

Actas de las IX Jornadas Argentinas de Robótica 15-17 de noviembre, Córdoba, Argentina

# Planificador jerárquico de trayectorias para sistemas multi-robots

## Hierarchical trajectory planning for multi-robots systems

Presentación: 25/09/2017

Aprobación: 30/09/2020

### M. Estefanía Pereyra

Centro de Investigación en Informática para la Ingeniería, Facultad Regional Córdoba, Universidad Tecnológica Nacional - Córdoba, Argentina  
mepereyra@frc.utn.edu.ar

### David Gaydou

Centro de Investigación en Informática para la Ingeniería, Facultad Regional Córdoba, Universidad Tecnológica Nacional - Córdoba, Argentina  
dgaydou@frc.utn.edu.ar

### R. Gastón Araguás

Centro de Investigación en Informática para la Ingeniería, Facultad Regional Córdoba, Universidad Tecnológica Nacional - Córdoba, Argentina  
garaguas@frc.utn.edu.ar

### Resumen

En este trabajo se presenta un método de planificación jerárquico de trayectorias para guiar un sistema multi-robot desde un punto inicial hasta un punto final dado, considerando la posibilidad de división y reunión de la formación. Un planificador global basado en el algoritmo de Dijkstra computa los caminos óptimos para cada robot de la formación en forma secuencial, mientras que un planificador local determina el conjunto de movimientos necesarios para cada robot. Este conjunto de movimientos permite desplazar robots voladores tipo cuadricópteros por el/los caminos obtenidos por el planificador global, evitando colisiones con los obstáculos del entorno y entre sí. Los vectores de movimiento se obtuvieron mediante simulación, utilizando un controlador no lineal de seguimiento de posición, y un modelo de repulsión basado en función potencial para evitar colisiones entre robots.

Palabras claves: planificador jerárquico, multi-robot, representación por grafos, controlador no lineal.

### Abstract

In this work, a hierarchical trajectory planning method to guide a multi-robot system from an initial point to a given end point is presented, considering the possibility of division and meeting of the training. A global planner based on the Dijkstra algorithm computes sequentially the optimal paths for each robot in the formation, while a local planner determines the set of movements required for each robot. This set of movements makes it possible to move a quadcopter type flying robots along the path(s) obtained by the global planner, avoiding collisions with obstacles in the environment and between them. The motion vectors were obtained by simulation, using a non-

linear position-tracking controller, and a repulsion model based on a potential function to avoid collisions between robots.

Keywords: Hierarchical planning, multi-robot system, graphs representation, non-linear controller.

## I Introducción

Los avances realizados en las últimas décadas en la localización y en la estimación de estado de robots móviles permiten realizar tareas cada vez más autónomas. En esta dirección, la comunidad se ha centrado su interés en tareas más avanzadas como la planificación de caminos/trayectorias en un entorno con obstáculos, especialmente para sistemas multi-robot y robots en formación. El problema consiste en encontrar un conjunto de caminos libre de obstáculos a través de un entorno conocido, desde una configuración inicial a una configuración final. Los caminos se obtienen optimizando algún parámetro específico (como distancias, consumos de energía, tiempo de operación, etc.) y aplicando luego restricciones respecto del tamaño y/o geometría de la formación.

La planificación de caminos y movimientos para sistemas multi-robot pueden agruparse en diferentes categorías. Los algoritmos basados en comportamiento (Balch& Arkin, 1998; Reynolds, 1999; Pereira et al., 2008; Zhong&Peng, 2015) son descentralizados y reactivos, es decir cada robot es controlado individualmente usando sólo la información local de su entorno. El comportamiento del robot se conforma típicamente de varios comportamientos simples (por ejemplo separación, alineamiento, y cohesión en Reynolds (1999)), que describen la acción básica a realizar. Este enfoque es de fácil implementación y puede aplicarse a formaciones grandes, pero presenta cierta dificultad para encontrar una solución en entornos complejos y no garantiza un control preciso de la formación. Existen algunas alternativas para mejorar su desempeño en entornos complejos, como el propuesto por Bai et al. (2009) que se basa en optimización de la formación utilizando partículas, o por Qu et al. (2013) basado en algoritmos genéticos, o el de optimización de colonias de hormigas propuesto por Noormohammadi et al. (2014). Sin embargo, ninguno logra mantener correctamente la geometría de la formación. Los algoritmos centralizados por otro lado consideran a la formación como un cuerpo, y la planificación del camino se realiza en un espacio de configuración compuesto (CCS por sus siglas en inglés) de alta dimensionalidad. Las soluciones de Aronov et al. (1999) son exactas y completas, pero su complejidad es exponencial en la dimensión del CCS, por lo que usualmente se utilizan métodos similares que muestrean el CCS. Por ejemplo, en Clark (2005) se introduce un planificador probabilístico con estrategias de muestreo diseñadas especialmente para sistemas multi-robot, y en Kala (2013) se presenta una generalización del método de rápida exploración aleatoria de árboles (RRT por sus siglas en inglés) para estructuras de grafos. Otra rama de las investigaciones considera la arquitectura líder-seguidor, computando la trayectoria del líder y a partir de esta las trayectorias de los seguidores (Barfoot& Clark, 2004; Chen et al., 2009), o coordinando los movimientos de los robots seguidores sobre caminos planeados previamente (Olimi et al., 2008; Liu et al., 2011). Dentro de este enfoque se encuentran también aquellos que emplean el concepto de campos de potencial artificial (Zhang et al., 2010; Garrido et al., 2011), que luego de encontrar la trayectoria del líder por diversos métodos, computan dinámicamente las trayectorias de los seguidores para mantener la distancia deseada entre robots.

Los métodos centralizados y de solución exacta son computacionalmente muy demandantes y por lo tanto prácticamente inaplicables para formaciones grandes, mientras que los de muestreo aleatorio parecen más promisorios. Por ejemplo, los autores en Saska et al. (2014) presentan una combinación de RRT y optimización de partículas para exploración cooperativa, demostrando una separación de la formación en un escenario simplificado y con un obstáculo simple. Un evasor de obstáculos con reglas agregadas específicamente para separación y unión de la formación se presenta en Ogren (2004), mientras que una extensión del comportamiento de bandadas (de pájaros) presentado en Reynolds (1999) utilizando teoría de juegos es propuesto en Dasgupta& Cheng (2013).

En este trabajo se presenta, como extensión de Pereyra et al. (2017), un método de planificación jerárquico general, que consiste en un planificador de caminos global y un planificador de movimientos local. Mientras que el planificador global busca los caminos topológicos para la formación y genera la dirección de los movimientos primarios, el planificador de movimientos local determina (basado en los caminos encontrados por el

planificador global) el movimiento para cada robot en la formación. Esta combinación evita que el planificador quede atrapado en mínimos locales, y permite computar rápidamente trayectorias posibles.

La contribución principal del trabajo consiste en la incorporación de un planificador de movimiento local al planificador global de caminos presentado en Pereyra et al. (2017), el cual extiende el bien conocido algoritmo de Dijkstra para ser aplicado a sistemas multi-robot, que además considera la posibilidad de separación y unión de la formación. Los resultados de simulación muestran que mediante la integración de ambos planificadores se obtiene el conjunto de movimientos individuales necesarios para trasladar cada robot de una formación desde un punto inicial a un punto final de un entorno con obstáculos.

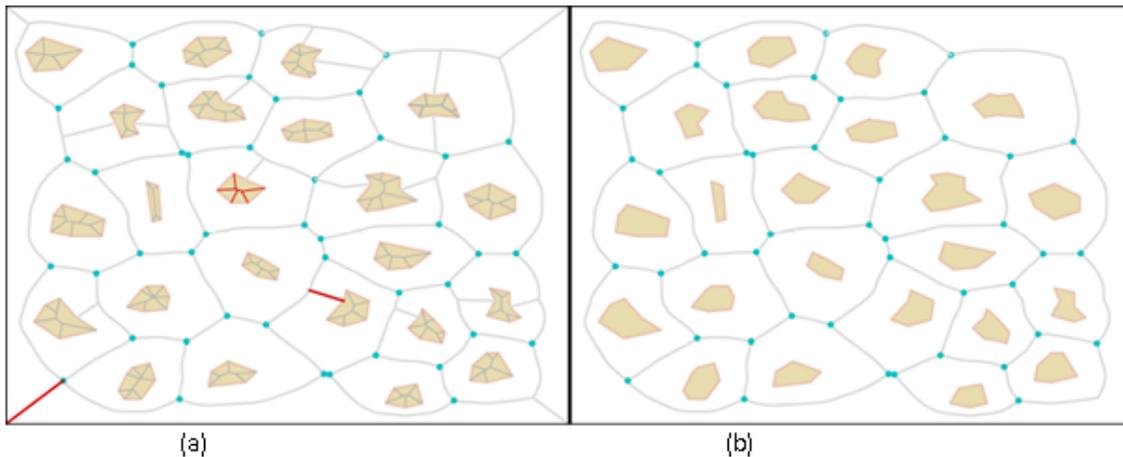


Figura 1. Construcción del grafo. (a) Diagrama Voronoi (en color cian) de un mapa con obstáculos (en marrón) y borde rectangular. La línea roja superpuesta en el diagrama representa una arista no deseada que debe ser eliminada. (b) Diagrama Voronoi donde las aristas innecesarias fueron eliminadas.

## II Planificación de caminos para multi-robots

El problema de planificación de caminos para una flota de robots (MPP, multi-robotpathplanning) se puede interpretar de forma general como la búsqueda de una secuencia continua de configuraciones posibles de la flota desde una configuración inicial hasta una configuración final dada. Una configuración de la flota se considera posible si sus robots no entran en colisión con el entorno que los rodea ni entre ellos. En este trabajo se investiga la problemática de guiar un conjunto de robots desde un punto inicial a un punto destino dentro de un mapa conocido. Los caminos de cada robot se obtienen minimizando un funcional de costo dado. De ser necesario se permiten divisiones del grupo de robots pero con la condición que se reúnan lo antes posible para mantener la formación. Por lo tanto, cada robot que llega a un punto de unión debe esperar a los demás robots que también contienen ese punto en sus caminos. Consecuentemente, el costo de la solución completa queda determinado

---

### Algoritmo 1: Planificador jerárquico multi-robot

---

1. Construcción del grafo conectado  $G(V,E)$
  2. Asignación de costo a cada arista  $e \in E$
  3. Determinar el camino de menor costo en  $G(V,E)$  desde el nodo  $Q_{ini} \in V$  hasta el nodo  $Q_{goal} \in V$  para cada robot  $r_i \in R$
  4. Generar las trayectorias a lo largo de cada camino encontrado
- 

por el costo del peor camino del conjunto de caminos individuales. El MPP debe realizarse evitando las colisiones con los obstáculos del mapa y evitando las colisiones entre robots. Desde el punto de vista del planificador global, una colisión entre robots ocurre cuando dos o más robots comparten parte de sus caminos pero en direcciones

opuestas. Desde el punto de vista del planificador local, ocurre cuando dos o más robots ocupan el mismo espacio en el mismo tiempo. Con el fin de evitar estas colisiones entre robots se imponen una serie de restricciones a los caminos resultantes, que se describen en la sección II-B, y se utiliza un modelo de repulsión por función potencial asociado a cada robot, descrito en la sección II-D. Para evitar colisiones con los obstáculos se realiza el MPP utilizando una representación gráfica del mapa basa en su diagrama de Voronoi.

## II-A. Descripción general

En el Algoritmo 1 se describe en forma general el método de planificación jerárquico propuesto. Dada una representación poligonal del mapa Fig. 1(a), las posiciones inicial y final, y el número de robots que conforman la formación  $R$ , se construye un grafo  $G = (V, E)$  de vértices  $V$  y aristas  $E$  (línea 1). A partir del diagrama de Voronoi del mapa, se obtiene una serie de polígonos equidistantes a los obstáculos y al borde como se muestran en la Fig. 1. Estos polígonos se encuentran conectados formando una red o grafo, el cual será utilizado por los robots para trasladarse dentro del mapa. Luego, las aristas que no conforman un camino libre de colisiones son eliminadas (internas a los obstáculos y externas conectadas a los obstáculos), algunas de ellas se marcan en rojo como ejemplo. El grafo final se muestra en la Fig. 1(b).

Una vez obtenido el grafo del mapa propuesto se evalúan y asignan costos a sus aristas (línea 2). El costo de una arista en particular es un vector  $c = (c_1, c_2, \dots, c_r)$ , donde  $c_r$  corresponde al costo de atravesar con  $r$  robots dicha arista. Este costo puede determinarse a partir del tiempo necesario para atravesar la arista mediante algún algoritmo de planificación de movimientos local, o puede aproximarse mediante una función de costo. La función de costo utilizada en nuestro caso considera la longitud y ancho del corredor (arista), y la cantidad de robots que lo atraviesan. Sobre el grafo con costos ya asignados,  $G(V, E)$ , se realiza la búsqueda de los  $R$  caminos mediante el planificador global (Pereyra et al., 2017) (línea 3), cada uno de estos caminos es una secuencia de nodos de forma  $p = \{v_0=Q_{ini}, v_1, v_2, \dots, v_n=Q_{goal}\}$ . Finalmente, a partir del camino  $p$ , el planificar local genera el conjunto de movimientos individuales para cada robot, respetando las posiciones relativas según las restricciones geométricas dadas por la formación (línea 4).

## II-B Propiedades y restricciones de los caminos

Sea  $G = (V, E)$  el grafo conectado que representa el entorno de trabajo, con vértices  $V = \{v_1, v_2, \dots, v_i\}$  y aristas  $E = \{e_1, e_2, \dots, e_j\}$ , y una flota de robots  $R = \{r_1, r_2, \dots, r_R\}$ , el conjunto de caminos que guiará a los robots por el mapa se denota  $P = \{p_1, p_2, \dots, p_R\}$ , con  $p_i: Z^+ \rightarrow V$ . Además, el camino de un robot individual  $r_i$  es una secuencia de vértices  $p_i = \{v_{i1}, v_{i2}, \dots, v_{ik}\}$  tal que  $(v_{ij}, v_{ij+1})$  es una arista del grafo.

La posibilidad de implementación de un camino implica el cumplimiento de las siguientes propiedades: (1) en un inicio, todos los robots de la formación se encuentran en la posición inicial, esto es  $p_i(0) = Q_{ini}, \forall p_i \in P$ . (2) para cada camino existe un estado  $k_{min} \in Z^+$  tal que  $p_i(k_{min}) = Q_{goal}$ , donde el robot  $r_i$  alcanzó su objetivo utilizando el camino de menor costo. (3) dos caminos de  $P$  no colisionan, es decir, dados dos estados intermedios cualquiera  $m, l \in (0, k_{min})$ , los caminos  $p_i, p_j$  se encuentran en colisión si  $(p_i(m), p_i(m+1)) = (p_j(l+1), p_j(l))$ . (4) sean dos caminos  $p_i, p_j$  y dos estados  $m, l \in (0, k_{min})$ , si  $p_i(m) = p_j(l)$  entonces  $m = l \leftarrow \max(m, l)$ , lo cual significa que el primer robot en alcanzar un vértice compartido espera por el segundo, con el fin de mantener la formación unida tanto como sea posible.

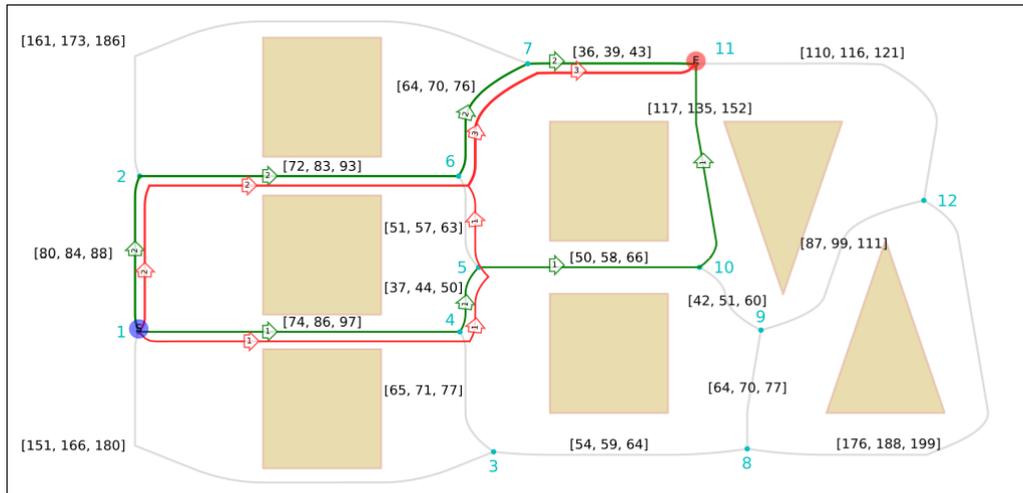


Figura 2. Planificación de caminos global para tres robots en un mapa con obstáculos. Los caminos obtenidos se muestran en color rojo (antes de la optimización) y en color verde (después de la optimización). Las flechas sobre cada arista de los caminos indican la dirección de desplazamiento y la cantidad de robots que se trasladan por ella.

## II-C Planificador global

En esta sección se describe brevemente el algoritmo propuesto en (Pereyra et al., 2017) para la planificación global. Basado en el conocido método de búsqueda de Dijkstra para un robot, el algoritmo computa secuencialmente los caminos para cada robot de un sistema multi-robot en un grafo conectado. El algoritmo estándar de Dijkstra encuentra los caminos óptimos desde un nodo inicial  $Q_{ini}$  al resto de los nodos del grafo  $G$ . Este algoritmo almacena para cada nodo  $v \in G$  el costo mínimo  $C_{vmin}$  para alcanzar dicho nodo, y el nodo predecesor  $v_{prev}$  desde donde el nodo  $v$  es alcanzado. El camino de menor costo para llegar a un nodo  $v$  puede ser fácilmente determinado recorriendo consecutivamente los predecesores comenzando por  $v$ .

Para el caso de sistemas multi-robots, la búsqueda de caminos puede realizarse en base al método de búsqueda estándar de Dijkstra y de forma secuencial, pero considerando posibles divisiones y reuniones de la formación de robots. Para poder extender el método a sistemas multi-robots deben tenerse en cuenta las siguientes consideraciones. Por una parte, se debe generar un grafo conectado  $G$  cuyos costos contemplen la posibilidad de desplazar  $R$  robots por sus aristas, es decir que cada arista tiene asignado un vector de costos. Por otra parte, para cada nodo debe computarse el costo de arribar a él con diferente cantidad de robots. Con el fin de mantener un control sobre la formación se define además un coeficiente  $K$ , llamado coeficiente de control de formación. Este coeficiente modifica el costo de cada arista proporcionalmente con el número de robots, de forma que valores altos del mismo provocan una gran dispersión de la formación, mientras que valores bajos aumentan su concentración.

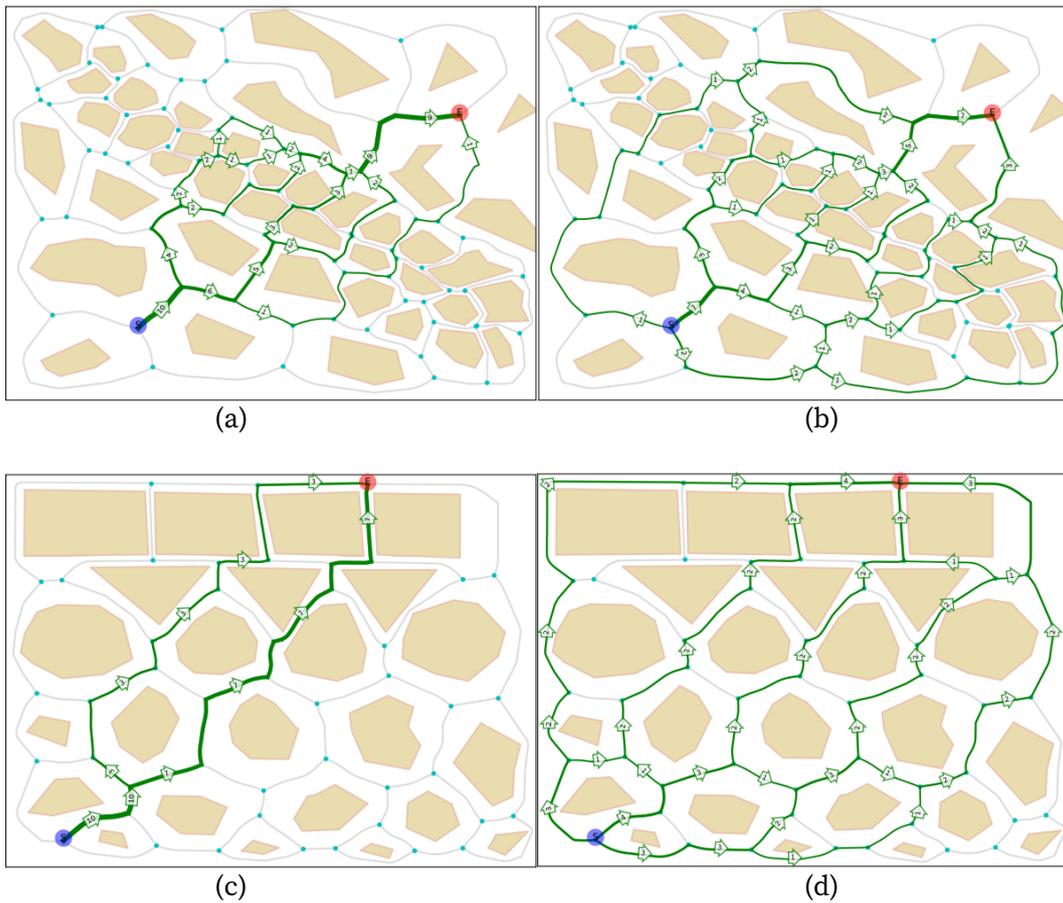


Figura 3. MPP para 10 robots en dos mapas con diferentes complejidades y con dos valores del coeficiente de control de la formación,  $K = 10$  en (a) y (c), y  $K = 100$  en (b) y (d).

Dada una formación de  $R$  robots, un grafo conectado  $G(V,E)$ , y un vector de costos  $c^e = (c_1^e, c_2^e, \dots, c_R^e)$  para cada arista  $e$ , donde  $c_r^e$  es el costo de atravesar  $e$  con  $r \in \{1, R\}$  robots, el algoritmo busca secuencialmente los caminos para cada robot de una formación. De forma similar al estándar de Dijkstra, el algoritmo comienza con el computo del camino óptimo para el primer robot. Luego modifica los costos del grafo considerando el paso de este primer robot, y computa de la misma manera el camino óptimo para el siguiente robot de la formación. El proceso continúa hasta que se hayan computado los caminos  $P_R = \{p_1, p_2, \dots, p_R\}$  para los  $R$  robots de la formación.

Los caminos  $P_R = \{p_1, p_2, \dots, p_R\}$  para  $R$  robots determinados de esta forma secuencial son óptimos si los caminos  $P_{R-1}$  obtenidos para  $R-1$  robots forman parte de la solución óptima para  $R$  robots, es decir  $P_{R-1} \subset P_R$ . Un ejemplo de esto se muestra en la Fig. 2, donde se representan por líneas de color los caminos computados y con una flecha sobre cada arista se indica la cantidad de robots que la atraviesan. Sobre cada arista del grafo se muestra el vector de costo  $c^e = (c_1^e, c_2^e)$  correspondiente. En este ejemplo, los caminos encontrados por el algoritmo desde  $Q_{ini} = 1$  a  $Q_{goal} = 11$  son  $p_1 = p_2 = \{1, 2, 6, 7, 11\}$ , con costo  $C_1 = C_2 = 286$  y  $p_3 = \{1, 4, 5, 6, 7, 11\}$ , con costo  $C_3 = 281$ , representado en color rojo. Sin embargo, los caminos óptimos entre esos nodos para tres robots son  $p_1 = p_2 = \{1, 2, 6, 7, 11\}$ , con costo  $C_1 = C_2 = 276$  y  $p_3 = \{1, 4, 5, 10, 11\}$ , con costo  $C_3 = 278$ , que se muestran en verde. Este problema puede resolverse de forma parcial por medio de un proceso de optimización aplicado en una segunda etapa del algoritmo. Una vez determinado el nuevo camino, todos los caminos previamente determinados son recalculados secuencialmente, comenzando desde el primero. Si uno de los caminos recalculados resulta en un menor costo para la formación, el camino original es reemplazado por este nuevo. En la Fig. 3 se muestran algunos ejemplos de caminos determinados por el planificador global, para diferente cantidad de robots y utilizando diferentes valores del coeficiente  $K$ .

## II-D Planificador local

Para determinar el movimiento de cada robot de la formación según los caminos dados por el planificador global y su interacción con los demás robots se propone un planificador local descentralizado basado en comportamiento, con reglas simples para la generación de puntos de paso que procuren que los robots no colisionen entre sí. Este planificador local computa para cada robot un punto de paso corregido  $wpc$ , a partir del punto de paso dado por el planificador de caminos global  $wp$  y la distancia respecto a los demás robots que compiten en el seguimiento del mismo punto  $wp$ . Para evitar que los robots colisionen entre sí se utiliza el modelo de repulsión por función potencial representado en la Fig. 4(a). Luego, para un instante de muestreo  $k$  y por cada robot  $r_i$  se computa un vector tridimensional de repulsión  $f_{r_i}$ , considerando todos los posibles vectores de repulsión individuales generados por la presencia de los demás robots de la formación. Cada vector de repulsión individual entre el robot  $r_a$  y cualquier robot vecino  $r_b$  tiene la dirección dada por el eje que une sus centros, y su magnitud  $V_{ab}$  se determina según

$$V_{ab} = \text{máx} \left( 0, \frac{d_e - d_{ab}}{d_{ab} - d_i} \right), \quad (1)$$

donde

$$d_{ab} = \sqrt{\left( \frac{x_b - x_a}{\alpha} \right)^2 + \left( \frac{y_b - y_a}{\beta} \right)^2 + \left( \frac{z_b - z_a}{\gamma} \right)^2} \quad (2)$$

Aquí,  $(x_a, y_a, z_a)$  y  $(x_b, y_b, z_b)$  son las posiciones de los robots  $r_a$  y  $r_b$  en el instante discreto de muestreo  $k$ , y los coeficientes  $\alpha, \beta, \gamma > 0$  ajustan el factor de forma del campo repulsivo. Los vectores  $f_{rab}$  y  $f_{rba}$  obtenidos para cada robot se suman a la posición del punto de paso  $wp$  correspondiente al instante de muestreo  $k$ , y se proyectan ortogonalmente sobre el plano  $P$  que es perpendicular a la trayectoria global y contiene dicho punto de paso  $wp$  (Fig. 4(b)). Estas proyecciones determinan los puntos de paso corregidos  $wpc_a$  y  $wpc_b$ , que sirven como nueva referencia para el controlador seguidor de posición del robot correspondiente. Cabe destacar que el mecanismo propuesto no compromete las garantías de estabilidad y convergencia del controlador no lineal, desde que las correcciones se realizan sobre la referencia del mismo.

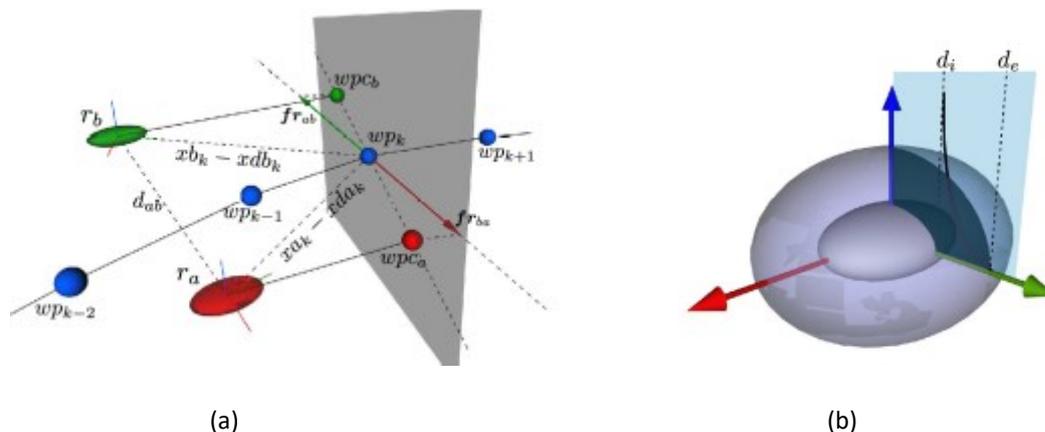


Figura 4. Modelo de repulsión por función potencial (a). Cómputo de los puntos de paso corregidos por efecto de la acción potencial de repulsión (b).

## III Análisis de resultados y simulaciones

Todos los resultados fueron obtenidos bajo las mismas condiciones, utilizando una notebook con un procesador Intel CORE i5 y 4GB de RAM, con Debian GNU/Linux. El planificador global fue implementado en Python 2.7, mientras que el planificador local se implementó en GNU/Octave. Los mapas utilizados para la evaluación fueron gaps, dense, staggeredbrickwall (staggeredbk), potholes, vardensity, var density2 y var density3 de <http://imr.ciirc.cvut.cz/planning/maps.xml>. Sobre un total de  $\approx 3000$  caminos computados por el

planificador global aproximadamente el 94 % de las soluciones fueron óptimas luego de aplicar el proceso de optimización.

### III-A Simulación

El simulador está compuesto por un lazo de tiempo discreto que computa la dinámica en 6 grados de libertad y un controlador no lineal de seguimiento de posición globalmente asintóticamente estable bajo algunas restricciones de orientación inicial y ganancias de lazo (Lee et al., 2010). La Fig. 5 presenta las trayectorias simuladas para dos robots tipo cuadricópteros idénticos siguiendo un tramo de camino obtenido por el planificador global (Fig. 5(a)). Las trayectorias son computadas numéricamente resolviendo las ecuaciones de la dinámica de los dos robots, comandados por sendos controladores no lineales de seguimiento de posición, y sujetos a las condiciones de repulsión antes explicadas. En color azul se muestran los puntos de pasos  $w_p$  generados por el planificador global, en rojo y en verde respectivamente las trayectorias de los robots  $r_a$  y  $r_b$ , y se superponen a la trayectoria los puntos de paso corregidos  $w_{pc}$  generados por el planificador local.

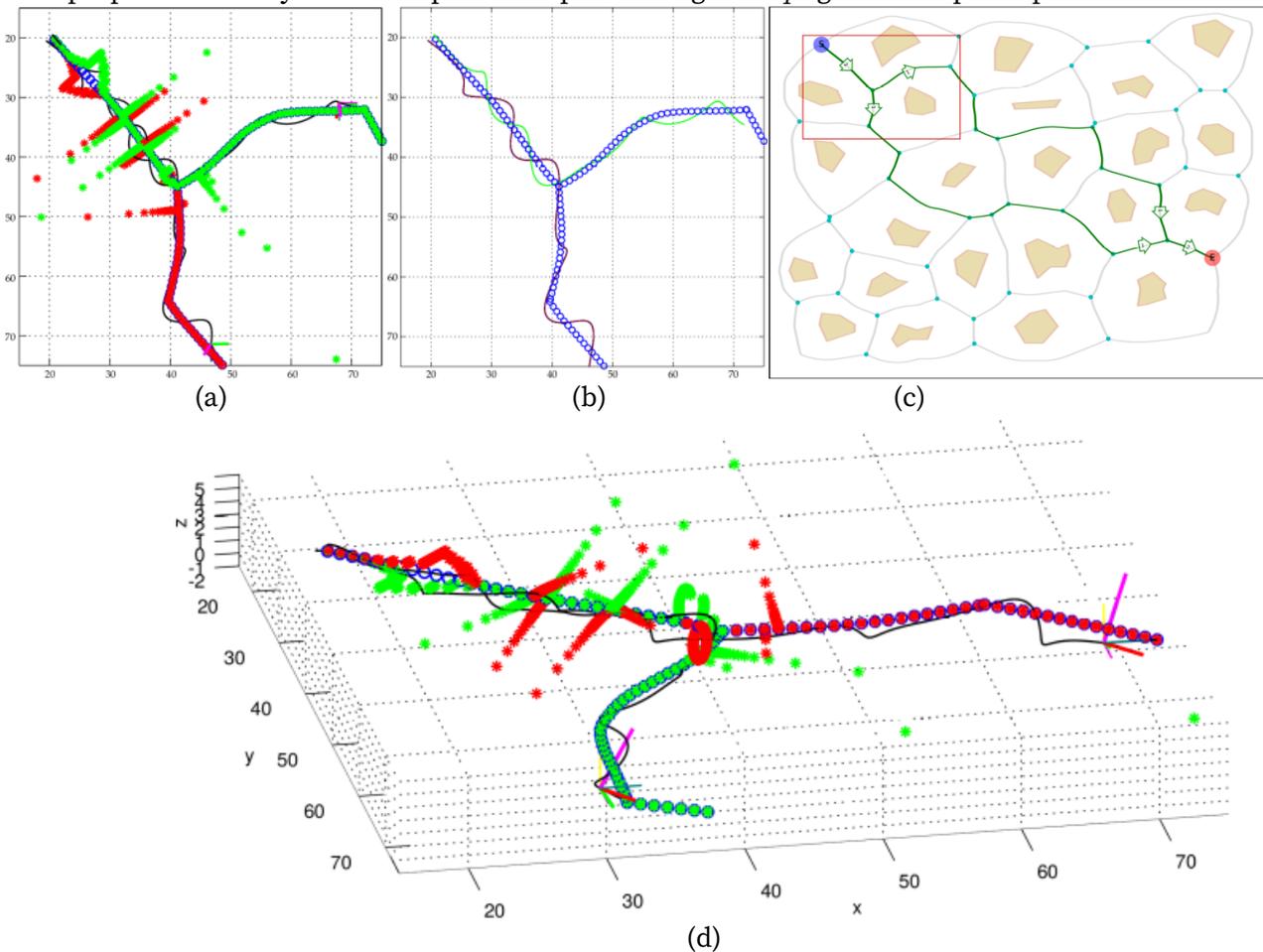


Figura 5. Simulación del camino deseado y trayectoria dinámica de dos robots en formación. En (a) se muestra el mapa completo con los caminos obtenidos por el planificador global, y en (b), (c) y (d) una porción del camino simulado. En (d) se muestran una vista en perspectiva de la simulación, donde puede apreciarse que las trayectorias para cada robot sufren correcciones por efecto del modelo repulsivo en todas las direcciones.

En la Fig. 5(d) se aprecia que si bien los puntos de paso generados por el planificador global tienen coordenada  $z = 1$ , correspondiente a una altura de vuelo constante, la respuesta del sistema actuando bajo los efectos dinámicos propios de los robots sumado al efecto repulsivo potencial introducido en forma artificial, produce trayectorias que fluctúan en tres dimensiones y que mantienen la coherencia de la formación. Esto se visualiza en los puntos de paso corregidos distantes del camino planificado, generados como producto de un acercamiento excesivo entre los robots que interactúan constantemente en el tramo compartido.

## IV Conclusión

En este trabajo se presentó un nuevo método de planificación jerárquico para sistemas multi-robots tipo cuadricópteros. Un planificador global de búsqueda de caminos que permite la separación y reunión de la formación, en conjunto con un planificador local de 3 grados de libertad que evita que los robots colisionen entre sí fueron integrados para su funcionamiento conjunto. Las simulaciones de los cuadricópteros utilizando un controlador no lineal seguidor de posición permitieron determinar las trayectorias completas para dos robots que se desplazan a lo largo de un entorno con obstáculos y sin colisionar entre sí. Como trabajo futuro se planea utilizar esta estructura jerárquica de planificación de caminos y control de trayectorias para comandar cuadricópteros reales de dinámica similar a la utilizada por el simulador.

## Agradecimientos

Este trabajo se desarrolló en el marco del proyecto homologado PID-UTI4534 “Multirrotores Autónomos para Aplicaciones en Ambientes Exteriores”, del Centro de Investigación en Informática para la Ingeniería - CIII, de la Universidad Tecnológica Nacional, Facultad Regional Córdoba.

## Referencias

- T. Balch & R. Arkin (1998). "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939.
- C. W. Reynolds (1999). "Steering behaviors for autonomous characters," in *Game developers conference*, vol. 1999, pp. 763–782.
- G. A. S. Pereira, V. Kumar, & M. F. M. Campos (2008). "Closed loop motion planning of cooperating mobile robots using graph connectivity," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 373–384.
- X. Zhong, & X. Peng (2015). "VCS-based motion planning for distributed mobile robots: collision avoidance and formation," *Soft Computing*.
- C. Bai, H. Duan, C. Li, & Y. Zhang (2009). "Dynamic multi-uavs formation reconfiguration based on hybrid diversity-pso and time optimal control," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 775–779.
- H. Qu, K. Xing, & T. Alexander (2013). "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing*, vol. 120, pp. 509–517.
- A. Noormohammadi Asl, M. B. Menhaj, & A. Sajedin (2014). "Control of leader-follower formation and path planning of mobile robots using asexual reproduction optimization (aro)," *Appl. Soft Comput.*, vol. 14, pp. 563–576.
- B. Aronov, M. de Berg, A. F. van der Stappen, P. Švestka, & J. Vleugels (1999). "Motion planning for multiple robots," *Discrete & Computational Geometry*, vol. 22, no. 4, pp. 505–525.
- C. M. Clark (2005). "Probabilistic Road Map sampling strategies for multirobot motion planning," *Robotics and Autonomous Systems*, vol. 53, no. 3-4, pp. 244–264.
- R. Kala (2013). "Rapidly exploring random graphs: motion planning of multiple mobile robots." *Advanced Robotics*, vol. 27, no. 14, pp. 1113–1122.
- T. Barfoot & C. Clark (2004). "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65–78.
- J. Chen, D. Sun, J. Yang, & H. Chen (2009). "A leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme," *The International Journal of Robotics Research*.
- R. Olmi, C. Secchi, & C. Fantuzzi (2008). "Coordination of multiple agvs in an industrial application," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, 2008*, pp. 1916–1921.
- S. Liu, D. Sun, and C. Zhu (2011). "Coordinated Motion Planning for Multiple Mobile Robots Along Designed Paths With Formation Requirement," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 6, pp. 1021–1031.
- M. Zhang, Y. Shen, Q. Wang, & Y. Wang (2010). "Dynamic artificial potential field based multi-robot formation control," in *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, pp. 1530–1534.
- S. Garrido, L. Moreno, and P. U. Lima (2011). "Robot formation motion planning using Fast Marching," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 675–683.
- M. Saska, J. Chudoba, L. Přeučil, J. Thomas, G. Loianno, A. Třešňák, V. Vonásek, & V. Kumar (2014). "Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pp. 584–595.
- P. Ogren (2004). "Split and join of vehicle formations doing obstacle avoidance," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 2. IEEE, 2004, pp. 1951–1955 Vol.2.
- P. Dasgupta & K. Cheng (2013). "Robust multi-robot team formations using weighted voting games," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 373–387.

M. E. Pereyra, R. G. Araguás, & M. Kulich (2017). “Sequential path planning for a formation of mobile robots with split and merge,” in 4th IEEE Latin American Conference on Computational Intelligence, LA-CCI 2017. IEEE, Nov. 2017.

T. Lee, M. Leoky, & N. H. McClamroch (2010). “Geometric tracking control of a quadrotor UAV on SE(3),” in 49th IEEE Conference on Decision and Control (CDC), pp. 5420–5425.