**MÉTODO GENERALIZADO PARA LA PLANIFICACIÓN DE LOCOMOCIÓN EN ROBOTS CON PATAS EN AMBIENTES IRREGULARES**Bernardo Aceituno C.^{1*}, José Cappelletto y Juan C. Grieco¹¹ Grupo de Investigación y Desarrollo de Mecatrónica, Universidad Simón Bolívar.[*b.a.ve@ieee.org](mailto:b.a.ve@ieee.org)**RESUMEN**

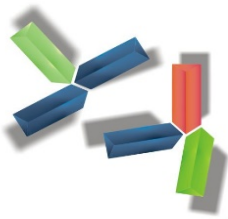
La mayoría de los métodos de planificación de locomoción se basan en técnicas tales como búsqueda en grafos, mallas o programación dinámica. Éstos suelen limitarse a un número de patas en específico, y son altamente dependientes de la complejidad del ambiente. El presente trabajo incorpora el uso de técnicas de optimización de costos y restricciones para adaptar la locomoción a la geometría del robot y al ambiente. Para ello se desarrolló un algoritmo de optimización convexa con programación cuadrática de enteros mixtos (MIQP), que permite calcular de forma rápida y robusta movimientos en ambientes irregulares. Se efectuaron simulaciones dinámicas sobre una plataforma hexápoda de 3 grados de libertad por pata. Este trabajo representa a conocimiento de los autores la primera implementación de un método de planificación de locomoción generalizado basado en optimización convexa.

Palabra Clave: Planificación de locomoción, programación de enteros mixtos, optimización convexa, restricciones geométricas, ambientes irregulares.

ABSTRACT

The majority of locomotion planning methods are based on techniques such as graph search, meshes or dynamic programming. These are often limited by a specific number of legs and are highly dependent on the complexity of the environment. This work introduces the use of constrained objective optimization techniques to adapt the locomotion to the geometry of the robot and the environment. To do this, a convex optimization algorithm with mixed integer quadratic programming (MIQP) was developed, which allows computing movements on a fast and robust way for rough environments. This approach was tested through dynamic simulations on a hexapod platform with 3 degrees of freedom (DOF) by leg. This work represents, to the knowledge of the authors, the first implementation of a generalized method for locomotion planning based on convex optimization.

Keywords: Locomotion planning mixed integer programming, convex optimization, geometric constraints, and rough environments.



INTRODUCCIÓN

La planificación de locomoción en robots con patas basada en optimización convexa fue introducida por Deits et al [1], donde su uso permitía planificar movimientos en ambientes irregulares para robots bípedos de forma rápida. Dicho método se basa en programación cuadrática de restricciones cuadráticas de enteros mixtos (MIQCQP), para representar las regiones libres, las rotaciones de cada pie y el alcance de cada pata, y fue implementado exitosamente por el equipo MIT en el Desafío de Robótica de DARPA [2]. Sin embargo, el método anterior se limita a caminados de dos patas y no permite la introducción de geometrías de mayor complejidad o cantidad de patas. El auge de los robots con patas requiere extender estos métodos mas allá de los robots humanoides, en especial desde la aparición de robots cuadrúpedos y hexápodos altamente articulados [3][4].

Otros métodos de uso común son los basados en algoritmos de búsqueda como A*, A* de Reparación en Cualquier Momento (ARA*), Árboles Aleatorios de Exploración Rápida (RRT), Programación Dinámica (DP), entre otros [5], tras los cuales se planifica el movimiento del centro de masa y el Zero-Moment Point del robot [6] para generar movimientos del cuerpo completo. Sin embargo, los métodos anteriores submuestran el espacio de trabajo, no todos aseguran trayectorias óptimas y su complejidad escala drásticamente con la complejidad del ambiente.

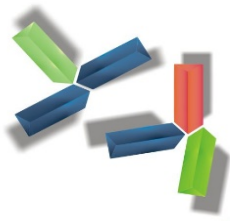
Este trabajo presenta una generalización del método de planificación de locomoción basado en optimización convexa, para robots con un número arbitrario de patas en ambientes complejo. Adicionalmente este método es implementado en un robot hexápodo BH3-R de LynxMotion para validar su funcionamiento al generar planes de movimiento de forma rápida y óptima.

METODOLOGÍA

El algoritmo diseñado se puede dividir en dos etapas principales: una etapa de planificación de pisadas y una etapa de planificación de movimientos. También se incluye una etapa de descripción de regiones libres. Estas etapas son explicadas a detalle a continuación:

Descripción de regiones libres:

Antes de planificar caminado es necesario describir cada región libre como un polígono convexo proyectado en el plano del piso (en este caso el plano XY). Para esto se hace uso del algoritmo de Inflación Iterativa de Regiones con programación Semi-definida (IRIS) [7] el cual genera el mayor polígono convexo que describe cada región libre mediante hiper-planos separadores, luego este polígono se proyecta en el plano cartesiano de la superficie y se procede a la planificación.



Planificación de pisadas:

Para representar cada pisada se usa una descripción de la forma:

$$f = (f_x \ f_y \ f_z \ \theta)$$

Donde f_x , f_y y f_z son las componentes de la posición de la pisada en el espacio y θ la orientación. El algoritmo de planificación de pisadas se representa como un programa cuadrático de restricciones cuadráticas de enteros mixtos (MIQCQP) similar a [1], el cual recibe como entradas las regiones libres, el número máximo de pisadas y la meta, y retorna el plan óptimo para llegar a la configuración deseada. Se toman como variables de decisión:

- Posición de cada pisada: un vector en \mathbb{R}^4 que describe cada pisada.
- Posición relativa aproximada del centro de presión: un vector en \mathbb{R}^2 que describe una aproximación lineal del centro de masa con respecto a cada pisada.
- Asignación de regiones: una matriz binaria que para R regiones y N pisadas será $\{0,1\}^{R \times N}$ asigna una región libre a cada pisada y es tomada como una variable entera.
- Recorte de paso: asignada a los pasos restantes del plan una vez llegado a la meta, es un vector t binario en $\{1,0\}^N$.

Este programa entonces se describe la siguiente forma:

A. Restricciones de caminado:

Se debe asegurar que cada pisada se encuentre en una región libre, para ello se agrega la siguiente restricción sobre la matriz de regiones para cada pisada i :

$$\sum_{j=1}^R H(i, j) = 1$$

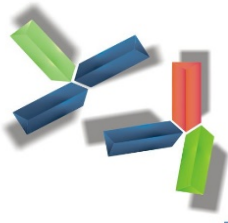
de forma que se cumple que si la región libre j se describe como $\{x \mid A_j x < b_j\}$, entonces se restringe que:

$$H(i, j) = 1 \Rightarrow A_j f_i \leq b_j$$

donde el operador \Rightarrow se representa como una restricción lineal en los software optimización. Luego se debe asegurar que cada pisada se encuentre en el espacio de trabajo de la pisada anterior, para ello se aproxima el espacio de trabajo como una región circular centrada en la pisada anterior (o en $i - n_{patas}$) con un radio r_{max} , y se restringe que la pisada i se encuentre dentro de esta región:

$$(f_{ix} - f_{(i-n_{patas})x})^2 + (f_{iy} - f_{(i-n_{patas})y})^2 \leq r_{max}^2, \quad i = 1, 2, \dots, N$$

Lo que se representa como una restricción cuadrática. Por otro lado, se debe asegurar que cada pisada i no se encuentra a una altura ni a una orientación inalcanzable desde la pisada $i - n_{patas}$, para ello se agrega la siguiente restricción lineal:



$$\Delta f_{min} \leq f_{i+1} - f_i \leq \Delta f_{max}$$

Finalmente, al igual que en [1] se agrega una restricción para que la pisadas recortadas sean iguales a las pisadas iniciales, lo cual es representado como una restricción lineal en el recorte en la pisada i respecto a la primera pisada de la pata asignada a esta:

$$t_i = 1 \Rightarrow f_i = f_{inicial_{pata_i}}$$

lo que permite reducir el numero de pisadas a ejecutar.

B. Restricciones geométricas:

Para lograr describir la geometría del robot, y obtener planes congruentes se agregan dos restricciones: la primera restringe la posición de cada pisada respecto a sus pisadas vecinas, esto se representara como la siguiente restricción cuadrática:

$$(f_{ix} - f_{i\pm 1x})^2 + (f_{iy} - f_{i\pm 1y})^2 \leq D_{max}^2$$

Donde D_{max}^2 es el cuadrado de la distancia máxima entre una pata y su vecina, esta desigualdad se representa fácilmente como una restricción cuadrática. Por otro lado, también se restringe la posición de cada pisada nominal respecto al centro de presión aproximado del robot. Para ello se define el centro de presión aproximado ($x_{app} \in \mathbb{R}^2$) con respecto a la pisada i como:

$$x_{app|f_i} = x_{app} + L_i \begin{pmatrix} s_i \\ c_i \end{pmatrix}$$

Donde L_i es la distancia nominal entre la pata y el centro de presión aproximado, s_i abrevia a $\sin(\theta_i)$ y c_i a $\cos(\theta_i)$ de allí se restringe que cada pisada se encuentre ubicada en una región apoyo valida (R_i), descrita como círculo centrado en la posición de la pisada nominal (ver figura 1), descrito como:

$$f_i \in R_i \Rightarrow (f_{ix} - x_{appix} - L_i c_i)^2 + (f_{iy} - x_{appiy} - L_i s_i)^2 \leq R_{pie}^2$$

Donde R_{pie}^2 es una constante definida por el usuario.

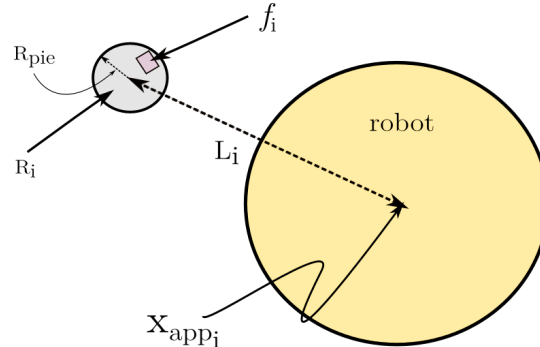
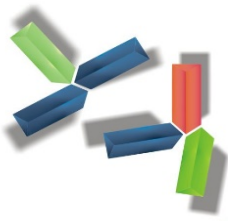


Figura 1. Restricción geométrica sobre la pisada i y el centro de presión aproximado



En el caso de las rotaciones, la presencia de una función trigonométrica en las restricciones hace que se pierda la convexidad por lo que se debe aproximar de forma lineal. Esto se puede lograr mediante restricciones lineales enteras (método aplicado en [1]) o manteniendo fija la rotación en el plan; ambos métodos se implementaron obteniendo resultados mixtos. Sin embargo, la aproximación lineal representa una mejor alternativa para robots con geometrías rectangulares, mientras que para robots de simetría circular aumenta de forma significativa la complejidad del problema.

C. Función objetivo:

Se proponen dos objetivos principales a minimizar por el programa:

1. Diferencia entre la posición final y la configuración deseada.
2. Numero de pisadas

Por ello dadas las n_{patas} pisadas finales $f_{gN-n_{patas}}, \dots, f_{gN-1}, f_{gN}$ para un plan de N pasos. se debe minimizar la distancia entre las N pisadas finales del plan y las n_{patas} pisadas finales deseadas, entonces se tendrá como primer objetivo un termino proporcional a la distancia cuadrática entre cada pisada final y su valor deseado:

$$\sum_{i=N-n_{patas}}^N (f_{g_i} - f_i)^T Q_g (f_{g_i} - f_i)$$

donde $Q_g \in \mathbb{R}^{4 \times 4}$ es un peso cuadrático. Por otro lado se tiene que la cantidad de pasos recortados estará dada por la suma de las componentes de $t \in \mathbb{R}^N$. Entonces la cantidad de pasos será $N - \sum_{i=1}^N t_i$ y como se busca minimizar el numero de pasos entonces se agrega el objetivo:

$$\sum_{i=1}^N q_t t_i$$

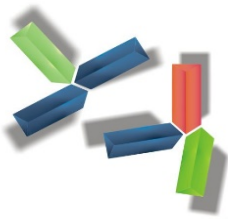
donde $q_t \in \mathbb{R}$ es un peso asignado a cada recorte, de esta manera se asegura que todas las pisadas posteriores al final del plan se solapen con la configuración inicial y de allí puedan ser recortadas.

Planificación de movimientos:

Para planificar los movimientos de cada pata y el centro de masa del robot se hace uso de un planificador de “Zero-Moment-Point”. Este está basado en un regulador cuadrático lineal variante en el tiempo, para entonces así aplicar cinemática inversa usando un modelo lineal de péndulo invertido. Para mas detalles en el funcionamiento se recomienda referirse a [8].

RESULTADOS Y DISCUSIÓN

Entorno de trabajo:



Para la implementación de los algoritmos de planificación se hizo uso del paquete de planificación y control “Drake” [9] el cual funciona bajo el software de simulación MATLAB R2015a. Para resolver el programa se hizo uso del software de optimización de gran escala “Gurobi” [10], otros software fueron considerados sin embargo Gurobi presento el mejor desempeño y mayor rapidez para detectar problemas inviables.

Para verificar el funcionamiento del método propuesto se hizo uso de un robot hexápodo circular BH3-R [4]. Para éste, las restricciones geométricas se veían simplificadas puesto que la distancia L_i se mantiene constante en cada pisada debido a la simetría radial del robot.

Ambientes planos:

Para analizar el funcionamiento en ambientes sencillos sin obstáculos ni restricciones, se inicializa un plan de 67 pisadas sobre un terreno plano simple con rotaciones fijas el cual es resuelto en un tiempo de 647 ms. Las figura 2 muestra los resultados obtenidos, la posición de cada pisada y los movimientos a realizar por cada pata.

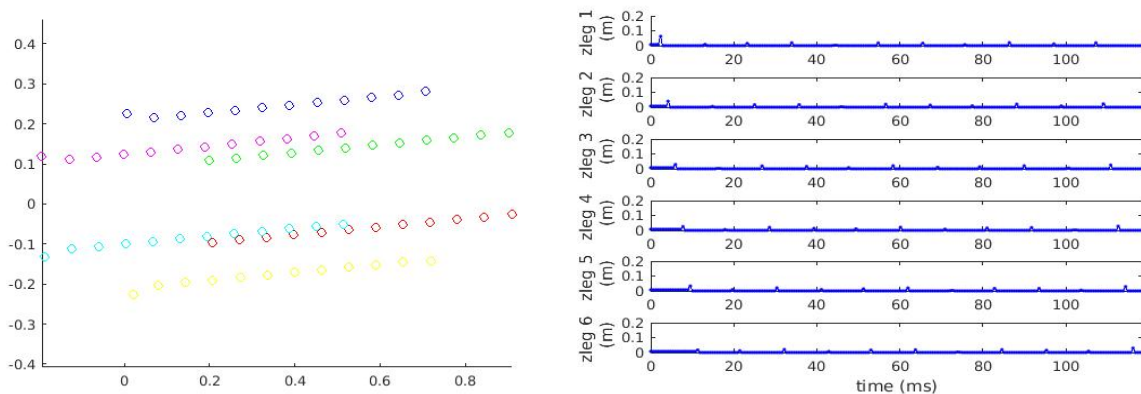


Figura 2. En un terreno plano Izq: Plan de pisadas generado Der: Movimiento para cada pata.

Por otro lado, la figura 3 muestra un dibujo en 3D de este plan a ser ejecutado en el ambiente de interés.

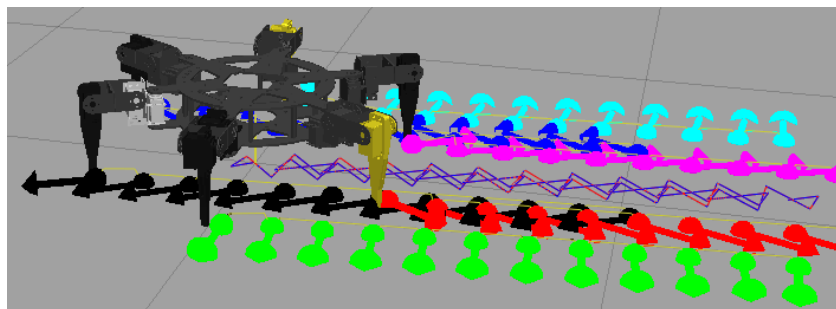
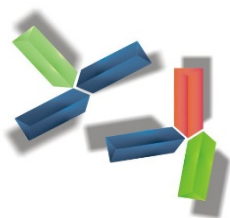


Figura 3. Plan de locomoción generado en un terreno plano.



Ambientes irregulares:

Para verificar el funcionamiento de la implementación en un ambiente mas irregular se hace uso de dos bloques inclinados sobre los cuales se deba planificar el caminado, de nuevo las regiones libre son descritas como regiones convexas en la superficie de cada bloque. El plan es resuelto de forma optima para 67 pisadas en 4.67 s y se obtiene el plan mostrado en la figura 4.

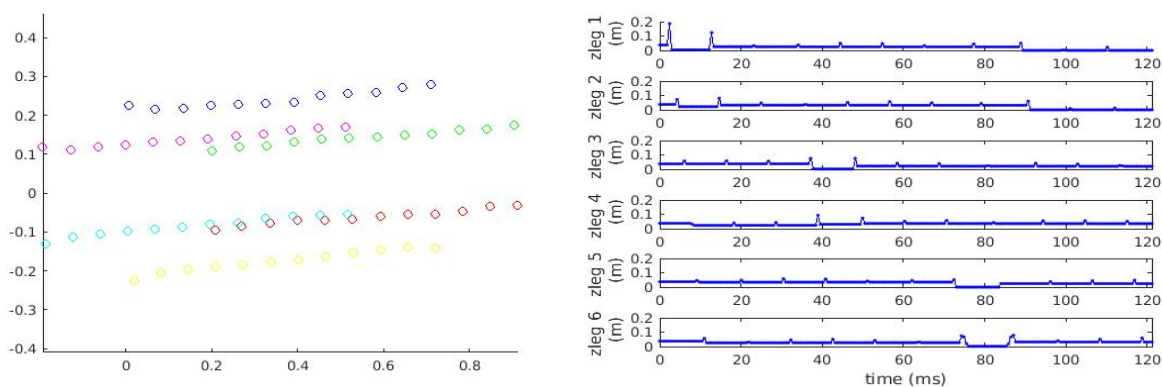


Figura 4. En un terreno irregular Izq: Plan de pisadas generado Der: Movimiento para cada pata.

La figura 5 muestra este plan dibujado en 3D (nótese como las pisadas se ajustan a la geometría del terreno).

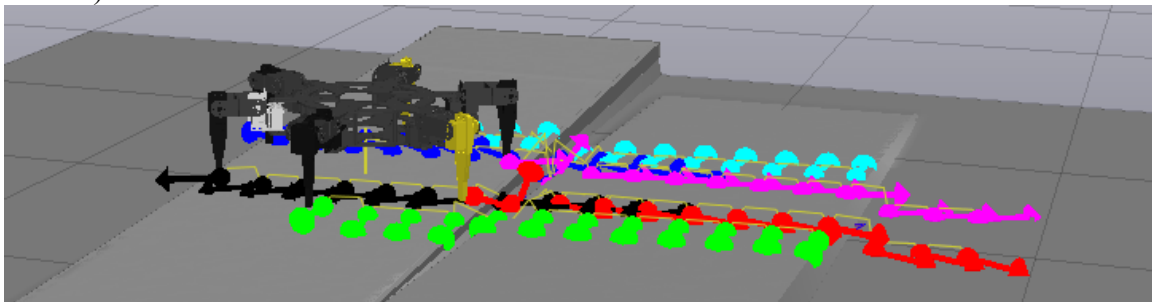
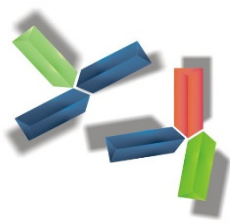


Figura 5. Plan de locomoción generado en un terreno irregular.

CONCLUSIONES

Este trabajo presenta un método generalizado para planificación de locomoción en robots con patas de forma optima y eficiente. El mismo se basa en optimización convexa de enteros mixtos y planificación del ZMP para generar movimientos en ambientes irregulares. También presentamos el concepto de restricciones geométricas el que permite generar planes de pisadas adaptados a cada robot, de esta forma generalizando el método para un robot con patas cualquiera.



En el futuro es de interés asegurar robustez en el método propuesto mediante métodos planificación de movimiento de centro de masa alternos al método de ZMP, por ejemplo por planificación de Cono de Llave de Contactos (Contac Wrench Cone) lo que asegura estabilidad en el plan a ejecutar mediante restricciones sobre momento angular centroidal del robot [11].

Finalmente los autores consideran que es importante validar experimentalmente el funcionamiento del planificador en una plataforma física, por esto se debe desarrollar un controlador para el robot y un sistema de percepción que permita describir las regiones libres en el ambiente de forma robusta.

REFERENCIAS

- [1] Robin Deits and Russ Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization", In *Proceedings of the 2014 IEEE/RAS International Conference on Humanoid Robotics*, 2014.
- [2] Scott Kuindersma et al, "Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot", *Autonomous Robots*, 40(3):429–455, 2016.
- [3] Maureen Rojas, Novel Certad, Jose Cappelletto and Juan C. Grieco, "Foothold Planning and Gait Generation for a Hexapod Robot Transversing Terrains with Forbidden Zones", In *12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, 2015.
- [4] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Hierarchical planning of dynamic movements without scheduled contact sequences", in *IEEE International Conference on Robotics and Automation*, 2016.
- [5] Steven M. LaValle, "Planning Algorithms", Cambridge University Press, 2006.
- [6] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point", In *IEEE International Conference on Robotics and Automation*, 2003.
- [7] Robin Deits and Russ Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming", On *Workshop on the Algorithmic Foundations of Robotics*, 2014.
- [8] Russ Tedrake, Scott Kuindersma, Robin Deits, and Kanako Miura, "A closed-form solution for real-time ZMP gait generation and feedback stabilization", In *Proceedings of the IEEE/RAS International Conference on Humanoid Robotics, November 2015*.
- [9] Russ Tedrake, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2014.
- [10] Gurobi Optimization et al, "Gurobi optimizer reference manual", URL: <http://www.gurobi.com>, 2012.
- [11] Hongkai Dai and Russ Tedrake, "Planning robust walking motion on uneven terrain via convex optimization", *Under Review*, 2016.