

Identificación del campo de trabajo de un robot hexápodo utilizando optimización multiobjetivo

Josué Domínguez Guerrero¹, Roberto Sepúlveda¹, Oscar Montiel¹, Oscar Castillo²

¹ Instituto Politécnico Nacional, Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI), Tijuana, B.C., Mexico

² Tijuana Institute of Technology, Tijuana, B.C., México

jdominguez@citedi.mx, rsepulvedac@ipn.mx, oross@ipn.mx, ocastillo@tectijuana.mx

Abstract. In this paper, an innovative method to find the workspace of a hexapod mobile robot is presented. Differently, to the current state of the art methods that allows to determine the working spaces for walking in straight line, the proposed method allows estimating the optimal set of configurations for walking in any viable direction. The method takes advantage of the existing similitudes between the hexapod and the Delta robot during the tripod walk; however, there are some movement restrictions between them, which were conveniently solved by handling them as a multiobjective optimization problem. Particularly, the MOEA/D algorithm was used to solve the problem.

Keywords. Hexapod robot, multiobjective optimization, delta robot, evolutionary algorithms, gait.

1. Introducción

En la actualidad, la mayoría de los robots están diseñados para desplazarse en superficies lisas, niveladas o ligeramente inclinadas. Los ambientes externos con superficies irregulares e inestables como arena, nieve, lodo o grava, y los interiores como escaleras, puertas y esquinas, se consideran ambientes difíciles para el desplazamiento de robots móviles [17]. En la naturaleza existe una gran diversidad de animales de los cuales se puede imitar su locomoción para diseñar robots capaces de sortear este tipo de terrenos [10], por ejemplo, los animales hexápodos, cuya estructura y modelo de locomoción les permite evitar pequeños obstáculos y caminar por terrenos

inclinados, lo que ha inspirado la creación de robots que imiten su locomoción. La habilidad de un robot hexápodo para eludir obstáculos, caminar en distintos tipos de terreno y su eficiencia energética depende en gran medida de la relación entre el largo de sus patas y su peso.

En el movimiento y desplazamiento de un robot hexápodo se tienen que controlar y coordinar al mismo tiempo sus seis patas, cada una de las cuales se representa normalmente utilizando la estructura de una cadena cinemática de tres eslabones con tres grados de libertad [3], por lo que el control de un robot hexápodo es un problema en el que se tienen que controlar 18 uniones al mismo tiempo.

El caminado de un hexápodo de acuerdo al número de patas que se encuentran en apoyo, se puede clasificar en secuencias de caminado, que pueden ser de tipo onda, tetrápodo, transición y trípode. Para crear estas secuencias de caminado se puede utilizar un generador central de patrones [3, 12, 6], redes neuronales [7], algoritmos genéticos [11] o un generador aleatorio basado en aprendizaje [5]. En el caminado cada pata puede tener dos estados: avance y recuperación.

Un robot hexápodo al considerarse como un sistema aislado sus patas se pueden mover libremente sin embargo al poner las patas sobre la superficie estas ya no se pueden mover libremente sino que estarán restringidas por la fricción entre el piso y las patas, sus movimientos se limitan de

manera similar a lo que sucede cuando una mano robótica con tres dedos sujeta un objeto uniforme [2]. Durante la etapa de avance en el caminado trípode un robot hexápodo se asemeja a un robot paralelo tipo Delta como el presentado en [18], la plataforma móvil es el cuerpo del robot y la plataforma fija es el triángulo de apoyo que forman las tres patas que se encuentran en contacto con el suelo. En [1] se presenta el diseño de una estructura de un robot hexápodo paralelo, el cual de forma similar al robot tipo Delta, su plataforma superior está fija, la longitud de cada una de sus patas varía utilizando uniones tipo pistón para mover la plataforma inferior, de este tipo de robots en [8] se presenta un análisis de su exactitud y de la simetría en el campo de trabajo; en [13], se presenta un método para resolver el problema de la cinemática directa de este tipo de robots utilizando un algoritmo genético.

Este trabajo contribuye con un método para encontrar el campo de trabajo de un robot hexápodo, utiliza la similitud existente entre un robot hexápodo y uno tipo Delta. La representación del robot hexápodo utilizada es similar a la presentada en [8, 13, 1] con la diferencia que éstos son robots estáticos, la representación de un robot hexápodo móvil como un robot paralelo también se utiliza en [15] donde se hace un análisis de la estabilidad del robot antes del caminado. El método propuesto en este trabajo permite encontrar todas las posibles combinaciones de las articulaciones de las tres patas en soporte del hexápodo que hacen que su cuerpo avance o gire manteniendo las puntas de las patas en el mismo triángulo de apoyo.

De acuerdo con la investigación realizada, no hay un trabajo donde se encuentre el campo de trabajo para un robot hexápodo móvil, las investigaciones reportadas generan solo el movimiento con patrones de caminado en línea recta [3, 12, 6, 5]. Para encontrar el campo de trabajo del robot, se requiere una solución que satisfaga de manera simultánea y óptima varios objetivos, por lo que este problema intrínsecamente es un problema de optimización multiobjetivo, en el cual se deberán minimizar los errores de los ángulos, el perímetro entre el triángulo de apoyo de la posición actual, y

el de la nueva posición. La propuesta utiliza el algoritmo MOEA/D (del inglés Multiobjective Evolutionary Algorithm Based on Decomposition) propuesto en [19], cada punto en la frontera de Pareto representa una posición viable para un robot hexápodo.

Este trabajo está organizado de la siguiente manera: En la sección 2 se describe el modelo matemático de los robots hexápodos. En la sección 3 se describe la secuencia de caminado utilizado para el robot hexápodo y la formulación multiobjetivo del problema de mantener el triángulo de apoyo en la misma posición para encontrar el campo de trabajo, también se describe el algoritmo MOEA/D. Los resultados y experimentos se muestran en la sección 4. Finalmente, en la sección 5 se presentan las conclusiones.

2. Modelo cinemático de un robot hexápodo

En la Figura 1, se observan las patas del modelo del robot hexápodo utilizadas las cuales están distribuidas simétricamente al eje y , que es el eje sobre el cual se desplaza el robot. En la Figura 2, se presenta el modelo cinemático utilizado para representar cada pata del robot. Considerando que el origen se encuentra en el punto de anclaje de la pata con el cuerpo, la altura z_1 es la distancia entre el origen y la superficie de contacto, cada pata de un robot hexápodo se puede analizar como una cadena cinemática de tres eslabones, la cual inicia en la unión 0 donde está sujeta la pata al cuerpo del robot, el eslabón 1 es la coxa, el eslabón 2 es el fémur y el eslabón 3 es la tibia [9, 14]. La Tabla 1 muestra los parámetros de Denavit-Hartenberg (D-H) del robot: longitud del eslabón (a_i), giro del eslabón (α_i), distancia de la unión (d_i) y ángulo de la unión (θ_i).

Tabla 1. Parámetros D-H

Eslabón	a_i	α_i	d_i	θ_i
1	L_1	$\pm 90^\circ$	0	θ_1
2	L_2	0	0	θ_2
3	L_3	0	0	θ_3

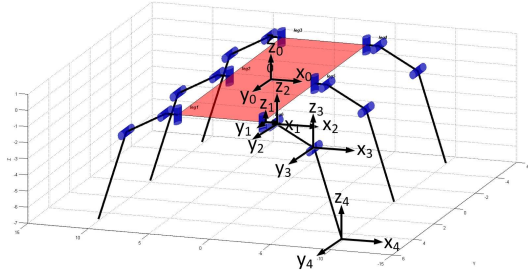


Fig. 1. Modelo del robot

La matriz de transformación homogénea que describe la traslación y rotación entre el i -ésimo y el $(i - 1)$ -ésimo sistema de coordenadas se representa en (1). La ubicación de la punta de la pata con respecto al origen está definida en las ecuaciones (5), (6) y (7), las cuales se obtienen utilizando las matrices de transformación homogénea definidas en (2), (3) y (4) [14]:

$${}^{i-1}\mathbf{T}_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & \alpha_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & \alpha_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1)$$

$${}^0\mathbf{T}_1 = \begin{pmatrix} \cos \theta_1 & 0 & \sin \theta_1 & L_1 \cos \theta_1 \\ \sin \theta_1 & 0 & -\cos \theta_1 & L_1 \sin \theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2)$$

$${}^1\mathbf{T}_2 = \begin{pmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & L_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & L_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

$${}^2\mathbf{T}_3 = \begin{pmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & L_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & L_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (4)$$

$$x = [L_1 + L_2 \cos \theta_2 + L_3 \cos(\theta_2 + \theta_3)] \cos \theta_1, \quad (5)$$

$$y = [L_1 + L_2 \cos \theta_2 + L_3 \cos(\theta_2 + \theta_3)] \sin \theta_1, \quad (6)$$

$$z = L_2 \sin \theta_2 + L_3 \sin(\theta_2 + \theta_3). \quad (7)$$

Resolviendo las ecuaciones de posición (5), (6) y (7), se pueden determinar los valores de

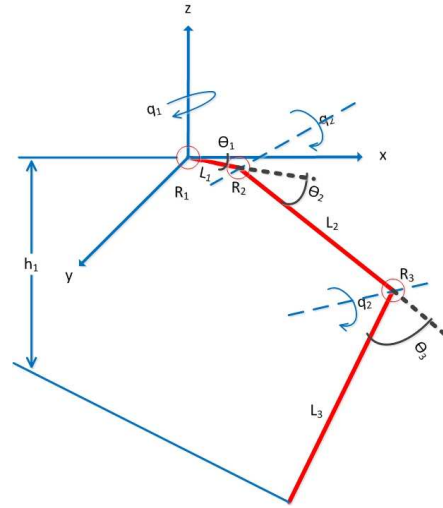


Fig. 2. Modelo cinemático de una pata

los ángulos de las articulaciones de cada pata utilizando las ecuaciones (8), (9) y (10):

$$\theta_1 = a \tan 2(y, x), \quad (8)$$

$$\theta_2 = a \tan 2(c, \pm \sqrt{a^2 + b^2 + c^2}) - a \tan(a, b), \quad (9)$$

donde $a = 2L_2(\sqrt{x^2 + y^2} - L_1)$; $b = 2zL_2$; $c = [(\sqrt{x^2 + y^2} - L_1)^2 + z^2 + L_2^2 - L_3^2]$, and

$$\theta_3 = \cos^{-1} \left[\frac{(\sqrt{x^2 + y^2} - L_1)^2 + z^2 - L_2^2 - L_3^2}{2L_2L_3} \right]. \quad (10)$$

3. Identificación del campo de trabajo del robot hexápodo

El caminado trípode del robot hexápodo se dividió en seis estados a partir de la posición inicial en la cual las patas 1, 3 y 5 están en apoyo y las patas 2, 4 y 6 están levantadas. El proceso de caminado se muestra en la Figura 3.

En el primer estado las dos patas extremas del lado izquierdo 1 y 3, y la pata central 5 del lado derecho se encuentran en apoyo, estas patas se

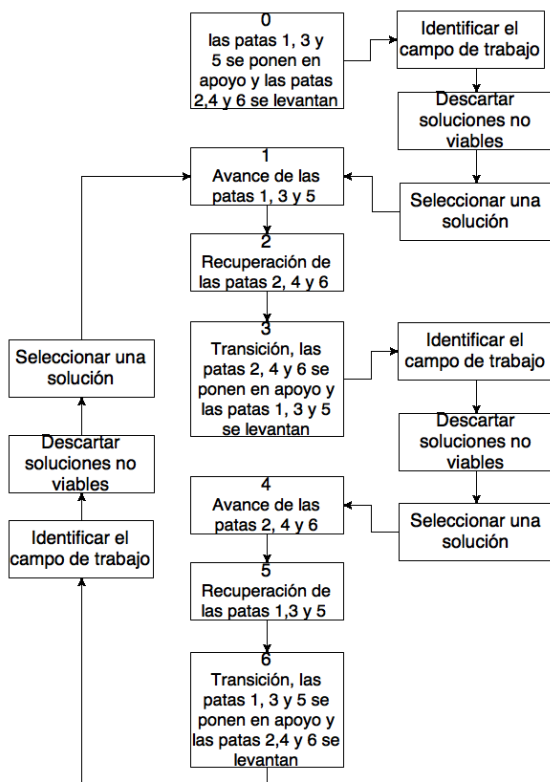


Fig. 3. Proceso de caminado del robot hexápodo

desplazan hacia atrás cambiando el valor de los ángulos $\theta_{1,1}$ y $\theta_{1,3}$ por $-\Delta\theta_1$, y el ángulo $\theta_{1,5}$ por $\Delta\theta_1$, esto provocará que el robot avance en línea recta hacia enfrente; en el segundo estado (recuperación), las dos patas extremas del lado derecho 4 y 6, y la pata central 2 del lado izquierdo se ponen en la posición inicial, a los ángulos $\theta_{1,4}$ y $\theta_{1,6}$ y $\theta_{1,2}$ se les asigna el valor cero; en el tercer estado (transición), las patas 4, 6 y 2 se ponen en apoyo, posteriormente las patas 1, 3 y 5 se levantan; en el estado cuatro las patas 4, 6 y 2 realizan el avance; en el estado cinco las patas 1, 3 y 5 se recuperan a su posición inicial; en el estado seis se realiza la transición y se vuelve a repetir el ciclo.

El primer y el cuarto estado son los que controlan el desplazamiento del hexápodo. En el avance en línea recta el ángulo $\theta_{1,i}$ de las patas del lado derecho y del lado izquierdo en apoyo

deben tener la misma magnitud pero con sentido diferente. La acción de girar el cuerpo de un robot hexápodo es compleja ya que se tiene que encontrar los valores de los ángulos $\Theta_{1,a}$ que permitan al robot girar manteniendo el triángulo de apoyo en la misma posición inicial. En el primer estado el movimiento del cuerpo de un robot hexápodo es controlado por los ángulos $\Theta_{1,a} = \{\theta_{1,1}, \theta_{1,3}, \theta_{1,5}\}$ y en el cuarto estado es controlado por los ángulos $\Theta_{1,a} = \{\theta_{1,2}, \theta_{1,4}, \theta_{1,6}\}$.

Durante la etapa de avance de un robot hexápodo con caminado tripode se forma un robot paralelo tipo Delta (Figura 4). Esta clase de robots está compuesta por una plataforma fija y una plataforma móvil conectadas por tres cadenas cinemáticas idénticas e igualmente distribuidas [16]. En el robot hexápodo la plataforma móvil es el cuerpo y la plataforma fija es el triángulo de apoyo que se forma con las tres patas en el piso.

3.1. Definición multiobjetivo del problema

Un problema de optimización multiobjetivo se define como minimizar o maximizar las funciones objetivo $F(\mathbf{x})$ en el espacio objetivo Z , sujeto a restricciones de igualdad, de desigualdad y los límites de las variables de decisión $\mathbf{x} \in \Omega$. La meta en un problema de optimización multiobjetivo es encontrar el conjunto óptimo de Pareto \mathcal{P}^* , es decir el conjunto de soluciones $\mathbf{x} \in \Omega$ para las cuales no existe ninguna solución $\mathbf{x}' \in \Omega$ que domine \mathbf{x} . Una solución \mathbf{x}' domina a otra solución \mathbf{x} ($\mathbf{x}' \preceq \mathbf{x}$) si \mathbf{x}' es parcialmente menor que \mathbf{x} , asumiendo que todos los objetivos se minimizarán (\mathbf{x}' sera menor o igual que \mathbf{x} para todos los objetivos y menor al menos que uno de los objetivos).

Para mantener el triángulo de apoyo en la etapa de avance se debe asegurar que el triángulo formado por la posición de las patas 1 (x_1^t, y_1^t, z_1^t), 3 (x_3^t, y_3^t, z_3^t) y 5 (x_5^t, y_5^t, z_5^t) en apoyo en el primer estado antes de avanzar, sea el mismo que el formado por las patas 1 ($x_1^{t+1}, y_1^{t+1}, z_1^{t+1}$), 3 ($x_3^{t+1}, y_3^{t+1}, z_3^{t+1}$) y 5 ($x_5^{t+1}, y_5^{t+1}, z_5^{t+1}$) después de avanzar. El triángulo de apoyo antes y después del avance será igual si dos de sus ángulos y su perímetro es igual. Para cumplir con lo anterior, se planteó una formulación matemática de optimización multiobjetivo, la cual se muestra en

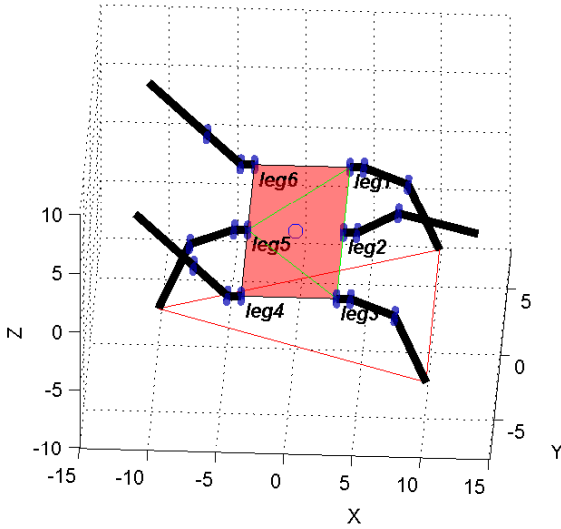


Fig. 4. Representación de un robot hexápodo como robot paralelo

la ecuación (11). El primer objetivo, es minimizar el error entre el ángulo β_1^t , y el mismo ángulo β_1^{t+1} después del avance; el segundo objetivo, es minimizar el error entre los ángulos β_5^t y β_5^{t+1} ; el tercer objetivo, es minimizar el error del perímetro del triángulo de apoyo en el primer estado p^t y el perímetro del triángulo después de avanzar p^{t+1} :

Minimizar

$$\begin{aligned} f_1(\Theta_{1,a}) &= |\beta_1^t - \beta_1^{t+1}|, \\ f_2(\Theta_{1,a}) &= |\beta_5^t - \beta_5^{t+1}|, \\ f_3(\Theta_{1,a}) &= |p^t - p^{t+1}|, \end{aligned} \quad (11)$$

sujeto a

$$-\pi/6 \leq \Theta_{1,a} \leq \pi/6.$$

Los ángulos β_1^t y β_5^t son dos de los ángulos formados por el triángulo de apoyo de las patas en el tiempo t los cuales permanecen constantes. Para encontrar sus valores se utilizan las ecuaciones (12) y (13), las cuales se obtienen descomponiendo el triángulo de apoyo en tres triángulos rectángulos como se muestra en la Figura 5, y utilizando las ecuaciones de la cinemática (5) y (6) se encuentra la posición de

las patas 1, 3 y 5. El perímetro p^t del triángulo de apoyo se calcula utilizando la ecuación (14). Los ángulos β_1^{t+1} y β_5^{t+1} y el perímetro p^{t+1} , después de que el cuerpo del robot se movió, se calculan utilizando las mismas ecuaciones con los valores de las patas en la nueva posición.

$$\beta_1^t = \tan^{-1}\left(\frac{x_1^t - x_5^t}{y_1^t - y_5^t}\right) + \tan^{-1}\left(\frac{x_1^t - x_3^t}{y_1^t - y_3^t}\right), \quad (12)$$

$$\beta_5^t = \sin^{-1}\left(\frac{\sin(\beta_1^t)d_1}{d_2}\right), \quad (13)$$

$$p^t = d_1 + d_2 + d_3, \quad (14)$$

en donde:

$$d_1 = \sqrt{(x_1^t - x_3^t)^2 + (y_1^t - y_3^t)^2},$$

$$d_2 = \sqrt{(x_3^t - x_5^t)^2 + (y_3^t - y_5^t)^2},$$

$$d_3 = \sqrt{(x_1^t - x_5^t)^2 + (y_1^t - y_5^t)^2}.$$

Las variables de búsqueda del problema de optimización multiobjetivo son los ángulos $\Theta_{1,a}^{t+1} = \{\theta_{1,1}^{t+1}, \theta_{1,3}^{t+1}, \theta_{1,5}^{t+1}\}$, estos ángulos provocan que las patas se coloquen en una nueva posición, la cual se encuentra utilizando las ecuaciones (5), (6) y (7). Con esta nueva posición se calculan los ángulos β_1^{t+1} y β_5^{t+1} y el perímetro p^{t+1} que definen el nuevo triángulo de apoyo.

En el problema de restricción del triángulo de apoyo el conjunto óptimo de Pareto \mathcal{P}^* son todas las soluciones $\Theta_{1,a}^{t+1}$ en el espacio de búsqueda Ω tal que no exista ninguna solución $\Theta_{1,a}^{t+1'}$ que sea dominante de $\Theta_{1,a}^{t+1}$. Para que $\Theta_{1,a}^{t+1'}$ domine a $\Theta_{1,a}^{t+1}$, se debe cumplir $f_j(\Theta_{1,a}^{t+1'}) \leq f_j(\Theta_{1,a}^{t+1})$ para todo $j = 1, 2, 3$ y $f_j(\Theta_{1,a}^{t+1'}) < f_j(\Theta_{1,a}^{t+1})$ para al menos una $j = 1, 2, 3$.

Todos los triángulos de apoyo que cumplen esta condición, es decir todas las posiciones en las que se puede mover el hexápodo manteniendo sus patas en el mismo lugar es el campo de trabajo, el cual como se muestra en la Figura 3 se tiene que encontrar al iniciar el caminado del robot entre los

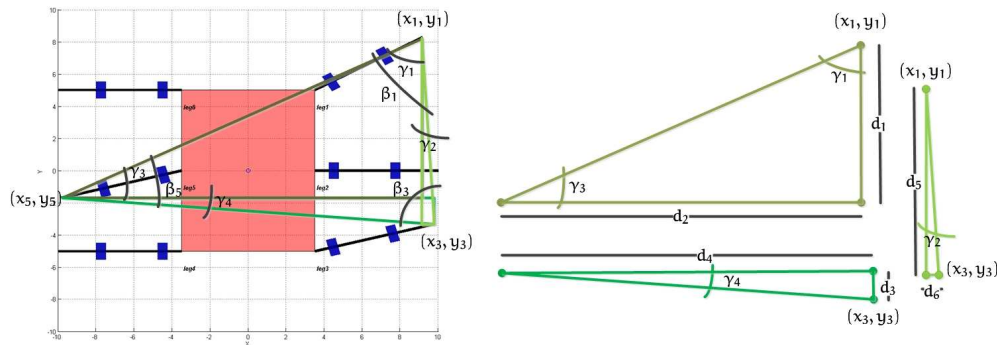


Fig. 5. Descomposición del triángulo de apoyo

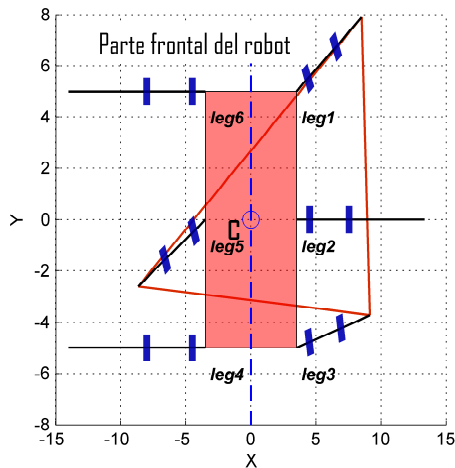


Fig. 6. Centro del robot dentro del triángulo de apoyo

estados 0 y 1, y después en la transición de los estados 3 y 4, y 6 y 1.

La condición para que el robot hexápodo no se caiga durante el caminado, es que el centro de gravedad de la parte superior del cuerpo permanezca siempre dentro del polígono de apoyo, como se muestra en la Figura 6. Esta condición se cumple mientras la distancia S_f sobre el eje x a la diagonal que se forma entre la pata central y la pata de enfrente, o la distancia S_b sobre

el eje x a la diagonal que se forma entre la pata central y la pata de atrás, sean mayor que cero.

3.2. Algoritmo para optimización multiobjetivo basado en descomposición (MOEA/D)

Los problemas de optimización multiobjetivo se resuelven principalmente usando métodos que asignan un valor de peso a cada objetivo de acuerdo a su importancia en el problema (métodos escalados), que lo convierten en un problema de un solo objetivo o usando algoritmos evolutivos los cuales ofrecen la ventaja de poder encontrar múltiples soluciones óptimas de Pareto en una sola corrida. También es relativamente más sencillo adaptarlos a diferentes problemas de aplicación. En optimización multiobjetivo es importante encontrar la mayor cantidad de soluciones óptimas de Pareto, pero también que éstas estén distribuidas uniformemente sobre el frente de Pareto.

Para encontrar las posiciones en las que se puede mover un robot hexápodo se utilizó el algoritmo MOEA/D propuesto en [19], el cual divide un problema de optimización multiobjetivo en N subproblemas de un solo objetivo que descompone el espacio objetivo en espacios igualmente distribuidos. La dispersión de las soluciones en el frente de Pareto es superior a otros algoritmos evolutivos de acuerdo a lo reportado en [19].

Algoritmo 1: Algoritmo MOEA/D**Entradas**

MOP;
 Criterio de paro;
 N : número de subproblemas en MOEA/D;
 N vectores de pesos: $\lambda^1, \dots, \lambda^N$;
 τ : número de vectores de pesos colindantes.

Salida

EP .

Paso 1) Inicialización:

Paso 1.1) $EP = \emptyset$

Paso 1.2) Para cada $i = 1, 2, \dots, N$ encontrar $B(i) = \{i_1, i_2, \dots, i_\tau\}$, donde $\lambda^{i_1}, \dots, \lambda^{i_\tau}$ son los τ vectores mas cercanos a λ^i

Paso 1.3) Inicializar la población x^1, \dots, x^N .
 Encontrar los vector de funciones objetivo $FV^i = F(x^i)$

Paso 1.4) Inicializar $z = (z_1, \dots, z_m)^T$

Paso 2) Actualización:

for $i = 1$ to N do

Paso 2.1) Reproducción: Seleccionar k y l de $B(i)$, usando operadores genéticos generar y de x^k y x^l .

Paso 2.2) Mejoramiento: Mejorar la solución y para producir y'

Paso 2.3) Actualización de z :

```

foreach  $j = 1, \dots, m$  do
  if  $z_j < f_j(y')$  then
     $z_j = f_j(y')$ 
  end
end

```

end

Paso 2.4) Actualización de soluciones colindantes:

```

foreach  $j \in B(i)$  do
  if  $g^{te}(y'|\lambda^j, z) < g^{te}(x^j|\lambda^j, z)$  then
     $x_j = y'$ ;  $FV^j = F(y')$ 
  end
end

```

end

Paso 2.5) Actualización de EP :

Eliminar todos los vectores dominados por $F(y')$ de EP .

Agregar $F(y')$ a EP si no hay ningún vector en EP que lo domine.

end

Paso 3) Criterio de paro

Si se cumple el criterio de paro, terminar y devolver EP , sino ir al **Paso 2**.

asignado un conjunto B_i con los τ vectores de peso más cercanos. Para resolver cada subproblema se utiliza el enfoque de Tchebycheff definido en la ec. (15):

$$g^{te}(x|\lambda^j, z^*) = \max_{1 \leq i \leq m} \{\lambda_i^j |f_i(x) - z_i^*|\}, \quad (15)$$

donde $\lambda^j = \lambda_1^j, \lambda_2^j, \dots, \lambda_m^j$ es el vector de pesos del individuo j , cada subproblema se resuelve utilizando los τ vectores de pesos uniformemente distribuidos $\lambda^1, \lambda^2, \dots, \lambda^N$ más cercanos a cada λ^j . El algoritmo MOEA/D resuelve todo los N subproblemas en una sola corrida, su funcionamiento se muestra en el Algoritmo 1.

4. Resultados experimentales

La comprobación del método propuesto se realizó utilizando la herramienta Robotics [4] de Matlab para simular un robot hexápodo y ver el movimiento de sus articulaciones. En el experimento se consideró que las patas 1, 3 y 5 están en apoyo como se muestra en la Figura 4 y los ángulos iniciales son $\theta_{1,1}^0 = \theta_{1,3}^0 = \theta_{1,5}^0 = 0$, $\theta_{2,1}^0 = \theta_{2,3}^0 = \theta_{2,5}^0 = -\pi/8$ y $\theta_{3,1}^0 = \theta_{3,3}^0 = \theta_{3,5}^0 = -\pi/4$. Las longitudes de los eslabones de las patas son $l_1 = 1$ cm., $l_2 = 3,5$ cm. y $l_3 = 6$ cm, el ancho del cuerpo del robot es $W = 10$ cm y el largo $L = 7$ cm.

Estas medidas se eligieron porque son las características físicas de un robot hexápodo que se tiene en el laboratorio de sistemas digitales del CITED-IPN. Las variables de búsqueda son los ángulos $\theta_{1,1}^1, \theta_{1,3}^1$ y $\theta_{1,5}^1$ todos entre $-\pi/6$ y $\pi/6$. Los ángulos $\theta_2 = -\pi/8$ y $\theta_3 = -\pi/4$ permanecen constantes, por lo tanto la altura de las tres patas también permanece constante $z = -6,8827$ cm.

En el algoritmo evolutivo se utilizó una población de 300 individuos con un vecindario $\tau = 20$, y se ejecutó durante 100 generaciones. En la Figura 7 se observa el frente de Pareto obtenido después de 100 generaciones, en esta Figura también se indica la ubicación en el espacio objetivo de las soluciones a, b, c, y d, de las cuales se muestra el movimiento que provocan en las patas del robot en la Figura 9, estas soluciones fueron elegidas aleatoriamente para ejemplificar el comportamiento del robot hexápodo. En la Tabla

Cada subproblema i tiene un vector de pesos diferente para cada función objetivo, y tiene

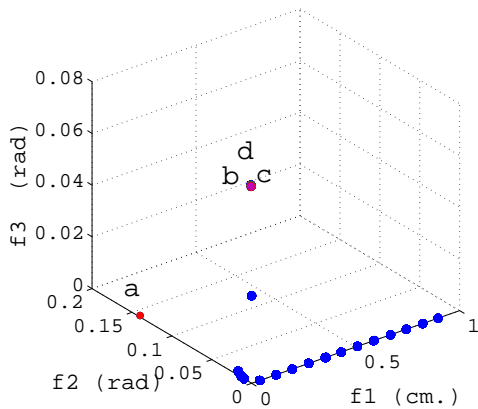


Fig. 7. Frente de Pareto obtenido después de 100 generaciones

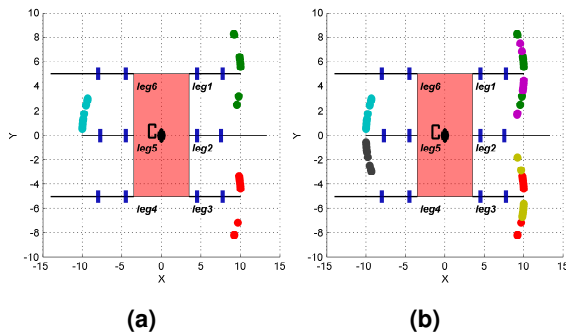


Fig. 8. (a) Campo de trabajo, (b) Campo de trabajo completo de robot hexápodo

2 se muestran los valores de ángulos, la posición de las tres patas en apoyo en los ejes x y y , así como los valores de las funciones objetivo de las soluciones a, b, c, y d.

En la Figura 8a se observa el campo de trabajo del robot hexápodo obtenido por las mismas soluciones del frente de Pareto mostrado en la Figura 7, estas soluciones solo provocan que el robot se mueva en un sentido ya que el algoritmo multiobjetivo no utiliza ningún criterio para determinar la posición de las patas.

El campo de trabajo completo mostrado en la Figura 8b se obtiene creando soluciones

simétricas a las obtenidas por el algoritmo y uniendo estos dos conjuntos de soluciones.

En la Figura 9 se observan cuatro soluciones representativas del conjunto óptimo de Pareto, donde se muestra la nueva posición de las tres patas (1, 3 y 5) que provocan el movimiento del robot y el triángulo de apoyo permanece en la misma posición.

En la misma figura se muestra que lo que se mueve son las patas que forman el triángulo de apoyo y no el cuerpo, esto se debe a que en la herramienta *Robotics* de Matlab, el sistema de referencia del modelo cinemático del robot es la parte superior del cuerpo, sin embargo, en un robot real, el sistema de referencia es el piso, por lo que el cuerpo del robot es lo que se mueve.

En la Figura 9b se muestra una solución dentro del espacio de trabajo que no es viable ya que el centro geométrico del cuerpo superior del robot esta fuera del triángulo de apoyo.

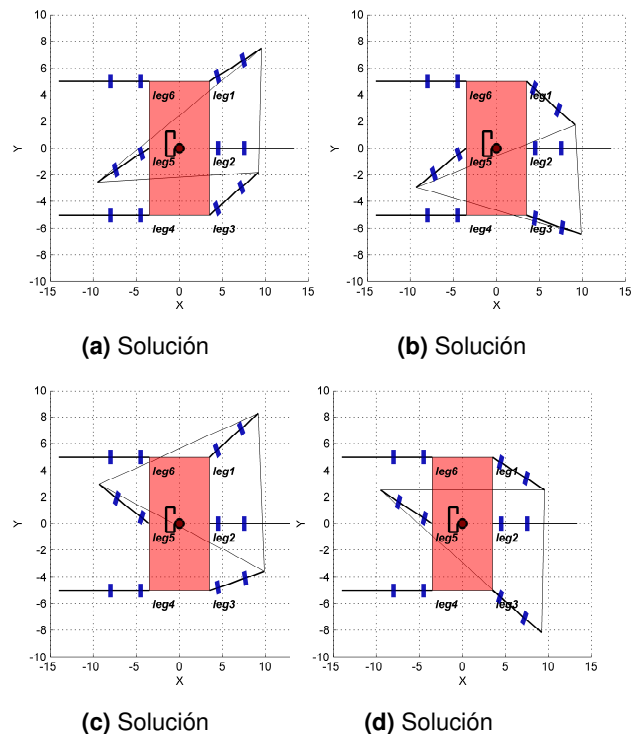


Fig. 9. Comportamiento de un robot hexápodo para cuatro soluciones óptimas de Pareto

Tabla 2. Muestra de soluciones óptimas de Pareto

Solución	Pata	Ángulo	Posición		Función objetivo		
		θ_1	x	y	f_1	f_2	f_3
a	1	0.3883	9.544	7.472	0.0001	0.0009	0.0764
	3	0.5103	9.198	-1.81			
	5	0.3975	-9.52	-2.528			
b	1	-0.5236	9.155	1.735	0	0.014	0.0006
	3	-0.2223	9.869	-6.44			
	5	0.4682	-9.327	-2.947			
c	1	-0.3883	9.544	2.528	0.0001	0.0009	0.0764
	3	-0.5103	9.198	-8.19			
	5	-0.3975	-9.521	2.528			
d	1	-0.3883	9.544	2.528	0.0001	0.0009	0.0759
	3	-0.5069	9.209	-8.17			
	5	-0.3975	-9.521	2.528			

5. Conclusiones

En este trabajo se analizó un robot hexápodo como un robot paralelo tipo Delta, y se propuso un método innovador para obtener el campo de trabajo, ya que de acuerdo a las fuentes bibliográficas consultadas, éstas se enfocan a que el robot avance en línea recta.

El método propuesto permite encontrar los posibles movimientos que puede realizar un robot hexápodo, y tiene como propósito mantener en la misma posición sobre el piso al triángulo de apoyo después del movimiento, lo cual se planteó como un problema de optimización multiobjetivo. El método propuesto permite que un robot hexápodo se pueda mover en todas las direcciones durante la etapa de avance del caminado trípode.

Referencias

- Akda, M., Karagulle, H., & Malgaca, L. (2012).** An integrated approach for simulation of mechatronic systems applied to a hexapod robot. *Mathematics and Computers in Simulation*, Vol. 82, No. 5, pp. 818–835.
- Borras, J. & Dollar, A. M. (2014).** Analyzing dexterous hands using a parallel robots framework. *Autonomous Robots*, Vol. 36, No. 1-2, pp. 169–180.
- Chen, W., Ren, G., Zhang, J., & Wang, J. (2012).** Smooth transition between different gaits of a hexapod robot via a central pattern generators algorithm. *Journal of Intelligent and Robotic Systems*, Vol. 67, No. 3-4.
- Corke, P. I. (2011).** *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer.
- Erden, M. S. & Leblebicioglu, K. (2008).** Free gait generation with reinforcement learning for a six-legged robot. *Robotics and Autonomous Systems*, Vol. 56, No. 3, pp. 199–212.
- Fuente, L., Lones, M., Turner, A., Caves, L., Stepney, S., & Tyrrell, A. (2013).** Adaptive robotic gait control using coupled artificial signalling networks, hopf oscillators and inverse kinematics. *Evolutionary Computation (CEC), IEEE Congress on*, pp. 1435–1442.
- Juang, C.-F., Chang, Y.-C., & Hsiao, C.-M. (2011).** Evolving gaits of a hexapod robot by recurrent neural networks with symbiotic species-based particle swarm optimization. *Industrial Electronics, IEEE Transactions on*, Vol. 58, No. 7, pp. 3110–3119.
- Karimi, D. & Nategh, M. J. (2014).** Kinematic nonlinearity analysis in hexapod machine tools: Symmetry and regional accuracy of workspace. *Mechanism and Machine Theory*, Vol. 71, pp. 115–125.
- Mănoiu-Olaru, S. & Ntulescu, M. (2013).** Stability analysis software platform dedicated for a hexapod robot. In **Dumitrache, L.**, editor, *Advances in Intelligent Control Systems and Computer Science*, volume 187 of *Advances in Intelligent Systems and Computing*. Springer Berlin Heidelberg, pp. 143–156.

10. **Nishii, J. (2000).** Legged insects select the optimal locomotor pattern based on the energetic cost. *Biological Cybernetics*, Vol. 83, No. 5, pp. 435–442.
11. **Pap, Z., Kecskes, I., Burkus, E., Bazso, F., & Odry, P. (2010).** Optimization of the hexapod robot walking by genetic algorithm. *Intelligent Systems and Informatics (SISY), 8th International Symposium on*, pp. 121–126.
12. **Ren, G., Chen, W., Dasgupta, S., Kolodziejcki, C., Worgotter, F., & Manoonpong, P. (2015).** Multiple chaotic central pattern generators with learning for legged locomotion and malfunction compensation. *Information Sciences*, Vol. 294, pp. 666–682. *Innovative Applications of Artificial Neural Networks in Engineering*.
13. **Rolland, L. & Chandra, R. (2016).** The forward kinematics of the 6-6 parallel manipulator using an evolutionary algorithm based on generalized generation gap with parent-centric crossover. *Robotica*, Vol. 34, pp. 1–22.
14. **Roy, S. S. & Pratihar, D. K. (2012).** Effects of turning gait parameters on energy consumption and stability of a six-legged walking robot. *Robotics and Autonomous Systems*, Vol. 60, No. 1, pp. 72–82.
15. **Rushworth, A., Cobos-Guzman, S., Axinte, D., & Raffles, M. (2015).** Pre-gait analysis using optimal parameters for a walking machine tool based on a free-leg hexapod structure. *Robotics and Autonomous Systems*, Vol. 70, pp. 36–51.
16. **Simionescu, I., Ciupitu, L., & Ionita, L. C. (2015).** Static balancing with elastic systems of DELTA parallel robots. *Mechanism and Machine Theory*, Vol. 87, pp. 150–162.
17. **Sorin, M. & Mircea, N. (2012).** Matlab simulation interface for locomotion analysis of a hexapod robot structure. *System Theory, Control and Computing (ICSTCC), 16th International Conference on*, pp. 1–6.
18. **Wang, J. & Gosselin, C. M. (1999).** Static balancing of spatial three-degree-of-freedom parallel mechanisms. *Mechanism and Machine Theory*, Vol. 34, No. 3, pp. 437–452.
19. **Zhang, Q. & Li, H. (2007).** A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, Vol. 2007.

*Article received on 09/05/2016; accepted on 07/02/2017.
Corresponding author is Josué Domínguez Guerrero.*