

# **ANÁLISIS CINEMÁTICO A UN ROBOT MÓVIL KRYPTON 1 PARA PLANEACIÓN DE TRAYECTORIAS**

## *KINEMATIC ANALYSIS OF A KRYPTON 1 MOBILE ROBOT FOR PATH PLANNING*

**Salvador Manuel Malagón Soldara**

Tecnológico Nacional de México / IT de Celaya, México  
*salvador.malagon@itcelaya.edu.mx*

**Sara Marcela Arellano Díaz**

Tecnológico Nacional de México / IT de Celaya, México  
*sara.arellano@itcelaya.edu.mx*

**José Luis Hurtado Chávez**

Tecnológico Nacional de México / IT de Celaya, México  
*luis.hurtado@itcelaya.edu.mx*

**David Gasca Figueroa**

Tecnológico Nacional de México / IT de Celaya, México  
*david.gasca@itcelaya.edu.mx*

**Lisseth Abilene Mata Ramírez**

Tecnológico Nacional de México / IT de Celaya, México  
*17030049@itcelaya.edu.mx*

**Recepción:** 13/abril/2023

**Aceptación:** 23/diciembre/2023

### **Resumen**

El presente proyecto de ingeniería se basó en el desarrollo de la programación de un robot Krypton 1. Dicha programación, ayudará en competencias de robótica WER para calcular los movimientos de los motores y las articulaciones del robot. Se utilizó el software de Abilix para el diseño de la programación del robot. Además, se empleó un análisis matemático con el método de cinemática inversa, el cual ayudó a obtener la velocidad y la posición de cada movimiento del robot. Posteriormente, se realizó la implementación de la cinemática a la programación para valorar los resultados obtenidos en la trayectoria que seguirá el robot. Por último, el algoritmo se comprobó mediante diversos ángulos en la pista WER. La finalidad fue el minimizar los errores de orientación y posición en los participantes de la competencia.

**Palabras Clave:** Abilix, cinemática, robot diferencial.

## **Abstract**

*The present engineering project was based on the development of the programming of a Krypton 1 robot. This programming will help in WER robotics competitions to calculate the movements of the motors and joints of the robot. Abilix software was used to design the robot programming. In addition, a mathematical analysis with the inverse kinematics method was used, which helped to obtain the velocity and position of each movement of the robot. Subsequently, the implementation of the kinematics to the programming was carried out to evaluate the results obtained in the trajectory that the robot will follow. Finally, the algorithm was tested using different angles on the WER track. The purpose was to minimize the orientation and position errors in the participants of the competition.*

**Keywords:** *Abilix, kinematics, differential robot.*

## **1. Introducción**

La robótica móvil es una herramienta que permite extender las aplicaciones, en comparación a los robots fijos, puesto que se pueden desarrollar nuevas tecnologías con la implementación de robots no sedentarios, que puedan realizar navegación autónoma. En la actualidad se aplican una gran variedad de métodos para la planeación y seguimiento de trayectorias que permitan a los robots cumplir con su objetivo. Dentro de la robótica móvil, una de las arquitecturas más utilizadas es la configuración diferencial. Sin embargo, un problema que presenta la de tracción diferencial es desplazarse en línea recta, ya que para que el robot pueda lograr esto, sus dos ruedas deben girar a la misma velocidad, pero si cada rueda experimenta una fricción diferente, sus velocidades van a variar [Valencia, 2009].

Este problema debe ser solucionado con un sistema de control dinámico que vaya variando las velocidades a medida que el robot lo necesite. En este sentido, primero se realiza el modelo del robot diferencial, con el objetivo de tener una representación matemática que describa la cinemática del sistema [González, 2015]. De la misma manera, el avance de las ruedas del robot ya sea en línea recta o si es necesario

girar 90° o 45° se puede determinar mediante el estudio de la cinemática directa e inversa del robot.

En la actualidad la industria de los robots mantiene un amplio campo de aplicación, ya sea para trabajos industriales, como lo son los brazos robóticos soldadores hasta los SCARA. Donde, este tipo de robots siguen un patrón repetitivo, sin embargo, el margen de aplicación es sumamente extenso. Debido al alcance de las nuevas tecnologías, la robótica ha sido capaz de pasar a ser una asignatura al alcance de todos, debido a los softwares que facilitan la programación de autómatas y empresas dedicadas a rediseñar el ensamble de diferentes tipos de robots [Zapata-Ros, 2014].

Es por ello por lo que, debido a su fácil acceso, manejo, ensamble y programación, se han creado diferentes competencias alrededor del mundo. Las cuales establecen requisitos y requerimientos para formar parte de dicho concurso. Ejemplo de esto, es investigación de [Toca, 2019] acerca del diseño de robots de competencia en categorías seguidor de línea y robot laberinto. En la cual, se desarrolla el diseño e implementación de un robot seguidor de línea en el ámbito de velocidad y un robot laberinto, permitiendo la implementación de la robótica en la parte mecánica, eléctrica y programación. Este tipo de trabajos son importantes, ya que dejan un historial acerca de las características a cuidar cuando se construye un robot de competencia. Esto nos ayuda a visualizar de mejor manera la estructura y factores a tomar en cuenta para el desarrollo de un algoritmo.

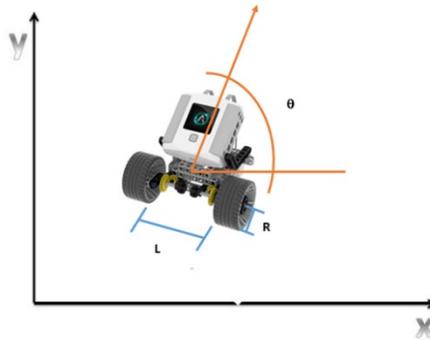
Por otro lado, existen publicaciones las cuales se enfocan en gran medida al cálculo de la cinemática del robot, la cual proporciona información que ayudará a controlar las trayectorias de este mismo, aportando de esta manera información para el correcto modelado de un robot móvil. [González, 2009] describe la implementación de estrategias de navegación de un robot móvil tipo diferencial a partir de la estimación odométrica y su implementación utilizando hardware reconfigurable FPGAs (arreglo de compuerta en campo).

Dentro de este trabajo se muestra la potencialidad de las FPGAS en la implementación de aplicaciones de robótica móvil. Además, se muestran los resultados de algunos experimentos realizados con la plataforma y los errores

odométricos de estos, los cuales se utilizarán para generación y corrección de trayectorias.

## 2. Métodos

De acuerdo con el objetivo, se planteó un sistema cinemático para la solución de trayectorias de un robot diferencial. Para plantear el modelado del sistema se tienen que identificar las variables de interés, que en este caso son las velocidades angulares de cada uno de los motores ( $\omega_l$ ) y ( $\omega_r$ ), el radio de las ruedas ( $R$ ), la longitud del eje entre las ruedas ( $L$ ) y el ángulo de posición en el que se encuentra el robot ( $\theta$ ). Una vez identificadas estas variables, se deben tener algunas consideraciones para analizar; idealmente el robot se desplaza en superficies planas y se mueve por las fuerzas de los motores de las ruedas. Además, el robot diferencial es un sistema no holonómico, por lo tanto, no se puede mover de forma lateral por sí solo, su desplazamiento es hacia adelante o hacia atrás. De esta forma, la figura 1 muestra las variables de interés en el robot diferencial Abilix Krypton 1.



Fuente: elaboración propia

Figura 1 Variables físicas del robot diferencial armado con el kit Krypton 1.

Para obtener las velocidades de las ruedas derecha ( $v_r$ ) e izquierda ( $v_l$ ), se emplean las ecuaciones 1 y 2. Donde ( $\omega_r$ ) es la velocidad angular y  $R$  el radio de la llanta.

$$v_l = \omega_r R \quad (1)$$

$$v_r = \omega_r R \quad (2)$$

Esto significa que para obtener la velocidad lineal de cada rueda se multiplica la velocidad angular ( $\omega_r$ ) de cada motor por el radio de las llantas ( $R$ ). Una vez

teniendo las velocidades lineales de las ruedas, se puede calcular la velocidad lineal total ( $v_T$ ) del robot, como se muestra en la ecuación 3.

$$v_T = \frac{v_r + v_l}{2} \quad (3)$$

Ahora, para que el robot tenga un movimiento rotatorio debe existir una velocidad angular ( $\omega_T$ ). Ésta se plantea como la diferencia de las velocidades de las llantas derecha e izquierda ( $v_r$  y  $v_l$ ) entre la longitud del eje ( $L$ ), como se muestra en la ecuación 4.

$$\omega_T = \frac{v_r - v_l}{L} \quad (4)$$

Además, para facilitar el desarrollo cinemático del robot se tomaron en cuenta ciertos aspectos:

- En un instante el ángulo del robot se encuentra en el origen.
- Pasado un tiempo el robot se habrá movido y estará en una orientación determinada.
- Suponiendo además que: el robot será considerado como un simple punto.
- El objetivo del modelo es determinar la posición horizontal ( $X$ ), la posición vertical ( $Y$ ) y la orientación con respecto al eje X del punto en el robot ( $\theta$ ).
- Aplicando una velocidad lineal ( $v_T$ ) y una velocidad angular ( $\omega_T$ ).
- El punto de control está ubicado en el centro del eje de las 2 ruedas.

Para que el robot se desplace en línea recta la velocidad de sus ruedas debe de ser igual. Por lo tanto, se puede definir la velocidad lineal como un promedio de la velocidad de ambas llantas, ecuaciones 5 y 6.

$$v = R \frac{v_r + v_l}{2} \quad (5)$$

$$w = R \frac{v_r - v_l}{L} \quad (6)$$

Donde  $h$  es una literal usada para representar la componente en cada eje. Posteriormente, se reemplazan los componentes de las velocidades en cada uno de los ejes, ecuaciones 7,8 y 9.

$$\dot{x}_r = v \cos\theta \quad (7)$$

$$\dot{y}_r = v \sin\theta \quad (8)$$

$$\dot{\phi} = \omega \quad (9)$$

Las tres ecuaciones anteriores se unirán de manera que las tres velocidades formen las ecuaciones 10, 11 y 12. Una vez que se obtuvieron estos valores y ecuaciones podemos plantear la cinemática inversa. Partiendo de la ecuación matricial y despejando las variables de interés, se obtiene la ecuación 13.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R \frac{v_r + v_l}{2} \\ R \frac{v_r + v_l}{L} \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} R \frac{\cos\theta}{2} & R \frac{\cos\theta}{2} \\ R \frac{\sin\theta}{2} & R \frac{\sin\theta}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix}^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (13)$$

La matriz de los coeficientes es llamada ("A") a partir de la cual es necesario encontrar la pseudoinversa de dicha matriz. Esta matriz se define con ecuación 14.

$$A^+ = (A^T * A)^{-1} * A^T \quad (14)$$

Primero se calcula la matriz transpuesta definida como la ecuación 15. Enseguida, se multiplica la Matriz transpuesta por la matriz original para obtener la ecuación 16. A partir de este resultado, se calcula la matriz inversa de la ecuación 17.

$$A^T = \begin{bmatrix} R \frac{\cos\theta}{2} & R \frac{\sin\theta}{2} & \frac{R}{L} \\ R \frac{\cos\theta}{2} & R \frac{\sin\theta}{2} & -\frac{R}{L} \end{bmatrix} \quad (15)$$

$$A^T A = \begin{bmatrix} R \frac{\cos\theta}{2} & R \frac{\sin\theta}{2} & \frac{R}{L} \\ R \frac{\cos\theta}{2} & R \frac{\sin\theta}{2} & -\frac{R}{L} \end{bmatrix} \cdot \begin{bmatrix} R \frac{\cos\theta}{2} & R \frac{\cos\theta}{2} \\ R \frac{\sin\theta}{2} & R \frac{\sin\theta}{2} \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} = \begin{bmatrix} \frac{R^2}{C^2} + \frac{R^3}{4} & \frac{R^2}{4} - \frac{R^2}{C^2} \\ \frac{R^2}{4} - \frac{R^2}{C^2} & \frac{R^2}{C^2} + \frac{R^2}{4} \end{bmatrix} \quad (16)$$

$$(A^T * A)^{-1} = \begin{bmatrix} \frac{4 + C^2}{4R^2} & \frac{-C^2 - 4}{4R^2} \\ \frac{-C^2 - 4}{4R^2} & \frac{4 + C^2}{4R^2} \end{bmatrix} \quad (17)$$

Después se multiplica la matriz inversa por la matriz transpuesta obteniendo así la matriz pseudoinversa de la ecuación 18.

$$A^+ = \begin{bmatrix} \frac{4 + C^2}{4R^2} & \frac{-C^2 - 4}{4R^2} \\ \frac{-C^2 - 4}{4R^2} & \frac{4 + C^2}{4R^2} \end{bmatrix} \cdot \begin{bmatrix} R \frac{\cos\theta}{2} & R \frac{\sin\theta}{2} & \frac{R}{L} \\ R \frac{\cos\theta}{2} & R \frac{\sin\theta}{2} & -\frac{R}{L} \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{R} & \frac{\sin\theta}{R} & \frac{L}{2R} \\ \frac{\cos\theta}{R} & \frac{\sin\theta}{R} & -\frac{L}{2R} \end{bmatrix} \quad (18)$$

Por lo tanto, esto resulta en la ecuación 19.

$$A^+ = \begin{bmatrix} \frac{\cos\theta}{R} & \frac{\sin\theta}{R} & \frac{L}{2R} \\ \frac{\cos\theta}{R} & \frac{\sin\theta}{R} & -\frac{L}{2R} \end{bmatrix} \quad (19)$$

Se sustituyen las ecuaciones para sacar la cinemática inversa de la ecuación 20.

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{R} & \frac{\sin\theta}{R} & \frac{L}{2R} \\ \frac{\cos\theta}{R} & \frac{\sin\theta}{R} & -\frac{L}{2R} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (20)$$

Finalmente. se obtienen las ecuaciones para sacar la velocidad de cada rueda  $\omega_r$  y  $\omega_l$ , ecuaciones 21 y 22.

$$\omega_r = \frac{2(\dot{x}\cos\theta + \dot{y}\sin\theta) + L\dot{\phi}}{2R} \quad (21)$$

$$\omega_l = \frac{2(\dot{x}\cos\theta + \dot{y}\sin\theta) - L\dot{\phi}}{2R} \quad (22)$$

En los cálculos se puede trabajar con las ecuaciones de interés. Primero, es necesario conocer  $\dot{x}$ ,  $\dot{y}$ ,  $\dot{\phi}$  que son las posiciones deseadas a las que debe llegar el robot y, posteriormente se pueden obtener con las ecuaciones 23, 24 y 25.

$$\dot{x} = \frac{R\omega_r \cos\theta + R\omega_l \cos\theta}{2} \quad (23)$$

$$\dot{y} = \frac{R\omega_r \sin\theta + R\omega_l \sin\theta}{2} \quad (24)$$

$$\dot{\phi} = \frac{R\omega_r + R\omega_l}{L} \quad (25)$$

Una vez que se tienen estos valores, son sustituidos en las ecuaciones obtenidas con la pseudoinversa, es decir, las ecuaciones 26 y 27.

$$\omega_r = \frac{2(\dot{x}\cos\theta + \dot{y}\sin\theta) + L\dot{\phi}}{2R} \quad (26)$$

$$\omega_l = \frac{2(\dot{x}\cos\theta + \dot{y}\sin\theta) - L\dot{\phi}}{2R} \quad (27)$$

De esta forma, se conoce el valor de las velocidades de las ruedas y las velocidades que debemos ingresar al robot. Por lo tanto, para saber su velocidad lineal total y su velocidad angular total del robot, se utilizan las ecuaciones de cinemática directa. Donde, se deben de ingresar los valores obtenidos de las ecuaciones 26 y 27.

### Descripción de los códigos de programación con cinemática inversa y cinemática directa

Para comenzar a obtener los valores que caracterizan al robot Krypton 1, es necesario analizar su estructura. La cual está conformada por un PLC, dos motores, un sistema de transmisión, dos llantas fijas y una llanta caster (Figura 2).



*Fuente: elaboración propia*

Figura 2 Kit Krypton 1 en su forma de robot móvil diferencial.

En particular, los dos motores son parte del kit de Krypton y se trata de dos motoredutores alimentados por el PLC con una sola velocidad. Donde, su

funcionamiento está basado en un control on/off programado desde la aplicación Abilix. Esta aplicación, se instala en la tableta y activa o desactiva los motores por medio de órdenes ejecutadas desde el PLC. Por lo tanto, como el control es on/off, los parámetros de las llantas serán constantes.

Radios de las ruedas:	Velocidades de las ruedas:
$R_r = R_l = 35.5 \text{ mm} = 0.0355 \text{ m}$	$v_r = 80 \text{ rpm} = 8.3775802 \text{ rad/s}$
$L = 106. \text{ mm} = 0.1063 \text{ m}$	$v_l = 80 \text{ rpm} = 8.3775802 \text{ rad/s}$

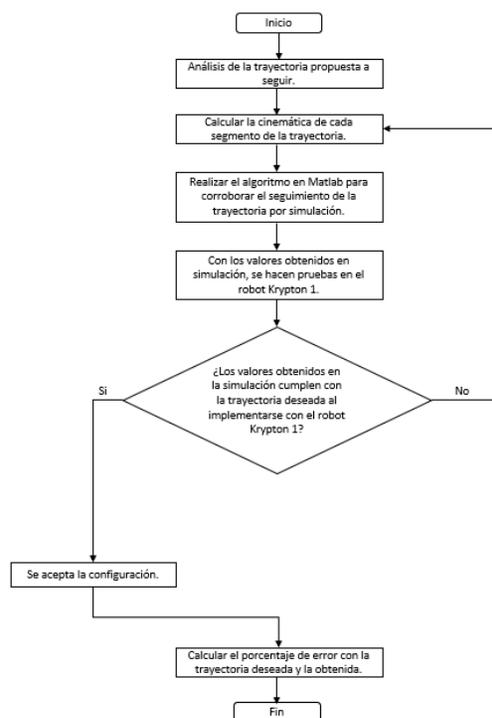
Velocidad lineal del robot:

$$v = 0.2974041046 \text{ m/s}$$

Y velocidad angular del robot:

$$\omega = 5.59556 \frac{\text{rad}}{\text{s}}$$

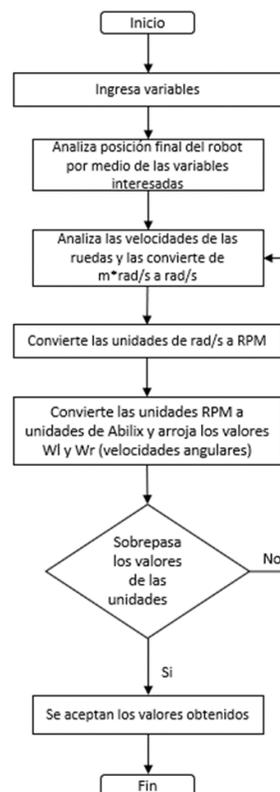
Adicionalmente, para comprobar estos movimientos se realizarán simulación en Matlab. Por lo tanto, los algoritmos en esta aplicación deben de respetar la cinemática calculada anteriormente. Para lograrlo, se desarrolló un diagrama de flujo que explica el comportamiento que debe de seguir el robot (Figura 3).



Fuente: elaboración propia

Figura 3 Diagrama de flujo de la cinemática directa.

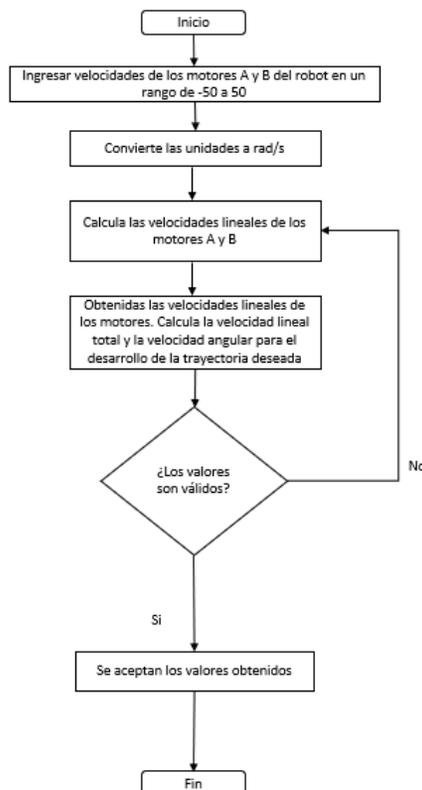
En el algoritmo de cinemática directa se deben ingresar un valor de ángulo del robot y la equivalencia de los RPM. Por medio de estas entradas se encontrarán las velocidades angulares de los motores del robot. Posteriormente, se deben realizar conversiones para obtener las unidades que sean aceptadas en el segundo algoritmo de cinemática inversa. Obteniendo las velocidades angulares del robot se puede proceder a la realización del segundo algoritmo. Este nuevo programa analiza la cinemática directa, esto con el fin de obtener las velocidades lineal y angular totales necesarias para un buen desarrollo de los movimientos del robot. En las entradas del segundo algoritmo se toman en cuenta el radio de las ruedas, la distancia entre ellas, el parámetro ( $L$ ) y los valores de las velocidades angulares obtenidas en el algoritmo anterior. Además, se debe tomar en cuenta el rango de -50 a 50 grados en las vueltas de en nuestra aplicación de Abilix. Hecho esto, se realiza la conversión de unidades de radianes a rpm. En la figura 4 se aprecia un diagrama de flujo con el procedimiento.



Fuente: elaboración propia

Figura 4 Diagrama de flujo para la conversión entre unidades de velocidad.

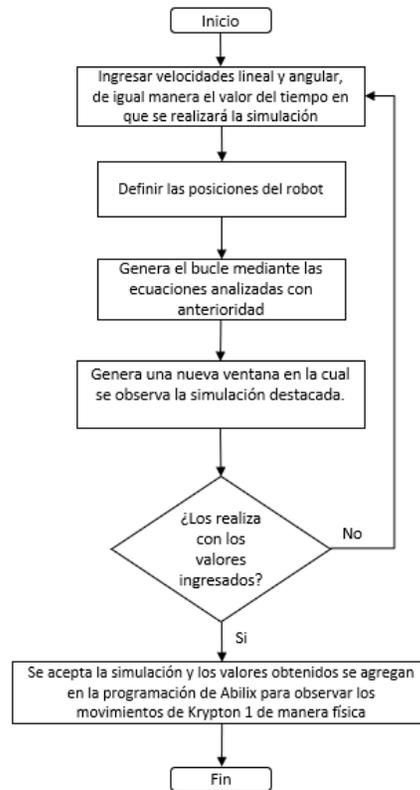
El tercer código de programación desarrollado funciona con la integración de los coeficientes obtenidos en el cálculo de las cinemáticas. Este proceso se realiza por medio de Matlab, asignando el valor de tiempo en el cual se realizará la simulación. Ingresados los coeficientes obtenidos y el tiempo específico, posteriormente se definen las posiciones iniciales del robot. En el algoritmo de la figura 5, se realiza un bucle para la simulación en donde se agrega el modelo de las ecuaciones obtenidas en la cinemática inversa, recordando que la idea principal es verificar la posición y orientación del robot. Cabe mencionar que la naturaleza de la trayectoria deseada depende de los valores obtenidos en los algoritmos anteriores, esto quiere decir que, si se desea una trayectoria lineal, la velocidad angular total deberá ser cero y la velocidad lineal un valor diferente a cero. Por otro lado, para un giro sobre su propio eje debe ser lo contrario, la velocidad lineal tendrá un valor de cero y, por último, para un movimiento en curva el valor de la velocidad lineal total debe ser mayor a la velocidad angular total.



Fuente: elaboración propia

Figura 5 Diagrama de flujo de la cinemática inversa.

En la figura 6 se muestra un diagrama de flujo en donde se explica el funcionamiento más a detalle del tercer algoritmo desarrollado.



Fuente: elaboración propia

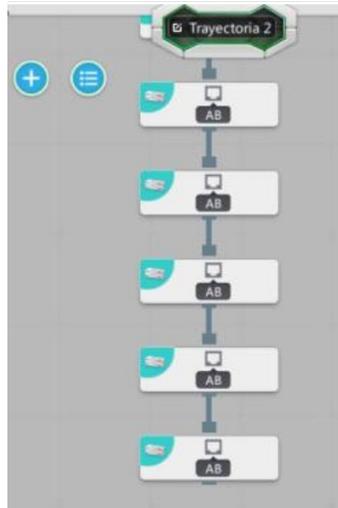
Figura 6 Diagrama de flujo del tercer algoritmo para realizar la simulación.

La programación en el software Abilix se puede realizar mediante: arrastrar y soltar, scratch y flow-chart [Gomes, 2020]. Para este proyecto, los algoritmos anteriormente descritos se realizaron por medio de flow-chart, el cual trabaja por medio de diagramas de flujo secuenciales. En adición, otra ventaja que ofrece esta plataforma es que el software de Abilix se puede instalar en el celular y sólo es necesario seleccionar el tipo de bloque que será introducido en el diagrama de flujo. En la figura 7 se muestra un ejemplo de la programación realizada en flow-chart.

### 3. Resultados

Mediante la aplicación Matlab fue posible la programación de un algoritmo de cinemática directa. En él, se traducen las velocidades que maneja el robot Abilix

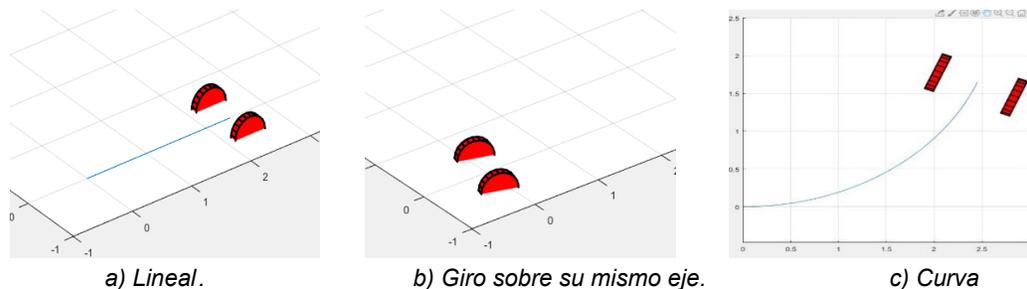
Krypton 1 y se calculan las velocidades necesarias en cada eje, así como su posición. Después de integrar dichas velocidades, se obtuvieron los valores necesarios para programar en el lenguaje Abilix el robot Krypton 1. Dentro de este software se introdujeron los valores de velocidad y tiempo.



Fuente: elaboración propia

Figura 7 Bloques utilizados para programar en flow-chart.

En las imágenes de figura 8, se pueden apreciar las distintas trayectorias realizadas en la simulación: una trayectoria lineal, un giro sobre su propio eje, y un movimiento en curva. Una vez realizadas las simulaciones necesarias, se procedieron a ingresar las velocidades obtenidas en el lenguaje de programación de Abilix. La finalidad fue el comprobar que las simulaciones estuvieran en lo correcto y, el robot Krypton 1 realizara los mismos movimientos que dicha simulación.



Fuente: elaboración propia

Figura 8 Simulación en Matlab con trayectoria lineal, giro sobre su eje y curva hacia la izquierda.

Por otro lado, se propuso una pista de competencia WER donde fueron seleccionadas tres trayectorias (Figura 9). En cada estación se probaron los valores arrojados por el algoritmo programado en Matlab. Donde, los valores fueron introducidos mediante la aplicación de Abilix en un lenguaje de chart, arrastrando bloques que activan uno o ambos motores, dependiendo de la velocidad y tiempo ingresados.



Fuente: elaboración propia

Figura 9 Pista de competencia WER seleccionada.

Una vez establecidas las trayectorias de la pista a programar en el algoritmo, fueron ingresados los valores de velocidad lineal y angular del robot. Posteriormente, se obtuvieron los siguientes resultados (Tabla 1) para la trayectoria 1.

Tabla 1 Tabla de valores de programación para trayectoria 1.

Movimiento	MOTOR	Segundos (s)	Velocidad (v)	Dirección	Trayectoria avanzada (cm)	Coordenadas
1	A	3 s	33	Lineal	60 cm	x=0, y=60
	B	3 s	-33			
2	A	2 s	15	Curva 90° Derecha	Curva	x= -15, y=75
	B	2 s	-32			
3	A	1 s	30	Lineal	30 cm	x= -45, y=75
	B	1 s	-30			
4	A	1 s	18	Curva 45° Derecha	Curva	x= -55, y=73
	B	1 s	-28			
5	A	1 s	16	Giro 90° Izquierda	Giro sobre su propio eje	x= -55, y=73
	B	1 s	10			
6	A	1 s	35	Lineal	23 cm	x= -75, y=95
	B	1 s	-35			

Fuente: elaboración propia

Ya programados los valores de tiempo y velocidad mostrados en la tabla 1. se obtuvo la trayectoria mostrada en la figura 10. Posteriormente, se implementó un marcador al robot, para mostrar los movimientos que seguía el robot y comprobar su correcto avance con líneas en el piso.



Fuente: elaboración propia

Figura 10 Trayectoria 1 realizada por el robot Abilix Krypton 1.

En la tabla 2 se puede apreciar la diferencia de errores con un número de 10 iteraciones. En ella se puede ver que los resultados tienen un error menor a 5%.

Tabla 2 Tabla de errores calculados en la trayectoria 1.

TRAYECTORIA 1						ERRORES							
Bloques	MOTOR	TIEMPO (s)	VELOCIDAD (v)	DIRECCIÓN	TRAYECTORIA AVANZADA (cm)	INTENTOS					PROM	ERROR CALCUL	VALOR ABS
						1	2	3	4	5			
1	A	3s	33	Lineal	60 cm	90	94	94	90	90	45.8	0	0
	B	3s	-33										
2	A	2s	15	Curva 90 Derecha	Curva	70	65	45	90	91	36.1	0.00599	0.00599
	B	2s	-32										
3	A	1s	30	Lineal	30 cm	90	90	95	90	90	45.5	0.00494	0.00494
	B	1s	-30										
4	A	1s	18	Curva 45 Derecha	Curva	45	45	45	45	40	22	0.00511	0.00511
	B	1s	-28										
5	A	1s	16	Giro 90 Izquierda	Giro sobre su propio eje	60	65	58	50	30	26.3	0.00416	0.00416
	B	1s	10										
6	A	1s	35	Lineal	23 cm	90	90	90	90	90	45	0.005	0.005
	B	1s	-35										

Fuente: elaboración propia

Este resultado es satisfactorio, ya que después de implementar la cinemática se pudo notar que los errores disminuyeron hasta un 50% en la programación de Abilix. Dicha comparación, se puede notar entre la tabla 1 y la tabla 3.

En un robot de configuración diferencial es crítico controlar la velocidad de cada rueda, pues se presentan variaciones de las velocidades debido a fricciones, deslizamientos o las perturbaciones de los motores. En este sentido, se logra observar que el robot sigue la trayectoria logrando tener un error cercano a cero.

La figura 11 presenta las simulaciones para la trayectoria 1, realizadas mediante el algoritmo calculado por cinemática directa presentadas en el software de Matlab.

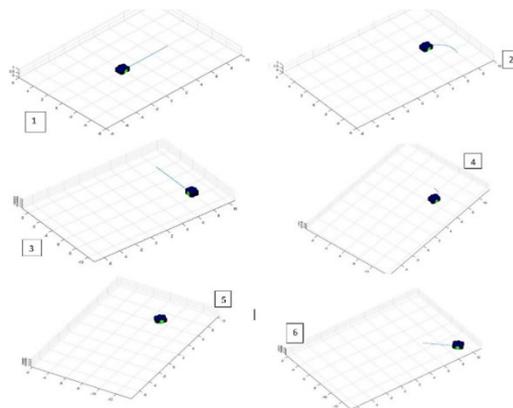
Por último, se obtiene como respuesta un comportamiento en el cual el robot cumple su misión de seguir la trayectoria planteada, realizando un movimiento suave y

efectivo pues logra sobrepasar las perturbaciones en un tiempo reducido. Se ve que el robot logra un error cercano a cero y sigue la referencia de manera correcta, dando como resultado un sistema de control que llega a la meta propuesta por la cinemática inversa.

Tabla 3 Tabla de valores de la trayectoria 1 prueba experimental.

Movimiento	Motor	Segundos (s)	Velocidad (v)	Dirección	Trayectoria avanzada
1	A	3 s	+27 v	Lineal	60 cm
	B	3 s	-27 v		
2	A	1 s	+20 v	Lineal	15 cm
	B	1 s	-20 v		
3	A	1 s	-5 v	Giro	
	B	1 s	-5 v		
4	A	1 s	+12 v	Lineal	5 cm
	B	1 s	-12 v		
5	A	1 s	-5 v	Giro	
	B	1 s	-5 v		
6	A	1 s	+12 v	Lineal	5 cm
	B	1 s	-12 v		
7	A	1 s	-5 v	Giro	
	B	1 s	-5 v		
8	A	3 s	+27 v	Lineal	60 cm
	B	3 s	-27 v		
9	A	1 s	-5 v	Giro	
	B	1 s	-5 v		
10	A	1 s	+8 v	Lineal	5 cm
	B	1 s	-8 v		
11	A	1 s	+12 v	Giro	
	B	1 s	+12 v		
12	A	2 s	+28 v	Lineal	35 cm
	B	2 s	-28 v		

Fuente: elaboración propia



Fuente: elaboración propia

Figura 11 Trayectoria 1 simulada mediante el algoritmo realizado en Matlab.

## **4. Discusión**

Con los manuales proporcionados por Abilix se construyó satisfactoriamente el robot Krypton 1 con una configuración diferencial. Mediante el cual, se realizaron una serie de programaciones empíricas y experimentales a través del software Abilix Krypton. Donde, se tuvo como finalidad la planeación de trayectorias con el algoritmo generado mediante cinemática directa.

Por otro lado, mediante la aplicación de ecuaciones matriciales fue posible desarrollar la cinemática del robot diferencial. Con la cual, se obtuvieron las velocidades en los ejes  $X$ ,  $Y$  y  $\varphi$ , al ingresar como entradas las velocidades angulares de las llantas. Con ello, se implementó un algoritmo mediante programación para obtener la posición del robot con el uso de integración. Sin embargo, para la creación del algoritmo de cinemática inversa se incluyen las restricciones holonómicas, las cuales, impiden que se tenga sólo una solución de posiciones para alcanzar el punto final asignado para una trayectoria. Por lo tanto, este problema de infinitas soluciones deja como opción, generar una secuencia de movimientos directos alternados con giros sobre su mismo eje.

Al final, usando el algoritmo desarrollado de cinemática directa en el software Matlab, se plantearon las posibles soluciones de 3 trayectorias. Posteriormente se hizo simulación en el mismo software, y fueron aplicadas en el robot Krypton 1. Por último, el robot las realizó en la pista de competición WER satisfactoriamente, con error menor a 5%. En este sentido, se ejecutaron distintas pruebas de las 3 trayectorias seleccionadas, variando valores de entrada en el algoritmo, para optimizar la programación y, el traslado del robot, en cada una de ellas.

## **5. Conclusiones**

Cumpliendo las reglas de cinemática directa, se obtuvieron resultados satisfactorios para el seguimiento de los puntos finales mediante la construcción de una pista de competencia WER. Además, se lograron completar diferentes estaciones de la pista propuesta, así como el seguimiento para curvas de una manera suave sin necesidad de dividir en secciones. Mediante la programación de la cinemática directa en el software de Matlab se simplificó el proceso de cálculo

para mejorar los tiempos del usuario con una interfaz más amigable. Con el procesador Abilix, también se redujo significativamente la cantidad de bloques en lenguaje chart. Por último, cabe mencionar que la programación realizada en este proyecto es una buena alternativa para mejorar el rendimiento de los competidores que usan los kits Krypton.

## **6. Bibliografía y Referencias**

- [1] González Vaca, J., Caro Peña, C. A., & González, H. (2015). Cinemática inversa de robot serial utilizando algoritmo genético basado en MCDS. *Tecnura*, 19(44), 33-45.
- [2] Gomes, A. S. A., Da Silva, J. F., & Teixeira, L. R. D. L. (2020). Educational robotics in times of pandemic: challenges and possibilities. *In 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, 1-5.
- [3] González, L. H. R., Valencia, J. A., & Montoya, A. (2009). Modelo cinemático de un robot móvil tipo diferencial y navegación a partir de la estimación odométrica. *Scientia et Technica*, 1(41), 191-196.
- [4] Toca, L., Romo, D., Ruiz, C., & Machay, B. (2019). Diseño de robots de Competencia en Categorías Seguidor de Línea y Robot Laberinto. *Nexus Científicos*, 3(2), 28-42.
- [5] V Valencia, A. J., Montoya O., Ríos A., & Hernando L. (2009). Modelo cinemático de un robot móvil tipo diferencial y navegación a partir de la estimación odométrica. *Scientia et technica*, XV (41), 191-196.
- [6] Zapata-Ros, M. (2014). Gestión del aprendizaje en Educación Superior y web social. *Revista de Educación a Distancia (RED)*, 42(2014). 1-17.