

Aprendizaje automático de modos de caminar para un robot humanoide

Francisco M. Rivas, José M. Cañas y Juan González

Abstract—Uno de los problemas abiertos en robots humanoides es el modo de coordinar sus múltiples actuadores para dotarlos de capacidad de movimiento robusto y ágil. El artículo presenta una técnica que encuentra automáticamente modos de caminar estables y rápidos para el humanoide Nao. El trabajo modeliza la caminata del robot como ondas acopladas reduciendo el espacio de parámetros manteniendo a la vez la representatividad de gran diversidad de movimientos. Una función de calidad combina varios criterios y permite buscar automáticamente en ese espacio de parámetros la caminata óptima. Se han realizado numerosos experimentos tanto en simulación como con el robot real que validan la propuesta.

I. INTRODUCCIÓN

Una línea pujante dentro de la robótica en los últimos años son los robots humanoides. Numerosos grupos de investigación centran sus trabajos en ellos. Por ejemplo en la competición científica internacional RoboCup dos ligas los utilizan como plataforma, la liga standard (robot Nao) y la específica de humanoides. Además, casi todas las grandes firmas japonesas han construido algún prototipo, tal vez pensando en un incipiente mercado de robots de servicio en los hogares y en que su apariencia humana les hace más fácilmente aceptables por las personas. Especialmente relevante es el trabajo realizado por Honda en su proyecto ASIMO (*Advanced Step in Innovative MObility*), en el que han conseguido que su humanoide corra a velocidades por encima de 6 km/h.

Pese a los grandes avances recientes, la locomoción de los humanoides sigue siendo un problema abierto, aún se está muy lejos de la flexibilidad, robustez y plasticidad de los movimientos naturales de las personas, por ejemplo.

La generación de modos de caminar en robots humanoides es parte de un problema más general: la coordinación de actuadores en robots de n articulaciones. Dado un robot con n articulaciones y una determinada morfología (humanoide, cuadrúpedo, hexápodo, ápedo...) el problema es encontrar las funciones que establecen la posición de cada articulación para que el robot pueda desplazarse. En la literatura se pueden encontrar tres enfoques diferentes: clásico, tablas de control y bio-inspirado. En el enfoque clásico se establecen funciones de trayectoria para los extremos de las patas y se calculan las posiciones de los actuadores mediante cinemática inversa. La realización de estos controladores requiere de un modelo matemático exhaustivo del robot y

en general el coste computacional será alto. Un enfoque diferente son las tablas de control, propuestas por Yim [1] para la locomoción de sus primeros robots modulares. Estas tablas almacenan el vector de posición de las articulaciones para cada instante. El controlador recorre la tabla enviando las posiciones a los actuadores. Este controlador es sencillo y se puede realizar usando microcontroladores de gama baja. Sin embargo, es muy poco flexible. Para cambiar el movimiento hay que recalcular la tabla.

El tercer enfoque es el bio-inspirado, en el que las funciones de posición se obtienen a partir de modelos provenientes de la naturaleza. IJspeert[2] fue pionero en la aplicación de los generadores centrales de patrones (CPGs) de la lamprea para la locomoción de robots, logrando generar posteriormente movimiento en su salamandra robótica Amphibot[3]. Los CPGs son grupos de neuronas especializadas que producen ritmos para controlar la actividad muscular de los seres vivos. Kurakawa et al. [4] utilizaron como base el modelo de oscilador de medio centro (*half center oscillator*) de Matsuoka[5] para la generación de ritmos en el robot modular M-tran III, que puede adoptar diferentes morfologías. Herrero et al. [6] empleó un CPG con la misma arquitectura pero con una dinámica más rica, tanto de la neurona como de la sinapsis, para la locomoción de robots gusanos en una dimensión. Otros autores utilizan redes neuronales en vez de modelos específicos de CPGs. Tal es el caso de Avinash et al. [7] que entrena las redes mediante algoritmos genéticos para generar la locomoción de robots de tipo serpiente/gusano.

Los CPGs tienen la propiedad de que en régimen permanente se comportan como osciladores de frecuencia fija por lo que Gonzalez-Gomez[8] propuso un modelo simplificado de osciladores sinusoidales con el que logró la locomoción de una serpiente robótica en dos dimensiones, implementando cinco modos de caminar diferentes y obteniendo unos movimientos suaves y naturales. Esta idea de funciones simplificadas está dando muy buenos resultados, sobre todo por la posibilidad de su realización en hardware de gama baja. Prieto et al. [9] aplicó este modelo para la generación de caminatas en línea recta de un robot cuadrúpedo de 12 articulaciones. Aunque los parámetros los estableció empíricamente mediante prueba/error, fue suficiente para conseguir movimientos más estables y suaves que los alcanzados en prototipos anteriores.

La técnica más empleada para la locomoción de robots humanoides ha sido durante muchos años el ZMP (Zero

Francisco M. Rivas y José M. Cañas trabajan en la Universidad Rey Juan Carlos

Juan González trabaja en la Universidad Carlos III

Moment Point), en la que se calcula la trayectoria del centro de masas para que la caminata sea estable. Robots tan complejos como ASIMO o HRP utilizan este algoritmo. Sin embargo, para funcionar correctamente es necesario realizar un modelado muy preciso tanto del robot como de los actuadores. Por ello, cada vez más autores están aplicando modelos bioinspirados en los humanoides. Endo et al. [10] aplicaron con éxito un modelo simple de CPG a la locomoción de un bípido de 5 grados de libertad y posteriormente lo refinaron para lograr el movimiento de uno completo (38 DOF) mediante aprendizaje[11]. Asa et al[12] utilizaron la flexibilidad de los CPGs para realizar una transición entre locomoción bípeda y cuadrúpeda, de manera que si se perdía la estabilidad en modo bípedo, el robot podría continuar caminado con cuatro patas. Acosta et al[13] han usados CPGs para crear locomoción estable, probada tanto en simulación como en un robot real usando un giróscopo. El enfoque ZMP en combinación con CPGs ha sido propuesto por Or et al [14] para el control en tiempo real de un robot humanoide con columna vertebral flexible.

Una vez presentado el contexto de este artículo y el estado del arte en la sección II se describe el modelo de movimiento que proponemos para robots humanoides. Seguidamente presentamos una manera automática de evaluar diferentes modos de caminar tentativos y el algoritmo de búsqueda que hemos desarrollado para encontrar una caminata rápida y estable. En la sección IV ajustamos el modelo genérico al robot Nao, con el que hemos realizado numerosos experimentos, y analizamos la influencia de varios parámetros en la calidad de la caminata. Finalmente resumimos las conclusiones alcanzadas.

II. MODELO DE MOVIMIENTO CON ONDAS ACOPLADAS

La marcha de un robot humanoide es similar a la forma de andar de una persona humana. Los humanoides tienen 2 patas con 3 articulaciones en cada pata: la cadera, la rodilla y el tobillo (Figura 1). La marcha de un robot bípedo, como la humana, es periódica. Conseguimos desplazarnos encadenando cierto número de *pasos*. Si nos movemos a una velocidad constante estos pasos que componen la marcha son exactamente iguales. Por ello simplificaremos la marcha de un robot humanoide en pasos. Otra característica importante es la simetría: para conseguir una marcha correcta y lineal el movimiento de una pata es simétrico al de la contraria pero con un cierto retardo, exactamente la mitad del tiempo que dura un paso. Para que la marcha sea lo suficientemente estable y el robot no se caiga todos los actuadores que intervienen deben estar correctamente acompasados y sincronizados, en definitiva, acoplados.

Una manera de modelizar un paso es almacenar la secuencia de posiciones intermedias de todas las articulaciones que se emplean en la realización de un paso, generando las ya mencionadas tablas de control [1]. La repetición de esas posiciones a lo largo del tiempo,

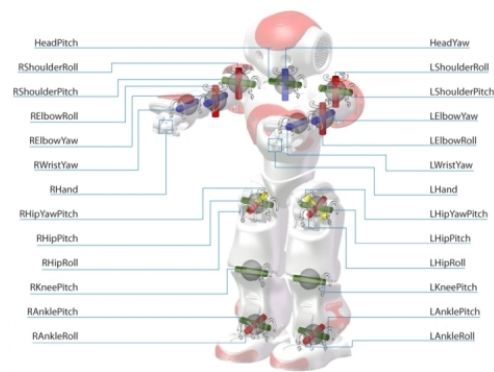


Fig. 1. Actuadores del humanoide

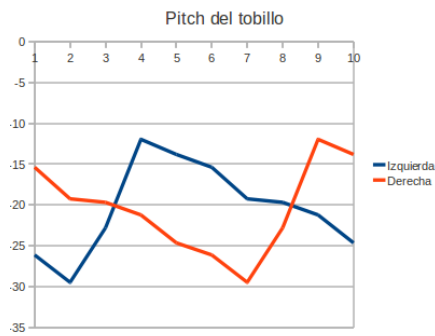
como si fueran fotogramas de una película, genera el movimiento deseado. Por ejemplo en la Figura 2 se aprecia las secuencias seguidas por varias articulaciones mientras el robot daba un paso.

Esta forma de andar ofrece una difícil parametrización, únicamente podemos variar la velocidad de forma indirecta, modificando el intervalo entre fotogramas. Para generar formas distintas de locomoción tendríamos que modificar el valor de los actuadores en cada fotograma, tarea muy compleja teniendo en cuenta que en la marcha del robot intervienen 12 actuadores (6 en cada pata: Pitch cadera, Yaw cadera, Roll cadera, Pitch rodilla, Roll tobillo y Pitch tobillo), suponiendo que utilizamos movimientos de 10 fotogramas serían *120 parámetros* que tenemos que fijar para conseguir una nueva forma de andar.

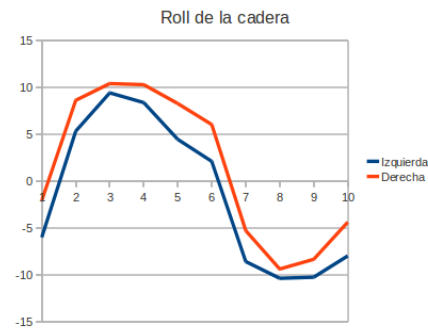
La parametrización de un paso con 120 parámetros es excesiva para buscar en el espacio de las posibles caminatas. En este trabajo proponemos una forma de disminuirlos significativamente para llegar a un espacio que sí sea factible de explorar de modo automático. Siguiendo la línea de los generadores centrales de patrones (CPG), por cada actuador que interviene en la locomoción conseguimos una función que defina su movimiento en cada paso. Así, los parámetros no dependen del número de fotogramas que componen el movimiento sino de los parámetros de esa función característica, por ejemplo sinusoidales.

Como podemos ver en la Figura 2 los movimientos que describen los actuadores no corresponden a ondas sinusoidales en la mayoría de los casos. Sin embargo, trataremos de modelizar estos recorridos con funciones base que queden caracterizadas con los mismos parámetros que una onda sinusoidal: frecuencia (ω), amplitud (A_0), fase (β) y desplazamiento (γ), correspondiendo a la función $a(t) = A_0 * \sin(\omega t + \beta) + \gamma$. Por cada actuador utilizaremos 4 parámetros.

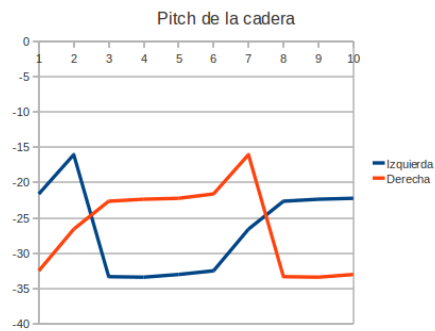
De los 12 actuadores involucrados en la caminata obviaremos los *yaw* de las caderas izquierda y derecha. Estos *yaw* controlan únicamente la rotación de la cadera, es



(a)



(b)



(c)

Fig. 2. Gráficas de actuadores que intervienen en la marcha

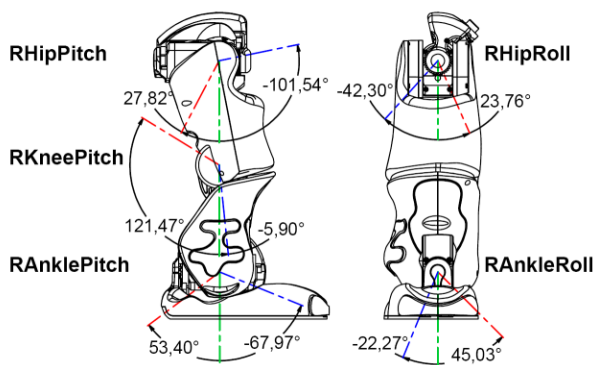


Fig. 3. Actuadores involucrados en la caminata

decir la apertura de las piernas respecto de la verticalidad del tronco a la hora de andar. Como queremos que el robot ande con las piernas cerradas y lo más erguido posible para obtener una mayor estabilidad éstos estarán fijados siempre a 0. Basta por tanto fijar valores para los 10 actuadores que intervienen en la marcha. Cada uno tendrá una función parametrizada en frecuencia, amplitud, fase y desplazamiento, con lo que tendremos 40 parámetros para generar una marcha en el robot humanoide.

La frecuencia de todos los actuadores debe ser la misma ya que el movimiento del paso debe empezar y terminar al mismo tiempo para todos ellos. Así tendremos 3 parámetros independientes en cada función (fase, amplitud, y desplazamiento) y un parámetro común en todos que es la frecuencia, por lo que tendremos realmente 31 parámetros.

En las caminatas rectas el paso del robot es simétrico por lo que podemos obviar uno de los lados del robot. Podemos fijar el movimiento del lado izquierdo desde el derecho sumándole un desfase fijo a la función que define su movimiento. Con ello reducimos el número de parámetros a 16: 5 actuadores a 3 parámetros por actuador más la frecuencia común.

Fijando el desfase de cadera como referencia y definiendo el desfase del resto de articulaciones de modo relativo a él, nos ahorramos un parámetro. Esa referencia marca la posición inicial del paso, pero es irrelevante respecto a la búsqueda.

Otra simplificación significativa viene de fijar la dependencia directa en los actuadores que realizan el *balanceo* en la marcha del robot. El balanceo es el movimiento que realiza el robot para compensar el peso a la hora de levantar una de las patas durante la marcha (Figura II). En él intervienen únicamente el *roll* de la cadera y el *roll* del tobillo. Variar las fases de estos movimientos significa que la marcha no esté acompasada y sea descoordinada por lo que podemos fijar una dependencia directa con la fase de la cadera de forma que sólo varíen las fases de los *roll* cuando varíe la fase de la cadera. Para conseguir un balanceo más suave o más pronunciado fijamos un movimiento base con unas amplitudes determinadas y multiplicaremos el valor de estas amplitudes por una constante que llamamos *amplitud de balanceo*. Con esto caracterizamos todo el movimiento del balanceo con ese único parámetro. Así conseguimos eliminar las dos amplitudes, que dependerán del nuevo parámetro, las dos fases, que serán directamente dependientes de la fase del *pitch* de la cadera para que el balanceo sea siempre acompasado y los dos desplazamientos, ya que al igual que las amplitudes serán fijas y situarán a los actuadores en la base del balanceo. Aplicando esta idea reducimos el número de parámetros de nuestra caminata a los 10 de la tabla I.

Parámetro	min	max	niveles
1. Frecuencia (ω), común para todos los actuadores	0	-	-
2. Amplitud del pitch de la cadera	0	62	62
3. Desplazamiento (γ) del pitch de la cadera	-100	25	125
4. Amplitud del pitch de la rodilla	0	65	65
5. Fase (β) del pitch de la rodilla	-2π	2π	12
6. Desplazamiento (γ) del pitch de la rodilla	0	130	130
7. Amplitud del pitch del tobillo	0	60	60
8. Fase (β) del pitch del tobillo	$-\pi$	π	6
9. Desplazamiento vertical (γ) del pitch del tobillo	-75	45	120
10. Amplitud de balanceo	0	100	100

TABLE I

PARÁMETROS FINALES DE LA MODELIZACIÓN DEL MOVIMIENTO

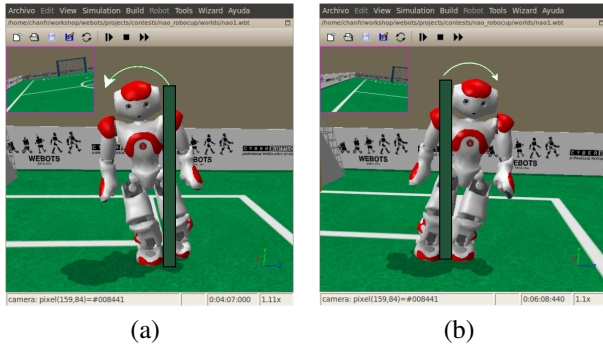


Fig. 4. Balanceo en la forma de andar parametrizada

III. BÚSQUEDA AUTOMÁTICA BASADA EN FUNCIÓN DE COSTE

Con la modelización explicada en la sección anterior conseguimos representar un gran número de formas de andar. No cubre todas las posibles caminatas del humanoide, pero sí nos ofrece una gran variedad donde buscar. Para hacernos una idea, discretizando los rangos de los 10 parámetros a un subconjunto acotado de valores (para cada parámetro los de la columna “niveles” en la tabla I) y fijando la frecuencia fundamental tendríamos aproximadamente 2^{51} posibles caminatas.

De todas las caminatas que describe nuestro modelo hay muchas de ellas que no tienen sentido, nos llevan a la caída del robot, otras generan pasos largos y lentos o rápidos con pasos cortos, etc. La mejor forma de realizar la búsqueda de la marcha óptima es realizando una búsqueda sistemática que recorra todas las posibilidades, sin embargo esta opción no es factible debido al gran número de caminatas que es capaz de generar nuestro modelo. En lugar de eso hemos utilizado un algoritmo que explora ese enorme espacio de manera muestreada, guiándose por los valores de cierta función de calidad.

III-A. Evaluación de una caminata

Para la evaluación de caminatas no utilizamos ningún método analítico debido a la gran complejidad que supone.

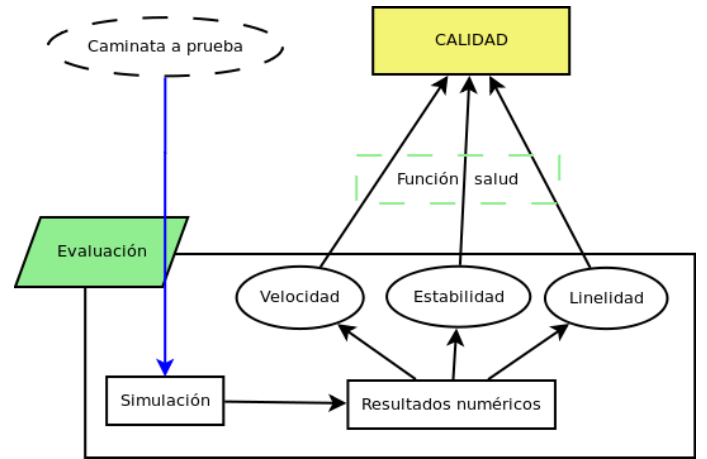


Fig. 5. Evaluación de la caminata basada en simulador.

Utilizamos un simulador en el que desplegaremos el modelo durante un cierto tiempo y caracterizamos numéricamente si la caminata realizada es buena o mala observando cualidades como estabilidad, velocidad y linealidad del movimiento generado (Figura 5).

Estas cualidades las medimos cuantitativamente a partir de datos de la simulación concreta de cada caminata, con características numéricas. A partir de ahí, combinamos todos estos factores en una única función de salud que indica la calidad final de esa caminata.

Gracias al software realizado en este trabajo, el simulador almacena la trayectoria generada por cada caminata y analizando automáticamente esa trayectoria podemos caracterizar las cualidades de la caminata. La *velocidad* se mide explícitamente como *distancia* recorrida durante el *tiempo* de simulación, entre el punto inicial y el final. La *linealidad* se mide como desviación final respecto de la trayectoria recta ideal, distancia entre el punto final ideal y el conseguido por la caminata. La *estabilidad* se mide iterativamente como desviación acumulada respecto de la trayectoria recta ideal en todo momento, midiendo los bamboleos alrededor de la recta ideal, tanto por su izquierda como por su derecha. Si durante la realización del movimiento el robot se cae no se evaluará la función salud y se asigna calidad nula al movimiento.

Hemos probado múltiples maneras de combinar estos tres criterios. Finalmente nos hemos decantado por la ecuación (1). La combinación de factores mediante multiplicación y los exponentes elegidos se han mostrado suficientemente discriminantes en los experimentos. Se premia la distancia recorrida frente a cualquier otro parámetro porque conseguir una forma de andar en la que el robot de pasos largos y sea capaz de moverse de forma rápida y estable es lo más complicado. Damos tanto peso a la distancia recorrida porque las diferencias de este valor entre unas caminatas y otras es mínimo.

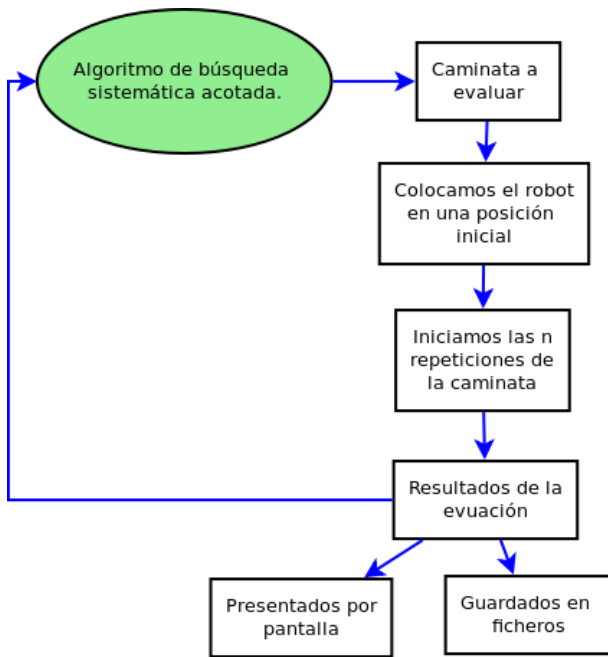


Fig. 6. Secuencia de evaluación.

$$salud = \sqrt{\frac{distancia^3}{tiempo}} \cdot \sqrt{\frac{1}{estabilidad}} \cdot linealidad \quad (1)$$

III-B. Búsqueda sistemática discreta

Una manera de explorar el espacio de posibles caminatas consiste en fijar los valores máximos y mínimos para cada parámetro así como un incremento en ese rango, que marca su nivel de discretización, es decir, cuántos posibles valores de cada parámetro se probarán. El tamaño de ese incremento marca la granularidad de la búsqueda.

Siguiendo los pasos de la Figura 6 se prueban todas las combinaciones de valores. Para cada una de ellas se sitúa al robot en una posición inicial erguida en el simulador y se le deja moverse un tiempo según marca la combinación de parámetros. La trayectoria y su calidad se miden y quedan almacenada, rastreando así en el espacio de posibles caminatas y quedándonos con aquellas de mayor calidad. Para hacer más fiable la medida de salud cada caminata se evalúa varias veces, conservando la calidad media de esas ejecuciones.

IV. EXPERIMENTOS

El robot concreto con el que hemos probado la técnica de aprendizaje automático de modos de caminar es el humanoide Nao (versión V3+) (Figura 8). La evaluación de las caminatas tentativas se realiza en el simulador Webots, que incluye soporte para este humanoide. Una vez aprendida, la caminata se ha probado y validado en el robot

real. Se tienen más detalles de todos experimentos llevados a cabo en [15].

Para realizar las pruebas hemos desarrollado dos componentes software dentro del entorno de programación *jderobot* (versión 4.3), que está orientado a componentes, es distribuido y multilinguaje. La aplicación robótica se plantea como un conjunto de componentes que interoperan entre sí. Los componentes están conectados mediante el middleware de comunicaciones *Ice*, pueden ejecutarse en la misma o diferentes máquinas y estar escritos en distintos lenguajes de programación. *Jderobot* incluye soporte para sensores láser, múltiples cámaras, robot Pioneer, varios cuellos mecánicos, simulador Gazebo, simulador Stage, etc.

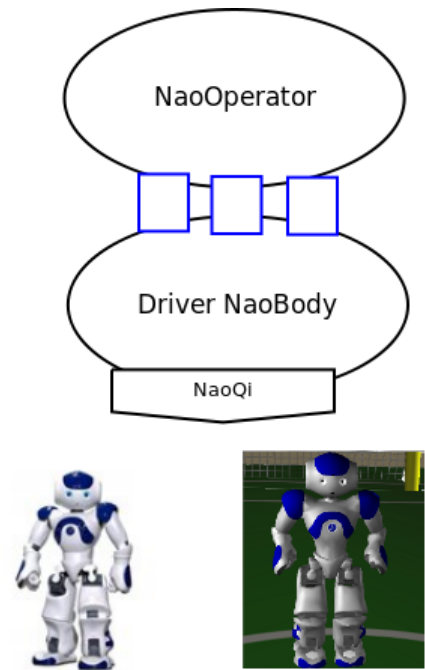


Fig. 7. Componentes software desarrollados.

Primero se ha desarrollado el componente *NaoBody*, que usa el middleware del fabricante (llamado *Naoqi*, versión 1.3.17) para acceder a los sensores y actuadores del humanoide y ofrecerlo a otros componentes. Incluye soporte tanto para conectarse al Nao real como al robot simulado en Webots. En el caso de usar el simulador, este componente ofrece también la posición verdadera del robot en el mundo simulado.

En segundo lugar, el componente *NaoOperator* permite teleoperar al robot a voluntad y realizar las búsquedas sistemáticas en el espacio de posibles caminatas. Se conecta a *NaoBody*, que a su vez está conectado al simulador. Este componente permite fijar los valores máximos y mínimos de cada parámetro, así como sus niveles de discretización, estableciendo con ello el espacio concreto de todas las posibles caminatas. Permite explorar todos

los parámetros simultaneamente o restringir la búsqueda a subespacios con sólo uno o dos parámetros libres.

El modelo genérico de caminata basado en ondas acopladas para un humanoide, descrito en la sección II, se adaptó al caso concreto del robot Nao. En primer lugar se eligió la función base para cada una de las articulaciones. Para escoger bien estas funciones base pusimos a caminar al robot con el software del fabricante y muestreamos las posiciones intermedias para cada actuador, consiguiendo los patrones de la Figura 2. Nuestro algoritmo utiliza esas funciones como base, cambiando su frecuencia, su amplitud, su desplazamiento y su desfase.

En segundo lugar, se fijaron los límites máximos y mínimos para cada grado de libertad, que se muestran en la tabla II. También se fijaron los niveles de discretización para cada parámetro dependiendo de su naturaleza. Aparecen en la columna de la derecha de dicha tabla.

Dejamos 45 días algoritmo ejecutando en un ordenador con procesador Intel E6600 dual core. El algoritmo encontró una manera de caminar estable y rápida, de unos 18 cm/segundo, que se puede apreciar en la Figura 8. Se puede ver el video con el robot caminando según los parámetros encontrados por el algoritmo en la web ¹.



Fig. 8. Caminata conseguida en el robot real

IV-1. Impacto de los cambios de amplitud y desplazamiento de la cadera en la calidad de la caminata: El pitch de la cadera supone uno de los actuadores fundamentales a la hora de la locomoción del robot, es el actuador encargado de fijar la verticalidad y de elevar cada una de las piernas al comenzar una zancada (Figura

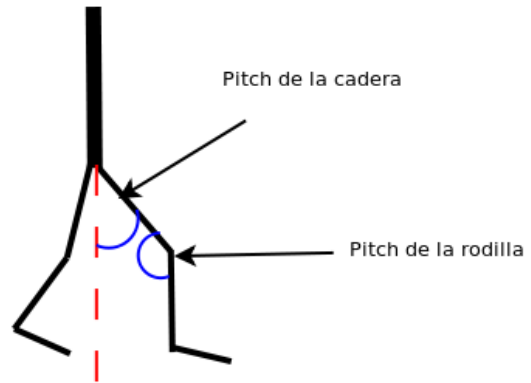


Fig. 9. Pitch de cadera y de rodilla son fundamentales en el paso

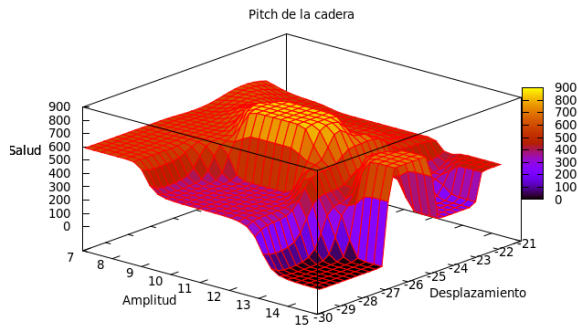
9). Como podemos ver en la gráfica de la figura 10(a), la calidad del movimiento disminuye si el desplazamiento es superior a 27 o inferior a 24. Ya que este actuador fija la verticalidad del Nao es normal que el rango para este actuador en el que la caminata es óptima sea muy reducido, si nos salimos de ese rango el robot andará inclinado hacia atrás o hacia delante, tal y como queda reflejado en la figura 10(b). Esta verticalidad puede ser compensada por los desplazamientos de los otros dos actuadores que intervienen en la marcha, por ejemplo, podemos compensar una inclinación frontal con una mayor flexión de la rodilla.

Con la amplitud sucede lo mismo, si realizamos una caminata con valores muy grandes, el robot, o bien se caerá, o bien no conseguirá un buen desplazamiento. Si la amplitud es pequeña el robot no es capaz de levantar las piernas por lo que no se caerá pero tampoco conseguirá avanzar, con una amplitud grande el robot sí que conseguirá levantar la pierna pero la levantará en exceso por lo que realizará una caminata con grandes desequilibrios o se caerá.

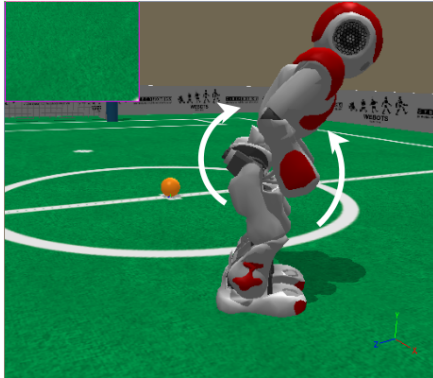
IV-2. Impacto de los cambios de amplitud y desplazamiento de la rodilla en la calidad de la caminata: En la Figura 11(a) se aprecia que no tiene sentido la evaluación de caminatas de este tipo con un valor mayor a, aproximadamente 41. Esto sucede porque a partir de este valor el Nao flexiona mucho las rodillas (figura 11(b)). Los motores que pueden compensar este desfase son el pitch de la cadera y el pitch del tobillo, aumentando sus valores, pero no varían ya que son tomados como valores fijos. El robot al comenzar el movimiento se caerá en la mayoría de los casos. Probablemente en una forma de andar donde estos valores estén ajustados sí tendría sentido buscar con un desplazamiento mayor, con lo que conseguiríamos una caminata en la que el robot anduviera agachado.

En lo referente a la amplitud de este actuador podemos ver que tanto para valores bajos como altos la calidad de la locomoción disminuye. Este hecho es razonable ya que la amplitud de oscilación del movimiento debe estar ajustada

¹<https://jderobot.org/index.php/Frivas-pfc-itis#BEST>

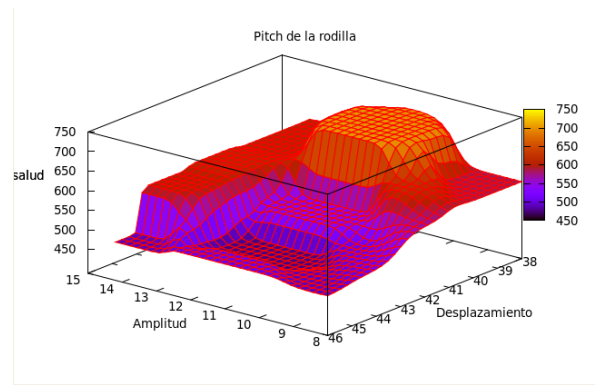


(a)

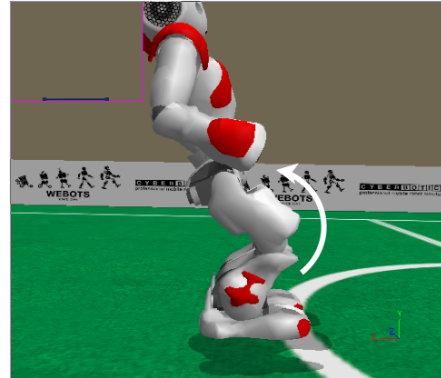


(b)

Fig. 10. Variaciones en pitch de cadera y efecto de valores altos.



(a)



(b)

Fig. 11. Variaciones en el pitch de rodilla y efecto de valores altos.

al tipo de locomoción, si existe muy poca amplitud el robot no será capaz de realizar el movimiento necesario para desplazar el peso del robot mientras que si este movimiento es excesivo perderá estabilidad por la elevada flexión de la rodilla y acabará en el suelo.

V. CONCLUSIONES

Se ha presentado un modo de aprendizaje automático de modos de caminar en robots humanoides. Está basado en la modelización de la caminata como ondas acopladas en sus distintos actuadores y una búsqueda sistemática en el espacio de parámetros de ese modelo.

La mejor caminata encontrada funciona correctamente, es decir, consigue que el robot avance de manera estable y rápida. Esta caminata ha sido validada tanto en simulación como en el robot real. La velocidad conseguida es similar a la del modo de caminar que propone el fabricante.

Pese a que se ha utilizado una búsqueda muy tosca, con una discretización relativamente gruesa de los parámetros, el algoritmo planteado en este trabajo preliminar ha igualado la velocidad conseguida por el fabricante. En la búsqueda por velocidades mayores el modelo propuesto permite búsquedas más finas aumentando los niveles de discretización de cada

parámetro, aunque el coste temporal de la búsqueda aumenta.

La búsqueda sistemática sin embargo es ciega y lenta. Al discretizar en incrementos los valores de los parámetros deja zonas del espacio sin explorar. Además es lenta por exhaustiva, porque efectivamente recorre todo el espacio, aunque sean zonas poco prometedoras. Una de las líneas en la que estamos continuando el trabajo descrito es utilizar un algoritmo genético para explorar el espacio de parámetros en lugar de la búsqueda sistemática. El algoritmo evolutivo puede emplear como función de salud la misma función de calidad que hemos definido. Este algoritmo evolutivo busca en el continuo de esos valores y su búsqueda es más refinada, más eficiente. Al rastrear de modo más preciso el espacio de caminatas posibles es probable que encuentre aún mejores modos de caminar y en menor tiempo.

Otra línea a explorar es ajustar las funciones base de cada de las articulaciones, las usadas en la modelización como ondas acopladas.

Una cuestión abierta es la extensión del aprendizaje a movimientos más generales, como giros, movimientos en arco, desplazamientos laterales, etc.

VI. AGRADECIMIENTOS

Este trabajo se ha realizado financiado parcialmente por la Comunidad de Madrid a través del proyecto RoboCity2030-II (S2009/DPI-1559).

REFERENCES

- [1] M. Yim, *Locomotion with unit-modular reconfigurable robot*. PhD thesis, Stanford University, 1995.
- [2] A. J. Ijspeert, *Design of artificial neural oscillatory circuits for the control of lamprey- and salamander-like locomotion using evolutionary algorithms*. PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 1998.
- [3] A. J. Ijspeert and A. Crespi, "Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA 2007)*, pp. 262–268, 2007.
- [4] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, and S. Murata, "Distributed self-reconfiguration of m-tran iii modular robotic system," *I. J. Robotic Res.*, vol. 27, no. 3-4, pp. 373–386, 2008.
- [5] M. Kiyotoshi, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological Cybernetics*, vol. 56, pp. 345–353, July 1987.
- [6] F. H. Carron, *Novel Central Pattern Generator elements for Autonomous Modular Robots*. PhD thesis, Escuela Politecnica Superior, Universidad Autonoma de Madrid, 2011.
- [7] L. M. Avinash Ranganath, Juan Gonzalez-Gomez, "A distributed neural controller for locomotion in linear modular robotic configurations," in *Proceedings of the 8th Workshop of RoboCity2030*, pp. 129–144, 2011.
- [8] J. Gonzalez-Gomez, *Robotica Modular y Locomocion: Aplicacion a Robots Apodos*. PhD thesis, Escuela Politecnica Superior, Universidad Autonoma de Madrid, 2008.
- [9] A. Prieto-Moreno, "Estudio de la locomocion de un robot cuadrupedo mediante la generacion de patrones biologicos," Master's thesis, Escuela Politecnica Superior, Universidad Autonoma de Madrid, 2007.
- [10] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng, "An empirical exploration of a neural oscillator for biped locomotion control," in *Proc. of ICRA, 2004*, pp. 3036–3042, 2004.
- [11] E. Gen, M. Jun, M. Takamitsu, N. Jun, and C. Gordon, "Learning cpg-based biped locomotion with a policy gradient method: Application to a humanoid robot," *Int. J. Rob. Res.*, vol. 27, pp. 213–228, February 2008.
- [12] A. K., I. K., and W. M., "Adaptive behavior to environment of a humanoid robot with cpg," in *SICE 2004 Annual Conference*, vol. 1, pp. 208–213 vol. 1, aug. 2004.
- [13] C. Acosta Calderon, R. Mohan, and C. Zhou, "Rhythmic locomotion control of humanoid robot," in *MICAI 2008: Advances in Artificial Intelligence* (A. Gelbukh and E. Morales, eds.), vol. 5317 of *Lecture Notes in Computer Science*, pp. 626–635, Springer Berlin / Heidelberg, 2008.
- [14] O. Jimmy, "A hybrid cpg-zmp controller for the real-time balance of a simulated flexible spine humanoid robot," *Trans. Sys. Man Cyber Part C*, vol. 39, pp. 547–561, September 2009.
- [15] F. R. Montero, "Caminata basada en ondas acopladas para el robot humanoide nao," Master's thesis, Escuela Técnica Superior Ingeniería Informática, Universidad Rey Juan Carlos, 2010.