofisticados de geometría computacional y combinatoria relativos a arreglos de curvas y superficies [2].

En una reciente publicación [11] señalamos la necesidad de la implementación y experimentación de algoritmos exactos de planificación del movimiento de varios robots. En [11] reportamos el estudio de un algoritmo exacto del movimiento de dos robots circulares e independientes [1] y propusimos un esquema para su implementación consistente en varias etapas que de manera constructiva conducen a la descomposición exacta en celdas del espacio de configuración libre del sistema y a la formación del grafo de conectividad.

En este artículo presentamos el desarrollo de la implementación del planificador y su aplicación a casos particulares. La notación utilizada ha sido la misma que en el artículo [11] al cual remitimos al lector para toda consulta sobre los conceptos teóricos, el problema de la planificación del movimiento y la descripción del algoritmo. E1 artículo está organizado en torno a las etapas propuestas en [11] de la siguiente manera. En la Sección 2 se determinan los espacios admisibles [A.sub.1] y [A.sub.2] del espacio de trabajo y se calculan las curvas críticas. El cálculo y ordenamiento de los puntos de intersección entre las curvas críticas y la descomposición de las curvas críticas en secciones se describe en la Sección 3. El ordenamiento topológico de las secciones se realiza en la Sección 4. La determinación de las regiones regulares y la descomposición en celdas son tratadas en las Secciones 5 y 6 respectivamente. El cálculo del grafo de conectividad, la búsqueda en el grafo y la determinación de los nodos inicial y final, se discuten en las Secciones 7 y 8. Finalmente en las Secciones 9 y 10 se presentan algunos resultados del planificador y se dan las principales conclusiones.

2 Preprocesamiento y Cálculo de las Curvas Críticas

En esta primera etapa se procesa la geometría de los obstáculos del espacio de trabajo, calculándose los espacios admisibles, [A.sub.1] y [A.sub.2], y las curvas críticas del sistema.

El espacio de trabajo del sistema consiste en un conjunto W [subconjunto] [R.sup.2] del plano, acotado por un conjunto finito [EXPRESIÓN MATEMÁTICA IRREPRODUCIBLE EN ASCII.] de obstáculos poligonales y estacionarios. La frontera [delta]W del espacio de trabajo está formada por una frontera exterior [[delta]W.sub.ext] y posibles fronteras interiores [EXPRESIÓN MATEMÁTICA IRREPRODUCIBLE EN ASCII.]. Cada una de las fronteras consiste en una secuencia finita de aristas [a.sub.i] de los obstáculos. Los robots circulares [B.sub.1] y [B.sub.2] con radios [r.sub.1] y [r.sub.2] respectivamente se mueven libremente en W. Denominaremos [P.sub.1] y [P.sub.2] los puntos en los cuales están centrados los robots [B.sub.1] y [B.sub.2].

[FIGURA 1 OMITIR]

En un preprocesamiento inicial se determinan los espacios admisibles [A.sub.1] y [A.sub.2] engrosando isotrópicamente la frontera [delta]W las cantidades [r.sub.l] y [r.sub.2] respectivamente. Las fronteras [delta] [A.sub.1] y [delta][A.sub.2] consisten en aristas desplazadas y vértices desplazados. Las aristas desplazadas se obtienen trasladando paralelamente las aristas de los obstáculos. Los vértices desplazados se obtienen formando los arcos de circunferencia centrados en los vértices convexos de los obstáculos y con extremos en las aristas desplazadas adyacentes. La figura 1 ilustra un espacio de trabajo W con los robots circulares [B.sub.1] y [B.sub.2]. En la figura 2 se muestran las fronteras [delta][A.sub.1] y [delta][A.sub.2] de los espacios admisibles.

Las curvas críticas corresponden a posiciones del robot [B.sub.1] tales que el robot [B.sub.2] es tangente al mismo tiempo a [B.sub.1] y a la frontera [delta]W. En el espacio de trabajo considerado las curvas críticas son segmentos de líneas rectas ó arcos de circunferencia [11].

[FIGURA 2 OMITIR]

Existen dos representaciones de curvas importantes: la representación implícita y la representación paramétrica. La formulación implícita, F(x, y) = 0, es adecuada para el cálculo de intersecciones y para la clasificación de puntos; por ejemplo, nos permite saber si un punto se encuentra en la crezca ó en un lado u otro de la misma. Por otra parte, la representación paramétrica, (x, y) = (x(t), y(t)), evita muchos de los inconvenientes de las formulaciones no-paramétricas. En particular, la representación paramétrica es muy adecuada para el ordenamiento de puntos a lo largo de una curva. En las diferentes etapas del algoritmo se realizan muy frecuentemente tanto intersecciones como clasificaciones y ordenamiento de puntos, por tanto hemos adoptado ambas representaciones para todas las curvas. Además, el problema del cálculo de los puntos de intersección se simplifica considerablemente si ambas representaciones son utilizadas.

Las ecuaciones paramétricas de los segmentos de rectas que unen los puntos ([x.sub.1], [y.sub.1]) Y ([x.sub.2], [y.sub.2]) son: x(t) = [x.sub.1] + t ([x.sub.2 - [x.sub.1]) y y(t) = [y.sub.l] + t (Y2 - Yl), donde t [elemento de] [0, 1]. Los arcos circulares de radio r aceptan la siguiente parametrización racional:

[EXPRESIÓN MATEMÁTICA IRREPRODUCIBLE EN ASCII.]

Cada arista [a.sub.i] de [delta]W genera una curva crítica consistente en un segmento recto que se obtiene trasladando paralelamente la arista [a.sub.i] la cantidad [r.sub.l] + 2[r.sub.2] hacia el interior del espacio de trabajo. Así mismo, los vértices [a.sub.ij] de [delta]W y los vértices de [delta][A.sub.2] son centros de arcos críticos de radios [r.sub.1] + 2[r.sub.2] y [r.sub.1] + [r.sub.2] respectivamente. A cada una de las curvas críticas se les asocia un sistema de referencia local coordenado con el fin de facilitar el cálculo y el procesamiento de los puntos de intersección.

3 Determinación de las Secciones

Las curvas críticas se cortan entre sí y con la frontera [delta][A.sub.1] dividiendo el espacio [A.sub.l] en

regiones R, abiertas y disjuntas, denominadas regiones regulares. La frontera [delta]R de una región regular consiste en secciones de curvas críticas y en secciones de la frontera [delta][A.sub.1]. Denominamos sección de una curva a la parte comprendida entre dos puntos de intersección consecutivos. La determinación de las seccionei implica en primerlugar el cálculo de todos los puntos de intersección y en segundo lugar el ordenamiento de los puntos de intersección en la curva con el propósito de identificar puntos adyacentes.

El cálculo de los puntos de intersección entre todas las curvas: cm'vas críticas y curvas de la frontera [delta] [A.sub.1] conduce a resolver las ecuaciones de la intersección entre rectas, entre arcos y rectas y entre arcos, las cuales se reducen a buscar raíces de polinomios de grado menor o igual a dos. Una vez obtenido el punto de intersección candidato ([x.sub.i], [y.sub.i]) se verifica si es un verdadero punto de intersección. Para ello verificamos que los valores de los parámetros t del punto ([x.sub.i], [y.sub.i]) a lo largo de las dos curvas que se intersectan pelenecen a los intervalos de definición de las curvas. Los puntos de intersección que no pertenecen al cierre de [A.sub.1] son eliminados. Una vez obtenidos los puntos de intersección [EXPRESIÓN MATEMÁTICA IRREPRODUCIBLE EN ASCII.] de cada curva, se ordenan en una lista con respecto a los valores de los parámetros [t.sup.i.sub.j]. Las listas ordenadas de puntos de intersección son procesadas. Cada par de puntos de intersección consecutivos, [EXPRESIÓN MATEMÁTICA IRREPRODUCIBLE EN ASCII.], definene una sección [EXPRESIÓN MATEMÁTICA IRREPRODUCIBLE EN ASCII.]. En adelante indicaremos por S([P.sub.1], [P.sub.2]) la sección cuyos extremos son [P.sub.1] Y [P.sub.2] respectivamente.

4 Ordenamiento Topológico de las Secciones

Las regiones regulares se determinan identificando sus fronteras constituidas por las secciones calcula(tas en la sección precedente. Cada frontera de una región regular se representa como la lista ordenada ([S.sub.1], ..., [S.sub.j], ..., [S.sub.N]) de las secciones que la constituye. Evidentemente, cada par de secciones consecutivas en la lista, [S.sub.i]([P.sub.[alpha]], [P.sub.b]) y [S.sub.i+l]([P.sub.b], [P.sub.c]), tiene un punto común [P.sub.b] que corresponde al punto extremo en que se unen. La determinación de la lista de secciones que forman una frontera requiere la identificación de las secciones adyacentes a una sección en sus puntos extremos. Para establecer las relaciones de adyacencia entre secciones es necesario previamente construir y ordenar el conjunto de secciones que salen de un mismo punto, lo cual se realiza en dos pasos:

1) Constmlcción de la función P [flecha diestra] L(P), donde P es un punto de intersección y L(P) equivalente a] {S([P.sub.[alpha]], [P.sub.b]): P = [P.sub.a] ó P = [P.sub.b]} es el conjunto de las secciones tales que P es uno de sus puntos extremos.

2) Ordenamiento de las secciones de L(P) con respeto a la pendiente de la tangente saliente [T.sub.out] de las secciones en el punto P en el sentido de las agujas del reloj.

El paso 1) se realiza al mismo tiempo que se calculan las secciones. Aquí, es importante tener una representación precisa de la orientación de las secciones [8]. Para cada sección S([P.sub.1], [P.sub.2]) consideramos el par sección/punto extremo [[sigma].sub.i] = (S, [P.sub.i]), i = 1, 2 que denominaremos secciones salientes y que son consideradas como secciones orientadas que emergen del punto [P.sub.i]. En el paso 2) se ordenan las listas de las secciones salientes L(P) con respecto a la tangente saliente [T.sub.out] de la sección en P en el sentido de las agaljas del reloj. La tangente saliente se define de la manera siguiente: [T.sub.out]([[sigma].sub.i], [P.sub.i]) = sT(S, [P.sub.i]), donde s = 1 si i= 1 y s = -1 si i = 2, T(S, [P.sub.i]) es el vector tangente a la sección S([P.sub.1], [P.sub.2]) en [P.sub.i], el cual se calcula cuando la lista de los puntos de intersección son procesados. Las secciones salientes adyacentes son las secciones consecutivas en la lista ordenada L(P).

En el ordenamiento dos o más secciones pueden tener la misma dirección tangencial saliente. En tales casos, que ocurren muy frecuentemente en la práctica, hacemos uso de la curvatura k(t) y de la dirección normal N. Las secciones con curvatura nula son segmentos de recta en la dirección de la tangente saliente. Las secciones con curvatura negativa, arcos de circunferencia, se encuentran a uno u otro lado de la tangente

dependiendo de la posición relativa de N con respecto a [T.sub.out]. En definitiva, las secciones salientes con la misma dirección de la tangente saliente son ordenadas con respecto al valor sign((N x [T.sub.out])* N k(t)).

[FIGURA 3 OMITIR]

En las figuras 3.a y 3.b se muestra la diferencia entre la tangente ordinaria y la tangente saliente. Los vectores T' y T" representan las tangentes ordinarias a la sección S(Q, P) en los extremos Q y P respectivamente. Los vectores [T.sub.out] y [T.sub.out] representan las tangentes salientes a la misma sección S(Q, P) en los extremos Q y P respectivamente.

[FIGURA 4 OMITIR]

[FIGURA 5 OMITIR]

En la figura 4 se muestran las secciones que emergen del mismo punto P junto con sus tangentes salientes en P. Recorriendo las tangentes en el sentido de las agujas de un reloj el conjunto/:(P) queda ordenado de la manera siguiente ([S.sub.4], [S.sub.1], [S.sub.3], [S.sub.2]).

5 Determinación de las Regiones Regulares

[FIGURA 6 OMITIR]

El ordenamiento topológico realizado en la sección anterior permite construir la relación de adyacencia entre secciones. Para ello definimos el operador next(,) que aplica el par (S, P) a la sección que sigue a S en la lista ordenada de secciones L(P). El operador nexí(,) hace corresponder a cadaección S secciones adyacentes en cada uno de sus extremos.

El procedimiento para identificar una región regular es la de comenzar por una sección y obtener todas las secciones de la misma frontera mediante aplicaciones sucesivas de la función nex$(,).

La figura 5 ilustra la aplicación del operador next(,) en cada uno de los extremos de la sección S(Q, P); en el extremo P se obtiene la sección S' = next(S, P) y en el otro extremo Q se obtiene la sección S" = next(S, Q). Las secciones S y S' forman parte de la misma frontera de la región [R.sub.1]. Análogamente, las secciones S y S" forman parte de la frontera de la región [R.sub.2]. Las regiones regulares [R.sub.1] y [R.sub.2] están separadas por la sección 3. La aplicación sucesiva de la función nexO(,) a una sección S permite recorrer las secciones pertenecientes a las fronteras de las regiones [R.sub.1] y [R.sub.2].

El ejemplo de la figura 6 ilustra el funcionamiento de la función next(,) en la determinación de la frontera de una región regular. Comenzando por la sección [S.sub.a], se determina la sección [S.sub.b] adyacente a [S.sub.a] en el punto [P.sub.a] de la siguiente manera: [S.sub.b] = next([S.sub.a], [P.sub.a]). Análogamente, la sección adyacente a [S.sub.b] en el punto [P.sub.b] es [S.sub.c] = next([S.sub.b], [P.sub.b]), y así sucesivamente hasta obtener la sección [S.sub.a] = next ([S.sub.d], [P.sub.d]) de partida. En este ejemplo la región regular está formada por las secciones {[S.sub.a], [S.sub.b], [S.sub.c], [S.sub.d]}.

6 Cálculo de las Componentes

Una vez determinadas las regiones regulares se procede al análisis de sus componentes [K.sub.i]. Recordemos [11] que las componentes son los subconjuntos maximales abiertos y conexos en que se divide el espacio. [A.sub.2] cuando el robot grande está centrado en una posición P1 [elemento de] R. Para determinar el número y la indexación de las clases de equivalencia [sigma](R) de componentes en una región regular R es suficiente el análisis de las componentes que aparecen en el conjunto [A.sub.2] - [rho]([P.sub.1]) para una posición particular [P.sub.1] [elemento de] R donde [rho]([P.sub.1]) indica el círculo de radio [r.sub.1] + [r.sub.2] centrado en [P.sub.1].

La selección de la posición [P.sub.1] se realiza tomando el punto medio M de un segmento [bar.FG] incluido en la región R. El punto medio de una de las secciones S de la frontera de la región R se toma como el extremo F. El otro extremo G se determina tomando la recta normal en F a la sección escogida S y calculando la intersección con todas las secciones de la frontera de R. Entre todos los puntos de intersección nos quedamos con el más cercano G al primer extremo F, lo cual garantiza que el segmento [bar.FG] está contenido en R.

El análisis de la descomposición de [A.sub.2] - [rho]([P.sub.1]) en componentes disjuntas se realiza calculando los puntos de intersección entre la frontera de [A.sub.2] y la circunferencia [rho][P.sub.1]. El número de puntos de intersección debe ser par. Si el número de puntos de intersección es nulo el conjunto [A.sub.2] - [rho][P.sub.1] es una sola componente cuyas fronteras son las fronteras de [A.sub.2] mas una frontera interior formada por el circulo [rho][P.sub.1]

7 Grafo de Conectividad y Búsqueda

Mediante el grafo de eonectividad CG reducimos el problema "continuo" de la búsqueda de un camino libre de eolisiones en una variedad diferencial en el problema "discreto" de la búsqueda de un camino en un grafo entre los nodos inicial [c.sub.I] y final [c.sub.F]. Los nodos del grafo de eonectividad CG, denominados celdas, están formados por los pares c = [R, T] donde R es una región regular y T [elemento de] [sigma](R). Dos nodos, c y c', están conectadas por una arista si y solo si satisfacen las reglas de adyacencia [1, 11].

Habiendo ya determinado las regiones regulares R y sus clases de equivalencia [sigma](R), la construcción de los nodos de CG se reduce a formar todos los pares c = [R, T] y a crear las aristas entre todos los pares de nodos aplicando las reglas de adyacencia. El conjunto de todas las celdas representa una descomposición exacta del espacio de configuración libre del sistema.

Entre los métodos mas utilizados para la búsqueda de caminos en planificación del movimiento de robots está el algoritmo de búsqueda en grafos [A.sup.*] basado en los principios de la programación dinámica [12]. La razón principal es que en múltiples ocasiones se necesita la generación de un camino cuyo costo sea mínimo para cierta función previamente definida. El algoritmo [A.sup.*] garantiza la búsqueda del CalItino de costo mínimo en el grafo de conectividad si tal camino existe. La función costo es en general una función de una cierta magnitud física que varia dm'ante el recorrido, por ejemplo, puede indicar la longitud de un recorrido. En el presente trabajo hemos utilizado el algoritmo [A.sup.*] para encontrar el camino en el grafo de conectividad. No se ha tenido en cuenta ninguna optimización en la bústtueda de caminos. Dejamos para un trabajo posterior la utilización del algoritmo [A.sup.*] para la generación de un camino optimizado para ciertas funciones de costo.

8 Identificación de los Nodos Inicial y Final

Dadas las configuraciones libres inicial y final [q.sub.init], [q.sub.goal] [elemento de] [C.sub.free] debemos determinar las celdas inicial y final, [c.sub.init] y [c.sub.goal], que contengan las configuraciones [q.sub.init] y [q.sub.goal] respectivamente. Esto se realiza en dos etapas:

1. Determinación de las regiones inicial y final [R.sub.init] y [R.sub.goal]. Sean [q.sub.init] = ([P.sup.init.sub.1], [P.sup.init.sub.2]) y [q.sub.goal] = ([P.sup.goal.sub.1], [P.sup.goal.sub.2]) las configuraciones inicial y final. Consideremos una línea recta que pase por los puntos [P.sup.init.sub.1] y [P.sup.goal.sub.1], calculamos las intersecciones entre la línea y todas las secciones críticas y ordenamos los puntos de intersección con respecto a su orden en la línea. Inspeccionando los puntos de intersección mas cercanos a [P.sup.init.sub.1] and [P.sup.goal.sub.1] se deducen las regiones no críticas inicial y final: [R.sub.init] y [R.sub.goal].

2. Determinación de las componentes inicial y final: [T.sub.init] y [T.sub.goal]. Se determinan los contornos

de las componentes de [A.sub.2] - [rho]([P.sup.init.sub.1]). Se traza una línea recta que pase por el punto [P.sup.init.sub.2] y se calculan los puntos de intersección con los contornos de las fronteras de las componentes calculadas. Estudiando los puntos de intersección mas próximos a [P.sup.init.sub.2] es fácil determinar la componente inicial [T.sub.init]. Análogamente se determina la componente final [T.sub.goal].

Una vez calculadas las regiones no críticas [R.sub.init] y [R.sub.goal], y las componentes inicial y final, [T.sub.init] y [T.sub.goal], Las celdas inicial y final, [c.sub.init] y [c.sub.goal], están completamente identificadas.

9 Aplicación

La implemetación del algoritmo ha sido realizada en lenguaje C en una estación de trabajo Sun-Sparc 20. Para ilustrar los resultados de la implementación del planificador vamos a considerar el entorno de trabajo representado en la figura 1.

Los espacios admisibles [A.sub.1] y [A.sub.2] se muestran en la figura 2. [A.sub.1] y [A.sub.2] son recintos acotados por una frontera exterior.

La figura 7 muestra las curvas críticas del sistema recortadas por el espacio admisible [A.sub.1]. Las curvas críticas dividen a [A.sub.1] en [A.sub.2] regiones regulares que aparecen numeradas en la figura. Cada región regular está acotada por secciones críticas.

[FIGURA 7 OMITIR]

El resultado de la determinación del número y de la indexación de las componentes que surgen de cada una de las 22 regiones regulares se muestra en la tabla 1.1, en donde aparece el número de la región seguido de las indexaciones de sus componentes. Recordamos que los índices utilizados para la indexación son: [[alfa].sub.i] para las aristas desplazadas, [[alfa].sub.i], j para los vértices desplazados, y [rho] para los arcos del círculo [rho]([P.sub.1]).

Con las regiones regulares y sus componentes obtenemos el grafo de conectividad del sistema. Los nodos del grafo de conectividad corresponden a los pares región regular--componente y se muestran indexados en la tabla 1. La indexación de los nodos del grafo de conectividad ha sido realizada con el número de la región regular seguido de una ó dos letras ("L", "" R", "" U") que indican la posición relativa de la componente respecto a la región regular. Así por ejemplo la región regular 0 da lugar a tres nodos que indexamos por "'0L", "'0R", "'0U". El nodo "'0L'" se forma con la región 0 y la componente que se extiende a la izquierda de la región regular 0. Análogamente los nodos "0R" y "0U" se forman con la región 0 y las componentes que se ubican respectivamente a la derecha y hacia arriba de la región 0. Los arcos del grafo de conectividad se obtienen aplicando las reglas de adyacencia a todos los pares de nodos.

La figura 8 muestra el grafo de conectividad CG del sistema. Se trata de un grafo conexo consistente en 45 nodos y 51 arcos. El tiempo de procesamiento en construir el grafo de conectividad es de 50 milisegundos.

[FIGURA 8 OMITIR]

Las figuras 9 y 10 muestran dos caminos generados por el planificador. En la figura 9 las configuraciones inicial y final de los robots pertenecen a los nodos 7UL y 21LR respectivamente. El camino discreto encontrado en el grafo CG es: 7UL, 6UL, 5L, 4L, 3L, 1L, 0L, 16L, 17L, 18LR, 20LR Y 21LR. En la figura 10 las conïiguraciones inicial y final de los robots pertenecen a los nodos OL y 14UR respectivamente. El camino en el grafo CG es: 0L, 16L, 17L, 18LR, 17R, OR, 8R, 9R, llR, 12R, 13UR y 14UR. El planificador ha sido aplicado a varios entornos de trabajo con geometrías diferentes. Los resultados serán publicarlos proximamente.

[FIGURA 9 OMITIR]

[FIGURA 10 OMITIR]

10 Conclusiones

Hemos presentado la primera implementación del algoritmo clásico de Schwartz y Sharir [1] para planificar el movimiento sin colisiones de dos robots circulares y hemos mostrado la factibilidad de la implementación. La descomposición exacta del espacio de configuración libre [C.sub.free] de dos robots circulares ha sido implementada mediante procedimientos eficientes de geometría computacional y estructuras de datos geométricas adecuadas con el fin de evitar cálculos no necesarios.

Los resultados obtenidos con el planificador son prometedores desde el punto de vista práctico pues demuestran la aplicabilidad del algoritmo y confirman que caminos libre de colisiones pueden generarse eficientemente mediante métodos exactos.

Queremos señalar algunos problemas abiertos que surgen en la implementación. El primero consiste en el problema de la generación de las trayectorias de los robots en el interior de la secuencia de nodos caracterizados por la irregularidad de las regiones regulares y las componentes. Proponemos la utilización de los diagramas de Voronoi para generar las trayectorias de los robots en cada nodo. El segundo problema consiste en la inestabilidad numérica de las implementaciones de algoritmos geométricos. Hemos tratado de reducir en lo posible las inexactitudes aritméticas considerando una determinada tolerancia en cada punto [13].

Trabajo parcialmente financiado por Colciencias con el código: 1106-14-364-95.

Referencias

[1] J. T. Schwartz and M. Sharir, On ¿he piano movers problem: III. Coordinating the motion of several independent bodies: The special cases of circular bodies moving amidst polygonal barriers, International Journal of Robotics Research, 2(3), (1983), pp. 46-75.

[2] M. Sharir, Algorithmic Motion Planning, ch. 40 of Handbook of Discrete and Computational Geometry, J.E. Goodman and eds., CRC Press, 1997.

[3] J. C. Latombe, Robot Motion planning, Kluwer Academic Publishers, 1991.

[4] Y. Hwang and P. Chen, A heuristic and complete planner .for the classical mover's problem, Proc. of the IEEE ICRA, Nagoya, Japan, (1995), pp. 729-736.

[5] D. Halperin and M. Sharir, A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. Discrete Comput. Geom.~ 16, (1996)~ pp. 121-134.

[6] M. Sharir and S. Sifrony, Coordinated motion planning for two independent robots, Ann. Math. Artif. Intell., 3, (1991), pp. 107130.

[7] F. Avnain, J. D. Boissonnat and B. Faverjon, A practieal exact motion planning algorithm for polygonal objects amidst polygonal obstacles, Technical report 890, INRIA, Sophia-Antipolis, France, 1988.

[8] J. Bañón, Implementation and extension of the ladder algorithm, IEEE Proceedings of the International Conference on Robotics and Automation, (1990), pp. 1548-1553.

[9] B. Drace-Wright, J. P. Laumond, and R. Alami, Motion planning of a robot and a movable object amidst

polygonal obstacles, IEEE Proceedings of the International Conference on Robotics and Automation, 1992, pp. 2474-2480.

[10] J. P. Laumond and R. Alami, A new geometrical approach to manipulation task planning; the case of a circular robot amidst polygonal obstacles anda movable circular object, Technical report 88314, LAAS, CNRS, Toulouse, France, 1988.

[11] J. M. Bañon y O. Arcila, Geometría del movimiento de dos robots circulares, Enseñanza Matemática Universitaria, 1999.

[12] P. E. Hart, N. J. Nilsson and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE International Conference on Robotics and Automation, San Francisco, pp. 1044-1049, 1986.

[13] S. Fang, B. Bruderlin and X. Zhu, Robustness in solid modeling--a tolerant based, intuitionistic approach, Computer Aided Design, 25:9, 1993.

[14] C. Yap, T. Dubé, The exact computation paradigm, in Computing in Euclidean Geometry, Wold Scientific Press, (1995), pp. 452-492.

JOSÉ MARÍA BAÑÓN, OLMEDO ARCILA

e-mail : {banon, olarcila}@borabora.univalle.edu.co

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD DEL VALLE,

CALI, COLOMBIA

Tabla 1.1: Regiones Regulares, Componentes y Nodos.

Región	Componente	Nodo
0	[rho][[alpha].sub.7][[alpha].sub.0][[alpha]sub.1]	OL
	[rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6]	0U
	[rho][[alpha].sub.1][[alpha].sub.2][[alpha]sub.3]	OR
1	[rho][[alpha].sub.7][[alpha].sub.0][[alpha].sub.1]	1L
	[rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6]	1U
	[rho][[alpha].sub.1][[alpha].sub.2][[alpha]sub.3]	1R
2	[rho][[alpha].sub.7][[alpha].sub.0][[alpha]sub.1]	2L
	[rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6]	2U
	[[rho][[alpha].sub.1][[alpha].sub.2]	2R
3	[rho][[alpha].sub.7][[alpha].sub.0][[alpha]sub.1]	31
	[rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6]	3U
	[rho][[alpha].sub.2] [[alpha].sub.3]	3R
4	[[rho][[alpha].sub.6,7][[alpha].sub.7]	41
	[[alpha].sub.0][[alpha].sub.1]	
	[rho][[alpha].sub.4] [[alpha].sub.5]	4U
	[[alpha].sub.6]	
5	[rho][[alpha].sub.6,7][[alpha].sub.7]	5L
	[[alpha].sub.0][[alpha].sub.1]	
	[rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6]	5U

6	[rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6] [[alpha].sub.6,7][[a.sub.7][[alpha].sub.O] [[alpha].sub.1]	6UL
7	[rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6] [[alpha].sub.6],7[[alpha].sub.7] [[alpha].sub.0][[alpha].sub.1]	7UL
8	[rho][[alpha].sub.7][[alpha].sub.0][[alpha].sub.1] [rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6] [rho][[alpha].sub.l][[alpha].sub.2][[alpha].sub.3]	81 8U 8R
9	[rho][[alpha].sub.0][[alpha].sub.1] [rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6] [rho][[alpha].sub.1][[alpha].sub.2][[alpha].sub.3]	91 9U 9R
10	[rho][[alpha].sub.7][[alpha].sub.0] [rho][[alpha].sub.4] [[alpha].sub.5] [[alpha].sub.6] [rho][alpha].sub.1][[alpha].sub.2] [[alpha].sub.3][[alpha].sub.3],4	IOL 10U lOR
11	[rho][[alpha].sub.4][[alpha].sub.5][[alpha]sub.6] [rho][[alpha].sub.l] [alpha].sub.2] [[alpha].sub.3][[alpha].sub.3],4	11U 1IR
12	[rho][[alpha].sub.1][[alpha].sub.2][[alpha]sub.3] [rho][[alpha].sub.4] [[alpha].sub.5] [[alpha].sub.6]	12R 12U
13	[rho][[alpha].sub.1][[alpha.sub.2][[alpha].sub.3] [[alpha].sub.3,4][[alpha].sub.4][[alpha].sub.5]	13UR
14	[rho][alpha].sub.l][[alpha].sub.2][[alpha].sub.3] [[alpha].sub.3,4][[alpha].sub.4] [[alpha].sub.5][[alpha].sub.6]	14UR
15	[rho][[alpha].sub.7][[alpha].sub.0][[alpha]sub.1] [rho][[alpha].sub.4] [[alpha].sub.5] [rho][[alpha].sub.l][a.sub.2][[alpha].sub.3]	15L 15U 15R
16	[rho][[alpha].sub.7][[alpha].sub.0][[alpha]sub.1] [rho][[alpha].sub.5][[alpha].sub.6] [rho][[alpha].sub.l][[alpha].sub.2] [[alpha].sub.3]	16L 16U 16R
17	[rho][[alpha].sub.7][[alpha].sub.0] [[alpha].sub.1] [rho][[alpha].sub.1][[alpha].sub.2][[alpha]sub.3]	17L 17R
18	[rho][[alpha].sub.7][[alpha].sub.0] [[alpha].sub.1][[alpha].sub.2][[alpha].sub.3]	1,811
19	[rho][[alpha].sub.7]UO[[alpha].sub.1] [[alpha].sub.2][[alpha].sub.3]	19LR
20	[rho][[alpha].sub.7][[alpha].sub.0] [[alpha].sub.1][[alpha].sub.2][[alpha].sub.3]	20LR
21	[rho][[alpha].sub.7][[alpha].sub.0] [[alpha].sub.1][[alpha].sub.2] [[alpha].sub.3][[alpha].sub.3,4]	21LR

Número de Documento:A236334565