

En español

Métodos para generar trayectorias libres de colisiones en entornos multidimensionales

Edwin Francis Cárdenas¹, Luis Miguel Mendez²,
Jorge Sofronye Esmeral³

RESUMEN

Planear los movimientos de un robot de forma fiable sin que colisione con su entorno se ha convertido en un problema relevante en la última década. La idea central de este artículo es exponer dos de los principales métodos que permiten generar trayectorias libre de colisiones en cualquier espacio n -dimensional. La importancia de conocer estas estrategias radica en que estos mismos algoritmos se han aplicado en áreas distintas a la robótica, como diseño de medicamentos, animación computarizada e inteligencia artificial. También presentamos un referente histórico de varios métodos de generación de trayectorias y mostramos mediante simulaciones como las estrategias de los campos potenciales artificiales y de los mapas probabilísticos funcionan, siendo los más apropiados en virtud de los aspectos teóricos expresados en otros textos especializados.

Palabras clave: robótica, generación de trayectorias, evasión de obstáculos, sistemas multidimensionales.

Recibido: Octubre 2 de 2009
Aceptado: noviembre 4 de 2010

Introducción

En las décadas de los ochenta y noventa la popularidad de los robots y sus múltiples usos en la industria hicieron que los investigadores y programadores se dedicaran a resolver el problema de concebir trayectorias libres de colisiones entre los robots y su entorno (Goldman, 1990). Una de las metas de la robótica es desarrollar sistemas autónomos, y lo que para el hombre parece fácil e intuitivo, desplazarse de un sitio a otro o manipular objetos, no es una labor tan sencilla para las máquinas.

Una primera aproximación a los métodos de planificación de caminos fue realizada por Lozano-Pérez (Lozano, 1981). Una década después se dio la aparición del libro de Jean-Claude Latombe (Latombe, 1991), y desde ese entonces un buen número de publicaciones y artículos han abordado el tema de genera-

In English

Collision-free path planning in multi-dimensional environments

Edwin Francis Cárdenas⁴, Luis Miguel Mendez⁵,
Jorge Sofronye Esmeral⁶

ABSTRACT

Reliable path-planning and generation of collision-free trajectories has become an area of active research over the past decade where the field robotics has probably been the most active area. This paper's main objective is to analyse the advantages and disadvantages of two of the most popular techniques used in collision-free trajectory generation in n -dimensional spaces. The importance of analysing such techniques within a generalised framework is evident as path-planning is used in a variety of fields such as designing medical drugs, computer animation and artificial intelligence and, of course, robotics. The review provided in this paper starts by drawing a historical map of path-planning and the techniques used in its early stages. The main concepts involved in artificial potential fields and probabilistic roadmaps will be addressed as these are the most influential methods and have been widely used in specialised literature.

Keywords: robotics, path planning, obstacle avoidance.

Received: October 2th 2009
Accepted: November 4th 2010

Introduction

The widespread use of robots in a variety of industrial processes during the 1980s and 1990s brought the need for generating trajectories to the foreground which were free from collisions between a robot and the environment it interacted with (Goldman, 1990). Although this task was initially performed by different specialists, autonomy has become an important aspect of robotics in more recent years and hence the need to develop more efficient algorithms. The task of collision-free trajectory planning (CFTP) comes naturally to humans and a variety of living organisms, but an abstraction of generating such a trajectory from start to goal position is not trivial and may become highly complex to implement.

The first attempt to plan the path of a robot was presented by Lozano-Pérez (Lozano, 1981). After the seminal work by Jean-Claude Latombe (Latombe, 1991), many researchers have addressed the problem of CFTP, but very few publications have closed the gap between newly-developed methods and the

¹Ingeniero Electromecánico, M.Sc., Universidad Nacional de Colombia, Colombia. ecardenasc@unal.edu.co

²Ingeniero Mecánico, M.Sc., Profesor asistente, Universidad Nacional de Colombia. Immendezm@unal.edu.co

³Ingeniero Eléctrico, Ph.D., Profesor asistente, Universidad Nacional de Colombia. jsufronye@unal.edu.co

⁴Electromechanical Engineer, M.Sc., Universidad Nacional de Colombia, Colombia. ecardenasc@unal.edu.co

⁵Mechanical Engineer, M.Sc., Assistant Professor, Universidad Nacional de Colombia, Colombia. Immendezm@unal.edu.co

⁶Electrical Engineer, Ph.D., Assistant Professor, Universidad Nacional de Colombia, Colombia. jsufronye@unal.edu.co

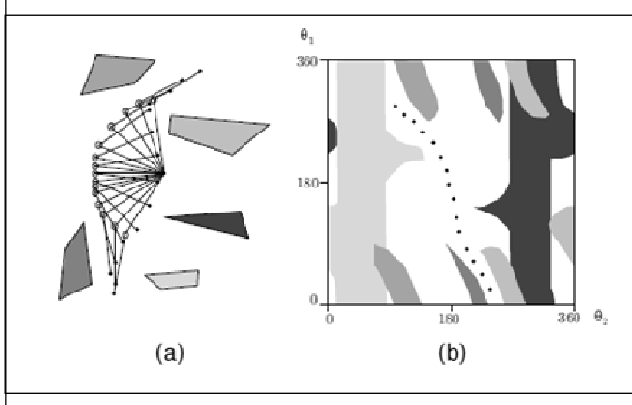
En español

ción de trayectorias sin colisiones, pero pocos libros como el de Howie Choset (Choset, 2005), han cerrado la brecha entre los conceptos fundamentales y los recientes progresos teóricos y prácticos.

El advenimiento de mayores velocidades en el procesamiento computacional ha permitido que estos métodos de planificación de caminos hagan parte de innovadoras aplicaciones prácticas, como es el caso de la cirugía robotizada (Sim, 2001; Goo, 2005), la animación en 3D (Kuffner, 1998; Wenhui, 2007), películas con actores virtuales, el desarrollo de videojuegos (Funge, 2004), el uso de robots asistentes en museos y hospitales (Maio, 1995), en la química realizando el modelamiento de cadenas moleculares (Kirillova, 2008) y en la exploración de otros mundos empleando robots móviles (Bresina, 2005; Carsten, 2009).

Existen varios métodos de origen geométrico para resolver el problema básico de planeamiento de caminos, y desde Latombe (1991) se han destacado tres métodos: campos potenciales artificiales (*Potential Functions*), mapas de carreteras (*Roadmaps*) y descomposición de celdas (*Cell Decompositions*). En la última década se ha desarrollado una nueva metodología denominada mapas probabilísticos (*Probabilistic Roadmaps*) (Choset, 2005), debido a que en entornos complejos, con un elevado número de grados de libertad o restricciones cinemáticas, los métodos geométricos pueden presentar limitaciones en cuanto al tiempo de cómputo necesario y el éxito en la generación de trayectorias.

Los algoritmos de generación de trayectorias requieren de la definición de *espacio de trabajo* (W) y *espacio de configuración* (Q). W es el entorno o lugar geométrico donde interactúa el robot en forma real con su entorno (espacio cartesiano); el espacio Q se define como el conjunto de todas las posibles configuraciones cinemáticas de un robot (Spong, 2006) y puede ser tratado como un espacio dimensional de orden dos, tres, cuatro, o n -dimensional.



theoretical aspects involved.

Recent technological developments have increased processing and computational capability, allowing novel path-planning techniques to be implemented as part of state-of-the-art applications such as robot-aided surgery (Sim, 2001; Goo, 2005), 3D animation (Sim, 2001; Goo, 2005), virtual actors in movies and video games (Funge, 2004), robot-assisted museum tours and hospital navigation (Maio, 1995), modelling chemicals and their molecular structure (Kirillova, 2008) and deep space exploration using mobile robots such as rovers (Bresina, 2005; Carsten, 2009).

Although several geometric methods can be used for solving basic path-planning problems, Latombe has highlighted three methods (Latombe, 1991): artificial potential fields (APF), roadmaps and cell decomposition. researchers have developed a technique termed probabilistic roadmaps during the last decade (Choset, 2005). It should be mentioned that geometric methods may be limited for robots having a large number of degrees of freedom or cinematic restriction sin highly complex environments due to long computational times and convergence issues.

Trajectory generation algorithms require the definition of two spaces: W denotes the *work space* and Q denotes the *configuration space*. W is the environment or geometric Cartesian space where a robot interacts with the "real" world; Q is defined as the set of all possible robot configurations defined by a cinematic relationship (Spong, 2006) and can be seen as an n -dimensional space, where n is the number of robot joints.

Figura 1. (a) Trayectoria para un manipulador de dos articulaciones por medio del espacio de trabajo. (b) Trayectoria en el espacio de configuración. Figura extraída de (Goldberg, 2009).

Figure 1: (a) Path of 2 DOF planar robot through the workspace. (b) Path in configuration space. Figure extracted from (Goldberg, 2009).

La representación del espacio de configuración permite establecer al robot como punto dentro de un espacio n -dimensional, con lo cual se puede trazar una trayectoria continua libre de colisiones desde una configuración inicial (*start*) hasta una final (*goal*).

Para el caso de un robot planar articulado de dos grados de libertad (GDL), Q corresponde a los valores que toman cada una de las dos articulaciones del robot, donde $Q \in \mathbb{R}^2$ y la trayectoria se calcula en un espacio bidimensional (figura 1). Si un robot tiene 6 GDL, la búsqueda de una trayectoria sin colisiones se

It should be stressed that configuration space Q allows a robot to be represented as a point in an n -dimensional space; it is thus possible to generate a continuous trajectory from an initial condition (i.e. *starting* configuration) to a final state (i.e. *goal* configuration)

Consider the case of a 2 degree-of-freedom (DOF) planar robot, where $Q \subset \mathbb{R}^2$ is defined as the space of all possible values which each joint (in this case q_1 and q_2) may take. In the particular case of a 2 DOF robot, Q is a two-dimensional space (Figure 1); for robots with 6 DOF, $Q \subset \mathbb{R}^6$ and the search have to be done in a

puede realizar dentro de $Q \in \mathbb{R}^6$.

Se dice que un objeto tiene n grados de libertad si su configuración puede ser especificada de forma mínima con n parámetros. Así pues, el número de GDL es igual a la dimensión de Q (Spong, 2006).

En las secciones 2 y 3 se exhiben los principios que rigen los dos principales métodos de generación de trayectorias (campos potenciales y mapas probabilísticos) dentro de entornos multidimensionales, esto debido a que la mayoría de estrategias tradicionales, como descomposición de celdas y *Roadmaps* se tornan imprácticas (Choset, 2005) y sólo asegurarían una trayectoria libre de colisiones para el efector final. Por último, se presentan las conclusiones y perspectivas en la sección 4.

Campos potenciales artificiales

El método de campos potenciales fue descrito primero por (Khatib, 1985) para aplicaciones de evasión de colisiones de forma *on-line* usando robots móviles con sensores de proximidad. La filosofía de este método es concebir el robot como una partícula eléctrica que es atraída hacia su configuración final q_{goal} mediante un potencial atractivo U_{att} , a la vez que es repelida por los obstáculos por medio de un potencial repulsivo propio U_{rep} . El campo potencial resultante es la adición de estos dos componentes al incluir todos los obstáculos.

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (1)$$

Así, el movimiento de la partícula puede ser tratado como un problema de optimización donde existe un mínimo global q_{goal} empezando desde una configuración q_{start} ; este problema puede abordarse utilizando el método de descenso del gradiente.

Potencial atractivo

Existen varios criterios para seleccionar un campo potencial atractivo, sin embargo la definición de mayor aceptación es la de que U_{att} incremente según la distancia desde q_{goal} , de acuerdo con la siguiente expresión:

$$U_{att}(q) = \begin{cases} \frac{1}{2} \zeta d^2(q, q_{goal}) & \text{Cuando: } d(q, q_{goal}) \leq d_{goal}^* \\ d_{goal}^* \zeta d(q, q_{goal}) - \frac{1}{2} \zeta (d_{goal}^*)^2 & \text{Cuando: } d(q, q_{goal}) > d_{goal}^* \end{cases} \quad (2)$$

Se define $d(q, q_{goal})$ como la distancia euclidiana entre una configuración q y la configuración final q_{goal} ; el parámetro ζ es usado para escalar el efecto atractivo del potencial. La fuerza atractiva se determina en dos partes: inicialmente la intensidad del campo varía linealmente con la distancia, al pasar el umbral $d(q, q_{goal}) \leq d_{goal}^*$, la atracción pasa a ser de orden cuadrática para que la intensidad del campo disminuya paulatinamente, o sea, es como una partícula que es atraída hacia su configuración final de manera uniforme hasta cierto punto, donde para asegurar una llegada suave a la meta se debe reducir su velocidad.

space having at least 6th order dimensionality.

A robot is said to have n -DOF if its configuration has minimal representation with n parameters (i.e. a robot's DOF are equal to the number of joints or Q dimensionality) (Spong, 2006).

Artificial potential fields

The philosophy behind APF was first described by (Khatib, 1985) and used as an "on-line" strategy for collision avoidance in mobile robots having proximity sensors. The main idea behind APF is to conceive the robot as an electrically-charged particle influenced by a charged field generated by the goal position and the obstacles (i.e. the particle is attracted by the goal configuration and repelled by obstacles through potential fields U_{att} and U_{rep} respectively). The resulting potential field is the sum of these two components and can be generated by

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (1)$$

Particle motion can be stated as an optimisation problem having a *global* minimum at q_{goal} and a set of initial conditions q_{start} . The most common approach is to use gradient descent methods to solve this problem.

Attractive potential field

Several criteria may be used to select an appropriate attractive potential field. Nonetheless, the most commonly used potential is a mixture of conic and parabolic convex hulls. It should be noticed that potential grows according to the distance between the particle (the robot) and q_{goal} . An expression is given by

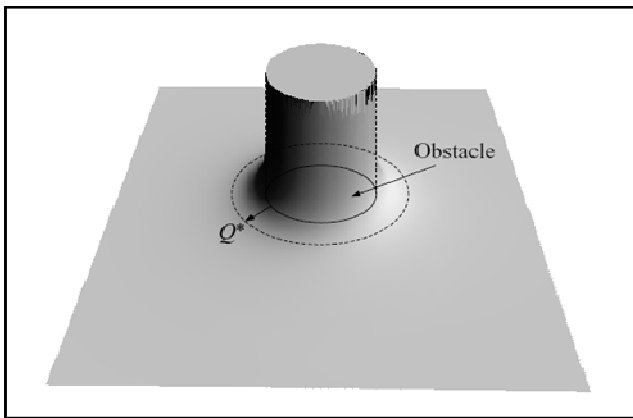
$$U_{att}(q) = \begin{cases} \frac{1}{2} \zeta d^2(q, q_{goal}) & \text{Cuando: } d(q, q_{goal}) \leq d_{goal}^* \\ d_{goal}^* \zeta d(q, q_{goal}) - \frac{1}{2} \zeta (d_{goal}^*)^2 & \text{Cuando: } d(q, q_{goal}) > d_{goal}^* \end{cases} \quad (2)$$

The Euclidian (norm) distance between actual configuration q and goal configuration q_{goal} is given by $d(q, q_{goal}) = \|q_{goal} - q\|$; the user-defined parameter ζ is used to scale the effect of the attractive potential. The resulting attractive field can be determined in one of two ways: initially, the field varies linearly with distance when $d(q, q_{goal})$ is above a certain threshold but once $d(q, q_{goal}) \leq d_{goal}^*$, the attractive field becomes quadratic, ensuring that the field is continuously differentiable around q_{goal} .

Potencial repulsivo

El objeto del potencial repulsivo es el de mantener a la partícula (robot como un punto en el Q) apartada de un obstáculo. La magnitud de la fuerza del campo debe aumentar con la proximidad al obstáculo. Por tanto, el potencial repulsivo es definido en términos de la distancia más corta $d_i(q)$ que existe a un obstáculo dado QO_i . Para evitar colisiones con el obstáculo QO_i el campo potencial debe producir un efecto repulsivo asintótico sólo en las cercanías del obstáculo, por lo cual se puede definir el potencial repulsivo como:

$$U_{rep_1}(q) = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{d_1(q)} - \frac{1}{Q_i^*} \right)^2, & d_1(q) \leq Q_i^* \\ 0, & d_1(q) > Q_i^* \end{cases} \quad (3)$$



Repulsive potential field

The repulsive field's main purpose is to keep a particle (i.e. the robot as a point in Q) away from any obstacle. The potential field's magnitude is inversely proportional to the distance between the obstacle and the particle; the field is thus defined in terms of the shortest distance $d_i(q)$ to an obstacle given by QO_i . In order to avoid any collisions with the i -th obstacle, the potential field must asymptotically repel the particle in the vicinity of the obstacle. Bearing this in mind, the potential can be defined as

$$U_{rep_1}(q) = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{d_1(q)} - \frac{1}{Q_i^*} \right)^2, & d_1(q) \leq Q_i^* \\ 0, & d_1(q) > Q_i^* \end{cases} \quad (3)$$

Figura 2. Representación del campo repulsivo de un obstáculo QO_i . Imagen tomada del libro de (Choset, 2005).

Figure 2: The repulsive field only operates in a domain near obstacle QO_i . Figure extracted from Choset(Choset, 2005)

$Q_i^* \in \mathbb{R}$ es un umbral que permite al robot ignorar un obstáculo que se encuentra lo suficientemente lejos, y η puede ser visto como la ganancia en el gradiente repulsivo. Estos escalares son generalmente determinados por ensayo y error. Una representación de este efecto repulsivo sobre un obstáculo circular se puede apreciar en la figura 2. La figura 3 muestra a $d_i(q)$ medida desde c_o , que es el punto más cercano al obstáculo QO_i , y cuyo gradiente es

$$\nabla d_i(q) = \frac{q - c_o}{d(q, c_o)} \quad (4)$$

Un inconveniente de esta formulación es el de que el cálculo de la distancia mínima a un robot no siempre es evidente en W , por tanto una alternativa expuesta en algunos de los textos especializados es seleccionar un subconjunto de puntos pertenecientes al robot, denominados *puntos de control* (r_j) y luego se define un potencial U_j por cada uno de estos puntos. Con la determinación del efecto sobre todos los puntos de control se realiza un mapeo de las fuerzas actuantes desde el espacio de trabajo al espacio de configuración y posteriormente se calcula una sumatoria de fuerzas en Q para obtener el efecto combinado sobre todo el robot, el cual ahora sí puede ser representado como un punto.

$Q_i^* \in \mathbb{R}$ is a threshold that allows a robot to ignore any faraway obstacles. Scalar η is a weighted repulsive field. These constants are generally chosen by trial and error. Figure 2 depicts a circular obstacle's repulsive field. Figure 3 shows $d_i(q)$ measured from c_o , which is the nearest point to obstacle QO_i and whose gradient is given by

$$\nabla d_i(q) = \frac{q - c_o}{d(q, c_o)} \quad (4)$$

A drawback to such an approach is that the calculation of the shortest distance in W is not always trivial; an alternative approach (which has been explored in several texts) is thus to define a set of points belonging to the robot (i.e. control points). A potential field U_j is then generated for each point (r_j) within the control set. After determining the effect of the field on each point, total field effect in W can be mapped as forces acting on a robot in the Q space. The combined effect of the field acting on the robot is calculated as the sum of all the forces in the q space, where the robot is represented as a particle. It should be noticed that summing the forces in the W space and then transforming this vector to the Q space is not equal to summing the transformed forces in the Q space. The latter is the correct way.

En español

In English

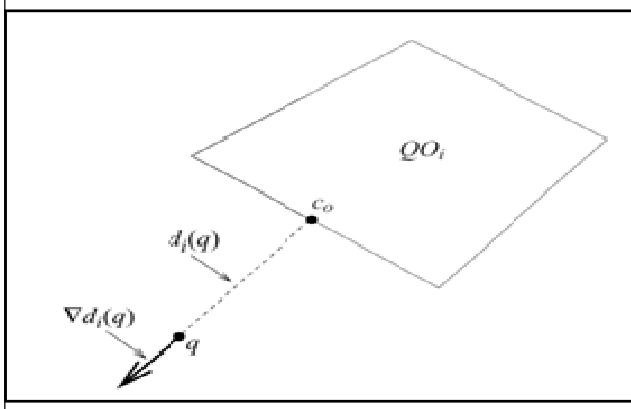


Figura 3. Distancia más corta a un obstáculo QO_i , en dirección del gradiente de $d_i(q)$ (Choset, 2005).

Figure 3: Shortest distance to obstacle QO_i and direction to gradient of $d_i(q)$. Figure extracted from Choset (Choset, 2005)

Efecto combinado

En una situación en la que un robot esté formado por uno o varios cuerpos sólidos, se pueden establecer varios puntos de control j , pero a la vez pueden existir uno o varios obstáculos i , de ahí que el campo potencial artificial resultante puede ser visto como:

$$U(q) = \sum_j U_{att_j} + \sum_{ij} U_{rep_{ij}} \quad (5)$$

$$\nabla U(q) = \sum_j J_j^T(q) \nabla U_{att_j}(q) + \sum_{ij} J_j^T(q) \nabla U_{rep_{ij}}(q)$$

El término $J_j(q)$ es la matriz jacobiana para cada punto de control r_j . El análisis de jacobianos es estudiado en la teoría de robótica y permite enlazar los espacios de trabajo y configuración mediante la cinemática directa, tal como se encuentra en (Craig, 1989). Para cada punto de control las expresiones de gradiente de los campos potenciales atractivo y repulsivo j son calculadas de las expresiones (2) y (3), obteniendo:

$$\nabla U_{att}(q) = \begin{cases} \zeta(q - q_{goal}) \\ \text{if: } d(q, q_{goal}) \leq d_{goal}^* \\ \frac{d_{goal}^* \zeta(q - q_{goal})}{d(q, q_{goal})} \\ \text{if: } d(q, q_{goal}) > d_{goal}^* \end{cases} \quad (6)$$

$$\nabla U_{rep_i}(q) = \begin{cases} \eta \left(\frac{1}{d_i(q)} - \frac{1}{Q_i^*} \right)^2 \frac{1}{d_i^2(q)} \nabla d_i(q), d_i(q) \leq Q_i^* \\ 0, d_i(q) > Q_i^* \end{cases} \quad (7)$$

Es razonable que exista mayor o igual cantidad de puntos de control que el número de grados de libertad. En el caso de un robot con articulaciones rotacionales los puntos de control pueden ser ubicados en los marcos de referencia articulares.

Una vez obtenido el efecto del campo potencial artificial sobre el robot en un momento dado, es necesario obtener la siguiente configuración; esto se puede lograr con el método de descenso del gradiente, método comúnmente utilizado para resolver problemas de optimización. Las direcciones de las fuerzas resultantes del campo potencial están dadas por las velocidades de descenso, "colina abajo", hacia un estado mínimo.

La idea del algoritmo de descenso del gradiente es simple: se

Combined field effect

Considering a robot to be one or more rigid bodies (in the case of robotic manipulators), several control points r_j can be determined, as expressed above. The effect of having more than one obstacle must also be considered. As a result, the total artificial potential field is given by

$$U(q) = \sum_j U_{att_j} + \sum_{ij} U_{rep_{ij}} \quad (5)$$

$$\nabla U(q) = \sum_j J_j^T(q) \nabla U_{att_j}(q) + \sum_{ij} J_j^T(q) \nabla U_{rep_{ij}}(q)$$

The term $J_j(q)$ is the Jacobian matrix for each control point r_j . The Jacobian matrix is widely used in robotics and allows forces and speeds to be mapped in the work-space and transferred to the configuration space (Craig, 1989). The gradient of the attractive and repulsive potential fields of the j -th component are calculated for each control point by means of the expressions give in (2) and (3). This results in the following expressions

$$\nabla U_{att}(q) = \begin{cases} \zeta(q - q_{goal}) \\ \text{if: } d(q, q_{goal}) \leq d_{goal}^* \\ \frac{d_{goal}^* \zeta(q - q_{goal})}{d(q, q_{goal})} \\ \text{if: } d(q, q_{goal}) > d_{goal}^* \end{cases} \quad (6)$$

$$\nabla U_{rep_i}(q) = \begin{cases} \eta \left(\frac{1}{d_i(q)} - \frac{1}{Q_i^*} \right)^2 \frac{1}{d_i^2(q)} \nabla d_i(q), d_i(q) \leq Q_i^* \\ 0, d_i(q) > Q_i^* \end{cases} \quad (7)$$

It is reasonable to have a number of control points equal to or greater than the number of DOF. In the case of a rotational-joint manipulator, it is common to place this control point on the joints' coordinate frames.

The subsequent configuration must be calculated once the effect of the potential field on a robot has been determined for a given step; this can be achieved by using a gradient descent algorithm which has been widely used for solving optimisation problems. The direction of the resulting forces produced by the field is calculated using "down-hill" velocities taking the particle to a minimum state.

The general idea of gradient descent methods is simple and intuitive; a start is made from an initial configuration and then a

empieza desde una configuración inicial, luego se toma un pequeño paso en dirección del gradiente negativo, lo cual nos da como resultado una nueva configuración; este proceso se repite hasta que el gradiente sea cero. La definición de dicho algoritmo para la implementación en aplicaciones de campos potenciales se aprecia en la Figura 4.

Descenso del gradiente	
1	$q(0) \leftarrow q_{start}$
2	$i \leftarrow 0$
3	while $\ U(q(i))\ > \epsilon$ do
4	$q(i+1) \leftarrow q(i) + \alpha(i)U(q(i));$
5	$i \leftarrow i+1$
6	end while

Figura 4. Algoritmo del descenso del gradiente.

La variable $q(i)$ es el valor de q en la i -ésima iteración; la solución consiste de una secuencia de configuraciones $\{q(0), q(1), q(2), \dots, q(i)\}$. El valor escalar $\alpha(i)$ debe ser lo suficientemente pequeño para que el robot no se salte obstáculos. Es improbable que el gradiente llegue a ser exactamente cero, por tal razón se establece la tolerancia como un valor suficientemente pequeño según los requerimientos de operación.

Ejemplos

Las figura 5 y 6 muestran la generación de trayectorias usando campos potenciales artificiales. El robot móvil rectangular de la figura 5(a) está en un W bidimensional, pero el Q es tridimensional, pues el robot es representado por la posición (x, y) del centroide y el ángulo θ que indica la orientación; es decir, en cualquier instante el robot tiene una configuración $q(x, y, \theta)$. En W se eligen los puntos de control r_j en los vértices del rectángulo. Usando el algoritmo del descenso del gradiente se calcularon iterativamente las distancias entre los r_j actuales y los de la posición final para determinar el efecto atractivo, además se calculan las distancias mínimas entre cada r_j y los diferentes obstáculos para determinar el efecto repulsivo total.

“small” step is taken in the gradient's negative direction; this results in a new configuration. The gradient is calculated in this new configuration and a step is taken in this gradient's negative direction. This process is repeated until the gradient reaches zero. Figure 4 gives a basic gradient descent algorithm which has been effectively implemented in APF path-planning applications.

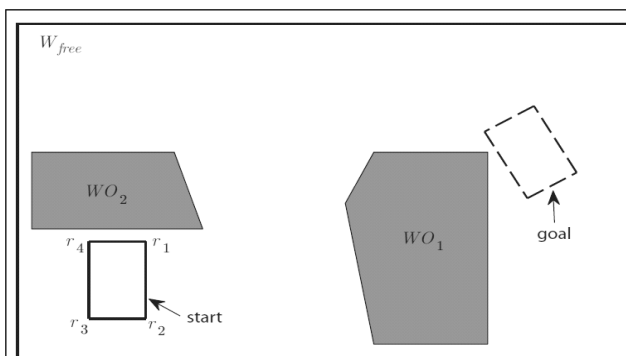
Gradient descent	
1	$q(0) \leftarrow q_{start}$
2	$i \leftarrow 0$
3	while $\ U(q(i))\ > \epsilon$ do
4	$q(i+1) \leftarrow q(i) + \alpha(i)U(q(i));$
5	$i \leftarrow i+1$
6	end while

Figure 4. Gradient descent algorithm

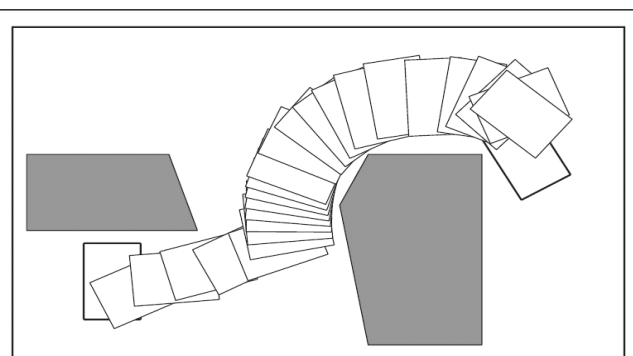
The variable $q(i)$ is the vector of joint values in the i -th iteration; the solution to the path-planning problem consists of a sequence of configurations $\{q(0), q(1), q(2), \dots, q(i)\}$. The scalar value $\alpha(i)$ must be sufficiently small so that a robot does not “jump” over obstacles. As the gradient may never reach zero in practice, a small tolerance value ϵ must be introduced; the size of this scalar depends on the application at hand.

Examples

Figures 5 and 6 show how APFs are applied to path-planning. The rectangular mobile robot shown in Figure 5(a) is in a two-dimensional space Q and a three-dimensional space W defined by the (x, y) position of centroid and orientation angle θ (i.e. the robot has a $q(x, y, \theta)$ configuration at any given instant). Control points r_j are defined in W for each of the rectangle's vertices. Using the gradient descent method, the shortest distances between j control points and obstacles are calculated in each iteration and the attractive force of the goal configuration on each control point is used for calculating the total force acting on the particle.



(a)



(b)

Figura 5. Ejemplo de generación de trayectoria con campos potenciales para un robot móvil rectangular. (a) Estados inicial y final en el espacio de trabajo. (b) Desplazamiento del robot en su entorno

Figure 5: Path-planning example with potential fields for a rectangular mobile robot. (a) initial and final positions in the workspace, (b) robot movement in its environment.

En cada iteración del algoritmo de descenso del gradiente se calcula el total del campo potencial y se obtiene por resultado una nueva configuración q del robot con el fin de utilizarse en la siguiente iteración. El parámetro α debe ser pequeño y ajustado

Figure 5(b) shows various positions of the path generated by an APF algorithm using the gradient descent method. It can be observed that the repulsive and attractive fields must be calculated for each control point r_j ; hence, if there are a large number of

En español

manualmente para que la trayectoria generada permita la evasión de obstáculos, esto es, que el robot no pase por alto el obstáculo más pequeño requerido. La figura 5(b) exhibe varias posiciones intermedias que representan la trayectoria calculada para el robot rectangular. Es evidente que se requiere calcular varias veces el efecto repulsivo y atractivo para cada r_j ; si existen muchos obstáculos y muchos r_j , el proceso de cómputo emplea más tiempo, por tal razón esta estrategia de generación de trayectorias libres de colisiones es más apropiada para robots móviles con ejecución *online* y realimentación sensorial de distancias mínimas a los obstáculos. La ventaja del método radica en que se puede expandir fácilmente la metodología a entornos multidimensionales, por ejemplo, un robot aéreo podría tratarse en un espacio Q de seis dimensiones correspondientes a sus seis grados de libertad.

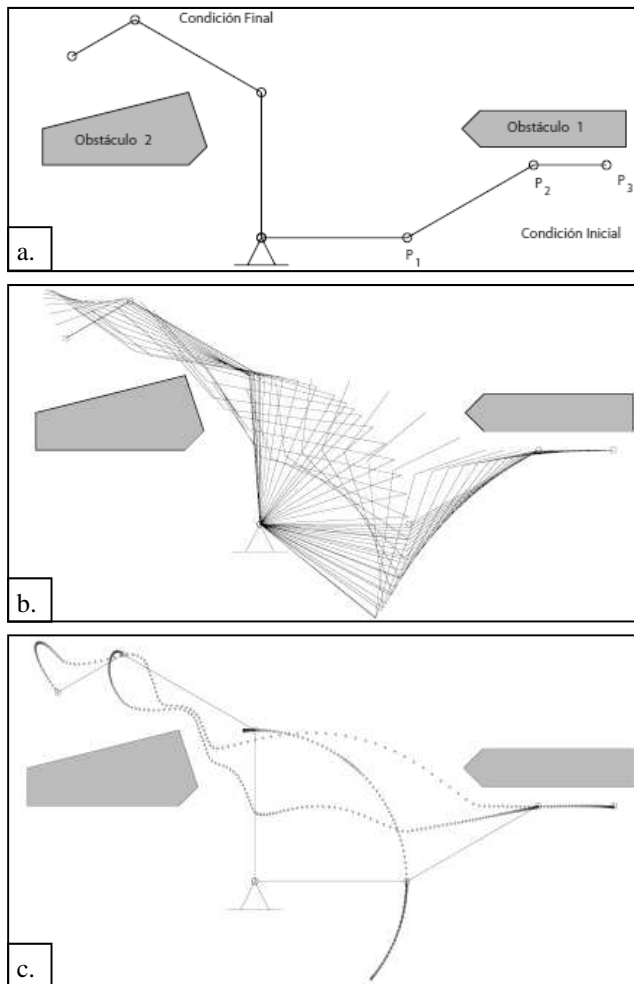


Figura 6: Comportamiento de un robot redundante de tres eslabones sometido a campos potenciales. (a) Estado inicial y final, (b) Estados intermedios debido al campo potencial artificial, y (c) Generación de la trayectoria libre de colisiones.

Con las simulaciones, dos efectos peculiares fueron detectados, el primero resulta en la presentación de vibraciones cuando el robot intenta sobrepasar un obstáculo, el cual está cerca a una situación de un mínimo local. Esto es debido a un paso α grande, lo cual es una consecuencia directa de la estrategia de descenso del gradiente. Segundo, el robot no llega exactamente a la posición de meta, pues el algoritmo de descenso del gradiente

In English

obstacles, and the number of control points is also large, computational cost increases and so does the calculation time. APF path planning methods are thus well-suited for on-line applications in mobile robotics where the minimum distance to the obstacles is feedback by a distance sensor array. One of the advantages of APF techniques is that they can be easily implemented in highly dimensional spaces. For example, an aircraft can be treated in a six dimensional space corresponding to its six DOF.

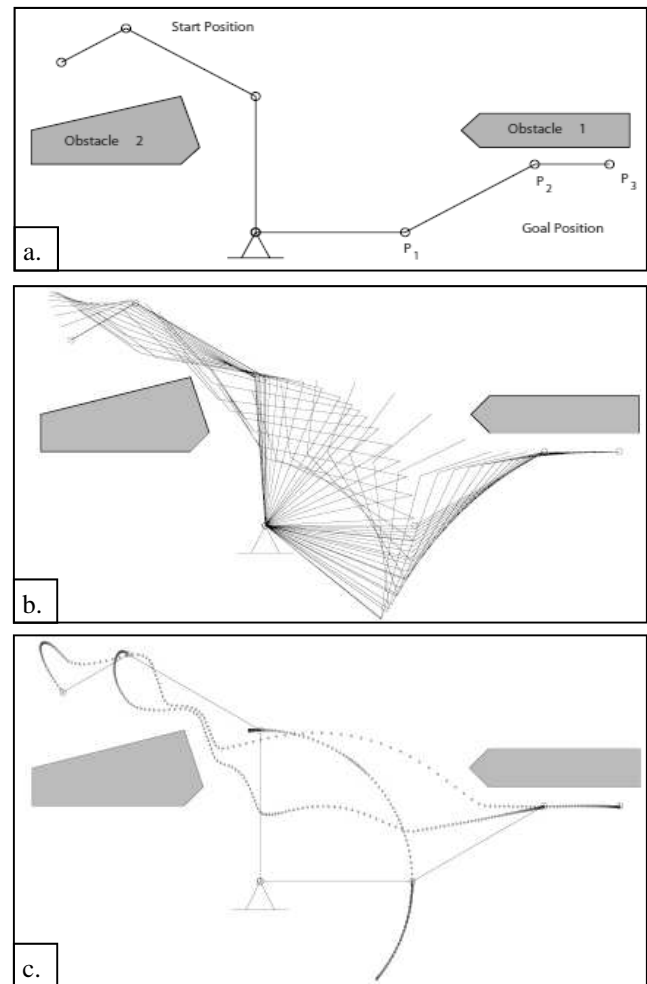


Figure 6: Planar three-joint robot behaviour using the APF technique. (a) Start position and goal position, (b) Intermediate position due to the APF, and (c) Collision-free path.

Two particular patterns were observed during simulation. It was observed that some vibrations were present when the robot tried to avoid an obstacle near a local minimum. This vibration resulted from large step size α and is common in gradient descent methods. Secondly, the robot did not reach goal configuration as the gradient descent method never reaches $U(q(i)) = 0$. Post-processing algorithms can be used to avoid such pattern as they smooth-out such vibrations and allow a robot to reach its goal

En español

In English

no acaba cuando $U(q(i)) = 0$. Para evitar efectos indeseados se sugiere realizar un posprocesamiento para afinar la trayectoria resultante.

En el caso de un robot articulado la metodología que se aplica es la misma que se expuso, la diferencia radica en la ubicación de los puntos de control, que se pueden localizar en el efector final y en los ejes articulares, con la advertencia de agregar nuevos puntos de control en el cuerpo de los eslabones para evitar la inadvertencia de obstáculos debido a geometrías irregulares. Se puede controlar la agilidad de los algoritmos implementados restringiendo los efectos repulsivos en aquellas articulaciones de las cuales se sabe de antemano que nunca van a colisionar con ciertos obstáculos. En la figura 6 se presentan los resultados de aplicar el método de campos potenciales artificiales a un robot planar de tres eslabones y tres articulaciones rotacionales.

Las simulaciones con robots articulados redundantes han demostrado que aunque se presentan mínimos locales para un punto de control en particular, el efecto combinado del campo atractivo y repulsivo en los restantes puntos de control hace que se pueda escapar de esa condición de atascamiento, y aunque usualmente se requiere un número grande de iteraciones, al emplear un algoritmo de posprocesado se consigue afinar fácilmente la trayectoria resultante.

Mapas probabilísticos

Los mapas probabilísticos (PRM), desde sus inicios, han demostrado un tremendo potencial para solucionar el problema de encontrar una ruta libre de colisiones en entornos multidimensionales (Kavraki, 1996).

La idea central del PRM es distribuir un conjunto de puntos (nodos) de manera aleatoria en el espacio de configuración libre de colisiones, Q_{free} . Posteriormente, con un "planificador local", se establecen las conexiones (enlaces) entre nodos y se descartan aquellas que produzcan colisión dentro de W . Al conjunto de nodos y enlaces se le denomina *Roadmap*. La configuración inicial q_{start} se agrega al *Roadmap* mediante un enlace libre de colisiones hasta el nodo más cercano; se hace lo mismo con q_{goal} . Luego se traza una trayectoria entre q_{start} y q_{goal} utilizando algoritmos como A* o el algoritmo de Dijkstra, con la finalidad de encontrar la ruta más corta desde q_{start} a q_{goal} (figura 7). Un enlace entre nodos es una línea recta compuesta por puntos, que representan una sucesión de configuraciones del robot en el espacio de trabajo, las cuales no deben presentar un choque con el entorno para que este enlace sea válido.

position exactly.

The methodology applied for articulated robot is the same as mentioned previously where the main difference is control point location. Control points are usually located at the end effectors, the joints and some extra control points on the body of each joint. The latter is recommended to avoid any unwanted joint contact with any of the obstacles due to irregular geometries. Figure 6 shows the results of applying an APF path-planning algorithm to a planar three-joint robot.

Simulating redundant-joint robots has shown that even though local minima may exist for a particular control point, the attractive/repulsive fields' combined effect on the remaining points lets the system escape such "trapped" configuration. Escaping such trapped configurations requires a large number of iterations which can be smoothed-out later on by post-processing algorithms.

Probabilistic roadmaps

Since their first appearance, probabilistic roadmaps (PRM) have been shown to be well-suited to the problem of finding collision-free trajectories in large dimension spaces and highly-cluttered environments (Kavraki, 1996).

The main idea behind PRM is simple. A set of node points is randomly distributed within collision-free space Q_{free} . Connections between these nodes (links) are then established by using local planners where only the links lying within the collision-free space are used. The set of all nodes and links is called a *roadmap*. The initial configuration is added through a link connecting q_{start} to the nearest node. Once a roadmap has been determined, a path can be established from q_{start} to q_{goal} by using algorithms such as A* or Dijkstra. The shortest path between start and goal configurations can thus be found (see Figure 7). A link is a straight line joining two nodes and represents a series of successive robot configurations in work space W . A valid link must guarantee that no configuration composing it collides with any of the obstacles. This may be seen as a disadvantage.

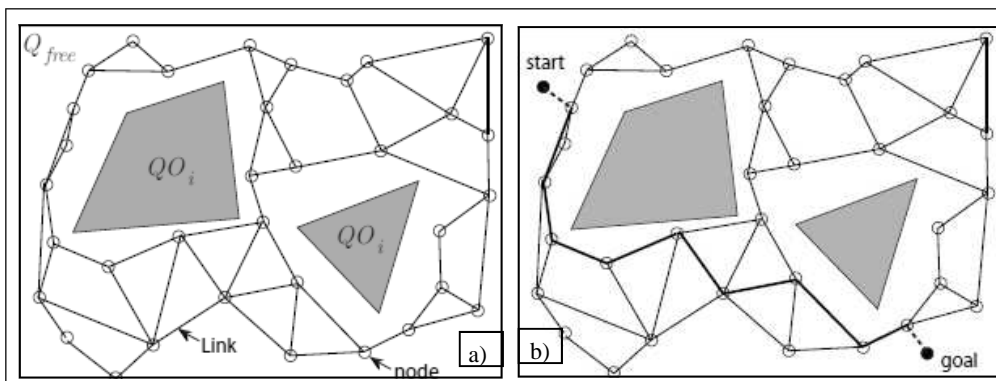


Figura 7: Generalización del funcionamiento de los algoritmos PRM en un entorno bidimensional. (a) *Roadmap* compuesto de nodos y enlaces. (b) Adición de los puntos inicial y final, junto con la ruta más corta generada sobre el *Roadmap*.

Figure 7: an example of how PRM algorithms work in a two-dimensional environment. (a) nodes and links depicted on the roadmap. (b) Adding start and end nodes, with the shortest route generated on the roadmap.

El costo computacional para los algoritmos PRM es determinado por el número de veces que se invoque el algoritmo de detección de colisiones, y para tornar más eficiente al PRM se debe tratar de reducir el tamaño del *Roadmap*. De la anterior necesidad han surgido los métodos de exploración de entorno por medio de árboles: *Expansive Spaces Trees* (EST) y *Rapidly Exploring Random Trees* (RRT). El EST se encuentra explicado a cabalidad en el trabajo de David Hsu (Hsu, 2000), sin embargo el RRT es el más empleado actualmente.

Algoritmo RRT

El RRT fue introducido como un algoritmo capaz de explorar eficientemente el espacio de configuración desde q_{start} (LaValle, 1999; LaValle, 2001). El RRT básico opera con la construcción de un árbol T compuesto de nodos y enlaces que se incrementan paulatina y aleatoriamente desde un punto de origen, tal y como se propone en el algoritmo de la figura 8. En este algoritmo se determina una configuración aleatoria q_{rand} libre de colisión y a una distancia cercana al árbol; éste a su vez crece con un enlace hacia esa configuración, para ello se usa la función EXTEND presentada en la figura 9.

BUILD_RRT	
1	$T(0) \leftarrow q_{start}$
2	for $i = 1$ to K do
3	$q_{rand} \leftarrow \text{RANDOM_CONFIG}$
4	EXTEND(T, q_{rand})
5	Return T

Figura 8. Construcción del RRT básico.

Figure 8: The basic RRT construction algorithm

The computational cost of PRM algorithms relies heavily on the number of times the collision detection algorithm has to be invoked. Many algorithms thus aim at reducing the number of links in the roadmap. This has given rise to algorithms exploring their environment through expansive trees, for example expansive-space trees (EST) and rapidly-exploring random trees (RRT). For a detailed description of EST algorithms, see the work of David Hsu (Hsu, 2000). Although RRT algorithms are widely used, no work has explained recent developments in detail.

Rapidly-exploring random trees

Rapidly-exploring random trees (RRT) was introduced by LaValle (LaValle, 1999; LaValle, 2001) and soon proved an efficient way of exploring configuration space and thus generating roadmaps joining q_{start} and q_{goal} . The basic RRT algorithm constructs a tree T by using nodes and links that gradually growing in a random fashion; they stop once start and goal configurations become joined. The result of this procedure is shown in Figure 8. In the RRT algorithm, a random, collision-free configuration q_{rand} is created within a neighbourhood of the tree. Once the configuration has been created, the tree tries to grow towards this configuration by using an EXTEND function, as shown in Figure 9

EXTEND(T, q)	
1	$q_{near} \leftarrow \text{NEAREST-NEIGHBOR}(T, q)$
2	if NEW-CONFIG(q, q_{near}, q_{new})
3	ADD-VERTEX(T, q_{new})
4	ADD-EDGE(T, q_{near}, q_{new})
5	if $q_{new} = q$ then
6	Return <i>Reached</i>
7	else
8	Return <i>Advanced</i>
9	Return <i>Trapped</i>

Figura 9. Función EXTEND del algoritmo RRT básico. Tomado de (Kuffner, 2000).

Figure 9: EXTEND function of RRT-basic algorithm. Extracted from Kuffner (Kuffner, 2000)

La naturaleza del RRT le hace avanzar con más avidez en aquellas zonas en las que haya mayor espacio libre, pues es allí donde hay más posibilidad de encontrar puntos q_{rand} factibles. Esto puede comprenderse observando el crecimiento del árbol en la figura 10.

Para encontrar una trayectoria libre de colisiones, el RRT inicia su exploración en q_{start} como raíz del árbol T . Si el crecimiento de este árbol alcanza q_{goal} o se cumple un número límite de iteraciones K , mostrado en la Figura 8, el algoritmo RRT culmina y luego se traza una ruta desde q_{goal} a q_{start} . En otras palabras, se puede establecer un camino continuo entre nodos empezando desde las ramas, pasando por el tronco, hasta llegar a la raíz.

The nature of RRT allows for rapid expansion in regions where no obstacles are present. This is the case as obstacle-free regions present a greater chance of feasible point q_{rand} being found. This can be observed in Figure 10 depicting the expansion of a tree.

The RRT algorithm starts exploring the neighbourhood of q_{start} as the root of T to find a collision-free trajectory. If this tree's growth reaches goal configuration, or a maximum number of iterations is reached, the algorithm stops and a path is drawn from q_{goal} to q_{start} ; this process is depicted in Figure 8. A continuous path can thus be established starting from the branches, extending through the tree trunk and finally reaching the root.

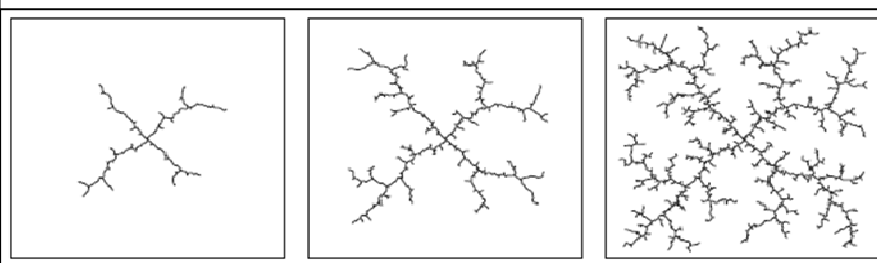


Figura 10. Representación en 2D del crecimiento en forma de árbol, proporcionado por el algoritmo RRT. Tomado de (Kuffner, 2000).

Figure 10: RRT algorithm growing like a tree in a 2D environment. Extracted from Kuffner (Kuffner, 2000)

En español

In English

En los últimos años se han producido diferentes mejoras al RRT básico con resultados comprobados en la práctica (LaValle, 2006). En particular, la versión denominada *RRT_Connect*, desarrollada por Kuffner y LaValle (Kuffner, 2000), ha demostrado ser una de las variantes más confiables para todo tipo de aplicaciones.

La idea principal del *RRT_Connect* es el desarrollo de dos árboles que inician en los puntos inicial y final, ambos árboles crecen simultáneamente uno hacia el otro, permitiendo una mejora en la convergencia del algoritmo, así como en el tiempo de cómputo, debido al menor número de comprobaciones de colisión.

La figura 11 presenta el resultado de utilizar el *RRT_Connect* en un entorno de búsqueda R^2 que contiene tres obstáculos poligonales (obsérvense las semejanzas con el espacio de configuración de un robot planar de dos articulaciones ilustrado en la figura 1). Para las rutas obtenidas con el método RRT es conveniente aplicar algoritmos de posprocesado con el objetivo de reducir su irregular topología; lo deseable es utilizar algoritmos de simplificación rápidos y sencillos para no comprometer el tiempo de cómputo total. Se ha observado que los algoritmos RRT son bastante rápidos en entornos abiertos (LaValle, 2001; Yershova, 2005) y además, más efectivos que otros métodos (Brandt, 2006).

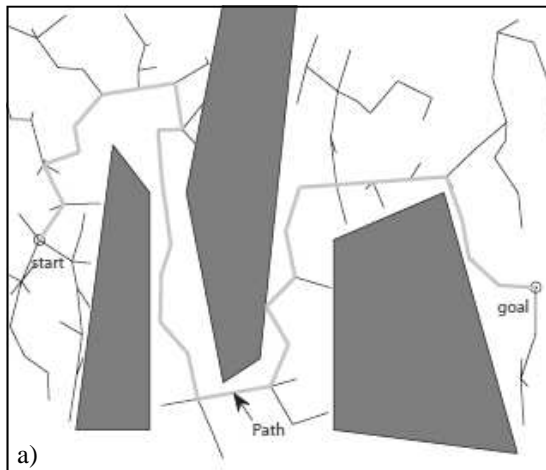


Figura 11. (a) Ruta generada por el algoritmo *RRT_Connect* en un entorno de búsqueda en 2D. (b) Ruta conseguida por posprocesado.

Conclusiones y prospectivas

En este artículo se han presentado las bases teóricas necesarias para la implementación de dos estrategias de planeamiento de trayectorias libres de colisiones en sistemas n -dimensionales. La primera, campos potenciales artificiales, permite una rápida respuesta ante el entorno, generando un camino continuo y generalmente suave. Su principal desventaja está en la aparición de mínimos locales que llegan a producir un estancamiento del algoritmo empleado (Latombe, 1991). Adicionalmente, existe otro contratiempo: el número de puntos de control. Si no hay suficientes, existe el riesgo de colisionar obstáculos inadvertidos (Choset, 2005). Es concluyente que el método de campos poten-

Modifications have been made to the basic RRT in recent years and particle results have been detailed in LaValle (LaValle, 2006). The algorithm termed *RRT_Connect*, developed by Kuffner and LaValle (Kuffner, 2000), has been shown to be one of the most reliable variations. It is also worth mentioning that the *RRT_Connect* algorithm has been used in a vary of applications with encouraging results.

The main idea behind *RRT_Connect* is to use two trees: one beginning from the start configuration, the other starting from goal configuration. Both trees grow towards each other simultaneously, allowing better convergence rates, as well as less computational cost due to the reduced number of collision checks.

Figure 11 shows the result of applying the *RRT_Connect* algorithm to a search space in R^2 . The environment contains three obstacles represented by irregular polygons. Notice the resemblance with the configuration space of the 2 DOF planar robot depicted in Figure 1. For paths generated using the RRT algorithm, and its variations, it is recommended to use post-processing algorithms to reduce irregular topologies and sharp turns; the objective is to smooth the path without compromising calculation time due to overly complex post-processing algorithms. It has been noted that RRT algorithms are extremely fast in open spaces (i.e. an environment having few obstacles) (LaValle, 2001; Yershova, 2005) and it has been shown to be more efficient than other probabilistic methods (Brandt, 2006).

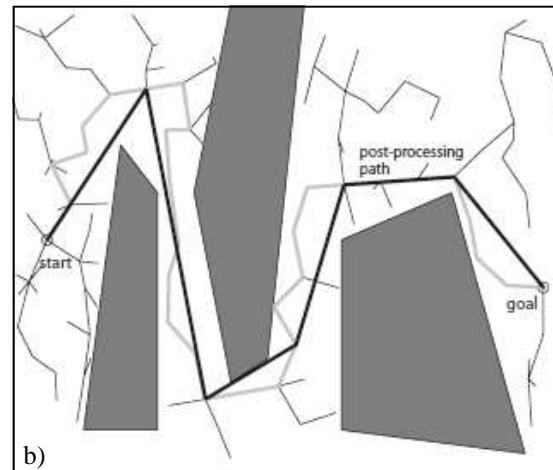


Figure 11: (a) Path generated by the *RRT_Connect* algorithm in two-dimensional search space with three obstacles. (b) Path obtained after post-processing.

Conclusions and future work

This article has presented the fundamental theoretical concepts needed for implementing two of the most commonly-used collision-free path-planning strategies for robotic systems in an n -dimensional search space. The first methodology, APF, allows a quick response to its environments and usually generates a smooth path requiring little post-processing. Its most noticeable weakness concerns regarding local minima and possible entrapment that may arise (Latombe, 1991) and the number of control points. If a reduced number of control points is used, the algorithm exhibits better convergence rates but the robot may collide with objects if we consider that it is a rigid body and not just a

ciales requiere ajustar varios parámetros para que la trayectoria generada sea satisfactoria. Además, su implementación y análisis en el campo de la robótica demostró gran dependencia del método con el cálculo de los jacobianos del sistema, y de la obtención de la distancia mínima de un punto de control a un polígono o a un sólido.

En la segunda estrategia, que contempla los mapas de enrutamiento probabilístico o PRM, se destacaron en el uso e implementación el algoritmo RRT y su variante RRT-Connect, cuyas fortalezas son: la concepción generalizada para cualquier dimensión del entorno de búsqueda, donde no se requiere conocer totalmente el espacio de trabajo (Choset, 2005); su desempeño es rápido en entornos con pocos obstáculos, siendo capaz de sortear mínimos locales, y se logró comprobar su fiabilidad en entregar una respuesta (LaValle, 2001), pues se explora el espacio libre de forma sistemática. Algunas dificultades o contratiempos que presenta este método, son: la necesidad de verificar rápidamente si una configuración produce una colisión en el espacio de trabajo, ya que se deben hacer cientos de comprobaciones por segundo antes de generar una trayectoria. La configuración de la ruta encontrada es difícil de seguir en algunos casos prácticos, por lo cual es necesario un posprocesado rápido y eficiente. Adicionalmente, no se puede asegurar que la ruta obtenida sea óptima, incluso utilizando un posprocesado intensivo. Crear una ruta suavizada requiere un segundo posprocesado, utilizando geometría avanzada para generar curvas (Angeles, 2007; Kwangjin, 2008a; Kwangjin, 2008b).

Comparando los dos métodos analizados, implementados y probados, encontramos que el algoritmo RRT (y sus variantes), es más fiable en la consecución de una trayectoria libre de colisiones en sistemas multi-dimensionales en comparación con el método que utiliza campos potenciales artificiales. Hemos efectuado distintas simulaciones que implican trazar trayectorias empleando el algoritmo RRT_Connect y los resultados siempre han sido satisfactorios, aunque su costo computacional puede ser alto. Una característica que favorece la estrategia de los campos potenciales artificiales es la posibilidad de aplicación en robots móviles terrestres y voladores inmersos en entornos cambiantes.

A la hora de implementar cualquiera de los métodos expuestos en este texto se deben tener en cuenta los siguientes aspectos: para el método de campos potenciales artificiales se requiere obtener la distancia mínima entre polígonos o poliedros convexos y no convexos, lo cual no siempre es sencillo o evidente en el campo de la geometría computacional.

En el caso del algoritmo RRT el uso de una librería de software que permita determinar si existe colisión entre poliedros es un requerimiento para encontrar una trayectoria libre de colisiones. Sea cual fuere el método empleado, no se puede asegurar que la ruta encontrada sea la óptima. La estrategia que aborda los campos potenciales artificiales necesita, cuando se amerite, de un subsistema que elimine oscilaciones en la trayectoria generada. Por otra parte, al recurrir a los PRM la ruta obtenida después de un posprocesado generalmente resulta en una trayectoria que

particle (Choset, 2005); on the other hand, if a large number of control points are used, then computational times may increase and make the algorithm extremely slow. It is worth mentioning that APF methods require trial and error tuning of several constant parameters which may influence the type of path generated, thus an ad-hoc method for their selection is necessary. Its use in the field of robotics has shown that the algorithm is highly dependent on the method used to calculate the system's Jacobian matrix and the algorithm used to calculate the shortest distance between a control point and obstacles represented as polygons.

The PRM method involves basic RRT and the RRT_Connect variant. The main strength of such algorithms is that they are conceived so that they can easily be applied to multidimensional search spaces. It is not necessary to have full information about the work space (Choset, 2005). It has also been shown that convergence rate is very small if the number of obstacles is small and that the algorithm can escape trapped configurations (i.e. escape local minima). It must be mentioned that RRT algorithms produce extremely reliable solutions (LaValle, 2001) since the search space is explored methodically. Some drawbacks include constant collision-detection algorithm use where algorithm must be invoked several times per iteration before producing a collision-free path. In many practical cases, the generated path has highly irregular topology, making it difficult to follow. Consequently, post-processing algorithms must be used to smooth-out the trajectory and increased computational times must be expected. There is no guarantee that the generated path is optimal and a second post-processing stage must be implemented to generate acceptable curves using advanced geometrical applications (Angeles, 2007; Kwangjin, 2008a; Kwangjin, 2008b).

Extensive simulation-based comparison has shown that RRT and its variations are more reliable in large dimensionality spaces where a collision-free path is sought. A drawback is that even though a collision-free path is always found, the computational cost may be high. It is worth mentioning that APF methods can be easily applied to mobile robots, including aerial vehicles, even in dynamic environments.

The following points must be born in mind when implementing any of the algorithms described in this paper. The shortest distance between the particle and different convex and non-convex geometries must be found for APF methods, something which is not always simple or intuitive in the field of computational geometry.

Collision-detection libraries must be used for RRT algorithms and the choice of library has a big impact on computational time. Once the path generated by RRT methods has been smoothed using post-processing algorithms, trajectories usually lie very close to the obstacle, a property that is not usually desirable.

pasa muy cerca de los obstáculos, lo cual no es deseable en algunas aplicaciones.

El auge de los PRM y las variantes del algoritmo RRT han desembocado en nuevos desarrollos en diferentes áreas, como en la medicina (Jijie, 2008), concepciones para desembalaje de objetos articulados (Cortés, 2008a), exploración y navegación sobre terrenos accidentados (Cortés, 2008b) y sistemas autónomos de conducción de vehículos (Kuwata, 2008).

Referencias / References

- Angeles, J., *Fundamentals of robotic mechanical systems: theory, methods, and algorithms.*, New York, Berlin: Springer, 2007.
- Brandt, D., Comparison of A* and RRT-Connect Motion Planning Techniques for Self-Reconfiguration Planning., *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Vol., Oct. 2006, pp.892-897.
- Bresina, J.L., Jónsson, A.K., Morris, P.H., Rajan, K., *Ctivity planning for the mars exploration rovers.*, 15th International Conference on Automated Planning and Scheduling, 2005, pp. 40-49.
- Carsten, J., Rankin A., Ferguson D., Stentz A., *Global planning on the Mars Exploration Rovers: Software integration and surface testing.*, J. Field Robot., Vol. 26, No. 4, 2009, pp. 337-357.
- Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S., *Principles of robot motion: theory, algorithms, and implementation.*, Cambridge, Mass.: MIT Press, 2005.
- Cortes, J., Jaillet, L., Simeon, T., *Disassembly Path Planning for Complex Articulated Objects.*, Robotics, IEEE Transactions on, Vol.24, No.2, April 2008a, pp.475- 481.
- Cortes, J., Jaillet, L., Simeon, T., *Transition-based RRT for path planning in continuous cost spaces.*, Intelligent Robots and Systems, 2008, IROS 2008. IEEE/RSJ International Conference on, 22-26 Sept. 2008b, pp.2145- 2150.
- Craig, J., *Introduction to Robotics: Mechanics and Control.*, Addison-Wesley, 1989.
- Funge, J.D., *Artificial intelligence for computer games: an introduction.*, Wellesley, Mass.: Peters, 2004.
- Goldberg, K., *Planar Robot Simulator with Obstacle Avoidance (Configuration Space).*, Dponible en: <http://ford.ieor.berkeley.edu/cspace/>.
- Goldman, A., *Path planning problems and solutions.*, in Book Path planning problems and solutions, Series Path planning problems and solutions, Vol.1, 1994, pp. 105-108.
- Goo Bong, C., Soo Gang, L., Sungmin, K., Byung-Ju, Y., Whee-Kuk, K., Se Min, O., Young Soo, K., Jong, I.I.P., Seong Hoon, O., *A robot-assisted surgery system for spinal fusion.*, in Book A robot-assisted surgery system for spinal fusion, Series A robot-assisted surgery system for spinal fusion, 2005, pp. 3015-3021.
- Hsu, D., *Randomized single-query motion planning in expansive spaces.*, PhD thesis, Department of Computer Science, Stanford University, 2000, Automation Science and Engineering, 2008. CASE 2008, IEEE International Conference on, Aug. 2008, pp.41-46, 23-26.
- Jijie, X., Duindam, V., Alterovitz, R., Goldberg, K., *Motion planning for steerable needles in 3D environments with obstacles using rapidly-exploring Random Trees and backchaining.*, Automation Science and Engineering, 2008, CASE 2008, IEEE International Conference on, Aug. 2008, pp.41-46, 23-26.
- Kavraki, L., Svestka, P., Latombe, J.C., Overmars, M.H., *Probabilistic roadmaps for path planning in high-dimensional configuration spaces.*, Robotics and Automation, IEEE Transactions on, Vol. 12, No. 4, 1996, pp. 566-580.
- Khatib, O., *Real-time obstacle avoidance for manipulators and mobile robots.*, in Book Real-time obstacle avoidance for manipulators and mobile robots, Vol. 2, 1985, pp. 505, 500-505.
- Kirilova, S., Cortes, J., Stefanu, A., Simeon, T., *An NMA-guided path planning approach for computing large-amplitude conformational changes in proteins.*, Proteins, Vol. 70, No. 1, 2008, p. 131.
- Kuffner, J.J., *Goal-Directed Navigation for Animated Characters Using Real-Time Path Planning and Control.*, Lecture notes in computer science., No. 1537, 1998, p. 171.
- Kuffner, J. J., LaValle, S. M., *RRT-connect: An efficient approach to single-query path planning.*, Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, Vol.2, 2000, pp.995-1001.
- Kuwata, Y., Fiore, G.A., Teo, J., Frazzoli, E., How, J.P., *Motion planning for urban driving using RRT.*, Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 22-26 Sept. 2008, pp.1681-1686.
- Kwangjin, Y. and Sukkarieh, S., *3D smooth path planning for a UAV in cluttered natural environments.*, Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, Sept. 2008a, pp.794-800, 22-26.
- Kwangjin, Y., Sukkarieh, S., *Real-time continuous curvature path planning of UAVS in cluttered environments.*, Mechatronics and Its Applications, 2008. ISMA 2008, 5th International Symposium on, May 2008b, pp.1-6, 27-29.
- Latombe, J.-C., *Robot motion planning.*, Boston, MA: Kluwer Academic Publishers, 1991.
- LaValle, S. M., *Planning algorithms*, Cambridge; New York: Cambridge University Press, 2006.
- LaValle, S. M., Kuffner, J. J., *Randomized kinodynamic planning.*, Robotics and Automation, 1999, Proceedings, IEEE International Conference on, Vol.1, 1999, pp.473-479.
- LaValle, S. M., Kuffner, J. J., *Rapidly-exploring random trees: Progress and prospects.*, In B. R. Donald, K. M. Lynch, and D. Rus, editors, *Algorithmic and Computational Robotics: New Directions*, A K Peters, Wellesley, MA, 2001, pp. 293-308.

- Lozano-Pérez, T., Automatic planning of manipulator transfer movements., *IEEE Trans. Systems, Man, and Cybernetics SMC-11*, No. 110, 1981, pp. 681-689-681-689.
- Maio, D., Rizzi, S., CICERO: An Assistant for Planning Visits to a Museum., *Lecture notes in computer science*, No. 978, 1995, pp. 564, 1995.
- Sim, C., Wan Sing, N., Ming Yeong, T., Yong-Chong, L., Tseng Tsai, Y., Image-guided manipulator compliant surgical planning methodology for robotic skull-base surgery., in *Book Image-guided manipulator compliant surgical planning methodology for robotic skull-base surgery*, Series Image guided manipulator compliant surgical planning methodology for robotic skull-base surgery, pp. 26-29, 2001.
- Spong, M., Hutchinson, S., Vidyasagar, M., *Robot modeling and control.*, Hoboken, NJ: John Wiley & Sons, 2006.
- Wenhu, Q., Xiaomei, L., Zhengxu Z., Study on Action Control of Virtual Actors., in *Book Study on Action Control of Virtual Actors*, Series Study on Action Control of Virtual Actors, 2007, pp. 907-912.
- Yershova, A., Jaillet, L., Simeon, T., LaValle, S. M., Dynamic-Domain RRTs: Efficient Exploration by Controlling the Sampling Domain., *Robotics and Automation ICRA 2005*, Proceedings of the 2005 IEEE International Conference, 18-22 April 2005, pp. 3856-3861.