

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Màster:

AUTOMÀTICA I ROBÒTICA

Tesi de Màster

MÉTODO DE MUESTREO ADAPTATIVO BASADO EN EL ANÁLISIS DE  
COMPONENTES PRINCIPALES PARA LA PLANIFICACIÓN DE MOVIMIENTOS DE  
UN SISTEMA ROBÓTICO (BRAZO-MANO)

**Luis Yordano Cruz C.**

Director: Jan Rosell Gratacòs

Curs Acadèmic 2010/11

Marzo 2011

*Gracias a Dios por brindarme esta  
oportunidad tan hermosa.*

*A mi familia con todo mi amor.*

## **Agradecimientos**

Mis más sinceros agradecimientos a:

Profesor Dr. Jan Rosell Gratacòs, por todo el apoyo y confianza, durante el desarrollo de la presente tesis.

Profesor Dr. Raúl Suárez y Alexander Pérez, como coautores en la publicación de dos artículos, relacionados con el contenido de la presente tesis.

Profesor Dr. Vicenç Puig, por su cordial y atenta gestión, durante la realización del Master de Automática y Robótica de la Universidad Politécnica de Cataluña.

Familiares y amigos de los que siempre obtuve voces de aliento y confianza.

## Indice

Agradecimientos .....	i
Lista de Figuras.....	iv
Lista de Tablas .....	vi
Resumen.....	vii
1. Introducción.....	1
1.1. Motivación.....	2
1.2. Objetivos .....	2
1.3. Contenido .....	3
2. Preliminares .....	4
2.1. Espacio de Configuraciones .....	4
2.2. Planificación Probabilística.....	5
2.3. Algoritmo 1: PRM básico .....	7
2.4. Análisis de Componentes Principales .....	8
2.5. Planificación de Movimientos usando PCA.....	9
3. Método de Muestreo Basado en PCA .....	10
3.1. Algoritmo 2: Muestreo basado en PCA (versión 1.0) .....	11
3.2. Algoritmo 3: Evaluación Muestreo PCA .....	13
3.3. Algoritmo 4: Muestreo basado en PCA (versión 2.0) .....	18
3.4. Implementación .....	21
4. Aplicación a PRMs .....	22
4.1. PRM basado en Muestreo Gaussiano .....	22
4.2. Algoritmo 5: PRM Básico Gaussiano .....	23
4.3. PRM basado en Muestreo PCA .....	24
4.4. Algoritmo 6: PRM Básico PCA .....	25
4.5. PRM Gaussiano vs PRM PCA .....	26
5. Aplicación a PRMs para Sistemas Robóticos Brazo-Mano .....	30
5.1. Espacio de Configuraciones de un Sistema Robótico Brazo-Mano .....	31
5.2. Regiones de Muestreo Gaussiano .....	32
5.3. PRM Brazo-Mano Gaussiano.....	33
5.4. Algoritmo 7: PRM Brazo-Mano Gaussiano.....	34
5.5. Regiones de Muestreo PCA .....	35
5.6. PRM Brazo-Mano PCA .....	36
5.7. Algoritmo 8: PRM Brazo-Mano PCA .....	37

5.8. Implementación PRM Brazo-Mano Gaussiano y PCA .....	38
5.9. PRM Brazo-Mano Gaussiano vs PRM Brazo-Mano PCA .....	39
6. Conclusiones.....	45
Publicaciones.....	47
Bibliografía.....	48
Apéndice A.....	51
Código C++ .....	51

## Lista de Figuras

<b>Nº Figura</b>	<b>Título de la Figura</b>	<b>Pags.</b>
2.1	Los ángulos $q_1, q_2$ especifican la configuración de las dos articulaciones del robot.	5
2.2	Un ejemplo de un roadmap para un robot.	6
2.3	Un ejemplo de cómo resolver una consulta del roadmap de la figura 2.2.	7
2.4	Uso del PCA para reducir la dimensión del espacio de trabajo[16].	9
3.1	Región $R_s$ y la región $V_s$ , obtenida usando PCA, para 2 dimensiones.	10
3.2	Interfaz Gráfica del Problema A.	12
3.3	Interfaz Gráfica del Problema B.	12
3.4	Interfaz Gráfica del Problema C.	12
3.5	Pasajes Estrechos, cubiertos con circunferencias para los tres problemas.	13
3.6	Secuencia de muestreo en $R_s$ y $V_s$ , para el problema A.	14
3.7	Secuencia de muestreo en $R_s$ y $V_s$ , para el problema B.	14
3.8	Secuencia de muestreo en $R_s$ y $V_s$ , para el problema C.	15
3.9	Numero de Muestras requeridas como una función de $\alpha$ y $k$ para el problema A.	16
3.10	Numero de Muestras requeridas como una función de $\alpha$ y $k$ , para el Problema B.	17
3.11	Numero de Muestras requeridas como una función de $\alpha$ y $k$ , para el Problema C.	17
3.12	Secuencia de muestreo en $R_s$ y $V_s$ , para el problema A.	19
3.13	Secuencia de muestreo en $R_s$ y $V_s$ , para el problema B.	20
3.14	Secuencia de muestreo en $R_s$ y $V_s$ , para el problema C.	20
4.1	Secuencia de Muestreo Gaussiano.	22
4.2	Problema D.	26
4.3	Gráfico de Resultados para el Problema D.	27
4.4	Problema E.	27

<b>Nº Figura</b>	<b>Título de la Figura</b>	<b>Pags.</b>
4.5	Gráfico de Resultados para el Problema E.	28
4.6	Problema F.	28
4.7	Problema G.	29
5.1	Sistema Robótico Brazo-Mano (IOC).	30
5.2	Interfaz Gráfica The Kautham Project (IOC).	38
5.3	Espacio de Trabajo para las Simulaciones Brazo-Mano.	38
5.4	Impresión de los Resultados.	39
5.5	Problema H.	39
5.6	Gráfico de Resultados para el Problema H.	40
5.7	Problema I.	40
5.8	Gráfico de Resultados para el Problema I.	41
5.9	Problema J.	41
5.10	Problema K.	42
5.11	Posturas Brazo-Mano para encontrar la solución al problema H.	44
5.12	Posturas Brazo-Mano para encontrar la solución al problema I.	44

## Lista de Tablas

<b>Nº Tabla</b>	<b>Título de la Tabla</b>	<b>Pags.</b>
I	Numero de Muestras $N$ , generadas como una función de $\alpha$ y $k$ , para el problema A.	16
II	Numero de Muestras $N$ , generadas como una función de $\alpha$ y $k$ , para el problema B.	16
III	Numero de Muestras $N$ , generadas como una función de $\alpha$ y $k$ , para el problema C.	17
IV	Desempeño de la librería Armadillo	21
V	Resultados para el Problema D	26
VI	Resultados para el Problema E	27
VII	Resultados para el Problema F	28
VIII	Resultados para el Problema G	29
IX	Resultados para el Problema H	40
X	Resultados para el Problema I	41
XI	Resultados para el Problema J	42
XII	Resultados para el Problema K	42



# Resumen

---

El presente tema de tesis presenta un nuevo método de muestreo adaptativo basado en el análisis de componentes principales (PCA). El método de muestreo es iterativo y se basa en ir adaptando periódicamente la base del espacio de configuraciones del sistema robótico usado para la obtención de las muestras, a partir del análisis de componentes principales de las muestras libres de colisión obtenidas hasta el momento. Al método de muestreo se aplicara a un planificador del tipo PRM (*Probalistic RoadMap Method*).

El desarrollo, se ha realizado en dos partes, la primera parte consiste en implementar el método de muestreo propuesto, realizando ejemplos y pruebas para garantizar la viabilidad de dicho método.

La segunda parte está enfocada en optimizar la planificación de movimientos aplicando un PRM PCA (usa el método de muestreo desarrollado en la primera parte) el cual se aplica a un sistema robótico brazo-mano (brazo robot Stäubli TX90 de seis grados de libertad, equipado con una mano mecánica Shunk SAH de 13 grados de libertad), con una configuración final de aprehensión constreñida.

A sí mismo para realizar una comparación y verificar el rendimiento del PRM PCA, se ha implementado un PRM Gaussiano (basado en muestreo Gaussiano), el cual tiene características similares para solucionar problemas de planificación de movimientos en espacios reducidos o pasajes estrechos. Ambos PRMs (PCA y Gaussiano) han sido bien implementados dentro del entorno de programación para aplicaciones de planificación The Kautham Project, desarrollado en el Instituto de Organización y Control de Sistemas Industriales (IOC-UPC).

# Capítulo 1

## Introducción

---

¿Cómo puede decidir un robot qué movimientos debe realizar para lograr tareas en el mundo real?

Los sistemas de programación existentes de los robots industriales todavía tienen limitado sus capacidades de planificación de movimientos. Además el campo de la robótica está creciendo: exploración espacial, trabajo submarino, intervención en ambientes peligrosos, robótica en servicios. La planificación de movimientos aparece como uno de los componentes necesarios para la autonomía de los robots en tales contextos reales. También es un tema fundamental en la simulación de robots para ayudar a los diseñadores para determinar caminos libres de colisión, para robots que realizan tareas específicas.

Planificación de Caminos en robótica consiste en el diseño de un camino geométrico entre dos configuraciones dadas. El camino geométrico debe evitar todos los obstáculos presentes en el espacio físico, y también satisfacer cualquier restricción cinemática o dinámica del movimiento [1].

### Clasificación de Algoritmos

El problema de la planificación de caminos (path-planning) ha sido profundamente investigado en las últimas 3 décadas y ha sufrido bastantes transformaciones. A continuación se describen en forma resumida los principales algoritmos para el diseño de caminos:

- **Algoritmos Clásicos (completos):** se tiene como entradas tanto el estado inicial y final de robot como la geometría de los obstáculos. Un ejemplo de este tipo de algoritmos son los basados en campos potenciales. En este tipo de algoritmos, los obstáculos son modelados como cargas del mismo signo al robot móvil y el objetivo atrae a dicho robot. Lo malo de este tipo de algoritmos son tanto su coste computacional como su tendencia a caer en mínimos locales. Se han hecho esfuerzos en minimizar este tipo de inconvenientes con ayuda de algoritmos predictivos.

- **Algoritmos basados en muestreo:** se basan en recoger información de estados accesibles (que pertenecen al espacio libre de configuraciones) por el robot. Para conocer si el estado es accesible se consulta a un módulo de detección de colisiones. Una vez seleccionada una muestra válida, se intenta conectar con otras, formando mapas o árboles que capturan la conectividad del espacio libre.

## 1.1. Motivación

Actualmente los sistemas robóticos brazo-mano se encuentran en investigación en áreas como: Aprensión, Planificación de Movimientos, Teleoperación. Su elevado número de grados de libertad hace difícil su manejo. Los algoritmos basados en muestreo permiten abordar sistemas con un número elevado de grados de libertad, sin embargo en espacios constreñidos como los que se pueden encontrar en sistemas brazo-mano cerca de las configuraciones de agarre, el número de muestras requerido puede ser excesivamente elevado.

El Análisis de Componentes Principales (PCA: Principal Component Analysis), puede ayudar a mejorar la eficiencia del muestreo mediante la reducción del espacio de búsqueda.

## 1.2. Objetivos

El objetivo de la presente Tesis, es la mejora de la eficiencia de los métodos de planificación de movimientos basados en muestreo, como el método probabilístico de caminos (PRM) para la planificación de movimientos de un brazo robot equipado con una mano mecánica. Esta mejora se centra en facilitar la obtención de muestras libres de colisión alrededor de las configuraciones de aprehensión cuando el espacio libre del espacio de configuraciones del sistema es muy reducido o constreñido.

Para alcanzar el objetivo, se describe brevemente los dos aportes principales que se presentan en el tema de tesis:

- El primero consiste en presentar un nuevo método de muestreo adaptativo aplicando PCA; para obtener muestras libres de colisión de forma eficiente, independientemente del grado de complejidad del problema planteado.
- Para la realización del segundo aporte, se ha implementado un PRM, en el cual se aplica un muestreo basado en PCA iterativamente para obtener muestras

libres de colisión alrededor de las configuraciones de aprehensión, en espacios reducidos, de esta manera se logra optimizar el camino para alcanzar una posición deseada, dado una posición inicial y final. El PRM propuesto en la presente tesis ha sido aplicado inicialmente a problemas con pocos grados de libertad (2 y 3 GDL) y posteriormente a un sistema robótico brazo-mano para optimizar el camino en espacios reducidos o pasajes estrechos, centrado en la configuración de aprehensión constreñida, donde un muestreo aleatorio no consigue la solución.

### **1.3. Contenido**

La Tesis se encuentra estructurada de la siguiente manera:

Capítulo 1: Se presenta el marco científico, la motivación y se define los objetivos de la presente tesis, y su aplicación en el mundo real.

Capítulo 2: Se da un bosquejo general de la planificación de movimientos, dando énfasis a temas como la reducción de la dimensión en sistemas robóticos brazo-mano.

Capítulo 3: Se presenta un nuevo método de muestreo basado en el Análisis de Componentes Principales (PCA), y que con una alta probabilidad puede localizar configuraciones libre de colisiones en espacios reducidos o constreñidos, así mismo se presentan los resultados obtenidos.

Capítulo 4: Se describe la implementación de PRMs basados en muestreo Gaussiano y PCA, para la planificación de movimientos en robots con pocos grados de libertad, sobre problemas que implican pasajes estrechos, o espacios constreñidos, reportando el rendimiento, en base al tiempo que tardan en alcanzar la solución, el número de colisiones, y el total de configuraciones libres de colisiones.

Capítulo 5: Se presenta la extensión a la planificación de movimientos de un sistema robótico brazo-mano, con una configuración final de aprehensión en espacios reducidos o constreñidos, mediante la aplicación de PRMs basados en muestreo Gaussiano y PCA. Se reportan los resultados obtenidos de ambos métodos.

Publicaciones: artículos publicados durante el desarrollo de la presente tesis.

Apéndice A: Se describe el software desarrollado en la presente tesis.

# Capítulo 2

## Preliminares

---

### 2.1. Espacio de Configuraciones

El problema básico de la planificación de movimientos es conceptualmente muy simple usando las ideas del espacio de configuraciones ( $\mathcal{C}$ ).

Entonces antes de continuar debemos definir que es una configuración:

- Una configuración indica la pose que el robot puede tomar en el espacio de trabajo (*workspace*).

El  $\mathcal{C}$  de un robot consiste en el conjunto de todas las configuraciones que puede tomar.

La dimensión del  $\mathcal{C}$  esta dada por el número de grados de libertad del robot.

Por ejemplo consideremos un robot planar de dos grados de libertad (Figura 2.1), entonces  $\mathcal{C}$  tiene dimensión 2, y puede ser localmente representado por  $\mathbb{R}^2$  o más precisamente por el subconjunto:

$$\mathcal{C} = \{q = (q_1, q_2): q_1 \in [0, 2\pi), q_2 \in [0, 2\pi)\}$$

Esto garantiza que la representación sea inyectiva, es decir, que para un simple valor de  $q$  existe una postura del robot manipulador.

Ahora podemos definir el problema de la planificación de movimientos como el mapeo continuo, de configuraciones libres de colisiones (entre el robot y el obstáculo que conecta una configuración inicial con una configuración final). Se define el subespacio de configuraciones de obstáculos  $\mathcal{C}_{O_i}$ , como el conjunto de configuraciones que el robot intersecta con un obstáculo ( $\mathcal{W}O_i$ ), en el espacio de trabajo, es decir:

$$\mathcal{C}_{O_i} = \{q \in \mathcal{C} | R(q) \cap \mathcal{W}O_i \neq \emptyset\}$$

El espacio libre o espacio libre de configuraciones  $\mathcal{C}_{free}$  es el conjunto de configuraciones en la cual el robot no intersecta con ningún obstáculo, es decir:

$$\mathcal{C}_{free} = \mathcal{C} \setminus (\cup_i \mathcal{C}_{O_i})$$

Con esta notación una ruta libre se expresa como  $c: [0,1] \rightarrow cl(\mathcal{C}_{free})$ , en el cual  $cl(\mathcal{C}_{free})$  denota el cierre de  $\mathcal{C}_{free}$ . Una ruta libre no permite el contacto entre el robot y los obstáculos, mientras que una ruta semi-libre permite el contacto en la frontera de un obstáculo [1].

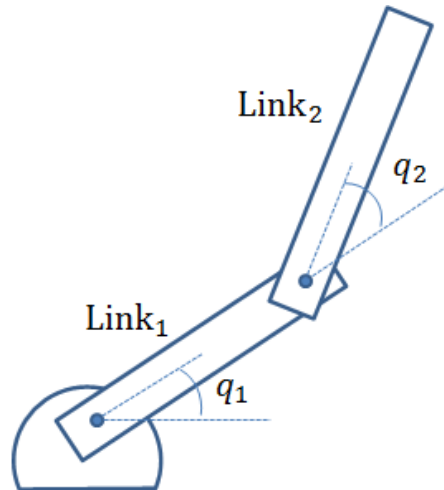


Figura 2.1. Los ángulos  $q_1, q_2$  especifican la configuración de las dos articulaciones del robot.

## 2.2. Planificación Probabilística

Los planificadores probabilísticos representan una clase de métodos de remarcable eficiencia especialmente en problemas que involucran espacios de configuraciones de grandes dimensiones. Pertenecen a la familia de métodos basados en muestreo, cuya idea básica consiste en determinar un conjunto finito de configuraciones libres de colisiones y su interconexión, que adecuadamente represente la conectividad de  $\mathcal{C}$  como un mapa de caminos (roadmap) que puede ser empleado para resolver problemas de planificación de movimientos [2]. En cada iteración se muestrea y se verifica la colisión entre el robot y los obstáculos del espacio de trabajo ( $\mathcal{W}$ ). Si hay colisión, se descarta la muestra. En caso contrario se adiciona al roadmap y se conecta si es posible con otras configuraciones almacenadas anteriormente.

Estos métodos se conocen como métodos probabilísticos de caminos (PRM : *Probabilistic Roadmap Method*), y se describe su funcionamiento detallado a continuación.

### 2.3. Algoritmo 1: PRM básico

El roadmap se representa por un grafo no dirigido  $G = (V, E)$ . Los nodos en  $V$  son un conjunto de configuraciones libres ( $\mathcal{C}_{free}$ ) del robot. Se asume que la generación de muestras es aleatoria con una distribución uniforme. Las aristas en  $E$  corresponden a una ruta; una arista  $(q_1, q_2)$ , corresponde a una ruta libre de colisiones conectando configuraciones  $q_1$  y  $q_2$ .

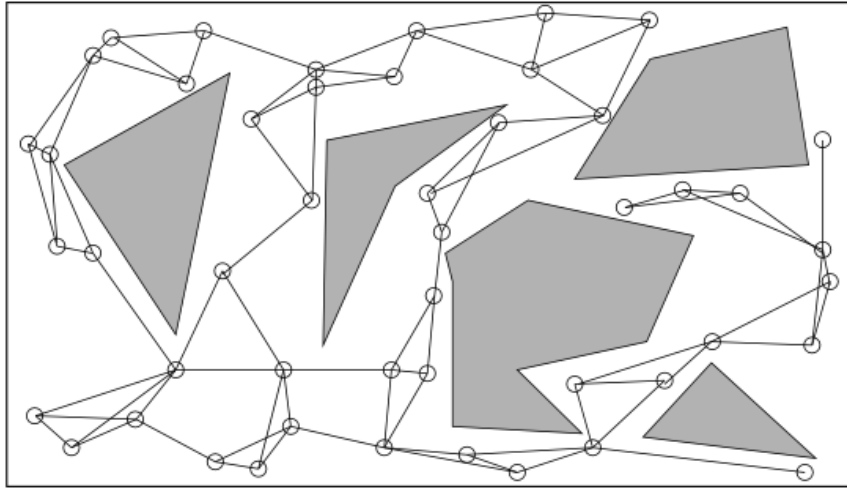


Figura 2.2: Un ejemplo de un roadmap para un robot (espacio de dos dimensiones en el espacio euclidiano). El área gris son obstáculos. Los círculos vacíos corresponden a los nodos del roadmap. Las líneas rectas entre los círculos corresponden a las aristas. El número de  $k$  vecinos más cercanos es tres [1].

#### Fase de Construcción del RoadMap

Sea  $\Delta$  el planificador local que sobre la entrada  $(q, q') \in \mathcal{C}_{free} \times \mathcal{C}_{free}$  retorna entre una ruta libre de colisiones de  $q$  a  $q'$  o NULL (si no puede encontrar una ruta). Se asume que  $\Delta$  es simétrico y determinista. Sea  $dist$  una función  $\mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}^+ \cup \{0\}$ , llamada función distancia, usualmente una métrica sobre  $\mathcal{C}$ . La fase de construcción se muestra en el algoritmo 1.

#### Fase de Consulta

En la fase de consulta, el roadmap es usado para resolver problemas individuales de planificación de movimientos.

Dado una configuración inicial  $q_{init}$  y una configuración final  $q_{goal}$ , el método prueba primero a conectar  $q_{init}$  y  $q_{goal}$  a dos nodos  $q'$  y  $q''$ , respectivamente, en  $V$ . Si hay éxito, el planificador busca el grafo  $G$  para una secuencia de aristas en  $E$  conectando  $q'$  y  $q''$  (Figura 2.3).

### Algoritmo 1: RoadMap (Fase de Construcción)

Entrada:

$n$ : numero de nodos para poner en el mapa de caminos

$k$ : numero de vecinos cercanos para examinar en cada configuración

Salida:

roadmap  $G = (V, E)$

$V \leftarrow \emptyset, \quad E \leftarrow \emptyset$

**while**  $|V| < n$  **do**

**repeat**

$q \leftarrow$  una configuración aleatoria en  $\mathcal{C}$

**until**  $q$  esta libre de colisiones

$V \leftarrow V \cup \{q\}$

**end while**

**for all**  $q \in V$  **do**

$N_q \leftarrow k$  vecinos más cercanos de  $q$  escogidos de  $V$  acorde a  $dist$

**for all**  $q' \in N_q$  **do**

**If**  $(q, q') \notin E$  **and**  $\Delta(q, q') \neq \text{NULL}$  **then**

$E \leftarrow E \cup \{(q, q')\}$

**end if**

**end for**

**end for**

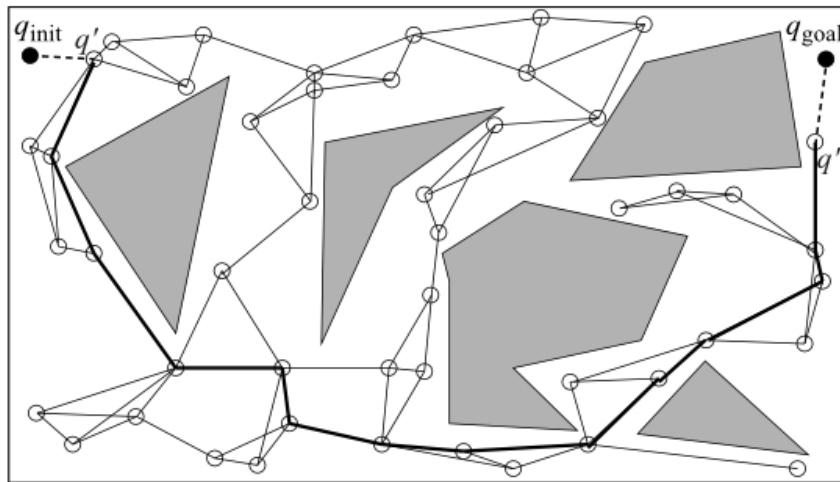


Figura 2.3: Un ejemplo de cómo resolver una consulta del roadmap de la figura 2.2. Las configuraciones  $q_{init}$  y  $q_{goal}$  son conectadas al roadmap a través de  $q'$  y  $q''$ , entonces un algoritmo búsqueda en grafos retorna la ruta más corta denotado por la línea más oscura [1].

### Métodos para sesgar el Muestreo

Para mejorar la eficiencia de los PRMs es conveniente obtener muestras que permitan capturar la conectividad de  $\mathcal{C}_{free}$ .

A continuación, se enumeran las estrategias de muestreo que persiguen este objetivo las cuales han sido bien clasificadas por Hsu [3] en:



- a) Estrategias basadas en el espacio de trabajo (workspace): Las que utilizan información del espacio trabajo para determinar las regiones de interés (por ejemplo [4], [5]).
- b) Estrategias de filtrado: Las que actúan sobre-muestreando  $\mathcal{C}$  pero filtran rápidamente cualquier configuración no válida (por ejemplo [6], [7]).
- c) Estrategias Adaptativas: Las que utilizan la información acumulada durante la construcción del roadmap (por ejemplo [8], [9]).
- d) Estrategias de Deformación: Los que deforman (dilatan)  $\mathcal{C}$  para hacerlo más expansivo y así captar más fácilmente su conectividad (por ejemplo [10], [11]).

## 2.4. Análisis de Componentes Principales

Otra forma de mejorar la eficiencia es reduciendo la dimensión, es decir definiendo los submanifolds de  $\mathcal{C}$  donde esta la solución (o donde la solución se puede encontrar con más facilidad). Entre las técnicas de reducción de dimensión, el Análisis de Componentes Principales (PCA) ha sido utilizado para sintetizar el movimiento humano en aplicaciones gráficas [13], y también captar el acoplamiento entre las articulaciones de los dedos de manos mecánicas antropomórficas (para la búsqueda de configuraciones de aprehensión [14], o para la planificación movimientos de un sistema brazo-mano [16]).

El Análisis de Componentes Principales (PCA: Principal Component Analysis) es una técnica estadística de síntesis de la información, o reducción de la dimensión (número de variables). Es decir, ante un banco de datos con muchas variables, el objetivo será reducirlas a un menor número perdiendo la menor cantidad de información posible.

Hay diferentes maneras de llevar a cabo el PCA [16]. Básicamente, se puede hacer mediante el cálculo de la descomposición de valores propios de la matriz covarianza de datos o la descomposición de una matriz de datos de valores singulares, generalmente después de centrar la media por cada atributo de los datos. Cuanto mayor sean los valores propios o los valores singulares, la dispersión de los datos será mayor, a lo largo de la dirección de los vectores propios correspondientes; los vectores propios son directamente utilizados para definir las direcciones de la nueva base.

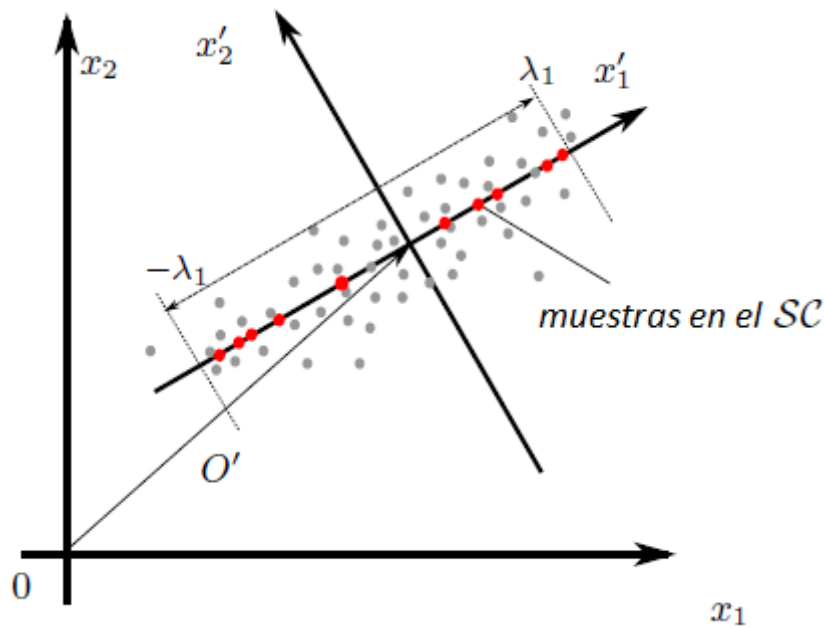


Figura 2.4. Uso del PCA para reducir la dimensión del espacio de trabajo [16].

## 2.5. Planificación de Movimientos usando PCA

En [15] y [16] se ha usado PCA en la planificación de movimientos de un sistema robótico brazo-mano para reducir la dimensión de la mano mecánica y así tratar el problema con menos grados de libertad. A continuación se describe brevemente el procedimiento propuesto de este enfoque:

- Captura de posturas de la mano humana, mediante un guante sensorizado CyberGlove, usado por un operador realizando movimientos naturales en el espacio libre.
- Mapeo al espacio de trabajo de las articulaciones de una mano mecánica SAH con 17 articulaciones y 13 grados de libertad.
- Aplicación de PCA: los tres primeros vectores cubren un 85% del total de la varianza del conjunto de datos analizados. Entonces aplicando la idea de reducir la dimensión solo se utilizan estos tres para generar muestras en el espacio de la mano mecánica, más los seis grados de libertad de un brazo robot Staubli TX 90, para buscar una solución usando un PRM.

# Capítulo 3

## Método de Muestreo Basado en PCA

Al implementar un muestreo aleatorio con una distribución uniforme y aplicarlo al algoritmo PRM, este algoritmo resulta probabilísticamente completo. Esto quiere decir que si existe una solución al problema planteado, la probabilidad de que el algoritmo alcance una solución tiende a uno. En el presente tema de tesis se presenta un nuevo método de muestreo basado en PCA, el cual garantiza una alta probabilidad de encontrar configuraciones libres de colisiones en áreas difíciles de  $\mathcal{C}$  con el objetivo de poder encontrar solución con el menor número de muestras.

A continuación se describe su funcionamiento:

- Dado una región  $R_S$  de  $\mathcal{C}$ , se aplica un muestreo aleatorio y se obtienen un conjunto mínimo de muestras (igual a la dimensión del  $\mathcal{C}$ ) de  $\mathcal{C}_{free}$ .
- Se usa el PCA, y se determina una región  $V_S$ , para obtener nuevas muestras con una alta probabilidad de pertenecer a  $\mathcal{C}_{free}$ .
- Iterativamente, se muestrea en  $R_S$  y  $V_S$ , que se va recalculando con todas las muestras libres obtenidas (Figura 3.1).

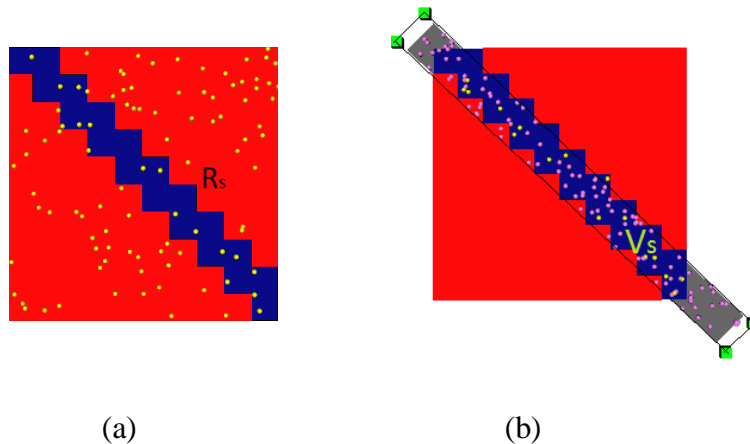


Figura 3.1: Región  $R_S$  y la región  $V_S$ , obtenida usando PCA, para 2 dimensiones.

El procedimiento de muestreo se describe en el algoritmo 2:

### 3.1. Algoritmo 2: Muestreo basado en PCA (versión 1.0)

---

Entrada:

$R_s$ : Región de  $\mathcal{C}$

$V_s$ : Región de muestreo (calculado con el PCA).

$S$ : Un conjunto de al menos  $d$  muestras libres de colisión de  $R_s$ .

$\alpha$ : Un valor en un rango de  $[0, 1]$ , probabilidad de la muestra

$k$ : El número de configuraciones muestreadas en el tiempo

---

Salida:

$S$ : con  $k$  nuevas muestras libre de colisiones.

$V_s$ : actualizado usando las nuevas muestras libre de colisiones

---

$r = \text{RAND}()$

**if**  $r \geq \alpha$  **then**

$S_k = \text{SAMPLE-FROM}(R_s, k)$

**else**

$S_k = \text{SAMPLE-FROM}(V_s, k)$

**end if**

**if**  $\text{CARD}(S_k) > 0$  **then**

$V_s = \text{PCA}(S \cup S_k)$

**end if**

$S = S \cup S_k$

**return**  $S, V_s$

---

El algoritmo 2 se inicia generando un número aleatorio  $r$  entre  $[0,1]$ , el cual se compara con el valor de  $\alpha$ , si  $r$  es mayor que  $\alpha$  el muestreo se realiza en la región  $R_s$ , en caso contrario el muestreo se realiza en la región  $V_s$ , luego en ambos casos se comprueba si se han obtenido configuraciones libres de colisiones con la función  $\text{CARD}(S_k)$ , que devuelve la cardinalidad del conjunto  $S_k$ , y si es mayor que cero, se llama a la función  $\text{PCA}(S \cup S_k)$ , la cual calcula el análisis de componentes principales del conjunto  $S \cup S_k$  y retorna una nueva región de muestreo  $V_s$ , en caso contrario no se calcula el PCA. El algoritmo retorna el conjunto  $S$  ampliado con las nuevas muestras libres de colisiones,  $S_k$ , obtenidas de  $R_s$  o de  $V_s$  y la nueva región  $V_s$ .

El cálculo del PCA, genera una nueva región  $V_s$  (hiper-caja), que está centrada en la media de los datos y con una longitud de cada uno de los lados igual a dos veces la desviación estándar en el eje correspondiente.

La función  $\text{SAMPLE-FROM}(A, k)$ , obtiene  $k$  muestras aleatorias de la Región  $A$  ( $R_s$  o de  $V_s$ ) y devuelve las que son libres.

## Evaluación

A continuación se presentan tres problemas ilustrativos (Figura 3.2, 3.3 y 3.4), en los cuales se muestra el funcionamiento del algoritmo explicado anteriormente.

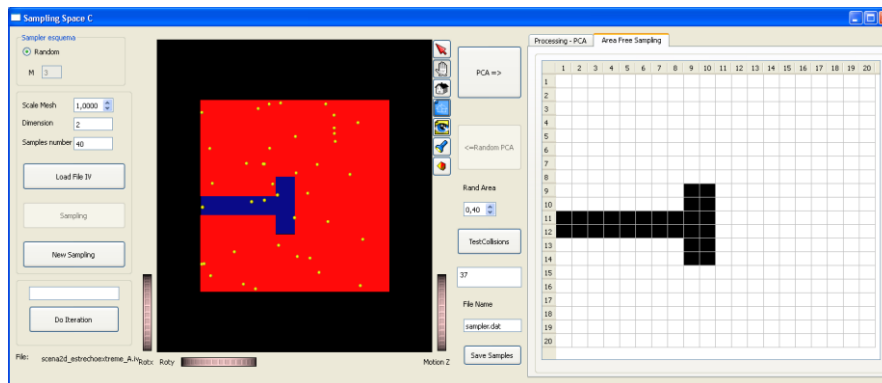


Figura 3.2. Interfaz Gráfica del Problema A.

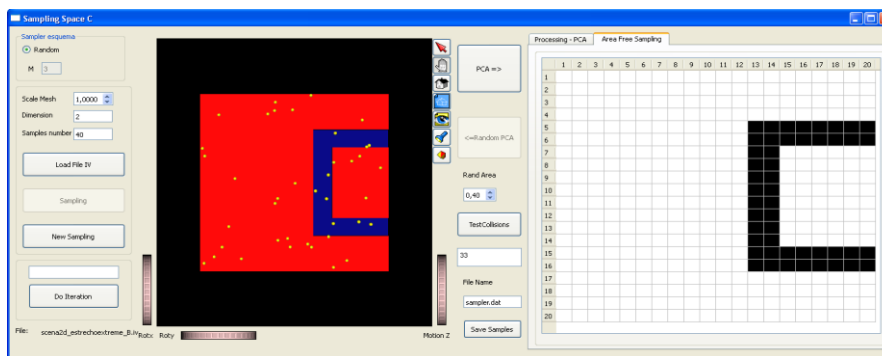


Figura 3.3. Interfaz Gráfica del Problema B.

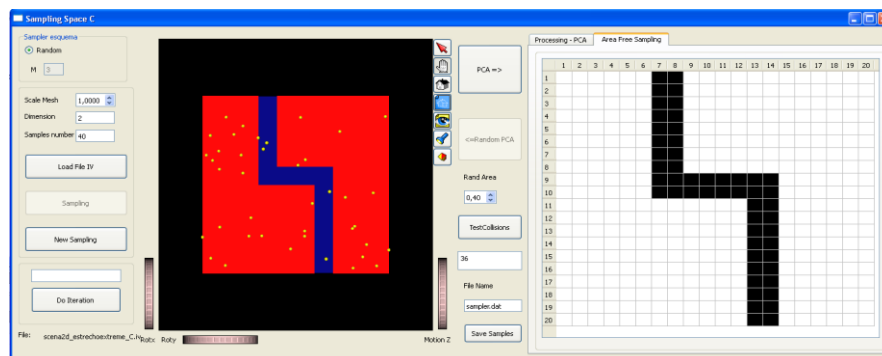


Figura 3.4. Interfaz Gráfica del Problema C.

Para la evaluación de los tres problemas, el pasaje estrecho ha sido cubierto con circunferencias de radio  $R$  igual al espacio libre del pasaje, cuyos centros están situados en el eje central del pasaje y una distancia  $R$  (Figura 3.5). Entonces, el parámetro para evaluar el procedimiento de muestreo es el número de muestras  $N$  para cubrir cada circunferencia con al menos una muestra, es decir el número de muestras necesarias dada una dispersión requerida de las muestras en el pasaje

estrecho (con estas muestras en el pasaje estrecho, un roadmap puede conectar la entrada a la salida, aunque pueda ser logrado con menos muestras).

Este método de evaluación es sencillo, pero útil para la demostración. La función EVALUATE(S), retorna TRUE cuando todas las circunferencias están cubiertas, o en caso contrario retorna FALSE (Algoritmo 3).

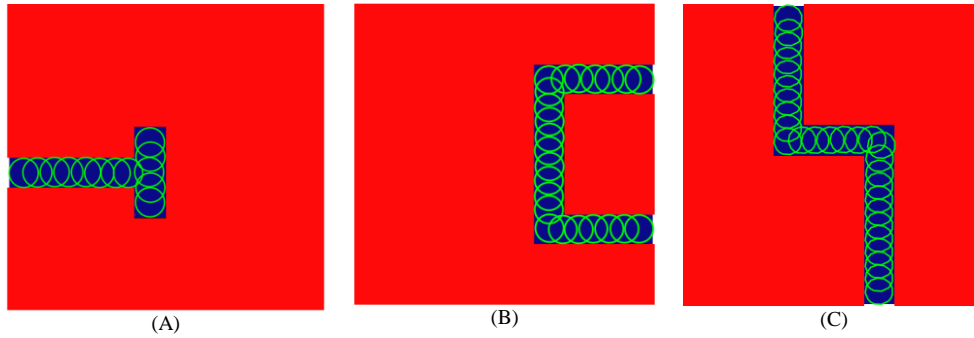


Figura 3.5. Pasajes Estrechos, cubiertos con circunferencias para los tres problemas.

### 3.2. Algoritmo 3: Evaluación Muestreo PCA

Entrada:

$R_s$ : Región de  $\mathcal{C}$

$\alpha$ : Un valor en un rango de  $[0, 1]$

$k$ : El número de configuraciones muestreadas en el tiempo

Salida:

$N$ : Número total de muestras generadas.

$S = \text{SAMPLE-FROM}(R_s, k)$

**if**  $\text{CARD}(S) < d$  **then**

**repeat**

$S = \text{SAMPLE-FROM}(R_s, k)$

**until**  $\text{CARD}(S) \geq d$

**end if**

$V_s = \text{PCA}(S)$

$N = d$

**repeat**

$(S, V_s) = \text{Muestreo basado en PCA}(R_s, V_s, S, \alpha, k)$

$N = N + k$

**until**  $\text{EVALUATE}(S) = \text{TRUE}$

**return**  $N$

A continuación se muestra, la secuencia del muestreo en  $R_s$  y  $V_s$  para los tres problemas, aplicando el algoritmo de evaluación (Figura 3.6, 3.7 y 3.8). Los parámetros son  $k = 40$  y  $\alpha = 0.4$ .

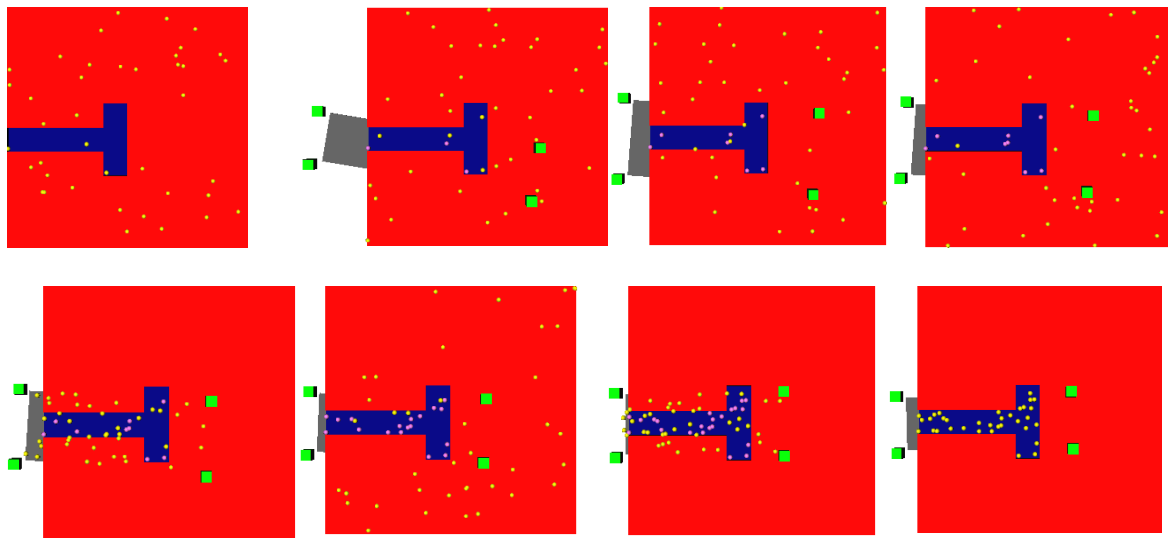


Figura 3.6. Secuencia de muestreo en  $R_s$  y  $V_s$ , para el problema A.

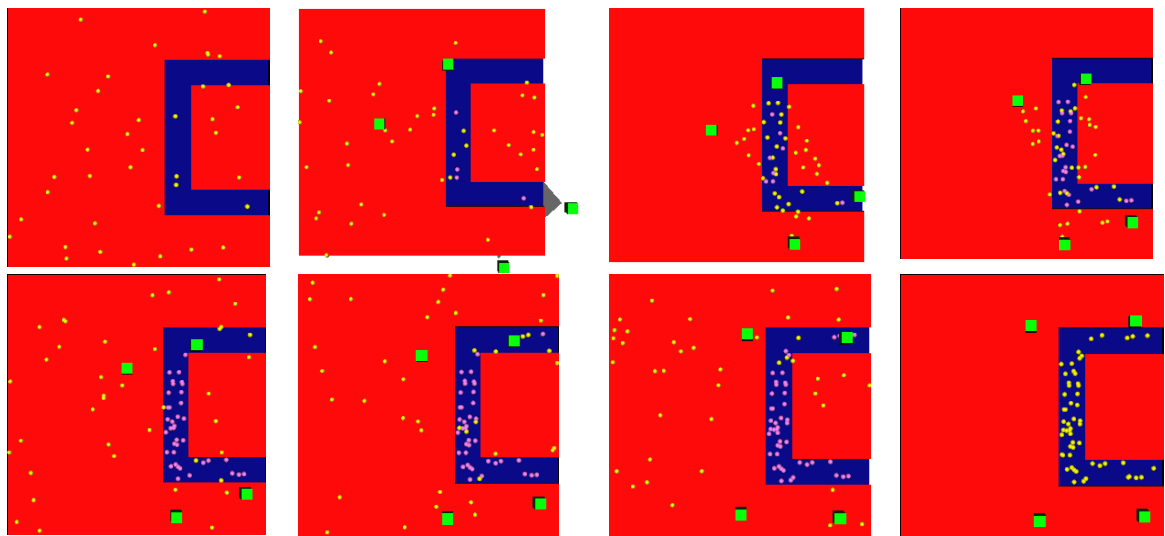


Figura 3.7. Secuencia de muestreo en  $R_s$  y  $V_s$ , para el problema B.

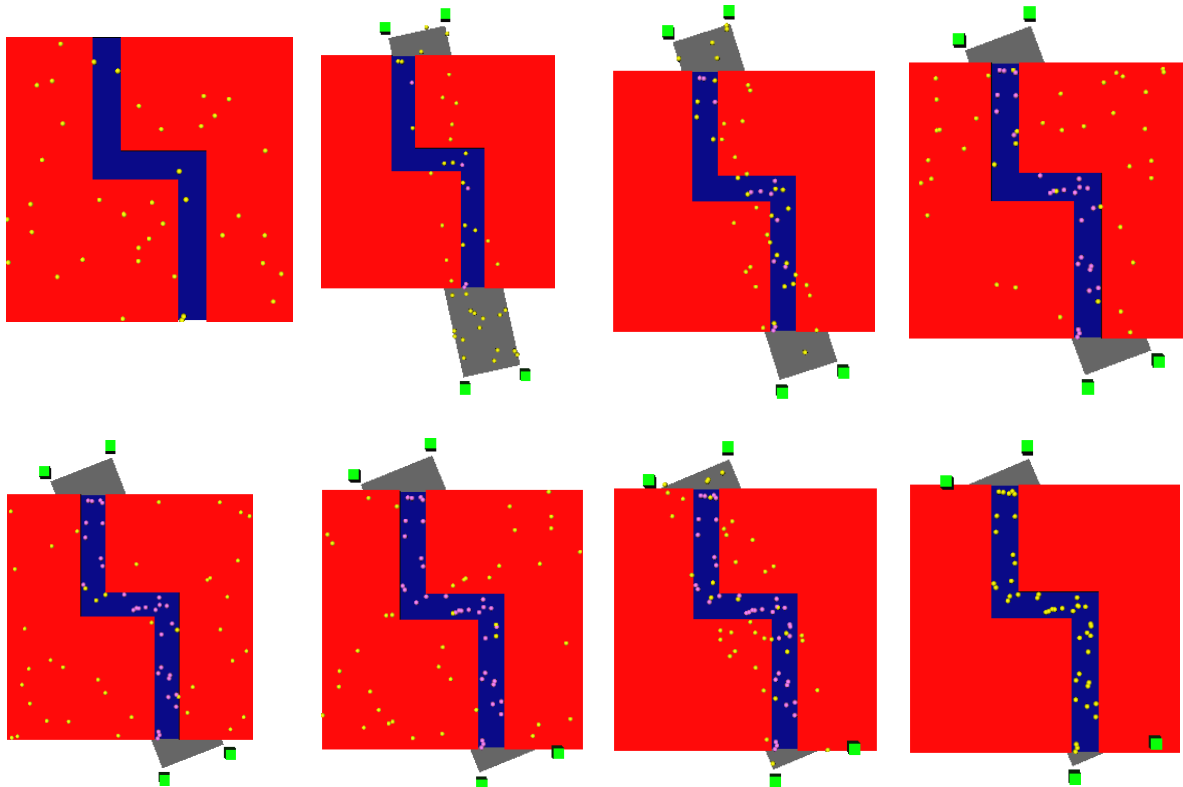


Figura 3.8. Secuencia de muestreo en  $R_s$  y  $V_s$ , para el problema C.

A continuación se describen las pruebas y los resultados obtenidos de los tres problemas de pasajes estrechos aplicando el nuevo método de muestro adaptativo usando PCA.

#### Problema A:

Usando un muestreo aleatorio uniforme, el número de muestras requeridas para resolver el problema ha sido  $N=430\pm 22$ , este valor se compara con el valor requerido usando el algoritmo de muestreo basado en PCA. El valor de  $N$  ha sido calculado con diferentes valores de  $\alpha$  y  $k$ . Los casos de  $\alpha=0$  y  $\alpha=1$ , no son considerados por que corresponden a muestreo aleatorio uniforme, y muestreo solo en  $V_s$ .

Los resultados se muestran en la Tabla I y la representación gráfica en la Figura 3.9, y como se puede apreciar, el método propuesto siempre reduce el número de muestras para cubrir todo el espacio del pasaje estrecho, con un rendimiento de 43% (se necesita solo el 43% de las muestras que requiere el muestreo aleatorio puro) obtenido cuando  $\alpha=0.8$  y  $k=80$ .



K	Alfa				
	0.2	0.4	0.5	0.6	0.8
20	254±37	236±35	230±52	256±40	280±22
40	240±47	224±39	225±23	244±29	248±50
80	220±64	222±65	220±55	218±56	216±55

Tabla I: Número de Muestras  $N$ , generadas como una función de  $\alpha$  y  $k$ , para el problema A.

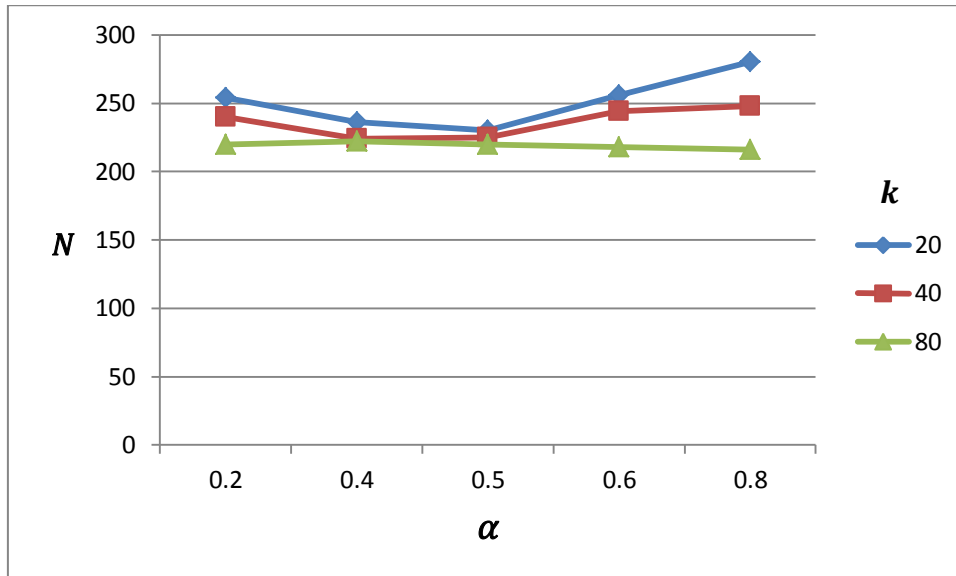


Figura 3.9: Número de Muestras requeridas como una función de  $\alpha$  y  $k$ , para el problema A

### Problema B:

De manera análoga al problema A, se ha realizado la evaluación del problema B donde las muestras para resolver el problema con muestreo uniforme ha sido  $N=420\pm34$ , con un rendimiento del 41% (se necesita solo el 41% de las muestras que requiere el muestreo aleatorio puro), cuando  $\alpha=0.8$  y  $k=80$ , los datos se muestran en la Tabla II y su gráfico correspondiente en la Figura 3.10.

K	Alfa				
	0.2	0.4	0.5	0.6	0.8
20	324±71	322±51	326±37	340±33	336±47
40	314±67	312±34	312±25	320±36	308±31
80	290±54	304±53	286±30	278±62	250±43

Tabla II: Número de Muestras  $N$ , generadas como una función de  $\alpha$  y  $k$ , para el problema B.

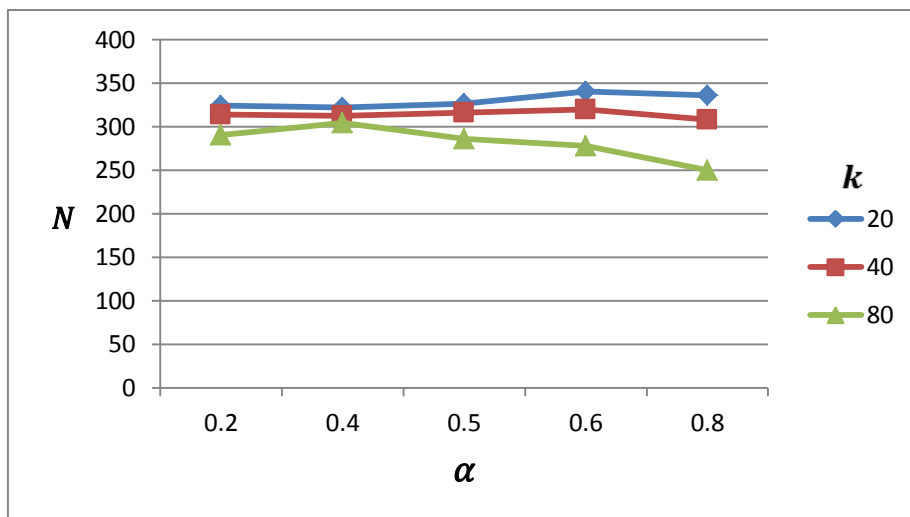


Figura 3.10: Número de Muestras requeridas como una función de  $\alpha$  y  $k$ , para el Problema B.

Problema C:

Así como en los problemas A y B, el método propuesto basado en PCA, alcanza un rendimiento de 57% (se necesita solo el 57% de las muestras que requiere el muestreo aleatorio puro) cuando  $\alpha=0.8$  y  $k=80$  y con un muestreo aleatorio son necesarias  $N=484\pm 29$  muestras, los datos para diferentes valores de  $\alpha$  y  $k$  se muestran en la Tabla III y respectivo gráfico en la Figura 3.11.

K	Alfa				
	0.2	0.4	0.5	0.6	0.8
20	388±82	366±53	344±55	350±42	368±50
40	376±40	308±49	268±43	228±47	224±26
80	336±73	320±51	224±36	216±35	208±33

Tabla III: Número de Muestras  $N$ , generadas como una función de  $\alpha$  y  $k$ , para el problema C.

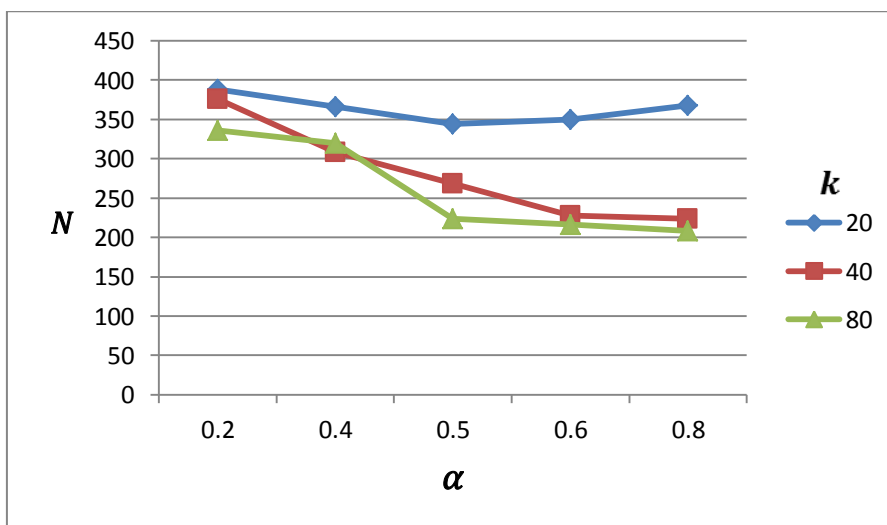


Figura 3.11: Número de Muestras requeridas como una función de  $\alpha$  y  $k$ , para el Problema C.

Con los ejemplos ilustrados anteriormente se puede notar lo siguiente:

- Usando valores grandes de  $k$  disminuye el número de muestras necesarias. La razón es que cuando se utilizan valores grandes de  $k$ , el primer volumen de muestreo  $V_S$  obtenido con el PCA cubre bastante bien el pasaje estrecho, y por lo tanto los mejores resultados se obtienen cuando la probabilidad de muestreo en  $V_S$  es más alta (es decir, para valores grandes de  $\alpha$ ).
- En cambio cuando se usan valores pequeños de  $k$ , el valor más bajo de  $N$  se obtiene para valores de  $\alpha$  igual o inferior a 0,5. En este caso, el volumen  $V_S$  calculado inicialmente no es muy bueno, ya que se obtiene con un conjunto de configuraciones no muy representativa, es decir las muestras generadas aleatoriamente no cubren lo suficientemente bien el pasaje estrecho, y por lo tanto la búsqueda de más configuraciones fuera de  $V_S$  es necesario (muestreo en  $R_S$ ). Esto se puede lograr con un valor más bajo de  $\alpha$  con el fin de aumentar la probabilidad de tomar muestras en la región  $R_S$ . Sin embargo el valor de  $\alpha$ , no debe ser demasiado pequeño, porque entonces la información capturada por el PCA prácticamente no se utilizan y los resultados tienden a acercarse a los obtenidos con muestreo aleatorio uniforme. Por el contrario para valores grandes de  $\alpha$  y  $k$  muy pequeño, el método no puede lograr la dispersión determinada utilizando una cantidad razonable de muestras, por lo cual las pruebas se inician a partir de veinte muestras.

### 3.3. Algoritmo 4: Muestreo basado en PCA (versión 2.0)

---

Entrada:

$R_S$ : Región de  $\mathcal{C}$

$S$ : Un conjunto de al menos  $d$  muestras libres de colisión de  $R_S$ .

$K_V, K_R$ : Número de muestras de configuraciones libres de colisiones de  $V_S, R_S$ ; respectivamente.

---

Salida:

$S$ : actualizado con hasta  $K_V + K_R$  nuevas muestras libre de colisiones.

---

$V_S = \text{PCA}(S)$

$S_{k_R} = \text{SAMPLE-FROM}(R_S, K_R)$

$S = S \cup S_{k_R}$

**for**  $i=1$  **to**  $K_V$  **do**

$s = \text{SAMPLE-FROM}(V_S, 1)$

**if**  $s \in R_S$  **then**

$S = S \cup s$

**end if**

**end for**

**return**  $S$

---

Después de realizar las pruebas de los tres ejemplos, y notar las características cuando se variaba los valores  $\alpha$ , y  $k$ , se muestra a continuación una variación del algoritmo 2, el cual se describe en el algoritmo 4:

La principal diferencia entre el algoritmo 4 y el algoritmo 2, radica en que ahora en el algoritmo 4, se pide explícitamente al algoritmo el número de muestras libres de colisiones tanto en la región  $R_S$  como en la región  $V_S$ , en forma intercalada hasta alcanzar la solución al problema. El algoritmo retorna el conjunto  $S$  que contiene el número de muestras libres de colisiones que pertenecen a la región  $R_S$ . También se realiza una comprobación que las muestras libres de  $V_S$  pertenezcan a la región  $R_S$ , en caso contrario la muestra es descartada.

A continuación se muestran las simulaciones realizadas para los tres problemas tratados anteriormente, los valores para  $K_R = 10$ ,  $K_V = 10$  y una muestra inicial de 40 muestras.

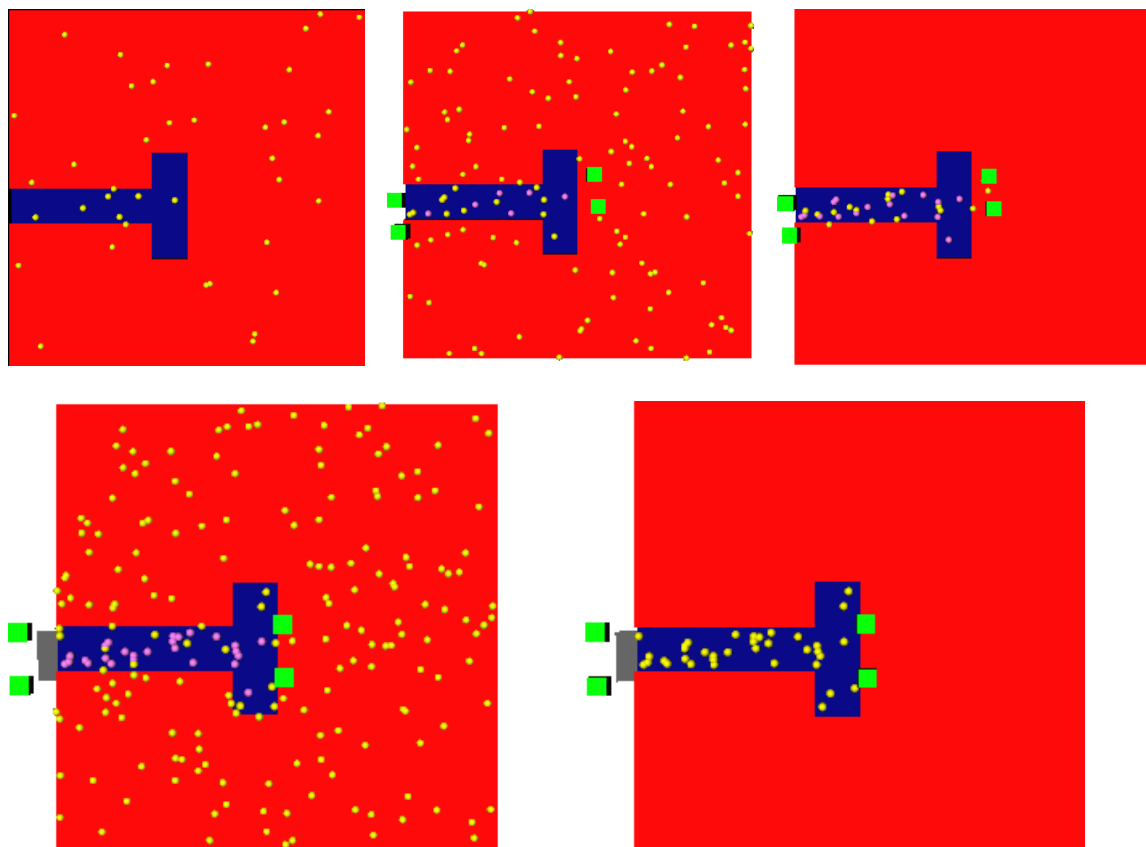


Figura 3.12. Secuencia de muestreo en  $R_S$  y  $V_S$ , para el problema A.

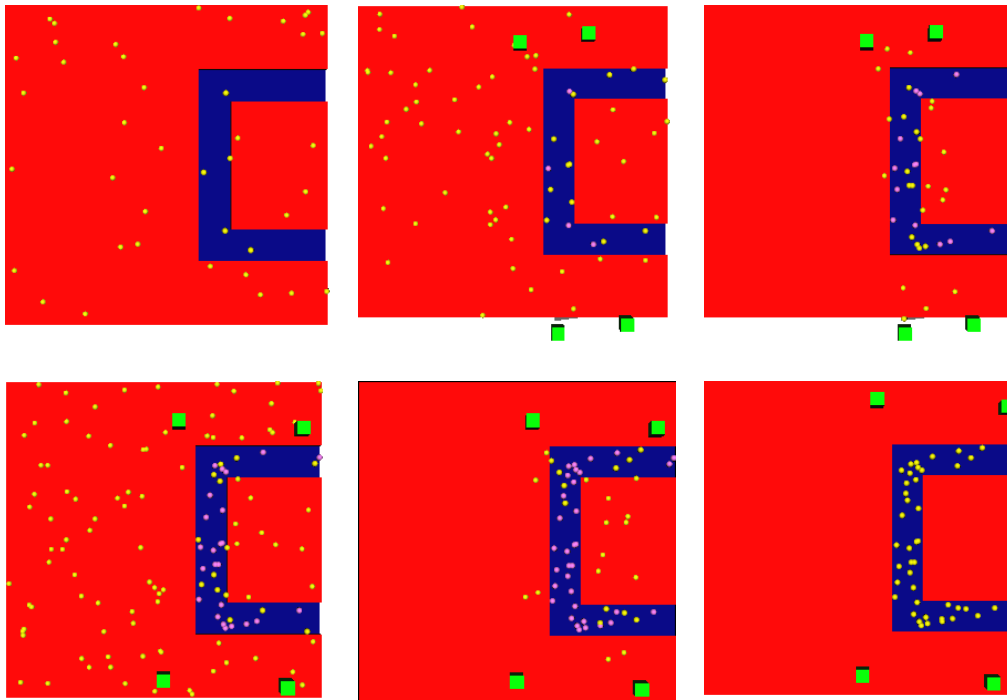


Figura 3.13. Secuencia de muestreo en  $R_s$  y  $V_s$ , para el problema B.

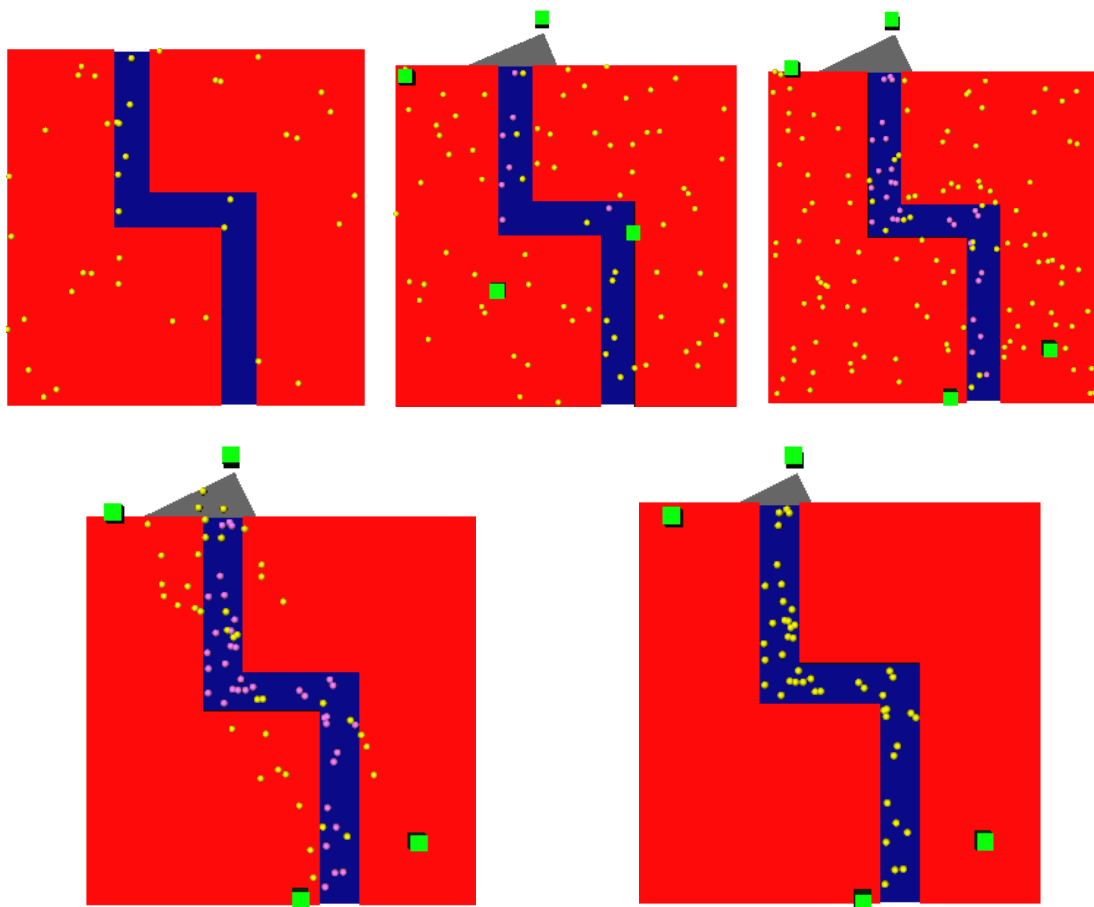


Figura 3.14. Secuencia de muestreo en  $R_s$  y  $V_s$ , para el problema C.

### 3.4. Implementación

El enfoque propuesto ha sido implementado en C++ utilizando librerías de código abierto y multiplataforma:

- Qt [18] (cross-platform application and UI framework) para la interfaz de usuario.
- Coin3D [19] (real-time 3D visualization), para el entorno gráfico.
- PQP [20] (A Proximity Query Package), para la detección de colisiones.

Para el cálculo del PCA una posibilidad era utilizar Octave [25] o R [26] con una interface como el RCPP [27]. Sin embargo, la aplicación requiere realizar el PCA dentro del bucle de muestreo, y los criterios de desempeño son un factor clave, y por esta razón se ha usado Armadillo C++ Linear Algebra Library [28]. Esta librería es de código abierto y tiene un buen desempeño en tiempo de respuesta para grandes volúmenes de datos, las pruebas realizadas se muestran en la Tabla IV.

<b>G.D.L.</b>	<b>2</b>	<b>3</b>	<b>5</b>	<b>7</b>	<b>9</b>	<b>11</b>	<b>13</b>
Tiempo de Procesamiento PCA(ms)	0.14	0.26	0.55	1.04	1.51	2.20	3.01

Tabla IV: Desempeño de la librería Armadillo para la ejecución de conjuntos de 10.000 muestras con diferente número de grados de libertad.

# Capítulo 4

## Aplicación a PRMs

A continuación se presentan dos enfoques probabilísticos, para solucionar problemas de planificación de movimientos en espacios reducidos basado en dos tipos de muestreo:

- PRM basado en Muestreo Gaussiano [6]
- PRM basado en Muestreo Adaptativo usando PCA

### 4.1. PRM basado en Muestreo Gaussiano

En [6], se planteó un enfoque de muestreo Gaussiano para la planificación de movimientos, el cual cubre un aspecto crítico de los PRMs como estrategia de muestreo en el espacio libre de configuraciones (espacios reducidos o estrechos).

En la presente tesis, se ha realizado una implementación de un PRM básico para solidos que pueda trasladarse y rotar libremente en el espacio (free-flying robot) usando muestreo Gaussiano, el cual se detalla en el Algoritmo 5.

El Algoritmo 5, se inicia calculando un valor aleatorio de  $c_t, c_r$  (traslación y rotación) para la primera muestra, y para la segunda muestra se calcula una distancia  $d$  desde  $c_t$ , esta distancia  $d$  es elegida acorde una distribución gaussiana con media 0 y una desviación estándar de 1, una vez obtenidas las dos muestras se comparan si al menos una pertenece al espacio libre de colisiones. La Figura 4.1 muestra la secuencia del muestreo, en caso de no cumplir con la condición; son descartadas las dos muestras, si se ha obtenido una muestra libre se comprueba si esta configuración resuelve el problema, el algoritmo se repite hasta un máximo de número muestras y retorna 1, si ha encontrado una ruta válida para conectar  $c_{ini}$  y  $c_{goal}$ , en caso contrario devuelve 0, lo cual indica que no ha encontrado la solución al problema.

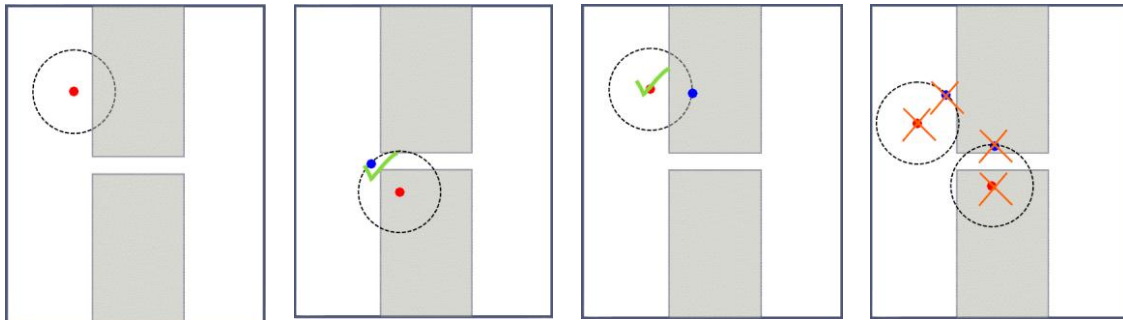


Figura 4.1. Secuencia de Muestreo Gaussiano.

## 4.2. Algoritmo 5: PRM Básico Gaussiano

---

Entrada:

$c_{ini}$  : Configuración inicial  
 $c_{goal}$  : Configuración final  
 $c_t$  : parámetro de traslación del robot  
 $c_r$  : parámetro de rotación del robot  
 $c_1$  : una configuración aleatoria ( $c_t, c_r$ )  
 $d$  : distancia elegida acorde a una distribución normal.  
 $c_2$  : una configuración ( $c'_t, c_r$ ), donde  $c'_t$  está a una distancia  $d$  de  $c_t$

---

Salida:

$P$  ruta a conectar  $c_{ini}$  y  $c_{goal}$

---

```
numsamples = 0
while numsamples < MaxSamples do
     $c_t, c_r = \text{RAND\_U}()$ 
     $c_1 = (c_t, c_r)$ 
     $d = \text{RAND\_G}(0,1)$ 
     $c'_t = c_t + d$ 
     $c_2 = (c'_t, c_r)$ 
    numsamples = numsamples + 2
    if  $c_1 \in \mathcal{C}_{free}$  and  $c_2 \notin \mathcal{C}_{free}$  then
         $S = S \cup c_1$ 
        PRM_ADD( $c_1$ )
         $P = \text{PRM\_SOLVE}(c_1)$ 
        if  $P \neq \emptyset$  then
            return P
        end if
    else if  $c_2 \in \mathcal{C}_{free}$  and  $c_1 \notin \mathcal{C}_{free}$  then
         $S = S \cup c_2$ 
        PRM_ADD( $c_2$ )
         $P = \text{PRM\_SOLVE}(c_2)$ 
        if  $P \neq \emptyset$  then
            return P
        end if
    else discard both
    end if
end while
return  $\emptyset$ 
```

---

El Algoritmo 5 describe el procedimiento de un PRM básico basado en muestreo Gaussiano en el cual se ha utilizado la siguiente nomenclatura y funciones:

- $\text{RAND\_U}()$ : Devuelve una nueva posición y orientación aleatoria del robot.
- $\text{RAND\_G}(0,1)$ : Devuelve un número aleatorio con una distribución Gaussiana.



- $PRM\_ADD(s)$ : adiciona una muestra libre de colisiones al PRM, y la conecta si es posible a las muestras vecinas.
- $PRM\_SOLVE(s)$ : devuelve 1, si ha encontrado una ruta desde  $c_{ini}$  y  $c_{goal}$ , en caso contrario devuelve 0.

El procedimiento de muestreo Gaussiano tiene cinco parámetros:

- $c_t$ : parámetro de traslación del robot
- $c_r$ : parámetro de rotación del robot
- $c_1$ : una configuración aleatoria ( $c_t, c_r$ )
- $d$ : distancia elegida acorde a una distribución normal.
- $c_2$ : una configuración ( $c'_t, c_r$ ), donde  $c'_t$  está a una distancia  $d$  de  $c_t$

### 4.3. PRM basado en Muestreo PCA

El algoritmo 6, presenta un PRM básico usando un muestreo adaptativo PCA, para solucionar problemas de planificación de movimientos en espacios reducidos o pasajes estrechos. A continuación se describe su funcionamiento.

El algoritmo se inicia calculando un conjunto de muestras libres de colisiones aleatorias igual a la dimensión del problema ( $d$ ). Una vez obtenidas estas muestras se adicionan al grafo PRM y con estas muestras iniciales ( $S$ ) se calcula el PCA. A continuación obtenemos un nuevo conjunto muestras de  $V_s$  las cuales se procesan de forma unitaria hasta alcanzar  $N_v$  muestras libres. Estas se comprueban si pertenecen a la región de muestreo  $R_s$  así como si pertenecen al conjunto  $\mathcal{C}_{free}$  (muestras libres de colisiones) y si es verdadera esta condición se adicionan al grafo PRM y se busca a través de la función  $PRM\_SOLVE$  si hay una ruta desde  $c_{ini}$  y  $c_{goal}$ . Si existe dicha ruta, el algoritmo retorna 1, en caso contrario el algoritmo calcula nuevas muestras aleatorias en la región  $R_s$  hasta  $N_M$  muestras libres de colisiones y se adicionan al grafo PRM (intenta conectar con las muestras vecinas si es posible) el bucle continua hasta el máximo número de muestras a generar ( $MaxSamples$ ), si no ha se ha encontrado solución al problema el algoritmo retorna 0.

#### 4.4. Algoritmo 6: PRM Básico PCA

---

Entrada:

$c_{ini}$  : Configuración inicial

$c_{goal}$  : Configuración final

$d$  : Dimensión del espacio de configuraciones

---

Salida:

$P$  ruta a conectar  $c_{ini}$  y  $c_{goal}$

---

**for**  $i=1$  **to**  $d$  **do**

$S_{RAND} = \text{SAMPLE\_FROM}(R_s, 1)$

**end for**

$\text{PRM\_ADD}(S_{RAND})$

$S = S_{RAND}$

$\text{samples} = 0$

$N_v = 0$

$N_M = 0$

**while**  $\text{samples} < \text{MaxSamples}$  **do**

$\text{Samples} = \text{samples} + N_v + N_M$

$V_s = \text{PCA}(S)$

**for**  $i=1$  **to**  $N_v$  **do**

$s = \text{SAMPLE\_FROM}(V_s, 1)$

**if**  $s \in R_s$  **and**  $s \in C_{free}$  **then**

$S = S \cup s$

$\text{PRM\_ADD}(s)$

$P = \text{PRM\_SOLVE}(s)$

**if**  $P \neq \emptyset$  **then**

**return**  $P$

**end if**

**end if**

**end for**

**end for**

**for**  $i=1$  **to**  $N_M$  **do**

$s = \text{SAMPLE\_FROM}(R_s, 1)$

**if**  $s \in C_{free}$  **then**

$S = S \cup s$

$\text{PRM\_ADD}(s)$

**end if**

**end for**

**end while**

**return**  $\emptyset$

---

#### 4.5. PRM Gaussiano vs PRM PCA

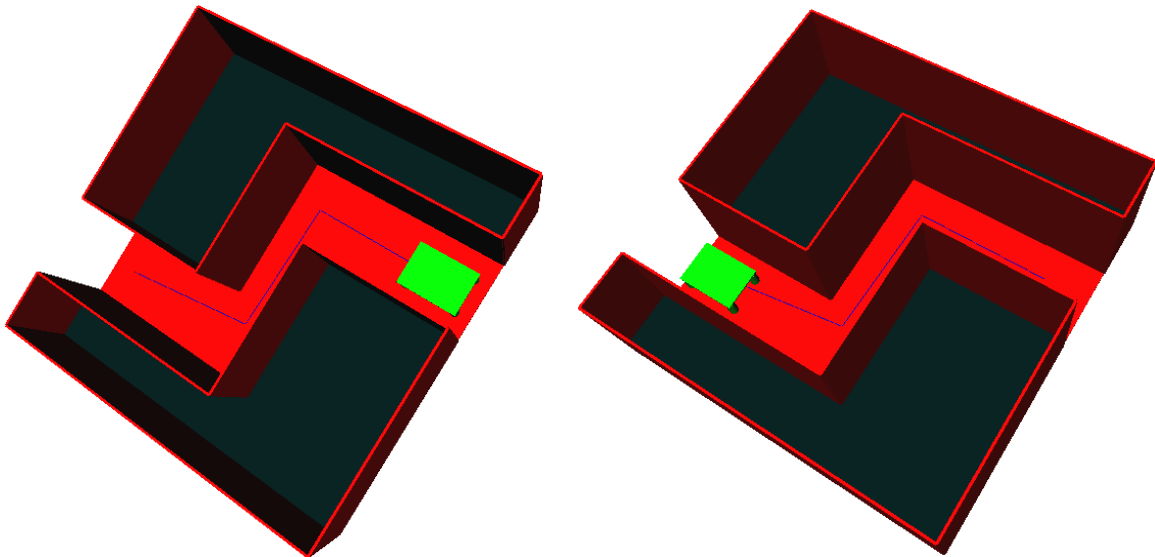
Ahora que ya se han explicado los algoritmos, se procede a compararlos en tres aspectos como son:

- El número de Colisiones
- El número de muestras generadas para alcanzar la solución
- El tiempo necesario para encontrar la solución.

Para comprobar el desempeño y el rendimiento de ambos métodos probabilísticos se han desarrollado cuatro ejemplos didácticos, los cuales se muestran a continuación.

##### Problema D:

Consiste en atravesar un pasaje estrecho (Figura 4.2), dado una posición inicial y una posición final, el objetivo es alcanzar la solución en el menor tiempo posible y con el menor número de muestras y colisiones.



(a) Posición Inicial

(b) Posición Final

Figura 4.2: Problema D.

Los resultados obtenidos de ambos métodos se muestran en la Tabla V y su gráfico correspondiente en la Figura 4.3, en la cual se puede apreciar que el método PRM basado en PCA tiene un rendimiento superior comparado con el método PRM basado en muestreo Gaussiano.

PLANIFICADOR	Tiempo Solución(s)	Número Colisiones	Número de Muestras
PRM GAUSSIAN	1,5906	1363,2	39,8
PRM PCA	1,1056	230,6	30
<b>Reducción</b>	<b>30,49 %</b>	<b>83,08 %</b>	<b>24,62 %</b>

Tabla V: Resultados para el Problema D.

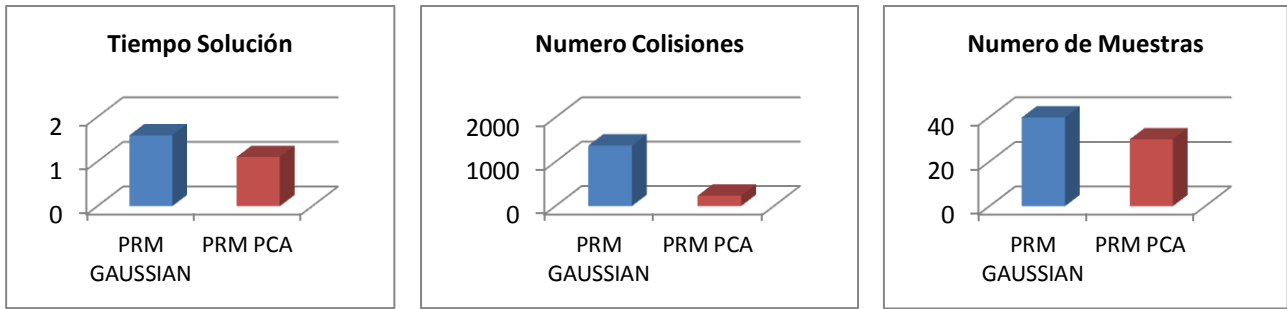
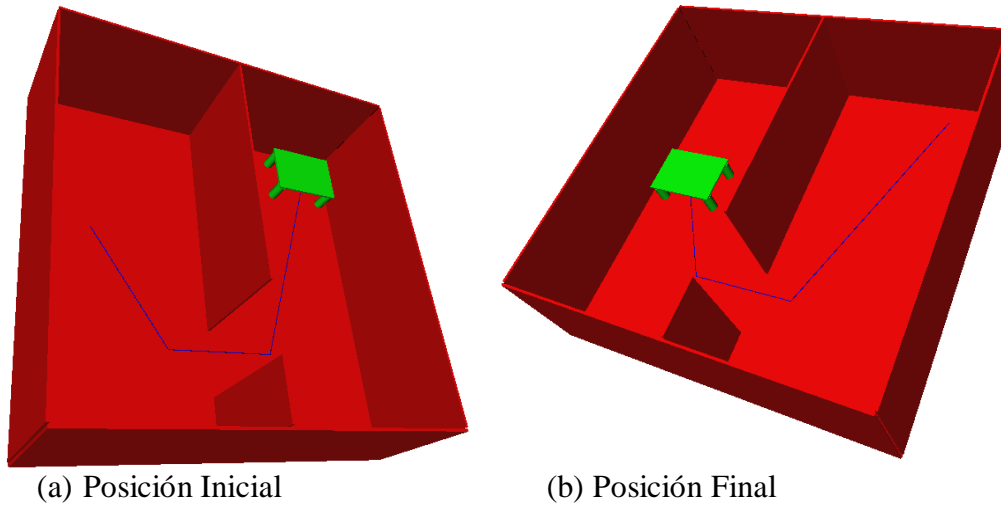


Figura 4.3. Gráfico de Resultados para el Problema D.

Problema E:

Consiste en alcanzar la posición final donde el espacio de configuraciones es disperso (Figura 4.4), el objetivo es alcanzar la solución en el menor tiempo posible y con el menor número de muestras, y colisiones.



(a) Posición Inicial  
Figura 4.4: Problema E.

(b) Posición Final

Los resultados obtenidos de ambos métodos se muestran en la Tabla VI y su gráfico correspondiente en la Figura 4.5. Se puede apreciar, como en el problema E, que el método PRM basado en PCA tiene un rendimiento superior comparado con el método PRM basado en muestreo Gaussiano.

PLANIFICADOR	Tiempo Solución(s)	Número Colisiones	Número de Muestras
PRM GAUSSIAN	1,6826	2716	79,2
PRM PCA	1,373	100,8	50,2
<b>Reducción</b>	<b>18,40 %</b>	<b>96,28 %</b>	<b>36,61 %</b>

Tabla VI: Resultados para el Problema E.

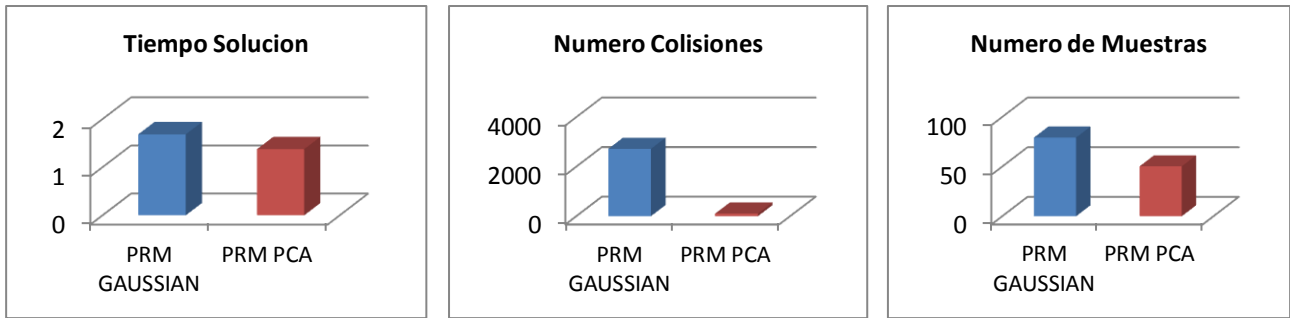
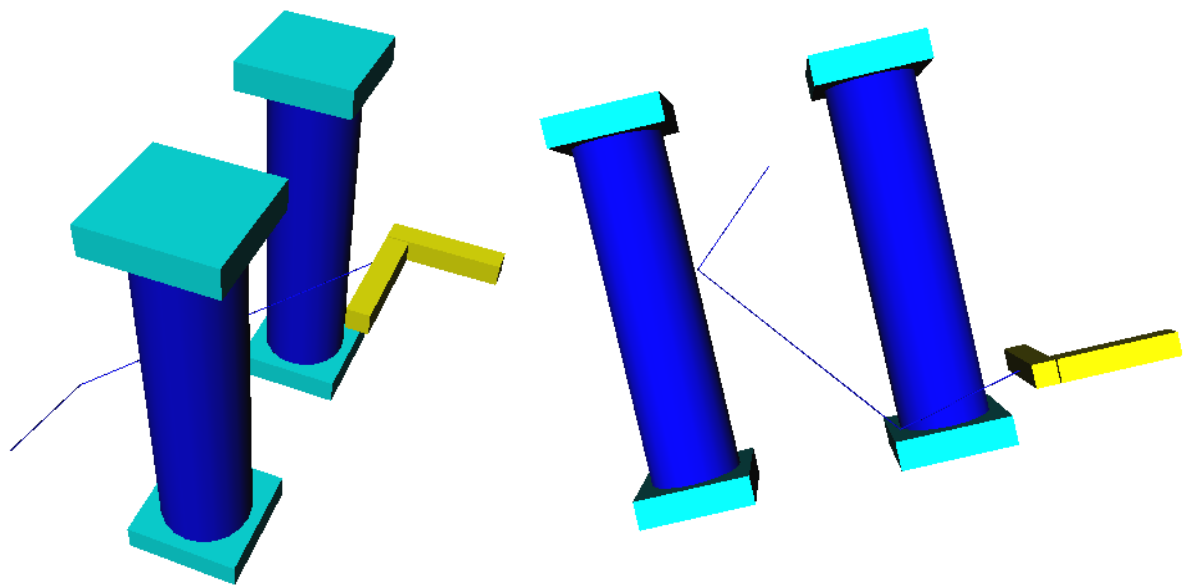


Figura 4.5. Gráfico de Resultados para el Problema E.

Problema F:

El siguiente problema ilustra la planificación de movimientos de un free-flying robot de 3GDL, que tiene que cruzar dos columnas (Figura 4.6).



(a) Posición Inicial

(b) Posición Final

Figura 4.6. Problema F.

Los resultados obtenidos de ambos métodos se muestran en la Tabla VII, en la cual se puede apreciar como en el problema E, que el método PRM basado en PCA tiene un rendimiento superior comparado con el método PRM basado en muestreo Gaussiano.

PLANIFICADOR	Tiempo Solución(s)	Número Colisiones	Número de Muestras
PRM GAUSSIAN	4,22832	5146,4	194,6
PRM PCA	0,4264	22	6,8
<b>Reducción</b>	<b>89,91 %</b>	<b>99,57 %</b>	<b>96,50 %</b>

Tabla VII: Resultados para el Problema F.

Problema G:

El siguiente problema ilustra la planificación de movimientos de un brazo robot Stäubli TX90 de 6 GDL (para este ejemplo solo se considera 3 GDL), que tiene una posición inicial y una posición final en la que debe evitar el contacto con las dos columnas (Figura 4.7).

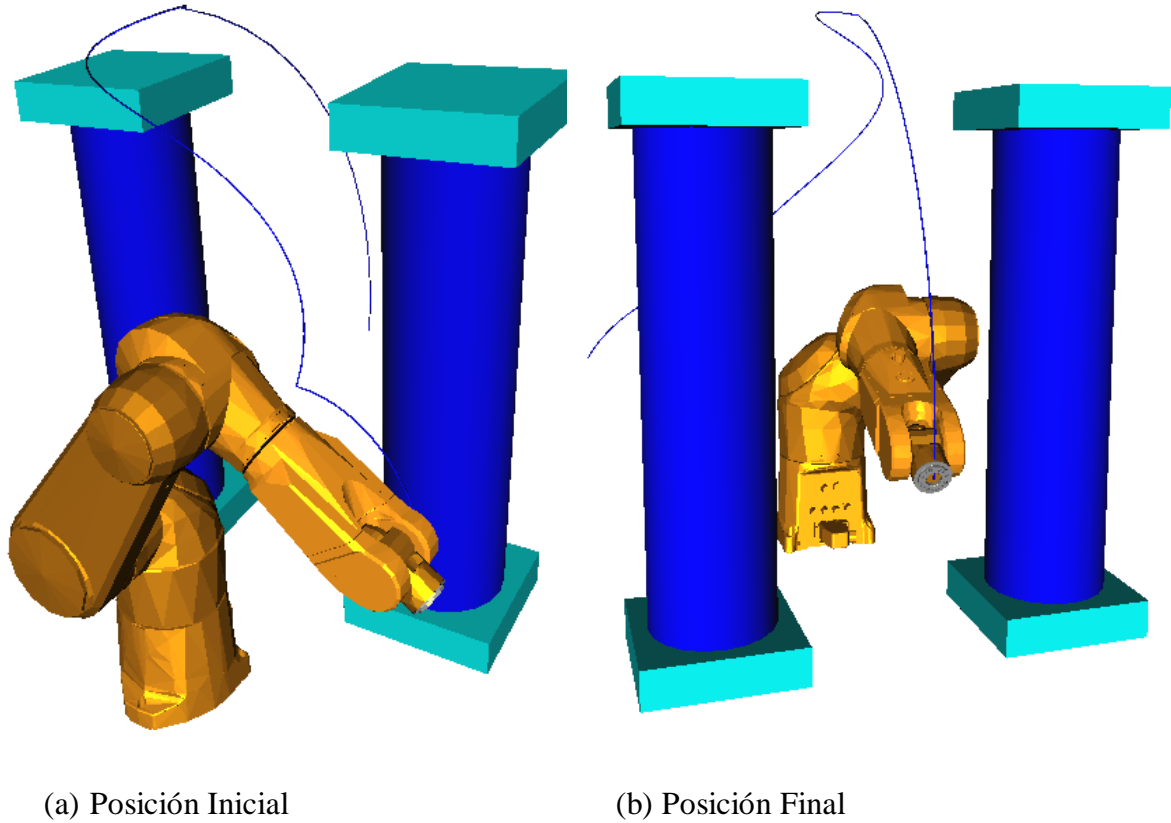


Figura 4.7. Problema G.

Los resultados obtenidos de ambos métodos se muestran en la Tabla VIII, en la cual se puede apreciar como en el problema F, que el método PRM basado en PCA tiene un rendimiento superior comparado con el método PRM basado en muestreo Gaussiano.

PLANIFICADOR	Tiempo Solución(s)	Número Colisiones	Número de Muestras
PRM GAUSSIAN	124,8254	10903,4	2625,2
PRM PCA	33,5282	200,8	101
Reducción	<b>73,13 %</b>	<b>98,15 %</b>	<b>96,15 %</b>

Tabla VIII: Resultados para el Problema F.

# Capítulo 5

## Aplicación a PRMs para Sistemas Robóticos Brazo-Mano

---

En este capítulo, se presenta un PRM usando el método de muestreo basado en PCA, para resolver problemas de planificación de movimientos de un sistema robótico brazo-mano (Figura 5.1). El planificador propuesto es un PRM concebido para contestar una única pregunta (single query), es decir no se persigue la captura de la conectividad de todo el espacio libre de  $\mathcal{C}$  del problema, sino únicamente la parte relevante para conectar dos configuraciones  $c_{init}$  y  $c_{goal}$ , siendo esta última una configuración de aprehensión.

En la presente tesis se enfoca el uso del PCA para la solución de problemas de planificación de movimientos de un sistema robótico brazo-mano como un solo sistema, en comparación con trabajos anteriores [15] y [16] en los que se usó PCA únicamente para los grados de libertad de la mano.

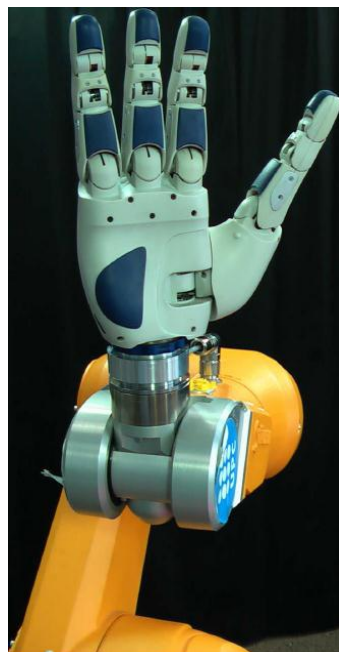


Figura 5.1. Sistema Robótico Brazo-Mano (IOC)

Para contrastar la eficiencia y rendimiento de la propuesta se han implementado de manera independiente dos planificadores:

- PRM Brazo-Mano basado en Muestro Gaussiano
- PRM Brazo-Mano basado en Muestro PCA

Dado que ambos métodos tienen características bien definidas, en como realizan el muestreo en el espacio de configuraciones para localizar configuraciones libres de colisiones en espacios reducidos o estrechos, parten de un punto en común que es el espacio de configuraciones del problema planteado, es así que antes de desarrollar los algoritmos para ambos planificadores locales, se explicara el espacio de configuraciones del sistema robótico brazo-mano.

### 5.1. Espacio de Configuraciones de un Sistema Robótico Brazo-Mano

A continuación se explica la concepción del problema del sistema robótico brazo-mano, para luego aplicar los dos métodos probalísticos (Gaussiano y PCA), para solucionar problemas de planificación de movimientos, en espacios reducidos o estrechos.

Sea  $\mathcal{C}$  el espacio de configuraciones del sistema robótico brazo-mano:

$$\mathcal{C} = \mathcal{C}^a \times \mathcal{C}^h$$

Donde  $\mathcal{C}^a$  y  $\mathcal{C}^h$  son respectivamente, el espacio de configuraciones del brazo y de la mano mecánica. Usando PCA para reducir los grados de libertad de la mano [15] y [16], la planificación de movimientos se hará en el sub-espacio  $\mathcal{SC}$  definido como:

$$\mathcal{SC} = \mathcal{C}^a \times \mathcal{SC}^h$$

Donde  $\mathcal{SC}^h$  es el sub-espacio  $H$  dimensional de  $\mathcal{C}^h$  definido por los primeros  $H$  componentes del PCA (llamados PMDs: Principal Motion Directions), entonces, si  $A$  es la dimensión de  $\mathcal{C}^a$ , la planificación se hará en un espacio  $d$  dimensional dónde:

$$d = A + H$$

Una configuración  $c \in \mathcal{SC}$  será un vector  $d$  dimensional cuyos primeros  $A$  componentes corresponden a las articulaciones del brazo y los últimos  $H$  corresponden a los PMDs que determinan las articulaciones de la mano mecánica.



Una vez definido el espacio de configuraciones del brazo y la mano mecánica, queda definir las regiones de muestreo de donde se obtendrán las muestras.

## 5.2. Regiones de Muestreo Gaussiano

En [6], se demostró la eficiencia de muestreo para obtener muestras cercanas a los obstáculos, en comparación con el método de muestro aleatorio uniforme.

En un sistema robótico brazo-mano, el muestreo Gaussiano se realizará alrededor de la configuración final de aprehensión, donde la holgura del espacio libre es reducida.

Sea:

- $pos(c)$  una función que retorna las coordenadas de posición del brazo cuando el sistema brazo-mano está localizado en la configuración  $c$ .
- $dist(p_1, p_2)$ , una función que calcula la distancia euclidiana entre los dos puntos  $p_1, p_2 \in \mathbb{R}^3$ .
- $B$  la región de  $\mathcal{SC}$  definida como:

$$B(p, \delta) = \{c \in \mathcal{SC} | dist(pos(c), p) \leq \delta\}$$

Donde  $p \in \mathbb{R}^3$  y  $\delta$  es una distancia dada como umbral.

Entonces la región  $M_s$  donde se realizara el muestreo Gaussiano quedará definido como sigue:

*Region  $M_s$* : región de  $\mathcal{SC}$  que satisface que la posición del brazo (es decir las coordenadas  $x, y$  y  $z$  del TCP del robot), está a una distancia  $\delta_R$  de la configuración final de aprehensión  $c_{goal}$  es decir:

$$M_s = B(pos(c_{goal}), \delta_R)$$

Para obtener cada par de muestras de  $M_s$  se ha hecho de la siguiente manera:

- Configuración del Brazo: la posición del brazo se obtiene muestreando un cubo de lado  $\delta_M$  centrado en  $pos(c_{goal})$ ; la orientación del brazo se obtiene muestreando alrededor de la orientación objetivo. Después se aplica la cinemática inversa para obtener la configuración del brazo  $(\theta_1, \dots, \theta_A) \in \mathcal{C}^a$ . En el algoritmo 6, las funciones que lo implementan son  $RAND\_ARM()$  y  $RAND\_HAND()$  respectivamente.
- La configuración de la mano: se obtiene muestreando cada PMD dentro del rango  $(\theta_{A+1}, \dots, \theta_{A+H}) \in \mathcal{SC}^h$ .

- La segunda configuración difiere de la primera en la posición del robot (se localiza aleatoriamente dentro de un radio  $d$  el cual es obtenido acorde a una distribución normal) y en la configuración de la mano.
- El valor de  $\delta_M$  no será fijo, sino que irá incrementándose en cada iteración del algoritmo, desde un valor inicial  $\delta_I$  hasta  $dist(pos(c_{ini}), pos(c_{goal}))$ , con incrementos  $\Delta_M$ .

### 5.3. PRM Brazo-Mano Gaussiano

El Algoritmo 7, describe un PRM que usa el método de muestreo Gaussiano, para resolver problemas de un sistema robótico real brazo-mano.

El algoritmo inicia con un bucle en el cual se obtiene un par de configuraciones alrededor de la configuración final muestreando dentro de  $M_S$ . Estas dos configuraciones de brazo-mano se comprueban, y si al menos una configuración está libre de colisiones, se almacena la libre y se busca si hay solución, en caso contrario ambas muestras son descartadas. Si no se encuentra solución se incrementa el umbral de distancia para generar dos nuevas configuraciones. El bucle continua hasta alcanzar la solución o hasta que el número total de muestras se haya generado.

Cuando se ha encontrado una configuración libre de colisiones en la región  $M_S$  se adiciona al conjunto  $S \in \mathcal{C}_{free}$ , y el roadmap intenta conectar con las muestras vecinas si es posible, finalmente se busca una solución.

El valor inicial de  $\delta_M$  es igual a  $\delta_I$ , y es incrementado por  $\Delta_M$  en cada iteración, y restableciendo a  $\delta_I$  siempre que  $\delta_M$  sea más grande que  $dist(pos(c_{ini}), pos(c_{goal}))$ .

El planificador solo tiene definido dos parámetros para ser usados ( $\delta_I$ , y  $\Delta_M$ ) aparte de los parámetros básicos de un PRM básico (es decir el número máximo de vecinos por nodo,  $N_n$ , y la distancia de umbral de los vecinos,  $D_n$ ).

#### 5.4. Algoritmo 7: PRM Brazo-Mano Gaussiano

Entrada:

$c_{ini}$  : Configuración inicial

$c_{goal}$  : Configuración final

$c_1$  : una configuración aleatoria brazo-mano ( $c_1^a, sc_1^h$ )

$d$  : distancia elegida acorde a una distribución normal.

$c_2$  : una configuración brazo-mano ( $c_2^a, sc_2^h$ ), donde  $c_2^a$  está a una distancia  $d$  de  $c_1^a$

$\delta_M, \delta_I, \Delta_M$  : Umbrales de distancia

Salida:

$P$  ruta a conectar  $c_{ini}$  y  $c_{goal}$

$numsamples = 0; d_{ig} = dist(pos(c_{ini}), pos(c_{goal})); \delta_M = \delta_I$

```
while  $numsamples < MaxSamples$  do
  ( $c_1^{aT}, c_1^{aO}$ ) =  $RAND\_ARM()$ 
   $c_1^a = (c_1^{aT}, c_1^{aO})$ 
   $d = RAND\_G(0,1)$ 
   $c_2^{aT} = c_1^{aT} + d$ 
   $c_2^{aO} = c_1^{aO}$ 
   $c_2^a = (c_2^{aT}, c_2^{aO})$ 
   $sc_1^h = RAND\_HAND()$ 
   $sc_2^h = RAND\_HAND()$ 
   $c_1 = (c_1^a, sc_1^h)$ 
   $c_2 = (c_2^a, sc_2^h)$ 
   $numsamples = numsamples + 2$ 
  if  $c_1 \in \mathcal{C}_{free}$  and  $c_2 \notin \mathcal{C}_{free}$  then
    if  $c_1 \in R_s$  then
       $S = S \cup c_1$ 
       $PRM\_ADD(c_1)$ 
       $P = PRM\_SOLVE(c_1)$ 
      if  $P \neq \emptyset$  then
        return  $P$ 
      end if
    end if
  else if  $c_2 \in \mathcal{C}_{free}$  and  $c_1 \notin \mathcal{C}_{free}$  then
    if  $c_2 \in R_s$  then
       $S = S \cup c_2$ 
       $PRM\_ADD(c_2)$ 
       $P = PRM\_SOLVE(c_2)$ 
      if  $P \neq \emptyset$  then
        return  $P$ 
      end if
    end if
  else discard both
  end if
   $\delta_M = \delta_M + \Delta_M$ 
  if  $\delta_M > d_{ig}$  then
     $\delta_M = \delta_I$ 
  end if
end while
return  $\emptyset$ 
```

## 5.5. Regiones de Muestreo PCA

En el capítulo 3, se realizó un estudio detallado del método de muestreo adaptativo usando PCA, y en capítulo 4 se desarrolló un PRM básico usando PCA, dando buenos resultados en comparación con el método de muestreo Gaussiano, ahora se explicara cómo aplicar este PRM basado en PCA, pero en un entorno mucho más complejo como son los sistemas robóticos brazo-mano.

Se definen las siguientes regiones:

- a) *Region*  $R_S$ : región de  $\mathcal{SC}$  que satisface que la posición del brazo (es decir las coordenadas  $x, y$  y  $z$  del TCP del robot), está a una distancia  $\delta_R$  cuando está localizado en la configuración final de aprehensión  $c_{goal}$  es decir:

$$R_S = B(pos(c_{goal}), \delta_R)$$

El análisis de componentes principales usa las configuraciones libres de colisiones de  $R_S$  para recalcular el muestreo de la región  $V_S$ .

- b) *Region*  $I_S$ : Es la región inicial donde se obtiene un conjunto de al menos  $d$  (dimensión del problema) muestras libres de colisiones para calcular la primera aproximación  $V_S$ . Estas muestras son obtenidas alrededor de  $c_{goal}$  :

$$I_S = B(pos(c_{goal}), \delta_I)$$

Donde  $\delta_I$  es un valor de umbral menor o igual que  $\delta_R$ .

Para obtener las muestras de la región  $I_S$  se ha realizado de la siguiente manera:

- Configuración del Brazo: la posición del brazo se obtiene muestreando un cubo de lado  $\delta_I$  centrado en  $pos(c_{goal})$ ; la orientación del brazo se obtiene muestreando alrededor de la orientación objetivo. Después la cinemática inversa se aplica para obtener la configuración del brazo  $(\theta_1, \dots, \theta_A) \in \mathcal{C}^a$ .
- La configuración de la mano: se obtiene muestreando cada PMD dentro del rango  $(\theta_{A+1}, \dots, \theta_{A+H}) \in \mathcal{SC}^h$ .

- c) *Region*  $V_S$ : es la región resultante del análisis de componentes principales, aplicado a todas las configuraciones libre de colisiones de  $R_S$ . Esto es una hipercaja de dimensión  $d$ , y las muestras dentro de  $V_S$  son una configuraciones brazo-mano  $(\theta_1, \dots, \theta_A, \dots, \theta_{A+H}) \in \mathcal{SC}$ .

d) *Region General de Muestreo*  $M_S$ : las muestras que están más lejos de  $c_{goal}$  son necesarias para construir el mapa de carreteras para resolver la consulta para conectar  $c_{init}$  y  $c_{goal}$ . También existen muestras fuera de  $V_S$  y son necesarias para recalculer  $V_S$  para cubrir mejor el área de interés. Para estos propósitos, se define una región  $M_S$  similar a  $R_S$  e  $I_S$ :

$$M_S(\delta_M) = B(pos(c_{goal}), \delta_M)$$

Donde  $\delta_I \leq \delta_M \leq dist(pos(c_{ini}), pos(c_{goal}))$ .

Las muestras de  $M_S$  se obtienen de la siguiente manera:

- Configuración del brazo: como en  $I_S$  pero usando  $\delta_M$  en lugar de  $\delta_I$ .
- Configuración de la mano: se obtiene igual que en  $I_S$ .

## 5.6. PRM Brazo-Mano PCA

El Algoritmo 8 describe un PRM que usa el método de muestreo basado en PCA, para resolver problemas de un sistema robótico real brazo-mano.

A continuación se describe el funcionamiento del algoritmo:

El algoritmo primero obtiene configuraciones libres de colisiones alrededor de la configuración final muestreando dentro de  $I_S$ , entonces el bucle se inicia siguiendo en tres pasos, hasta que la solución es encontrada o hasta que se ha generado el número total de muestras :

- a) Recalcula  $V_S$  usando las muestras del roadmap dentro de la región  $R_S$ .
- b) Obtiene muestras libres de colisiones de  $V_S$ , las adiciona al conjunto de muestras libres, y al roadmap, y luego busca una solución.
- c) Obtiene muestras libres de colisiones de  $M_S$ , y las adiciona al conjunto de muestras libres, y al roadmap.

El tamaño de  $M_S$  está definido con una variable de valor  $\delta_M$ : el valor inicial de  $\delta_M$  es igual a  $\delta_I$ , y se incrementa  $\Delta_M$  en cada iteración, restableciéndolo a  $\delta_I$  siempre que  $\delta_M$  sea más grande que  $dist(pos(c_{ini}), pos(c_{goal}))$ .

El planificador solo tiene definido tres parámetros para ser usados ( $\delta_I$ ,  $\delta_R$ , y  $\Delta_M$ ) aparte de los parámetros básicos de un PRM básico (es decir el número máximo de vecinos por nodo,  $N_n$ , y la distancia de umbral del vecino,  $D_n$ ).

### 5.7. Algoritmo 8: PRM Brazo-Mano PCA

---

Entrada:

$c_{ini}$  : Configuración inicial  
 $c_{goal}$  : Configuración final  
 $d$  : Dimensión del espacio de configuraciones  
 $\delta_I$  : Umbral de distancia inicial  $I_s$   
 $\delta_R$  : Umbral de distancia de la región  $R_s$   
 $\Delta_M$  : Incremento de distancia

---

Salida:

$P$  ruta a conectar  $c_{ini}$  y  $c_{goal}$

---

```
 $S_{PCA} = \text{SAMPLE\_FROM}(I_s, d)$   
 $\text{PRM\_ADD}(S_{PCA})$   
 $\delta_M = \delta_I$   
 $\text{numsamples} = 0$   
while  $\text{numsamples} < \text{MaxSamples}$  do  
     $\text{numsamples} = \text{numsamples} + N_v + N_M$   
     $V_s = \text{PCA}(S_{PCA})$   
    for  $i=1$  to  $N_v$  do  
         $s = \text{SAMPLE-FROM}(V_s, 1)$   
        if  $s \in R_s$  and  $s \in C_{free}$  then  
             $S_{PCA} = S_{PCA} \cup s$   
             $\text{PRM\_ADD}(s)$   
             $P = \text{PRM-SOLVE}(s)$   
            if  $P \neq \emptyset$  then  
                return  $P$   
            end if  
        end if  
    end for  
    for  $i=1$  to  $N_M$  do  
         $s = \text{SAMPLE-FROM}(M_s(\delta_M), 1)$   
        if  $s \in R_s$  and  $s \in C_{free}$  then  
             $S_{PCA} = S_{PCA} \cup s$   
             $\text{PRM\_ADD}(s)$   
        end if  
    end for  
     $\delta_M = \delta_M + \Delta_M$   
    if  $\delta_M > \text{dist}(\text{pos}(c_{ini}), \text{pos}(c_{goal}))$  then  
         $\delta_M = \delta_I$   
    end if  
end while  
return  $\emptyset$ 
```

---

## 5.8. Implementación PRM Brazo-Mano Gaussiano y PCA

La implementación se ha desarrollado dentro del entorno de programación para aplicaciones de planificación The Kautham Project, desarrollada en el Instituto de Organización y Control de Sistemas Industriales (Figuras 5.2, 5.3 y 5.4).

Las librerías utilizadas son Qt [18], para la interfaz gráfica, Coin3D [19] para el entorno grafico de las simulaciones y PQP [20] para el detector de colisiones, así mismo esta aplicación proporciona al desarrollador; modelos de robots incluidos la cinemática directa e inversa.

Para el cálculo del PCA, se ha utilizado la librería: Armadillo C++ Linear Algebra [28], pero ahora incorporado dentro de The Kautham Project, para ejecutar el PCA en tiempo real cuando se está resolviendo un problema de planificación.

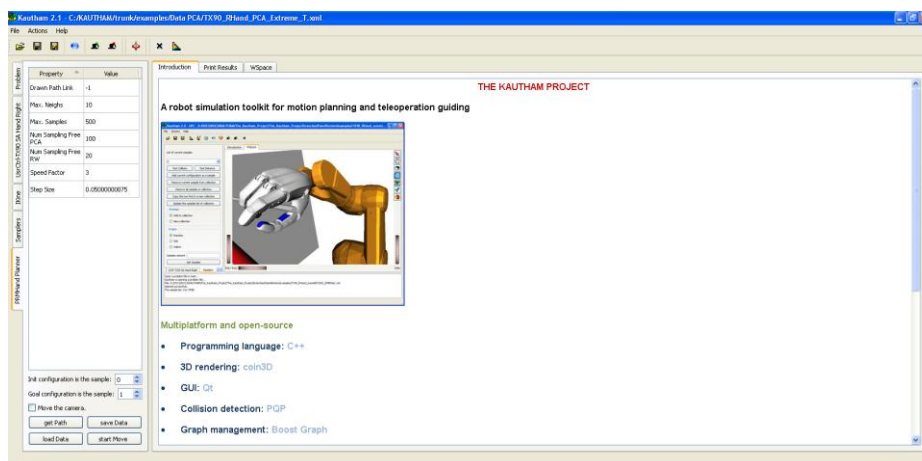


Figura 5.2: Interfaz Gráfica The Kautham Project (IOC)

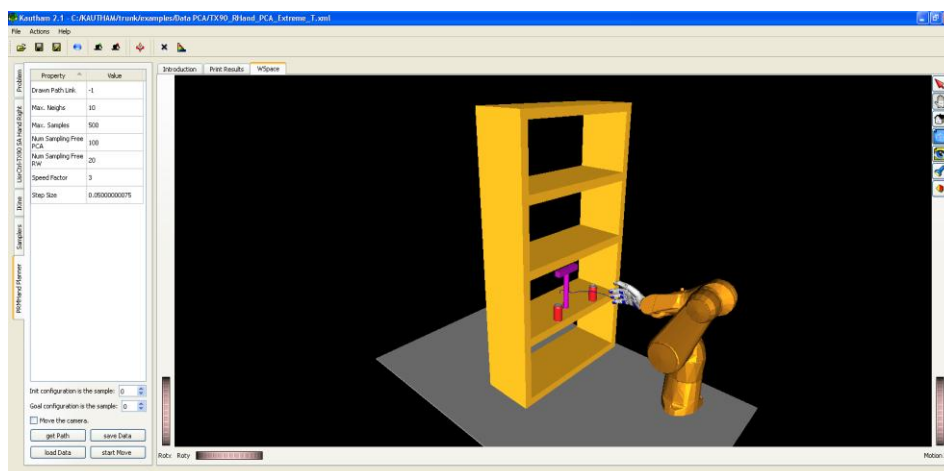


Figura 5.3: Espacio de Trabajo para las Simulaciones Brazo-Mano.

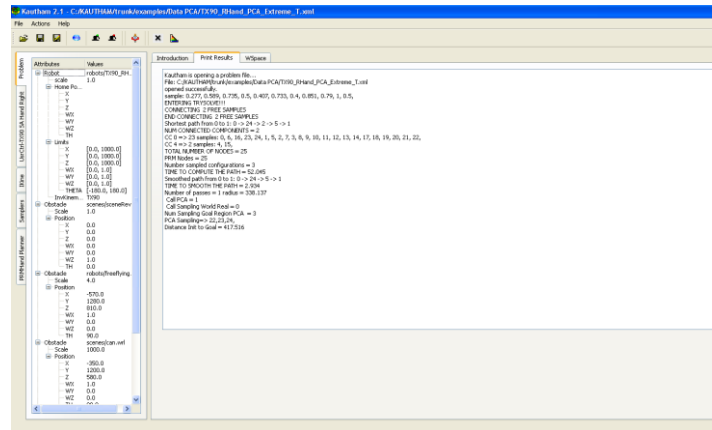


Figura 5.4: Impresión de los Resultados.

### 5.9. PRM Brazo-Mano Gaussiano vs PRM Brazo-Mano PCA

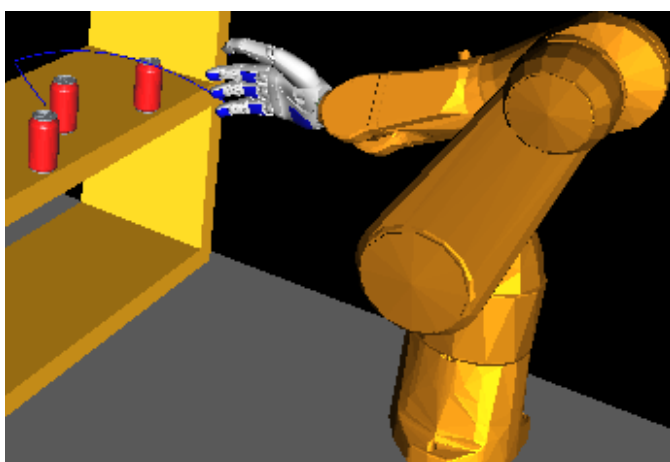
Ahora que ya se han explicado los algoritmos de sistemas robóticos brazo-mano, se procede a compararlos en tres aspectos como son:

- El número de Colisiones
- El número de muestras generadas la solución
- El tiempo necesario para encontrar la solución.

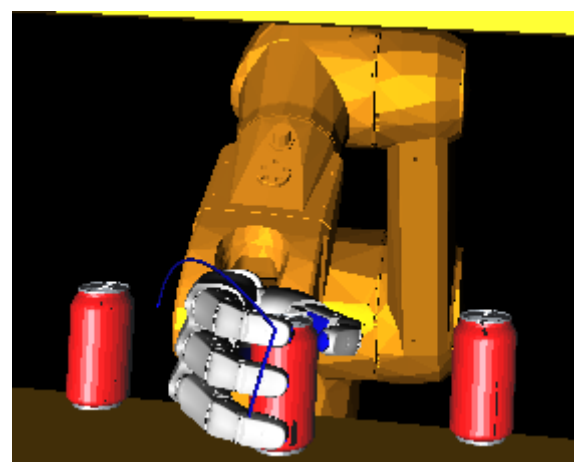
Los parámetros para el PRM Brazo-Mano Gaussiano son:  $\delta_I = 5$  y  $\Delta_M = 2,5$  y para el PRM Brazo-Mano PCA son:  $\delta_I = 5, \delta_R = 30$ , y  $\Delta_M = 2,5$ , todas las unidades están en centímetros. A continuación se presentan cuatro ejemplos didácticos.

#### Problema H:

Consiste en un problema de planificación de movimientos de un sistema robótico brazo-mano, dado una posición inicial y una posición final, para este ejemplo consiste en sujetar la lata central sobre la estantería (Figura 5.5)



(a) Posición Inicial



(b) Posición Final

Figura 5.5: Problema H



Los resultados obtenidos de ambos métodos se muestran en la Tabla IX y su grafico correspondiente en la Figura 5.6.

PLANIFICADOR	Tiempo Solución(s)	Número Colisiones	Número de Muestras
PRM B-M GAUSSIAN	11,811	102,4	11
PRM B-M PCA	13,2336	160	12
<b>Aumento</b>	<b>10,74 %</b>	<b>36 %</b>	<b>8,33 %</b>

Tabla IX: Resultados para el Problema H.

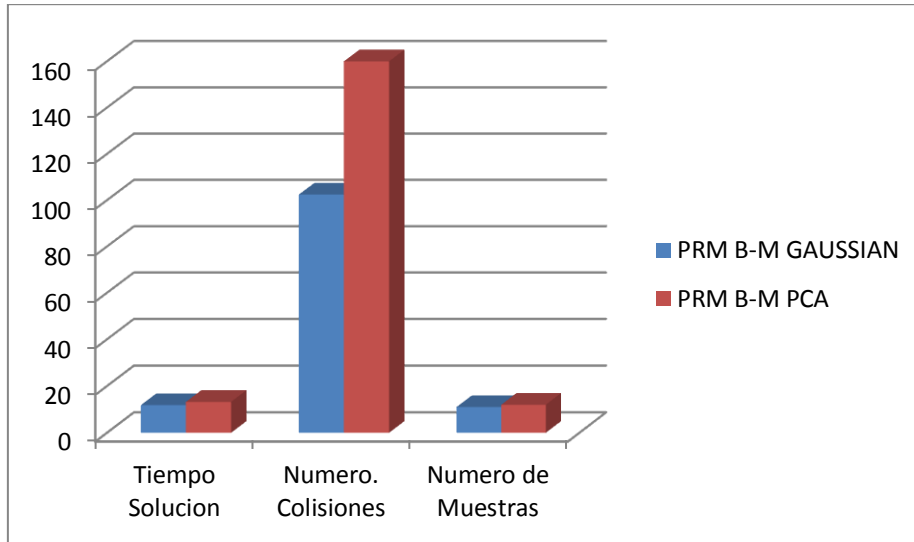
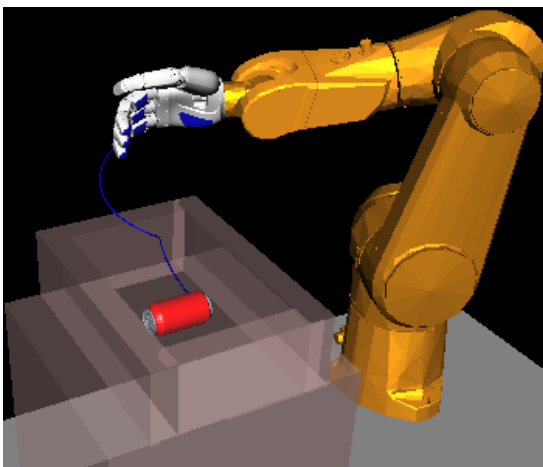


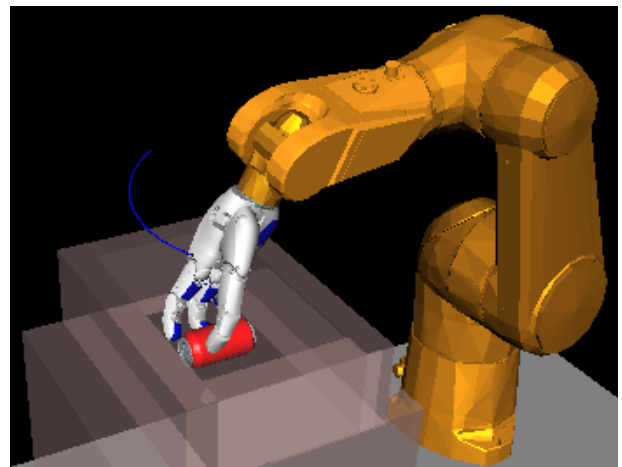
Figura 5.6. Gráfico de Resultados para el Problema H.

### Problema I:

Consiste en un problema de planificación de movimientos de un sistema robótico brazo-mano, dado una posición inicial y una posición final, para este ejemplo consiste en sujetar la lata que se encuentra dentro de una caja (Figura 5.7).



(a) Posición Inicial



(b) Posición Final

Figura 5.7: Problema I.

Los resultados obtenidos de ambos métodos se muestran en la Tabla X y su grafico correspondiente en la Figura 5.8.

PLANIFICADOR	Tiempo Solución(s)	Número Colisiones	Número de Muestras
PRM B-M GAUSSIAN	4,993	70	5,8
PRM B-M PCA	11,36828	244,6	13
<b>Aumento</b>	<b>56,07 %</b>	<b>71,38 %</b>	<b>55,38 %</b>

Tabla X: Resultados para el Problema I.

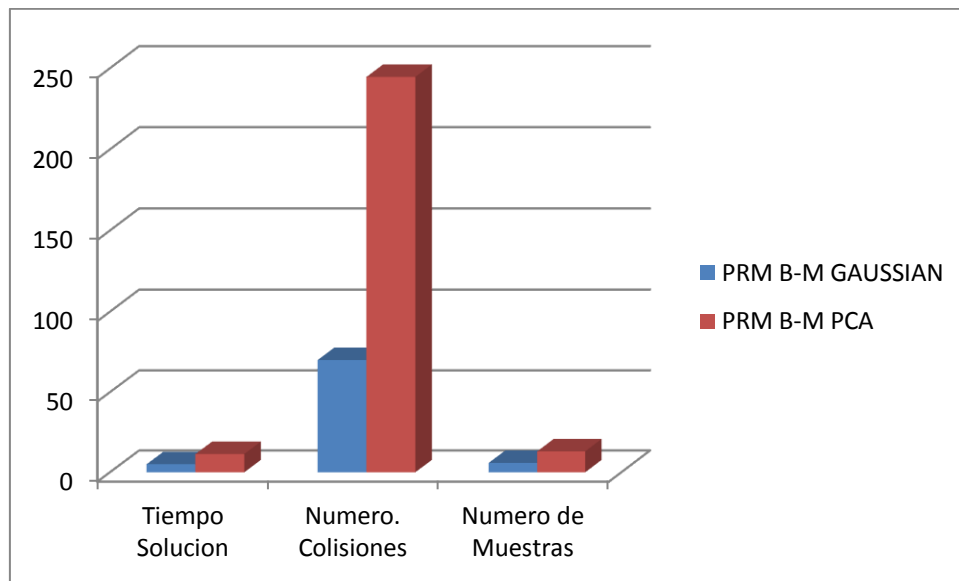
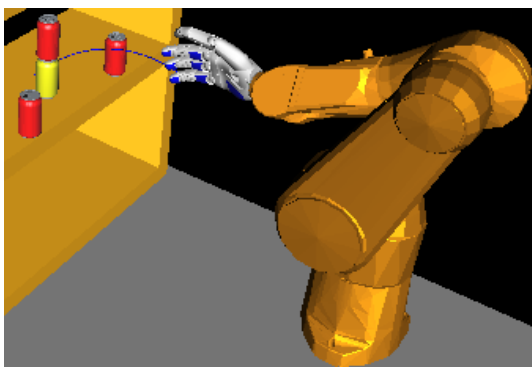


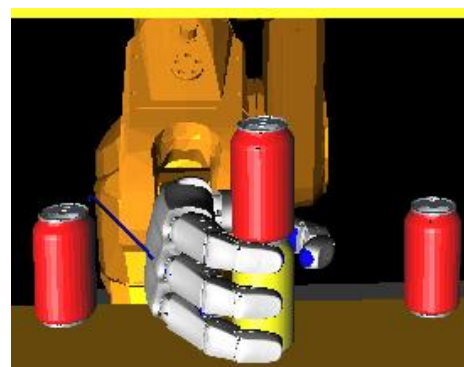
Figura 5.8. Gráfico de Resultados para el Problema I.

### Problema J:

Consiste en un problema de planificación de movimientos de un sistema robótico brazo-mano, dado una posición inicial y una posición final (configuración de aprehensión), para este ejemplo consiste en sujetar la lata de color amarillo (Figura 5.9).



(a) Posición Inicial



(b) Posición Final

Figura 5.9: Problema J.

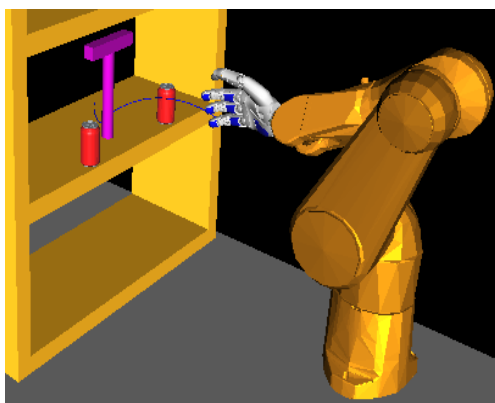
Los resultados obtenidos solo se muestran los correspondientes al PRM Brazo-Mano PCA (Tabla XI), porque encuentra solución al problema planteado, sin embargo el PRM Brazo-Mano Gaussiano no logra encontrar una solución (habiendo fijado el número máximo de 500 muestras libre de colisiones).

PLANIFICADOR	Tiempo Solución(s)	Número Colisiones	Número de Muestras
PRM B-M PCA	54,88	852,6	27,4

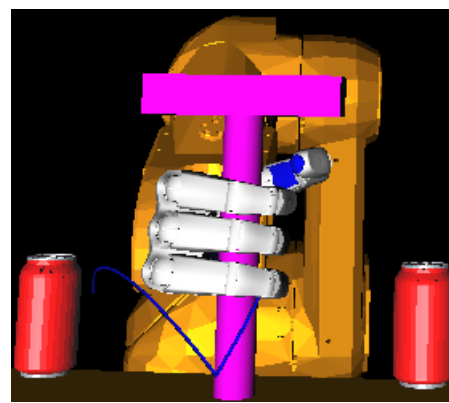
Tabla XI: Resultados para el Problema J.

Problema K:

Consiste en un problema de planificación de movimientos de un sistema robótico brazo-mano, dado una posición inicial y una posición final (configuración de aprehensión), para este ejemplo consiste en sujetar una herramienta tipo T (Figura 5.10)



(a) Posición Inicial



(b) Posición Final

Figura 5.10: Problema K.

Como en el caso anterior el PRM Brazo-Mano PCA, es el único planificador que logra encontrar solución al problema planteado, el PRM Brazo-Mano Gaussiano no logra encontrar una solución (habiendo fijado el número máximo de 500 muestras libres de colisiones).

Los resultados son mostrados en la Tabla XII.

PLANIFICADOR	Tiempo Solución(s)	Número Colisiones	Número de Muestras
PRM B-M PCA	36,11	618,50	15,10

Tabla XII: Resultados para el Problema K.

Como se ha podido apreciar en los cuatro ejemplos planteados de un sistema robótico brazo-mano, el PRM Brazo-Mano Gaussiano, tiene una ventaja razonable en comparación con el PRM Brazo-Mano PCA en los dos primeros ejemplos (ver resultados en la Tabla IX y X), esto se debe generalmente por que el muestreo Gaussiano; desde la primera muestra ya se está buscando una solución, en cambio en el otro lado para poder aplicar el primer PCA es necesario contar con al menos 11 (dimensión del problema: los seis primeros corresponden a las articulaciones del brazo y los cinco últimos a los 5 PMDs de la mano) muestras libres del espacio de configuraciones, por lo cual para alcanzar estas muestras iniciales se producen muchas llamadas al método de detector de colisiones.

Para problemas simples el PRM Brazo-Mano Gaussiano es eficiente en tiempo de respuesta, menor número de llamadas al método de detección de colisiones y numero de muestras para alcanzar la solución; pero para problemas más complejos, como los problema H, I; que tienen una configuración final de aprensión, no puede alcanzar la solución, por lo cual dicho PRM, no es el adecuado para utilizar, sin embargo el PRM Brazo-Mano PCA, siempre ha encontrado solución (con los parámetros  $\delta_I = 5$ ,  $\delta_R = 30$ , y  $\Delta_M = 2.5$ ), independientemente de la complejidad del problema ya sea una configuración de aprensión o no, lo cual se ha demostrado en la presente tesis.

A continuación se muestran la secuencia de posturas Brazo-Mano, para solucionar los problemas H,I; que tiene una configuración final de aprensión, Figura 5.11 y 5.12.

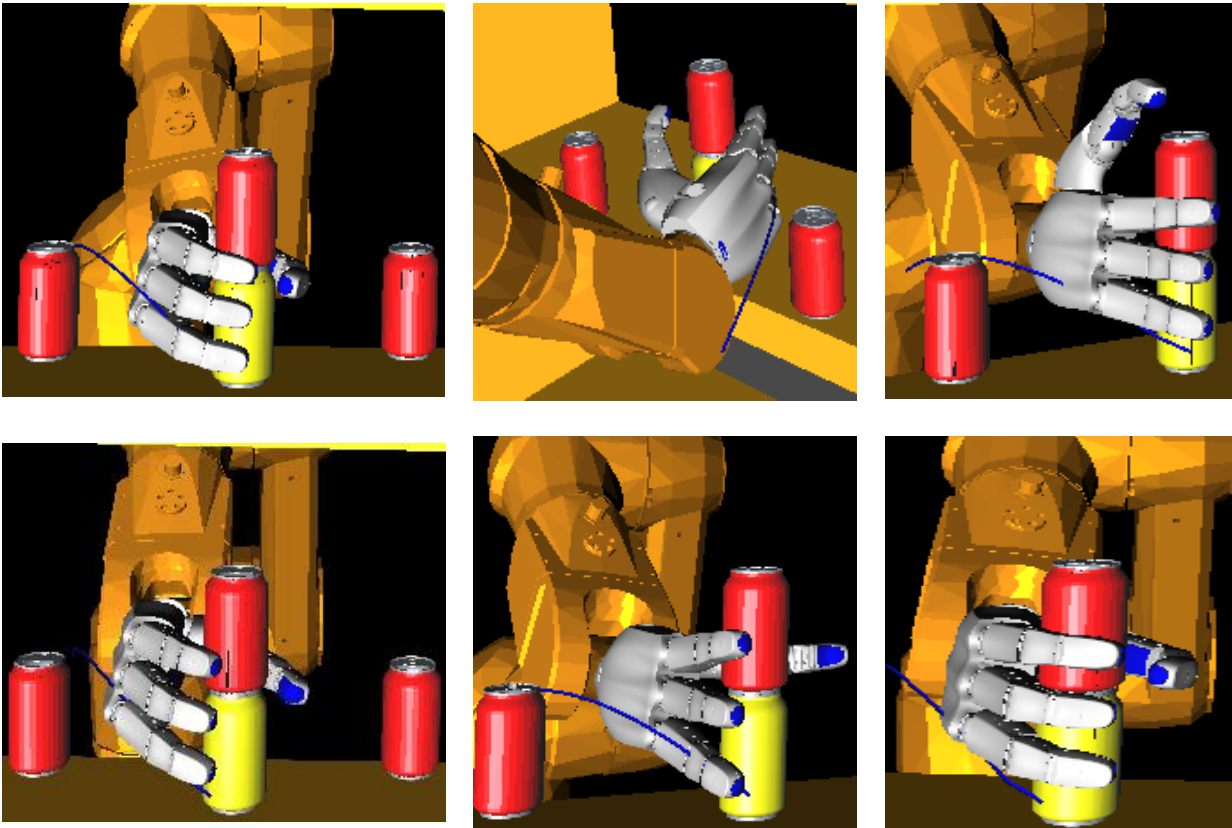


Figura 5.11: Posturas Brazo-Mano para encontrar la solución al problema H.

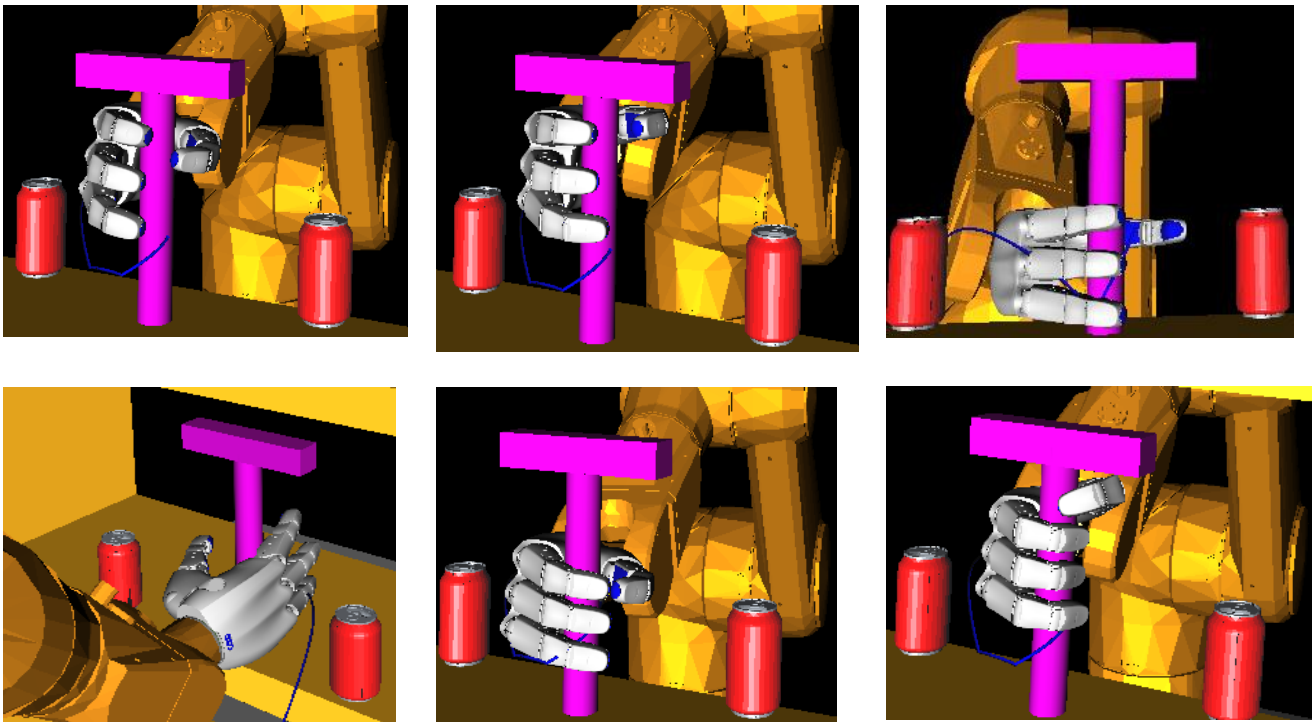


Figura 5.12: Posturas Brazo-Mano para encontrar la solución al problema I.

# Capítulo 6

## Conclusiones

---

Para mejorar la eficiencia de métodos basados en muestreo, en la presente Tesis se ha propuesto un nuevo método de muestreo, que con una alta probabilidad de encontrar muestras hacia regiones difíciles del espacio de configuraciones, como espacios reducidos o pasajes estrechos, basado en el uso de Análisis de Componentes Principales, aplicado a las configuraciones libre de colisiones. El procedimiento del muestreo propuesto ha sido concebido como un método local, es decir para ser aplicado a una región  $R_s$  del espacio de configuraciones donde se encuentra el área de interés. El uso de PCA tiene como resultado una nueva base del espacio de configuración con la que se define un volumen de muestreo. Las configuraciones muestreadas de ese volumen están más propensas para ser libre de colisiones, así mejorar la eficiencia de muestreo. Este volumen del muestreo es actualizado iterativamente para asegurarse de alcanzar la región de interés, con nuevas configuraciones libre de colisiones.

Se ha implementado y evaluado la propuesta de manera satisfactoria, y se han reportado los resultados de los ejemplos.

Asimismo una vez probada la viabilidad del muestreo propuesto, se ha implementado un PRM PCA, el cual tiene la característica principal de utilizar el muestreo basado en PCA, para obtener configuraciones libre de colisiones del espacio de configuraciones.

Para realizar la comparación era necesario un PRM que tenga características afines en la solución de problemas de planificación de movimientos sobre espacios reducidos o pasajes estrechos, por lo cual se optó por implementar un PRM Gaussiano, el cual tiene la particularidad que para obtener las muestras libres de colisiones utiliza un muestreo Gaussiano. Dado que ambos PRM son probabilísticos por su tipo de muestreo que utilizan internamente (PCA y Gaussiano, respectivamente), se realizaron ejemplos para compararlos y verificar su rendimiento. Dicha comparación se llevó a cabo teniendo en cuenta el tiempo de respuesta, el número de colisiones y las muestras generadas hasta alcanzar la solución.

Para resolver el problema de planificación de movimientos de un sistema robótico brazo-mano de 19 grados de libertad (Seis corresponden al brazo robot y trece a la mano mecánica antropomorfa) se ha implementado un PRM Brazo-Mano PCA aplicado a un sistema robótico brazo-mano, al cual para compararlo y verificar su rendimiento, también se ha implementado un PRM Brazo-Mano Gaussiano. El PRM Brazo-Mano PCA, resuelve problemas complejos como configuraciones finales de aprensión que el PRM Brazo-Mano Gaussiano es incapaz de resolver.

La aplicación del PCA en grandes dimensiones como: reconocimiento de imágenes, clasificación de patrones, aprensión, planificación de movimientos, están ahora en desarrollo e investigación por la comunidad científica. En la presente tesis en particular se ha aplicado el PCA para crear un nuevo método de muestreo y un PRM PCA para solucionar un problema de planificación de movimientos a un sistema robótico brazo-mano, en ambos casos dando buenos resultados.

# Publicaciones

---

A continuación se enumeran las publicaciones que se han realizado durante la realización de la presente Tesis:

- J. Rosell, **L. Cruz**, R. Suárez, and A. Pérez, “Importance sampling based on adaptive principal component analysis,” *Accepted to the 2011 IEEE Int. Symposium on Assembly and Manufacturing*.
- J. Rosell, **L. Cruz**, R. Suárez, and A. Pérez, “Path planning for grasping Operations using an Adaptive PCA-based Sampling Method, *submitted to the 2011 IEEE Int. Conf. on Robotics and Automation*.



# Bibliografía

---

- [1] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion*. The MIT Press, 2005.
- [2] L. E. Kavraki and J.-C. Latombe, “Randomized preprocessing of configuration for fast path planning,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 3, 1994, pp. 2138–2145.
- [3] D. Hsu, J.-C. Latombe, and H. Kurniawati, “On the probabilistic foundations of probabilistic roadmap planning,” *The International Journal of Robotics Research*, vol. 25, no. 7, pp. 627 – 643, 2006.
- [4] J. P. van der Berg and M. H. Overmars, “Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners ”. *The International Journal of Robotics Research*, vol. 24 (12), pp. 1055–1071, 2005.
- [5] H. Kurniawati and D. Hsu, “Workspace-based connectivity oracle: An adaptive sampling strategy for PRM planning,” in *Algorithmic Foundations of Robotics VII*, S. Akella and et.al., Eds. Springer–Verlag, 2006.
- [6] V. Boor, M. H. Overmars, and A. F. van der Stappen, “The Gaussian sampling strategy for probabilistic roadmap planners,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 1018–1023.
- [7] D. Hsu, T. Jiang, J. Reif, and Z. Sun, “The bridge test for sampling narrow passages with probabilistic roadmap planners,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 4420–4426.
- [8] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. K. Overmars, “Probabilistic roadmaps for path planning in high -dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, August 1996.
- [9] D. Hsu, G. Sanchez-Ante, and Z. Sun, “Hybrid PRM sampling with a cost-sensitive adaptive strategy,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 3874 – 3880.
- [10] M. Saha, J. C. Latombe, Y. C. Chang, and F. Prinz, “Finding narrow passages with probabilistic roadmaps: The small-step retraction method,” *Autonomous robots*, vol. 19(3), pp. 301–319, 2005.

- [11] H. L. Cheng, D. Hsu, J. C. Latombe, and G. Sanchez-Ante, “Multi-level free space dilation for sampling narrow passages in prm planning,” in Proc. of the IEEE Int. Conf. on Robotics and Automation, 2006, pp.1255– 1260.
- [12] D. Berenson, S. Srinivasa, D. Ferguson, and J. Kuffner, “Manipulation planning on constraint manifolds,” in Proc. of the IEEE Int. Conf. on Robotics and Automation, 2009, pp. 625 – 632.
- [13] A. Safonova, J. K. Hodgins, and N. S. Pollard, “Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces,” ACM Transactions Graph., vol. 23, no. 3, pp. 514–521, 2004.
- [14] M. T. Ciocarlie and P. K. Allen, “Hand posture subspaces for dexterous robotic grasping,” The International Journal of Robotics Research, vol. 28, no. 7, pp. 851–867, July2009.
- [15] R. Suárez, J. Rosell, A. Pérez, and C. Rosales, “Efficient search of obstacle-free paths for anthropomorphic hands,” in Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2009, pp. 1773– 1778.
- [16] J. Rosell, R. Suárez, C. Rosales, J. A. Garcia and A. Perez, “Motion planning for high DOF anthropomorphic hands,” in Proc. of the IEEE/RSJ Int. Conf. on Robotics and Automation, 2009, pp. 4025– 4030.
- [17] I. Jolliffe, Principal Component Analysis. Upper Saddle River, NJ, USA: Springer Series in Statistics, 2002.
- [18] J. Blanchette and M. Summerfield, C++ GUI Programming with Qt 4. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2006.
- [19] Kongsberg Oil & Gas Technologies, “Coin3D - 3d graphics development tools,” [www.coin3d.org](http://www.coin3d.org), December 2010.
- [20] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, “Fast proximity queries with swept sphere volumes,” in Proc. of Int. Conf. on Robotics and Automation, 2000, pp. 3719–3726.
- [21] A. Bicchi, “Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 652–662, 2000.
- [22] L. Biagiotti, F. Lotti, C. Melchiorri, and G. Vassura, “How far is the human hand? a review on anthropomorphic robotic end-effectors,” tech. rep., University of Bologna, 2004.

- [23] M. F. M. Santello and J. F. Soechting, “Postural hand synergies for tool use,” *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, December 1998.
- [24] M. Ciocarlie, C. Goldfeder, and P. Allen, “Dimensionality reduction for hand-independent dexterous robotic grasping,” in *Proc. of theIEEE/RSJ Conf. on Intelligent Robots and Systems*, 2007, pp. 3270–3275.
- [25] J. W. Eaton, GNU Octave Manual. Network Theory Limited, 2002.
- [26] T. R. Foundation, “The R project for statistical computing,” [www.rproject.org](http://www.rproject.org), Marzo 2011.
- [27] D. Eddelbuettel, “Rcpp: Seamless R and C++ integration,” [dirk.eddelbuettel.com/code/rcpp.html](http://dirk.eddelbuettel.com/code/rcpp.html), Marzo 2011.
- [28] C. Sanderson, “Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments,” NICTA, Tech. Rep., 2010.

# Apéndice A

---

## Código C++

El software desarrollado en la presente tesis se adjunta en el CD, el cual tiene la siguiente estructura:

### Capítulo 3:

#### Sampling :

- Sampling\_PCA (version\_1.0): implementa el algoritmo 2.
- Sampling\_PCA (version\_2.0): implementa el algoritmo 4.

### Capítulo 4:

#### Gaussian:

- gaussianplanner.cpp y gaussianplanner.h: planificador de movimientos que implementa el algoritmo 5.

#### PCA :

- prmplanner\_pca.cpp y prmplanner\_pca.h: planificador de movimientos que implementa el algoritmo 6.

#### Problemas:

- Archivos .xml, para probar los planificadores (Gaussiano y PCA).

### Capítulo 5:

#### Gaussian:

- gaussianplannerArmHand.cpp y gaussianplannerArmHand.h: planificador de movimientos que implementa el algoritmo 7.

#### PCA :

- prmplannerArmHandPCA.cpp y prmplannerArmHandPCA.h: planificador de movimientos que implementa el algoritmo 8.

#### Problemas:

- Archivos .xml, para probar los planificadores (Gaussiano y PCA).