

CAPÍTULO 4

RESULTADOS

4.1 METODOS DE PLANEACION DE MOVIMIENTOS

Los resultados mostrados en éste capítulo corresponden a distintas pruebas realizadas durante el desarrollo de la esta tesis, las cuales fueron ejecutadas en una computadora personal de escritorio marca *Sony* modelo *Vaio* con procesador *Intel Pentium IV*, velocidad de 2.4 Ghz. y 512 Mb. en *RAM* y en una computadora portátil marca *Toshiba* modelo *Satellite* con procesador *Intel Pentium IV*, velocidad de 2.4 Ghz. y 256 Mb. en *RAM*.

Los tres métodos de planeación de movimientos implementados en esta tesis (*PRM*, *OBPRM* y *Visibilidad*), descritos en el capítulo 2, se probaron en espacios de trabajo sencillos y con robots de forma regular y poligonal, con el fin de demostrar su efectividad al buscar rutas entre las configuraciones inicial y final.

Las configuraciones generadas, para el robot, en éstos problemas consistieron en generar aleatoriamente cada uno de los parámetros tanto de la posición (x, y, z) como de la orientación (α, β, γ) , estas configuraciones serán explicadas detalladamente en la sección 4.3. Las rutas encontradas por los diferentes métodos para los distintos espacios de trabajo se visualizaron con el sistema GEMPA y son mostradas a continuación:

4.1.1 PRM

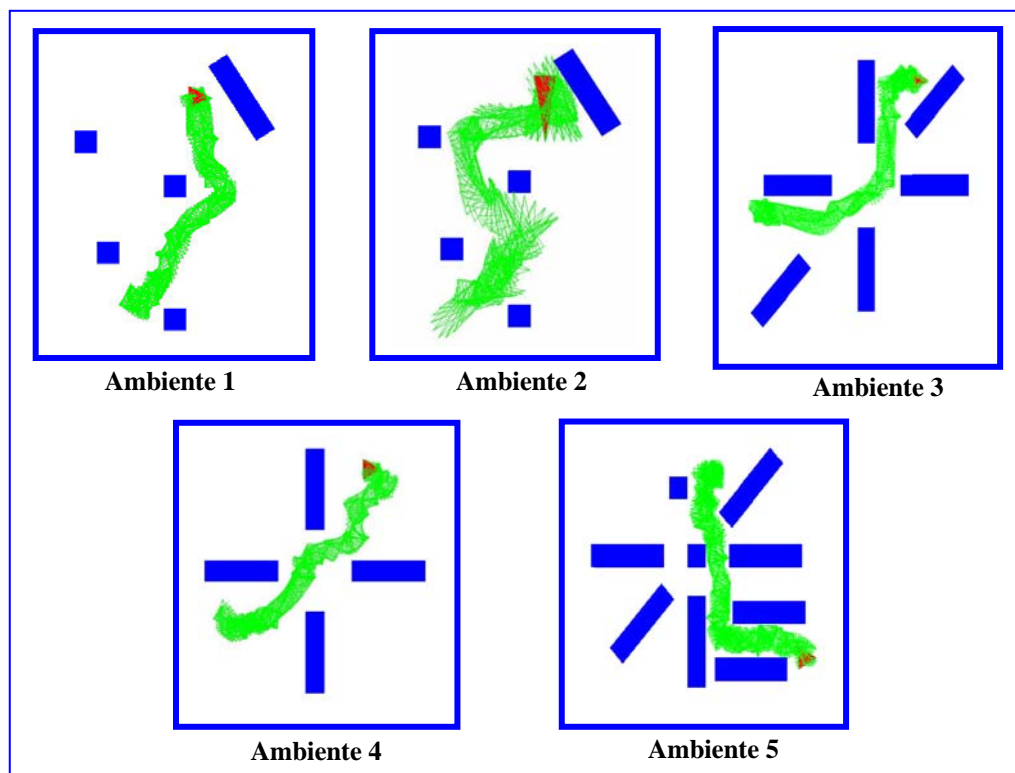


Figura 4.1 Las rutas encontradas para el robot (*tetrahedro*) son desplegadas como una secuencia de configuraciones en color gris claro a través de los obstáculos.

La tabla mostrada a continuación resume los resultados generados por el método después de realizar 10 pruebas en cada uno de los espacios de trabajo ilustrados. Tanto el número de configuraciones como el número de vecinos cercanos son parámetros definidos por el usuario, los cuales participan en la etapa de construcción del roadmap.

Ambiente	No. Configuraciones	Vecinos cercanos	Tiempo (h : m : s : cs)	Ruta
1	50	10	0:0:9:14	Si
2	50	10	0:0:6:22	Si
3	50	10	0:0:7:25	Si
4	50	10	0:0:8:68	Si
5	500	10	0:1:30:32	Si

Tabla 1 Resultados obtenidos por el método *PRM*.

4.1.2 OBPRM

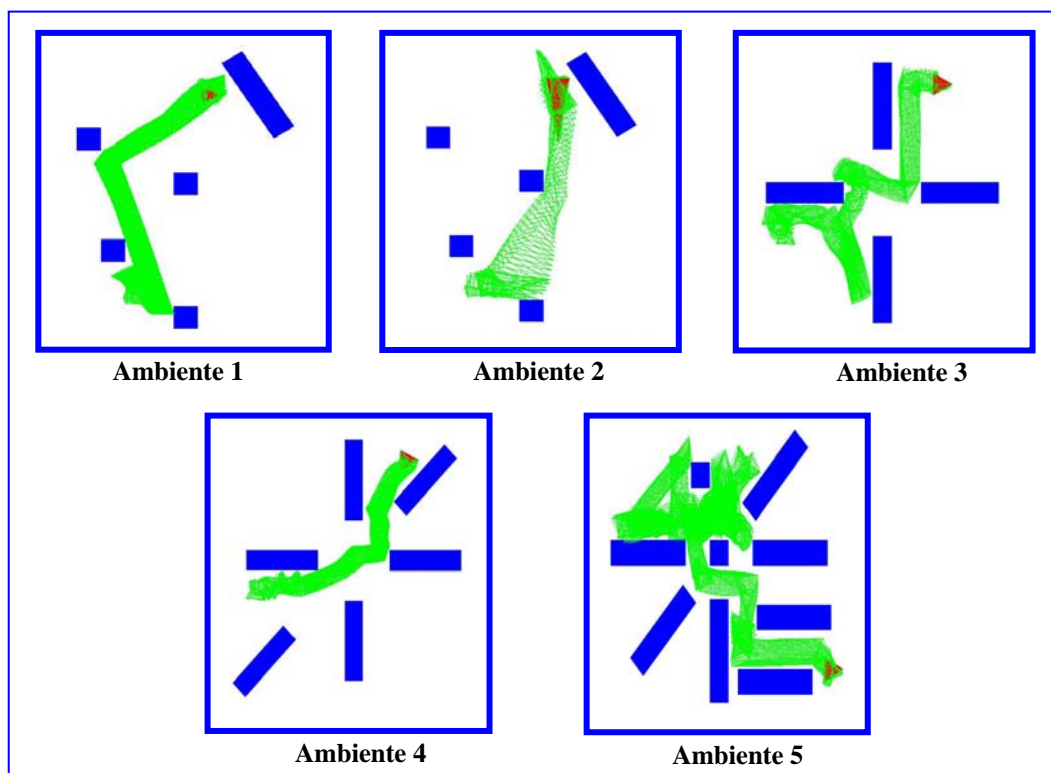


Figura 4.2 Las rutas encontradas para el robot (*tetrahedro*) son desplegadas como una secuencia de configuraciones en color gris claro a través de los obstáculos.

La tabla 2 muestra los resultados generados por el método después de realizar 10 pruebas en cada uno de los espacios de trabajo ilustrados. Tanto el número de configuraciones como el número de vecinos cercanos son parámetros definidos por el usuario, los cuales participan en la etapa de construcción del roadmap.

Ambiente	No. Configuraciones	Vecinos cercanos	Tiempo (h : m : s : cs)	Ruta
1	20	10	0:2:2:18	Si
2	20	10	0:0:39:7	Si
3	20	10	0:0:31:52	Si
4	20	10	0:1:53:46	Si
5	20	10	0:3:28:32	Si

Tabla 2 Resultados obtenidos por el método *OBPRM*.

4.1.3 Visibilidad

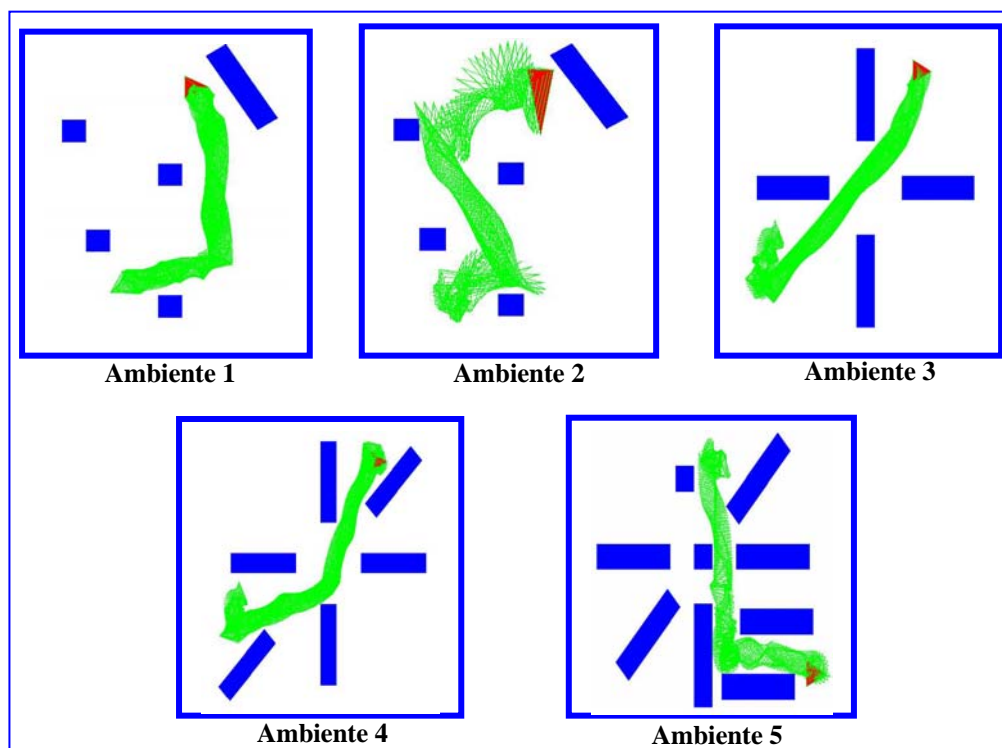


Figura 4.3. Las rutas encontradas para el robot (*tetrahedro*) son desplegadas como una secuencia de configuraciones en color gris claro a través de los obstáculos.

La tabla 3 resume los resultados generados por el método después de realizar 10 pruebas en cada uno de los espacios de trabajo ilustrados. Tanto el número de configuraciones como el número de vecinos cercanos son parámetros definidos por el usuario, los cuales participan en la etapa de construcción del roadmap.

Ambiente	No. Configuraciones	Vecinos cercanos	Tiempo (h : m : s : cs)	Ruta
1	50	10	0:0:8:94	Si
2	50	10	0:0:2:95	Si
3	50	10	0:0:6:13	Si
4	50	10	0:0:9:40	Si
5	100	10	0:0:29:31	Si

Tabla 3 Resultados obtenidos por el método *Visibilidad*.

4.2 CONFIGURACIONES GENERADAS POR LA HEURÍSTICA

Durante el diseño de la heurística desarrollada en ésta tesis, se evaluaron diferentes estrategias para la generación de configuraciones adecuadas que permitieran resolver

algunas de las versiones del problema de los clavos, tal y como se mencionó en el capítulo 3.

Las distintas estrategias evaluadas generaron una gran diversidad de configuraciones, muchas de las cuales se encuentran en la región difícil del espacio de trabajo (*pasaje estrecho formado por el obstáculo*) y que llevaron al robot a salir de la configuración entrelazada en que se encontraba con el obstáculo, en algunas versiones del problema.

La tabla 4 muestra los diferentes tipos de configuraciones generados así como sus características, llevándonos a distinguir las estrategias más apropiadas en la generación de configuraciones para solucionar las diferentes versiones del problema de los clavos.

Por otra parte señalamos que existen muchas otras variantes posibles para la generación de configuraciones (*por ejemplo utilizar las aristas en lugar de los triángulos que forman la malla del obstáculo*), pero las que se han implementado en este trabajo, pueden ser consideradas las más representativas.

Tipo	Posición	Orientación	Región más probable	Características	Versión del problema de los clavos
I	Baricentro del triángulo	Cuaterniones <i>Eje:</i> Vector normal del triángulo.	Pasaje estrecho	Configuraciones entrelazadas con el obstáculo y en el pasaje estrecho	1.5, 1.2
II	Baricentro del triángulo	Aleatoria	Distribuidas a lo largo del obstáculo	Configuraciones entrelazadas con el obstáculo	1.5, 1.2, 1.1
III	Vértice aleatorio del triángulo	Aleatoria	Aquellas donde el obstáculo tiene más vértices, (<i>partes curvas</i>)	Configuraciones entrelazadas con el obstáculo	1.5, 1.2, 1.1
IV	Baricentro del triángulo	Cuaternión <i>Eje:</i> Lado aleatorio del triángulo	Cerca del pasaje estrecho	Configuraciones entrelazadas con el obstáculo y cerca del pasaje estrecho	1.5, 1.2

Tabla 4. Tipos de configuraciones generadas con la heurística.

Tal y como observamos en la tabla anterior, los tipos I y IV generan las configuraciones más importantes y más difíciles de encontrar al tratar de resolver el problema de los clavos en las versiones 1.5, 1.2 respectivamente. Recordemos que los tipos de configuraciones generados son factibles dadas las características del cuerpo del obstáculo y del robot, explicadas en la sección 3.7.

Los tipos II y III generan configuraciones colocando el robot en los puntos seleccionados y la orientación se genera aleatoriamente a través de un generador de números pseudoaleatorios *Random* entre los rangos A y B (comúnmente $A = 0$ y $B = 2\pi$), los cuales pueden ser configurados por el usuario.

Este tipo de configuraciones son generadas con el fin de incrementar la diversidad de configuraciones generadas de manera entrelazada con el obstáculo, además de ser muy sencillas de implementar. La figura 4.4 muestra ejemplos de éste tipo de configuraciones:

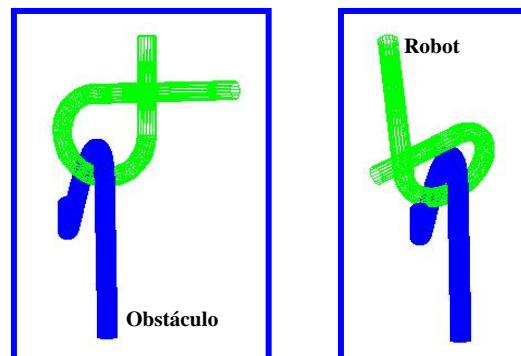


Figura 4.4 Ejemplos de configuraciones para los tipos II y III.

4.3 CONFIGURACIÓN ALEATORIA

Otro tipo de configuraciones generadas son las configuraciones con posición y orientación aleatorias cuyo método no es parte de la heurística diseñada en éste trabajo sino del conjunto de clases definidas en el sistema GEMPA. Estas configuraciones desempeñan un papel importante en los resultados obtenidos en ésta tesis, mismo que será explicado más adelante.

Generar posiciones y orientaciones aleatorias, que permita construir una configuración, que coloque al robot en el espacio de trabajo con cierta orientación, implica el utilizar una función generadora de números *pseudoaleatorios* que permita obtener un número entre cierto rango.

Para generar los parámetros de posición se define un rango entre los límites del espacio de trabajo, de la siguiente manera:

$$\begin{aligned}x &= \text{Random}(X_{min}, X_{max}) \\y &= \text{Random}(Y_{min}, Y_{max}) \\z &= \text{Random}(Z_{min}, Z_{max})\end{aligned}$$

dónde

Random es la función generadora de número pseudoaleatorios y X_{min} , X_{max} , Y_{min} , Y_{max} , Z_{min} y Z_{max} son los límites del espacio de trabajo respectivamente. Para los parámetros de orientación se define lo siguiente:

$$\begin{aligned}\alpha &= \text{Random}(0, 2\pi) \\ \beta &= \text{Random}(0, 2\pi) \\ \gamma &= \text{Random}(0, 2\pi)\end{aligned}$$

y las configuraciones generadas son del siguiente tipo:

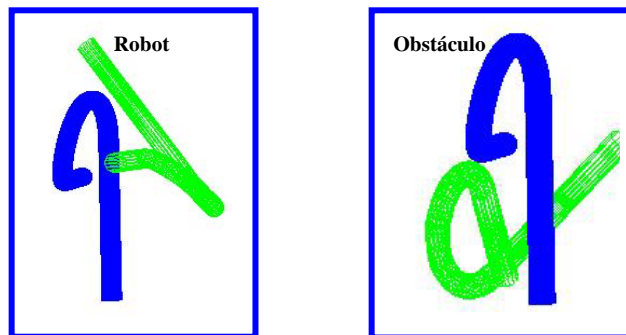


Figura 4.5 Ejemplos de configuraciones generadas aleatoriamente en el espacio de trabajo.

4.4 EMPLEO DE LOS MÉTODOS PROBABILÍSTICOS IMPLEMENTADOS

Los métodos probabilísticos implementados en ésta tesis (*PRM*, *OBPRM* y *Visibilidad*) generaron rutas con las configuraciones generadas por la heurística para las versiones 1.5 y 1.2 del problema de los clavos.

En estas rutas el robot se desplaza exitosamente de una configuración inicial a una configuración final pasando por el corredor estrecho. El procedimiento realizado es el siguiente:

4.4.1 Pasos para solucionar la versión 1.5

1. Ejecutar el método *Visibilidad* generando 2500 configuraciones de los siguientes tipos (ver tabla 4):
 - a. 1000 configuraciones, generadas del tipo *I*,
 - b. 500 configuraciones generadas del tipo *II*,
 - c. 500 configuraciones generadas del tipo *III*,
 - d. 500 configuraciones generadas del tipo *IV*.

El roadmap construido y las configuraciones generadas se conservan en estructuras temporales.

2. Ejecutar el método *OBPRM* generando 100 configuraciones del tipo *IV*. El roadmap construido y las configuraciones generadas se conservan en estructuras temporales.
3. Ejecutar el método *UneGrafo(...)*, el cuál recibe como parámetros los dos roadmaps construidos en los pasos 1 y 2 así como ambas listas de las configuraciones generadas. El método mezcla ambos roadmaps en uno solo de la misma manera que en la etapa de conexión del método *PRM*.

Es decir, cada uno de los nodos que forman uno de los roadmaps se intenta conectar con los k nodos vecinos más cercanos perteneciente al otro roadmap, agregando estas nuevas aristas a un tercer roadmap resultante, que finalmente será la mezcla de los dos anteriores.

El roadmap construido y las configuraciones generadas se conservan en estructuras temporales.

4. Ejecutar el método *PRM* generando 50 configuraciones aleatorias. El roadmap construido y las configuraciones generadas se conservan en estructuras temporales.
5. Ejecutar el método *UneGrafo(...)* para unir los roadmaps generados en los pasos 3 y 4. El roadmap construido y las configuraciones generadas se conservan en estructuras temporales.
6. Determinar que configuraciones pertenecen a componentes disconexos del roadmap resultante del paso 5 y realizar 30 rotaciones a cada configuración (*rotar la configuración usando cuaterniones contruidos con ejes definidos en la heurística*).

Buenos resultados se obtuvieron al rotar la configuración con respecto al vector normal del triángulo seleccionado aleatoriamente. Las nuevas configuraciones generadas serán verificadas por colisión a fin de detectar aquellas que se encuentren en el espacio libre.

7. Intentar conectar éstas configuraciones con los k nodos vecinos más cercanos del roadmap construido en el paso 5 y adicionar las nuevas aristas generadas, esto con el fin de mejorar aún más la conectividad en el roadmap.
8. Ejecutar la fase de consulta para determinar si existe una ruta entre las configuraciones inicial y final.

4.4.2 Pasos para solucionar la versión 1.2

Para la versión 1.2 se siguen los mismos pasos que para la versión 1.5 con solamente una variante en el paso 1:

1. Ejecutar el método *Visibilidad* generando 6500 configuraciones de los siguientes tipos (ver tabla 4):
 - a. 5000 configuraciones, generadas del tipo *I*,
 - b. 500 configuraciones generadas del tipo *II*,
 - c. 500 configuraciones generadas del tipo *III*,
 - d. 500 configuraciones generadas del tipo *IV*.

La tabla siguiente explica el papel que desempeñan cada uno de los métodos probabilísticos en la generación de estas rutas:

Método Probabilístico	Función	Contribución
Visibilidad	Tiene como objetivo el generar configuraciones en el pasaje estrecho, cerca de éste y configuraciones entrelazadas con el obstáculo.	Configuraciones que desplacen al robot por el pasaje estrecho y generar un nodo, lo más cerca posible, que conecte con la configuración inicial.
OBPRM	Generar configuraciones entrelazadas a lo largo de todo el cuerpo del obstáculo con orientaciones en diversas direcciones.	Mejorar la conectividad del roadmap aumentando conexiones entre las configuraciones generadas en el método anterior.
PRM	Generar configuraciones en el espacio abierto presente entre el obstáculo y la configuración final.	Generar un nodo, lo más cerca posible, que conecte con la configuración final.

Tabla 5 Función principal de cada método probabilístico en la ruta generada.

Mejoramiento	Función	Contribución
Rotación de configuraciones	Detectar los nodos aislados del roadmap (<i>que no conectan con ningún otro nodo</i>) y generar configuraciones en la misma posición pero con diferentes orientaciones.	Aumentar la conectividad del roadmap construido uniendo componentes disconexos.

Tabla 5a Función principal del mejoramiento en la ruta generada.

El mejoramiento realizado al roadmap construido, por los tres métodos probabilísticos, debe ser ejecutado siempre después de que las ejecuciones de los métodos han terminado, sin embargo el orden en el cual estos métodos se ejecuten es indistinto.

A continuación se muestran configuraciones representativas, obtenidas en las rutas generadas con este procedimiento para las versiones 1.5 y 1.2 del problema de los clavos.

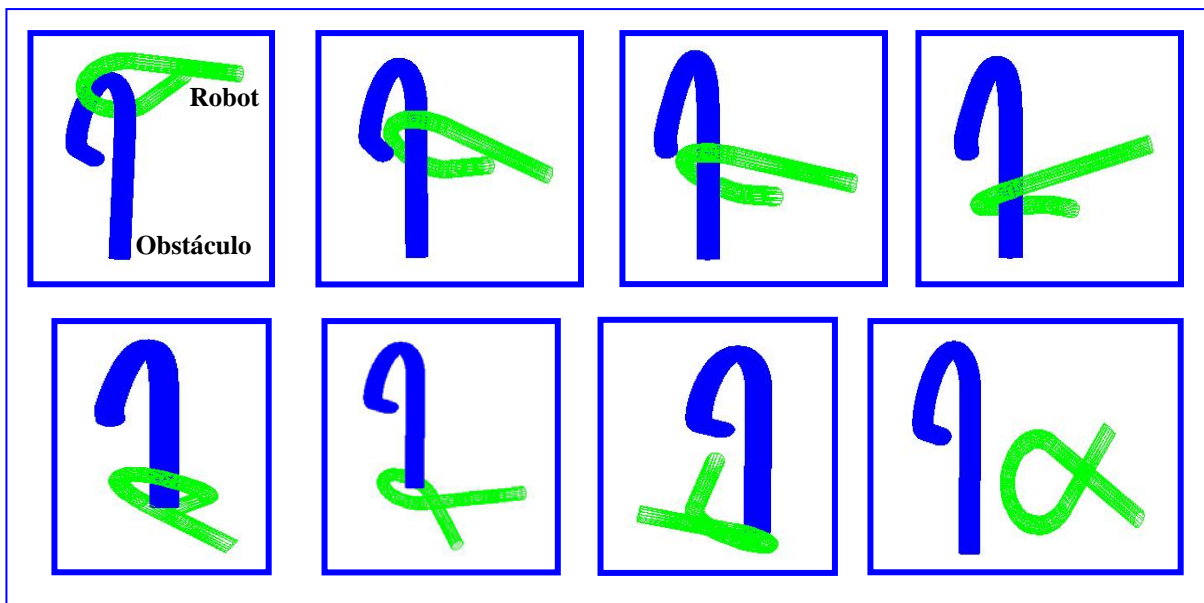


Figura 4.6 Secuencia de configuraciones en la ruta de la versión 1.5.

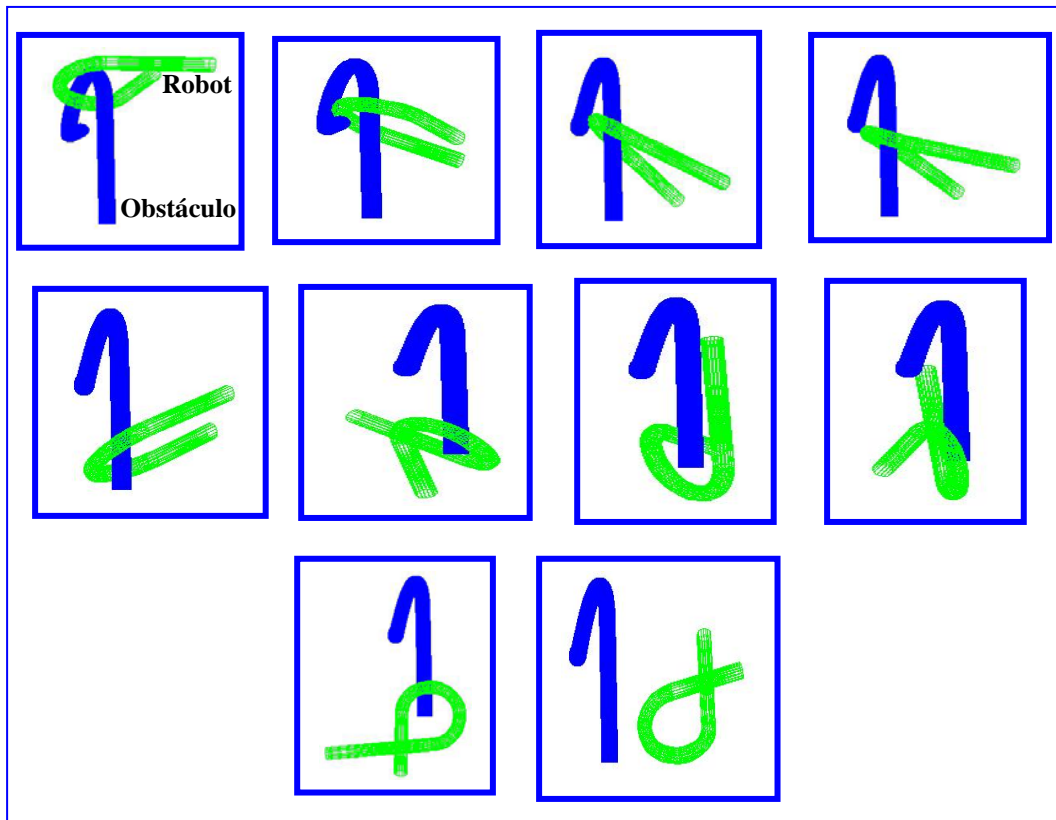


Figura 4.7 Secuencia de configuraciones en la ruta de la versión 1.2.

El procedimiento desarrollado para obtener estas rutas es muy costoso, en cuanto a tiempo, ya que los tiempos de ejecución finales fueron aproximadamente de 10 horas para la *versión 1.5* y 17 horas para la *versión 1.2*.

Estos tiempos se deben aproximadamente en un 90% al método de *Visibilidad*, dado el número grande de configuraciones que se generan en él y la fase de selección de configuraciones que éste realiza.

Podemos afirmar que el método *Visibilidad* obtiene las mejores configuraciones presentes en la ruta construida para las versiones *1.5* y *1.2* del problema de los clavos, sin embargo un análisis minucioso del método, en cuanto a su desempeño, debería ser realizado posteriormente para mejorar estos tiempos (ver sección 6.2 inciso *h*, *Trabajo futuro*).

4.5 AVANCES OBTENIDOS PARA LA VERSIÓN 1.1

4.5.1 Combinación de estrategias para generar una configuración particular

Una vez diseñada la heurística se desarrolló un método que combina diferentes estrategias para generar una configuración particular, obteniendo configuraciones temporales intermedias en cada paso; el nuevo tipo de configuraciones obtenido es el *Tipo V* referenciado en la tabla siguiente:

Tipo	Posición	Orientación	Región más probable	Características	Versión del problema de los clavos
V	Baricentro del triángulo <i>(Triángulo con mayor probabilidad acumulada respecto de su área)</i>	Cuaternión <i>Ejes:</i> Vector normal + vector definido por el usuario	Pasaje estrecho	Configuraciones entrelazadas con el obstáculo y en el pasaje estrecho	1.1
VI	Punto aleatorio dentro del triángulo	Cuaternión <i>Eje:</i> Vector normal del triángulo	Distribuidas a lo largo del obstáculo	Configuraciones entrelazadas con el obstáculo	1.1
VII	Punto aleatorio dentro del triángulo	Cuaternión <i>Eje:</i> Lado aleatorio del triángulo	Distribuidas a lo largo del obstáculo y relativamente cerca del pasaje estrecho	Configuraciones entrelazadas con el obstáculo	1.1

Tabla 6. Tipos de configuraciones generadas con la heurística.

El método consiste en primero generar una configuración del *Tipo I*, (posición: *baricentro*, orientación: *cuaterniones*, eje de rotación: *vector normal al triángulo*), seleccionando un triángulo aleatorio de la malla de triángulos del obstáculo considerando la probabilidad acumulada de acuerdo a su área, tal y como se explica en la sección 3.5.1.

Posteriormente ésta configuración se coloca paralela con respecto al obstáculo, es decir se coloca uno de los tubos del clavo, que representa al robot, paralelo a uno de los tubos del clavo, que representa al obstáculo, como se muestra en la figura 4.8. Este método se encuentra definido en el conjunto de clases pertenecientes al sistema GEMPA.

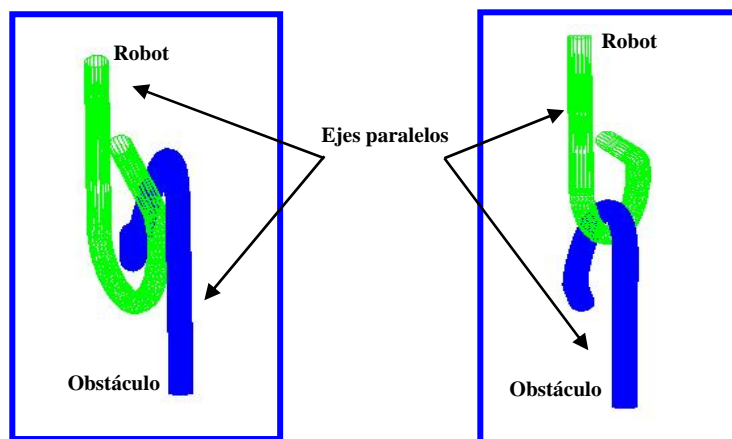


Figura 4.8 Ejemplos de configuraciones del robot paralelas con el obstáculo.

Una vez que ésta configuración está paralela, en esa misma posición se rota utilizando cuaterniones, construidos con el eje definido por el usuario en el archivo que contiene las características del espacio de trabajo. La configuración final tiene la posición definida en el *Tipo I* y la orientación obtenida por los cuaterniones, el siguiente diagrama esquematiza lo anterior:

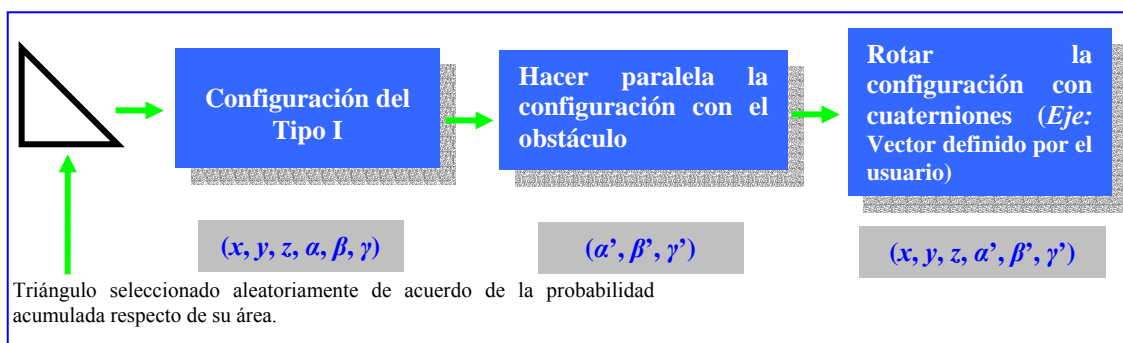


Figura 4.9 Combinación de estrategias para generar una configuración particular.



Figura 4.10 Ejemplos de configuraciones del tipo V.

Esta combinación de estrategias se diseñó con un propósito muy específico, el generar configuraciones como las mostradas en la figura 4.10, ya que éstas se identificaron en el modelo físico del problema de los clavos como las más adecuadas y promisorias para que el clavo salga, ligeramente desplazándose pero más aún rotando, de la configuración entrelazada en que se encuentra con el obstáculo.

Tal y como se explicó anteriormente estas configuraciones son necesarias para solucionar la versión original (*versión 1.0*) del problema de los clavos, sin embargo quizás también son necesarias para la *versión 1.1*, ya que aunque con los nuevos tipos (*V*, *VI* y *VII*) se encontraron configuraciones en el corredor (figura 4.11) éste parece ser muy reducido o tan exacto que dificulte al robot salir del obstáculo con solo trasladarse, como en las versiones *1.5* y *1.2*.

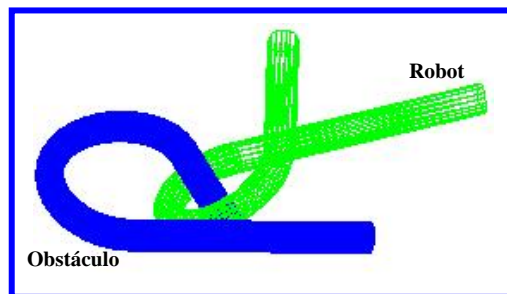


Figura 4.11 Configuración generada exactamente en el corredor de la *versión 1.1* utilizando la combinación de estrategias.

Aunque se tienen muy buenas configuraciones para la *versión 1.1* no se obtuvieron rutas con las pruebas realizadas, ya que éstas a pesar de que fueron muy pequeñas pues cuyo fin era solo el observar los tipos de configuraciones generados por los tipos *V*, *VI* y *VII*, no fueron lo suficientemente grandes (*número de configuraciones generadas*) para llevarnos a obtener una ruta.

Lo que nos lleva a asegurar que una ejecución grande del procedimiento descrito en la sección 4.4.1 con los nuevos tipos generados o bien otra combinación de los métodos probabilísticos implementados nos llevaría a obtener rutas para esta versión, sin embargo el costo en tiempo sería bastante considerable.