

Asistente de planificación de trayectorias libres de colisión

María Alejandra Godoy
Departamento de Ingeniería de Sistemas
Universidad de los Andes
e-mail: mgodoy@uniandes.edu.co

Jorge E. Morales
Facultad de Ingeniería de Sistemas
Fundación Universitaria San Martín
e-mail: jmorales@uniandes.edu.co

Resumen: El objetivo de este artículo es presentar un asistente para un planificador de trayectorias libres de colisión para robots manipuladores. La función del asistente es automatizar la definición de los parámetros de un planificador de trayectorias libres de colisión a partir de un análisis del robot y del entorno que rodea el robot. Aunque el asistente se detalla para planificación en robots manipuladores, puede extenderse para planificación en otros tipos de robots.

Palabras claves: Robótica, inteligencia artificial, planificación de trayectorias, manipuladores.

1. Introducción

El problema de la planificación de trayectorias libres de colisión trata con la generación de caminos para un objeto en movimiento o manipulador desde una configuración inicial a una configuración final sin colisionar con los obstáculos que rodean el manipulador. Con base en pruebas experimentales [1] se ha notado que el desempeño de un planificador de trayectorias depende de información del mundo que rodea el robot como por ejemplo: distribución de los objetos en la escena, claridad entre ellos, geometría, etc. Otra característica de algunos planificadores es que éstos requieren de ciertos parámetros para su ejecución los cuales son definidos por el usuario y, en la mayoría de los casos, éste no puede escoger correctamente un valor que esté de acuerdo con la información de la escena ya que no conoce la relación entre ésta y el parámetro. Estos parámetros, además, son invariantes en tiempo de ejecución del planificador.

La solución que se propone es desarrollar un **asistente de planificación** que, a partir de un análisis de las características de la escena y de la información de comportamiento del planificador, pueda definir automáticamente, sin interacción alguna con el usuario, los valores para los diferentes parámetros del método. Así mismo, el asistente debe servir para lograr que estos parámetros sean dinámicos, es decir, que varíen de acuerdo a la localidad que rodea la configuración de la trayectoria que se está analizando.

A continuación se hace un repaso del planificador de Choque y Arrastre [2] sobre el cual se va a implementar la solución propuesta; en la sección 3 se presenta una descripción formal del problema; en la sección 4 se presenta una visión general de la solución; en la sección 5 se expone el caso particular de un asistente para el planificador mencionado y por último, en la sección 6 se presentan las conclusiones.

2. Método de planificación de Choque y Arrastre

El planificador de Choque y Arrastre [2] intenta viajar hacia la configuración final por la trayectoria de interpolación articular lineal. En el momento en que detecta una colisión con un obstáculo, el robot trata de “arrastrarse” sobre éste buscando enrutarse nuevamente sobre la trayectoria inicial. Este planificador trabaja sobre el EC sin precalcular y genera solamente lo indispensable de éste durante el proceso de búsqueda de soluciones.

La manera como el robot rodea el obstáculo en el EC está guiada por una estrategia de evasión, la cual involucra el desplazamiento de una o dos articulaciones principalmente (Figura 2.1.). El robot continúa guiándose con la estrategia inicialmente escogida hasta cuando logre evitar el obstáculo o hasta que las articulaciones lleguen a sus límites, en cuyo caso debe intentar por otra dirección, es decir, escoger otra estrategia de evasión.

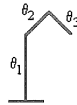


Figura 2.1. Articulaciones involucradas en las diferentes estrategias de evasión

El método está desarrollado para soportar tres tipos de estrategias de evasión: en la estrategia de evasión vertical (Figura 2.2a.) el robot trata de evitar el obstáculo rodeándolo en el EC, por encima y por debajo moviendo θ_3 ó θ_2 y por la izquierda y por la derecha moviendo θ_1 . En la estrategia diagonal (Figura 2.2b.), el robot trata de evitar el obstáculo rodeándolo por las diagonales superior izquierda, superior derecha, inferior derecha e inferior izquierda, moviendo las articulaciones θ_1 y θ_2 ó θ_1 y θ_3 ó θ_2 y θ_3 . En la estrategia combinada (Figura 2.2c.), el robot intenta rodear el obstáculo tanto por las direcciones de la estrategia vertical como por las de la diagonal.

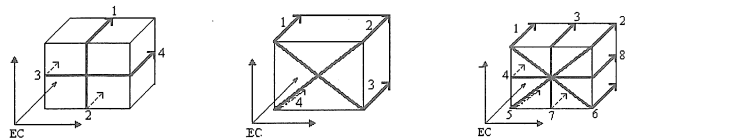


Figura 2.2. (a) Estrategia de evasión vertical en el EC. (b) Estrategia diagonal. (c) Estrategia combinada

Para probar el desempeño del método de Choque y Arrastre se realizó un proceso de experimentación y evaluación [1] el cual fue llevado a cabo en una estación IBM RISC-6000, utilizando MSIM [3]. Durante el proceso de evaluación se identificaron y caracterizaron las tareas que éste resuelve de acuerdo al tiempo de respuesta:

- Las tareas simples son aquellas en las cuales hay pocos obstáculos entre CI y CF, y en donde las necesidades de planificación a nivel de pinza son bajas (tiempo promedio de respuesta: 2 seg).

- Las tareas no triviales son aquellas en las que son necesarias pequeñas maniobras en la mano para encontrar solución y/o cuando aumenta el número de obstáculos que impiden el acercamiento al punto final (tiempo promedio de respuesta: 4 a 5 seg).
- Las tareas más difíciles de resolver se presentan cuando se hacen necesarios movimientos precisos de la mano para encontrar una solución como por ejemplo, una planificación en la que el robot debe insertar un objeto en otro. Para estos casos el tiempo promedio de respuesta moviendo la articulación θ_2 es de 9 seg. y moviendo la articulación θ_3 es de 13 seg. El orden en que son aplicadas las estrategias para evadir los obstáculos influye de una forma directa en el costo de la solución y en el tiempo de respuesta.
- Los casos de falla son aquellos en los que la planificación en la pinza es la única forma de sacar al robot de algún hueco. La mayoría de estos casos fallan debido a que el método no está pensado para trabajar en dichas situaciones.

3. Descripción formal de la planificación

Un planificador de trayectorias es una herramienta compuesta por un conjunto de algoritmos especializados en encontrar una trayectoria libre de colisiones para llevar a un manipulador de un lugar a otro en un mundo dado [2] donde una trayectoria libre de colisiones entre CI y CF se define como:

$$TLC = \langle C_1, \dots, C_n \rangle \mid C_1 = CI, C_n = CF \wedge \forall i \neg \exists \text{ colisión en } C_i, \text{ ni al pasar de } C_i \rightarrow C_{i+1}$$

Un planificador de trayectorias pt es una función dependiente de ciertos parámetros de entrada:

$$pt(R, M, CI, CF, P) = TLC$$

donde R es el robot, M es el mundo, CI y CF son las configuraciones inicial y final respectivamente y P es el conjunto de valores de los parámetros que necesita el planificador para calcular la trayectoria solución entre CI y CF.

4. La solución propuesta

En la actualidad, un planificador de trayectorias se puede ver como una caja negra a la cual entran los siguientes datos: el robot R, el mundo M, las configuraciones inicial CI y final CF y los parámetros P y de la cual sale una trayectoria libre de colisión TLC de CI a CF:

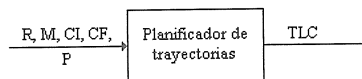


Figura 4.1. Etapas del planificador de trayectorias, actualmente

El asistente de planificación propuesto realizaría dos etapas que buscan mejorar el desempeño del planificador: una de resolución de parámetros y una etapa de retroalimentación. El objetivo de la primera etapa es definir los valores de los parámetros de entrada del planificador a partir de un análisis global de la

información de la escena que puede ser relevante para un buen desempeño del planificador. En las pruebas experimentales realizadas en [1], se puede ver que entre los parámetros del planificador y las características de la escena debe existir alguna función de dependencia que debe ser aprovechada por el asistente de planificación para calcular los valores para cada uno de dichos parámetros, así :

$$P_i = f(C), \quad 1 \leq i \leq n$$

donde P_i es un parámetro de planificación y C es el conjunto de características de las escenas.

El objetivo de la etapa de retroalimentación es permitir que los valores de los parámetros del planificador sean dinámicos, ya que el mundo local a lo largo de la trayectoria solución es diferente. En realidad, ésta no es una etapa adicional sino más bien un llamado a la anterior etapa de resolución de parámetros cada vez que sea necesario recalcular los parámetros para hacerlos más acordes con el mundo local.

En la Figura 4.2. se puede ver cómo el asistente entraría a apoyar al planificador: la idea es que el usuario ya no deba definir ningún parámetro; sólo tendría que escoger las configuraciones entre las cuales quiere hallar una trayectoria libre de colisión. Entonces, al asistente deben entrar el robot R , el mundo M y las dos configuraciones CI y CF y con base en el análisis de estos parámetros se pretende definir el conjunto de parámetros P' , propios del planificador. Adicionalmente, se desea que cuando el entorno local cambie, cambie también el conjunto P' . Con el fin de lograr este objetivo, se crea una etapa de retroalimentación en la que el planificador se apoyaría en el asistente para que éste defina nuevamente el conjunto P' de acuerdo al nuevo entorno local. Debido a que se haría un análisis local, sería necesario pasarle al asistente la configuración actual C_i del robot.

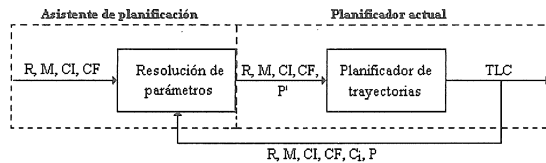


Figura 4.2. Etapas del planificador de trayectorias propuesto

5. Un caso particular

En el desarrollo de esta propuesta se escogió llevarla a cabo sobre el planificador de Choque y Arrastre por las siguientes razones:

- Los resultados de las pruebas realizadas en [1] sugieren la existencia de una posible dependencia entre el desempeño del planificador y características del entorno.
- Actualmente, el orden de aplicación de las estrategias de evasión es siempre el mismo sin importar la geometría o posición del objeto a evadir.

- Es un planificador muy eficiente aún con la desventaja del orden estático de aplicación de las estrategias: sus medidas de respuesta en términos de tiempo de planificación, costo de la solución, número de nodos y número de movimientos, son comparativamente mucho más bajas con respecto a los otros planificadores de MSIM.

5.1. Resolución de parámetros

Recordemos que la función de esta etapa es definir los valores de los parámetros de entrada del planificador de trayectorias para evitar la intervención del usuario y poder calcular unos valores de parámetros lo más acorde posible con la información de la escena.

Para el caso particular del planificador de Choque y Arrastre los parámetros que deben ser resueltos son:

- el grado de la discretización del espacio de configuraciones en el proceso de búsqueda del choque.
- el orden de aplicación de las diferentes direcciones en la estrategia de arrastre.

En la sección 2 se habló de la estrategia de evasión, otro parámetro existente en el planificador actual. Este parámetro fue “eliminado” porque se tomó la decisión de que el asistente analice todas las direcciones de evasión de la estrategia total porque es un proceso generalizable al número de direcciones que se requiera. A continuación se hace describen los parámetros que deben ser resueltos por el asistente, las diferentes alternativas tenidas en cuenta para resolver dichos parámetros y la alternativa escogida.

5.1.1. Discretización del EC en el proceso de búsqueda del choque

Es el nivel de partición que se va a hacer sobre el espacio de configuraciones cuando el robot se encuentra cerca de un obstáculo, es decir, de una posible colisión. Esto significa que cuando no se está cerca de un objeto, el robot puede moverse con libertad pero cuando éste se encuentra cerca de algún obstáculo, debe tener más cuidado en los movimientos que realiza debido a la poca libertad que tiene para moverse. Entonces, para definir este parámetro debe analizar el **nivel de libertad de movimiento** que puede ser medido con diferentes criterios:

a) Conos generalizados en el espacio físico: es un modelo de representación del espacio libre en el espacio físico, por medio de un conjunto de conos formados a partir del barrido y deformación de una sección unidimensional a lo largo de una curva en el espacio llamada columna o espina [4]. La idea es conectar los diferentes conos formando un grafo por el cual viaja el robot (Figura 5.2a.)

b) Zonas de barrido en el espacio físico: son los sectores circulares en los que cada una de las articulaciones se puede mover de su posición actual sin chocar con un objeto en el espacio físico. Esta medida está representada por una n-tupla de la siguiente forma

$$Z = (\theta_{1p}, \theta_{1n}, \dots, \theta_{Np}, \theta_{Nn})$$

donde N es el número de grados de libertad del robot, θ_{ip} y θ_{in} indican que el ángulo es en la dirección positiva o negativa de la articulación, respectivamente, tal como lo muestra la Figura 5.2b.

c) Claridad en el espacio de configuraciones: esta medida se define como la mínima distancia desde la configuración actual hasta los vértices de los obstáculos en el espacio de configuraciones. Esta alternativa fue la que se implementó en el asistente de planificación (Figura 5.2c.)

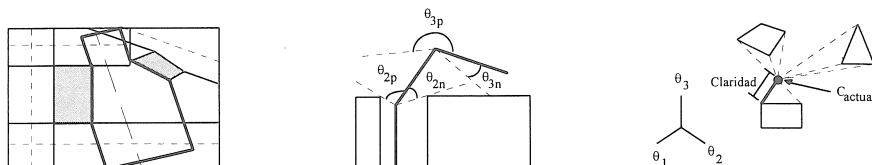


Figura 5.2. (a) Conos generalizados en el EF. (b) Zonas de barrido en el EF. (c) Claridad en el EC.

5.1.2. Orden de aplicación de las direcciones de evasión

El orden de aplicación de las direcciones de evasión indica la dirección hacia la cual debe moverse cada articulación y debe ser una función del **estado actual del mundo local** que es resultado del estado de vecindad del objeto de choque, del estado del objeto de choque y del estado del robot en el momento del choque.

5.1.2.1. Estado de vecindad del objeto de choque

El estado de la vecindad del objeto de choque da información acerca de los objetos que rodean este obstáculo. Esta medida puede ser obtenida con criterios como:

- Densidad alrededor del objeto: con esta medida se conoce la densidad de las n zonas que rodean el obstáculo de choque donde n es el número de direcciones que se van a tener en cuenta para rodear el obstáculo. Las diferentes direcciones se ordenan de acuerdo a qué tan densas están sus respectivas zonas.
- Tipo de objetos: es una clasificación de los objetos que rodean el actual y da información sobre el aspecto externo de un objeto: qué tan feo es, si tiene huecos, etc. Esta clasificación ayuda a que el robot evite rodear estos objetos ya que hacen más difícil el arrastre en caso de un posible choque.
- Número de obstáculos alrededor del objeto: con esta medida se conoce la distribución de las n zonas que rodean el objeto de choque. Esta alternativa fue la que se implementó en el asistente de planificación.

Los aspectos de comparación de las diferentes alternativas son la capacidad de medida, es decir, qué tanto dice dicha medida acerca de la vecindad del objeto de choque y la facilidad de implementación de cálculo.

	Capacidad de medida	Implementación
Densidad	✓	½
No. de objetos	½	✓
Tipo de objetos	½	✗

Tabla 5.1. Comparación de medidas del estado de la vecindad del objeto de choque

5.1.2.2. Estado del objeto de choque

El estado del objeto de choque da información del objeto con el cual se colisiona el robot. Esta medida puede ser obtenida con criterios como los que se exponen a continuación pero para el asistente no se tomó en cuenta ninguna por estar fuera del alcance del desarrollo.

- Distancia del punto de choque a los vértices y las aristas de la cara de choque: esta medida tiene la desventaja de que es incompleta ya que esta distancia es muy local y si tomamos la dirección hacia la cual obtuvimos la menor distancia, podemos encontrar que el objeto hacia esa dirección es más largo o complicado de recorrer.
- Distancia del punto de choque a los vértices y las aristas del minimax del objeto proyectado en el plano de la cara de choque: esta medida es completa ya que se está tomando en cuenta todo el objeto de choque.
- Orientación del objeto de choque: esta medida indica si el objeto está acostado o parado (es decir, es más ancho que largo o viceversa).

	Capacidad de medida	Implementación
Distancias en la cara de choque	*	✓
Distancias en el minimax de la cara de choque	✓	✓
Orientación del objeto	✓	*

Tabla 5.2. Comparación de las medidas del estado del objeto de choque

5.1.2.3. Estado del robot en el momento del choque

Esta medida nos indica cómo se encuentra el robot en el momento de colisionar con un objeto y se puede obtener con los siguientes criterios:

- Posición del robot en el EC: es la configuración actual en la que se encuentran las articulaciones del robot. Esta información sirve para saber si el robot se encuentra cerca de los límites de sus articulaciones.
- Tipo de configuración: indica en qué configuración se encuentra el robot. Por ejemplo, para un PUMA560 indica si su hombro está en configuración derecha o izquierda, si el codo está arriba o abajo y si la muñeca está arriba o abajo.
- Rango de movilidad: es la medida de cuanto le falta a cada articulación para llegar a sus límites mínimo y máximo. Esta medida sirve para saber si el robot se encuentra cerca de los límites de sus articulaciones y ayuda a evitar tomar una dirección de evasión que lo acerque más a dichos límites.

	Capacidad de medida	Implementación
Posición del robot en el EC	½	✓
Rango de movilidad	½	✓
Tipo de configuración	*	✓

Tabla 5.3. Comparación de las medidas del estado del robot en el momento del choque

5.1.2.4. Cálculo del número de objetos alrededor del objeto de choque

Para encontrar el número de objetos que rodea el objeto de choque, se proyecta el centroide de cada obstáculo en el plano perpendicular a la dirección de la articulación θ_3 (Figura 5.5a.) de la siguiente forma: se calcula primero el sistema de referencia de la articulación, se trasladan los centroides de cada objeto a este nuevo sistema y se proyecta el punto al plano xy, es decir, se elimina la coordenada z del punto (Figura 5.5b.). Una vez se ha proyectado el punto, se analiza a qué zona pertenece y se incrementa el número de objetos de dicha zona (Figura 5.5c.).

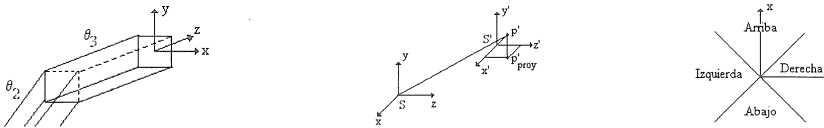


Figura 5.5. (a) Sistema de referencia de la articulación θ_3 . (b) Traslación y proyección del centroide a S' . (c) Zonas de análisis para la estrategia vertical

5.1.2.5. Cálculo del rango de movilidad del robot

Esta medida está dada por la siguiente definición

$$RM+ = |C_{actual\langle i \rangle} - C_{max\langle i \rangle}|, \text{ para } 1 \leq i \leq 3$$

$$RM- = |C_{actual\langle i \rangle} - C_{min\langle i \rangle}|, \text{ para } 1 \leq i \leq 3$$

donde $RM+$ y $RM-$ son los rangos de movilidad en la dirección positiva y negativa de la articulación, respectivamente.

De los resultados de los dos anteriores cálculos se obtiene el orden de las direcciones de evasión así: los resultados del cálculo del número de objetos son ordenados de menor a mayor (se le da prioridad a una dirección con un número de objetos mínimo) y los resultados del rango de movilidad son ordenados de mayor a menor (se le da prioridad a una dirección con mayor movilidad). Finalmente, se mezclan las dos listas ordenadas para obtener el orden final de las direcciones de evasión:

$$O = peso1 * estado_vecindad_{dirección} + peso2 * estado_robot_{dirección}$$

Para asignar $peso1$ y $peso2$ se revisa si existe alguna articulación cuyo rango de movilidad sea más bajo que un epsilon dado. Si este es el caso se le da más peso al estado del robot con el fin de tratar de evitar que el robot tome una dirección en la cual no pueda moverse más. De lo contrario, se le da un mayor peso a la vecindad para tratar de evitar colisiones futuras con otros objetos que rodean el objeto de choque.

5.2. Resultados experimentales

Todas las pruebas realizadas para el método de Choque y Arrastre sin asistente, se realizaron con un valor de un grado para la discretización del espacio de configuraciones debido a que, con este valor, el método siempre encuentra solución si existe una ruta desde la configuración inicial hasta la final. Así mismo, es

importante anotar que se tomó la misma pareja de configuraciones para cada serie de pruebas logrando que los resultados no fueran dependientes de la ruta escogida. Las abreviaturas de las tablas de desempeño son las siguientes:

T: Tiempo de planificación (A: asistente y C: choque)

N: Número de nodos en la solución

C: Costo en grados de la solución

M: Número de movimientos de la solución

I: Número de intentos fallidos antes de encontrar la dirección de evasión final

Por último, los puntos oscuros en los gráficos pertenecen a los resultados del asistente y los puntos claros pertenecen a Choque y Arrastre.

Los resultados obtenidos para la prueba de Claridad vs. Desempeño se lograron a partir de escenas con dos objetos en las que se varió la distancia de separación entre ellos.

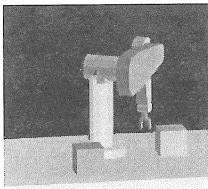


Figura 5.6. Tipo de escena para medir Claridad vs. Desempeño

Tipo de prueba	TA	TC	NA	NC	CA	CC	MA	MC
Muy cerca	0.55	0.69	101	235	231.40	234.41	103	236
Cerca	0.67	0.74	131	252	252.82	251.83	134	254
Lejos	0.75	0.78	151	250	256.83	249.83	153	252
Muy Lejos	0.94	0.82	180	266	306.06	265.41	186	268

Tabla 5.1. Medidas para Claridad vs. Desempeño

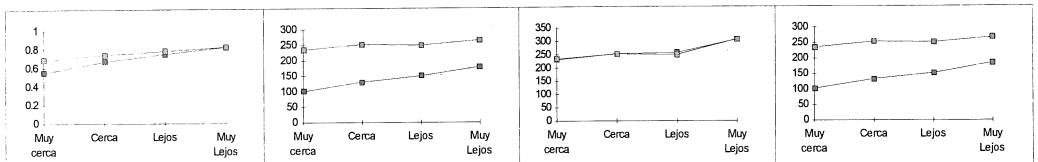


Gráfico 5.1. (a) Claridad vs. Tiempo de planificación. (b) Claridad vs. Nodos en la solución. (c) Claridad vs. Costo de la solución. (d) Claridad vs. Número de movimientos

Para la prueba de Distribución de los obstáculos alrededor del objeto de choque vs. Desempeño se utilizaron escenas en las que el robot sólo colisiona con un objeto y alrededor de éste hay un grupo de objetos que en cada escena está en una posición distinta.

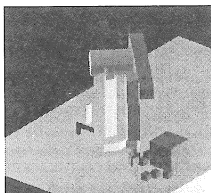


Figura 5.7. Tipo de escena para medir Distribución vs. Desempeño

Tipo de prueba	TA	TC	NA	NC	CA	CC	MA	MC	IA	IC
Arriba	0.48	0.64	71.0	89.5	86.41	88.42	71.0	90.5	0	1
Derecha	0.50	0.63	80.0	96.5	93.41	95.42	80.0	97.5	0	1
Abajo	0.49	0.81	86.5	151.0	104.41	153.7	86.5	152.0	0	1
Izquierda	0.55	0.64	71.5	89.5	87.39	88.42	71.5	90.5	0	1

Tabla 5.2. Medidas para Distribución vs. Desempeño

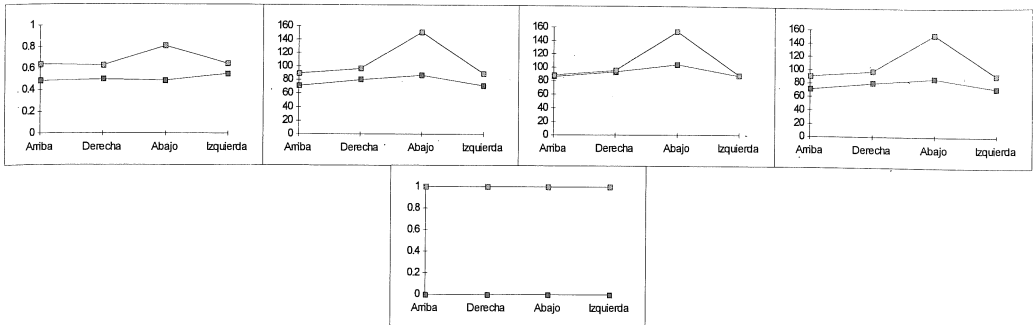


Gráfico 5.2. (a) Distribución vs. Tiempo de planificación (b) Distribución vs. Nodos en la solución (c) Distribución vs. Costo de la solución. (d) Distribución vs. Número de movimientos (e) Distribución vs. Número de intentos fallidos

Para la prueba de Rango de movilidad del robot vs. Desempeño se utilizaron escenas en las que alguna de las tres articulaciones de posicionamiento del robot se encuentra cerca de sus límites en el momento de la colisión, obligando al robot a que no evada el objeto en esa dirección.

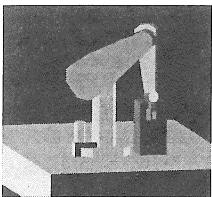


Figura 5.8. Tipo de escena para medir Movilidad vs. Desempeño

Prueba	TA	TC	NA	NC	CA	CC	MA	MC	IA	IC
1	0.54	1.95	91	147	147.37	146.36	91	148	0	1
2	0.67	2.01	101	153	163.49	162.50	101	153	0	2
3	0.55	1.97	98	142	145.10	147.38	99	144	0	1
4	0.72	2.11	151	164	183.74	185.23	153	164	0	2

Tabla 5.3. Medidas para Movilidad vs. Desempeño

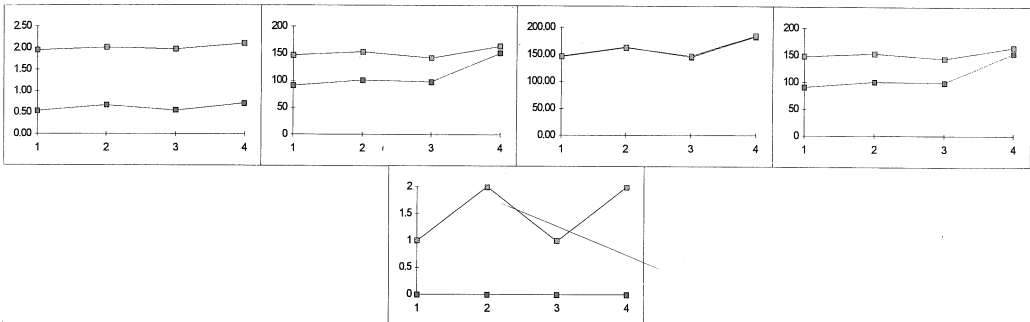


Gráfico 5.3. (a) Movilidad vs. Tiempo de planificación (b) Movilidad vs. Nodos en la solución (c) Movilidad vs. Costo de la solución (d) Movilidad vs. Número de movimientos (e) Movilidad vs. Número de intentos fallidos

En la primera prueba, el tiempo de planificación mejora al tener en cuenta la claridad, ya que el robot avanza con movimientos más largos cuando tiene libertad para moverse. Si analizamos el gráfico vemos que la diferencia entre los tiempos es mayor cuando los objetos están cerca que cuando la distancia de separación es mayor. Esto es debido a que, cuando los dos objetos están cerca, el asistente hace que el robot recorra las distancias de aproximación en un número menor de movimientos que en el planificador original, recorriendo una mayor distancia por cada movimiento. Cuando los objetos se alejan, las zonas de

movilidad se hacen más pequeñas obligando al robot a moverse un poco más suave, lo que incrementa el tiempo de la planificación.

El número de nodos y de movimientos en la solución, que arroja la planificación, es menor por el hecho de que cuando en la ruta hay una gran distancia que recorrer, sin obstáculos cerca, el robot ejecuta menos movimientos y por lo tanto hay menos puntos en la trayectoria.

El costo de la solución no presenta este mismo síntoma. En el gráfico se observa que los costos de la solución para ambos planificadores es similar. Esto parece sugerir que el costo de la solución es independiente del grado de discretización escogido en cada punto y que es dependiente de la dirección de evasión que escoja el método. Entonces, el costo de la trayectoria difiere para los dos métodos cuando alguno de ellos escogió una dirección de evasión distinta en algún momento de la planificación.

Para la prueba de Distribución vs. Desempeño se puede concluir que las diferencias en los resultados son uniformemente variables porque en Choque y Arrastre siempre hubo un intento fallido antes de escoger la dirección final que el asistente escogió directamente, razón por la cual el planificador de Choque y Arrastre se tomó un poco más de tiempo y de movimientos, mientras intentaba por otra dirección por la cual no pudo evadir el obstáculo.

Con respecto a la prueba de Rango de movilidad vs. Desempeño, las diferencias de los resultados son más amplias porque cuando Choque y Arrastre escoge una dirección de evasión no sabe si más adelante la articulación o articulaciones involucradas en el movimiento llegarán a su límite, situación en la cual se devuelve para intentar por otra dirección. Si la distancia recorrida entre la escogencia de dicha dirección y la falla en la evasión es considerable, todos los parámetros del desempeño se elevan considerablemente.

Sintetizando, con las pruebas realizadas se puede concluir que el costo de la solución depende del orden de aplicación de las direcciones de evasión lo que indica que depende de las medidas del estado actual del mundo local. Por otro lado, el tiempo de planificación, el número de nodos y el número de movimientos dependen tanto de este parámetro como del grado de discretización del EC, es decir, que son dependientes del estado actual del entorno y del nivel de libertad de movimiento en cada posición de la trayectoria.

6. Conclusiones

Se ha propuesto y desarrollado un asistente de planificación con capacidad de automatizar la definición de los parámetros de entrada a un planificador de trayectorias libres de colisión. Este planificador, en su búsqueda de una trayectoria, realiza dos tipos de movimientos: aproximación al objeto de choque y evasión del mismo. Para cada uno de estos movimientos, el asistente apoya al planificador definiendo el parámetro necesario: discretización en el espacio de configuraciones para la aproximación y el orden de

aplicación de las direcciones de evasión para el arrastre. Después de llevar a cabo el desarrollo del asistente de planificación se llegan a ciertas conclusiones que se exponen a continuación:

- Cuando se definen los parámetros de planificación con base en información del entorno y del robot, se mejoran las medidas de respuesta del planificador: el número de nodos, el número de movimientos y el tiempo de planificación disminuyen al igual que el número de intentos fallidos. El costo de la solución se mantiene constante a no ser que cada planificador escoja una dirección de evasión distinta.
- El asistente de planificación mejora el método de Choque y Arrastre porque, con su ayuda, encuentra siempre una solución (si la hay) ya que planifica con un mínimo de discretización en los lugares de la escena donde es necesario.
- Los objetivos perseguidos por el asistente y los resultados obtenidos justifican incluir directamente, en planificadores futuros, e implementar en los actuales, un asistente de tal naturaleza.
- El asistente de planificación es un aporte al estudio de la planificación de trayectorias ya que ayuda a convertir un planificador en un elemento con un grado de inteligencia más alto.

7. Bibliografía

- [1] Toro, J., Morales, J., Villalobos, J., "*Planificación de Trayectorias usando Estrategias de Evasión en el Espacio de Configuraciones*", Memo de investigación, Universidad de los Andes, 1994.
- [2] Toro, J., "*Planificación de Trayectoria para Robots Manipuladores*", Tesis de Magister, Universidad de los Andes, 1994.
- [3] Villalobos, J. "*MSIM: Un Ambiente de Programación de Robots con Capacidad de Planificación de Tareas*", Proc. CLEI'91.
- [4] Brooks, R. "*Solving the Find-Path Problem by Good Representation of Free Space*", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-13, No. 3, marzo-abril 1983.