



# UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA

CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO

POSGRADO EN INGENIERÍA MECÁNICA

Tesis

## MODELADO Y GENERACIÓN DE TRAYECTORIAS PARA UN ROBOT CAMINANTE DE 2 GRADOS DE LIBERTAD

Para obtener el grado de

**Maestro en Ingeniería Mecánica**

Presentada por

**Ing. Jesús Raúl Alonso Armadillo**

Asesor

**Dr. Mauro Eduardo Maya Méndez**

Co-Asesor

**Dr. Eduardo Morales Sánchez**

San Luis Potosí, S.L.P. Febrero de 2012





# Agradecimientos

A Dios por darme la vida y rodearme de grandes personas.

Gracias papá y mamá por apoyarme a lo largo de esta etapa que hoy termina, y que es un logro más de los tres.

A mis asesores de tesis, Dr. Mauro y Dr. Morales, por el tiempo y esfuerzo dedicados junto conmigo a este trabajo.

También debo agradecer a las latosas de mis hermanas (sí! leyeron bien, latosas!) que siempre han estado al pendiente de mis logros y fracasos. A mi hermano y sobrinitos que, con cada sonrisa me dan un motivo más para yo sonreírle a la vida.

A mi pekeña... TE AMO PRECIOSA!!... Gracias por brindarme tu apoyo incondicional!

Gracias a todos mis maestros y compañeros de la Universidad que en algún momento dedicaron parte de su tiempo en ayudarme. En particular, gracias a mis sinodales, Dres. Emilio, Gilberto y Hugo, por las sugerencias realizadas a lo largo de este trabajo.

Al personal administrativo del posgrado. Gracias Zory y Sra. Chela!

Finalmente, gracias al programa de becas de CONACYT y al Fondo de Apoyo a la Investigación de la UASLP.



# Resumen

En este trabajo se presenta el análisis y modelado cinemático de un robot no holónimo de 2 grados de libertad con patas. En una primera instancia se obtiene un modelo cinemático analizando los mecanismos de un robot comercial. Sin embargo estas ecuaciones resultan en un modelo complicado y poco manejable. Posteriormente se obtiene un modelo simplificado haciendo una aproximación de las ecuaciones encontradas en el punto anterior. El modelo simplificado es validado en simulación comparándolo contra las ecuaciones obtenidas analíticamente.

En base al modelo simplificado, se obtienen las trayectorias primitivas del robot. A partir de estas se desarrolla una propuesta de planeación de trayectorias considerando un espacio libre de obstáculos. Se consideró el criterio de mínima distancia como criterio de optimización para la planeación de trayectorias. La minimización de distancia fue obtenida usando métodos de programación entera, esto debido al tipo de desplazamiento que presenta el robot.

Finalmente se desarrolla una plataforma de simulación para validar la planeación de trayectorias. La plataforma consta de dos partes: la primera parte contiene un algoritmo que calcula la trayectoria óptima en base a las ecuaciones obtenidas de la planeación de trayectorias; la segunda parte se encarga de animar la trayectoria solicitada por el algoritmo.

Palabras clave: **Robots móviles, robots con patas, modelado cinemático, generación de trayectorias, trayectorias discretas, criterios de optimización.**



# Tabla de Contenido

Agradecimientos.....	i
Resumen .....	iii
Tabla de Contenido .....	v
Introducción .....	1
Capítulo 1. Estado del Arte de Robots con Patas .....	5
1.1 Clasificación de los Robots .....	5
1.2 Robots Móviles .....	6
1.2.1 Robots con Ruedas .....	6
1.2.2 Robots Articulados .....	8
1.2.3 Robots Submarinos o Aéreos .....	8
1.2.4 Robots con Patas .....	9
1.3 Modelado de Robots con Patas .....	14
1.4 Trayectorias en Robots con Patas .....	17
Capítulo 2. Modelado Cinemático.....	21
2.1 Descripción del Robot .....	21
2.1.1 Arquitectura.....	21
2.1.2 Movimiento .....	25
2.2 Modelado Cinemático .....	26
2.2.1 Análisis de la Pata 1 .....	26
2.2.2 Análisis de la Pata 2 .....	32
2.2.3 Ecuaciones del Robot .....	34
Capítulo 3. Generación de Trayectorias .....	43

3.1 Planeación de Trayectorias .....	45
3.1.1 Minimización de Energía.....	47
3.1.2 Minimización de Tiempo.....	50
3.1.3 Minimización de Distancia.....	52
3.1.4 Propuestas de Trayectorias .....	56
3.2 Algoritmo de Seguimiento de Trayectorias .....	73
3.2.1 Simulador .....	73
Capítulo 4. Resultados.....	89
4.1 Resultados en Caso de Distancia Entera.....	90
4.2 Resultados en el Caso de Corta Distancia .....	92
4.3 Resultados en el Caso de Larga Distancia.....	94
Conclusiones.....	101
Anexo .....	105
Anexo A .....	105
Anexo B.....	109
Anexo C.....	111
Bibliografía.....	131



# Introducción

Los robots se han introducido poco a poco en la vida cotidiana de distintas formas, desde aquellos que son desarrollados para el entretenimiento como los juguetes hasta brazos robóticos automatizados encargados del ensamble de automóviles. La gama en la que se presentan los robots es muy extensa y para facilitar su estudio se suelen clasificar según diferentes criterios. Una de las clasificaciones más comunes es la realizada por T.M. Knasel (Ollero, 2001); en donde los robots son clasificados dependiendo del año donde comenzaron a ser desarrollados. A esta clasificación también se le conoce como *clasificación por generación*.

Una de las clases de robots que más han despertado el interés en los últimos años es la de los llamados robots móviles (terrestres, aéreos, submarinos, etc.). Una de las principales razones para ello es que, al poder desplazarse en su entorno, el espacio de trabajo de los robots móviles se extiende casi sin límite. Esto los habilita para la realización de diversas tareas que requieren grandes desplazamientos comparados con el tamaño del robot.

Dentro de los robots móviles terrestres, los robots con patas parecen destacar por su capacidad de moverse en terrenos irregulares debido a su movilidad. Sin embargo, el diseño y el control de este tipo de robots son más complejos que por ejemplo los de los robots con ruedas. En efecto, las patas son mecanismos más complejos que las ruedas y para lograr la estabilidad de los robots con patas se requiere una gran coordinación en el movimiento de sus extremidades. En ambos casos los robots están sujetos a restricciones no holónomas; es decir, el sistema no puede moverse en ciertas direcciones de manera instantánea.

El diseño de robots con extremidades está inspirado en los seres vivos. El hecho de que un robot tome la forma de un ser vivo le permite un desempeño adecuado en diversos ambientes. La naturaleza misma proporciona una extensa variedad de modelos para imitar; insectos, mamíferos e incluso al mismo ser humano.

Algunas de las ventajas que presentan los robots con patas según (Díaz & García, 2009) son: movilidad, superación de obstáculos, suspensión activa, desplazamiento en diferentes terrenos, etc. Por otro lado, la estructura mecánica de los robots con patas es relativamente complicada, así como la electrónica necesaria, los algoritmos de control asociados, el costo, etc.

En (Hernández & Díaz) se presenta el desarrollo del robot ANGELO. Este es un robot que consta de cuatro extremidades y en cada una de ellas tiene tres articulaciones, o sea tres grados de libertad por pata, asemejándolo así a un cuadrúpedo viviente. Esto incrementa la dificultad en la coordinación de los movimientos para lograr la estabilidad del robot, dado que cuenta con un número relativamente pequeño de puntos de apoyo. El control de este robot está hecho mediante un micro controlador dsPIC30F4013 que controla a los servomotores en cada extremidad.

En (Ávila & Zambrano, 2007) se presenta el desarrollo de REMLED, un robot con patas con tres articulaciones en cada extremidad. Para la selección de los actuadores de REMLED fueron necesarios los resultados obtenidos del análisis dinámico del robot, debido a los altos torques que son generados en cada articulación. También se buscó que éstos tuvieran un sistema de autobloqueo para mantener la estabilidad estática sobre el sistema mecánico y no sobre el actuador, y que contara de igual manera con un sistema de reducción para aumentar el par de salida y disminuir la velocidad.

A pesar de las ventajas de estos sistemas, en los desarrollos (Hernández & Díaz) y (Ávila & Zambrano, 2007) queda de manifiesto la complejidad del diseño y la dificultad para controlar los robots con patas. Mientras que con ANGELO fueron necesarios numerosos intentos para lograr la estabilidad del robot, con REMLED fueron necesarios los resultados del análisis cinemático y dinámico para el mismo propósito. En general, para aspirar a tener una garantía de la estabilidad del sistema controlado, es necesario contar con el modelo del sistema y analizar sus propiedades.

## ***Justificación***

Tratando de conservar las ventajas y paliar las dificultades de estos sistemas, una alternativa que merece ser estudiada es la propuesta de robots con patas con arquitectura simple. Este trabajo va en esa dirección. Aquí se realiza el estudio de los movimientos que un robot con arquitectura simple es capaz de realizar. Teniendo como resultado, que a pesar de su arquitectura es capaz de llegar de un punto a otro cualquiera sobre la superficie mediante una serie de maniobras. De esta manera se tiene que no necesariamente debe disponerse de un complejo diseño, en un robot con patas, para lograr la principal tarea de los robots móviles que es poder alcanzar cualquier punto en su espacio de trabajo

En este trabajo se estudia un robot no holónimo con patas de arquitectura simple. La no holonomía de este robot se debe a las restricciones que le impiden desplazarse instantáneamente en ciertas direcciones. Esto resulta en dificultades para controlar este tipo de sistemas. Un ejemplo de este tipo de restricciones, es la restricción de desplazamiento lateral que impide estacionar un vehículo sin

necesidad de recurrir a una serie de maniobras. El robot en este trabajo cuenta con un mayor número de restricciones que las presentes en un vehículo sobre ruedas. Por lo anterior es necesario un estudio detallado del modelado del sistema y la valoración de sus características.

El robot que fue utilizado como referencia para la obtención del modelo matemático y la generación de trayectorias, fue el micro robot “Inchworm” de la marca Hexbug®. Este micro robot cuenta únicamente con dos actuadores para realizar la tarea de caminado; lo cual lo hace un robot relativamente sencillo de controlar en comparación de otros robots con patas con un mayor número de actuadores.

### ***Motivación***

La motivación del trabajo es múltiple: en primer lugar se analiza una clase de robot no holónimo con patas que tiene una arquitectura simple pero resistente. Estas características permiten prever ciertas ventajas de estos robots como el bajo costo y la alta capacidad de carga para una variedad de aplicaciones. Por otro lado, este análisis proporciona información esencial para, a futuro, construir prototipos de este tipo de robots y sintetizar algoritmos de navegación y algoritmos de control.

### ***Objetivo general***

El objetivo general de este trabajo consiste en desarrollar un generador de trayectorias para un robot móvil con patas. Debido a las restricciones que presenta el robot en su desplazamiento, es necesario obtener un modelo matemático que determine el tipo de trayectorias que el robot es capaz de realizar.

### ***Objetivos particulares***

Como objetivos particulares en esta tesis se tienen los siguientes:

- Obtener las relaciones de posición y velocidad presentes en los mecanismos del robot.
- Obtener el modelo cinemático considerando un marco de referencia inercial sobre la superficie de desplazamiento y un marco de referencia móvil en el cuerpo del robot.
- Obtener las trayectorias admisibles para el robot.
- Realizar una planeación de trayectorias para el robot bajo el criterio de mínima distancia y considerando un espacio libre de obstáculos.
- Desarrollar una plataforma de simulación para validar la planeación de trayectorias.
- Reportar los resultados obtenidos del simulador.

### ***Organización del trabajo***

El presente documento está organizado de la siguiente manera: en el capítulo 1 se realiza la revisión de las aportaciones más significativas sobre robots móviles que guardan relación con el presente

proyecto. En dichas aportaciones se aborda la problemática de modelado y la obtención de las trayectorias en diferentes tipos de robots móviles. Posteriormente se presenta el capítulo de *Modelado Cinemático*, donde se obtienen las ecuaciones que describen matemáticamente los movimientos del robot. Con estas ecuaciones también es posible describir las limitaciones de desplazamiento que posee el robot. Después se encuentra el capítulo de *Generación de Trayectorias*, en el que se obtienen las trayectorias primitivas del robot con base en su modelo cinemático. Dentro del mismo capítulo, se desarrolla un generador de trayectorias que consiste en un algoritmo cuya finalidad es llevar el robot de una posición inicial a otra final, respetando las restricciones de movimiento propias del robot. Además, las trayectorias generadas por el algoritmo deben cumplir con el criterio de *mínima distancia*, el cual se refiere a que el robot debe ser capaz de alcanzar cierta posición recorriendo la menor distancia posible. También se busca un bajo costo computacional del algoritmo, con la idea de implementar a futuro el generador de trayectorias en un prototipo. Finalmente se presenta el capítulo de *Resultados* donde, de manera gráfica, se muestran diferentes trayectorias que el robot opta por seguir en base al algoritmo de seguimiento de trayectorias. El aporte de esta tesis se ve reflejado en el apartado de *Conclusiones*. En dicho apartado se sugieren algunas aplicaciones que pueden recurrir a alguno de los métodos utilizados para el modelado y generación de trayectorias en esta tesis. De igual manera se presenta el trabajo a futuro para desarrollar el control en lazo cerrado del robot.

# Capítulo 1. Estado del Arte de Robots con Patas

Lo que hoy día se conoce como robótica, está compuesta por diversas ramas como la electrónica y la mecánica, el control y la computación por mencionar algunas. A lo largo de los últimos años se ha presentado un significativo avance en cada una de las ramas que la componen y por tanto en la robótica como tal. De cierta manera, lo que ha propiciado este avance es el darse cuenta de lo útil que resultan los robots en el desempeño múltiples tareas. De aquí se tiene que, debido a la diversidad de aplicaciones para los robots, existen distintos tipos de robots con diferentes características.

## 1.1 Clasificación de los Robots

La gama en la que se presentan los robots es muy extensa y para facilitar su estudio es conveniente realizar una clasificación de los mismos. Sin embargo, para la gran diversidad de robots no existe aún alguna clasificación que se utilice de manera general. Existen pues distintas clasificaciones basadas en diferentes criterios; nivel de control, nivel de inteligencia, nivel de lenguaje de programación y generación (Intriago, 2009), entre otros. En este trabajo se toma como referencia la llamada clasificación por generación de T. M. Knasel (Ollero, 2001), tabla 1.1. En esta clasificación, los diferentes tipos de robots son organizados dependiendo del año en que comenzaron a ser desarrollados.

En la tabla 1.1 se presentan diferentes tipos de robots además de los móviles. Es posible realizar una descripción más detallada de cada uno de los diferentes tipos de robots. Sin embargo, solo se realizará la descripción de los robots móviles debido a que esta tesis trata de un subtipo de estos robots.

Generación	Nombre	Tipo de Control	Grado de movilidad	Usos más frecuentes
1ª (1982)	Pick & Place	Fines de carrera, aprendizaje	Ninguno	Manipulación. Servicio de máquinas.
2ª (1984)	Servo	Servo control, trayectoria continua, programación condicional.	Desplazamiento por vía	Soldadura Pintura.
3ª (1989)	Ensamblado	Servos de precisión, visión, tacto, programación offline	Guiado por vía	Ensamblado Desbardado
4ª (2000)	Móvil	Sensores inteligentes	Patas, ruedas.	Construcción Mantenimiento
5ª (2010)	Especiales	Controlados con técnicas de IA	Andante, saltarín	Uso militar Uso especial

Tabla 1.1. Clasificación de los robots según T.M. Knasel

## 1.2 Robots Móviles

Los robots móviles son aquellos que brindan la posibilidad de desplazarse a través de distintos terrenos. Su principal característica es la de poseer un espacio de trabajo relativamente grande en comparación al tamaño del robot.

La variedad de robots móviles que existe merece el realizar una diferenciación entre los mismos. En (Ollero, 2001) se presenta una clasificación para robots móviles, la cual se basa en el tipo de locomoción que éstos poseen. Dicha clasificación está dividida de la siguiente manera:

- 1) Robots con ruedas
- 2) Robots articulados
- 3) Robots submarinos y aéreos
- 4) Robots con patas

A continuación se presenta una breve descripción de cada uno de los tipos de robots móviles involucrados en esta clasificación. Cabe mencionar que las características que se mencionan a continuación se basan en modelos de robots comunes, excluyendo aquellos con características particulares.

### 1.2.1 Robots con Ruedas

Este tipo de robots presentan ser una solución más simple (en comparación a los demás sistemas de locomoción) para conseguir la movilidad en terrenos rígidos y libres de obstáculos, permitiendo conseguir velocidades relativamente altas. Un ejemplo de robot móvil con ruedas se ilustra en la figura 1.1.



Figura 1.1. Ejemplo de robot móvil con ruedas

Una característica digna de mención en los robots con ruedas, son las restricciones no holónomas que la mayoría de estos sistemas posee. Estas restricciones impiden el movimiento de los mismos en ciertas direcciones de manera instantánea. Esto ocasiona que el robot tenga trayectorias realizables (las que puede ejecutar) y trayectorias no realizables (que no puede seguir debido a sus restricciones). Desde el punto de vista control, estas restricciones presentan retos importantes para controlar estos sistemas. Una de las tareas más complicadas de realizar es la estabilización de un punto fijo (en posición y orientación), que de manera práctica puede asociarse a una tarea de estacionado. Estas temáticas han sido abordadas durante la última década por muchos investigadores reconocidos a nivel mundial.

Como limitación más significativa para este tipo de vehículos cabe mencionar el deslizamiento en la impulsión (derrape). Dependiendo de las características del terreno pueden presentarse deslizamientos y/o vibraciones. De esta manera es posible deducir que la locomoción sobre ruedas es poco eficiente en terrenos blandos o accidentados.

Es verdad que mediante un estudio pertinente, es posible guiar a un robot con ruedas a través de un camino libre de colisiones, sin embargo, es posible que en dicho camino existan ciertos obstáculos que deben ser superados. En tal caso, los robots con ruedas parecen no ser la mejor opción, esto debido a que la relación entre el tamaño del obstáculo y el radio de las llantas debe ser muy pequeña.

Cabe mencionar que las limitaciones (así como las restricciones de movimiento) en este tipo de robots se refieren a un tipo de robots convencional, es decir, aquellos que no cuentan con adaptaciones especiales como suspensión o ruedas omnidireccionales.

Los robots con ruedas emplean a su vez, diferentes tipos de locomoción mediante ruedas que les confieren características y propiedades diferentes respecto a dimensiones, carga útil, eficiencia energética y maniobrabilidad.

## 1.2.2 Robots Articulados

Este tipo de robots poseen similitud con animales rastreros, por ejemplo los gusanos y serpientes; se destacan por su capacidad de adaptación al terreno y son apropiados para caminos estrechos.

Dependiendo de las adaptaciones de las que se puede proveer a este tipo de robots, el fenómeno de derrape se presenta en menor o mayor medida.

El diseño mecánico de los robots articulados es en buena medida complicado, ya que generalmente están compuestos por varios mecanismos encadenados entre sí.

El control en estos robots suele ser complejo. Esto debido a que poseen un gran número de actuadores. Existen casos donde no sólo están compuestos por articulaciones, sino por otro tipo de mecanismos como ruedas y/o patas (Ollero, 2001). En la figura 1.2 se muestra un ejemplo de robot articulado.



Figura 1.2. Ejemplo de robot articulado

Actualmente el estudio de robots articulados se encuentra en reciente desarrollo, a diferencia de los robots con ruedas de los cuales es posible encontrar una gran cantidad de trabajos dedicados a los mismos.

## 1.2.3 Robots Submarinos o Aéreos

Los robots aéreos y submarinos son construidos principalmente con fines de exploración. Con este tipo de robots se busca obtener información de lugares que, para el hombre, son inaccesibles o peligrosos; por ejemplo las profundidades inexploradas del mar o atmosferas de planetas en el espacio exterior.

Comúnmente estos robots son el resultado de la evolución de vehículos completamente tele operados o pilotados por el hombre; como aviones y submarinos. En la figura 1.3 se observa el



robot submarino Odyssey IV desarrollado en el Instituto Tecnológico de Massachusetts (Morales, 2008).

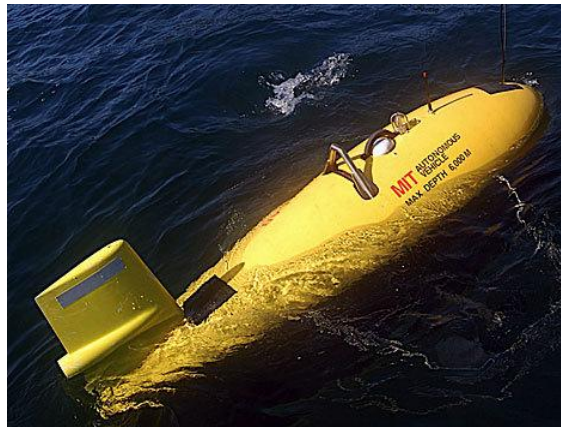


Figura 1.3. Robot submarino Odissey IV

La arquitectura en este tipo de robots es muy variable y no es posible hacer una generalización de sencillez o complejidad en el diseño de los mismos. Como ejemplo de lo anterior, se tiene que un robot con arquitectura similar a la de un submarino presenta un diseño relativamente sencillo en comparación a un robot aéreo que imite el vuelo de cualquier ave. De aquí se tiene que el control para este tipo de robots puede ser desde muy sencillo hasta muy complejo.

## 1.2.4 Robots con Patas

En terrenos suaves o accidentados, los robots con patas son la mejor opción según diferentes autores ((Ollero, 2001), (Liu, Tan & Zhao, 2007), (Díaz & García)). Lo anterior se debe a diversas características que posee este tipo de robots. En la figura 1.4 se muestra un ejemplo de robot con patas.

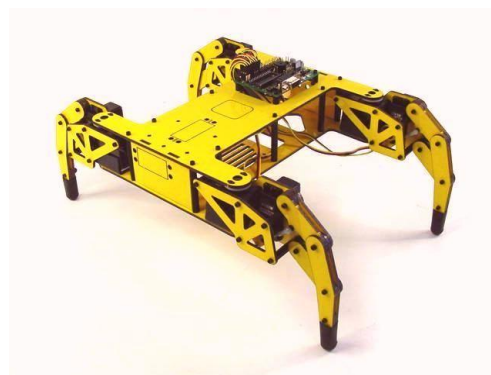


Figura 1.4. Ejemplo de robot móvil con patas

Las características que se mencionan a continuación, se describen con la finalidad de comparar las diferentes propiedades que poseen los robots con patas y los robots con ruedas. Esta comparación directa se debe a que ambos tipos de robots se desplazan sobre superficies sólidas; ya sea accidentadas o completamente lisas.

- **Buena movilidad:** Los robots con patas presentan una mayor movilidad que los que los que utilizan ruedas u orugas. Lo anterior debido a las diferentes configuraciones que pueden tomar sus articulaciones sin tener que modificar los puntos de apoyo con la superficie. Con esto el robot podrá realizar movimientos como: inclinación; el cuerpo del robot no está obligado a mantenerse paralelo a la superficie. Rotación; el robot es capaz de girar su cuerpo sin modificar los puntos de apoyo (el máximo de giro depende de las propiedades de las articulaciones de cada modelo de robot). Posicionamiento; el cuerpo es capaz de desplazar su centro de gravedad sin necesidad de modificar los puntos de contacto con la superficie (ésta propiedad también está limitada por las propiedades que poseen las articulaciones de cada modelo). En la figura 1.5 se ilustran los movimientos mencionados,

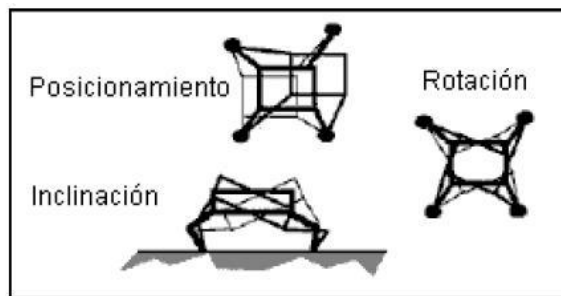


Figura 1.5. Movilidad

- **Buena capacidad de superar obstáculos:** Un robot con patas puede superar obstáculos que sean relativamente altos en comparación a su altura, a diferencia de los robots con ruedas los cuales no superan obstáculos de un tamaño mayor al radio de sus llantas (haciendo referencia a un robot con ruedas convencional). Para ilustrar este concepto está la figura 1.6.
- **Buena suspensión:** Intrínsecamente un robot con patas posee “suspensión por adaptación”, la cual es posible debido a la capacidad que tiene de variar la altura de su cuerpo mediante la posición de sus patas. De esta manera su movimiento puede ser más suave que un robot con ruedas. La figura 1.7 muestra explícitamente esta descripción.

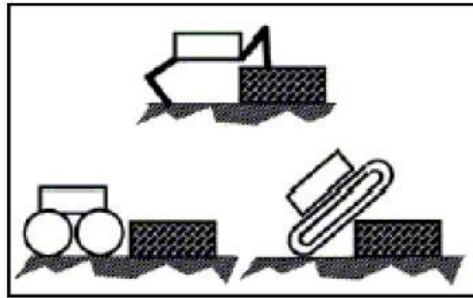


Figura 1.6. Superación de obstáculos

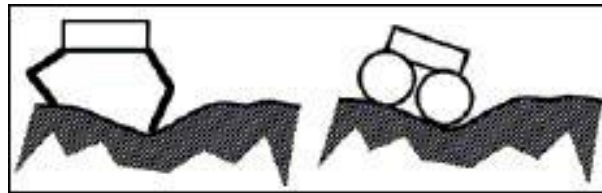


Figura 1.7. Suspensión activa

- **No precisa de terrenos continuos:** Los robots con patas no necesitan precisamente superficies continuas para poder desplazarse. A diferencia de los robots convencionales con ruedas para los cuales alguna discontinuidad en el camino podría representar un gran problema. Para una mejor comprensión de lo anterior observar la figura 1.8.



Figura 1.8. Terreno discontinuo

- **Buen agarre con la superficie:** Las patas de un robot transfieren el peso del mismo de forma directa sobre el suelo y eso hace menor la posibilidad de que resbale. En cambio una rueda puede deslizarse sobre una superficie dependiendo de su adherencia a la misma, como se observa en la figura 1.9.

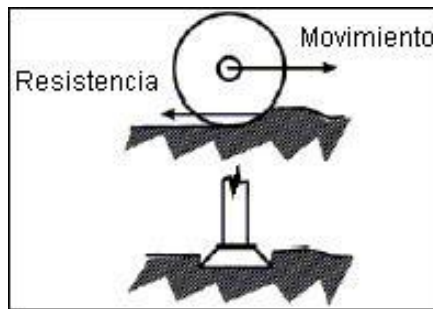


Figura 1.9. Deslizamiento

- **Consideración del medio ambiente:** Los robots con patas tienen contacto con la superficie de la Tierra de manera puntual. Caso contrario de los que poseen ruedas, los cuales tienen un contacto continuo y pueden ocasionar erosión al derrapar dejando desprotegida la superficie.
- **Capacidad de mantener velocidad media constante:** Un robot con patas puede superar pequeños obstáculos sin tener que modificar su velocidad, figura 1.10.

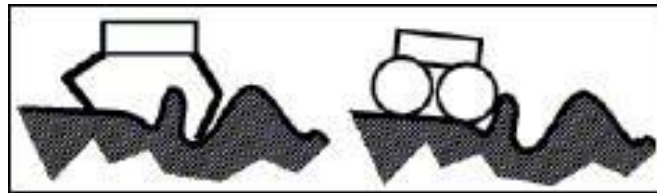


Figura 1.10. Velocidad media constante

Desde luego, los robots con patas también presentan una serie de problemas que requieren de atención. De igual forma que se enunciaron las ventajas en el uso de este tipo de robot, también se hará mención de las desventajas que éstos presentan.

- **Diseño mecánico complejo:** Una pata consiste de varias articulaciones y enlaces, a diferencia de la rueda es un mecanismo extremadamente sencillo.
- **Electrónica compleja:** La cantidad de articulaciones en un robot con patas es claramente mayor que en un robot con ruedas. Por lo tanto se necesita un considerable número de actuadores para poder tener control en los movimientos de un robot con patas.
- **Algoritmos de control:** Un robot caminante debe coordinar la posición de todas sus articulaciones para realizar cualquier movimiento. Es por esto que el algoritmo de control en los robots con patas se vuelve más complejo que en los robots con ruedas.

- **Velocidad máxima baja:** El tipo de locomoción que poseen los robots con patas, impide que estos mecanismos alcancen grandes velocidades en comparación de los robots con ruedas. Esto se debe principalmente a que a grandes velocidades, la estabilidad de un robot con patas peligra, a diferencia de un robot con ruedas que posee una arquitectura muy estable.
- **Costo de producción:** El costo total de cualquier sistema, por lo regular, es proporcional a su complejidad. Por tanto un mecanismo robótico de locomoción por patas será de un costo mayor que uno con ruedas. Dependiendo del número de patas con las que cuenta un robot, puede haber un mayor o menor grado de complejidad para encontrar la solución a los problemas mencionados ya anteriormente.

Como se puede apreciar en las desventajas mencionadas anteriormente, el principal problema de los robots con patas es su complejidad en general. Sin embargo, existen mecanismos con patas que son relativamente sencillos, los cuales presentan algunas de las características mencionadas anteriormente. El estudio de robots simples resulta importante debido a los altos costos involucrados en el diseño y control de los robots con patas. Un ejemplo de robots simples es el robot que sirve como modelo en este trabajo de tesis, el cual es un robot que posee únicamente dos actuadores que proveen de movimiento a las patas del mismo.

A lo largo de este trabajo se obtienen las propiedades y restricciones presentes en el caminado del robot modelo. Para esto se requiere la obtención de un modelo matemático que dicte la forma de interacción entre el robot y la superficie de desplazamiento del mismo. También, como parte de este trabajo, se tiene la generación de trayectorias para el mismo robot. Para la generación de trayectorias inicialmente se obtienen las trayectorias *admisibles* del robot; estas trayectorias están definidas por el modelo matemático antes descrito, en otras palabras, la arquitectura del robot es la que dicta las trayectorias admisibles del mismo. Con mira a implementar un algoritmo generador de trayectorias en un prototipo real, se desarrollará una planificación de trayectorias basada en el criterio de distancia mínima, la cual se valida mediante un simulador (que también fue desarrollado en esta tesis).

En los siguientes dos apartados se citan diferentes trabajos con robots móviles. Una parte de estos trabajos se refiere a la obtención del modelo cinemático para algunos robots con patas. Otra parte hace referencia a la generación de trayectorias de algunos robots de este mismo tipo y de algunos robots con ruedas.

## 1.3 Modelado de Robots con Patas

Un modelo cinemático describe la relación entre el movimiento de las uniones y el movimiento de los cuerpos rígidos que componen al robot (Roque & Aguilera, 2010). Esta relación es la velocidad de cada uno de los elementos del robot, ya sean velocidades angulares o lineales.

En este apartado se manejan el término de *cadena cinemática*, el cual se define a continuación para un mejor entendimiento del lector.

**Cadena cinemática** (figura 1.11). Conjunto o subconjunto de miembros de un mecanismo enlazados entre sí. Por ejemplo, la cadena de transmisión de un vehículo, el mecanismo pistón-biela-manivela, etc. Los miembros de una cadena cinemática se denominan *eslabones*.

- *Cadena cerrada o anillo*. Cadena cinemática tal que cada uno de sus miembros está enlazado nada más con dos miembros de la misma cadena.
- *Cadena abierta*. Cadena cinemática que no tiene ningún anillo (Cardona & Clos, 2001).

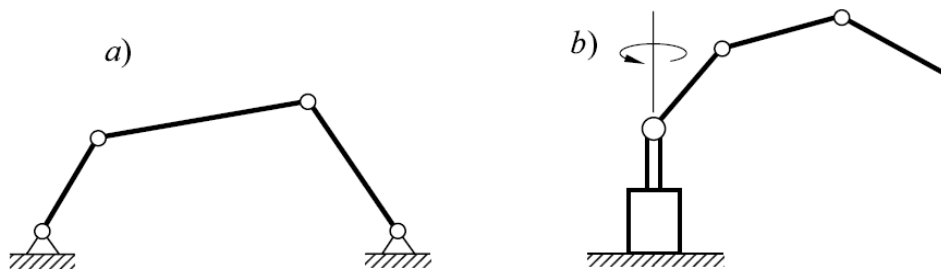


Figura 1.11. Cadena cinemática a) cerrada y b) abierta

La forma de obtener un modelo cinemático para cada robot, varía dependiendo de la arquitectura del mismo (entendiendo por “arquitectura” a la forma particular de cada robot, medidas y ensambles). Sin embargo, existen similitudes en la manera de obtener tal modelo para cada tipo de robots.

Para este proyecto se revisaron diferentes trabajos sobre el modelado de robots con patas, de los cuales se presenta un breve resumen a continuación.

### ***Robot ROSEP***

El modelo cinemático del ROSEP (Robot de Seis Patas) se basa en las ecuaciones que determinan la posición y la orientación del cuerpo, tomando en cuenta la posición final que adoptan las patas sobre un terreno irregular, así como la inherente redundancia del complejo sistema mecánico articulado formado por los dieciocho grados de libertad (3 grados por pata) que permiten dirigir la orientación del robot (Vargas, Gorrostieta & Pérez, 2002).

El tipo de mecanismo que presenta en cada una de las patas de ROSEP es una cadena cinemática abierta. Aquí se traza un vector posición desde el origen del sistema de referencia (posicionado en el centro de gravedad del robot) hasta el extremo de cada pata, figura 1.12.

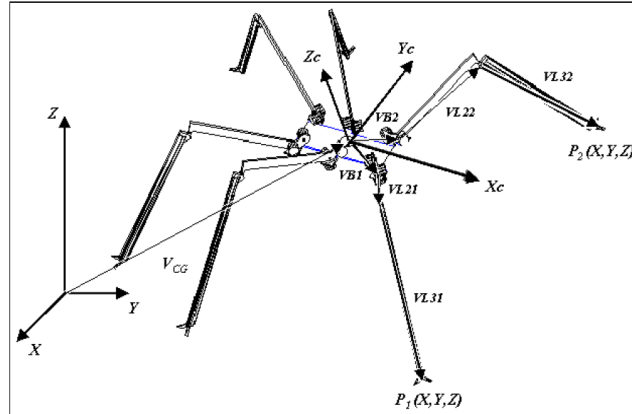


Figura 1.12. Relaciones vectoriales de ROSEP

Con las relaciones vectoriales en la figura 1.12 se desea encontrar el comportamiento que posee el extremo de cada pata respecto al sistema de referencia inercial  $(X, Y, Z)$ , para ello se traza un vector  $\vec{V}_{Pi}$  que va desde el origen del sistema de referencia inercial hasta el extremo de la  $i$ -ésima pata. Para determinar el vector  $\vec{V}_{Pi}$  de la  $i$ -ésima pata, se traza un vector  $\vec{V}_{CG}$  desde el origen del sistema de referencia inercial hasta el punto donde se encuentra el centro de gravedad del robot, de allí se traza otro vector  $\vec{V}_{Bi}$  hacia la primera articulación de la  $i$ -ésima pata, posteriormente otro vector  $\vec{V}_{L2i}$  que vaya desde la primera hasta la segunda articulación, y finalmente un vector  $\vec{V}_{L3i}$  que apunte desde la segunda articulación hasta el término de la cadena cinemática abierta. De este modo se tiene que  $\vec{V}_{Pi} = \vec{V}_{CG} + \vec{V}_{Bi} + \vec{V}_{L2i} + \vec{V}_{L3i}$ , con lo cual es posible encontrar la posición del extremo de cada pata respecto al sistema de referencia inercial.

### **Robot REMLED**

Otro trabajo donde se presenta el modelado cinemático de un robot con patas, es el realizado con REMLED (Robot Escalador para el Mantenimiento y Limpieza de Edificios), figura 1.13. La estructura básica de REMLED está constituida por el cuerpo y cuatro patas. Cada una de estas patas está conformada por cuatro eslabones unidos a través de articulaciones. Tres de estos eslabones están unidos mediante articulaciones actuadas, esto quiere decir que cada pata posee tres grados de libertad rotacionales gracias al movimiento del eje de un motor. El último eslabón está unido por medio de una articulación universal no actuada, la cual proporciona dos grados de libertad más extremo final de la pata.

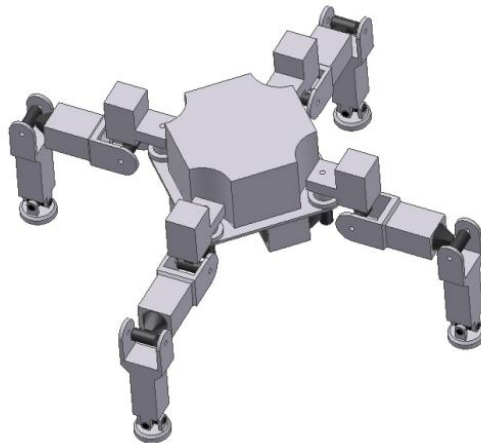


Figura 1.13. Vista general del diseño de REMLED

En la figura 1.14 se muestra el esquema de la cinemática de una pata del robot. Donde

- $l_1, l_2, l_3$  y  $l_4$  son las longitudes de los cuatro eslabones respectivos que conforman la pata.
- $\theta_1, \theta_2$  y  $\theta_3$  son los desplazamientos angulares de las articulaciones uno, dos y tres respectivamente.
- $\theta_4$  es el desplazamiento angular de la articulación final alrededor del eje paralelo al eje “z”.

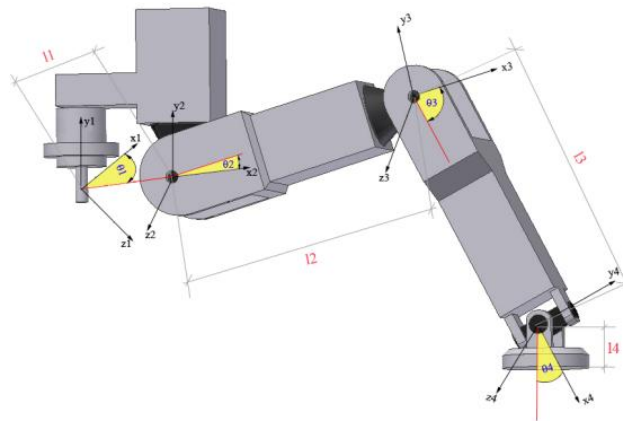


Figura 1.14. Esquema de una de las patas de REMLED

A diferencia de ROSEP, el sistema de referencia, respecto al cual se obtiene la posición del extremo de cada pata, se coloca de manera independiente en la unión de la pata con el cuerpo del robot.

En el caso de éste robot el modelo cinemático se realiza de manera inversa, es decir, se desea determinar la configuración que deben adoptar las articulaciones para situar el extremo final de la pata en una posición deseada. Como resultado de dicho análisis se tiene las ecuaciones (1.1-1.3).



$$\theta_1 = \operatorname{atan}^{-1} \frac{a_z}{a_x} \quad (1.1)$$

$$\theta_2 = \sin^{-1} \left[ \frac{l_3 \sin \theta_3}{\sqrt{(\sqrt{a_x^2 + a_z^2} - l_1)^2 + a_y^2}} \right] + \sin^{-1} \left[ \frac{a_y}{\sqrt{(\sqrt{a_x^2 + a_z^2} - l_1)^2 + a_y^2}} \right] \quad (1.2)$$

$$\theta_3 = \cos^{-1} \left[ \frac{(\sqrt{a_x^2 + a_z^2} - l_1)^2 + a_y^2 - l_2^2 - l_3^2}{2l_2 l_3} \right] \quad (1.3)$$

Los resultados expuestos en (Avila & Zambrano, 2007) son únicamente los que relacionan las posiciones y no incluyen las velocidades de las articulaciones en el robot. Debido a que el análisis anterior no expone los resultados de velocidad, no es posible considerarlo como un análisis cinemático del todo (recordar la definición de modelo cinemático al inicio de este apartado). Sin embargo, el desarrollo realizado es de ayuda para este trabajo de tesis puesto genera una idea de cómo definir sistemas de referencia y puntos de interés en el cuerpo del robot.

### ***Otros robots***

A lo largo de la revisión bibliográfica se encontraron diferentes métodos para obtener matemáticamente las relaciones de movimiento en el cuerpo de cada robot. Sin embargo, la mayoría coincide en obtener el modelo cinemático considerando cada una de las patas como una cadena cinemática abierta y como efector final el extremo de cada una de las patas; posteriormente se obtiene el comportamiento del efector final respecto a un sistema de referencia posicionado en el cuerpo del robot (preferentemente en el centro de masa).

## **1.4 Trayectorias en Robots con Patas**

El término de *generación de trayectorias* puede prestarse a confusión o a ambigüedad; de modo que es importante definir algunos conceptos que se manejan a lo largo de este documento.

Cuando se habla de **trayectoria temporal** (por omisión simplemente trayectoria) se está haciendo referencia a un camino que tiene asociado un perfil cinemático; es decir, a cada configuración perteneciente al camino se le asocia una velocidad (Gómez, et al.).

**Trayectoria aceptable o realizable**, se define como aquella que resulte libre de obstáculos, además de ser admisible desde el punto de vista cinemático (Muñoz, 1995).

Convencionalmente el problema de **generación de trayectorias** para un robot móvil indica que dado un robot y un ambiente, se puede determinar una trayectoria entre dos puntos específicos si el robot está libre de colisiones y si satisface cierto criterio de desempeño (Espitia, 2011).

Como se mencionó anteriormente, para la generación de trayectorias en un robot debe contarse con un *criterio de desempeño* (también conocido como *criterio de optimización*). Elegir un criterio de desempeño, depende de las necesidades que requieren ser cubiertas cuando el robot se desplaza de un punto a otro dentro de su espacio de trabajo. Como ejemplos de criterio de desempeño pueden citarse los siguientes: que el robot consuma la menor energía posible para llegar de un punto a otro; llegar en el menor tiempo posible; trazar una trayectoria forzosamente recta o en contraparte, compuesta de curvas suaves; recorrer la menor distancia posible para llegar de un punto a otro; etc.

Un ejemplo de la aplicación del criterio de optimización de energía, se encuentra en el trabajo realizado en (González P, et al., 2009). En dicho trabajo se busca la minimización de la energía que consume un robot hexápodo (llamado SILO 6) al realizar el caminado. Cabe mencionar que las trayectorias que se proponen en (González P, et al., 2009) no son las que realiza el robot para llegar de un punto a otro; sino las trayectorias del extremo de cada una de las patas respecto, a un sistema de referencia fijo en el centro de gravedad del robot.

Para obtener las ecuaciones de consumo de energía en el robot SILO 6, se hicieron ciertas consideraciones que hacen más sencilla la obtención del modelo energético. Como ejemplo de dichas consideraciones están; que el centro de gravedad del robot siempre se encuentra a una distancia constante de la superficie y que el robot se mueve a una velocidad constante. Finalmente mediante un algoritmo que calcula el consumo total de energía en base a la velocidad y peso del robot, se proponen las trayectorias que deben realizar las patas del robot.

En (Avila & Zambrano, 2007) se desarrolla un algoritmo de generación de trayectorias con diferente criterio. Este algoritmo tiene como finalidad llevar a un robot cuadrúpedo de un punto a otro en línea recta. Además de esto, el algoritmo debe lograr mantener la proyección del centro de gravedad (del cuerpo del robot) con la superficie, dentro de un triángulo formado por las tres patas que se encuentran en contacto con el piso en la transición de un paso a otro, figura 1.15.

En la mayoría de los trabajos de robots con patas, las trayectorias que se buscan son aquellas que realizan los extremos de las patas respecto a un sistema de referencia fijo en el cuerpo del robot. En el caso de los robots con ruedas, las trayectorias son respecto a un sistema inercial que se encuentra en el piso y sugieren el camino que el robot debe seguir para ir de un punto a otro sobre la superficie. Cabe mencionar que este último tipo trayectorias es el que se busca en este trabajo de tesis; es decir, se busca encontrar trayectorias que el robot sea capaz de realizar para llegar de una posición a otra en un plano. Por lo que también se realizó una revisión de algunos trabajos realizados con robot con ruedas.

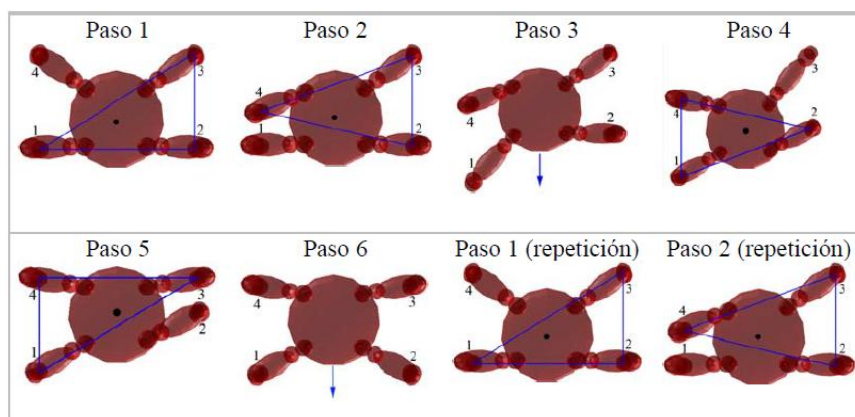


Figura 1.15. Algoritmo de locomoción del robot cuadrúpedo

Para los robots móviles con ruedas, lo más común es usar métodos geométricos para obtener las trayectorias que deben seguir. Como ejemplo de esto están (Gómez, et al.), (Lamiroux & Laumond, 2001), (Laumond, 1998). El método geométrico consiste, a grandes rasgos, en proponer trayectorias curvilíneas que el robot sea capaz de seguir dependiendo de las restricciones que él mismo posee debido a su arquitectura. El criterio de optimización que se maneja para este método, consiste en minimizar la distancia que el robot debe recorrer para llegar de un punto a otro en el plano.

### ***Discusión***

En cuanto al modelado de robots con patas, la mayoría de trabajos revisados coinciden en obtener las ecuaciones que relacionan las posiciones y velocidades de las articulaciones presentes en cada una de las patas. De esta manera, es posible conocer el comportamiento del extremo de cada una de las patas respecto a un sistema de referencia que se encuentra en el cuerpo del robot. Al conjunto de ecuaciones que resultan de tal análisis se le considera como único modelo de interés, es decir, no se modela el desplazamiento que tiene el robot sobre la superficie. Como se verá más adelante, en el presente trabajo se realiza la misma metodología; se obtiene una ecuación que describe el comportamiento del extremo de cada pata respecto a un marco de referencia ubicado en el centro del robot. Sin embargo, dicha ecuación se utiliza para modelar el desplazamiento que tiene el robot sobre una superficie.

En la revisión bibliográfica para la generación de trayectorias en robots con patas, se encontró que las trayectorias de interés son aquellas que trazan cada una de las patas del robot respecto a un sistema inercial dentro del cuerpo del mismo. Estas trayectorias varían según el comportamiento de los actuadores en cada pata. En el caso de este trabajo, dichas trayectorias no pueden variar debido a que las patas del robot se están compuestas por cadenas cinemáticas cerradas. Además, las trayectorias que son de interés para este proyecto, son aquellas que el robot describe al desplazarse sobre una superficie. De manera que los procedimientos descritos en otros trabajos de robots con patas, no resultan de utilidad para el objetivo de esta tesis.



# Capítulo 2. Modelado Cinemático

En este capítulo se presenta la descripción del robot utilizado como referencia para este trabajo de tesis. De igual manera se presenta el desarrollo para la obtención del modelo cinemático de dicho robot

## 2.1 Descripción del Robot

### 2.1.1 Arquitectura

Para la realización de este trabajo se toma como referencia un micro robot de juguete de la marca HEXBUG®, figura 2.1. El robot está compuesto por 3 patas. En la figura 2.2 se muestra como están consideradas y numeradas cada una de ellas.



Figura 2.1. Robot de juguete marca HEXBUG®

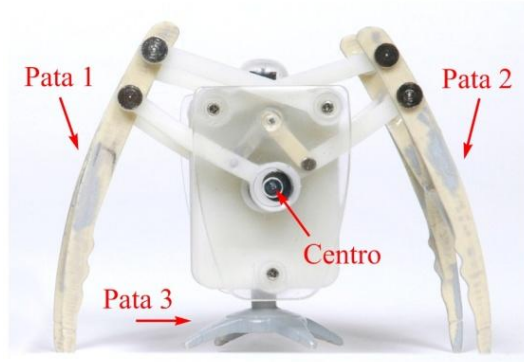


Figura 2.2. Descripción general del robot

A continuación se realiza la descripción de cómo está compuesta cada una de las patas del robot. En el caso de las patas 1 y 2 basta con describir solo una de ellas ya que están construidas de la misma manera.

### ***Pata 1***

Observando el robot se ve que la pata 1 está compuesta por dos mecanismos; uno de 4 barras unido a otro de 5 barras; en este último se encuentra el efector final (el extremo de la pata).

En la figura 2.3 se muestra como fue considerado el mecanismo de 4 barras. Las barras 1,2 y 3 (en rojo) son eslabones móviles, el eslabón número 4 (azul) se considera como la barra de *tierra* del mecanismo.

El mecanismo de 5 barras se muestra en la figura 2.4. Las barras 1 a 4 (rojo) son eslabones móviles y la barra número 5 se refiere la tierra del mecanismo. El efector final (negro) se encuentra pegado al eslabón número 3 y va desde el mismo eslabón hasta el piso.

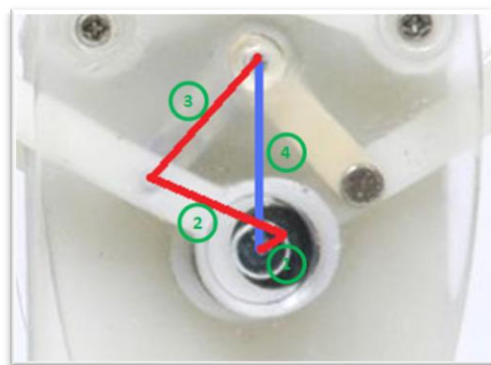


Figura 2.3. Mecanismo de 4 barras

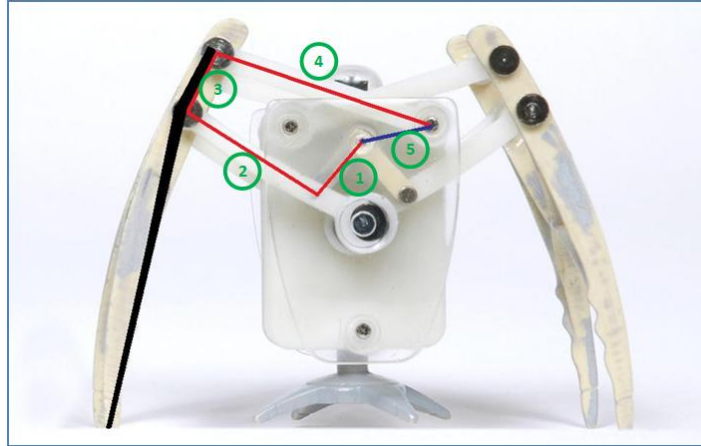


Figura 2.4. Mecanismo de 5 barras y efector final

Para el mecanismo de 5 barras existe una propuesta diferente a la que se muestra en la figura 2.4. Esta propuesta consiste en tomar el centro del robot como origen del mecanismo de 5 barras. De esta manera se tiene que el eslabón 1 es el mismo en ambos mecanismos (4 y 5 barras) y que el eslabón 2 en el mecanismo de 4 barras forma parte del eslabón 2 en el mecanismo de 5 barras. Para fines prácticos, puede usarse cualquiera las dos propuestas para este mecanismo. Esto debido a que no se hace presente ninguna ventaja o desventaja al trabajar con una u otra propuesta en particular.

El mecanismo de 4 barras en el robot posee un excéntrico como eslabón actuador, ver figura 2.5. El eje de acción se encuentra en el centro del círculo con mayor radio. La distancia entre los centros de ambas circunferencias se considera como la medida del primer eslabón (eslabón actuador) del mecanismo de 4 barras.

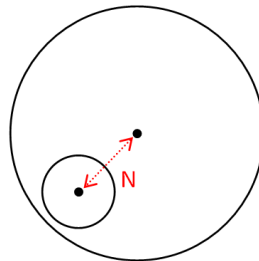


Figura 2.5. Excéntrico. La distancia N se considera como medida del eslabón actuador

Cabe mencionar que en realidad la pata 1 posee otro par de mecanismos (4 y 5 barras), los cuales no están visibles en la figura 2.2 debido a que se encuentran en contraparte a la cara mostrada en la misma figura. Entendiendo por cara a las dos vistas que se tienen del robot considerando la figura 2.2 y un giro de  $180^\circ$  de la misma respecto a la vertical. Sin embargo, sólo se considera un mecanismo de 4 y uno de 5 barras para la pata, debido a que cada mecanismo está unido rígidamente a su correspondiente en la cara contraria (el de 4 barras con el otro de 4 barras y el de 5 con el otro de 5).

Otro aspecto importante son los puntos de apoyo que posee la pata 1; es posible que cada punto de apoyo llegue a confundirse como una pata por sí sólo, sin embargo se trata de varios puntos de apoyo de un mismo cuerpo rígido. En la figura 2.6 se muestran las uniones rígidas y los puntos de apoyo mencionados.

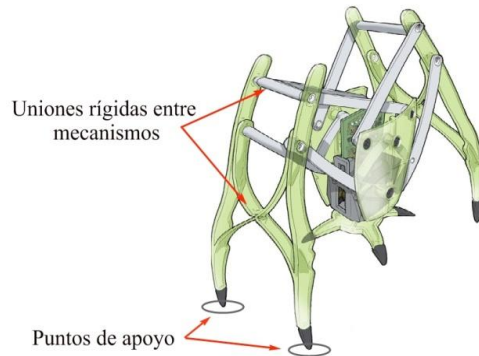


Figura 2.6. Uniones rígidas y puntos de apoyo

### ***Pata 2***

Como se mencionó anteriormente, la pata 2 del robot está compuesta de la misma forma que la pata 1. Las consideraciones, consecuencia de las uniones rígidas y puntos de apoyo de la pata 1, fueron tomadas de la misma forma en la pata 2.

### ***Pata 3***

La pata 3 es un mecanismo más sencillo comparado con las otras dos patas. Es un mecanismo de un eslabón unido al cuerpo del robot con una junta de revolución. En la figura 2.7 se observa el mecanismo de la pata 3.

Físicamente esta pata cuenta con 4 puntos de apoyo, sin embargo se considera como una sola pata debido a que se trata de un solo cuerpo rígido. Cabe mencionar que el centro de gravedad del robot se encuentra sobre la recta colineal al eje longitudinal de la pata.

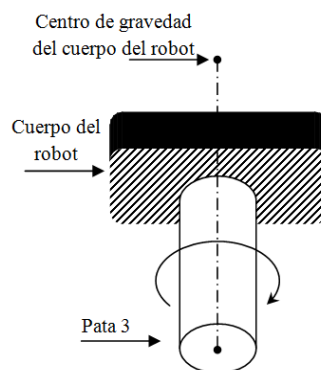


Figura 2.7. Mecanismo considerado en la pata 3



## 2.1.2 Movimiento

Ahora se describen los movimientos que realiza el robot. Las patas 1 y 2 están controladas por un solo motor. El eje de dicho motor se encuentra en el centro del cuerpo del robot (figura 2.2). Por lo tanto, las patas 1 y 2 no pueden moverse de manera independiente.

El movimiento que producen los extremos de las patas 1 y 2 es semi-elíptico y es el que permite el avance lineal del robot. En figura 2.8 se ilustra el tipo de movimiento que describen las patas de avance del robot (por *patas de avance* se refiere a las patas 1 y 2).

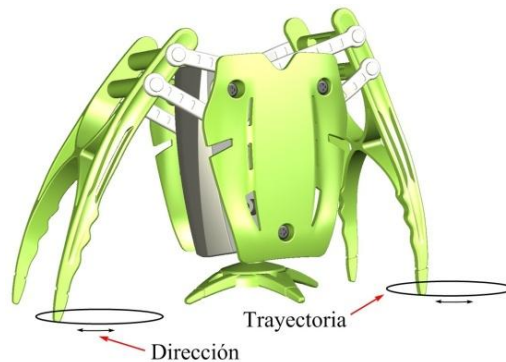


Figura 2.8. Trayectoria descrita por el extremo de las patas 1 y 2

Las patas de avance siempre presentan la misma dirección de desplazamiento con respecto al cuerpo del robot.

La pata 3 es responsable de dos funciones en el caminado del robot:

1. Es la que proporciona el soporte del robot sobre la superficie de desplazamiento cuando las patas de avance se encuentran recorriendo la parte superior de la elipse, es decir, cuando no están apoyadas sobre la superficie.
2. Se encarga del giro del cuerpo del robot cuando es la única pata en contacto con la superficie.

Descritos los movimientos anteriores es posible imaginar que el robot se desplaza como una persona usando muletas, lo cual puede verse como un caminado discreto. Las trayectorias que describe el robot al caminar se componen de *pasos* de magnitud constante y giros. Por *paso* se refiere al movimiento de traslación que realiza el cuerpo del robot. Este movimiento se presenta cuando las patas de avance recorren la parte inferior de la semi-elipse y se encuentran apoyadas sobre la superficie.

Es claro que el robot analizado en este trabajo presenta una mayor cantidad de restricciones en su desplazamiento que un robot móvil con ruedas tradicional e inclusive que otros robots con patas. A diferencia de otros robots móviles este no puede girar y avanzar al mismo tiempo. Sin embargo,

cabe mencionar que es capaz de desplazarse sobre una superficie utilizando únicamente dos motores, uno para avance en línea recta y otro para girar, a diferencia de la mayoría de los robots con patas que necesitan una gran número de actuadores para poder realizar la tarea de caminado.

El caminado que presenta el robot brinda una idea de las trayectorias que es capaz o no de realizar. Sin embargo, hay que recordar la definición de trayectoria realizable, la cual dice que una trayectoria realizable es aquella que resulte libre de obstáculos y es admisible desde el punto de vista cinemático. De manera que para la generación de trayectorias deben considerarse las restricciones de velocidad que son propias del robot

Para obtener las restricciones de velocidad se recurre al modelado cinemático, el cual describe matemáticamente las relaciones de velocidad que están presentes en el cuerpo del robot.

## **2.2 Modelado Cinemático**

En esta sección se busca obtener un modelo cinemático del robot el cual incluya o ponga en evidencia las restricciones del robot. En particular, el modelo cinemático deseado consiste en las ecuaciones de movimiento del cuerpo del robot en el plano horizontal.

Las patas 1 y 2 son las responsables de proporcionarle la velocidad de avance y retroceso al cuerpo del robot. Suponiendo que no hay deslizamiento entre los apoyos de estas patas y el piso, se puede relacionar la velocidad del cuerpo del robot referenciada al piso con la velocidad de las patas referenciadas al cuerpo del robot. Para encontrar la velocidad en las patas del robot se debe obtener primero las relaciones geométricas en cada una de ellas para después calcular sus derivadas.

Es posible analizar las patas 1 y 2 de manera independiente puesto que no comparten eslabones entre sí, sólo comparten el eje de acción donde está sujeto el eslabón actuador de cada pata, por lo que inicialmente se obtiene la relación geométrica en la pata 1.

### **2.2.1 Análisis de la Pata 1**

Hay que recordar que la pata 1 está compuesta por dos mecanismos; uno de 4 y uno de 5 barras. Los movimientos de los eslabones en el mecanismo de 4 barras definen el comportamiento del mecanismo de 5 barras, esto debido a los eslabones que comparten ambos mecanismos. Por lo cual se debe conocer primero y en su totalidad el comportamiento del mecanismo de 4 barras.

A continuación se presenta en análisis realizado para cada uno de éstos dos mecanismos. Cabe mencionar que la cinemática de los mecanismos (4 y 5 barras) se resuelve de manera clásica y solamente se recuerdan esas soluciones aplicadas a los mecanismos del robot.

### 2.2.1.1 Mecanismo de 4 Barras

Para obtener el modelo cinemático en éste mecanismo se debe calcular el número de grados de libertad en el mismo. El número de grados de libertad (*Ngdl*) en un sistema físico, se refiere al número mínimo de parámetros que es necesario especificar para determinar completamente su estado.

Para calcular el número de grados de libertad se recurre al criterio de *Grübler – Kutzbach* (Norton, 2000) para mecanismos planos que dice:  $Ngdl = 3(n-1) - 2j_1 - j_2$ . Donde  $n$  es el número total de eslabones en el mecanismo,  $j_1$  es el número de uniones de un grado de libertad y  $j_2$  es el número de uniones de 2 grados de libertad.

En el mecanismo de 4 barras se tiene que  $n = 4$  por los cuatros eslabones que componen el mecanismo (incluyendo la *tierra* que es la parte fija del mecanismo),  $j_1 = 4$  puesto que el mecanismo posee 4 juntas de revoluta (las cuales son uniones con un grado de libertad) y  $j_2 = 0$  ya que en el mecanismo no existen uniones con 2 grados de libertad. Sustituyendo los valores anteriores en la ecuación de *Grübler – Kutzbach*, se tiene como resultado:  $Ngdl = 3(4-1) - 2(4) - 0 = 1$ .

Con lo anterior se sabe que debe existir una variable de entrada conocida para poder determinar las velocidades en el mecanismo de 4 barras. La variable conocida es la velocidad de revolución del excéntrico en el mecanismo, puesto que es la única que le proporciona movimiento.

En la figura 2.9 se muestra como son considerados los sentidos de giro en cada eslabón (flechas rojas), así como el nombre que se le da a cada una de las longitudes de los eslabones y los ángulos que poseen. Notar que los ángulos de los eslabones están medidos de manera absoluta y respecto a la vertical. Las longitudes L, M, N y P de los eslabones son constantes.

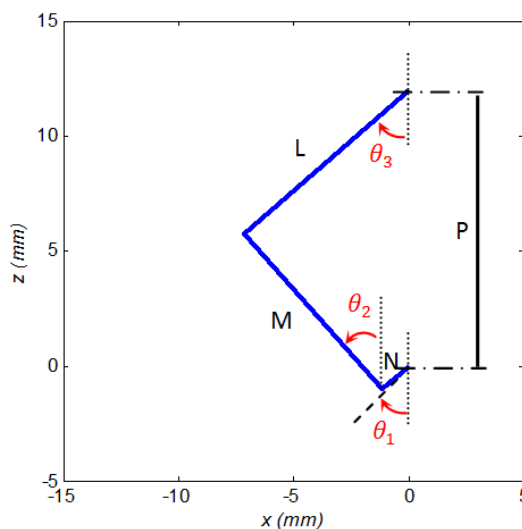


Figura 2.9. Longitudes y ángulos en el mecanismo de 4 barras

Ahora se plantean las ecuaciones de restricción del mecanismo mediante la proyección de los eslabones sobre los ejes horizontal y vertical (ecuaciones (2.1-2.3)).

$$\theta_1 - \epsilon = 0 \quad (2.1)$$

$$N \sin \theta_1 + M \sin \theta_2 - L \sin \theta_3 = 0 \quad (2.2)$$

$$N \cos \theta_1 - M \cos \theta_2 - L \cos \theta_3 + P = 0 \quad (2.3)$$

A la orientación del eslabón actuador se le llama  $\theta_1$ . Dicha orientación posee un valor conocido  $\epsilon$  puesto que pertenece a la entrada con la cual se controla el mecanismo, en otras palabras  $\theta_1$  es nuestra variable de control y se le puede asignar un valor conocido  $\epsilon$  entre  $0^\circ$  y  $360^\circ$ .

Resolviendo el sistema de ecuaciones anterior para los ángulos asociados a los eslabones se tiene:

$$\theta_1 = \epsilon \quad (2.4)$$

$$\theta_2 = 2 \arctan \left( \frac{L \sin \theta_3 - N \sin \theta_1}{M + N \cos \theta_1 - L \cos \theta_3 + P} \right) \quad (2.5)$$

$$\theta_3 = 2 \arctan \left( \frac{2LN \sin \theta_1 + \sqrt{(-2LN \sin \theta_1)^2 + (-2L(N \cos \theta_1 + P))^2 - (2NP \cos \theta_1 + \Omega)^2}}{2NP \cos \theta_1 + \Omega + 2L(N \cos \theta_1 + P)} \right) \quad (2.6)$$

Con  $\Omega = L^2 + P^2 + M^2 + N^2$ ; la cual es constante debido a que se consideran eslabones rígidos y por tanto sus medidas son constantes.

Es importante resaltar que  $\theta_2$  está en términos de  $\theta_1$  y  $\theta_3$ ; a su vez  $\theta_3$  está únicamente en términos de  $\theta_1$ . Lo cual quiere decir que ambas variables dependen únicamente de  $\theta_1$ , la entrada de control.

Para encontrar las velocidades en el mecanismo de 4 barras, se derivan las ecuaciones de restricción del mismo, ecuaciones (2.1-2.3), de lo cual se obtiene:

$$\dot{\theta}_1 - \dot{\epsilon} = 0 \quad (2.7)$$

$$N \dot{\theta}_1 \cos \theta_1 + M \dot{\theta}_2 \cos \theta_2 - L \dot{\theta}_3 \cos \theta_3 = 0 \quad (2.8)$$

$$-N \dot{\theta}_1 \sin \theta_1 + M \dot{\theta}_2 \sin \theta_2 + L \dot{\theta}_3 \sin \theta_3 = 0 \quad (2.9)$$

Resolviendo el sistema de ecuaciones (2.7-2.9) para las velocidades  $\dot{\theta}_1$ ,  $\dot{\theta}_2$  y  $\dot{\theta}_3$  se tiene:

$$\dot{\theta}_1 = \dot{\epsilon} \quad (2.10)$$

$$\dot{\theta}_2 = \frac{N \dot{\theta}_1 \sin(\theta_1 - \theta_3)}{M \sin(\theta_2 + \theta_3)} \quad (2.11)$$

$$\dot{\theta}_3 = \frac{N\dot{\theta}_1 \sin(\theta_1 + \theta_2)}{L \sin(\theta_2 + \theta_3)} \quad (2.12)$$

Las velocidades  $\dot{\theta}_2$  y  $\dot{\theta}_3$  dependen únicamente de la posición y velocidad del eslabón actuador, es decir  $\theta_1$  y  $\dot{\theta}_1$ . Los resultados se presentan en función de más variables ya que si se expresaran en términos únicamente de  $\theta_1$  y  $\dot{\theta}_1$  representaría una ecuación mucho más extensa y difícil de interpretar.

### 2.2.1.2 Mecanismo de 5 Barras

Ahora que son conocidas las posiciones y velocidades del mecanismo de 4 barras, se obtendrán las ecuaciones de restricción para el mecanismo de 5 barras.

Se observa que al igual que en el mecanismo de 4 barras, la orientación de cada eslabón es medida de manera absoluta y respecto a la vertical, como se indica en la figura 2.10. Las longitudes de los eslabones A, B, D, E, L y R son constantes.

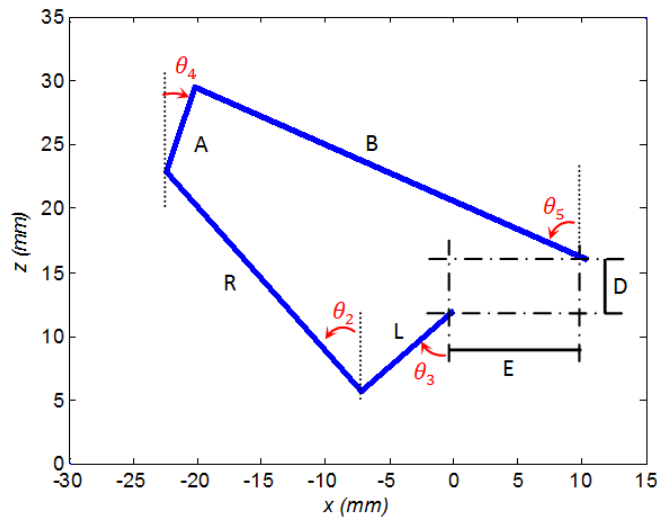


Figura 2.10. Longitudes y ángulos en el mecanismo de 5 barras

Es importante mencionar que el eslabón M (en el mecanismo de 4 barras) y el eslabón R son en realidad un solo eslabón rígido de magnitud  $M + R$ , por lo tanto el eslabón R tendrá la misma orientación que el eslabón M (la cual se resolvió en el análisis anterior).

También conviene mencionar que el eslabón L forma parte de ambos mecanismos (4 y 5 barras), por lo tanto, la orientación ( $\theta_3$ ) y velocidad ( $\dot{\theta}_3$ ) ya fueron resultados en el análisis del mecanismo de 4 barras.

Las ecuaciones de restricción para el mecanismo de 5 barras son las siguientes:

$$-R \sin \theta_2 - L \sin \theta_3 + A \sin \theta_4 + B \sin \theta_5 - E = 0 \quad (2.13)$$

$$R \cos \theta_2 - L \cos \theta_3 + A \cos \theta_4 - B \cos \theta_5 - D = 0 \quad (2.14)$$

Hay que recordar que  $\theta_2$  y  $\theta_3$  fueron resueltos en el análisis anterior, por lo tanto son valores ya conocidos. Resolviendo para  $\theta_4$  y  $\theta_5$  de las ecuaciones (2.13) y (2.14) se tiene:

$$\theta_4 = 2 \arctan \left( \frac{L \sin \theta_3 + R \sin \theta_2 - B \sin \theta_5 + E}{L \cos \theta_3 - R \cos \theta_2 + B \cos \theta_5 + D + A} \right) \quad (2.15)$$

$$\theta_5 = 2 \arctan \left( \frac{-a_5 - \sqrt{a_5^2 + b_5^2 - c_5^2}}{c_5 - b_5} \right) \quad (2.16)$$

en donde

$$a_5 = -2B(R \sin \theta_2 + L \sin \theta_3 + E)$$

$$b_5 = 2B(L \cos \theta_3 - R \cos \theta_2 + D)$$

$$c_5 = 2L(E \sin \theta_3 + D \cos \theta_3) + 2R(E \sin \theta_2 - D \cos \theta_2) - 2LR \cos(\theta_2 + \theta_3) + \Psi$$

$$\Psi = L^2 + R^2 + B^2 + E^2 + D^2 - A^2$$

Nótese que  $\Psi$  es un término de valor constante, usado únicamente para simplificar la expresión de  $c_5$ .

$\theta_4$  y  $\theta_5$  también dependen únicamente de  $\theta_1$ , pero se expresan en términos de las otras variables para mostrar una expresión más reducida.

Para obtener las velocidades en el mecanismo de 5 barras se derivan las ecuaciones (2.13) y (2.14), dando como resultado las ecuaciones (2.17) y (2.18) respectivamente.

$$-R\dot{\theta}_2 \cos \theta_2 - L\dot{\theta}_3 \cos \theta_3 + A\dot{\theta}_4 \cos \theta_4 + B\dot{\theta}_5 \cos \theta_5 = 0 \quad (2.17)$$

$$-R\dot{\theta}_2 \sin \theta_2 + L\dot{\theta}_3 \sin \theta_3 - A\dot{\theta}_4 \sin \theta_4 + B\dot{\theta}_5 \sin \theta_5 = 0 \quad (2.18)$$

Resolviendo el sistema de ecuaciones anterior para las variables  $\dot{\theta}_4$  y  $\dot{\theta}_5$  se tiene:

$$\dot{\theta}_4 = \frac{L\dot{\theta}_3 \sin(\theta_3 + \theta_5) + R\dot{\theta}_2 \sin(\theta_5 - \theta_2)}{A \sin(\theta_4 + \theta_5)} \quad (2.19)$$

$$\dot{\theta}_5 = \frac{L\dot{\theta}_3 \sin(\theta_4 - \theta_3) + R\dot{\theta}_2 \sin(\theta_2 + \theta_4)}{B \sin(\theta_4 + \theta_5)} \quad (2.20)$$

Ahora ya se conocen las posiciones y velocidades en los mecanismos que componen una de las dos patas de avance. Sin embargo, aún no se ha considerado la parte donde la pata hace contacto con la superficie la cual se considera el *efector final* del mecanismo de 5 barras.

En la figura 2.11 se observan ya acoplados los mecanismos de 4 y 5 barras, además se señala el efector final.

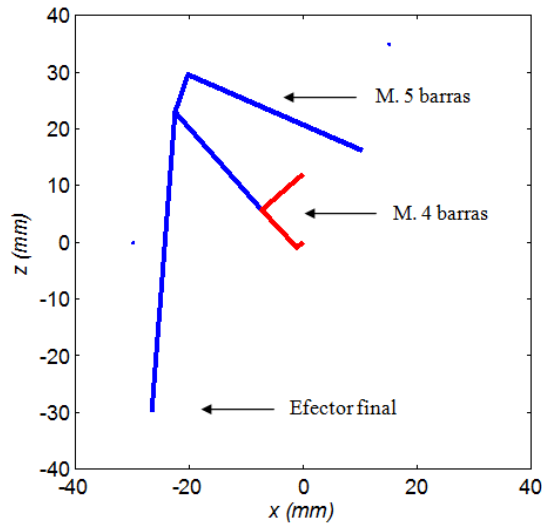


Figura 2.11. Mecanismos acoplados y efector final

Para una mejor comprensión de la figura anterior, es recomendable observar la figura 2.4 donde se aprecia la forma de los eslabones que componen ambos mecanismos. Haciendo la comparación de ambas figuras, se puede observar que el eslabón A del mecanismo de 5 barras y el efector final son parte de un mismo cuerpo rígido.

Pasando ahora al análisis del efector final; se desea conocer la posición y velocidad de dicho efector. Para obtener la posición del efector final se traza un vector ( $\vec{r}_{p1}$ ) que va desde la junta de revoluta del eslabón actuador (considerada como el origen en el mecanismo de 4 barras) hasta el final de la pata 1 del robot figura 2.12. El vector  $\vec{r}_{p1}$  queda descrito en la ecuación (2.21).

$$\vec{r}_{p1} = [-N \sin \theta_1 - (R + M) \sin \theta_2 - F \sin(\theta_4 - \delta)]\mathbf{i} + [-N \cos \theta_1 + (R + M) \cos \theta_2 - F \cos(\theta_4 - \delta)]\mathbf{k} \quad (2.21)$$

Notar que  $\delta$  es un ángulo constante debido a que A y F son parte de un mismo cuerpo rígido.

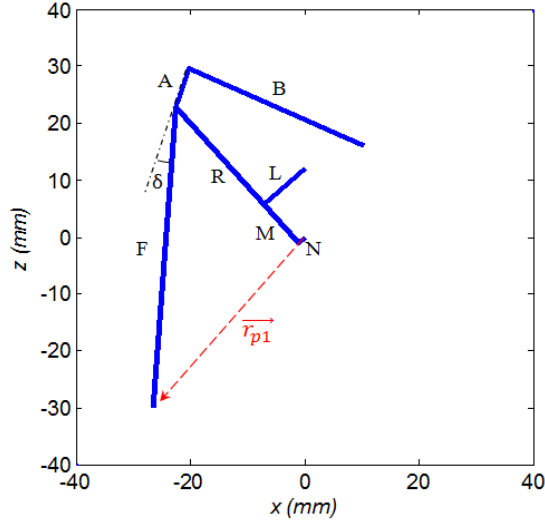


Figura 2.12. Vector de posición  $\vec{r}_{p1}$

Para conocer la velocidad que se imprime al robot cuando del efector final está en contacto con la superficie, se calcula la primera derivada del vector de posición (2.21) y de ésta manera se obtiene el vector de velocidad  $\dot{\vec{r}}_{p1}$  (2.22).

$$\begin{aligned} \dot{\vec{r}}_{p1} = & [-N\dot{\theta}_1 \cos \theta_1 - (R+M)\dot{\theta}_2 \cos \theta_2 - F\dot{\theta}_4 \cos(\theta_4 - \delta)]i + \\ & [N\dot{\theta}_1 \sin \theta_1 - (R+M)\dot{\theta}_2 \sin \theta_2 - F\dot{\theta}_4 \sin(\theta_4 - \delta)]k \end{aligned} \quad (2.22)$$

Con esto se conoce finalmente el comportamiento de la pata 1 en función de la única entrada de control que posee.

## 2.2.2 Análisis de la Pata 2

La pata 2 está compuesta de la misma manera que la pata 1; un mecanismo de 4 y uno de 5 barras más un efector final, y con mismas magnitudes en los eslabones. De manera que los resultados obtenidos al analizar la pata 1 pueden ser utilizados para la pata 2 bajo ciertas consideraciones:

- Los mecanismos de 4 barras en ambas patas comparten el mismo origen y el mismo actuador.
- El sentido de giro del eslabón actuador de la pata 2, siempre será el mismo que el sentido de giro en el actuador de la pata 1, puesto que ambos se encuentran en la misma pieza excéntrica, figura 2.13.
- La variable de control para la pata 2 ( $\phi_1$ ), siempre estará desfasada  $180^\circ$  de la variable de control en la pata 1 ( $\theta_1$ ).



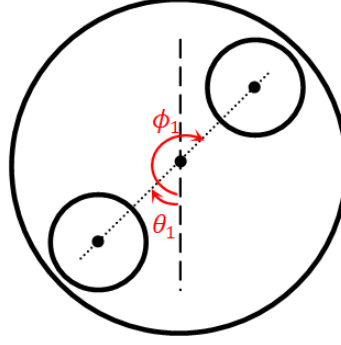


Figura 2.13. Excéntrico en el origen de las patas 1 y 2

Teniendo en cuenta las consideraciones anteriores se encuentra que la relación entre ambas variables de control es:  $\phi_1 = \theta_1 + 180^\circ$ .

Del análisis en la pata 1, se sabe que todas las posiciones  $\theta_i$  ( $i=1,2,3,4,5$ ) dependen de la entrada de control  $\theta_1$ , de manera que los movimientos en la pata 2 estarán en función únicamente de  $\phi_1$ . Los ángulos restantes en los mecanismos de 4 y 5 barras de la pata 2 tendrán por nombre  $\phi_i$  y serán nombrados en el mismo orden que fueron nombrados los ángulos  $\theta_i$  en la pata 1. Ahora se muestran los resultados más relevantes del análisis en la pata 2.

El vector de posición que va desde el origen del mecanismo de 4 barras hacia la parte inferior de la pata 2, es llamado  $\vec{r}_{p2}$  y se está descrito por la siguiente ecuación:

$$\begin{aligned} \vec{r}_{p2} = & [N \sin \phi_1 + (R + M) \sin \phi_2 + F \sin(\phi_4 - \delta)]i + \\ & [-N \cos \phi_1 + (R + M) \cos \phi_2 - F \cos(\phi_4 - \delta)]k \end{aligned} \quad (2.23)$$

Para conocer la velocidad del efector final en la pata 2, se deriva  $\vec{r}_{p2}$  y así se obtiene:

$$\begin{aligned} \dot{\vec{r}}_{p2} = & [N \dot{\phi}_1 \cos \phi_1 + (R + M) \dot{\phi}_2 \cos \phi_2 + F \dot{\phi}_4 \cos(\phi_4 - \delta)]i + \\ & [N \dot{\phi}_1 \sin \phi_1 - (R + M) \dot{\phi}_2 \sin \phi_2 + F \dot{\phi}_4 \sin(\phi_4 - \delta)]k \end{aligned} \quad (2.24)$$

Las ecuaciones que describen la posición de los eslabones ( $\theta_i$  y  $\phi_i$ ) fueron introducidas en Matlab para crear una simulación de movimiento de las patas del robot. La medida de cada eslabón fue tomada del prototipo.

Para la simulación, la entrada de control  $\theta_1$  tomó valores entre  $0^\circ$  y  $360^\circ$  con incrementos de 1 grado. El centro de gravedad del robot se toma como referencia inercial y las patas se mueven libremente. En la figura 2.14 pueden verse los mecanismos que componen ambas patas del robot y la trayectoria que realizan los dos efectores finales (elipse punteada) en un ciclo donde  $\theta_1$  va de  $0^\circ$  a  $360^\circ$ . La línea horizontal indica la altura a la cual se encuentra la pata de giro.

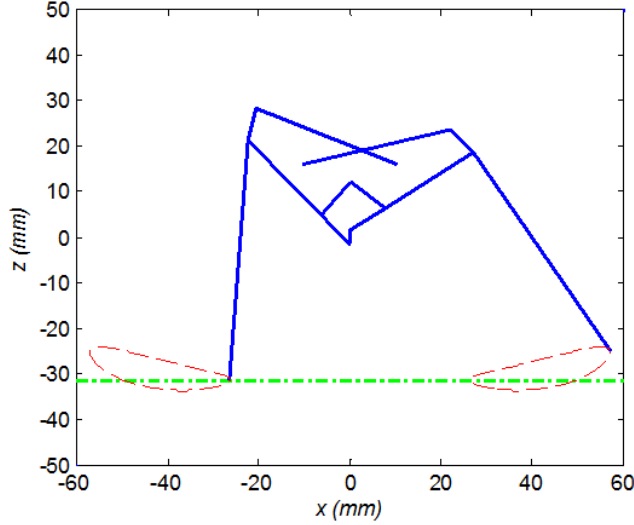


Figura 2.14. Simulación de las patas de avance del robot

El código que se generó para esta simulación se encuentra en el anexo A de este documento, así como las medidas que fueron utilizadas para eslabón.

## 2.2.3 Ecuaciones del Robot

Después de los análisis anteriores, finalmente se está en la posibilidad de modelar el movimiento del cuerpo del robot, que es el modelado buscado. Para ello se considera el caso cuando las patas de avance están apoyadas (fijas) en el piso y es el cuerpo del robot el que se desplaza. Asumiendo que no hay deslizamiento, el vector  $-\vec{r}_{p1}$ , que va de la punta de la pata de avance al cuerpo del robot, y su derivada temporal nos describen el movimiento deseado. De este movimiento particularmente interesa la velocidad lineal  $v$  (2.25) del cuerpo del robot que está asociada con la proyección del vector de velocidad (2.22) sobre el plano horizontal (componente  $i$  con signo negativo). La componente  $k$  en (2.22) indica la velocidad a la que se modifica la altura del cuerpo del robot y no aporta información para el modelo de desplazamiento sobre la superficie, es por ello que se desprecia en los análisis posteriores.

$$v = N\dot{\theta}_1 \cos \theta_1 + (R + M)\dot{\theta}_2 \cos \theta_2 + F\dot{\theta}_4 \cos(\theta_4 - \delta) \quad (2.25)$$

Recordando que el robot avanzará solamente cuando las patas de avance estén apoyadas en el piso (la parte inferior del ciclo ilustrado en la figura 2.14) es necesario describir apropiadamente esa situación, comenzando por determinar el intervalo de  $\theta_1$  que corresponde a este caso. Asumiendo, que las patas de avance se apoyan sobre el piso precisamente cuando (2.25) es igual a cero. Dada la alta no linealidad de los términos involucrados en  $v$  se determinó numéricamente que esto se cumple para

$$\theta_1 = 5.2445^\circ = \rho_1 \quad \text{y} \quad \theta_1 = 184.6300^\circ = \rho_2 \quad (2.26) \text{ y } (2.27)$$

Con  $v_1$  como la velocidad de avance y eligiendo una dirección de giro  $\theta_1$  como positiva (sentido horario) se puede escribir

$$v_1 = \begin{cases} v, & 0^\circ \leq \theta_1 \leq \rho_1 \cup \rho_2 \leq \theta_1 \leq 360^\circ \\ 0, & \rho_1 < \theta_1 < \rho_2 \end{cases} \quad (2.28)$$

Para modelar la velocidad de giro del robot se coloca un marco de referencia  $F_r(O_r, \mathbf{i}_r, \mathbf{j}_r)$  con  $O_r$  coincidente con el centro de gravedad del cuerpo del robot; el eje  $\mathbf{i}_r$  en su dirección de avance lineal y el plano  $\mathbf{i}_r\mathbf{j}_r$  paralelo al piso, figura 2.15. En el piso se coloca un marco de referencia inercial  $F_o(O_o, \mathbf{i}_o, \mathbf{j}_o)$  con el plano  $\mathbf{i}_o\mathbf{j}_o$  paralelo a la superficie.  $x_r, y_r$  son la coordenadas del centro del robot en el marco de referencia inercial.

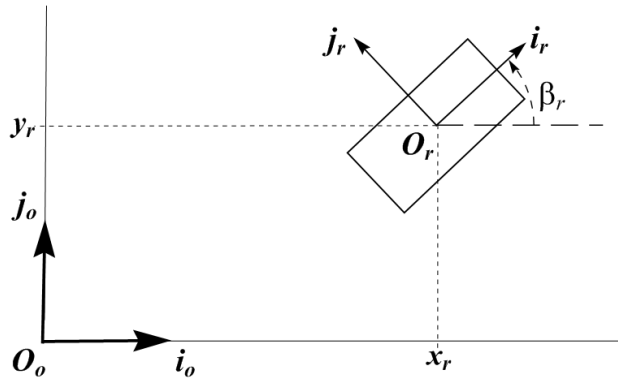


Figura 2.15. Sistemas de referencia fijo y en el robot

Sea  $\beta_r$  el ángulo del eje  $\mathbf{i}_r$  con respecto a  $\mathbf{i}_o$  y  $\omega$  la velocidad angular de la pata de giro, la velocidad de giro del robot está dada por

$$\dot{\beta}_r = \begin{cases} 0, & 0^\circ \leq \theta_1 \leq \rho_1 \cup \rho_2 \leq \theta_1 \leq 360^\circ \\ -\omega, & \rho_1 < \theta_1 < \rho_2 \end{cases} \quad (2.29)$$

la superficie, la velocidad de giro de la tercera pata provoca el giro del cuerpo del robot pero en sentido contrario. Es por ello que cuando la velocidad angular  $\beta_r$  no es cero tiene la misma magnitud que la velocidad  $\omega$  pero con sentido contrario.

Sean  $x_r, y_r$  las coordenadas de  $O_r$  (es decir el centro del robot) en el marco  $F_o$ . Con esta notación el modelo cinemático del robot puede escribirse como

$$\begin{aligned} \dot{x}_r &= v_1(\theta_1, \dot{\theta}_1) \cos \beta_r \\ \dot{y}_r &= v_1(\theta_1, \dot{\theta}_1) \sin \beta_r \\ \dot{\beta}_r &= \dot{\beta}_r(\theta_1) \end{aligned} \quad (2.30)$$

Con el sistema de ecuaciones anterior (2.30) se cumple uno de los objetivos, el cual era proponer un modelo cinemático para el robot móvil en este trabajo.

Es conveniente mencionar algunas características del modelo anterior. Las variables  $\dot{x}_r$  y  $\dot{y}_r$  son las componentes de la velocidad lineal con la cual el robot se desplaza sobre el piso. Dentro de la expresión para estas variables se encuentra  $v_1$ , la cual está condicionada (por un rango de valores de  $\theta_1$ ) a ser o no diferente de cero. Tal condición también existe para la velocidad con que puede girar el robot aunque manera inversa. Dicho de otra manera, mientras que exista una velocidad de avance diferente de cero, la velocidad de giro debe ser igual a cero, y mientras que exista una velocidad de giro diferente de cero, el robot no puede avanzar linealmente.

Las ecuaciones en (2.30) tienen cierto parecido con el modelo cinemático de un robot móvil sobre ruedas de tipo unicycle. Sin embargo, las expresiones de las velocidades son muy diferentes. En particular, la velocidad  $v_1$  del robot con patas es altamente no lineal en  $\theta_1$  y en la variable de control  $\dot{\theta}_1$ , mientras que en el unicycle las velocidades son variables de control directamente.

Aunque (2.30) representa bien el comportamiento del robot, las ecuaciones no son muy manejables para análisis posteriores. Asimismo también puede consumir recursos de cómputo que puedan resultar significativos para algunas aplicaciones como el control en tiempo real del robot. Por lo anterior se requiere un modelo simplificado del robot, particularmente en el caso de la expresión de  $v_1$ .

Para lograr una simplificación de  $v_1$ , se utilizaron métodos analíticos. Sin embargo, dada la complejidad de la expresión de  $v_1$ , no fue posible obtener dicha simplificación por estos métodos. Por lo anterior se optó por obtener una función más sencilla y que se aproximara de manera considerable a  $v_1$ .

### 2.2.3.1 Modelo Cinemático Simplificado

Para poder encontrar un modelo más simple y que siguiese representando una buena aproximación para el movimiento del robot, se optó por analizar el comportamiento de  $v_1$ . Dicho análisis se llevó a cabo mediante simulación. En ésta fueron introducidas las ecuaciones para calcular las posiciones y velocidades en los mecanismos de 4 y 5 barras y así obtener la velocidad de desplazamiento  $v_1$ . Las medidas de los eslabones fueron las mismas que se utilizaron para la simulación del movimiento de los mecanismos (figura 2.14).

Conviene recordar que  $v_1$  depende de manera última de  $\theta_1$  y  $\dot{\theta}_1$  únicamente (las demás variables que intervienen en (2.25) pueden expresarse en términos de esas dos variables). Es por ello que únicamente interesa observar el comportamiento de  $v_1$  con respecto a  $\theta_1$  y  $\dot{\theta}_1$ , para luego proponer una función que tenga el mismo comportamiento que  $v_1$ .

El valor para la velocidad angular  $\dot{\theta}_1$  se introduce como una constante para después ser integrada continuamente y así generar los valores de  $\theta_1$  ( $0^\circ \leq \theta_1 < 360^\circ$ ). Los cálculos se realizaron en Simulink y los valores de  $v_1$  y  $\theta_1$  se graficaron con ayuda de Matlab.

A continuación se presentan algunos de los resultados obtenidos de la simulación. En la figura 2.16 se observan las gráficas que muestran el comportamiento de  $v_1$  (eje de las ordenadas) respecto a diferentes valores de  $\dot{\theta}_1$ .

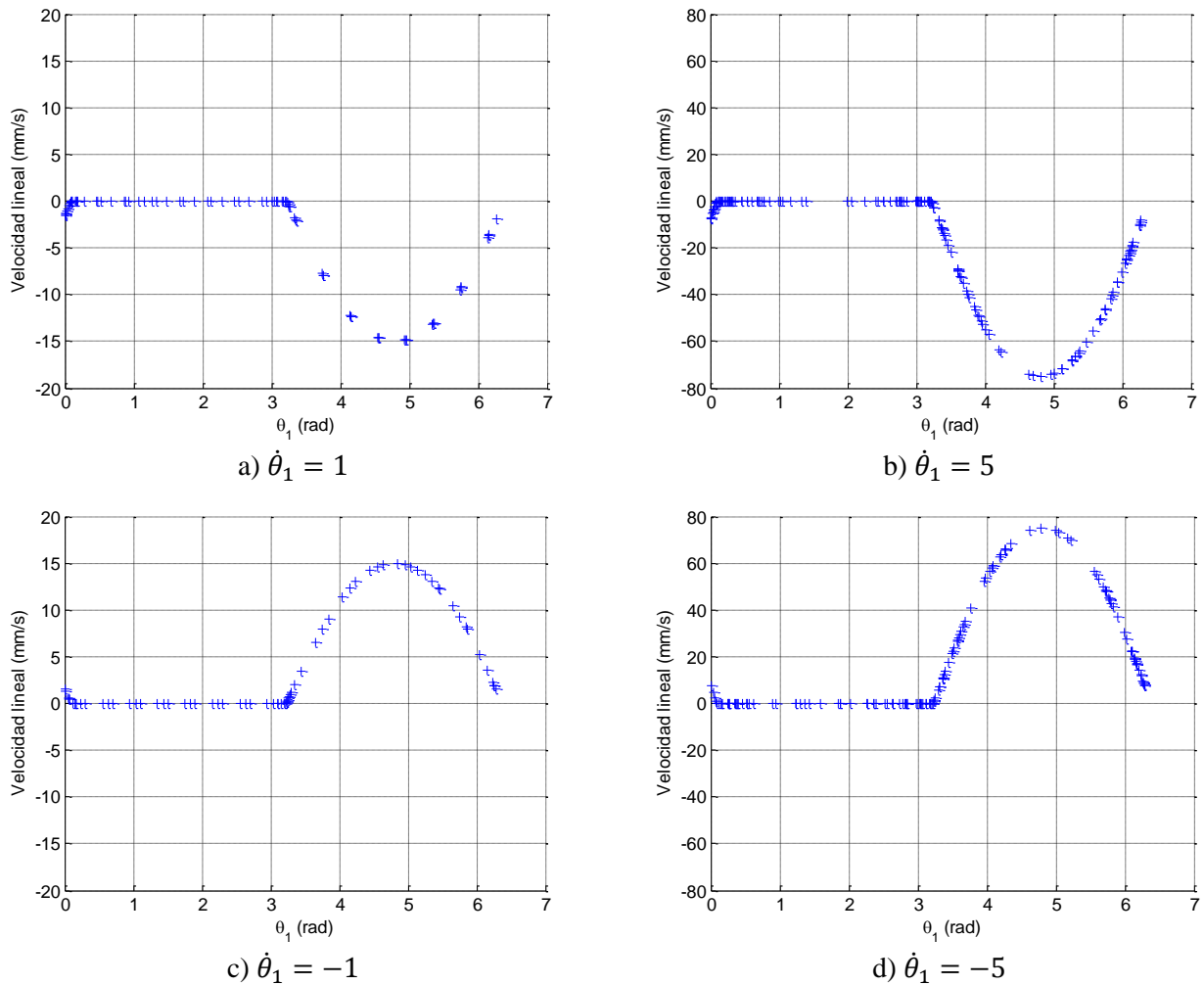


Figura 2.16. Gráficas del comportamiento de  $v_1$

Nótese que para valores positivos de  $\dot{\theta}_1$  los valores de velocidad son negativos, esto solo significa que cuando  $\dot{\theta}_1$  gira en sentido anti horario el cuerpo del robot se desplazará hacia atrás (o de reversa), y cuando tiene un sentido de giro horario los desplazamientos serán hacia adelante.

Cabe mencionar que se realizó un mayor número de pruebas con diferentes valores de  $\dot{\theta}_1$ , sin embargo todas presentaban características muy similares a las que se presentan en la gráficas anteriores.

Con ayuda de las gráficas obtenidas fue posible observar el comportamiento de  $v_1$  respecto a diferentes valores de  $\theta_1$  y  $\dot{\theta}_1$ . Tal comportamiento presenta en todos los casos las siguientes características:

- $v_1$  se asemeja a una función senoidal, o más precisamente, a la semionda senoidal positiva con un cierto desfaseamiento, excepto en la región donde  $\theta_1$  toma valores entre  $\rho_1$  y  $\rho_2$  (los límites que indican cuando las patas de avance no tienen contacto con la superficie, o sea  $v_1 = 0$ ).
- El desfaseamiento de la semionda siempre es el mismo.
- La frecuencia es la misma independientemente del valor de  $\dot{\theta}_1$ .
- $\dot{\theta}_1$  solo se comporta como un factor multiplicativo de  $v_1$ .
- El signo de  $\dot{\theta}_1$  no cambia ninguna de las propiedades anteriores.

Basado en lo anterior, se propone una función con el modelo:

$$\bar{v} = \dot{\theta}_1 K_1 \sin(\theta_1 - K_2) \quad (2.31)$$

donde  $K_1$  es la constante encargada de escalar la amplitud y  $K_2$  se ocupa de mantener el desfase constante. Para encontrar el valor de estas constantes que logre una aproximación aceptable de  $\bar{v}$  a  $v_1$  se recurrió al método de *corrección diferencial de mínimos cuadrados* (González E.J., 2010). Este método consiste en la minimización del residuo  $\Delta X$  en la expresión:

$$\Delta X = (A^T W A)^{-1} A^T W \Delta Y \quad (2.32)$$

donde  $A$  es la matriz de derivadas parciales de  $m \times n$  ( $m$  es el número de muestras y  $n$  el número de parámetros a estimar en la función).  $W$  es la matriz de pesos asignada a cada muestra (para este caso todos los pesos se consideraron de valor unitario).  $\Delta Y$  es el vector resultante de restar el valor original de las muestras menos los valores actuales en cada iteración.

Para  $\bar{v}$  se realizó un programa con 20 iteraciones del método de mínimos cuadrados y un total de 32524 muestras de  $v_1$ . El programa realizado para la convergencia de los valores de  $K_1$  y  $K_2$  se encuentra en el Anexo B de este documento. En dicho programa se logró la convergencia de los valores para  $K_1$  y  $K_2$ , dando como resultado:

$$K_1 = 15.4589 \quad \text{y} \quad K_2 = 4.7498^\circ \quad (2.33) \text{ y } (2.34)$$

Con estos valores, la función propuesta queda de la siguiente manera:

$$\bar{v} = \dot{\theta}_1 h_1(\theta_1) \quad (2.35)$$

con

$$h_1(\theta_1) = \begin{cases} 15.4589 \sin(\theta_1 - 4.7498^\circ), & 0^\circ \leq \theta_1 \leq \rho_1 \cup \rho_2 \leq \theta_1 \leq 360^\circ \\ 0, & \rho_1 < \theta_1 < \rho_2 \end{cases} \quad (2.36)$$

Para verificar que la nueva función propuesta  $\bar{v}$  era capaz de reemplazar a  $v_1$  en el modelo cinemático, se realizaron numerosas pruebas de las cuales se muestran algunos resultados a continuación, figura 2.17. La curva con “+” (color azul) representa el comportamiento de  $v_1$  y la curva con “o” (en rojo) describe el comportamiento de la nueva función propuesta  $\bar{v}$ :

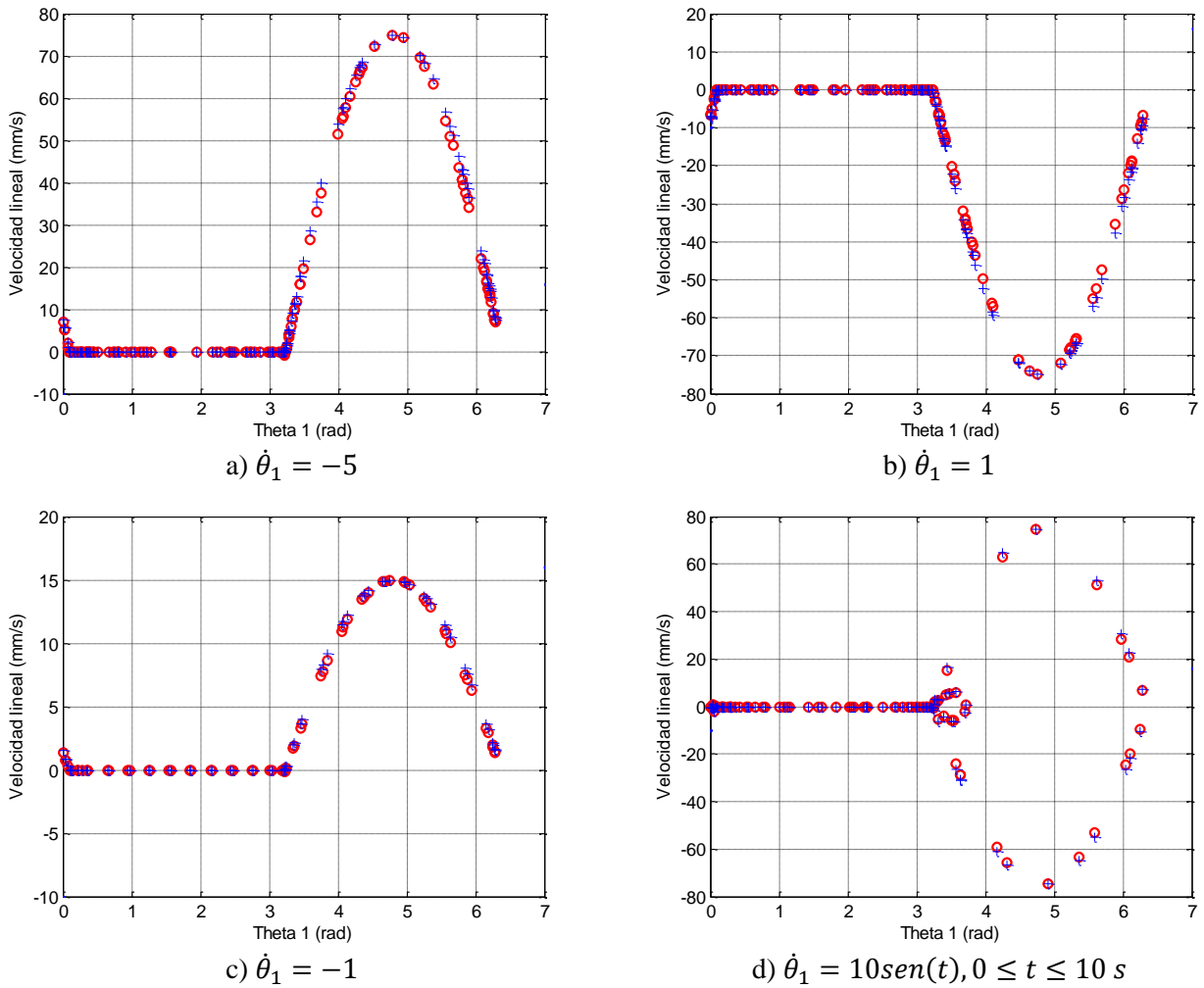


Figura 2.17.  $v_1$  vs.  $\bar{v}$

Además de tener un comportamiento muy similar a  $v_1$ , la nueva función  $\bar{v}$  presenta otras características que merecen ser mencionadas:

- Es una función claramente más sencilla y fácil de manejar a comparación de  $v_1$ .
- Las variables de las cuales depende ( $\theta_1$  y  $\dot{\theta}_1$ ) están explícitas en la función. A diferencia de  $v_1$  en donde dichas variables se encuentran implícitas dentro de otros términos.
- $\bar{v}$  se aproxima bastante bien a  $v_1$ , por lo tanto  $\bar{v}$  también se aproxima a la velocidad de avance lineal que posee el robot.

Por los puntos expuestos anteriormente, se adopta a  $\bar{v}$  como la función que describe el avance del robot.

Ahora, recordando que la velocidad de giro del robot está dada por (2.29), de manera que es posible reescribirla como

$$\dot{\beta}_r = \omega h_2(\theta_1) \quad (2.37)$$

con

$$h_2(\theta_1) = \begin{cases} 0, & 0^\circ \leq \theta_1 \leq \rho_1 \cup \rho_2 \leq \theta_1 \leq 360^\circ \\ -1, & \rho_1 < \theta_1 < \rho_2 \end{cases} \quad (2.38)$$

De manera que el modelo cinemático, ahora simplificado, quedará de la siguiente manera:

$$\begin{aligned} \dot{x}_r &= \dot{\theta}_1 h_1(\theta_1) \cos \beta_r \\ \dot{y}_r &= \dot{\theta}_1 h_1(\theta_1) \sin \beta_r \\ \dot{\beta}_r &= \omega h_2(\theta_1) \end{aligned} \quad (2.39)$$

A simple vista no parece ser muy diferente del antiguo modelo obtenido, sin embargo hay que tener en cuenta todos los cálculos implícitos para obtener la velocidad de desplazamiento en (2.25). En cambio, para obtener la solución de  $\bar{v}$  en (2.35) simplemente hay que calcular la función seno y multiplicarla por una constante.

## ***Discusión***

El haber obtenido una nueva función de velocidad, no implica que el análisis de los mecanismos de 4 y 5 barras haya sido un cálculo innecesario. En este caso fueron utilizados como referencia para proponer una función más sencilla y que de igual manera describiera lo mejor posible la velocidad de desplazamiento del robot.

Cabe mencionar que, en el modelo cinemático obtenido no se consideran algunos efectos que influyen en el desplazamiento del robot, por ejemplo el derrape entre la superficie y las patas del mismo. El error que provocan dichos efectos puede ser minimizado mediante técnicas de control.



Los resultados obtenidos del modelo cinemático del robot fueron publicados en las memorias del XIII Congreso Mexicano de Robótica, con el trabajo titulado *Modelado Cinemático de un Robot no-Holónimo de 2 Grados de Libertad con Patas* (Alonso, 2011).



# Capítulo 3. Generación de Trayectorias

En este capítulo se obtienen las trayectorias que se el robot es capaz de seguir. Estas trayectorias se determinan en base a los movimientos que el robot es o no capaz de realizar, es decir, en base a su modelo cinemático.

Utilizando el modelo cinemático que se obtuvo en la sección anterior, se realizó una simulación de movimiento del robot en Simulink. En la simulación fueron introducidas las ecuaciones de velocidad lineal y de giro, para posteriormente ser integradas y obtener la posición del móvil respecto al marco de referencia fijo  $(x_r, y_r)$ . Cabe mencionar que fueron utilizadas las ecuaciones de velocidad debido a que el paso de integración que maneja Simulink es variable y se ajusta de manera automática para lograr mayor precisión en el cálculo de la integral.

En la figura 3.1 se muestra el resultado de la simulación de movimiento del robot sobre un plano. Para la velocidad de giro  $\omega$  en la pata de apoyo se introdujo un valor constante y negativo, esto significa que el cuerpo girará en sentido anti horario cuando las condiciones para  $\theta_1$  permitan el giro. Para la velocidad de avance  $\bar{v}$  se introdujo un valor constante y negativo para  $\dot{\theta}_1$ . Hay que recordar que  $\omega$  y  $\dot{\theta}_1$  son las únicas variables que controlan el movimiento del robot. En la misma figura, también se aprecia que el robot se desplaza dando pasos de magnitud constante. Cada uno de los pasos ocurre cuando  $\theta_1$  (ángulo del eslabón actuador) realiza un ciclo de vuelta completa.

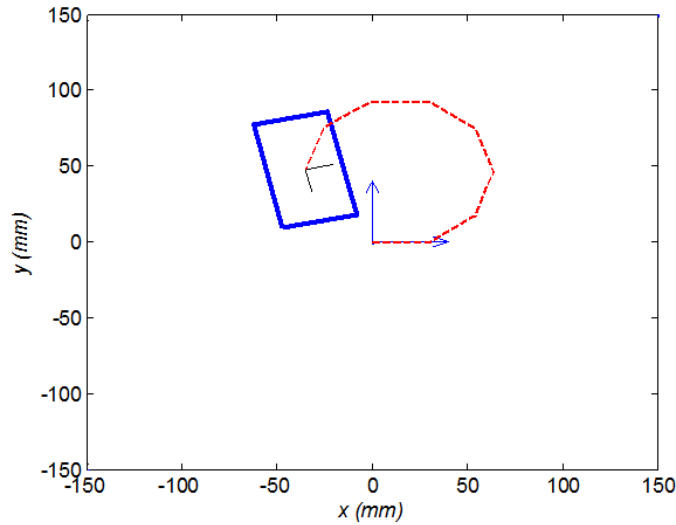


Figura 3.1. Simulación de desplazamiento. El rectángulo representa el cuerpo del robot y la línea punteada muestra la trayectoria que describe.

### ***Trayectorias primitivas del robot***

Con la simulación anterior, se observa que el robot sólo es capaz de seguir trayectorias compuestas por líneas rectas de magnitud constante y giros con magnitud y dirección arbitrarias. Esto genera una serie de trayectorias *primitivas* (Marín, et al., 2011), estas son trayectorias básicas del robot. Con ellas es posible construir una trayectoria compuesta que alcance cualquier punto sobre la superficie.

Las trayectorias primitivas para el robot se muestran la figura 3.2. Las líneas rectas representan la magnitud de un paso constante de magnitud  $R$ . Los puntos en cada inicio y final de las líneas rectas, representan los puntos donde el robot es capaz de girar y comenzar una nueva trayectoria primitiva. El ángulo  $\sigma$  representa el giro que el robot puede realizar para orientarse y avanzar un paso. Este giro no está restringido en magnitud (a diferencia del avance lineal del robot) y puede ser en cualquier sentido.

### ***Magnitud del paso del robot***

Pese a que es posible parametrizar el valor de  $R$  (magnitud del paso en el avance del robot) y usarlo como una variable constante en futuros trabajos, es importante saber cómo calcular su valor. Para conocer la magnitud del paso, se hace uso del modelo cinemático obtenido en el capítulo anterior (2.39). En este modelo se encuentra la función de velocidad de avance que está dada por una ecuación diferencial (2.32). Por lo tanto, la manera de encontrar el valor de  $R$  es integrando dicha ecuación.

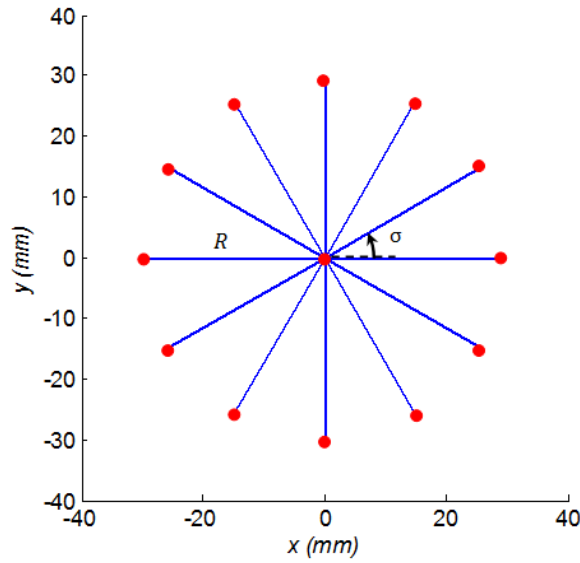


Figura 3.2. Trayectorias primitivas en el robot

Hay que recordar que  $\bar{v}$  se compone de dos funciones diferentes, resultado de la condición para  $\theta_1$ . Lo anterior es importante debido a que la velocidad de avance se calcula mediante una función senoidal, y la integral de 0 a  $360^\circ$  de dicha función es igual a cero.

Recordando los valores de las constantes involucradas ( $K_1 = 15.4589, K_2 = 4.7498, \rho_1 = 5.2445^\circ$  y  $\rho_2 = 184.63^\circ$ ) la integral de (2.32) se calcula de la siguiente manera:

$$|R| = \left| \int_0^{\rho_1} K_1 \sin(\theta_1 - K_2) d\dot{\theta}_1 + \int_{\rho_1}^{\rho_2} 0 d\dot{\theta}_1 + \int_{\rho_2}^{360^\circ} K_1 \sin(\theta_1 - K_2) d\dot{\theta} \right| = |-30.9178| \approx 31 \quad (3.1)$$

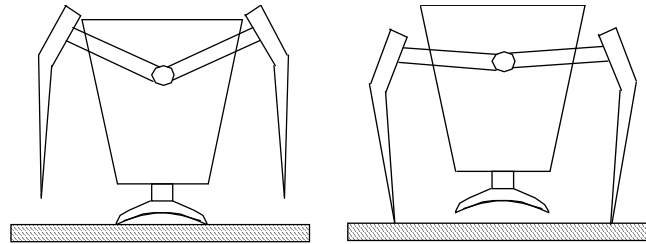
El signo “-” en el resultado de la integral se refiere a que  $\theta_1$  crece positivamente (como se aprecia en los límites de la integral). El cuerpo del robot se desplazará aproximadamente 31 unidades hacia atrás. Lo cual es de poco interés, ya que con este cálculo sólo se busca la magnitud del paso sin importar el sentido. Por lo tanto la magnitud del paso del robot queda definida como  $R = 31$  unidades (en el caso de este prototipo las unidades son milímetros).

## 3.1 Planeación de Trayectorias

Anteriormente se determinaron las trayectorias primitivas del robot. Ahora, se requiere proponer trayectorias que el robot sea capaz de seguir, y que lo lleven de un punto a otro en un *espacio de trabajo* libre de obstáculos (por espacio de trabajo se entiende a la superficie sobre la cual se desplaza el robot).

A diferencia de los robots móviles con ruedas, el robot de este trabajo no puede realizar trayectorias curvilíneas; se desplazada sobre la superficie con una serie de pasos constantes. A pesar de la restricción que el robot presenta para seguir cierto tipo de trayectorias, es capaz de llegar de una posición inicial a cualquier otra mediante una serie de maniobras.

Otro aspecto importante que considerar para la planeación de trayectorias, es el tipo de *configuración interna* en la que se encuentra el robot en cada instante. En este trabajo se le llama *configuración interna* a cualquiera de las dos posturas en las puede estar el robot; pata de giro apoyada (configuración A) o pata de giro no apoyada (configuración B), figura 3.3. Analizando las características de ambas configuraciones, se llegó a la conclusión que la configuración A es la que el robot debe adoptar en los puntos inicial y final de la trayectoria. Debido a que en dicha configuración el robot está listo para orientarse hacia cualquier lugar, y también puede desplazarse hacia adelante o hacia atrás de manera inmediata. En la configuración B sólo es capaz de avanzar linealmente.



a) Configuración interna A    b) Configuración interna B

Figura 3.3. Configuraciones internas del robot

La planeación de trayectorias debe contar con un criterio de optimización, el cual se determina en base a las necesidades propias del sistema. Haciendo mención de algunos criterios de optimización, se tienen los criterios de *minimización de energía*, *minimización de tiempo* o *minimización de distancia*. Estos criterios se refieren a planear una trayectoria que, además de ser realizable por el robot, consuma la menor cantidad de energía posible para su ejecución (minimización de energía); demande el menor tiempo posible para llegar de un punto a otro (minimización de tiempo); posea la menor longitud que otras posibles trayectorias (minimización de distancia).

A continuación se describen algunos conceptos que son necesarios para la planeación de trayectorias en este trabajo.

### ***Variables en el problema***

Las variables que intervienen en el problema de planeación de trayectorias son aquellas que describen la posición del robot en el plano, además de aquellas que describen los movimientos que debe realizar el robot a lo largo de una trayectoria.

Las posiciones inicial y final están medidas respecto a un sistema de referencia inercial  $F_o(O_o, \mathbf{i}_o, \mathbf{j}_o)$  pegado al piso, figura 3.4. Las variables de posición inicial  $(x_i, y_i)$  y final  $(x_f, y_f)$  son conocidas, al igual que las variables de orientación inicial  $(\beta_i)$  y final  $(\beta_f)$ . En la misma figura se muestran las variables que describen los movimientos que debe realizar el robot a lo largo de una trayectoria:  $R$  que es la magnitud del paso constante del robot;  $D$  es la distancia entre las posiciones inicial y final;  $\sigma_i$  ( $i = 1, 2, \dots, n, n + 1$ ) se refiere a cada uno de giros que el robot debe realizar a lo largo de la trayectoria;  $\zeta_j$  ( $j = 1, 2, \dots, n$ ) son ángulos medidos de la horizontal hacia la línea recta que traza el robot en cada paso realizado.

### ***Cálculo de la distancia a recorrer***

La distancia  $Z$  que debe recorrer el robot a lo largo de una trayectoria se calcula mediante (3.2). Donde  $n$  es el número de pasos enteros a lo largo de la trayectoria y  $R$  es la magnitud constante de un paso del robot.

$$Z = nR \quad (3.2)$$

### ***Restricciones***

Debido a que las trayectorias deben estar compuestas por *pasos enteros* del robot, la variable  $n$  debe tener un valor entero positivo. De esta manera, se tiene que una de las restricciones para la planeación de trayectorias está dada por (3.3).

$$(n \in \mathbb{N}) \quad (3.3)$$

Como ya se mencionó anteriormente, existen diferentes criterios para desarrollar la planeación de trayectorias. A continuación se presentan tres de los posibles criterios para realizar la planeación de trayectorias en este trabajo.

## **3.1.1 Minimización de Energía**

Para realizar la planeación de trayectorias bajo el criterio de mínima energía, es necesario considerar el torque requerido para realizar los dos movimientos característicos del robot; *traslación* y *rotación*. El movimiento de traslación se refiere al avance lineal que realiza el robot a lo largo de una trayectoria. El movimiento de rotación es el que le permite modificar su orientación cuando le es permitido (condiciones en el modelo cinemático).

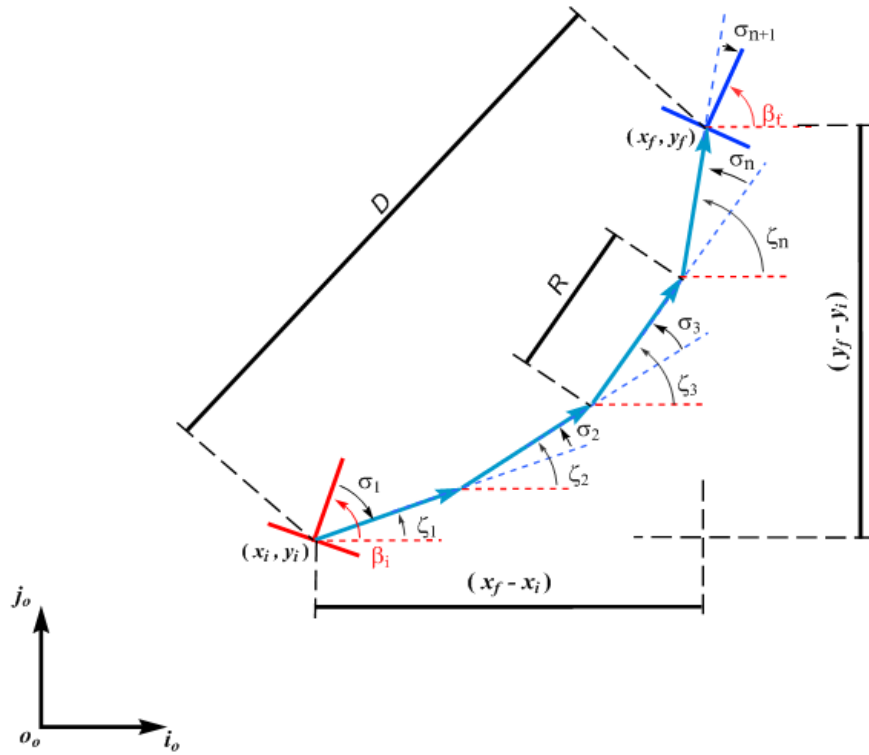


Figura 3.4. Variables implícitas en la planeación de trayectorias

Considerando que para dar un paso (movimiento de traslación) el robot necesita un torque constante  $\tau_t$ , la energía requerida a lo largo de una trayectoria por el movimiento de traslación está dada por

$$E_t = 360^\circ n \tau_t \quad (3.4)$$

donde  $n$  es el número entero de pasos en la trayectoria.

De igual manera, es posible considerar un torque constante  $\tau_r$  para realizar el movimiento de rotación en el robot. Bajo esta consideración, es posible calcular la energía requerida para realizar todos los giros implicados en la trayectoria ( $E_r$ ), teniendo como resultado que

$$E_r = \tau_r \sum_{i=1}^{n+1} |\sigma_i| \quad (3.5)$$

donde  $\sigma_i$  representa cada uno de los giros realizados por el robot a lo largo de la trayectoria. Para el cálculo de  $E_r$  sólo importa la magnitud de cada giro, es por ello que en (3.5) se calcula el valor absoluto de los mismos.

Con los resultados obtenidos en (3.4) y (3.5) es posible calcular la energía total  $E_T$  que se requiere para realizar cualquier trayectoria. De manera que



$$E_T = E_t + E_r \quad (3.6)$$

Sustituyendo (3.4) y (3.5) en (3.6) se tiene

$$E_T = 360^\circ n\tau_t + \tau_r \sum_{i=1}^{n+1} |\sigma_i| \quad (3.7)$$

Los valores de  $\sigma_i$  dependen de los valores que tomen los ángulos absolutos  $\zeta_j$ . Observando la figura 3.4, es posible deducir que el valor de cada  $\sigma_i$  se calcula de la forma siguiente:

$$\begin{aligned} \sigma_1 &= \zeta_1 - \beta_i \\ \sigma_2 &= \zeta_2 - \zeta_1 \\ \sigma_3 &= \zeta_3 - \zeta_2 \\ &\vdots \\ \sigma_n &= \zeta_n - \zeta_{n-1} \\ \sigma_{n+1} &= \beta_f - \zeta_n \end{aligned} \quad (3.8)$$

Por lo tanto, la función de objetivo en el problema de mínima energía se obtiene sustituyendo (3.8) en (3.7), dando como resultado la ecuación (3.9).

$$E_T = 360^\circ n\tau_t + \tau_r \left( |\zeta_1 - \beta_i| + |\zeta_2 - \zeta_1| + \dots + |\zeta_n - \zeta_{n-1}| + |\beta_f - \zeta_n| \right) \quad (3.9)$$

Dentro de este problema de minimización se encuentran algunas restricciones. Estas restricciones aseguran que el robot llegue a la posición  $(x_f, y_f)$  y orientación  $(\beta_f)$  deseadas. En el caso del número de pasos  $n$ , se tiene como primera restricción la enunciada en (3.3), la cual se refiere a que  $n$  sólo puede tomar valores enteros positivos. Como segunda restricción se tiene que el número de paso debe ajustar para alcanzar la posición final deseada, es decir, la distancia que debe recorrer el robot (3.2) debe ser al menos de igual magnitud que  $D$ . Por lo tanto se tiene que

$$n \geq \frac{D}{R} \quad (3.10)$$

Para el caso de los giros en la trayectoria, se debe asegurar que el robot llegue a la posición deseada modificando  $n + 1$  veces su orientación a lo largo de la trayectoria. Lo anterior da origen a dos restricciones más al problema de minimización. Estas restricciones se enuncian en (3.11) y (3.12), siendo (3.11) la restricción que indica que la trayectoria debe alcanzar la coordenada deseada sobre el eje horizontal  $(x_f)$  y (3.12) hace referencia a la coordenada en el eje vertical  $(y_f)$ . Ver figura 3.4.

$$R(\cos \zeta_1 + \cos \zeta_2 + \dots + \cos \zeta_n) - (x_f - x_i) = 0 \quad (3.11)$$

$$R(\sin \zeta_1 + \sin \zeta_2 + \dots + \sin \zeta_n) - (y_f - y_i) = 0 \quad (3.12)$$

Con la función objetivo (3.9) y sus restricciones (3.2) y (3.10-3.12), se tiene que el problema de minimización tiene la forma (3.13).

$$\begin{aligned} \min(E_T) \quad E_T &= 360^\circ n \tau_i + \tau_r \left( |\zeta_1 - \beta_i| + |\zeta_2 - \zeta_1| + \dots + |\zeta_n - \zeta_{n-1}| + |\beta_f - \zeta_n| \right) \\ \text{s.a} \quad n &\geq \frac{D}{R} \\ n &\in \mathbb{N} \\ R(\cos \zeta_1 + \cos \zeta_2 + \dots + \cos \zeta_n) - (x_f - x_i) &= 0 \\ R(\sin \zeta_1 + \sin \zeta_2 + \dots + \sin \zeta_n) - (y_f - y_i) &= 0 \end{aligned} \quad (3.13)$$

Para obtener la minimización de (3.13) debe recurrirse a métodos de programación entera, debido a que la variable  $n$  sólo puede tomar valores de este tipo.

De esta manera se tiene el planteamiento para realizar la planeación de trayectorias con en base al criterio de mínima energía. Cabe mencionar que la energía calculada en el proceso, es una aproximación de la verdadera energía necesaria para realizar una trayectoria. Esto se debe a que el torque en cada uno de los motores está considerado como constante, cuando en la realidad ambos están sujetos a cambios debido a la relación variante entre la masa y el desplazamiento del robot. Para una mejor aproximación de la energía es posible que se requiera el modelo dinámico del robot.

Es importante recordar que el robot de este trabajo posee una arquitectura simétrica, la cual le permite avanzar con la misma velocidad de frente o de reversa. En el análisis anterior se considera que el robot debe avanzar siempre de frente y se descarta un posible cambio de sentido de avance. Si se quisiera considerar el cambio de sentido de avance, sería necesario analizar una mayor cantidad de funciones. Lo anterior se debe a que el cálculo para cada uno de los giros es diferente cuando se desea que el robot avance en sentido contrario. En apartados posteriores se describe con mayor precisión el concepto anterior.

### 3.1.2 Minimización de Tiempo

Para obtener una función que calcule el tiempo total que tarda el robot en llegar de un punto a otro, se sigue una metodología muy similar a la usada para obtener la función de energía en el apartado anterior.

Considerando una velocidad constante  $\omega_t$  en el motor encargado del desplazamiento lineal del robot, es posible decir que el tiempo que tarda el robot en dar un paso es  $360^\circ / |\omega_t|$ . Por lo tanto, el

tiempo que tarda en desplazarse de manera lineal ( $T_t$ ) en una trayectoria compuesta de  $n$  pasos enteros se calcula mediante (3.14).

$$T_t = \frac{360^\circ n}{|\omega_t|} \quad (3.14)$$

En (3.14) se calcula la magnitud de  $\omega_t$  debido el motor encargado de la traslación puede girar en ambos sentidos, y un valor negativo para  $\omega_t$  ocasiona una estimación errónea del tiempo (el tiempo debe tener magnitud positiva).

Para calcular el tiempo invertido en modificar su orientación  $n + 1$  veces a lo largo de una trayectoria ( $T_r$ ) se tiene la ecuación (3.15).

$$T_r = \frac{1}{|\omega_r|} \sum_{i=1}^{n+1} |\sigma_i| \quad (3.15)$$

De manera que el tiempo total ( $T_T$ ) que tarda el robot en llegar de un punto a otro se determina sumando los tiempos anteriormente calculados:

$$T_T = T_t + T_r \quad (3.16)$$

Sustituyendo (3.14) y (3.15) en (3.16) se tiene

$$T_T = \frac{360^\circ n}{|\omega_t|} + \frac{1}{|\omega_r|} \sum_{i=1}^{n+1} |\sigma_i| \quad (3.17)$$

Al igual que en la función para el cálculo total de energía (3.7), es posible sustituir (3.8) en la función anterior. Teniendo como resultado que la función para determinar el tiempo consumido a lo largo de la trayectoria es:

$$T_T = \frac{360^\circ n}{|\omega_t|} + \frac{1}{|\omega_r|} \left( |\zeta_1 - \beta_i| + |\zeta_2 - \zeta_1| + \dots + |\zeta_n - \zeta_{n-1}| + |\beta_f - \zeta_n| \right) \quad (3.18)$$

Las restricciones presentes en la función anterior, son las mismas que se encontraron para el caso de la energía (ecuaciones (3.2), (3.10), (3.11) y (3.12)). Por lo tanto, el problema de optimización para el caso de tiempo queda expresado en (3.19).

$$\begin{aligned}
\min(T_T) \quad T_T &= \frac{360^\circ n}{|\omega_t|} + \frac{1}{|\omega_r|} \left( |\zeta_1 - \beta_i| + |\zeta_2 - \zeta_1| + \dots + |\zeta_n - \zeta_{n-1}| + |\beta_f - \zeta_n| \right) \\
s.a \quad n &\geq \frac{D}{R} \\
n &\in \mathbb{N} \\
R(\cos \zeta_1 + \cos \zeta_2 + \dots + \cos \zeta_n) - (x_f - x_i) &= 0 \\
R(\sin \zeta_1 + \sin \zeta_2 + \dots + \sin \zeta_n) - (y_f - y_i) &= 0
\end{aligned} \tag{3.19}$$

Al igual que en la minimización de la energía en (3.13), debe recurrirse a métodos de programación entera para obtener la trayectoria que demande menor tiempo en llevar al robot de un punto a otro.

El tiempo calculado en (3.19) es una aproximación del tiempo real necesario para realizar una trayectoria. Esto se debe a que la velocidad en cada uno de los motores se considera constante, cuando en la realidad ambas velocidades están sujetas a cambios debido al arranque y paro de los motores (aceleración y desaceleración en los motores).

En este caso también se considera un sólo sentido de avance para el robot.

### 3.1.3 Minimización de Distancia

La planeación de trayectorias en este trabajo, se realizó bajo el criterio de mínima distancia debido a diferentes motivos:

- La tarea de aplicación del robot suele estar plasmada en desplazamientos, por ejemplo exploración de un entorno desconocido, evitamiento de obstáculos, etc.
- Es una minimización sencilla, comparada por ejemplo con la minimización de una función en energía o tiempo. En este caso la minimización se realizó en forma desacoplada, es decir, en primera instancia se calcula la mínima distancia para llegar de un punto a otro y posteriormente se minimiza la magnitud total de giro en la trayectoria. Lo anterior es posible debido a que no existe ninguna relación entre las unidades de desplazamiento y orientación propias del robot.
- Relacionado con lo anterior, para las funciones de costo en tiempo y en energía hay que tomar consideraciones (simplificaciones o hipótesis), por ejemplo un perfil de velocidad determinado para la minimización por tiempo o un perfil de torque para la minimización de energía las cuales no son condiciones de operación reales del robot.
- Para una buena clase de sistemas mecánicos, las tres minimizaciones suelen dar resultados similares.

Para cumplir con el criterio de mínima distancia, se recurrió a métodos gráficos para la planeación de trayectorias (Laumond, 1998), (Marín, et al., 2011), (Pivtoraiko, et al., 2009), y métodos de programación entera pura para el cálculo de la distancia mínima (Tovar & Renaud, 2010), (Arora, 2004).

Dando inicio a la formulación del problema de minimización, se tiene que la distancia que recorre el robot se calcula con el producto del número de pasos en la trayectoria y la magnitud de cada uno de ellos (3.2).

Debido a que el problema de minimización será tratado en forma desacoplada (distancia y giros por separado), las únicas restricciones de interés son (3.3) y (3.10). Estas son las únicas restricciones asociadas con el desplazamiento lineal del robot. De manera que el problema de minimización de distancia queda de la forma (3.20), donde  $Z$  es la función objetivo a minimizar.

$$\begin{aligned} \min(Z) \quad Z &= nR \\ \text{s.a} \quad n &\geq \frac{D}{R} \\ n &\in \mathbb{N} \end{aligned} \tag{3.20}$$

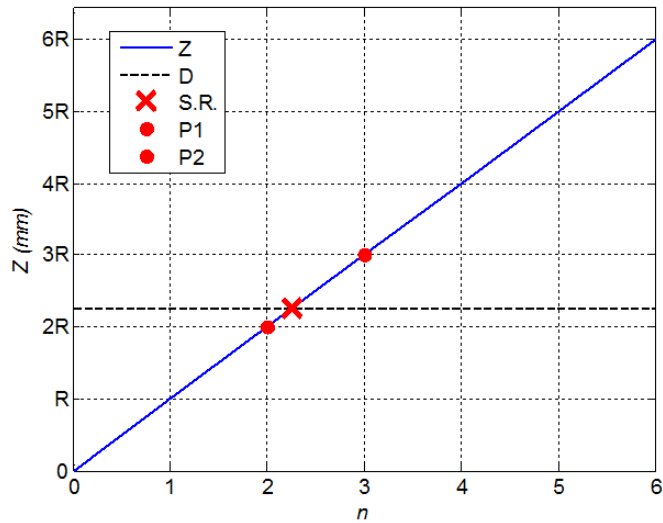
La variable  $n$ , de la cual depende la minimización de (20), es una variable que sólo puede tomar valores enteros. Por lo tanto, el problema de minimización puede resolverse por métodos gráficos considerándose como un problema de Programación Entera (Arora, 2004).

El método gráfico consiste en trazar las restricciones de la función objetivo y analizar aquellos puntos que son candidatos para cumplir con la minimización.

Inicialmente se obtiene la *solución relajada* del problema, es decir, se realiza la minimización del problema ignorando la restricción que indica que la variable  $n$  pertenece al conjunto de los números naturales ( $n = 0,1,2,3,4, \dots$ ). De manera que la solución relajada para (3.20), se obtiene igualando la función objetivo a la mínima cantidad posible ( $Z = D$ ). Por lo tanto el valor para  $n$  de la solución relajada esta dado por

$$n = \frac{D}{R} \tag{3.21}$$

Si el valor de  $n$  resulta ser un número entero, la solución converge y el valor mínimo para  $n$  se calcula mediante (3.21). En el caso contrario se recurre al método gráfico, en el cual es posible observar los puntos vecinos a al valor de la solución relajada, gráfica 3.1.



Gráfica 3.1. Método gráfico para la optimización de la distancia;  $n$  se refiere al número entero de pasos (adimensional) y  $Z$  es la distancia a recorrer por el robot ( $mm$ ).

En la gráfica anterior se muestra un ejemplo donde  $n$  de la solución relajada (S.R., marcada con una  $\times$ ) no toma un valor entero, y se encuentra en la intersección de la constante  $D$  (línea punteada) y la función objetivo  $Z$ . A la izquierda y derecha de  $n$  se encuentra dos puntos vecinos (puntos gruesos en la gráfica 3.1). Cada uno de los puntos vecinos genera un valor diferente para la función objetivo; de manera que es necesario evaluar cada uno de ellos y elegir aquel que genere el valor mínimo de  $Z$ . Sin embargo, hay que recordar que una de las restricciones en (3.20) indica que el valor de  $Z$  debe ser igual o mayor que  $D$ . Por lo que el punto vecino que cumple con la minimización del problema, será el valor entero más próximo y mayor a  $D/R$ . Es decir

$$n = \text{int}\left(\frac{D}{R}\right) + 1 \quad (3.22)$$

donde  $\text{int}(D/R)$  se refiere a conservar únicamente la parte entera que resulta de la división de  $D/R$ .

El procedimiento anterior muestra una manera general de cómo obtener un valor de  $n$  que minimice (3.20). Los resultados se analizaron de manera gráfica en el espacio de trabajo del robot, y se concluyó que era posible construir trayectorias compuestas por  $n$  pasos que llevase al robot de un punto a otro de la superficie. Sin embargo, existe un rango de valores de  $D$  para el cual no fue posible construir una trayectoria realizable con el resultado de  $n$  en (3.20).

### 3.1.3.1 Caso Particular

El rango de valores para  $D$  donde no fue posible aplicar el método descrito anteriormente es  $0 < D < R$ . Aplicando el método anterior se obtiene que el valor de  $n$  (usando (3.22) debido a que resulta no ser número entero) es igual a 1, cuando  $0 < D < R$ . Lo que sugiere que el robot puede llegar a una *posición cercana* con únicamente un paso, pero en la realidad esto no es posible. Se considera *posición cercana* a todo punto dentro de una circunferencia de radio  $R$  con centro en la posición inicial del robot, figura 3.5. Recordando las trayectorias primitivas del robot (figura 3.2), este sólo puede desplazarse mediante pasos enteros. De manera que una trayectoria compuesta por un solo paso, lo llevaría más allá de la posición final deseada (perímetro de la circunferencia en la figura 3.5).

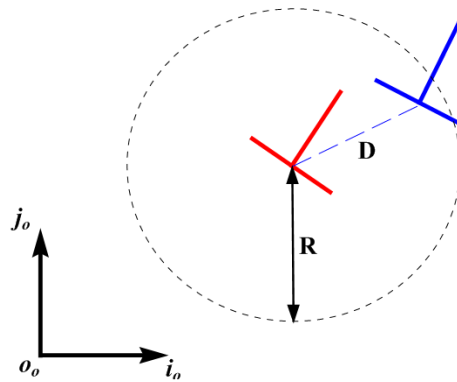


Figura 3.5. Posición cercana ( $0 < D < R$ )

El problema anterior lleva a proponer una trayectoria compuesta por el menor número de pasos posibles y que alcance la posición final deseada. De manera que se propone una trayectoria donde  $n$  sea igual a 2; esto debido a que es el número entero más próximo 1 y con el cual  $Z$  resulta ser mayor a  $D$  (restricción en (3.20)). En la figura 3.6 se muestra que con únicamente dos pasos ( $n = 2$ ) es posible alcanzar una posición que se encuentra a una *corta distancia* ( $0 < D < R$ ).

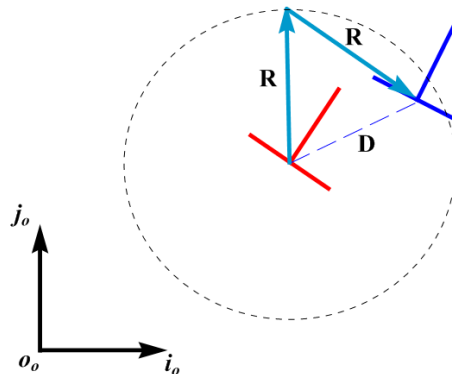


Figura 3.6. Trayectoria realizable para corta distancia

De lo anterior se puede decir que, para el caso cuando  $D$  posea un valor mayor que 0 y menor que  $R$ , la trayectoria realizable con menor magnitud está dada por  $Z = nR$ , donde  $n = 2$ .

Haciendo una pequeña reflexión de lo que se ha visto en este apartado, es posible identificar tres casos donde el valor de  $n$  (mínimo número de pasos) se calcula de diferente forma:

- Caso 1. *Distancia entera*. Ocurre cuando  $D$  es un múltiplo entero de  $R$ .  $n$  se calcula mediante la relación  $\frac{D}{R}$ .
- Caso 2. *Larga distancia*.  $D > R$  y no múltiplo de  $R$ . Entonces  $n = \text{int}\left(\frac{D}{R}\right) + 1$ .
- Caso 3. *Corta distancia*.  $0 < D < R$ . En este caso, la menor trayectoria realizable consta de dos pasos,  $n = 2$ .

De manera que la minimización de la función objetivo  $Z$ , puede expresarse de la siguiente forma:

$$Z = \begin{cases} D \Leftrightarrow D/R \in \mathbb{N} \\ R[\text{int}(D/R) + 1] \Leftrightarrow \{D \mid D > R\} - \{D \mid Z = D\} \\ 2R \Leftrightarrow 0 < D < R \end{cases} \quad (3.23)$$

Ahora se conoce la distancia mínima que el robot es capaz de recorrer para llegar de un punto a otro de la superficie.

En la siguiente sección se presentan diferentes tipos de trayectorias para cada uno de los casos de  $D$ .

### 3.1.4 Propuestas de Trayectorias

En esta sección se presentan las trayectorias que el robot debe seguir para los diferentes casos que han resultado de la minimización de distancia. Inicialmente se presenta la situación que se considera más sencilla de resolver ( $D$  es múltiplo entero de  $R$ ), para después presentar los otros dos casos en orden de menor a mayor complejidad.

#### 3.1.4.1 Trayectoria para Distancia Entera

En este caso se tiene que la magnitud de la distancia  $D$ , que separa a las posiciones inicial y final del robot, resulta ser un múltiplo entero de la magnitud del paso  $R$ . Para una rápida identificación, puede llamarse también como caso de *distancia entera*. Para este caso, la mínima distancia  $Z$  que el robot es capaz de recorrer es  $D$  misma. Gráficamente, esto indica que el robot debe llegar de un punto en línea recta. En la figura 3.7 se muestra cómo sería la trayectoria para este caso; las flechas continuas representan los pasos necesarios para llegar de una posición a otra de la superficie;  $\beta_i$  y  $\beta_f$  representan la orientación inicial y final del robot, respectivamente; los ángulos  $g_{ai}$  ( $i=1,2$ ) son las opciones que el robot tiene para girar en el punto inicial para alinearse con la línea recta que une



ambas posiciones; los ángulos  $g_{bi}$  ( $i=1,2$ ) son giros que el robot puede realizar para finalmente alcanzar la orientación final deseada;  $\gamma$  es la orientación de la línea recta que uno los puntos  $a$  y  $b$  respecto al sistema de referencia inercial.

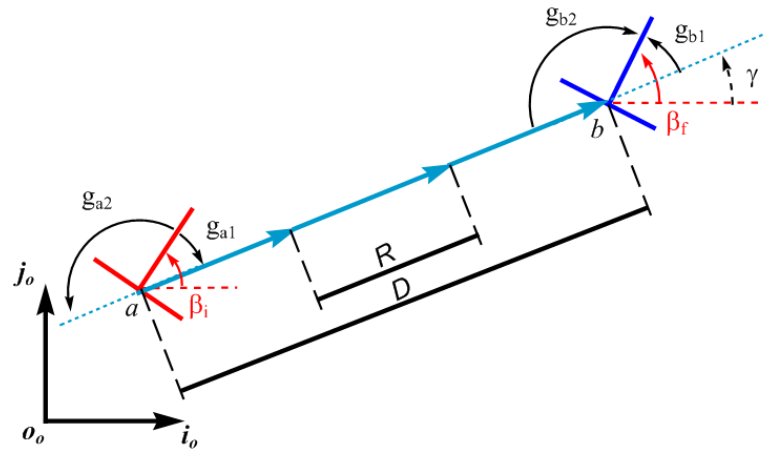


Figura 3.7. Trayectoria en línea recta

En este momento comienzan a tener relevancia los giros que el robot realizará a lo largo de la trayectoria. Claramente los ángulos  $g_{ai}$  no son iguales entre sí (salvo el caso particular donde ambos tengan un magnitud de  $90^\circ$ ), del mismo modo los ángulos  $g_{bi}$  poseen diferente magnitud y sentido entre sí. Es por ello que una vez que se encontró la distancia mínima, se decide minimizar la magnitud de giro total que realiza el robot a largo de la trayectoria.

### ***Minimización de la cantidad de giro***

Antes de dar comienzo a la minimización del total de giro, debe analizarse la existencia de diferentes combinaciones entre los giros en el punto  $a$  y los giros del punto  $b$ . El robot es capaz de seguir cualquiera de los ángulos  $g_{ai}$  inicialmente, sin embargo, al llegar al final de la trayectoria debe seguir el ángulo  $g_{bi}$  que logre alinearlos con la orientación final deseada  $\beta_f$ .

Se tiene que en cada una de las posiciones ( $a$  y  $b$ ), el robot tiene dos opciones de ángulos a seguir. De manera que existe un total de 4 combinaciones ( $2^2$ ). Las combinaciones obtenidas se describen en la tabla 3.1.

Combinación	Punto a	Punto b
1	$g_{a1}$	$g_{b1}$
2	$g_{a1}$	$g_{b2}$
3	$g_{a2}$	$g_{b1}$
4	$g_{a2}$	$g_{b2}$

Tabla 3.1. Combinaciones entre los giros

Ahora es necesario verificar si las combinaciones en la tabla anterior son capaces de llevar al robot hacia la orientación final deseada. Esto puede realizarse verificando que la suma del ángulo inicial  $\beta_i$  y la suma entre los giros en los puntos  $a$  y  $b$ , de como resultado  $\beta_f$ .

Para la combinación 1 se debe cumplir que

$$\beta_i + g_{a1} + g_{b1} - \beta_f = 0 \quad (3.24)$$

con

$$g_{a1} = -\beta_i + \gamma \quad \text{y} \quad g_{b1} = \beta_f - \gamma \quad (3.25) \text{ y } (3.26)$$

donde  $\gamma$  es el ángulo medido de la horizontal hacia la línea recta que une los puntos inicial y final del robot. Sustituyendo las ecuaciones para  $g_{a1}$  y  $g_{b1}$  en (3.24) se tiene

$$\beta_i - \beta_i + \gamma + \beta_f - \gamma - \beta_f = 0 \quad (3.27)$$

finalmente

$$\beta_f = \beta_f \quad (3.28)$$

Lo cual quiere decir que siguiendo el camino descrito por la combinación 1, el robot es capaz de alcanzar la orientación final deseada.

Analizando la combinación 2 se tiene que

$$\beta_i + g_{a1} + g_{b2} - \beta_f = 0 \quad (3.29)$$

con

$$g_{a1} = -\beta_i + \gamma \quad \text{y} \quad g_{b2} = -180^\circ + \beta_f - \gamma \quad (3.30) \text{ y } (3.31)$$

sustituyendo en (3.29)

$$\beta_i - \beta_i + \gamma - 180^\circ + \beta_f - \gamma - \beta_f = 0 \quad (3.32)$$

finalmente

$$-180^\circ + \beta_f \neq \beta_f \quad (3.33)$$

Lo que significa que la combinación 2 no es capaz de llevar al robot a la orientación final deseada.

De la misma forma se prueba para cada una de las combinaciones 3 y 4, resultando que sólo las combinaciones 1 y 4 logran llevar al cuerpo del robot hacia la orientación final deseada.

Ahora sólo interesan dos combinaciones. Cada combinación posee una magnitud de giro  $f$  diferente de la otra, la cual se calcula de la siguiente forma

$$f_1 = |g_{a1}| + |g_{b1}| \quad \text{y} \quad f_2 = |g_{a2}| + |g_{b2}| \quad (3.34) \text{ y } (3.35)$$

Para este caso donde se tienen únicamente dos variables a comparar ( $f_1$  y  $f_2$ ), no se considera necesario utilizar un criterio adicional a la simple comparación entre valores; es decir, si  $f_1 \leq f_2$  el robot debe seguir la combinación 1 para realizar la menor cantidad de giro posible, en el caso contrario ( $f_1 > f_2$ ) debe seguir el camino descrito por la combinación 4.

### ***Sentido de avance lineal***

Cabe mencionar que dependiendo de la combinación de giros a seguir, será el sentido de avance lineal (de frente o reversa) para el robot a lo largo de la trayectoria. Seguir la combinación 1, implica que el robot (después de alinearse con la línea recta) avanzará de *frente* hacia el punto  $b$ . Seguir la combinación 2, implica que el robot avanzará de *reversa* hacia el punto  $b$ .

Dentro del problema de distancia entera existe un caso particular para la minimización en la cantidad de giro; este caso ocurre cuando  $D = 0$ .

### ***Caso particular $D = 0$***

Para este caso se tiene que los puntos  $a$  y  $b$  coinciden en el plano. Por lo que el robot únicamente debe realizar un giro hasta orientarse con  $\beta_f$ . Existen dos posibles soluciones en este caso;  $g_1$  y  $g_2$ , figura 3.8.

El valor para  $g_1$  y  $g_2$  se calcula de la siguiente manera

$$g_1 = \beta_f - \beta_i \quad \text{y} \quad g_2 = -360^\circ + \beta_f - \beta_i \quad (3.36) \text{ y } (3.37)$$

Para seleccionar el ángulo que demande menor cantidad de giro para el robot, basta comparar las ambas magnitudes ( $|g_1|$  y  $|g_2|$ ), y así elegir el ángulo de menor magnitud.

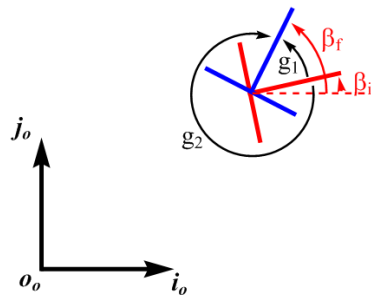


Figura 3.8. Caso particular,  $D = 0$

Con el caso particular presentado anteriormente, se da por concluido el caso de *distancia entera* ( $0 < D < R$ ).

### 3.1.4.2 Trayectoria para Corta Distancia

Como anteriormente se ilustró para este caso (figura 3.6), la trayectoria con la menor distancia posible está compuesta por dos pasos enteros, formando así lo que puede apreciarse como un triángulo isósceles. Hay que hacer la observación que es posible proponer una segunda trayectoria con el mismo número de pasos; la cual se trata de un segundo triángulo con las mismas dimensiones que el primero (área y perímetro), figura 3.9. Ambos triángulos (superior e inferior) llevan al robot de un lugar a otro con la menor cantidad de pasos. Sin embargo, hay que recordar que minimizar la cantidad de giro total que realiza el robot también es de interés. Por lo tanto es necesario definir los ángulos involucrados en cada una de las posibles trayectorias.

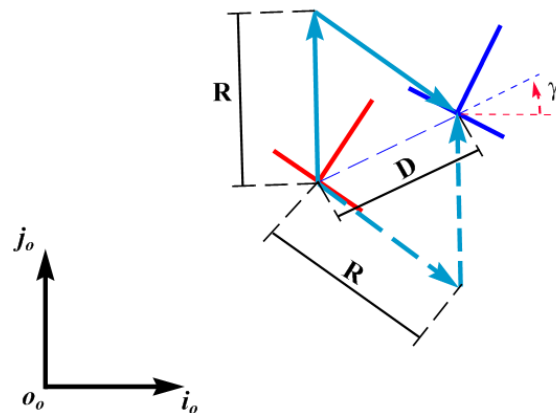


Figura 3.9. Dos posibles trayectorias a corta distancia

En la figura 3.10 se muestra cómo están medidos cada uno de los giros en cada uno de los triángulos. A diferencia del problema de distancia entera, ahora existen 3 puntos (en cada uno de los triángulos) donde debe elegirse sólo uno de dos posibles ángulos a seguir.

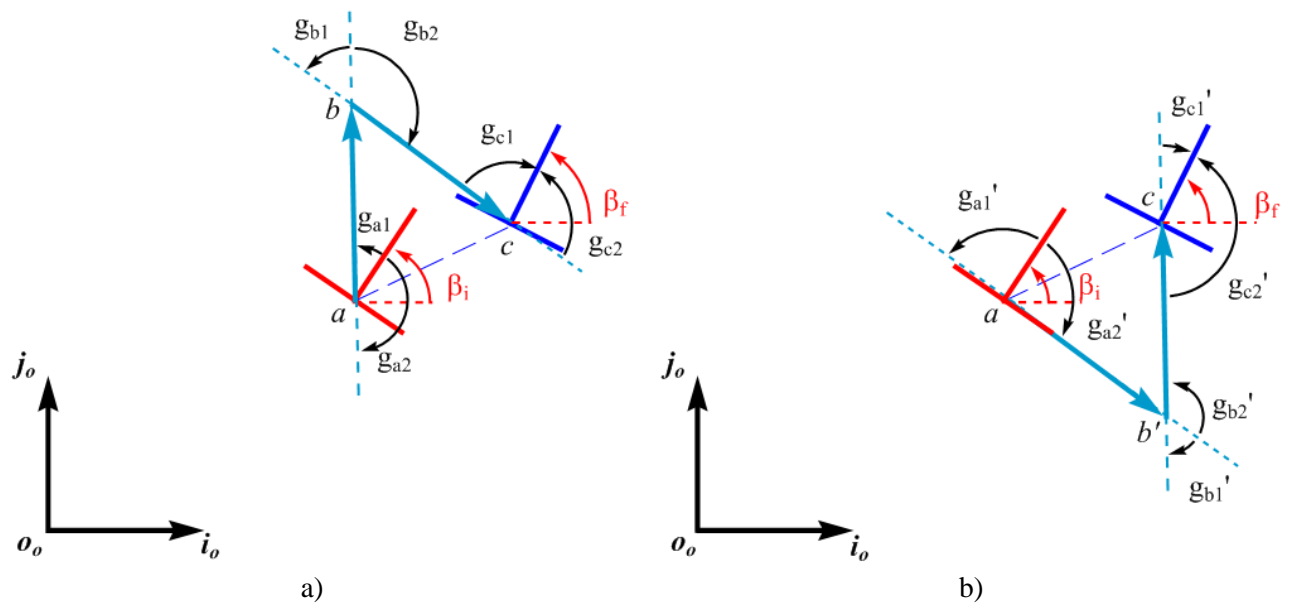


Figura 3.10. Ángulos de giro para el triángulo superior (a) e inferior (b)

De la misma forma que en caso de *distancia entera*, se debe analizar cada una de las posibles combinaciones de los giros, para verificar que sean capaces de llevar al robot a la orientación final deseada. En el caso del triángulo superior se tiene un total de  $2^3$  combinaciones. Empleando el mismo criterio que en el caso de distancia entera para evaluar cada combinación, debe cumplirse

$$\beta_i + g_a + g_b + g_c = \beta_f \quad (3.38)$$

siendo  $g_a, g_b, g_c$  el ángulo seleccionado en cada vértice del triángulo superior. De esta manera se tiene que sólo 4 de las 8 combinaciones en el triángulo superior, son capaces de orientar al robot con  $\beta_f$ . Las 4 combinaciones que ahora son de interés se muestran en la tabla 3.2.

Combinación	Ángulo elegido		
	Punto a	Punto b	Punto c
1	$g_{a1}$	$g_{b1}$	$g_{c1}$
2	$g_{a1}$	$g_{b2}$	$g_{c2}$
3	$g_{a2}$	$g_{b1}$	$g_{c2}$
4	$g_{a2}$	$g_{b2}$	$g_{c1}$

Tabla 3.2. Combinaciones de giros para el triángulo superior

Analizando de la misma forma el triángulo inferior, se tienen que las 4 combinaciones que llevan al robot hasta la orientación final deseada son

Combinación	Ángulo elegido		
	Punto a	Punto b'	Punto c
5	$g_{a1}'$	$g_{b1}'$	$g_{c1}'$
6	$g_{a1}'$	$g_{b2}'$	$g_{c2}'$
7	$g_{a2}'$	$g_{b1}'$	$g_{c2}'$
8	$g_{a2}'$	$g_{b2}'$	$g_{c1}'$

Tabla 3.3. Combinaciones de giros para el triángulo inferior

Se tiene que el total de caminos (combinaciones) en el problema de *corta distancia* es 8. De estos posibles caminos, el robot debe seguir aquel con la menor magnitud de giro. Para ello debe evaluarse la magnitud de cada una de las combinaciones y así obtener la menor de ellas.

La magnitud de cada camino se obtuvo de la siguiente forma

$$\begin{aligned}
 f_1 &= |g_{a1}| + |g_{b1}| + |g_{c1}| & f_5 &= |g_{a1}'| + |g_{b1}'| + |g_{c1}'| \\
 f_2 &= |g_{a1}| + |g_{b2}| + |g_{c2}| & f_6 &= |g_{a1}'| + |g_{b2}'| + |g_{c2}'| \\
 f_3 &= |g_{a2}| + |g_{b1}| + |g_{c2}| & f_7 &= |g_{a2}'| + |g_{b1}'| + |g_{c2}'| \\
 f_4 &= |g_{a2}| + |g_{b2}| + |g_{c1}| & f_8 &= |g_{a2}'| + |g_{b2}'| + |g_{c1}'|
 \end{aligned} \tag{3.39}$$

Una vez calculadas todas la magnitudes  $f_j$  ( $j = 1, \dots, 8$ ) se elige la menor de ellas, y el robot debe seguir los ángulos que componen dicha combinación. Cabe mencionar que se obtuvo un método analítico para encontrar el camino de menor magnitud, sin necesidad del cálculo de las 8 magnitudes. Sin embargo, tal método requiere una serie extensa de comparaciones entre los valores de orientación inicial, final y el ángulo  $\gamma$  ( $\gamma$  es la orientación de la línea recta entre las posiciones inicial y final, figura 3.9). Por lo tanto, para fines prácticos de programación y tiempo de ejecución, el camino de menor magnitud se encuentra mediante la *búsqueda exhaustiva*.

### **Sentido de avance lineal**

De igual forma que en el caso de *distancia entera*, el sentido de avance lineal está determinado por los ángulos que componen el camino de menor magnitud. En este caso la trayectoria puede cambiar su sentido de avance en el punto  $b$  del triángulo (punto  $b'$  en el triángulo inferior). La tabla 3.4 muestra como resulta el sentido de avance para combinación de giros.

Camino	Sentido de avance $a \rightarrow b$	Sentido de avance $b \rightarrow c$	Camino	Sentido de avance $a \rightarrow b'$	Sentido de avance $b' \rightarrow c$
1	Frente	Reversa	5	Reversa	Frente
2	Frente	Frente	6	Reversa	Reversa
3	Reversa	Frente	7	Frente	Reversa
4	Reversa	Reversa	8	Frente	Frente

Tabla 3.4. Sentido de avance para cada trayectoria

Los cálculos para cada uno de los involucrados en la trayectoria, se muestran más adelante en el apartado de *Algoritmo de Seguimiento de Trayectorias*.

### 3.1.4.3 Trayectoria para Larga Distancia

El proponer trayectorias para el caso de larga distancia ( $D > R$ ), resulta ser un problema más complejo que los otros dos mencionados anteriormente. Esto debido a la gran variedad de trayectorias que pueden proponerse para que el robot llegue de un punto a otro del plano.

A continuación se presentan 3 propuestas que se analizaron, las cuales pueden dar solución al problema de larga distancia.

#### *Propuesta 1*

Esta propuesta se realizó inspirada en las trayectorias que realizan los robots con ruedas, es decir trayectorias curvilíneas (línea discontinua en la figura 3.11). Claramente, para el robot en este trabajo es muy difícil seguir tal tipo de trayectorias. Sin embargo, es posible construir una trayectoria discreta (construida a partir de pasos constantes) que se asemeje a una trayectoria curva, como se aprecia en la figura siguiente.

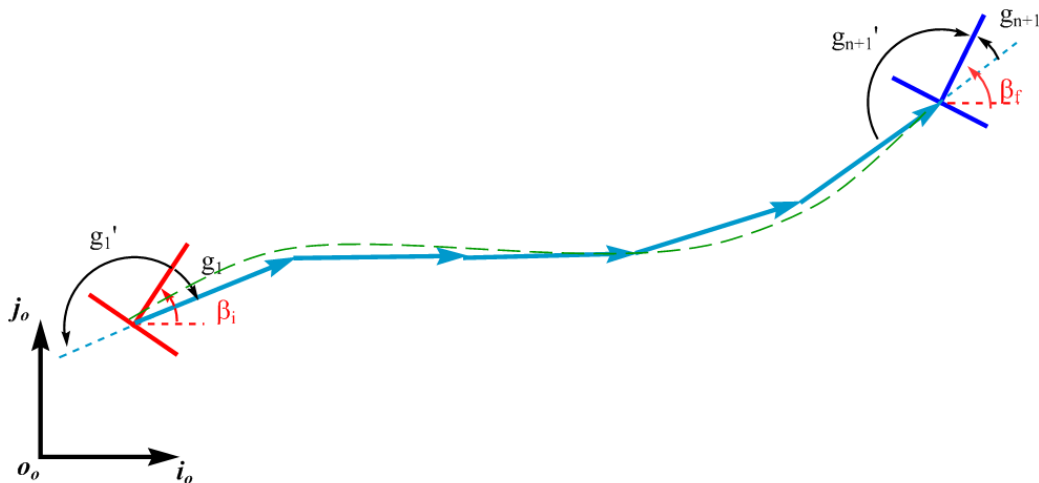


Figura 3.11. Propuesta 1 para el caso de larga distancia

Algunas características que posee esta propuesta son:

- El grado de complejidad se ve incrementado debido al número de giros involucrados. El número de giros en la trayectoria es de  $n + 1$ , donde  $n$  es el número mínimo de pasos calculado en la ecuación (3.22), figura 3.11.

- Mientras la distancia  $D$  entre la posición inicial y final del robot sea mayor, las trayectorias del robot pueden aproximarse en mayor medida a trayectorias curvas. Cabe mencionar que esta propiedad no es relevante en este trabajo.
- Independientemente de la forma que tenga la trayectoria (arco o senoidal), la distancia a recorrer siempre será  $nR$ .
- A pesar de que en este trabajo se considera que no existe derrape entre la pata de giro del robot y la superficie, no es posible ignorar que en la realidad va existir dicho derrape. Por lo tanto, un mayor número de giros ocasionará un error mayor a lo largo de la trayectoria. Cabe mencionar que existen otras condiciones que afectan al desplazamiento del dispositivo y que tampoco se están considerando en los análisis realizados.

## Propuesta 2

La siguiente propuesta combina los casos de *distancia entera* y *corta distancia*. Su objetivo es llevar al robot en línea recta hacia la posición final para finalmente realizar una maniobra muy similar al estacionado de un vehículo, figura 3.12.

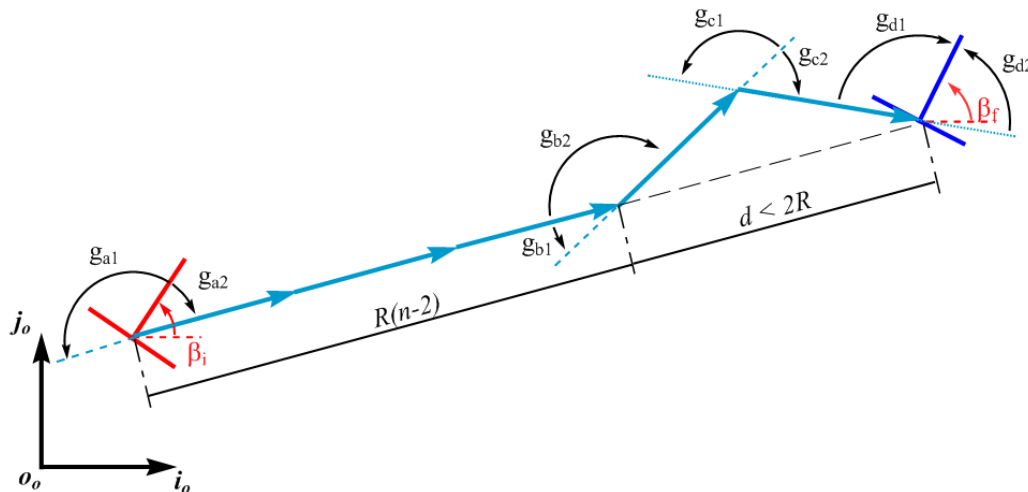


Figura 3.12. Propuesta 2 para el caso de larga distancia

Al igual que en caso para *corta distancia*, el triángulo que se aprecia en la figura anterior posee una contraparte con diferentes magnitudes y sentidos de giro (triángulo inferior). Esta contraparte también merece ser considerada si se desea minimizar la cantidad de giro total a lo largo de la trayectoria.

En esta trayectoria es posible modificar la ubicación de la maniobra; es decir, el robot puede realizar inicialmente la trayectoria trazada como un triángulo, para luego seguir en línea recta hasta la posición final y orientarse con  $\beta_f$ . De igual manera, es posible realizar la maniobra a mitad de la trayectoria, sin embargo esto incrementaría en 1 el número de giros a lo largo de trayectoria. Debido



a la poca practicidad que representa realizar la maniobra a la mitad de la trayectoria, esta opción queda descartada y sólo se conserva la opción de maniobrar al inicio o fin de la trayectoria. Por lo tanto, son cuatro posibles trayectorias en esta propuesta:

1. Triángulo superior al inicio de la trayectoria.
2. Triángulo inferior al inicio de la trayectoria.
3. Triángulo superior al final de la trayectoria
4. Triángulo inferior al final de la trayectoria

Como principales características de la propuesta 2 se tienen:

- Posee un número limitado de soluciones a analizar. Cada una de las 4 trayectorias mencionadas anteriormente posee 8 combinaciones diferentes de giros. Por lo tanto, existen 32 caminos diferentes que llevan al robot a la posición y orientación final deseadas, de los cuales se debe elegir el de menor magnitud.
- Puede *reutilizarse* parte del algoritmo (presentado más adelante) de *distancia entera y corta distancia* para construir el algoritmo de *larga distancia*.
- El giro utilizado para orientar al robot con la línea recta que une ambas posiciones, podría llegar a ser redundante. Esto por el hecho de utilizar un giro para llevar al robot a una orientación independiente de  $\beta_i$  y  $\beta_f$ .

### ***Propuesta 3***

Existe la posibilidad de crear trayectorias a partir de dos círculos de diferente radio y con centro en la posición inicial y final del robot cada uno. Estos círculos deben poseer un radio que sea múltiplo entero de la magnitud del paso del robot ( $R$ ). De manera que inicialmente se propone que el radio del círculo con centro en  $(x_i, y_i)$  tenga un radio de magnitud  $mR$ , y el círculo con centro en  $(x_f, y_f)$  tenga un radio  $lR$  de longitud, donde  $m, l \in \mathbb{N}$ . La finalidad de proponer ambos círculos, es utilizar los puntos donde se intersecan como punto intermedio de la trayectoria. Es decir, el robot debe dirigirse en línea recta de la posición inicial hacia una de las intersecciones de los círculos; para luego orientarse hacia la posición final y avanzar de nueva cuenta en línea recta; finalmente el robot debe alinearse con el ángulo  $\beta_f$ .

Es importante definir la relación que existe entre  $m$  y  $l$ . Debido al valor de mínima distancia obtenido en este caso  $D > R$ , se tiene que el número de pasos en esta trayectoria debe ser  $n = \text{int}\left(\frac{D}{R}\right) + 1$ . Por lo tanto, se tiene de manera estricta que

$$m + l = n \tag{3.40}$$

Nótese que si  $m + l$  fuese menor a  $n$ , no existiría intersección entre ambas circunferencias. En el caso donde  $m + l > n$  no se estaría respetando el criterio de mínima distancia.

En la figura 3.13 se aprecian los conceptos antes mencionados. Cada una de las flechas contiguas indican un paso del robot, los puntos donde intersecan las circunferencias están indicados con líneas punteadas.

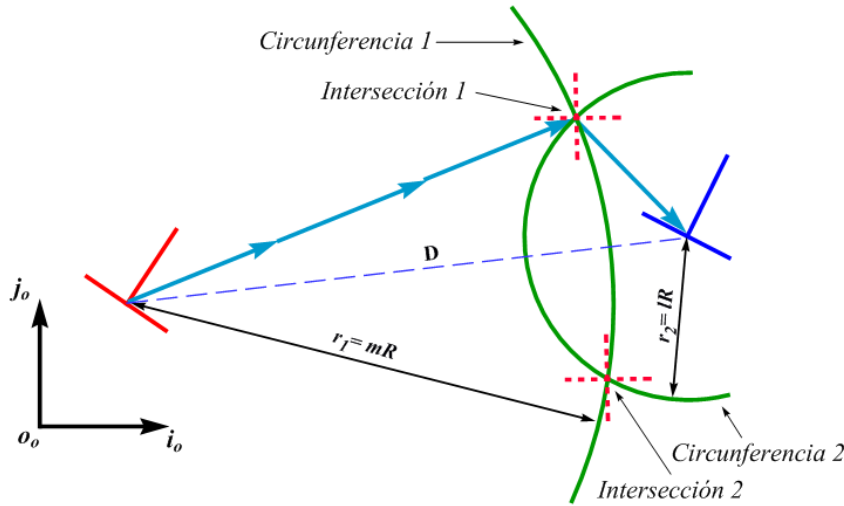


Figura 3.13. Propuesta 3 para el caso de larga distancia

A diferencia de la propuesta 2 donde se analizan triángulos isósceles; en esta propuesta puede apreciarse la forma de un triángulo escaleno, el cuál puede analizarse por métodos algebraicos. En la presente propuesta también se manejan los conceptos de *triángulo superior* y *triángulo inferior*. Esto debido a las 2 intersecciones que se generan debido al tamaño de ambos círculos.

Una vez encontrados los puntos de intersección, se deja a un lado el concepto de los círculos y se analiza como un problema que puede resolverse mediante un triángulo escaleno; el cual tiene la posibilidad de variar la magnitud de sus lados ( $mR$  y  $lR$ ), más no su base ( $D$ ) ni perímetro ( $nR + D$ ).

En la figura 3.14 se muestra como pueden ser definidos los ángulos de giro involucrados en la trayectoria (notar similitud con los ángulos definidos para el caso de *corta distancia*). Como ya se mencionó antes, debe considerarse la opción de un triángulo con las mismas dimensiones en la parte inferior. A pesar de que el triángulo inferior cuenta con las mismas dimensiones que el triángulo superior, los ángulos involucrados en el primero resultan ser distintos de los involucrados en el segundo.

De igual manera que el caso de *corta distancia*, cada triángulo en esta trayectoria posee 4 combinaciones que llevan al robot a la orientación final deseada. De igual manera, cada una de estas combinaciones posee diferente magnitud de giro, por lo que requieren ser evaluadas y así elegir la de menor magnitud.

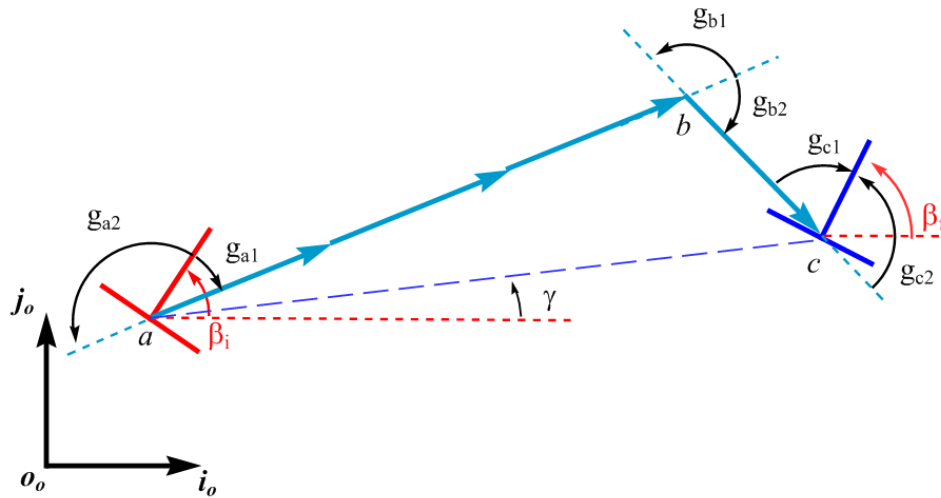


Figura 3.14. Definición de ángulos en la trayectoria

Las características presentes en la tercera propuesta son:

- Al igual que en la segunda propuesta, existe un número finito de soluciones. El número de triángulos que pueden formarse no es infinito, y cada triángulo a su vez posee un número finito de soluciones que pueden ser analizadas.
- A diferencia de la propuesta 2, ninguno de los giros se considera redundante.
- Parte del algoritmo de *corta distancia* puede ser reutilizado para encontrar las posibles soluciones en la presente trayectoria.

### ***Elección de propuesta para el caso de larga distancia***

De las tres propuestas anteriores, la propuesta 3 parece ser la más conveniente para el robot en este trabajo. Comparando las propuestas entre sí, se tiene:

- *Propuesta 3 vs. Propuesta 1.* Generar la solución para los giros involucrados, resulta ser mucho más sencillo en la propuesta 3 que en la propuesta 1. Lo cual implica que es posible realizar un algoritmo que se ejecute en tiempo real. Es decir que el robot pueda procesar al mismo tiempo que realiza la trayectoria.
- *Propuesta 3 vs. Propuesta 1.* El número de giros que se ejecutan en la propuesta 1 es visiblemente mayor que en la propuesta 3. Por lo tanto es muy probable que el error acarreado en la orientación final del robot, sea mayor en a propuesta 1.
- *Propuesta 3 vs. Propuesta 2.* La propuesta 2 contiene un número menor de posibles soluciones y resulta más sencilla la selección de la óptima entre ellas. Sin embargo, la cantidad de posibles soluciones en la propuesta 3 no es infinito y pueden calcularse mediante operaciones sencillas.

- *Propuesta 3 vs. Propuesta 2.* En la propuesta 2 el robot debe realizar un giro más que en la propuesta 3, lo cual, como se mencionó anteriormente, puede resultar un giro innecesario. Hablando en términos de tiempo de ejecución, es obvio que el robot consumirá más tiempo en realizar un giro (por pequeño que este sea), que en procesar una solución con un número finito de posibilidades.

De manera que la propuesta 3 fue la que se decidió desarrollar a profundidad y la cual se analiza a continuación.

### 3.1.4.3.1 Análisis de la Trayectoria para el Caso de Larga Distancia

Como se mencionó anteriormente, esta trayectoria se basa en el análisis de triángulos escalenos y la relación que guardan con los ángulos que debe realizar el robot en cada uno de sus vértices.

Ahora, hay que recordar que la menor distancia está dada por  $Z = nR$ , donde  $n = \text{int}\left(\frac{D}{R}\right) + 1$ . Por lo tanto sólo resta minimizar la cantidad total de giro a lo largo de la trayectoria.

La mínima cantidad de giro puede ser encontrada de dos formas distintas:

#### 1. Optimizando una función objetivo

De igual manera que en el caso de *corta distancia*, cada triángulo (superior e inferior) en la trayectoria posee 4 posibles soluciones. Pero ahora los triángulos pueden variar la longitud de sus lados de manera discreta. Es decir, modificar los valores de  $m$  y  $l$  con las restricciones de  $m + l = n$  y  $m, l \in \mathbb{N}$ . Modificar la magnitud de sus lados, afecta de manera directa la magnitud de los giros involucrados en la trayectoria.

Este problema puede considerarse como un problema de *programación entera*, debido a que la función de costo (para la magnitud de giro) está en término de variables que sólo pueden tomar valores enteros  $(m, l)$ . De este modo, es posible encontrar el valor de  $m$  y  $l$  con el cual se minimiza la cantidad de giro en cada uno de los dos triángulos. Una vez obtenido el valor mínimo en cada triángulo, se elige el triángulo con el valor mínimo de giro.

#### 2. Búsqueda exhaustiva

En lugar de considerar que se trata de sólo dos triángulos capaces de variar la magnitud de sus lados, puede considerarse que se trata de un número  $s$  de triángulos, con lados y combinaciones de giro distintas. En la figura 3.15 se observa un ejemplo del concepto anterior. Es posible calcular el número de triángulos que pueden formarse en base al número de pasos en la trayectoria. Por ejemplo, en la figura se muestra una trayectoria compuesta por 4 pasos enteros, y en la parte superior se forman 3 triángulos  $(n - 1)$ , y el mismo número de triángulos en la parte inferior  $(n - 1)$ . Por lo tanto, el número de triángulos que pueden formarse con una trayectoria de  $n$  pasos es

$s = 2(n - 1)$ . Anteriormente se observó que cada triángulo posee 4 posibles combinaciones de giros, de manera que el número total de soluciones es  $n. s. = 4 \cdot 2(n - 1) = 8(n - 1)$ .

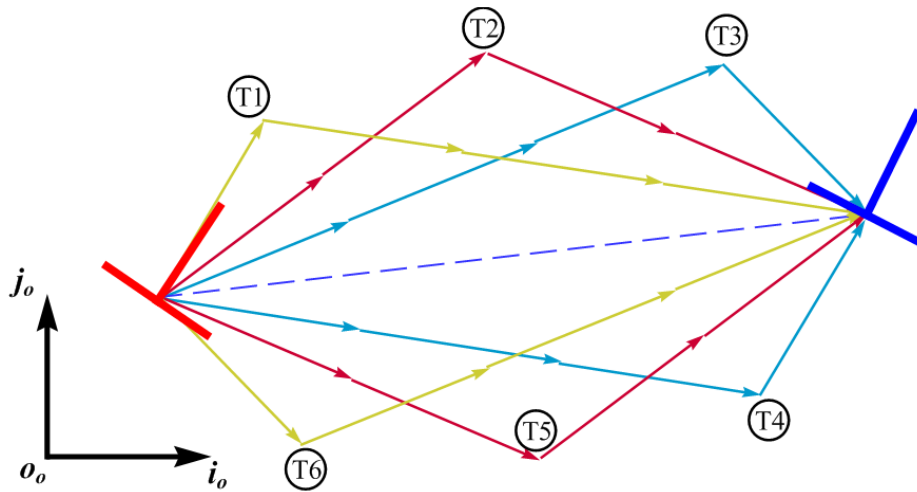


Figura 3.15. Número de triángulos en una trayectoria de 4 pasos

El método de búsqueda exhaustiva requiere la evaluación de todas las posibles soluciones en el problema a optimizar. Es por ello que inicialmente se busca dar solución optimizando una función objetivo.

### ***Función objetivo***

Cabe mencionar que se realizó un esfuerzo por diseñar una sola función objetivo, que englobara todas las posibles soluciones en este problema; esto con la intención de tener que minimizar únicamente una sola función. Sin embargo, esto no fue posible debido a que la magnitud de cada una de las posibles combinaciones (4 por cada triángulo) es diferente de las otras. Por lo tanto se tiene un total de 8 funciones a minimizar, una por cada combinación de giros. Una vez que se realice la minimización de las 8 funciones, deben compararse los valores (ya minimizados) de cada una de ellas y elegir la combinación con la magnitud más pequeña.

La función objetivo para cada una de las combinaciones debe estar en términos de  $m$  y  $l$ , puesto que son las variables que modifican el valor de magnitud de giro en cada combinación. Teniendo en cuenta lo anterior, se comienza con la combinación 1 de giros:  $C_1 = g_{a1} \rightarrow g_{b1} \rightarrow g_{c1}$ .

La función para calcular la magnitud de  $C_1$  es

$$f_1 = |g_{a1}| + |g_{b1}| + |g_{c1}| \quad (3.41)$$

Las variables  $m$  y  $l$  se encuentran implícitas en  $g_{ab}$ ,  $g_{bl}$  y  $g_{cl}$ . Por lo tanto es necesario encontrar el valor de cada uno de los ángulos en términos de  $m$  y  $l$ .

El valor de cada uno de los giros puede obtenerse geoméricamente con ayuda de la figura 3.16.

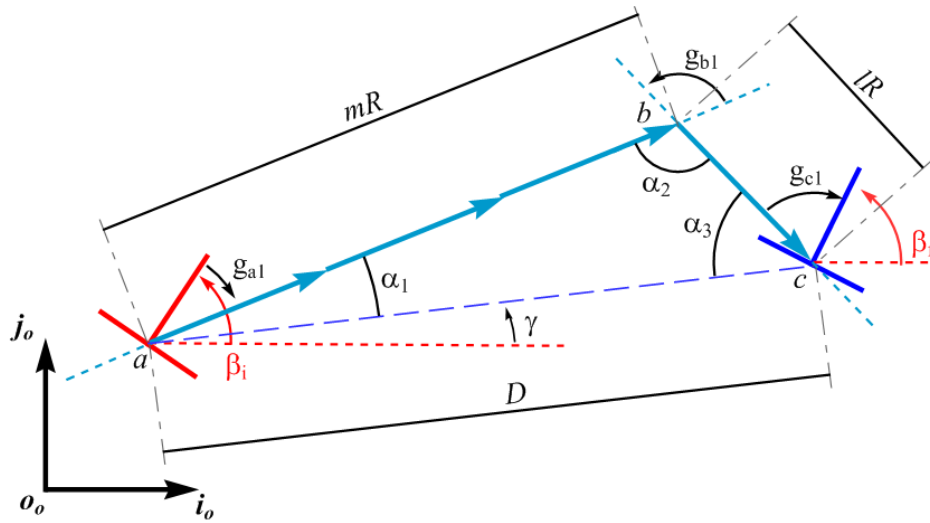


Figura 3.16. Relación de giros y ángulos en la combinación 1

De la figura anterior se tiene que

$$\begin{aligned} g_{a1} &= -\beta_i + \gamma + \alpha_1 \\ g_{b1} &= \alpha_2 \\ g_{c1} &= -180^\circ + \beta_f - \gamma + \alpha_3 \end{aligned} \quad (3.42)$$

donde  $\alpha_1$ ,  $\alpha_2$  y  $\alpha_3$  son los ángulos internos del triángulo;  $\beta_i$  y  $\beta_f$  son la orientaciones inicial y final del robot; y  $\gamma$  es el ángulo medido de la horizontal a la línea recta que une los puntos  $a$  y  $b$ .

Sustituyendo (3.42) en (3.41) se tiene

$$f_1 = |-\beta_i + \gamma + \alpha_1| + |\alpha_2| + |-180^\circ + \beta_f - \gamma + \alpha_3| \quad (3.43)$$

Es necesario expresar los ángulos  $\alpha_1$ ,  $\alpha_2$  y  $\alpha_3$  en términos de  $m$  y  $l$ . Para esto se recurre a la *ley de cosenos*, donde se encuentra que:

$$\alpha_1 = \arcsin\left(\frac{lR}{D} \sin\left(\arccos\left(\frac{(mR)^2 + (lR)^2 - D^2}{2mlR^2}\right)\right)\right) \quad (3.44)$$

$$\alpha_2 = \arccos\left(\frac{(mR)^2 + (lR)^2 - D^2}{2mlR^2}\right) \quad (3.45)$$

$$\alpha_3 = \arcsin\left(\frac{mR}{D} \sin\left(\arccos\left(\frac{(mR)^2 + (lR)^2 - D^2}{2mlR^2}\right)\right)\right) \quad (3.46)$$

Sustituyendo (3.44 -3.46) en (3.43) se tiene

$$\begin{aligned}
f_1 = & \left| -\beta_i + \gamma + \arcsin \left( \frac{lR}{D} \sin \left( \arccos \left( \frac{(mR)^2 + (lR)^2 - D^2}{2mlR^2} \right) \right) \right) \right| \\
& + \left| \arccos \left( \frac{(mR)^2 + (lR)^2 - D^2}{2mlR^2} \right) \right| \\
& + \left| -180^\circ + \beta_f - \gamma + \arcsin \left( \frac{mR}{D} \sin \left( \arccos \left( \frac{(mR)^2 + (lR)^2 - D^2}{2mlR^2} \right) \right) \right) \right|
\end{aligned} \tag{3.47}$$

Cabe mencionar que se realizó un intento por simplificar la función anterior el cual no tuvo resultados satisfactorios.

Una vez que se tienen los valores de posición inicial y final, es posible calcular los valores para  $\gamma$ ,  $D$  y  $n$ . A partir de aquí, los valores de  $\beta_i$ ,  $\beta_f$ ,  $\gamma$ ,  $D$ ,  $R$  y  $n$ , son valores que permanecen constantes a lo largo del problema de optimización, de manera que la únicas variables en la ecuación anterior son  $m$  y  $l$ . Sin embargo, es posible expresar  $l$  en términos de  $m$  debido a que  $n = m + l$ , de aquí se tiene que  $l = n - m$ . Por lo tanto, la función objetivo puede re expresarse como

$$\begin{aligned}
f_1(m) = & \left| -\beta_i + \gamma + \arcsin \left( \frac{(n-m)R}{D} \sin \left( \arccos \left( \frac{(mR)^2 + ((n-m)R)^2 - D^2}{2m(n-m)R^2} \right) \right) \right) \right| \\
& + \left| \arccos \left( \frac{(mR)^2 + ((n-m)R)^2 - D^2}{2m(n-m)R^2} \right) \right| \\
& + \left| -180^\circ + \beta_f - \gamma + \arcsin \left( \frac{mR}{D} \sin \left( \arccos \left( \frac{(mR)^2 + ((n-m)R)^2 - D^2}{2m(n-m)R^2} \right) \right) \right) \right|
\end{aligned} \tag{3.48}$$

Para la minimización de  $f_1$  debe obtenerse en primera instancia la solución relajada del problema, es decir, ignorar que  $m$  debe ser un número natural. Para obtener la solución relajada, es posible recurrir a la primera y segunda derivada de  $f_1$  para obtener los máximos y mínimos de esta función. Sin embargo, es de esperarse que  $f_1'$  y  $f_1''$  resulten de una complejidad similar a  $f_1$ . Por ello vale la pena analizar de manera gráfica el comportamiento de  $f_1$  respecto a  $m$ , y de esta manera observar la cantidad de puntos de inflexión que deben ser analizados mediante  $f_1$ ,  $f_1'$  y  $f_1''$ .

Para un caso particular donde  $D$  resulta ser mayor que  $R$ , se observa el siguiente comportamiento de  $f_1$ , figura 3.17.

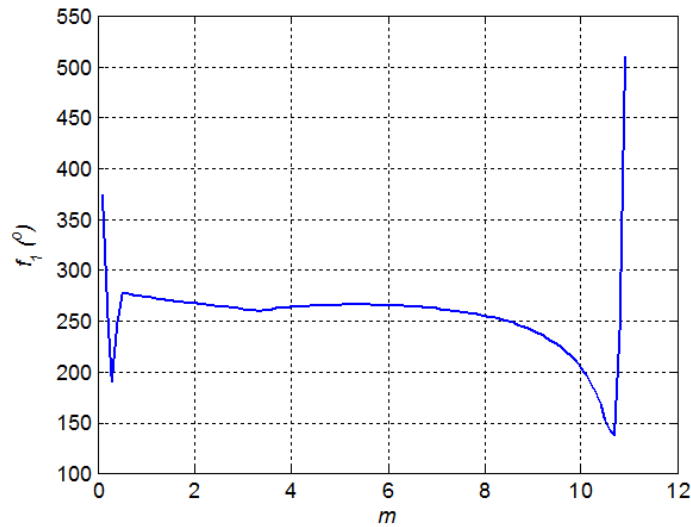


Figura 3.17. Comportamiento de  $f_1$  para un caso particular

En la figura anterior puede apreciarse un número considerable de puntos de inflexión. Por lo que la obtención del valor mínimo de  $f_1$  no es tarea sencilla, considerando la complejidad de las funciones involucradas.

Suponiendo que se obtienen los diferentes valores de  $m$  correspondientes a los puntos mínimos en  $f_1$ , resta por ver si estos son valores enteros. De no serlo, deben evaluarse en  $f_1$  los puntos vecinos con valor entero, para posteriormente realizar una comparación entre los diferentes valores de  $f_1$  generados y así seleccionar el menor de ellos.

Claramente, el procedimiento anterior demanda una serie de cálculos complejos, además hay que recordar que debe realizarse para 8 funciones distintas. Por ello no se considera como una solución práctica realizar la minimización de giro en una trayectoria.

### ***Búsqueda exhaustiva***

Anteriormente se calculó el número de soluciones que existen para el problema de larga distancia  $n.s. = 8(n - 1)$ . Afortunadamente no se trata de un número que crece exponencialmente respecto de  $n$ , y es posible para un procesador actual calcular todas las soluciones en un tiempo relativamente pequeño. Por lo que el método de *búsqueda exhaustiva* representa una buena alternativa para el cálculo y la selección de las variables  $m$  y  $l$ . Es importante recordar que lo que se busca en este trabajo de tesis es minimizar distancia y magnitud de giro a lo largo de una trayectoria, y no el minimizar el costo computacional que demande el cálculo de la trayectoria.

Una ventaja que presenta este método en comparación al anterior, es que no se usa la función en su máxima expresión. Esto debido a que los cálculos realizados para una combinación de giros, pueden



ser reutilizados en el cálculo de otras combinaciones. Otra ventaja es que no requiere la evaluación de los puntos candidatos, en la primera y segunda derivada de la función objetivo. Una explicación detallada de cómo se implementó el método de *búsqueda exhaustiva* se verá en la siguiente sección.

## 3.2 Algoritmo de Seguimiento de Trayectorias

Está visto que el robot para el cual se desarrolló este trabajo de tesis, tendrá diferentes formas de resolver el problema de llegar de un punto a otro. Esto debido al tipo de trayectorias que es capaz de realizar, y a los requerimientos que debe cumplir en cada uno de ellas. En esta sección se presenta un algoritmo de seguimiento de trayectorias, el cual tiene la finalidad de conducir al robot a lo largo de una trayectoria sobre el plano.

Puede que inicialmente se tenga una impresión errónea de sencillez respecto a implementar un algoritmo para seguir las trayectorias propuestas. Sin embargo, hay que considerar que el algoritmo debe realizarse de manera que el robot sea capaz de procesar cada instrucción (no simplemente por ejemplo: “sigue  $g_{a1}$ ; después avanza hasta llegar a  $b$ ; gira  $g_{b1}$ ; avanza hasta llegar a  $c$ ; gira  $g_{c1}$ ”). Por lo tanto, es importante analizar la manera de darle las instrucciones al robot. Pensando en cómo puede procesar el robot las instrucciones para realizar una trayectoria, se desarrolló un simulador, el cual contiene el algoritmo de seguimiento de trayectorias y un código de animación de las mismas. Este último puede verse como la parte de control encargada de los movimientos en el robot. Más detalles se presentan enseguida.

### 3.2.1 Simulador

Para validar las trayectorias propuestas en este trabajo se desarrolló un simulador. Este está dividido en dos partes: en la primera se encuentra el algoritmo de generación de trayectorias y en la segunda se desarrolló un código encargado de la animación de la trayectoria elegida en la primera parte. Debido a que se buscaba que el código de animación funcionara de una manera general (para los tres tipos de trayectorias), se tuvo que adoptar una forma general en las variables que maneja la parte del algoritmo.

Ambas partes del simulador se explican con mayor detalle en los dos apartados siguientes.

#### 3.2.1.1 Algoritmo de Seguimiento de Trayectorias

A continuación se presenta el diagrama de flujo para el algoritmo de seguimiento de trayectorias. Debido a la extensión del mismo, el diagrama de flujo se presenta de manera seccionada usando conectores (círculos con numeración).

## Inicio

En esta parte se da inicio al algoritmo. Se declara el valor de la constante  $R$  y se reciben los datos de posición  $((x_i, y_i), (x_f, y_f))$  y orientación inicial  $(\beta_i)$  y final  $(\beta_f)$ . Se calcula el valor de la distancia  $D$  entre ambas posiciones, y el valor del ángulo entre la horizontal y la línea que une ambas posiciones  $\gamma$ . También se define que el tipo de problema en el que se encuentra el robot; distancia entera ( $\text{mod}(\frac{D}{R}) = 0$ ), corta ( $D < R$ ) o (larga  $D > R$ ) distancia. Ver figura 3.18.

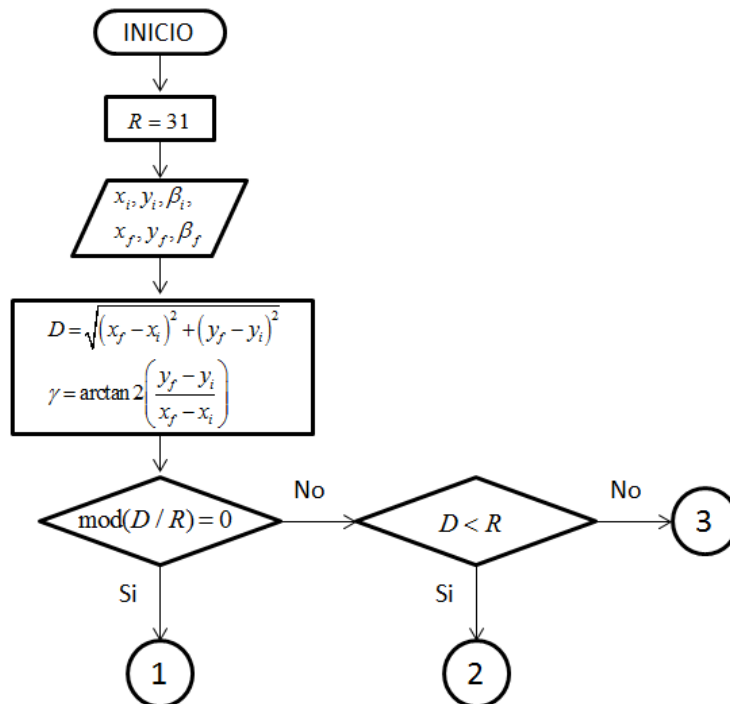


Figura 3.18. Sección de Inicio en el Algoritmo de Trayectorias

## Sección 1

En esta sección se realizan los cálculos necesarios cuando se trata de un problema de distancia entera. Inicialmente se calcula el número de pasos necesarios en la trayectoria ( $n = D/R$ ). Hay que recordar que dentro de este problema existe un caso particular ( $n = 0$ ). Ver figura 3.19.

**Caso  $n = 0$ .** En este caso el robot sólo tiene girar hasta orientarse con  $\beta_f$ , para ello debe realizar un giro más corto entre  $\beta_i$  y  $\beta_f$  ( $g_{al}$ ). De manera que la instrucción para el robot debe ser: girar  $G1$  grados en sentido  $s1$ . En donde  $G1$  es la magnitud de  $g_{al}$  y  $s1$  la dirección de primer giro; si  $s1 = 1$  entonces el sentido de giro es anti horario, si  $s1 = -1$  entonces será en sentido horario. Las magnitudes de giro restantes  $G2$  y  $G3$ , y las magnitudes de los avances  $m$  y  $l$ , son igual a cero debido a que no son necesarias. Finalmente estos resultados se envían al código de animación  $E$ .

**Caso  $n \neq 0$ .** Para este caso sólo es necesario usar un giro inicial ( $g_{ai}$ ), un avance lineal de  $n$  pasos ( $m$ ) y un giro final ( $g_{bi}$ ). Aquí aparecen dos combinaciones de giros con diferente magnitud ( $f_1$  y  $f_2$ ), las cuales es necesario comparar para obtener la trayectoria con menor cantidad de giro. Después de elegir la trayectoria con menor giro, se calculan los valores a  $G1$  y  $G2$  (magnitud de giro en el punto  $a$  y  $b$  respectivamente) dependiendo de la combinación elegida. La condición para el sentido de giro en el punto  $a$  ( $s_1$ ), ahora debe considerar que hay dos posibles combinaciones de giro (ver condición en la figura 3.19).

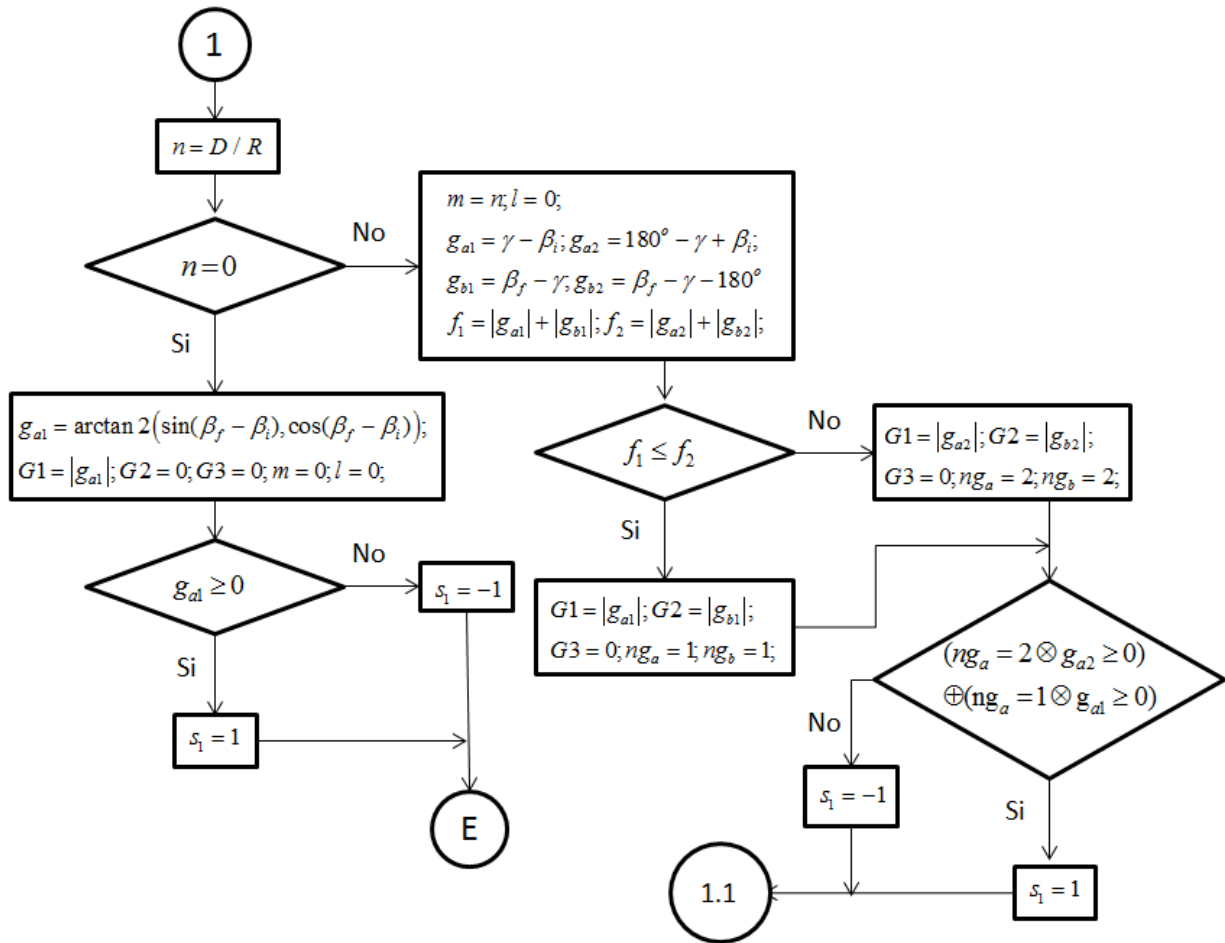


Figura 3.19. Sección 1 del Algoritmo de Trayectorias

### Sección 1.1

De igual manera existe una condición para el sentido de giro en el punto  $b$  ( $s_2$ ). Para obtener el sentido de avance lineal basta con saber la elección del giro en el punto  $a$  ( $ng_a$ ). Si el primer giro fue  $g_{a1}$  ( $ng_a = 1$ ) entonces el sentido de avance será de *frente* ( $av_1 = -1$ ), de lo contrario será de *reversa* ( $av_1 = 1$ ). Los resultados son enviados a  $E$ . Ver figura 3.20.

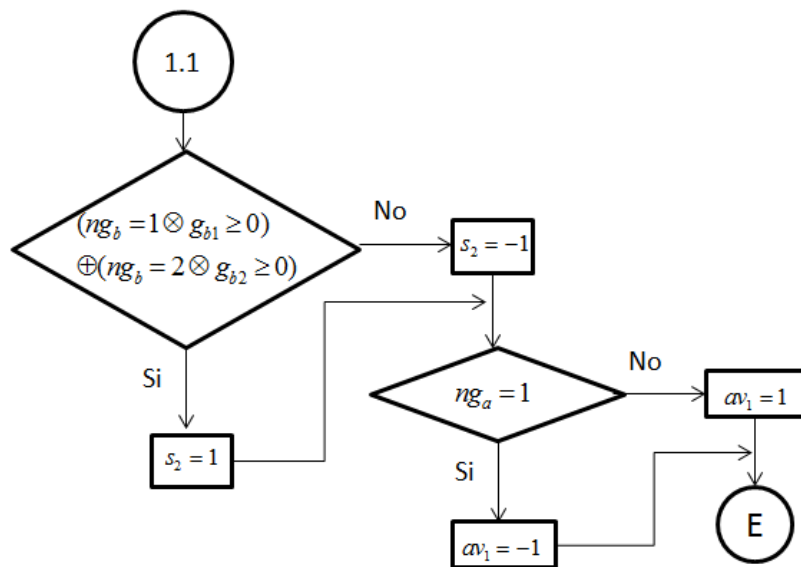


Figura 3.20. Sección 1.1 del Algoritmo de Trayectorias

## Sección 2

Aquí se comienza con el análisis para el caso de *corta distancia*. La intención en este caso es seguir una trayectoria en forma de triángulo isósceles, donde sus lados ( $mR$  y  $lR$ ) tengan la magnitud de un paso entero cada uno ( $m = l = 1$ ). Los ángulos internos del triángulo  $\alpha_1, \alpha_2, \alpha_3$ , están relacionados con los puntos  $a, b, c$  del triángulo (donde  $a$  indica la posición inicial del robot,  $b$  ahora es el punto intermedio de la trayectoria y  $c$  se encuentra en la posición final). Ver figura 3.21.

Como se mencionó anteriormente en este capítulo, en este caso existen 8 posibles combinaciones de giro, las cuales están representadas por las funciones  $f_1$  a  $f_8$  (ver diagrama para esta sección). Mediante un ciclo *for* se realiza la comparación de las 8 posibilidades y se encuentra la de menor magnitud.

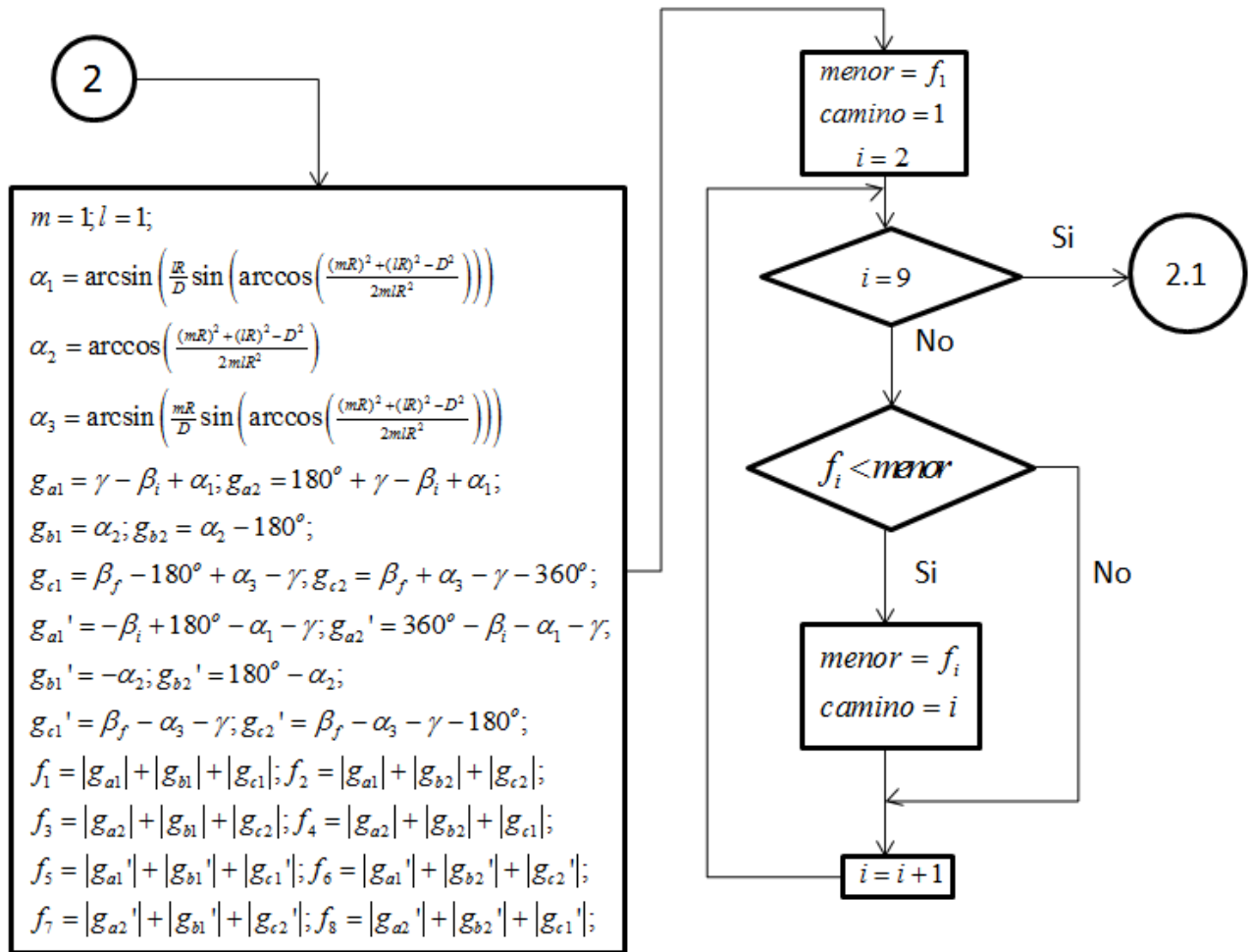


Figura 3.21. Sección 2 del Algoritmo de Trayectorias

## Sección 2.1

Dependiendo de la combinación elegida en la sección anterior, se calcula la magnitud del giro en cada uno de los vértices del triángulo ( $G1, G2, G3$ ). También se guarda la selección de giro en cada vértice, variables auxiliares ( $ng_a, ng_b, ng_c$ ), debido a que esta información es necesaria para obtener el sentido de giro del robot en cada punto. Los resultados son enviados a partes diferentes del algoritmo, esto debido que los caminos 1 a 4 hacen referencia al triángulo superior y las restantes al triángulo inferior. Ver figura 3.22.

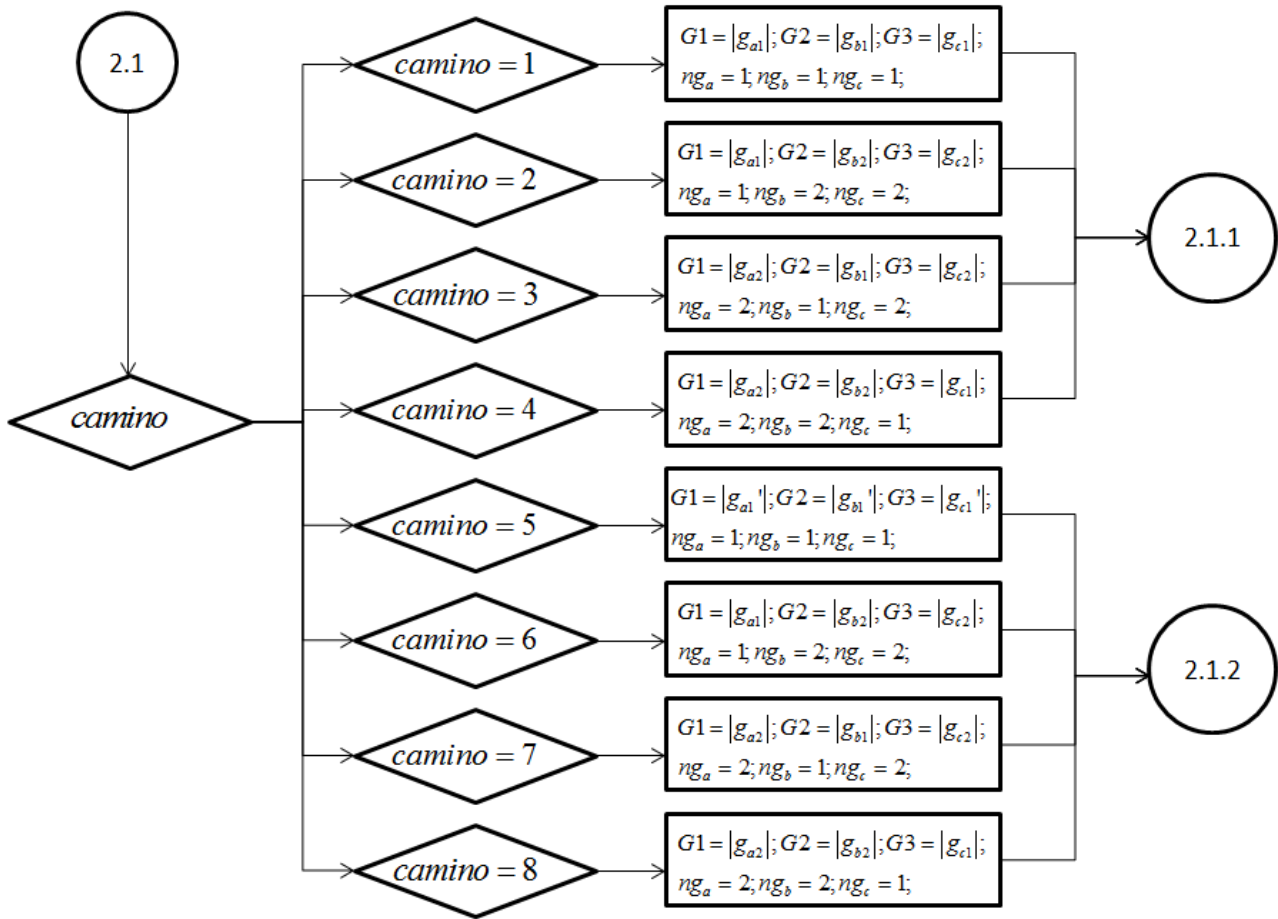


Figura 3.22. Sección 2.1 del Algoritmo de Trayectorias

### Sección 2.1.1

En esta parte se obtienen los sentidos de giro  $s1, s2, s3$  en los puntos  $a, b, c$ , respectivamente. De igual manera se obtiene el sentido de avance lineal ( $av_1$ ) que el robot debe llevar al recorrer el lado  $mR$  del triángulo. Nótese que se usó el mismo criterio que en la sección 1.1 del algoritmo. Para obtener el segundo avance lineal ( $av_2$ ) se utiliza un criterio un poco más complejo usando además la variable auxiliar  $ng_b$ , Ver figura 3.23. Finalmente se envían los resultados para realizar la animación.

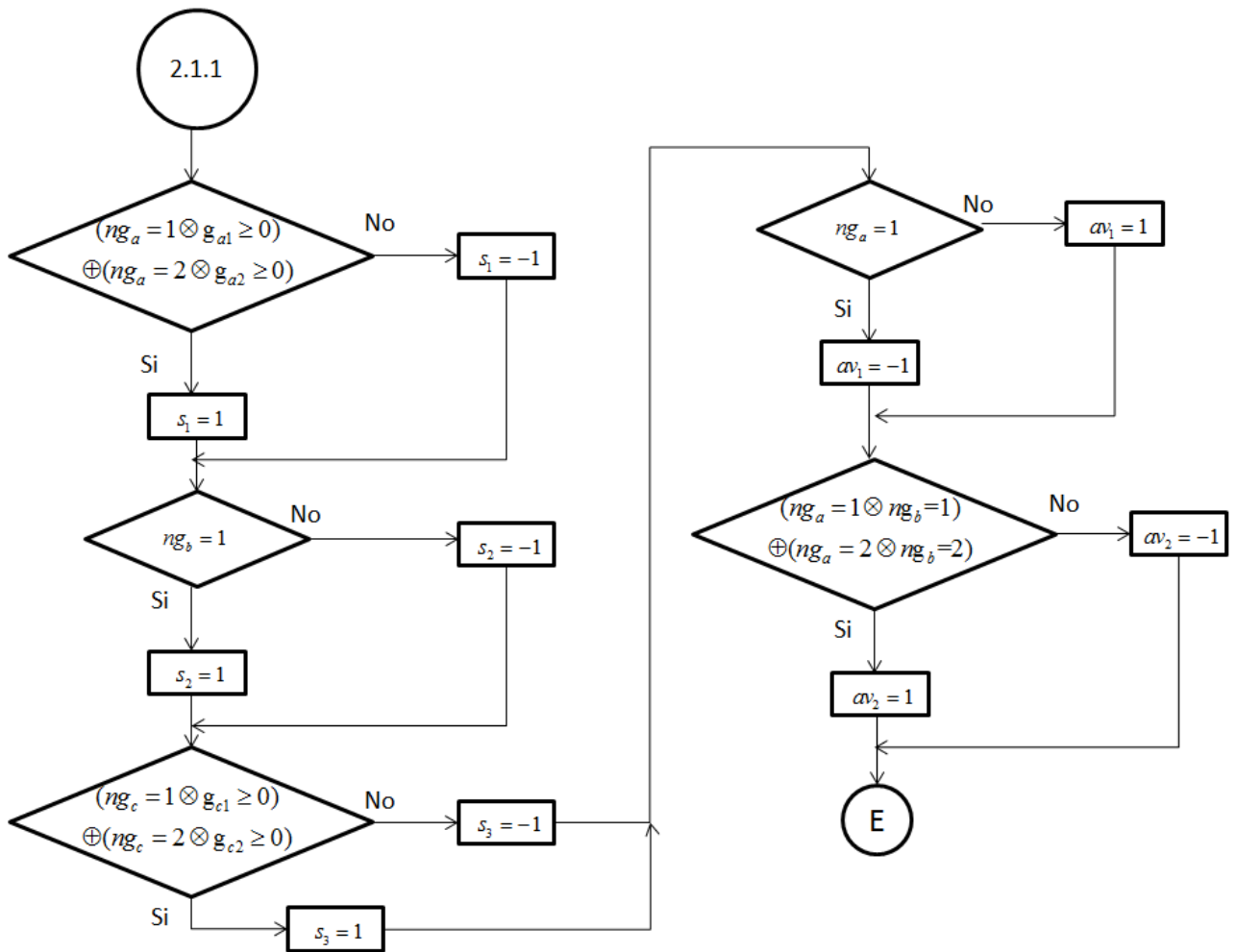


Figura 3.23. Sección 2.1.1 del Algoritmo de Trayectorias

### Sección 2.1.2

Esta sección realiza el mismo tipo de análisis que la sección 2.1.1, únicamente cambian las condiciones para elegir los sentidos de giro y de avance a lo largo de la trayectoria. Ver figura 3.24.

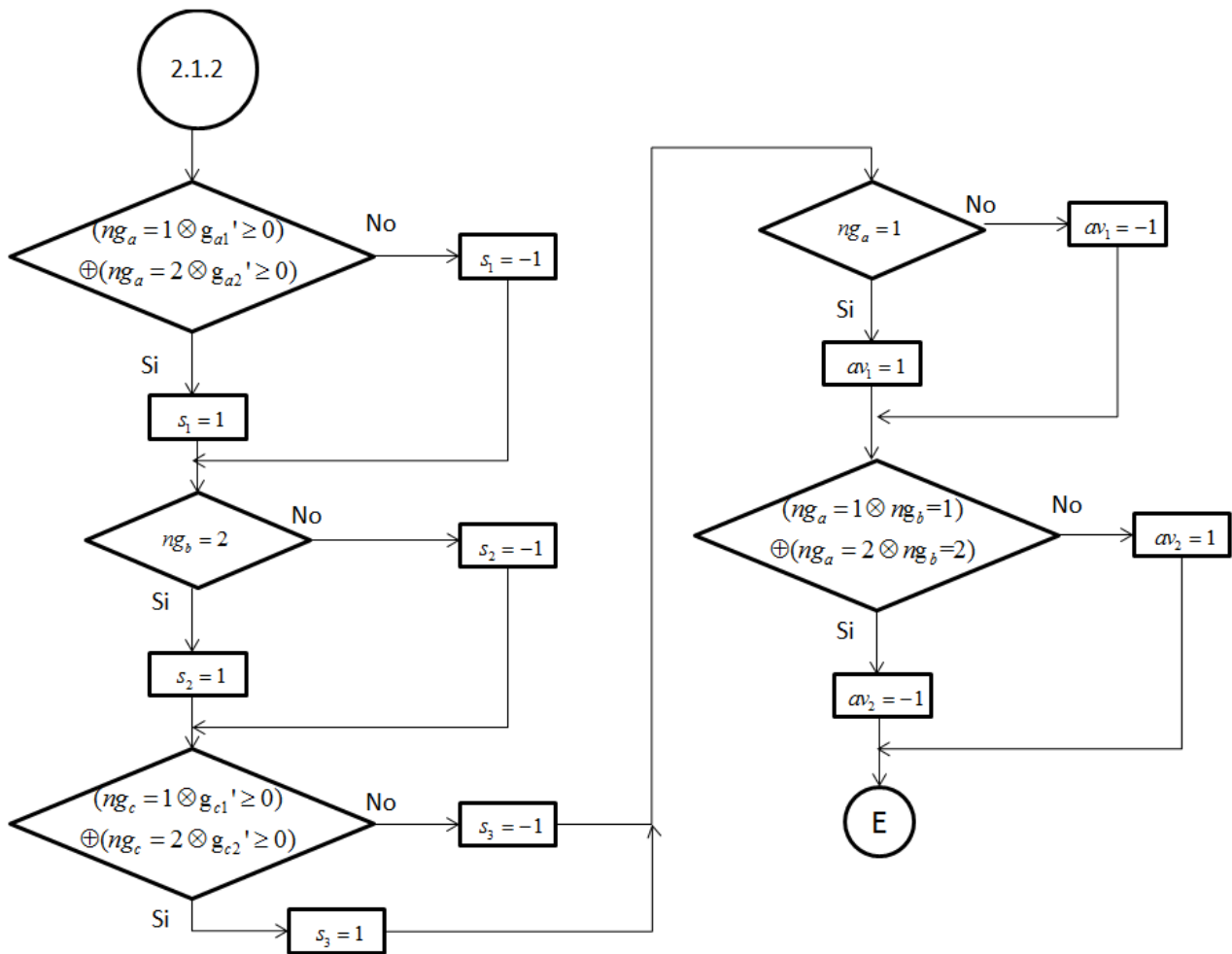


Figura 3.24. Sección 2.1.2 del Algoritmo de Trayectorias

### Sección 3

La sección 3 (junto con sus subsecciones) se encarga de resolver el problema de *larga distancia*. En el siguiente diagrama se calcula el número de pasos necesarios  $n$  para llegar de un punto a otro. Se inicializa la variable  $minimo_{global}$ , la cual se actualizará cada vez que el algoritmo encuentre un camino de menor magnitud que la variable misma. Se inicializa con el valor de  $540^0$  debido a que la máxima magnitud que puede alcanzar cada giro es  $180^0$ . Ver figura 3.25.

La primera condición presente en el diagrama ( $j = n - 1$ ), hace referencia al número de parejas de triángulos (superior e inferior) que existen dependiendo de  $n$ . La forma de calcular los ángulos interiores de los triángulos  $\alpha_1, \alpha_2, \alpha_3$ ; los ángulos de los posibles giros; y las funciones  $f_1, \dots, f_8$ , es la misma que en el problema de corta distancia. Sin embargo, en este caso los valores para  $m$  y  $l$  van



cambiando en forma gradual, dependiendo del valor con que vaya tomando  $j$  a lo largo del ciclo *for*. En forma anidada se encuentra el mismo ciclo que se utilizó para búsqueda de camino óptimo en el problema de *corta distancia*. Notar que el ciclo anidado se repetirá hasta  $j$  alcance el valor de  $n - 1$ .

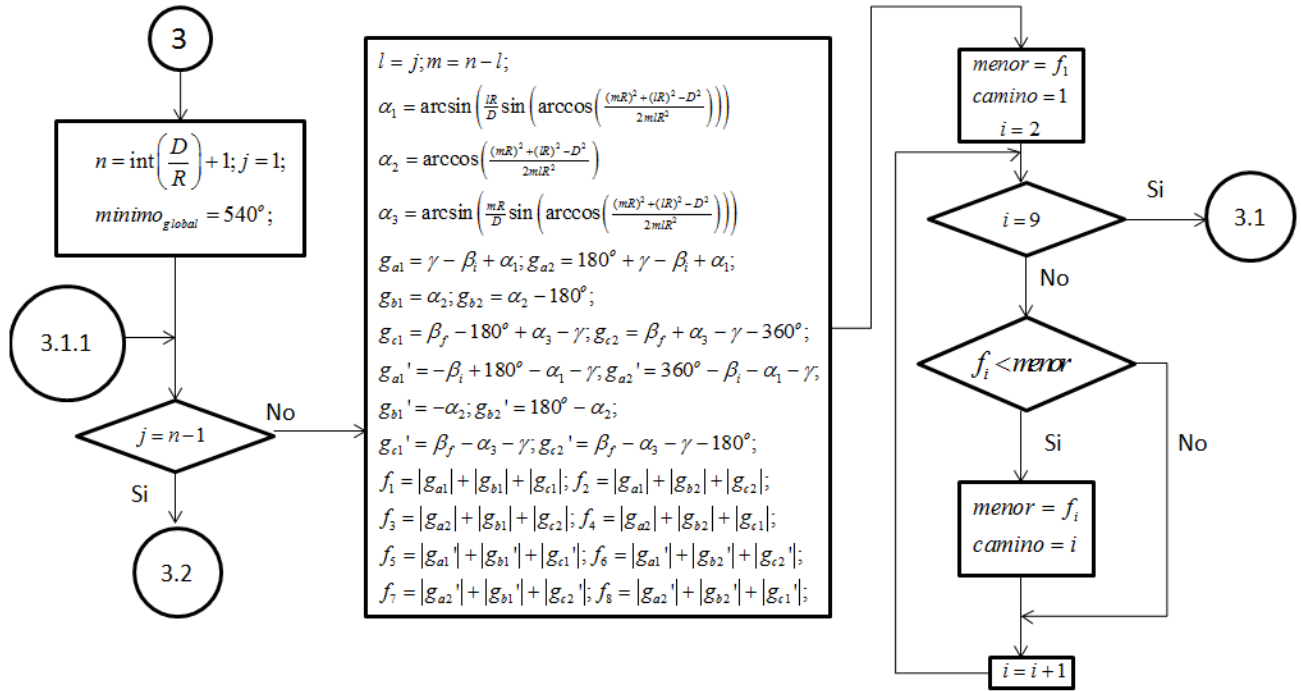


Figura 3.25. Sección 3 del Algoritmo de Trayectorias

### Sección 3.1

En esta parte se va actualizando la variable  $minimo_{global}$  con la magnitud de la función que representa la menor de todas las evaluadas hasta ese momento. Cada vez que se encuentre un valor mínimo respecto a los anteriores de  $f$ , se actualiza de igual manera la variable  $l_{optima}$ ; esta variable guarda el valor de  $l$  donde apareció la mínima magnitud de giros. Por último, se incrementa la variable  $j$  en +1 para regresar a la condición  $j = n - 1$  (Etiqueta 3.1.1). Ver figura 3.26.

### Sección 3.2

Una vez encontrado el valor de  $l$  donde se presenta la menor magnitud de giro, se evalúa de nueva cuenta los 8 caminos posibles y se obtiene el menor, para este caso resulta el menor global de esta trayectoria. Ver figura 3.26.

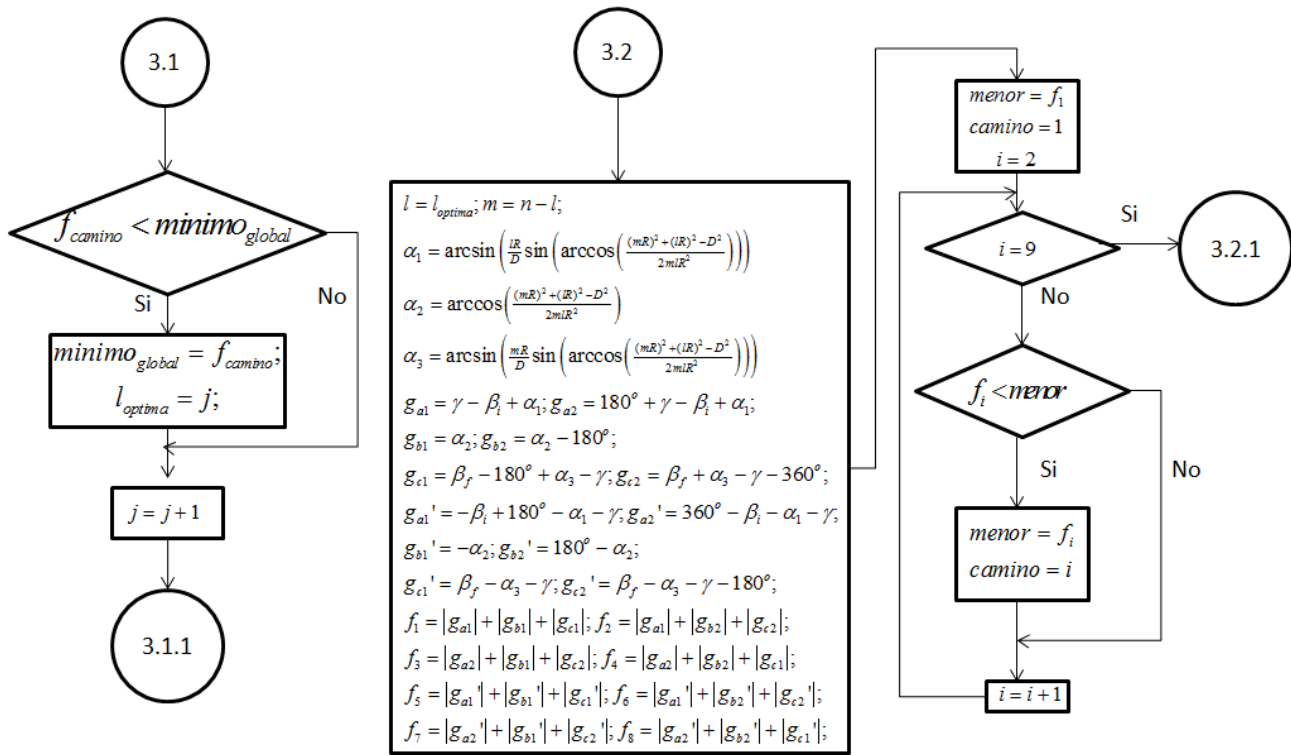


Figura 3.26. Secciones 3.1 y 3.2 del Algoritmo de Trayectorias

### Sección 3.2.1

A partir de aquí la forma de calcular las magnitudes y sentidos de giro es la misma (incluyendo las condiciones) que en el caso de *corta distancia*. De igual manera, las condiciones que determinan los sentidos de avance son las mismas que en el caso mencionado.

Notar la igualdad entre las secciones: 2.1 y 3.2.1 (figura 3.27); 2.1.1 y 3.2.1.1 (figura 3.28); 2.1.2 y 3.2.1.2 (figura 3.29).

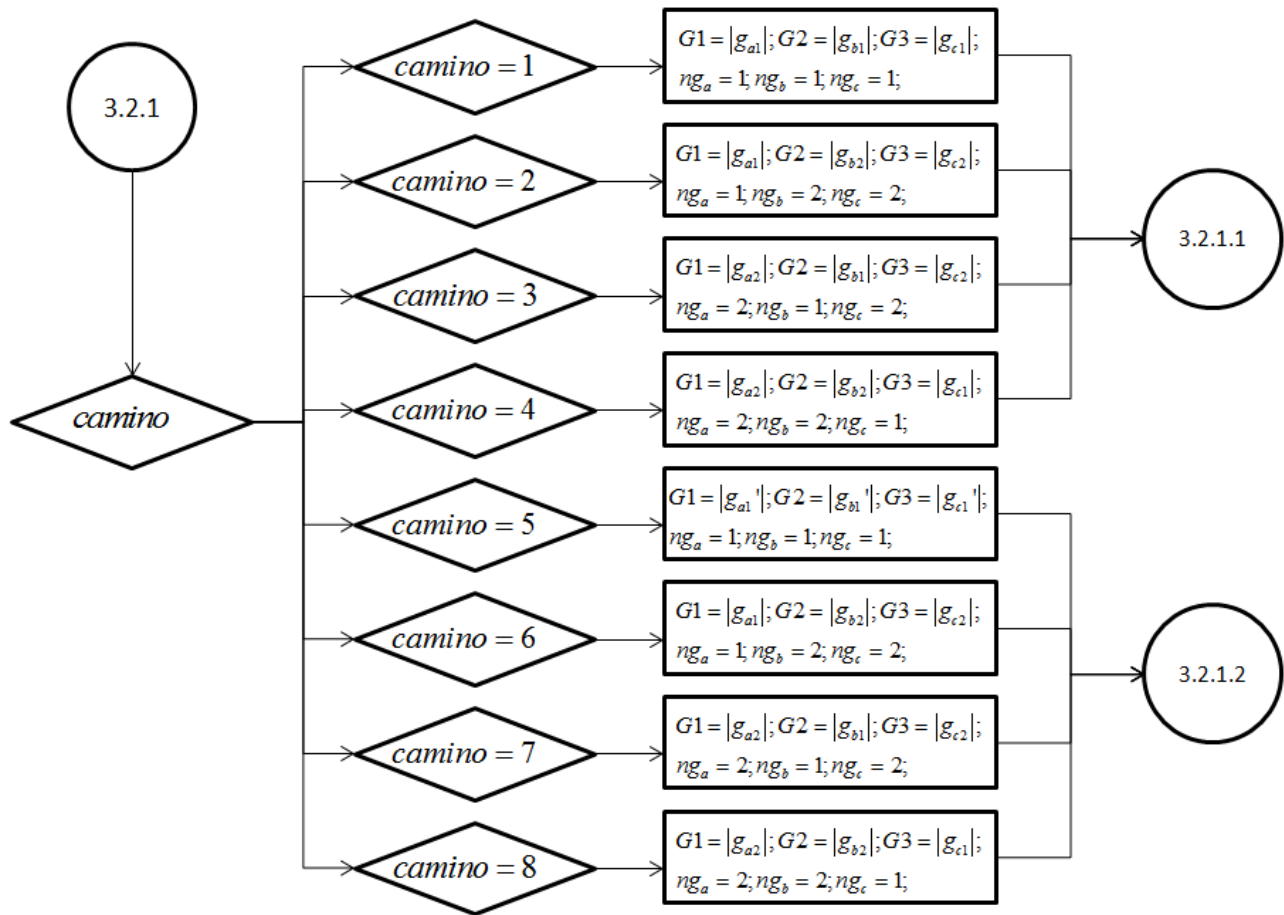


Figura 3.27. Sección 3.2.1 del Algoritmo de Trayectorias

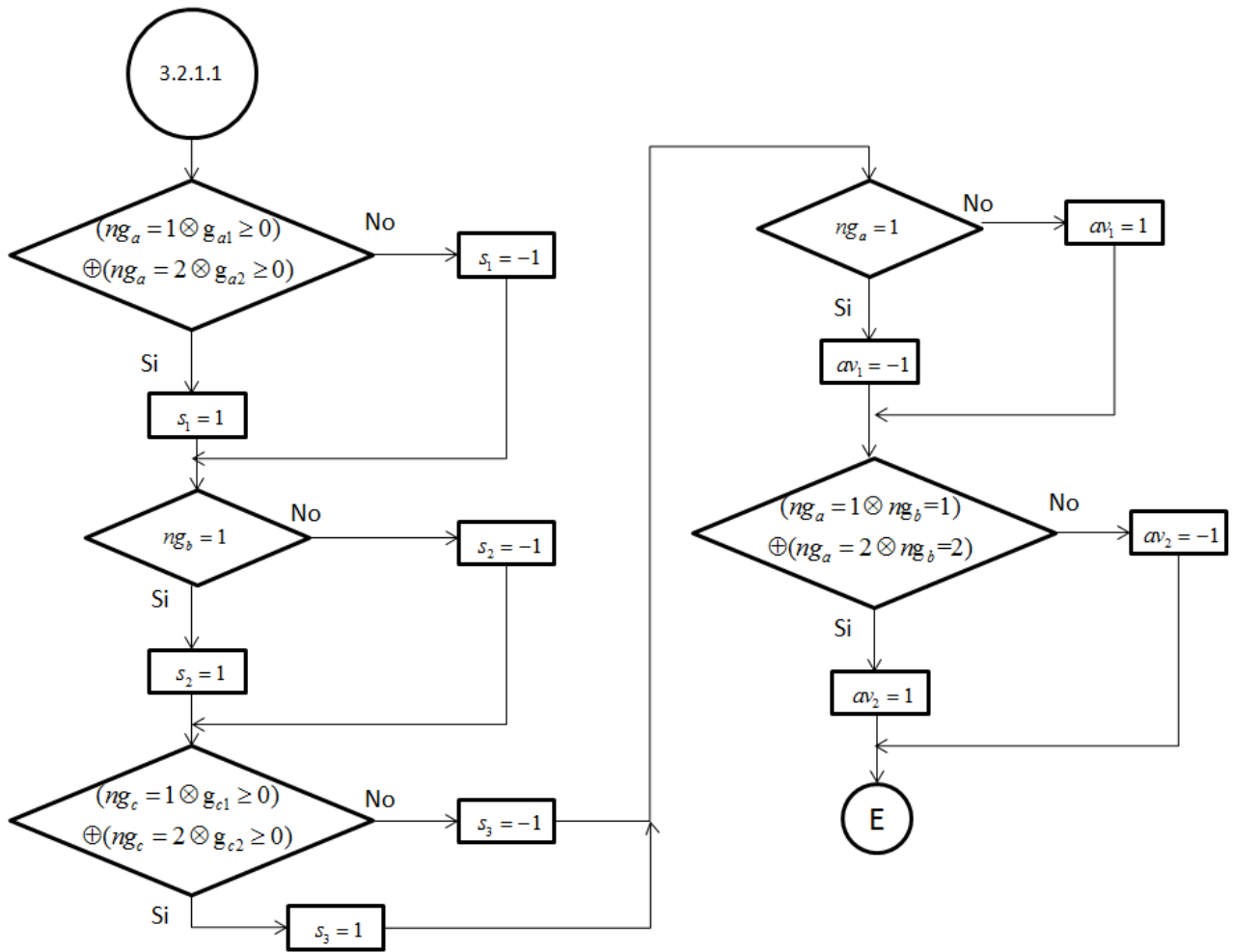


Figura 3.28. Sección 3.2.1.1 del Algoritmo de Trayectorias

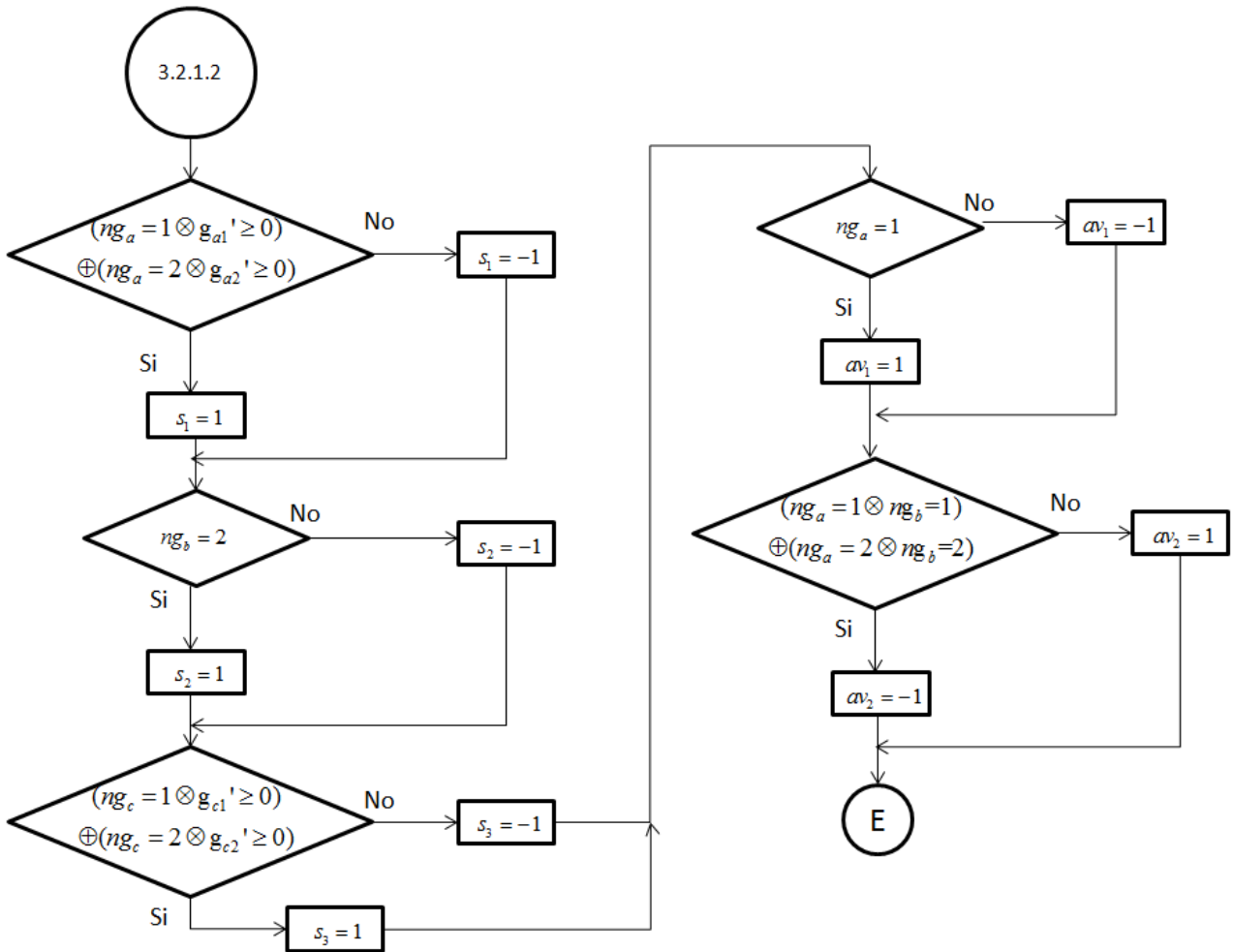


Figura 3.29. Sección 3.2.1.2 del Algoritmo de Trayectorias

### 3.2.1.1 Animación

La animación consta de 5 etapas que consisten en ciclos *for*. Cada etapa recibe distintos valores generados por el algoritmo de seguimiento antes visto.

La **primera etapa** recibe el valor de las variables  $\beta_i, G1, s_1, x_i, y_i$ . Con los cuales se realiza una interpolación entre los valores  $\beta_i$  y  $s_1 G1$  con incrementos de  $G1/10$ , para después graficar de manera secuencial los puntos y así obtener una animación. Esta etapa se encarga de animar el primer giro a realizar en la trayectoria. En caso que no se requiera un giro al comienzo de la trayectoria ( $G1 = 0$ ), no se ejecuta esta instrucción debido a las condiciones declaradas en el ciclo, *for*  $inc = 0: G1/10 : G1$ ; de tal modo que si  $G1 = 0$  el ciclo no se ejecuta ni una sola vez.

La **segunda etapa** se encarga del avance que debe realizar el robot para llegar del punto inicial al final en el caso de distancia entera. Para los casos de corta o larga distancia, este avance se refiere al lado de magnitud  $mR$  del triángulo que el robot debe recorrer. De igual manera que en la etapa anterior, la animación de avance se realiza con una interpolación entre los puntos  $a$  y  $b$  de la trayectoria. Los datos que recibe esta etapa son  $m, R, av_1, \beta_i + s_1G1, x_i, y_i$ , los datos que hereda a la siguiente etapa son los que describen la posición y orientación con la que termina el robot después del primer avance lineal:  $x_i + av_1mR\cos(\beta_i + s_1G1)$ ,  $y_i + av_1mR\sin(\beta_i + s_1G1)$  y  $\beta_i + s_1G1$  (los dos primeros describen la posición y el último la orientación).

**Tercera etapa.** En el caso de distancia entera esta es la última etapa, puesto que una vez posicionado en el punto  $b$  sólo resta orientarse con  $\beta_f$ . En el caso de las trayectorias triangulares, esta etapa sirve para orientar al robot de manera que llegue en línea recta del punto  $b$  al punto  $c$ . En esta tercera etapa el robot únicamente modifica su orientación a  $\beta_i + s_1G1 + s_2G2$ . La animación se realiza de la misma forma que en la etapa 1.

La **cuarta etapa** (esta etapa sólo existe en los casos de corta y larga distancia) consiste en animar el recorrido que realiza el robot del punto  $b$  al punto  $c$ . Para la animación se realiza una interpolación entre el punto  $b$  y el punto  $c$ . Finalmente el robot alcanzara la posición  $[x_i + av_1mR\cos(\beta_i + s_1G1) + av_2lR\cos(\beta_i + s_1G1 + s_2G2), y_i + av_1mR\sin(\beta_i + s_1G1) + av_2lR\sin(\beta_i + s_1G1 + s_2G2)]$  y su orientación estará dada por  $\beta_i + s_1G1 + s_2G2$ .

Finalmente, en la **quinta etapa** el robot sólo debe modificar su orientación (de no ser necesario, el robot conserva la orientación de la etapa anterior). Al igual que en las etapas 1 y 3, se realiza un incremento en la orientación del robot de  $s_3G3$  grados. Por lo que la orientación final del robot está dada por  $\beta_i + s_1G1 + s_2G2 + s_3G3$ , lo cual es equivalente a  $\beta_f$ .

El código desarrollado para el simulador se encuentra en el Anexo C de este documento. La principal finalidad de desarrollar el simulador anterior, es la de validar el algoritmo de seguimiento de trayectorias de manera numérica y gráfica.

La interfaz para introducir los valores de posición inicial y final deseados, así como las orientaciones respectivas en cada uno de estos puntos, se encuentran en el encabezado del código del simulador.

## ***Discusión***

En este capítulo se obtuvieron las trayectorias primitivas del robot en base al modelo cinemático del mismo. Estas trayectorias se componen básicamente de pasos constantes y giros de magnitud y sentido no arbitrarios. Debido a los efectos no considerados en el modelo cinemático, es posible que existan variaciones en el desplazamiento y orientación al momento de probar dichas trayectorias en

un prototipo real. Sin embargo, es posible lidiar con estos efectos implementando un control en lazo cerrado que logre minimizar el error en cada desplazamiento y cambio de orientación del robot.

De los criterios de optimización que se presentaron en este capítulo (mínima energía, tiempo mínimo y distancia mínima), el criterio seleccionado para la planeación de trayectorias fue el de distancia mínima. Esto debido a que la aplicación principal del robot es la de desplazamiento, sin importar, por el momento, el consumo energético o de tiempo. Además, con el criterio elegido es posible tratar en forma desacoplada la minimización de distancia y giros, obteniendo así una solución que no demanda cálculos más complejos que los involucrados en el caso de energía o tiempo.

En base a las ecuaciones obtenidas de la planeación de trayectorias, se desarrolló un algoritmo de seguimiento de trayectorias. Dicho algoritmo fue implementado en una plataforma de simulación, la cual fue desarrollada para validar los resultados de la planeación de trayectorias. Los resultados obtenidos de la plataforma de simulación se presentan en el capítulo siguiente.





# Capítulo 4. Resultados

La parte experimental en este trabajo consistió de realizar numerosas pruebas para diferentes situaciones en las cuales podría estar el robot. Por diferentes situaciones se hace referencia a diferentes condiciones iniciales (posición y orientación en el plano) y finales.

Debido a la falta de un prototipo real en cual realizar pruebas, se utilizó el simulador descrito en el capítulo anterior para observar los resultados del algoritmo de trayectorias.

Los resultados que se presentan a continuación muestran, de manera gráfica, la trayectoria seguida por el robot en diferentes casos; así como los valores de posición y orientación deseados en caso.

Para un mejor entendimiento de los resultados en este documento, se presenta la tabla 4.1, la cual describe la simbología en cada una de las trayectorias.







Símbolo	Descripción	Símbolo	Descripción
	Sistema de referencia inercial. Coordenadas (0,0)		Posición final del robot. Coordenadas $(x_f, y_f)$ . La orientación se mide respecto a la línea de mayor longitud.
	Cuerpo del robot. La intersección de las líneas indica el centro de masa		Indica el giro realizado en el punto correspondiente.
	Posición inicial del robot. Coordenadas $(x_i, y_i)$ . La orientación se mide respecto a la línea de mayor longitud.		Trayectoria seguida por el robot.

Tabla 4.1. Simbología utilizada en la representación de los resultados

Cabe mencionar que se realizó un número (en el orden de decenas) exhaustivo de pruebas en el simulador, sin embargo sólo se muestran los resultados obtenidos en algunas de ellas con motivo de no extender innecesariamente este documento. En todos los casos se consideró a la magnitud del paso  $R = 31$ .

## 4.1 Resultados en Caso de Distancia Entera

Hay que recordar que existe un caso particular dentro del caso de distancia entera; este ocurre cuando la distancia  $D$  entre la posición inicial y final es igual a cero. A continuación se presentan un caso donde ocurre tal situación, figura 4.1.

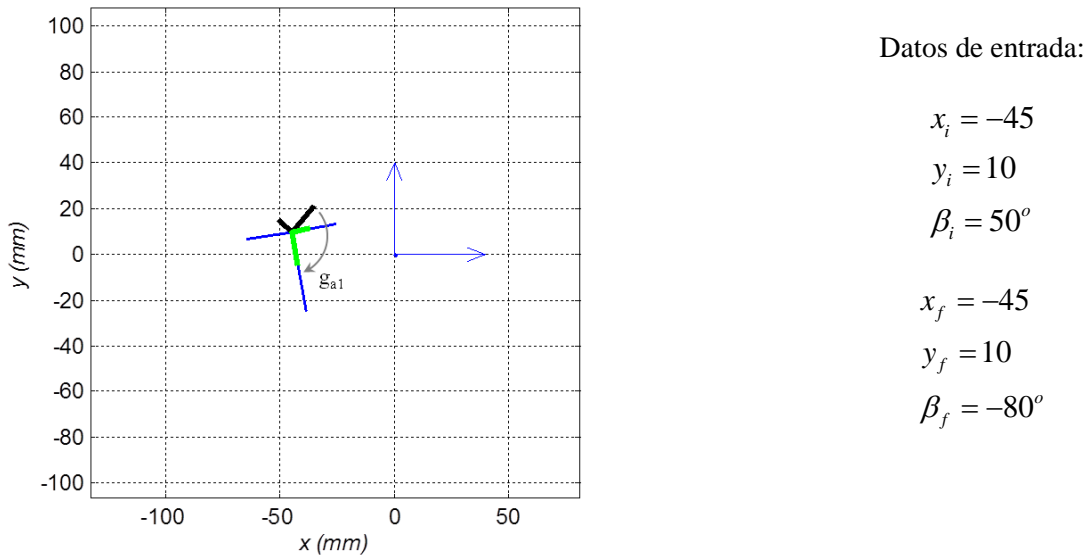


Figura 4.1. Caso  $D = 0$

La siguiente tabla 4.2 muestra los valores arrojados por el algoritmo de trayectorias para el caso anterior, donde  $f_i$  indica la magnitud de giro para cada combinación  $i$  posible (en grados), y  $\min(f_i)$  indica la combinación de menor magnitud.

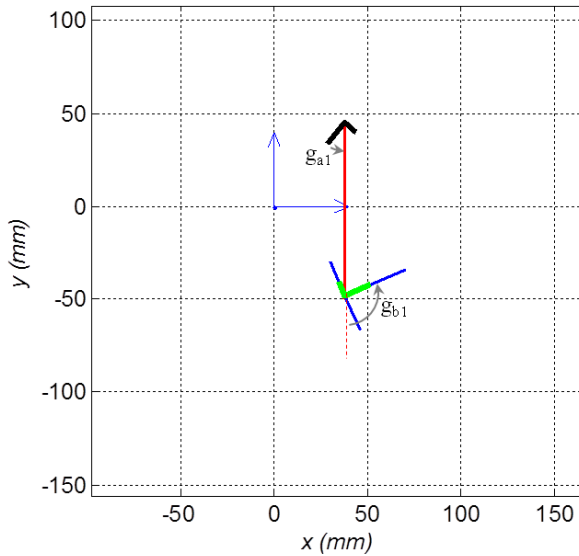
$D$	$n$	$m$	$l$	$f_i$	$\min(f_i)$
0	0	0	0	$f_1 = 130$ $f_2 = 230$	$f_1$

Tabla 4.2. Resultados en el ejemplo para  $D = 0$

Los valores en las cuatro primeras columnas indican que no debe existir ningún avance en línea recta por parte del robot, lo cual puede apreciar en la figura alusiva a este ejemplo. La combinación de menor magnitud resulta ser la número 1, la cual se compone únicamente de  $g_{a1}$ .

Como se observa en la figura anterior, la trayectoria realizada por el robot fue la sugerida por el algoritmo de seguimiento de trayectorias.

A continuación se presenta un caso donde la distancia  $D$  resulta ser múltiplo entero de  $R$  y diferente de cero, figura 4.2.



Datos de entrada:

$$\begin{aligned} x_i &= 38 \\ y_i &= 45 \\ \beta_i &= -130^\circ \\ \\ x_f &= 38 \\ y_f &= -48 \\ \beta_f &= 23^\circ \end{aligned}$$

Figura 4.2. Caso  $D$  múltiplo de  $R$  y diferente de cero

Tabla de valores para este caso:

$D$	$n$	$m$	$l$	$f_i$	$\min(f_i)$
93	3	3	0	$f_1 = 153$ $f_2 = 207$	$f_1$

Tabla 4.3. Resultados en el ejemplo para  $D \neq 0$

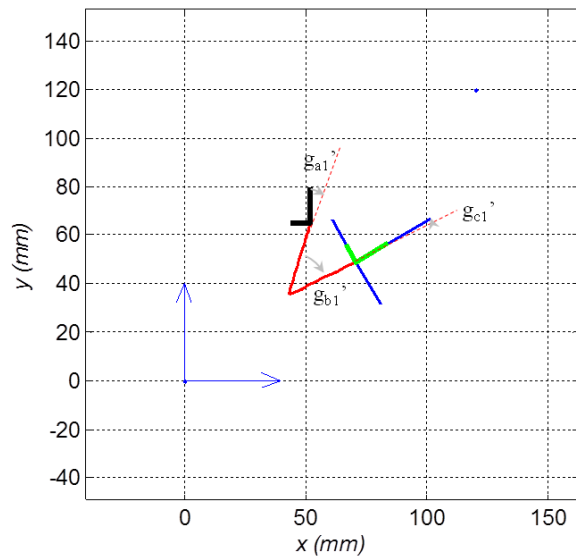
La tabla 4.3 sugiere que en este caso debe existir un avance en línea recta con una magnitud de  $3R$ ; en la figura anterior se puede apreciar que la distancia recorrida por el robot es de 93 milímetros. Ambos giros realizados en la trayectoria pertenecen a la combinación 1 para el caso de distancia entera.

En los dos casos presentados en esta sección, se observa gráficamente que el algoritmo de seguimiento de trayectorias genera resultados que son confiables cuando se trata de un problema de *distancia entera*.

## 4.2 Resultados en el Caso de Corta Distancia

Ahora se presentan dos casos en donde la distancia  $D$  resulta ser mayor a 0 y menor que  $R$ .

El primer caso se muestra en la figura 4.3.



Datos de entrada:

$$x_i = 52$$

$$y_i = 65$$

$$\beta_i = 90^\circ$$

$$x_f = 71$$

$$y_f = 49$$

$$\beta_f = 30^\circ$$

Figura 4.3. Primer caso donde  $D < R$

Tabla de valores para este caso:

$D$	$n$	$m$	$l$	$f_i$	$\min(f_i)$
24.8395	2	1	1	$f_1 = 154.47$ $f_2 = 332.96$ $f_3 = 300.00$ $f_4 = 292.56$ $f_5 = 67.43$ $f_6 = 325.03$ $f_7 = 387.03$ $f_8 = 300.00$	$f_5$

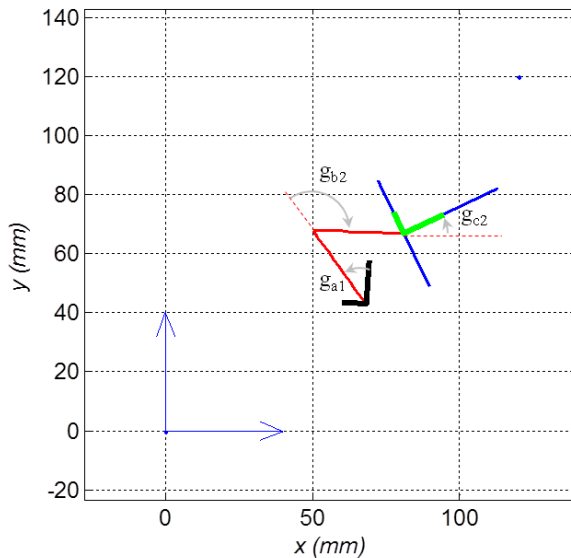
Tabla 4.4. Resultados en el primer ejemplo para  $D < R$

Notar que ahora  $l$  posee un valor diferente de cero, esto geométicamente indica que la trayectoria a seguir posee una forma triangular.

En este ejemplo se puede apreciar la importancia de haber considerado ambos triángulos en la planeación de trayectorias. La combinación número 5 pertenece al triángulo inferior y notablemente posee una magnitud menor a las demás combinaciones.

Cabe hacer la observación que en la tabla anterior existen funciones que presentan valores iguales entre sí ( $f_3$  y  $f_8$ ). Este fenómeno se presenta repetidamente en diferentes ocasiones. Sin embargo, la relación no siempre es la misma; es decir, para diferentes valores de entrada,  $f_3$  podría presentar el mismo valor que  $f_7$  o cualquier otra de las 6 funciones restantes, o simplemente poseer un valor que no se repite en ninguna otra función. Es por ello que este fenómeno no es tomado en cuenta para descartar ninguna de las funciones.

Un segundo ejemplo donde se resuelve un problema de corta distancia es el que se ilustra en la figura 4.4.



Datos de entrada:

$$x_i = 68$$

$$y_i = 43$$

$$\beta_i = 86^\circ$$

$$x_f = 81$$

$$y_f = 67$$

$$\beta_f = 26^\circ$$

Figura 4.4. Segundo caso donde  $D < R$

Tabla de valores para este caso:

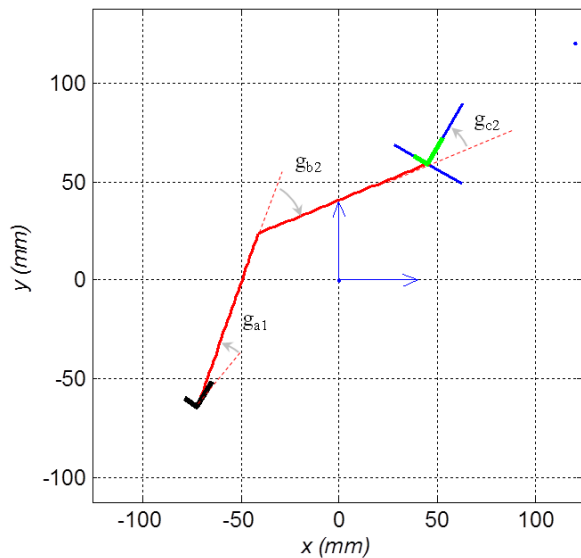
$D$	$n$	$m$	$l$	$f_i$	$\min(f_i)$
27.2947	2	1	1	$f_1 = 243.35$ $f_2 = 195.52$ $f_3 = 221.12$ $f_4 = 420.00$ $f_5 = 243.35$ $f_6 = 300.00$ $f_7 = 221.12$ $f_8 = 315.52$	$f_2$

Tabla 4.5. Resultados en el segundo ejemplo para  $D < R$

En este ejemplo el robot recorre la trayectoria descrita por el triángulo superior, puesto que  $f_2$  pertenece a dicho triángulo A diferencia del caso anterior, no existe una combinación de giros con una magnitud tan pequeña. Sin embargo, sigue existiendo una gran diferencia entre el mínimo y el máximo  $f_i$ .

### 4.3 Resultados en el Caso de Larga Distancia

En esta sección se presentan dos ejemplos para el caso de larga distancia. Inicialmente se describen los resultados del primer ejemplo en la figura 4.5.



Datos de entrada:

$$x_i = -72$$

$$y_i = -64$$

$$\beta_i = 60^\circ$$

$$x_f = 45$$

$$y_f = 59$$

$$\beta_f = 60^\circ$$

Figura 4.5. Ejemplo 1 en el caso de larga distancia

Los resultados en cualquier caso de larga distancia son más extensos que los resultados en los casos de corta distancia y distancia entera. Esto debido a que se evaluando las 8 combinaciones posibles con valores distintos de  $m$  y  $l$ ;  $m$  es evaluado desde 1 hasta  $n - 1$  y  $l = n - m$ . La tabla 4.6 contiene los valores para este ejemplo.

El valor de  $m$  donde se obtiene la menor magnitud de giro es  $m = 3$ , y  $f_2$  es la función que posee tal magnitud en este caso. Por lo tanto el robot debe seguir la combinación de giros número 2, como se puede apreciar en la figura alusiva a este ejemplo.

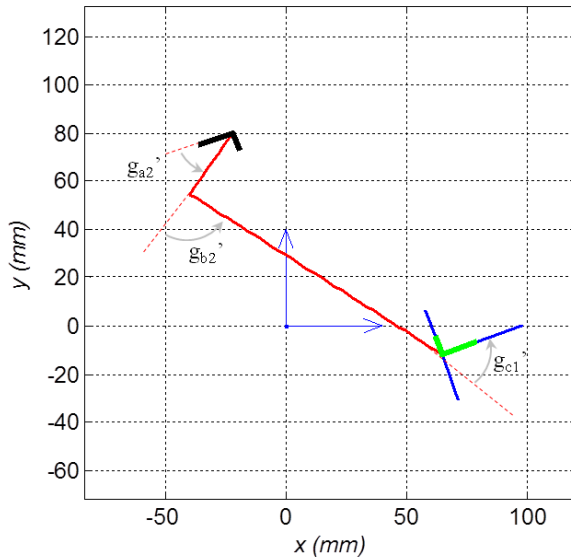
En la tabla 4.5 se observa que no para todos los valores de  $m$ ,  $f_2$  resulta ser la función de menor magnitud; por lo que no es posible asegurar que la función de menor magnitud para un valor cualquiera de  $m$  y  $l$  será la menor en todos los valores restantes

$D$	$n$	$m$	$l$	$f_i$	$\min(f_i)$
169.7587	6	1	5	$f_1 = 316.36$ $f_2 = 105.97$ $f_3 = 297.38$ $f_4 = 360.00$ $f_5 = 254.02$ $f_6 = 349.08$ $f_7 = 360.00$ $f_8 = 116.88$	$f_2$
169.7587	6	2	4	$f_1 = 305.88$ $f_2 = 82.60$ $f_3 = 331.5$ $f_4 = 360.00$ $f_5 = 277.39$ $f_6 = 359.84$ $f_7 = 360.00$ $f_8 = 82.75$	$f_2$
169.7587	6	3	3	$f_1 = 294.02$ $f_2 = 77.68$ $f_3 = 348.29$ $f_4 = 360.00$ $f_5 = 294.02$ $f_6 = 360.00$ $f_7 = 348.29$ $f_8 = 77.68$	$f_2$
169.7587	6	4	2	$f_1 = 277.39$ $f_2 = 82.75$ $f_3 = 360.00$ $f_4 = 359.84$ $f_5 = 305.88$ $f_6 = 360.00$ $f_7 = 331.51$ $f_8 = 82.60$	$f_8$
169.7587	6	5	1	$f_1 = 254.02$ $f_2 = 116.88$ $f_3 = 360.00$ $f_4 = 349.08$ $f_5 = 316.36$ $f_6 = 360.00$ $f_7 = 297.38$ $f_8 = 105.97$	$f_8$

Tabla 4.6. Resultados en el primer ejemplo para  $D > R$



Otro ejemplo para el caso de larga distancia se observa en la figura 4.6. En este caso la trayectoria que sigue el robot es la descrita por el triángulo inferior. La tabla 4.7 contiene los diferentes valores obtenidos en este ejemplo.



Datos de entrada:

$$\begin{aligned}
 x_i &= -22 \\
 y_i &= 80 \\
 \beta_i &= -160^\circ \\
 x_f &= 65 \\
 y_f &= -12 \\
 \beta_f &= 20^\circ
 \end{aligned}$$

Figura 4.6. Ejemplo 2 en el caso de larga distancia

Es posible notar la repetitividad el valor de 180 para distintas funciones de la tabla 4.7, el cual no sólo se repite en dos funciones (no siempre las mismas) dentro de las 8 para un valor de  $m$ ; también se repite en todos los casos de  $m$ . Inesperadamente, 180 es el valor mínimo que puede lograrse al minimizar la magnitud de giro en este ejemplo. Esto puede indicar que existe alguna relación entre las 8 combinaciones dependiendo de los parámetros de entrada. Sin embargo, en este trabajo no fue posible encontrar dicha relación.

El algoritmo de trayectorias indicó que el mínimo de giro se logra con la combinación de giros número 8 (ver figura 4.6), con  $f_8 = 180.00$  y  $m = 1$ . Es verdad que no sólo bajo estas circunstancias ocurre que la magnitud de giro es igual a 180, pero hay considerar que los resultados en la tabla 4.7 se redondean a 2 cifras después del punto y no se muestran en su totalidad. En cambio, el cálculo, a nivel de procesamiento de datos, implica cifras más allá de este redondeo y son las que hacen la distinción entre un valor y otro de  $f_i$ .

$D$	$n$	$m$	$l$	$f_i$	$\min(f_i)$
126.6215	5	1	4	$f_1 = 355.46$ $f_2 = 341.52$ $f_3 = 180.00$ $f_4 = 203.01$ $f_5 = 284.87$ $f_6 = 364.53$ $f_7 = 250.58$ $f_8 = 180.00$	$f_8$
126.6215	5	2	3	$f_1 = 351.24$ $f_2 = 324.24$ $f_3 = 224.50$ $f_4 = 180.00$ $f_5 = 257.64$ $f_6 = 324.24$ $f_7 = 318.10$ $f_8 = 180.00$	$f_4$
126.6215	5	3	2	$f_1 = 318.10$ $f_2 = 324.24$ $f_3 = 257.64$ $f_4 = 180.00$ $f_5 = 224.50$ $f_6 = 324.24$ $f_7 = 351.24$ $f_8 = 180.00$	$f_4$
126.6215	5	4	1	$f_1 = 250.58$ $f_2 = 364.53$ $f_3 = 284.87$ $f_4 = 180.00$ $f_5 = 180.00$ $f_6 = 341.52$ $f_7 = 355.46$ $f_8 = 203.01$	$f_4$

Tabla 4.7. Resultados en el segundo ejemplo para  $D > R$

### ***Discusión***

A lo largo de este capítulo se presentaron diferentes problemas para cada una de las situaciones en las que puede estar el robot; larga distancia, corta distancia y distancia entera. Los resultados obtenidos en estos problemas, y en los que no fueron incluidos en este documento, son satisfactorios. La distancia recorrida en cada una de las trayectorias fue la mínima necesaria para alcanzar el punto de destino del robot y la magnitud de giro no alcanza a ser mayor o igual que  $360^\circ$ .

En los resultados anteriores se observa que el robot no presenta errores al seguir las trayectorias. Esto se debe a que, para simulación, se considera un desplazamiento ideal del robot, además de una superficie completamente lisa. Sin embargo, es de esperarse que en un prototipo real se hagan presentes diversos factores que afecten la precisión en el seguimiento de trayectorias. De aquí la importancia de implementar un control retroalimentado en prototipos con similares características de desplazamiento.



# Conclusiones

En este trabajo se obtuvo el modelo cinemático de un robot móvil de dos grados de libertad con patas el cual, a pesar de su simplicidad, es capaz de llegar de un lugar a otro cualquiera en su espacio de trabajo mediante una serie de maniobras.

El modelo cinemático que se obtuvo en primera instancia representa muy bien el comportamiento del robot. Sin embargo, resultó ser altamente no lineal y un tanto complicado para manejar análisis posteriores de sus propiedades (p.e. controlabilidad). Una aportación del presente trabajo consistió en obtener una función que aproxima bastante bien una buena parte de las no linealidades del sistema. El modelo simplificado tiene las características de representar adecuadamente el comportamiento del sistema y una simplicidad que lo hace manejable para estudios complementarios. Ambos modelos cinemáticos pueden aplicarse a una clase de robots con patas que sean simples en su construcción y que por ello tienen ventajas para algunas aplicaciones.

El tipo de movimientos que el robot es capaz de realizar no es común en otros robots de su tipo, de aquí se desprende la segunda parte de este trabajo. Una vez obtenidas las ecuaciones que modelan el comportamiento del robot sobre una superficie, se determinan las trayectorias primitivas propias del mismo. Estas trayectorias primitivas hacen visible lo que matemáticamente sugiere el modelo cinemático.

A partir de las trayectorias primitivas, se desarrolla una propuesta para la planeación de trayectorias, en donde se considera un espacio de trabajo libre de obstáculos. En la planeación de trayectorias se propone como criterio de optimización minimizar la distancia que debe recorrer el robot para llegar de un punto a otro. Debido a que el tipo de desplazamiento del robot se considera discreto, no fue posible realizar la optimización por métodos convencionales y se recurrió a métodos numéricos de programación entera. Llevada a cabo la minimización de distancia, se llegó a la conclusión que los giros involucrados en las trayectorias también demandaban movimiento por parte del robot. Por lo que se realizó un análisis para la minimización de giro total a lo largo de la trayectoria. Este análisis

resultó ser más complejo que el realizado para minimizar la distancia. De igual manera se tuvo que recurrir a métodos numéricos para obtener la menor cantidad de giro posible.

Las trayectorias propuestas en este trabajo resultan ser trayectorias simples, que no demandan gran costo computacional y que son sencillas de aplicar para cualquier otro robot móvil que presente un desplazamiento similar al robot de este trabajo.

Para la validación de las trayectorias propuestas fue necesario desarrollar un programa de simulación por computadora. Este programa de simulación está basado en el algoritmo de generación de trayectorias y realiza la animación de la trayectoria solicitada.

Existen algunas limitantes en cuanto al trabajo desarrollado. Una de estas limitantes, es que en el modelo cinemático no fueron incluidos aquellos efectos que pueden generar errores en el seguimiento de las trayectorias, p. e. derrape. Sin embargo, es posible minimizar este error implementando un control en lazo cerrado para el robot prototipo. Como segunda limitante, se tiene que las trayectorias son calculadas respecto a un sistema inercial. Lo anterior implica que el robot debe conocer su posición y orientación al menos en dos puntos (inicial y final); esto se traduce en que se debe proveer al robot de aditamentos que le permitan conocer su posición y orientación respecto a un sistema inercial, p. e. GPS.

### ***Trabajo a futuro***

Dentro de las acciones para extender este estudio están contempladas las siguientes:

- Estudio de las propiedades de controlabilidad del modelo. En este trabajo se diseñó lo que puede considerarse como un control en lazo abierto. Sin embargo, es importante el cálculo del error que se genera a lo largo de las trayectorias, especialmente si se pretende realizar un prototipo del robot.
- Estudio de las restricciones no-holónomas del robot para complementar el estudio cinemático. Dicho estudio representa la base para el diseño de un control en lazo cerrado para el robot.
- Construcción de un prototipo de laboratorio. En forma paralela a este trabajo se está construyendo un prototipo con características muy similares al robot descrito en esta tesis. El prototipo está siendo desarrollado en el CICATA-IPN Unidad Querétaro.
- Validación del modelo cinemático y trayectorias en el prototipo. En este trabajo se realizó la validación de ambos conceptos en un simulador. Sin embargo, la simulación no considera el error que puede producirse entre las trayectorias ideales (robot en el simulador) y las reales (robot prototipo).
- Implementar un sistema de retroalimentación (mediante sensores de posición) en el prototipo para poder lidiar con los efectos que no fueron considerados en el modelado, p. e. derrape.

- Desarrollo de algoritmo de navegación para el robot. La planeación de trayectorias en este trabajo sugiere los movimientos que el robot puede realizar en un espacio libre de obstáculos. Un algoritmo de navegación debe ser capaz de guiar al robot en un ambiente con obstáculos y lograr una trayectoria libre de colisiones. Se pretende que dicha trayectoria sea desarrollada en base a las trayectorias propuestas en este trabajo.





# Anexo

## Anexo A

### *Código de simulación de las patas del robot*

```
clear all

%Medidas de los eslabones
A = 7.0;
B = 33.5;
D = 4.0;
E = 10.5;
F = 53.0;
R = 23.0;
L = 9.5;
M = 9.0;
N = 1.5;
P = 12.0;
psi = 10*pi/180;
alpha = pi;
S = 31.5;

for x=1:1:361
    th1(x)=x*pi/180; %tetal en radianes
end

for x=1:1:361

    %-----Resolviendo las ecuaciones de posición
    %Coeficientes para encontrar theta3
    coefall(x) = -2*L*N*sin(th1(x));
    coefb11(x) = -2*L*(N*cos(th1(x)) + P);
    coefc11(x) = 2*P*N*cos(th1(x)) + L^2 + P^2 + N^2 - M^2;

    %Encontramos theta2 y theta3
```

```

th3(x) = 2*atan((-coefa11(x) + sqrt(coefa11(x)^2+coefb11(x)^2-
    coefc11(x)^2))/(coefc11(x)-coefb11(x)));
th2(x) = 2*atan((L*sin(th3(x)) - N*sin(th1(x)))/(M + N*cos(th1(x)) -
    L*cos(th3(x)) + P));

%Coeficientes para encontrar theta5
coefa12(x) = -2*B*(L*sin(th3(x)) + R*sin(th2(x)) + E);
coefb12(x) = 2*B*(L*cos(th3(x)) - R*cos(th2(x)) + D);
coefc12(x) = 2*L*E*sin(th3(x)) + 2*L*D*cos(th3(x)) - 2*R*D*cos(th2(x)) +
    2*R*E*sin(th2(x)) - 2*L*R*cos(th2(x) + th3(x)) + L^2 + R^2 + B^2 + E^2
    + D^2 - A^2;

%Encontramos theta4 y theta5
th5(x) = 2*atan((-coefa12(x) - sqrt(coefa12(x)^2+coefb12(x)^2-
    coefc12(x)^2))/(coefc12(x)-coefb12(x)));
th4(x) = 2*atan((L*sin(th3(x)) + R*sin(th2(x)) - B*sin(th5(x)) + E)/(A
    + L*cos(th3(x)) - R*cos(th2(x)) + B*cos(th5(x)) + D));

%Para mostrar theta2 a theta5 en grados convertimos:
gth2(x) = th2(x)*180/pi;
gth3(x) = th3(x)*180/pi;
gth4(x) = th4(x)*180/pi;
gth5(x) = th5(x)*180/pi;

%El vector de posición del centro al efector de la pata trasera
rplx(x) = -N*sin(th1(x)) - (R+M)*sin(th2(x)) - F*sin(th4(x)-psi);
rply(x) = -N*cos(th1(x)) + (R+M)*cos(th2(x)) - F*cos(th4(x)-psi);

%---Para la pata delantera
ph1(x) = -th1(x) + alpha;

%-----Resolviendo las ecuaciones de posición
%Coeficientes para encontrar phi3
coefa21(x) = -2*L*N*sin(ph1(x));
coefb21(x) = -2*L*(N*cos(ph1(x)) + P);
coefc21(x) = 2*P*N*cos(ph1(x)) + L^2 + P^2 + N^2 - M^2;

%Encontramos phi2 y phi3
ph3(x) = 2*atan((-coefa21(x) + sqrt(coefa21(x)^2+coefb21(x)^2-
    coefc21(x)^2))/(coefc21(x)-coefb21(x)));
ph2(x) = 2*atan((L*sin(ph3(x)) - N*sin(ph1(x)))/(M + N*cos(ph1(x)) -
    L*cos(ph3(x)) + P));

%Coeficientes para encontrar phi5
coefa22(x) = -2*B*(L*sin(ph3(x)) + R*sin(ph2(x)) + E);
coefb22(x) = 2*B*(L*cos(ph3(x)) - R*cos(ph2(x)) + D);
coefc22(x) = 2*L*E*sin(ph3(x)) + 2*L*D*cos(ph3(x)) - 2*R*D*cos(ph2(x)) +
    2*R*E*sin(ph2(x)) - 2*L*R*cos(ph2(x) + ph3(x)) + L^2 + R^2 + B^2 + E^2
    + D^2 - A^2;

%Encontramos phi4 y phi5
ph5(x) = 2*atan((-coefa22(x) - sqrt(coefa22(x)^2+coefb22(x)^2-
    coefc22(x)^2))/(coefc22(x)-coefb22(x)));

```

```

ph4(x) = 2*atan((L*sin(ph3(x)) + R*sin(ph2(x)) - B*sin(ph5(x)) + E)/(A +
L*cos(ph3(x)) - R*cos(ph2(x)) + B*cos(ph5(x)) + D));

%Para mostrar phi2 a phi5 en grados convertimos:
gph2(x) = ph2(x)*180/pi;
gph3(x) = ph3(x)*180/pi;
gph4(x) = ph4(x)*180/pi;
gph5(x) = ph5(x)*180/pi;

%El vector de posición del centro al efector de la pata delantera
rp2x(x) = N*sin(ph1(x)) + (R+M)*sin(ph2(x)) + F*sin(ph4(x)-psi);
rp2y(x) = -N*cos(ph1(x)) + (R+M)*cos(ph2(x)) - F*cos(ph4(x)-psi);

end

x=1:361;

axes('fontsize',12)

for x=1:1:361

plot(-60,-50)
hold on
plot(60,50)
apoyo1=[-60 60]';
apoyo2=[-S -S]';
line(apoyo1,apoyo2,'LineWidth',2.5,'LineStyle','-.','Color','g');
daspect([1,1,1])

%PARA DIBUJAR LA PATA TRASERA
xa0 = 0;
xa1 = -N*sin(th1(x));
xa2 = xa1 - M*sin(th2(x));
xa3 = 0;
xa4 = xa2 - R*sin(th2(x));
xa5 = xa4 + A*sin(th4(x));
xa6 = E;
xa7 = xa4 - F*sin(th4(x)-psi);

ya0 = 0;
ya1 = -N*cos(th1(x));
ya2 = ya1 + M*cos(th2(x));
ya3 = P;
ya4 = ya2 + R*cos(th2(x));
ya5 = ya4 + A*cos(th4(x));
ya6 = D + P;
ya7 = ya4 - F*cos(th4(x)-psi);

Xa1 = [xa0 xa1 xa2 xa3]';
Ya1 = [ya0 ya1 ya2 ya3]';

Xa2 = [xa2 xa4 xa5 xa6]';

```

```

Ya2 = [ya2 ya4 ya5 ya6]';

Xa3 = [xa4 xa7]';
Ya3 = [ya4 ya7]';

%PARA LA PATA DELANTERA
xb0 = 0;
xb1 = N*sin(ph1(x));
xb2 = xb1 + M*sin(ph2(x));
xb3 = 0;
xb4 = xb2 + R*sin(ph2(x));
xb5 = xb4 - A*sin(ph4(x));
xb6 = -E;
xb7 = xb4 + F*sin(ph4(x)-psi);

yb0 = 0;
yb1 = -N*cos(ph1(x));
yb2 = yb1 + M*cos(ph2(x));
yb3 = P;
yb4 = yb2 + R*cos(ph2(x));
yb5 = yb4 + A*cos(ph4(x));
yb6 = D + P;
yb7 = yb4 - F*cos(ph4(x)-psi);

Xb1 = [xb0 xb1 xb2 xb3]';
Yb1 = [yb0 yb1 yb2 yb3]';

Xb2 = [xb2 xb4 xb5 xb6]';
Yb2 = [yb2 yb4 yb5 yb6]';

Xb3 = [xb4 xb7]';
Yb3 = [yb4 yb7]';

line(Xa1,Ya1,'LineWidth',2.5);
line(Xa2,Ya2,'LineWidth',2.5);
line(Xa3,Ya3,'LineWidth',2.5);

line(Xb1,Yb1,'LineWidth',2.5);
line(Xb2,Yb2,'LineWidth',2.5);
line(Xb3,Yb3,'LineWidth',2.5);

%Trayectoria de las patas
TXa(x)=xa7;
TYa(x)=ya7;
line(TXa',TYa','LineStyle','--','Color','r');

TXb(x)=xb7;
TYb(x)=yb7;
line(TXb',TYb','LineStyle','--','Color','r');
pause(.001)
hold off
end

```

# Anexo B

## *Código para el cálculo de los parámetros $K_1$ y $K_2$*

```
%En estas variables se guardan los valores generados en Simulink
V1=v1.signals(1).values;
T1=t1.signals(1).values;
thetalp=th1p.signals(1).values;

n=size(V1);      %Número de valores generados en Simulink
n=n(1);
D(:,1)=T1;
D(:,2)=V1;

for i=1:n
    if (D(i,2)~=0)
        F(i,:)=D(i,:);
    end
end

T1=F(:,1);
V1=F(:,2);

axes('fontsize',12)
hold on
plot(T1,V1,'+')
xlabel('Theta 1 (rad)')
ylabel('Velocidad de avance')
grid on

%Valores iniciales propuestos
C=[10 1]';

%Metodo de Minimos cuadrados
for j=1:20

    for i=1:n
        %Se evalúa la matriz de derivadas parciales
        A(i,:) = [thetalp(i)*sin(T1(i) - C(2)) -C(1)*thetalp(i)*cos(T1(i) -
            C(2))];
        fy(i,1) = C(1)*thetalp(i)*sin(T1(i)-C(2));
    end

%Calculo del error
DeltaY = V1 - fy;
DeltaC=inv(A'*A)*(A'*DeltaY);

%Actualización de valores
C=C+DeltaC;
```

```
C;  
end  
%Se generan los valores para la función propuesta, con los valores K1 y K2  
obtenidos  
for i=1:n  
m(i)= C(1)*theta1p(i)*sin(T1(i)-C(2));  
end  
  
plot(T1,m,'or')  
grid on  
clear all
```

# Anexo C

## *Código Simulador*

```
close all
clear all
clc

%%%POSICIÓN Y ORIENTACIÓN INICIAL
xi=-95;
yi=10;
b_i=30;    %Grados

%%%POSICIÓN Y ORIENTACIÓN FINAL
xf=20;
yf=10;
b_f=10;    %Grados

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R = 31;

b_i=atan2(sin(b_i*pi/180),cos(b_i*pi/180));
b_f=atan2(sin(b_f*pi/180),cos(b_f*pi/180));

D = sqrt((xf-xi)^2+(yf-yi)^2);
gamma = atan2((yf-yi),(xf-xi));

%_%_%_% Condición para D = 0
if(D==0)
    m=0;
    l=0;

    ga(1)=atan2(sin(-b_i+b_f),cos(-b_i+b_f));
    p(1)= abs(ga(1));
    G1=abs(ga(1));

    if (ga(1)>=0)
        s1=1;
    else
        s1=-1;
    end

    G2=0;    G3=0;
end
%_%_%_% Termina condición para D = 0

%_%_%_% Condición para D múltiplo entero de R
if (mod(D,R)==0) && (D~=0)
```

```

m=D/R;
l=0;

zeta_i = b_i-gamma;
zeta_f = b_f-gamma;

zeta_i=atan2(sin(zeta_i),cos(zeta_i));
zeta_f=atan2(sin(zeta_f),cos(zeta_f));

%Giros del robot
ga(1)=-zeta_i;
ga(2)=atan2(sin(pi-zeta_i),cos(pi-zeta_i));

gb(1)=zeta_f;
gb(2)=atan2(sin(-pi+zeta_f),cos(-pi+zeta_f));

%Valores absolutos de los ángulos para c. menor camino de angulos
for i=1:1:2
    gam(i)=abs(ga(i));
    gbm(i)=abs(gb(i));
end

%Calculo de caminos
p(1) = gam(1) + gbm(1);
p(2) = gam(2) + gbm(2);

%Selección del camino menor
if (p(1)<=p(2))
    camino=1;
else
    camino=2;
end

%Selección de giros
if (camino==1)
    G1=gam(1);    G2=gbm(1);    G3=0;
    ng1=1;        ng2=1;        ng3=1;
elseif (camino==2)
    G1=gam(2);    G2=gbm(2);    G3=0;
    ng1=2;        ng2=2;        ng3=2;
end

%Dirección del Giro 1
if ((ng1==2) && (ga(2)>=0)) || ((ng1==1) && (ga(1)>=0))
    s1=1;
else
    s1=-1;
end

%Dirección del Giro 2
if ((ng2==1) && (gb(1)>=0)) || ((ng2==2) && (gb(2)>=0))
    s2=1;

```



```

else
    s2=-1;
end

%Dirección del Giro 3
s3=1;

%Dirección del primer avance (adelante=-1, reversa=1)
if (ng1==1)
    av1=-1;
else
    av1=1;
end

%Dirección del segundo avance (adelante=-1, reversa=1)
av2=1;
end
% _ _ _ _ % Termina Condición para D multiplo de R

% _ _ _ _ % Condición para 0<D<R
if (D<R) && (mod(D,R)~=0)

    m=1;
    l=1;

    alp2=acos(((m*R)^2)+(l*R)^2-D^2)/(2*m*l*(R^2));
    alp1=asin((l*R*sin(alp2))/D);
    alp3=asin((m*R*sin(alp2))/D);

    zeta_i = b_i-alp1-gamma;
    zeta_f = b_f-pi+alp3-gamma;

    zeta_i=atan2(sin(zeta_i),cos(zeta_i));
    zeta_f=atan2(sin(zeta_f),cos(zeta_f));

    b_zeta_i = b_i-pi+alp1-gamma;
    b_zeta_f = b_f-alp3-gamma;

    b_zeta_i=atan2(sin(b_zeta_i),cos(b_zeta_i));
    b_zeta_f=atan2(sin(b_zeta_f),cos(b_zeta_f));

    %Giros del robot (triángulo de arriba)
    ga(1)=-zeta_i;
    ga(2)=atan2(sin(pi-zeta_i),cos(pi-zeta_i));

    gb(1)=alp2;
    gb(2)=alp2-pi;

    gc(1)=zeta_f;
    gc(2)=atan2(sin(-pi+zeta_f),cos(-pi+zeta_f));

    %Giros del robot (triángulo de abajo)

```

```

b_ga(1)=-b_zeta_i;
b_ga(2)=atan2(sin(pi-b_zeta_i),cos(pi-b_zeta_i));

b_gb(1)=-alp2;
b_gb(2)=pi-alp2;

b_gc(1)=b_zeta_f;
b_gc(2)=atan2(sin(-pi+b_zeta_f),cos(-pi+b_zeta_f));

%Valores absolutos de los angulos
for i=1:1:2
    gam(i)=abs(ga(i));
    gbm(i)=abs(gb(i));
    gcm(i)=abs(gc(i));

    b_gam(i)=abs(b_ga(i));
    b_gbm(i)=abs(b_gb(i));
    b_gcm(i)=abs(b_gc(i));
end

%Calculo de caminos
p(1) = gam(1) + gbm(1) + gcm(1);
p(2) = gam(1) + gbm(2) + gcm(2);
p(3) = gam(2) + gbm(1) + gcm(2);
p(4) = gam(2) + gbm(2) + gcm(1);

p(5) = b_gam(1) + b_gbm(1) + b_gcm(1);
p(6) = b_gam(1) + b_gbm(2) + b_gcm(2);
p(7) = b_gam(2) + b_gbm(1) + b_gcm(2);
p(8) = b_gam(2) + b_gbm(2) + b_gcm(1);

%Selección del camino menor
p_min=p(1);
s=length(p);
camino=1;

    for i=2:1:s
        if (p(i)<p_min)
            p_min=p(i);
            camino=i;
        end
    end

%Selección de giros
if (camino==1)
    G1=gam(1);    G2=gbm(1);    G3=gcm(1);
    ng1=1;        ng2=1;        ng3=1;
elseif (camino==2)
    G1=gam(1);    G2=gbm(2);    G3=gcm(2);
    ng1=1;        ng2=2;        ng3=2;
elseif (camino==3)
    G1=gam(2);    G2=gbm(1);    G3=gcm(2);
    ng1=2;        ng2=1;        ng3=2;

```

```

elseif (camino==4)
    G1=gam(2);    G2=gbm(2);    G3=gcm(1);
    ng1=2;        ng2=2;        ng3=1;

elseif (camino==5)
    G1=b_gam(1);  G2=b_gbm(1);    G3=b_gcm(1);
    ng1=1;        ng2=1;        ng3=1;
elseif (camino==6)
    G1=b_gam(1);  G2=b_gbm(2);    G3=b_gcm(2);
    ng1=1;        ng2=2;        ng3=2;
elseif (camino==7)
    G1=b_gam(2);  G2=b_gbm(1);    G3=b_gcm(2);
    ng1=2;        ng2=1;        ng3=2;
elseif (camino==8)
    G1=b_gam(2);  G2=b_gbm(2);    G3=b_gcm(1);
    ng1=2;        ng2=2;        ng3=1;
end

%%PARA DISTINGUIR ENTRE LOS GIROS Y AVANCES DE CADA TRIANGULO
if(camino<=4)

    %Dirección del Giro 1
    if ((ng1==1) && (ga(1)>=0)) || ((ng1==2) && (ga(2)>=0))
        s1=1;
    else
        s1=-1;
    end

    %Dirección del Giro 2
    if (ng2==1)
        s2=1;
    else
        s2=-1;
    end

    %Dirección del Giro 3
    if ((ng3==1) && (gc(1)>=0)) || ((ng3==2) && (gc(2)>=0))
        s3=1;
    else
        s3=-1;
    end

    %Dirección del primer avance (adelante=-1, reversa=1)
    if(ng1==1)
        av1=-1;
    else
        av1=1;
    end

    %Dirección del segundo avance (adelante=-1, reversa=1)
    if((ng1==1) && (ng2==1)) || ((ng1==2) && (ng2==2))
        av2=1;
    else

```

```

        av2=-1;
    end
else

    %Dirección del Giro 1
    if ((ng1==1) && (b_ga(1)>=0)) || ((ng1==2) && (b_ga(2)>=0))
        s1=1;
    else
        s1=-1;
    end

    %Dirección del Giro 2
    if (ng2==2)
        s2=1;
    else
        s2=-1;
    end

    %Dirección del Giro 3
    if ((ng3==1) && (b_gc(1)>=0)) || ((ng3==2) && (b_gc(2)>=0))
        s3=1;
    else
        s3=-1;
    end

    %Dirección del primer avance (adelante=-1, reversa=1)
    if(ng1==1)
        av1=1;
    else
        av1=-1;
    end

    %Dirección del segundo avance (adelante=-1, reversa=1)
    if((ng1==1) && (ng2==1)) || ((ng1==2) && (ng2==2))
        av2=-1;
    else
        av2=1;
    end
end
end
end

%%_%%_%%_%%_%%Termina condición 0<D<R

%%_%%_%%_%%_%%Condición de larga distancia y D no multiplo de R
if (D>R) && (mod(D,R)~=0)

    min_giros=3*pi;
    n=floor(D/R)+1;

    for j=1:1:n-1
        l=j;
        m=n-1;

        alp2=acos(((m*R)^2)+(l*R)^2-D^2)/(2*m*l*(R^2));

```

```

alp1=asin((l*R*sin(alp2))/D);
alp3=asin((m*R*sin(alp2))/D);

zeta_i = b_i-alp1-gamma;
zeta_f = b_f-pi+alp3-gamma;

zeta_i=atan2(sin(zeta_i),cos(zeta_i));
zeta_f=atan2(sin(zeta_f),cos(zeta_f));

b_zeta_i = b_i-pi+alp1-gamma;
b_zeta_f = b_f-alp3-gamma;

b_zeta_i=atan2(sin(b_zeta_i),cos(b_zeta_i));
b_zeta_f=atan2(sin(b_zeta_f),cos(b_zeta_f));

%Giros del robot (triángulo de arriba)
ga(1)=-zeta_i;
ga(2)=atan2(sin(pi-zeta_i),cos(pi-zeta_i));

gb(1)=alp2;
gb(2)=alp2-pi;

gc(1)=zeta_f;
gc(2)=atan2(sin(-pi+zeta_f),cos(-pi+zeta_f));

%Giros del robot (triángulo de abajo)
b_ga(1)=-b_zeta_i;
b_ga(2)=atan2(sin(pi-b_zeta_i),cos(pi-b_zeta_i));

b_gb(1)=-alp2;
b_gb(2)=pi-alp2;

b_gc(1)=b_zeta_f;
b_gc(2)=atan2(sin(-pi+b_zeta_f),cos(-pi+b_zeta_f));

%Valores absolutos de los angulos
for i=1:1:2
    gam(i)=abs(ga(i));
    gbm(i)=abs(gb(i));
    gcm(i)=abs(gc(i));

    b_gam(i)=abs(b_ga(i));
    b_gbm(i)=abs(b_gb(i));
    b_gcm(i)=abs(b_gc(i));
end

%Calculo de caminos
p(1) = gam(1) + gbm(1) + gcm(1);
p(2) = gam(1) + gbm(2) + gcm(2);
p(3) = gam(2) + gbm(1) + gcm(2);
p(4) = gam(2) + gbm(2) + gcm(1);

```

```

p(5) = b_gam(1) + b_gbm(1) + b_gcm(1);
p(6) = b_gam(1) + b_gbm(2) + b_gcm(2);
p(7) = b_gam(2) + b_gbm(1) + b_gcm(2);
p(8) = b_gam(2) + b_gbm(2) + b_gcm(1);

%Selección del camino menor
p_min=p(1);
s=length(p);
camino=1;

    for i=2:1:s
        if (p(i)<p_min)
            p_min=p(i);
            camino=i;
        end
    end

    %Condición para obtener l_optima
    if(p(camino)<min_giros)
        min_giros=p(camino);
        l_optima=j;
        camino_optimo=camino;
    end
end

%%Evaluando con la "l" optima
l=l_optima
m=n-l;

alp2=acos(((m*R)^2)+(l*R)^2-D^2)/(2*m*l*(R^2));
alp1=asin((l*R*sin(alp2))/D);
alp3=asin((m*R*sin(alp2))/D);

zeta_i = b_i-alp1-gamma;
zeta_f = b_f-pi+alp3-gamma;

zeta_i=atan2(sin(zeta_i),cos(zeta_i));
zeta_f=atan2(sin(zeta_f),cos(zeta_f));

b_zeta_i = b_i-pi+alp1-gamma;
b_zeta_f = b_f-alp3-gamma;

b_zeta_i=atan2(sin(b_zeta_i),cos(b_zeta_i));
b_zeta_f=atan2(sin(b_zeta_f),cos(b_zeta_f));

%Giros del robot (triángulo de arriba)
ga(1)=-zeta_i;
ga(2)=atan2(sin(pi-zeta_i),cos(pi-zeta_i));

gb(1)=alp2;
gb(2)=alp2-pi;

```

```

gc(1)=zeta_f;
gc(2)=atan2(sin(-pi+zeta_f),cos(-pi+zeta_f));

%Giros del robot (triángulo de abajo)
b_ga(1)=-b_zeta_i;
b_ga(2)=atan2(sin(pi-b_zeta_i),cos(pi-b_zeta_i));

b_gb(1)=-alp2;
b_gb(2)=pi-alp2;

b_gc(1)=b_zeta_f;
b_gc(2)=atan2(sin(-pi+b_zeta_f),cos(-pi+b_zeta_f));

%Valores absolutos de los angulos
for i=1:1:2
    gam(i)=abs(ga(i));
    gbm(i)=abs(gb(i));
    gcm(i)=abs(gc(i));

    b_gam(i)=abs(b_ga(i));
    b_gbm(i)=abs(b_gb(i));
    b_gcm(i)=abs(b_gc(i));
end

%Calculo de caminos
p(1) = gam(1) + gbm(1) + gcm(1);
p(2) = gam(1) + gbm(2) + gcm(2);
p(3) = gam(2) + gbm(1) + gcm(2);
p(4) = gam(2) + gbm(2) + gcm(1);

p(5) = b_gam(1) + b_gbm(1) + b_gcm(1);
p(6) = b_gam(1) + b_gbm(2) + b_gcm(2);
p(7) = b_gam(2) + b_gbm(1) + b_gcm(2);
p(8) = b_gam(2) + b_gbm(2) + b_gcm(1);

%Selección del camino menor
p_min=p(1);
s=length(p);
camino=1;

    for i=2:1:s
        if (p(i)<p_min)
            p_min=p(i);
            camino=i;
        end
    end

%Selección de giros
if (camino==1)
    G1=gam(1);    G2=gbm(1);    G3=gcm(1);
    ng1=1;        ng2=1;        ng3=1;
elseif (camino==2)

```

```

        G1=gam(1);    G2=gbm(2);    G3=gcm(2);
        ng1=1;        ng2=2;        ng3=2;
elseif (camino==3)
        G1=gam(2);    G2=gbm(1);    G3=gcm(2);
        ng1=2;        ng2=1;        ng3=2;
elseif (camino==4)
        G1=gam(2);    G2=gbm(2);    G3=gcm(1);
        ng1=2;        ng2=2;        ng3=1;

elseif (camino==5)
        G1=b_gam(1);    G2=b_gbm(1);    G3=b_gcm(1);
        ng1=1;        ng2=1;        ng3=1;
elseif (camino==6)
        G1=b_gam(1);    G2=b_gbm(2);    G3=b_gcm(2);
        ng1=1;        ng2=2;        ng3=2;
elseif (camino==7)
        G1=b_gam(2);    G2=b_gbm(1);    G3=b_gcm(2);
        ng1=2;        ng2=1;        ng3=2;
elseif (camino==8)
        G1=b_gam(2);    G2=b_gbm(2);    G3=b_gcm(1);
        ng1=2;        ng2=2;        ng3=1;
end

%%PARA DISTINGUIR ENTRE LOS GIROS Y AVANCES DE CADA TRIANGULO
if(camino<=4)

    %Dirección del Giro 1
    if ((ng1==1) && (ga(1)>=0)) || ((ng1==2) && (ga(2)>=0))
        s1=1;
    else
        s1=-1;
    end

    %Dirección del Giro 2
    if (ng2==1)
        s2=1;
    else
        s2=-1;
    end

    %Dirección del Giro 3
    if ((ng3==1) && (gc(1)>=0)) || ((ng3==2) && (gc(2)>=0))
        s3=1;
    else
        s3=-1;
    end

    %Dirección del primer avance (adelante=-1, reversa=1)
    if(ng1==1)
        av1=-1;
    else
        av1=1;
    end
end

```



```

%Dirección del segundo avance (adelante=-1, reversa=1)
if ((ng1==1) &&(ng2==1)) || ((ng1==2) &&(ng2==2))
    av2=1;
else
    av2=-1;
end
else

%Dirección del Giro 1
if (( (ng1==1) &&(b_ga(1)>=0) ) || ( (ng1==2) &&(b_ga(2)>=0) ) )
    s1=1;
else
    s1=-1;
end

%Dirección del Giro 2
if (ng2==2)
    s2=1;
else
    s2=-1;
end

%Dirección del Giro 3
if ((ng3==1) &&(b_gc(1)>=0) ) || ((ng3==2) &&(b_gc(2)>=0) )
    s3=1;
else
    s3=-1;
end

%Dirección del primer avance (adelante=-1, reversa=1)
if(ng1==1)
    av1=1;
else
    av1=-1;
end

%Dirección del segundo avance (adelante=-1, reversa=1)
if ((ng1==1) &&(ng2==1)) || ((ng1==2) &&(ng2==2))
    av2=-1;
else
    av2=1;
end
end
end

%_ _ _ _ _ _ _ _ Termina condición de larga distancia y D no multiplo de R

sism = 15; %largo de las flechas del sistema de
referencia móvil
sism1= 8;
Dim=150;

%-----PRIMER GIRO-----

```

```

for i=0:G1/10:G1
    if (s1==1)
        inc=i;
    else
        inc=-i;
    end

    plot(-Dim,-Dim)
    hold on
    plot(Dim,Dim)
    daspect([1,1,1]);

    %Robot
    xa0 = xi;
    xa1 = xa0 + 35*cos(b_i+inc);
    xa2 = xa0 - 20*sin(b_i+inc);
    xa3 = xa0 + 20*sin(b_i+inc);

    ya0 = yi;
    ya1 = ya0 + 35*sin(b_i+inc);
    ya2 = ya0 + 20*cos(b_i+inc);
    ya3 = ya0 - 20*cos(b_i+inc);

    Xa = [xa2; xa0; xa1; xa0; xa3];
    Ya = [ya2; ya0; ya1; ya0; ya3];

    %sistema de referencia inercial
    xi0 = 0;
    xi1 = 40;

    yi0 = 0;
    yi1 = 40;

    Xi = [xi0; xi0; xi1];
    Yi = [yi1; yi0; yi0];

    %Posición inicial del robot
    xb0 = xa0;
    xb1 = xb0 - sism1*sin(b_i);
    xb2 = xb0 + sism*cos(b_i);

    yb0 = ya0;
    yb1 = yb0 + sism1*cos(b_i);
    yb2 = yb0 + sism*sin(b_i);

    Xb = [xb1; xb0; xb2];
    Yb = [yb1; yb0; yb2];

    %Posición final del robot
    xf0 = xf;
    xf1 = xf0 - sism1*sin(b_f);
    xf2 = xf0 + sism*cos(b_f);

```

```

yf0 = yf;
yf1 = yf0 + sism1*cos(b_f);
yf2 = yf0 + sism*sin(b_f);

Xfinal = [xf1; xf0; xf2];
Yfinal = [yf1; yf0; yf2];

compass(Xi,Yi);
line(Xa,Ya,'LineWidth',1.5);
line(Xb,Yb,'Color','black','LineWidth',1.5);
line(Xfinal,Yfinal,'Color','g','LineWidth',1.5)

hold off
pause(.1)
grid on
b_act=b_i+inc;
end

%Puede que no haya que no sea neceserio girar G1 grados por ello
%necesitamos dar un valor a b_act la cual se heredará al siguiente ciclo
if (G1==0)
    b_act=b_i;
end

%-----PRIMER AVANCE-----
for i=0:R/10:m*R
    if (av1<0)
        inc_x=i*cos(b_act);
        inc_y=i*sin(b_act);
    else
        inc_x=-i*cos(b_act);
        inc_y=-i*sin(b_act);
    end

    plot(-Dim,-Dim)
    hold on
    plot(Dim,Dim)
    daspect([1,1,1]);

    %Robot
    xa0 = xi+inc_x;
    xa1 = xa0 + 35*cos(b_act);
    xa2 = xa0 - 20*sin(b_act);
    xa3 = xa0 + 20*sin(b_act);

    ya0 = yi+inc_y;
    ya1 = ya0 + 35*sin(b_act);
    ya2 = ya0 + 20*cos(b_act);
    ya3 = ya0 - 20*cos(b_act);

    Xa = [xa0; xa2; xa1; xa3; xa0; xa1];

```

```

Ya = [ya0; ya2; ya1; ya3; ya0; ya1];

if (mod(i,R)==0)
    Xa = [xa2; xa0; xa1; xa0; xa3];
    Ya = [ya2; ya0; ya1; ya0; ya3];
end

%sistema de referencia inercial
xi0 = 0;
xi1 = 40;

yi0 = 0;
yi1 = 40;

Xi = [xi0; xi0; xi1];
Yi = [yi1; yi0; yi0];

%Posición inicial
xb0 = xi;
xb1 = xb0 - sism1*sin(b_i);
xb2 = xb0 + sism*cos(b_i);

yb0 = yi;
yb1 = yb0 + sism1*cos(b_i);
yb2 = yb0 + sism*sin(b_i);

Xb = [xb1; xb0; xb2];
Yb = [yb1; yb0; yb2];

%Posición final del robot
xf0 = xf;
xf1 = xf0 - sism1*sin(b_f);
xf2 = xf0 + sism*cos(b_f);

yf0 = yf;
yf1 = yf0 + sism1*cos(b_f);
yf2 = yf0 + sism*sin(b_f);

Xfinal = [xf1; xf0; xf2];
Yfinal = [yf1; yf0; yf2];

%Trayectoria
TX(j) = xa0;
TY(j) = ya0;

compass(Xi,Yi);
line(TX',TY', 'Color','r', 'LineWidth',1.5);
line(Xa,Ya, 'LineWidth',1.5);
line(Xb,Yb, 'Color','black', 'LineWidth',1.5);
line(Xfinal,Yfinal, 'Color','g', 'LineWidth',1.5)

hold off

```

```

    pause(.1)
    grid on

    x_act=xa0;
    y_act=ya0;
end

%Puede que no haya que no sea neceserio girar G1 grados por ello
%necesitamos dar un valor a b_act la cual se heredar  al siguiente ciclo
if (m==0)
    x_act=xi;
    y_act=yi;
end

%-----SEGUNDO GIRO-----
for i=0:G2/10:G2
    if (s2==1)
        inc=i;
    else
        inc=-i;
    end

    plot(-Dim,-Dim)
    hold on
    plot(Dim,Dim)
    daspect([1,1,1]);

    %Robot
    xa0 = x_act;
    xa1 = xa0 + 35*cos(b_act+inc);
    xa2 = xa0 - 20*sin(b_act+inc);
    xa3 = xa0 + 20*sin(b_act+inc);

    ya0 = y_act;
    ya1 = ya0 + 35*sin(b_act+inc);
    ya2 = ya0 + 20*cos(b_act+inc);
    ya3 = ya0 - 20*cos(b_act+inc);

    Xa = [xa2; xa0; xa1; xa0; xa3];
    Ya = [ya2; ya0; ya1; ya0; ya3];

    %sistema de referencia inercial
    xi0 = 0;
    xi1 = 40;

    yi0 = 0;
    yi1 = 40;

    Xi = [xi0; xi0; xi1];
    Yi = [yi1; yi0; yi0];

```

```

%Posición inicial
xb0 = xi;
xb1 = xb0 - sism1*sin(b_i);
xb2 = xb0 + sism*cos(b_i);

yb0 = yi;
yb1 = yb0 + sism1*cos(b_i);
yb2 = yb0 + sism*sin(b_i);

Xb = [xb1; xb0; xb2];
Yb = [yb1; yb0; yb2];

%Posición final del robot
xf0 = xf;
xf1 = xf0 - sism1*sin(b_f);
xf2 = xf0 + sism*cos(b_f);

yf0 = yf;
yf1 = yf0 + sism1*cos(b_f);
yf2 = yf0 + sism*sin(b_f);

Xfinal = [xf1; xf0; xf2];
Yfinal = [yf1; yf0; yf2];

compass(Xi,Yi);
line(TX',TY', 'Color','r', 'LineWidth',1.5);
line(Xa,Ya, 'LineWidth',1.5);
line(Xb,Yb, 'Color','black', 'LineWidth',1.5);
line(Xfinal,Yfinal, 'Color','g', 'LineWidth',1.5)

hold off
pause(.1)
grid on
end

%Puede que no haya que no sea neceserio girar G1 grados por ello
%necesitamos dar un valor a b_act la cual se heredará al siguiente ciclo
if (G2~=0)
    b_act=b_act+inc;
end

%-----SEGUNDO AVANCE-----
for i=0:R/10:1*R
    if (av2<0)
        inc_x=i*cos(b_act);
        inc_y=i*sin(b_act);
    else
        inc_x=-i*cos(b_act);
        inc_y=-i*sin(b_act);
    end

    plot(-Dim,-Dim)

```

```

hold on
plot(Dim,Dim)
daspect([1,1,1]);

%Robot
xa0 = x_act+inc_x;
xa1 = xa0 + 35*cos(b_act);
xa2 = xa0 - 20*sin(b_act);
xa3 = xa0 + 20*sin(b_act);

ya0 = y_act+inc_y;
ya1 = ya0 + 35*sin(b_act);
ya2 = ya0 + 20*cos(b_act);
ya3 = ya0 - 20*cos(b_act);

Xa = [xa0; xa2; xa1; xa3; xa0; xa1];
Ya = [ya0; ya2; ya1; ya3; ya0; ya1];

if (mod(i,R)==0)
    Xa = [xa2; xa0; xa1; xa0; xa3];
    Ya = [ya2; ya0; ya1; ya0; ya3];
end

if (l==0)
    Xa = [xa2; xa0; xa1; xa0; xa3];
    Ya = [ya2; ya0; ya1; ya0; ya3];
end

%sistema de referencia inercial
xi0 = 0;
xi1 = 40;

yi0 = 0;
yi1 = 40;

Xi = [xi0; xi0; xi1];
Yi = [yi1; yi0; yi0];

%Posición inicial
xb0 = xi;
xb1 = xb0 - sism1*sin(b_i);
xb2 = xb0 + sism*cos(b_i);

yb0 = yi;
yb1 = yb0 + sism1*cos(b_i);
yb2 = yb0 + sism*sin(b_i);

Xb = [xb1; xb0; xb2];
Yb = [yb1; yb0; yb2];

%Posición final del robot
xf0 = xf;

```

```

xf1 = xf0 - sism1*sin(b_f);
xf2 = xf0 + sism*cos(b_f);

yf0 = yf;
yf1 = yf0 + sism1*cos(b_f);
yf2 = yf0 + sism*sin(b_f);

Xfinal = [xf1; xf0; xf2];
Yfinal = [yf1; yf0; yf2];

%Trayectoria
TX1(j) = xa0;
TY1(j) = ya0;

compass(Xi,Yi);
line(TX',TY', 'Color','r', 'LineWidth',1.5);
line(TX1',TY1', 'Color','r', 'LineWidth',1.5);
line(Xa,Ya, 'LineWidth',1.5);
line(Xb,Yb, 'Color','black', 'LineWidth',1.5);
line(Xfinal,Yfinal, 'Color','g', 'LineWidth',1.5)

hold off
pause(.1)
grid on
end
x_act = xa0;
y_act = ya0;

%-----TERCER GIRO-----
for i=0:G3/10:G3
    if (s3==1)
        inc=i;
    else
        inc=-i;
    end

    plot(-Dim,-Dim)
    hold on
    plot(Dim,Dim)
    daspect([1,1,1]);

    %Robot
    xa0 = x_act;
    xa1 = xa0 + 35*cos(b_act+inc);
    xa2 = xa0 - 20*sin(b_act+inc);
    xa3 = xa0 + 20*sin(b_act+inc);

    ya0 = y_act;
    ya1 = ya0 + 35*sin(b_act+inc);
    ya2 = ya0 + 20*cos(b_act+inc);
    ya3 = ya0 - 20*cos(b_act+inc);

```



```

Xa = [xa2; xa0; xa1; xa0; xa3];
Ya = [ya2; ya0; ya1; ya0; ya3];

%sistema de referencia inercial
xi0 = 0;
xi1 = 40;

yi0 = 0;
yi1 = 40;

Xi = [xi0; xi0; xi1];
Yi = [yi1; yi0; yi0];

%Posición inicial
xb0 = xi;
xb1 = xb0 - sism1*sin(b_i);
xb2 = xb0 + sism*cos(b_i);

yb0 = yi;
yb1 = yb0 + sism1*cos(b_i);
yb2 = yb0 + sism*sin(b_i);

Xb = [xb1; xb0; xb2];
Yb = [yb1; yb0; yb2];

% Posición final del robot
xf0 = xf;
xf1 = xf0 - sism1*sin(b_f);
xf2 = xf0 + sism*cos(b_f);

yf0 = yf;
yf1 = yf0 + sism1*cos(b_f);
yf2 = yf0 + sism*sin(b_f);

Xfinal = [xf1; xf0; xf2];
Yfinal = [yf1; yf0; yf2];

compass(Xi,Yi);
line(TX',TY', 'Color','r', 'LineWidth',1.5);
line(TX1',TY1', 'Color','r', 'LineWidth',1.5);
line(Xa,Ya, 'LineWidth',1.5);
line(Xb,Yb, 'Color','black', 'LineWidth',1.5);
line(Xfinal,Yfinal, 'Color','g', 'LineWidth',1.5)

hold off
pause(.1)
grid on
end

if(G3~=0)
    b_act=b_act+inc;
end

```



# Bibliografía

- Alonso J.R., Maya M., Morales E., Félix L. & Guarneros O. (2011). *Modelado Cinemático de un Robot no-Holónimo de 2 Grados de Libertad con Patas*. XII Congreso Mexicano de Robótica. San Luis Potosí, México. Octubre de 2011.
- Arora J.S. (2004). *Introduction to Optimum Design*. Elsevier Academic Press, 2da edición. (2004).
- Avila S.I. & Zambrano J.C. (2007). *Estudio, Diseño y Construcción de un Robot Cuadrúpedo Omnidireccional*. Universidad Politécnica Salesiana. Cuenca, Ecuador. Recuperado el 2 de Diciembre de 2011 de <http://dspace.ups.edu.ec/handle/123456789/188>.
- Cardona S. & Clos D. (2001). *Teoría de Máquinas*. Ediciones UPC, Barcelona, 2001.
- Díaz L.I. & García L.A. *Estado del Arte en Robótica Cuadrúpeda Semiautónoma*. Universidad Europea de Madrid. Recuperado el 7 de Julio de 2010 de: <http://www.r4p.es/PFC/docs/Estado%20del%20Arte%20Rob%F3tica%20Cuadrupeda.pdf>.
- Espitia H. (2011). *Propuesta de un Algoritmo para la Planeación de Trayectorias de Robots Móviles Empleando Campos Potenciales y Enjambres de Partículas Activas Brownianas*. Tesis de Magister. Universidad Nacional de Colombia, Bogotá.
- Font J.M. & Battle J. (2005), *Posicionamiento dinámico y guiado de un robot móvil con cinemática de triciclo*. Universidad Politécnica de Catalunya. Barcelona, España.
- Gómez F., Cuesta F. & Ollero A. *Planificación de Trayectorias en Robots Móviles Basada en Técnicas de Control de Sistemas no Holónomos*. Depto. Ingeniería de Sistemas y Automática. Escuela Superior de Ingenieros, Universidad de Sevilla. Recuperado el 9 de Junio de 2010 de: <http://www.ceautomatica.es/old/actividades/jornadas/XXIV/documentos/ro/174.pdf>.

- González E.J. (2010). *Notas sobre Teoría de Estimación aplicada a la Robótica*. Centro de Investigación y Estudios de Posgrado, Universidad Autónoma de San Luis Potosí. San Luis Potosí, México.
- González P., Garcia E., Ponticelli R. & Armanda M. (2009). *Minimizing Energy Consumption in Hexapod Robots*. *Advanced Robotics* 23 (2009) 681-704.
- Hernández I. & Diaz F. *Desarrollo de un robot cuadrúpedo con base en servomotores, ANGELO*. Reporte Técnico. Universidad Pontificia Bolivariana.
- Intriago E. (2009). *Historia y Evolución de la Robótica Industrial*. Universidad Técnica de Ambato. Recuperado el 3 de enero de 2012 de: <http://eduardointriago.wikispaces.com/file/view/UNIDAD+I+FUNDAMENTOS+GENERAL+LES.pdf>.
- Lamiroux F. & Laumond J. (2001). *Smooth Motion Planning for Car-Like Vehicles*. *IEEE Transactions on Robotics and Automation*. Vol. 17, No.4, Agosto 2001.
- Laumond J. (1998). *Robot Motion Planning and Control*. Laboratoire d'Architecture des Systèmes Centre National de la Recherche Scientifique. 1998.
- Liu J., Tan M. & Zhao X. (2007). *Legged robots – an overview*. *Transactions of the Institute of Measurement and Control* 29, 2 pp.185-202.
- Marín F.J., Casillas J., Musientes M., Transeth A.A., Fjerdinegen I. & Schjolberg I. (2011). *Aprendizaje de Controladores Difusos para Seguimiento de Trayectorias en Robots Multiarticulados*. *Actas ROBOT 2011*. 28-29 de Noviembre de 2011. Sevilla, España.
- Morales R. (2008). *Crean un robot submarino que se suspende en el mar como un helicóptero*. Recuperado el 10 de Diciembre de 2011 de: [http://www.tendencias21.net/Crean-un-robot-submarino-que-se-suspende-en-el-mar-como-un-helicoptero\\_a2594.html](http://www.tendencias21.net/Crean-un-robot-submarino-que-se-suspende-en-el-mar-como-un-helicoptero_a2594.html).
- Muñoz V. (1995). *Planificación de Trayectorias para Robots Móviles*. Tesis Doctoral. Universidad de Málaga.
- Norton R.L. (2000). *Diseño de Maquinaria*. Editorial McGraw Hill. Año 2000. México,
- Ollero A. (2001). *Robótica: Manipuladores y Robots Móviles*. Editorial Marcombo.
- Párraga C.R. (2008), *Diseño de un Prototipo de Robot Recogedor de Basura que Funciona con Energía Solar*. Escuela Superior Politécnica del Litoral. Guayaquil, Ecuador.
- Pivtoraiko M., Knepper R. & Kelly A. (2009). *Differentially Constrained Mobile Robot Motion Planning in State Lattices*. *Journal of Field Robotics*, vol. 26:3, pp. 308-333, Marzo 2009.

- Roque R.V. & Aguilera M.I. (2010), *Modelado, simulación y control de un robot hexápodo: lecciones aprendidas*. Instituto Tecnológico de Nuevo Laredo. Nuevo Laredo, Tamaulipas. Recuperado el 16 de Junio de 2010 de: <http://www.depi.itchihuahua.edu.mx/electro/archivo/electro2000/mem2000/articulos/RO3.pdf>
- Tovar A., & Renaud J.E. (2010). *Optimum Design of Mechanical Elements: Class notes for AME60661*. Mayo 21, 2010.
- Vargas E., Gorrostieta E. & Pérez A. (2002). *Desarrollando una Pata para un Robot Hexápodo*. VII Reunión Internacional de Ingeniería Mecánica, Octubre 30-Noviembre 1, 2002, San Luis Potosí.