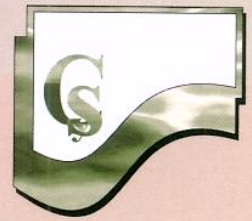




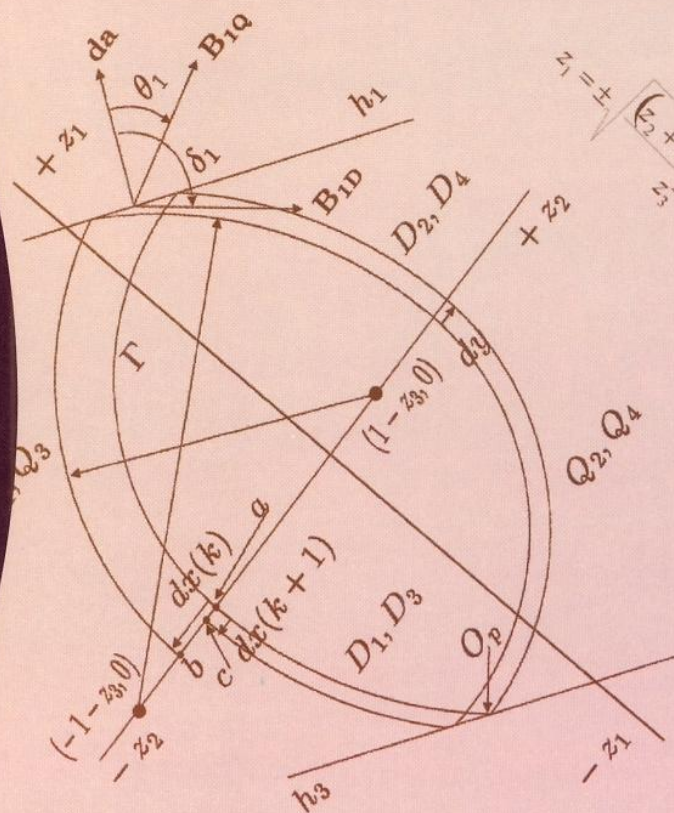
Instituto Politécnico Nacional

Computación y Sistemas



Revista Iberoamericana de Computación
Publicación Trimestral

ISSN 1405-5546

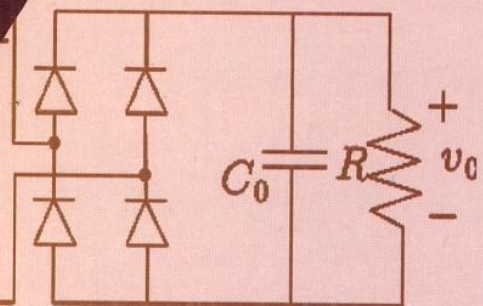


$$z_1 = \pm \frac{(z_2 + z_3 + z_3^2)}{z_3^2} - (z_2 + 1 + z_3)^2$$

$$G(x_1(t), x_2(t)) = \begin{bmatrix} \sigma(x_2(t) - x_1(t)) - \gamma(x_1(t)) \\ \mu_1(t) - \beta_2(t) - x_1(t) \\ x_1(t)x_2(t) - b_2x_1(t) \end{bmatrix}$$

$$C_{1k} = F(X_k); \quad C_{2k} = F(X_k + C_{1k}/2);$$

$$C_{3k} = F(X_k + C_{2k}/2); \quad C_{4k} = F(X_k + C_{3k}).$$



Computación y Sistemas

Revista Iberoamericana de Computación

Editores en Jefe Chief Editors

Juan Humberto Sossa Azuela
Centro de Investigación en
Computación (IPN),
México

Isaac Scherson
University of California,
Irvine, USA

Gerhard X Ritter
University of Florida,
USA

Coordinación Editorial
Blanca Lidia Miranda Valencia

Coordinación de Arbitraje
Lourdes Salazar Rodríguez

Diseño y Formación
Lourdes Salazar Rodríguez

Coordinación de Producción
Miguel Angel Tenorio Trejo

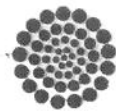
Impresión
Talleres Gráficos del
Instituto Politécnico Nacional
Dirección de Publicaciones
Portadas: Diseños y Sistemas de
México, S.A., de C.V.

Suscripciones
Claudia Elizabeth Andrade Trujillo

Publicación incluida en: el Índice de Revistas Mexicanas de
Investigación Científica y Tecnológica del Conacyt-México,
Periódica: Índice de Revistas Latinoamericanas en
Ciencias, E-Journal, Latindex, Redalyc y Scielo.



INSTITUTO POLITÉCNICO
NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN



CONACYT
CONSEJO NACIONAL
DE CIENCIA Y TECNOLOGÍA

Editores Asociados Associate Editors

Alberto Del Bimbo
U. de Florencia, Italia

Alberto Sanfeliú
U. Politécnica de
Cataluña, España

Alexander Gelbukh
Centro de Investigación
en Computación (IPN),
México

Alexander Poznyak
Centro de Investigación
y Estudios Avanzados (IPN),
México

Angélica de Antonio Jiménez
U. Politécnica de Madrid, España

Carlos Coello
Centro de Investigación
y Estudios Avanzados (IPN),
México

Claude Girault
U. de Paris VI, Francia

Cristina Verde
U. Nacional Autónoma
de México, México

David Muñoz
I. Tec. Estudios Superiores
de Monterrey, México

Edgar Sánchez
Centro de Investigación
y Estudios Avanzados (IPN),
México

Enrique Sucar
INAOE
México

Ernesto Bribiesca
U. Nacional Autónoma
de México, México

Ernesto Suaste
Centro de Investigación
y Estudios Avanzados (IPN),
México

Evguenii Kourmychev
Centro de Investigación en
Óptica, México

Gustavo Rossi
U. Nacional de
la Plata, Argentina

Hebert Sira
Centro de Investigación
y Estudios Avanzados (IPN),
México

Ignacio Canals
U. Autónoma Metropolitana,
México

Ioannis Kakadiaris
U. de Houston, USA

Ivan Terol
CENIDET
Cuernavaca, Mor., México

Jesús Favela
CICESE, México

Jihad El-Sana
Ben-Gurion University of the
Nagev, Israel

Joaquín Álvarez
CICESE, México

José Luis Marroquín
Centro de Investigación
en Matemáticas,
México

José Ruiz Shulcloper
Centro de Aplicaciones
de Tecnologías Avanzadas,
Cuba

Juan Manuel Ahuactzin
Probayes, Francia

Leo Joskowics
U. de Jerusalem, Israel

Leonid Sheremetov
Instituto Mexicano del
Petróleo, México

Luis Camarinha Matos
U. Nueva de Lisboa,
Portugal

María Vargas
The Open University,
England

Mateo Valero
U. Politécnica de
Cataluña, España

Miguel Katrib
U. de la Habana,
Cuba

Oscar Pastor
U. Politécnica de
Valencia, España

Pedro Ma. Larrañaga
U. Politécnica de Valencia, España

Ramón Rodríguez
I. Tec. Estudios Superiores
de Monterrey, México

Raúl Rojas
Universidad Libre de Berlín,
Alemania

Rodolfo Pazos
CENIDET
Cuernavaca, Mor., México

Rogelio Lozano
U. de Technologie de Compiègne,
Francia

Serguei Levachkine
Centro de Investigación
en Computación (IPN),
México

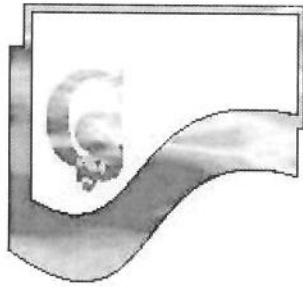
Tomás Lang
U. de California, USA

Ulises Cortés
U. Politécnica de
Cataluña, España

Valeri Kontorovitch
Centro de Investigación
y Estudios Avanzados (IPN),
México

Vladimir Riazanov
Academia de Ciencias
de Rusia, Rusia





Computación y Sistemas

Vol. 11 No. 3 Enero – Marzo / 2008
January – March / 2008

Editorial

Juan Manuel Ahuactzin

Using UML State Diagrams for Modelling the Performance of Parallel Programs 199
Jorge L. Ortega Arjona

Potential and Actual Collaboration Support for Distributed Pair-Programming 211
*Alberto L. Morán, Jesús Favela, Raúl Romero, Hiroshi Natsu, Cynthia Pérez,
Omar Robles and Ana María Martínez Enriquez*

Color Matching using Time Series Searching in Color Database 230
*E. M. Felipe Riverón, J. G. Figueroa Nazuno, A. F. Gutiérrez Tornés,
R. Barrón Fernández and F. I. Cervantes Alarcón*

Dinamyc Behavior of Contaminants in the Water Distribution Network of Cuernavaca México, a Real Application of Multiobjective Distributed Reinforcement Learning 242
Carlos Eduardo Mariano Romero

Algoritmo de Locomoción Libre en un Robot Caminante de Seis Patas 260
Efrén Gorrostieta y Emilio Vargas Soto

Resumen de Tesis Doctoral
Recuperación de Información con Resolución de Ambigüedad de Sentidos de Palabras para el Español 288
Yoel Ledo Mezquita

Anuncios y Eventos

Algoritmo Difuso de Locomoción Libre para un Robot Caminante de Seis Patas

Fuzzy Algorithm of Free Locomotion for a Six Legged Walking Robot

Efrén Gorrostieta¹ y Emilio Vargas Soto²

¹Universidad Autónoma de Querétaro Facultad de Informática, ²Universidad Anáhuac México Sur, Facultad de Ingeniería.
efrengorrostieta@gmail.com; emilio.vargas@anahuac.mx

Artículo recibido en Octubre 03, 2003; aceptado en Julio 02, 2007

Resumen

Este artículo muestra un conjunto de algoritmos que permiten a un robot caminante de seis patas realizar su desplazamiento mediante acciones no establecidas ni periódicas. Este modo de caminar es conocido como: locomoción libre. El desarrollo de estos algoritmos utiliza técnicas de lógica difusa para la toma de decisiones. La valoración de los algoritmos se realiza mediante la simulación del proceso de locomoción del robot. Así mismo, se muestra un algoritmo de control neuronal utilizado para dirigir adecuadamente la ejecución de la locomoción del robot.

Palabras Claves: Robot Caminante, Locomoción libre, Espacio de trabajo, Margen de Estabilidad, Lógica Difusa, Algoritmo Adaptivo, Algoritmo Predictivo.

Abstract

In this paper a set of algorithms that allows a six-legged walking robot perform its displacement by means of a non-established and non-periodic tasks is presented. The development of these algorithms uses fuzzy logic techniques in order to take decisions. To evaluate the algorithms, a robot locomotion process simulation is performed. Also, a neural network control is used to adequately direct the execution of the robot's locomotion.

Keywords: Walking Robot, free locomotion, working space, stability margin, fuzzy logic, adaptive algorithm, predictive algorithm.

1 Introducción

En aplicaciones donde la superficie de desplazamiento es totalmente irregular y desconocida, el uso de los robot móviles con ruedas presenta un sin número de problemas principalmente de estabilidad del vehículo y acceso a ciertas partes (Celaya, 1998). Es en éste tipo de aplicaciones donde el uso de robots caminantes presenta un mejor desempeño (Estremera y González 2002).

Al considerar nuestro entorno, éste está fabricado para las características antropomórficas de un ser caminante natural bípedo, que es el ser humano. Si analizamos casi todas las construcciones, éstas están diseñadas para que el ser humano se desplace de un lugar a otro. Es tan marcado éste tipo de ambiente que las personas que utilizan una silla de ruedas para desplazarse requieran de construcciones especiales para poder compartir los espacios.

En los últimos años se han estudiado diversos diseños de robots caminantes, con el propósito de lograr nuevos tipos de negocios en aplicaciones de robots caminantes tanto comerciales, industriales y de entretenimiento. Como ejemplo, uno de estos robots es el desarrollado en la Universidad Politécnica Salesiana (Calle, 2007) o el robot desarrollado por la empresa japonesa Sony Co. (Masuhiro, 2001). Las investigaciones de máquinas caminantes ayudan a explicar de cierta manera el comportamiento de la locomoción animal (Shaoping, 2001), teniendo en cuenta que muchas de las características de la locomoción, para éste tipo de máquinas caminantes, se basan en la observación de algunos animales. Por otro lado es necesario considerar varios de los siguientes aspectos:

Evitar que el robot pierda el equilibrio. Esto se logra observando la ubicación del centro de gravedad dentro de las patas que soportan el cuerpo cuando éste se desplace o esta fijo.

Distribución de la carga soportada por el robot en cada una de las patas que esta en contacto con la superficie.

Control de la movilidad de las patas. El desplazamiento de las patas dentro de su espacio de trabajo, evitando movimientos fuera de su límite, o bien, colisiones entre las patas; o entre una pata y el cuerpo. Reduciendo así daños al robot o deterioros serios en alguna de sus partes.

Búsqueda de soporte de las patas. Precisar el lugar donde las patas van a soportar el robot, en el caso de superficies planas esta tarea no es complicada, pero en el caso donde tenemos una superficie irregular y con cierto tipo de obstáculos esta definición se complica.

El desarrollo de las técnicas más recientes de reconocimiento de imágenes, así como el sensado de las variables asociadas al control dinámico y cinemático del robot, permiten un buen desempeño de su andar.

Gestionar la manera de caminar libremente, es una tarea sumamente compleja. El desarrollo de los algoritmos de locomoción libres en los robots caminantes ha sido por varios años tema de estudio e investigación en diferentes lugares del mundo. El problema no se ha resuelto del todo, por lo que aun se requiere de una mayor investigación que permita un mejor conocimiento del proceso de caminar en éste tipo de máquinas.

En éste trabajo se presentan el desarrollo de los algoritmos de locomoción libre en un robot caminante de seis patas, así como las estructuras de control requeridas para que estos se realicen. Específicamente, se utilizan técnicas de control neuronal en cada una de las patas articuladas del robot.

La configuración del robot hexápodo se presenta en la figura 1. La configuración del robot está basada en la morfología natural de una hormiga como se plantea en (Gorrostieta, 2000, y Solano, 2000).

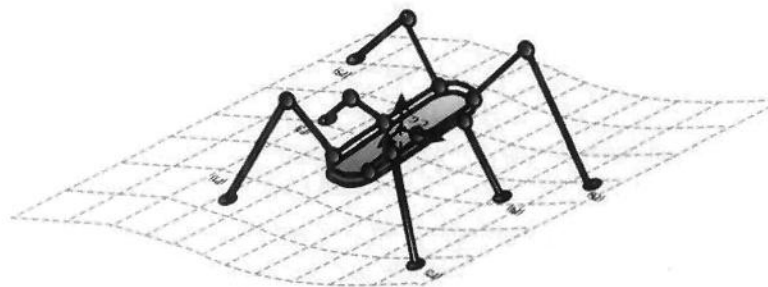


Fig. 1. Configuración de Robot Caminante

Los algoritmos de locomoción del robot caminante estudiados, consideran de forma relevante las características del terreno y el margen de estabilidad del robot. Entre las características del terreno podemos mencionar: espacios prohibidos, pendientes y obstáculos.

2 Trayectoria del Paso

Una de las tareas más importantes que tiene el control es el seguimiento de la trayectoria del paso a realizar por el robot. Esta trayectoria está definida principalmente por el tipo de paso a realizar, las restricciones mecánicas, la cinemática y las características del terreno. En la figura 2 se muestra la trayectoria de movimiento conocido como: parabólico. Este movimiento se puede observar mejor en la proyección del plano xz de la trayectoria del paso. Para realizar la construcción de los algoritmos de locomoción, se parametrizó la trayectoria del paso. De forma que a partir de la altura y la longitud del paso, se construye automáticamente dicha trayectoria. De esta forma, a partir de las ecuaciones de trayectoria del paso, se definen puntos específicos de posición y velocidad, los cuales se consideran los puntos de consigna del sistema de control mencionado en la sección anterior.

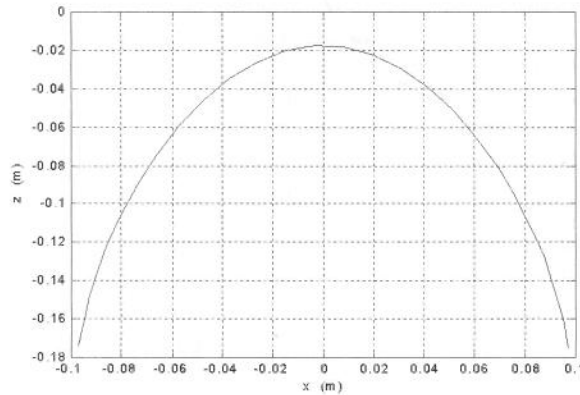


Fig. 2. Trayectoria de Movimiento Parabólico

La trayectoria del paso que se muestra en la figura 2 esta basada en un movimiento similar al que presentan algunos animales con patas como se reportó en (Ilg W. and Mühlfriedel 1999). La ecuación 1 describe el comportamiento de éste movimiento. La trayectoria fue desarrollada e implementada con el propósito de verificar las posiciones en cada uno de los grados de libertad de las patas del robot.

$$\begin{aligned}\theta_1 &= d\gamma - A\gamma(\cos \xi - 1) \\ \theta_2 &= d\beta - A\beta(\cos \xi - 1) \\ \theta_3 &= d\chi - A\chi(\cos \xi - 1)\end{aligned}\quad (1)$$

Donde $d\gamma$, $d\beta$, $d\chi$ son los valores de los ángulos θ_1 , θ_2 , y θ_3 , respectivamente. Estos valores determinan la posición inicial de la articulación en el espacio. La variable $A\gamma$ nos define la longitud del paso y los valores de $A\beta$, $A\chi$, nos define la altura del paso, mientras que ξ se considera como el ángulo de movilidad del robot y varía de θ a π .

En la ecuación 2 se muestran los valores específicos utilizados en la ecuación 1, estos valores incluyen la trayectoria en el espacio de trabajo de la articulación.

$$\begin{aligned}d\gamma &= 70 & A\gamma &= 20 \\ d\beta &= 4 & A\beta &= 15 \\ d\chi &= 310 & A\chi &= 10\end{aligned}\quad (2)$$

En la ecuación 3 se muestran las ecuaciones desarrolladas para determina las trayectorias para generar un paso de la articulación en forma triangular, donde la constante kl es la variable que introduce la pendiente al final del paso.

$$\begin{aligned}\theta_1 &= d\gamma - A\gamma(\cos \xi - 1) \\ \theta_2 &= d\beta - A\beta(\cos \xi - 1)e^{kl\xi} \\ \theta_3 &= d\chi - A\chi(\cos \xi - 1)e^{kl\xi}\end{aligned}\quad (3)$$

Para la generación de cualquier movimiento en las articulaciones fue necesario verificar la definición del espacio de trabajo de la pata, esto para asegurar que el movimiento generado fuera posible.

Con el propósito de dar seguimiento a las trayectorias generadas automáticamente, y aplicarlas como parte de la estrategia de los algoritmos de locomoción libre, se utilizó, diseño y aplicó un sistema de control neuronal, mismo que se representa en la figura 3.

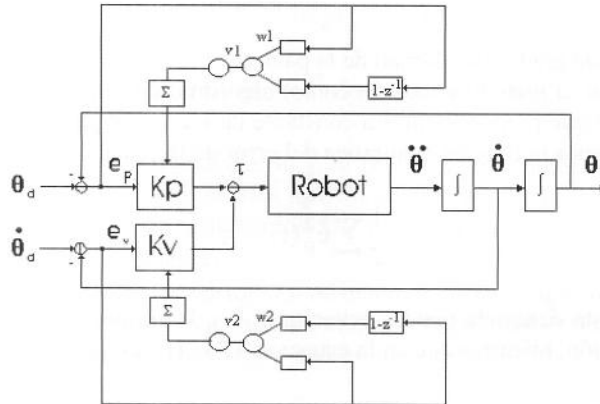


Fig. 3. Esquema del Control Dinámico de la Pata

El sistema no lineal a controlar se expresa en la ecuación 4, que representa el comportamiento dinámico del robot. Donde $D(q)$ es la matriz de inercia, $c(q, \dot{q})$ la matriz de coriolis e inercias, $g(q)$ la componente de la gravedad y $b(\dot{q})$ es el efecto de la fricción, τ el torque requerido por cada motor para realizar el movimiento

$$D(q)\ddot{q} + c(q, \dot{q}) + g(q) + b(\dot{q}) = \tau + \lambda(q, \dot{q}) \quad (4)$$

La ecuación de la ley de control $\tau = f(q, v, r)$ será la que asegure la generación de las trayectorias para construir el paso mostrado en la figura 2. El sistema de control utilizado es el control de PD por compensación de gravedad, propuesto por Tanekgaki y Arimoto (Gorrostieta y Vargas 2002), éste modelo considera la ley de control mostrada en la ecuación 5.

$$\tau = Kpe + Kve + g(q) \quad (5)$$

La ecuación 6 muestra el cálculo del error. Este error se obtiene de la posición deseada para cada una de las articulaciones del robot (q) y la posición real que se esta midiendo del sistema (r).

$$e = r - q \quad (6)$$

Los parámetros de ajuste del control de la ecuación 5 son: los coeficientes del error Kp y los coeficientes de la derivada del error Kv , están definidos como matrices simétricas positivas de $n \times n$. El término de gravedad $g(q)$ que compensa el efecto de la fuerza de gravedad fue aplicado al torque .

Los ajustes de los parámetros K_p y K_v se realizan mediante la ayuda de una red neuronal para cada uno de los mecanismos, en la figura 3 se muestra el sistema de control donde se puede ver como interactúa la red neuronal en el auto ajuste de cada uno de estos parámetros.

La entrada en cada uno de las dos redes es el error y la primera diferencia de error, esto es:

$$x(t) = [e_p(t) \quad \Delta e_p(t)] \quad (7)$$

Siendo $e_p(t)$ el error de cada uno de grados de libertad de la pata.

Las redes son entrenadas por el método conocido como: algoritmo de retro propagación, el cuál se desarrolló para el caso del ajuste de la constante proporcional. La constante derivativa se puede deducir de manera similar. La razón de minimización esta dado por la función cuadrática del error de la ecuación 8.

$$E(t) = \frac{1}{2} \sum e_p^2(t) \quad (8)$$

La ganancia proporcional esta denotada por las ecuaciones 9 y 10, donde v representa los pesos de la última neurona y h la función de activación, mientras que en la ecuación 11 se representa la suma de los pesos de la neurona en la primera capa.

$$K_p = k_p + \Delta k_p \quad (9)$$

$$\Delta k_p = v h \quad (10)$$

$$s = w_1 e_p(t) + w_2 \Delta e_p(t) \quad (11)$$

La función de activación se muestra en la ecuación 12 de las neuronas intermedias es:

$$h = \frac{1}{1 + e^{-s}} \quad (12)$$

Para aplicar el algoritmo de retro propagación fue necesario calcular el gradiente de la función E_p con respecto a los coeficientes v y w

$$\nabla E(t) = \begin{bmatrix} \frac{\partial E(t)}{\partial v_1} \\ \frac{\partial E(t)}{\partial w_j} \end{bmatrix} \quad (13)$$

$$\frac{\partial E_p}{\partial v_1} = -e_p(t)^2 h \text{ sign}(R) \quad (14)$$

$$\frac{\partial E_p}{\partial v_1} = -e_p(t)^2 \text{ sign}(R) v_1 h(1-h)x_j \quad (15)$$

La función $sign(R)$ se evaluó con los valores de $+1$ o -1 . Dependiendo del signo de la relación entre la posición angular y el esfuerzo de torsión aplicado, que se pueden considerar, para el caso de nuestra aplicación se consideró siempre positivo. Las ecuaciones 14 y 15 se desarrollaron para obtener las ecuaciones de la adaptación para los coeficientes del peso de la red, del w y v de las neuronas como se desarrollan en (Gorrostieta et al. 2004). Se utilizó el método de máximo descenso.

$$v_1(t+1) = v_1(t) + \eta e_p(t)^2 h \text{sign}(R) \tag{16}$$

$$w_j(t+1) = w_j(t) + \eta e_p(t)^2 \text{sign}(R) v_1 h(1-h)x_j \tag{17}$$

Las ecuaciones adaptables 16 y 17 se calcularon y se probaron en simulación. Ambas ecuaciones permiten adaptar los valores de los coeficientes proporcionales Kp . Un desarrollo similar, se puede utilizar para encontrar las ecuaciones de adaptación de los coeficientes derivados Kd . El valor η es el coeficiente de aprendizaje de la red como se ha realizado en (Gorrostieta et al. 2004).

El esquema adaptivo de las neuronas desarrollado mediante la metodología anterior permite alcanzar un buen desempeño del control comparado con los esquemas de control PD, control PID control PD con compensación de Gravedad como se presentó en (Gorrostieta y Vargas 2002).

3 Estabilidad del Robot

La estabilidad del robot es una de las características más importantes de su locomoción. En éste trabajo utilizamos el margen de estabilidad, éste parámetro permite valorar el mejor desempeño del robot sobre una trayectoria determinada. Uno de los propósitos del estudio de la estabilidad es primeramente evitar que el robot se caiga cuando éste lleve a cabo un desplazamiento o algún movimiento de las patas.

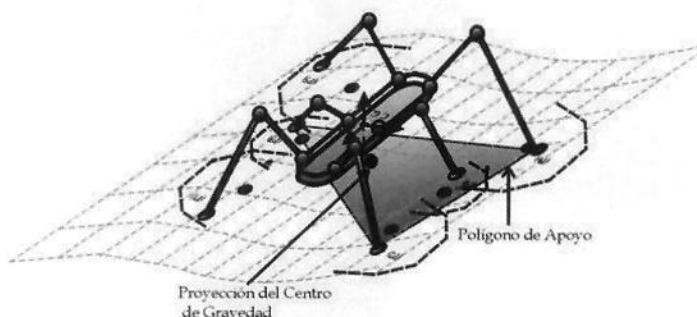


Fig. 4. Polígono de Apoyo del Robot

El robot bajo estudio cuenta con seis patas, las cuales tienen la función de soportar el cuerpo y proporcionar un desplazamiento. Se considera que el número mínimo de patas que aseguran una buena estabilidad es de tres, las cuales al ser apoyadas en la superficie generan un triángulo donde el peso del robot será soportado. Por otra parte, si se apoyan cuatro patas se formará un polígono irregular de cuatro lados, o bien; un polígono de cinco lados en el caso que solamente una de las patas esté en movimiento o fuera del contacto con la superficie de apoyo. Independientemente del número de patas que estén realizando contacto con la superficie de apoyo, la proyección del centro de gravedad debe de estar dentro del área del polígono que se forma con las patas que están soportando el cuerpo del robot. Esta es la condición indispensable que evitará que el robot caiga como se analizó en (Estremera y González 2002). Al polígono que se forma en la superficie por las patas que soportan el cuerpo se le conoce como:

polígono de apoyo. En éste caso en particular, puede ir desde tres lados hasta seis lados cuando todas las patas se encuentren en la superficie. El polígono de apoyo es usado en el proceso de locomoción, la definición del polígono de apoyo es calculado bajo condiciones estáticas, no obstante, éste parámetro se usa en las funciones de movimiento (Vargas 1994). Esto es posible debido a que se considera que el polígono de apoyo no cambia, aún cuando se realiza movimiento en alguna pata a fin del lograr un cambio en la configuración del robot para llegar a la posición deseada.

El margen de estabilidad S_m , permite conocer el grado de estabilidad del robot y está definido como la mínima distancia que hay entre la proyección del centro de gravedad del cuerpo dentro de un margen de seguridad y la frontera que existe en cada uno de los lados del polígono de apoyo generado en el estado actual del robot. La ecuación 18 define el cálculo del margen de estabilidad.

$$S_m = \min(S_1, S_2, S_3, S_4, S_5, S_6) \tag{18}$$

En la figura 5 se muestra el polígono de apoyo. Para el caso en que se tienen tres de las patas apoyadas en la superficie P2- P3-P6, se requiere calcular tres distancias S_1 S_2 y S_3 como se observa en la figura 5.

El margen de seguridad, es un valor que permite asegurar que el centro de gravedad se encuentre dentro del polígono de apoyo generado por la situación actual del robot. Por otro lado, permitirá evitar los problemas que se pueden presentar cuando hay valores muy cerca del límite, complicando la toma de decisión en el algoritmo de locomoción, por éste efecto de valores frontera dentro de los límites del polígono.

El radio de seguridad define un círculo que se encuentra alrededor del centro de gravedad proyectado en el polígono de apoyo. Este parámetro facilita la determinación segura de la locomoción, sobre todo cuando el centro de gravedad esta cerca de uno de los limites del polígono de apoyo.

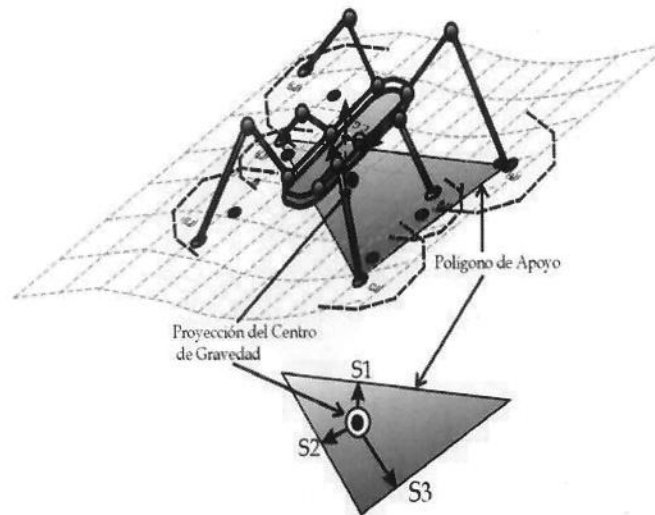


Fig. 5. Margen de Estabilidad del Robot

4 Espacios de Trabajo

El espacio de trabajo de cada una de las patas está definido por el volumen descrito por la cinemática del mecanismo que conforman la pata, particularmente interesa el punto final del la extremidad que tiene contacto con la superficie.

El conocimiento del espacio de trabajo permite saber cuál será la movilidad de cada una de las patas. En la figura 6 se muestra las características del espacio de trabajo de una pata del robot.

En la figura 7 se observa el límite del espacio de trabajo de cada una de las patas del robot. Como puede observarse existen espacios de trabajo que se sobreponen. Por ejemplo: el espacio de trabajo de la pata P4 con los espacios de trabajo de las patas P2 y P6. Así mismo, se presenta esta misma situación con el espacio de trabajo de la pata P3 y los espacios de trabajo de las patas P1 y P5.

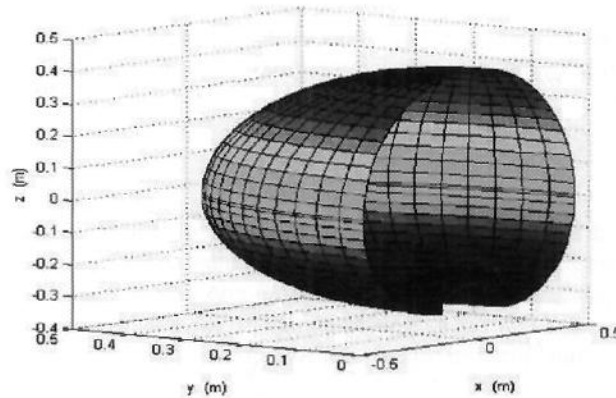


Fig. 6. El Espacio de Trabajo de Una Pata

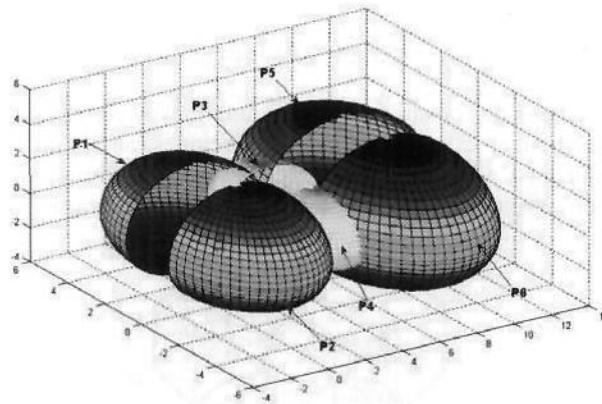


Fig. 7. Los 6 Espacios de Trabajo del Robot

A partir del estudio de estos modelos y las restricciones mecánicas del robot, se determinó que los espacios de trabajo varían en función de la altura del cuerpo del robot con relación a la superficie. Si la superficie está más cerca del cuerpo del robot se logran espacios de trabajo compartidos y a medida que se aleja el cuerpo de la superficie de apoyo se logran espacios de trabajo excluyentes debido a la configuración del robot. En la ecuación 19 se muestra las relaciones que permiten ubicar cada uno de los puntos en el espacio cartesiano con respecto a los ángulos.

$$\begin{aligned}
 x &= (l_2 \cos\theta_2 + l_3 \cos\theta_3) \cos\theta_1 \\
 y &= (l_2 \cos\theta_2 + l_3 \cos\theta_3) \sin\theta_1 \\
 z &= l_2 \sin\theta_2 + l_3 \sin\theta_3
 \end{aligned}
 \tag{19}$$

Donde l_2 y l_3 son las longitudes efectivas de los eslabones que conforman la pata. Siendo θ_1 , θ_2 , y θ_3 la orientación de los eslabones.

El efecto de las diferentes alturas del cuerpo del robot en el plano xy se muestra en la figura 8. En esta figura se muestran las trayectorias obtenidas definidas por la intersección del espacio de trabajo en tres planos xy a diferentes alturas.

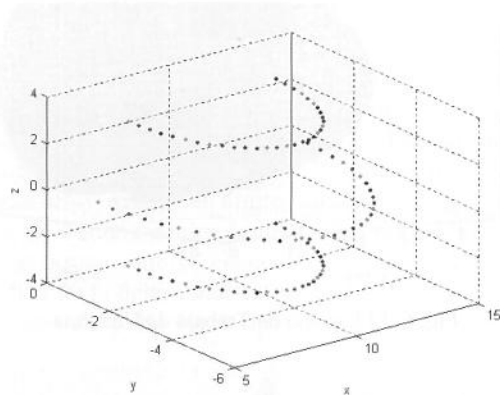


Fig. 8. Cambio de Dimensiones en el Espacio de Trabajo Para Diferentes Valores de Altura

Estas geometrías de los espacios de trabajo permitieron una mayor certeza en el diseño de los algoritmos de locomoción libre. A fin de facilitar el cálculo de estas restricciones, se simplificó la forma de las trayectorias obtenidas, mediante una aproximación por líneas rectas, de acuerdo a la forma que se muestra en la Figura 9. En algunos trabajos previos también se simplifican los espacios de trabajo (Celaya, 1998).

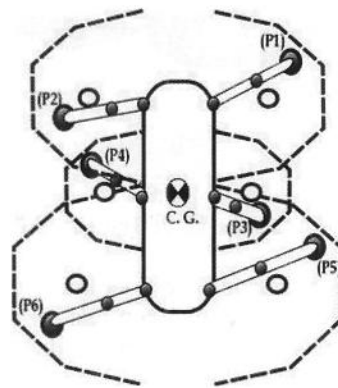


Fig. 9. Espacios de Trabajo Simplificados en el Robot

En la figura 9 se presenta una vista de planta del robot donde se muestran los espacios de trabajo que realmente se utilizaron en el diseño del algoritmo de locomoción, las líneas interrumpidas determinan el límite de espacio de trabajo para cada una de las patas, estas líneas consideran un pequeño margen de seguridad para cada una de las

áreas de trabajo. Otro parámetro importante que se deriva de los espacios de trabajo, es el centro geométrico de cada espacio; así tenemos que en cada uno de los espacios aparece en el centro de éste, un punto que representa dicho centro.

Dado que las dimensiones de los espacios de trabajo varían dependiendo de la posición en que se encuentre el robot con respecto a la superficie, es importante realizar los cálculos de los espacios de trabajo en función de la situación del robot.

El desplazamiento del robot está condicionado a límites de los espacios de trabajo que se presentan en cada una de las patas. Por otra parte, el centro geométrico de los espacios de trabajo nos permite ubicar dimensionalmente los límites de movilidad en el robot, tomando en cuenta dicha ubicación y la distancia máxima posible permitida en la dirección de locomoción deseada.

5 Distancias y Direcciones

Una de las características principales de movimiento del robot es la dirección deseada. Esta dirección es determinada por el navegador, evaluando previamente la tarea a realizar, el objetivo, o tipo de obstáculos que se encuentran en el camino del robot. En la figura 10 se muestra una gráfica donde en una primera aproximación se requiere ir a la posición deseada. El primer cálculo realizado fue el vector de dirección que se genera del centro de gravedad C.G. (x_g, y_g, z_g) al destino. A partir de éste vector se obtiene un vector unitario, que es utilizado para dar sentido y dirección a cada uno de los movimientos de las patas.

El siguiente problema que se presenta es determinar cuál será el punto final del movimiento de cada una de las patas. Hasta éste momento se tiene el punto actual, definido por la posición de la pata del robot o los ángulos de cada uno de los eslabones que forman la articulación. Para encontrar el punto final se debe de conocer la ubicación de cada una de las patas dentro de su espacio de trabajo y las distintas posibilidades de movimiento en éste espacio. Así que, si se conoce un vector que nos permite saber la dirección de éste movimiento V_1, V_2, V_3, V_4, V_5 y V_6 , también se puede restringir el movimiento dentro de las múltiples posibilidades del espacio de trabajo. El punto máximo de desplazamiento se obtiene en la intersección del vector de dirección de cada pata con el límite del espacio de trabajo de la pata que va realizar el movimiento.

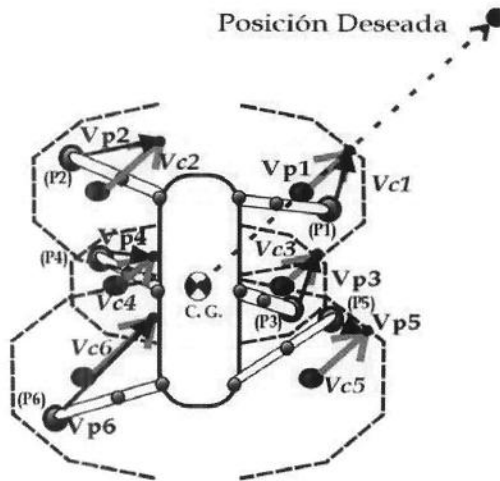


Fig. 10. Vectores de Movimiento en el Espacio de Trabajo

áreas de trabajo. Otro parámetro importante que se deriva de los espacios de trabajo, es el centro geométrico de cada espacio; así tenemos que en cada uno de los espacios aparece en el centro de éste, un punto que representa dicho centro.

Dado que las dimensiones de los espacios de trabajo varían dependiendo de la posición en que se encuentre el robot con respecto a la superficie, es importante realizar los cálculos de los espacios de trabajo en función de la situación del robot.

El desplazamiento del robot está condicionado a límites de los espacios de trabajo que se presentan en cada una de las patas. Por otra parte, el centro geométrico de los espacios de trabajo nos permite ubicar dimensionalmente los límites de movilidad en el robot, tomando en cuenta dicha ubicación y la distancia máxima posible permitida en la dirección de locomoción deseada.

5 Distancias y Direcciones

Una de las características principales de movimiento del robot es la dirección deseada. Esta dirección es determinada por el navegador, evaluando previamente la tarea a realizar, el objetivo, o tipo de obstáculos que se encuentran en el camino del robot. En la figura 10 se muestra una gráfica donde en una primera aproximación se requiere ir a la posición deseada. El primer cálculo realizado fue el vector de dirección que se genera del centro de gravedad C.G. (x_g, y_g, z_g) al destino. A partir de éste vector se obtiene un vector unitario, que es utilizado para dar sentido y dirección a cada uno de los movimientos de las patas.

El siguiente problema que se presenta es determinar cuál será el punto final del movimiento de cada una de las patas. Hasta éste momento se tiene el punto actual, definido por la posición de la pata del robot o los ángulos de cada uno de los eslabones que forman la articulación. Para encontrar el punto final se debe de conocer la ubicación de cada una de las patas dentro de su espacio de trabajo y las distintas posibilidades de movimiento en éste espacio. Así que, si se conoce un vector que nos permite saber la dirección de éste movimiento $V1, V2, V3, V4, V5$ y $V6$, también se puede restringir el movimiento dentro de las múltiples posibilidades del espacio de trabajo. El punto máximo de desplazamiento se obtiene en la intersección del vector de dirección de cada pata con el límite del espacio de trabajo de la pata que va realizar el movimiento.

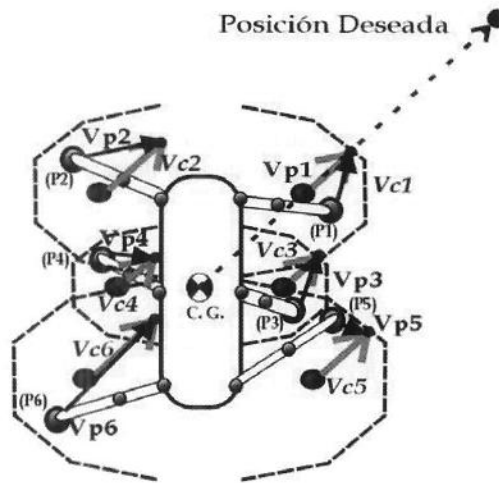


Fig. 10. Vectores de Movimiento en el Espacio de Trabajo

En la figura 10 se muestra la ubicación de las patas dentro del espacio de trabajo, así como también se ilustran los centros geométricos de cada uno de ellos representados en la ecuación 20. Para determinar el movimiento se consideró un vector de dirección, calculado desde la posición actual de la pata, el vector del centro geométrico y la intersección con el límite del espacio de trabajo, determinando I . Ver las coordenadas en la ecuación 21.

Como se puede observar en la figura 10, el vector formado desde el centro geométrico hacia el destino, se denota como \vec{V}_c y el número de pata respectiva. Este vector intercepta la frontera del espacio de trabajo, es precisamente en ese punto donde consideramos el punto final del movimiento de la pata. En la ecuación 22 se muestran los vectores para tres de las patas. El movimiento de cada uno de las patas se describe por el vector \vec{V}_p y el número de la pata en movimiento, representado en la ecuación 23 para las tres primeras patas, de manera similar se representan las otras tres patas faltantes.

$$\begin{matrix} c_1(x_{c1}, y_{c1}) & c_2(x_{c2}, y_{c2}) & c_3(x_{c3}, y_{c3}) \\ c_4(x_{c4}, y_{c4}) & c_5(x_{c5}, y_{c5}) & c_6(x_{c6}, y_{c6}) \end{matrix} \quad (20)$$

$$\begin{matrix} I_1(x_{i1}, y_{i1}) & I_2(x_{i2}, y_{i2}) & I_3(x_{i3}, y_{i3}) \\ I_4(x_{i4}, y_{i4}) & I_5(x_{i5}, y_{i5}) & I_6(x_{i6}, y_{i6}) \end{matrix} \quad (21)$$

$$\begin{aligned} \vec{V}_{c1} &= (x_{i1} - x_{c1})\hat{i} + (y_{i1} - y_{c1})\hat{j} \\ \vec{V}_{c2} &= (x_{i2} - x_{c2})\hat{i} + (y_{i2} - y_{c2})\hat{j} \\ \vec{V}_{c3} &= (x_{i3} - x_{c3})\hat{i} + (y_{i3} - y_{c3})\hat{j} \end{aligned} \quad (22)$$

$$\begin{aligned} \vec{V}_{p1} &= (x_{i1} - x_1)\hat{i} + (y_{i1} - y_1)\hat{j} \\ \vec{V}_{p2} &= (x_{i2} - x_2)\hat{i} + (y_{i2} - y_2)\hat{j} \\ \vec{V}_{p3} &= (x_{i3} - x_3)\hat{i} + (y_{i3} - y_3)\hat{j} \end{aligned} \quad (23)$$

La magnitud del desplazamiento de la pata se calculó a partir de la distancia de cada pata al punto I como se muestra para tres de las patas en la ecuación 24. Este valor de desplazamiento puede ser modificado por una variable δ de tal manera que el movimiento de la pata se vea limitado y no siempre se desplace al límite del espacio de trabajo.

$$\begin{aligned} dez_1 &= \delta_1 \sqrt{(x_{i1} - x_1)^2 + (y_{i1} - y_1)^2} \\ dez_2 &= \delta_2 \sqrt{(x_{i2} - x_2)^2 + (y_{i2} - y_2)^2} \\ dez_3 &= \delta_3 \sqrt{(x_{i3} - x_3)^2 + (y_{i3} - y_3)^2} \end{aligned} \quad (24)$$

El cálculo de δ como se muestra en la ecuación 25 es un exponencial decreciente.

$$\delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{bmatrix} = \begin{bmatrix} e^{-\gamma_1} \\ e^{-\gamma_2} \\ e^{-\gamma_3} \\ e^{-\gamma_4} \\ e^{-\gamma_5} \\ e^{-\gamma_6} \end{bmatrix} \quad (25)$$

La potencia de la exponencial γ se ordenó en un vector como se puede ver en la ecuación 26. En una posición inicial éste vector se inicializa en ceros, para posteriormente ir determinado el valor de cada uno de estos valores según la distancia del paso que se desea ejecutar.

$$\gamma = [\gamma_1 \ \gamma_2 \ \gamma_3 \ \gamma_4 \ \gamma_5 \ \gamma_6] \quad (26)$$

6 Algoritmos de Locomoción

La investigación de la locomoción libre para los robots caminantes, es un desarrollo que aspira a realizar los desplazamientos necesarios por el robot cuando éste camina en diferentes tipos de superficie. Así como también se pretende lograr un entendimiento sobre la forma caminar que tienen algunos insectos.

Para explicar el diseño de los algoritmos de locomoción libre aplicados a un robot hexápodo, primeramente se propone una división en siete áreas de trabajo, las cuales son formadas por las intersecciones de las líneas que van entre cada una de las patas tal como se muestra en la figura 11. Cada área es etiquetada por una letra.

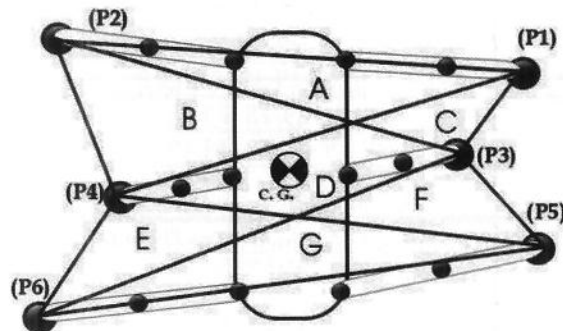


Fig. 11. Áreas Determinadas del Polígono de Apoyo

Tabla 1. Coordenadas de los límites de cada una de las Áreas

Área	Coordenadas
A	$P1(x_1, y_1), P2(x_2, y_2), Pi1(x_{i1}, y_{i1})$ $x_{i1} = \frac{(x_1 y_4 - x_4 y_1)(x_3 - x_2) - (x_2 y_3 - x_3 y_2)(x_4 - x_1)}{(y_2 - y_3)(x_4 - x_1) - (y_1 - y_4)(x_3 - x_2)}$ $y_{i1} = \frac{y_1 - y_4}{x_4 - x_1} x_{i1} - \frac{x_1 y_4 - x_4 y_1}{x_4 - x_1}$
B	$P2(x_2, y_2), P4(x_4, y_4), Pi1(x_{i1}, y_{i1})$ $x_{i1} = \frac{(x_1 y_4 - x_4 y_1)(x_3 - x_2) - (x_2 y_3 - x_3 y_2)(x_4 - x_1)}{(y_2 - y_3)(x_4 - x_1) - (y_1 - y_4)(x_3 - x_2)}$ $y_{i1} = \frac{y_1 - y_4}{x_4 - x_1} x_{i1} - \frac{x_1 y_4 - x_4 y_1}{x_4 - x_1}$
C	$P1(x_1, y_1), P3(x_3, y_3), Pi1(x_{i1}, y_{i1})$ $x_{i1} = \frac{(x_1 y_4 - x_4 y_1)(x_3 - x_2) - (x_2 y_3 - x_3 y_2)(x_4 - x_1)}{(y_2 - y_3)(x_4 - x_1) - (y_1 - y_4)(x_3 - x_2)}$ $y_{i1} = \frac{y_1 - y_4}{x_4 - x_1} x_{i1} - \frac{x_1 y_4 - x_4 y_1}{x_4 - x_1}$
D	$P4(x_4, y_4), P3(x_3, y_3), Pi1(x_{i1}, y_{i1}), Pi2(x_{i2}, y_{i2})$ $x_{i1} = \frac{(x_1 y_4 - x_4 y_1)(x_3 - x_2) - (x_2 y_3 - x_3 y_2)(x_4 - x_1)}{(y_2 - y_3)(x_4 - x_1) - (y_1 - y_4)(x_3 - x_2)}$ $y_{i1} = \frac{y_1 - y_4}{x_4 - x_1} x_{i1} - \frac{x_1 y_4 - x_4 y_1}{x_4 - x_1}$ $x_{i2} = \frac{(x_3 y_6 - x_6 y_3)(x_5 - x_4) - (x_4 y_5 - x_5 y_4)(x_6 - x_3)}{(y_4 - y_5)(x_6 - x_3) - (y_3 - y_6)(x_5 - x_4)}$ $y_{i2} = \frac{y_3 - y_6}{x_6 - x_3} x_{i2} - \frac{x_3 y_6 - x_6 y_3}{x_6 - x_3}$
E	$P6(x_6, y_6), P4(x_4, y_4), Pi2(x_{i2}, y_{i2})$ $x_{i2} = \frac{(x_3 y_6 - x_6 y_3)(x_5 - x_4) - (x_4 y_5 - x_5 y_4)(x_6 - x_3)}{(y_4 - y_5)(x_6 - x_3) - (y_3 - y_6)(x_5 - x_4)}$ $y_{i2} = \frac{y_3 - y_6}{x_6 - x_3} x_{i2} - \frac{x_3 y_6 - x_6 y_3}{x_6 - x_3}$
F	$P5(x_5, y_5), P3(x_3, y_3), Pi2(x_{i2}, y_{i2})$ $x_{i2} = \frac{(x_3 y_6 - x_6 y_3)(x_5 - x_4) - (x_4 y_5 - x_5 y_4)(x_6 - x_3)}{(y_4 - y_5)(x_6 - x_3) - (y_3 - y_6)(x_5 - x_4)}$ $y_{i2} = \frac{y_3 - y_6}{x_6 - x_3} x_{i2} - \frac{x_3 y_6 - x_6 y_3}{x_6 - x_3}$
G	$P5(x_5, y_5), P6(x_6, y_6), Pi2(x_{i2}, y_{i2})$ $x_{i2} = \frac{(x_3 y_6 - x_6 y_3)(x_5 - x_4) - (x_4 y_5 - x_5 y_4)(x_6 - x_3)}{(y_4 - y_5)(x_6 - x_3) - (y_3 - y_6)(x_5 - x_4)}$ $y_{i2} = \frac{y_3 - y_6}{x_6 - x_3} x_{i2} - \frac{x_3 y_6 - x_6 y_3}{x_6 - x_3}$

En la tabla 1 se presenta las coordenadas que delimitan cada una de las áreas definidas por los polígonos de apoyo. La región D que esta formada por cuatro vértices. La dificultad de apoyo radica en evaluar las condiciones de soporte, movilidad y estabilidad en las diferentes alternativas. Por ejemplo, el área de trabajo A es un triángulo cuyos vértices que delimitan su área se encuentran en la posiciones de la pata P1, P2 y el punto de intersección Pi1, con posibilidades de gestionar el movimiento del robot soportando el peso con una pata adicional a P1 y P2, bien sea P4 o P3. O bien soportando al robot con dos patas diferentes, adicionalmente a P1 y P2, con P4 o P6, o bien con P3 y P5.

Una vez determinadas las siete áreas posibles, se realiza una proyección de centro de gravedad del robot considerando el margen de seguridad y se observa en qué área queda incluido el centro de gravedad, en caso de que

éste se encuentre en los límites se realiza un movimiento del cuerpo del robot de tal manera que el centro de gravedad quede incluido completamente dentro de las áreas definidas por las patas que soportan el peso del robot.

La tabla 2 muestra las diferentes posibilidades de movimiento para una sola pata considerando que el centro de gravedad se encuentra en las distintas regiones mostradas en la figura 11.

Tabla 2. Movilidad en Una Pata

Área	Movilidad de una Pata
A	P3 P4 P5 P6
B	P3 P4 P5 P6
C	P3 P4 P5 P6
D	P1 P2 P3 P4 P5 P6
E	P1 P2 P3 P4
F	P1 P2 P3 P4
G	P1 P2 P3 P4

Cuando se levanta la pata como es el caso que se propone en la tabla 2, el robot queda soportado en un polígono de cinco lados, en donde el centro de gravedad proyectado en la superficie, deberá de estar dentro de esta área para asegurar la estabilidad como se describió con anterioridad. Para la primera situación donde el centro de gravedad se encuentre en el área A el robot podrá levantar las patas P3, P4, P5 y P6 sin perder el equilibrio.

Tabla 3. Movilidad en Dos Patas

Área	Movilidad de dos Patas
A	P1-P5 P1-P6 P3-P4 P3-P6 P4-P5
B	P3-P4 P3-P5 P4-P5 P4-P6 P5-P6
C	P2-P5 P2-P6 P3-P6 P4-P6 P5-P6
D	P1-P5 P1-P6 P2-P5 P2-P6 P3-P4
F	P1-P2 P1-P3 P1-P5 P2-P5 P3-P4
F	P1-P2 P1-P4 P1-P3 P2-P3 P3-P4
G	P1-P2 P1-P6 P2-P4 P2-P6 P3-P4

En la tabla 3 se muestra las posibilidades del movimiento de las patas del robot considerando, dos patas en movimiento, en éste caso el polígono de apoyo tendrá cuatro lados en donde nuevamente el centro de gravedad debe de estar comprendido dentro de dicha área.

Tabla 4. Movimiento de Tres Patas

Área	Movilidad de tres Patas
A	P1-P4-P5 P1-P5-P6 P3-P4-P6 P3-P5-P6
B	P3-P4-P5 P3-P4-P6 P3-P5-P6 P4-P5-P6
C	P2-P3-P6 P4-P5-P6 P2-P5-P6 P3-P4-P6
D	P1-P4-P5 P2-P3-P6
E	P1-P2-P3 P1-P4-P5 P1-P2-P5 P1-P4-P3
F	P1-P2-P4 P1-P4-P3 P1-P2-P3 P2-P4-P3
G	P1-P2-P4 P1-P2-P6 P2-P5-P3 P2-P3-P4

En la tabla 4 se muestran los posibles movimientos para las tres patas, en éste caso el polígono de apoyo será un triángulo; quedando así la posibilidad mínima de estabilidad en el robot.

Con la información que se presenta hasta el momento se plantean los algoritmos de locomoción libre, para una sola de las patas, el diagrama de flujo de éste algoritmo se puede ver en la figura 12. Considerando que se requiere definir la dirección de avance del robot, en éste caso se calculan los vectores de dirección para cada uno de los centros geométricos del espacio de trabajo de cada una de las patas. Esto se realiza en el primer bloque del algoritmo. Posteriormente se determina la posición actual de cada una de las patas y se calculan los vectores de movimiento de cada pata, así como la movilidad del cuerpo del robot para la dirección establecida. En esta parte se realiza el cálculo de un vector de movimiento de patas MP y un vector de movimiento del cuerpo MC ecuaciones 27 y 28 respectivamente. Para el cálculo de estas ecuaciones se requiere de las formulas 23 y 24. Con base en la posición actual de las patas, se proyecta el centro de gravedad en las áreas geométricas que se muestran en la figura 11.

$$MP = (Vp1 \ Vp2 \ Vp3 \ Vp4 \ Vp5 \ Vp6) \quad (27)$$

$$MC = (Vc1 \ Vc2 \ Vc3 \ Vc4 \ Vc5 \ Vc6) \quad (28)$$

Como se observó en la tabla 1 existe un número determinado de movimientos de una pata, según el área donde se encuentre la proyección del centro de gravedad.

De acuerdo al diagrama de flujo desarrollado, el siguiente paso es evaluar el algoritmo de decisión, éste algoritmo se encarga en determinar la pata se moverá. Como dato de entrada de éste algoritmo, se requieren las longitudes de movimiento de cada una de las patas en la situación actual del robot, que es un vector de distancias Vd definido por la ecuación 29. Este vector se calcula de acuerdo con la dirección de avance del robot y la información de los vectores anteriores que se calcularon previamente ecuaciones 27 y 28.

$$Vd = (d1 \ d2 \ d3 \ d4 \ d5 \ d6) \quad (29)$$

Después de esta etapa se aplica un algoritmo difuso para definir la pata que será seleccionada a fin de continuar con el nuevo movimiento de locomoción libre del robot. El resultado de éste algoritmo puede ser: a) Mover una pata, b) mover el cuerpo del robot y c) Mover el cuerpo y una o más patas al mismo tiempo. En el caso en que el resultado del algoritmo sea el movimiento de una de las patas, se realiza una validación si las condiciones del terreno son buenas, es decir, si el espacio a donde llega la pata esta definido o esta libre para apoyar la pata, de ser correctas estas apreciaciones, se realiza el movimiento y se termina la tarea. En caso contrario, se realizará un reubicación de cada una de las patas para cambiar las condiciones y poder realizar el movimiento de alguna otra pata.

7 Algoritmo Difuso

Para el desarrollo de éste algoritmo de decisión, se tienen dos posibilidades de realizar, una de ellas es, utilizar un algoritmo con lógica difusa denominado por su término inglés *fuzzy* y la otra es utilizar un algoritmo con la combinación de lógica difusa conjuntamente con redes neuronales denominado por el término inglés *neurofuzzy*.

En la figura 13 se presenta el esquema de un sistema difuso, en éste esquema se puede observar que se tiene una primera etapa conocida como *fuzzification*, término utilizado para la conversión de las variables de entrada en variables difusas, el segundo bloque es el que se conoce como inferencia o reglas *fuzzy*, y el tercer bloque conocido como *defuzzification*. Es en éste último bloque donde se determinan los resultados que serán las salidas del sistema.

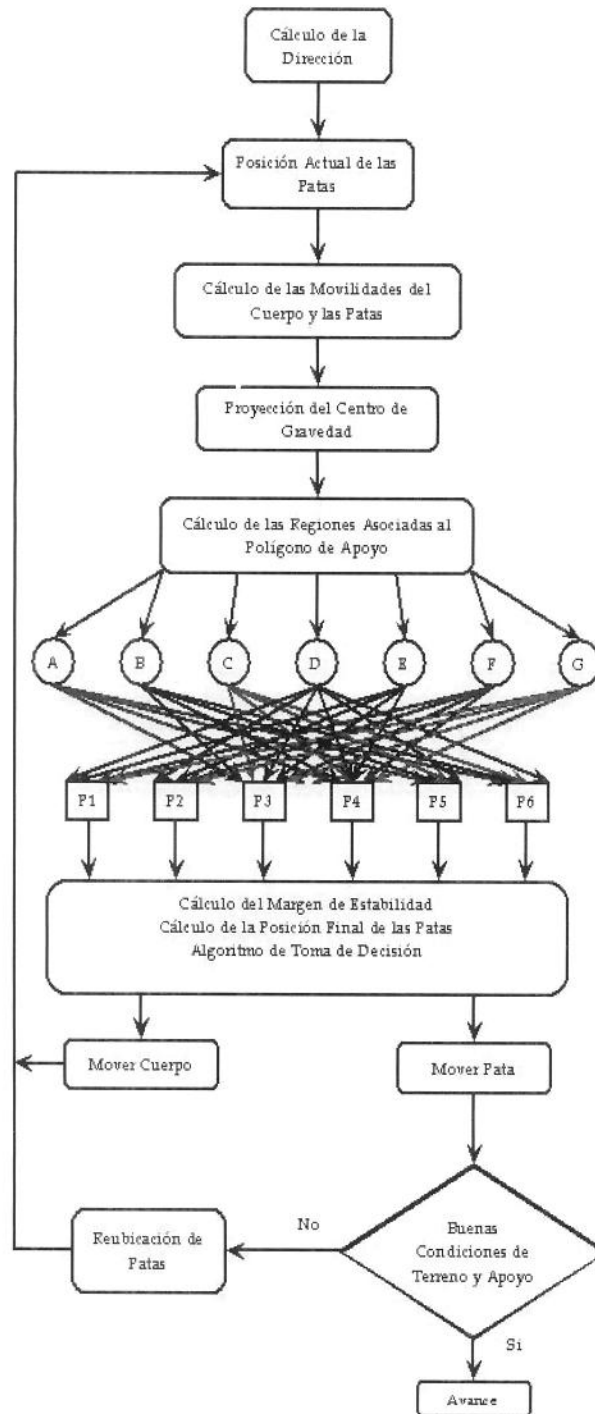


Fig. 12. Algoritmo de Locomoción Libre para una Pata

En la figura 13 a) se muestra un diagrama de flujo más detallado de los cálculos para determinar la solución de un sistema difuso. En la primera etapa hay un procesamiento de los datos de entrada, esto va desde una validación hasta una normalización de los mismos, el procesamiento anterior a esta etapa puede ayudar al desarrollo del algoritmo difuso, posteriormente los datos modificados son pasados a la siguiente etapa.

En la ecuación 30 se presenta la relación de un sistema difuso. Donde $O(t)$ es la salida del algoritmo, considerando que las salidas son función de tiempo puesto que éste se va a ejecutar constantemente durante el avance del robot y el movimiento de cada una de las patas. Los valores asignados en la función difusa D consiste en la composición de las reglas de inferencia, por lo general se usan reglas "si entonces", las cuales son determinados por la información pertinente al sistema. Los datos de entrada se representan como $I(t)$.

$$O(t) = \psi [D, I(t)] \tag{30}$$

Donde la función ψ representa el algoritmo difuso, que en éste caso consiste en un conjunto de reglas de inferencia, que son las que determinan la decisión según las condiciones de las variables de entrada.

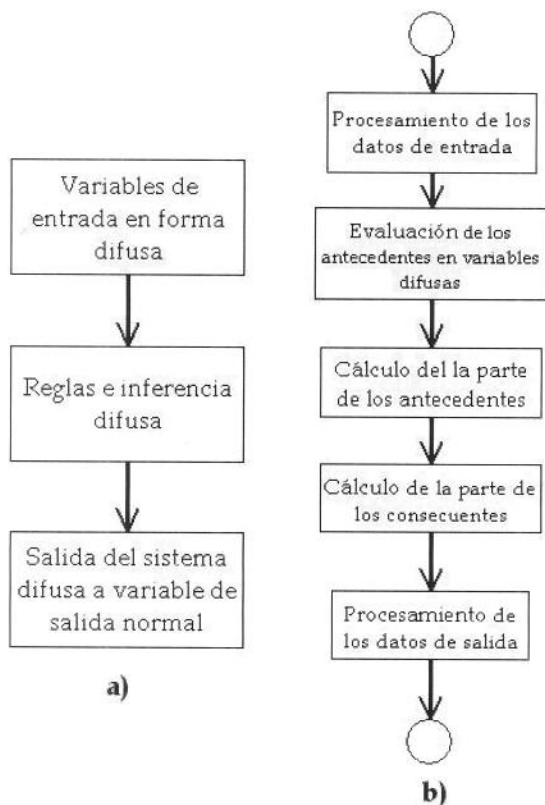


Fig. 13. Diagrama de Sistema de Decisión Difuso



Fig. 14. Diagramas de Funciones Difusas

En la figura 14 se muestra las funciones de membresía posibles para aplicar al algoritmo difuso de tres variables, la primera es cuando tenemos una movilidad baja de la pata, la segunda cuando tenemos una movilidad media y la tercera cuando tenemos la movilidad alta. En la ecuaciones 31, 32 y 33 (Berkan and Trubatch, 1996), se presenta la representación de dichas funciones, donde la función de membresía se denota como $\mu(d)$ y dn es la distancia calculada en el vector de movimiento, en éste caso el universo de D se considera la distancia máxima a la cual se puede mover la pata. Para esa situación actual éste valor puede variar de acuerdo a las condiciones de espacio de trabajo de cada una de las patas.

$$A1 = \{(dn, \mu_b(dn))\} \quad dn \in D \tag{31}$$

$$A2 = \{(dn, \mu_m(dn))\} \quad dn \in D \tag{32}$$

$$A3 = \{(dn, \mu_a(dn))\} \quad dn \in D \tag{33}$$

Por otro lado, es necesario considerar que entre mayor sea la partición del universo en discurso D en variables difusas, serán necesarias más reglas de inferencia para obtener un algoritmo de locomoción. En la ecuación 34 tenemos la relación de las reglas de inferencia en función de las variables de entrada

$$\eta r = \eta f^v \tag{34}$$

Así tenemos que en la ecuación 34 ηr que representa el número de reglas del sistema difuso, ηf^v es el número de particiones del universo en variables difusas y v es el número de variables que intervienen en el sistema. En el caso de considerar la partición de la figura 14 se necesitarán 729 reglas y si consideramos la estabilidad como una variable más, entonces se necesitarán 2187. Esta cantidad de reglas es muy grande sobre todo numéricamente y por consecuencia requerirá de un tiempo de cómputo muy grande haciendo muy lento el sistema, por lo que se requiere un trabajo de análisis para la reducción de reglas (Gorrostieta, 2000).

En éste trabajo se propone el avance de cada una de las patas hacia un punto definido. Este avance está definido por la intersección del vector de dirección ubicado en la pata y el espacio de trabajo de dicha pata.

En la elaboración de los algoritmos se consideraron como resultado dos funciones de membresía, una denominada baja y la otra función de membresía denominada alta, de acuerdo a la figura 15. De esta forma, se podrían considerar 64 reglas como máximo, esto sin considerar la variable de estabilidad. Una vez definido el número de variables en cada una de las patas se encuentra la relación matemática de cada una de las funciones de membresía difusa.



Fig. 15. Diagramas de Funciones Difusas del Algoritmo

De esta forma, cada movilidad de las patas puede pertenecer a los conjuntos difusos de movilidad baja y movilidad alta.

Es necesario considerar que las funciones de pertenencia μ , varían dependiendo de las características de cada una de las patas. En el estudio realizado, se agruparon en tres grupos para determinar la función de pertenencia, las patas P1 y P2 tienen el espacio de trabajo idéntico por las características físicas del robot. Así queda definida la función de pertenencia para la movilidad baja de éste par de patas como μ_{B1} y en la movilidad alta será μ_{A1} . De manera similar se determina para las patas centrales P3 y P4 y las P5 P6 de la parte trasera del robot. Este procedimiento se denomina con el término inglés como *fuzzification* y lo denotamos en el bloque de la figura 13 a) como variables de entrada en forma difusa.

Las reglas de inferencia difusa se expresan en función de las variables de entrada al sistema difuso y son las que juegan el papel más importante en la toma de decisión. Estas reglas se construyen con el conocimiento del experto como se conoce en el lenguaje de la lógica difusa o por métodos heurísticos. En el caso concreto del algoritmo de locomoción libre, para el movimiento de una sola pata las posibilidades de salida son seis. En la tabla 4 se observan las nomenclaturas asignadas a cada un de las salidas, de esta forma se procede a construir las reglas correspondientes a cada una de las áreas del polígono de apoyo, es decir, que dependiendo de la proyección del centro de gravedad se aplica un algoritmo difuso específico.

Tabla 4. Salidas de las Reglas de Inferencia

Salida	Pata
S1	P1
S2	P2
S3	P3
S4	P4
S5	P5
S6	P6

En la ecuación 35 se presenta una muestra de las reglas de inferencia difusa que se usan en estos algoritmos.

$$\begin{aligned}
 &si\ Pa1 \ominus Pb2 \ominus Pb3 \ominus Pb4 \ominus Pb5 \ominus Pb6 \rightarrow S1 \\
 &si\ Pb1 \ominus Pa2 \ominus Pb3 \ominus Pb4 \ominus Pb5 \ominus Pb6 \rightarrow S2 \\
 &si\ Pb1 \ominus Pb2 \ominus Pa3 \ominus Pb4 \ominus Pb5 \ominus Pb6 \rightarrow S3 \\
 &si\ Pb1 \ominus Pb2 \ominus Pb3 \ominus Pa4 \ominus Pb5 \ominus Pb6 \rightarrow S4 \\
 &si\ Pb1 \ominus Pb2 \ominus Pb3 \ominus Pb4 \ominus Pa5 \ominus Pb6 \rightarrow S5 \\
 &si\ Pb1 \ominus Pb2 \ominus Pb3 \ominus Pb4 \ominus Pb5 \ominus Pa6 \rightarrow S6
 \end{aligned}
 \tag{35}$$

El operador Θ que se usa queda definido por las distintas definiciones de operaciones difusas que puede ser desde un operador lógico conjunción o disyunción hasta un operador especial como el de Zade, Mandami ó Lukasiewiski. En éste estudio muy particular se utilizó la conjunción y la manera más sencilla de obtenerla es usando el mínimo de las funciones difusas de cada una de las reglas (Tanaka 1997).

En éste sistema, el punto crítico se presenta cuando existen varias salidas con la misma posibilidad de resultado para el algoritmo. En éste caso, se utiliza una modificación a la función de pertenencia, a la cual se conoce como corte alfa y el objetivo principal de estas modificaciones es dar un peso a la función de pertenencia según los datos de la entrada, de tal forma que se obtenga una sola salida de un sistema redundante como el que se presenta. Es decir, si tenemos como salida el movimiento de tres patas por ejemplo, entonces hay que decidir cuál de estas tres es la que se va a mover.

$$A^\alpha = \{dn \mid \mu(d) < \alpha\} \tag{36}$$

$$\mu_{A^\alpha}(d) = \vee [\alpha \bullet \mu_{A^\alpha}(d)] \quad d \in D \tag{37}$$

Es algo muy similar a lo que sucede cuando iniciamos una caminata y cualquiera de los dos pies tienen la misma posibilidad de dar éste primer paso. Esta situación se presenta en el algoritmo pero con la diferencia que se tienen seis posibilidades. Para esta condición, se introdujo una variable aleatoria, dicha variable se utiliza para dar un peso específico a la nueva función de pertenencia con el corte alfa. Dicho peso específico se denomina también como un cierto umbral de pertenencia, denotado por las ecuación 36 y 37 (Berkan and Trubatch, 1996).

Este es el paso correspondiente a la inferencia difusa de diagrama de flujo mostrado en la figura 13 b.

La última parte que muestra el diagrama se refiere a determinar la salida del algoritmo difuso. A éste proceso se le conoce comúnmente como *defuzzification* y es en éste proceso en donde se toman los resultados obtenidos de las diferentes reglas de inferencia y se determina una salida en las variables que se pueda interpretar en la aplicación, en éste caso lo que interesa como salida es determinar cuál de las seis patas se moverá con las condiciones de entrada presentadas.

Existen varios métodos utilizados en la última etapa de un sistema difuso, estos se pueden dividir en dos grupos, los métodos de centroides y los métodos máximos. En los métodos de centroides tenemos el centro de pesos, centro de gravedad, centro de masas y el centro del área más grande, de los métodos máximos se tienen el de máxima posibilidad y promedio de máximos. El método más usual por esta etapa es el conocido como centro de gravedad, en la ecuación 38 se puede ver la ecuación general para el cálculo de éste (Pedrycz, 1993).

$$S0 = \frac{\int \mu(d)dc \, dd}{\int \mu(d) \, dd} \tag{38}$$

Con la finalidad de simplificar e implementar una solución de la ecuación 38, se usa su aproximación en series denotada por la ecuación 39.

$$S0 = \frac{\sum_{i=1}^N dc_i \mu(d)}{\sum_{i=1}^N \mu(d)} \tag{39}$$

Para organizar la ejecución del algoritmo el diagrama de flujo que se muestra en la figura 13 b, el primer paso es un tratamiento de los datos que intervienen en el sistema después de esta etapa las variables de salidas se evalúan en las funciones de membresía correspondientes. Las reglas de inferencia son compuestas por dos partes, la parte de

los antecedentes y las partes de los consecuentes o también llamada la parte izquierda y derecha de la ecuación respectivamente. En la parte del antecedente se obtiene un vector con el resultado de las operaciones lógicas.

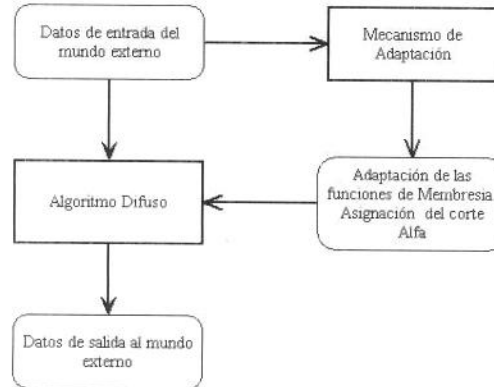


Fig. 16. Sistema Difuso Adaptivo

En el diagrama mostrado fue necesario realizar ajustes en las reglas de inferencias. Concretamente, en las funciones de pertenencia. Las características de los espacios de trabajo introducen una variación, ya que las condiciones de estos espacios cambian con respecto a la posición del robot en el eje coordenado z , razón fundamental para que la definición de las funciones de membresía cambiarán para saber qué tan grande o qué tan pequeña es la movilidad. Es importante mencionar que la movilidad pequeña se puede convertir en grande en caso de que se mantenga constante los valores de la función de membresía cuando el robot se aleje del suelo. Por esta razón se considera un ajuste en las funciones de membresía que se adapten a estos cambios. En la figura 16 se muestra el esquema del sistema de adaptación difuso. Como se expresó la ecuación general de un sistema difuso, ecuación 30, ahora el sistema difuso adaptable puede expresarse por la ecuación 40 (Berkan and Trubatch, 1996).

$$O(t) = \psi [D, D_a(t), I(t)] \quad (40)$$

La parte adaptable de la ecuación 40 se denota con la D_a , esta parte es la que realiza los cambios en el sistema adaptivo, otro de los cambios que se realizaron en esta etapa son los valores del corte alfa que también altera las funciones de membresía como se mencionó con anterioridad. En éste caso no es necesario rediseñar la regla de inferencia ya que los cambios que realizan de la adaptación sólo son redimensionamiento del universo y en otra es el valor de umbral alfa. Con esta última consideración los algoritmos de decisión quedan resueltos con algunas variantes que se van a ir explicando según sea el caso de cada uno de los diferentes algoritmos de locomoción plateados en la presente investigación.

8 Algoritmo de Locomoción Libre

En éste algoritmo se construye de forma no repetitiva, ni predeterminada, la decisión respecto a qué número de patas va a mover el robot, como se ilustran en la figura 18. Se evaluaron tres posibilidades: el movimiento, de una pata, de dos patas y tres patas. Estas posibilidades se presentaron en los algoritmos anteriores. En lo que respecta a las primeras cuatro acciones, la misma situación que en los algoritmos anteriores, es decir se calcula la dirección, la posición actual de las patas los vectores de movilidad para el cuerpo y las patas, posteriormente se proyecta el centro de gravedad.

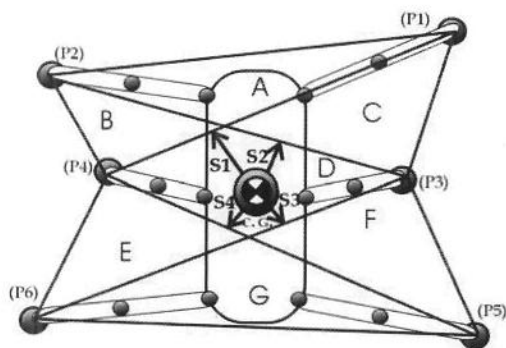


Fig. 17. Margen de Estabilidad Dentro de las Areas Determinadas

Para lograr una decisión, es indispensable el cálculo del margen de estabilidad. Este parámetro se utiliza para decidir cuántas patas se van a mover en el futuro desplazamiento del robot. Para calcular en éste punto el margen de estabilidad, se determinan las distancias del centro de gravedad proyectado a los límites del área que describirá posteriormente el polígono de apoyo como se puede observar en la figura 17. Esta variante del cálculo del margen de estabilidad permitirá tener información antes de llegar al algoritmo de decisión.

La salida de éste algoritmo permitirá mover entre una y tres patas, dependiendo de las condiciones de entrada y el cálculo de la estabilidad actual del robot. Por lo que se requiere introducir en el algoritmo una variable difusa perteneciente al margen de estabilidad. Como se muestra en la figura 19 se consideran tres funciones de membresía para el margen de estabilidad.

1.- Punto Destino

$$Pd = (xd, yd)$$

2.- Posición Actual de las Patas

$$C_1(x_{c1}, y_{c1}) \quad C_2(x_{c2}, y_{c2}) \quad C_3(x_{c3}, y_{c3})$$

$$C_4(x_{c4}, y_{c4}) \quad C_5(x_{c5}, y_{c5}) \quad C_6(x_{c6}, y_{c6})$$

3.- Cálculo de los límites del espacio de trabajo

$$I_1(x_{i1}, y_{i1}) \quad I_2(x_{i2}, y_{i2}) \quad I_3(x_{i3}, y_{i3})$$

$$I_4(x_{i4}, y_{i4}) \quad I_5(x_{i5}, y_{i5}) \quad I_6(x_{i6}, y_{i6})$$

4.- Cálculo de los vectores de movimientos de las patas y cuerpo

$$\vec{V}c_1 = (x_{i1} - x_{c1})\hat{i} + (y_{i1} - y_{c1})\hat{j} \quad \vec{V}p_1 = (x_{i1} - x_1)\hat{i} + (y_{i1} - y_1)\hat{j}$$

$$\vec{V}c_2 = (x_{i2} - x_{c2})\hat{i} + (y_{i2} - y_{c2})\hat{j} \quad \vec{V}p_2 = (x_{i2} - x_2)\hat{i} + (y_{i2} - y_2)\hat{j}$$

$$\vec{V}c_3 = (x_{i3} - x_{c3})\hat{i} + (y_{i3} - y_{c3})\hat{j} \quad \vec{V}p_3 = (x_{i3} - x_3)\hat{i} + (y_{i3} - y_3)\hat{j}$$

5.- Cálculo del desplazamiento de cada una de la patas

$$dez_1 = \delta_1 \sqrt{(x_{i1} - x_1)^2 + (y_{i1} - y_1)^2}$$

$$dez_2 = \delta_2 \sqrt{(x_{i2} - x_2)^2 + (y_{i2} - y_2)^2}$$

$$dez_3 = \delta_3 \sqrt{(x_{i3} - x_3)^2 + (y_{i3} - y_3)^2}$$

$$\delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{bmatrix} = \begin{bmatrix} e^{-\gamma_1} \\ e^{-\gamma_2} \\ e^{-\gamma_3} \\ e^{-\gamma_4} \\ e^{-\gamma_5} \\ e^{-\gamma_6} \end{bmatrix} \quad \gamma = [\gamma_1 \quad \gamma_2 \quad \gamma_3 \quad \gamma_4 \quad \gamma_5 \quad \gamma_6]$$

6.- Cálculo de las moviidades de las Patas y el Cuerpo

$$MP = (Vp1 \quad Vp2 \quad Vp3 \quad Vp4 \quad Vp5 \quad Vp6) \quad MC = (Vc1 \quad Vc2 \quad Vc3 \quad Vc4 \quad Vc5 \quad Vc6)$$

7.- Cálculo de las líneas rectas entre cada una de la patas para definir las áreas

A B C D E F G y calcular en donde esta la proyección del centro de gravedad y margen de estabilidad.

$$Sm = \min(S1, S2, S3, S4, S5, S6)$$

8.- Ajuste de la funciones de membresía

$$A^\alpha = \{dn \mid \mu(d) < \alpha\} \quad \mu_A(d) = \sqrt{[\alpha \cdot \mu_{Aa}(d)]} \quad d \in D$$

9. Aplicación las reglas de membresía a cada uno de la patas

$$P_a = \{(dn, \mu_A(dn))\} \quad dn \in D$$

$$P_b = \{(dn, \mu_B(dn))\} \quad dn \in D$$

10.- Cálculo de las salidas por el algoritmo difuso para el movimiento de una, dos o tres patas

$$SO = \frac{\sum_1^N S1d\mu_n(d) + \sum_1^N S2d\mu_n(d) + \sum_1^N S3d\mu_n + \sum_1^N S4d\mu_n + \sum_1^N S5d\mu_n + \sum_1^N S6d\mu_n}{\sum_1^N \mu_n(d)}$$

11.- Si no hay posibilidad de movimiento de las patas se mueve el cuerpo e ir al paso 2

12.- Cálculo de la posición final de las patas y

Determinar los ángulos de cada uno de ellas

$$\theta_1 = d\gamma - A\gamma(\cos \xi - 1)$$

$$\theta_2 = d\beta - A\beta(\cos \xi - 1)$$

$$\theta_3 = d\chi - A\chi(\cos \xi - 1)$$

13.- Realización de los movimientos a cargo del control neuronal

14.- Si no se ha llegado al punto final ir al paso dos, en caso contrario terminar.

Fig. 18. Algoritmo de Locomoción Libre



Fig.19. Funciones de Membresía Para el Margen de Estabilidad

Las relaciones difusas para el margen de estabilidad se muestran en las ecuaciones 41, 42 y 43. En donde Sm es el margen de estabilidad calculado y S es el universo de éste.

$$S_b = \{(Sm, \mu_B(Sm))\} \quad Sm \in S \quad (41)$$

$$S_m = \{(Sm, \mu_m(Sm))\} \quad Sm \in S \quad (42)$$

$$S_a = \{(Sm, \mu_A(Sm))\} \quad Sm \in S \quad (43)$$

Ahora las reglas de inferencia tendrán en consecuencia una variable más, en el caso de las variables de salida estas están comprendidas en las tablas 1,2 y 3.

La toma de decisiones puede verse como el proceso aplicado por una persona que tiene condiciones de equilibrio no muy favorables, en ese caso se realizan movimientos suaves y cortos, por lo que en el robot si las condiciones de estabilidad no son muy buenas, es decir que cae en una categoría baja de las funciones difusas, entonces éste sólo moverá una sola pata como se puede ver en la ecuación 35. Si las condiciones de estabilidad ahora son regulares o medianas, entonces el robot pasará a mover dos patas y en el último caso cuando el robot cuenta con condiciones de estabilidad muy buenas, éste moverá tres patas. Este algoritmo integra los tres algoritmos anteriores mediante la nueva variable difusa.

El algoritmo llamado con predictor se proyecta el movimiento de las patas en el siguiente instante futuro de desplazamiento del robot, en éste punto se calcula el valor de la estabilidad para cada una de las tres posibilidades de movimiento de patas. El valor más grande de margen de estabilidad resultante de cada una de las tres posibilidades es el que se toma para el movimiento de las patas. Este último algoritmo tiene como objetivo principal mejorar el desempeño del robot

Para probar la funcionalidad de estos dos últimos algoritmos, se propuso una trayectoria específica en uno de los simuladores desarrollados con esta finalidad.

El parámetro de comparación utilizado como indicador de desempeño del robot, es el margen de estabilidad. En la figura 20 se presenta el desempeño del margen de estabilidad en el algoritmo de locomoción libre y el algoritmo de locomoción que utiliza un predictor. Estos algoritmos son ejecutados simultáneamente para una misma trayectoria y condiciones de terreno.

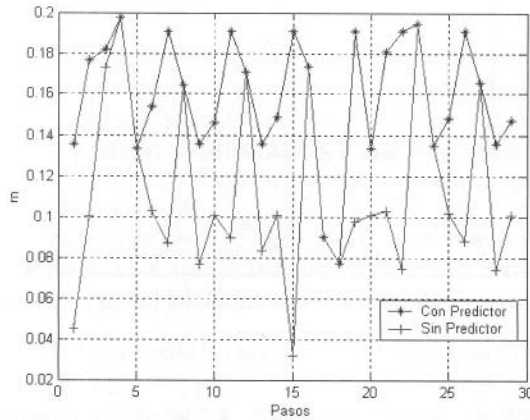


Fig. 20. Comportamiento del Margen de Estabilidad

En la gráfica correspondiente al margen de estabilidad del algoritmo predictivo se puede observar que éste trata de buscar los valores más altos para éste parámetro, lo que permitirá tener una mejor estabilidad en el robot al caminar.

9 Simuladores

Como una parte del trabajo desarrollado, se diseñaron dos programas en computadora que nos permitieron verificar la movilidad del robot y su estabilidad.

El simulador en su primera versión fue realizado en lenguaje C. Las ecuaciones de movimiento programadas nos permitieron leer una base de datos que contiene las posiciones relativas de las patas del robot con respecto al centro de gravedad del cuerpo. De esta forma, las posiciones fueron generadas a través de un programa en donde se evaluaron los algoritmos de locomoción libre presentados en éste trabajo. El simulador permite visualizar las configuraciones del robot bajo diferentes puntos de vista. Esta visualización nos permitió ubicar la geometría del robot al caminar y su estabilidad.

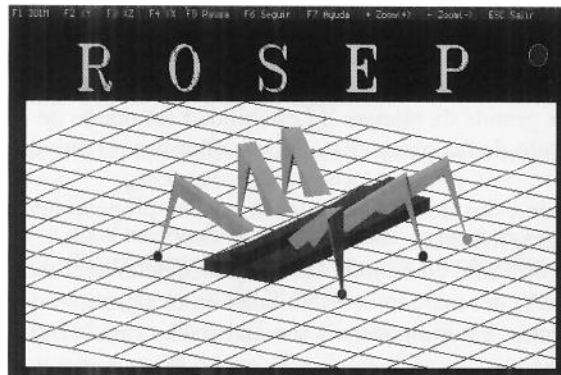


Fig. 21. Simulador en Tridimensional del Robot

En la Figura 21 se muestra una de las vistas que presenta el programa de visualización implementado en lenguaje C. La rapidez de su ejecución nos hace suponer que es posible construir en un mismo programa los algoritmos y el simulador.

Con el propósito de integrar los algoritmos de locomoción libre, se diseñó un segundo simulador. Dicho simulador incluye a su vez las librerías de OpenGL para C. En la Figura 22 se muestra una de las vistas del simulador, en donde se representa al robot caminando de frente. Este simulador presenta las dimensiones de las patas proporcionales a las dimensiones reales de una hormiga.

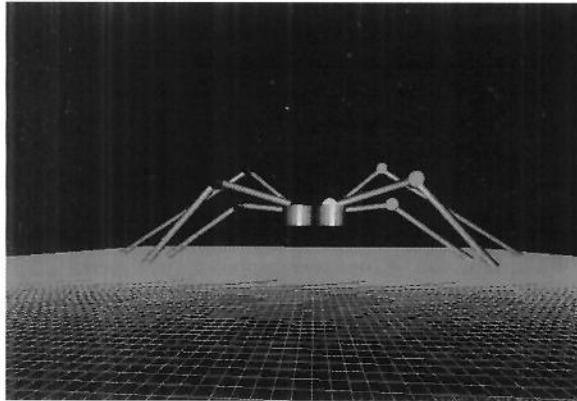


Fig. 22. Simulador Tridimensional del Robot

10 Conclusiones

Los algoritmos de locomoción libre mostrados en éste trabajo permitieron evaluar la movilidad de las patas, y del cuerpo del robot de forma que no existe un plan previo de secuencias en la forma de caminar del robot. Así mismo, se presenta una contribución en la parametrización del control de éste tipo de algoritmos de locomoción libre.

El desempeño de la movilidad y estabilidad se analizó únicamente para terreno plano. El uso de la adaptación de las funciones de membresía está enfocado a la locomoción en diferentes tipos de terrenos irregulares los cuales se proponen como trabajos de investigación a futuro.

El objetivo planteado en la investigación se alcanzó con el desarrollo de estos algoritmos. El sistema de adaptación de los algoritmos permite resolver la redundancia de la movilidad de las patas del robot.

Se cuenta con un sistema de control apropiado para el desarrollo de los movimientos de cada una de las patas del robot, así como sistema de auto-ajuste que puede realizarse en tiempo real.

El algoritmo difuso que se utilizó en la construcción de los algoritmos, resultó ser adecuado para lograr controlar en simulación el movimiento del robot. Es importante recordar que esta técnica nace con la imitación del razonamiento de realizar una cierta tarea, lo que resulto muy propio para éste tipo de aplicación.

Finalmente, se concluye que el algoritmo que presentó mejores resultados, tanto de movilidad como de estabilidad; fue el algoritmo predictivo.

Referencias

1. **Arkin, R.** "Behavior-Based Robotics". The MIT Press Cambridge. Massachusetts London England. 1998.
2. **Berkan, C., y Trubatch, S.** "Fuzzy Systems Design Principles". IEEE Press. 1996.
3. **Celaya, E., y Porta, M.** "A Control Structure for the Locomotion of a Legged Robot on Difficult Terrain". IEEE Robotics and Automation Magazine. pp. 43-51, 1998.

4. **Estremera, J., y González-de-Santos, P.** "Free Gait For Cuadruped Robots Over Irregular Terrain". International Robotics Research. Volume 21, Number 2. Sage Science Press. pp. 115-130,2002.
5. **Gorrostieta, E., y Vargas, E.** "Diseñando un Robot Caminante de Seis Patas ". Segundo Congreso Nacional de Robótica. Asociación Mexicana de Robótica e Instituto Tecnológico de Toluca. pp. 181-186, 2002.
6. **Gorrostieta, E., y Vargas, E.** "Designig a PD control with Gravity Compensation for a six Legged Robot ". 3rd International Symposium on Robotics and Automation, ISRA – IEEE. pp. 70-74, 2002.
7. **Gorrostieta, E., Aguado, A., y Vargas, E.** "Diseño de un Controlador Neuronal Aplicado a la Generación de Pasos de un Robot Caminante". Ubamari . Año XIX, No. 53. pp. 32-48, 2002.
8. **Gorrostieta, E., Vargas, E., y Aguado A.** "A Neuro PD Control Applied for Free Gain on a Six Legged Robot". WSEAS Transactions on Computer. Issue 3, Volumen 3. pp. 612-619, 2004.
9. **Gorrostieta, E.** "Modelado Neuro Difuso de un Quemador de Biomosas". Tesis de Maestría. Centro de Ingeniería y Desarrollo Industrial. Querétaro, México. 2000
10. **Masuhiro, F.** "AIBO : Toward The Era Digital Creatures". International Robotics Research. Volume 20, Number 10. Sage Science Press. pp. 781-794, 2001.
11. **Pedrycz, W.** "Fuzzy Control and Fuzzy Systems". Research Studies Press and John Wiley and Son Inc. London England. 1993
12. **Shaoping, B., Low, H., y Weimiao, G.** "Kinematographic Experiments on The Leg Movements and Body Trayectories of Cockroach Walking on Different Terrain". Proceedings of International Conferences on Robotics and Automation. pp. 2605-2610, 2001.
13. **Solano, J., Vargas, E., Gorrostieta, E., y Morales, C.** "Designing a Walking Robot of Six Legs ". Proceedings of International Symposium on Robotics and Automation ISRA'2000, Monterrey, N. L. pp. 115-119, 2000.
14. **Ilg, W., Mühlfriedel, T., Berns, K. y Dillmann, R.** "Hybrid Learning Concepts for a Biologically Inspired Control of Periodic Movements for Walking Machines". In. Soft Computing in Mechatronics. Germany. pp. 19-40, 1999.
15. **Tanaka, K.** "An Introduction to Fuzzy Logic for Practical Applications ". Springer USA. 1997.
16. **Hironori, A., Noriho, K., Tatsuya, N., y Eiji, N.** "Gait Analysis of a Cuadruped Walking Robot". Proceedings of ISART'91. pp. 397-402, 1991.
17. **Vargas, E.** "Diseño y Realizaron de Algoritmos de Locomoción Libre para un Robots Caminantes de Cuatro Patas ". Tesis Doctoral. Universidad Complutense de Madrid. Facultad de Ciencias Físicas. 1994.
18. **Yoshihiro, S., Kazutoshi, K., Yuuji, H., Makoto, H., y Masakatsu, F.** "Foot Trajectory for Quedruped Walking Machine". IEEE International Workshop on Intelligent Robots and Systems. IROS90. pp. 315-322, 1990.
19. **Calle, E., Ávila, I., y Zambrano, J.** " Diseño e Implementación de un Robot Móvil Cuadrúpedo". Revista Tecnológica. ESPOL. Vol. 20, N. 1. pp. 65-72, ISSN : 0257-1749. 2007



Efrén Gorrostieta Hurtado es Doctor en Ingeniería, en la especialidad en Mecatrónica. Maestro en Ciencia con la especialidad en Control y Automatización e Ingeniero en Electrónica del Instituto Tecnológico de Estudios Superiores de Occidente. Actualmente Profesor investigador de la Universidad Autónoma de Querétaro en la Facultad de Informática. Las áreas de interés son Control Inteligente, Inteligencia Artificial, Robótica y Bioingeniería. Fundador del Concurso Nacional de Minirobótica, Presidente de la Academia Mexicana de Robótica 2003-2005. Senior Member IEEE., Fundador y actual presidente de la Asociación Mexicana de Mecatrónica A.C..



José Emilio Vargas Soto es Doctor en Ciencias Físicas, en el área de Informática y Automática por la Universidad Complutense de Madrid. The Electrocommunication University of Tokyo le otorgó un Postdoctoral Degree por sus investigaciones sobre la locomoción libre de robots caminantes. Es Maestro en Ingeniería por la Universidad Politécnica de Madrid en el área de Control, e Ingeniero Mecánico con mención honorífica por la Universidad Nacional Autónoma de México. Actualmente se desempeña como Director de la Facultad de Ingeniería en la Universidad Anáhuac México Sur, y combina esta actividad con investigación en el área de mecatrónica industrial y robots móviles.