

Diseño e implementación de controladores inteligentes en procesadores suaves para la caminata de un robot bípedo

Jorge A. Quiñones¹, Oscar Montiel-Ross¹, Roberto Sepúlveda¹, Oscar Castillo²

¹ Instituto Politécnico Nacional (CITEDI-IPN),
Centro de Investigación y Desarrollo de Tecnología Digital, Tijuana,
México

² Tijuana Institute of Technology, Tijuana,
México

arturo.quinones@uabc.edu.mx, {orross, rsepulvedac@ipn.mx}, ocastillo@tectijuana.mx

Resumen. Se presenta el diseño, construcción, y control inteligente para la caminata de un robot bípedo. Los controladores están basados en lógica difusa y redes neuronales artificiales. La estabilidad del robot se controla mediante la técnica conocida como "Zero Moment Point". Los controladores se implementaron en tecnología FPGA debido a la versatilidad que ofrecen para combinar técnicas de procesamiento secuencial con multiprocesadores suaves, y procesamiento concurrente mediante lenguajes de descripción de hardware, lo cual permite ejecutar las tareas en tiempo real dándole al robot un comportamiento más natural. Se muestran resultados experimentales.

Palabras clave. FPGA, cinemática inversa, redes neuronales, procesadores suaves, robot bípedo.

Design and Implementation of Intelligent Controllers in Soft Processors for the Walking of a Biped Robot

Abstract. We present the design, construction, and intelligent control for the walking of a biped robot. The controllers are based on fuzzy logic and neural network. The stability of the robot is controlled by the Zero Moment Point technique. The controllers were implemented into FPGA technology since its versatility to combine sequential processing for softcore multiprocessors systems as well as the use of concurrent processing using hardware description languages, which allows executing tasks in real time providing the robot

with a more natural behavior. We show experimental results.

Keywords. FPGA, inverse kinematics, neural networks, soft processors, biped robot.

1. Introducción

La idea de que los robots humanoides interactúen con el hombre ha sido uno de los mayores retos en el ámbito científico y tecnológico en la actualidad. Se espera que estas máquinas puedan realizar las actividades tediosas y peligrosas que se presentan en la vida cotidiana.

Obtener el modelo matemático de los robots bípedos es sumamente complicado debido al alto grado de no linealidad [8] y a la gran cantidad de parámetros necesarios en los controladores tradicionales. Por tal razón, son de gran interés las investigaciones que proponen soluciones alternas, planteadas con algoritmos de control innovadores mediante lógica difusa [11], redes neuronales [12], y algoritmos genéticos [3] para posicionamiento y control.

El estudio de robots bípedos manifiesta una gran complejidad en relación al control de su locomoción, debido a la dificultad de controlar el caminar y mantener el equilibrio generado por los movimientos de sus articulaciones, lo que representa intrínsecamente un sistema sumamente no lineal [2]. Otro aspecto importante

es la reacción dinámica producida por el contacto con la tierra, donde se genera una fricción con fuerza inercial que debe ser cero para mantener el equilibrio.

La gran mayoría de los trabajos de investigación sobre robots bípedos se centran en desarrollar controladores para que el robot no pierda la estabilidad al caminar, y es donde toma gran importancia el concepto de ZMP (Zero Moment Point) [10] o también conocido como CoP (Centro de presión) [4], que se define como un criterio de estabilidad ampliamente utilizado en el desarrollo de robots, por ejemplo, en el ASIMO creado por la empresa Honda Inc. [5], que representa uno de los humanoides más avanzados en la actualidad. En el ciclo de la caminata bípeda, el robot se apoya sobre una sola extremidad y en la planta del pie se forma el polígono de *Soporte Único*.

En este punto la fuerza de reacción con la tierra es cero (ZMP) [9], y siguiendo el ciclo cuando se apoya mediante los dos pies se le denomina polígono de *Doble Soporte*, por lo consiguiente el ZMP existe y el robot no sufre una caída mientras el centro de masa se encuentre dentro de dichos polígonos de soporte. En la caminata de robots bípedos existen dos formas de equilibrio: el caminado Estático [7] con movimientos lentos, sin estética y el caminado Dinámico, con gran eficiencia e incluso con una movilidad más natural [1].

En este trabajo mostramos el diseño y construcción del robot bípedo que utilizamos para realizar los experimentos, se presenta el diseño de los algoritmos de control inteligente desarrollados para la caminata del robot, principalmente basados en lógica difusa y redes neuronales artificiales (RNA). Estos controladores se implementaron en tecnología FPGA por la versatilidad que ofrecen para mezclar técnicas de procesamiento secuencial con multiprocesadores suaves, y procesamiento concurrente mediante lenguajes de descripción de hardware, lo cual permitirá ejecutar las tareas en tiempo real.

El contenido se encuentra organizado como sigue: En la Sección 2 se detalla la construcción, el hardware y la plataforma de cómputo utilizada en el robot bípedo. La Sección 3 describe la metodología para la obtención de la cinemática

inversa a partir de la cinemática directa mediante RNA. La Sección 4, presenta los controladores mediante lógica difusa para controlar la caminata bípeda, y finalmente, en la Sección 5, se detallan los experimentos realizados y los resultados obtenidos.

2. Configuración de la plataforma de hardware

Con el objetivo de realizar pruebas experimentales se diseñó en SolidWork, un robot bípedo construido de aluminio, ver Figura 1, con 12 grados de libertad totalmente actuado en sus articulaciones mediante servomotores digitales marca Dynamixel con un torque máximo de 26 kg.cm a 12V consumiendo 2.4A.

El servomotor, además de incorporar un reductor de engranajes, tiene un motor de corriente continua de precisión y un circuito de control con funciones de conectividad de red. La altura total del robot es 47 cm y un peso de 1.71 kg.



Fig. 1. Robot bípedo experimental JoKariT SocAr

Para llevar a cabo el control mediante el uso del cálculo del ZMP a base de sensores usamos (1) y (2); se colocan en los vértices de la superficie del pie, Figura 2, cuatro sensores de fuerza para así obtener las coordenadas ZMPx y ZMPy del

polígono de soporte:

$$ZMP_x = \frac{\sum_{i=0}^4 R_i \cdot X_i}{\sum_{i=0}^4 R_i}, \quad (1)$$

$$ZMP_y = \frac{\sum_{i=0}^4 R_i \cdot Y_i}{\sum_{i=0}^4 R_i}. \quad (2)$$

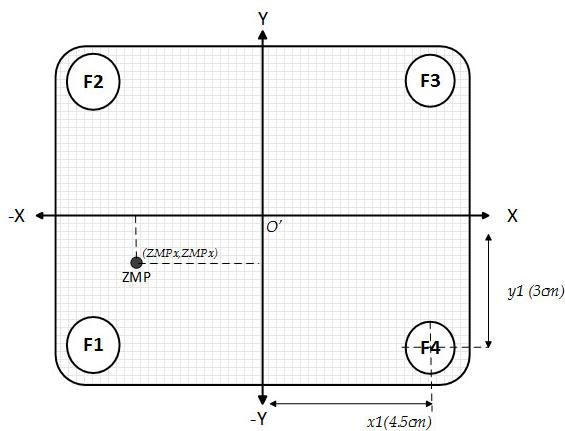


Fig. 2. Distribución de sensores en la planta del pie del robot

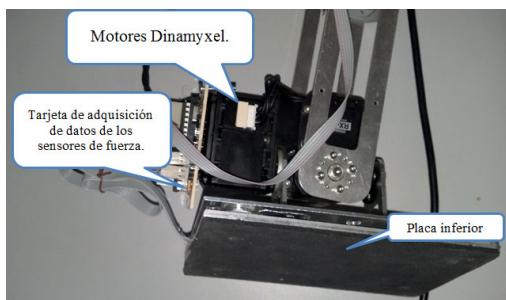


Fig. 3. Planta del pie del robot bápedo

Para calcular el ZMP fue necesario un controlador de señales digitales (DSC) de Microchip modelo 33FJ12GP201 a 50 Mhz en modo esclavo con comunicación I2C y cuatro sensores de fuerza

modelos Flexiforce 301 ubicados en la planta del pie del robot bápedo. Para eliminar el ruido introducido cuando hacen contacto las placas de metal que contienen los sensores de presión con la superficie del suelo, se implementó un filtro digital pasa bajas tipo FIR de ventana Chebyshev de orden 32, con una frecuencia de corte de 200 Hz, y una frecuencia de muestreo por canal de 1Khz (4 canales).

Para incorporar autonomía en el robot se utiliza como unidad de procesamiento el núcleo Microblaze, que es un "Procesador Suave" ejecutándose a una velocidad de 100 Mhz. Para implementar la lógica de su arquitectura de un procesador, usa como base los recursos de lógica programable existente en la FPGA de la serie Spartan 6 de Xilinx Inc., tal como se muestra en la Figura 4. Lo anterior es una gran ventaja, ya que se tiene la capacidad de poder personalizar los periféricos por software.

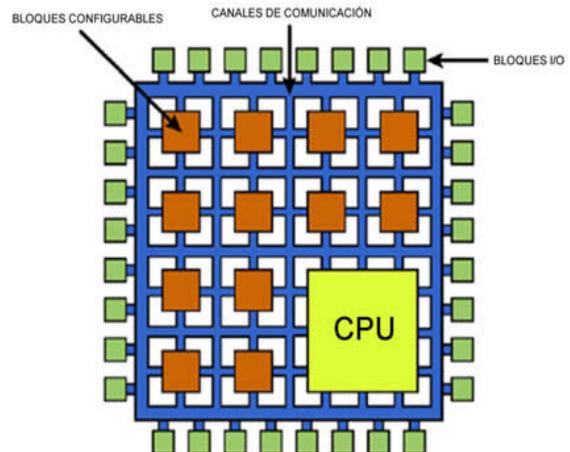


Fig. 4. Procesador suave

En la Figura 5 se ilustra el esquema general del sistema. En el procesador Microblaze se encuentra embebida la RNA que recibe a la entrada los vectores de datos correspondientes a la posición deseada en la planta del pie del robot bápedo, donde X, Y, Z se refiere a la posición y α , β , γ a su orientación en el plano cartesiano.

Una vez procesada la información en tiempo real, se obtiene como respuesta un vector con nueve datos de salida ($q_1, q_2, q_3, \dots, q_9$), que

pertenecen a la posición que debe mantener las articulaciones del robot, a su vez estos datos son enviados individualmente mediante el protocolo RS-232C, a los servomotores.

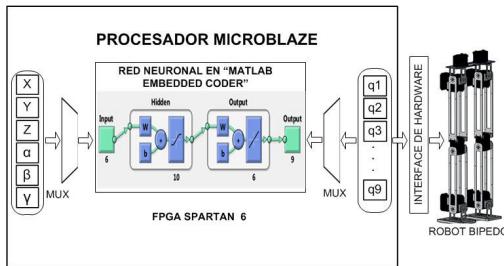


Fig. 5. Esquema general del sistema RNA

3. Implementación de la cinemática inversa del robot bípedo en FPGA

En este apartado se describe cómo resolver e implementar en una FPGA la cinemática directa e inversa, siendo de gran interés para la planificación de las trayectorias del robot bípedo.

Tabla 1. Parámetros DH de la cadena cinemática el pie derecho del robot bípedo

qD	θ	d	a	α
1	$\theta_1 + \pi/2$	0	0	0
2	$\theta_2 + \pi/2$	-L21	0	$-\pi/2$
3	$\theta_3 + \pi/2$	0	0	$-\pi/2$
4	$\theta_4 - \pi/2$	-L23	0	$-\pi/2$
5	θ_5	0	L34	$-\pi/2$
6	θ_6	0	L45	0
7	θ_7	0	L56	0
8	θ_8	0	L67	$\pi/2$
9	θ_9	0	L78	0

3.1. Cinemática directa

La cinemática directa, determina las coordenadas de posición en función de los ángulos que forma cada articulación, es decir:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \underset{f}{\rightarrow} (\theta_1, \theta_2, \theta_3, \theta_4, \dots, \theta_n).$$

Para resolver la cinemática directa, es común recurrir a relaciones geométricas, pero cuando los grados de libertad aumentan se utiliza un método sistemático de matrices de transformación homogénea, donde Denavit - Hartenberg emplean un procedimiento ordenado para pasar de un eslabón a otro mediante cuatro transformaciones básicas.

A partir de ésto conseguimos las coordenadas y los cuatro parámetros de nuestra configuración de la cadena cinemática, del pie izquierdo y derecho del robot, ver Figura 6, con ésto obtenemos las Tablas 1 y 2.

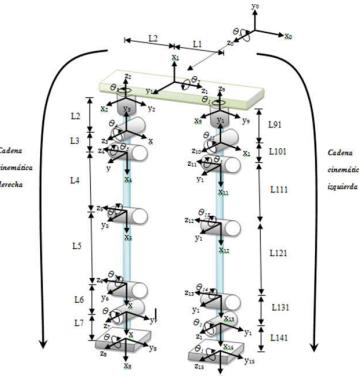


Fig. 6. Cadena cinemática del robot bípedo

Tabla 2. Parámetros DH de la cadena cinemática del pie izquierdo del robot bípedo

qI	θ	d	a	α
1	$\theta_1 + \pi/2$	0	0	0
2	$\theta_2 + \pi/2$	-L19	0	$-\pi/2$
10	$\theta_{10} + \pi/2$	0	0	$-\pi/2$
11	$\theta_{11} - \pi/2$	-L910	0	$-\pi/2$
12	θ_{12}	0	L1011	$-\pi/2$
13	θ_{13}	0	L1112	0
14	θ_{14}	0	L1213	0
15	θ_{15}	0	L1314	$\pi/2$
16	θ_{16}	0	L1415	0

Para llevar a cabo la simulación se utiliza el Toolbox de Matlab de Robótica desarrollado por Peter Corke. Esta herramienta proporciona muchas funciones que son útiles para el estudio y simulación de diferentes tipos de robots,

incluyendo la cinemática directa e inversa, la dinámica y la generación de trayectorias.

El toolbox mantiene la notación y usa los métodos generales para la representación de la cinemática y dinámica que son pasados como parámetros en las funciones escritas en el lenguaje de Matlab. Obtenidos los parámetros DH del análisis de la cinemática directa del robot bípedo, se ingresan en la función del Toolbox llamada "link" que relaciona y crea un objeto con las seis articulaciones de la cinemática de una pierna y sus respectivos eslabones necesarios para la simulación.

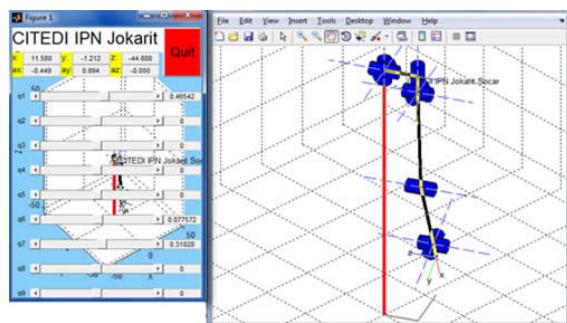


Fig. 7. Simulación de la cinemática directa

Representado el robot bípedo como se muestra en la Figura 7, se evalúan las posiciones de las articulaciones (q) y gráficamente se observa la posición de cada eslabón y la posición del efecto final (Planta del pie del robot).

3.2. Cinemática inversa

Por el contrario, la *cinemática inversa*, determina los ángulos que deben formar las articulaciones para poder llevar el efecto final a la posición deseada denotada por:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \theta_n \end{bmatrix} = \underset{f}{\rightarrow}^{-1} (x, y, z).$$

Para conseguir la cinemática inversa es común recurrir a la solución analítica cuando se refiere a robots de pocos grados de libertad, para obtener una solución cerrada, en este caso para doce grados de libertad divididas en dos cadenas cinemáticas resulta muy difícil, si no es que imposible, la obtención de la misma, por lo cual consideramos recurrir a las redes neuronales como se ilustra en la (Figura 8).

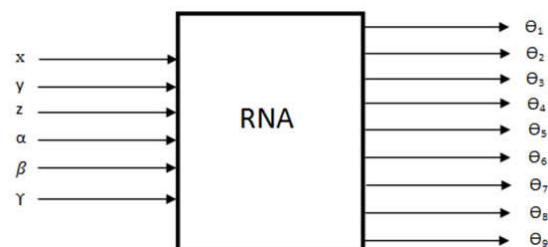


Fig. 8. Relación de entradas y salidas de la RNA para la resolución de la Cinemática Inversa

La estructura de la red neuronal consta de seis entradas donde X, Y, Z corresponden a la posición deseada del efecto final, y α, β, γ representan su orientación; en la salida se cuenta con nueve términos, donde se toman los datos que producen la posición de cada articulación y así producir la posición final deseada. La RNA también tiene 10 capas ocultas y se entrenaron en el toolbox de RNA de Matlab, mediante 75639 datos calculados de la cinemática directa. En la Tabla 4 se muestran los parámetros de entrenamiento de la red.

Tabla 3. Parámetros de entrenamiento de la red neuronal para la cinemática inversa

Parámetros de la red	Valor
Épocas	265
Error cuadrático medio	4.44×10^{-8}
Tamaño de la matriz de entrenamiento de entrada	9x75639
Tamaño de la matriz de entrenamiento de salida	6x75639
Método de entrenamiento	L-M
Función de activación (Ocultas/Salida)	tansig/purelin

La duración del entrenamiento fué de tres horas y media, calculando 265 épocas para dos matrices de entrenamiento de 6x75639 (Entradas) y 9x75639 (Salidas) mediante el método de Levenberg-Marquardt y una función de activación sigmoide y lineal, obteniendo un error mínimo cuadrado de 4×10^{-8} que implica un error aproximado de 1 cm en el efecto final.

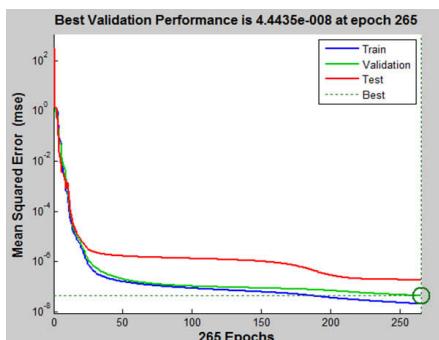


Fig. 9. Error cuadrático medio

3.3. Implementación y metodología de la cinemática inversa en hardware

Para la implementación de la cinemática inversa del robot bípedo, primero se relacionan las articulaciones y eslabones al plano cartesiano para extraer los parámetros de DH y a su vez se crea el objeto del robot con la función “link” y se resuelve la cinemática directa con el comando “fkine” usando el toolbox de Robótica de Peter Corke para crear un vector de datos con diferentes posiciones.

Con los resultados obtenidos de la cinemática directa se crea un patrón de entrada y entrenamos una RNA con el Toolbox de Matlab, comparamos y validados ingresando diversos datos a los patrones de entrenamiento y la red responde con la cinemática inversa. Como siguiente paso exportamos la RNA a un bloque de Simulink, donde se hace uso de la herramienta “Embedded Coder” que convierte nuestro bloque de Simulink, con nuestra RNA previamente traducida a código C ANSI.

Para compilar los archivos generados en C, se crea y se personaliza en la herramienta XPS de Xilinx, la arquitectura de hardware que incluye el procesador Microblaze y los periféricos necesarios para procesar nuestra RNA. Posteriormente se exporta la arquitectura a la herramienta SDK de Xilinx para editar, compilar, depurar, y descargar a la FPGA los archivos generados en C que albergan la RNA. En Figura 10 se ilustra el proceso para la implementación de la RNA en el robot bípedo.

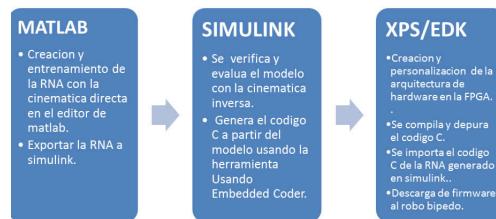


Fig. 10. Etapas del proceso para implementar RNA en procesador suaves

4. Implementación de controladores difusos para la caminata bípeda

Para garantizar la caminata bípeda es necesario mantener el equilibrio dinámico de robot bípedo en su desplazamiento sin que caiga, por lo que es necesario utilizar estrategias de control para garantizar la estabilidad en la trayectoria deseada. En esta sección se muestran el uso de controladores basados en lógica difusa lograr este objetivo.

4.1. Implementación del controlador difuso para el seguimiento de trayectoria mediante ZMP

En la Figura 11 se muestra el sistema general para efectuar la caminata de un robot bípedo implementando un controlador de postura ZMP difuso, cuya función principal es mantener en equilibrio la posición erguida del robot en relación al plano terrestre. Al inicio del sistema se establecen los “Parámetros de la caminata”, dichos valores de parámetros dictan la forma de caminar del robot. En esta se fijan el tamaño del paso, tipo de caminata, velocidad de la caminata, etc.

Todos estos parámetros son preparados por el bloque de planificación, el cual contiene los algoritmos de sincronización en función de la capacidad mecánica del robot y principalmente genera las posiciones deseadas de las piernas del robot para llevar a cabo el avance en la caminata. La posición que cada articulación debe mantener para llevar al objetivo al efecto final (Pie del robot) se calcula a través de la cinemática inversa mediante redes neuronales pre entrenadas con datos calculados por la cinemática directa.

En esta etapa se calculan dos cinemáticas, una en dirección del centro de masa hacia los pies y el otro de los pies al origen. Los ángulos obtenidos son enviados a los motores Dynamixel y establece la posición final de las articulaciones.

Después de obtener una determinada posición es necesario calcular el ZMP para poder definir la trayectoria del centro de masa y realizar una compensación de dicha trayectoria, como se observa en la Figura 12, esto se realiza mediante cuatro sensores de fuerza colocados en la planta del pie formando el polígono de soporte.

El cálculo del ZMP real es comparado con el ZMP deseado, el error generado forma parte de la primera variable de entrada de un sistema de inferencia difuso (FIS) que en tiempo real es derivado para corresponder a la segunda variable de entrada del FIS (Cambio de error), ver Figura 13. La respuesta del FIS actúa como compensación para establecer la trayectoria de la cadera en la referencia deseada.

La variable lingüística de entrada "Error" está constituida por 7 términos lingüísticos distribuidos en el universo de discurso, como se ve en la Figura 14, y que corresponden a 5 funciones triangulares y 2 trapezoidales, que va desde -13cm hasta 13cm. Los términos para el error usados son: very negative, negative, small negative, zero, small positive, positive, very positive.

La variable Lingüística de entrada "Variación" está formada por 5 términos lingüísticos distribuidos en el universo de discurso, ver Figura 15, y representados por 5 funciones de membresía triangulares y 2 trapezoidales, que van desde -13cm hasta 13cm. Los términos usados usados son: very negative, negative, small negative, zero, small positive, positive, very positive.

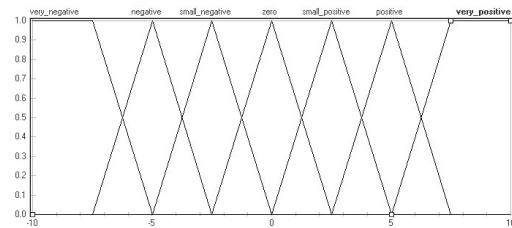


Fig. 15. Universo de discurso de la variable "Variación del error"

La variable lingüística de salida "Compensación" está formada por 7 términos lingüísticos distribuidos en el universo de discurso y representados mediante 5 funciones triangulares y 2 trapezoidales, que va desde -10cm hasta 10cm. Los términos usados son: very negative, negative, small negative, zero, small positive, positive, very positive, según se observa en la Fig. 16.

Tabla 4. Reglas del sistema de inferencia ZMP

ZMP'/ ZMP	VN	N	SN	Z	SP	P	VP
VN	VP						
N	VP	P	P	SP	SP	SP	P
SN	P	P	SP	SP	Z	SP	SP
Z	P	SP	Z	Z	Z	SN	NE
SP	SN	SN	Z	SN	SN	NE	NE
P	N	SN	SN	SN	N	N	VN
VERY POSITIVE	VN						

El sistema difuso está gobernado por cuarenta y nueve reglas como se aprecia en la Tabla 4, que establecen la acción correspondiente para efectuar la compensación más adecuada. Para observar la respuesta del sistema de inferencia, en la Figura 17 se ilustra la superficie de control del sistema.

4.2. Implementación de controlador difuso para la velocidad de movimiento

Además del controlador de trayectoria también se implementa un controlador de velocidad de movimiento, el cual se utiliza para reducir la inercia que producen los seis motores Dynamixel coordinados para generar el movimiento en las

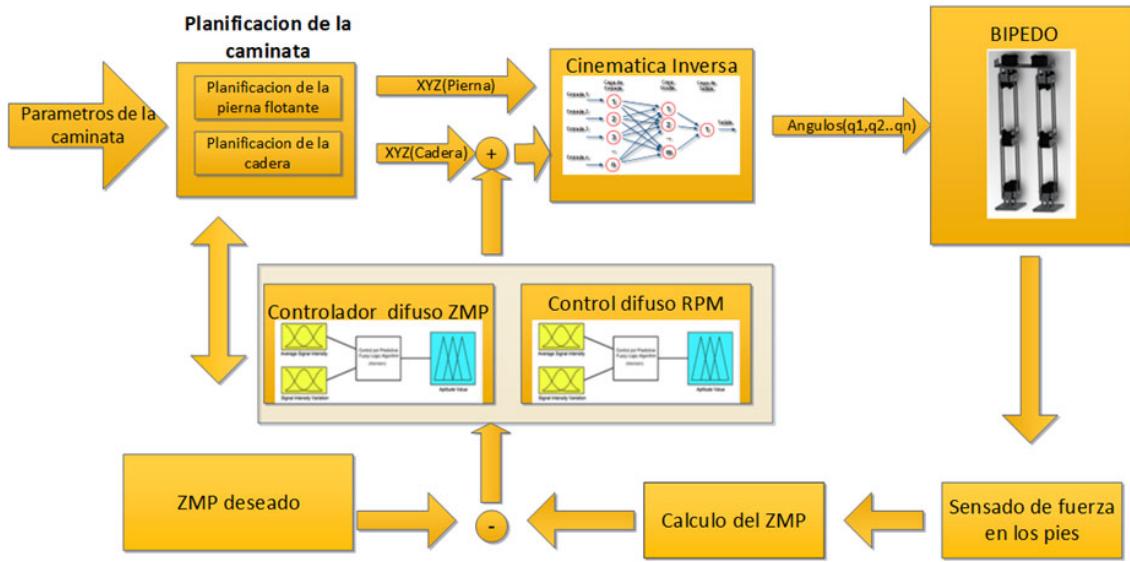


Fig. 11. Esquema del controlador difuso de postura mediante ZMP para el robot bípedo

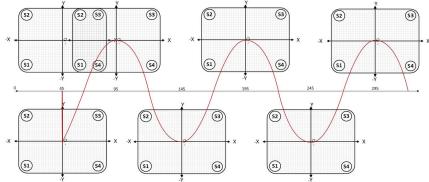


Fig. 12. Planeación de la trayectoria deseada del ZMP del robot bípedo

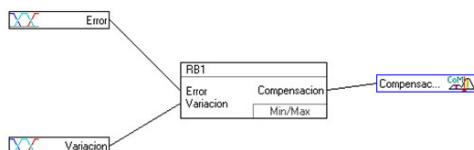


Fig. 13. Sistema de inferencia difusa del controlador ZMP basado en sensores

cadenas cinemáticas. El objetivo es reducir la velocidad cuando se aproxima al objetivo y aumentarla cuando éste se aleja. Para ello se usan las variables lingüísticas de entrada “Error” y “Cambio de error”, para la variable de salida “RPM” que configura directamente las RPMs de los motores cuyos parámetros se encuentran entre 0 y 33.3 RPM.

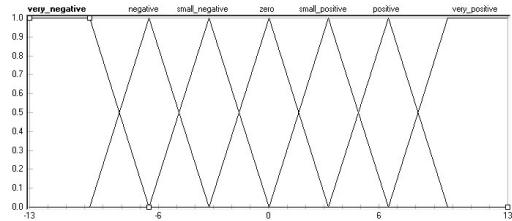


Fig. 14. Universo de discurso de la variable “Error”

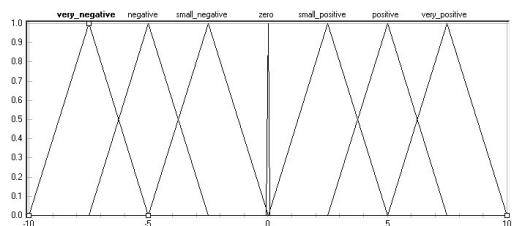


Fig. 16. Universo de discurso de la variable “Compensación”.

La variable lingüística de entrada “Error” representa los valores absolutos de la diferencia del valor deseado y el valor actual ZMP y está constituida por tres términos lingüísticos que son distribuidos en el universo de discurso, y representados por una función triangular y dos

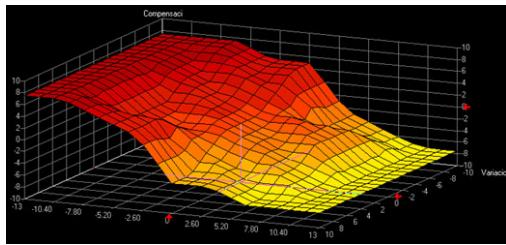


Fig. 17. Superficie de control del controlador ZMP

trapezoidales, que va desde 0 hasta 10cm. Los términos para la variable “Error” usados son small, medium y large, como se observa en la Figura 18.

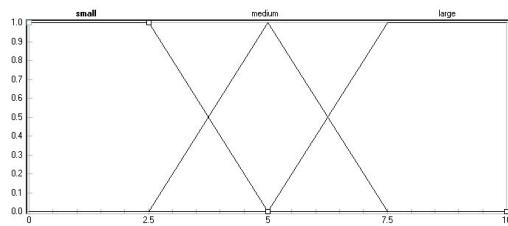


Fig. 18. Universo de discurso de la variable de entrada “Error” del controlador de velocidad de movimiento

La variable lingüística de entrada “Cambio de error” representa los valores absolutos de la diferencia del Error actual y error anterior ZMP, está constituida por tres términos lingüísticos que son distribuidos en el universo de discurso por una función triangular y dos trapezoidales que va desde 0 hasta 10cm. Los términos para la variable “Cambio de error” usados son small, medium y large, ver Figura 19.

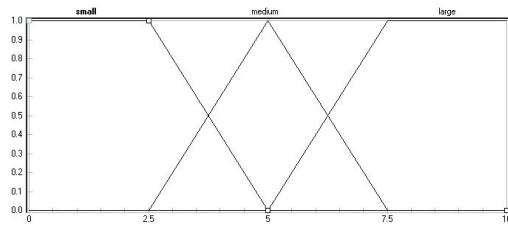


Fig. 19. Universo de discurso de la variable de entrada “Cambio de error” del controlador de velocidad de movimiento

La variable lingüística de salida “RPM” representa las velocidades de los motores, está constituida

por tres términos lingüísticos que son distribuidos en el universo de discurso por cinco funciones de membresía triangulares que va desde 0 hasta 1 que normalizan respectivamente los parámetros de fábrica de los motores. Los términos para la variable “RPM” usados son VeryLow, Low, Medium, High, VeryHigh, según se ve en la Figura 20.

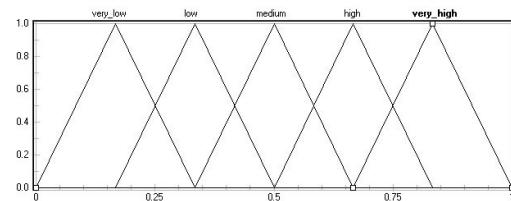


Fig. 20. Universo de discurso de la variable de salida “RPM” del controlador de velocidad de movimiento

La acción del controlador difuso está basada en nueve reglas las cuales se ilustran en la Tabla 5, y que permiten tener una respuesta de mayor precisión y suavidad en la ejecución de los movimientos del robot bípedo. Para apreciar el comportamiento, en la Figura 21 se muestra la superficie de control del FIS.

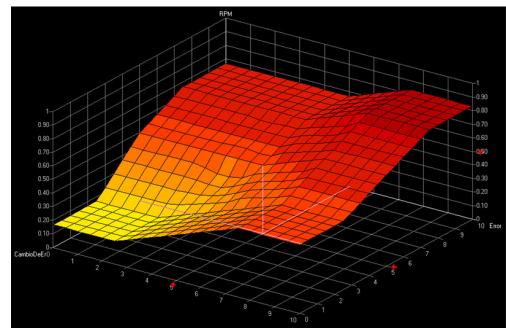


Fig. 21. Superficie de control para el controlador de velocidad de movimiento

5. Resultados

La red se entrena previamente en el ToolBox RNA de Matlab para su posterior uso en la herramienta de generación de código C en los bloques de Simulink Embedded, donde se prueban y se comparan físicamente los resultados de la

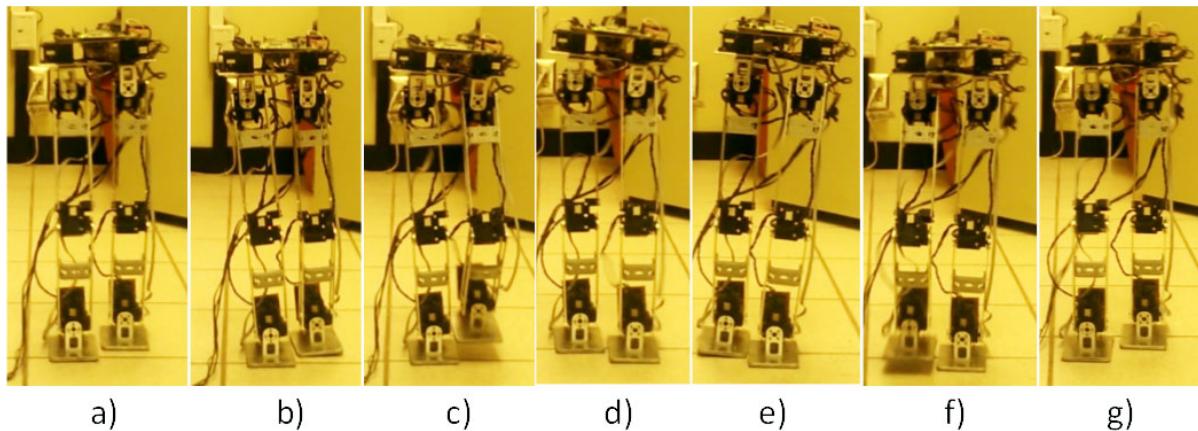


Fig. 22. Plano frontal de la caminata bípeda

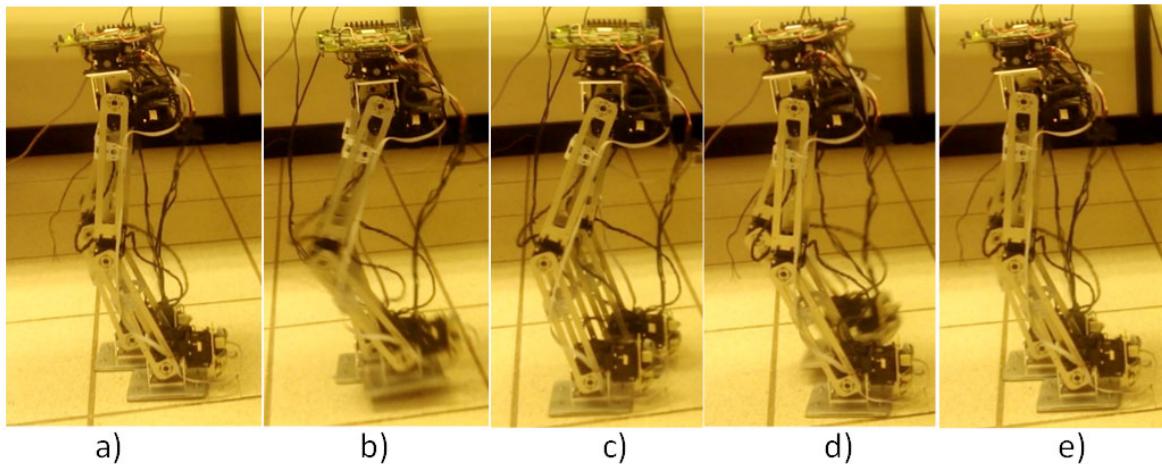


Fig. 23. Plano sagital de la caminata bípeda

Tabla 5. Reglas del sistema de inferencia para el FIS de velocidad de movimiento

Error ¹ / CambioDe error	Low	Medium	High
Small	Very Low	Low	Medium
Medium	Medium	Medium	High
Large	High	High	Very High

simulación con las pruebas de laboratorio. En las Figuras 9, 10, 11 y 12 se muestran las similitudes de los resultados de la simulación y las

pruebas con el prototipo, evaluando los vectores de posición y orientación obteniendo resultados favorables.

Para planear la caminata se localizaron puntos estables en los planos frontal, ver Figura 22, y sagital, ver Figura 23, para llevar a cabo la trayectoria que debe seguir el centro de masa para realizar el ciclo de caminata dinámica mediante la manipulación de la cinemática inversa. Esto dió como resultado una aproximación a un caminado estético similar al ser humano.

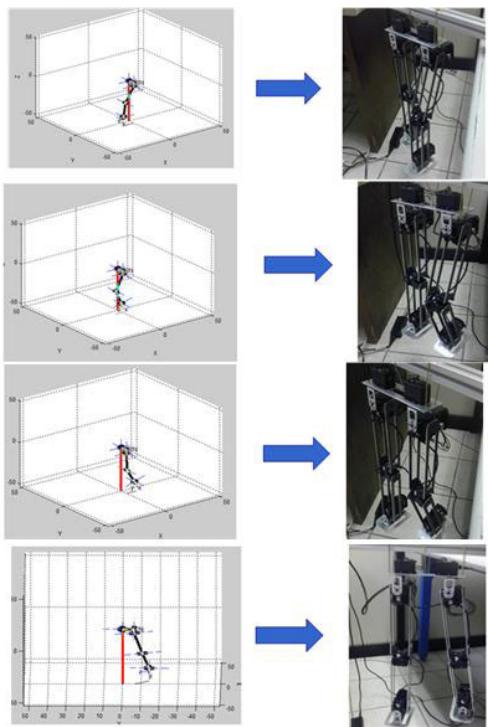


Fig. 24. Evaluacion de la cinematica inversa en el robot bípedo

Para facilitar el modelado del robot bípedo, el análisis se basa en el Péndulo Invertido [6] y se simula en la herramienta Simulink el controlador PD+I difuso para obtener la respuesta de escalón unitario mostrada en la Figura 25, obteniendo una estabilización en la señal de salida en un tiempo de 8 ms y con un reducido efecto de Gibbs, por consiguiente tambien evaluamos el controlador en el seguimiento de una señal cuadrada obteniendo una respuesta satisfactoria, ver Figura 26.

Las pruebas realizadas de la caminata del robot se muestran en Figura 27, se observa la trayectoria planeada de referencia contra la señal de salida ZMP del robot realizando una trayectoria recta con un avance de 2 cm por cada paso con una altura de 46.6 cm, el robot se mantiene de forma estable sin caerse.

En la Tabla 6 se muestran las velocidades máximas alcanzadas por el robot bípedo, en una superficie plana concluyendo que si el robot

reduce la altura se puede incrementar el avance y por lo tanto aumenta la velocidad de la caminata.

Tabla 6. Velocidades máximas alcanzadas por el robot bípedo

Altura del robot (cm)	Longitud de paso (cm)	Velocidad (m/min)
45.4	5	6
45.6	5	8
45.8	4.5	10
46.0	4	11
46.4	4	12
46.6	3	15

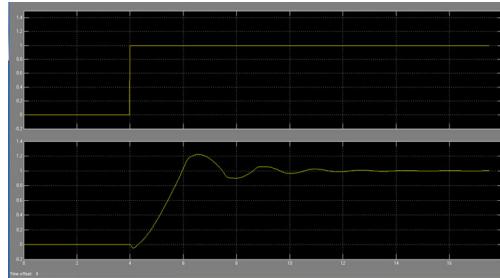


Fig. 25. Simulación del controlador ZMP para la respuesta al escalón unitario

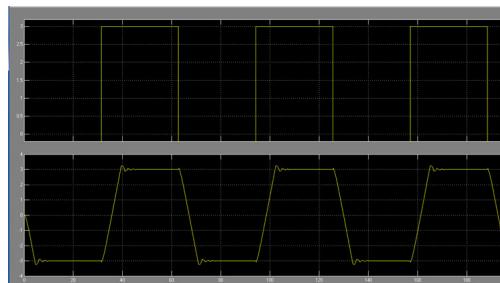


Fig. 26. Simulación de seguimiento del controlador ZMP

6. Conclusiones

Como resultado de esta investigación se concluye que la metodología presentada resuelve eficientemente el problema de locomoción en el robot bípedo experimental construido en el

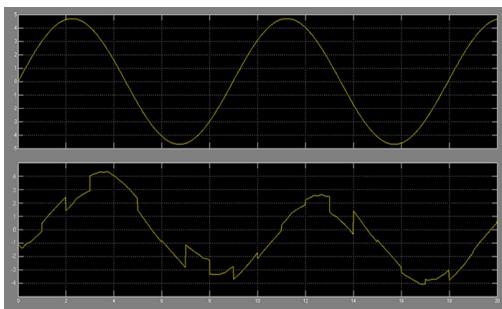


Fig. 27. Respuesta real del controlador ZMP en el robot bípedo a trayectorias deseadas

laboratorio, y que es factible emplear redes neuronales para calcular la cinemática inversa para robots con muchos grados de libertad, obteniendo gran precisión en los movimientos, además que la ejecución del algoritmo pre-entrenado, en el hardware propicia una reducción de latencias, idóneo en estos sistemas embebidos.

También esta propuesta comprueba que usar como base el conocimiento experimental, para diseñar el controlador de postura ZMP con lógica difusa, favorece en una implementación más simple que los métodos de control clásico, pero sin sacrificar su la operabilidad y eficiencia.

Referencias

1. **Braun, D. & Goldfish, M. (2009).** A control approach for actuated dynamic walking in biped robots. *IEEE Transactions on Industrial Electronics*, Vol. vol. 25, no. 6, No. 1, pp. 1292–1303.
2. **c. Choi, K., j. Lee, H., & Lee, M. C. (2006).** Fuzzy posture control for biped walking robot based on force sensor for zmp. *SICE-ICASE, 2006. International Joint Conference*, pp. 1185–1189.
3. **Cheng, M.-Y. & Lin, C.-S. (1997).** Genetic algorithm for control design of biped locomotion. *Journal of Robotic Systems*, Vol. 14, No. 5, pp. 365–373.
4. **Goswami, A. (1999).** Foot rotation indicator (fri) point: a new gait planning tool to evaluate postural stability of biped robots. *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pp. 47–52 vol.1.
5. **Honda Motor Co. (2007).** *The Honda Humanoid Robot*.
6. **Hong, S., Oh, Y., Kim, D., & You, B. J. (2014).** Real-time walking pattern generation method for humanoid robots by combining feedback and feed-forward controller. *IEEE Transactions on Industrial Electronics*, Vol. 61, No. 1, pp. 355–364.
7. **McGeer, T. (1990).** A control approach for actuated dynamic walking in biped robots. *Int. J. Robot. Res.*, Vol. 9, No. 2, pp. 62–82.
8. **Shieh, M. Y., Chang, K. H., Chuang, C. Y., Chiou, J. S., & Li, J. H. (2007).** Anfis based controller design for biped robots. *Mechatronics, ICM2007 4th IEEE International Conference on*, pp. 1–6.
9. **Shimojo, M., Araki, T., Ming, A., & Ishikawa, M. (2006).** A zmp sensor for a biped robot. *Robotics and Automation (ICRA), Proceedings 2006 IEEE International Conference on*, pp. 1200–1205.
10. **Vukobratovic, M. & Borovac, B. (2004).** Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, Vol. 01, No. 01, pp. 157–173.
11. **Vukobratović, M. & Timčenko, O. (1996).** Experiments with nontraditional hybrid control technique of biped locomotion robots. *Journal of Intelligent and Robotic Systems*, Vol. 16, No. 1, pp. 25–43.
12. **Zhou, C. (2000).** Neuro-fuzzy gait synthesis with reinforcement learning for a biped walking robot. *Soft Computing*, Vol. 4, No. 4, pp. 238–250.

Article received on 07/04/2017; accepted on 10/11/2017.
Corresponding author is Jorge A. Quiñones.