

Algoritmo estocástico para la generación automática de trayectorias de un robot humanoide

Stochastic algorithm for automatic path planning of a humanoid robot

DOI: <http://dx.doi.org/10.17981/ingecuc.14.1.2018.03>


Artículo de investigación científica. Fecha de recepción: 05/10/2017-. Fecha de aceptación: 06/03/2018

Cristian David Villate Martínez 

Universidad de Pamplona. Pamplona (Colombia).
cristian.villate@unipamplona.edu.co

César Augusto Peña Cortés 

Universidad de Pamplona. Pamplona (Colombia).
cesarapc@unipamplona.edu.co

Oscar Eduardo Gualdrón Guerrero 

Universidad de Pamplona. Pamplona (Colombia).
oscar.gualdron@unipamplona.edu.co

Para citar este artículo:

C. D. Villate Martínez, C. A. Peña Cortés y O. E. Gualdrón Guerrero, "Algoritmo estocástico para la generación automática de trayectorias de un robot humanoide," *INGE CUC*, vol. 14, no. 1, pp. 30-40, 2018. DOI: <http://doi.org/10.17981/ingecuc.14.1.2018.03>

Resumen

Introducción– La incorporación de sistemas de aprendizaje autónomos en la robótica permitirá la resolución de una gran cantidad de problemas. Uno de ellos es la marcha autónoma para el caso de los robots humanoides debido a la complejidad que tiene por la gran cantidad de variables que influyen en este proceso.

Objetivo– Desarrollar algoritmos que generen marchas autónomas en un robot humanoide con varios grados de libertad.

Metodología– El estudio inicia con el desarrollo de algoritmos estocásticos con pocas dimensiones; luego, se extiende a situaciones n-dimensionales. Posteriormente, se realizan pruebas en simulación, y, por último, las pruebas experimentales.

Resultados– Se generó un algoritmo basado en el modelo físico del robot para crear las trayectorias de marcha estocásticamente. Se implementó un simulador que contempla las restricciones cinemáticas incluyendo colisiones para verificar los resultados. Adicionalmente, se realizaron cien pruebas experimentales donde se verificó el correcto funcionamiento de las trayectorias.

Conclusiones– Se pudo corroborar que es posible crear un algoritmo estocástico que mezcla reglas determinantes y aleatorias para generar marchas automáticamente en robots humanoides, extendiendo conceptos generados en espacios bidimensionales y tridimensionales a coordenadas articulares n-dimensionales.

Palabras clave– Robots humanoides, planificación de trayectorias, robots autónomos

Abstract

Introduction– The incorporation of an autonomous learning system in robotics will allow the resolution of a large number of problems. One is the autonomous march of the humanoid robots due to its complexity in the great number of variables regarding this process.

Objective– Develop algorithms that generate autonomous paths in a humanoid robot with various degrees of freedom.

Methodology– The study begins with the development of stochastic algorithms with few dimensions. Then, it will be extended to n-dimensional situations. Afterwards, simulation tests will be carried out. And finally, the experimental tests are performed.

Results– An algorithm was generated based on the physical model of the robot to create walking paths stochastically. A simulator that contemplates the kinematic constraints, including collisions, was implemented to verify the results. In addition, one hundred experimental tests were done. With these tests, the correct operation of the trajectories was verified.

Conclusions– It was verified that it is possible to create a stochastic algorithm that mixes determinant and random rules to automatically generate paths in humanoid robots, hence, extending concepts generated in two-dimensional and three-dimensional spaces to n-dimensional articulated coordinates.

Keywords– Humanoid Robots, Path Planning, Autonomous Robot

I. INTRODUCCIÓN

La robótica humanoide es un campo vanguardista y de alto impacto que está relacionado con el avance de la sociedad como imagen primaria del futuro. Uno de los principales problemas que se presentan en este tipo de dispositivos es la locomoción.

En robótica humanoide existen diversos tipos de alternativas en locomoción, dentro de las cuales se encuentran el uso de ruedas [1], orugas, piernas [2][3] o estructuras mixtas [4]. Una de las estructuras en las que se hace mayor referencia son las bípedas, dado que se pueden movilizar en entornos similares a los que habita el hombre. Por esta razón, en un robot humanoide se debe hacer hincapié en las piernas, ya que deben ser capaces de realizar una variedad de tareas, incluyendo la interacción con el entorno mientras se mantiene el equilibrio [5]. Existen delimitaciones básicas para hacer frente a estos escenarios, como por ejemplo, que el robot debe ser capaz de moverse en entornos no estructurados [6].

Se ha abordado la problemática del desplazamiento bípedo en humanoides analizando los puntos ciegos que puede tener un robot al desplazarse, llegando a estrategias de soluciones como la utilización del método "dual-SLIP 3D" [7], el cual consiste en un sistema que detecta cuándo el terreno es plano e irregular para así caminar de una manera más eficiente y economizando recursos tanto energéticos como computacionales.

Debido a la semejanza con la morfología humana, los robots humanoides presentan características similares en su locomoción, encontrándose básicamente con las mismas dificultades. El equilibrio es uno de los conflictos más recurrentes dentro de su operatividad. Los humanos normalmente tardan entre tres y cuatro meses desde que dan sus primeros pasos hasta que caminan bien (emplean otros dos meses en hacerlo con una mayor velocidad) [8]. En ese tiempo aprenden a mover eficientemente los miembros de sus cuerpos. Las caídas hacen parte del proceso de aprendizaje pues éstas les permiten, de forma intuitiva, aprender cuáles movimientos realizar y cuáles no. Lógicamente, estos procesos de aprendizaje requieren de un esfuerzo mental [9]. Por esta razón, se propone crear un algoritmo que genere trayectorias por medio de su propio aprendizaje y, de esta forma, almacene los resultados hasta que se generen otros con mejores prestaciones.

Mantener el equilibrio en un sistema robótico humanoide es una tarea compleja, por lo tanto, realizar una marcha que involucre sobrepasar obstáculos lo es más. Por esta razón, se han creado estrategias de navegación multi-objetivo midiendo las distancias que existen entre el objetivo, el obstáculo y el robot, activando así algoritmos de marcha para las diferentes situaciones que se le presenten [10].

Otras estrategias adoptadas pueden ser la imitación de movimientos [11], donde se toman los patrones de la marcha humana como puntos de partida

para generar trayectorias para un robot. Este tipo de estrategias aumentan la velocidad de procesamiento en el aprendizaje o generación de rutas [12].

Desde un enfoque tradicional en robótica, existen dos tipos de trayectorias que se pueden generar para realizar una determinada tarea. Éstas son: las trayectorias punto a punto y las trayectorias continuas [13]. En las primeras solo interesa que el robot parta de un punto inicial y logre llegar al punto final, mientras que en las trayectorias continuas es necesario tener en cuenta el recorrido para que el robot cumpla con una trayectoria analítica previamente definida, usualmente una línea o un arco-círculo.

Para el caso de los robots humanoides es necesario que las articulaciones partan de una posición inicial y lleguen a su objetivo final, sin embargo, a pesar de que su cometido no es que los puntos de apoyo del robot cumplan con una trayectoria analítica, se debe tener en cuenta el recorrido de las articulaciones para que no causen caídas o colisiones, lo cual genera unas restricciones altamente acopladas entre el valor de las mismas [14].

Existen un número infinito de posibilidades para generar trayectorias en los robots humanoides dado su nivel de redundancia en grados de libertad, sin embargo, es obligatorio escoger soluciones que cumplan con las restricciones cinemáticas y dinámicas para asegurar su estabilidad [15]. Por estos motivos, en este artículo se propone un algoritmo computacional estocástico que genera trayectorias basándose en reglas predecibles sumado a reglas aleatorias, lo cual le permite al robot realizar una marcha sobrepasando las dificultades asociadas dentro de las cuales se recalca la conservación del equilibrio para evitar caídas.

La plataforma robótica empleada en esta investigación para hacer las pruebas experimentales es el robot Humanoide Bioid Premium (tipo A). Este ha sido objeto de estudio en varias investigaciones dado su nivel de accesibilidad, características mecánicas y su sistema de control. Unos ejemplos fueron: el análisis cinemático directo e inverso, contemplando todos los puntos singulares presentes en el robot [16]; el análisis de velocidades utilizando una fusión de sensores propioceptivos para realizar análisis energéticos [17], entre otros.

Para el análisis cinemático del robot Bioid se empleó el método de los *screws*, el cual hace uso del teorema de Charles para modelar el desplazamiento general de un cuerpo rígido en el espacio, representado por medio una rotación más una translación. A esta combinación de movimientos se le conoce como desplazamiento *screw* [18].

Este artículo está organizado de la siguiente manera: en la sección dos se presenta la metodología implementada para el desarrollo de los algoritmos propuestos; en la sección tres se presentan las pruebas, los resultados y su respectiva discusión; y, por último, se presentan las conclusiones.

II. METODOLOGÍA

La investigación se realizó de forma sinérgica, integrando dos temáticas: la robótica humanoide y la programación computacional. Por esta razón, se debe abordar desde dos enfoques diferentes pero complementarios entre sí.

A. Fundamento matemático

Los robots son máquinas que involucran aspectos como: el análisis cinemático directo, el inverso, el modelamiento dinámico y la planificación de trayectorias. En este proyecto se plantea un análisis de movimiento de forma pseudoestática (en el cual las velocidades desarrolladas por las articulaciones son bajas) con la finalidad de desestimar las fuerzas dinámicas involucradas en el movimiento (fuerzas de Coriolis, centrífuga, centrípeta, etc.).

El análisis cinemático directo del robot humanoide se puede desarrollar por medio de diversos métodos. El más tradicional es el método de Denavit-Hartenberg [13], el cual tendría un grado elevado de complejidad dada su estructura que involucra 16 reglas. Sin embargo, existe un método moderno denominado desplazamiento *screw*, que a pesar de que su concepción involucra un manejo matemático más complejo, su aplicación es muy simple y eficaz, lo cual lo hace idóneo para aplicar en robots humanoides que contengan eslabones con geometrías complejas. El método de desplazamiento *screw* [18] plantea:

$$p_2 = s_0 + ts + (p_1 - s_0)\cos\theta + s \times (p_1 - s_0)\sin\theta + [(p_1 - s_0)^T s]s(1 - \cos\theta) \quad (1)$$

Donde p_n representa el punto de desplazamiento de un punto, s_0 es el vector perpendicular del sistema de origen al punto de desplazamiento y s representa el vector *screw* (Fig. 1), esto según la fórmula de Rodrigues. El método de desplazamiento *screw* se encuentra ampliamente explicado en [19]. Cabe recalcar que los vectores *screw* se ubican sobre el eje de cada una de las articulaciones del robot. En la fig. 2 se puede apreciar un ejemplo de la ubicación de los vectores *screws* sobre las articulaciones de las extremidades inferiores del robot.

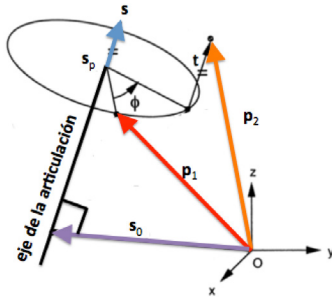


Fig. 1. Diagrama esquemático del desplazamiento Screw.
Fuente: Autores.

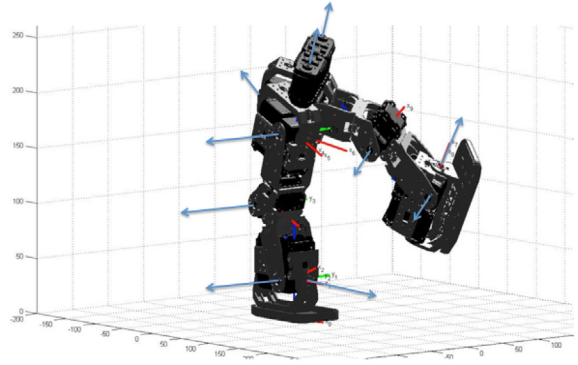


Fig. 2. Vectores screw de las extremidades inferiores del robot Bioloid.
Fuente: Autores.

La fórmula de Rodrigues también se puede escribir en términos de una matriz de transformación homogénea definida por $A_{4 \times 4}$, donde sus componentes son:

$$a_{11} = (s_x^2 - 1)(1 - \cos(\theta)) + 1 \quad (2)$$

$$a_{12} = s_x s_y (1 - \cos(\theta)) - s_z \sin(\theta) \quad (3)$$

$$a_{13} = s_x s_z (1 - \cos(\theta)) + s_y \sin(\theta) \quad (4)$$

$$a_{21} = s_y s_x (1 - \cos(\theta)) + s_z \sin(\theta) \quad (5)$$

$$a_{22} = (s_y^2 - 1)(1 - \cos(\theta)) + 1 \quad (6)$$

$$a_{23} = s_y s_z (1 - \cos(\theta)) - s_x \sin(\theta) \quad (7)$$

$$a_{31} = s_z s_x (1 - \cos(\theta)) - s_y \sin(\theta) \quad (8)$$

$$a_{32} = s_z s_y (1 - \cos(\theta)) + s_x \sin(\theta) \quad (9)$$

$$a_{33} = (s_z^2 - 1)(1 - \cos(\theta)) + 1 \quad (10)$$

$$a_{14} = t s_x - s_0 x (a_{11} - 1) - s_0 y a_{12} - s_0 z a_{13} \quad (11)$$

$$a_{24} = t s_y - s_0 x a_{21} - s_0 y (a_{22} - 1) - s_0 z a_{23} \quad (12)$$

$$a_{34} = t s_z - s_0 x a_{31} - s_0 y a_{32} - s_0 z (a_{33} - 1) \quad (13)$$

$$a_{41} = a_{42} = a_{43} = 0 \quad (14)$$

$$a_{44} = 1 \quad (15)$$

Para calcular el desplazamiento del efector final del robot desde la posición de referencia (q_0) hasta la posición objetivo (q), se considera como la resultante de desplazamiento de n *screws* sucesivos, lo cual puede ser calculado por medio de la denominada ecuación de cierre:

$$q = A_1 A_2 A_3 A_4 A_5 A_{n-1} A_n \quad (16)$$

En la Tabla 1 se aprecia un ejemplo del vector *screw* y su perpendicular para el caso de las 6 primeras articulaciones correspondientes al pie de apoyo (donde L_i corresponde a las dimensiones de los eslabones). Al aplicar (2)-(15) se obtienen las matrices de transformación homogénea con las cuales se plantea la ecuación de cierre para obtener la localización del sistema del eslabón 6 con respecto al origen. De forma análoga, se aplica el procedimiento para el resto de las articulaciones del robot, incluyendo las del torso y las extremidades superiores.

TABLA 1. ANÁLISIS POR EL MÉTODO DE LOS SCREW DEL PIE DE APOYO DEL ROBOT BIOLOID.

| Análisis del pie de apoyo del robot Bioloid | | | |
|---|------------|-------------------|-----|
| Articulación | S_i | S_{0i} | MTH |
| 1 | (0, -1, 0) | (0, 0, L1) | A1 |
| 2 | (1, 0, 0) | (0, 0, L1) | A2 |
| 3 | (0, -1, 0) | (-L4, 0, L1+ L2) | A3 |
| 4 | (0, -1, 0) | (0, 0, L1+ L2+L3) | A4 |
| 5 | (1, 0, 0) | (0, 0, L1+ L2+L3) | A5 |
| 6 | (0, 0, 1) | (0, 0, 0) | A6 |

Fuente: Autores.

En la Fig. 3, se ilustra la disposición inicial de los sistemas coordenados del robot. Se puede apreciar que los brazos, en primera instancia, se encuentran separados del cuerpo con el fin de no interferir con los movimientos de las extremidades inferiores.

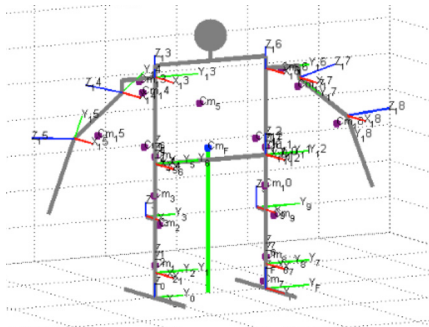


Fig. 3. Sistemas coordenados del robot Bioloid. Fuente: Autores.

B. Modelo CAD

Se realizó un diseño asistido por computador (CAD) para modelar gráficamente el robot Bioloid Premium tipo A. Éste permite analizar los movimientos del robot antes que la plataforma experimental los ejecute en la realidad y, de esta forma, evitar daños en la misma. Adicionalmente, por medio del modelo, se pudo determinar con precisión los centros de masa de los eslabones. En la Fig. 4 se aprecia una imagen del modelo.

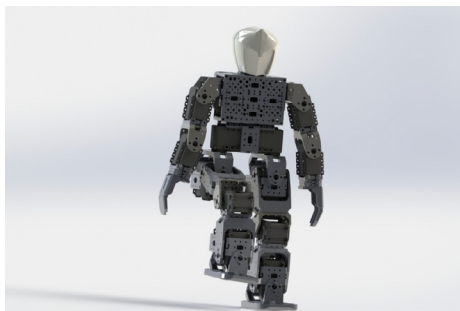


Fig. 4. Diseño CAD del robot Bioloid. Fuente: Autores.

C. Simulador de marchas

Con el fin de visualizar con mayor precisión los movimientos de la marcha del robot, se generó un simulador que recrea en tiempo real las piezas del robot generadas en el CAD y que posee las mismas características del robot real. Este simulador puede ejecutarse desde Matlab u Octave. En la Fig. 5, se puede apreciar una imagen del simulador desarrollado.

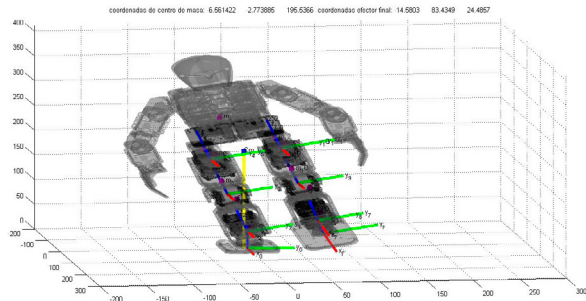


Fig. 5. Simulador de trayectorias del robot Bioloid. Fuente: Autores.

El objetivo principal de la creación del simulador es poder visualizar y corroborar los ángulos obtenidos por los algoritmos computacionales sin golpear o estropear el robot, además, permite verificar el funcionamiento de los resultados obtenidos de una forma más rápida que en la vida real. Su desarrollo se hizo con base en el análisis cinemático realizado anteriormente.

Las piezas del diseño CAD del robot se transformaron en datos matriciales utilizando el formato estéreo-litografía (basado en la triangulación de superficies) con el fin de manipular su posición y orientación al acoplar los conceptos matemáticos de las matrices de transformación homogéneas.

Se calcularon y visualizaron los centros de masa, tanto de cada eslabón como del sistema total, esto sirve, desde un análisis pseudoestático, para estudiar el movimiento de desplazamiento de un pie del robot cuando el otro pie se utiliza como apoyo. La idea general es cumplir la restricción de que la proyección del centro de masa total del robot sobre el suelo debe coincidir con el área de contacto del pie de apoyo para evitar que el robot pierda el equilibrio y se caiga [6]. En caso de que el robot esté apoyado en sus dos piernas, la proyección debe estar entre los dos puntos de apoyo (Fig. 3).

Los movimientos válidos tomados para el desarrollo de las trayectorias corresponden únicamente cuando se cumple la restricción previamente mencionada, además que su movimiento concuerda morfológicamente para evitar choques entre los elementos de su estructura. Adicionalmente, el simulador permite conocer la posición espacial del efector final del robot, que sería el pie con el cual está realizando el desplazamiento.

Se utilizó la alternativa de referenciar o acoplar tres articulaciones con respecto a las otras articulaciones para permitir que el pie móvil del robot estuviera siempre paralelo al piso. De esta forma, se pudo reducir el problema de 12 articulaciones a 9 (para las piernas), con el fin de disminuir la complejidad del mismo y, por ende, aumentar la velocidad de cómputo de los algoritmos.

D. Generación de marchas autónomas

El objetivo de la marcha autónoma propuesta es crear un algoritmo que, basado en unas coordenadas articulares iniciales y finales predeterminadas, genere la evolución de cada una de las articulaciones correspondientes a los ángulos ejecutados por los servomotores para que el robot realice el desplazamiento sin que se produzca ningún choque ni pierda su equilibrio.

La realización de estos algoritmos se puede realizar por medio de diferentes alternativas, como: el empleo de reglas heurísticas, estocásticas o técnicas avanzadas de inteligencia artificial. Puntualmente, para esta investigación, en un inicio y con el desarrollo de algoritmos de evolución articular de forma plenamente no determinista (aleatoria), con estos algoritmos se proponía que la evolución de cada articulación tuviese tres opciones en cada iteración:

- Giro en un sentido con un ángulo fijo
- Giro en sentido contrario
- Permanecer inmóvil durante la iteración

Lógicamente, en cada iteración se corrobora si las coordenadas articulares corresponden a puntos válidos o no, es decir, si existe o no una colisión o pérdida de equilibrio.

Con el fin de abordar el problema de lo simple a lo complejo, en términos de tiempo de ejecución y costos computacionales, se inició por estudiar casos con pocos grados de libertad y llegar a conjeturas que permitan su extensión a más grados de libertad. En principio, se plantearon pruebas con solo tres grados articulares, teniendo como meta partir de la posición inicial y llegar a tres posiciones diferentes y preestablecidas. Adicionalmente, se planteaban como restricciones algunas regiones en las cuales las articulaciones no podían pasar. Estas regiones representarían las zonas del espacio articular que corresponden a situaciones donde el centro de masa del robot se proyecta fuera de la superficie de apoyo del robot, haciendo que éste pierda el equilibrio o algunas de sus partes colisionen. Como inicialmente se estudia el caso de 3 grados de libertad, estas regiones se definen de forma heurística. En la sección de resultados se presenta una analogía (entre el espacio de la tarea y el espacio articular) con un robot que se desplaza en 3 dimensiones y existen obstáculos que debe evadir.

En la Fig. 6 se ilustra un ejemplo en el cual todas las articulaciones parten de un mismo punto y en cada iteración avanzan de forma independiente en

un ángulo preestablecido según un sentido que se define de forma aleatoria. Este proceso se realiza de forma indefinida hasta que todas las articulaciones lleguen a su meta. Cabe resaltar que se descartan todas las trayectorias que implique el paso de las articulaciones por zonas restringidas (en este caso se tomaron los ángulos absolutos mayores a 25 grados).

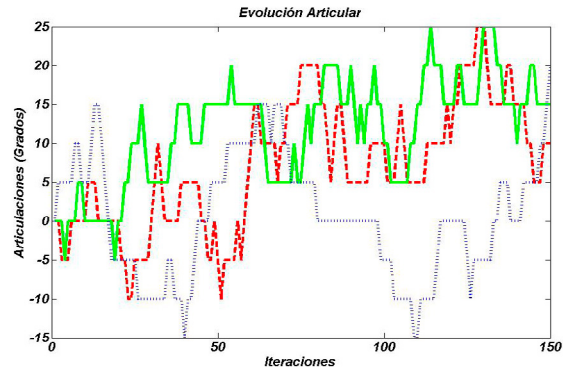


Fig. 6. Evolución articular de tres grados de libertad. Fuente: Autores.

La evolución articular implementada, por pertenecer a un método plenamente no determinista, no siempre es la misma; por esta razón, su resultado varía cada vez que se aplica, generando así múltiples soluciones al problema de locomoción. Algunas de ellas son muy rápidas y otras muy lentas. Adicionalmente, dentro de las soluciones consideradas como válidas pueden existir tramos de las trayectorias que no son útiles, por ejemplo: situaciones en las cuales las articulaciones se muevan y vuelvan a una posición sin producir un desplazamiento total o sin evadir una restricción. Con el fin de disminuir el número de iteraciones, se generó un algoritmo que elimina estos tramos, haciendo que la trayectoria final sea más corta o eficiente. En la Fig. 7 se ilustra un ejemplo de una trayectoria empleando esta técnica para el caso de 3 grados de libertad (puede observarse que el número de iteraciones es significativamente menor).

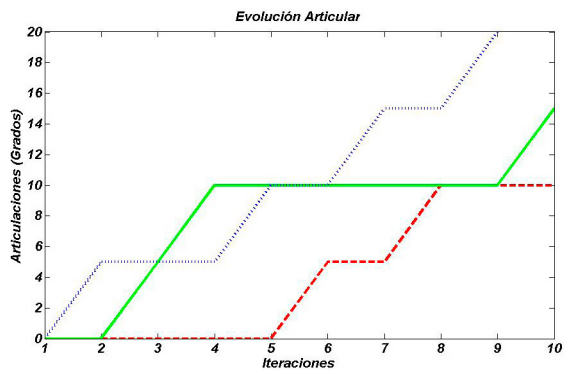


Fig. 7. Evolución articular de tres grados de libertad. Fuente: Autores.

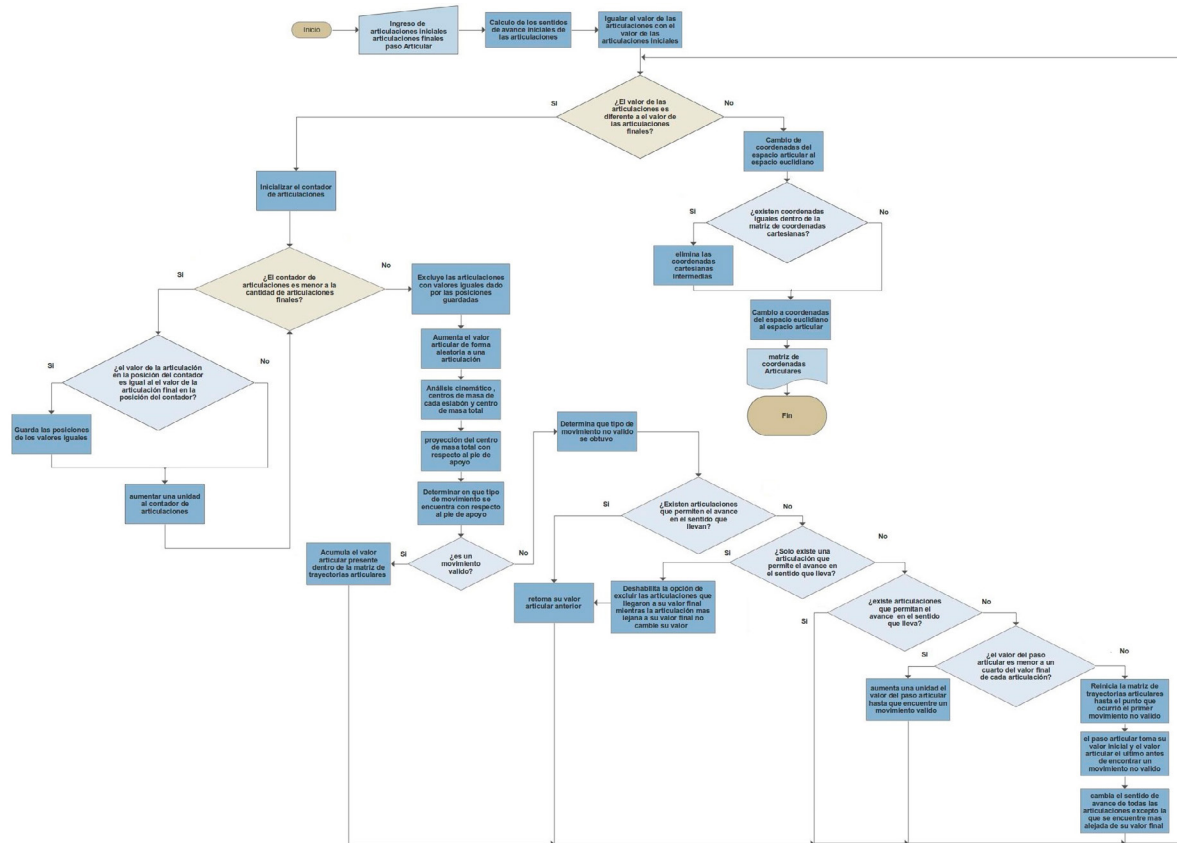


Fig. 8. Esquema del algoritmo de generación de marchas autónomas.
Fuente: Autores.

Aunque este algoritmo es efectivo para encontrar soluciones al problema planteado, este método hace que el tiempo de ejecución crezca significativamente con el número de grados de libertad. Así, para el caso de 12, requirió de 5 días para encontrar una sola solución (utilizando un procesador Intel Core I7 de 2.8Ghz), lo cual lo hace inviable para su implementación. Por esta razón, para el cálculo de las trayectorias articulares del robot humanoide con todos los grados de libertad, se decidió agregar algunas reglas deterministas que actúan en conjunto con las aleatorias para disminuir el número de iteraciones para obtener una solución válida. En la Fig. 8, se ilustra el esquema del algoritmo de generación de marchas.

Las reglas que describen el algoritmo son las siguientes:

En primer lugar, se definen el sentido de avance de cada articulación, de tal forma que, si el valor final es mayor o igual al valor inicial de la articulación, el avance sea positivo; y si el valor final es menor, el avance sea negativo. Esta regla determinista actúa siempre y cuando no exista una colisión o el robot se desestabilice, en caso contrario, se aplican unas reglas no deterministas que se enuncian posteriormente.

Dado que el número de iteraciones se incrementa con el número de grados de libertad, se optó por ir excluyendo las articulaciones a medida que llegaran a sus valores finales (logrando sus objetivos). Esta técnica funciona correctamente siempre y cuando no se llegue a unas coordenadas articulares correspondientes a un punto no válido (colisión o desequilibrio). Como ejemplo equivalente en el caso bidimensional, se puede pensar en el desplazamiento de un robot a lo largo de los ejes X y Y para llegar a un punto predeterminado. Imagínese que después de varias iteraciones el robot logró ubicarse sobre la coordenada objetivo del eje Y, haciendo que la diferencia radique solamente sobre el eje X, esto hace que la ruta más eficiente sea una línea recta a lo largo de ese eje; sin embargo, en caso de que el robot se encuentre con un obstáculo en esa dirección, es necesario cambiar su orientación para evadir el desplazamiento (requiriendo nuevamente realizar un desplazamiento sobre el eje Y a pesar de cumplir con su cometido final en esa coordenada).

Para el caso del robot con 9 grados de libertad, cuando se llega a un punto no válido, se almacena la última posición articular válida y se procede a analizar cuáles articulaciones no han llegado a su objetivo. Posteriormente, se busca la articulación que se

encuentra más lejos de cumplir su objetivo dado que es la que requiere de un mayor número de iteraciones. En este caso, se realiza una excepción para que las articulaciones que ya habían cumplido su cometido se muevan cambiando de sentido de giro (aleatoriamente) más allá de sus valores objetivo siempre y cuando no superen sus límites máximos (u otras restricciones). Luego, se compara en cada iteración si el valor de la articulación más alejada ha llegado o no a su valor final o si difiere del valor guardado anteriormente; si es diferente, en teoría, ha sobrepasado el obstáculo en ese punto. Posteriormente, se procede a recalcularse el sentido del avance de las otras articulaciones para retomar la dirección hacia el objetivo final.

Cada vez que se presente una colisión, se analiza la cercanía de los valores articulares de su objetivo final en el instante justo antes de este suceso, y luego se compara la cantidad de articulaciones que llegaron a su límite final con la cantidad de valores articulares con el fin de detectar la trayectoria que esté más cerca de sus valores finales y que no presente ninguna colisión, para luego ser asignada cuando ocurra una colisión y no exista una posibilidad de avance.

Al aplicar los algoritmos dentro del simulador, los movimientos que se realizan se representan con una variable lógica binaria cuyos estados son: “movimiento válido” y “movimiento no válido”. Cuando se analizó la generación de trayectorias se detectó que en algunas ocasiones se requiere aumentar el ángulo de avance de las articulaciones (denominado paso) para superar un obstáculo (movimiento no válido), y, de esta forma, saltar de un movimiento válido a otro válido. Al representar este fenómeno en un espacio bidimensional, parecería como si existiesen agujeros en suelo, requiriendo aumentar el avance para lograr superarlo y llegando así a otro punto válido.

Finalmente, al obtener la trayectoria en valores articulares, se transforman a coordenadas cartesianas (cinemática directa del robot tomando el extremo de un pie como el efector final y el extremo del otro pie como la base), se compara qué coordenadas poseen el mismo valor y se eliminan todas las coordenadas intermedias que existen entre estos dos puntos con la finalidad de recortar la trayectoria en los puntos redundantes. Una vez realizada esta depuración, se procede a realizar nuevamente una transformación al espacio articular.

La trayectoria total del robot, denominada marcha, está compuesta por una serie de pasos o sub-trayectorias a las cuales se les aplica el procedimiento previamente descrito. A continuación, se enuncian los pasos:

- El robot parte con los dos pies en el suelo y procede a situar la proyección de su centro de gravedad sobre el pie de apoyo.
- Se apoya sobre un solo pie y levanta el otro a una altura máxima de 10mm y lo lleva una distancia de 50 mm.
- Coloca el segundo pie en el suelo (uno más adelante que otro).

- Vuelve con los pies haciendo que su centro de gravedad se ubique en el medio de los puntos de apoyo.
- Repite la acciones solo que cambiando el pie de apoyo y levantando el otro.

V. PRUEBAS Y RESULTADOS

En la Fig. 9, se presenta un ejemplo del primer concepto planteado (algoritmo netamente no determinista), donde la evolución de las articulaciones se ejerce por medio de reglas aleatorias (girar un ángulo en un sentido, no avanzar, girar un ángulo en sentido opuesto). En primera instancia, se realizaron pruebas en un espacio bidimensional para partir de un punto (coordenadas 7, 17) y llegar a otro objetivo (coordenadas 5, 2), evadiendo un obstáculo en su camino (área no permitida de color verde comprendida en X: 2 a 9 y Y: 3 a 12). En este caso, los ángulos de las articulaciones se cambian por la distancia de avance sobre el eje X y Y. Debe notarse que además de las restricciones correspondientes a los obstáculos se encuentran las de los límites exteriores (0 y 10 en X, 0 y 20 en Y). Se puede ver cómo el robot logra llegar a su destino.

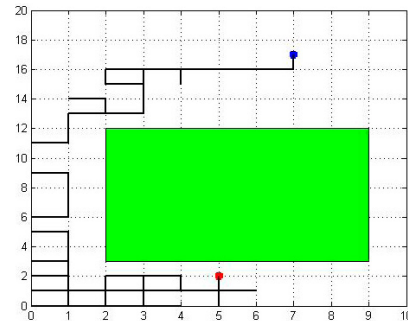


Fig. 9. Simulación de trayectorias en 2D con obstáculos.
Fuente: Autores.

Haciendo una extrapolación (donde cada eje representa una articulación), se realizaron experimentos tridimensionales con las mismas condiciones. En la Fig. 10, se puede apreciar como el robot también logra llegar a su cometido. Sin embargo, en estas pruebas se aprecia claramente que existen movimientos innecesarios que ralentizan la trayectoria total.

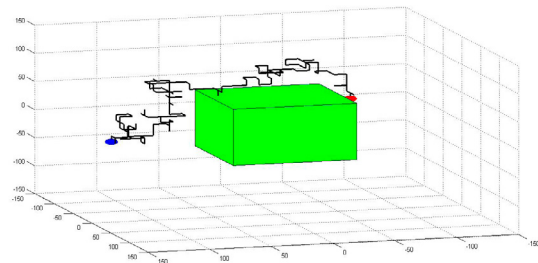


Fig. 10. Simulación de trayectorias en 3D con obstáculos.
Fuente: Autores.

Posteriormente, se procedieron a realizar pruebas con la extrapolación para las 9 articulaciones (Fig. 11). En estas pruebas, la generación de una trayectoria, partiendo de una posición articular inicial hasta llegar a una final, se realizó de forma correcta, aunque su cálculo y su respuesta fue muy lenta (16 horas), lo cual hace inoperable el algoritmo en la práctica.

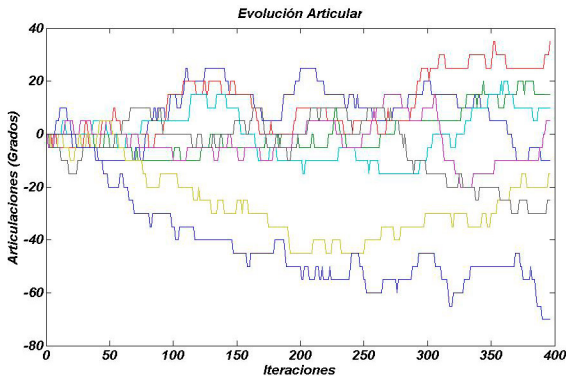


Fig. 11. Generación de trayectorias de 9 grados articulares en su fase de prueba.
Fuente: Autores.

En contraste con estos resultados, al emplear el algoritmo estocástico que mezcla reglas deterministas con aleatorias (mencionada anteriormente), se pudo apreciar una mejora significativa en la generación de los movimientos que realiza el robot en experimentos en un espacio tridimensional. Se aprecia claramente cómo el robot supera los obstáculos de una forma eficiente y rápida (Fig. 12).

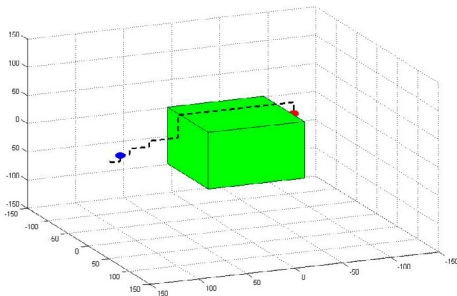


Fig. 12. Simulación de trayectorias optimizadas en 3D con obstáculos.
Fuente: Autores.

Los resultados del algoritmo estocástico tuvieron una mejora bastante notoria, reduciendo tanto en cantidad de iteraciones y cantidad de movimientos, como en el tiempo de ejecución. El algoritmo netamente no determinista contó con un tiempo de ejecución de aproximadamente una hora, mientras que al aplicar el nuevo algoritmo propuesto se redujo a dos segundos y la cantidad de movimientos para realizar la trayectoria se redujo de 300 a 94. Teniendo en cuenta estos resultados, se dispuso a dar paso a

la implementación con las 9 articulaciones del robot (correspondientes a las de los pies), añadiéndole, a su vez, los cálculos cinemáticos y las restricciones mecánicas que posee el robot (siendo éstas análogas a los obstáculos en los casos de dos y tres dimensiones).

Al realizar las pruebas, se mostró igualmente una mejora significativa en la cantidad de iteraciones, cantidad de movimientos y tiempo de ejecución, permitiendo que su operatividad esté dentro de los parámetros permitidos, generando así la posibilidad de ser un algoritmo *on line* en el robot (Fig. 13). La respuesta del primer algoritmo fue de aproximadamente 16 horas, y con la optimización la pudo realizar en 3 segundos. Sus movimientos se redujeron de 470 a 190, aproximadamente.

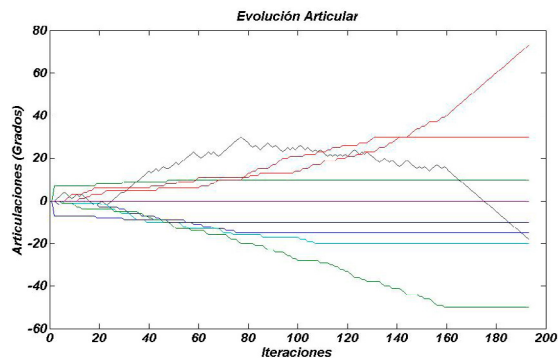


Fig. 13. Simulación de trayectorias empleando 9 articulaciones.
Fuente: Autores.

Para visualizar los movimientos del robot, se pasaron los resultados obtenidos al simulador donde se encuentra el modelo gráfico (Fig. 14). Como una marcha está compuesta por diferentes pasos, se decidió dividirlo en secciones. De esta forma, se parte de un reposo; luego, se sube el pie izquierdo hasta un punto máximo; después, se baja este pie nuevamente hasta el piso, pero más alejado; seguidamente, el pie izquierdo se convierte en el pie fijo y el derecho lo desplaza hasta un punto máximo y lo pasa nuevamente hasta el piso, pero más alejado. Nuevamente, el pie izquierdo vuelve al punto más alto, y finalmente, apoya los dos pies de forma paralela en el piso, generando así una marcha.

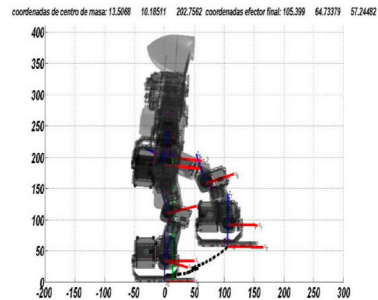


Fig. 14. Generación de trayectorias con nueve grados articulares en su fase final implementado en el simulador.
Fuente: Autores.

Adicionalmente, el simulador permite la distancia de la proyección del centro de gravedad con respecto al punto de apoyo (centro del pie de apoyo del robot humanoide), el cual se determina como “error”. Este error será una medida que indica el equilibrio del robot en una posición determinada cuando se apoya en un solo pie (Fig. 15).

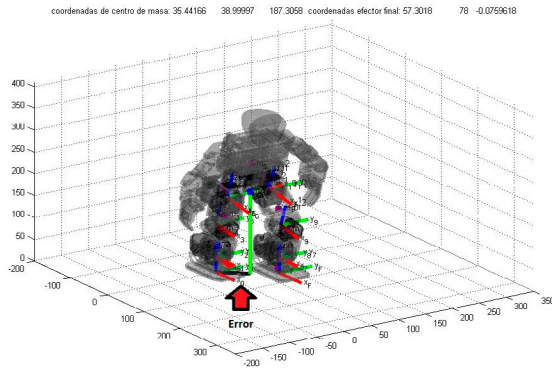


Fig. 15. Error o distancia de la proyección del centro de gravedad con respecto al punto de apoyo.
Fuente: Autores.

Una vez superada la etapa de simulación, se procedió a la verificación de los resultados por medio de una etapa experimental. La idea general es ejecutar las trayectorias generadas por el algoritmo estocástico en la plataforma robótica (Fig. 16) y, de esta forma, probar si se cumple o no. En caso que el robot pierda equilibrio, se le considera como un intento fallido.

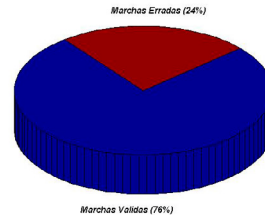


Fig. 16. Modelo físico del robot Bioloid.
Fuente: Autores.

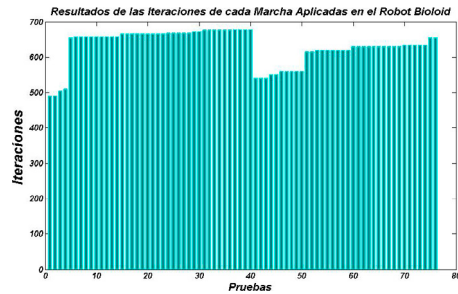
Se realizaron cien pruebas con el robot real implementando diferentes trayectorias generadas y se obtuvo un 76% de efectividad a la hora de realizar una marcha. En la Fig. 17, se pueden apreciar algunos datos estadísticos que ponen en evidencia la tendencia en la generación de las marchas.

En la Fig. 17b, se aprecia el número total de iteraciones de la marcha en cada uno de los experimentos (debe recalarse que la marcha está compuesta por varias subtrayectorias descritas anteriormente). Al aplicarse un número significativo de pruebas, como el mencionado, se da la posibilidad de mejorar el proceso de aprendizaje de la marcha, pues se escoge el mejor de los resultados de acuerdo a los requerimientos.

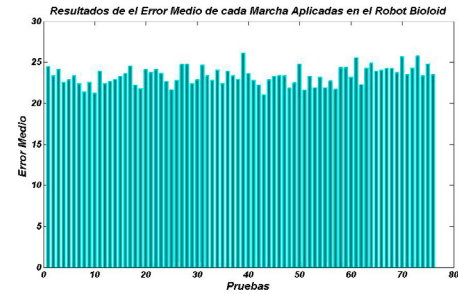
Porcentaje de Aciertos de las Marchas Realizadas en el Robot Bioloid



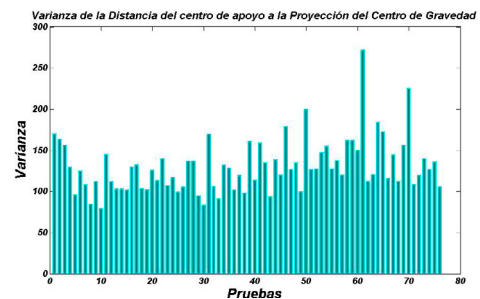
(a) Porcentaje de aciertos



(b) Número de iteraciones por prueba



(c) Error medio



(d) Varianza

Fig. 17. Resultados de la varianza de las marchas del robot Bioloid.
Fuente: Autores.

Por ejemplo, si se desean mejorar los tiempos de ejecución, basta con seleccionar la trayectoria con el menor número de iteraciones. Si se desea mejorar la estabilidad, un buen indicador es el error medio (distancia de la proyección del centro de masas del robot con el centro del pie de apoyo del robot) (Fig. 17c). Sin embargo, es bueno asegurarse, adicionalmente, que la varianza del error sea pequeña, garantizando que el robot no se acerque a movimientos no válidos durante la ejecución de la trayectoria.

La marcha más estable se determinó al analizar las desviaciones estándar y las varianzas de las pruebas que tuviera menor valor. En este caso, indica que la proyección del centro de masas está más cerca del pie de apoyo, dando como resultado la prueba diez con una desviación estándar de 8,879 y una varianza de 78,849.

Por otra parte, se pudo poner en evidencia, a través de las pruebas experimentales, que la respuesta de la plataforma robótica correspondía con la respuesta del simulador (Fig. 18). Esto es un gran logro teniendo en cuenta las consideraciones de un movimiento pseudoestático.

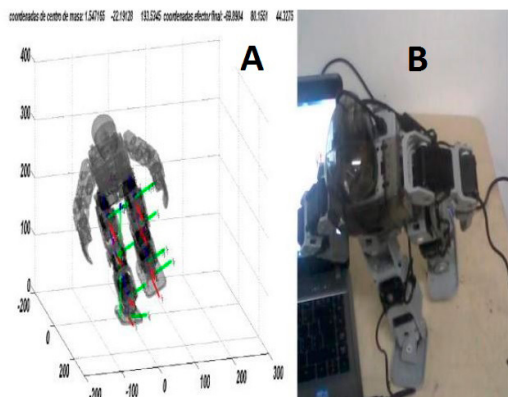


Fig. 18. Comparación de resultados.
(a) Simulador robot Bioloid. (b) Robot Bioloid real.
Fuente: Autores.

VI. CONCLUSIONES

Realizando una comparación entre los algoritmos implementados, se puede concluir que las reglas no deterministas ofrecen un aporte importante a la hora de sobrepasar obstáculos. Sin embargo, si las reglas son netamente de carácter aleatorio, la eficiencia es mínima, dado que los tiempos de ejecución son muy elevados para casos con varios grados de libertad (e.g. 16h para 9 GDL). Por esta razón, el algoritmo estocástico propuesto disminuye significativamente el número de iteraciones, ya que aplica reglas deterministas en las zonas donde no hay obstáculos (colisiones, límites u otro tipo de restricciones), generando trayectorias más cortas.

Se pudo corroborar que es posible analizar la marcha final automática del robot extrapolando el concepto utilizado en los espacios bidimensionales y tridimensionales. En este orden de ideas, se podría asumir que el algoritmo generador de trayectorias funciona para n dimensiones donde cada dimensión se representa con una variable articular.

El algoritmo se puede utilizar también para generar una trayectoria a cualquier punto válido que el robot posea. Por su naturaleza autónoma, puede funcionar en entornos cambiantes siempre y cuando contemplen las restricciones de velocidad, eje gravitatorio y se tomen como válidas las trayectorias con menor error.

El tiempo de ejecución del nuevo algoritmo es significativamente menor, lo cual permitirá su aplicación en línea por medio de un sistema de control embebido, adaptándose automáticamente a las necesidades de locomoción que se presente en ese preciso instante. Adicionalmente, se propone, para un próximo trabajo, incluir restricciones dinámicas que permitan aumentar el grado de efectividad sobre la plataforma experimental.

FINANCIAMIENTO

Artículo de investigación derivado del proyecto de investigación titulado: Generación Automática de Marchas para un Robot Humanoide (Bioloid). Financiado por la Universidad de Pamplona. Fecha de Inicio: 22 Agosto de 2016. Fecha de terminación: 15 Diciembre de 2017.

REFERENCIAS

- [1] Z. Mohamed y G. Capi, "Development of a new mobile humanoid robot for assisting elderly people," *Procedia Engineering*, vol. 41, no. Iris, pp. 345–351, 2012. <http://dx.doi.org/10.1016/j.proeng.2012.07.183>
- [2] G. Wiedebach *et al.*, "Walking on partial footholds including line contacts with the humanoid robot atlas," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 1312–1319, 2016. <http://dx.doi.org/10.1109/HUMANOID.2016.7803439>
- [3] B. Ding, A. Plummer y P. Iravani, "Investigating Balancing Control of a Standing Bipedal Robot With Point Foot Contact," *IFAC-PapersOnLine*, vol. 49, no. 21, pp. 403–408, 2016. <http://dx.doi.org/10.1016/j.ifacol.2016.10.587>
- [4] E. Ackerman y E. Guizzo, "Its Wheel-Leg Robot: 'Best of Both Worlds,'" *IEEE Spectrum*, 2017. [En línea]. Disponible en: <https://spectrum.ieee.org/automaton/robotics/humanoids/boston-dynamics-handle-robot>
- [5] Y. Hosoda, S. Egawa, J. Tamamoto, K. Yamamoto, R. Nakamura y M. Togami, "Basic design of human-symbiotic robot EMIEW," in *IEEE International Conference on Intelligent Robots and Systems*, no. c, pp. 5079–5084, 2006. <http://dx.doi.org/10.1109/IROS.2006.282596>
- [6] B. Henze, A. Dietrich y C. Ott, "An Approach to Combine Balancing with Hierarchical Whole-Body Control for Legged Humanoid Robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 700–707, 2016. <http://dx.doi.org/10.1109/LRA.2015.2512933>
- [7] Y. Liu, P. M. Wensing, J. P. Schmiedeler y D. E. Orin, "Terrain-Blind Humanoid Walking Based on a 3-D Actuated Dual-SLIP Model," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1073–1080, 2016. <https://doi.org/10.1109/LRA.2016.2530160>

- [8] M. W. Clearfield, "Learning to walk changes infants' social interactions," *Infant Behavior and Development*, vol. 34, no. 1, pp. 15–25, 2011. <http://dx.doi.org/10.1016/j.infbeh.2010.04.008>
- [9] E. P. Shaw *et al.*, "Measurement of attentional reserve and mental effort for cognitive workload assessment under various task demands during dual-task walking," *Biological Psychology*, vol. 134, no. January, pp. 39–51, 2018. <https://doi.org/10.1016/j.biopsycho.2018.01.009>
- [10] K. B. Lee, H. Myung y J. H. Kim, "Online multiobjective evolutionary approach for navigation of humanoid robots," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5586–5597, 2015. <https://doi.org/10.1109/TIE.2015.2405901>
- [11] D. A. López, J. E. Hernández y C. A. Peña Cortes, "Advances in the control of bipedal platforms using the system," *Revista Colombiana de Tecnologías de Avanzada*, vol. 2, pp. 117–124, 2013. <https://doi.org/10.24054/16927257.v22.n22.2013.419>
- [12] K. Teachasrisaksakul, Z. Q. Zhang, G. Z. Yang y B. Lo, "Imitation of dynamic walking with bsn for Humanoid robot," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 794–802, 2015. <https://doi.org/10.1109/JBHI.2015.2425221>
- [13] A. Barrientos, L. Peñin, C. Balager y R. Aracil, *Fundamentos de Robótica*, 2nd ed. Madrid: McGraw-Hill, 2007.
- [14] E. Luhta, "Walk Cycles," in *How to Cheat in Maya 2010*, Boston: Focal Press, pp. 177–221, 2010. <https://doi.org/10.1016/B978-0-240-81188-8.50008-4>
- [15] K. H. Koch, K. Mombaur y P. Soueres, "Optimization-based walking generation for humanoid robot," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 45, no. 22, pp. 498–504, 2012. <http://dx.doi.org/10.3182/20120905-3-HR-2030.00189>
- [16] J. V. Nunez, A. Briseno, D. A. Rodriguez, J. M. Ibarra y V. M. Rodriguez, "Explicit Analytic Solution for Inverse Kinematics of Bioloid Humanoid Robot," in *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, pp. 33–38, 2012. <https://doi.org/10.1109/SBR-LARS.2012.62>
- [17] P. Wawrzynski, J. Mozaryn y J. Klimaszewski, "Robust estimation of walking robots velocity and tilt using proprioceptive sensors data fusion," *Robotics and Autonomous Systems*, vol. 66, pp. 44–54, 2015. <http://dx.doi.org/10.1016/j.robot.2014.12.012>
- [18] L. W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Maryland, USA: Wiley, 1999. URL: <http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471325937.html>
- [19] J. Zhao, Z. Feng, F. Chu y N. Ma, "A Brief Introduction to Screw Theory," in *Advanced Theory of Constraint and Motion Analysis for Robot Mechanisms*, pp. 29–79, 2014. <https://doi.org/10.1016/B978-0-12-420162-0.00002-3>

Cristian David Villate Martínez received the Mechatronic engineering degree from the Pamplona University, Pamplona, Colombia, in 2017. He has been a researcher of the SIARC-A&C (Research Group in Automation and Control) since 2015. <http://orcid.org/0000-0002-5427-730X>

César Augusto Peña Cortés is currently a full professor in the Department of Mechanical, Mechatronics and Industrial Engineering at the University of Pamplona (since 2004). He is part of the Automation and Control research group. He holds a PhD in Automation and Robotics from the Universidad Politécnica de Madrid, Spain (2006). He has a Master's degree in Electronics and Computer Engineering from the Universidad de los Andes, Colombia (2003) and a professional degree as an Electromechanical Engineer from the Pedagogical and Technological University of Colombia (2001). His research topics revolve around service robots, artificial vision and neurosignals, in which he has several publications in journals and congresses lectures. <https://orcid.org/0000-0003-4148-2168>

Oscar Eduardo Gualdrón Guerrero received his PhD degree in Electronic Engineering from the Rovira I Virgili university, Tarragona, Spain (2006). He is currently the research manager at the University of Pamplona and a full professor in the Department of Electronic Engineering (since 2007). He is part of the Automation and Control research group and the Multisensorial System research group. <https://orcid.org/0000-0002-7854-6842>