

**REVISIÓN DEL ANÁLISIS MATEMÁTICO DEL MOVIMIENTO CABECEO–
CABECEO CABECEO–VIRAJE PARA UN SISTEMA ROBÓTICO MODULAR.****MATHEMATICAL REVIEW OF THE ANALYSIS OF MOVEMENT NOD-NOD
NOD-TURN FOR A MODULAR ROBOTIC SYSTEM.**

**Ing. Paola Natalia Lancheros Guzmán, Ing. Laura Beatriz Sanabria Galvis,
PhD. Ricardo Andrés Castillo Estepa**

Universidad Militar Nueva Granada, Programa de Ingeniería en Mecatrónica,
Grupo de Investigación DaVinci.
Carrera 11 #101-80, Bogotá, Cundinamarca, Colombia.
+57 1 6500000.

E-mail: {u1801917, u1801943, ricardo.castillo}@unimilitar.edu.co.

Resumen: El siguiente informe presenta el modelo matemático para la locomoción de un sistema robótico modular basado en el enfoque bio-inspirado, tomando como referencia el estado del arte relativo al generador central de patrones (CPG) simplificado y representado mediante osciladores sinusoidales de frecuencia fija. Esto, con el fin de generar movimiento en configuraciones tipo oruga y serpiente del sistema robótico modular MECABOT 3.0 por medio de la simulación realizada en el software Webots y de esta manera verificar la aplicabilidad del modelo matemático.

Palabras clave: CPG, Locomoción, Osciladores De Frecuencia Fija, Series De Fourier, Sistema Robótico Modular.

Abstract: In the following report the mathematical model for locomotion of a modular robotic system is presented according to the bio-inspired approach, regarding the state of the art relative to the central pattern generator (CPG) simplified and represented by sinusoidal fixed frequency oscillators. In caterpillar and snake type configurations, to implement it in the simulation of modular robotic system MECABOT Webots 3.0 through software and thus verify the applicability of the mathematical model.

Keywords: CPG, Fixed Frequency Oscillators, Fourier series, Locomotion, Modular Robotic System.

1 INTRODUCCIÓN

La robótica modular hace referencia a un grupo de sistemas robóticos compuestos por módulos pequeños, fácilmente interconectados. Éstos pueden ser de dos tipos: configurables, que cuentan con interfaces de conexión y auto-configurables, que son aquellos que basan su decisión en reglas geométricas para su accionamiento con módulos vecinos detectados, dando paso a la reconstrucción requerida (Rus, Salemi, Shen, & Yim, 2007). Un sistema robótico modular presenta ventajas

económicas y funcionales comparadas con las de un robot de estructura fija (Hoyos Gutiérrez, Cardona, Capacho, Prieto, & Pérez Cisneros, 2011), ya que puede ser reconfigurado morfológicamente según las condiciones en las que se encuentre por medio de módulos adicionales que se ajusten al nuevo arreglo morfológico dando la posibilidad de que se reduzcan los costos de inversión (Althoff, Giusti, & Icer, 2016).

El objetivo de este documento es obtener un modelo matemático que permita la locomoción en configuraciones tipo oruga y serpiente del sistema robótico modular MECABOT 3.0 existente en la Universidad Militar Nueva Granada.

El documento presentado a continuación se dividirá en las siguientes secciones: 1. INTRODUCCIÓN 2. REFERENTES TEÓRICOS 3. APLICACIÓN DEL MODELO MATEMÁTICO EN SISTEMA ROBÓTICO MODULAR MECABOT 3.0 4. ANÁLISIS DE RESULTADOS 5. CONCLUSIONES.

2 REFERENTES TEÓRICOS

La robótica modular tiene como objetivo el ensamble de módulos simples en una gran variedad de configuraciones para adaptarse a diferentes tareas, su diseño va enfocado a la generalidad y optimización en un rango de aplicaciones. La locomoción depende de la estructura y configuración elegidas de tal modo que cumpla con las restricciones del medio en el que está inmerso (obstáculos, estabilidad, función, entre otras).

Los robots modulares pueden ser clasificados según su arquitectura por la disposición geométrica de cada unidad, así:

- Arquitectura móvil: Están compuestos de unidades que emplean su entorno para ejecutar maniobras de desplazamiento y conformar cadenas o *lattice* (mallas) complejas.

- Arquitectura *lattice* (malla): Los módulos están conectados por un patrón tridimensional (malla). El control y movimiento se pueden realizar en paralelo y la reconfiguración es sencilla ya que se tiene un número limitado de ubicaciones para el desplazamiento de los módulos. La simulación computacional se puede escalar con facilidad para representar sistemas más complejos.

- Arquitectura en cadena: La interconexión es en cadena o árbol, es aquella en la que los módulos están conectados en serie, es muy versátil ya que puede alcanzar casi cualquier punto en el espacio, el movimiento y control es secuencial; sin embargo, son más difíciles de analizar, simular y controlar. Dentro de esta categoría se encuentra topología 1D, 2D y 3D, los cuales pueden ser apreciados en la Fig. 1 (Gonzales Gómez, 2008).

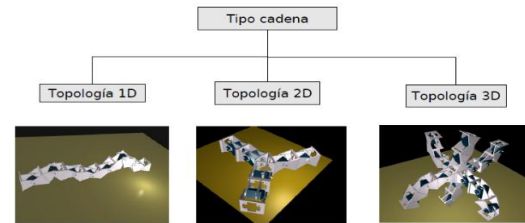


Fig. 1 Topologías para robot tipo cadena (Gonzales Gómez, 2008).

Dentro de la topología unidimensional se encuentran dos clases: Los ápodos y los ápodos autopropulsados. El término ápodo hace referencia al desplazamiento terrestre por medio de movimientos corporales, imitando el movimiento de los gusanos y serpientes que tienen una estructura segmentada que permite movimientos ondulatorios y cambiar su forma fácilmente. Estos se clasifican según su configuración en: viraje-viraje, cabeceo-cabeceo, cabeceo-viraje. Este tipo de configuración para los robots ápodos se ilustran en la Fig. 2.

- Cabeceo-Cabeceo (Oruga): Los módulos están conectados de tal forma que todos giran en el plano xz , generando la configuración tipo oruga. Su desplazamiento es hacia adelante o atrás.

- Viraje-Viraje: Los módulos están conectados de tal forma que todos giran en el plano xy su desplazamiento se da en forma lateral, perpendicular a la longitud de la cadena. Esta conexión genera el movimiento apoyando su cuerpo en el suelo completamente; emplea ruedas pasivas para reducir la fricción.

-Cabeceo-Viraje (Serpiente): Esta conexión resulta de alternar el mismo número de módulos de cabeceo y viraje, generando locomoción de dos dimensiones y generando una curva en tres dimensiones. Genera diferentes tipos de movimiento como el desplazamiento lateral, trepar, rodar, entre otras.

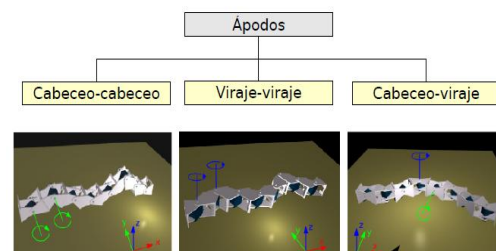


Fig. 2 División de robots ápodos (Gonzales Gómez, 2008)

Este trabajo se limitará al estudio de un sistema robótico modular con configuración cabeceo-cabeceo y cabeceo-viraje.

Para lograr que un sistema robótico modular tipo cadena se desplace únicamente por el giro en sus

articulaciones, se deben calcular las funciones y parámetros que determinan los ángulos necesarios para que el robot coordine los M módulos. Existen diversos enfoques que permiten la obtención de estas funciones tales como el clásico, tablas de control de la marcha (Gait Control Tables), y bio-inspirado. El enfoque bio-inspirado, basado en redes (Medina Durán, Ferreira Jaimes, & Gualdrón Guerrero, 2009) (Gualdrón Guerrero & Gamarra Sierra, 2009) o circuitos neuronales de animales vertebrados e invertebrados denominados CPG (Central Pattern Generators) que se encargan de producir estímulos rítmicos para acciones como la respiración y la marcha, sin necesidad de recibir estímulos externos (Bucher & Marder, 2001); este enfoque busca obtener un modelo matemático que mediante una aproximación al CPG, permita el control de la locomoción de los sistemas robóticos. Los modelos de control basados en CPG se han empleado un gran número de aplicaciones como cuadrúpedos (Fukuoka, Hada, Kimura, & Takase, 2003), hexápodos (Arai, Inagaki, & Yuasa, 2003), nadadores (Ijspeert & Crespi, 2008), terrestres (Dario, Menciassi, Sfakiotakis, la Spin, & Tsakiris, 2005), entre otros ya que presenta como ventaja que actúa directamente en las articulaciones, lo que produce un movimiento más armonioso y eficiente (Crespi & Ijspeert, 2007). Es por esto, que se emplea un modelo matemático de control basado en el CPG como controlador para los módulos del sistema robótico a trabajar.

Para lograr la locomoción de un sistema robótico modular en cadena, se requiere de un modelo matemático que permita la locomoción en configuraciones tipo oruga y serpiente deseadas, para esto se emplea el modelo matemático bio-inspirado presentado por Gómez; este modelo plantea que “En estado estacionario, (las redes CPG) se comportan como osciladores de frecuencia fija lo que permite sustituirlas por un modelo simplificado formado por generadores sinusoidales. La ventaja es que son sencillos de implementar y se requieren muy pocos recursos para su realización.” (Gonzales Gómez, 2008).

González describe matemáticamente los osciladores de frecuencia fija gracias al desarrollo por series de Fourier (1) las variables se presentan en la Tabla 1.

$$\varphi_i(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2\pi n}{T}t\right) + b_n \sin\left(\frac{2\pi n}{T}t\right) \right) \quad (1)$$

Tabla 1: Variables serie de Fourier

Símbolo	Variable
a_n, b_n	Coefficientes de Fourier
T	Periodo

t Tiempo

Se realiza una aproximación de la serie con el fin de limitar el problema a las funciones sinusoidales de frecuencia fija y se realiza el cambio de variables obteniendo la ecuación (2).

$$\varphi_i(\phi) = A_i \text{Sen}(\phi + \psi_i) + O_i, i \in \{1 \dots M\} \quad (2)$$

Las variables de la ecuación (2) se presentan en la Tabla 2.

Tabla 2: Variables onda sinusoidal

Símbolo	Variable	Símbolo	Variable
i	Generador	O_i	Offset
A_i	Amplitud	T	Periodo
ψ_i	Fase		

Se tiene en cuenta que la cadena está conformada por M generadores sinusoidales con amplitud A y diferencia entre fases $\Delta\phi$ de un par de módulos consecutivos, la ecuación que describe esta onda es (3)

$$\varphi_i(\phi) = A \sin(\phi + (i-1)\Delta\phi + \psi_i), i \in \{1 \dots M\} \quad (3)$$

La anterior ecuación se obtiene tomando en cuenta las siguientes restricciones:

- Para lograr simetría de las oscilaciones frente al origen, todos los generadores tienen un offset $O_i = 0$ y la misma amplitud A .
- Las fases de todos los módulos se expresan en función de la fase del primer módulo ψ_1 . Al tener limitado el estudio a un sistema en estado estacionario, el valor de la fase inicial se omite.
- Para obtener un movimiento armónico en el conjunto de módulos, la diferencia de fase $\Delta\phi$ es constante.

2.1 Curva Serpentinoide 2D

El sistema robótico modular trabajado se desplaza en una dimensión, por tanto, se modela con una curva continua s , que indica la distancia entre los extremos del robot. Estas curvas vienen descritas por el ángulo de doblaje $\theta(s)$ cuando es continuo, y φ_i cuando es discreto.

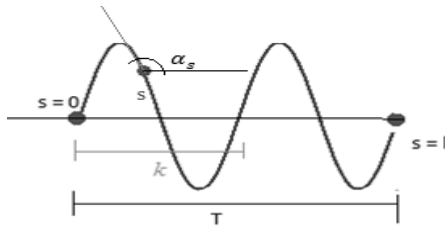


Fig. 3 Parámetros de la curva serpentinoide

La curva serpentinoide (Fig. 3) describe una onda sinusoidal (Tabla 3) que varía a lo largo del eje corporal de un sistema robótico modular tipo oruga o serpiente y se representa por la ecuación

(4).

$$\theta(s) = A \text{Sen} \left(\frac{2\pi k}{l} s \right) \quad (4)$$

Tabla 3: Parámetros Curva Serpentinoide

Símbolo	Variable	Símbolo	Variable
$\theta(s)$	Ángulo de doblaje	l	Longitud
T	Periodo	ds	Distancia
k	Ondulación		

El ángulo que se forma desde una línea tangente al punto s con respecto al eje x se representa por α_s donde, según el estudio sobre los robots biológicamente inspirados realizado por Hirose [10], la ecuación de α_s es (5) descrita en la Tabla 4.

$$\alpha_s = \alpha \text{Cos} \left(\frac{2\pi k}{l} s \right), 0 < \alpha_s < 120^\circ \quad (5)$$

Tabla 4: Parámetros de Onda Serpentinoide

Símbolo	Variable	Rango
α	Ángulo de serpenteo	$\alpha \in [0^\circ, 120^\circ]$
k	Número de ondulaciones	$k \geq 2$
ϕ	Fase	
h	Altura	
w	Anchura	

Por otro lado α es el ángulo de serpenteo que se forma entre la línea tangencial en el punto inicial de la curva ($s=0$) con respecto al eje x . El valor máximo del ángulo de serpenteo es 120° ya que, de superarse, el robot presentaría colisión entre los módulos [3]. En la Fig. 4 se observa la variación del ángulo de doblaje en la curva serpentinoide.

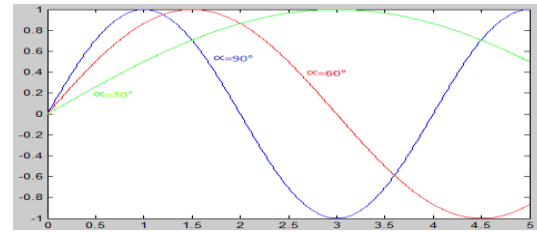


Fig. 4 Curva serpentinoide para diferentes ángulos de doblaje

Es posible generar un sistema coordenado, que permita definir la forma del robot, al igual que la región en la que es estáticamente estable, es decir, en la que existe movimiento.

2.2 Curva Serpentinoide 3D

Esta curva se forma por la superposición de dos ondas serpentinoides; la de cabeceo y la de viraje, cada una de estas tiene su propio ángulo de serpenteo y número de ondulaciones (Fig. 5). La curva serpentinoide 3D está descrita según las ecuaciones (8) y (9):

$$\theta_h(s, \phi) = 2\alpha_h \sin \left(\frac{2\pi k_h}{M} \right) \sin \left(\phi + \frac{2\pi k_h}{l} s + \Delta\phi_{vh} \right) \quad (6)$$

$$\theta_v(s, \phi) = 2\alpha_v \sin \left(\frac{2\pi k_v}{M} \right) \sin \left(\phi + \frac{2\pi k_v}{l} s \right) \quad (7)$$

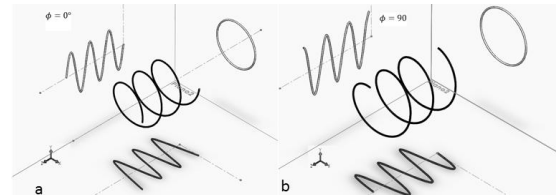


Fig. 5 Ángulos de serpenteo $k=3$ a. 0° b. 90°

2.3 Generadores Sinusoidales para el Grupo Cabeceo-Cabeceo

El grupo Cabeceo-Cabeceo requiere para su locomoción un único generador vertical, dado por la forma de la onda serpentinoide, a continuación, se presenta la ecuación que define este movimiento en tiempo continuo y discreto (Rodríguez O et al., 2012).

2.3.1 Sistema en tiempo continuo:

Para determinar correctamente una onda serpentinoide se deben tener en cuenta la ecuación (3) donde los parámetros se describen en la Tabla 5Tabla 4, que definen los valores en los que el

movimiento es estáticamente estable, el desplazamiento es nulo o el movimiento es muy brusco. Dependiendo del sentido de propagación de la onda, se observa si el robot avanza (cola - cabeza) o si retrocede (cabeza - cola). Otro aspecto que describe el comportamiento del sistema robótico modular es el parámetro k , encargado de la estabilidad de la locomoción del sistema robótico modular. Si el robot es estable en todas las fases ϕ , entonces la locomoción es estáticamente estable, esto ocurre cuando existen al menos dos puntos de contacto con la superficie $k \geq 2$ (Fig. 6). Sin embargo, es posible que en puntos intermedios se presente $k = 1$, por tanto, se caerá en la región inestable, provocando movimientos bruscos en el robot. Esta transición se puede suavizar variando el ángulo α .

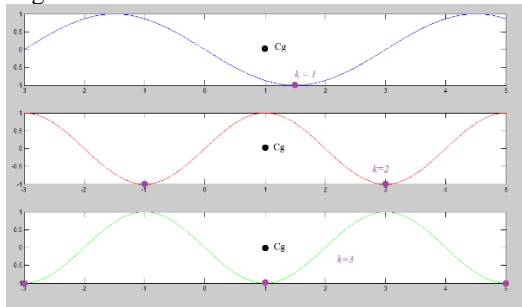


Fig. 6 Puntos de contacto en onda serpentina

2.3.2 Sistema en tiempo discreto:

Los robots modulares reales son sistemas discretos constituidos por múltiples módulos M que, al unirse conforman el sistema robótico y permiten su movimiento. Para el sistema descrito en este documento, se parte del desarrollo matemático en tiempo continuo, teniendo en consideración la distancia d que separa los diferentes módulos. Para determinar la onda generadora del movimiento del sistema se emplea un nuevo parámetro M_u (8).

$$M_u = \frac{M}{k} \quad (8)$$

Mientras mayor sea el parámetro M_u , la curva formada por los módulos será más cercana a la curva serpentina del modelo continuo. Así, la locomoción del robot discreto queda definida bajo la ecuación (9) y determinada por los parámetros de la Tabla 5.

$$\varphi_i(\phi) = 2\alpha \sin\left(\frac{\pi k}{M}\right) * \sin\left(\phi + \frac{2\pi k}{M} \left((i-1) + \frac{d_{0v}}{d}\right)\right) \quad (9)$$

Tabla 5: Parámetros de Locomoción del Robot en Discreto

Símbolo	Variable
$\Delta\phi$	Variación de fase [°]
M	Número de módulos
M_u	Número de módulos por ondulación
d	Longitud del sistema
d_0	Distancia al origen del sistema

2.4 Generadores Sinusoidales para el Grupo Cabeceo-Viraje

La locomoción en 2D de los sistemas robóticos modulares en cadena, presentan 5 modos de caminar diferentes en configuración tipo cadena: movimiento en línea recta, trayectoria circular, rodar, desplazamiento lateral y rotación (Gonzales Gómez, 2008) cuando tienen M articulaciones, estos movimientos se caracterizan según los grados de libertad del movimiento definidos según los parámetros característicos del generador de ondas sinusoidales ($\alpha_v, \alpha_h, k_v, k_h, \Phi_{vh}$), donde (Tabla 6):

Tabla 6: Parámetros de onda sinusoidal en 3D

Símbolo	Variable	Unidades
α_v	Ángulo de serpiente vertical	rad
α_h	Ángulo de serpiente horizontal	rad
k_v	Número de ondulaciones verticales	s^{-1}
k_h	Número de ondulaciones horizontales	s^{-1}
Φ_{vh}	Diferencia de fase entre módulo vertical y horizontal.	rad

Este trabajo se limita al desplazamiento en dos dimensiones en configuración tipo serpiente, es decir, desplazamiento lateral considerando una superficie homogénea y sin obstáculos.

2.4.1 Sistema en tiempo continuo

La configuración tipo serpiente permite que el robot se desplace hacia los lados sin variar la orientación de su eje corporal mediante dos grupos de generadores independientes: horizontal o de viraje φ_{hi} (10) y vertical o cabeceo φ_{vi} (11), que al superponerse, forman ondas tridimensionales que varían de acuerdo a la fase entre ellas Φ_{vh} (Tabla 7).

$$\varphi_{vi}(\phi) = A_v \sin(\phi + (i-1)\Delta\phi_v), i \in \left\{1.. \frac{M}{2}\right\} \quad (10)$$

$$\varphi_{hi}(\phi) = A_h \sin(\phi + (i-1)\Delta\phi_h + \Delta\phi_{vh}), i \in \left\{1.. \frac{M}{2}\right\}$$

(11)

Tabla 7: Parámetros de la onda sinusoidal 3D en tiempo continuo

Símbolo	Variable	Símbolo	Variable
A_h	Amplitud horizontal	$\Delta\phi_h$	Diferencia de fase entre dos módulos horizontales consecutivos.
A_v	Amplitud vertical	$\Delta\phi_v$	Diferencia de fase entre dos módulos verticales consecutivos.
φ_{hi}	Angulo de doblaje horizontal	$\Delta\phi_{vh}$	Diferencia de fase entre módulo vertical y horizontal.
φ_{vi}	Angulo de doblaje vertical		

Para esta configuración se debe seguir el criterio de estabilidad presentado en la sección (2.3.1, que indica que deben existir al menos dos ondulaciones en el sistema robótico modular.

2.4.2 Sistema en tiempo discreto

Para poder implementar la curva serpentina 3D en el sistema robótico modular, es necesario discretizar teniendo en cuenta que está conformado por articulaciones de tipo cabeceo situadas en (12) y viraje en (13) alternadamente.

$$s = (i - 1)d + d_0 \quad (12)$$

$$s = (i - 1)d + d_0 + \frac{d}{2} \quad (13)$$

Los ángulos de doblaje se obtienen teniendo en consideración la ubicación de los puntos s en las articulaciones verticales y horizontales (14)(15):

$$\varphi_{vi}(\phi) = 2 \alpha_v \sin\left(\frac{2\pi k_v}{M}\right) \sin\left(\phi + \frac{4\pi k_v}{M}\left(i - 1 + \frac{d_0}{d}\right)\right) \quad (14)$$

$$\varphi_{hi}(\phi) = 2 \alpha_h \sin\left(\frac{2\pi k_h}{M}\right) \sin\left(\phi + \frac{4\pi k_h}{M}\left(i - 1 + \frac{d_0}{d} + \frac{1}{2}\right) + \Delta\phi_{vh}\right) \quad (15)$$

3 APLICACIÓN DEL MODELO MATEMÁTICO EN SISTEMA ROBÓTICO MODULAR MECABOT 3.0

3.1 MECABOT 3.0

MECABOT 3.0 Fig. 1 es la tercera generación del sistema robótico modular MECABOT desarrollado en la Universidad Militar Nueva Granada con el fin de ser empleado en operaciones de búsqueda y rescate urbano. En base a este diseño se realizarán las pruebas de simulación para las configuraciones tipo oruga y serpiente con el fin de verificar su funcionalidad como robot móvil. MECABOT 3.0 cuenta con un cuerpo (segmento) y una articulación (pivote).

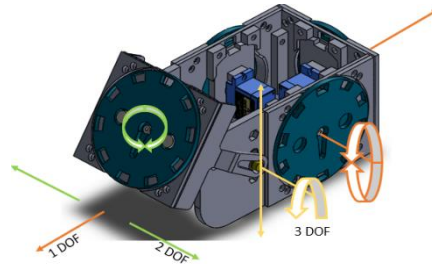


Fig. 7 MECABOT 3.0.




3.2 Pruebas de Simulación de MECABOT 3.0

Para constatar la aplicabilidad del modelo matemático presentado en este documento se realiza la simulación de un sistema robótico modular. El software utilizado es Webots, un simulador que proporciona un ambiente de prototipado rápido para crear mundos virtuales en 3D con propiedades físicas. El usuario puede agregar objetos pasivos o activos dentro del nodo mundo (world); los objetos activos son llamados robots móviles y pueden tener diferentes formas de locomoción, según se requiera; además, es posible emplear sensores y actuadores, cámaras, motores, emisores, receptores, entre otros, que el software suministra para definir el comportamiento del robot (Torres et al., 2013). Webots requiere de un archivo denominado "controlador" en que se puede definir el comportamiento de MECABOT 3.0 mediante el uso de las ecuaciones del modelo matemático. Este archivo es desarrollado en código C. Para las pruebas se varían los parámetros de la ecuación tanto para la configuración oruga (3) como para la configuración serpiente (14)(15), obteniendo datos de velocidad como indicador de rendimiento del sistema robótico modular MECABOT 3.0 con el fin de analizar su comportamiento en cada configuración y obtener un rango de funcionamiento adecuado para la locomoción. Para la obtención de datos de velocidad, se agrega un nodo en el software Webots llamado "floor", un piso métrico dividido en baldosas con escala 2:1; un marcador ubicado en la superficie y el contador de tiempo de simulación con el que cuenta el software.

3.2.1 Configuración Tipo Oruga

Para las pruebas en esta configuración se mantiene constante el parámetro $k = 3$ y se varían los parámetros A , f y M , registrando los resultados en gráficas para ser analizadas. En la Tabla 8 se observa de color azul los segmentos y en verde las articulaciones que conforman la cadena, evidenciando la formación de la onda sinusoidal al variar el parámetro A .

Tabla 8: Simulación Mecabot 3.0 Configuración Oruga

Amplitud d	$k = 3$
0,1	
0,3	
0,5	

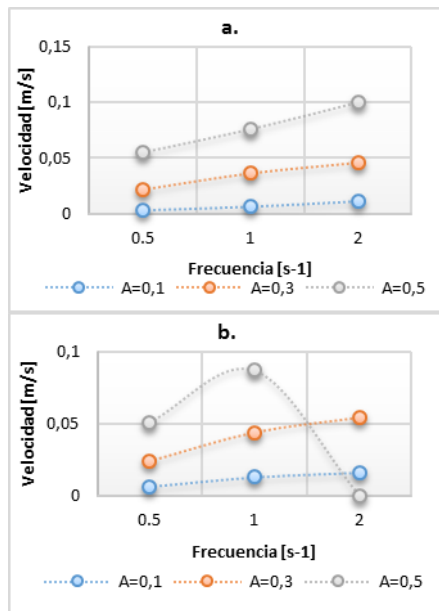
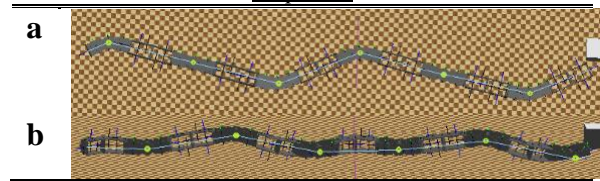


Fig. 8 Gráficos velocidades configuración tipo Oruga Pivote - Pivote $k=3$ a. $M=5$ b. $M=7$

En la Fig. 8a. se observa que incrementa la velocidad al aumentar la frecuencia y la amplitud. En la Fig. 8b. se observa que para $f > 1$ en $A = 0.5$ se pierde la estabilidad del robot.

3.2.2 Configuración Tipo Serpiente

Tabla 9: Simulación Mecabot 3.0 Configuración Serpiente



En la Tabla 9 se observan los dos componentes a. horizontal b. vertical que conforman la onda serpentina 3D para generar la locomoción en configuración tipo serpiente.

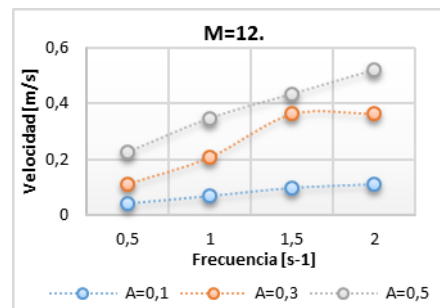


Fig. 9 Gráfico velocidad configuración tipo Serpiente Pivote - Pivote $k=2$

Se mantienen constantes los valores de $A_p = 0.1$, $k = 2$ y $M = 12$ variando A y f obteniendo las velocidades presentadas en la Fig. 9. Para todos los valores de A_h se tiene un incremento de la velocidad al aumentar la frecuencia. La velocidad máxima 0.52 m/s se da en $f = 2$ y $A_h = 0.5$ y la mínima es 0.04 m/s $f = 0.5$ y $A_h = 0.1$.

4 ANÁLISIS DE RESULTADOS

Mediante el desarrollo de las simulaciones para el sistema robótico modular MECABOT 3.0 se observa la aplicabilidad del modelo de osciladores de frecuencia fija propuesto por (Gonzales Gómez, 2008) ya que se evidencia en la Tabla 8 y Tabla 9 la formación de la onda serpentina a lo largo de la cadena teniendo en cuenta que siempre se tienen un mínimo de 2 puntos de apoyo $k \geq 2$ que permiten el avance del sistema. Las Fig. 8 y Fig. 9 muestran las velocidades obtenidas para cada configuración con respecto a la frecuencia y amplitud; para el caso de la oruga, también se varía el número de módulos M . El rango de funcionamiento obtenido para es $0.1 \leq A \leq 0.5$, $0.1 \leq A_h \leq 0.5$, $0.5 \leq f \leq 2$.

5 CONCLUSIONES

Con en el desarrollo del análisis matemático se evidencia que la locomoción de un sistema robótico modular basado en el enfoque bio-inspirado puede ser simplificado mediante el uso de osciladores

sinusoidales de frecuencia fija, representados por series de Fourier. De esta manera la ecuación describe una curva serpentina que, al variar la fase, genera una onda serpentina.

A partir del modelo matemático en tiempo continuo de la curva serpentina en 2D, se pasan a tiempo discreto las ecuaciones que definen el movimiento del sistema y se implementan en el software Webots. Así, fue posible realizar el análisis de la relación entre número de módulos, frecuencia y amplitud con respecto a la velocidad del robot, concluyendo que el parámetro frecuencia es directamente proporcional a la velocidad, al igual que el de amplitud.

Se comprobó la aplicabilidad de los generadores sinusoidales de frecuencia fija propuestos por (Gonzalez Gómez, 2008) al sistema robótico modular en cadena MECABOT 3.0, ya que, mediante la simulación del modelo en tiempo discreto, se observa la representación de la curva serpentina a lo largo de la estructura, generando la locomoción en una y dos dimensiones.

Se verificó que el modelo de los generadores sinusoidales de frecuencia fija se puede implementar en un archivo controlador en lenguaje de código C y permite la generación de la locomoción en configuración tipo oruga y serpiente.

RECONOCIMIENTO

Este documento es el resultado del proyecto de investigación en desarrollo ING – 2112 titulado “Rediseño y construcción de módulos para un sistema robótico reconfigurable de búsqueda y rescate, patrocinado por la Universidad Militar Nueva Granada.

REFERENCIAS

- Althoff, M., Giusti, A., & Icer, E. (2016). A task-driven algorithm for configuration synthesis of modular robots. *International Conference on Robotics and Automation (ICRA)* (págs. 5203-5209). Stockholm: IEEE.
- Arai, T., Inagaki, S., & Yuasa, H. (2003). CPG model for autonomous decentralized multi-legged robot system—generation and transition of oscillation patterns and dynamics of oscillators. *Robotics and Autonomous Systems*, 171 - 179.
- Bucher, D., & Marder, E. (2001). Central pattern generators and the control of rhythmic movements. *Current Biology*, 986-996.
- Crespi, A., & Ijspeert, A. J. (2007). Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. *IEEE International Conference on Robotics and Automation* (págs. 262-268). Roma: IEEE.
- Dario, P., Menciassi, A., Sfakiotakis, M., la Spin, G., & Tsakiris, D. P. (2005). Polychaete-like Undulatory Robotic Locomotion. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 3018-3023.
- Fukuoka, Y., Hada, Y., Kimura, H., & Takase, K. (2003). Adaptive Dynamic Walking of a Quadruped Robot 'Tekken' on Irregular Terrain Using a Neural System Model. *International Conference on Robotics & Automation* (págs. 2037 - 2042). Taipei: IEEE.
- Gonzales Gómez, J. (2008). *Robótica Modular y Locomoción: Aplicación a robots ápodos*. Madrid.
- Gualdrón Guerrero, O. E., & Gamarra Sierra, C. A. (2009). Diseño e implementación de un vehículo autónomo evasor de obstáculos empleando redes neuronales en un procesador digital de señales DSP56f801. *Revista Colombiana de Tecnologías de avanzada*, 83 - 88.
- Hirose, S. (1993). *Biologically inspired robots (Snake-like locomotor and manipulator)*. Oxford Science Press.
- Hoyos Gutiérrez, J. G., Cardona, J. E., Capacho, L. M., Prieto, F., & Pérez Cisneros, M. (2011). Control visual basado en posición de un brazo robótico. *Revista Colombiana de Tecnologías de avanzada*, 19 - 22.
- Ijspeert, A. J., & Crespi, A. (2008). Online Optimization of Swimming and Crawling in an Amphibious Snake Robot. *IEEE Transactions on Robotics*, 75-87.
- Medina Durán, J. O., Ferreira Jaimes, J., & Gualdrón Guerrero, O. E. (2009). Redes neuronales recurrentes en dispositivos lógicos programables para el control de un robot móvil. *Revista Colombiana de Tecnologías de Avanzada*, 75 -81.
- Rodríguez Oscar Oswaldo, Pineda Pinto Ronald Fernando, Cárdenas Pedro Fabián. (2012). Herramientas EJS 3D/MATLAB para el control del sistema no lineal aplicado al péndulo invertido sobre carro deslizante. *Revista colombiana de tecnologías de Avanzada*. 1 (19). Pág. 28 – 34.
- Rus, D., Salemi, B., Shen, W.-m., & Yim, M. (2007). Modular Self-Reconfigurable Robot Systems [Grand Challenges of Robotics]. *IEEE Robotics & Automation Magazine*, 43-52.

Torres Clayton José, Archila John Faber, Tronco Mário Luiz, Becker Marcelo, Viera Porto Arthur José, Tiberti Alexander José. (2013). Estudio cinemático de una plataforma robótica para agricultura. Revista colombiana de tecnologías de Avanzada. 2 (22). Pág. 131 – 137.