

## Planificación simultánea de trayectoria y contactos de un robot caminante

Mollá-Santamaría, P.<sup>a,\*</sup>, Peidró, A.<sup>a</sup>, Payá, L.<sup>a</sup>, Ballesta, M.<sup>a</sup>, Reinoso, Ó.<sup>a,b</sup>

<sup>a</sup> Instituto de Investigación de Ingeniería de Elche (I3E), Universidad Miguel Hernández de Elche, Avda. de la Universidad s/n, Edificio Innova, 03202 Elche, Alicante, España.

<sup>b</sup> ValgrAI: Valencian Graduate School and Research Network of Artificial Intelligence, Camí de Vera s/n, Edificio 3Q, 46022 Valencia, España.

**To cite this article:** Mollá-Santamaría, P., Peidró, A., Payá, L., Ballesta, M., Reinoso, O. 2023. Simultaneous trajectory and contact planning of a walking robot. XLIV Jornadas de Automática, 679-684. <https://doi.org/10.17979/spudc.9788497498609.679>

### Resumen

En este artículo estudiamos la planificación simultánea de la trayectoria y de los apoyos de las patas de un robot en terrenos irregulares. Partimos del algoritmo A estrella ( $A^*$ ), que busca el camino más corto entre dos puntos. Primero se modela el terreno con un mallado triangular y se subdividen los triángulos, para tomar como espacio de búsqueda del  $A^*$  los centroides de dichos triángulos. En segundo lugar, con respecto al  $A^*$  original, se ha considerado la estabilidad del robot en cada centroide, de modo que el coste en un centroide sea penalizado si el robot es inestable y nulo si es estable. La estabilidad en cada punto de contacto se determina calculando que la reacción del terreno en ese punto esté contenida en un cono de fricción con una aproximación lineal. Por último, se obtiene la trayectoria, los puntos de contacto de cada pata, así como la postura del robot en cada posición.

*Palabras clave:* Planificación trayectoria, planificación apoyos, estabilidad, contactos no-coplanares, robot caminante.

### Simultaneous trajectory and contact planning of a walking robot

#### Abstract

In this paper we study the simultaneous planning of the trajectory and leg supports of a robot on uneven terrain. We start with the A-star algorithm ( $A^*$ ), which searches for the shortest path between two points. First, the terrain is modelled with a triangular mesh and the triangles are subdivided to take the centroids of these triangles as the search space of the  $A^*$ . Secondly, with respect to the original  $A^*$ , the stability of the robot at each centroid is considered, so that the cost at a centroid is penalised if the robot is unstable and zero if it is stable. The stability at each contact point is determined by calculating that the ground reaction at that point is contained in a linear approximation of the friction cone. Finally, the trajectory, the contact points of each leg, as well as the robot's posture at each position are obtained.

*Keywords:* Path planning, support planning, stability, non-coplanar contacts, walking robot.

### 1. Introducción

En este artículo se estudia la planificación de la trayectoria y de los puntos de contacto de un robot con múltiples patas cuyo objetivo es desplazarse por terrenos irregulares sin volcar ni deslizar, por lo que se deberá tener en cuenta la estabilidad del robot. En el ámbito de los robots caminantes, la planificación de movimientos con múltiples contactos no-

coplanares ha generado en los últimos años un creciente interés.

El método del Punto de Momento Nulo (ZMP, por sus siglas en inglés Zero Moment Point) es comúnmente conocido para analizar la estabilidad de robots caminantes (Vukobratović y Borovac, 2004). El ZMP es el punto en el que los momentos de las fuerzas de reacción del terreno en los apoyos del robot son nulos. El robot será estable si dicho punto está dentro del polígono de soporte que forman los puntos de contacto. No

obstante, este método presenta limitaciones para analizar la estabilidad de robots que caminan por terrenos accidentados, con puntos de contacto no-coplanares.

Hirukawa et al., (2006) propuso un método que aborda las limitaciones del ZMP con el concepto de “Cono de Wrenches de Contacto” o CWC. Según este concepto, en cada punto de contacto se define un cono con ápice en dicho punto, con eje perpendicular al terreno, y con una apertura que depende del coeficiente de fricción. La reacción del terreno en dicho punto debe estar contenida en este cono. Es común que se realice una aproximación lineal de este cono como una pirámide. Combinando todas las fuerzas y momentos de contacto se define el CWC, que es un polítopo 6D tal que el robot será estable (ni volcará, ni deslizará) si la resultante de todas las fuerzas y momentos que actúan sobre él (sin considerar las fuerzas de reacción del terreno) están dentro del CWC.

Numerosos estudios de robots bípedos humanoides (Dai y Tedrake, 2016; Navaneeth, Sudheer, y Joy, 2022) y cuadrúpedos (Aceituno-Cabezas et al., 2017) se basan en una secuencia precalculada de puntos de apoyo en el terreno donde el robot deberá colocar sus patas durante el recorrido, y calculan la trayectoria del centro de masas del robot asegurando que las fuerzas gravitatorias e inerciales están dentro del CWC. Otros trabajos recientes como (Aceituno-Cabezas et al., 2018) resuelven un problema de optimización convexa mixta-entera para planificar simultáneamente la ubicación de los contactos y la locomoción global de un robot cuadrúpedo, lo que permite una solución óptima más completa, para terrenos planos y con pequeñas inclinaciones de  $10^\circ$ .

Otros estudios (Jenelten, Grandia, Farshidian, y Hutter, 2022) emplean una técnica de optimización graduada para determinar en tiempo real la posición y los puntos de apoyo del robot, utilizando sensores montados en el robot para sensar y mapear el terreno. Esto aumenta la robustez del método ante perturbaciones durante la trayectoria, permitiendo al robot realizar tareas complejas como subir escaleras.

Por otro lado, destaca el trabajo de (Orsolino et al., 2018) que proporciona la planificación de la trayectoria del centro de masas asegurando la estabilidad a través de la intersección del CWC con un polítopo que considera los límites de los actuadores del robot.

En el presente artículo se estudia la planificación de trayectorias de un robot con ocho patas mediante el algoritmo A\*, para determinar la trayectoria óptima desde un punto inicial hasta otro final, donde ambos puntos están sobre un terreno irregular que puede emular un terreno natural. Además de determinar la secuencia de posturas en las que centrar el robot, también se determinan los puntos de apoyo de las ocho patas en el terreno a lo largo del camino. El camino óptimo tiene en cuenta la estabilidad, para que el robot no vuelque ni deslice en ningún momento. Se muestran un par de trayectorias de ejemplo, comparando resultados al considerar o ignorar la estabilidad. Además, también se tiene en cuenta la transición entre posturas, lo que permite evitar cambios bruscos en las posturas del robot a medida que avanza en la trayectoria.

Este artículo está organizado como sigue. La Sección 2 describe el robot estudiado. La Sección 3 describe el modelado del terreno por el que se mueve el robot, como un mallado triangular. Seguidamente, en la Sección 4 se describe un test de estabilidad que resuelve un sistema de ecuaciones mediante

Newton evitando construir el CWC. La Sección 5 desarrolla el algoritmo A\* utilizado para planificar la trayectoria. Los resultados se ilustran y comparan mediante ejemplos de simulación en la Sección 6. Por último, la Sección 7 presenta las conclusiones y sugiere mejoras y trabajos futuros.

## 2. Descripción del robot

En esta sección se presenta un robot caminante modular diseñado para moverse por entornos irregulares. Este robot se utiliza para planificar la trayectoria óptima desde una posición inicial hasta una final y para determinar los puntos de contacto en el terreno.

El robot está formado por dos módulos idénticos unidos por una articulación esférica actuada como se ilustra en la Figura 1. Ambos módulos constan de un cuerpo central que tiene la capacidad de desplazarse y orientarse alrededor de los ejes  $x$ ,  $y$ ,  $z$  (6 grados de libertad, gdl) y de cuatro patas que le permiten moverse de manera eficiente. Cada pata del robot tiene 3 gdl ( $q_1, q_2, q_3$ ), donde  $q_1$ , proporciona el movimiento hacia delante y hacia atrás de la pata,  $q_2$  permite la subida y bajada de la pata, y  $q_3$  facilita la flexión o estiramiento de la pata. Por tanto, cada módulo del robot cuenta con 18 gdl que le dotan de gran libertad de movimiento para desplazarse por terrenos accidentados.

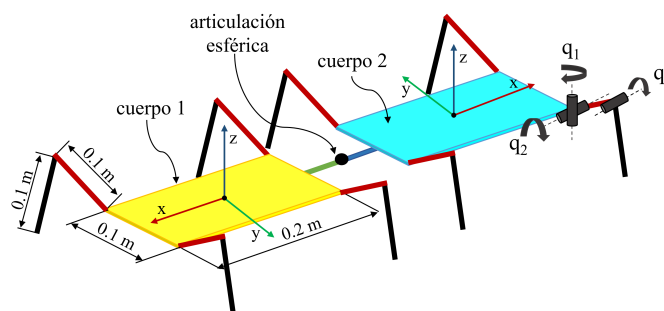


Figura 1: Robot cuadrúpedo modular con estructura alámbrica.

## 3. Modelado y subdivisión del terreno

En esta sección, el objetivo es adaptar el terreno para determinar la trayectoria y sus puntos de contacto del robot con el entorno, considerando que el robot tiene todos sus pies apoyados en el terreno en cada posición. El terreno modelado ( $2\text{m} \times 2\text{m} \times 0.9\text{m}$ ) consiste en varias rampas con distintas inclinaciones que rodean el entorno, lo que requiere que el robot suba las tres rampas para llegar al punto más alto.

Cabe destacar que el terreno de partida está definido por un mallado triangular en formato STL, el cual en primer lugar se ha modelado con un software de diseño 3D, concretamente Autodesk Inventor, y posteriormente se ha exportado en formato de archivo STL gracias a su capacidad para representar objetos sólidos de manera gráfica mediante la aproximación de su superficie con triángulos.

Atendiendo a la planificación de la trayectoria del robot, es necesario partir de una nube de puntos para poder determinar los nodos o puntos que forman parte del camino óptimo desde una posición inicial a una final. Dado que el terreno está formado por triángulos, los puntos de búsqueda para el recorrido son los centroides de los triángulos. Para conseguir una malla con mayor cantidad de nodos y así una planificación más precisa y realista, se dividen los triángulos en otros más

pequeños mediante una subdivisión recursiva del terreno hasta que cada triángulo esté circunscrito en un círculo de radio menor a 0.2m.

El método de subdivisión adoptado consiste en fragmentar cada triángulo uniendo los centros de cada lado, lo que resulta en cuatro nuevos triángulos. Sin embargo, para triángulos alargados o achatados, como el destacado en la Figura 2b, el método puede requerir numerosas divisiones hasta que los triángulos queden circunscritos, generando triángulos muy pequeños en comparación con el resto del terreno. En nuestro caso, como el triángulo achatado está situado en uno de los lados del terreno por donde el robot no caminará, no afecta a la planificación del recorrido. No obstante, para conseguir una división más equiaxial en el terreno, se podría implementar otro método para los triángulos que generan triángulos muy pequeños con el otro método. El procedimiento consistiría en dividir el lado más largo del triángulo en  $N$  segmentos y unirlos con el vértice opuesto.

Inicialmente, el terreno se compone de 42 triángulos, Figura 2a. Tras aplicar el método de subdivisión, el terreno subdividido está formado por un total de 1176 triángulos, Figura 2b. Los centroides de estos nuevos triángulos subdivididos son los nodos que definen el espacio de búsqueda para la planificación de la trayectoria del robot.

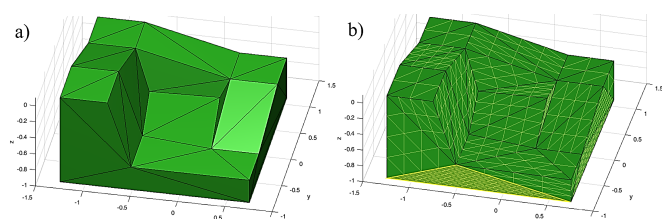


Figura 2: a) Terreno original. b) Terreno subdividido.

Uno de los objetivos del artículo es identificar la localización de los puntos de apoyo del robot en el terreno subdividido. Para ello, se determina el triángulo del terreno más cercano al pie del robot y se proyecta el pie en ese triángulo. Este procedimiento se realiza para cada una de las patas del robot.

Inicialmente se realizó una búsqueda por fuerza bruta, para encontrar el triángulo más cercano buscando entre todos los del entorno, esto fue computacionalmente costoso. Para mejorar el proceso de búsqueda, finalmente nos limitamos a examinar los triángulos que se encuentran a una pequeña distancia del hombro del robot. Para ello, se define un k-d tree con los centroides de los triángulos subdivididos que forman el terreno y, mediante este k-d tree, se determina aquellos centroides que están dentro de una esfera centrada en el hombro de la pata de radio  $R*1.5$ , donde  $R$  corresponde a la longitud de la pata del robot completamente estirada (0.2m).

Una vez se obtiene el triángulo más próximo al pie de cada una de las patas, se calculan las proyecciones sobre el terreno. De esta manera, el robot en cada posición tendrá cada una de sus patas apoyadas en el terreno, lo cual permitirá el estudio de la estabilidad.

#### 4. Análisis de estabilidad

La estabilidad juega un papel fundamental en la planificación del recorrido del robot, ya que si este se desplaza

por un camino donde no es estable se puede enfrentar a situaciones de vuelco o deslizamiento que ponen en riesgo tanto su seguridad como la precisión en la ejecución de tareas.

Para asegurar la estabilidad del robot es esencial realizar un análisis de los puntos de apoyo. El punto de momento cero (ZMP) es el punto en el cual las fuerzas de reacción no generan momentos de vuelco en el plano de contacto del robot con el terreno. En planos horizontales, para que el robot se encuentre en una posición estable basta con que el ZMP esté dentro de la envoltura convexa formada por los puntos de contacto del robot con el terreno. Sin embargo, esta tarea se complica para el análisis de la estabilidad de robots con múltiples contactos no coplanares. En este contexto, Hirukawa et al. (2006) propusieron un método que aborda esta problemática al tener en cuenta las restricciones de fricción entre los puntos de contacto y el terreno. En dicho método, la estabilidad de un punto de contacto  $C_i$  se evalúa mediante la fuerza de contacto  $\mathbf{f}_i$ , la cual se compone de la suma vectorial de la fuerza en la dirección normal en el punto de contacto  $\mathbf{n}_i$  y la fuerza de fricción que depende del coeficiente de fricción del terreno en ese punto ( $\mu_i$ ). Para que un punto de contacto sea considerado estable, es necesario que la fuerza de contacto se encuentre dentro del cono de fricción.

El cono de fricción es una representación geométrica que describe las diferentes combinaciones de fuerza normal y fuerza de fricción que deben cumplir con las restricciones de fricción en un punto de contacto (1).

$$\|\mathbf{n}_i \times \mathbf{f}_i \times \mathbf{n}_i\|_2 \leq \mu_i(\mathbf{f}_i \cdot \mathbf{n}_i) \quad (1)$$

En el artículo de Caron et al. (2017) y en otros numerosos artículos, se utiliza una aproximación lineal del cono de fricción, ya que ofrece una mayor eficiencia computacional en términos de velocidad de cálculo y procesamiento. La aproximación consiste en utilizar los vectores que definen los límites del cono de fricción, formando una pirámide, y se denotan como  $\mathbf{f}_{ij}$ . De manera que la estabilidad del robot requiere que se cumpla la condición (2), que involucra los vectores  $\mathbf{f}_{ij}$  y los momentos de estos vectores, considerando la posición de  $C_i$  en relación con el eje de coordenadas global, que se define como  $\mathbf{p}_{C_i}$ . La expresión matemática de esta condición es la siguiente:

$$\begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau}_O \end{bmatrix} = \sum_{ij} \lambda_{ij} \begin{bmatrix} \mathbf{f}_{ij} \\ \mathbf{p}_{C_i} \times \mathbf{f}_{ij} \end{bmatrix} \quad \lambda_{ij} \geq 0 \quad (2)$$

Donde el subíndice  $i$  recorre los puntos de contacto y  $j$  recorre las aristas empleadas en la aproximación de los conos de fricción mediante pirámides.

Esta expresión describe que la resultante de las fuerzas  $\mathbf{f}$  y momentos  $\boldsymbol{\tau}_O$  (de gravedad, de inercia, etc.) que pueden hacer volcar o deslizar el robot, debe ser una combinación lineal de  $\mathbf{f}_{ij}$  y sus momentos, con coeficientes  $\lambda$  no-negativos. Esto garantiza la estabilidad del robot al asegurar que las reacciones  $\mathbf{f}_i$  están dentro de la pirámide de fricción. En este artículo se han utilizado 10 vectores  $\mathbf{f}_{ij}$  para aproximar el cono.

El método de resolución utilizado consiste en encontrar una solución a partir de valores aleatorios de  $\lambda$  para que se cumpla (2) mediante el método iterativo de Newton. En primer lugar, las desigualdades  $\lambda_{ij} \geq 0$  de (2) se convierten en ecuaciones

equivalentes, introduciendo una variable auxiliar  $t_{ij}$  (que puede adquirir cualquier valor) y viene dada por la expresión:

$$\lambda_{ij} = t_{ij}^2 \quad (3)$$

El sistema de ecuaciones resultante, formado por (2) y (3), que es indeterminado, se resuelve por el método de Newton para sistemas no-lineales. En el caso de que encuentre una solución, esa posición del robot será estable, de lo contrario, se podrá afirmar que no es estable.

## 5. Planificación de trayectoria y puntos de contacto

El presente artículo tiene como objeto planificar simultáneamente la trayectoria y los puntos de apoyo sobre el terreno de un robot con múltiples patas, descrito en la sección 2. Para lograrlo, se utiliza el algoritmo de búsqueda A\*, que busca el camino con el menor coste desde una posición inicial a una final.

Los centroides de los triángulos subdivididos forman la malla de puntos que conforman el espacio de búsqueda del algoritmo. Para ello, con estos puntos se construye un k-d tree utilizado al iniciar el algoritmo para buscar el nodo inicial y final de la malla más cercanos a la posición inicial y a la final de la trayectoria deseada. Posteriormente, también se utiliza para buscar los  $n$  nodos vecinos más cercanos al nodo actual, siendo en este ejemplo  $n$  igual a 10. El uso de k-d trees nos permite realizar tales búsquedas de manera más eficiente que mediante una búsqueda exhaustiva de fuerza bruta.

Para obtener el recorrido óptimo, en este artículo se utiliza el algoritmo A\* para realizar una búsqueda a lo largo de todos los centroides (nodos) de los triángulos que modelan el terreno. Se utiliza el algoritmo A\* tradicional, pero modificando las distancias o costes de cada nodo, teniendo en cuenta la estabilidad del robot y la similitud entre la postura adoptada por el robot entre nodos vecinos. El algoritmo aplicado en este artículo se puede resumir en los siguientes pasos:

1) El algoritmo recorre todos los nodos que forman el mapa del terreno, comenzando por el nodo inicial, y priorizando la búsqueda entre aquellos nodos que presentan un menor coste  $f$  (el coste  $f$  se definirá más abajo).

2) A continuación, se obtienen los 10 nodos vecinos más cercanos al actual (inicialmente, el nodo actual es el inicial).

3) Para cada nodo *vecino*, si no ha sido evaluado o explorado anteriormente, se calculan dos costes  $f$  y  $g$ :

$$\begin{aligned} g(\text{vecino}) &= g(\text{actual}) + d + q + e \\ f(\text{vecino}) &= g(\text{vecino}) + h \end{aligned} \quad (4)$$

Donde:

-  $g(x)$  es el coste real de llegar desde el nodo inicial hasta el nodo “ $x$ ”.

-  $f(\text{vecino})$  es una estimación de lo que costaría llegar desde el nodo inicial hasta el destino, pasando por el nodo “*vecino*”. Este coste se estima mediante una distancia heurística  $h$ , que no debe sobreestimar la distancia real. Por simplicidad, se toma:  $h$  = distancia en línea recta entre *vecino* y *destino*.

-  $d$  es la distancia real entre el nodo actual y el *vecino*.

-  $q$  es la diferencia entre las posturas adoptadas por el robot cuando se sitúa sobre el nodo actual, y sobre el nodo *vecino*. Se tiene en cuenta la diferencia en posiciones y orientaciones de los cuerpos del robot, y de los ángulos girados por sus patas,

entre la postura en el nodo actual y en el *vecino*. El objetivo es minimizar  $q$  para favorecer continuidad entre las posturas adoptadas por el robot a lo largo de los diferentes nodos que va a ir recorriendo para ejecutar su trayectoria.

-  $e$  es una posible penalización por inestabilidad. Para su cálculo, se sitúa el robot sobre el nodo *vecino* y se realiza el test de estabilidad, descrito en la Sección 4. En caso de que el robot sea inestable, se penaliza dicho nodo tomando un valor elevado de  $e$  de  $10^{12}$ . Si es estable, se toma  $e = 0$ .

Hay que tener en cuenta que tanto  $q$  como  $e$  dependen de la orientación que adopte el robot al situarlo encima del nodo *vecino*. Para ello, se tantean cuatro posibles orientaciones separadas entre sí  $45^\circ$ , utilizando la orientación que proporcione un menor valor para  $q + e$ .

En la siguiente subsección se describe cómo se “construye” la postura del robot cuando éste se sitúa sobre cada nodo *vecino*, con el fin de poder evaluar los costes descritos.

### 5.1. Cálculo de estabilidad

El coste de la estabilidad  $e$  toma un valor nulo en caso de que el robot sea estable, y un valor significativamente alto si no es estable. Así se logra que el algoritmo A\* descarte ese nodo como parte del posible recorrido hacia el nodo objetivo.

Para determinar el coste de la estabilidad, en primer lugar, se sitúa la articulación esférica que une ambos módulos del robot,  $C$ , a una distancia  $b$  del nodo o centroide del triángulo que se está evaluando. Los cuerpos del robot se disponen en la dirección del eje principal, denotado como  $ep$ , que es perpendicular a la normal del triángulo del nodo que se está evaluando.

Una vez los centros de los cuerpos están centrados en el eje principal, se calcula los ejes de coordenadas de cada cuerpo como se ilustra en la Figura 3. Los cuerpos son perpendiculares a la normal del triángulo que se encuentra justo debajo del centro del cuerpo. La normal corresponde con el eje  $z$  del cuerpo. El eje  $x$  es la proyección del eje principal en el plano del cuerpo y el eje  $y$  se obtiene a partir del producto vectorial entre el eje  $z$  y el  $x$ .

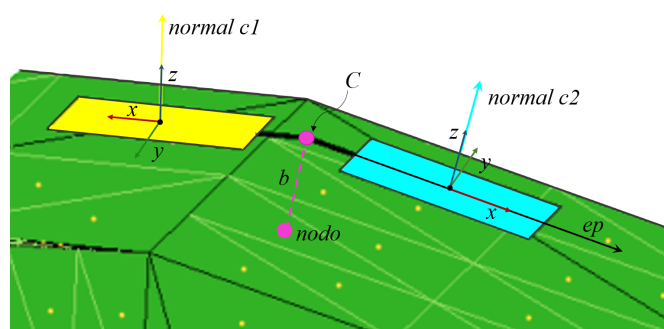


Figura 3: Cuerpos del robot sin patas en la dirección  $ep$ .

Asimismo, cabe destacar que los centros de los cuerpos se definen como el punto de intersección entre una circunferencia de radio  $L$  centrada en  $C$  con cada uno de los ejes  $x$  de cada cuerpo, como se muestra en la Figura 4.



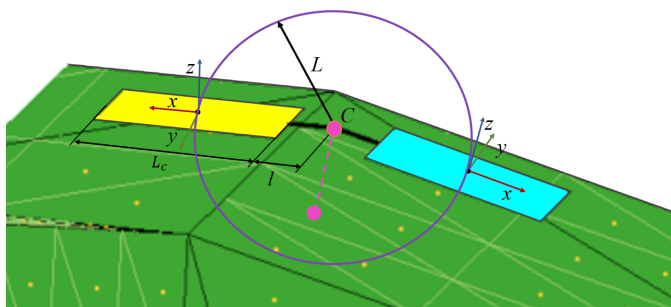


Figura 4: Intersección circunferencia en C con el eje x de cada cuerpo.

De manera que las posiciones de los centros de los cuerpos vienen dadas por la siguiente ecuación:

$$(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2 = L^2$$

$$L = l + \frac{L_c}{2} \tag{5}$$

Donde L es la suma de la longitud de uno de los segmentos de unión l y la mitad de la longitud del cuerpo L<sub>C</sub>, x<sub>o</sub>, y<sub>o</sub>, z<sub>o</sub> corresponde al punto de unión C y las coordenadas x, y, z son las coordenadas del centro de cada cuerpo, obtenidas como:

$$\begin{aligned} x &= x_o + L \cdot ep_x \\ y &= y_o + L \cdot ep_y \\ z &= z_o + L \cdot ep_z \end{aligned} \tag{6}$$

Siendo ep<sub>x</sub>, ep<sub>y</sub>, ep<sub>z</sub> las coordenadas del eje principal proyectado en el plano del cuerpo, es decir, el eje x de cada cuerpo y L el radio de la circunferencia de intersección.

A continuación, se añaden las patas del robot formando un ángulo recto en el aire y se crea una función que determina el punto de apoyo en el terreno más cercano a cada pie del robot en el que debería apoyar los pies, logrando así definir los contactos del robot con el terreno.

Una vez conocidos los puntos de apoyo del robot (C<sub>i</sub>) en ese nodo, se resuelve la posición y orientación de los cuerpos centrales y los ángulos de las patas del robot mediante el método iterativo de Newton. Para ello, se define un sistema de ecuaciones no lineales para garantizar que ambos cuerpos siempre se mantengan unidos y que el pie de cada pata se coloque en los puntos de contacto calculados (Figura 5).

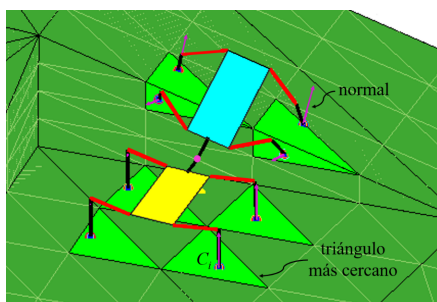


Figura 5: Robot con todas las patas apoyadas sobre el terreno.

Con el robot sobre el terreno se estudia la estabilidad del robot para las distintas orientaciones en cada uno de los nodos examinados por el algoritmo A\*. Finalmente, este algoritmo devuelve los centroides de los triángulos subdivididos que forman parte de la trayectoria, la posición del robot (orientaciones y posiciones de los cuerpos centrales y ángulos de las patas) y los puntos de apoyo en cada uno de los nodos.

## 6. Resultados

Esta sección ilustra mediante ejemplos el algoritmo descrito para evaluar su factibilidad a la hora de planificar la trayectoria del robot y determinar sus posturas y puntos de contacto en cada nodo. Para ello, se definen dos posiciones finales (P<sub>A</sub> y P<sub>B</sub>) para un mismo punto de partida (P<sub>O</sub> = [0, 0, -0.5]). Esto permite comparar los recorridos resultantes considerando o no la influencia de la estabilidad del robot. Para llevar a cabo este estudio se ha tomado un coeficiente de fricción del terreno de 0.4. El algoritmo se ha implementado en Matlab 2015a en Windows, y se ha ejecutado en un procesador Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz con 16 GB RAM.

El primer punto objetivo P<sub>A</sub> [0, 0.75, -0.2] es una posición en la cual el robot debe subir la primera rampa y llegar a una posición intermedia de la segunda rampa. Para evaluar el camino óptimo sin considerar la estabilidad del robot, Figura 6a, el algoritmo requiere 135 segundos. Los puntos rojos representan los centroides que forman parte de la trayectoria óptima donde se situaría la proyección de C. En este camino, el robot va directo a P<sub>A</sub>, subiendo por la parte empinada de la segunda rampa, donde el robot perdería la estabilidad y volcaría o resbalaría. No obstante, cuando se tiene en cuenta la estabilidad del robot, Figura 6b, el robot sube por las rampas que tienen una inclinación que no compromete su estabilidad, aunque esto implique un camino más largo. En este caso, el algoritmo tarda 544 segundos en calcular la trayectoria óptima.

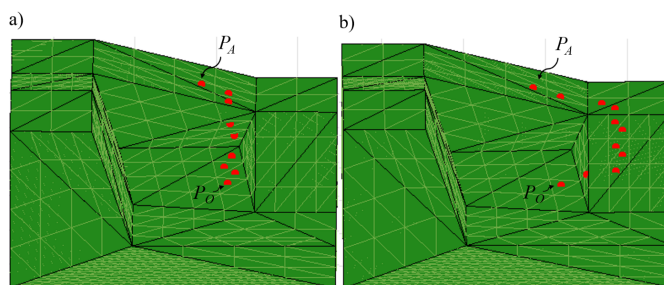


Figura 6: a) Trayectoria óptima del robot hasta P<sub>A</sub> sin estabilidad (puntos rojos). b) Con estabilidad.

El algoritmo A\* obtiene simultáneamente las posturas del robot y los puntos de contacto junto con la trayectoria óptima. La Figura 7 muestra algunas de las posiciones del robot siguiendo la trayectoria óptima obtenida teniendo en cuenta la estabilidad y la orientación del robot, Figura 6b. En estas posiciones se busca mantener una continuidad en las posturas del robot, evitando cambios bruscos entre nodos.

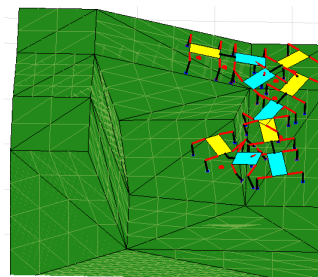


Figura 7: Posiciones y contactos del robot en la trayectoria hasta P<sub>A</sub>.

El segundo punto final P<sub>B</sub> [-1, -0.4, 0.2] está situado en la parte más alta del terreno, Figura 8. De manera similar al punto

$P_A$ , en primer lugar, se calcula la trayectoria del robot mediante el algoritmo A\* sin tener en cuenta la estabilidad, Figura 8a. En este caso, el algoritmo tarda 290 segundos en determinar la trayectoria con menor coste, que es aquella que va directa a  $P_B$  por la rampa con más pendiente en la que el robot no sería estable. Por otro lado, cuando se considera la estabilidad del robot, Figura 8b, el tiempo de respuesta del algoritmo es de 1 hora y 44 minutos. Esto se debe a que el robot no puede subir directamente y debe recorrer más nodos para llegar al punto final.

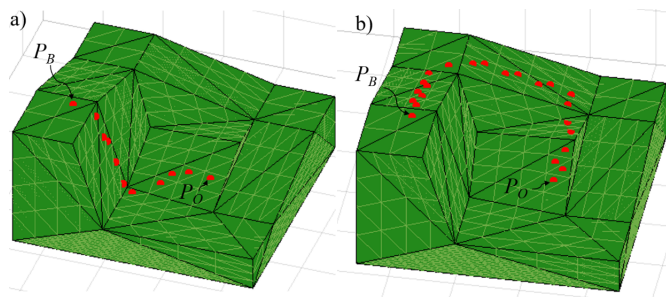


Figura 8: a) Trayectoria óptima del robot hasta  $P_B$  sin estabilidad (puntos rojos). b) Con estabilidad.

La Figura 9 muestra algunas de las posiciones del robot, posturas y puntos de contacto, para alcanzar el punto  $P_B$  recorriendo la trayectoria óptima del robot obtenida por el algoritmo A\* considerando la estabilidad y la continuidad de las posturas. Como muestran todos los ejemplos de esta sección, el robot adopta posturas razonables y factibles a pesar de no haber considerado límites en los giros de las patas.

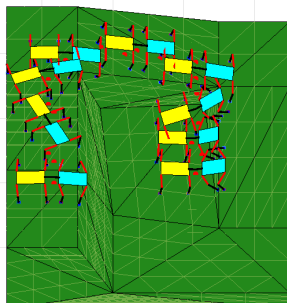


Figura 9: Posiciones y contactos del robot en la trayectoria hasta  $P_B$ .

## 7. Conclusiones

En este artículo presenta un método para planificar las trayectorias de un robot con múltiples patas en un entorno con desniveles. Por un lado, se define un terreno con un mallado triangular para determinar los puntos de contacto de las patas y establecer la malla de nodos para planificar la trayectoria. Por otro lado, se utiliza el algoritmo A\* que busca el recorrido óptimo, desde una posición inicial hasta la posición objetivo, en el que el robot es estable a lo largo del camino y mantiene una postura lo más realista y parecida a la anterior. Además, se identifica en cada posición del robot en el camino los puntos de contacto y posturas.

No obstante, dado que en la realidad el robot debería ir calculando la trayectoria siguiente mientras se ejecuta la trayectoria actual, en el futuro sería necesario mejorar los tiempos de búsqueda de los puntos óptimos. Una posible

mejora podría ser utilizar un árbol aleatorio de exploración rápida (RRT, por sus siglas en inglés *Rapidly Exploring Random Tree*) en lugar del algoritmo A\*. El algoritmo A\* realiza una búsqueda exhaustiva capaz de encontrar la trayectoria óptima absoluta, pero es muy costoso computacionalmente. A cambio, el algoritmo RRT es más rápido, pero no tiene por qué encontrar solución ni la mejor absoluta, sino que encontrará soluciones subóptimas.

Como otra línea de investigación futura, dado que ya se conoce la secuencia de posturas a adoptar por el robot a lo largo del terreno, se tendrá que resolver la planificación de movimiento entre posturas sucesivas. Y finalmente, se implementará sobre un prototipo real.

## Agradecimientos

Este trabajo forma parte del Programa INVESTIGO 2022 (número de expediente: INVEST/2022/432) financiado por la Conselleria de Innovación, Universidades, Investigación y Sociedad Digital, y por la Unión Europea (Next Generation EU). También es parte del proyecto CIGE/2021/177, financiado por la Conselleria d'Innovació, Universitats, Ciència i Societat Digital, y del proyecto TED2021-130901B-I00, financiado por MCIN/AEI/10.13039/501100011033 y por la Unión Europea "NextGenerationEU"/PRTR".

## Referencias

- Aceituno-Cabezas, B., Dai, H., Cappelletto, J., Grieco, J. C., Fernández-López, G., 2017. A mixed-integer convex optimization framework for robust multilegged robot locomotion planning over challenging terrain. En: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4467–4472. DOI: 10.1109/IROS.2017.8206313
- Aceituno-Cabezas, B., Mastalli, C., Dai, H., Focchi, M., Radulescu, A., Caldwell, D. G., Cappelletto, J., Grieco, J. C., Fernández-López, G., Semini, C., 2018. Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization. *IEEE Robotics and Automation Letters* 3(3), 2531–2538. DOI: 10.1109/LRA.2017.2779821
- Caron, S., Pham, Q. C., Nakamura, Y., 2017. ZMP support areas for multicontact mobility under frictional constraints. *IEEE Transactions on Robotics* 33(1), 67–80. DOI: 10.1109/TRO.2016.2623338
- Dai, H., Tedrake, R., 2016. Planning robust walking motion on uneven terrain via convex optimization. En: 2016 IEEE-RAS 16th International Conference on Humanoid Robots, pp. 579–586. DOI: 10.1109/HUMANOIDS.2016.7803333
- Hirukawa, H., Hattori, S., Harada, K., Kajita, S., Kaneko, K., Kanehiro, F., Fujiwara, K., Morisawa, M., 2006. A universal stability criterion of the foot contact of legged robots-Adios ZMP. En: 2006 IEEE International Conference on Robotics and Automation, pp. 1976–1983. DOI: 10.1109/ROBOT.2006.1641995
- Jenelten, F., Grandia, R., Farshidian, F., Hutter, M., 2022. TAMOLS: Terrain-aware motion optimization for legged systems. *IEEE Transactions on Robotics* 38(6), 3395–3413. DOI: 10.1109/TRO.2022.3186804
- Navaneeth, M. G., Sudheer, A. P., Joy, M. L., 2022. Contact wrench cone-based stable gait generation and contact slip estimation of a 12-DOF biped robot. *Arabian Journal for Science and Engineering* 47, 15947–15971. DOI: 10.1007/s13369-022-06763-z
- Orsolino, R., Focchi, M., Mastalli, C., Dai, H., Caldwell, D. G., Semini, C., 2018. Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots. *IEEE Robotics and Automation Letters* 3(4), 3363–3370. DOI: 10.1109/LRA.2018.2836441
- Vukobratović, M., Borovac, B., 2004. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics* 1(1), 157–173. DOI: 10.1142/S0219843604000083