



Universidad Autónoma de Querétaro
Facultad de Ingeniería

Análisis Cinemático de Algoritmo de Locomoción para un Robot Inspirado en
Serpientes con Múltiples Grados de Libertad.

TESIS

Que como requisito para obtener el Grado de

M. en. C. Ingeniería Matemática

Presenta:

Carlos Alberto Martínez Gómez

Dirigido por:

Dr. José Emilio Vargas Soto

Sinodales:

Dr. José Emilio Vargas Soto
Presidente

Dr. Efrén Gorrostieta Hurtado.
Secretario

Dr. Gerardo Israel Pérez Soto
Vocal

M.C Luisa Ramírez Granados
Suplente

Dr. Roberto Augusto Gómez Loenzo
Suplente

Centro Universitario, Querétaro, Querétaro.

Fecha de aprobación del Consejo Universitario (mes, año), México

Resumen

La presente investigación parte del interés en comprender el proceso de creación de un algoritmo de locomoción de bajo nivel para un robot inspirado en serpientes con la finalidad de ofrecer una alternativa en la realización de labores o maniobras en tareas de riesgo para el ser humano y que puedan ser complicadas de realizar para robots con otra morfología. Adicionalmente, el proyecto de investigación aporta elementos académicos que propician un mayor conocimiento sobre este tipo de robots, así como la posibilidad de desarrollar en un futuro máquinas que generen beneficios a la sociedad. El trabajo inicia presentando un modelo cinemático de locomoción en línea recta para un robot tipo serpiente, esto a modo de comprender el funcionamiento del algoritmo, la creación del código y la simulación de dicho algoritmo, para proseguir con un modelo de locomoción en el plano, que requirió mayor dificultad para plantear y resolver. Se emplean relaciones de recurrencia para disminuir el cómputo del algoritmo empleado, en el primer caso, y un tratamiento vectorial en el segundo obteniéndose como curiosidad matemática una relación entre el número de elementos del polinomio de la solución del problema cinemático directo y la sucesión de Fibonacci. Los resultados finales de este trabajo son los dos modelos y su comprobación mediante animación utilizando software, que pudieran ser de interés para posteriores investigaciones en caso de que se desee aportar a la cinemática, plantear la dinámica y posteriormente construir el robot.

Palabras clave: Cinematica, robot, locomoción, algoritmo.

Abstract

This research starting from understand the process of design a locomotion algorithm of a robot inspired by snakes in order to offer an alternative in carrying out tasks or dangerous maneuvers for humans and that can be complicated. for robots with other morphology. Besides, this research contributes academic elements in order to improve knowledge about this kind of robots, and in the near future build them to benefit society through their multiple applications. We decide to start solving a kinematic model in straight line for a snake robot, in need to develop and complain the algorithm, creation of code and simulation. After, we continue with a model of locomotion in a XY plane, which required greater difficulty in planning and solving. Recurrence relations are used to decrease the computation of the algorithm, in 1D case, and a numerical method in X Y problem, obtaining as a mathematical curiosity a relation between the number of elements of the polynomial of the solution of the direct kinematic problem and the Fibonacci sequence. The final results of this work are the two models and their verification by animation using software, in case of interest to improve kinematics, propose their dynamics and build the robot.

Keywords: Kinematics, robot, locomotion, algorithm.

Dedicatoria.

A mis padres: Por su esfuerzo y dedicación para brindarme una formación como ser humano y profesionalista, la cual hoy puedo retribuir a nivel personal obteniendo un grado académico del cual pueden ellos sentirse orgullosos.

A la comunidad científica, mediante los cuales, sobre “hombros de gigantes” podemos aprender, y descifrar cómo funciona el universo en términos del entendimiento humano.

Dirección General de Bibliotecas UNQ

Agradecimientos.

Al consejo Nacional de Ciencia y Tecnología (CONACYT) por haber financiado mis estudios de posgrado.

A la Universidad Autónoma de Querétaro por poner a mi alcance los medios para superarme como profesionista y aprender cosas nuevas.

Al Dr. José Emilio Vargas Soto, por su constante apoyo, asesoría, retroalimentación y enseñanza, así como la confianza brindada para desarrollar la presente investigación durante más de dos años.

A los profesores del posgrado en Ciencias con terminal en Ingeniería Matemática, en especial a la profesora Luisa Ramírez Granados y el Dr. Roberto Augusto Gómez Loenzo por hacer de sus asignaturas impartidas una gran experiencia. Así como al Dr. Gerardo Israel Pérez Soto y el Dr. Jesus Jerónimo Castro, por sus apreciables aportaciones y sugerencias durante la realización de la presente investigación. Al M.C Javier Valle Mora, que motivó la curiosidad y potencial de estudiar matemáticas

Índice general.

Resumen.....	I
Abstract.....	II
Dedicatoria.....	III
Agradecimientos.....	IV
1 Introducción.....	1
1.1 Descripción del problema.....	6
1.2 Justificación.....	7
2 Antecedentes.....	10
2.1 Tipos de locomoción.....	11
2.1.1 Movimiento Ondulatorio.....	12
2.1.2 Lateral.....	13
2.1.3 Movimiento Rectilíneo.....	13
2.1.4 Concertina.....	14
2.2 Ecuación Serpenoide de Hirose.....	15
2.3 Otras modelaciones.....	17
2.4 Robots ápodos en la actualidad.....	18
2.5 Clasificación por su morfología.....	22
2.5.1 Modelos discretos.....	22
2.5.2 Modelos Continuos.....	23
2.5.3 Modelos Mecánicos.....	24
3 Hipótesis.....	26
4 Objetivos.....	26
4.1 Objetivo General.....	26
4.2 Objetivos Específicos.....	26
5 Metodología.....	27
5.1 Definición de Cinemática.....	28
5.1.1 Definición de algoritmo de locomoción y problema cinemático directo.....	28
5.1.2 Generadores sinusoidales (Dominio).....	28
5.1.3 Posición (Imagen).....	29
5.1.4 Problema cinemático directo (P.C.D).....	29

5.1.5	Algoritmo de locomoción.	29
5.2	Desarrollo del modelo unidireccional.....	30
5.2.1	Problema cinemático directo.....	30
5.2.2	Generadores sinusoidales.	32
5.2.3	Análisis de la función, y propuesta de mejora.	32
5.2.4	Programación del algoritmo.	34
5.3	Modelo Tridimensional.	36
5.3.1	Problema cinemático directo.....	37
5.3.2	Generadores sinusoidales.	43
5.3.3	Análisis de la ecuación de calor.....	44
5.3.4	Obtención de las funciones generadoras de amplitud.	46
5.3.5	Limpieza de la señal.	52
5.3.6	Modelo final.	53
5.3.7	Algoritmo de locomoción.	54
5.3.8	Rutinas de viraje.	55
5.3.9	Evasión de obstáculos.....	56
6	Resultados y discusión.	58
6.1	Modelo unidireccional.....	58
6.2	Modelo bidireccional.....	62
6.3	Discusión de resultados.	79
6.3.1	Modelo unidireccional.	80
6.3.2	Modelo Tridimensional.....	82
7	Conclusiones.	87
	Bibliografía.	92
	Anexos.	95
	Anexo 1. Solución al problema cinemático directo en 3 dimensiones.....	95
	Anexo 2. Comprobación computacional de la ecuación 34.....	102
	Anexo 3. Reporte técnico donde se buscó obtener la relación expresada en la ecuación 21, de forma cerrada.	106
	Anexo 4. Publicaciones realizadas en base a 1. Desarrollo del algoritmo de locomoción, 2. Mejora del algoritmo.....	113

Índice de figuras.

Figura 1.Robots de servicio y sus aplicaciones.....	2
Figura 2.Robots de servicio móviles.....	3
Figura 3.Estudio de la locomoción en nivel bajo (1 y 2) y alto nivel (3) entre dos puntos.	5
Figura 4. Descripción del objetivo de un modelo matemático de locomoción de un Snake Robot.....	10
Figura 5. Bio-mecanismo de la serpiente que facilita el movimiento lateral, ubicado a lo largo de la espina dorsal.	11
Figura 6.Estrategia de locomoción de tipo rectilínea.....	14
Figura 7.Bourque, J. (1985). Patrón de movimiento tipo concertina, recuperado de Functional Vertebrate Morphology.	14
Figura 8.Patrones de locomoción serpentinoide, de arriba abajo y de izquierda a derecha: 1. Rectilíneo, 2. Serpentinoide, 3. Concertina, 4. Ondulación Lateral. Enciclopedia Británica (2012).....	15
Figura 9. ACM III's, primer robot ápodo desarrollado por Shigeo Hirose.	16
Figura 10. Robot MAKRO.....	17
Figura 11.Etapas dentro de una labor de inspección de un robot ápodo.	19
Figura 12.Aplicaciones de robots ápodos, de izquierda a derecha, arriba abajo: robot serpiente trabajando en labores de rescate, robot inspector de ductos, robot snake de inspección subacuática, robot de desplazamiento en 3 dimensiones	21
Figura 13.Mecanismo Elephant Trunk,.....	23
Figura 14.Metodología empleada dentro de la investigación.	27
Figura 15.Desarrollo para la construcción de los dos algoritmos de locomoción.	28
Figura 16.Cadena cinemática con desplazamiento en 2 dimensiones.....	30
Figura 17.Algoritmo de locomoción para el caso unidireccional.....	34
Figura 18.Algoritmo de programación para simulación del robot.	35
Figura 19.Pasos a seguir para la construcción de un algoritmo de locomoción para un robot tipo Snake.	36
Figura 20. Diagrama de robot tipo Snake 3D.	36
Figura 21. Solución de la ecuación de calor unidimensional.....	45

Figura 22.Movimiento de Tipo Concertina. (2012). Tomado de Enciclopedia Británica. 46

Figura 23.. Diagrama de flujo de algoritmo de locomoción en 3 dimensiones para un robot Snake.....55

Figura 24.Cambio de dirección aprovechando los instantes de amplitud cero.....56

Dirección General de Bibliotecas UAQ

Índice de gráficas.

Gráfica 1.Posición de los eslabones de una simulación de robot ápedo unidireccional.	24
Gráfica 2.Rango de trabajo para $\varphi(t)$ proporcionado por la literatura.	33
Gráfica 3.Comportamiento de la secuencia propuesta e implementada.	34
Gráfica 4.Análisis Vectorial del Problema de cinemática directa.	40
Gráfica 5. Visualización del robot enviando las señales expresadas en la ecuación 37.	41
Gráfica 6.Crecimiento del cómputo por n-ésima articulación del P.C.D.	43
Gráfica 7.Distribución del calor en una varilla durante $t= 1$ segundo.	45
Gráfica 8.Vista Superior de la expansión y contracción del robot en 3 dimensiones.	47
Gráfica 9.Comportamiento de la función amplitud con escalón unitario.	50
Gráfica 10.Implementación de la función amplitud al generador Serpenoide.	51
Gráfica 11.Implementación de los generadores al robot en 3D.	51
Gráfica 12.Movimiento del robot al procesar la señal corregida.	53
Gráfica 13.Estrategia de evasión de obstáculos.	57
Gráfica 14.Evolución del algoritmo de locomoción de la animación 1.	59
Gráfica 15. Visualización del desplazamiento de la animación 2.	61
Gráfica 16. Visualización de la animación 3.	63
Gráfica 17. Vista superior de la animación 3.	64
Gráfica 18.Evolución del algoritmo de locomoción en la animación 4.	66
Gráfica 19.Vista superior de la evolución de la animación 4.	66
Gráfica 20.Evolución del algoritmo de locomoción en la animación 5.	69
Gráfica 21.Vista superior de la evolución del algoritmo en la animación 5.	69
Gráfica 22.Evolución del algoritmo de locomoción en la animación 6.	71
Gráfica 23.Vista superior de la evolución del algoritmo de locomoción en la animación 6.	71
Gráfica 24. Evolución del algoritmo de locomoción.	73
Gráfica 25. Vista superior de la evolución del algoritmo de locomoción.	73
Gráfica 26. Resultados de la ejecución del algoritmo de la animación 8.	74
Gráfica 27.Funciones implementadas en el algoritmo de la animación 8.	75

Gráfica 28. Vista superior de la evolución del algoritmo de locomoción de la animación 8.	75
Gráfica 29. Animación del algoritmo en la animación 9.....	77
Gráfica 30. Vista superior de la evolución del algoritmo de locomoción en la animación 9.	77
Gráfica 31. Visualización de la animación 10.....	79
Gráfica 32. Vista superior de la evolución del algoritmo de locomoción de la animación 10.	79
Gráfica 33. Etapas de progresión dentro de un ciclo de locomoción de un snake robot empleando ambos algoritmos.	81

Dirección General de Bibliotecas UAG

Índice de tablas.

Tabla 1. Investigaciones previas en robots ápodos.....	20
Tabla 2.Cantidad de términos del polinomio solución del problema cinemático directo usando residuos.	42
Tabla 3.Comparativa del cómputo del P.C.D usando los tres métodos.	43
Tabla 4.Instantes y valores críticos.	48
Tabla 5. Parámetros implementados en la animación 1.....	58
Tabla 6. Función generadora de la animación 1.	59
Tabla 7. Mediciones obtenidas de la animación 2.....	59
Tabla 8. Parámetros implementados en la animación 2.....	60
Tabla 9. Funciones enviadas para generar la animación 2.....	60
Tabla 10.Mediciones obtenidas de la animación 2.....	61
Tabla 11. Señales enviadas al generador para la animación 3.....	62
Tabla 12. Ajuste de parámetros para la animación 3.....	63
Tabla 13. Resultados obtenidos de la animación 3.....	63
Tabla 14. Parámetros para generar la animación 4.	65
Tabla 15 Funciones implementadas en el algoritmo de locomoción de la animación 4.	65
Tabla 16. Mediciones obtenidas de la animación 4.....	67
Tabla 17.Parámetros implementados en el algoritmo de la animación 5.	68
Tabla 18.Funciones implementadas en el algoritmo para la animación 5.....	68
Tabla 19.Resultados de la ejecución del algoritmo de la animación 5.	68
Tabla 20.Parámetros implementados en el algoritmo de la animación 6.	70
Tabla 21.Funciones implementadas en el algoritmo de la animación 6.	70
Tabla 22.Resultados de la ejecución del algoritmo en la animación 6.....	70
Tabla 23 Parámetros implementados en el algoritmo en la animación 7.	72
Tabla 24.Funciones implementadas en el algoritmo de la animación 7.....	72
Tabla 25. Resultados de la ejecución del algoritmo de la animación 7.	72
Tabla 26. Parámetros implementados en el algoritmo en la animación 8.	74
Tabla 27.Funciones implementadas en el algoritmo de la animación 8.....	74
Tabla 28.Parámetros implementados en el algoritmo en la animación 9.	76
Tabla 29.Funciones implementadas en el algoritmo de la animación 9.....	76

Tabla 30. Resultados de la ejecución del algoritmo de la animación 9.	76
Tabla 31. Parámetros implementados en el algoritmo en la animación 10	78
Tabla 32. Funciones implementadas en el algoritmo de la animación 10	78
Tabla 33. Resultados de la ejecución del algoritmo de la animación 10.	78

Dirección General de Bibliotecas UAQ

1 Introducción.

La robótica Industrial y de Servicios comenzó a finales del primer tercio del Siglo XX, durante el periodo de expansión de la Industria, en el cual, las líneas de producción en serie se volvieron más complejas y exigían cada vez más procesos simultáneos, para lo cual la mano del obrero fue insuficiente (Sánchez-Martín, et al., 2007)

El primer robot industrial fue desarrollado por Bill Griffith P. Taylor en el año de 1937, el robot funcionaba mediante placas perforadas que activaban solenoides que permitían el control de una grúa.

Actualmente, se considera a un robot de tipo industrial a aquellos que cumplen con ser un “Manipulador multifuncional, controlado automáticamente, reprogramable en tres o más ejes, que puede estar fijo o móvil para uso en aplicaciones de automatización industrial” (ISO Standards, 2012).

Por contraparte, la robótica de servicios se enfoca en desarrollar robots que ayuden al ser humano en labores no necesariamente industriales, la Federación Internacional de Robótica, los define como “Robot que opera de forma parcial o totalmente autónoma al servicio del bienestar de los seres humanos y de equipamientos, excluyendo operaciones manufactureras” (IFR, 2016).

Este tipo de robots pueden ser fijos o móviles. En cuanto a los primeros, han sido empleados en labores que requieren precisión como asistentes en cirugías, atención médica, diagnóstico de enfermedades, apoyo a personas con capacidades diferentes, asistente personal, entre muchas otras (figura 1):

Sin embargo, la mayoría de los robots de servicios son móviles, su capacidad para desplazarse en el espacio facilita la realización de tareas, sobre todo en áreas donde el tránsito del ser humano es complicado ya sea por las condiciones del terreno, o es en extremo riesgoso por otros factores implicados (radioactividad, concentración de gases tóxicos, carencia de oxígeno).



Figura 1. Robots de servicio y sus aplicaciones.

Dentro de las aplicaciones de los robots móviles de servicios, podemos encontrar que estos han sido usados en múltiples aplicaciones, como: exploración de otros planetas, zonas radioactivas, robots caminantes en zonas volcánicas, asistentes de labores geriátricas, cuidados de personas con discapacidades, labores domésticas, caracterización de suelos. rescate de personas entre otras, algunas de ellas pueden apreciarse en la figura 1.

La realización de estas tareas por parte de los robots de servicios, ha traído como beneficio la continua generación de conocimiento, caracterización del entorno en el que el ser humano no podría acceder, así como reducción del riesgo a que un ser humano normalmente se expondría de realizar dichas tareas. Los robots móviles a su vez pueden clasificarse por su tipo de locomoción, algunos se aprecian en la figura 2, estos pueden ser:

- Con Ruedas.
- Con cintas de deslizamiento.
- Robots Caminantes.
- Robots Ápodos o Modulares
- Otros.

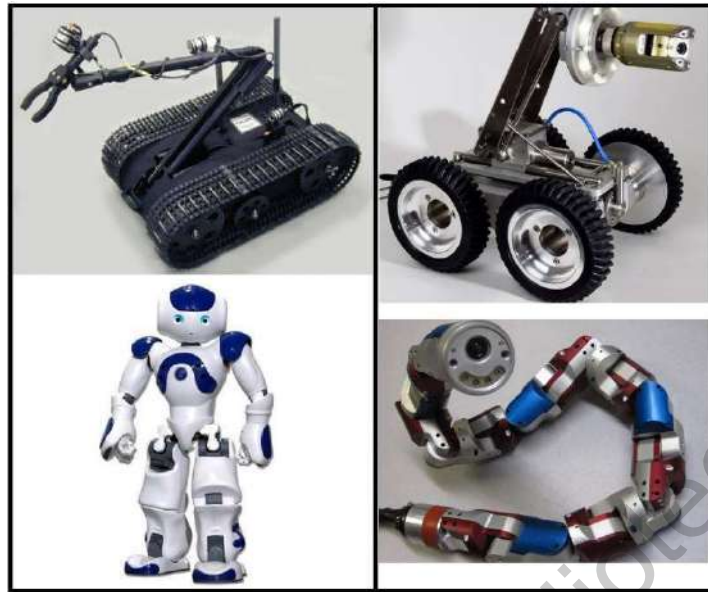


Figura 2. Robots de servicio móviles.

La elección del tipo de locomoción adecuada depende del tipo de tarea a realizar, generalmente se tienen en consideración los siguientes aspectos:

Maniobrabilidad: La capacidad de realizar una rutina de movimiento adaptable al entorno incluyendo casos críticos, esto, es decir, la capacidad de moverse dentro del entorno en curvas, secciones estrechas o de poco margen de movimiento. en cuanto a largo o ancho del espacio de desplazamiento.

Controlabilidad: La capacidad del robot para ser dirigido dentro del entorno sin que se atasque o caiga, también puede interpretarse como la capacidad del robot para leer las cualidades del entorno y operar en base a este.

Tracción: El robot debe tener la capacidad de desplazarse por cuenta propia hacia la dirección a la que se aplica el esfuerzo, lo cual puede dificultarse si la superficie de desplazamiento cambia de un momento a otro, y la rutina de desplazamiento debe modificarse sobre la marcha.

Capacidad de subir pendientes: Generalmente una superficie de desplazamiento en la cual interese que un robot supla a un ser humano, será irregular, ya que es una condición donde radica el riesgo de desplazarse para un ser humano,

por tanto, al elegir un robot móvil, se buscará que este pueda desplazarse por una pendiente.

Eficiencia: Debido a que un ciclo de trabajo o una rutina de movimiento, puede desarrollarse en un contexto de que el robot trabaje desconectado de un suministro de energía, está sea aprovechada al máximo para que el consumo de energía genere la mayor rentabilidad adecuada.

Mantenimiento: Al elegir un robot, se elegirá uno que pueda ser desmontado para reparación o sustitución de piezas, sin tener que acudir necesariamente a un taller, torno o sitio especializado o ya que estas labores podrían ser realizadas en el mismo lugar de trabajo y de forma rápida, además de que pueda contarse con un stock de refacciones para realizar dichas actividades

Consideraciones de navegabilidad: Una vez identificado el tipo de superficie de entorno (tipo de irregularidades, espacio disponible, inclinaciones, tipo de suelo), se buscará un robot que cumpla con poderse moverse en las condiciones planteadas.

En cuanto a la locomoción, su estudio puede dividirse en dos enfoques, en bajo y en alto nivel:

La locomoción de bajo nivel se encarga de determinar las formas de locomoción, mediante un algoritmo, o un modelo que permitan el avance del robot mediante los correctos giros de los servomotores.

Esta a su vez puede dividirse en nivel 1, donde se debe resolver la cinemática en al menos una línea recta, y el nivel 2, en donde se debe plantear la locomoción dentro de un plano.

Mientras que, en la locomoción de alto nivel, se hace énfasis en planificación de trayectorias, diseño de curvas, estrategias de locomoción en casos críticos. El presente trabajo se enfocará en locomoción de bajo nivel 1 y 2. la diferencia de estos estudios en la figura 3.

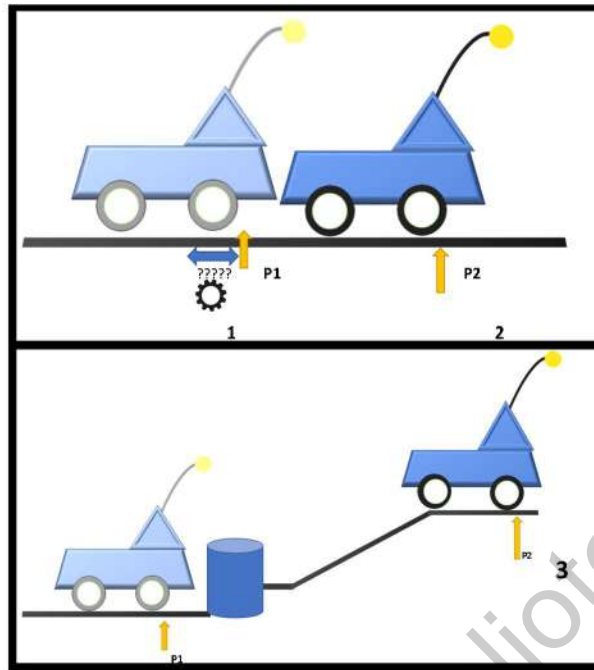


Figura 3. Estudio de la locomoción en nivel bajo (1 y 2) y alto nivel (3) entre dos puntos.

Dentro de los robots móviles de servicios, se hizo un especial enfoque en los robots de locomoción ápoda, debido a que es una morfología poco usual, y cuya línea de investigación está en vías de desarrollo, por lo cual se ha visualizó un área de oportunidad, además por ventajas en su morfología y su tipo de locomoción, se plantea que este pudiera aportar diversas soluciones a problemáticas actuales.

Los robots ápodos, reciben diversos nombres en la literatura, podemos encontrarlos como: Robot Tipo Snake, Robot Modular, Robot Ápodo, Cadena Cinemática abierta Móvil, Robot Bio-Inspirado en serpientes, Serpentinoide. Por lo cual, cuando estos términos sean mencionados, nos estaremos refiriendo al mismo tipo de robot.

Las características de los robots ápodos son:

Distribución modular: La estructura morfológica del robot se divide en módulos, que pueden ser dependientes de otro, o independiente, es decir con su propio servomotor. Lo cual permite que estos puedan oscilar para generar un patrón de movimiento.

Flexibilidad: Mientras más módulos tenga un robot ápedo y estos sean más pequeños, este tendrá la capacidad de doblarse más, y efectuar oscilaciones con mayor amplitud y frecuencia, lo cual puede ser útil en el diseño de algoritmos de locomoción y planificación de trayectorias.

Rutinas de movimiento: Su estructura puede modificarse para efectuar diversos patrones de movimiento según el caso, subir escalones, pendientes, trepar por un árbol, moverse por la arena o un suelo irregular.

Tracción: La tracción de un robot ápedo, se genera mediante un impulso causado por el movimiento sinusoidal característico de los organismos biológicos y este genera un esfuerzo contra la superficie, generando, con el patrón adecuado, que el robot se impulse en una dirección en específico.

Este patrón es generado por una señal de entrada al robot, la cual genera una respuesta en forma de movimiento del mecanismo.

Maniobrabilidad. dados puntos críticos dentro de una trayectoria, puede modificarse la señal para que el robot, disminuya o aumente su amplitud de oscilación o frecuencia, con la finalidad de liberarse de un obstáculo o desplazarse más rápido en caso que las circunstancias lo permitan, o cambiar de rutina, por ejemplo, puede pasar de una rutina para andar dentro de una tubería, a una en la cual deba subir por escaleras.

El diseño de algoritmos de locomoción de estos robots bioinspirados constituye una línea de investigación en robótica. Imitar patrones morfológicos y de desplazamiento, trae por ende como resultado, eficiencia energética, estabilidad y adaptación al entorno, debido a que la selección natural, a través de los procesos evolutivos, ha mejorado el proceso de locomoción de las especies animales a través de millones de años. Haciendo de estos una opción interesante a seleccionar al elegir un robot móvil para realizar una tarea.

1.1 Descripción del problema

Al construir el algoritmo de locomoción, el problema a resolver será el siguiente:

Dada una estructura morfológica, la cual posee movimientos se repiten recursivamente. y de la cual, al resolver el problema cinemático, se obtiene una relación en la cual se describe la posición de cada una de las articulaciones respecto al tiempo.

Se desea estudiar entonces el proceso de obtención de estas funciones, debido a que, si el patrón de movimiento es recursivo, hay posibilidad de obtener una función que pueda expresarse mediante una relación de recurrencia, y si, las señales de entrada al robot para generar movimiento, de igual forma pueden expresar como una función dentro de un ciclo de locomoción, entonces se obtendrá una función continua para la ejecución dentro del ciclo de locomoción.

La finalidad de este enfoque dentro del estudio será explorar la posibilidad de que con relaciones de recurrencia podamos disminuir el coste en cuanto a cómputo del problema y/o generar un algoritmo eficiente que pueda ser mejorado en estudios posteriores.

1.2 Justificación.

La presente tesis se enmarca dentro del análisis cinemático de un robot modular, centrado en la planeación de estrategias de locomoción que permitan al robot moverse de forma eficiente, con un espacio de trabajo reducido. El problema a resolver es que el movimiento del robot se efectúe de forma estable, por lo cual el estudio de la locomoción se llevará a cabo en bajo nivel. empleando herramientas matemáticas para disminuir el cómputo del problema.

Desde el punto de vista académico, esta investigación es justificable debido a que mediante ella puede generarse nuevo conocimiento a través de un modelo matemático que facilite el entendimiento sobre la forma en cómo se desplazan este tipo de robots.

Dicho conocimiento permitirá abatir la **brecha tecnológica** generada por países que llevan décadas con líneas de investigación con este tipo de robots bio-inspirados, y

de esta forma poder aportar alternativas para solucionar problemas relacionadas con la agilización de labores de inspección en zonas peligrosas.

Desde el punto de vista económico, el importar tecnologías de otros países es siempre más costoso, según la Conferencia de las Naciones Unidas sobre Comercio y Desarrollo, México importa el 3.5% de los bienes de tecnologías de la información a nivel mundial, lo cual equivale a 65,027 Millones de Dólares. Según (Menchaca, 2010). México importa 20 veces más tecnología de la que produce. Por lo cual, en la medida que se realice innovación y desarrollo de tecnología, como es el caso de este proyecto, puede reducirse la **brecha económica**.

La elaboración de este proyecto mecatrónico que inicia con la formulación del modelo matemático y el análisis dinámico del robot puede traer distintas aplicaciones futuras de utilidad al país, como podrían ser:

Rescate de personas: Las características morfológicas de un robot ápedo, permiten que este pueda desplazarse por espacios reducidos, como zonas con escombros, convirtiéndolo en una herramienta futura para rescate de personas en zonas de desastre, como terremotos, huracanes.

Reconocimiento de las condiciones ambientales: En ocasiones no hay garantía que un espacio de trabajo sea propicio para que el ser humano efectúe cierta actividad por diversos factores (concentración de oxígeno, gases tóxicos, temperaturas extremas), si el acceso a este lugar es complejo, un robot serpentiforme podría realizar esta actividad, y de esta forma evitar poner en riesgo vidas humanas.

Aplicaciones Militares: Las características de un robot modular, permiten que este pueda desplazarse sigilosamente por terrenos llenos de vegetación, y a simple vista no sea perceptible, por lo cual podría ser empleado por organismos nacionales de Defensa, en labores de seguridad y espionaje.

Arqueología: Acceder a lugares con muchos años de antigüedad y de alto valor histórico y cultural, representa en ocasiones una labor invasiva, ya que el desplazarse por estos lugares puede ocasionar la contaminación o daño físico del mismo, poniendo

en riesgo su conservación, incluso, en ocasiones ingresar a estos, solo puede hacerse mediante espacios muy reducidos, por lo cual la morfología de un robot inspirado en serpientes puede ser adecuada para estas tareas.

Dirección General de Bibliotecas UAQ

2 Antecedentes

Los robots con esta morfología, vieron su origen en la década de los 90's, en Japón. Cuando el Dr. Shigeo Hirose, comenzó a estudiar las serpientes biológicas, con la finalidad de aprovechar su morfología y su patrón de locomoción y mediante estos diseñar robots bio inspirados. Los robots ápodos se caracterizan por realizar un patrón de movimiento empleado por reptiles y algunos anfibios, el cual ya ha sido caracterizado matemáticamente, a través de la ecuación serpentinoide (Hirose & Mori, 1993), el objetivo de la investigación en ese entonces, fue:


1.- El interés por obtener una interpretación matemática de la locomoción de las serpientes.

2.- El desarrollo de futuras aplicaciones en robots que pudieran adoptar dicho algoritmo de locomoción.

De acuerdo con Hirose & Yamada, (2009), las investigaciones anteriores, empleando métodos poco consistentes, (observación de organismos ápodos en entornos artificiales), no habían dado resultados que permitieran obtener un modelo en concreto. Debido a esto, el problema de interpretar la locomoción ápoda en modelado dinámico y cinemático, comenzó a abordarse desde la física clásica. Una interpretación visual del problema a resolver, puede visualizarse en la figura 4.

¿Puede el patrón de movimiento recursivo de las serpientes caracterizarse matemáticamente?

• Shigeo Hirose
(1973)

$f(q_i) \Rightarrow$ 

Ecuación de Hirose



ACM III

Figura 4. Descripción del objetivo de un modelo matemático de locomoción de un Snake Robot.

Para esto, se replanteó el problema haciendo enfoque en describir matemáticamente los distintos tipos de locomoción que utilizan los organismos ápodos. los cuáles serán descritas a continuación.

2.1 Tipos de locomoción

Los mecanismos que llevan a un organismo que carece de extremidades a desplazarse, es algo muy complejo, ya que estos pueden moverse con facilidad por diversos entornos, desde una selva tropical hasta un desierto, nadar por ríos y mares, trepar e inclusive auto- propulsarse para lanzarse por copas de elevados árboles, todo esto anterior, con velocidad y pasando generalmente desapercibido por otros organismos. El movimiento de las serpientes proviene de un sistema de articulaciones que se compone por un juego de tendones y vertebras. (Lilywhite, 2014). Y es detallado a continuación en la figura 5.

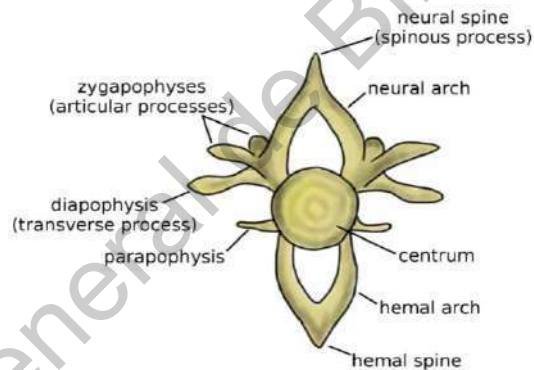


Figura 5. Bio-mecanismo de la serpiente que facilita el movimiento lateral, ubicado a lo largo de la espina dorsal (Galbusera, 2019).

El movimiento lateral es posible debido a la rotación posible sobre los ejes X & Y entre las vértebras de la serpiente, mientras que el movimiento que proporciona la capacidad de la serpiente para elevarse es generado por tendones anclados a las partes nombradas como *Zygapophyses*, *Diapophysis* y *Parapophysis* que se visualizan en la figura 5.

Como menciona Gray (1946), para que un organismo ápodo pueda generar desplazamiento, es necesaria la existencia de fuerzas en dirección normal hacia el cuerpo del organismo, esta investigación, si bien no tuvo como objetivo la construcción

de los robots, si dio ciertas descripciones matemáticas de las fuerzas de contacto para poder describir la locomoción.

El cuerpo de una serpiente puede visualizarse como un cuerpo cuya masa se encuentra prácticamente uniformemente distribuida, y a pesar de que podríamos decir que la superficie que presenta fricción con el suelo es bastante alargada, lo cierto es que generalmente al moverse, reduce la superficie de contacto considerablemente.

La clasificación del movimiento Serpenoide, se presenta a continuación, cabe destacar que el movimiento rectilíneo no es único de los reptiles en cuestión, sino puede ser útil para describir el movimiento de otros organismos ápodos.

2.1.1 Movimiento Ondulatorio.

Es el comúnmente usado por los organismos ápodos, describe, como su nombre lo indica, un movimiento en forma ondulatoria, la locomoción es sostenida por la superficie ventral y lateral de la serpiente, es aprovechado cuando la superficie de desplazamiento proporciona apoyo para el desplazamiento y posee pocas irregularidades.

Esta estrategia de movimiento consiste en aprovechar al máximo el entorno, cada superficie de contacto posible, se emplea para apoyarse y obtener una fuerza de reacción siguiendo una secuencia, la cual va desde la cola hasta la cabeza, la suma de las fuerzas de reacción obtenidas, dará como resultado un desplazamiento progresivo hacia adelante. Las serpientes más largas, tienen por ende mayor superficie de contacto por aprovechar. Sin embargo, las fuerzas de reacción no se aprovechan tanto en especies más robustas o que no tienen tanta elasticidad, mientras que las especies más delgadas, pueden obtener velocidades de 12 a 13 kilómetros por hora, como ejemplos, tenemos a serpientes arbóreas y mamba negra (Lilywhite, 2014).

Biomecánicamente, el movimiento ondulatorio emplea un amplio radio de rotación, entre 30 y 60 ° sobre la superficie de contacto mientras que la amplitud sobre esta misma es mucho más baja y variable dependiendo de la especie.

2.1.2 Lateral.

Este tipo de locomoción es comúnmente usado por serpientes del desierto, conocido como *Sidewinding*, la finalidad de este patrón de movimiento es reducir la superficie de contacto entre el cuerpo de la serpiente y el suelo, aprovechando además la componente de fricción que pueda ser obtenida por estas superficies, otro factor que lleva a estos organismos a emplear el serpenteo lateral, es el ahorro de energía cuando el alimento escasea. El patrón de movimiento consiste en fijar al menos dos puntos del vientre de la serpiente y ondular el cuerpo hacia un lado, aprovechando las irregularidades empleándolas como un punto para obtener apoyo, y producir un impulso.

Un aspecto clave de este patrón de movimiento es la habilidad de la serpiente para apoyar su el peso de su cuerpo en los puntos de contacto con la superficie, siendo muy importantes en este proceso la musculatura axial al movimiento.

El desplazamiento producido generalmente posee un desfase de 30° grados con respecto a la vertical, con una velocidad promedio de 3.7 kilómetros por hora.

2.1.3 Movimiento Rectilíneo.

Este patrón de movimiento es ejecutado con la finalidad de establecer un desplazamiento únicamente en línea recta. Las serpientes suelen usarlo para acechar o moverse sin ser detectadas.

Este patrón de movimiento emplea fuerza obtenida casi en su totalidad por los músculos ventrales. Con el objetivo de flexionar y comprimir el dorsal subcutáneo repetidas veces, apoyándose sobre la superficie de contacto para así generar una fuerza de reacción y desplazarse hacia adelante.

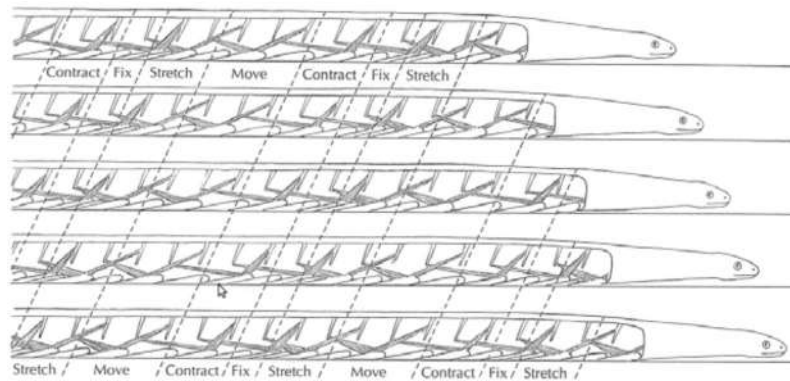


Figura 6. Estrategia de locomoción de tipo rectilínea.

El movimiento que se produce suele ser lento y debe observarse por un tiempo lo suficientemente largo para notar que se está produciendo un desplazamiento, es propio de ápodos de mayor peso, como son boas y pitones.

2.1.4 Concertina

En esta forma de locomoción, la serpiente fija ciertas partes de su organismo para obtener un punto de apoyo e impulsarse hacia adelante, realizando ciclos de contracción y expansión consecutivamente, este mecanismo consiste en obtener continuamente fuerzas de reacción, inducidas por la musculatura lateral y sustentadas por la columna vertebral. La progresión de este tipo de desplazamiento puede apreciarse en la figura 14.

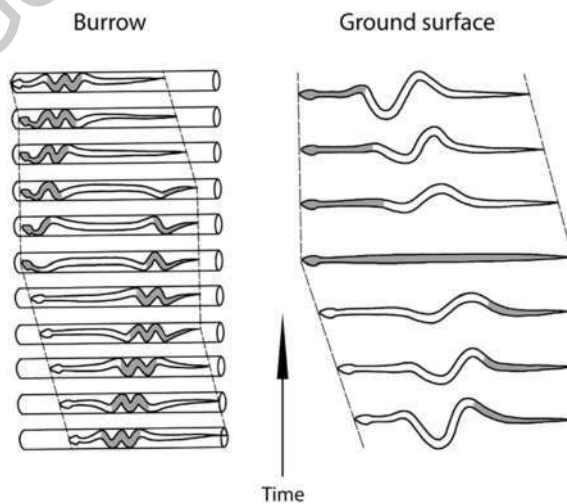


Figura 7. Patrón de movimiento tipo concertina. (Bourque, J. , 1985).

Este movimiento suele ser utilizado cuando estos animales desean desplazarse dentro de túneles o cavidades estrechas, usando las paredes como puntos de apoyo para poder desplazarse. Es un movimiento que en términos energéticos genera desgaste, debido al constante proceso de estirar e impulsarse debido a la complejidad del entorno en que suele emplearse. La velocidad alcanzada en promedio utilizando el patrón de movimiento concertina es de 0.1 kilómetros por hora. (Lilywhite, 2014).

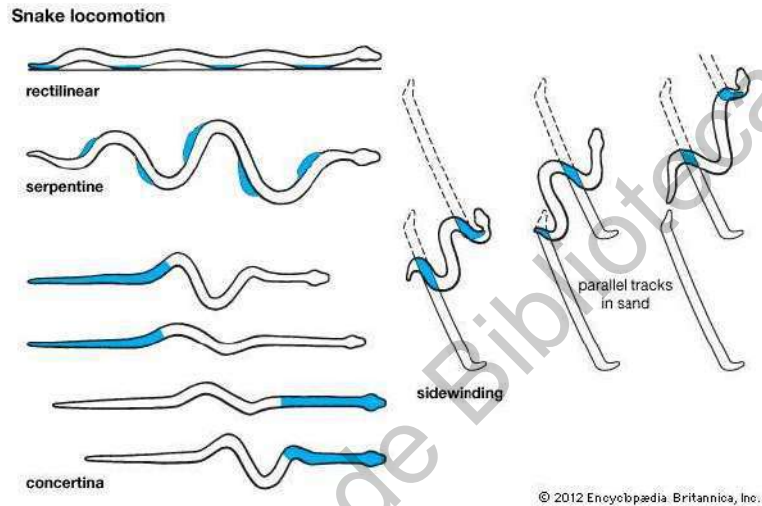


Figura 8. Patrones de locomoción serpentinoide, de arriba abajo y de izquierda a derecha: 1. Rectilíneo, 2. Serpentinoide, 3. Concertina, 4. Ondulación Lateral. (Enciclopedia Británica, 2012).

2.2 Ecuación Serpentina de Hirose

Dada esta descripción del movimiento de este tipo de animales, fue cuestión de tiempo para que surgieran investigaciones desde el punto de vista de la bio-mecánica.

Descubriendo que las curvas que hacen las serpientes al moverse también seguían una cierta ley de la naturaleza, pudo generalizarse este patrón mediante una ecuación matemática. Esto dio lugar al establecimiento de una fórmula, conocida como "Curva Serpentina". La cual, funciona como un generador senoidal, y puede caracterizar dicho patrón de locomoción, el cual ha sido aplicado al área de robótica con la finalidad de construir estructuras bio-inspiradas en estos animales.

La "Curva Serpentina de Hirose, se expresa de la siguiente forma:

$$K(s) = \frac{-2\pi k}{l} \alpha \sin\left(\frac{2\pi K}{l} s\right) \quad (1)$$

Donde:

l: longitud de la curva [m].

s: distancia a lo largo de la curva [m].

k: número de ondulaciones, $K > 0$.

α : Angulo de serpenteo [Rad]

Mediante esta función, podemos generar señales con cierto patrón sinusoidal conocidas como serpentinoideas, que, implementadas correctamente, generará un patrón de desplazamiento recurrente y estable con el movimiento característico de las serpientes (Garzón, 2011). La selección de estos patrones de locomoción dependiendo del entorno ha sido objeto de estudio, tanto teórico como en simulación computacional. cómo puede observarse en los trabajos de Zhang & Guo (2009) y San (2011).

Como conclusión de esta investigación se determinó que el movimiento ondulatorio, es el más suave y eficiente para la locomoción, además, se concluyó que los demás patrones de locomoción de arrastre en el espacio, son producto de la combinación de dos o más ondas serpentinoideas.



Figura 9. ACM III's, primer robot ápodico desarrollado por Hirose (1993).

Después del trabajo inicial de Shigeo Hirose, se han desarrollado diversas aportaciones a esta línea de investigación, así como otros tipos de modelaciones.

2.3 Otras modelaciones

Otra forma de tratar la locomoción ápoda, es considerar a todo el sistema de mecanismos, como un sistema no holonómico, es decir, un sistema en el cual, no podemos establecer un cambio de dirección en el robot, sin modificar previamente las directrices que controlan el vector de desplazamiento. Otra característica que diferencia a estos sistemas, es que tiene menos grados de libertad a controlar que el número total, lo cual, dificulta en sobremedida la tarea de controlarlo.

Ostrowski & Burdick (1998), abordaron el diseño de algoritmos de locomoción ápoda empleando diversas estrategias matemáticas para resolver el problema del control de estos robots, que incluye tratarlos como un sistema hiper- redundante.

Una aplicación usual para este tipo de modelos, es el de robots modulares que emplean ruedas auxiliares, para incrementar la velocidad en espacios en que esto es posible, y aumentar la capacidad de respuesta en caso de que la inspección lo requiera -En la figura. Puede observarse a MAKRO, un robot con ruedas empleado por la industria para labores de inspección.



Figura 10. Robot MAKRO

De forma similar, tomando como inspiración otro organismo ápodo como son las orugas, *Copernicus*, es un robot cuya construcción e investigación es financiado por la unión europea, con la finalidad de inspeccionar tuberías a pequeña escala (González, 2002).

2.4 Robots ápodos en la actualidad.

Estas investigaciones dieron paso a la creación de diversas gamas de robots ápodos para distintas aplicaciones como exploración en tanques, ductos de la industria petrolera, y/o exploración submarina, así como en entornos hostiles para el ser humano. (Liljebäck et al, 2011).

Según Rollinson (2014) Los robots serpiente son adecuados para entornos altamente reducidos porque su tamaño y su seccionamiento son altamente redundantes y les permiten entrar y moverse a través de espacios reducidos. Debido a esto, una aplicación de los robots apodos es la de operaciones de búsqueda y rescate en zonas de desastre y sismos.

Los robots ápodos poseen muchas ventajas que los convierten en una propuesta interesante para realizar diversas tareas de inspección, dentro de las cuales podemos destacar:

Convierte los obstáculos del entorno en puntos de apoyo, debido a su estructura morfológica, la locomoción de estos robots puede adaptarse al entorno, generando distintos arreglos de los módulos y moverse con mayor eficacia según sea el caso. (Pettersen, 2017)

Aunque el diseño de los algoritmos de locomoción y de sistemas de control es complicado (Rezapour, et al., 2016)., sus patrones de locomoción en zonas con obstáculos sobrepasan la movilidad de los robots con ruedas, orugas o extremidades. (Liljebäck et al, 2012)

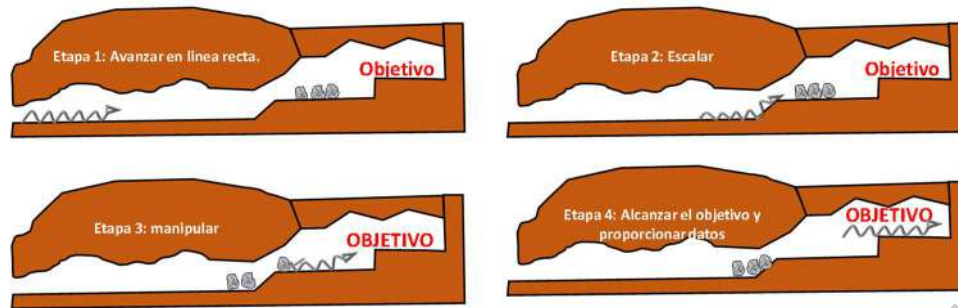


Figura 11. Etapas dentro de una labor de inspección de un robot ápodo.

El poder caracterizar el movimiento de una serpiente mediante ecuaciones matemáticas, también permitió modificar esta ecuación de onda para modelar los diversos patrones de locomoción de una serpiente biológica antes mencionados.

Otras aportaciones que consideramos importantes sobre el avance y la dirección de la línea de investigación sobre los robots ápodos en la actualidad son recabadas en la tabla 1, la cual se aprecia a continuación.

Autor(es)	Aportación.
(Kakogawa & Ma, 2018)	Propuesta de un sistema de locomoción basado en actuadores de transmisión elástica, sincronizado con patrones de locomoción bioinspirados.
(Mukherjee, Kar, & S., 2017)	Simulación de un sistema de locomoción, empleando señales de control con picos programados para que la señal de salida dependa del momento de trabajo y prioridad de demanda.
(Reyes & Ma, 2017)	Estudio de las interacciones del robot tipo Snake, con el entorno, considerando los obstáculos y las fuerzas de fricción generadas, y cómo influyen en la locomoción
(Arizumi & Matsuno, 2017)	Análisis dinámico de un robot tipo snake, considerando como entorno de trabajo un espacio acuático, incluyendo fricción viscosa, dentro de las ecuaciones de Euler-Lagrange.
(Abut, 2015)	Modelado dinámico y propuesta de un Sistema de control P.I.D para un robot tipo Snake.

Tabla 1. Investigaciones previas en robots ápodos.

Estos trabajos, forman parte del diseño, modelado y construcción de estos robots con morfologías bio inspiradas en estos animales, con aplicaciones diversas, como lo son:

Inspección en ductos: La capacidad de un robot ápodo de desplazarse por espacios cerrados, de doblar su morfología en los cambios de dirección de tuberías, y de moverse por un espacio muy limitado, al igual que lo hace una serpiente biológica, le ha convertido en una herramienta en la cual los robots ápodos pueden explorar ductos con la finalidad de localizar fallas o prevención de accidentes.

Tanques industriales: La localización de fisuras en tanques de aceite o de productos corrosivos o flamable es una labor sumamente riesgosa para que un ser humano pueda realizarla, además de que implicaría la implementación de cámaras móviles, que tuvieran la capacidad de realizar la actividad sin desgastarse en el entorno, por lo cual se han desarrollado robots ápodos que siguen patrones de locomoción similares a las serpientes marinas, que puedan realizar la inspección. Mediante el mismo principio, se han empleado robots ápodos para uso en la industria petrolera o exploración submarina. (Kelasidi, et al, 2015).

Búsqueda y rescate: La capacidad de desplazarse por terrenos donde el espacio es reducido y es difícil el acceso para el ser humano, lo han convertido en una potencial herramienta para labores de inspección en zonas de desastre, como inundaciones o terremotos, con la finalidad de localizar personas atrapadas en escombros de edificios colapsados. (Rollinson, 2014).

Otras aplicaciones identificadas en las líneas de investigación vigentes son:
Aplicaciones militares, espionaje, vigilancia de zonas restringidas.

Los robots ápodos, poseen propiedades interesantes, ya que el hecho de no contar con extremidades, no presenta impedimento para que estas se desplacen por cualquier terreno, así sea irregular o reducido. Por lo cual, nos permiten proponerlos

como solución al problema de cómo moverse a través de espacios sumamente reducidos, y con superficies de desplazamiento irregulares.

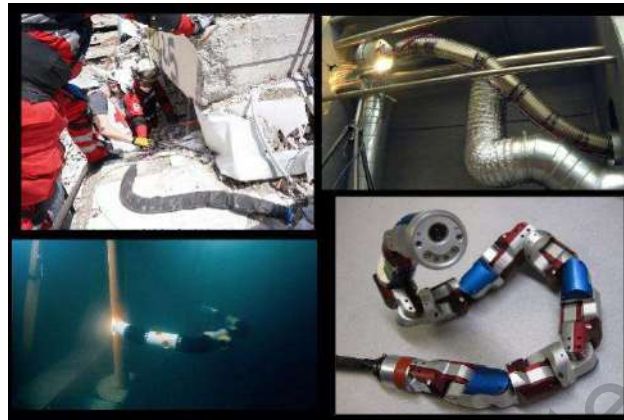


Figura 12. Aplicaciones de robots ápodos, de izquierda a derecha, arriba abajo: robot serpiente trabajando en labores de rescate, robot inspector de ductos, robot snake de inspección subacuática, robot de desplazamiento en 3 dimensiones

En contraparte, los robots serpiente también poseen un número de desafíos que limitan su uso en la práctica, estos incluyen la necesidad de coordinar un largo número de grados de libertad, decrementando la rentabilidad por la naturaleza del diseño, y la compleja interacción de la forma de robot con el espacio de trabajo lo cual, además, trae por ende un complicado diseño de algoritmos de locomoción y de sistemas de control.

Otra dificultad, es el alto consumo de energía que suelen presentar en trayectorias largas, lo cual ha dado origen a plantear problemas de optimización donde las variables objeto de estudio suelen ser el torque, la potencia y el consumo de energía. (Kelasidi & Pettersen, 2018).

El diseño de las trayectorias suele complicarse más, si las superficies son irregulares e inclinadas. como plantean Gong et al. (2014), Para solucionar este problema se han diseñado diversas estrategias y patrones de locomoción, donde se modifican parámetros como la amplitud de onda y el torque producido para romper con las dificultades físicas planteadas (Wang, et al, 2014).

2.5 Clasificación por su morfología.

El problema de diseñar una secuencia de locomoción eficiente de un robot de estas características, radica en la complejidad que representa el caracterizar el movimiento de una serpiente a través de un modelo matemático, que dependerá en gran medida de como esté estructurado el robot, las dos principales estructuras, son las continuas y las discretas.

2.5.1 Modelos discretos.

Las ecuaciones de Hirose, permiten trazar la posición de cada uno de los eslabones a partir de ecuaciones senoidales, a través de un modelo continuo, sin embargo, un robot, es en esencia, un sistema discreto, en el cual, para cada articulación $i=1,2,3,4\dots n$, le corresponde un valor en el espacio, por tanto las ecuaciones de Hirose, suelen utilizarse en su forma discreta, en la cual, la posición de los eslabones forma una aproximación de la onda senoidal continua. Lo cual, ha llevado a dos tipos diferentes de crear la modelar y construir estos robots.

Dentro de las estructuras discretas, podemos hacer énfasis en algunos modelos encontrados en bibliografía, de los que destacamos los siguientes.

(Sandoval, et al., 2018), proponen el siguiente modelo discreto para un robot oruga de múltiples grados de libertad:

$$\alpha(\varphi) = \alpha \cos \left[\varphi + \frac{2\pi K(i-1)}{M} \right] \quad (2)$$

Donde:

$\alpha_i(\varphi)$: Ángulo de la articulación i [Rad]

α : Amplitud, delimita la altura, tomada por el robot [Rad]

φ : Desfasamiento, o ángulo inicial. [Rad]

k: Número de ondulaciones M:
Número de articulaciones

l: Número de articulación, para $i= 1,2,3 \dots M$.

2.5.2 Modelos Continuos

Algunos mecanismos, como el *Elephant Trunk*, y robots tipo 'Snake' con muchas articulaciones, pueden modelarse mediante el envío de señales de tipo sinusoidal, en su forma continua similar a la ecuación 1. Debido a que el espacio entre módulos, con respecto a la longitud del robot, es mínima. (Hannan & Walker., 2000).

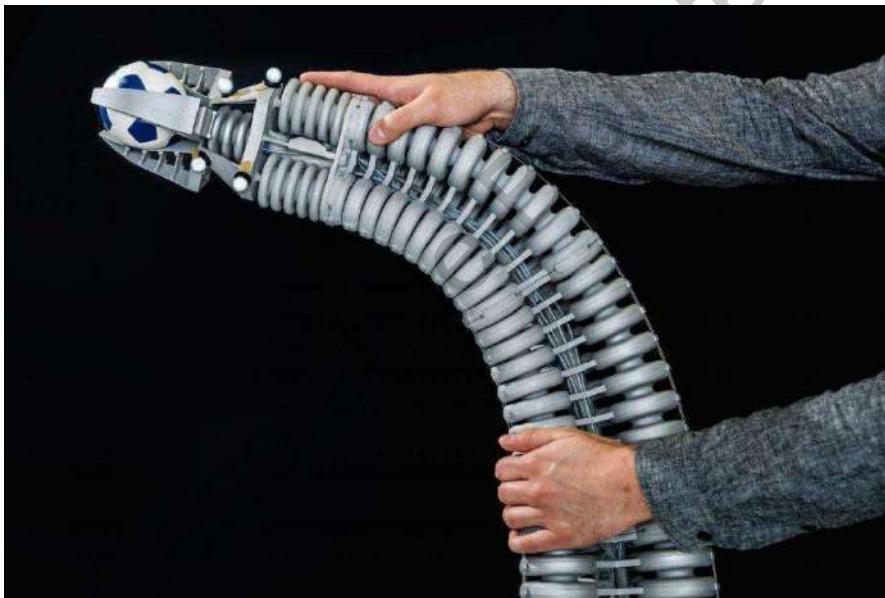


Figura 13. Mecanismo Elephant Trunk,

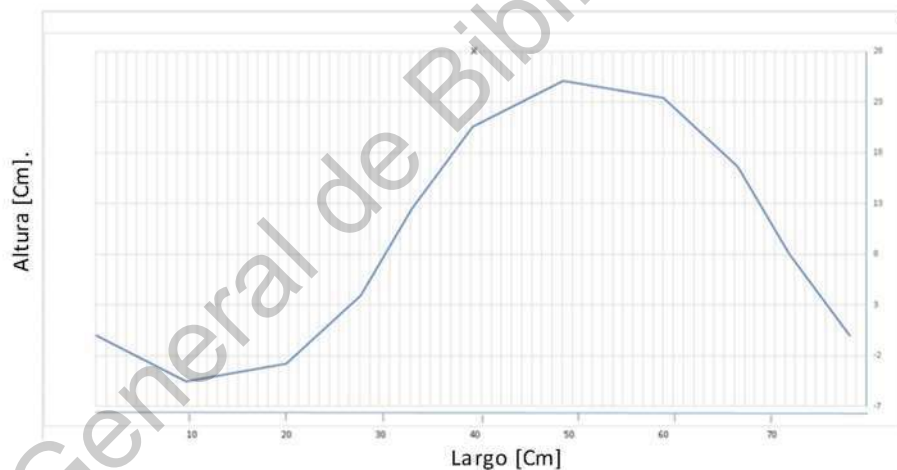
2.5.3 Modelos Mecánicos

No existe una sola forma de diseñar un robot ápodo de forma que se garantice su locomoción, estos se pueden clasificar de acuerdo a su topología.

2.5.3.1 Robots de topología 1D

Son robots ápodos, cuyo algoritmo de locomoción, esta bio-inspirado en el movimiento Serpenoide rectilíneo, por lo cual, su desplazamiento solo puede realizarse en línea recta, hacia atrás y hacia adelante, a través del plano X& Y.

En este caso, solo se necesitará de un generador u oscilador de Hirose para realizar su modelado.



Gráfica 1. Posición de los eslabones de una simulación de robot ápodo unidireccional.

2.5.3.2 Robots de Topología 2D

La trayectoria de este tipo de robots, emplea dos ecuaciones serpentínoides, una, se proyecta sobre el plano x, y mientras que la otra, sobre el plano x, z, generando así una trayectoria en el espacio.

(Garzón, 2011). Propone un modelo estable de locomoción, bajo las siguientes ecuaciones, que nos permiten conocer la componente de las senoidales sobre cada uno de los ejes del espacio.

$$x(s) = \int_0^s \cos[\varphi(\sigma)] d\sigma \quad (3)$$

$$y(s) = \int_0^s \sin[\varphi(\sigma)] d\sigma \quad (4)$$

$$z(s) = \int_0^s \sin[\varphi'(\sigma)] d\sigma \quad (5)$$

$$\varphi(\sigma) = a * \cos[b\sigma] + c\sigma \quad (6)$$

$$\varphi'(\sigma) = a' \sin(2b\sigma) \quad (7)$$

Dónde el escalar a determina la amplitud de onda, b Determina el número de ondulaciones por unidad de longitud y c determina la dirección de la curva, la función $\varphi(\sigma)$ nos permite generar una senoidal lo suficientemente suave, para ser aproximada por puntos.

A partir de esta senoidal, nosotros podemos aproximar dicha función en puntos discretos en el espacio. Físicamente, estos puntos representaran las articulaciones del robot, entonces podemos definir, la ubicación en el espacio de cada articulación, mediante la aproximación numérica dada por la siguiente ecuación.

3 Hipótesis

A través del trabajo de investigación se desea valorar la posibilidad de parametrizar de forma cualitativa y cuantitativa las relaciones de recurrencia y de funciones continuas para mejorar la modelación matemática de la cinemática que presenta un robot de inspección bio-inspirado en serpientes.

4 Objetivos

4.1 Objetivo General.

Desarrollar un modelo matemático que permita valorar la cinemática del desplazamiento de un robot bio-inspirado en serpientes, verificando su efectividad mediante simulación en software.

4.2 Objetivos Específicos.

- Estudiar y entender la mecánica de la locomoción de la serpiente para lograr una abstracción y comprensión matemática necesarias para plantear la cinemática del movimiento.
- Desarrollar modelos cinemáticos de locomoción de un robot móvil bio-inspirado en serpientes con las características propuestas.
- Comprobar la efectividad de los modelos matemáticos desarrollados mediante simulación.

5 Metodología.

El tratamiento del problema está dado bajo el desarrollo de un proyecto mecatrónico, dentro del cual se van superando etapas para llegar al cumplimiento de los objetivos (Vargas, et al., 2014). Dentro de la ejecución de este proyecto, solo está considerada la etapa 1 y 2, modelación cinemática y construcción del modelo respectivamente.

Estas etapas a su vez, las desglosaremos en este proyecto siguiendo el siguiente diagrama de operación.



Figura 14. Metodología empleada dentro de la investigación.

En los capítulos 1 al 2 del presente trabajo desarrollamos los primeros 3 pasos de esta metodología, se describirá las propuestas en 2D y 3D a continuación, así como la construcción de los algoritmos y su validación virtual a través de una simulación de la locomoción. la finalidad de la construcción del modelo en 2D es la comprensión a través de la práctica de la complejidad del problema planteado, lo cual nos permitirá retomar ideas para resolver el modelo 3D, así como delimitar los alcances de la investigación. Puede observarse la estructura del presente trabajo mediante el siguiente diagrama:

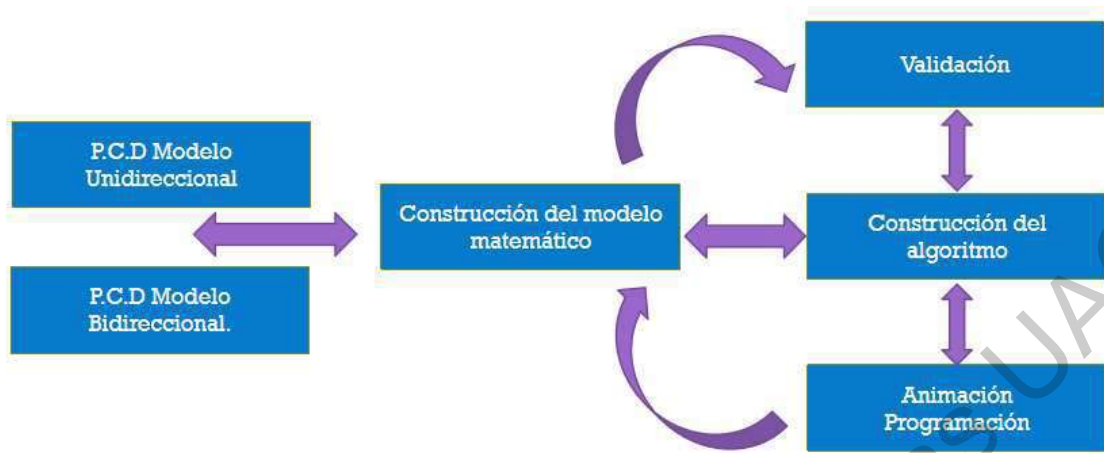


Figura 15. Desarrollo para la construcción de los dos algoritmos de locomoción.

5.1 Definición de Cinemática.

La cinemática estudia el movimiento de los cuerpos, sin involucrar las causas que lo generan, en nuestro caso de estudio, será la metodología empleada para conocer como los eslabones del robot, se mueven con respecto a los otros en el espacio en función del movimiento de los servomotores. A continuación, se dará la expresión del problema cinemático directo y algoritmo de locomoción para definir las variables con la que se desarrollará la parte matemática.

5.1.1 Definición de algoritmo de locomoción y problema cinemático directo.

Definiremos como:

m : el número de articulaciones de la cadena cinemática. $i \in \mathbb{N}$

i , el i -ésimo elemento de m , $i \in \mathbb{N}$

T : total de ciclos de locomoción [S].

t : instante en el cual se encuentra el algoritmo de locomoción [S].

5.1.2 Generadores sinusoidales (Dominio).

Sean:

$\alpha_{i,t}$, una función que genera para cada instante de tiempo t , y para cada elemento i , impar, un vector Q_α

$\gamma_{i,t}$, una función que genera para cada instante de tiempo t , y para cada objeto i par, un vector Q_γ

$$Q_\alpha = \{q_1, q_3 \dots q_{m-1}\}$$

$$Q_\gamma = \{q_2, q_4 \dots q_m\}$$

5.1.3 Posición (Imagen).

Sea el conjunto P , una colección de m vectores, los cuales representan la posición de cada elemento i . es decir.

$$p = \{p_1, p_2 \dots p_m\} \quad (10)$$

$$p_i = \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix} \quad (11)$$

Además, se debe cumplir que:

$$\|\overrightarrow{p_i p_{i+1}}\| = 1 \quad (12)$$

Se define $W \subset P$, como aquella colección de vectores que formen una posición estable del robot. y las funciones F , como aquellas que permitan obtener una imagen en R^3 , a partir de Q . La demostración se presenta en el anexo V

$$F = \{f_1, f_2 \dots f_m\} \quad (13)$$

5.1.4 Problema cinemático directo (P.C.D).

Hallar

$F(Q_\alpha, Q_\gamma) \rightarrow P$, de modo que:

$$f_i \rightarrow p_i \quad (14)$$

5.1.5 Algoritmo de locomoción.

Hallar:

$\alpha_{i,t}, \gamma_{i,t}$ de modo que:

$$F(Q_\alpha, Q_\gamma) \rightarrow W \forall t \in T \quad (15)$$

5.2 Desarrollo del modelo unidireccional.

Podemos describir al modelo unidireccional 2D o 1R, Como un robot el cuál su estructura morfológica consta de una cadena cinemática abierta, con $m-1$ grados de libertad, cuyas revolutas, se encuentran sobre el eje Z. Como se muestra en la figura 1.

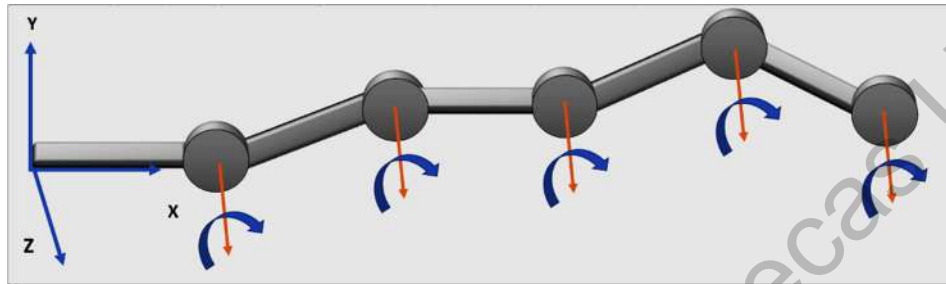


Figura 16. Cadena cinemática con desplazamiento en 2 dimensiones.

Problema Cinemático Directo.	Algoritmo De Locomoción
<p>Hallar</p> <p>$F(Q_\alpha) \rightarrow P$, de modo que:</p> <p>$f_i \rightarrow p_i$</p>	<p>hallar:</p> <p>α de modo que:</p> <p>$F(Q_\alpha) \rightarrow W \forall t \in T$</p>

5.2.1 Problema cinemático directo.

Para plantear la cinemática del robot, se planteó el diagrama de alambre y usando la convención D-H, se buscó la matriz de transformación homogénea para n articulaciones.

Para $M=1$:

$${}^0H_1 = \begin{bmatrix} \cos(q_1) & -\text{sen}(q_1) & 0 & l * \cos(q_1) \\ \text{sen}(q_1) & \cos(q_1) & 0 & l * \text{sen}(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (16)$$

Donde:

q_i : ángulo de la articulación i

l : longitud de los eslabones

Para M=2:

$${}^0H_2 = \begin{bmatrix} \cos(q_1+q_2) & -\text{sen}(q_1+q_2) & 0 & l[\cos(q_1) + \cos(q_1+q_2)] \\ \text{sen}(q_1+q_2) & \cos(q_1+q_2) & 0 & l[\text{sen}(q_1) + \text{sen}(q_1+q_2)] \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

Para M=3:

$${}^0H_3 = \begin{bmatrix} \cos(q_1+q_2+q_3) & -\text{sen}(q_1+q_2+q_3) & 0 & l[\cos(q_1) + \cos(q_1+q_2) + \cos(q_1+q_2+q_3)] \\ \text{sen}(q_1+q_2+q_3) & \cos(q_1+q_2+q_3) & 0 & l[\text{sen}(q_1) + \text{sen}(q_1+q_2) + \text{sen}(q_1+q_2+q_3)] \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

Podemos notar que se forma un patrón o relación de recurrencia, el cual nos permite generalizar el modelo cinemático para M articulaciones:

$${}^0H_M = \begin{bmatrix} \sum_{i=1}^M \cos(q_i) & -\sum_{i=1}^M \text{sen}(q_i) & 0 & l \sum_{i=1}^M \cos\left(\sum_{i=1}^M q_i\right) \\ \sum_{i=1}^M \text{sen}(q_i) & \sum_{i=1}^M \cos(q_i) & 0 & l \sum_{i=1}^M \text{sen}\left(\sum_{i=1}^M q_i\right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

Por tanto, la función que nos indica la posición de cada una de las articulaciones, obteniendo, por relación de recurrencia, la siguiente generalización.

$$P_i: \begin{cases} x_i \\ y_i \end{cases} = l \begin{cases} \sum_{i=1}^m \cos\left(\sum_{i=1}^m \theta_i\right) \\ \sum_{i=1}^m \sin\left(\sum_{i=1}^m \theta_i\right) \end{cases} \quad (20)$$

Donde:

$$\theta_i = \alpha_i - \alpha_{i-1} \quad (21)$$

El cálculo de esa diferencia es necesario ya que θ , referencia los ángulos obtenidos del generador, con el ángulo formado entre una articulación y la siguiente.

5.2.2 Generadores sinusoidales.

En este caso, como tenemos articulaciones con el mismo eje de rotación solo es necesario una ecuación Serpenoide, ya que con una será suficiente para generar las señales que se enviarán a los actuadores para poner en marcha el ciclo de locomoción.

Para este caso, empleamos la curva Serpenoide que emplea (Sandoval, et al., 2018), y que está dada de la siguiente manera:

$$\alpha_{i,t} = \alpha \cos\left(\varphi(t) + \frac{2\pi k(i-1)}{M}\right) \quad (22)$$

Donde:

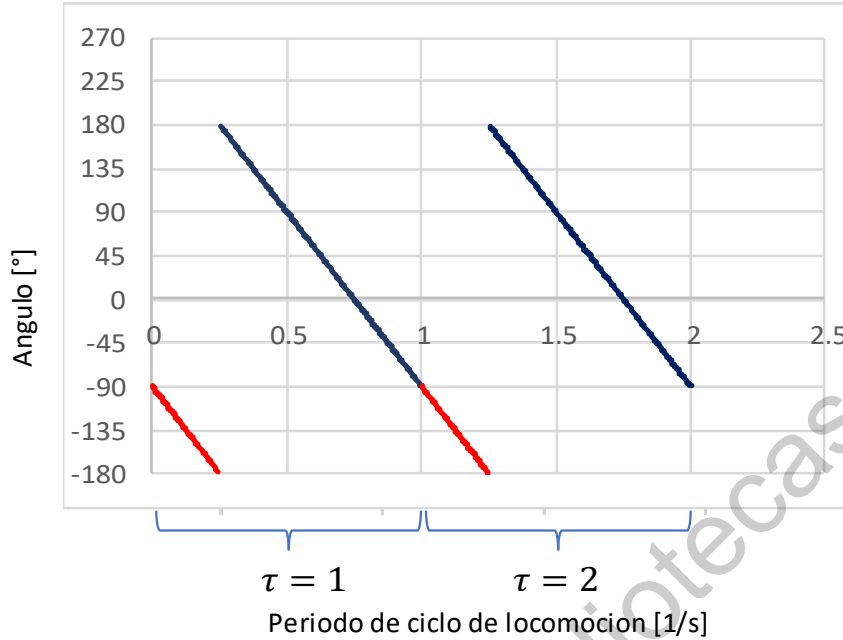
α : Amplitud del generador.

$\varphi(t)$: Secuencia que genera una señal para la i -ésima articulación por unidad de tiempo.

k : Numero de ondulaciones.

5.2.3 Análisis de la función, y propuesta de mejora.

Esta función, genera valores, en cada instante de tiempo t , para cada una de las articulaciones, si esta función fuera constante, no habría variación del ángulo de las articulaciones del robot, con respecto al tiempo. (González, 2008) propone el siguiente rango de trabajo para $\varphi(t)$. El cual se visualiza en la gráfica 2.



Gráfica 2. Rango de trabajo para $\varphi(t)$ proporcionado por la literatura.

Donde:

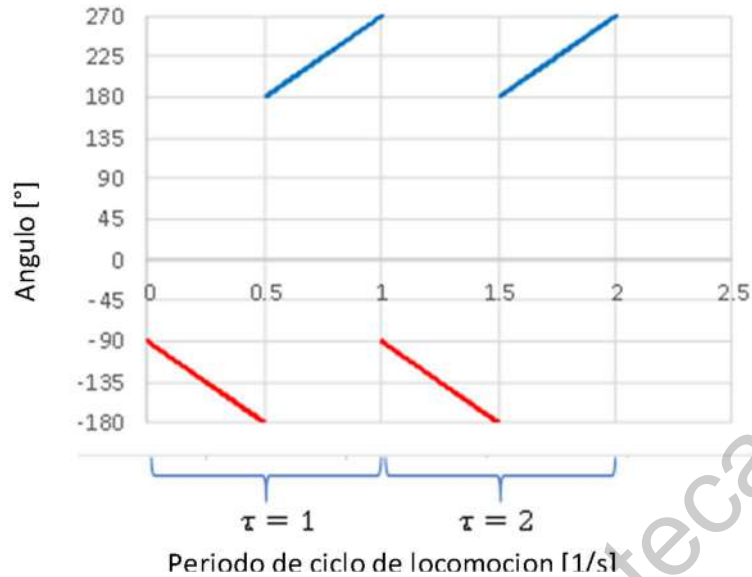
τ : Ciclo de locomoción dentro del cual se ejecuta el algoritmo.

Analizando el comportamiento de la secuencia, se plantea la siguiente función a trazos para describirla.

$$\varphi(t, \tau) = \begin{cases} -2\pi t - \frac{\pi}{2} + (\tau - 1)2\pi; & \tau - 1 \leq t < \tau - \frac{3}{4} \\ -2\pi t + \frac{3\pi}{2} + (\tau - 1)2\pi; & \tau - \frac{3}{4} \leq t < \tau \end{cases} \quad (23)$$

También pudo notarse, que la sección de la secuencia entre $T=0.5$ y 1 , es redundante, ya que ingresando los valores al algoritmo de locomoción nos genera posiciones por las que el robot ya ha transitado. Por tanto, se propone la siguiente secuencia, cuyo comportamiento puede apreciarse en la gráfica 3.

$$\varphi(t, \tau) = \begin{cases} -2\pi t - \frac{\pi}{2} + (\tau - 1)2\pi; & \tau - 1 \leq t < \tau - \frac{1}{2} \\ 2\pi t + \frac{\pi}{2} - (\tau - 1)2\pi; & \tau - \frac{1}{2} \leq t < \tau \end{cases} \quad (24)$$



Gráfica 3. Comportamiento de la secuencia propuesta e implementada.

5.2.4 Programación del algoritmo.

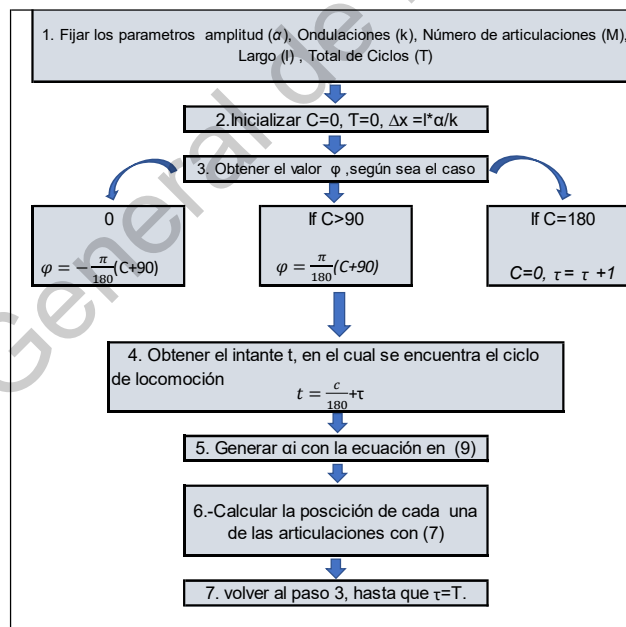


Figura 17. Algoritmo de locomoción para el caso unidireccional.

El algoritmo de locomoción para el robot, tendrá entonces los siguientes pasos a seguir, cuya secuencia se ilustra en la figura 17.

1. Definición de parámetros.

2. Inicialización de los generadores sinusoidales.
3. Etapa de limpieza de la señal y envío de esta al robot para generar una posición en el plano de ejecución.
4. Continuar generando posiciones hasta que el tiempo transcurrido sea igual al necesario para completar un ciclo de locomoción (T).

1. Definir parámetros (amplitud, ciclos totales de locomoción (T), longitud, dirección del robot).

Inicializar $t = 0$.

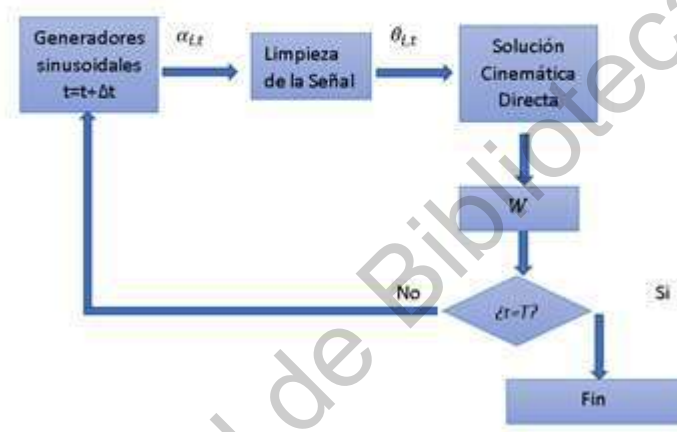


Figura 18. Algoritmo de programación para simulación del robot.

Mediante la construcción de este algoritmo en 2D, se desarrolló el conocimiento de las herramientas matemáticas y de programación necesarias para establecer una serie de pasos específicos que podemos ver en la figura 18, para la consecución del mismo.

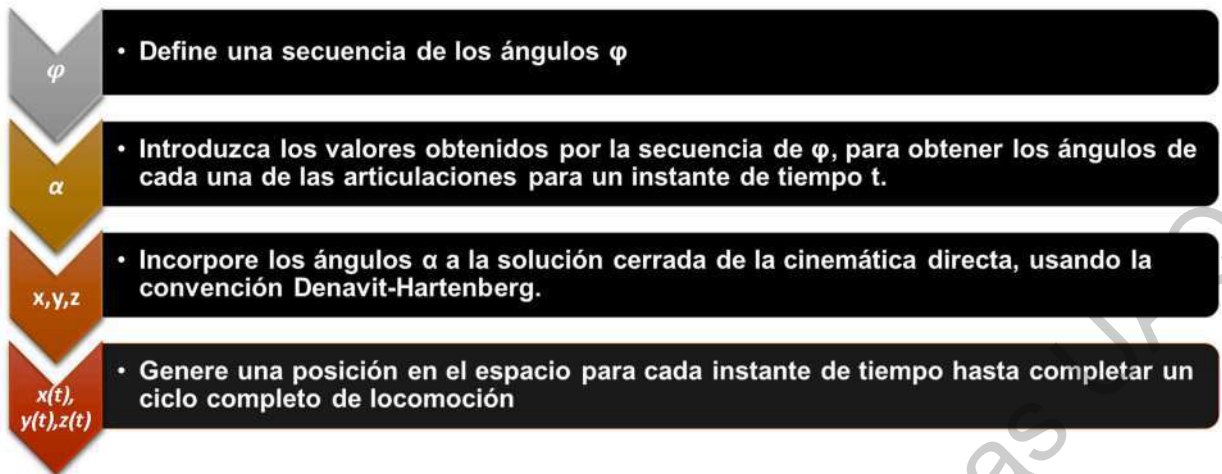


Figura 19. Pasos a seguir para la construcción de un algoritmo de locomoción para un robot tipo Snake.

5.3 Modelo Tridimensional.

Podemos describir al modelo bidireccional 3D o 2R, Como un robot el cuál su estructura morfológica consta de una cadena cinemática abierta, con $m-1$ grados de libertad, cuyas revolutas, se encuentran con eje de giro sobre el eje Y & Z intercaladamente, esto puede apreciarse de mejor manera en la figura 19.

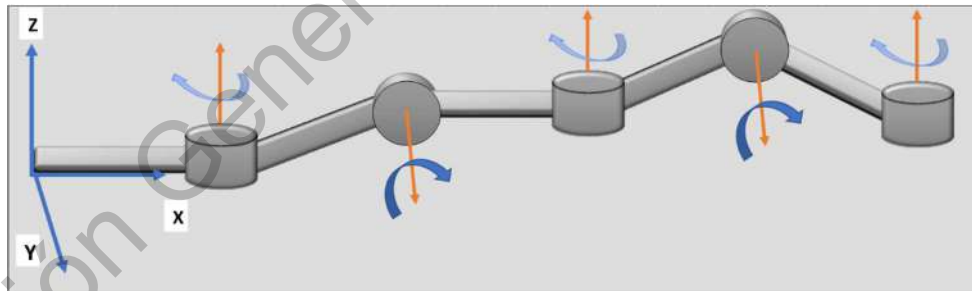


Figura 20. Diagrama de robot tipo Snake 3D.

Problema cinemático directo.	Algoritmo de locomoción
Hallar	hallar:
$F(Q_\alpha) \rightarrow P$, de modo que:	α de modo que:
$f_i \rightarrow p_i$	$F(Q_\alpha) \rightarrow W \forall t \in T$

5.3.1 Problema cinemático directo.

Para resolver el problema cinemático directo, de igual forma se empleó el algoritmo Denavit -Hartenberg, en este caso, al tener las revolutas, un eje de rotación diferente entre sí, a lo largo de toda la cadena cinemática. Quedando las matrices de transformación homogénea bajo la siguiente generalización.

$${}^{n-1}H_n = \begin{pmatrix} \cos[q_n] & 0 & (-1)^{n+1}\sin[q_n] & l * \cos[q_n] \\ \sin[q_n] & 0 & (-1)^{n+1}\cos[q_n] & l * \sin[q_n] \\ 0 & (-1)^{n+1} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (25)$$

Por tanto, para obtener la función posición del enésimo término, debe realizarse el siguiente producto matricial.

$${}^0H_n = \prod_{i=1}^n {}^{i-1}H_i \quad (26)$$

Se utilizó el programa *Wolfram Mathematica* para obtener las expresiones, se muestran a continuación las primeras 4, Se anexa al presente trabajo el desarrollo completo del problema cinemático directo (Anexo 1).

Efectuando las operaciones, obtendremos los siguientes vectores posición de las n articulaciones, haciendo previamente la siguiente convención con:

$$f_i = \cos(q_i)$$

$$g_i = \sin(q_i)$$

Tenemos, para n= 1

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} lf_1 \\ lg_1 \\ 0 \end{bmatrix} \quad (27)$$

para n=2

(28)

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = l \begin{bmatrix} f_1(1 + f_2) \\ g_1(1 + f_2) \\ g_2 \end{bmatrix}$$

para n =3

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = l \begin{bmatrix} f_1(1 + f_2 + f_2f_3) - g_1g_3 \\ g_1(1 + f_2 + f_2f_3) + f_1g_3 \\ g_2(1 + f_3) \end{bmatrix} \quad (29)$$

para n=4

Podemos darnos cuenta que comienzan a formarse patrones con una estructura de sumatoria de productos.

$$\begin{bmatrix} x_4 \\ y_4 \\ z_4 \end{bmatrix} = l \begin{bmatrix} f_1(1 + \sum_{i=2}^4 \prod_{k=2}^i f_k - g_2g_4) - g_1g_3(1 + f_4) \\ g_1(1 + \sum_{i=2}^4 \prod_{k=2}^i f_k - g_2g_4) - f_1g_3(1 + f_4) \\ g_2(1 + f_3 + f_3f_4) + f_2g_4 \end{bmatrix} \quad (30)$$

Y cómo podemos notar, en el caso de n=5, el número de términos para procesar la solución se incrementa.

$$\begin{bmatrix} x_5 \\ y_5 \\ z_5 \end{bmatrix} = l \begin{bmatrix} f_1(1 + \sum_{i=2}^5 \prod_{k=2}^i f_k - f_2g_3g_5 - g_2g_4(1 + f_5)) - g_1(f_3g_5 + g_3(1 + f_4 + f_4f_5)) \\ g_1(1 + \sum_{i=2}^5 \prod_{k=2}^i f_k - f_2g_3g_5 - g_2g_4(1 + f_5)) + f_1(f_3g_5 + g_3(1 + f_4 + f_4f_5)) \\ g_2((1 + \sum_{i=3}^5 \prod_{k=3}^i f_k - g_3g_5) + f_2g_4(1 + f_5)) \end{bmatrix} \quad (31)$$

No se observó que pudiera ser factible la obtención de una relación de recurrencia, tratando este problema como en el caso unidimensional, cuya expresión puede apreciarse en la ecuación 20, por tanto, se decidió realizar un análisis más a fondo.

Si observamos el número de términos en la tabla 2, podemos notar que estos aumentan muy rápido, por lo cual, el procesamiento de los datos podría requerir un excesivo coste computacional. Lo cual nos llevó a plantear la posibilidad de generar un método iterativo para reducir el cómputo.

Número de articulación	Número de elementos del polinomio (P.C.D)		
	X	Y	Z
1	1	1	1
2	2	2	1
3	4	4	2
4	7	7	4
5	12	12	7
6	20	20	12
7	33	33	20
8	54	54	33
9	88	88	54
10	142	142	88
TOTAL	364	364	222

Tabla 2. Número de elementos del polinomio solución del P.C.D

Para trabajar con el procesamiento de todos estos datos, se propone el siguiente procedimiento,

dada la cerradura de la suma en \mathbb{R}^3 , podemos asumir, que:

$$\vec{p}_i + \vec{r}_i = \vec{p}_{i+1} \quad (32)$$

donde \vec{r}_i son las componentes vectoriales que hay que añadir al i-ésimo vector p , para obtener el siguiente.

$$\vec{r}_i = \begin{pmatrix} rx_i \\ ry_i \\ rz_i \end{pmatrix} \quad (33)$$

Empleando la expresión (), también podemos decir que:

(34)

$$\|\vec{r}_i\| = l$$

Y, además, despejando \vec{r}_i

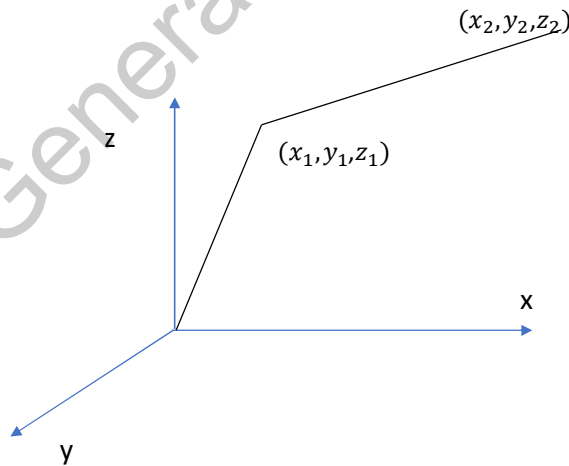
$$\vec{r}_i = \vec{p}_{i+1} - \vec{p}_i$$

Por tanto, podemos obtener la solución del P.C.D, trabajando con las componentes r , y sumándolas consecutivamente, obteniendo la siguiente generalización.

$$p_i = \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix} = \begin{Bmatrix} x_0 + \sum_{i=1}^{M-1} \vec{r}x_i \\ y_0 + \sum_{i=1}^{M-1} \vec{r}y_i \\ z_0 + \sum_{i=1}^{M-1} \vec{r}z_i \end{Bmatrix} \quad (36)$$

Lo anterior puede visualizarse de mejor forma en la gráfica 4.

$$(x_2, y_2, z_2) = (x_1 + RX_1, y_1 + RY_1, z_1 + RZ_1)$$



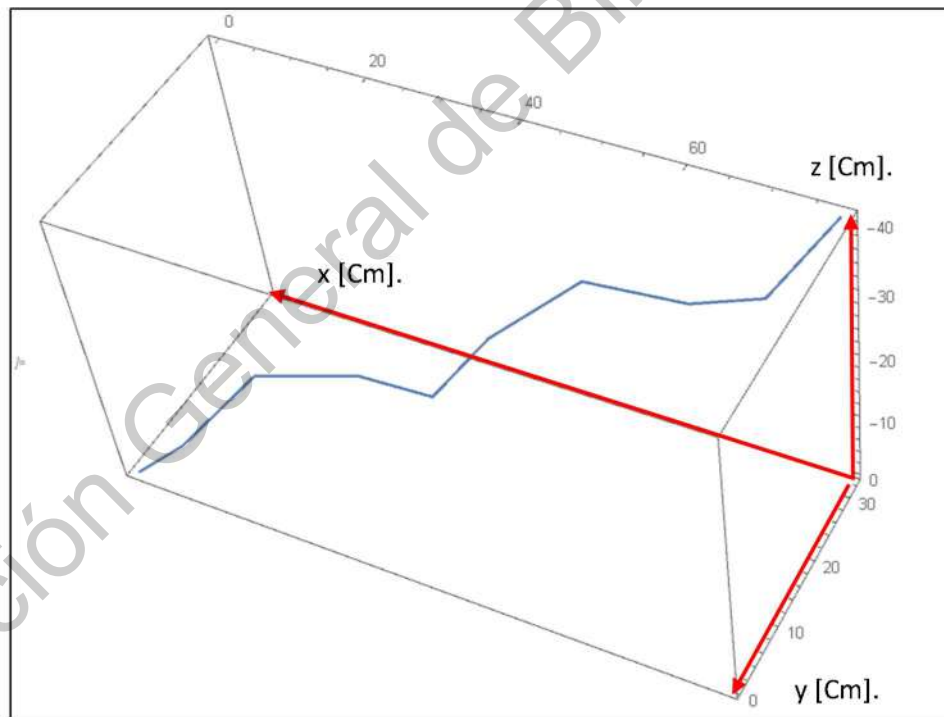
Gráfica 4. Análisis Vectorial del Problema de cinemática directa.

Se comprobó de forma algebraica que las expresiones polinomiales obtenidas realmente representan n puntos en el espacio, preservando al menos la norma en 3 dimensiones, esto se hizo empleando el software *Wolfram Mathematica*. Para llegar a esta conclusión, se puso a prueba que se cumpliera con la ecuación 34. (Anexo 2)

Lo anterior pudo realizarse hasta $n=9$, para $n= 10$, dado el elevado número de términos, el equipo de cómputo no termino de procesarlo después de 2 horas, por lo que se decidió comprobarlo de forma numérica con 50 iteraciones, la identidad expresada en la ecuación 34, se siguió preservando, por lo cual se determinó la expresión como válida.

Enviamos algunas señales con un valor referido al ángulo que debe tomar cada articulación para notar que la estructura generada fuera consistente, dichas expresiones están dadas en la ecuación 37, para las articulaciones pares y nones, respectivamente. La posición obtenida de la cadena cinemática puede observarse en la gráfica 6.

$$\begin{cases} q_{2n} = \frac{(-1)^{n+1}\pi}{4} \\ q_{2n+1} = \frac{(-1)^{n+1}\pi}{4} \end{cases} \quad (37)$$



Gráfica 5. Estructura del robot enviando las señales expresadas en la ecuación 37.

El número de elementos del polinomio para evaluar el P.C.D. quedó distribuido de la siguiente manera:

Número de Número de elementos del polinomio

articulación	(P.C.D)		
	X	Y	Z
1	1	1	1
2	2	2	1
3	3	3	2
4	5	5	3
5	8	8	5
6	13	13	8
7	21	21	13
8	34	34	21
9	55	55	34
Total	143	143	88

Tabla 3. Cantidad de términos del polinomio solución del problema cinemático directo usando residuos.

Observación: La relación entre número de articulaciones y elementos del polinomio del P.C.D, forma la **Sucesión de Fibonacci**.

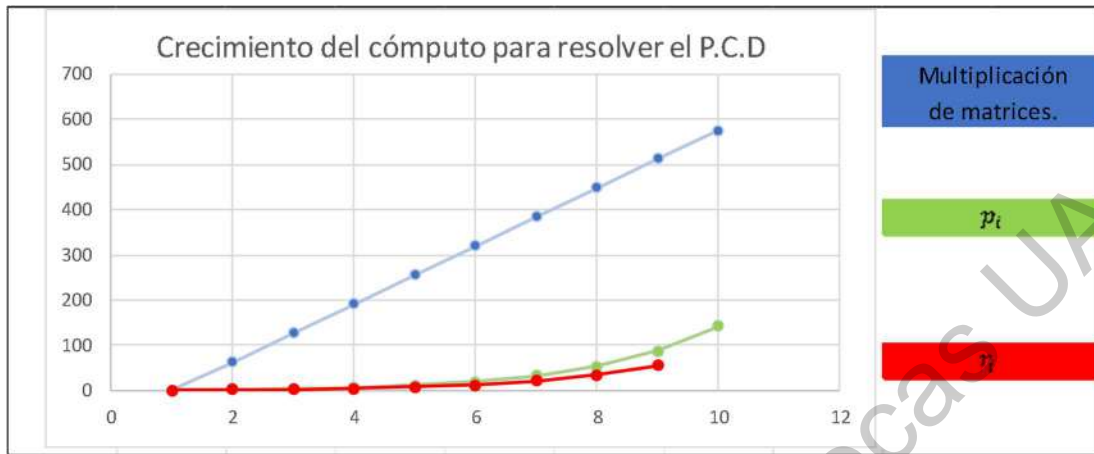
De esta forma, podríamos resolver el P.C.D por tres métodos:

- 1: Multiplicar directamente las matrices para cada instante de tiempo t.
- 2: Evaluar los polinomios obtenidos para cada instante de tiempo t.
- 3: Evaluar los residuos, sumar y obtener el siguiente elemento \vec{p}_i

Si analizamos el cómputo con mayor número de términos, resuelto por cada uno de los métodos obtenemos la siguiente tabla y gráfica:

Numero de articulación	Número de elementos del polinomio (P.C.D)		
	Producto de matrices.	p_i	r_i
1	1	1	1
2	64	2	2
3	128	4	3
4	192	7	5
5	256	12	8
6	320	20	13
7	384	33	21
8	448	54	34
9	512	88	55
10	576	142	
TOTAL	2881	364	143

Tabla 4. Comparativa del cómputo del P.C.D usando los tres métodos.



Gráfica 6. Crecimiento del cómputo por n-ésima articulación del P.C.D.

5.3.2 Generadores sinusoidales.

En el caso del modelo con proyección en 3 dimensiones, como menciona Garzón (2011), se plantean inicialmente dos generadores, uno para las articulaciones con rotación sobre el plano XY, y otro con rotación sobre el plano XZ.

Se comenzó proponiendo añadiendo dos generadores similares a la ecuación 38, a saberse:

$$\begin{cases} \alpha_{i,t} = \alpha \cos\left(\varphi(t) + \frac{2\pi k(i-1)}{M/2}\right) \\ \gamma_{i,t} = \gamma \cos\left(\varphi_2(t) + \frac{2\pi k(i-1)}{M/2}\right) \end{cases} \quad (38)$$

Sin embargo, al implementarse ambos generadores para resolver el P.C.D, se carecía de estructura, transiciones suaves y locomoción sobre dos puntos del robot. Por lo cual se modificó el algoritmo a partir de la observar cómo se mueven las serpientes, y modificando los generadores tomando como inspiración ecuación de calor unidimensional homogénea.

Partimos del hecho de que la amplitud, o análogamente, la medida de la cantidad que el cuerpo de un organismo puede doblarse, no siempre es la misma, tanto para

cualquier instante de tiempo, como para todo el largo de la estructura. Por tanto, los parámetros α, γ , no pueden ser constantes en este caso. Por lo cual la solución que se propone, consiste en hacerlos variables, a fin lograr mayor flexibilidad en la cadena cinemática.

5.3.3 Análisis de la ecuación de calor.

La ecuación de calor para el caso de una varilla unidimensional, nos ofrece un conjunto de soluciones para resolver el problema homogéneo. ⁽³⁹⁾

$$U_{xx} = U_t$$

Donde:

$U(x, 0)$, es la distribución inicial del calor, y:

$$\begin{cases} X(0, t) = f(t) \\ X(l, t) = g(t) \end{cases}$$

Las condiciones de frontera, o como se distribuye el calor a lo largo de la varilla.

Generalmente, $X(x)$, es una expansión en eigenfunciones en términos de la serie de Fourier. y nos dice como se distribuye el calor a lo largo de la varilla. Mientras que $T(t)$, nos dice como se disipa el calor conforme va pasando el tiempo (si es que esta se encuentra en enfriamiento).

Para solucionar este problema de ecuaciones diferenciales parciales, y desechando la idea de que la solución no es $U=0$, (ya que no nos daría ninguna información acerca de la disipación del calor) partiremos de una solución propuesta de la forma:

Disipación/ Acumulación de energía.

$$U = X(x) T(t)$$

Distribución de energía a lo largo de la materia

Figura 21. Solución de la ecuación de calor unidimensional.

Tomemos el siguiente ejemplo:

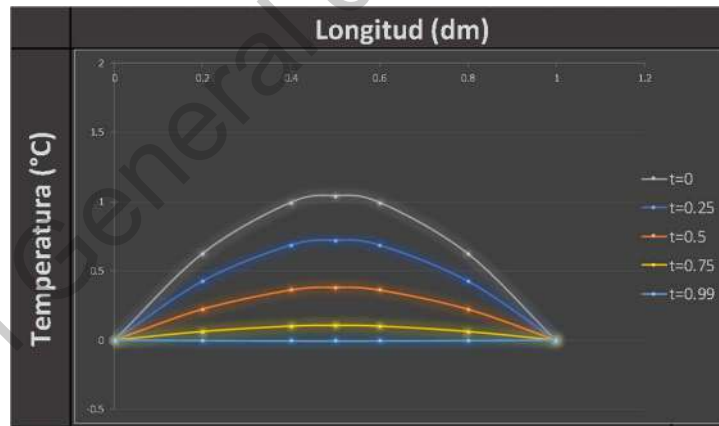
$$U_{x,t} = \sum_{n=0}^{\infty} \frac{8}{[(2n + 1)(\pi)]^3} \text{Sin}(\lambda_n x) * e^{-\pi^2 t} \quad (40)$$

Que satisface las siguientes condiciones de frontera:

Y cuya condición inicial es:

$$\varphi(x) = \text{Sin}(\pi x) \quad (41)$$

Graficando la función U, podemos notar la disipación del calor conforme pasa el tiempo.



Gráfica 7. Distribución del calor en una varilla durante t= 1 segundo.

En esta parte, el lector debe preguntarse, cual es la relación entre el fenómeno de disipación de la energía, y un algoritmo basado en serpientes. Bien, en el problema que estamos resolviendo, se está modelando la distribución de movimiento a lo largo de una distancia, o de una cadena que está oscilando o vibrando con cierta amplitud y frecuencia.

Dado que, en el movimiento de una serpiente, hay amplitud variable (o análogamente, la libertad de doblar su cuerpo) a lo largo del cuerpo del animal, haremos variable la amplitud de la señal de cada una de las articulaciones, tomando como inspiración, la solución de la ecuación de calor.

Ya que la deducción de esta solución parte de:

$$U_{x,t} = X(x)T(t) \quad (42)$$

Y los generadores de Hirose de la ecuación 38. también están en función de la distancia y el tiempo, entonces proponemos sustituir la amplitud constante por:

$$\begin{cases} \alpha_{i,t} = I(i)T(t) \cos\left(\varphi(t) + \frac{2\pi k(i-1)}{M/2}\right) \\ \gamma_{i,t} = I(i)T(t) \cos\left(\varphi(t) + \frac{2\pi k(i-1)}{M/2}\right) \end{cases} \quad (43)$$

De esta forma, la amplitud de la señal será variable a lo largo del tiempo, y se distribuirá de forma variable a lo largo de la estructura para dotar de mayor flexibilidad en los movimientos a la cadena cinemática.

5.3.4 Obtención de las funciones generadoras de amplitud.

Para obtener una función que nos genere una amplitud deseada en términos de dos variables, nos basamos en el movimiento tipo 'concertina', observada en algunas serpientes, el cual puede ser observado en la figura (21).

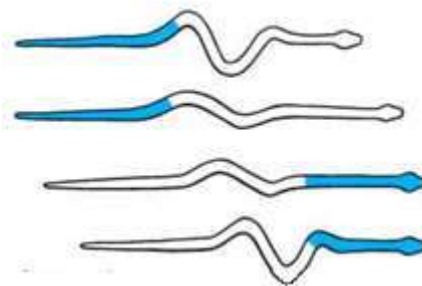


Figura 22. Movimiento de Tipo Concertina. Enciclopedia Británica (2012).

Puede observarse, que la amplitud de la onda que describe el movimiento se concentra en medio del organismo, y se distribuye hacia la izquierda y la derecha, por tanto, nuestra estrategia de movimiento tendrá la siguiente secuencia.

- Si el movimiento es simétrico.

-Contracción por derecha.

- Expansión.

-Contracción por izquierda.

-Expansión.

- Si el movimiento es asimétrico.

Contracción (por únicamente derecha, o izquierda)

Expansión.

Contracción.

Expansión.

El comportamiento de la estrategia de locomoción propuesta puede apreciarse de mejor manera en la gráfica 8.



Gráfica 8. Vista Superior de la expansión y contracción del robot en 3 dimensiones.

Para pasar el movimiento deseado a un modelo matemático que lo represente, asociaremos el ciclo de trabajo a distintas etapas del proceso de locomoción en que la amplitud alcanza valores críticos.

Instancia	Tiempo asociado del ciclo [S].
Inicio (partiendo del reposo)	t=0
Inicio de expansión	t=0.25
Expansión (amplitud máxima)	t=0.5
Inicio de contracción	t=0.75
Posición inicial (inicio del segundo ciclo de trabajo).	t=1

Tabla 5. Instantes y valores críticos.

Lo cual nos lleva a plantear las siguientes restricciones.

$$\begin{aligned}
 \max [\alpha_{i,t}] &= \begin{cases} \alpha_{5,0.25t} \\ \alpha_{5,0.75t} \end{cases} & \min [\alpha_{i,t}] &= \begin{cases} \alpha_{5,0t} \\ \alpha_{5,0.5t} \end{cases} \\
 \max [\gamma_{i,t}] &= \begin{cases} \gamma_{4,0.25t} \\ \gamma_{6,0.75t} \end{cases} & \min [\gamma_{i,t}] &= \begin{cases} \gamma_{4,0t} \\ \gamma_{6,0.5t} \end{cases}
 \end{aligned} \tag{44}$$

Además, dado que el serpenteo se aprecia más desde la vista superior que de la horizontal, se debe cumplir que:

$$\alpha_{i,t} > \gamma_{i,t}$$

Con esto garantizamos que:

El doblaje del cuerpo del robot se concentre en medio del cuerpo, y que esto ocurra en los instantes de tiempo asociados a la instancia crítica, regresando a una posición inicial y evitando transiciones bruscas si se busca una rutina simétrica, ya que habrá que pasar forzosamente por un punto de amplitud cero.

Podemos percatarnos que el comportamiento de la amplitud de onda sigue un patrón muy similar al de la solución de la ecuación de calor y cuyo comportamiento puede apreciarse en la gráfica 8.

Por lo cual, el siguiente paso, es obtener una expresión que distribuya el movimiento de forma que este se concentre en medio y hacia los lados durante cada ciclo de locomoción, el planteamiento se realiza de la siguiente manera:

Si i , es par:

$$I(i) | \max[I] = f(5), \quad (46)$$

$$\text{además, } \min[I] = f(1) = f(M - 1)$$

Si i , es impar:

$$I(i) | \max[I] = f\left(\frac{M}{2} + 1\right), \quad (47)$$

$$\text{además, } \min[I] = f(2) = f(M)$$

Ya que la tasa o razón de cambio de la exponencial de la ecuación 40, que describe la disipación de la energía en un periodo de cambio, suele ser elevada, optaremos por una con transición más suave, por lo cual se proponen las siguientes funciones que además cumplan con las restricciones propuestas en las ecuaciones (46) y 47.

$$I(i) = 2^{-\left|\frac{i-m}{m-1}\right|}, \text{ si } i \text{ es impar.}$$

$$I(i) = 2^{-\left|\frac{i-(m+1)}{m}\right|}, \text{ si } i \text{ es par.}$$

Donde : $m = M/2$.

Podemos observar, como sería la distribución de la amplitud, a través de cada una de las i -ésima articulaciones en la gráfica 9.

En caso de $T(t)$, si se desea un movimiento simétrico, en el cual, la señal se distribuya en la segunda mitad del ciclo de locomoción a igual magnitud, pero sentido contrario, se propone:

$$T(t) = K_n u(t) \sin(2\pi t) \quad (49)$$

Donde:

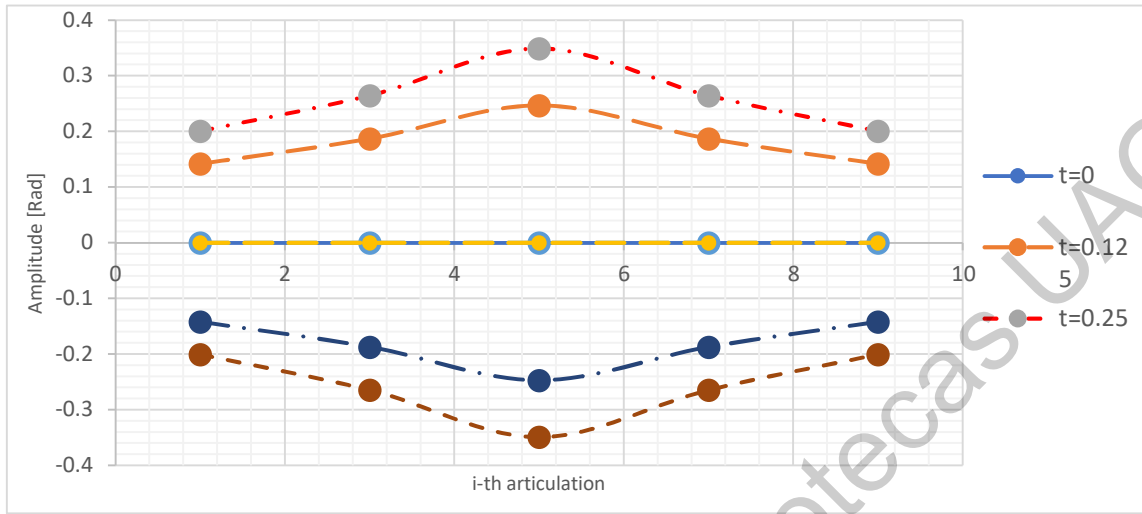
K_n = constantes de amplificación de la señal.

$u(t)$: función escalón unitario, el cual está dado de la siguiente forma.

$$u(t) = \begin{cases} 1, & t < 0.5 \\ -1, & t \geq 0.5 \end{cases} \quad (50)$$

En cambio, si se desea un movimiento asimétrico, simplemente:

$$T(t) = K_n \text{Sin}(2\pi t)$$



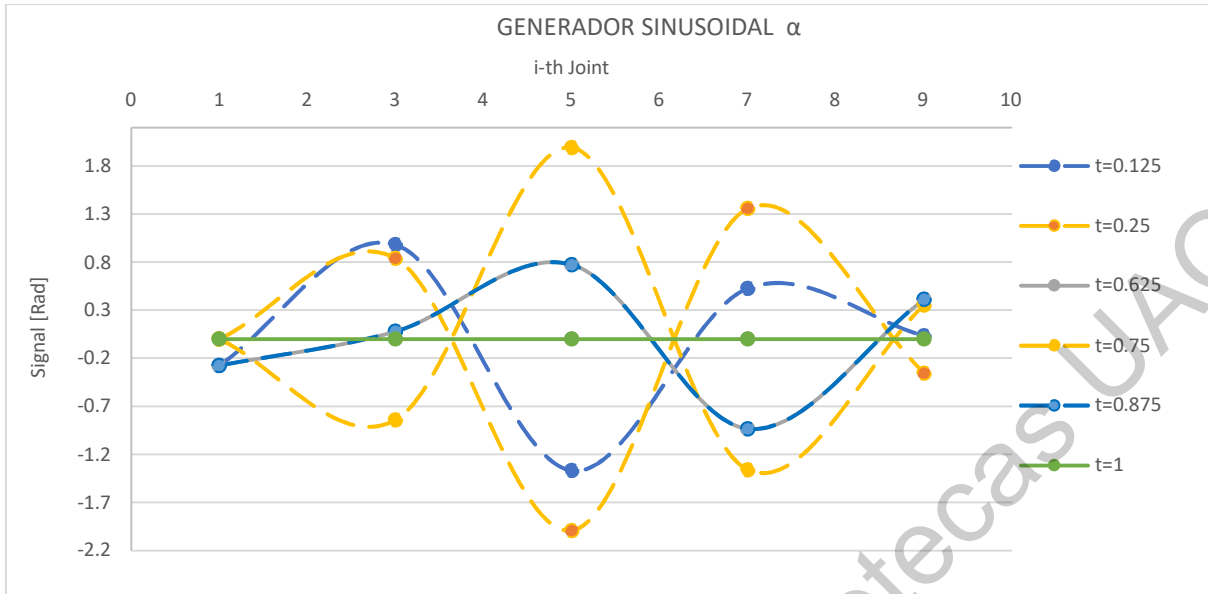
Gráfica 9. Comportamiento de la función amplitud con escalón unitario.

Por tanto, nuestros generadores sinusoidales quedarían de la siguiente manera.

$$\begin{cases} \overline{\alpha_{i,t}} = I(i)T(t)\alpha_{i,t} + \delta \\ \overline{\gamma_{i,t}} = I(i)T(t)\gamma_{i,t} + \hat{\delta} \end{cases} \quad (52)$$

Si, además, se desea dar dirección de desplazamiento al robot, este puede obtenerse añadiendo un ángulo constante δ , según sea el caso.

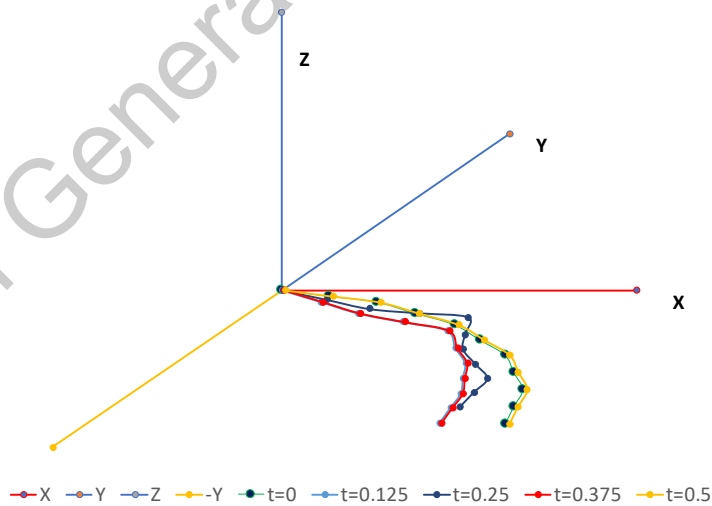
El comportamiento del generador sinusoidal puede observarse en la gráfica 10. Mientras que las funciones I y T , controlan la amplitud de la señal, el generador sinusoidal se encarga de distribuirla a lo largo de la cadena cinemática.



Gráfica 10. Implementación de la función amplitud al generador Serpenoide.

Alcanzado este paso, se procedió a implementar el algoritmo de locomoción empleando la ecuación 52. Obteniéndose lo siguiente.

IMPLEMENTACIÓN DEL ALGORITMO EN EL P.C.D



Gráfica 11. Implementación de los generadores al robot en 3D

5.3.5 Limpieza de la señal.

Podemos observar que no hay una transición recurrente, o que imite cualquier comportamiento o patrón de movimiento, esto es causado debido a que las proyecciones generadas sobre el eje Y, siempre están siempre en adición una tras otra, por lo cual, es necesario procesar la señal, para obtener las transiciones que nos generen el patrón de movimiento deseado. Por lo cual, en este algoritmo, la señal debe pasar forzosamente por una etapa de depuración en la cual se elimine la interacción de la señal de la n-ésima articulación, con la articulación n+1. Es decir, hallar una expresión que cumpla con la ecuación 15.

Lo cual implica encontrar un símil para la ecuación 21, en el caso unidimensional. No se pudo encontrar una solución por inducción para resolver el problema (Consultar Anexo 3). En vista de este problema, se optó por un tratamiento numérico de los datos que al final nos permitió obtener la expresión matemática deseada.

Este método consistió en restar de la señal inicial, la proyección generada sobre los ejes Y, Z, según sea el caso, con la finalidad de depurar la señal. Dicho procesamiento de datos debe efectuarse durante cada iteración dentro del ciclo de trabajo.

En principio, se genera una señal inicial, la cual es enviada a la estructura geométrica del robot, análogo a una “planta”, y posteriormente se determina el “error”, para producir una nueva señal que entra al sistema, por lo cual puede observarse cierta similitud con un sistema de control a lazo cerrado. Lo cual puede apreciarse en la figura 22.

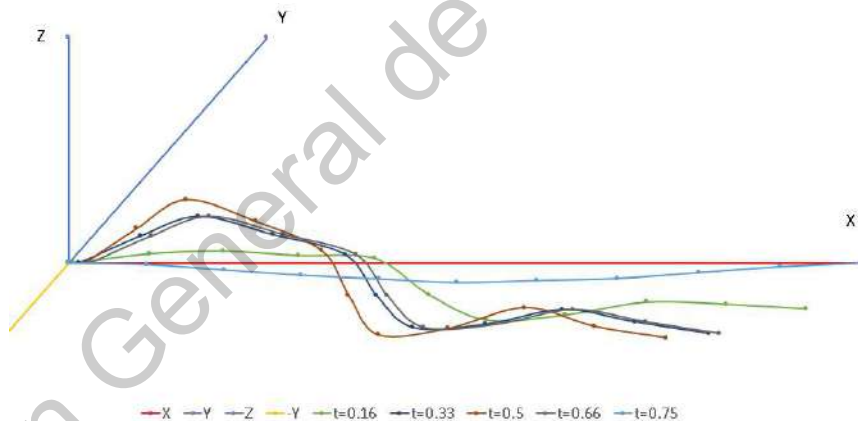
La ecuación 54 muestra el procesamiento de los datos dentro de nuestro método numérico. $\overline{\alpha}_{i,t}, \overline{\gamma}_{i,t}$ son las señales producidas por los generadores sinusoidales en la etapa 1. Mientras que $\overline{\alpha}_{i,t}$ y $\overline{\gamma}_{i,t}$, representan las señales ya depuradas en la etapa 2.

$$\begin{cases} \overline{\alpha}_{i,t} = \overline{\alpha}_{i,t} - \theta_{i-1,t} \\ \overline{\gamma}_{i,t} = \overline{\gamma}_{i,t} - \hat{\theta}_{i-1,t} \end{cases} \quad (53)$$

En la cual, los ángulos de las proyecciones están dados por:

$$\begin{cases} \theta_{i-1,t} = \tan^{-1} \left(\frac{P_{p_i,y}}{P_{p_i,x}} \right) \\ \hat{\theta}_{i-1,t} = \tan^{-1} \left(\frac{P_{p_i,z}}{P_{p_i,x}} \right) \end{cases} \quad (54)$$

Estas dos últimas expresiones fueron obtenidas por inducción, y son equivalentes a las obtenidas por (Transeth, 2007). Donde $P_{p_i,x}$, $P_{p_i,y}$ y $P_{p_i,z}$ son las proyecciones del vector p_i sobre los ejes X, Y, Z, del sistema coordenado absoluto, respectivamente. La principal ventaja de este algoritmo es no tener que buscar un generador sinusoidal que al aplicarse obtenga una imagen en W . Sino que, mediante el cálculo de la diferencia, puede modificarse. Mientras que la desventaja principal es el doble cómputo en la solución del P.C.D, lo que eleva el coste computacional del procesamiento de datos. La limpieza de la señal puede observarse en la figura (13).



Gráfica 12. Movimiento del robot al procesar la señal corregida

De esta forma, el algoritmo de locomoción con imagen en 3 dimensiones tiene con diferencia al caso unidireccional, una segunda etapa de procesamiento de la señal del generador.

5.3.6 Modelo final.

Las ecuaciones que producen el movimiento, quedan entonces de la siguiente manera:

$$\left\{ \begin{array}{l} \overline{\alpha}_{i,t} = \alpha \cdot u(t) \sin[2\pi t] \cdot 2^{\left|\frac{2i-M}{M}\right|} \alpha_{i,t} + \delta \\ \overline{\gamma}_{i,t} = \gamma \cdot \sin[2\pi t] \cdot 2^{\left|\frac{2i-M}{M}\right|} \gamma_{i,t} + \hat{\delta} \end{array} \right. \quad (55)$$

5.3.7 Algoritmo de locomoción.

El algoritmo de locomoción para el robot en 3 dimensiones, tendrá entonces los siguientes pasos a seguir:

1. Definición de parámetros.
2. Inicialización de los 2 generadores sinusoidales.
3. Etapa 1: Procesamiento de la señal, sin salida hacia la estructura (robot), sino generación de una posición "virtual".
4. Etapa 2: Obtención de los ángulos empleando las ecuaciones () y (), para recuperar la señal inicialmente enviada.
5. Obtención de una señal con salida hacia la estructura (robot) para generar una posición en el espacio.
6. Continuar generando posiciones hasta que el tiempo transcurrido sea igual al necesario para completar un ciclo de locomoción (T).

La estructura del algoritmo de locomoción, puede visualizarse en el diagrama presentado en la figura 22.

1. Definir parámetros (amplitud, ciclos totales de locomoción (T), longitud, dirección del robot).

tiempo total $T=0$

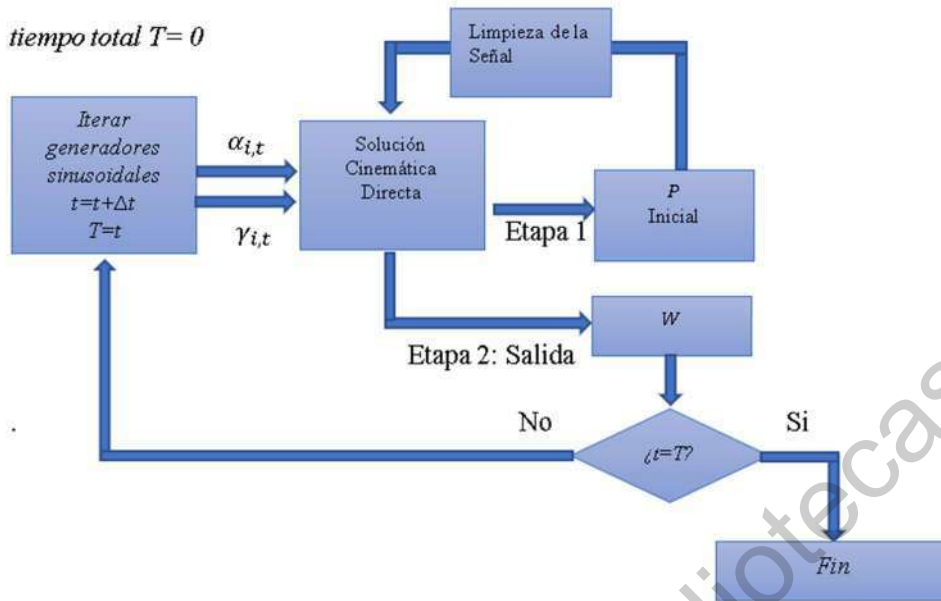


Figura 23.. Diagrama de flujo de algoritmo de locomoción en 3 dimensiones para un robot Snake.

A pesar que se obtiene un solo algoritmo de locomoción, esto no impide que puedan crearse distintos patrones de movimiento, lo cual se logra manipulando nuestros parámetros en las funciones generadoras de señal. Se crearon dos principales tipos de locomoción. a decirse, movimiento concertina con movimiento en 3 dimensiones, simétrico y asimétrico así como movimiento lateral o *Sidewinding* simétrico y asimétrico.

5.3.8 Rutinas de viraje.

Si se desea cambiar de dirección durante la ejecución de una rutina de locomoción, entonces nuestro ángulo de desfase cambiaría de ser constante, a ser variable en el tiempo, se propone, para esto, una función de tipo.

$$\delta(t) = at + \delta_0 \quad (56)$$

Donde:

a: Tasa de cambio del ángulo de viraje. [Rad/s]

t: Tiempo del ciclo local [S].

δ_0 : Valor Inicial, o ángulo previo antes del viraje [Rad/s].

En este punto, se vuelve una propiedad interesante el hecho de que este algoritmo de locomoción pase por un instante de tiempo t , en que la amplitud es cero (inicio y final), ya que, si se desea cambiar de estrategia de locomoción, puede implementarse una, en cuanto concluye otra, sin tener que elaborar una secuencia de transición. Este punto se ilustra en la figura 23

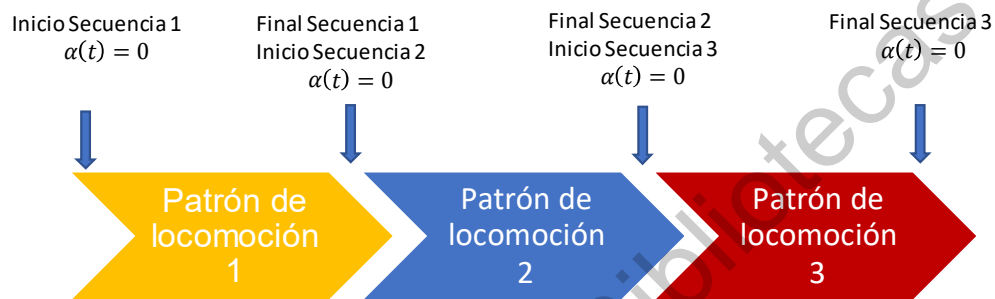


Figura 24. Cambio de dirección aprovechando los instantes de amplitud cero.

5.3.9 Evasión de obstáculos.

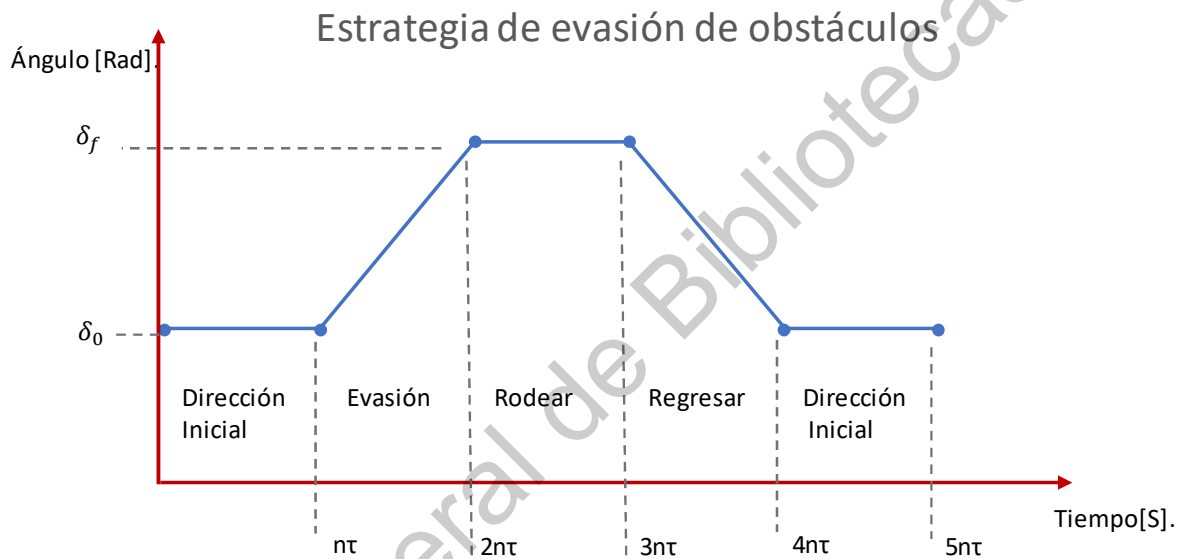
Para ilustrar el punto anterior, se presenta una situación útil, en la cual, es deseable implementar una estrategia de locomoción, con el objetivo de evadir un obstáculo, haciendo uso de diversos patrones de locomoción obtenidos.

Se hace la suposición, que, por métodos de lectura del entorno, no pertenecientes al estudio cinemático como sensado, visión, sistemas de control, se cuentan con los siguientes datos.

- Un ángulo de viraje inicial δ_0 [Rad].
- Un ángulo de viraje final para evadir el obstáculo δ_f [Rad].
- Un periodo de tiempo para virar, estabilizar y regresar a la dirección original $n\tau$ [S].

Dividimos el procedimiento para evadir el obstáculo en los siguientes, haciendo la suposición que el tiempo para cada uno es constante (Gráfica 13).

- 1: El robot se desplaza en una dirección, con un ángulo inicial δ_0 .
- 2: Se detecta un obstáculo, para lo cual, se inicia una secuencia de viraje, empleando los valores de la ecuación 57. Siendo de utilidad la propiedad de los generadores de locomoción expresada en la figura 24.
- 3: El robot se estabiliza, retomando la dirección inicial y rodeando al objeto,
- 4: Comienza la rutina de viraje para volver a la ruta inicial.
- 5: El Robot completa la evasión del obstáculo, retomando la dirección y el ángulo de trabajo.



Gráfica 13. Estrategia de evasión de obstáculos.

6 Resultados y discusión.

El algoritmo de locomoción se programó en Visual Basic, empleando una hoja de cálculo de Excel para visualizar la animación, para evaluarlo, fue empleado un procesador Intel Celeron 1.60 GHZ y 4.00 GB de memoria RAM. En el caso unidimensional, también se programó en C++- El simulador consiste en ejecutar un contador que da un valor $t+\Delta t$, generando una posición del robot, y actualizando la imagen en el espacio, puede apreciarse la locomoción del robot. Procederemos mostrando los resultados obtenidos en el modelo unidireccional para luego presentar los obtenidos en el modelo bidireccional. Se presentan los parámetros de ajuste de los generadores.

6.1 Modelo unidireccional.

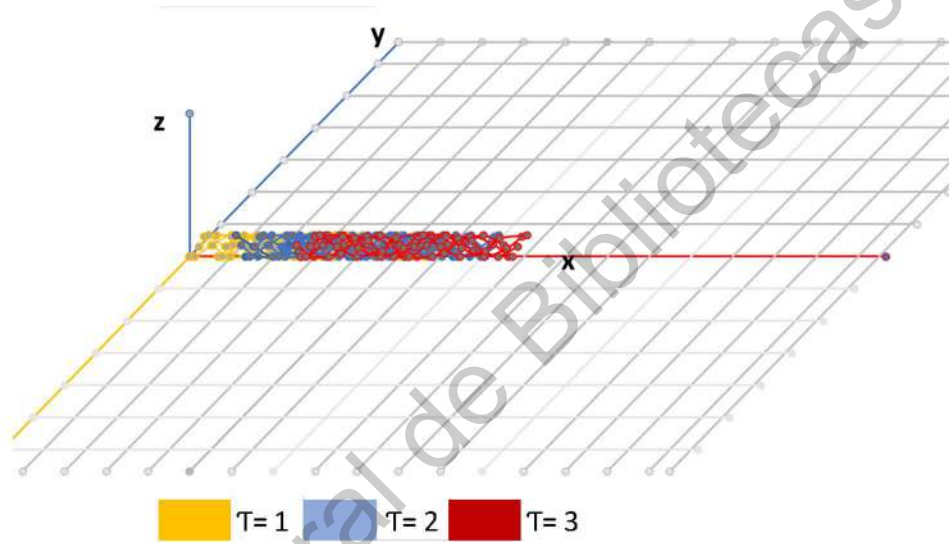
Animación 1. Descripción: "Movimiento de tipo línea recta, simétrico, con algoritmo 1D, en el plano X, Y Con 0 grados de dirección

<i>Parámetros del generador.</i>	Valor	Unidades
α	$\pi/3$	[Rad]
γ	No aplica	[Rad]
M	10	Adimensional.
δ	No aplica	[Rad]
$\hat{\delta}$	0	[Rad]
τ	3	Adimensional

Tabla 6. Parámetros implementados en la animación 1.

<i>Funciones del generador.</i>	<i>Expresión</i>
$u(t)$	1
$\alpha_{i,t}$	$\sin[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	No aplica

Tabla 7. Función generadora de la animación 1.



Gráfica 14. Evolución del algoritmo de locomoción de la animación 1.

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	60	0		
Altura Máxima Alcanzada			8.3927	
Tiempo de ejecución del algoritmo.				6.9847

Tabla 8. Mediciones obtenidas de la animación 2.

Animación 2: Descripción: “Movimiento de tipo línea recta, simétrico, con algoritmo 1D, en el plano X, Y Con 18° grados de dirección.

Parámetros del generador.	Valor	Unidades
α	$2\pi/5$	[Rad]
γ	No aplica	[Rad]
M	10	Adimensional.
δ	No aplica	[Rad]
$\hat{\delta}$	18°	[Rad]
τ	3	Adimensional

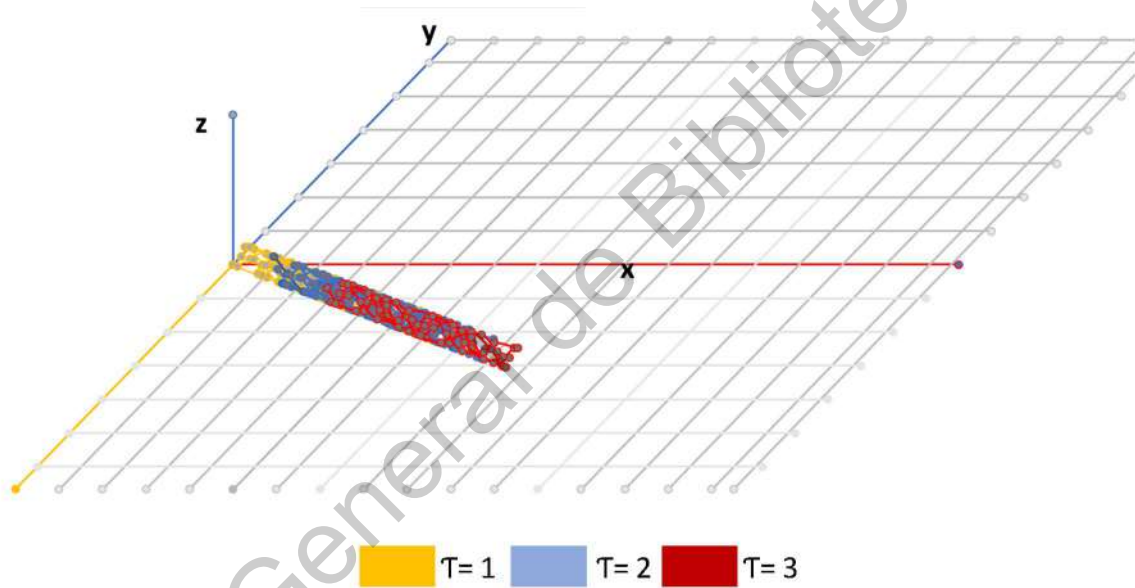
Tabla 9. Parámetros implementados en la animación 2.

Funciones del generador.	Expresión
$u(t)$	1
$\alpha_{i,t}$	$\sin[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	No aplica

Tabla 10. Funciones enviadas para generar la animación 2.

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	57.06339	18.54101		
Altura Máxima Alcanzada			16.03542	
Tiempo de ejecución del algoritmo.				7.36778

Tabla 11. Mediciones obtenidas de la animación 2.



Gráfica 15. Visualización del desplazamiento de la animación 2.

6.2 Modelo bidireccional.

Se probaron variantes del algoritmo, enviando distintos tipos de señales, simétricas y asimétricas, o enviando senos y cosenos,

obteniéndose diversos patrones de locomoción. El movimiento obtenido por los generadores durante un ciclo de locomoción puede visualizarse en la figura ().

Se presentan a continuación las rutinas de movimiento creadas, además de los parámetros fijados, las funciones generadoras de movimiento establecidas para cada rutina y el tipo de locomoción obtenido. Todas fueron para programar al menos 20 ciclos de locomoción, aunque solo se presentan 3 ciclos en las gráficas debido a que se superpone en cantidad la evolución del movimiento del mecanismo. Se registra el tiempo y el avance en con respecto a los ejes X & Y.

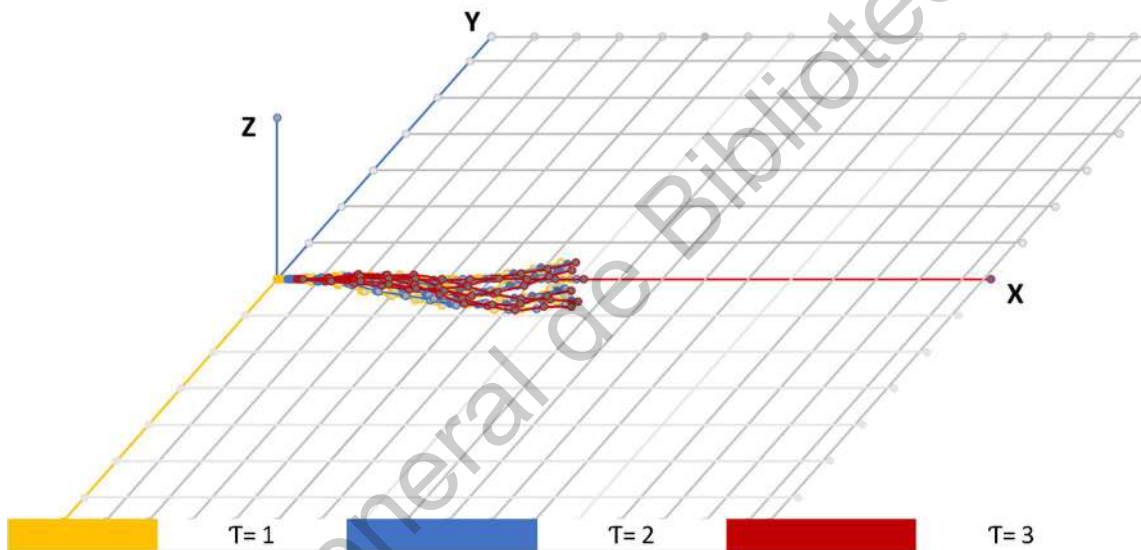
Animación 3. Descripción: “Movimiento de tipo ondulatorio, asimétrico, con señal sinusoidal como generador, con línea recta como trayectoria a seguir “

Funciones del generador.	Expresión
$u(t)$	-1 si $t < 0.5$ 1 si $t \geq 0.5$
$\alpha_{i,t}$	$\sin\left[\varphi(t) + \frac{2\pi k}{M}(i-1)\right]$
$\gamma_{i,t}$	$\sin\left[\varphi(t) + \frac{2\pi k}{M}(i-1)\right]$

Tabla 12. Señales enviadas al generador para la animación 3.

Parámetros del generador.	Valor	Unidades
α	$\pi/12$	[Rad]
γ	$\pi/36$	[Rad]
M	10	Adimensional.
δ	0	[Rad]
$\hat{\delta}$	0	[Rad]
τ	3	Adimensional

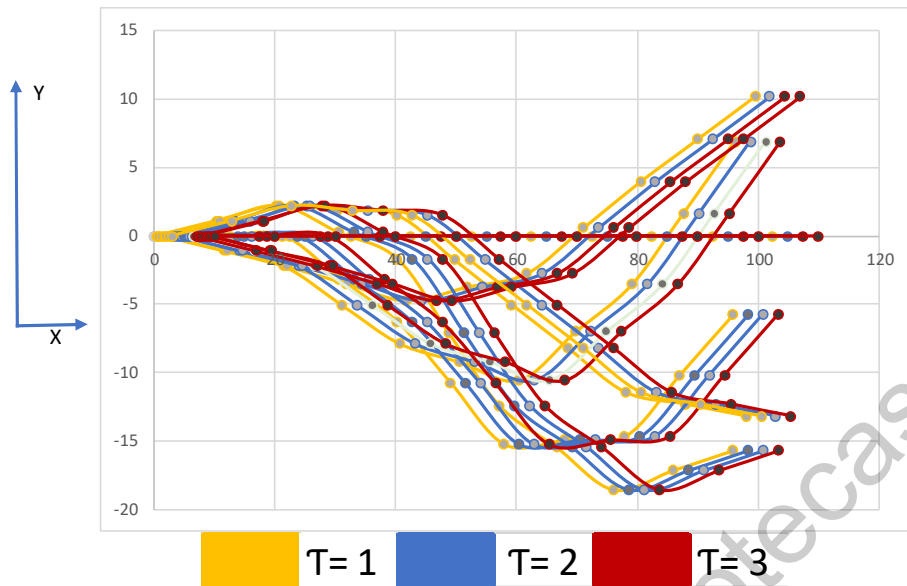
Tabla 13. Ajuste de parámetros para la animación 3.



Gráfica 16. Visualización de la animación 3

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	15	0		
Altura Máxima Alcanzada			1.93	
Tiempo de ejecución del algoritmo.				18.2164

Tabla 14. Resultados obtenidos de la animación 3



Gráfica 17. Vista superior de la animación 3.

Dirección General de Bibliotecas UAQ

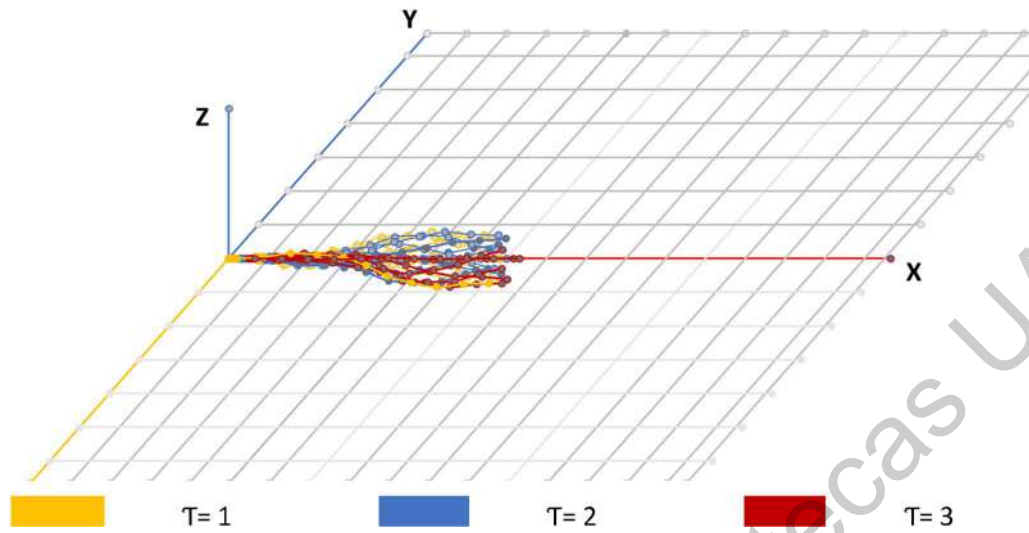
Animación 4. Descripción: “Movimiento de tipo concertina, simétrico, con señal sinusoidal como generador “

Parámetros del generador.	Valor	Unidades
α	$\pi/12$	[Rad].
γ	$\pi/36$	[Rad].
M	10	Adimensional.
δ	0	[Rad].
$\hat{\delta}$	0	[Rad].
τ	3	Adimensional

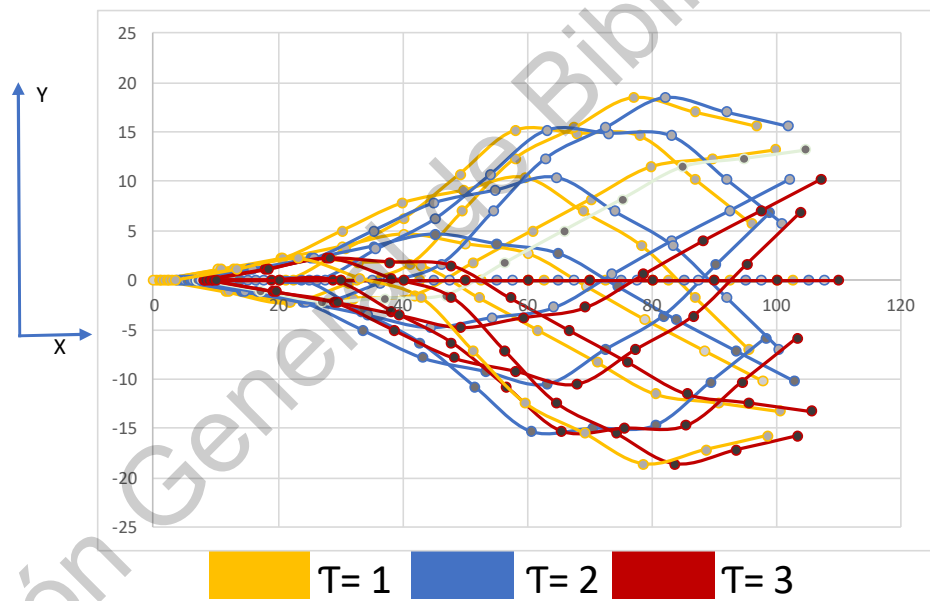
Tabla 15. Parámetros para generar la animación 4.

Funciones del generador.	Expresión
$u(t)$	-1
$\alpha_{i,t}$	$\sin[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	$\sin[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$

Tabla 16 Funciones implementadas en el algoritmo de locomoción de la animación 4.



Gráfica 18. Evolución del algoritmo de locomoción en la animación 4.



Gráfica 19. Vista superior de la evolución de la animación 4.

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	15	0		
Altura Máxima Alcanzada			1.93	
Tiempo de ejecución del algoritmo.				15.5664

Tabla 17. Mediciones obtenidas de la animación 4.

Dirección General de Bibliotecas UAQ

Animación 5. Descripción: “Movimiento de tipo concertina, asimétrico, con señal de tipo coseno como generador “

Parámetros del generador.	Valor	Unidades
α	$\pi/12$	[Rad]
γ	$\pi/36$	[Rad]
M	10	Adimensional.
δ	0	[Rad]
$\hat{\delta}$	0	[Rad]
τ	3	Adimensional

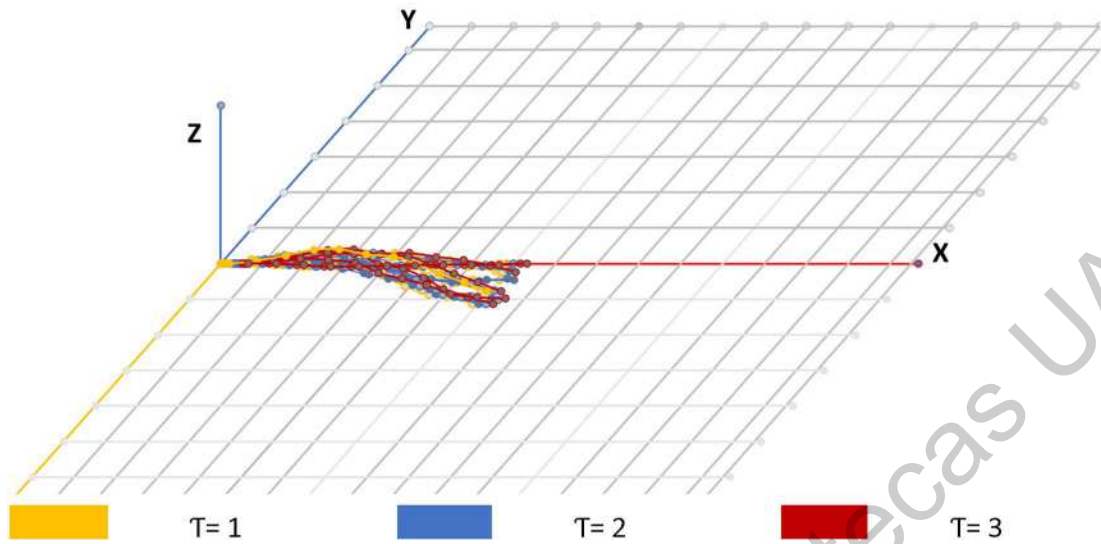
Tabla 18. Parámetros implementados en el algoritmo de la animación 5.

Funciones del generador.	Expresión
$u(t)$	-1 si $t < 0.5$ 1 si $t \geq 0.5$
$\alpha_{i,t}$	$\text{Cos}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	$\text{Cos}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$

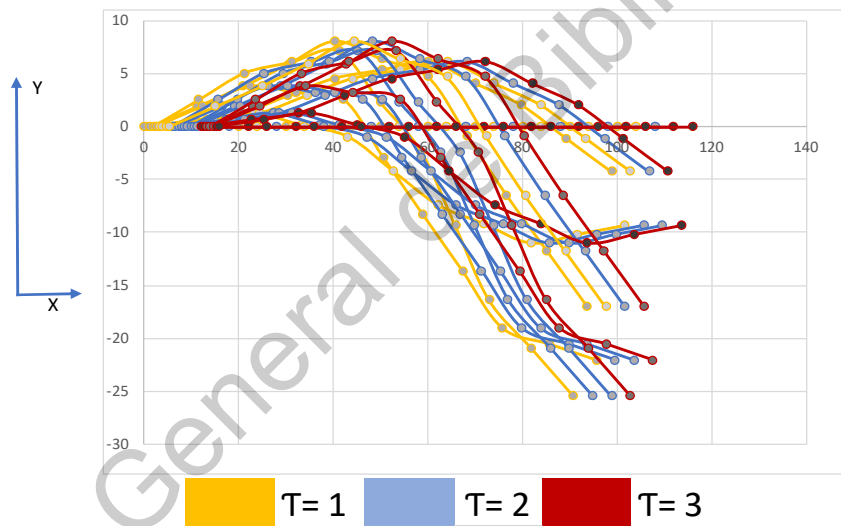
Tabla 19. Funciones implementadas en el algoritmo para la animación 5.

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	15	0		
Altura Máxima Alcanzada			1.628248	
Tiempo de ejecución del algoritmo.				15.43286

Tabla 20. Resultados de la ejecución del algoritmo de la animación 5.



Gráfica 20. Evolución del algoritmo de locomoción en la animación 5.



Gráfica 21. Vista superior de la evolución del algoritmo en la animación 5.

Animación 6. Descripción: “Movimiento de tipo concertina, simétrico, con señal de tipo coseno como generador”.

Parámetros del generador.	Valor	Unidades
α	$\pi/12$	[Rad]
γ	$\pi/36$	[Rad]
M	10	Adimensional.
δ	0	[Rad]
$\hat{\delta}$	0	[Rad]
τ	3	Adimensional

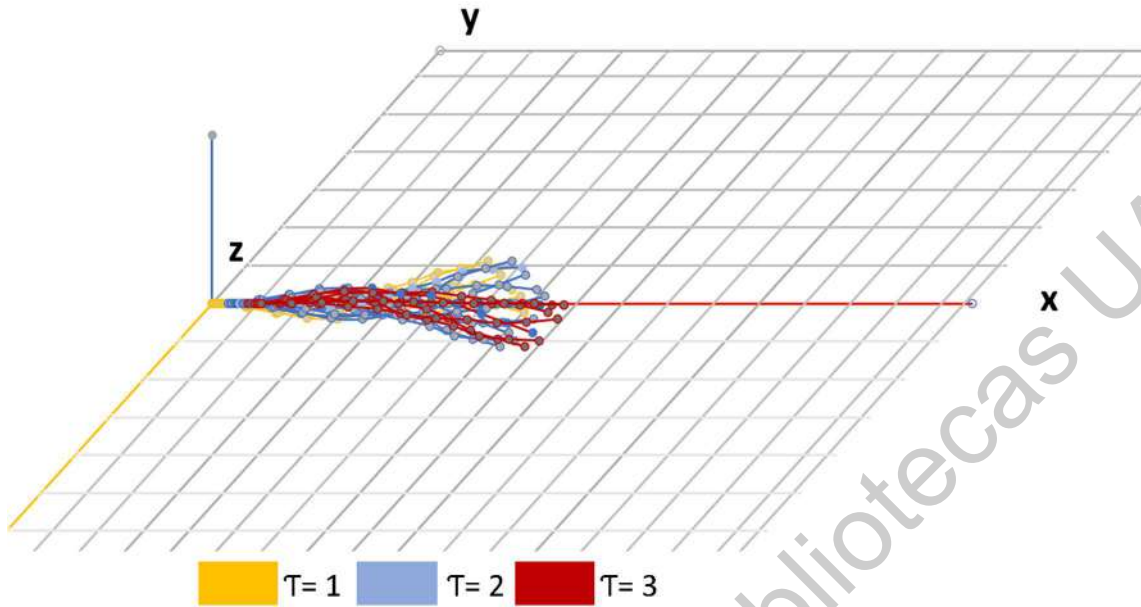
Tabla 21. Parámetros implementados en el algoritmo de la animación 6.

Funciones del generador.	Expresión
$u(t)$	-1
$\alpha_{i,t}$	$\text{Cos}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	$\text{Cos}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$

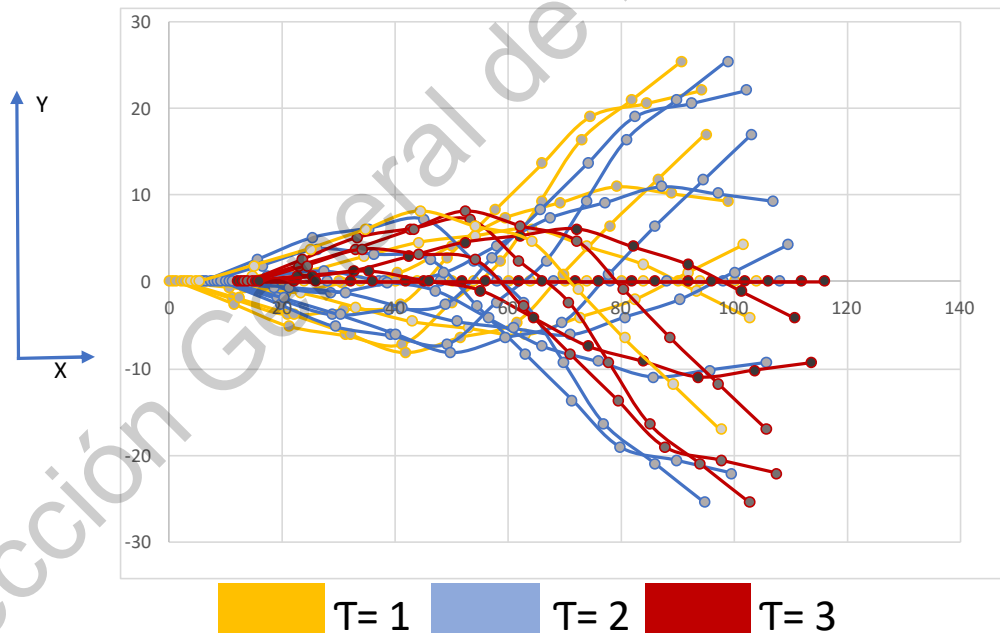
Tabla 22. Funciones implementadas en el algoritmo de la animación 6.

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	15	0		
Altura Máxima Alcanzada			1.628248	
Tiempo de ejecución del algoritmo.				15.43826

Tabla 23. Resultados de la ejecución del algoritmo en la animación 6



Gráfica 22. Evolución del algoritmo de locomoción en la animación 6.



Gráfica 23. Vista superior de la evolución del algoritmo de locomoción en la animación 6.

Animación 7. Descripción: “Movimiento de tipo *Sidewinding* asimétrico, con señal de tipo seno como generador “.

Parámetros del generador.	Valor	Unidades
α	$\pi/12$	[Rad]
γ	$\pi/72$	[Rad]
M	10	Adimensional.
δ	$-\pi/4$	[Rad]
$\hat{\delta}$	0	[Rad]
τ	3	Adimensional

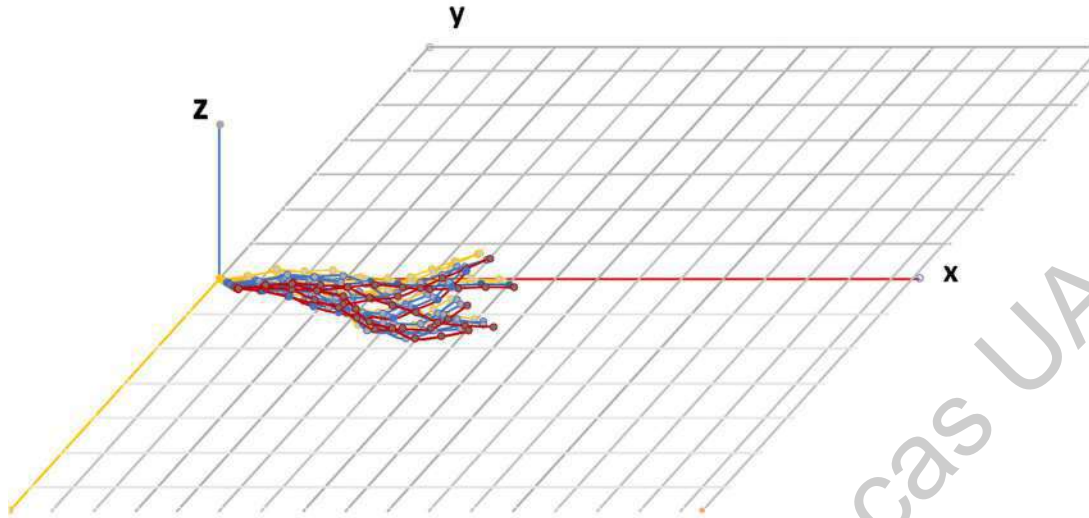
Tabla 24 Parámetros implementados en el algoritmo en la animación 7.

Funciones del generador.	Expresión
$u(t)$	-1 si $t < 0.5$ 1 si $t \geq 0.5$
$\alpha_{i,t}$	$\text{Sin}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	$\text{Sin}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$

Tabla 25. Funciones implementadas en el algoritmo de la animación 7

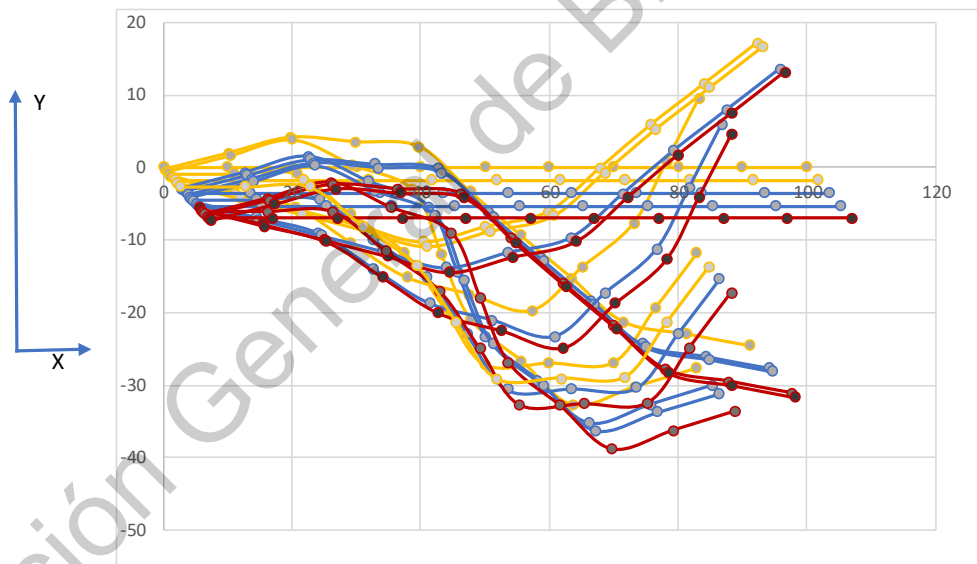
	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	10.60660172	10.60660172		
Altura Máxima Alcanzada			2.93607	
Tiempo de ejecución del algoritmo.				13.985

Tabla 26. Resultados de la ejecución del algoritmo de la animación 7.



T= 1
 T= 2
 T= 3

Gráfica 24. Evolución del algoritmo de locomoción



T= 1
 T= 2
 T= 3

Gráfica 25. Vista superior de la evolución del algoritmo de locomoción

Animación 8: “Movimiento de tipo *Sidewinding* simétrico, con señal de tipo seno como generador “.

Parámetros del generador.	Valor	Unidades
α	$\pi/12$	[Rad]
γ	$\pi/72$	[Rad]
M	10	Adimensional.
δ	$-\pi/4$	[Rad]
$\hat{\delta}$	0	[Rad]
τ	3	Adimensional

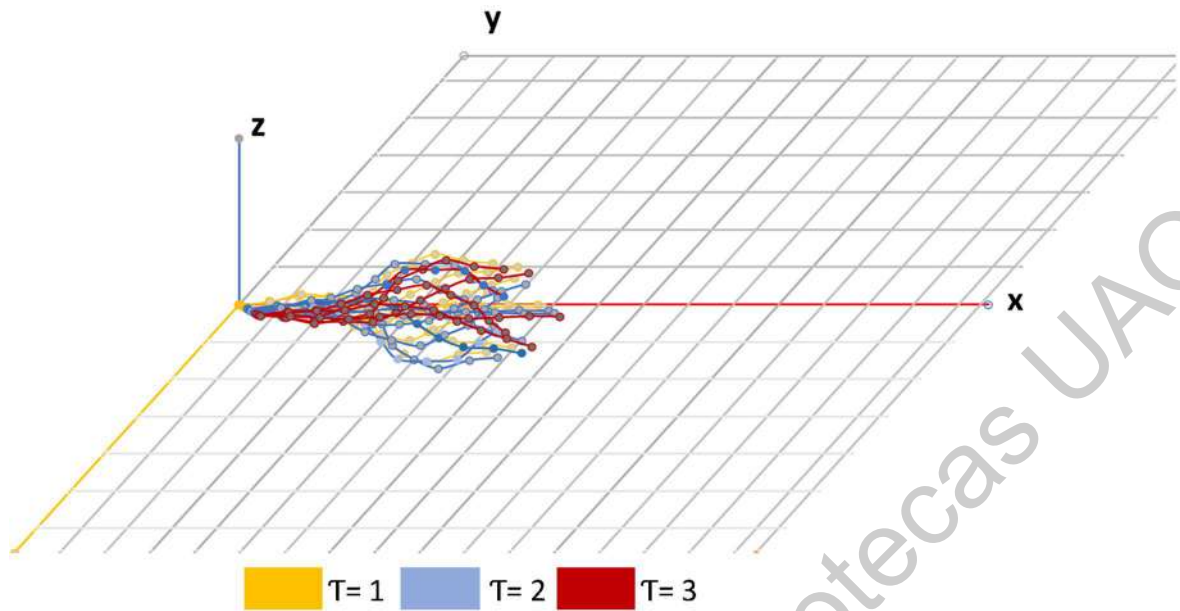
Tabla 27. Parámetros implementados en el algoritmo en la animación 8.

Funciones del generador.	Expresión
$u(t)$	-1
$\alpha_{i,t}$	$\text{Sin}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	$\text{Sin}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$

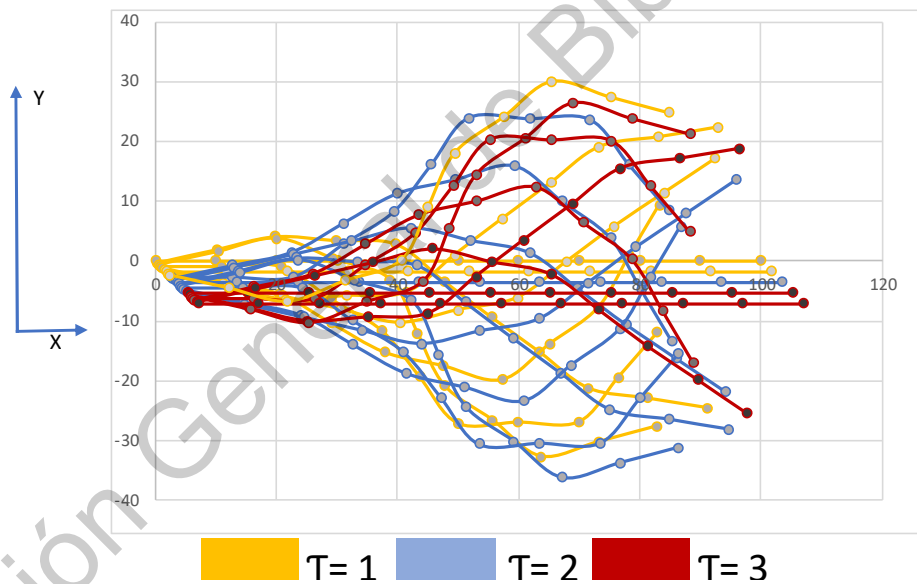
Tabla 28. Funciones implementadas en el algoritmo de la animación 8

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	10.60660172	10.60660172		
Altura Máxima Alcanzada			2.93607	
Tiempo de ejecución del algoritmo.				13.985

Gráfica 26. Resultados de la ejecución del algoritmo de la animación 8



Gráfica 27. Funciones implementadas en el algoritmo de la animación 8.



Gráfica 28. Vista superior de la evolución del algoritmo de locomoción de la animación 8.

Animación 9. Descripción: “Movimiento de tipo *Sidewinding* simétrico, con señal de tipo coseno como generador “.

Parámetros del generador.	Valor	Unidades
α	$\pi/36$	[Rad]
γ	$\pi/72$	[Rad]
M	10	Adimensional.
δ	$-11\pi/25$	[Rad]
$\hat{\delta}$	0	[Rad]
τ	3	Adimensional

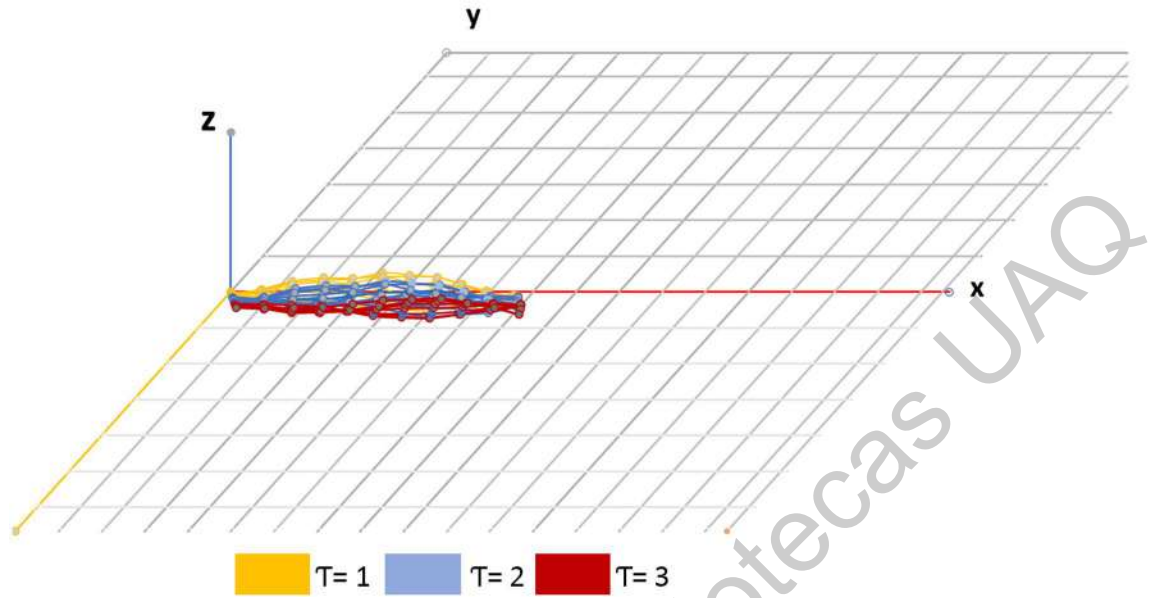
Tabla 29. Parámetros implementados en el algoritmo en la animación 9.

Funciones del generador.	Expresión
$u(t)$	-1
$\alpha_{i,t}$	$\text{Cos}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	$\text{Cos}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$

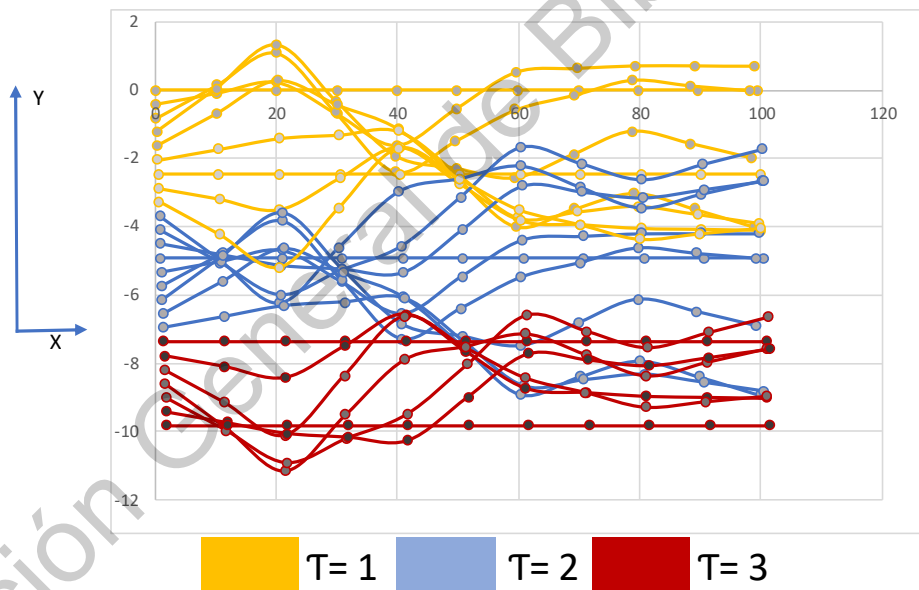
Tabla 30. Funciones implementadas en el algoritmo de la animación 9

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	14.7343	2.8107		
Altura Máxima Alcanzada			6.0351	
Tiempo de ejecución del algoritmo.				13.27

Tabla 31. Resultados de la ejecución del algoritmo de la animación 9.



Gráfica 29. Animación del algoritmo en la animación 9.



Gráfica 30. Vista superior de la evolución del algoritmo de locomoción en la animación 9.

Animación 10. Descripción: “Movimiento de tipo *Sidewinding* asimétrico, con señal de tipo coseno como generador “.

Parámetros del generador.	Valor	Unidades
α	$\pi/36$	[Rad]
γ	$\pi/72$	[Rad]
M	10	Adimensional.
δ	$-11\pi/25$	[Rad]
$\hat{\delta}$	0	[Rad]
τ	3	Adimensional

Tabla 32. Parámetros implementados en el algoritmo en la animación 10

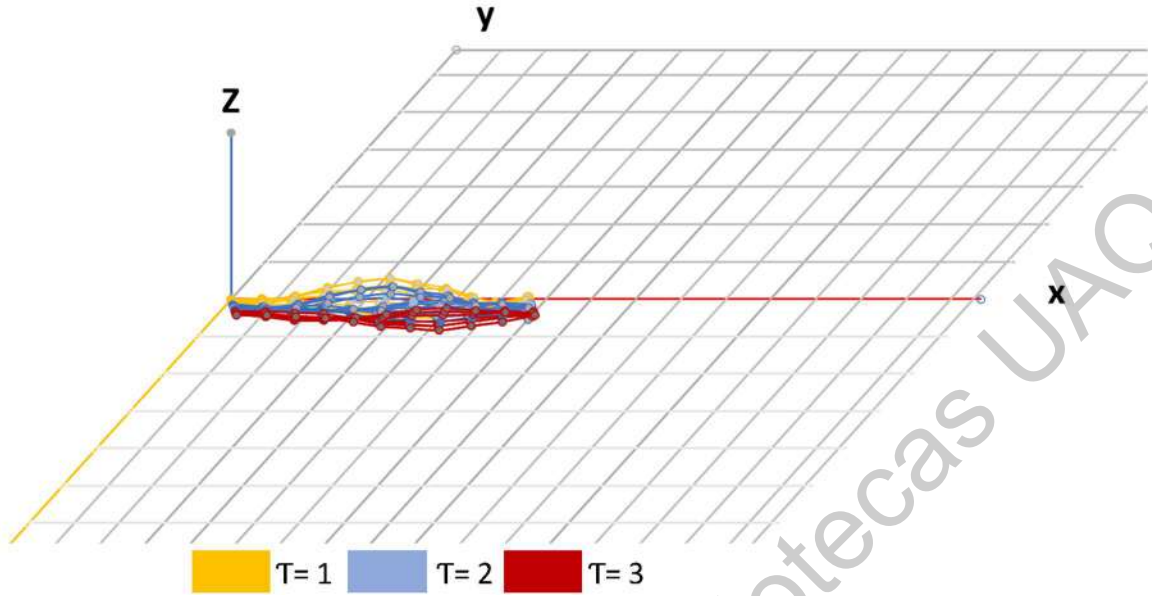
Tabla. Parámetros implementados en el algoritmo.

Funciones del generador.	Expresión
$u(t)$	-1 si $t < 0.5$ 1 si $t \geq 0.5$
$\alpha_{i,t}$	$\text{Cos}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$
$\gamma_{i,t}$	$\text{Cos}[\varphi(t) + \frac{2\pi k}{M}(i - 1)]$

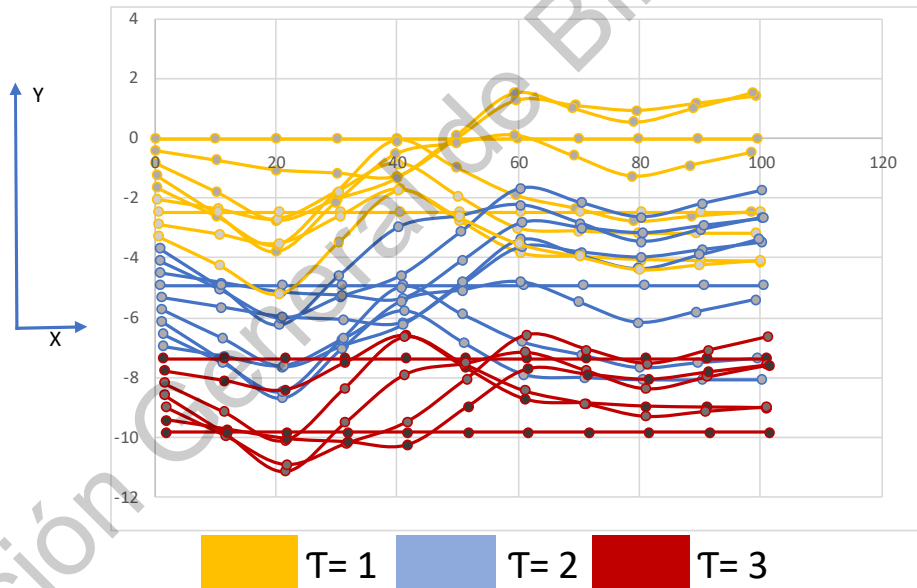
Tabla 33. Funciones implementadas en el algoritmo de la animación 10

	X [Cm].	Y [Cm].	Z [Cm].	Tiempo [S].
Distancia recorrida	2.8107	14.7343		
Altura Máxima Alcanzada			6.0351	
Tiempo de ejecución del algoritmo.				13.27

Tabla 34. Resultados de la ejecución del algoritmo de la animación 10.



Gráfica 31. Visualización de la animación 10.



Gráfica 32. Vista superior de la evolución del algoritmo de locomoción de la animación 10.

6.3 Discusión de resultados.

En base a los resultados obtenidos y las observaciones, puede apreciarse lo siguiente:

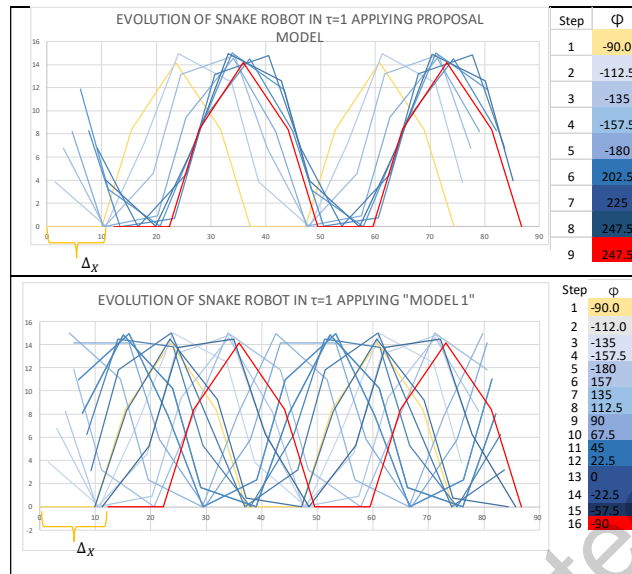
6.3.1 Modelo unidireccional.

El algoritmo fue posible implementar en Macro con VBA y C++. Observando que la complejidad del algoritmo es al menos $\Omega(n^2)$. Al implementarse el algoritmo desarrollado en la presente tesis, en código C++, se comparó con el propuesto en la literatura, obteniéndose, los siguientes resultados.

	Complejidad $f(n)$	Iteraciones por ciclo de locomoción	Tiempo [S]. Código	Tiempo [S].Animación	Δ_x [Cm]
Modelo 1	n^2	360	10,39	12.59	12.32
Modelo Propuesto	n^2	180	6,49	8.57	12.32

Tabla 35. Movimiento rectilíneo, comparación de resultados empleando algoritmo desarrollado y propuesto de literatura.

Se implementaron ambos algoritmos, obteniéndose las siguientes secuencias de movimiento dentro de un ciclo de locomoción, en cada transición, podemos notar que la estructura conserva al menos 2 puntos en el suelo, cumpliendo así con el criterio de estabilidad cinemático en robots ápodos (González, 2008).



Gráfica 33. Etapas de progresión dentro de un ciclo de locomoción de un snake robot empleando ambos algoritmos.

Podemos notar en la figura 33 que, con el algoritmo propio, solo fue necesario emplear 9 transiciones desde el reposo para completar un ciclo de locomoción, mientras en el código reportado por (González, 2008), se emplearon 16, por lo cual es una mejora en el ciclo de locomoción, Para consultar más a detalle, revisar (Anexo 4).

La mejora radica principalmente, en que el espectro en el que se mueve la función $\varphi(t)$, en el plano polar. En nuestra propuesta, no hay una transición por el cuadrante III, aprovechando que el valor de la función seno es la misma en 0 y $-\pi$, podemos saltar esa parte sin que haya un cambio brusco en el movimiento del robot. El estudio de la dinámica es necesario para validar este patrón de movimiento es efectivamente más eficiente.

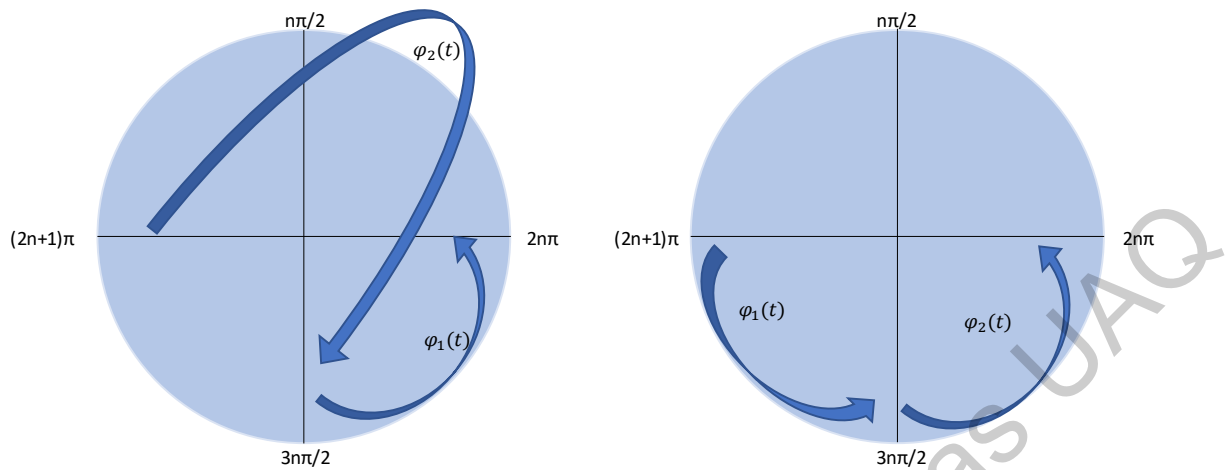


Figura 25. Perspectiva entre secuencias propuestas desde el plano polar.

Es importante notar que, en este perfil de locomoción, como en el bidireccional, solo se propuso una Δx , constante, ya que como estamos trabajando con cinemática, no podemos tener certeza que el movimiento generado por la estructura realmente va a proporcionar movimiento hacía adelante ya que faltan más variables, como el torque, la inercia o la fricción.

Sin embargo, dado que el modelo ya ha probado su funcionalidad como puede notarse en (Sandoval, 2018) y (González 2008). Podemos asumir un valor Δ , siempre y cuando la transición sea suave entre ciclos de trabajo, además la propuesta de mejora debería ser también validada mediante un análisis de la dinámica y posteriormente una implementación en una arquitectura.

6.3.2 Modelo Tridimensional.

El algoritmo presentado en la figura 19, al ser de mayor complejidad por la retroalimentación y el número de términos de polinomio, fue únicamente desarrollada en VBA, sin embargo, se anexa como memoria de cálculo las expresiones de los polinomios las cuales pueden ser programadas en cualquier lenguaje.

La relación entre el número de elementos del polinomio solución y el número de articulación está dada por la razón áurea, resulta interesante, para

determinar la complejidad computacional del algoritmo. Que podemos acotar al menos en:

$\Omega(\Phi(n))$, Donde Φ , es la razón aurea, dada por la siguiente expresión.

$$\Phi(n) = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right] \quad (57)$$

En la cual, para este caso

n , será el número de articulación.

Φ , será el número de términos del polinomio de tipo residual, para computar su P.C.D.

Consideramos que esta relación se produce, debido a que la solución del P.C.D, considera la rotación de la n -ésima articulación, en función de si misma y en función de la anterior, ya que estas están ligadas dada la estructura del robot, por lo cual el número de términos comienzan a acumularse bajo la secuencia antes mencionada.

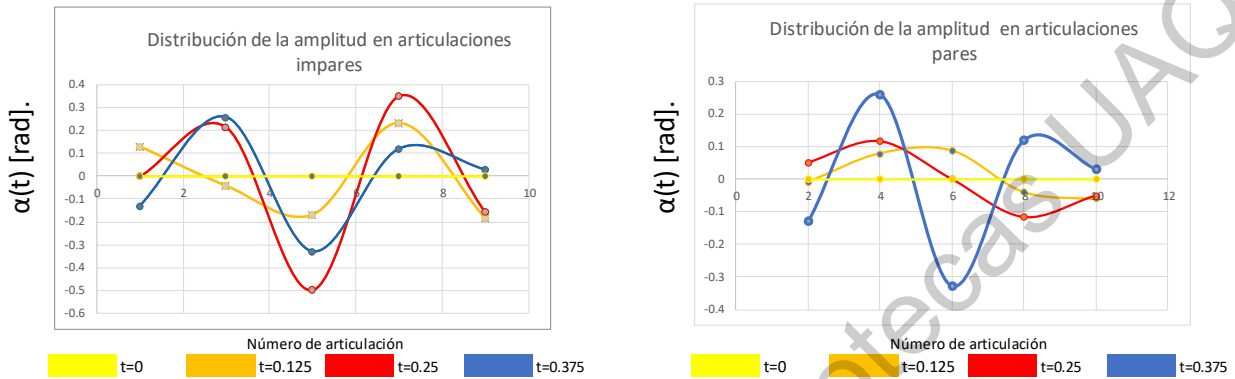
De igual forma que modelo unidireccional, este se compara con el presentado por (González, 2008). Para esto compararemos el modelo, enviando un seno como señal de onda, tanto un coseno, bajo los parámetros de la animación 3 y 4. Empleando nuestro modelo, con secuencia propuesta en la ecuación 21, y generadores de amplitud variable, expresados en la ecuación 55, mientras el modelo a comparar usa como secuencia el de la ecuación 20, y de generadores con amplitud constante.

Se muestra la distribución de la amplitud enviada al actuador durante la expansión durante medio ciclo de trabajo, dado que la señal es simétrica, a las articulaciones impares, pares y la progresión del robot desde el plano XZ. Para observar el cumplimiento de la estabilidad cinemática para robots ápodos.

Análisis de señal de la animación 1.

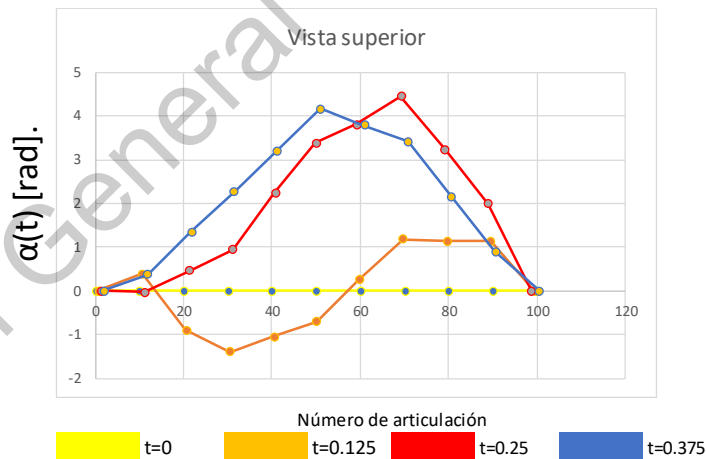
Podemos notar que la expansión máxima y la expansión mínima, se produce, de la forma deseada en la ecuación 45. en caso de las articulaciones pares, no siendo así así en caso de la amplitud en el caso de las articulaciones pares. Aunque ambas si

cumple con concentrar las señales de amplitud en el centro de la estructura y reducir las en los extremos.



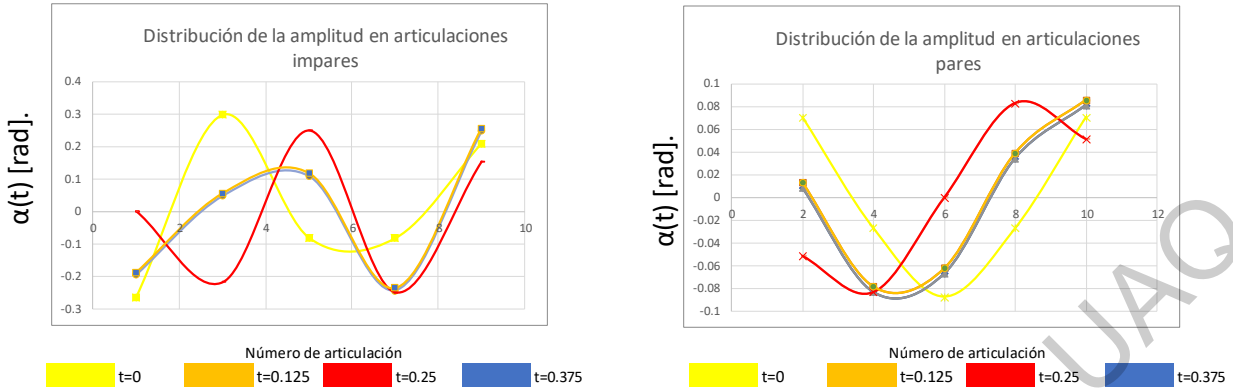
Gráfica 34. Distribución de la amplitud a través del robot sobre el plano XZ.

En la gráfica 35 podemos apreciar que este algoritmo no guarda el criterio de estabilidad cinemática, ya que al realizar la simulación las articulaciones 3 hasta la 7 están constantemente cruzando al eje -Z.



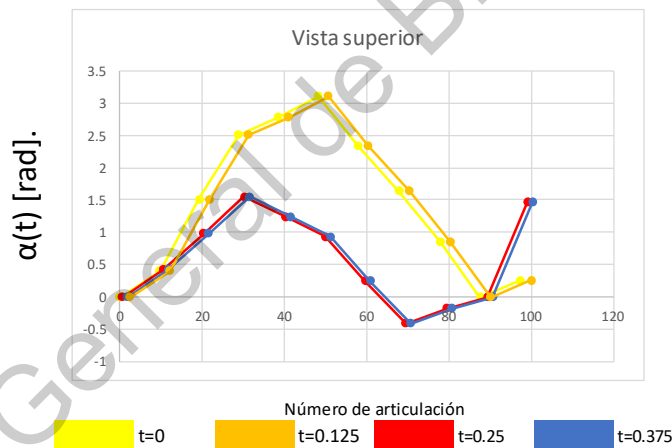
Gráfica 35. Evolución del algoritmo de locomoción en el plano XZ

Por lo cual decidimos analizar el algoritmo empleado por (González, 2008). Para comparar cómo se comporta. La gráfica 36, nos muestra que los valores de amplitud son más homogéneos a lo largo de la estructura del robot.



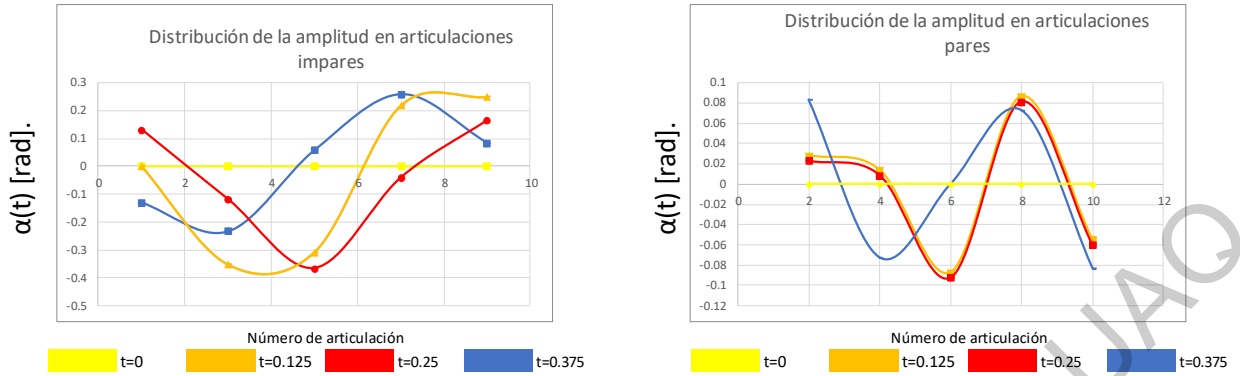
Gráfica 36. Distribución de la amplitud con el algoritmo. comparado.

Podemos observar en la gráfica 37 la proyección en XZ de este algoritmo, notamos que al menos dos articulaciones tienen su ubicación en eje -Z, siendo mínima (-0.4 cm). Muy probablemente, al funcionar el robot, este movimiento solo genere una acción de reacción en sentido opuesto dada la tercera ley de Newton.

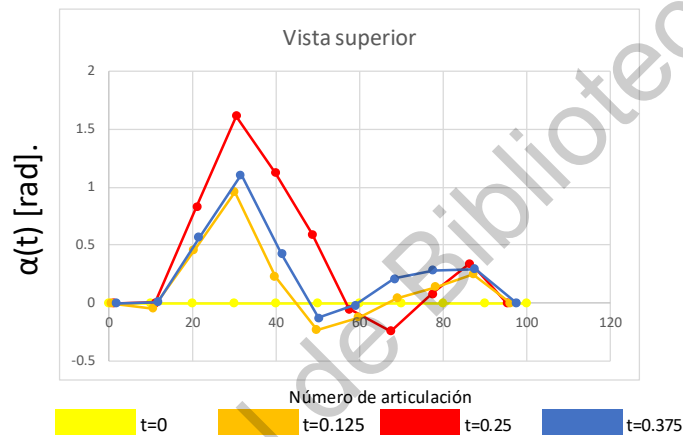


Gráfica 37. Proyección en el plano XZ del algoritmo de González (2008).

Dados estos resultados, decidimos cambiar las ecuaciones de desfase de la función $\varphi(t)$, pasando de ejecutar esta parte del algoritmo con la ecuación 23, a hacerlo con la ecuación 24. Conservando una función de amplitud variable para concentrarla en el centro del robot y expandirla hacia los costados.



Gráfica 38. Amplitud de onda en el algoritmo implementado.



Gráfica 39. Posición en el plano XZ del algoritmo implementado.

En esta última, podemos apreciar que la amplitud se concentra en la mitad de la estructura, en cuanto a la estabilidad, también ocurre que hay elementos del robot que están en el eje -z, aunque esto se da por -0.3 cm.

7 Conclusiones.

Respecto al modelo unidireccional, podemos aportar lo siguiente:

- 1) Permite la comprensión de cómo funciona la ecuación Serpenoide, partiendo de esto fue posible construir un algoritmo de locomoción, mismo que puede ser tratado con una relación de recurrencia y pudo aportarse una propuesta de mejora. **Cumpliendo con esto la generación de conocimiento y aporte académico, objetivos generales y específicos dentro de la presente investigación.**
- 2) . En ambas simulaciones, se cumple con la estabilidad cinemática del robot Por lo cual podemos dar este algoritmo como válido, cumpliendo nuestros objetivos específicos.
- 3) Facilita diseñar una secuencia de trabajo, la cual se muestra en la figura 19 y nos permite diseñar un algoritmo de locomoción para robots ápodos, para 2 o 3 dimensiones.
- 4) Este modelo nos facilita la comprensión de trabajos previos, así como el desarrollo de una propuesta, para un posterior análisis de su dinámica y diseño de una arquitectura, basado en una metodología de proyectos de índole mecatrónica.

Restricciones:

Si la estructura morfológica es con revolutas cuyo eje de rotación es intercalado, como la presentada en la figura 20, entonces este algoritmo podrá implementarse siempre y cuando las articulaciones impares tengan una señal de onda $\alpha = 0$, para que, de esta forma, solo las articulaciones con eje rotacional en Y, reciban la señal del algoritmo planteado en la figura 17.

Un contexto útil para este algoritmo sería en un caso en que no haya espacio hacia los lados para que la oscilación horizontal se lleve a cabo.

Si al estar trabajando con solo la mitad de las articulaciones, la aproximación de la curva generada por la Serpenoide, dada la naturaleza discreta de la morfología

discreta del robot, no será lo suficientemente adecuada y las transiciones serán muy bruscas. (González, 2008).

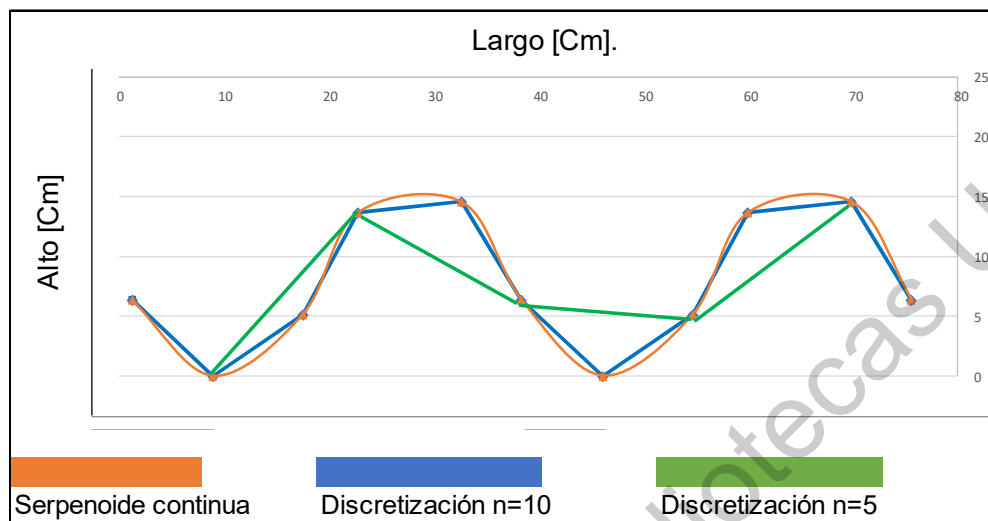


Figura 26. Discretización de la curva Serpenoide.

Por tanto, como consideraciones de diseño, el robot deberá tener la cualidad de reconfigurar los módulos, de forma que queden orientados respecto al mismo eje de rotación como el mostrado en la figura 16, para que pueda obtenerse una aplicación práctica de este algoritmo.

Sin embargo, para acercarnos más a la realidad en cuanto al movimiento, así como para validar la secuencia de movimiento propuesta, se recomienda plantear y resolver su dinámica en trabajos futuros.

Dadas las observaciones del modelo tridimensional, podemos aportar que:

La mejora propuesta en la ecuación 24, solo es útil en el modelo unidireccional, ya que, al implementarlo en 3 dimensiones, no cumple con la estabilidad cinemática, para este último, solo podríamos emplear la ecuación 23.

Los algoritmos empleados probablemente puedan mejorarse, si puede establecerse una expresión como la dada en la ecuación 21. Sin tener que generar una retroalimentación como la expresada en la ecuación 54, con la finalidad de generar un algoritmo como el presentado en la figura 17. A pesar de no haber obtenido esta

expresión, se planteó como se visualizaría el algoritmo (Anexo III). Sin embargo, basados en la información bibliográfica, sabemos que este problema ha sido tratado planteándolo como un sistema no holonómico, Por lo cuál el siguiente paso sería tratarlo con otra estructura algebraica, como lo es el álgebra de Lie (Ostrowski & Burdick, 1998).

El número de elementos del polinomio solución del PCD para la n-ésima articulación tiene una razón de crecimiento que obedece a la sucesión de Fibonacci. esto se debe a que las rotaciones de la arquitectura están basadas en la rotación de la n-ésima articulación, mientras que la siguiente articulación rota, dada la señal que le llega, mas la rotación de la anterior.

Esa transmisión de información dentro de la cadena cinemática origina que, para determinar la rotación de una articulación, se tenga que considerar la señal enviada por el generador a esta y la transmitida mecánicamente por las anteriores.

Restricciones:

El modelo planteado, que toma como inspiración la ecuación de calor, esta restringido al planteamiento y resolución de la dinámica para validarlo y pueda considerarse un algoritmo de locomoción valido, para implementarse en una arquitectura.

No todas las señales de tipo seno y coseno resultan validas tanto para el algoritmo planteado en literatura como el propuesto en el presente documento, ya que por sus propiedades en el plano polar pueden originar singularidades dentro de la locomoción, lo que se ve traducido en transiciones bruscas dentro de la animación.

7.1 Trabajos futuros

7.1.1 Modelo unidireccional:

Plantear y resolver su dinámica, proponer un diseño para este modelo, a pesar de que sería poco útil en una aplicación con un contexto real, por la restricción del movimiento, sería una actividad que resultaría educativa y de estudio para la comprensión del

movimiento de ese tipo de robots y para el planteamiento y solución del modelo en 3 dimensiones.

7.1.2 Modelo Tridimensional.

En cuanto al cómputo del P.C.D, podemos notar que la complejidad de este, está relacionado con la razón aurea, por lo cual, en investigaciones futuras, sería interesante explorar las siguientes líneas de investigación.

- 1: Realizar la demostración formal de dicha afirmación.
2. Determinar las 3 cotas de complejidad del algoritmo.
3. Dado que ya existen algoritmos para construir bloques a partir de la sucesión de Fibonacci (Ferrando & Segura, 2010), podríamos construir uno que genere todas las expresiones para construir el P.C.D, (ver anexo 2).

En cuanto al modelo basado en la ecuación de calor, consideramos que es un aporte inicial, la cual podría generar una propuesta de locomoción diferente y una buena propuesta para viraje y planeación de trayectorias en el sentido de que podemos secuenciar dos patrones de movimiento

El que ninguno de los 3 modelos presentados en la discusión de resultados conserve la dirección del movimiento sobre el eje Z positivo, no es prueba de que el algoritmo no sea útil, consideramos que, en esta parte del ciclo de trabajo, se generará un vector de igual magnitud pero de sentido opuesto por tercera ley de Newton (Stern, 2001). lo que al final de cuentas permite desplazar el robot en una dirección.

Proponer y resolver su dinámica es necesario para dar una correcta interpretación de estos resultados.

Estudiar, y repetir la secuencia de este trabajo de investigación considerando el robot como un sistema no holonómico (Ostrowski & Burdick, 1998),y verificar si mediante otra estructura algebraica, el algoritmo de locomoción no necesite de retroalimentación en el sistema, disminuyendo el coste computacional del algoritmo.

Volviendo al planteamiento de nuestra hipótesis. podemos concluir que mediante este estudio pudimos explorar a fondo el desarrollo de la cinemática de los robots ápodos, valorando de forma cuantitativa a través del desarrollo de dos algoritmos Y de forma cualitativa al desarrollarlos y validarlos en simulación.

En cuanto a la mejora de los algoritmos, pudimos obtener evidencia para **validarla únicamente en el caso unidireccional**, ya que nuestra propuesta en 3 dimensiones no cumple con los criterios de estabilidad cinemática. Al no cumplirse en el caso general, y solamente en el particular, se da la **hipótesis como falsa**, lo cual no disminuye el aporte de la presente investigación, sino que la acota.

Dirección General de Bibliotecas UNQ

Bibliografía

- Abut, T. (2015). Dynamic Model And Optimal Control Of A Snake Robot: TAROBOT – 1. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 4*, 1-6.
- Arizumi, R., & Matsuno, F. (2017). Dynamic Analysis of Three Snake Robot Gaits. *IEEE TRANSACTIONS ON ROBOTICS*, 1-3.
- Ferrando, I., & Segura, C. (2010). *La sucesión de Fibonacci como herramienta para modelizar la naturaleza*. Modelling in Science Education and Learning.
- Galbusera, F. (2019). The Spine. A Strong, Stable, and Flexible Structure with Biomimetics Potential. *Biomimetics*.
- Garzón, M. (2011). *Estrategias Bio-Inspiradas para Locomoción de Robots Ápodos*. Madrid.: Universidad Politécnica de Madrid.
- Gong, C., Tesch, M., Rollinson, D., & Choset, H. (2014). Snakes on an Inclined Plane: Learning an Adaptive Sidewinding. *International Conference on Intelligent Robots and Systems*, 3-5.
- González, J. (2002). *Principales Lineas de Investigación en Robots Ápodos*.
- González, J. (2008). *ROBÓTICAMODULAR Y LOCOMOCIÓN: Aplicación a Robot Apodos*. Madrid: Universidad Autónoma de Madrid.
- Gray, J. (1946). The mechanism of locomotion in snakes. *Journal of Experimental Biology*, 1-6.
- Hannan, & Walker. (2000). Analysis and initial experiments for a novel elephant's trunk robot. *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, 1-3.
- Hirose, S., & Mori, M. (1993). Biologically Inspired Snake-like Robots. *International Conference on Robotics and Biomimetics* (pp. 1-7). Shenyang: IEEE.
- Hirose, S., & Yamada, H. (2009). Snake-Like Robots, Machine Design of Biologically Inspired Robots. *IEEE & Robotics Magazine*.
- IFR. (2016, 10 02). *Service Robot*. Retrieved from <https://ifr.org/service-robots/>.
- ISO Standards. (2012, 03). *ISO 8373*. Retrieved from <https://www.iso.org/standard/55890.html>.
- Kakogawa, A., S, J., & Ma, S. (2018). Stiffness Design of a Resonance-Based Planar Snake Robot With Parallel Elastic Actuators. *IEEE ROBOTICS AND AUTOMATION LETTERS*, 1-4.

- Kelasidi, E. M., & Pettersen, K. ... (2018). Locomotion Efficiency Optimization of Biologically Inspired Snake Robots. *Applied Science*, 1-15.
- Kelasidi, E., Liljebäck, P., Pettersen, P., & Grasdahl, T. (2015). Biologically Inspired Swimming Snake Robots. Modeling, Control and Experimental Investigation. *IEEE ROBOTICS AND AUTOMATION MAGAZINE.*, 1,3,15-16.
- Liljebäck, P., Pettersen, K., Stavadahl, O., & Grasdahl, J. (2011.). Snake Robot Locomotion in Environments with Obstacles. *IEEE TRANSACTIONS ON MECHATRONICS*, 1,2, 11,12.
- Liljebäck, P., Pettersen, K., Stavadahl, O., & Graydahl, J. (2011). Snake Robot Locomotion in Environments with Obstacles. *IEEE TRANSACTIONS ON MECHATRONICS.*, 1-2.
- Liljebäck, P., Stavadahl, O., Pettersen, K., & T., G. (2014). Mamba ,A Waterproof Snake Robot with Tactile Sensing. *IEEE International Conference on Intelligent Robots and Systems.*, 1-8.
- Lilywhite, H. (2014). How Snake Works. In H. Lillywhite, *How Snake Works* (pp. 77-102). New York: Oxford University Press.
- Menchaca, A. (2010, o6 06). "México recurre a tecnología extranjera", *El informador*. Retrieved from <https://www.informador.mx/Tecnologia/Mexico-recurre-a-tecnologia-extranjera-20100606-0109.html>.
- Mukherjee, D., Kar, I., & S., M. (2017). Adaptive Sliding Mode Control for Head-Angle and Velocity Tracking of Planar Snake Robots. *11th Asian Control Conference (ASCC)*, 1,6.
- Ostrowski, J., & Burdick, J. (1998). The Geometric Mechanics of Undulatory Robotic Locomotion. *The International Journal of Robotics Research*, 1-3.
- Pettersen, K. (2017). Snake Robots. *Elsevier*, 1-3.
- Reyes, F., & Ma, S. (2017). Snake Robots in Contact with the Environment: Influence of the Friction on the Applied Wrench. *International Conference on Intelligent Robots and Systems (IROS)*, 1-4.
- Rezapour, T. P., Grasdahl, & Hofmann. (2016). Formation Control of Underactuated Bio-inspired Snake Robots. *Artificial Life and Robotica*, 1-3.
- Rollinson, D. (2014). *Control and Design of Snake Robots*. Pittsburgh: School of Computer Science.
- Sánchez-Martín, F., J, J., Millán Rodríguez, F., Salvador-Bayarri, J., Monllau Font, V., Palou Redorta, J., & Villavicencio Mavrigh, H. (2007). Historia de la robótica: de Arquitas de Tarento al Robot da Vinci.

- Sandoval, X, Castillo, E, Ruiz, & M. (2018). Desarrollo de un Robot Gusano Capaz de Desplazarse en Terreno Natural. In J. Ramos, E. Vargas, & E. Gorrosieta, *Robótica y Mecatrónica* (pp. 12-22). Querétaro.
- San-Millan, A. (2011). DISEÑO, CONSTRUCCIÓN Y CONTROL DE UNA SERPIENTE ROBÓTICA. *UNIVERSIDAD DE CASTILLA - LA MANCHA*, 20-100.
- Stern, P. (2001, 12 13). *Newton Laws of Motion*. Retrieved from De astrónomos a Astronaves.: <https://pwg.gsfc.nasa.gov/stargaze/Mnewton.htm>
- Transeth, A. (2007). *Modelling and Control of Snake Robots*. Trondheim: Norwegian University of Science and technology.
- Vargas, J., Pedraza, J., Ramos, J., López, F, A., G, & V. (2014). Desarrollo de proyectos mecatrónicos. *VIII Congreso Internacional de Ingeniería Mecánica y Mecatrónica.*, 1-5.
- Wang, Osbourne, & Alben. (2014). Optimizing snake locomotion on an inclined plane. 1-10.
- Zhang, S., & Guo, Y. (2009). Bio-inspired locomotion for a modular snake robot. *Proceedings Volume 7321, Bio-Inspired/Biomimetic Sensor Technologies and Applications*, 1-5,10.

Anexos

Anexo I. Solución al problema cinemático directo en 3 dimensiones.

n=1, hasta n=3

$$\text{Table}[f_i = \text{Cos}[q_i], \{i, 1, 10, 1\}]$$
{tabla} {coseno}

$$\text{Table}[g_i = \text{Sin}[q_i], \{i, 1, 10, 1\}]$$
{tabla} {seno}

$$\{\text{Cos}[q_1], \text{Cos}[q_2], \text{Cos}[q_3], \text{Cos}[q_4], \text{Cos}[q_5], \text{Cos}[q_6], \text{Cos}[q_7], \text{Cos}[q_8], \text{Cos}[q_9], \text{Cos}[q_{10}]\}$$

$$\{\text{Sin}[q_1], \text{Sin}[q_2], \text{Sin}[q_3], \text{Sin}[q_4], \text{Sin}[q_5], \text{Sin}[q_6], \text{Sin}[q_7], \text{Sin}[q_8], \text{Sin}[q_9], \text{Sin}[q_{10}]\}$$

PARA N = 1

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = 1 \begin{pmatrix} f_1 \\ g_1 \\ 0 \end{pmatrix}$$

PARA N = 2

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = 1 \begin{pmatrix} f_1 (1 + f_2) \\ g_1 (1 + f_2) \\ g_2 \end{pmatrix}$$

PARA N = 3

$$\begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix} = 1 \begin{pmatrix} f_1 (1 + f_2 + f_3) - g_1 g_3 \\ g_1 (1 + f_2 + f_3) + f_1 g_3 \\ g_2 (1 + f_3) \end{pmatrix}$$

n=4, hasta n=5

PARA N =4

$$In[] = \begin{pmatrix} X_4 \\ Y_4 \\ Z_4 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (1 + \sum_{i=2}^4 \prod_{k=2}^i f_k - g_2 g_4) - g_1 g_3 (1 + f_4) \\ g_1 (1 + \sum_{i=2}^4 \prod_{k=2}^i f_k - g_2 g_4) + f_1 g_3 (1 + f_4) \\ g_2 (1 + \sum_{i=3}^4 \prod_{k=3}^i f_k) + f_2 g_4 \end{pmatrix}$$

PARA N =5

$$In[] = \begin{pmatrix} X_5 \\ Y_5 \\ Z_5 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (1 + \sum_{i=2}^5 \prod_{k=2}^i f_k - f_2 g_3 g_5 - g_2 g_4 (1 + f_5)) - g_1 (f_3 g_5 + g_3 (1 + f_4 + f_4 f_5)) \\ g_1 (1 + \sum_{i=2}^5 \prod_{k=2}^i f_k - f_2 g_3 g_5 - g_2 g_4 (1 + f_5)) + f_1 (f_3 g_5 + g_3 (1 + f_4 + f_4 f_5)) \\ g_2 (1 + \sum_{i=3}^5 \prod_{k=3}^i f_k - g_3 g_5) + f_2 g_4 (1 + f_5) \end{pmatrix}$$

$$Out[] = \{ \{ \mathbf{1} (-\sin[q_1] ((1 + \cos[q_4] + \cos[q_4] \cos[q_5])) \sin[q_3] + \cos[q_3] \sin[q_5])) + \cos[q_1] (1 + \cos[q_2] + \cos[q_2] \cos[q_3] + \cos[q_2] \cos[q_3] \cos[q_4] + \cos[q_2] \cos[q_3] \cos[q_4] + \cos[q_2] \cos[q_3] \cos[q_4] \cos[q_5]) + \cos[q_2] \cos[q_3] \cos[q_4] \cos[q_5] - (1 + \cos[q_5]) \sin[q_2] \sin[q_4] - \cos[q_2] \sin[q_3] \sin[q_5]) \} \}, \{ \mathbf{1} (\cos[q_1] ((1 + \cos[q_4] + \cos[q_4] \cos[q_5])) \sin[q_3] + \cos[q_3] \sin[q_5]) + \sin[q_1] (1 + \cos[q_2] + \cos[q_2] \cos[q_3] + \cos[q_2] \cos[q_3] \cos[q_4] + \cos[q_2] \cos[q_3] \cos[q_4] \cos[q_5] - (1 + \cos[q_5]) \sin[q_2] \sin[q_4] - \cos[q_2] \sin[q_3] \sin[q_5])) \} \}, \{ \mathbf{1} (\cos[q_2] (1 + \cos[q_3] + \cos[q_3] \cos[q_4] + \cos[q_3] \cos[q_4] \cos[q_5]) - \sin[q_3] \sin[q_5]) \} \}$$

n= 6, n=7 parte 1 y 2.

PARA N=6

$$\begin{pmatrix} x_6 \\ y_6 \\ z_6 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (1 + \sum_{i=2}^6 \prod_{k=2}^i f_k - f_2 (f_3 g_4 g_6 + g_3 g_5 (1 + f_6)) - g_2 (f_4 g_6 + g_4 ((1 + \sum_{i=5}^6 \prod_{k=5}^i f_k)))) - g_1 (f_3 g_5 (1 + f_6) + g_3 (1 + \sum_{i=4}^6 \prod_{k=4}^i f_k - g_4 g_6)) \\ g_1 (1 + \sum_{i=2}^6 \prod_{k=2}^i f_k - f_2 (f_3 g_4 g_6 + g_3 g_5 (1 + f_6)) - g_2 (f_4 g_6 + g_4 ((1 + \sum_{i=5}^6 \prod_{k=5}^i f_k)))) + f_1 (f_3 g_5 (1 + f_6) + g_3 (1 + \sum_{i=4}^6 \prod_{k=4}^i f_k - g_4 g_6)) \\ g_2 (1 + \sum_{i=3}^6 \prod_{k=3}^i f_k - f_3 g_4 g_6 - g_3 g_5 (1 + f_6)) + f_2 (f_4 g_6 + g_4 (1 + f_5 + f_6)) \end{pmatrix}$$

PARA N=7

$$\begin{pmatrix} x_7 \\ y_7 \\ z_7 \end{pmatrix} =$$

$\mathbf{1}$

$$\begin{pmatrix} f_1 (1 + \sum_{i=2}^7 \prod_{k=2}^i f_k - (f_2 (f_3 (f_4 g_5 g_7 + g_4 g_6 (1 + f_7)) + g_3 (f_5 g_7 + g_5 (1 + f_6 + f_7))) + g_2 f_4 g_6 (1 + f_7)) - g_2 g_4 ((1 + \sum_{i=5}^7 \prod_{k=5}^i f_k) - g_1 (1 + \sum_{i=2}^7 \prod_{k=2}^i f_k - (f_2 (f_3 (f_4 g_5 g_7 + g_4 g_6 (1 + f_7)) + g_3 (f_5 g_7 + g_5 (1 + f_6 + f_7))) + g_2 f_4 g_6 (1 + f_7)) - g_2 g_4 ((1 + \sum_{i=5}^7 \prod_{k=5}^i f_k) - g_2 (1 + \sum_{i=3}^7 \prod_{k=3}^i f_k - f_3 (f_4 g_5 g_7 + g_4 g_6 (1 + f_7)) - g_3 (f_5 g_7 + g_5 (1 + f_6 + f_7))) + f_2 ($$

$$\begin{aligned} & - g_2 g_4 ((1 + \sum_{i=5}^7 \prod_{k=5}^i f_k) - g_5 g_7)) - g_1 (f_3 (f_5 g_7 + g_5 (1 + f_6 + f_7)) + g_3 (1 + \sum_{i=4}^7 \prod_{k=4}^i f_k - f_4 g_5 g_7 - g_4 g_6 (1 + f_7))) \\ & - g_2 g_4 ((1 + \sum_{i=5}^7 \prod_{k=5}^i f_k) - g_5 g_7)) + f_1 (f_3 (f_5 g_7 + g_5 (1 + f_6 + f_7)) + g_3 (1 + \sum_{i=4}^7 \prod_{k=4}^i f_k - f_4 g_5 g_7 - g_4 g_6 (1 + f_7))) \\ & + g_5 (1 + f_6 + f_7)) + f_2 (f_4 g_6 (1 + f_7) + g_4 ((1 + \sum_{i=5}^7 \prod_{k=5}^i f_k) - g_5 g_7)) \end{aligned}$$

n=8. Parte 1, Parte 2, Parte 3,

$$\begin{pmatrix} x_8 \\ y_8 \\ z_8 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (1 + \sum_{i=2}^8 \prod_{k=2}^i f_k - g_2 (g_4 ((1 + \sum_{i=5}^8 \prod_{k=5}^i f_k) - g_5 g_7 (1 + f_8) - g_6 g_8 f_5) + f_4 f_6 g_8) - (f_2 (f_3 (f_4 (f_5 g_6 g_8 + g_5 g_7 (1 + f_8))) + g_1 (1 + \sum_{i=2}^8 \prod_{k=2}^i f_k - g_2 (g_4 ((1 + \sum_{i=5}^8 \prod_{k=5}^i f_k) - g_5 g_7 (1 + f_8) - g_6 g_8 f_5) + f_4 f_6 g_8) - (f_2 (g_3 (g_5 (1 + \sum_{i=6}^8 \prod_{k=6}^i f_k - g_6 g_8) + g_2 (1 + \sum_{i=3}^8 \prod_{k=3}^i f_k - g_3 (g_5 (1 + \sum_{i=6}^8 \prod_{k=6}^i f_k - g_6 g_8) (1 + f_8)) + g_4 f_6 g_8) + g_3 (f_5 g_7 (1 + f_8) + g_5 (1 + \sum_{i=6}^8 \prod_{k=6}^i f_k - g_6 g_8))) + g_2 f_4 g_6 (1 + f_7 + f_8)) - g_1 (f_3 (f_5 g_7 (1 + f_8) - g_6 g_8) + f_5 g_7 (1 + f_8)) + f_3 (g_4 g_6 (1 + f_7 + f_8) + f_4 (g_5 g_7 (1 + f_8) + f_5 g_6 g_8) + f_6 g_4 g_8) + f_4 g_2 g_6 (1 + f_7 + f_8)) + f_1 (g_3 (1 + \sum_{i=4}^8 \prod_{k=4}^i f_k - g_6 g_8) + g_7 f_5 (1 + f_8)) - f_3 (g_4 (g_6 (1 + f_7 + f_8) + g_8 f_6) + f_4 (g_5 g_7 (1 + f_8) + f_5 g_6 g_8)) + f_2 (g_4 ((1 + \sum_{i=5}^8 \prod_{k=5}^i f_k) - g_5 g_7 (1 + f_8) - g_6 g_8 f_5) + g_5 (1 + \sum_{i=4}^8 \prod_{k=4}^i f_k - g_4 (g_6 (1 + f_7 + f_8) + g_8 f_6) + f_4 (g_5 g_7 (1 + f_8) + f_5 g_6 g_8))) + g_3 (1 + \sum_{i=4}^8 \prod_{k=4}^i f_k - f_4 (f_5 g_6 g_8 + g_5 g_7 (1 + f_8)) - g_4 (g_6 (1 + f_7 + f_8) + f_6 g_8)) + f_1 (g_3 (1 + \sum_{i=4}^8 \prod_{k=4}^i f_k - g_4 (g_6 (1 + f_7 + f_8) + f_6 g_8) - g_6 g_8 f_4 f_5) + g_5 (f_3 (1 + \sum_{i=6}^8 \prod_{k=6}^i f_k - g_6 g_8)) - (g_7 (1 + f_8) (f_4 g_3 g_5 - f_3 f_5)) + f_5 f_k) - g_5 g_7 (1 + f_8) - g_6 g_8 f_5) + f_4 (g_6 (1 + f_7 + f_8) + f_6 g_8) \end{pmatrix}$$

n=10. Parte 1, 2, 3, 4.

$$\begin{pmatrix} x_{10} \\ y_{10} \\ z_{10} \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (1 + \sum_{i=2}^{10} \prod_{k=2}^i f_k - (f_2 (f_3 (f_4 (f_5 (f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10})) + g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10}))) + g_5 (f_7 g_9 (1 + f_{10}) + g_7 (1 + \\ g_1 (1 + \sum_{i=2}^{10} \prod_{k=2}^i f_k - (f_2 (f_3 (f_4 (f_5 (f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10})) + g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10}))) + g_5 (f_7 g_9 (1 + f_{10}) + g_7 (1 + \\ + g_7 (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10}))) + g_4 (f_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) + g_6 (1 + \sum_{i=7}^{10} \prod_{k=7}^i f_k - g_7 g_9 (1 + f_{10}) - f_7 g_8 g_{10})) + g_3 (f_5 (g_7 (1 + \\ + g_7 (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10}))) + g_4 (f_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) + g_6 (1 + \sum_{i=7}^{10} \prod_{k=7}^i f_k - g_7 g_9 (1 + f_{10}) - f_7 g_8 g_{10})) + g_3 (f_5 (g_7 (1 + \\ g_2 (1 + \sum_{i=3}^{10} \prod_{k=3}^i f_k - g_3 (g_5 (1 + \sum_{i=6}^{10} \prod_{k=6}^i f_k - f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10}))) + f_4 g_8 \\ (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10}) + f_7 g_9 (1 + f_{10})) + g_5 (1 + \sum_{i=6}^{10} \prod_{k=6}^i f_k - g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) - f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10}))) + f_4 g_8 \\ (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10}) + f_7 g_9 (1 + f_{10})) + g_5 (1 + \sum_{i=6}^{10} \prod_{k=6}^i f_k - g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) - f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10}))) + f_4 g_8 \\ g_{10} + x g_9 (1 + f_{10}) - g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) + f_5 (f_7 g_9 (1 + f_{10}) + g_7 (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10})) - f_3 (g_4 (g_6 ((1 + \sum_{i=7}^{10} \prod_{k=7}^i f_k \\)) + f_4 g_2 g_6 (1 + \sum_{i=7}^{10} \prod_{k=7}^i f_k)) - g_2 (f_4 (f_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) - g_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10}))) + g_4 ((1 + \sum_{i=5}^{10} \prod_{k=5}^i f_k) - f_5 (g_6 \\)) + f_4 g_2 g_6 (1 + \sum_{i=7}^{10} \prod_{k=7}^i f_k)) - g_2 (f_4 (f_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) - g_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10}))) + g_4 ((1 + \sum_{i=5}^{10} \prod_{k=5}^i f_k) - f_5 (g_6 \\ + \sum_{i=7}^{10} \prod_{k=7}^i f_k - f_7 g_8 g_{10}) - g_7 g_9 (1 + f_{10})) + f_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) + f_4 (g_5 (f_7 g_9 (1 + f_{10}) + g_7 (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10})) + f_5 (f_6 (
 \end{pmatrix}$$

n=10, parte 5,6,7,8

$$\begin{aligned}
 & \dots) - \{g_1 (g_3 (1 + \sum_{i=4}^{10} \prod_{k=4}^i f_k - f_4 (f_5 (f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10})) + g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10}))) + g_5 (f_7 g_9 (1 + f_{10}) + g_7 ((1 + \sum_{i=4}^{10} \prod_{k=4}^i f_k - f_4 (f_5 (f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10})) + g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10}))) + g_5 (f_7 g_9 (1 + f_{10}) + g_7 ((1 + \sum_{i=4}^{10} \prod_{k=4}^i f_k - g_8 g_{10})) + f_5 (f_6 (-f_7 g_8 g_{10} - g_7 g_9 (1 + f_{10})) - g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10}))) + f_4 (g_6 (1 + \sum_{i=7}^{10} \prod_{k=7}^i f_k - f_7 g_8 g_{10} \\
 &) + g_7 ((1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k) - g_8 g_{10}))) - g_4 (g_6 (1 + \sum_{i=7}^{10} \prod_{k=7}^i f_k - f_7 g_8 g_{10} - g_7 g_9 (1 + f_{10})) + f_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10}))) + f_3 (f_5 (f_7 g_9 (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k) - g_8 g_{10}))) - g_4 (g_6 (1 + \sum_{i=7}^{10} \prod_{k=7}^i f_k - f_7 g_8 g_{10} - g_7 g_9 (1 + f_{10})) + f_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10}))) + f_3 (f_5 (f_7 g_9 (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10})) + f_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10}))) \\
 & (1 + f_9 + f_9 f_{10}))) + f_3 (f_5 (f_7 g_9 (1 + f_{10}) + g_7 (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10})) + g_5 (1 + \sum_{i=6}^{10} \prod_{k=6}^i f_k - \\
 & (1 + f_9 + f_9 f_{10}))) + f_3 (f_5 (f_7 g_9 (1 + f_{10}) + g_7 (1 + \sum_{i=8}^{10} \prod_{k=8}^i f_k - g_8 g_{10})) + g_5 (1 + \sum_{i=6}^{10} \prod_{k=6}^i f_k - \\
 & - g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) - f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10}))) \\
 & - g_6 (f_8 g_{10} + g_8 (1 + f_9 + f_9 f_{10})) - f_6 (f_7 g_8 g_{10} + g_7 g_9 (1 + f_{10}))) \\
 \end{aligned}$$

Anexo 2. Comprobación computacional de la ecuación 34.

Los vectores residuo, son:

$$\begin{aligned}
 & \text{Table}[f_i = \text{Cos}[q_i], \{i, 0, 10, 1\}] \\
 & \text{Table}[g_i = \text{Sin}[q_i], \{i, 0, 10, 1\}] \\
 & \begin{pmatrix} \text{RX}_1 \\ \text{RY}_1 \\ \text{RZ}_1 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 f_2 \\ g_1 f_2 \\ g_2 \end{pmatrix} \\
 & \begin{pmatrix} \text{RX}_2 \\ \text{RY}_2 \\ \text{RZ}_2 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 f_2 f_3 - g_1 g_3 \\ g_1 f_2 f_3 + f_1 g_3 \\ g_2 f_3 \end{pmatrix} \\
 & \begin{pmatrix} \text{RX}_3 \\ \text{RY}_3 \\ \text{RZ}_3 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (f_2 f_3 f_4 - g_2 g_4) - g_1 g_3 f_4 \\ g_1 (f_2 f_3 f_4 - g_2 g_4) + f_1 g_3 f_4 \\ f_2 g_4 + g_2 (f_3 f_4) \end{pmatrix} \\
 & \begin{pmatrix} \text{RX}_4 \\ \text{RY}_4 \\ \text{RZ}_4 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (\prod_{i=2}^5 f_i - f_2 g_3 g_5 - g_2 g_4 f_5) - g_1 (f_3 g_5 + g_3 f_4 f_5) \\ g_1 (\prod_{i=2}^5 f_i - f_2 g_3 g_5 - g_2 g_4 f_5) + f_1 (f_3 g_5 + g_3 f_4 f_5) \\ g_2 ((f_3 f_4 f_5) - g_3 g_5) + f_2 g_4 f_5 \end{pmatrix} \\
 & \begin{pmatrix} \text{RX}_5 \\ \text{RY}_5 \\ \text{RZ}_5 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (\prod_{i=2}^6 f_i - f_2 (f_3 g_4 g_6 + g_3 g_5 f_6) - g_2 (f_4 g_6 + g_4 f_5 f_6)) - g_1 (f_3 g_5 f_6 - g_3 (g_4 g_6 - f_4 f_5 f_6)) \\ g_1 (\prod_{i=2}^6 f_i - f_2 (f_3 g_4 g_6 + g_3 g_5 f_6) - g_2 (f_4 g_6 + g_4 f_5 f_6)) + f_1 (f_3 g_5 f_6 - g_3 (g_4 g_6 - f_4 f_5 f_6)) \\ g_2 ((f_3 f_4 f_5 f_6) - f_3 (g_4 g_6) - g_3 g_5 f_6) + f_2 (f_4 g_6 + g_4 f_5 f_6) \end{pmatrix}
 \end{aligned}$$

n=6. / n=7 parte 1 y parte 2.

$$\begin{pmatrix} R_{x6} \\ R_{y6} \\ R_{z6} \end{pmatrix} = 1 \begin{pmatrix} f_1 (\prod_{i=2}^7 f_i - f_2 (f_3 (f_4 85 87 + 84 86 f_7) + 83 (f_5 87 + 85 f_6 f_7)) - 82 (f_4 86 f_7 + 84 (f_5 f_6 f_7 - 85 87))) - 81 (f_3 (f_5 87 + 85 f_6 f_7) - 83 (84 86 f_7 - f_4 (f_5 f_6 f_7 - 85 87))) \\ 81 (\prod_{i=2}^7 f_i - f_2 (f_3 (f_4 85 87 + 84 86 f_7) + 83 (f_5 87 + 85 f_6 f_7)) - 82 (f_4 86 f_7 + 84 (f_5 f_6 f_7 - 85 87))) + f_1 (f_3 (f_5 87 + 85 f_6 f_7) - 83 (84 86 f_7 - f_4 (f_5 f_6 f_7 - 85 87))) \\ 82 ((\prod_{i=3}^7 f_i) - f_3 (f_4 85 87 + 84 86 f_7) - 83 (f_5 87 + 85 f_6 f_7)) + f_2 (f_4 86 f_7 + 84 (f_5 f_6 f_7 - 85 87)) \end{pmatrix}$$

$$\begin{pmatrix} R_{x7} \\ R_{y7} \\ R_{z7} \end{pmatrix} = \begin{pmatrix} f_1 (\prod_{i=2}^8 f_i - f_2 (f_3 (f_4 (f_5 86 88 + 85 87 f_8) + 84 (86 f_7 f_8 + f_6 88)) + 83 (f_5 87 f_8 + 85 (f_6 f_7 f_8 - 86 88))) - 82 (f_4 (f_6 88 + 86 f_7 f_8) + 84 (f_5 (f_6 f_7 f_8 - 86 88) - 85 86 88)) \\ 81 (\prod_{i=2}^8 f_i - f_2 (f_3 (f_4 (f_5 86 88 + 85 87 f_8) + 84 (86 f_7 f_8 + f_6 88)) + 83 (f_5 87 f_8 + 85 (f_6 f_7 f_8 - 86 88))) - 82 (f_4 (f_6 88 + 86 f_7 f_8) + 84 (f_5 (f_6 f_7 f_8 - 86 88) - 85 86 88)) \\ 82 ((\prod_{i=3}^8 f_i) - f_3 (f_4 (f_5 86 88 + 85 87 f_8) + 84 (f_6 88 + 86 f_7 f_8)) - 83 (f_5 87 f_8 + 85 (f_6 f_7 f_8 - 86 88))) + f_2 (f_4 (f_6 88 + 86 f_7 f_8) + 84 (f_5 (f_6 f_7 f_8 - 86 88) - 85 87 f_8)) \end{pmatrix}$$

$$\begin{pmatrix} - 85 87 f_8 \\ - 85 87 f_8 \\ + f_2 (f_4 (f_6 88 + 86 f_7 f_8) + 84 (f_5 (f_6 f_7 f_8 - 86 88) - 85 87 f_8)) \end{pmatrix}$$

$$\begin{pmatrix} R_{x8} \\ R_{y8} \\ R_{z8} \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (\prod_{i=2}^n f_i - f_2 (f_3 (f_4 (f_5 (f_6 87 89 + 86 88 f_9) + 85 (f_7 89 + 87 f_8 f_9)) + 84 (86 (f_7 f_8 f_9 - 87 89) + f_6 88 f_9)) + 83 (f_5 (f_7 89 + 87 f_8 f_9) + 85 (f_6 (f_7 f_8 f_9 - 87 89) + 81 (\prod_{i=2}^n f_i - f_2 (f_3 (f_4 (f_5 (f_6 87 89 + 86 88 f_9) + 85 (f_7 89 + 87 f_8 f_9)) + 84 (86 (f_7 f_8 f_9 - 87 89) + f_6 88 f_9)) + 83 (f_5 (f_7 89 + 87 f_8 f_9) + 85 (f_6 (f_7 f_8 f_9 - 87 89) + 82 (\prod_{i=3}^n f_i) - f_3 (f_4 (f_5 (f_6 87 89 + 86 88 f_9) + 85 (f_7 89 + 87 f_8 f_9)) \cdot \\ + 85 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9))) - 82 (f_4 (f_6 88 f_9 + 86 (f_7 f_8 f_9 - 87 89)) + 84 (f_5 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9) - 85 (f_7 89 + 87 f_8 f_9))) - 81 (f_3 (f_5 (f_7 89 + 85 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9))) - 82 (f_4 (f_6 88 f_9 + 86 (f_7 f_8 f_9 - 87 89)) + 84 (f_5 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9) - 85 (f_7 89 + 87 f_8 f_9))) + f_1 (f_3 (f_5 (f_7 89 + 85 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9))) + f_2 (f_4 (f_6 88 f_9 + 86 (f_7 f_8 f_9 - 87 89) - 85 (f_7 89 + 87 f_8 f_9)) \cdot \\ - 81 (f_3 (f_5 (f_7 89 + 87 f_8 f_9) + 85 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9)) - 83 (84 (f_6 88 f_9 + 86 (f_7 f_8 f_9 - 87 89)) - f_4 (f_5 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9) - 85 (87 f_8 f_9 + \\ + f_1 (f_3 (f_5 (f_7 89 + 87 f_8 f_9) + 85 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9)) - 83 (84 (f_6 88 f_9 + 86 (f_7 f_8 f_9 - 87 89)) - f_4 (f_5 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9) - 85 (87 f_8 f_9 + \\ 86 (f_7 f_8 f_9 - 87 89)) + 84 (f_5 (f_6 (f_7 f_8 f_9 - 87 89) - 86 88 f_9) - 85 (f_7 89 + 87 f_8 f_9))) \\ - 85 (87 f_8 f_9 + f_7 89))) \\ - 85 (87 f_8 f_9 + f_7 89))) \end{pmatrix}$$

n=8.

$$\begin{pmatrix} Rx9 \\ Ry9 \\ Rz9 \end{pmatrix} = \mathbf{1} \begin{pmatrix} f_1 (\prod_{i=2}^{10} f_i - f_2 (f_3 (f_4 (f_5 (f_6 (f_7 88 810 + 87 89 f_{10}) + 86 (f_8 810 + 88 f_9 f_{10})) + 85 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810))) + 84 (86 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10} \\ 81 (\prod_{i=2}^{10} f_i - f_2 (f_3 (f_4 (f_5 (f_6 (f_7 88 810 + 87 89 f_{10}) + 86 (f_8 810 + 88 f_9 f_{10})) + 85 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810))) + 84 (86 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10} \\ 82 ((\\) \end{pmatrix}$$

$$\begin{aligned}
 & - 87 89 f_{10}) + f_6 (f_8 810 + 88 f_9 f_{10})) + 83 (f_5 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810)) + 85 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 82 (f_4 (f_6 \\
 & - 87 89 f_{10}) + f_6 (f_8 810 + 88 f_9 f_{10})) + 83 (f_5 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810)) + 85 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 82 (f_4 (f_6 \\
 & 82 (\prod_{i=3}^{10} f_i) - f_3 (f_4 (f_5 (f_6 (f_7 88 810 + 87 89 f_{10}) + 86 (f_8 810 + 88 f_9 f_{10})) + 85 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810))) + 84 (f_6 (f_8 810 + 88 f_9 f_{10}) + 86 (f_7 (f_8 f_9 \\
 & - 82 (f_4 (f_6 (f_8 810 + 88 f_9 f_{10}) + 86 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) + 84 (f_5 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 85 (f_7 89 f_{10} + 87 \\
 & - 82 (f_4 (f_6 (f_8 810 + 88 f_9 f_{10}) + 86 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) + 84 (f_5 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 85 (f_7 89 f_{10} + 87 \\
 & 86 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 83 (f_5 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810)) + 85 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) + f_2 (f_4 (f_6 \\
 & - 85 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810))) - 81 (f_3 (f_5 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810)) + 85 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 86 (88 f_9 f_{10} + f_8 810))) - 83 \\
 & - 85 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810))) + f_1 (f_3 (f_5 (f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810)) + 85 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 86 (88 f_9 f_{10} + f_8 810))) - 83 \\
 & f_{10})) + f_2 (f_4 (f_6 (f_8 810 + 88 f_9 f_{10}) + 86 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) + 84 (f_5 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 85 (f_7 89 \\
 & - 83 (84 (f_6 (f_8 810 + 88 f_9 f_{10}) + 86 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - f_4 (f_5 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 85 (87 (f_8 f_9 f_{10} \\
 & - 83 (84 (f_6 (f_8 810 + 88 f_9 f_{10}) + 86 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - f_4 (f_5 (f_6 (f_7 (f_8 f_9 f_{10} - 88 810) - 87 89 f_{10})) - 85 (87 (f_8 f_9 f_{10} \\
 & f_7 89 f_{10} + 87 (f_8 f_9 f_{10} - 88 810))))))
 \end{aligned}$$

$$\begin{aligned}
 & - 85 (87 (f_8 f_9 f_{10} - 88 810) + f_7 89 f_{10})) \\
 & - 85 (87 (f_8 f_9 f_{10} - 88 810) + f_7 89 f_{10}))
 \end{aligned}$$

Probando que cada expresión cumpla con la ecuación 34, se da por sentado que $l > 0$.

Vector 1.

```
In[61]=
d =  $\sqrt{R_{x_1}^2 + R_{y_1}^2 + R_{z_1}^2}$ 
Simplify[d]
Out[61]=  $\sqrt{1^2 \cos^2[q_1] \cos^2[q_2] + 1^2 \cos^2[q_2] \sin^2[q_1] + 1^2 \sin^2[q_2]}$ 
Out[62]=  $\sqrt{1^2}$ 
```

Vector 2.

```
In[63]=
d =  $\sqrt{R_{x_2}^2 + R_{y_2}^2 + R_{z_2}^2}$ 
Simplify[d]
Out[63]=  $\sqrt{1^2 \cos^2[q_3] \sin^2[q_2] + 1^2 (\cos[q_2] \cos[q_3] \sin[q_1] + \cos[q_1] \sin[q_3])^2 + 1^2 (\cos[q_1] \cos[q_2] \cos[q_3] - \sin[q_1] \sin[q_3])^2}$ 
Out[64]=  $\sqrt{1^2}$ 
```

Vector 3.

```
In[65]=
i = 3
d_i =  $\sqrt{R_{x_i}^2 + R_{y_i}^2 + R_{z_i}^2}$ 
Simplify[d_i]
Out[65]=  $\sqrt{1^2}$ 
```

Vector 4.

```
In[66]=
i = 4
d_i =  $\sqrt{R_{x_i}^2 + R_{y_i}^2 + R_{z_i}^2}$ 
Simplify[d_i]
```

```
In[67]=
i = 5
d_i =  $\sqrt{R_{x_i}^2 + R_{y_i}^2 + R_{z_i}^2}$ 
Simplify[d_i]
```

Out[67]= 5

Out[68]= $\sqrt{1^2}$

Anexo 3. Reporte técnico para obtener la relación expresada en la ecuación 21, de forma cerrada.

Diseño de Algoritmo de Locomoción para la Cinemática Directa de un Robot Ápodo en \mathbb{R}^3

Por. Ing Carlos Alberto Martínez Gómez

1. INTRODUCCIÓN

Los robots ápodos, tienen su inspiración en las serpientes biológicas. Ambos, deben su proceso de locomoción a las curvas serpenoides [1], las cuales cumplen con la función de generar ondulaciones a lo largo de toda la cadena de articulaciones, a modo de que se genere un desplazamiento a lo largo del espacio.

Este tipo de locomoción busca aprovechar las características morfológicas y cualidades del desplazamiento de las serpientes biológicas, hacen de este diseño de robots una propuesta para resolver diversos problemas en ingeniería y servicios, relacionados con la búsqueda y rescate de personas, exploración de ductos, tanques y entornos hostiles al ser humano.

El primer paso dentro del desarrollo de un sistema robótico móvil, en este caso de un robot modular, antes de llegar a un modelo físico, es el diseño del algoritmo de locomoción, es decir, el cómputo de datos que debe realizarse, a fin lograr una secuencia en el patrón de desplazamiento del robot, que cumpla con ciertos criterios de movimiento, en los robots ápodos, los criterios de estabilidad son:

$\Delta x > 0$: Después de un ciclo de locomoción haber desplazamiento sobre el eje X, constante y mayor a cero.

Criterio de estabilidad: El robot debe tocar al menos dos puntos con el suelo.

El presente trabajo, detalla el diseño de un algoritmo para un robot ápodo de múltiples grados de libertad, en su etapa teórica. Empleando relaciones de recurrencia y álgebra vectorial.

3. DESARROLLO DEL ALGORITMO DE LOCOMOCIÓN

1. Secuencia $\varphi(t)$

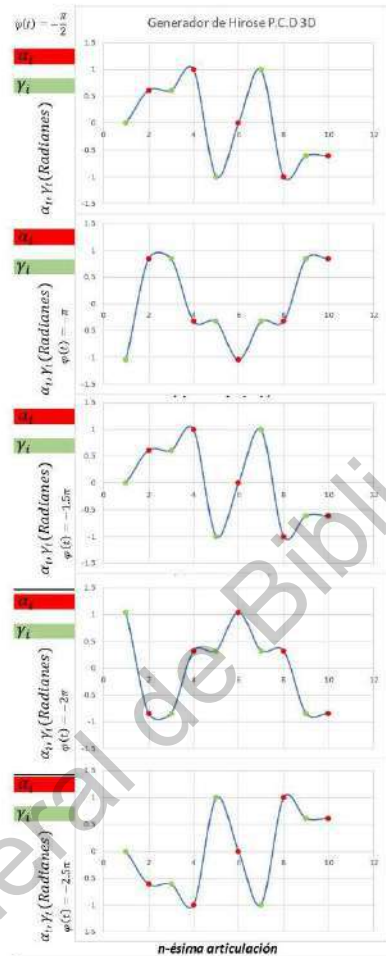
El primer paso, es definir una secuencia en función del tiempo, que nos permita generar ángulos que tomarán cada una de las articulaciones y el robot ápodo comience a "serpentear".

Se propondrá la secuencia presentada en la ecuación (1), que será introducidas en el generador de Hirose, empleándolas similarmente para el caso unidireccional.

2. Generador de Hirose.

Los ángulos obtenidos, son introducidos en el generador sinusoidal de Hirose [3]. Y de igual forma, se requiere que estos sean mutuamente independientes, debido a que uno servirá para las articulaciones nones, y otro para las pares

La diferencia de los generadores puede apreciarse en la gráfica (2).



Gráfica 2: Evolución de los generadores de Hirose.

Sin embargo, los ángulos generados por la ecuación (2) y (3), deben ser transformados, ya que estos están dados con respecto a un sistema de referencia absoluto (x_0, y_0, z_0) . Mientras que los ángulos q_i en el que expresamos la solución del P.C.D, son ángulos que consideran la rotación de la articulación i , dada una rotación de la articulación $i-1$, debemos hallar una interacción, en que estos deban expresarse en el sistema coordinado absoluto, dada por las ecuaciones (22) y (23)

Si n es non:

$$q_i = f(\alpha_i) \tag{22}$$

Si n , es par:

$$\tag{23}$$

$$q_i = f(\gamma_i)$$

Si Descomponemos, en coordenadas rectangulares el vector en \mathbb{R}^3 , siguiendo la regla de la mano derecha, tendremos nuestro espacio euclídeo en \mathbb{R}^3 , como se muestra en la figura (4).

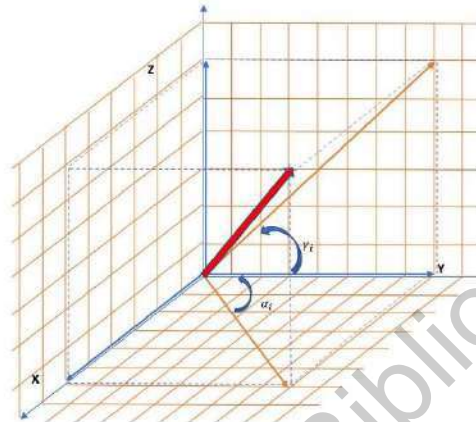


Figura 4: Descomposición rectangular de un vector en el espacio.

Cuya descomposición rectangular, dada por la ecuación (24).

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = l \begin{pmatrix} \sin(\gamma_i) \cos(\alpha_i) \\ \sin(\gamma_i) \sin(\alpha_i) \\ \cos(\gamma_i) \end{pmatrix} \quad (24)$$

Adicionalmente, Sea la terna $Q_x(q_i), Q_y(q_i), Q_z(q_i)$

Que expresa, las coordenadas X, Y, Z, de la i-ésima articulación, en función de los ángulos q_i , previamente establecidos en la convención Denavit-Hartenberg.

Entonces, el cambio de sistema coordenado, es básicamente hallar las funciones planteadas en (24) y (25.)

en el caso de $i=1$, e $i=2$, tendremos soluciones triviales, mientras que para $i=3$, el problema se complica.

$$l \begin{pmatrix} \sin(\gamma_i) \cos(\alpha_i) \\ \sin(\gamma_i) \sin(\alpha_i) \\ \cos(\gamma_i) \end{pmatrix} = l \begin{bmatrix} f_1 \\ g_1 \\ 0 \end{bmatrix} \quad (25)$$

ya que i , es non, se desea conocer la interacción entre q_1 y α_1 . Entonces, formando un cociente entre Q_x y Q_y , tendremos

$$\tan(\alpha_1) = \tan(q_1) \Rightarrow \alpha_1 = q_1$$

para $i=2$, usando el mismo cociente, llegamos a la identidad:

$$\alpha_2 = q_1 \quad (26)$$

y usando Q_2

$$\begin{aligned} \cos(\gamma_2) &= g_2 \\ q_2 &= a \sin(\cos(\gamma_2)) \end{aligned} \quad (27)$$

en $n=3$, el problema se complica, podemos notar, que las ecuaciones quedan de la forma.

$$\begin{cases} \sin(\gamma_i) \cos(\alpha_i) \\ \sin(\gamma_i) \sin(\alpha_i) \\ \cos(\gamma_i) \end{cases} = \begin{cases} f_1(1 + f_2 + f_2 f_3) - g_1 g_3 \\ g_1(1 + f_2 + f_2 f_3) + f_1 g_3 \\ g_2(1 + f_3) \end{cases} \quad (28)$$

Expresamos el segundo término en función del ángulo q_3 que es el que queremos despejar.

$$\begin{cases} \sin(\gamma_i) \cos(\alpha_i) \\ \sin(\gamma_i) \sin(\alpha_i) \\ \cos(\gamma_i) \end{cases} = \begin{cases} A_1 f_3 + B_1 g_3 + k_1 \\ A_2 f_3 + B_2 g_3 + k_2 \\ A_3 f_3 + B_3 g_3 + k_3 \end{cases} \quad (29)$$

Si usamos el cociente con Q_x y Q_y , tenemos

$$\tan(\alpha_i) = \frac{A_2 f_3 + B_2 g_3 + k_2}{A_1 f_3 + B_1 g_3 + k_1} \quad (30)$$

Si acomodamos la ecuación en la forma:

$$A f_3 + B g_3 = C$$

e introduciendo la identidad trigonométrica

$$f_3 = \sqrt{1 - g_3^2}$$

Y el cambio de variable

$$x = g_3$$

tendremos la ecuación cuadrática:

(31)

$$-(A^2 + B^2)X^2 + (2BC)X + (A^2 - C^2) = 0$$

resolviendo las raíces y dejando en función de q_3

$$q_3 = \text{asin}\left(\frac{BC \mp A\sqrt{A^2 + B^2 - C^2}}{A^2 + B^2}\right) \quad (32)$$

Donde

$$A(\alpha_3) = A_1 \tan(\alpha_3) - A_2$$

$$B(\alpha_3) = B_1 \tan(\alpha_3) - B_2$$

$$C(\alpha_3) = K_1 \tan(\alpha_3) - K_2$$

Por tanto, en forma general, si se desea conocer la relación entre α_i y q_i , tenemos que dejar el vector Q, como esta expresado en (32). Quedando la generalización expresada en la ecuación (33).

$$q_{2i+1} = \text{asin}\left(\frac{BC \mp A\sqrt{A^2 + B^2 - C^2}}{A^2 + B^2}\right) \quad (33)$$

Donde:

$$A(\alpha_i) = A_2 \tan(\alpha_i) - A_1$$

$$B(\alpha_i) = B_2 \tan(\alpha_i) - B_1$$

$$C(\alpha_i) = K_2 \tan(\alpha_i) - K_1$$

Para las articulaciones pares, trabajando con la componente Q_z tenemos

$$q_{2i} = \text{asin}\left(\frac{B_3 C \mp A_3 \sqrt{A_3^2 + B_3^2 - C^2}}{A_3^2 + B_3^2}\right) \quad (34)$$

Donde:

$$C(\alpha_i) = K_3 - \cos(\gamma_i)$$

Por tanto, si calculamos q_{i-1} , podemos conocer el siguiente elemento, ya que todos los términos de cualquier vector Q, estarán en función del sistema coordenado absoluto.

Por tanto, el modelado matemático sobre el cual se construirá el algoritmo de locomoción, ha sido concluido, por tanto, el diagrama de flujo resultante, se muestra en la figura 5.

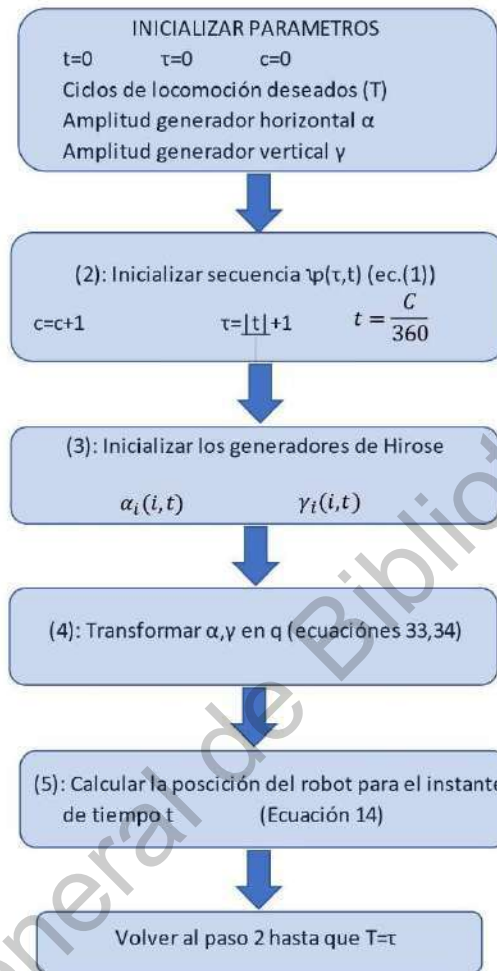


Figura 5: Diagrama de flujo del algoritmo de locomoción obtenido.

BIBLIOGRAFIA:

- [1] S. Hirose, and H. Yamada. "Snake-like robots: Machine design of biologically inspired robots". IEEE Robotics and Automation Magazine, 16(1), pp.88–98.2019.
- [2]: Martínez Gómez Carlos Alberto, Vargas Soto José Emilio, Pérez Soto Gerardo Israel, Ramírez Granados Luisa, Sandoval Castro Xóchitl Yamile, *Algoritmo para Resolver la Cinemática Directa de un Robot tipo Ápodo con Articulaciones Rotacionales*, La mecatrónica en México, Vol. 8, No. 1, 2018.
- [3]: Sandoval, X. Castillo E., *Desarrollo de un Robot Gusano Capaz de Desplazarse en Terreno Natural*. Robótica y Mecatrónica, Asociación Mexicana de Mecatrónica, 1ª edición, 2018

Anexo 4. Publicaciones realizadas en base a 1. Desarrollo del algoritmo de locomoción, 2. Mejora del algoritmo.

Se presenta el encabezado, así como los resultados y conclusión del artículo.

Artículo completo en: <http://www.mecamex.net/revistas/LMEM/revistas/LMEM-V08-N01-03.pdf>

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 37 – 50.
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C



Algoritmo para Resolver la Cinemática Directa de un Robot tipo Ápodo con Articulaciones Rotacionales

Martínez Gómez Carlos Alberto⁽¹⁾, Vargas Soto José Emilio⁽¹⁾, Pérez Soto Gerardo Israel⁽¹⁾, Ramírez Granados Luisa⁽¹⁾, Sandoval Castro Xóchitl Yamile⁽²⁾,

Universidad Autónoma de Querétaro, Campus Cerro de las Campanas⁽¹⁾
Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Unidad Querétaro⁽²⁾
✉ carlos.martho@hotmail.com

Resumen

Los robots ápodos poseen una estructura morfológica y principios de locomoción inspirados en las serpientes biológicas. En años recientes han originado líneas de investigación que buscan aprovechar sus cualidades en el desplazamiento a través de superficies irregulares y con espacios reducidos con la finalidad de emplearlos en diversas tareas como la localización de personas atrapadas en sismos, caracterización del entorno, inspección de ductos y plantas de generación de energía. El presente trabajo propone un algoritmo para resolver la cinemática directa de un robot móvil ápodo de múltiples grados de libertad con desplazamiento unidireccional, empleando propiedades matemáticas, como las relaciones de recurrencia y funciones continuas a trazos, presentando como resultados un modelo matemático que generaliza la solución para n articulaciones, su implementación en código C++ y la simulación del movimiento del robot.

Palabras clave: Robot, Algoritmo, Locomoción, Cinemática, Modelo Matemático, implementación.

Abstract

Snake robots have a morphological structure and locomotion principles inspired in biological snakes. In recent years, they have originated research lines that search for take advantage of their qualities for displace through irregular surfaces and narrow spaces having a purpose to use them in many tasks as localization of people trapped in earthquakes, characterization of the environment, duct inspection and safety of power generation plants. This work proposes an algorithm to solve the problem of direct kinematics problem of a unidirectional displacement multiple degrees of freedom snake robot, using mathematical properties like recurrence relations and piecewise functions, presenting as results a mathematical model that generalizes the solution for n -articulations, its implementation in C++ code and movement simulation of robot.

Keywords: Robot, Algorithm, Locomotion, Kinematics, Mathematical Model, Implementation.

1. Introducción

La robótica de servicios se enfoca en desarrollar robots que ayuden al ser humano en labores no necesariamente industriales, la Federación Internacional De Robótica (IFR) Define a un robot de servicios como "Robot que opera de forma parcial o totalmente autónoma al servicio del bienestar de los seres humanos y de equipamientos, excluyendo operaciones manufactureras" [1].

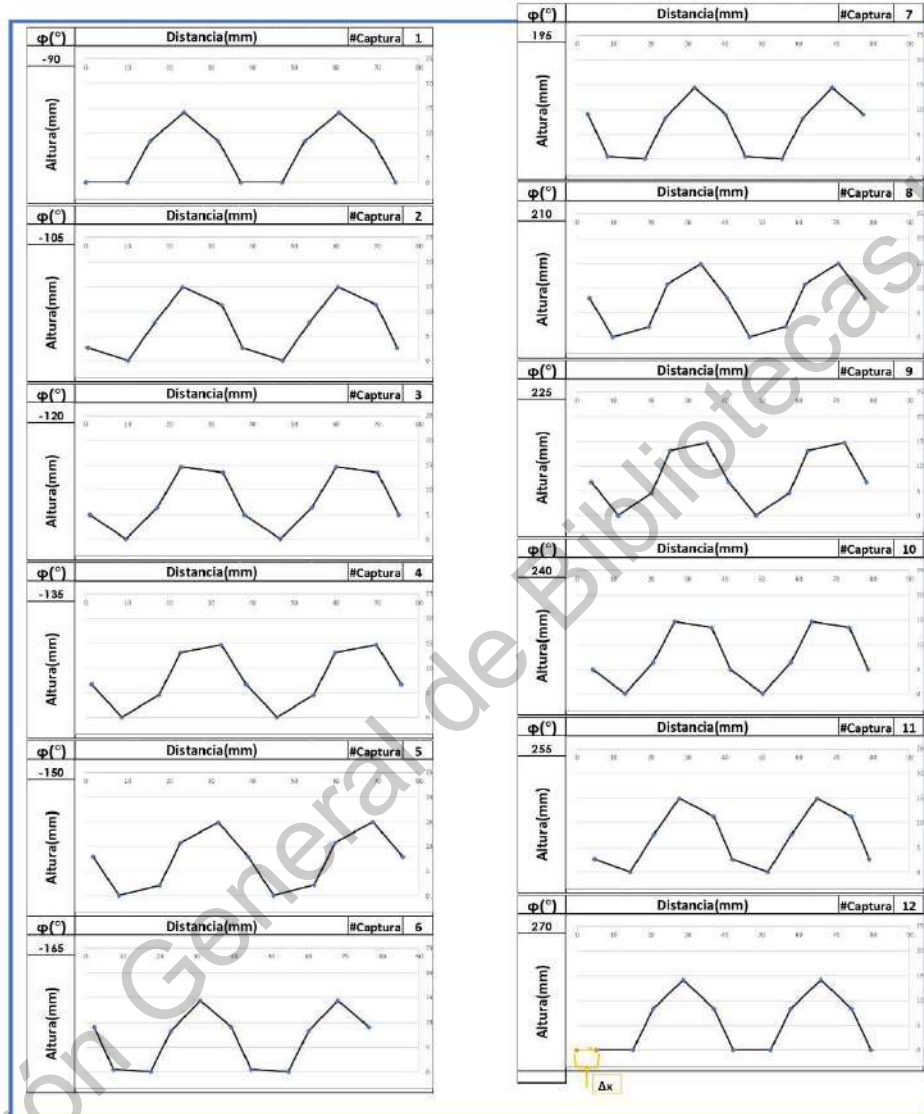


Figura 10: Simulación de la locomoción del robot en un ciclo.

3.1 Análisis del error:



Debido a que el algoritmo no toma el valor exacto de π , es de esperar un error de aproximación, y otro ocasionado por la discretización de ondas sinusoidales, como se presenta en [8]. La figura 11, muestra el valor esperado mediante cálculo, de la posición de la última articulación del robot, con respecto al obtenido en el algoritmo implementado.

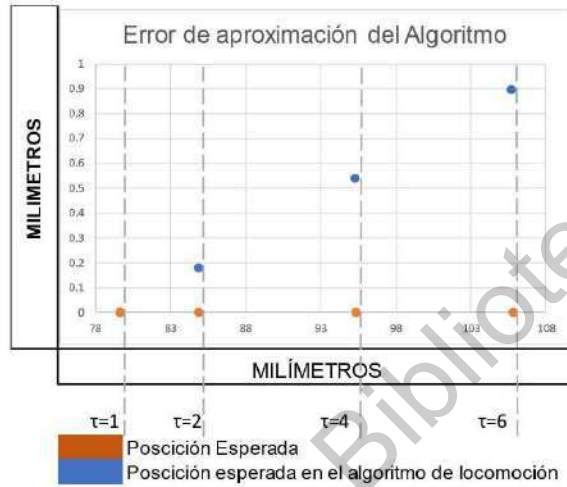


Figura 11: Comparativa entre valor esperado y el obtenido por el algoritmo.

Después del primer ciclo de trabajo, comienza a generarse un error, en forma de desfaseamiento, horizontal y vertical, estos son lineales respecto al ciclo de trabajo τ , y mediante ecuaciones de la recta, se presentan de la siguiente manera:

$$\begin{bmatrix} X_{alg} \\ Y_{alg} \end{bmatrix} = \begin{bmatrix} X_{esp} - 0.0291\tau \\ Y_{esp} + 0.01791\tau \end{bmatrix} \quad (20)$$

Donde:

- X_{alg} = Ubicación en X, obtenida por el algoritmo de locomoción, del último eslabón
- Y_{alg} = Ubicación en Y, obtenida por el algoritmo de locomoción, del último eslabón
- X_{esp} = Ubicación esperada en X, del último eslabón del robot.
- y_{esp} = Ubicación esperada en Y, del último eslabón del robot.

3.2 Tabla de resumen de variables y parámetros.

Símbolo	Tipo.	Definición.	Unidades	Domínio.
α_i	Variable	Ángulo de la articulación i, con respecto al eje x.	Radianes.	$\{0, \frac{2\pi}{3}\}$
α	Parámetro	Amplitud de onda	Radianes.	$\{0, \frac{2\pi}{3}\}$
$\Phi(c)$	Función	proporciona el ángulo Φ , para cada instante t	Radianes.	$\{\frac{\pi}{2}; -\pi, \pi; \frac{3\pi}{2}\}$
K	Parámetro	Numero de ondulaciones.	Escalar.	$K > 2$
M	Parámetro	Número de articulaciones.	Escalar.	$M \in \mathbb{Z}^+, M > 2.$



L	Parámetro	Longitud del Eslabón.	Milímetros	$l \in \mathbb{Q}^+$
q_i	Variable	Angulo de la articulación i, con respecto a la articulación i-1.	Radianes.	$\{-\pi, \pi\}$
\vec{x}	Vector	Vector que depende del instante t y de la constante de paso.	Milímetros	$\{0; \frac{l\alpha}{k}\}$
Δ_x	Parámetro	Avance del robot en un ciclo de trabajo.	Milímetros	$\frac{l\alpha}{k}$
t(c)	Función	Instante t en que se encuentra el ciclo de locomoción.		$\{0; T\}$
C	Variable	Contador	Escalar	$\{0; 180\}$
τ	Constante	Ciclo de trabajo.	Escalar	$\tau \in \mathbb{Z}^+, \tau > 1.$
T	Constante	Ciclos de trabajo a reproducir por el algoritmo.	Escalar	$T \in \mathbb{Z}^+, T > 1.$

Tabla 2. Resumen de Variables y Parámetros Implementados en el Modelo

4. Conclusiones

Durante el trabajo presentado, se desarrolló conocimiento matemático, para entender cómo se comporta este tipo de robots, lo cual permitió su validación mediante un diagrama de flujo y su implementación en código C++, aportando un modelo propuesto para describir la locomoción del robot, proponiendo una secuencia para el generador sinusoidal que disminuye el rango de iteraciones comparado con otros métodos presentados.

Actualmente se está desarrollando bajo esta misma metodología un modelo y un algoritmo de locomoción que logre desplazar un robot ápedo en el espacio, en una superficie con inclinación, por lo que este trabajo sirvió como parteaguas para comprender e implementar las curvas serpenoides. Competencias útiles para trabajar con el modelo en el espacio que es, por definición más complejo de obtener y validar.

Referencias

- [1] International Federation of Robotics. (2016). Service Robotics. Sitio Web: www.ifr.org/service-robots [Último acceso: febrero de 2019].
- [2] Darwin, C. (2009). El origen de las especies por medio de la selección natural. Editorial CSIC-CSIC Press. 1ª edición, Madrid, 2009.
- [3] Hirose, S., & Yamada, H. Snake-like robots: "Machine design of biologically inspired robots". IEEE Robotics and Automation Magazine, 16(1), 88–98. 2019.
- [4] Lilywhite, H. (2014). How Snakes Work: Structure, Function and Behavior of the World's Snakes. OXFORD. University Press, Nueva York, 2014.
- [5] Esta robo-serpiente buscó sobrevivientes tras el sismo de la CDMX. (04 de octubre de 2017). *El Financiero*. p.1
- [6] Atherton, K. (15 de diciembre de 2015). Snake Robot Slithers Through Pipes, Laser-Welds From Within. *Popular Science*. Recuperado de: <https://www.popsci.com/laser-welding-snake-robot-wants-to-repair-pipes>.
- [7] SINTEF. (2016) A giant subsea snake robot. Sitio Web: <https://www.sintef.no> [Último acceso : Marzo de 2019].
- [8] Sandoval, X. Castillo E., "Desarrollo de un Robot Gusano Capaz de Desplazarse en Terreno Natural". Robótica y Mecatrónica, Asociación Mexicana de Mecatrónica, 1ª edición, 2018
- [9] González, J. (2008). *Robótica Modular y Locomoción: Aplicación a Robots Ápedos* (Tesis Doctoral). Universidad Autónoma de Madrid, Madrid, España
- [10] Garzón, M. (2009). *Estrategias Bio-Inspiradas para Locomoción de Robots Ápedos*. (Tesis De Maestría). Universidad Politécnica de Madrid, Madrid, España.

Subaquatic inspection: localization of fissures in oil tanks or any toxic, flammable, or corrosive materials, are activities that imply risk to the human, Snake robots has been developed in way to displace in those environments and realize inspection.

Search and rescue: Snake robots are a potential tool in inspection tasks in disaster zones, like inundations, earthquakes, in the way of search and rescue trapped people in rubble.

Snake robots are characterized by employing a movement pattern bio-inspired in biological snakes, one of first locomotion algorithms was proposed by Shigeo Hirose in 1972 and implemented in ACM-III, first snake robot with wheeled legs [9].

Mathematical model that describes the movement of a biological snake depends on its morphological structure, movement of a continuous snake robot or similar mechanisms like the Elephant trunk [10]. Even biological snakes, is described by Serpenoid Curve, which is shown in equation (1).

$$K(s) = \frac{-2\pi k}{l} \alpha \sin\left(\frac{2\pi k}{l} S\right) \quad (1)$$

Where:

- l : length of the curve [Cm].
- S : Distance through the curve [Cm].
- k : Undulations, $k > 0$
- α : Serpentine angle [rad].

This function generates "serpenoid curves", which introduced in the direct kinematics solution, implies determinate position of the joints of the robot in a t instant.

Structure of snake robots, can be continuous if distance between joints is very small, or discrete, whose links are separated by joints. Morphological differences can be appreciated in figure 2.

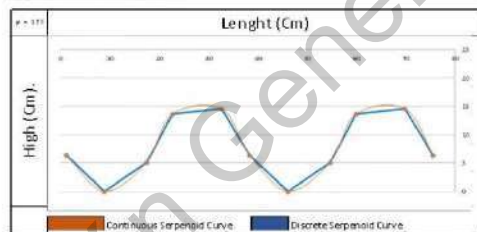


Fig. 2. Morphological Structure Kinds in IR Snake Robots.

Mathematical model for discrete snake robots is presented [11], and showed in equation (2).

$$\alpha_i(\varphi) = \alpha \cos\left(\varphi + \frac{2\pi k}{M}(i-1)\right) \quad (2)$$

Where:



$\alpha_i(\varphi)$: Angle of the i -th joint [rad].

φ : Phase shifting.

M : total of number of joints, in this case 10.

i : i -th joints.

There are 2 kinds of locomotion, which depend in dimensions employed. 1R locomotion is the movement, that take place in the X, Y plane. By the otherwise, we have 2R locomotion, in this, the movement that take place in 3D Space.

In this document only the problem of one-dimensional locomotion non continuous robots will be addressed.

Present work will introduce to the reader to the development of a mathematical model for snake robot locomotion, proposing it to solve problems related to locomotion in narrow or dangerous spaces.

II. DEVELOPMENT OF MATHEMATICAL MODEL

This problem is addressed as the development of a mechatronic project, we will use a methodology based in [12].

A. 2.1 Data Collect and synthesis of information

In this section we collected information about mathematical models that describes the movement of Snake robots with morphological discrete structure.

The development of a locomotion algorithm for a snake robot is based in the next definitions:

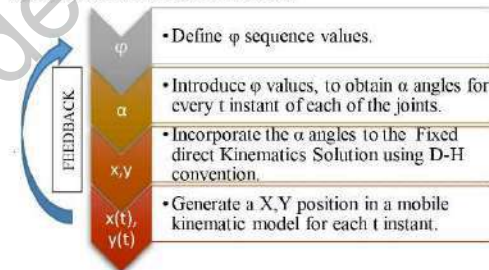


Fig. 3. Architecture of the mathematical model

On the first step, a sequence for a phase shifting needs to be defined, each value of the angle φ , create a different configuration of the robot through a locomotion cycle.

After, sinusoidal generator must be implemented, in the way to generate all the α angles for every joint, this generator must be discrete for this study case.

Third step implies to incorporate this angle to the kinematic model, obtained by Denavit-Hartenberg convention and homogeneous transformation matrices.

Algorithm Improvement Proposal to Solve the Direct Kinematics of a 10 Degrees of Freedom Snake Robot

Carlos Alberto Martínez-Gómez y José Emilio Vargas Soto
 Universidad Autónoma de Querétaro
 Facultad de Ingeniería- DIPFI.
 Querétaro, Mexico.
 carlos.martho@hotmail.com

Abstract—Morphological structure of Snake Robots, represents an advantage related to displacement on irregular surfaces. Researches on this kind of robots are in purpose to use them in tasks related to inspection in pipelines, danger zones or find people trapped in earthquakes. In this way, we propose an improvement of the locomotion algorithm, with respect to past works, employing a different strategy that reduce the computational cost to solve the Direct Kinematics Problem, we present a proposal of a mathematical model and its simulation. Analyzing results with the new algorithm and conclusions.

Keywords—Algorithm, Snake Robot, Locomotion, Kinematics.

I. INTRODUCTION

Service and Industrial Robotics started in first half of the twentieth century, in the industry expansion period, in which production lines, became more complex and simultaneous process were necessary, and for these activities, Manpower was not sufficient [1].

First industrial robot was developed by Bill Griffith Taylor in 1937, this robot worked through perforated plate which activates solenoids that controls all the mechanism [2].

Actually, ISO 8373 norm, classifies an industrial robot, all those who fulfill with the quality of being "An automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications." [3].

On the other hand, service robotics, focus on assists in no necessary industrial tasks, International Federation of Robotics (IFR) Define a service robotics all those "robots hat performs useful tasks for humans or equipment excluding industrial automation" [4].

This kind of robots can be fixed or mobiles, fixed robots, have been useful in tasks that high precision has required like surgery assistant, health care, disease diagnostic, support for people with different abilities.

Mobile Service robots are useful also, it's capacity for moving through space, facilitates the accomplishments of tasks, that the transit of the human being is not feasible, for dangerous environment or complicated displacement surface.

Example of applications of the mobile robots, are space exploration, radioactive areas, walking robots of volcanic zones, geriatric assistant, care of people with disabilities, home duties, soil and environment characterization, and rescue of people.

Mobile robot can be classified for its locomotion pattern, which can be:

- On wheels.
- Caterpillar / track belts.
- Legged robots.
- No legged robots
- Other.

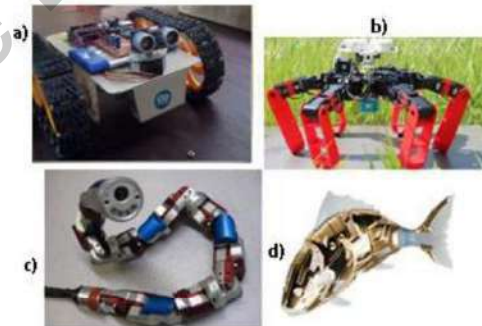


Fig. 1. kinds of locomotion: a) Track belt wheeled robot [5], b) Hexapod walking robot [6], c) snake robot [7], d) fish robot [8]

Study of no legged robot has created research lines, because this morphology, is very useful to solve many problems in industry and services, due to this robot can displace in irregular surfaces, or narrow spaces.

Most of the applications of no-legged robots are:

Pipeline inspection: ability to displace in narrow spaces, bend its body in trajectory changes, and move through very narrow spaces, in the same way than a biological snake, has made these robots useful to explore pipelines, searching failures inside.



At the end, is necessary to introduce the model in other that calculates its position for each t instant so that the model shows a displacement after a locomotion cycle. Feedback in every step is always possible.

The restrictions that validate a locomotion algorithm for these robots are:

- a) The locomotion pattern needs to be kinematic stable. This is one of the fundamental criteria in locomotion of robots: prevent the robot from falling. To comply this, the snake robot must have 2 points of its body on movement surface at least.
- b) The movement must form a recurrent pattern of locomotion.
- c) After a locomotion cycle, the robot must have the same configuration but to show a displacement $\Delta_x > 0$.

B. 2.2 Diagnosis and decision

After this process of data collection and information processing, implement the mathematical model is the next step, in order to improve our understanding of the displacement of this type of robots and verify the possibility of generating a more efficient locomotion algorithm.

In order to achieve these objectives, methodology established in figure (3) was followed.

C. 2.3 Model Development

1) Phase shifting

Model implemented in [11]. Propose the phase shifting seen in figure 5, and expressed in equation (3). φ Sequence values, have next mathematical form:

$$\varphi(t, \tau) = \begin{cases} -2\pi t - \frac{\pi}{2} + (\tau - 1)2\pi; & \tau - 1 \leq t < \tau - \frac{3}{4} \\ -2\pi t + \frac{3\pi}{2} + (\tau - 1)2\pi; & \tau - \frac{3}{4} \leq t < \tau \end{cases} \quad (3)$$

Where:

T: locomotion cycle.

t: Instant of time.

φ : Displacement angle in equation (2).

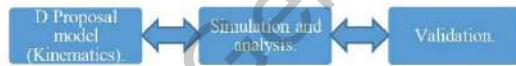


Fig. 4. Methodology to develop and validate the model.

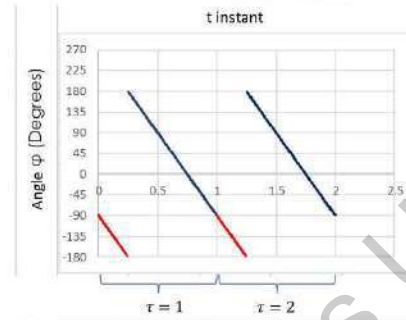


Fig. 5. Sequence of the phase shifting φ angle

From this sequence values this function takes equivalent values in the range of trigonometric functions.

Going from 0 to $-\pi/2$, is the same than going from π to $\pi/2$, so we decide to propose a new sequence in which we can save the computation time. The equation that describes our proposal improvement are described in (4).

$$\varphi(t, \tau) = \begin{cases} -\pi t - \frac{\pi}{2} + (\tau - 1)\pi; & \tau - 1 \leq t < \tau - \frac{1}{2} \\ \pi t + \frac{\pi}{2} - (\tau - 1)\pi; & \tau - \frac{1}{2} \leq t < \tau \end{cases} \quad (4)$$

In this function it is notable that de piecewise functions are a basis from the space (T, α) . Which is important in numerical analysis.

2) Sinusoidal generator

Once a working sequence has been established for the φ angle, this need to be inserted in discrete sinusoidal generator described in equation (2), with the aim of generating the angles of each of the joints, pattern of α angles, is described in figure 5, graphic is useful for only a t instant through a φ value.

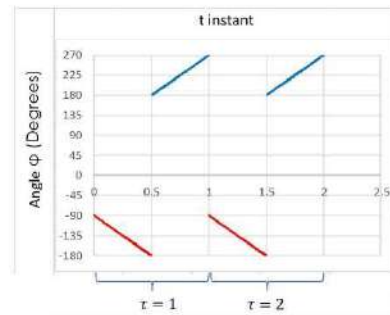


Fig.6. New sequence of face shifting.



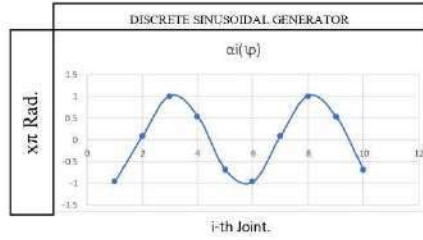


Fig. 7. Discrete sinusoidal generator at t instant.

3) Fixed kinematic model

Using D-H convention, is necessary to solve the kinematic model, this robot can be described like a union of many modules with a common, revolution axis.

For $M=1$:

$${}^0H_1 = \begin{bmatrix} \cos(q_1) & -\text{sen}(q_1) & 0 & l * \cos(q_1) \\ \text{sen}(q_1) & \cos(q_1) & 0 & l * \text{sen}(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Where:

q_i : i-th angle of the joint
 l : length of the link.

For $M=2$:

$${}^0H_2 = \begin{bmatrix} \cos(q_1+q_2) & -\text{sen}(q_1+q_2) & 0 & l[\cos(q_1)+\cos(q_1+q_2)] \\ \text{sen}(q_1+q_2) & \cos(q_1+q_2) & 0 & l[\text{sen}(q_1)+\text{sen}(q_1+q_2)] \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

For $M=3$:

$${}^0H_3 = \begin{bmatrix} \cos(Q) & -\text{sen}(Q) & 0 & l[\cos(q_1)+\cos(q_1+q_2)+\cos(Q)] \\ \text{sen}(Q) & \cos(Q) & 0 & l[\text{sen}(q_1)+\text{sen}(q_1+q_2)+\text{sen}(Q)] \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Where:

$$Q = q_1 + q_2 + q_3 \quad (8)$$

Solution of kinematics starts to form a recurrence pattern, through this, we can generalize that direct kinematic solution of a planar snake robot is given by the next equation:

$${}^0H_M = \begin{bmatrix} \sum_{i=1}^M \cos(q_i) & -\sum_{i=1}^M \text{sen}(q_i) & 0 & l \sum_{i=1}^M \cos(\sum_{j=1}^i q_j) \\ \sum_{i=1}^M \text{sen}(q_i) & \sum_{i=1}^M \cos(q_i) & 0 & l \sum_{i=1}^M \text{sen}(\sum_{j=1}^i q_j) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Therefore, indicates the position of each one of the joints is given by equation (10).

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} l \sum_{i=1}^M \cos(\sum_{j=1}^i q_j) \\ l \sum_{i=1}^M \text{sen}(\sum_{j=1}^i q_j) \end{bmatrix} \quad (10)$$

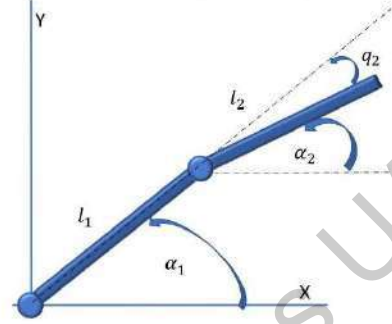


Fig. 8. 2 D.O.F mechanism.

In this point, is necessary to establish a relation between q_i and α_i , while the first one is the relative angle between the i-th joint and the previous one, the second one is the absolute angle formed between the i-th joint and the horizontal axis, that relations are showed in figure 8.

Analyzing the 2 D.O.F. (Degrees of Freedom) mechanism in figure 8, next relation is inferred:

$$q_2 = \alpha_2 - \alpha_1 \quad (11)$$

After experimentation, this generalization is assumable:

$$q_i = \alpha_i - \alpha_{i-1} \quad (12)$$

Substituting (12) in (11), we have:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} l \sum_{i=1}^M \cos(\sum_{i=1}^M (\alpha_i - \alpha_{i-1})) \\ l \sum_{i=1}^M \text{sen}(\sum_{i=1}^M (\alpha_i - \alpha_{i-1})) \end{bmatrix} \quad (13)$$

In which α_i , will be obtained by discrete sinusoidal generator expressed in (2). And the equation will allow obtain the position of each one of the joints for every t instant, assuming the behavior of the mechanism is an open kinematic modular chain formed by 10 D.O.F.

Nevertheless, if we want to represent this mechanism for many t instants, another problem appears: If φ sequence angle values is executed, the snake will take different positions on the plane, but without a fixed reference system respect to a x axis. In this point is necessary to feedback the model.

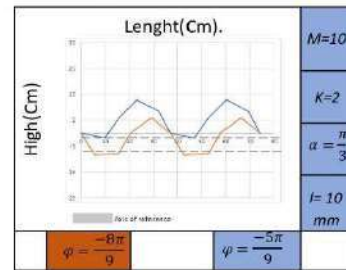


Figure 9: Same serpentine curves with different x reference axis



To solve this, a parameter is added in equation (13), with the purpose of fix the movement of the snake robot in a reference axis. Also, is desirable that this axis be $y=0$. This parameter will be:

$$\beta = \text{Min} \left\{ l \sum_{i=1}^M \sin \left(\sum_{i=1}^M (\alpha_i - \alpha_{i-1}) \right) \right\} \quad (14)$$

By subtracting the β parameter to each one of the y_i positions of i -th joints, some of these, will be zero, so, adding (14) in second row of (13), the expression obtained is:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} l \sum_{i=1}^M \cos \left(\sum_{i=1}^M (\alpha_i - \alpha_{i-1}) \right) \\ l \sum_{i=1}^M \sin \left(\sum_{i=1}^M (\alpha_i - \alpha_{i-1}) \right) - \beta \end{bmatrix} \quad (15)$$

Result is not just serpenoid curves with a same reference axis, but also a displacement in which the snake robot is touching the ground with two points of its body at least, therefore, the proposed model meets the first criterion of stability. That is showed in figure 10.

4) Mobile kinematic model

Second problem is related to the movement of the robot must generate a displacement along the horizontal axis reference. The proposal to solve this, is to add a X vector in the first row of equation (15), so that a vector is pushing the robot for each instant of time in the position, corresponding to the locomotion cycle.

Properties of this vector will be the next:

Let τ , a locomotion cycle of the snake robot.

If $t=0$, $\vec{X} = 0$

For $t = (\tau-1) + 1/2$, $\vec{X} = \frac{1}{2} \Delta_x$.

For $t=2\tau-1$, $\vec{X} = \Delta_x$.

Simultaneously, according to (4), is necessary that:

For a $\tau=0$, $\varphi = -\pi/2$

For a $\tau=1/2$, $\varphi = -\pi$

For a $\tau=1$, $\varphi = 3\pi/2$

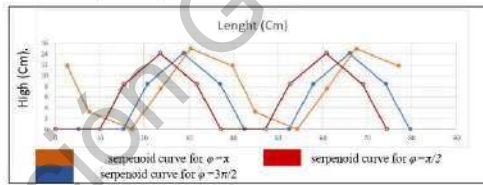


Fig. 10: Evolution of serpenoid curve with same parameters than figure 7, but applying equation (15).

Taking into account that range of φ is compound by a spectrum of π values, we will introduce (16), to represent each t instant in the locomotion cycle, in according to that range of φ .

$$t = \frac{c}{180} + (\tau - 1) \quad (16)$$

Where:

t : instant of time in locomotion cycle. ($C=0,1,2,3 \dots 180$).

c : whole number that works like a meter.

Thus, is possible to introduce (16) in definition \vec{X} , without altering its properties given the dimensionless of t , this is showed in (17).

$$\vec{X} = \Delta_x t \quad (17)$$

In the same way, c will control φ value sequence, functions that relations φ and c , are in (18).

$$\varphi(C) = \begin{cases} \frac{-\pi(C+90)}{180}, & C < 90 \\ \frac{\pi(C+90)}{180}, & C \geq 90 \end{cases} \quad (18)$$

Finally, we adjust our Δ_x , to a quotient from next considerations, doing some reasoning about snake locomotion.

If α is greater, the biological/robotic snake is going to produce a greater stride, therefore, this is proportional to displacement.

The longer the length (l) of each link, the more displacement is expected.

If number of undulations (K), is greater, the biologic/robotic snake, is going to descend more points of its body to the ground, thus, the displacement is inversely proportional.

The value obtained of for our parameters K , α and l , for this snake robot was $\Delta_x = 5.59 \text{ mm}$.

Our obtained model is expressed in (19).

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} l \sum_{i=1}^M \cos \left(\sum_{i=1}^M (\alpha_i - \alpha_{i-1}) \right) + \vec{X} \\ l \sum_{i=1}^M \sin \left(\sum_{i=1}^M (\alpha_i - \alpha_{i-1}) \right) - \beta \end{bmatrix} \quad (19)$$

Where α_i , are obtained in equation (2).

III. RESULTS

Validation of the model consists in two stages, implementation of the model in C++ code and simulation employing Excel spreadsheets and dynamic toolbars.

The other algorithm to compare this proposal, is showed in [11]. From this section onwards, reference will be made to this in the form "Model 1". The comparison consists of employing its φ piecewise function, showed in equation (3), again equation (4), and implemented in C++ Code.

To understand the steps to follow to implement the mathematical model in C++ code and simulation, a flowchart



was developed, that is showed in figure (9). This diagram is useful to only 1R snake robots fixed in any parameters.

For evaluate this code, was employed an Intel Celeron 1.60 GHZ and 4.00 GB RAM.

In table 1, parameters of snake robot for both algorithms were set. In table 2, it showed number of iterations per cycle of locomotion, and time for process one cycle, difference between two algorithms is the piecewise function in equations (3) and (4), if range of the equation is greater, the number of iterations will be.

The Algorithm implemented in C++ code consist in two "for" cycles, based in figure (9), through this, is inferred that the order of computational complexity is n^2 . From the kinematic point of view, proposal model is faster.

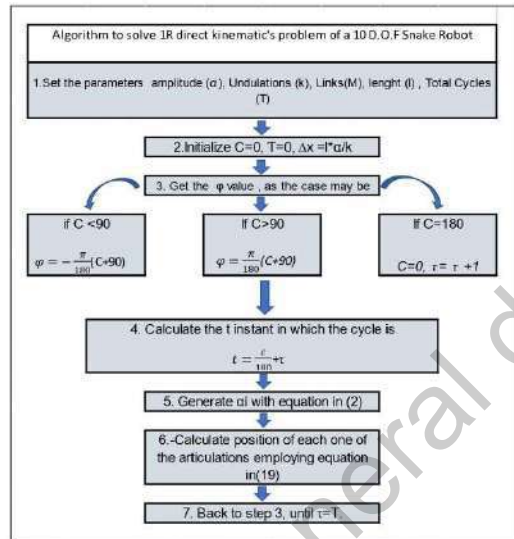


Fig 9: Necessary steps for implementation of mathematical model.

TABLE I: SETTING PARAMETERS OF SNAKE ROBOT.

K (Undulations)	α (Rad)	τ (cycle)	l (Cm)
2	$\pi/3$	1	10

TABLE II: EXECUTION OF BOTH ALGORITHMS FOR ONE CYCLE OF LOCOMOTION.

	Complex $f(n)$	Iterations per cycle of locomotion	Time (Sec.) Code	Time (Sec.) Animation	Δ_x (Cm)
Model 1	n^2	360	10,39		12.32
Proposal model	n^2	180	6,49		12.32

By the other hand, one cycle of locomotion was implemented in spreadsheet animation. Figure 10, shows evolution of both algorithms. Parameter on Δ_x , in the other model was setting in 12,32 Cm taking its stride equation.

Displacement of snake robot using both models, can be observed, its notable to advice that the number of iterations is greater than model proposal in present work. Also, it can be observed that any position in the locomotion cycle, has 2 points on the ground, and pattern of locomotion is recurrent, accomplishing stability criteria.

Table 3, resume variables and parameters employed in the develop of the algorithm.

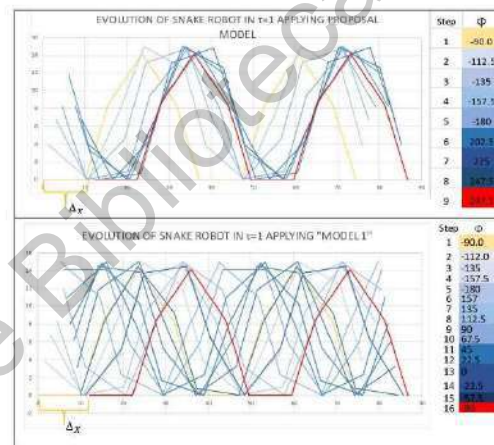


Fig. 11: Evolution of snake robot in both algorithms

TABLE III: SUMMARY OF VARIABLES AND PARAMETERS

Symbol.	Type.	Definition.	Unit.	Domain.
α_i	Variable.	Formed angle between i -link and x axis.	Rad.	$\{0; \frac{2\pi}{3}\}$
α	Parameter.	Amplitude.	Rad.	$\{0; \frac{2\pi}{3}\}$
$\Phi(c)$	Function.	Displacement for each t instant.	Rad.	$\{\frac{\pi}{2}; -\pi; \pi; \frac{3\pi}{2}\}$
K	Parameter.	Undulations.	Scalar.	$K > 2$
M	Parameter.	Number of Links.	Scalar.	$M \in \mathbb{Z}^+, M > 2$.
l	Parameter.	Length.	Cm.	$l \in \mathbb{Q}^+$
q_i	Variable.	Angle between i & $i-1$ link.	Rad.	$[-\pi; \pi]$
\vec{x}	Vector.	Displacement along x axis.	Cm.	$\{0; \frac{l\alpha}{k}\}$
Δ_x	Parameter.	Stride per cycle.	Cm.	$\frac{l\alpha}{k}$
$t(c)$	Function.	t instant in locomotion cycle.		$\{0; T\}$
C	Variable.	Meter.	Scalar.	$\{0; 180\}$
τ	Constant.	Locomotion cycle.	Scalar.	$\tau \in \mathbb{Z}^+, \tau > 1$.



T	Constant.	Total cycles of locomotion.	Scalar.	$T \in \mathbb{Z}^+, T > 1$.
-----	-----------	-----------------------------	---------	-------------------------------

IV. CONCLUSIONS

A proposal model has been developed accomplishing the stability criteria for one-dimensional snake robot locomotion.

This was possible by abstracting from the algorithm, piecewise functions than describes a basis of a vector space than generates all the positions for the 10 degrees of freedom of the snake robot in each locomotion cycle.

Even the parameter Δ_x can be set in any value, the algorithm will process the cycle of locomotion, the model is kinematics and real value will be restricted by physic limitations.

Friction allow bodies to move on a surface because is necessary for a joint to rest on the surface without sliding backwards. In this way, Δ_x is analogous to it, but no equal.

In later works, this basis could be orthogonalized, simplifying even more computational cost. The immediate benefit of this proposal, will be its implementation for a 3D Snake Robot, which compute work will be attenuated.

ACKNOWLEDGMENT

The author would like to thank the support and guidance of the Dr. José Emilio Vargas Soto, through the experimentation, analysis and redaction of results.

Present work would be impossible without support and financing from CONACYT, to the Master in science (Mathematical Engineering). Of which the author belongs.

The present work was carried out within the research and postgraduate division of the engineering faculty at the Universidad Autónoma de Querétaro, for facilitate its facilities and services to develop this research.

To professors Roberto Augusto Gómez Loenzo and Luisa Ramírez Granados by their suggestions in program and math involved in the model.

REFERENCES

- [1] D. Axehill, G. Hendeby, J. Sjöberg and D. Tömqvist. The history of Industrial Robot. (Publication: LiTH-isy-R-2853). Department of Electrical Engineering Linköping University. 2008.
- [2] An automatic Block -Setting Crane: Meccano Model by a Robot Unit. The Meccano magazine press. March 1937.
- [3] International Organization for Standardization. Robots and robotic devices vocabulary (ISO/DIS Standard No. 8373:2012). 2012.
- [4] International Federation of Robotics. Service Robotics, 2016.
- [5] S. Nagar (2014). "A caterpillar tracks based locomotive and obstacle avoiding robot" [Digests Conference: Recent Development in Science Engineering and Technology, At G D Goenka University, 2014]
- [6] Y. Perry, "AntBot: El robot araña que imita a las hormigas sin necesidad de un GPS. 2019.

- [7] "Esta robo-serpiente buscó sobrevivientes tras el sismo de la CDMX". El Financiero.p.1. October 2017.
- [8] N. Ahix, "L'équipe de corée en force au salon innorobo". Retrieved from robots et compagnie press. 2017.
- [9] S. Hirose, and H. Yamada. "Snake-like robots: Machine design of biologically inspired robots". IEEE Robotics and Automation Magazine, 16(1), pp.88-98.2019.
- [10] J. Yang, E. Peña-Pitarch, S. Beck and K. Abdel-Malek, "Synthesis and analysis of a flexible elephant trunk robot". Advanced Robotics. p132. 2006.
- [11] J. González Gómez. "Robótica Modular Y Locomoción: Aplicación A Robots Apodos". Universidad Autónoma de Madrid, Madrid, España. 2008.
- [12] J. E Vargas Soto, J.C Pedraza-Ortega, J.M. Ramos- Arreguin, and L. López-Vallejo. "Desarrollo de proyectos mecatrónicos", Asociación Mexicana de Mecatrónica, Querétaro, Mexico. 2012.



Sabemos que la imagen en 3 dimensiones del robot, existirá en el espacio \mathbb{R}^3 , y al ser una estructura rígida, debe conservar en sus articulaciones una distancia l , por tanto:

$$gen\{P\} = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in R^3; \|p_i p_{i+1}\| = l \right\} \quad (58)$$

Nota: esto quedó demostrado en el anexo II.

Mientras que la imagen en 3 dimensiones del robot, que cumpla con la restricción de estabilidad cinemática para robots ápodos puede representarse como:

$$gen\{W\} = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in R^3; \|p_i p_{i+1}\| = l, n(z_i = 0) \geq 2 \forall t \right\} \quad (59)$$

sean los vectores $(x_i, y_i, z_i), (x_{i+1}, y_{i+1}, z_{i+1}), (x_{i+2}, y_{i+2}, z_{i+2})$ generados por P.

$$(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 = l^2$$

$$(x_{i+1} - x_{i+2})^2 + (y_{i+1} - y_{i+2})^2 + (z_{i+1} - z_{i+2})^2 = l^2$$

Casos:

Si solo $z_i = 0$, no cumple con W

Si $z_i = 0$ & $z_{i+1} = 0$, cumple con las dos restricciones de W y es generado por P,

Si $z_i = 0$ & $z_{i+2} = 0$, cumple con W y es generado por P

Si $z_i = 0, z_{i+1} = 0, z_{i+2} = 0$, cumple con W y es generado por P

Esto último puede cumplirse para n vectores generados. por tanto, Concluimos que:

1: No toda colección generada por P, cumple con W.

2: P puede generar a W, por tanto, W es subconjunto de P