



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA



PLATAFORMA MÓVIL OMNIDIRECCIONAL DE CUATRO LLANTAS SUECAS (Mecanum) EN CONFIGURACIÓN “AB”

T E S I S

Que para obtener el título de:

INGENIERO MECATRÓNICO

P R E S E N T A N:

ADÁN ESTEBAN SUÁREZ ARRIAGA

ALINA MARÍA SÁNCHEZ BALPUESTA

DIRECTOR DE TESIS

DR. VICTOR JAVIER GONZÁLEZ VILLELA

Agradecimientos

Esteban:

A Dios por permitirme obtener un logro más en mi vida profesional, por su amor eterno. A mis padres por su amor y cuidado durante mi niñez, por su apoyo y respeto a mis decisiones. A mis hermanos Araceli, Sergio, Roberto y Miguel por su apoyo incondicional. A mis tíos por soportarme durante tanto tiempo, gracias por su amor, cariño y paciencia. A mi amada UNAM por todo lo que me ha brindado. Al Doctor Víctor Javier González Villela por su apoyo y paciencia durante la realización de este proyecto.

Alina:

A Dios que me ha dado todo, unos padres que me han inculcado sus valores éticos y morales, dos excelentes hermanos Omar y Samuel, a Esteban el mejor de los amigos, a mi amada UNAM que me ha hecho crecer en conocimientos y sobre todo me dio un fabuloso equipo para trabajar, el cual es dirigido por el Doctor Víctor Javier González Villela, a quien agradezco por haberme brindado esta oportunidad.

Introducción

Una de las ramas de la ingeniería moderna que más se ha desarrollado en la actualidad es la robótica. Gracias al avance vertiginoso del conocimiento en esta área de la ciencia y de la técnica se ha podido construir robots cada vez más sofisticados y con una mayor precisión en sus movimientos. Actualmente se tienen robots diseñados para mantenerse estáticos en laboratorios y en la industria, y otros se encuentran explorando el espacio y otros planetas.

Para lograr todo esto, no solo se requieren componentes mecánicos y electrónicos ensamblados, que respondan a nuestras diversas y cambiantes necesidades, también es necesario contar con los modelos matemáticos que nos permitan controlar los diversos sistemas y sean el fundamento en la toma de decisiones.

Este trabajo, pretende agregar un mayor conocimiento al desarrollo de la robótica móvil, en la actualidad, existen robots omnidireccionales utilizados en la industria, en la cual se requiere de mayor precisión y control de sus procesos de producción, por lo que deben ser manipulados con la mayor seguridad posible por el ser humano.

Las ecuaciones presentadas en el capítulo cuatro, acerca de la cinemática de este tipo de robots, permite tener un mejor control y comprensión del funcionamiento de los mismos.

Para probar la validez de las ecuaciones desarrolladas por el Dr. V. J. González Villela, se realizaron una serie de pruebas que permitieron evaluar el error tanto en el desplazamiento lineal como en el angular. Gracias a los resultados de las diversas pruebas se ha comprobado que la cinemática desarrollada es correcta y que un robot omnidireccional con Ruedas Mecanum es completamente controlable.

Esperamos que este trabajo sea útil para proseguir en el avance del conocimiento científico y tecnológico de la robótica móvil y ayude a generar nuevos y mejores prototipos, o ampliar el ámbito de aplicación de los robots omnidireccionales en diferentes áreas de la industria y la tecnología.

Objetivo General:

Comprobar la validez de la cinemática desarrollada para un robot móvil de cuatro ruedas Mecanum, mediante la utilización de campos potenciales, y así demostrar la omnidireccionalidad mediante pruebas donde se manipulen los tres grados de libertad que existen en el plano.

Objetivos Particulares:

- ✓ Evaluar las ecuaciones cinemáticas para un robot móvil con ruedas Mecanum tipo "A" y tipo "B".
- ✓ Utilizar un controlador basado en campos potenciales para comprobar la validez de la cinemática.
- ✓ Analizar el error de posición del robot móvil, tanto en lazo abierto como en lazo cerrado para conocer la disminución del error.

Índice

Agradecimientos

Introducción

Objetivos

1. Robots Móviles	7
1.1 Definición y estructura de un robot móvil.....	7
1.2 Clasificación de los robots móviles.....	8
1.2.1 Robots móviles con ruedas.....	9
1.2.1.1 Tipos de ruedas.....	10
1.2.1.1.1 Rueda Fija.....	12
1.2.1.1.2 Rueda orientable centrada.....	12
1.2.1.1.3 Rueda orientable descentrada (Castor).....	13
1.2.1.1.4 Rueda omnidireccional o sueca.....	13
1.2.1.2 Restricciones de movilidad en robots.....	14
1.2.1.3 Modelo cinemático de posición.....	16
1.2.2 Modelado de robots móviles con ruedas.....	17
1.2.2.1 Modelo cinemático de un Robot móvil diferencial.....	20
1.2.2.2 Modelo cinemático de un Robot Omnidireccional con tres ruedas suecas.....	21
1.2.2.3 Modelo dinámico de posición.....	22
2. Robots Móviles Omnidireccionales	23
2.1 Historia de la rueda sueca y omnidireccional.....	23
2.2 Definición y tipos de ruedas suecas.....	24
2.3 Configuraciones de robots con ruedas suecas.....	25
2.3.1 Configuración triangular con tres ruedas omnidireccionales.....	25
2.3.2 Configuración de cuatro ruedas Mecanum.....	25
2.3.3 Otras configuraciones con ruedas Mecanum.....	26
2.3.4 Configuración de cuatro ruedas suecas tipo "AB" y "BB"	26
3. Diseño del Robot Móvil Omnidireccional usado en las pruebas	28
3.1 Arquitectura.....	28
3.2 Plataforma móvil omnidireccional.....	28
3.3 Control Electrónico.....	33
4. Modelo cinemático y control del robot móvil con ruedas Mecanum	37
4.1 Plataforma móvil omnidireccional con ruedas suecas.....	41
4.1.1 Propagación de velocidades.....	43
4.1.2 Restricciones cinemáticas.....	45

4.1.3	Coordenadas generalizadas de configuración.....	48
4.2	Control por Campos potenciales.....	49
5.	Resultados de las pruebas con el robot móvil omnidireccional	52
5.1	Resultado de la prueba en lazo abierto.....	53
5.1.1	Movimiento del punto de análisis de la plataforma en línea recta diagonal...	53
5.2	Resultado de las pruebas en lazo cerrado.....	59
5.2.1	Seguimiento de una trayectoria rectilínea del punto P de la plataforma con orientación de cero grados.....	59
5.2.2	Seguimiento de una trayectoria circular del punto P de la plataforma con una velocidad angular ω	65
5.2.3	Seguimiento de Curva de Lissajous denominada "Kiss" con velocidad angular ω	71
5.2.4	Seguimiento de Curva de Lissajous denominada "Pop" con inclinación de 20° respecto al marco global.....	76
6.	Conclusiones y Trabajo a Futuro	80
6.1	Conclusiones.....	80
6.2	Trabajo a futuro.....	82
	Apéndices	86
	Referencias	100

Capítulo 1. Robots Móviles

La robótica es una técnica de aplicación de la informática, actualmente diseña y produce aparatos en diversos campos del conocimiento humano con el fin de sustituir a las personas en diferentes operaciones, principalmente las que son rutinarias, tanto en instalaciones industriales como la automotriz, en la medicina con la teleoperación, en los juguetes, o en los concursos escolares, se puede decir que dependiendo el área en la cual se requiera, ya sea por lo peligroso, aburrido, difícil de realizar o simplemente por pasatiempo; es una rama de estudio a la cual le queda mucho por hacer, pues se vive en una época en la que se está desarrollando de manera vertiginosa.

Los robots móviles se crean con muy diferentes propósitos, dependiendo de la función que se quiera o necesite realizar, ya sean grandes para poder cargar y manipular objetos pesados, como en la industria, o muy pequeños para procesos específicos como los drones o micro robots. A pesar de su rápido avance existen algunas restricciones, en el mercado a veces es muy difícil adquirir algunos elementos, debido a que siguen en fase experimental o están en proceso de desarrollo.

1.1 Definición y estructura de un robot móvil

Un robot móvil es "un vehículo que posee un sistema de navegación automática que le permite trasladarse en un ambiente determinado, donde se incluyen tareas de planificación, percepción y control" [1,2].

Se sabe que los robots móviles tienden a ser construidos con estructuras similares a los seres vivos, para esto podemos observar en la figura 1.1 las similitudes que hay entre los elementos que conforman a un robot móvil y un ser vivo:

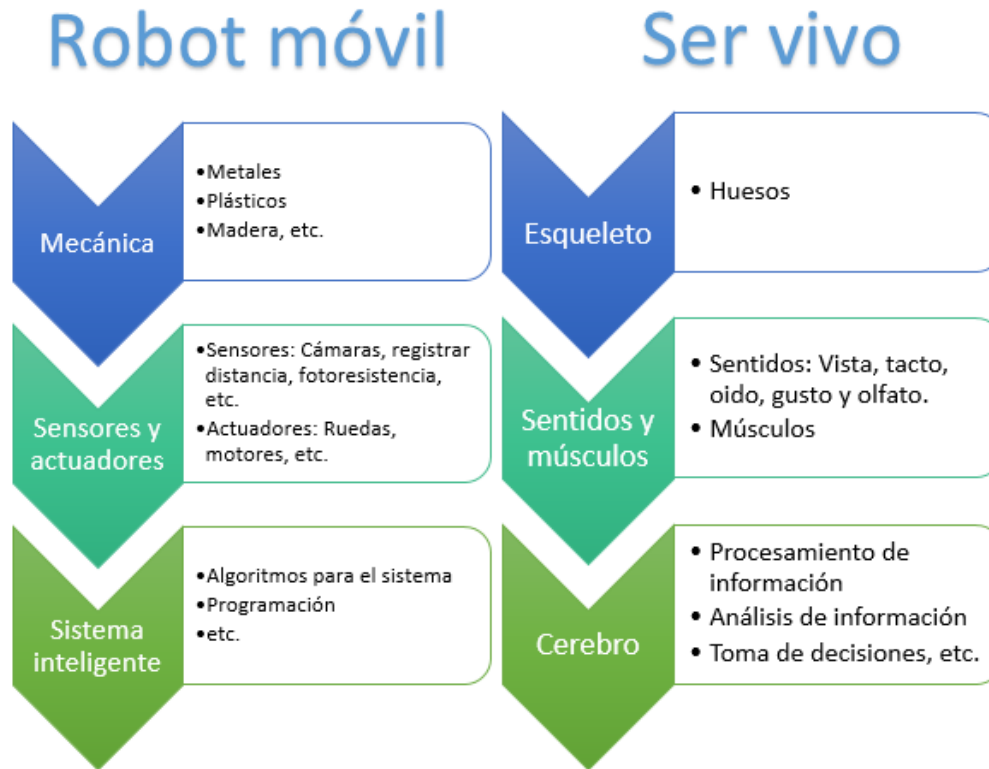


Figura 1.1a Esquema de comparación de robot móvil y un ser vivo [2,3].

1.2 Clasificación de los robots móviles.

En la actualidad existen diversos tipos de robots móviles, aunque no hay una clasificación muy clara, pueden ser ordenados de la siguiente manera:

- Por el medio en el cual se mueven: acuáticos, aéreos y terrestre.
- Por su sistema de locomoción: bípedos, hexápodos, orugas.
- Con ruedas: Ackerman, diferencial síncrono, triciclo y omnidireccional.

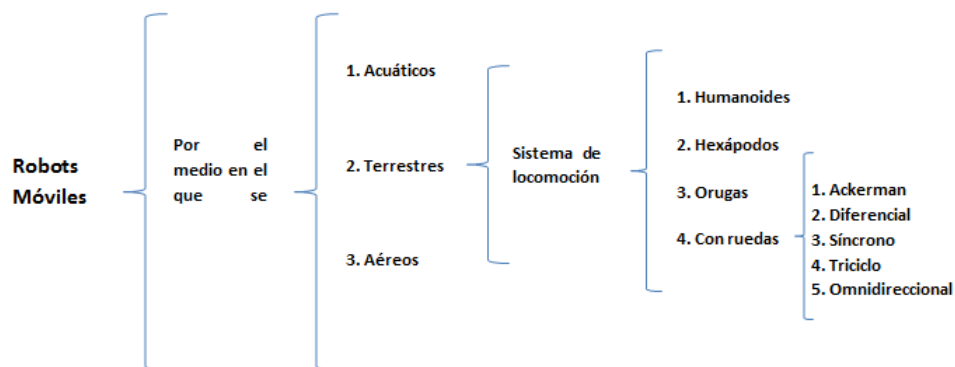


Fig. 1.2a Clasificación de los robots móviles [2,3].

1.2.1 Robots Móviles con Ruedas

El principal problema al que se enfrenta en los robots móviles, se encuentra en el sistema de locomoción y el número y tipo de ruedas que lo integran, ya que estos determinan su desplazamiento de un lugar a otro; puede ser en áreas muy pequeñas o grandes, dependiendo de las condiciones en las que se requiere su operación.

Hay cuatro tipos de ruedas que al combinarse, le confieren los diferentes tipos de desplazamiento, cada sistema de locomoción tiene su propio análisis cinemático, y por lo tanto, sistemas de ecuaciones muy diferentes.

En el análisis cinemático de los robots con ruedas se hacen las siguientes suposiciones:

1. El robot junto con sus partes forman un solo cuerpo rígido.
2. No hay desplazamiento en la dirección perpendicular a la rodadura.
3. No hay deslizamiento traslacional entre la rueda y el suelo (rotación pura).
4. Como máximo hay un eje de dirección por rueda.
5. Los ejes de la dirección son perpendiculares al suelo.

Cuando se realiza el análisis cinemático de un robot móvil, siempre habrá un punto P que por lo general, está asociado al centroide de la plataforma del robot, donde se coloca un sistema de referencia $(x_m - y_m)$. De esta manera se puede conocer en todo momento la posición y velocidad del robot móvil, al asociar, a su vez, la posición del punto P a un sistema de referencia global $(x_b - y_b)$.

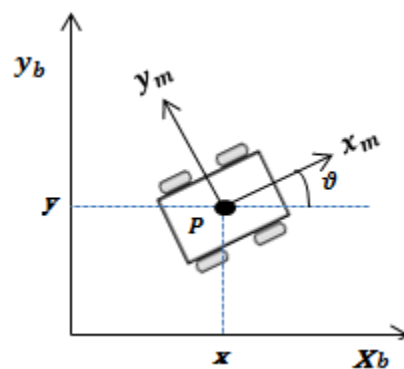


Fig. 1.2b Coordenadas de posición [1,3].

La posición del robot puede ser descrita en términos de las coordenadas; x e y que representan la posición del punto P respecto al marco de referencia global (límite espacial dentro del cual se realiza el movimiento del robot). El punto P se ubica en el origen del marco de referencia ($x_m - y_m$), y a su vez tanto el punto P como el origen de dicho sistema de referencia se colocan en el centroide de la plataforma. La orientación de este punto está descrita por ϑ . Se utiliza el siguiente vector para describir la posición del punto P ;

$$\xi = \begin{pmatrix} x \\ y \\ \vartheta \end{pmatrix} \quad (1.1)$$

Y la matriz de rotación que permite conocer la orientación del sistema de referencia móvil respecto al sistema de referencia fijo.

$$R(\vartheta) = \begin{pmatrix} \cos\vartheta & \sin\vartheta & 0 \\ -\sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.2)$$

1.2.1.1 Tipos de ruedas

La mayoría de los robots móviles diseñados y estudiados hasta ahora [1, 2, 4], utilizan tres tipos de ruedas y son: con ruedas fijas, con ruedas orientables y con ruedas omnidireccionales. Por lo general se combinan ruedas fijas y orientables, mientras que los robots con ruedas suecas (omnidireccionales), sólo contienen tres ruedas de este tipo. También hay robots móviles que contienen cuatro ruedas [1, 2], donde hay una combinación entre los diferentes tipos. De lo anterior se puede observar lo siguiente:

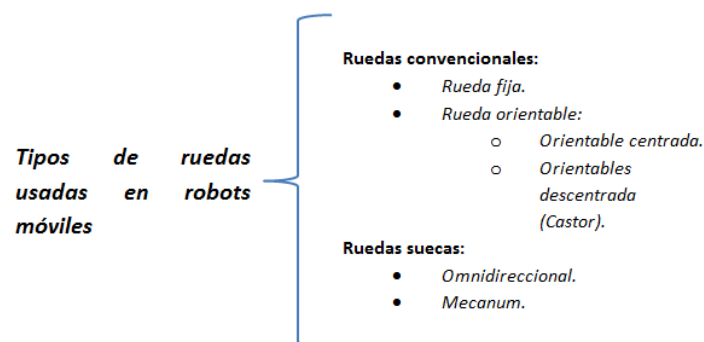


Fig. 1.2c Clasificación de ruedas usadas en robots móviles [1,4].

Para poder describir la posición y orientación del robot respecto a un marco de referencia global en el plano, es necesario contar primero con las restricciones cinemáticas de cada tipo de rueda respecto al sistema de referencia ($x_m - y_m$).

Se define de manera general los siguientes elementos que describen la posición de la rueda en un plano ubicado sobre ella;

- A es un punto fijo en la parte superior de la plataforma del robot, el cual está ubicado en el centro de la banda de deslizamiento de la rueda o llanta.
- l , es la distancia del punto P hasta A .
- α , representa la orientación de A respecto a x_m .
- β , representa la orientación del plano de la rueda respecto a l .
- φ , es el ángulo de rotación de la ruedas a través de su eje horizontal de giro.
- r , es el radio de la rueda, que se supone igual para todas.

A continuación, la figura 1.2d muestra los vectores de velocidad en el punto de contacto con el suelo, referidos a un plano ortogonal al eje de giro. En verde se muestra el vector velocidad de traslación \vec{v}_T , y en naranja el vector de velocidad tangencial \vec{v}_R , debido a la rotación de la rueda sobre su centro de masa.

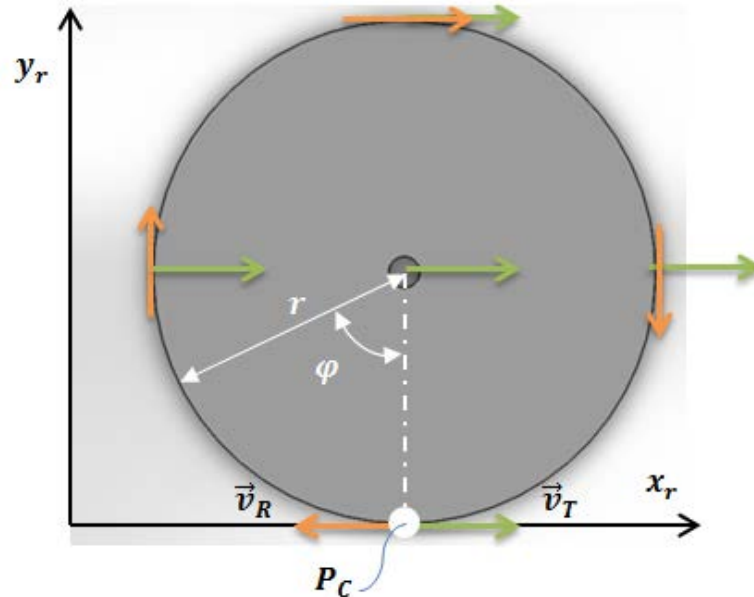


Fig. 1.2d Rueda simple con vectores de velocidad [3].

1.2.1.1.1 Rueda Fija

El centro de la rueda fija A , es un punto fijo de la plataforma donde se monta. El eje de la rueda está fijo a la estructura del robot. La posición del punto A respecto al marco de referencia en el cual se mueve, se expresa por medio de coordenadas polares; donde l es la distancia de A a P y el ángulo α . El ángulo β representa la orientación del plano de la rueda respecto a l . La orientación de la llanta respecto a su eje horizontal (φ). El radio de la rueda siempre estará expresado por r .

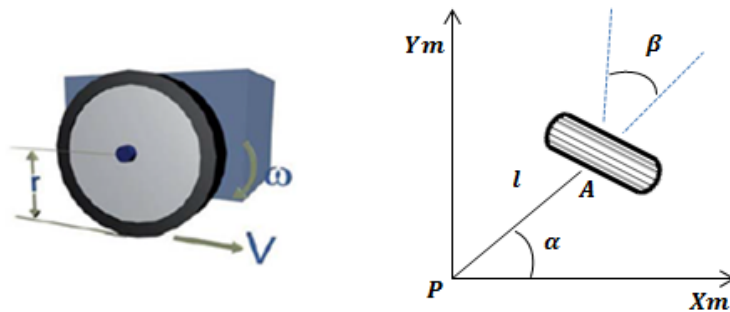


Fig. 1.2e Rueda fija con parámetros respecto al punto P [1,3].

1.2.1.1.2 Rueda Orientable Centrada

En este tipo de rueda el movimiento del plano de la rueda con respecto a la estructura (donde está montada) es una rotación alrededor de un eje vertical que pasa a través del centro de la rueda. Puede funcionar como rueda de tracción-dirección o sólo como de dirección. La descripción de esta rueda es exactamente la misma que para la anterior, a excepción de que el ángulo β es variable respecto al tiempo.

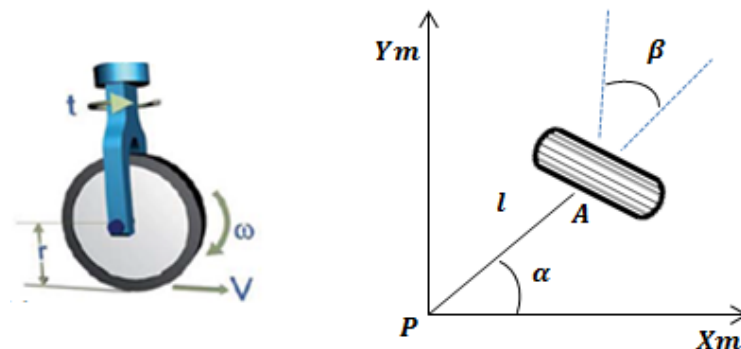


Fig. 1.2f Rueda orientable centrada [1,3].

1.2.1.1.3 Rueda Orientable Descentrada (Rueda de Castor)

Esta rueda es orientable respecto a la estructura del robot, de tal manera que la rotación del plano de la rueda es alrededor de un eje vertical que no pasa a través del centro de esta. El análisis de posición es muy parecido al de las dos anteriores, con la aparición de dos nuevos parámetros; B que representa la posición del centro de la rueda y d que es la longitud de una barra rígida que une A con B . Por lo general, este tipo de rueda tiene como función dar estabilidad a la estructura del robot.

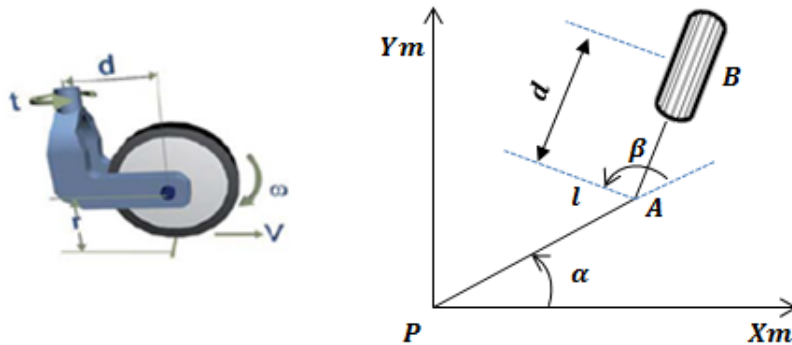


Fig. 1.2g Rueda orientable descentrada [1,3].

1.2.1.1.4 Rueda Sueca

La posición de una rueda sueca, respecto al centroide de la estructura del robot es exactamente igual que para la de una rueda fija, donde los parámetros α , β y l son constantes. Sin embargo, existe otro parámetro necesario para describir la dirección con respecto al plano de la rueda, ya que en el punto de contacto con el suelo, el vector velocidad tiene una componente representada por el ángulo γ .

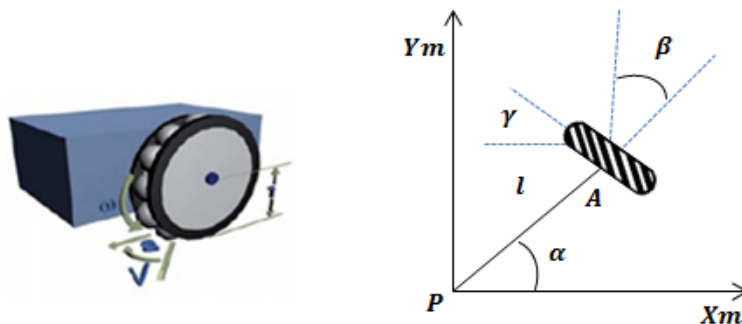


Fig. 1.2h Rueda omnidireccional [1,3].

1.2.1.2 Restricciones de movilidad en un robot

Una vez descritos los diferentes tipos de ruedas, se usan subíndices para indicar su tipo: **(f)** para las ruedas fijas, **(s)** para la rueda orientable centrada, **(c)** para la rueda orientable descentrada o castor y **(sw)** para la rueda sueca.

El número total de ruedas que podemos tener en un robot combinando cualquier tipo de estas queda así:

$$N = N_f + N_s + N_c + N_{sw} \quad (1.3)$$

La configuración de un robot está descrita por los siguientes vectores;

1. Coordenadas de posición: $\xi(t) = (x(t) \quad y(t) \quad \vartheta(t))^T \dots (1.4)$ que describe la posición en el plano.
2. Coordenadas de orientación: $\beta(t) = (\beta_s^T(t) \quad \beta_c^T(t))^T \quad (1.5)$ que describe la orientación de los ángulos de las ruedas orientables centradas y de castor.
3. Coordenadas de rotación: $\varphi(t) = (\varphi_f(t) \quad \varphi_s(t) \quad \varphi_c(t) \quad \varphi_{sw}(t))^T \quad (1.6)$ para los ángulos de rotación de las ruedas respecto a su propio eje horizontal.

Al combinar estos factores como lo muestra [1,4], se obtienen las restricciones para un robot móvil, que a su vez tienen que ver con la maniobrabilidad, la cual depende de la movilidad que permite el tipo de rueda y la movilidad debida al cambio de dirección de las ruedas orientables. Esto puede ser expresado por medio de dos parámetros:

1. δ_m que representa el grado de movilidad y,
2. δ_s la cual representa el grado de direccionabilidad.

Así, la maniobrabilidad para un robot queda expresada como:

$$\delta_M = \delta_m + \delta_s \quad (1.7)$$

Cuando el robot cuenta con ruedas orientables o fijas, se puede notar geoméricamente que existe un Centro Instantáneo de Rotación (ICR por sus siglas en inglés). Por lo que al prolongar los ejes de giro de cada rueda, estos se intersectan en un punto (ICR) tal como lo muestra la siguiente figura:

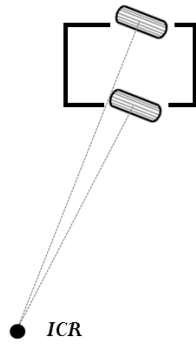


Fig. 1.2i Centro instantáneo de rotación.

De esta manera, el robot describe un movimiento rotacional que varía respecto al tiempo, alrededor de este punto. Este concepto es importante porque de ello dependerá el grado de movilidad para robots con ruedas fijas y orientables.

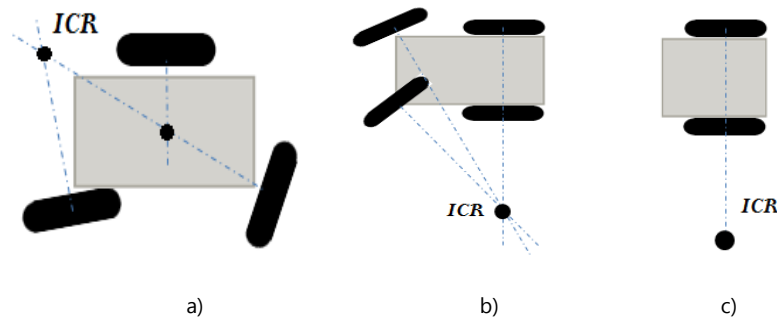


Fig. 1.2j a) Grado de movilidad 0, b) grado de movilidad 1, c) grado de movilidad 2 [1, 3, 4]

El grado de direccionabilidad depende del número de ruedas orientables centradas que pueden moverse de forma independiente para cambiar la dirección del robot.

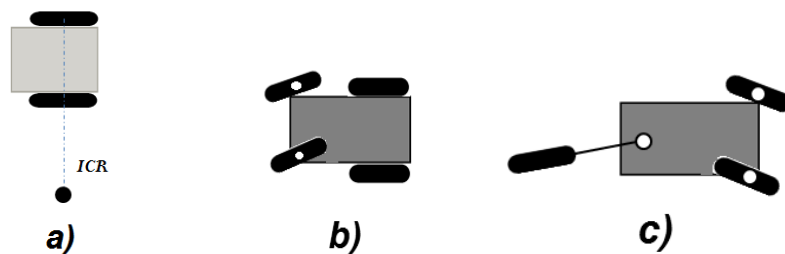


Fig. 1.2k a) Grado de direccionabilidad 0, b) Ruedas orientables dependientes con grado de direccionabilidad 1, c) ruedas orientables independientes con grado direccionabilidad 2 [1, 3, 4].

En base a lo anterior, se muestra la siguiente tabla donde se observan las diferentes combinaciones entre δ_m y δ_s para los robots móviles existentes hasta ahora:

Tabla 1.1 Combinación entre grados de movilidad y direccionalidad [1, 3, 4].

Tipos de robot					
δ_m	3	2	2	1	1
δ_s	0	0	1	1	2

1.2.1.3 Modelo cinemático de posición

Una consecuencia directa de δ_m es la facilidad para poder elegir el origen y orientación del marco de referencia que se ubica sobre el robot, ya que este cambia de posición respecto al tiempo y referido a sí mismo, como lo muestra la figura 1.2l. Específicamente, esto se conoce como modelo cinemático de posición.

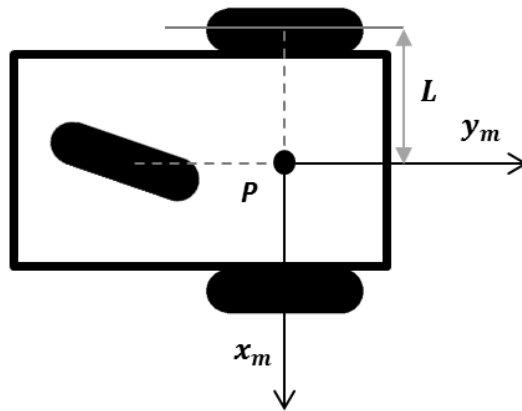


Fig. 1.2l Asignación de punto de análisis y sistema de referencia [1, 3, 4].

Si pensamos en el robot moviéndose, podemos intuir que existe un vector variable en el tiempo $\boldsymbol{\eta}(t)$, este vector tiene que ver directamente con el grado de movilidad del robot respecto a sí mismo, es decir $\delta_m = \boldsymbol{\eta}$;

$$\dot{\boldsymbol{\xi}} = \mathbf{R}^T(\boldsymbol{\vartheta})\boldsymbol{\Sigma}(\boldsymbol{\beta}_s)\boldsymbol{\eta} \quad (1.8)$$

Para robots con ruedas orientables y,

$$\begin{cases} \dot{\boldsymbol{\xi}} = \mathbf{R}^T(\boldsymbol{\vartheta})\boldsymbol{\Sigma}\boldsymbol{\eta} \\ \dot{\boldsymbol{\beta}} = \boldsymbol{\zeta} \end{cases} \quad (1.9)$$

Para robots con $\delta_s \geq 0$. Donde $\delta_s = \zeta$. Otra manera de expresar lo anterior es:

$$\dot{z} = B(z)u \quad (1.10)$$

Con $\delta_s = 0$;

$$z = \xi \quad B(z) = R^T(\vartheta)\Sigma \quad u = \eta$$

Con $\delta_s > 0$;

$$z = \begin{pmatrix} \xi \\ \beta_s \end{pmatrix} \quad B(z) = \begin{pmatrix} R^T(\vartheta)\Sigma(\beta_s) & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix} \quad u = \begin{pmatrix} \eta \\ \zeta \end{pmatrix}$$

Esto tiene una importancia en el control del robot, ya que ζ, η pueden ser elegidas como variables de entrada.

Al analizar la figura 1.2l, es notorio que el robot es del tipo (1,1), según la Tabla 1.1 Por lo tanto, este robot solo tiene una componente de velocidad en el eje y_m , o sea η_1 . Lo cual expresado según (1.5), queda así:

$$\begin{cases} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{pmatrix} = \begin{pmatrix} -L \sin \vartheta \sin \beta_{s3} \\ L \cos \vartheta \sin \beta_{s3} \\ \cos \beta_{s3} \end{pmatrix} \eta_1 \\ \beta_{s3} = \zeta_1 \end{cases} \quad (1.11)$$

1.2.2 Modelado cinemático de robots móviles con ruedas

El modelo cinemático de un robot permite conocer la velocidad y posición de este en cualquier lapso de tiempo. El modelo cinemático de un robot contempla las restricciones de movilidad, el modelo cinemático de posición, y el tipo de locomoción, por lo tanto la holonomía o no-holonomía del robot dependerá de todos estos elementos en conjunto.

Se puede hacer distinción entre la cinemática directa e inversa. La primera se refiere al hecho de que es posible conocer la posición final del robot, dada la posición inicial y los movimientos que realizó. La segunda le indica al robot los movimientos que debe realizar, para llegar a un punto final deseado y la orientación requerida.

Las restricciones de maniobrabilidad en los robots móviles, debido al desplazamiento que tienen hacia todas las direcciones en un plano, pueden ser

resueltas con los robots omnidireccionales, ya que estos se desenvuelven mejor hacia todas las direcciones dentro de un marco de referencia restringido.

Algunas consideraciones para el diseño de un robot móvil son: controlabilidad, maniobrabilidad, fuerza de tracción, eficiencia, estabilidad, mantenimiento, capacidad para subir pendientes, impacto ambiental, entre otras.

Existen diversas formas de locomoción en un robot móvil, algunas de ellas son las siguientes: diferencial, síncronas, Ackerman, ruedas/cintas de desplazamiento, con patas y omnidireccionales.

En la actualidad existen los modelados cinemáticos de varios tipos de robots móviles, tal como lo muestra [1, 4, 5], pero en este trabajo solo se presentan dos modelos cinemáticos, como ejemplo de lo existente hasta ahora.

Hay dos conceptos muy importantes en el diseño de robots móviles cuando se les considera como un sistema que se mueve en el plano:

- 1. Sistema holonómico:** La holonomía en robótica hace referencia a la proporción entre el número de grados de libertad que existen, y los grados de libertad que son controlables. En el plano, se tienen tres grados de libertad; la posición dada por (x, y) y la orientación (θ), respecto a un marco global.
- 2. Sistema no holonómico:** Se refiere a que el robot puede realizar movimientos hacia adelante y atrás, pero no lateralmente, además sus ecuaciones diferenciales no son integrables para la posición final del robot, se necesita conocer el movimiento en función del tiempo, ya que con la trayectoria realizada por la rueda o ruedas no es suficiente.

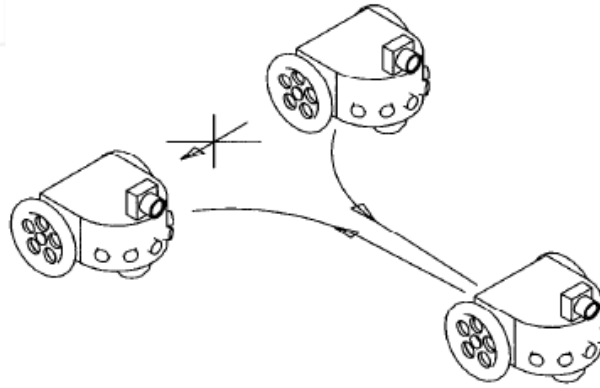


Fig. 1.2m Robot diferencial no holonómico [3, 6].

La figura anterior muestra un robot diferencial, con $\delta_m = 2$ y $\delta_s = 0$. El cual puede desplazarse hacia adelante y atrás, y girar alrededor del ICR, pero no puede hacerlo lateralmente. En este caso, el sistema es no holonómico, ya que solo tiene dos grados de libertad de los tres que existen en el plano.

El modelo cinemático de la posición del robot respecto al tiempo, puede ser reformulado en el espacio de estados [1,4], de la siguiente manera:

$$\dot{\xi} = R^T(\vartheta)\Sigma(\beta_s)\eta \quad (1.12)$$

Donde $\eta(t)$ es el vector que representa los grados de movilidad del robot.

El modelo de la configuración cinemática de un robot móvil muestra la evolución de la orientación y las velocidades de rotación de las llantas [1, 4], lo cual depende de $\dot{\beta}_c$ y $\dot{\varphi}$ que son las derivadas con respecto al tiempo de (1.5) y (1.6). Con estos vectores se define a q un nuevo vector que representa las coordenadas de configuración.

$$q = \begin{pmatrix} \xi \\ \beta_s \\ \beta_c \\ \varphi \end{pmatrix} \quad (1.13)$$

La evolución de estas coordenadas respecto al tiempo quedan expresadas como:

$$\dot{q} = S(q) \cdot u \quad (1.14)$$

Donde $S(q)$ contiene a las derivadas de (1.5), (1.6) y (1.8), mientras que u contiene a η .

1.2.2.1 Robot móvil diferencial

En esta configuración no hay ruedas directrices, el cambio de dirección del robot está dado por la diferencia de velocidades en sus ruedas laterales. La tracción se obtiene con las ruedas mismas, aunque pueden existir una o más ruedas que ayudan a dar mayor estabilidad al robot.

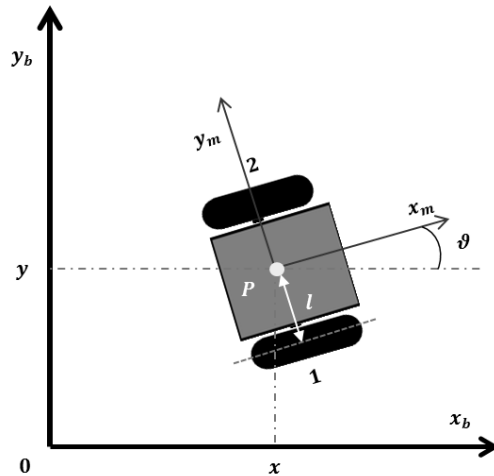


Figura 1.2n Robot diferencial en el plano [3, 4, 7]

Algunas de las ventajas de este tipo de configuración son la simplicidad del diseño, facilidad de implementación y, es un sistema barato. Entre los inconvenientes se tiene que es difícil de controlar y requiere de control de precisión para trayectorias no rectas [7, 8].

Uno de los problemas más relevantes de este tipo de diseño se encuentra en el acomodo de las llantas, ya que puede existir deformación (cambia el diámetro), y como consecuencia se distorsiona el control de la dirección del vehículo.

Su modelo cinemático en el espacio de estados queda de la siguiente manera [7, 8]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r \cdot \cos \theta}{2} & \frac{r \cdot \cos \theta}{2} \\ \frac{r \cdot \sin \theta}{2} & \frac{r \cdot \sin \theta}{2} \\ \frac{r}{2l} & -\frac{r}{2l} \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} \quad (1.15)$$

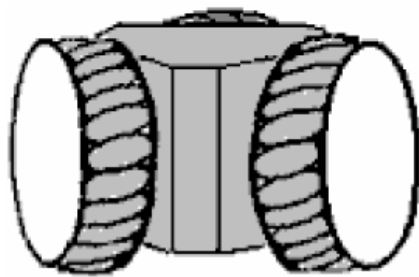
Donde v es la velocidad lineal del robot, ω representa la velocidad angular del robot respecto a si mismo, $\dot{\phi}_1$ y $\dot{\phi}_2$ representan la velocidad angular de la rueda

derecha e izquierda respectivamente, mientras que r es el radio de la rueda y l la distancia del centro de la rueda al marco de referencia ($x_m - y_m$).

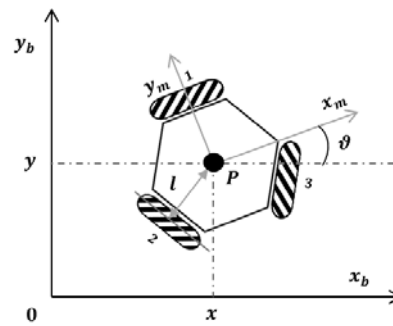
1.2.2.2 Robot Omnidireccional con tres ruedas suecas

El movimiento omnidireccional solo puede efectuarse mediante la instalación de dos ejes, uno vertical y otro horizontal respecto a la superficie de desplazamiento. Mientras que el desplazamiento en el eje y se debe al acomodo de los rodillos sobre la periferia de la rueda, cuyos ejes forman un ángulo de 90° respecto al eje de giro de la rueda.

Los robots de este tipo tienen mayor libertad de movimiento en comparación con los robots que usan ruedas convencionales (fijas, orientables y castor). Esto es debido a que pueden realizar movimientos más complejos con ayuda del control adecuado [9].



a)



b)

Figura 1.2ñ a) Modelo físico del robot omnidireccional, b) Robot omnidireccional en el plano [1, 3, 9]

Se presenta a continuación el modelo cinemático en el espacio de estados de un robot de tres ruedas omnidireccionales [9].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & \frac{r}{\sqrt{3}} & -\frac{r}{\sqrt{3}} \\ \frac{2r}{3} & -\frac{r}{3} & -\frac{r}{3} \\ -\frac{r}{3l} & -\frac{r}{3l} & -\frac{r}{3l} \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} \quad (1.16)$$

1.2.2.3 Modelo dinámico de posición

El modelo dinámico de posición se obtiene al considerar la relación que existe entre **(1.14)** y los torques generados por los actuadores que posee cada rueda según lo muestra [1,4]. Esto quiere decir que al dar un torque a la rueda, se genera una velocidad angular, que a su vez produce una velocidad lineal para cada rueda. Al juntar el efecto de velocidad lineal de cada rueda, dan la velocidad lineal o angular del robot respecto al marco de referencia global, en el cual el robot se está moviendo. Si se considera a **(1.14)** como invariante en el tiempo y con una realimentación estática, lo anterior, se expresa de la manera siguiente:

$$\begin{cases} \dot{q} = S(q)u \\ \dot{u} = v \end{cases} \quad (1.17)$$



Capítulo 2. Robots Móviles Omnidireccionales

2.1 Historia y definición de rueda sueca y rueda omnidireccional

Desde 1907 ya existían inventores que pensaban en diseños que permitieran a los vehículos una movilidad en varias direcciones. Las ruedas omnidireccionales fueron patentadas en 1919 por J. Grabowiecki en Estados Unidos. Su diseño era una rueda principal con pequeños rodillos colocados en su periferia;

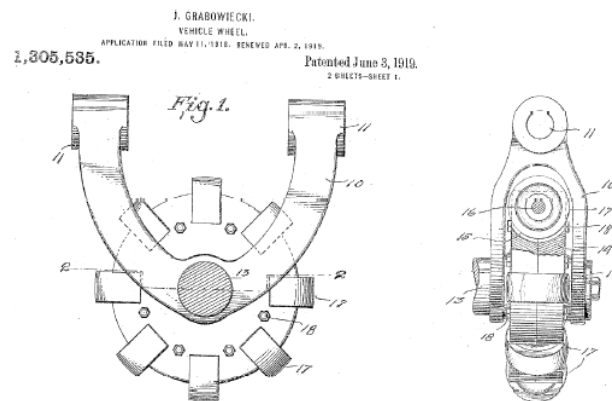


Fig. 2.1a. Diseño patentado por J. Grabowiecki [10].

Alrededor de 1973 el sueco Bengt Ilon desarrolló las primeras ruedas capaces de dar tres grados de movilidad a los vehículos. El diseño fue patentado por la compañía sueca Mecanum AB, estas tienen un perfil casi circular, aunque tienen un inconveniente, que los ejes de los rodillos pequeños tienen una fricción muy alta, generando así fuerzas transversales que impiden un desplazamiento adecuado.

En 1980 Bradbury patentó otro tipo de llantas que básicamente son dos esferas truncadas. Para 1987 ya existía el primer trabajo donde se utilizó un robot con cuatro ruedas Mecanum. En el año 2000 las ruedas patentadas por Ilon se comenzaron a usar en competencias de robótica, lo que dio como resultado que se empezaran a estudiar tanto a robots que utilizan este tipo de ruedas, como a la rueda misma [11].

2.2 Definición y tipos de ruedas suecas

Según lo dice Muñoz Martínez [12], "La rueda omnidireccional se define como una rueda estándar a la cual se le ha dotado de una corona de rodillos, cuyos ejes de giro resultan perpendiculares a la dirección normal de avance", mientras que Diegel [13] menciona que una rueda Mecanum según Ilion se basa en "el principio de una rueda central, con cierto número de rodillos colocados en un ángulo alrededor de la periferia de la rueda".

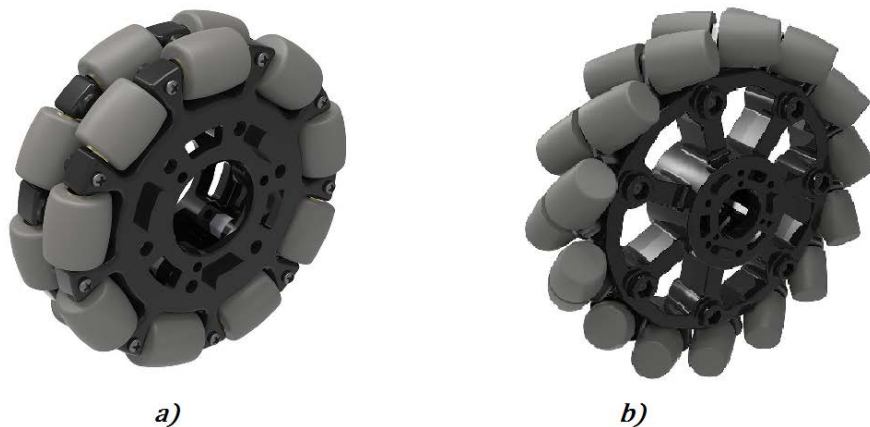


Fig. 2.2a. Imagen de ruedas suecas; a) Omnidireccional, b) Mecanum [14].

De lo anterior se concluye que una rueda sueca, "es aquella rueda que contiene en su periferia una corona de rodillos, colocados a cierto ángulo respecto a su eje de giro", y a su vez, esta se divide en omnidireccional y Mecanum. Lo anterior queda resumido en la figura 2.2b.

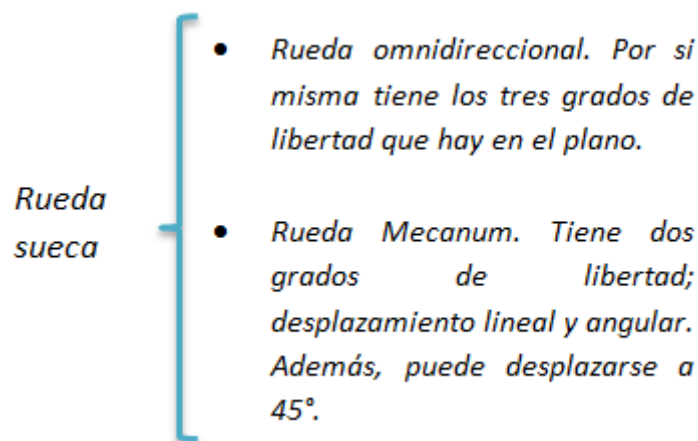


Fig. 2.2b Clasificación de la rueda sueca [3, 12, 13].

2.3 Configuraciones de robots móviles omnidireccionales

Este tipo de ruedas permiten realizar desplazamientos complicados; laterales, diagonales, giros de trescientos sesenta grados, y los típicos movimientos de adelante y atrás, para diferentes configuraciones de las ruedas. Dentro de las configuraciones de este tipo de ruedas se encuentran las siguientes:

1. Configuración de tres ruedas (triangular). En esta configuración se utilizan ruedas omnidireccionales.
2. Configuración de 4, 6 y 8 ruedas Mecanum del tipo "A" y "B".

2.3.1 Configuración triangular con tres ruedas omnidireccionales

Cuando se habla de configuración triangular, nos referimos al hecho de que se tienen solo tres ruedas, las cuales están colocadas de tal manera que forman un triángulo, y estas deben estar posicionadas a 120° una de la otra.

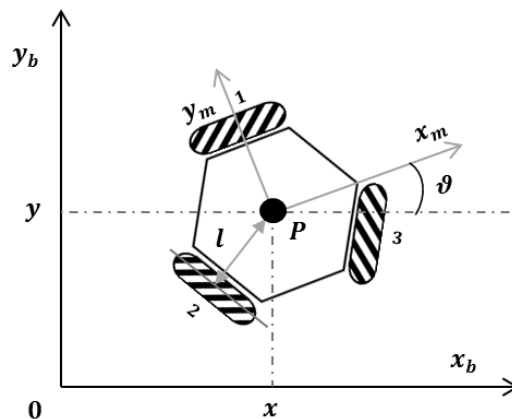


Fig. 2.3.1a. Disposición de ruedas suecas formando un triángulo, en una plataforma móvil [1, 3, 9].

2.3.2 Configuración de cuatro ruedas Mecanum

Este tipo de configuración es muy parecida a la Ackerman ya que los automóviles convencionales la utilizan, solo que en este caso, el eje delantero no gira. Consta de una plataforma rectangular y las ruedas se colocan en los lados paralelos que por lo general son los lados más grandes.

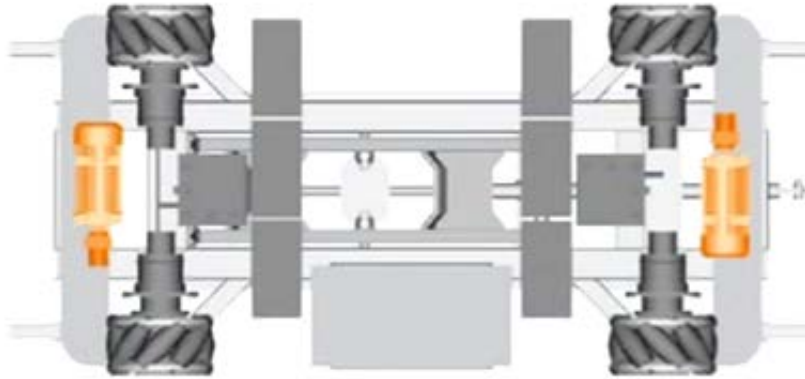


Fig. 2.3.2a. Configuración de cuatro ruedas del KUKA omniMove® [15].

2.3.3 Otras configuraciones con ruedas Mecanum

Se puede encontrar otros tipos de configuraciones utilizadas en la industria, que en realidad son una variación del caso anterior, esto se debe a que solamente se agregan más ruedas en los mismos lados, y se separan a cierta distancia; pueden ser seis ruedas, u ocho en lugar de cuatro. Hasta el año de publicación de este trabajo, aun no existen estudios en robots móviles con más de cuatro ruedas Mecanum.

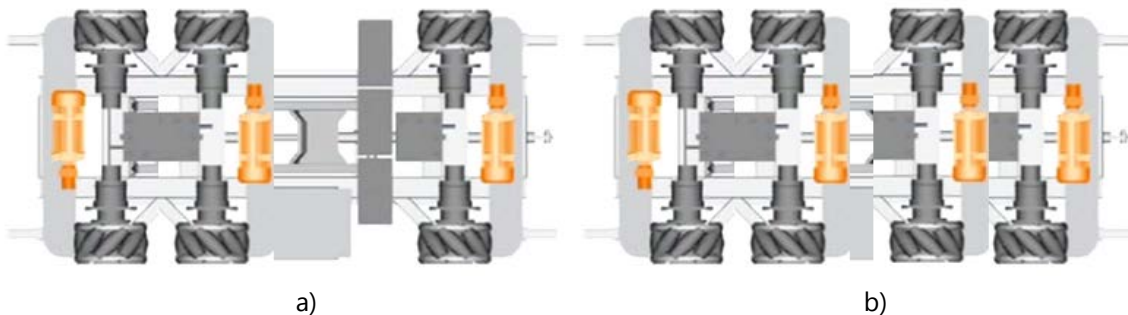


Fig. 2.3.3 a) Configuración de seis ruedas b) Configuración de ocho ruedas, ambas KUKA omniMove® [13].

2.3.4 Configuración de cuatro ruedas suecas tipo "AB" y "BB"

La plataforma diseñada consta de cuatro ruedas suecas. Las llantas adquiridas vienen por pares; cada par trae una rueda tipo "A" y tipo "B". La figura 2.3.4a muestra cuatro ruedas Fingertech.



Fig. 2.3.4a Dos pares de ruedas suecas Mecanum de tipo "A" y "B" [16].

La diferencia entre la llanta "A" y la "B" se muestra en la figura 2.3.4a. En dicha figura se puede observar que el ángulo de los rodillos de la rueda "B" llamado β tiene un valor de 45° , mientras que el ángulo α de la rueda "A" tiene un valor de 135° , ambos respecto al sistema de referencia que se muestra sobre la plataforma.

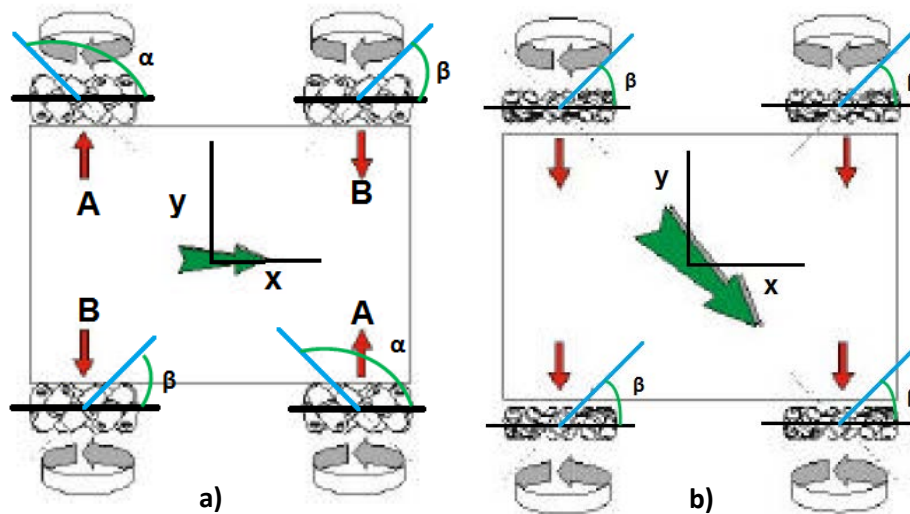


Fig. 2.3.4a a) Configuración de cuatro ruedas suecas "AB", b) Configuración de cuatro ruedas suecas "BB"[13].

Para la configuración "BB" que se muestra en la figura 2.3.4b, es visible que el ángulo β denotado en las cuatro ruedas tipo "B" mide 45° con respecto al sistema de referencia que se encuentra sobre la plataforma.

Capítulo 3. Diseño del Robot Móvil Omnidireccional usado en las pruebas

A continuación se presenta la información del proceso para poder crear un manipulador móvil omnidireccional, este solo es un modelo funcional con el cual se probará si el objetivo principal se cumple, es decir, no es algún prototipo que se desee por ahora llevar a la industria y sea comerciable.

3.1 Arquitectura

En el diseño y construcción de un robot se deben considerar algunos elementos importantes para darle versatilidad y eficacia como son: resistencia, equilibrio, funcionalidad, eficiencia, autonomía, capacidad de evolución, sea fiable, se adapte y sea reactivo.

3.2 Plataforma móvil omnidireccional

La figura 3.2a muestra la plataforma utilizada para el robot, esta tiene como característica principal las llantas omnidireccionales utilizadas para su funcionamiento.

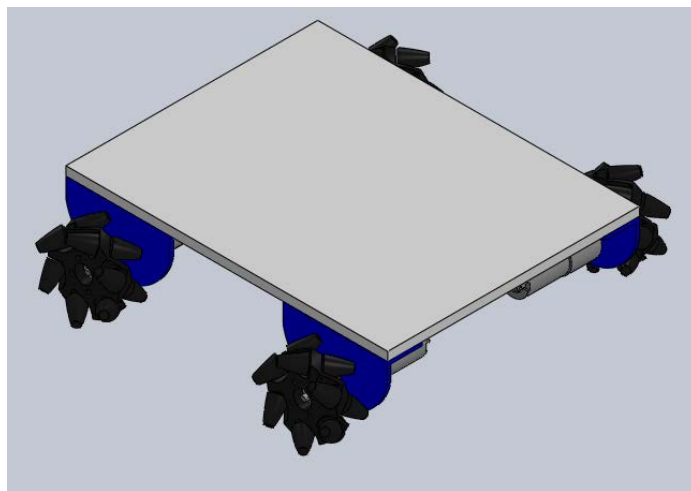


Fig. 3.2a CAD del robot móvil [3].

Plataforma

La plataforma tiene las siguientes características:

- Está hecha de acrílico transparente.
- Tiene un grosor de 62.3 [mm].
- Mide 180 [mm] de ancho y 220 [mm] de largo.

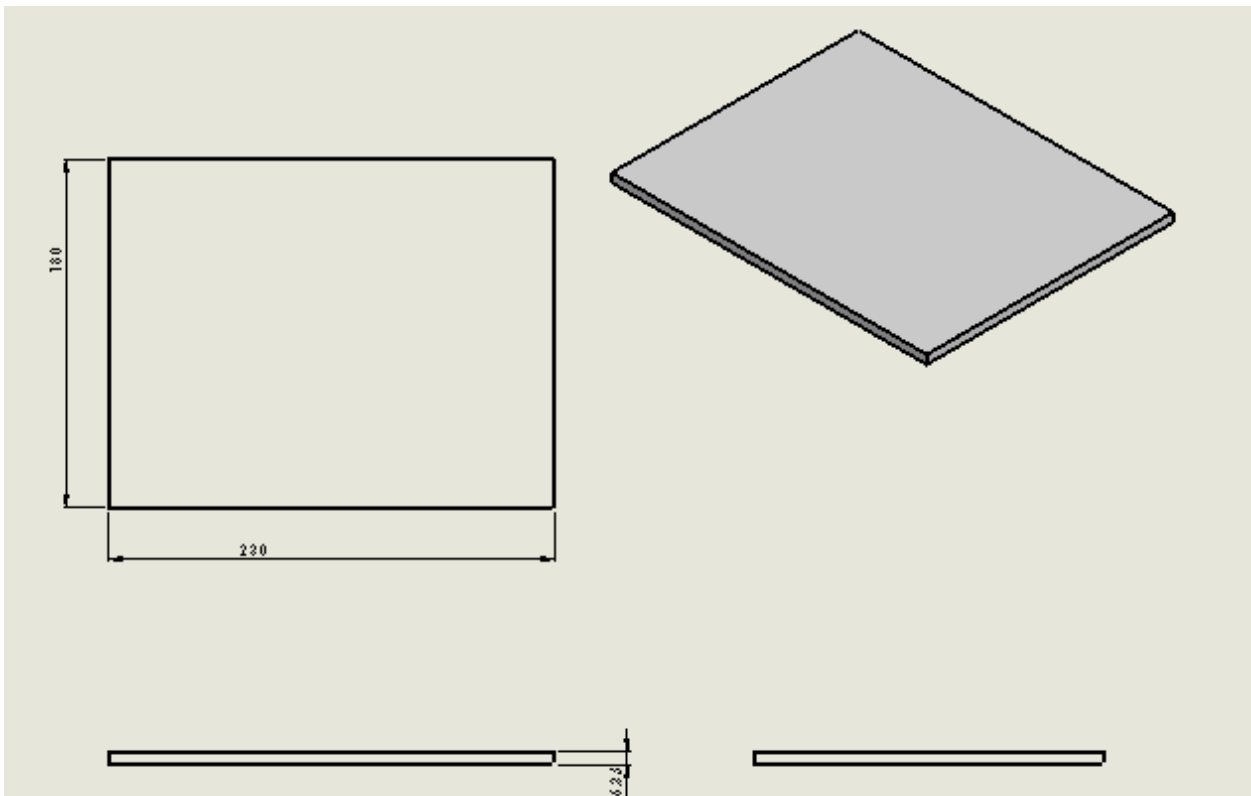


Fig. 3.2b Plano en CAD de la base [3].

La figura 3.2b tiene las dimensiones que se necesitan para poder acomodar todos los elementos que se requieren para el armado del robot omnidireccional.

Soporte motor-llantas-plataforma

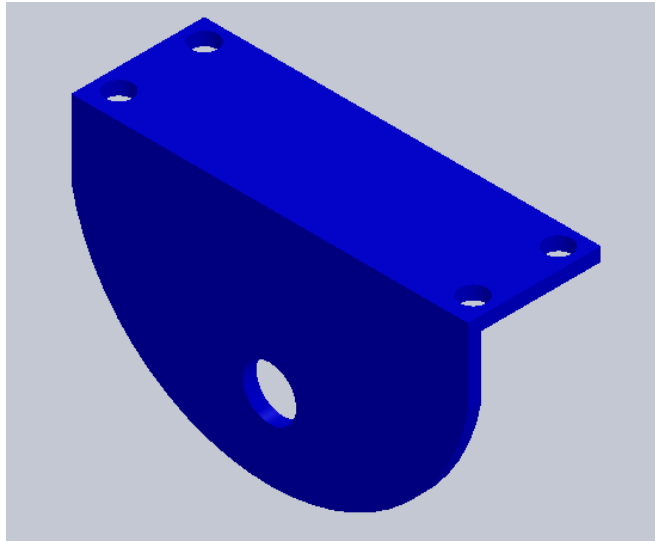


Fig. 3.2c Soporte de unión de un motor y una llanta [3].

En la figura 3.2c se puede revisar con atención uno de los cuatro soportes con los cuales se fijaron las llantas y los motores a la base principal, las dimensiones se muestran en la figura 3.2c1:

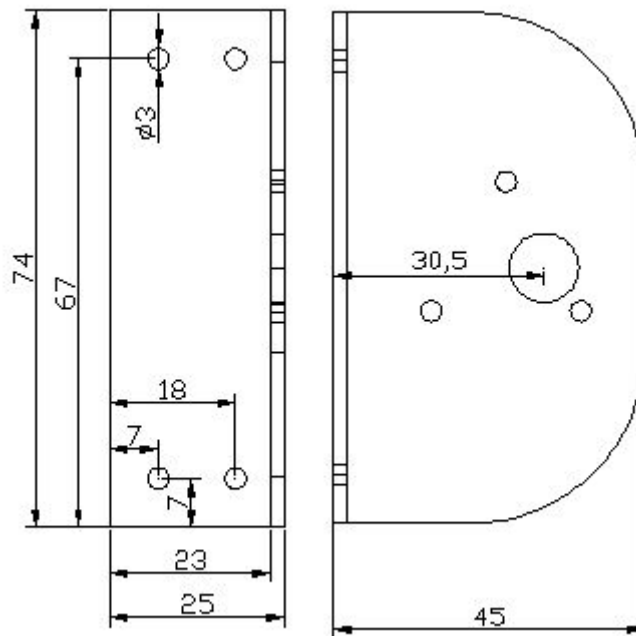


Fig. 3.2c1 Dimensiones del soporte [17].

Las dimensiones del soporte son importantes ya que estas permiten darle medidas límite y soporte a la base que sostendrá el brazo manipulador y todos los componentes electrónicos.

Motor



Fig. 3.2d Motor EMG30 [17].

La figura 3.2d muestra uno de los 4 motores EMG30 que se utilizaron para el movimiento de la base, estos nos ayudarán al desplazamiento del robot. En la figura 3.2d1 se muestran sus dimensiones:

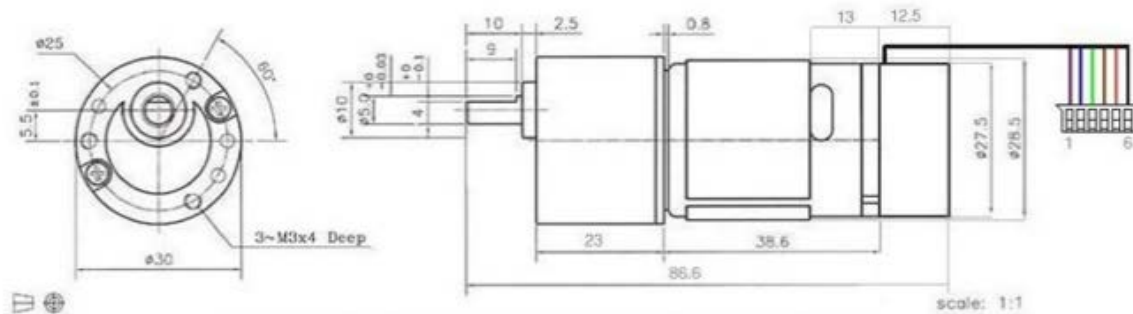


Fig. 3.2d1 Dimensiones del motor EMG30 [17].

Las especificaciones del motor EMG30 son las siguientes:

- Tensión nominal 12[VDC].
- Velocidad nominal 170 [rpm].
- Velocidad sin carga 216 [rpm].
- Consumo nominal 530 [mA].
- Consumo sin carga 150 [mA].
- Fuerza 1.5 [Kg/cm].

- Codificador de 90 pulsos por cada vuelta del eje de salida (4° de giro por cada pulso).

La siguiente es una tabla con la descripción de las salidas del motor por color y orden:

Tabla 3.1 Descripciones de salidas por cable del motor EMG30

CABLE	DESCRIPCIÓN
Morado	Señal de salida del sensor Hall del codificador B.
Azul	Señal de salida del sensor Hall del codificador A.
Verde	Alimentación GND para los sensores Hall.
Café	Alimentación +Vcc para los sensores Hall.
Rojo	+VDC para alimentación de los motores.
Negro	GND para alimentación de los motores.

Los sensores Hall necesitan resistencias pull-up de 4 [KΩ] que están incluidas si se utiliza el driver para la tarjeta MD25.

Ruedas

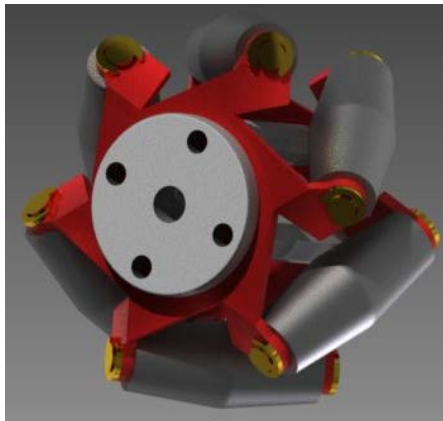


Fig. 3.2e Imagen en CAD de una de las llantas omnidireccionales suecas (Mecanum) [18].

La figura 3.2e muestra el CAD de las ruedas Mecanum que utilizó el prototipo funcional.

3.3 Control Electrónico

Este robot tiene comunicación inalámbrica con la computadora. Esto se logra con una tarjeta llamada Arduino Wifi Shield. Cada rueda cuenta con un motor que proporciona la fuerza necesaria para hacer girar las ruedas y que este siga la trayectoria deseada. A su vez, se utiliza una tarjeta Arduino Uno que manda las señales a los motores, y entre estos dos se encuentra otra tarjeta diseñada exclusivamente para controlar los motores utilizados en el robot. Toda la energía eléctrica que se requiere se suministra por medio de baterías.

Baterías

Las baterías son de litio recargables. Se muestra a continuación las características técnicas de ella. La figura 3.5 muestra la batería usada para el control de los motores de la base.



Fig. 3.3a Pila de litio [19].

Especificaciones de la batería:

Batería de polímero de Litio de 3 Celdas.

Capacidad de corriente: 5000 [mA] Hora.

Capacidad de Voltaje: 11.1 [Volts].

Constante de descarga: 20 [C].

Peso: 445 [grs].

Dimensiones: 145 X 49 X 27 [mm].

Plug: JST-XH.

Empaque: Es una carcasa de plástico de alta durabilidad.

Cargador de baterías de litio

Se utilizó un cargador de marca Turnigy de 3 celdas.



Fig. 3.3b Cargador de pilas de litio [19].

Wi-Fi Shield

Otra parte del control eléctrico consistió en determinar los elementos como las tarjetas de desarrollo y transmisión de datos para enviar información de la computadora al robot. La comunicación del robot es de manera inalámbrica, usando la tarjeta Wi-Fi Shield de Arduino.

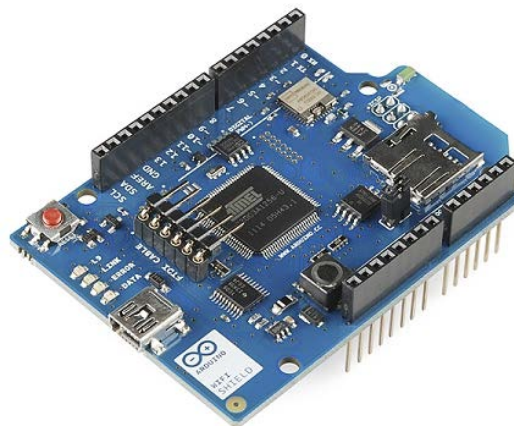


Fig. 3.3c Tarjeta Wi-fi Shield de Arduino [20].

Esta tarjeta permite conectar a la tarjeta Arduino con internet con la especificación inalámbrica 802.11, su sistema HDG104 inalámbrico a LAN 802.11 [b/g], el Atmega 32UC3 signa una IP que se puede usar con el protocolo UDP.

La tarjeta Arduino y el Wi-Fi Shield se conectan a través de los headers largos que pueden permitir colocar otro Shield en la parte superior.

Las características de la tarjeta Shield Wi-Fi es:

- Vía de conexión: redes 802.11 [b/g].
- Tipo de cifrado: WEP y WPA2 Personal.
- Ranura para tarjeta micro-SD incluida.
- Conexión estilo FTDI para depuración serial del Shield Wi-Fi.
- Conector micro-USB para actualizar el *firmware* del Shield Wi-Fi.
- El firmware de código abierto hace posible agregar nuevos protocolos directamente en el shield.

Arduino uno



Fig. 3.3d Arduino uno [21].

La tarjeta Arduino Uno es una tarjeta de desarrollo, que puede mediante la información que llega a sus pines de entrada, con ayuda de sensores, conocer qué hay a su alrededor, y a partir de esta información, realizar alguna actividad con ayuda de actuadores como motores o leds.

El microcontrolador que posee la tarjeta está basado en "Wiring" y su entorno de desarrollo es "Processing", el software que es requerido para poder programarlo es completamente gratis y se encuentra en la página oficial de Arduino.

Una de las tantas ventajas de esta tarjeta es que no es necesario conectarla a la computadora para que funcione, basta con suministrarle energía con una pila para que funcione lo que se tenga programado en ella.

Una de las principales características de esta tarjeta es su versatilidad, no es necesario crear una tarjeta para cada proyecto que se quiera realizar, basta con tener una de estas y se puede adaptar a cualquier necesidad. Las características de esta tarjeta se encuentran resumidas en la siguiente tabla:

Tabla 3.2 Características generales de la tarjeta Arduino Uno.

Microcontroller	ATmega328
Operating Voltage	5 [V]
Input Voltage (recommended)	7-12 [V]
Input Voltage (limits)	6-20 [V]
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 [mA]
DC Current for 3.3V Pin	50 [mA]
Flash Memory	32 [KB] (ATmega328) of which 0.5[KB] used by bootloader
SRAM	2 [KB] (ATmega328)
EEPROM	1 [KB] (ATmega328)
ClockSpeed	16 [MHz]

Tarjeta MD25

Esta es una tarjeta que sirve como controlador de dispositivos, en este caso la utilizamos para controlar dos motores modelo EMG30 de DC, desarrollada por la firma Devantech, Ltd. La figura 3.3e muestra la tarjeta.



Fig. 3.3e Tarjeta MD25 [22].

Las características de la tarjeta son:

- Tiene entradas de codificaciones para cuadratura desde los motores.
- La lectura de los codificadores proporciona un contador que determina el desplazamiento producido en los motores y el sentido de giro o dirección de los mismos.
- Actúa sobre dos motores que pueden ser controlados de forma autónoma o diferencial.
- Permite conocer el consumo de cada motor.
- La alimentación es de una sola tensión, +12 [VDC].
- Contiene un regulador de +5 [Vcc] a 300 [mA] para alimentar circuitos externos auxiliares.
- Puede llegar a suministrar una corriente de hasta 3 [A] por cada motor.
- Tiene una interfaz serie o I2C con posibilidad de conectar hasta 8 drivers MD25 en el mismo bus.
- Control de potencia y aceleración.
- Sus dimensiones son de 110 x 52 x 25 [mm].

Capítulo 4. Modelo Cinemático y Control de Robot Móvil con Ruedas Mecanum

La cinemática se encarga de estudiar el movimiento de un robot haciendo énfasis en su geometría. El modelado cinemático es una parte muy importante en el diseño de robots ya que permite simular el comportamiento del robot, y establecer el modelo matemático que puede ayudar a diseñar el controlador.

Cuando se habla de control, se piensa inmediatamente en encontrar leyes que ayuden a dar estabilidad al robot, mediante la realimentación al sistema dentro de un punto de equilibrio, lo cual es conocido como *regulación* y este tipo de control está más enfocado a robots móviles no-holonómicos [1,4,7]. A pesar de esto un robot móvil omnidireccional también necesita ser controlado.

Sin embargo, por lo general cuando se diseña un robot móvil, este tiene que seguir trayectorias que cambian respecto al tiempo, como consecuencia, hay que controlar al robot para que pueda realizar dicha tarea de manera estable y con el menor error posible. A este tipo de control se le denomina *estabilización sobre una trayectoria* [1].

Controlar a un robot de tres ruedas que no sean suecas para que siga una trayectoria, no es tan fácil. A pesar de esto, se puede realizar una linealización parcial de los sistemas [1, 4, 23] con tres ruedas no-holonómicas.

La tabla 4.1, tomada de [1], contiene los parámetros necesarios para hacer lineal las salidas del modelo dinámico de posición del robot, lo cual garantiza el control de la posición de un punto fijo del robot, para que este pueda seguir una trayectoria en el marco global.

La primera columna contiene los tipos de robots móviles según el grado de movilidad y direccionalidad. En la segunda columna se encuentran las coordenadas cartesianas del punto fijo de control \mathbf{P} , referidas al marco global. La tercer columna, contiene la primer derivada de \mathbf{z}_1 . Por último, la cuarta columna contiene las condiciones para que el determinante de \mathbf{z}_3 sea diferente de cero.

Tabla 41. Linealización de las salidas, coordenadas no lineales, y condiciones de regulación para diferentes tipos de robots.

(δ_m, δ_s)	$z_1 = h(\varepsilon, \beta_s)$	$z_3 = k(\varepsilon, \beta_s)$	$\det(K(\varepsilon, \beta_s)) \neq 0$
(2, 0)	$\begin{pmatrix} x + e \cos(\vartheta + \delta) \\ y + e \sin(\vartheta + \delta) \end{pmatrix}$	ϑ	$e \neq 0$ $\delta \neq 2k\pi$
(2, 1)	$\begin{pmatrix} x + e \cos(\vartheta + \delta) \\ y + e \sin(\vartheta + \delta) \\ \beta_s \end{pmatrix}$	ϑ	$e \neq 0$ $\delta \neq \beta_s + 2k\pi$
(1, 1)	$\begin{pmatrix} x + L \sin \vartheta + e \cos(\vartheta + \beta_s) \\ y - L \cos \vartheta + e \sin(\vartheta + \beta_s) \end{pmatrix}$	$\begin{pmatrix} \vartheta \\ \beta_s \end{pmatrix}$	$e \neq 0$
(1, 2)	$\begin{pmatrix} x + L \cos \vartheta - e \sin(\vartheta + \beta_{s1}) \\ y + L \sin \vartheta + e \cos(\vartheta + \beta_{s1}) \\ \beta_{s2} \end{pmatrix}$	$\begin{pmatrix} \vartheta \\ \beta_{s1} \end{pmatrix}$	$e \sin \beta_{s2} \neq 0$

Para el caso de robots omnidireccionales con cuatro llantas Mecanum, existen varios artículos y publicaciones al respecto, donde se utilizan controles adaptativos, o controladores basados en lógica difusa, tanto para probar la omnidireccionalidad del robot respecto a sí mismo, como para mejorar el seguimiento de una trayectoria lineal.

En [24] no se cuenta con un modelo físico del robot, pero se tiene un modelo cinemático, dinámico y con una ecuación de estado para una configuración de cuatro ruedas, empleando un modelo de referencia para el seguimiento del punto medio del sistema y verificar si se realiza la trayectoria propuesta.

En [25], se tiene un robot de cuatro ruedas para el seguimiento de trayectorias definidas y programadas en un microcontrolador, con la ayuda de un lector óptico de mouse y comunicación UART se registran los datos de la posición de un punto de este respecto al tiempo. Se hicieron pruebas en lazo abierto, adicionando tres controladores PID para corregir el desplazamiento en lazo cerrado. En este trabajo no se cuenta con un modelo cinemático.

En [26] se tiene un modelo cinemático y dinámico que fue utilizado en un robot diseñado por los autores. Se hicieron pruebas de desplazamiento en línea recta sobre el eje "x", "y" y movimiento angular $\dot{\vartheta}$. De esta manera comprobaron la omnidireccionalidad del robot respecto a sí mismo. Usaron una cámara para el

control correctivo del sistema y disminuir errores de posición, aunque faltó un método que ayudara a hacer correcciones de desplazamiento en trayectorias más complejas que las que se presentan en una línea recta.

En [27], se cuenta con la cinemática para un modelo físico diseñado y construido por los autores. Se utiliza un control autónomo con ayuda de un par de sensores de ultrasonido y unas barras de parachoques para evadir obstáculos, además se implementó una cámara de video para realizar la captura de la imagen del área de trabajo para localizar al robot por medio de un punto de referencia sobre este.

En [28] se utilizó un modelo físico para la comprobación del funcionamiento de la cinemática desarrollada, apoyándose en un láser telemétrico para conocer la posición del robot. Finalmente se realizaron las pruebas básicas sobre el eje " x " e " y " comprobando que, sí funcionan, las ecuaciones cinemáticas de ajuste a dichas trayectorias sobre los ejes.

En [29] se cuenta con un modelo físico mediante el cual, a partir del desarrollo de la cinemática y la implementación de lógica difusa, se realizaron pruebas donde se analizó el desplazamiento sobre el eje " y ", se observó que entre más funciones de pertenencia se utilicen, más eficiente era el robot para seguir trayectorias rectas sobre los ejes, la desventaja es que se requiere una mayor capacidad de procesamiento.

El robot construido para este trabajo tiene su propio modelo cinemático, obtenido por el método de propagación de velocidades [30]. Este modelo difiere de los obtenidos en los trabajos mencionados anteriormente. Se utilizó un sistema de realimentación mediante una cámara -parecido a [27]- para conocer la posición del robot en todo tiempo. No se hace un análisis dinámico del sistema como en [31], ya que como lo explica [32], el modelo dinámico para robots móviles es más complejo. El controlador está basado en campos potenciales [33], lo que permitió realizar trayectorias más complejas que una recta, para probar la omnidireccionalidad del robot respecto a sí mismo, y al sistema de referencia global. La comunicación con la computadora es de forma inalámbrica, mediante el protocolo de datagrama de usuario (UDP por sus siglas en inglés) [34].

4.1 Plataforma móvil omnidireccional con llantas suecas

Anterior a este trabajo existe [35, 36], donde se estudia el comportamiento de dos robots móviles, uno omnidireccional a partir de dos diferenciales [35] y uno diferencial [36]. En ambos casos, se implementó un controlador por campos potenciales para seguir una trayectoria.

El robot estudiado en este trabajo (descrito previamente en el capítulo dos y tres) posee cuatro ruedas suecas. Se menciona esto ya que algunas de las ecuaciones planteadas en [35 y 36] fueron retomadas para expresar la configuración cinemática del robot. La figura 4.1a muestra las plataformas omnidireccionales.

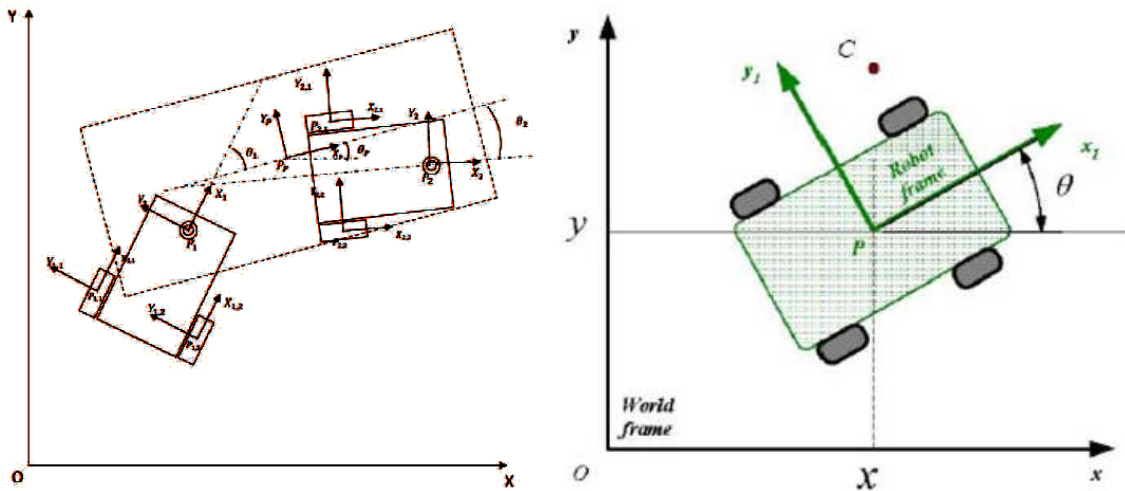


Fig. 4.1a a) Plataforma móvil con 2 robots diferenciales, b) Plataforma móvil con 4 ruedas suecas [35,3].

La plataforma móvil es considerada como un cuerpo rígido no deformable, así mismo, las ruedas no tienen deslizamientos y el robot se mueve en un plano horizontal. En el caso de la plataforma omnidireccional las ruedas siempre giran alrededor del eje del motor al cual están acopladas. Las notaciones y nomenclatura utilizada se tomó del documento de V.J. González Villela [37].

En la figura 4.1b se muestran dos marcos coordenados para la plataforma móvil. El primer marco es el global y se encuentra representado por $(x_0 - y_0)$ fijados al origen (0) . Este marco ayudará a registrar posiciones y velocidades de la plataforma.

El segundo marco de referencia está representado por $(x_1 - y_1)$ y fijado al punto P , el cual se localiza en el centroide de la plataforma móvil. Gracias a estos marcos

de referencia se define un vector de posición del robot móvil, tal como se muestra a continuación:

$$\xi = [x \quad y \quad \theta]^T \quad (4.1)$$

Este vector permite describir la posición del punto $P = (x_1, y_1)$ y la orientación θ del marco de la plataforma $(x_1 - y_1)$, que hace referencia al marco global $(x - y)$.

En el caso de la plataforma móvil, el ángulo θ se mide respecto al eje x . Después de las definiciones anteriores, se requiere obtener una matriz de rotación de la posición del punto P , esta consiste en una matriz ortonormal que mapea del marco fijo al de la plataforma móvil, y se expresa de la siguiente manera:

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

4.1.1 Propagación de velocidades

El siguiente problema es calcular las restricciones cinemáticas de la plataforma móvil. Dentro de estas restricciones están la velocidad lineal y angular de los eslabones del robot móvil. Se utilizó en este apartado el método descrito por Craig [30]. El cual se explica a continuación:

Se puede calcular la velocidad de cualquier eslabón, o comenzando de manera ordenada desde la base. De esta manera, la velocidad del eslabón $i + 1$ va a ser la del eslabón i más cualquier otra componente de velocidad producida por la articulación $i + 1$. La figura 4.1.1c muestra los elementos a considerar para aplicar la técnica.

Para el caso de los vectores de las velocidades de rotación, estos pueden agregarse solo si se han trazado en el mismo sistema de referencia. Debido a esto, la velocidad angular del eslabón $i + 1$ es igual a la del eslabón i más cualquier nuevo componente de velocidad producido por la velocidad rotacional que lleve el eslabón $i + 1$. Esto puede describirse de la siguiente manera:

$${}^i\omega_{i+1} = {}^i\omega_i + {}_{i+1}{}^iR \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} \quad (4.3)$$

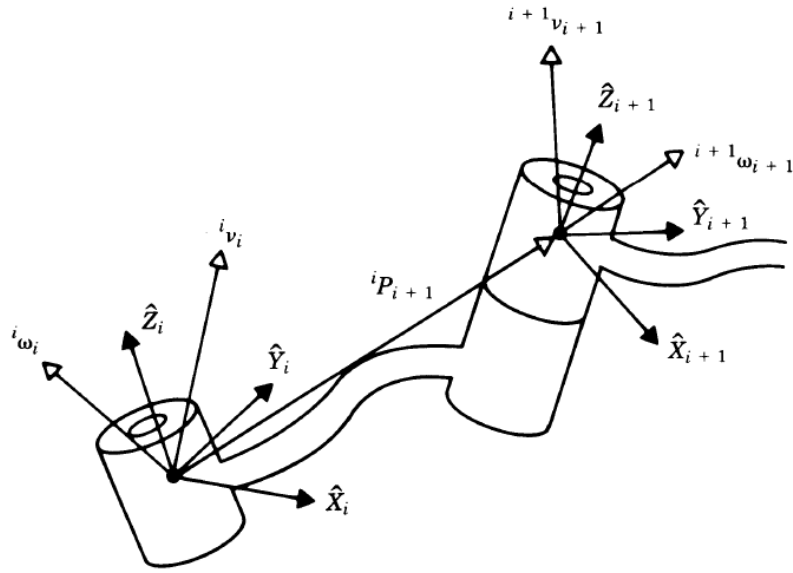


Fig. 4.1.1c Vectores de velocidad entre eslabones adyacentes, v_i y w_i [30]

En donde:

$$\dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} = {}^{i+1} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\theta}_{i+1} \end{bmatrix} \quad (4.4)$$

Al multiplicar la ecuación **4.41** en ambos lados por ${}^{i+1}\mathbf{R}$ se obtiene la descripción para la velocidad angular del eslabón $i + 1$ respecto al sistema de referencia $i + 1$:

$${}^i\omega_{i+1} = {}^{i+1}\mathbf{R}^i\omega_i + \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (4.5)$$

La velocidad lineal del origen del sistema de referencia $i + 1$ es la misma que la del sistema de referencia i , más un nuevo componente que se obtiene por la velocidad de la rotación del eslabón i , de esto se obtiene:

$${}^i v_{i+1} = {}^i v_i + {}^i\omega_{i+1} \times {}^i P_{i+1} \quad (4.6)$$

En donde ${}^i P_{i+1}$ representa un vector constante del sistema de referencia i , ya que indica la coordenada del origen del sistema $i + 1$ respecto al sistema de referencia i . Si se multiplican ambos lados de la ecuación **(4.4)** por ${}^{i+1}_i\mathbf{R}$ se obtiene:

$${}^{i+1}v_{i+1} = {}^{i+1}_i\mathbf{R}({}^i v_i + {}^i\omega_i \times {}^i P_{i+1}) \quad (4.7)$$

Las ecuaciones **(4.3)** y **(4.4)** solamente son válidas para eslabones unidos mediante juntas rotacionales. Al aplicar dichas ecuaciones a cada eslabón se puede obtener las velocidades angulares y lineales de cada uno, todo esto en su mismo sistema de referencia. Con estas ecuaciones se obtienen las restricciones cinemáticas de la plataforma móvil.



4.1.2 Restricciones cinemáticas de la plataforma móvil omnidireccional con ruedas suecas (Mecanum)

Considerando lo anterior es posible obtener las restricciones cinemáticas al propagar las velocidades $[\dot{x}_1 \ \dot{y}_1 \ \dot{\theta}]$ del marco de la plataforma móvil ($x_1 - y_1$), al marco de las llantas. Para poder hacerlo, se deben considerar algunos aspectos de suma importancia, y se utiliza el siguiente vector de velocidad:

$$\dot{\xi}_1 = [\dot{x}_1 \ \dot{y}_1 \ \dot{\theta}]^T \quad (4.8)$$

Este vector representa la velocidad del marco de la plataforma móvil respecto a sí misma. Las velocidades lineales del punto P están expresadas por \dot{x}_1 e \dot{y}_1 , y para la velocidad angular se tiene $\dot{\theta}$. Para poder representar este vector en el marco global se ocupa la siguiente ecuación:

$$\dot{\xi} = R^T(\theta)\dot{\xi}_1 \quad (4.9)$$

La figura 4.9 muestra una posición arbitraria de la plataforma móvil con las ruedas omnidireccionales.

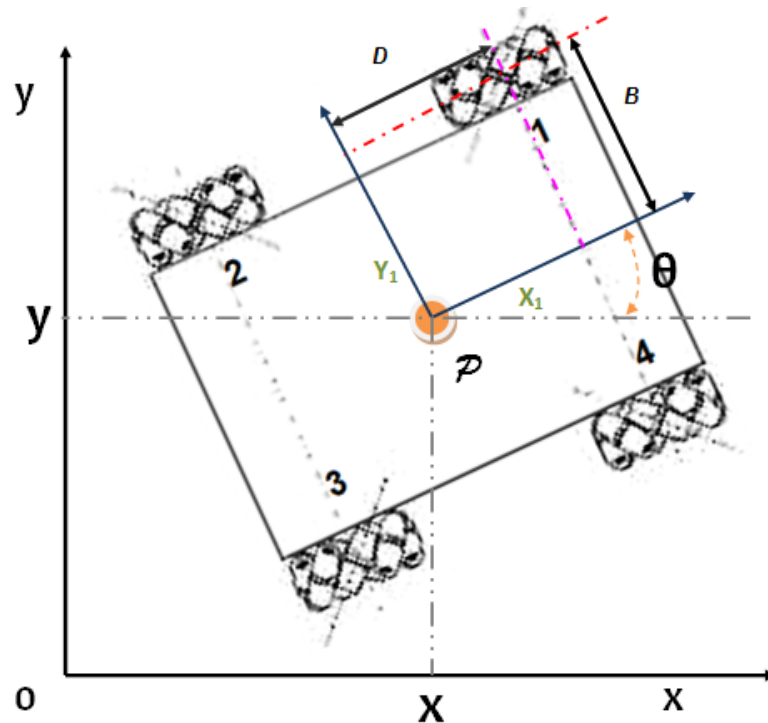


Fig. 4.1.2a Vista superior de la posición arbitraria de la plataforma móvil respecto al marco global. [16]

Primeramente se definen los elementos necesarios para modelar la rueda. En la figura 4.1.2b se muestran los parámetros tomados en cuenta para el modelado. Esto se hace a partir de definir el marco de referencia $x_1 - y_1$, ya que las ruedas se

modelan primero respecto a este sistema de referencia, tomando en cuenta el ángulo θ , que ayuda a determinar la matriz de rotación respecto al sistema global $x - y$.

Con la ayuda de la imagen 4.9 se definen las siguientes ecuaciones:

$${}^0P_1 = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \quad (4.01p)$$

$${}^0R_1 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.01r)$$

La ecuación 4.01p representa la posición del sistema de referencia {1}, respecto al sistema de referencia global {0}, mientras que la ecuación 4.01r representa la rotación del sistema de referencia {1}, respecto al sistema de referencia global {0}.

$${}^1P_2 = \begin{bmatrix} d \\ b \\ 0 \end{bmatrix} \quad (4.21p)$$

$${}^1R_2 = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.21r)$$

La ecuación 4.21p representa la posición del sistema de referencia {2}, respecto al sistema de referencia {1}, mientras que la ecuación 4.21r representa la rotación del sistema de referencia {2}, respecto al sistema de referencia {1}.

$${}^2P_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.32p)$$

$${}^2R_3 = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.32r)$$

La ecuación 4.32p representa la posición del sistema de referencia {3}, respecto al sistema de referencia {2}, mientras que la ecuación 4.32r representa la rotación del sistema de referencia {3}, respecto al sistema de referencia {2}.

Con estas ecuaciones podemos comenzar a definir los vectores tanto para velocidades angulares, así como para velocidades lineales.

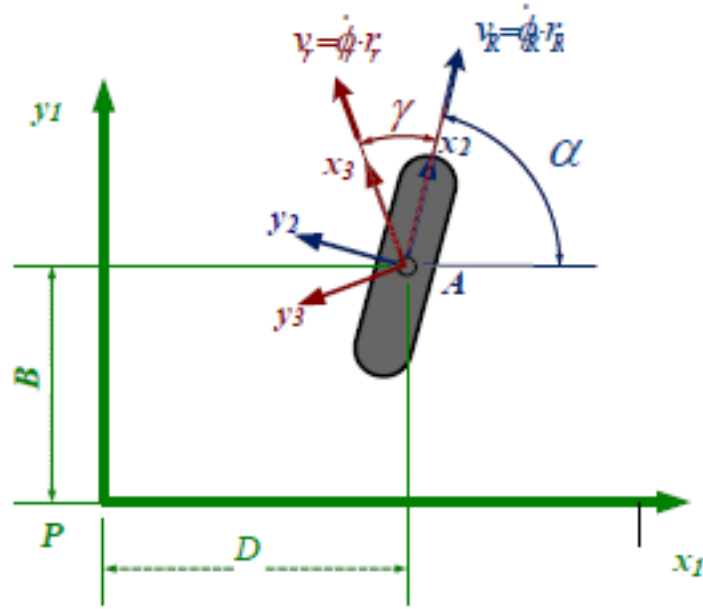


Fig. 4.1.2b Definición de parámetros para una rueda sueca [3].

Los parámetros se tienen son:

1. D , la coordenada de la posición de la rueda respecto a x_1 .
2. B , la coordenada de la posición de la rueda respecto a y_1 .
3. α , ángulo que forma la rueda respecto al sistema de referencia ($x_1 - y_1$)
4. γ , ángulo del vector velocidad de cada rodillo de las ruedas respecto al sistema ($x_2 - y_2$).
5. v_R , velocidad lineal de la rueda.
6. v_r , velocidad lineal del rodillo de la rueda.
7. $\dot{\phi}_r$, velocidad angular del rodillo.
8. $\dot{\phi}_R$, velocidad angular de la rueda sueca.
9. r_r , radio del rodillo.
10. r_R , radio de la rueda sueca.

Aplicando el método de propagación de velocidades, como se muestra en Craig [30] se obtienen las restricciones cinemáticas de la rueda respecto al sistema de referencia del robot:

$$\begin{pmatrix} \cos(\gamma + \alpha + \theta) & \sin(\gamma + \alpha) & d \cdot \sin(\gamma + \alpha + \theta) - b \cdot \cos(\gamma + \alpha) \\ -\sin(\gamma + \alpha + \theta) & \cos(\gamma + \alpha) & b \cdot \sin(\gamma + \alpha + \theta) - d \cdot \cos(\gamma + \alpha) \end{pmatrix} \cdot \begin{pmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta} \end{pmatrix} - \begin{pmatrix} r_R \cdot \cos(\gamma) & 1 \\ -r_R \cdot \sin(\gamma) & 0 \end{pmatrix} \cdot \begin{pmatrix} \dot{\phi}_R \\ v_r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

(4.10)

4.1.3 Coordenadas generalizadas de configuración

Para obtener las coordenadas generalizadas de esta plataforma móvil, se tomaron en cuenta los siguientes elementos:

1. Las coordenadas x, y ; definen la posición del punto P y el ángulo θ define la orientación de la plataforma.
2. Las coordenadas de rotación de las llantas a lo largo de su eje horizontal de rotación.
3. Las coordenadas de rotación de los rodillos que se encuentran sobre las llantas a lo largo del eje de rotación horizontal de estas.

Con los parámetros anteriormente mencionados se obtiene el vector de coordenadas generalizadas para la plataforma móvil, el cual queda de la manera siguiente:

$$\mathbf{q} = [x \quad y \quad \theta \quad \phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4]^T \quad (4.11)$$

Así mismo, se tiene el vector de velocidades:

$$\dot{\mathbf{q}} = [\dot{x} \quad \dot{y} \quad \dot{\theta} \quad \dot{\phi}_1 \quad \dot{\phi}_2 \quad \dot{\phi}_3 \quad \dot{\phi}_4]^T \quad (4.12)$$

Ya que se tienen las coordenadas generalizadas se busca una expresión en el espacio de estados para la cinemática. Seleccionando como variables de estado a la velocidad lineal $\mathbf{v} = [v_x \quad v_y]^T$ y angular ω de la plataforma. Con lo cual, se tiene el modelo cinemático para un robot de cuatro ruedas suecas o Mecanum; tal como lo muestra la matriz (4.13). Este modelo permite asignar al robot las velocidades angulares de cada llanta, y así poder seguir las trayectorias propuestas.

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & -\frac{1}{r} & -\frac{b+d}{r} \\ \frac{1}{r} & \frac{1}{r} & -\frac{b+d}{r} \\ \frac{1}{r} & -\frac{1}{r} & \frac{b+d}{r} \\ \frac{1}{r} & \frac{1}{r} & \frac{b+d}{r} \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (4.13)$$

4.2 Control por Campos potenciales

En la técnica de campos potenciales se toman en cuenta la suma de fuerzas (vectores) de atracción (meta) y rechazo (obstáculos) en el lugar donde se realice la prueba (área de trabajo) como se explica en [33]. Es importante entender que para mover el robot sólo es necesaria la orientación y los componentes de velocidad tanto en x como y , ya que en este caso no existen obstáculos.

Una vez que se tiene las variables a controlar, es posible aplicar una fórmula de control a la plataforma móvil, y así ver la adaptación a las diferentes trayectorias propuestas. El método de campos potenciales, puede aplicarse desde el punto de vista cinemático, cambiando los vectores de fuerza por los de distancia. El objetivo al que se desea llegar representa un vector de sentido que parte de la posición inicial del robot hacia dicho objetivo, siendo este el punto final. Para evitar colisiones se deben establecer áreas de seguridad y velocidades fijas y evitar las velocidad infinitas o demasiado grandes.

Se define un área de seguridad alrededor del objetivo y del robot, y una velocidad máxima tanto para el eje x , como para el eje y y una velocidad angular máxima. Mientras más cerca esté el robot del punto objetivo irá disminuyendo su velocidad hasta que los círculos sean tangenciales.

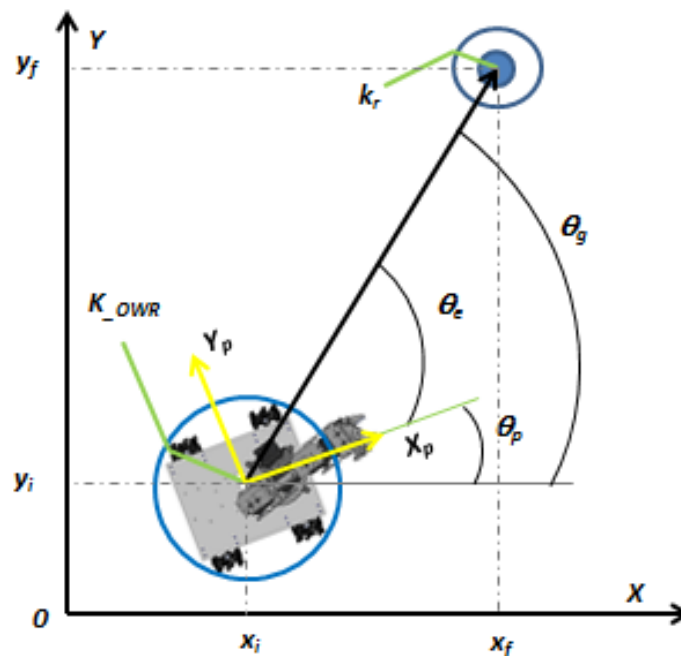


Fig. 4.2a Alcance de objetivo con campos potenciales [3].

En base a lo anterior se obtiene una serie de puntos que permitirán al robot llegar al punto objetivo. De esta manera se define el vector de distancia que va de la posición inicial del robot hasta el punto final (meta), el cual queda de la siguiente forma:

$$\mathbf{d}_g = (x_f - x_i, y_f - y_i) \dots \dots \dots (4.14)$$

Donde la magnitud de la distancia se obtiene mediante:

$$d_g = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2} \dots \dots \dots (4.15)$$

Para encontrar el ángulo de la meta:

$$\theta_g = \text{ang tan} \left(\frac{y_f - y_i}{x_f - x_i} \right) \dots \dots \dots (4.16)$$

En donde el ángulo de error se obtiene:

$$\theta_e = \theta_g - \theta_i \dots \dots \dots (4.17)$$

Para que el robot tenga la orientación deseada se necesita una velocidad angular que ayude a orientarlo por completo, haciendo que θ_e tienda a cero, para esto se cuenta con la siguiente ecuación:

$$w = w_{max} \sin \theta_e \dots \dots \dots (4.18)$$

En esta ecuación también se toma en cuenta w_{max} , la cual indica la velocidad máxima que puede alcanzar el móvil.

Otras dos ecuaciones que se tomaron en consideración para el experimento son:

$$w = w_{max} \sin(-\theta_p) \dots \dots \dots (4.19)$$

$$w = w_{max} \sin\left(\theta_e - \frac{\pi}{2}\right) \dots \dots \dots (4.20)$$

Para la velocidad que debe tomar el móvil, se tienen las siguientes ecuaciones tanto para su desplazamiento sobre el eje X, como en el eje Y:

$$v_{xp} = \begin{cases} v_{xmax} \cdot \cos(\theta_e) & \text{si } |d_g| > k_r + k_{OWR} \\ \frac{v_{xmax}}{k_r + k_{OWR}} \cdot d_g \cos(\theta_e) & \text{si } |d_g| < k_r + k_{OWR} \dots \dots \dots (4.21) \\ 0 & \text{si } |d_g| \leq k_{OWR} \end{cases}$$

$$v_{yp} = \begin{cases} v_{xmax} \cdot \sin(\theta_e) & \text{si } |d_g| > k_r + k_{OWR} \\ \frac{v_{xmax}}{k_r + k_{OWR}} \cdot d_g \sin(\theta_e) & \text{si } |d_g| < k_r + k_{OWR} \dots \dots \dots (4.22) \\ 0 & \text{si } |d_g| \leq k_{OWR} \end{cases}$$

Las ecuaciones anteriores indican la velocidad a la que debe ir dicho robot, está definida por el vector de atracción del robot al objetivo d_g , el ángulo de ese vector θ_g , el ángulo de inclinación del robot θ_i , y el error existente entre ambos ángulos θ_e , el área de seguridad (para evitar colisiones) con radio k_r y el área de seguridad del robot de k_{OWR} .



Capítulo 5. Resultados de las pruebas con el Robot Móvil Omnidireccional

Para probar la omnidireccionalidad de la plataforma móvil, se realizaron las siguientes pruebas en lazo abierto y en lazo cerrado. Las pruebas en lazo abierto permiten observar si el robot es o no controlable, también ayuda a conocer las causas de los distintos errores que pueden afectar al robot y que a su vez son ajenos a las ecuaciones de control.

Las pruebas en lazo cerrado, permiten comprobar la estabilidad del robot y el funcionamiento del sistema de control. Muestran además la capacidad de desplazamiento del robot.

Debido a que los movimientos son más complejos y por lo tanto, más difíciles de controlar se utilizaron las ecuaciones de las curvas de Lissajous [38] para el desplazamiento sobre trayectorias más complejas en comparación a la trayectoria de una línea recta o un círculo.

Lazo abierto:

1. Análisis del movimiento del punto **P** sobre la plataforma en línea recta diagonal.

Lazo cerrado:

1. Seguimiento de una trayectoria rectilínea del punto **P** en la plataforma con orientación de 0° .
2. Seguimiento de una trayectoria circular del punto **P** sobre la plataforma con una velocidad angular ω .
3. Seguimiento de Curva de Lissajous denominada "Kiss" con velocidad angular ω .
4. Seguimiento de Curva de Lissajous denominada "Pop" con inclinación de 20° .

A continuación se muestran las gráficas e imágenes de cada una de las pruebas, tanto en lazo cerrado como en lazo abierto, junto con los valores de los parámetros necesarios para llevarlas a cabo.

5.1 Resultado de la prueba en Lazo Abierto

5.1.1 Movimiento del punto de análisis de la plataforma en línea recta diagonal.

Se realizó un movimiento rectilíneo a 45° con una inclinación de 0° respecto al sistema {x y}. Para poder lograr este movimiento se introdujo una velocidad igual tanto en x como en y (v_x y v_y) mayores a cero.

La posición de la plataforma se seleccionó de forma arbitraria. **Las velocidades en v_x y v_y deben ser iguales.**

- Velocidades de entrada:
 - $v_x = 5 \left[\frac{cm}{s} \right]$
 - $v_y = 5 \left[\frac{cm}{s} \right]$
 - $\omega = 0 \left[\frac{rad}{s} \right]$

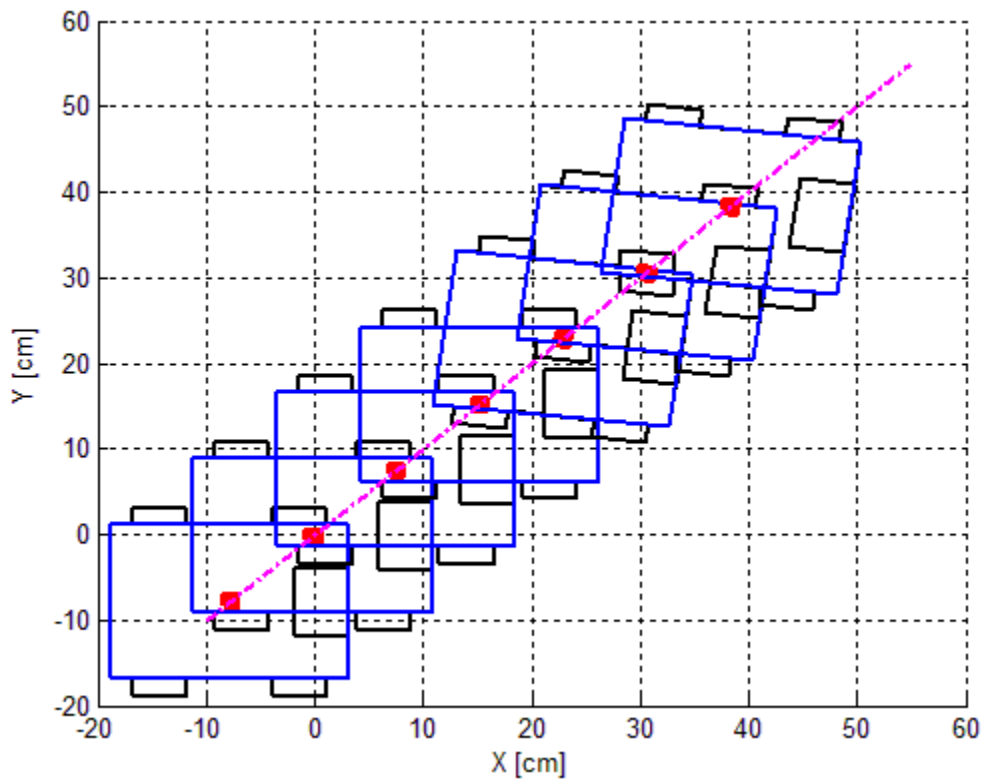


Fig. 5.1.1a Simulación de recta a 45° en lazo abierto [3].

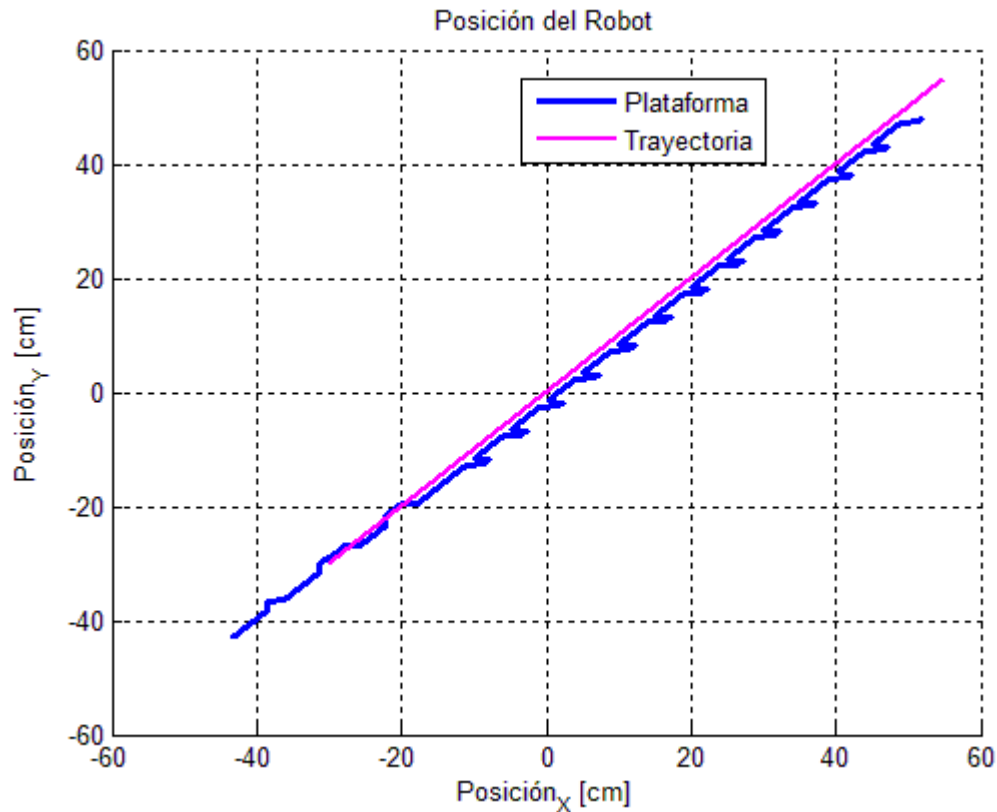


Fig. 5.1.1b Posición descrita vs posición seguida por el robot [3].

En la figura 5.1.1a se observa cómo el robot móvil se desplaza a través de la línea rosa, corroborándolo con el punto rojo que marca el punto central del mismo, en donde también se puede señalar que el análisis comienza desde las coordenadas (-10,-10), hasta (40,40).

Para la gráfica de la figura 5.1.1a se observa que tanto la trayectoria que se debe seguir según la fórmula establecida (recta rosa), como la que sigue el robot móvil omnidireccional (recta azul) son casi idénticas, esto muestra que la cinemática establecida en la fórmula con la cual se está manipulando el móvil es eficiente, ya que sigue la línea marcada para su trayectoria desde la fórmula, recordando que este se está trasladando con respecto a sí mismo y no al universo.

Además se puede agregar que el móvil (recta azul) está por debajo de la línea de la trayectoria que se le está indicando (recta rosa), esta es otra forma de corroborar que la cinemática establecida por la fórmula funciona aún sin ocupar campos potenciales.

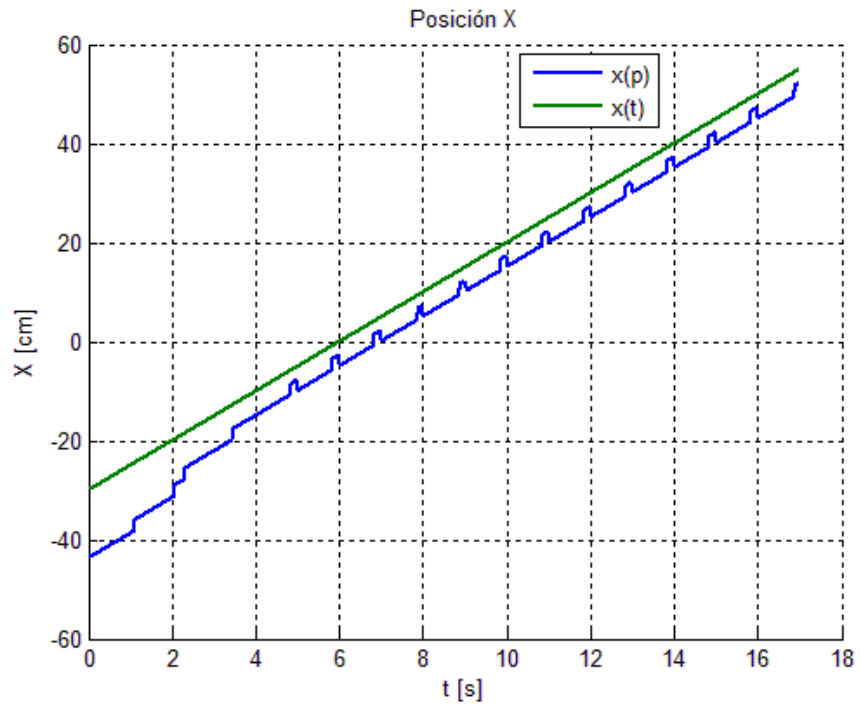


Fig. 5.1.1c Posición descrita para $x(t)$ vs $x(t)$ [3].

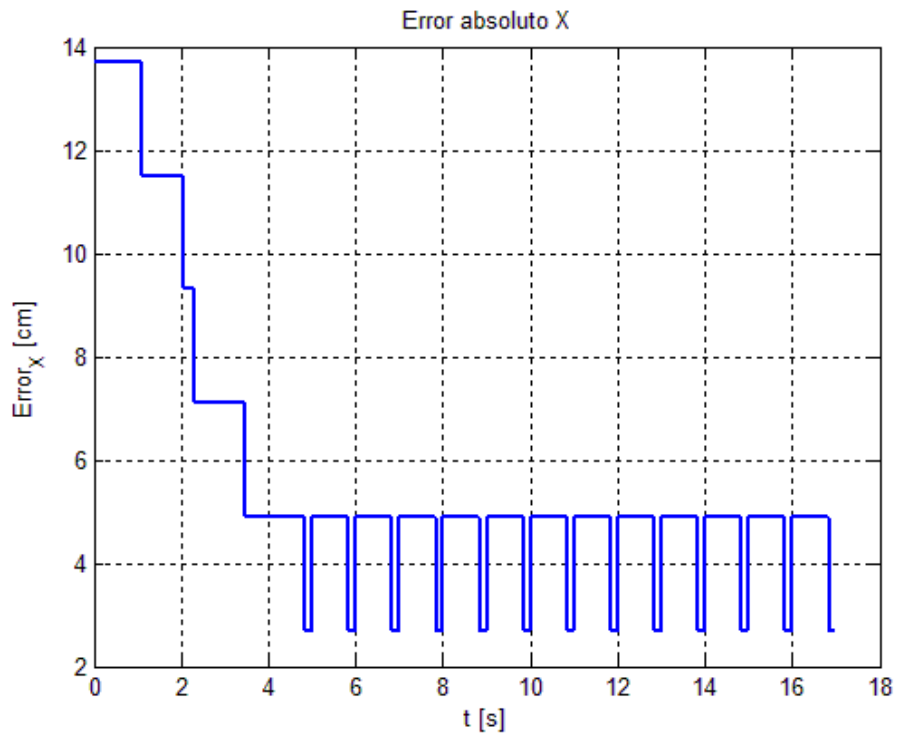


Fig. 5.1.1d Error absoluto en X [3].

Para la gráfica de la figura 5.1.1c se observa la trayectoria para el eje X, y se aprecia un desfase o error entre la línea verde y azul, de entre 4 a 5 cm, corroborando esto en la figura 5.1.1d.

La línea inicial en la figura 5.1.1d indica que el móvil inicia su desplazamiento en la trayectoria establecida por la fórmula, recordando también que este se mueve a través de él mismo como sistema de referencia.

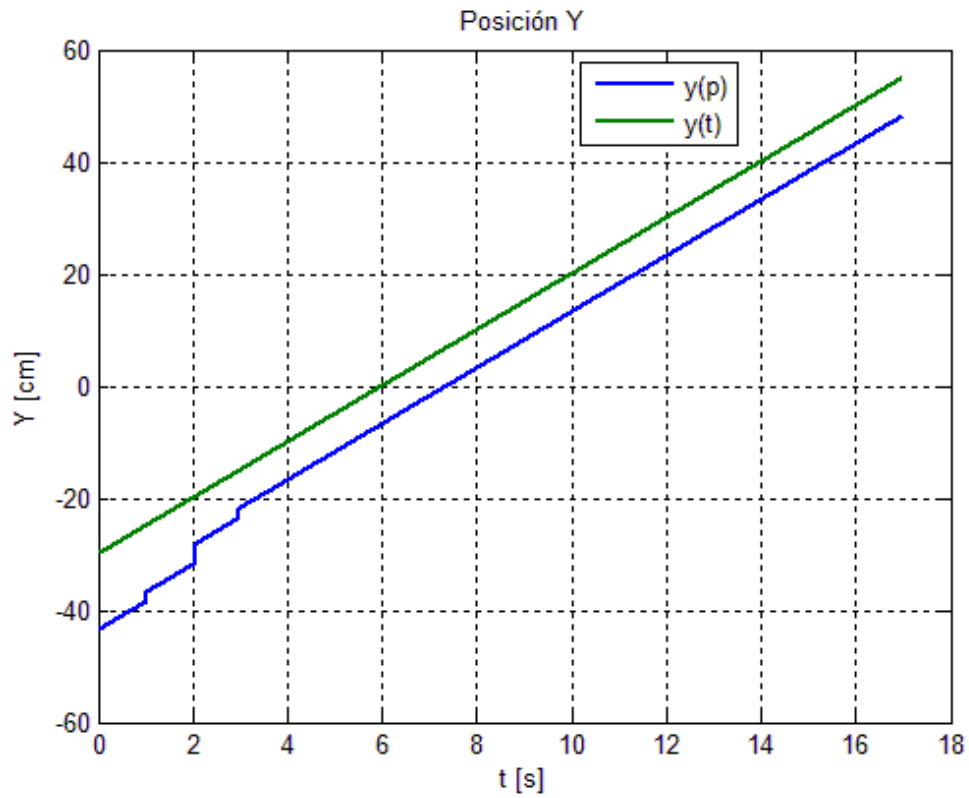


Fig. 5.1.1e Posición descrita para $y(t)$ vs $y(t)$ [3].

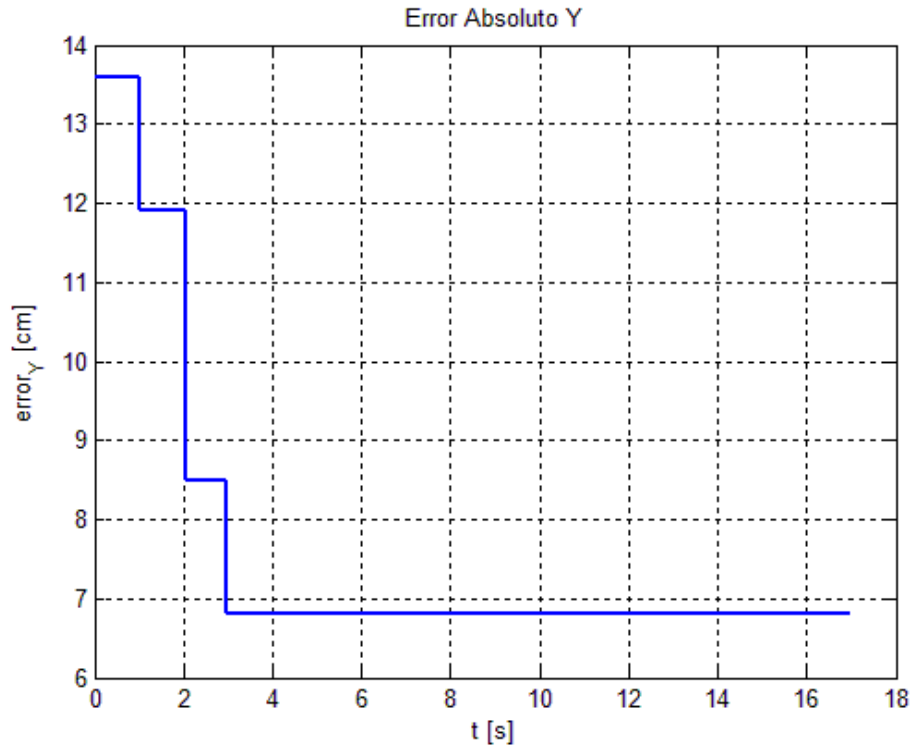


Fig. 5.1.1f Error absoluto en Y [3].

En la figura 5.1.1e, se aprecia que la trayectoria marcada para el eje Y es más continua que para el eje X, solo que en esta, el desfase o error con respecto a la recta verde es de 6.7 [cm] el cual podemos verificar en la figura 5.1.1f.

Al comienzo de la gráfica de la figura 5.1.1f se nota que existe un error en cuanto a la posición, el cual se va reduciendo. Esto se debe a que el móvil intenta seguir la trayectoria establecida mediante la fórmula.

En cuanto a la figura 5.1.1g, la gráfica muestra que el error es prácticamente cero durante toda la trayectoria, lo cual es lo esperado para esta prueba. La aseveración se corrobora en la figura 5.1.1h. Al final de la trayectoria se observa una ligera rotación en el robot provocada por un ligero deslizamiento, esto se puede ver en las dos figuras anteriores.

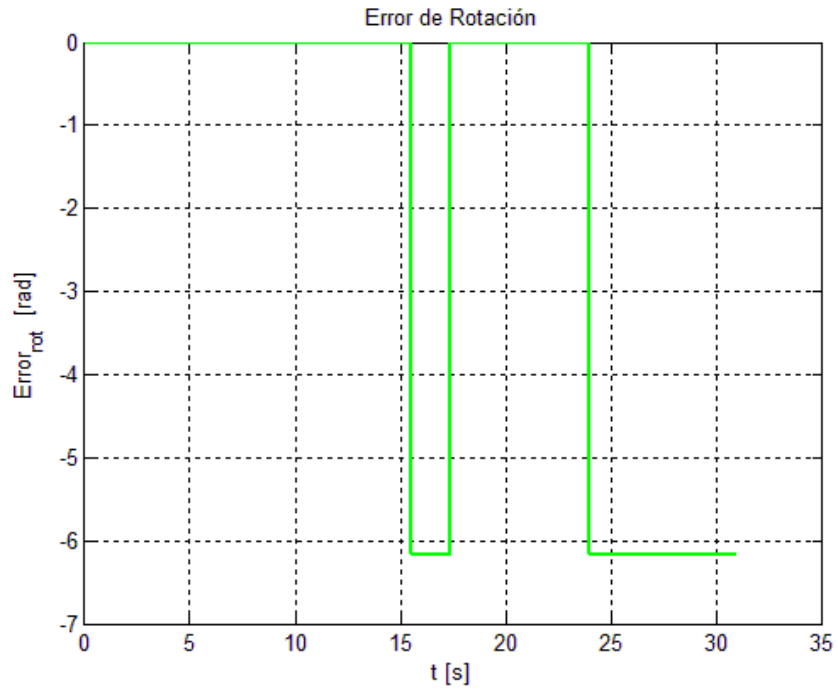


Fig. 5.1.1g Posición descrita para el giro de theta(P) vs theta(t) [3].

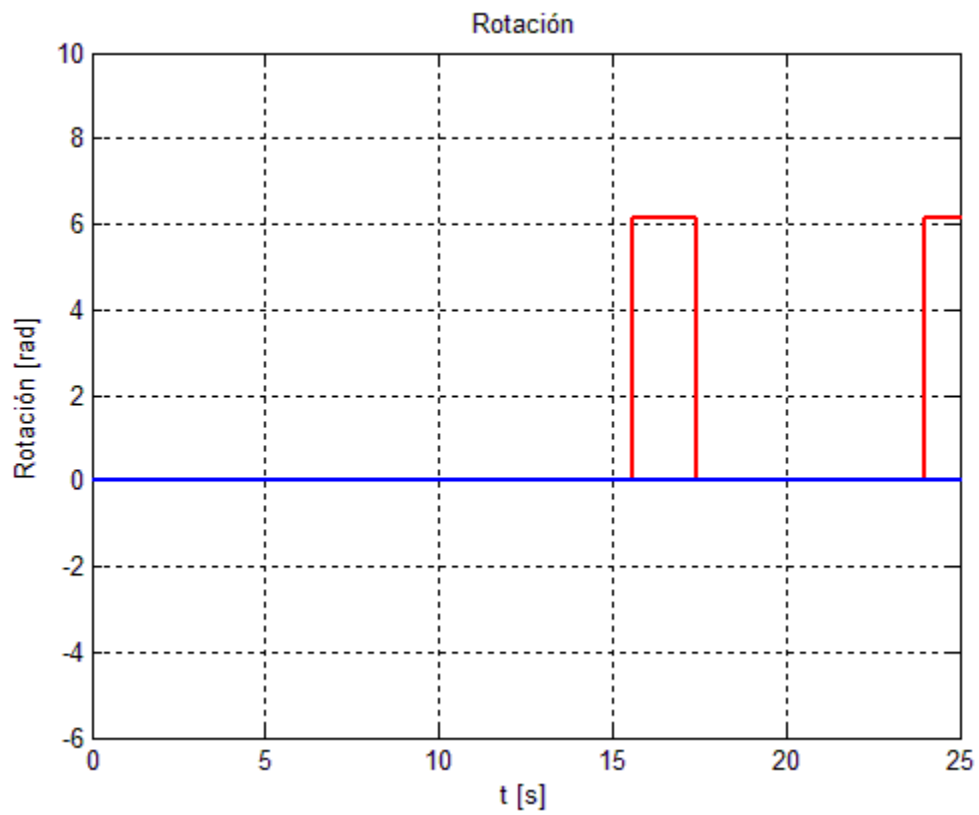


Fig. 5.1.1h Error absoluto de rotación del robot [3].

5.2. Resultados de las pruebas en Lazo Cerrado

5.2.1. Seguimiento de una trayectoria rectilínea del punto P de la plataforma con orientación de 0°.

Esta recta comienza en el punto $\{-30,-30\}$. El punto objetivo requerido por la ley de campos potenciales pertenece a la recta generada con las ecuaciones paramétricas que se muestran a continuación:

$$\begin{cases} x(t) = 5 \cdot t - 30 \\ y(t) = 5 \cdot t - 30 \end{cases} \quad (6.1)$$

Debido a que la ley de control de velocidad ya fue definida en el capítulo 4, para lograr la orientación a 0° respecto al sistema coordenado $\{X,Y\}$, se utiliza la ecuación 4.19 para el control de la velocidad angular.

- Parámetros
 - $v_{xmáx} = 3 \left[\frac{m}{s} \right]$
 - $v_{ymáx} = 2 \left[\frac{m}{s} \right]$
 - $\omega_{máx} = 0 \left[\frac{rad}{s} \right]$
 - $k_r = 20 [cm]$
 - $k_{OWR} = 0 [cm]$
 - $\theta_p = 0^\circ$

En esta prueba el robot empieza en un punto cualquiera, y con una orientación cualquiera. Puede observarse como rápidamente trata de llegar al punto inicial de la recta generada, al mismo tiempo que es capaz de orientarse como se le pidió.

En la figura 5.2a se muestra la simulación de la plataforma al tratar de seguir la trayectoria propuesta. La figura 5.2b muestra en color azul la trayectoria del robot y en color magenta la trayectoria propuesta.

Por supuesto que nunca va a alcanzar el punto de la trayectoria, ya que dicho punto avanza constantemente y eso es observable en los pequeños picos que sobresalen en la recta de color azul. También se observa que esta recta se ve más definida que la de lazo abierto, lo cual confirma el correcto funcionamiento del sistema de retroalimentación.

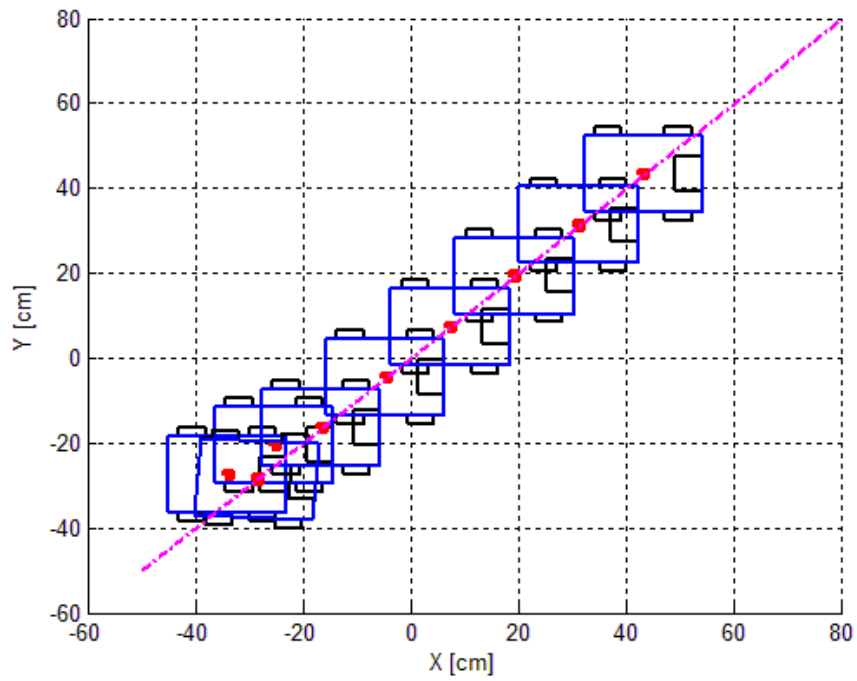


Fig. 5.2.1a Simulación de seguimiento de trayectoria rectilínea en lazo cerrado [3].

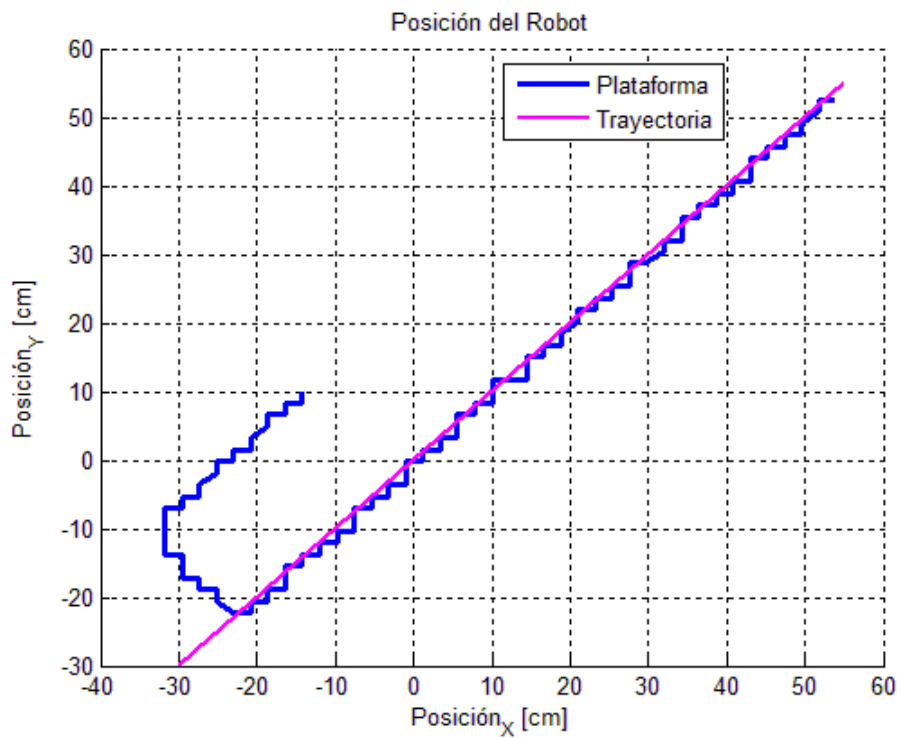


Fig. 5.2.1b Comparación de la trayectoria del robot vs la trayectoria propuesta [3].

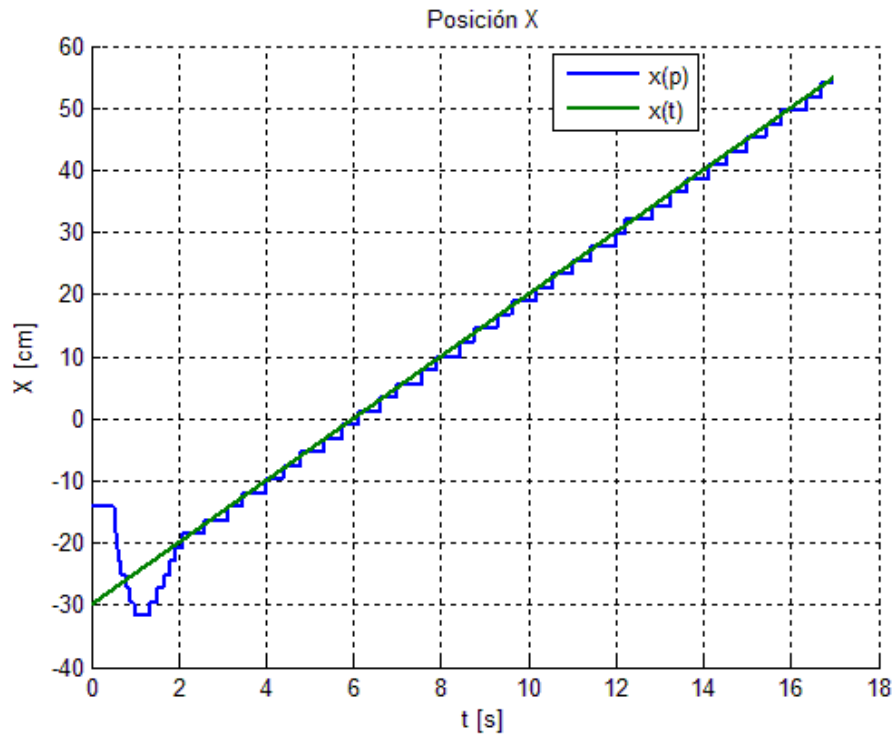


Fig. 5.2.1c Desplazamiento en X [3].

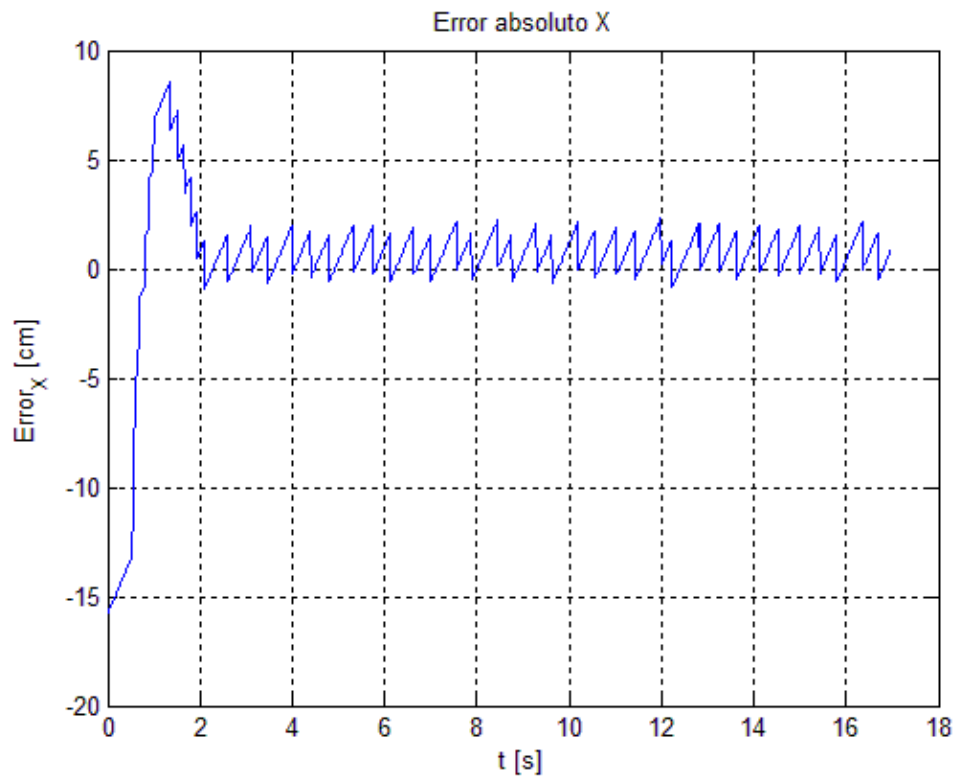


Fig. 5.2.1d Error de desplazamiento en X [3].

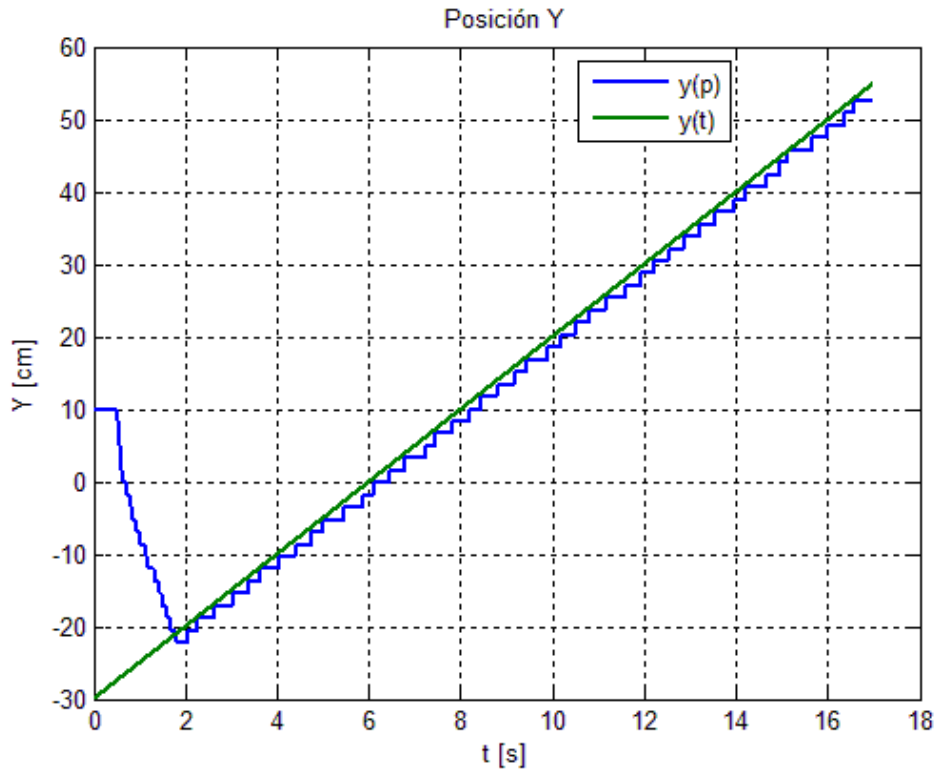


Fig. 5.2.1e Desplazamiento en Y [3].

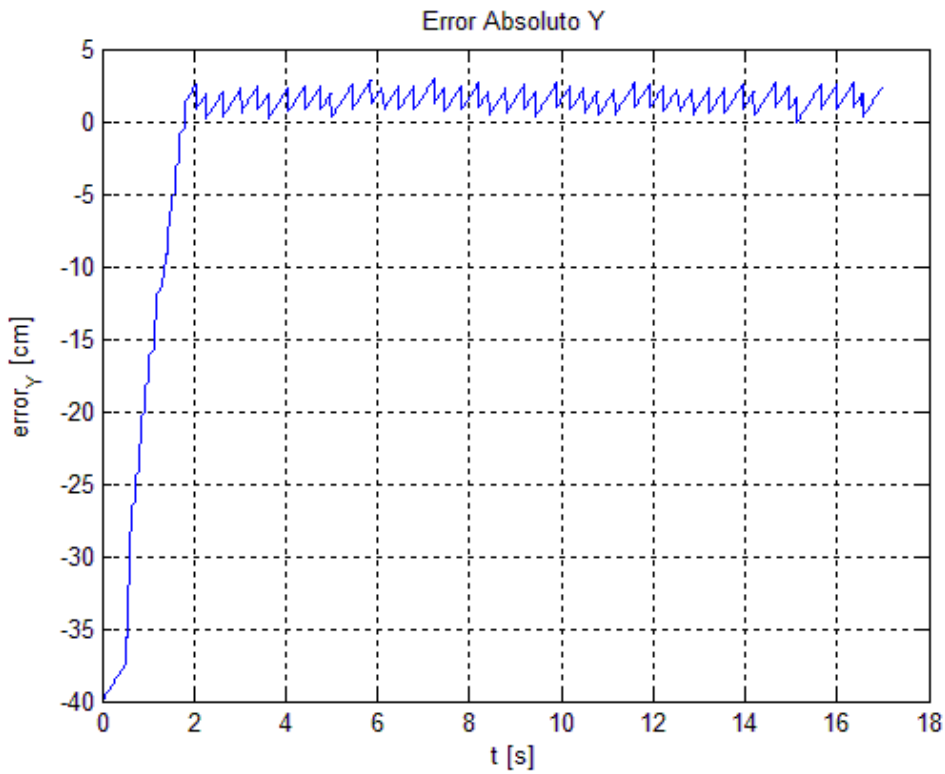


Fig. 5.2.1f Error de desplazamiento en Y [3].

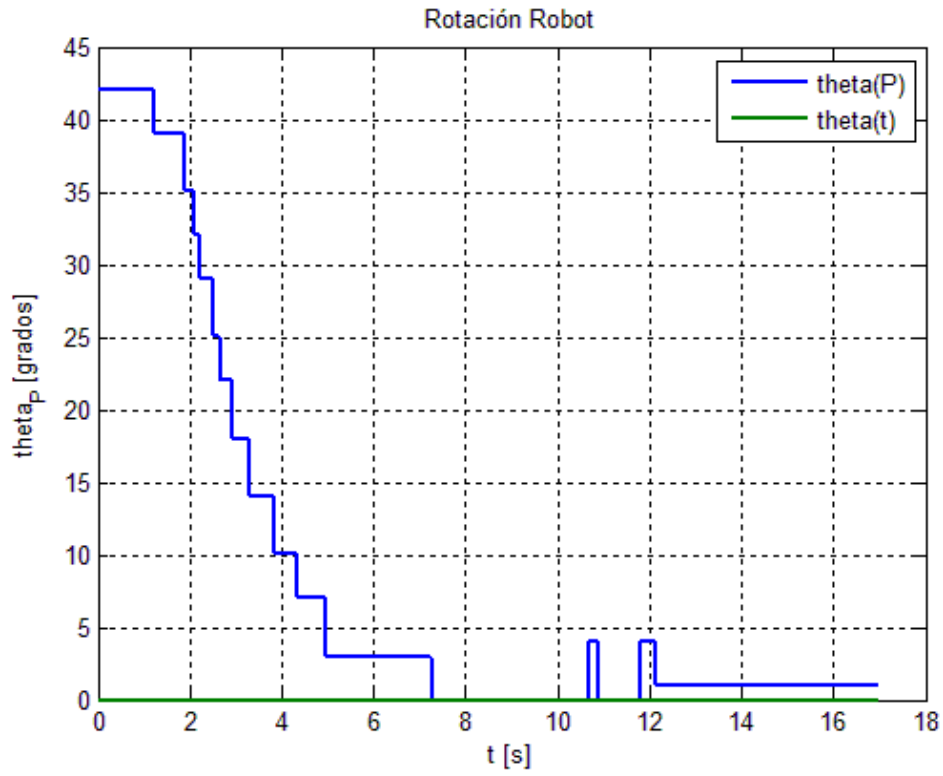


Fig. 5.2.1g Ángulo de rotación de la Plataforma [3].

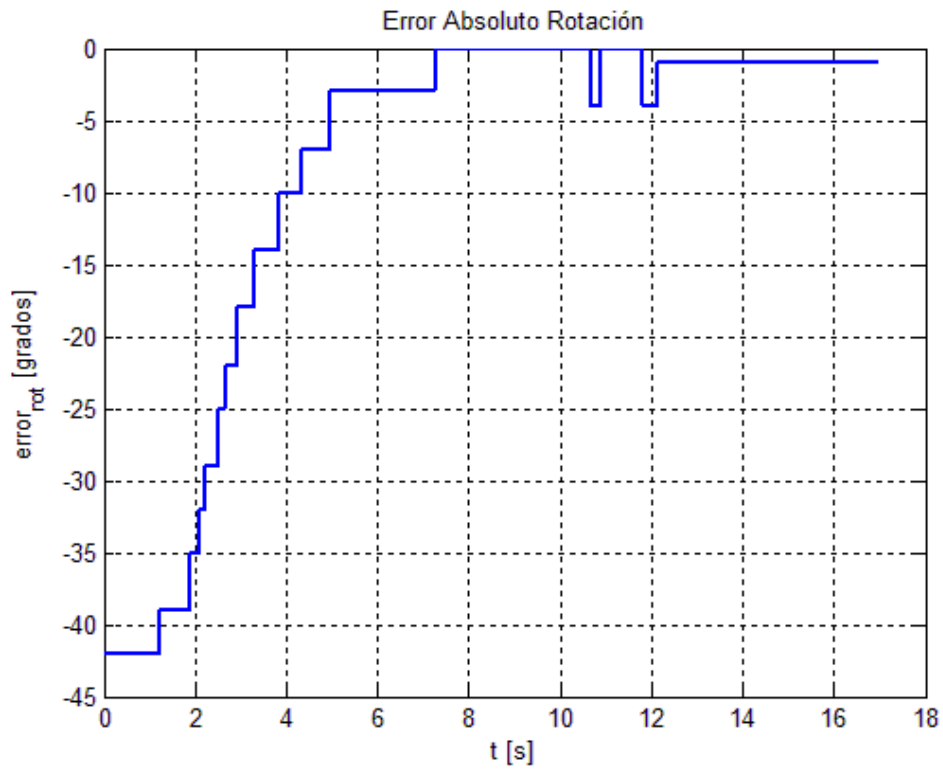


Fig. 5.2.1h Error de rotación de la Plataforma [3].

En estas gráficas se observa que el sistema de control tiene la capacidad de corregir errores, también es observable que el punto P no alcanza al punto objetivo. Mientras el error en el desplazamiento lineal tiende a estabilizarse, el del ángulo de rotación tiende a cero puesto que el ángulo de rotación es fijo. Así se comprueba de manera simulada que el sistema de control funciona correctamente.



5.2.2 Seguimiento de una trayectoria circular del punto P de la plataforma con una velocidad angular ω .

Para crear esta trayectoria se utilizaron las ecuaciones paramétricas que describen las curvas de Lissajous.

Con el propósito de demostrar algunas de las capacidades del robot se escogió una posición y orientación al azar para el controlador y los campos potenciales. La recta se rige por el siguiente sistema de ecuaciones paramétricas:

$$\begin{cases} x(t) = A \cdot \sin(\omega_x \cdot t + \delta_x) \\ y(t) = B \cdot \sin(\omega_y \cdot t + \delta_y) \end{cases} \quad (6.2)$$

Dónde:

1. A y B representan amplitudes.
2. ω_x y ω_y son frecuencias en radianes.
3. δ_x y δ_y son fases en radianes.

Para generar esta curva se utilizaron los siguientes parámetros:

- $v_{xmáx} = 3 \left[\frac{m}{s} \right]$
- $v_{ymáx} = 2 \left[\frac{m}{s} \right]$
- $\omega_{máx} = -\frac{\pi}{9} \left[\frac{rad}{s} \right]$
- $k_r = 25 [cm]$
- $k_{OWR} = 0 [cm]$
- $\theta_p = 0^\circ$
- Parámetros de las curvas de Lissajous
 - $A = B = 40$
 - $\omega_x = \omega_y = \frac{\pi}{20}$
 - $\delta_x = \frac{\pi}{2}$
 - $\delta_y = \pi$

En la figura 5.2.2b se puede apreciar la trayectoria que debe realizar el robot (círculo rosa) y la trayectoria que realiza (círculo azul).

Antes de que el robot empiece a realizar el círculo se observa cierta curva (figura 5.2.2b), esto se debe a que el robot se acomoda para comenzar la trayectoria

establecida antes de iniciar su recorrido, los otros picos que se ven son momentos en que el fiducial se pierde por la cámara.

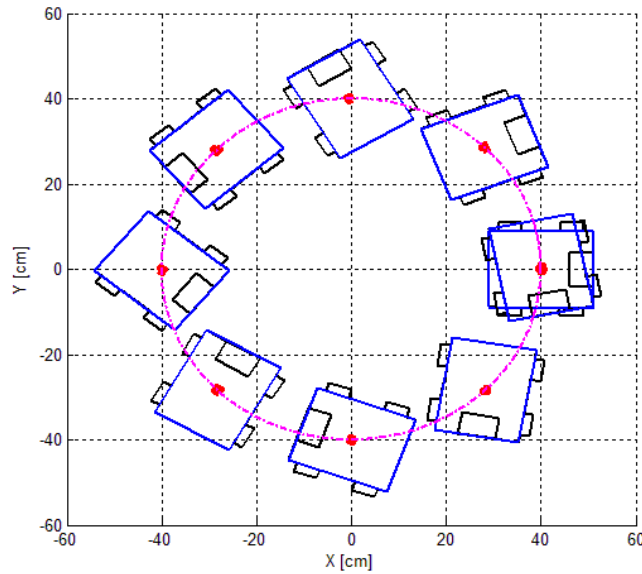


Fig. 5.2.2a Simulación de círculo en lazo cerrado con giro [3].

En la figura 5.2.2a se aprecia como el móvil está realizando una figura circular al mismo tiempo que gira sobre sí mismo, corroborando así su omnidireccionalidad. Se distingue la trayectoria que sigue el robot por medio de la línea azul, y se observa cómo este intenta realizar la circunferencia (de color rosa) lo más exacto posible.

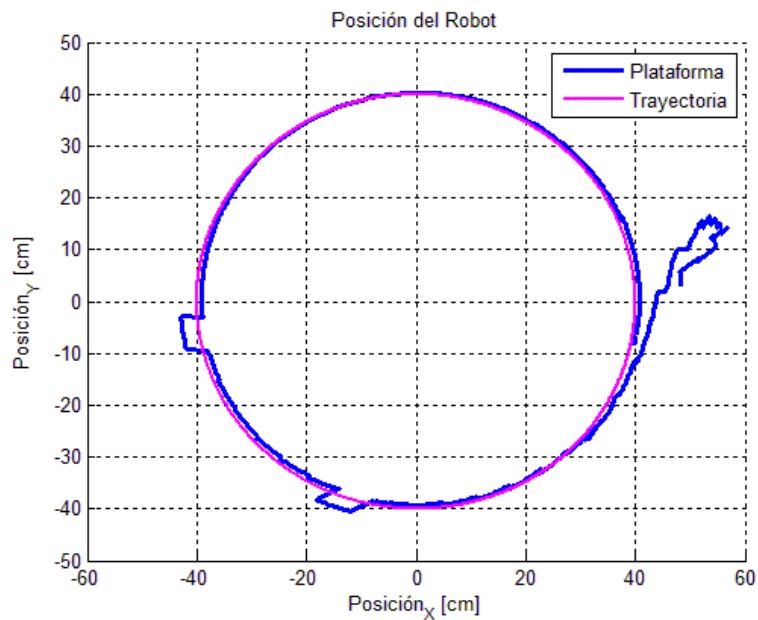


Fig. 5.2.2b Posición del robot vs trayectoria [3].

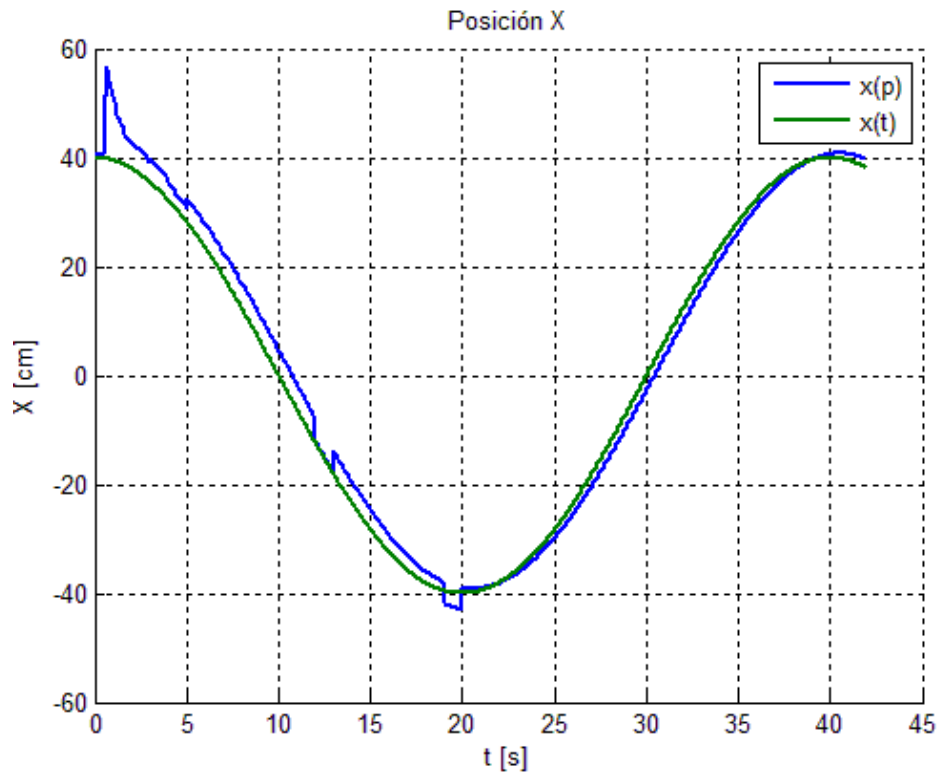


Fig. 5.2.2c Desplazamiento en X [3].

En la gráfica de la figura 5.2.2c se observa una sinusoidal que es necesaria para que nos proporcione los valores en X y así formar el círculo. Los valores graficados en azul, pertenecen al resultado del comportamiento de la prueba del robot en movimiento, al compararse con la curva verde, son casi idénticos. Los picos que salen de la trayectoria son errores provocados porque en algunos instantes el marcador de referencia (fiducial) no es captado por la cámara.

En esta gráfica también se comprueba el funcionamiento en conjunto de la cinemática y los campos potenciales; la primera para controlar al robot para que pueda seguir la trayectoria, y el segundo para que realice a la velocidad que se le indica sin perderse. En la figura 5.2.2d se muestra el error obtenido para el eje x .

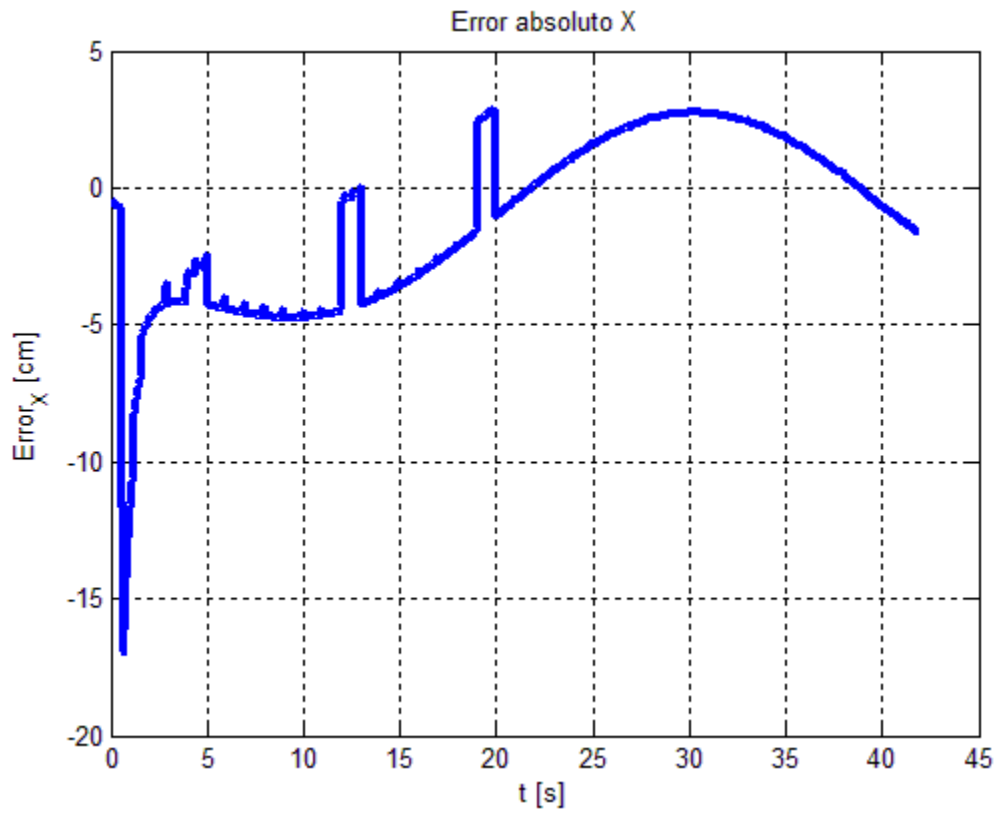


Fig. 5.2.2d Error de desplazamiento en X [3].

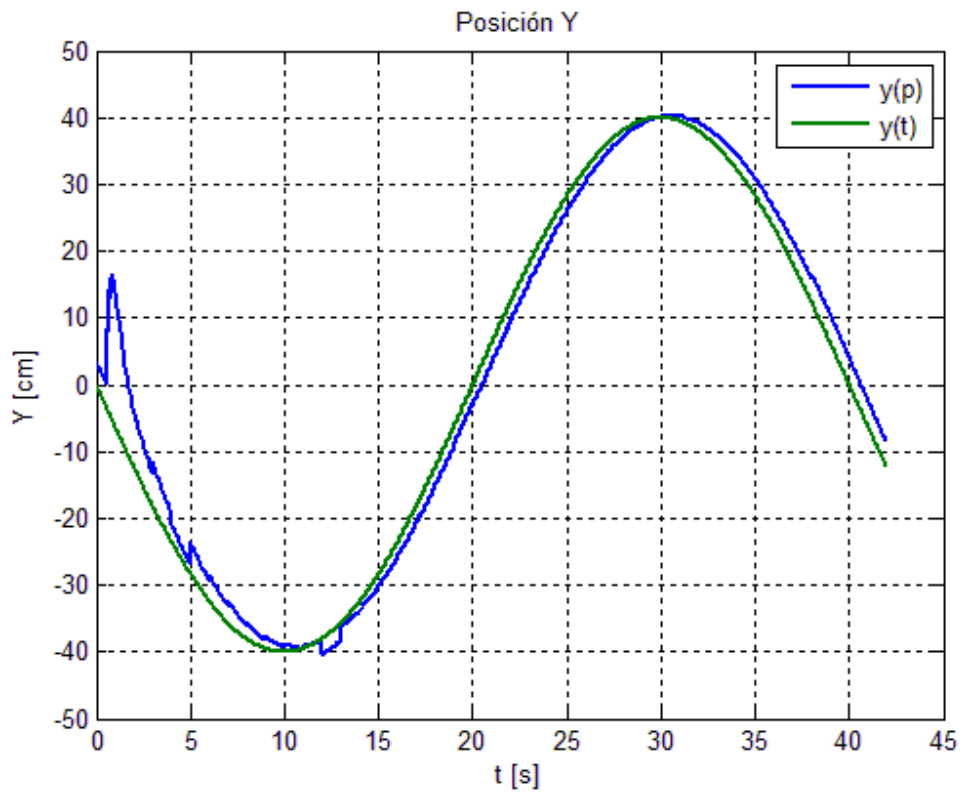


Fig. 5.2.2e Desplazamiento en Y [3].

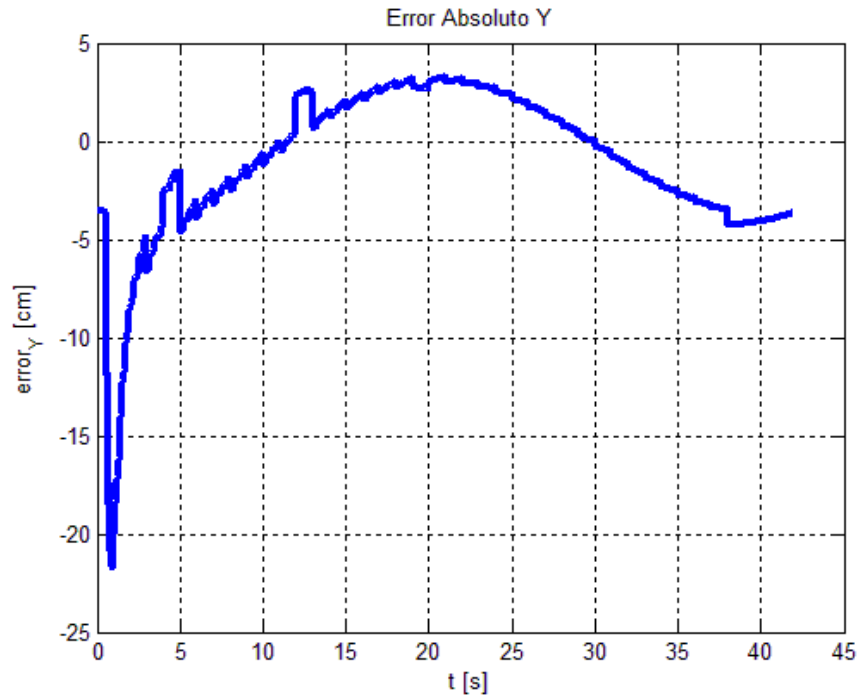


Fig. 5.2.2f Error de desplazamiento en Y [3].

Para la figura 5.2.2e y la figura 5.2.2f se observa una sinusoidal para la posición y error de desplazamiento en el eje Y. La figura 5.2.2e describe la posición de X y la figura 5.2.2c el error de X.

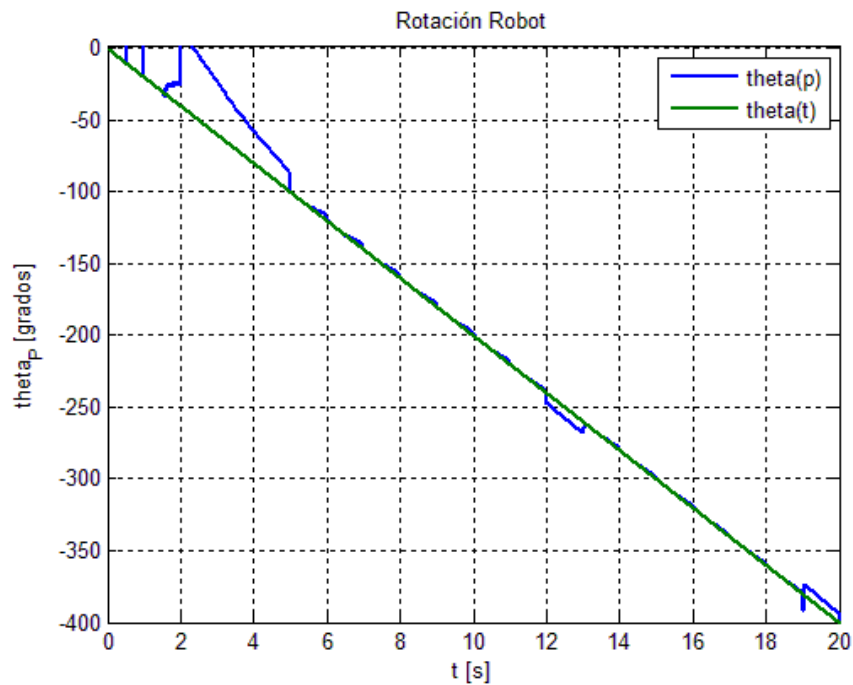


Fig. 5.2.2g Ángulo de rotación [3].

La figura 5.2.2g muestra los valores que toma theta conforme se está desplazando el robot. Al principio se ve cómo la posición angular del robot es casi cero, pero al momento de reaccionar tiene que girar varios grados para alcanzar la posición angular requerida, y después se aprecia en la figura 5.2.2h cómo el error de rotación tiende a cero.

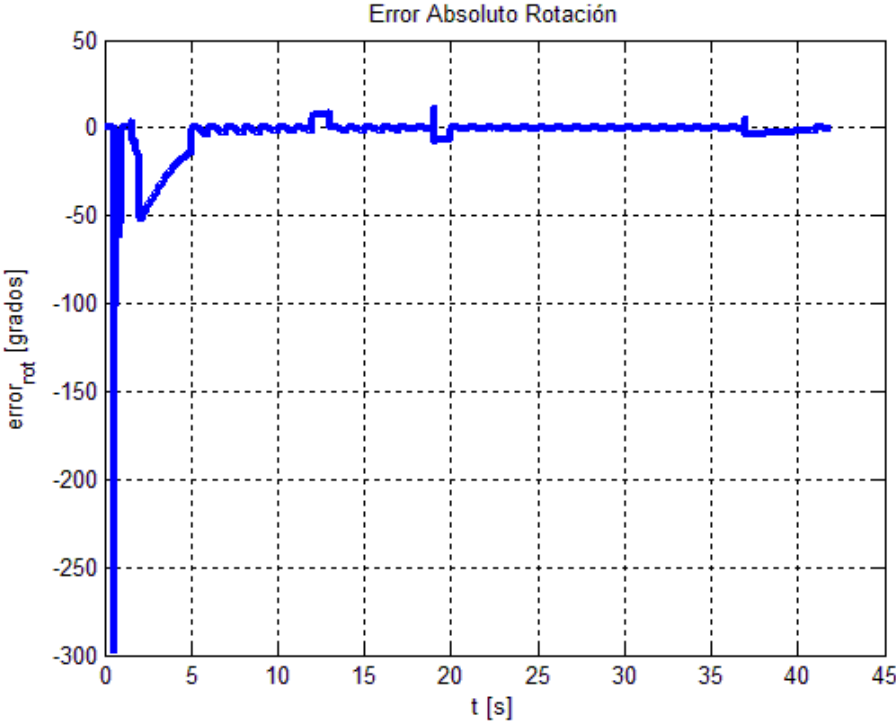


Fig. 5.2.2h Error del ángulo de rotación [3].

5.2.3 Seguimiento de Curva de Lissajous denominada "Kiss" con velocidad angular ω .

La trayectoria de la curva "Kiss" de Lissajous también fue generada utilizando las ecuaciones paramétricas descritas anteriormente. Esto demostró la capacidad del robot para seguir trayectorias más complejas.

Al igual que en el círculo, se escogió una posición y orientación aleatorias para observar la acción del controlador y los campos potenciales.

Para generar esta curva se utilizaron los siguientes parámetros:

- $v_{xmáx} = 3 \left[\frac{m}{s} \right]$
- $v_{ymáx} = 2 \left[\frac{m}{s} \right]$
- $\omega_{máx} = -\frac{\pi}{9} \left[\frac{rad}{s} \right]$
- $k_r = 25 [cm]$
- $k_{OWR} = 0 [cm]$
- $\theta_p = 0^\circ$
- Parámetros de las curvas de Lissajous
 - $A = B = 30$
 - $\omega_x = \frac{\pi}{24}, \omega_y = \frac{\pi}{36}$
 - $\delta_x = \frac{\pi}{2}$
 - $\delta_y = \frac{\pi}{2}$

La simulación muestra un movimiento mucho más complejo que una circunferencia o una recta, a pesar de que la plataforma se encuentra girando, el punto **P** no deja de seguir la trayectoria.

La figura 5.2.3b muestra la trayectoria de la plataforma y la curva a seguir. Se observa claramente que al existir una velocidad angular, el error en el desplazamiento aumenta.

En la figura 5.2.3g se observa que el giro de la plataforma no es completamente uniforme como debería ser, pero aun así, una vez que se estabiliza, mantiene un giro constante y el error también se hace constante ya que el ángulo, está cambiando respecto al tiempo.

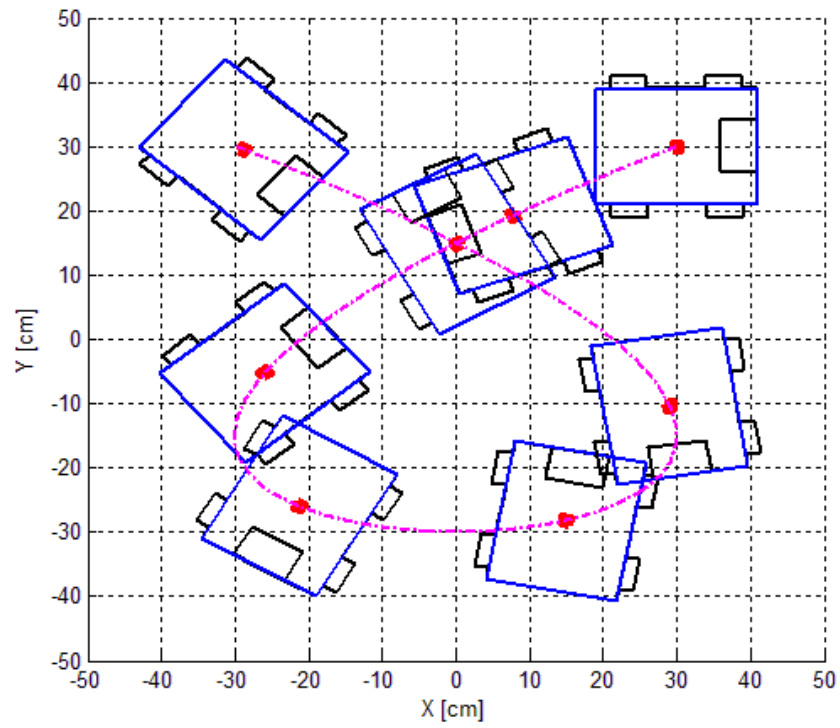


Fig. 5.2.3a Simulación de seguimiento de trayectoria compleja [3].

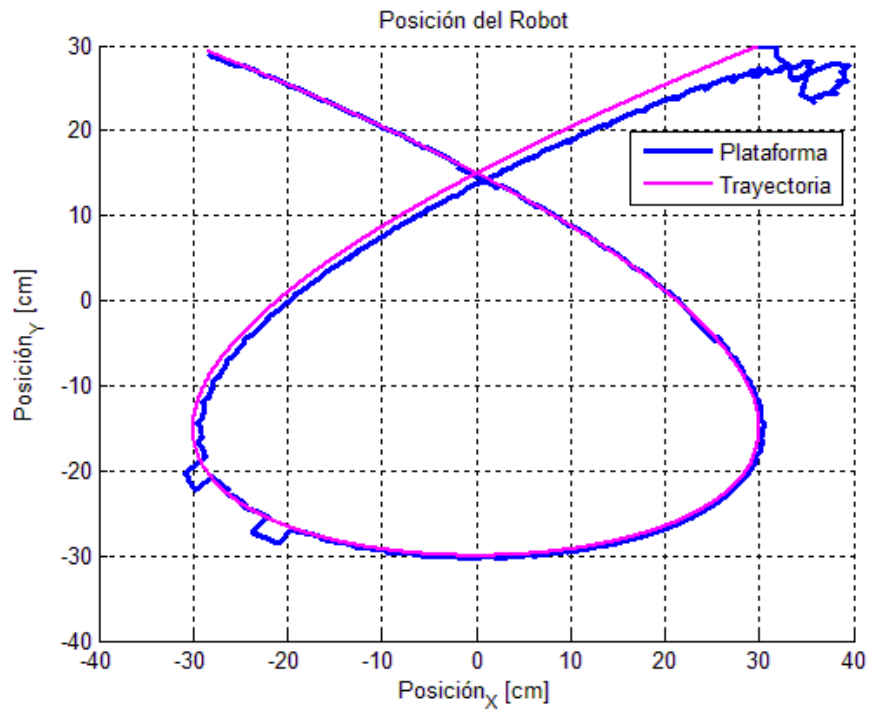


Figura 5.2.3b Comparación de la trayectoria del robot vs la trayectoria propuesta [3].

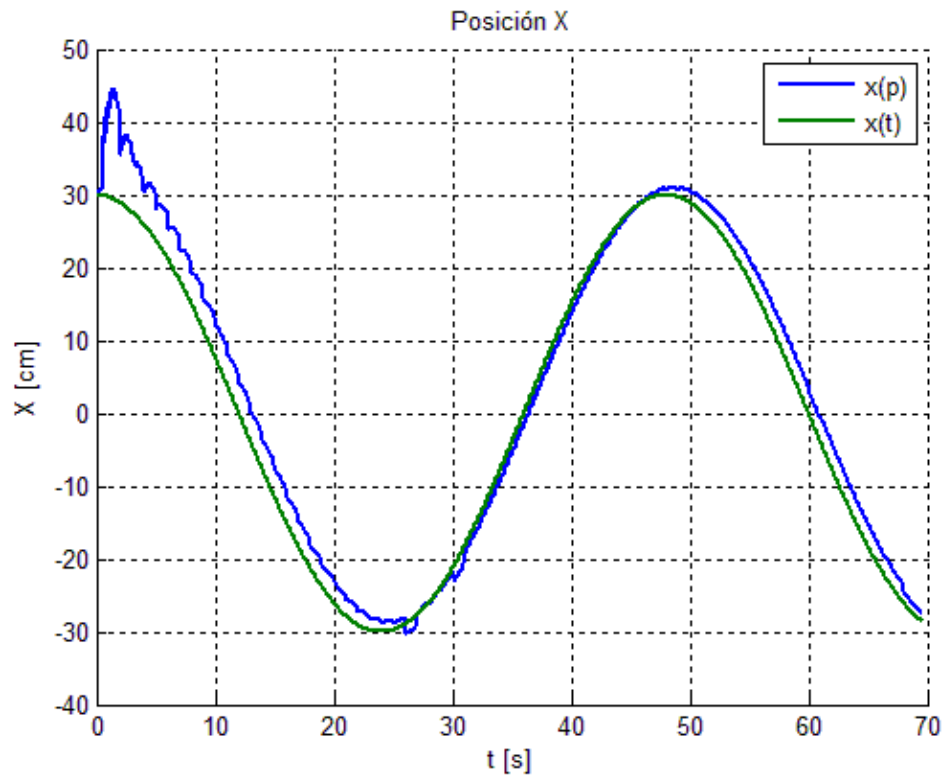


Fig. 5.2.3c Desplazamiento en X [3].

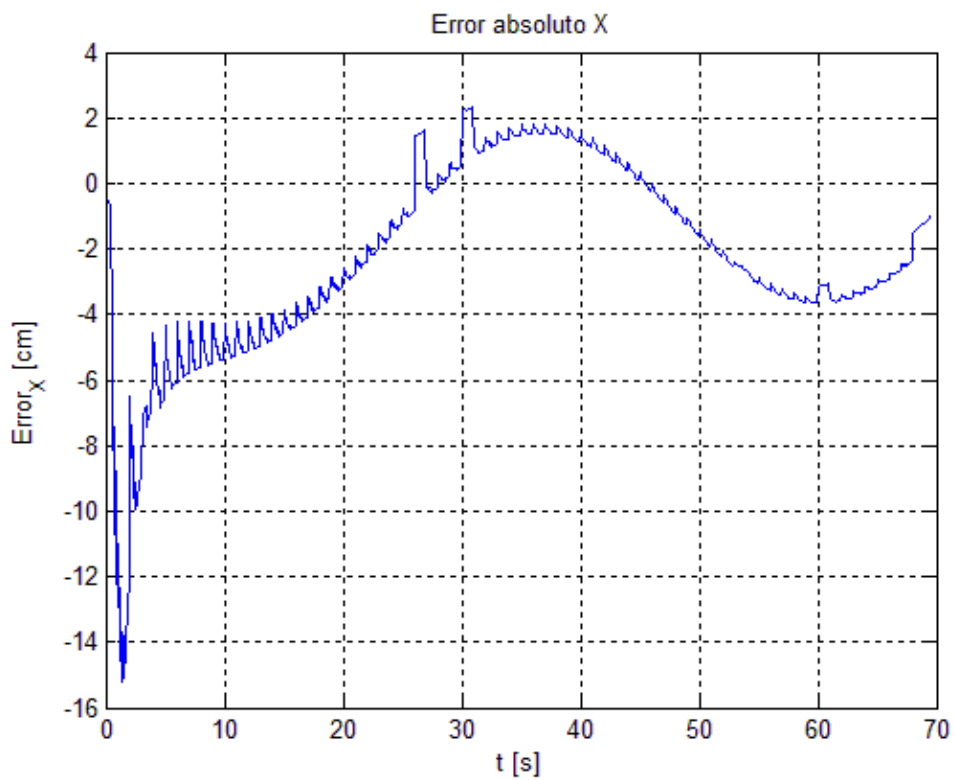


Fig. 5.2.3d Error de desplazamiento en X [3].

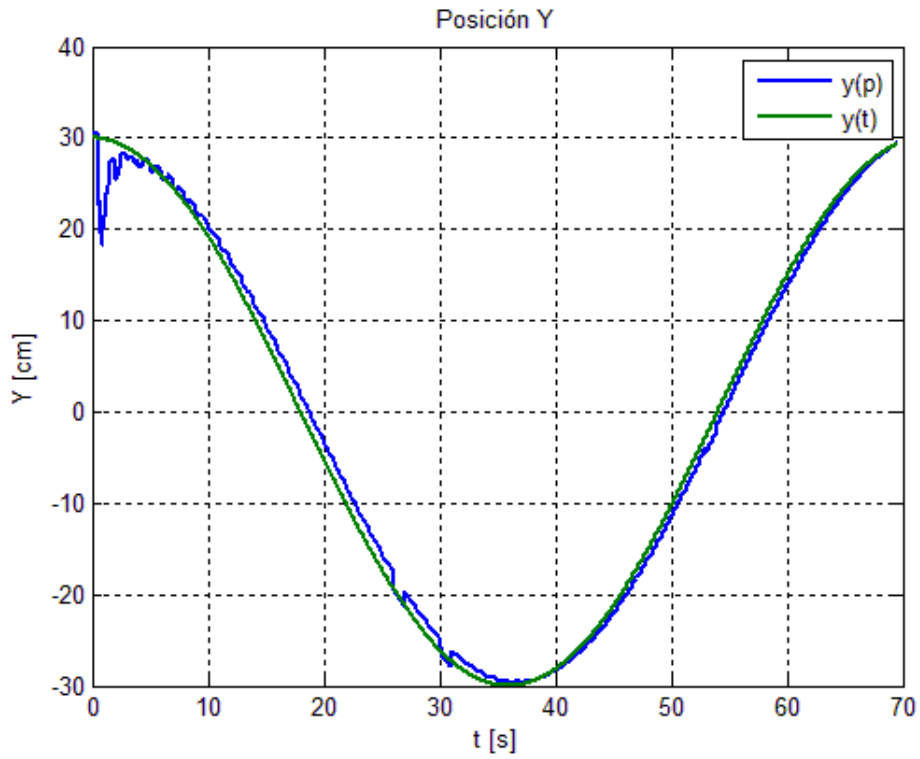


Fig. 5.2.3e Desplazamiento en Y [3].

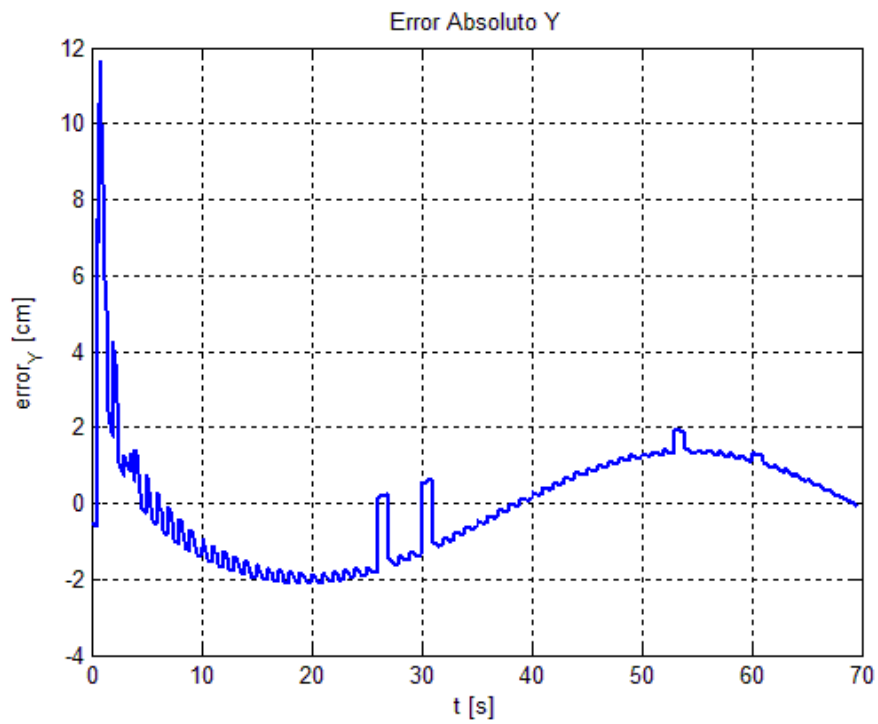


Fig. 5.2.3f Error de desplazamiento en Y [3].

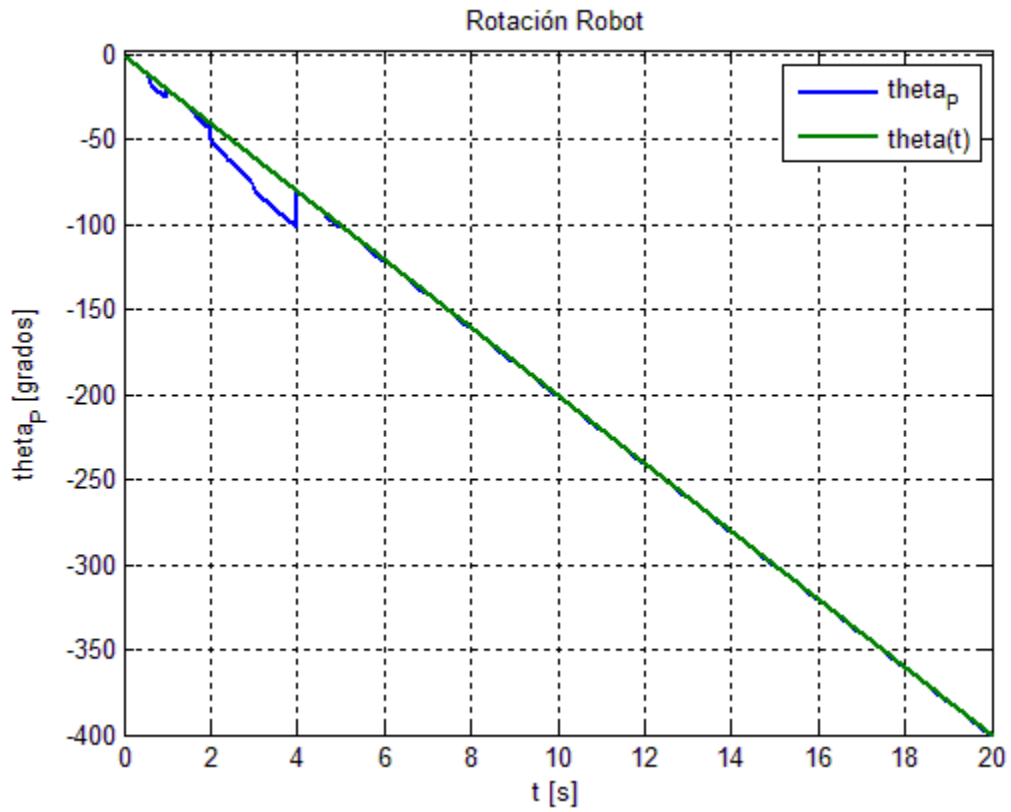


Fig. 5.2.3g Ángulo de rotación de la plataforma [3].

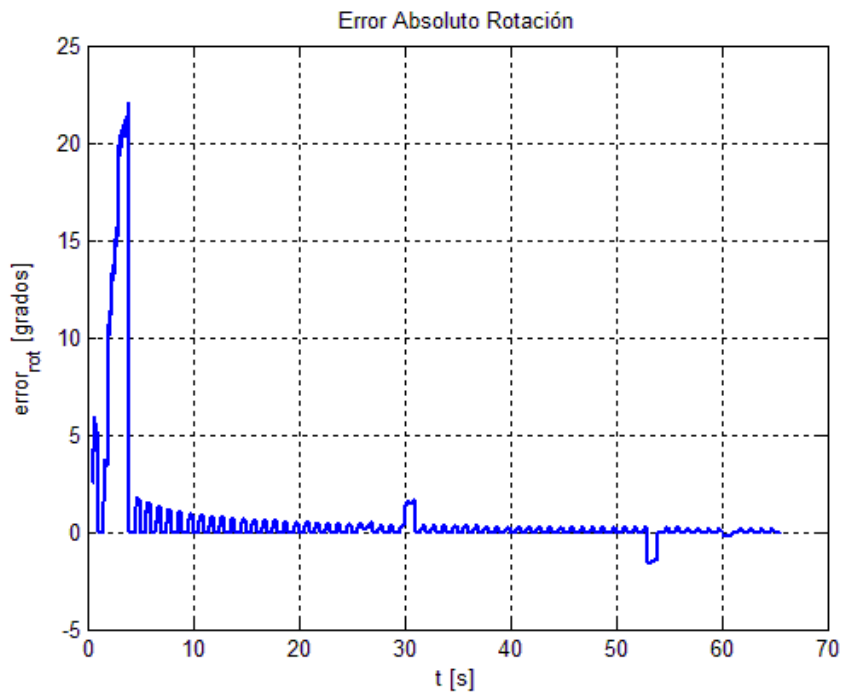


Fig. 5.2.3h Error del ángulo de rotación [3].

5.2.4 Seguimiento de Curva de Lissajous denominada "Pop" con inclinación de 20°.

Al igual que las dos trayectorias anteriores, esta también fue generada utilizando las ecuaciones paramétricas de las Curvas de Lissajous. La diferencia con la prueba anterior es que ahora se mantiene constante un ángulo de 20°.

Al igual que en la prueba anterior se escogió una posición y orientación aleatorias para que sea observable la acción del controlador y los campos potenciales. Para generar esta curva se utilizaron los siguientes parámetros:

- $v_{xmáx} = 3 \left[\frac{m}{s} \right]$
- $v_{ymáx} = 2 \left[\frac{m}{s} \right]$
- $\omega_{máx} = -\frac{\pi}{9} \left[\frac{rad}{s} \right]$
- $k_r = 20 [cm]$
- $k_{OWR} = 0 [cm]$
- $\theta_p = 20^\circ$
- Parámetros de las curvas de Lissajous
 - $A = B = 30$
 - $\omega_x = \frac{\pi}{24}, \omega_y = \frac{\pi}{36}$
 - $\delta_x = \frac{\pi}{9}$
 - $\delta_y = \frac{\pi}{2}$

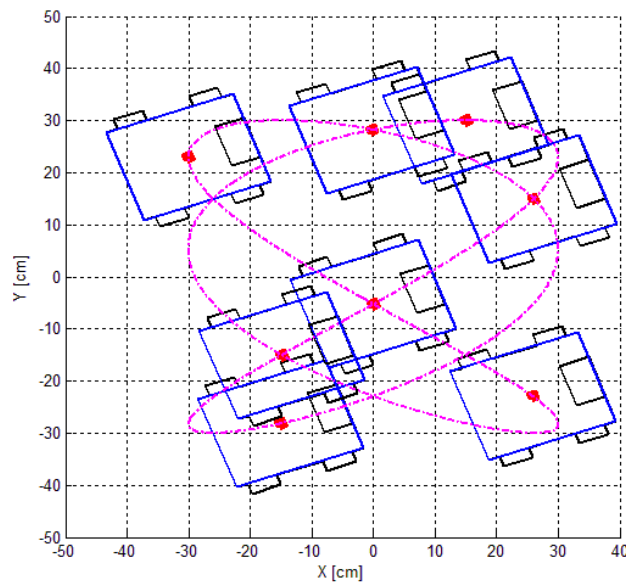


Fig. 5.2.4a Simulación de la trayectoria que sigue la Plataforma [3].

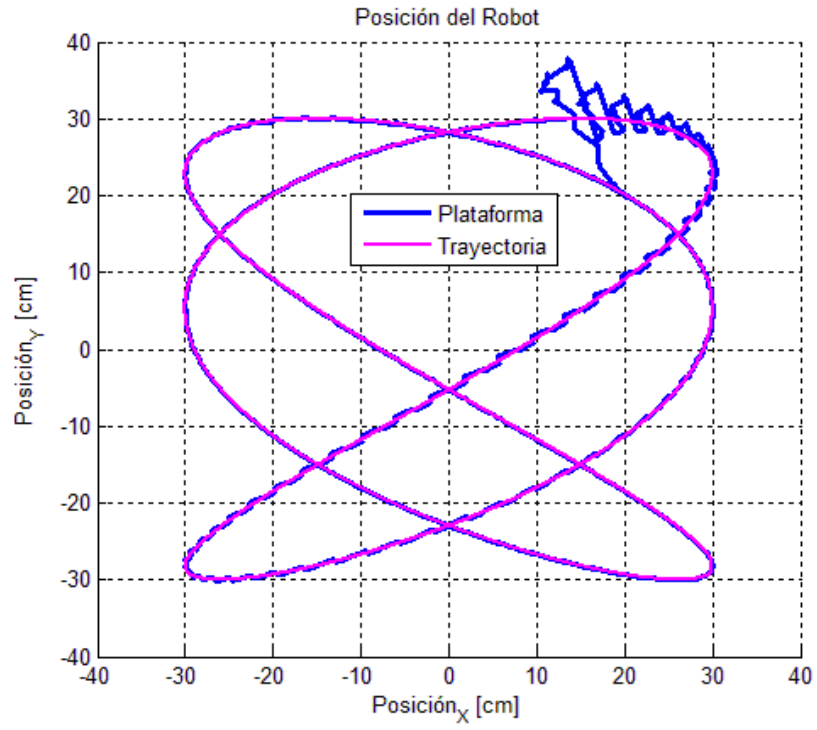


Fig. 5.2.4b Simulación de la trayectoria que sigue la plataforma vs la curva [3].

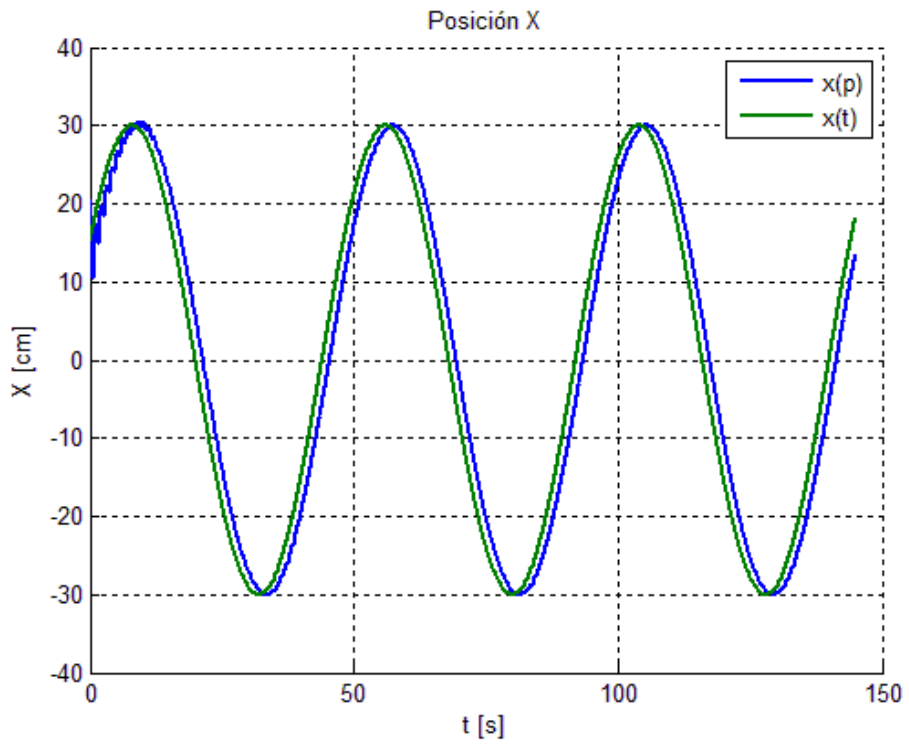


Fig. 5.2.4c Desplazamiento en X [3].

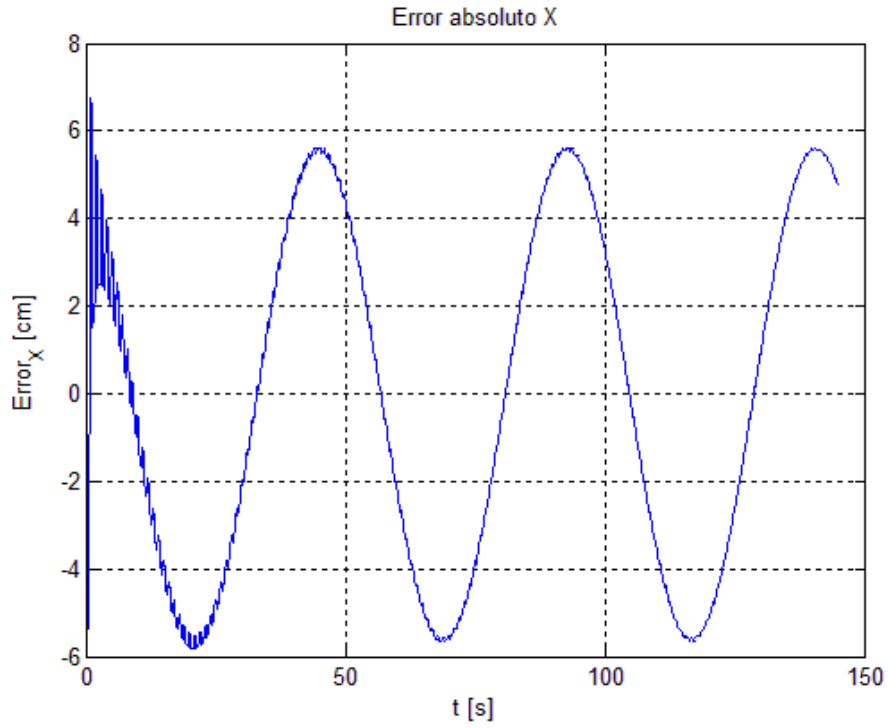


Fig. 5.2.4d Error de desplazamiento en X [3].

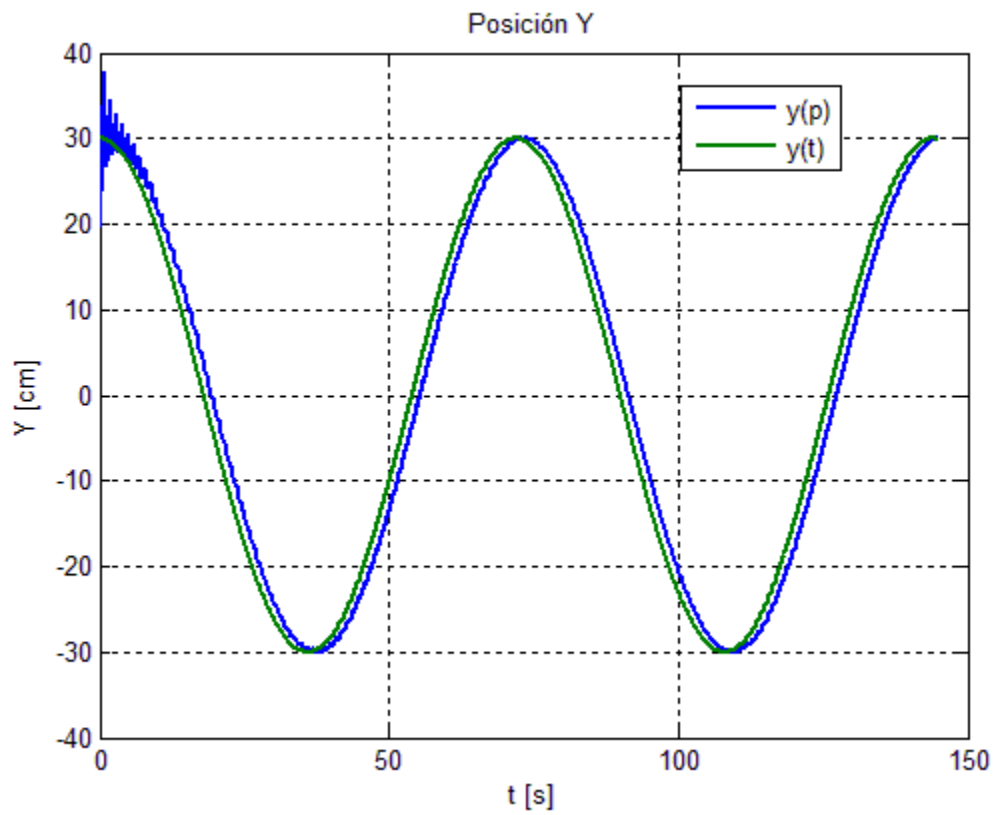


Fig. 5.2.4e Error de desplazamiento en Y [3].

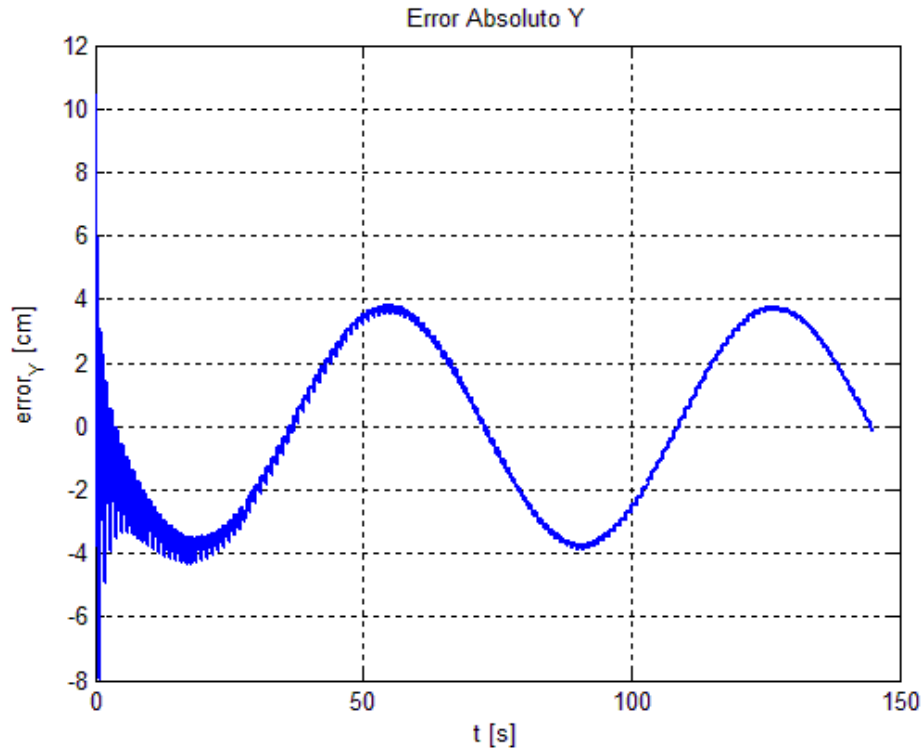


Fig. 5.2.4f Error de desplazamiento en X [3].

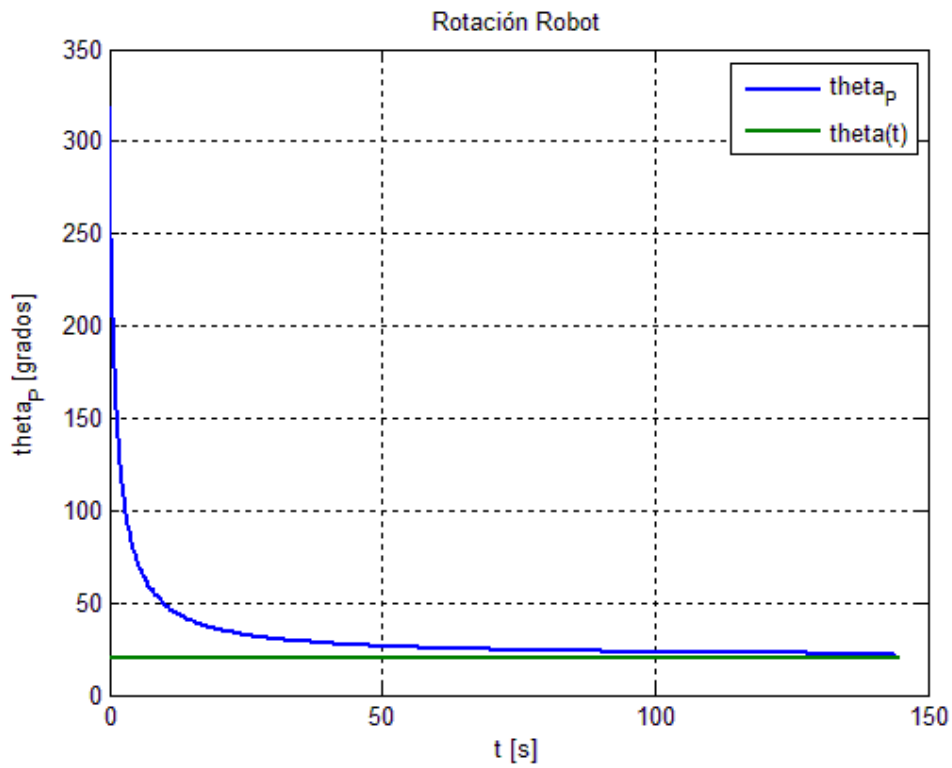


Fig. 5.2.4g Ángulo de inclinación del robot [3].

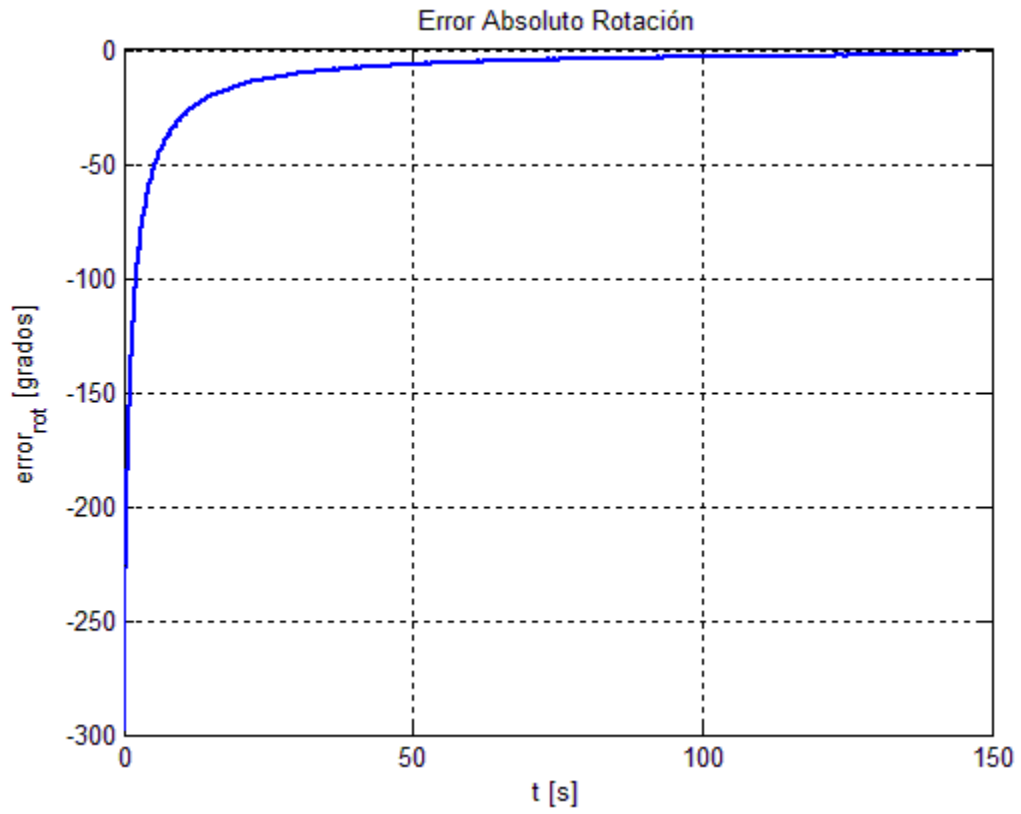


Fig. 5.2.4f Error del ángulo de inclinación [3].

Capítulo 6. Conclusiones y Trabajo a Futuro

6.1 Conclusiones

Podemos afirmar que la cinemática desarrollada por González Villela para este robot con llantas Mecanum es funcional. Esto quiere decir que el análisis cinemático es correcto y que se puede aplicar para cualquier otra configuración como las mostradas en el capítulo 2. Ya se ha estado utilizando este tipo de robots, pero con una configuración diferente.

Retomando el trabajo de tesis de Arturo [9], hay tres aspectos muy importantes que son necesarios mencionar:

1. El primero de ellos es que al no ser este un robot convencional, la teoría unificadora de González Villela, V. (2006) [6] sigue funcionando.
2. El otro aspecto es que la configuración de llantas combinada con la cinemática nos dice que el robot funciona omnidireccionalmente, esto se ha demostrado de manera teórica y también práctica gracias a las pruebas que se realizaron. En dichas pruebas se observó la manipulación de los tres grados de libertad del robot en el plano, todos ellos de manera independiente en la mayoría de las pruebas.
3. El tercer aspecto es que la ley de control para robots móviles omnidireccionales es aplicable a esta plataforma, pues se utilizó el mismo controlador de la tesis de Arturo y la ley de campos potenciales. Esto comprueba también que la plataforma sí es controlable.

En cuanto a los errores se puede observar que son muy parecidos tanto en lazo abierto (LA) como en lazo cerrado (LC), aunque la tendencia a cero se notó levemente más en lazo cerrado que en abierto, debido a la implementación del sistema de visión para poder ubicar el sistema de referencia en el que se debía mover el robot en LC.

En cuanto al sistema para LA solo se utilizó para poder visualizar la trayectoria que seguía, ya que este se utiliza como sistema de referencia.

En el sistema de visión implementado para LC se notó la eficiencia que tiene al usar campos potenciales para ubicar el robot y que siga la trayectoria definida previamente y por supuesto la ayuda que recibe de la cinemática utilizada, aunque con esto último podemos notar que en LA también es eficiente aunque con un desfase pequeño debido a la referencia que el robot tiene para moverse.

Comparando los resultados obtenidos para la prueba de la recta en LA y la recta en LC, podemos comentar lo siguiente:

Respecto al controlador:

Para el desplazamiento en "x" el error absoluto en LA fue de 5 [cm], mientras que en LC fue de 1.8 [cm], lo cual representa una mejora de 3.2 cm menos de error.

Para el desplazamiento en "y" tenemos un error de 6.8 [cm] en LA, mientras que en LC el error es de 3 [cm], la mejora es de 3.8 [cm] menos de error. Esto quiere decir que el controlador cumple su función de manera adecuada ya que está reduciendo el error en prácticamente un -64% en "x", mientras que en "y" lo disminuye en un -44%.

El controlador es funcional y adecuado para evaluar la cinemática y al mismo tiempo puede usarse en otras pruebas donde pueda existir un brazo manipulador sobre la plataforma.

Respecto al robot

El diseño utilizado para el robot fue el más adecuado, ya que no hubo problema alguno al momento de tomar los datos y los videos para el reporte, ni para generar las figuras correspondientes en el espacio utilizado y el campo de visión de la cámara, además la colocación de las llantas fue la correcta, lo cual le permitió un movimiento uniforme en su desplazamiento, lo hizo de la manera como fue establecida por la cinemática.

6.2 Trabajo a futuro

El robot es un prototipo que permite estudiar la validez de la cinemática desarrollada para este tipo de llantas, existen aspectos que no se contemplaron a detalle en su diseño, ya que el principal propósito del estudio fue el desempeño de

este tipo de llantas de robots. En trabajos futuros se deben hacer mejoras tanto en la parte mecánica como en la parte de visión y demás elementos que lo conforman.

- **Mecánica.**

- **Configuración de las Llantas.** En primer lugar, se puede retomar el trabajo de Diegel Olaf [7] donde propone mejoras al diseño de las llantas Mecanum. Dentro de estas mejoras se propone utilizar la siguiente configuración de llantas, tal como se muestra en la figura 6.2b donde se hace una comparación de la disposición de las llantas.

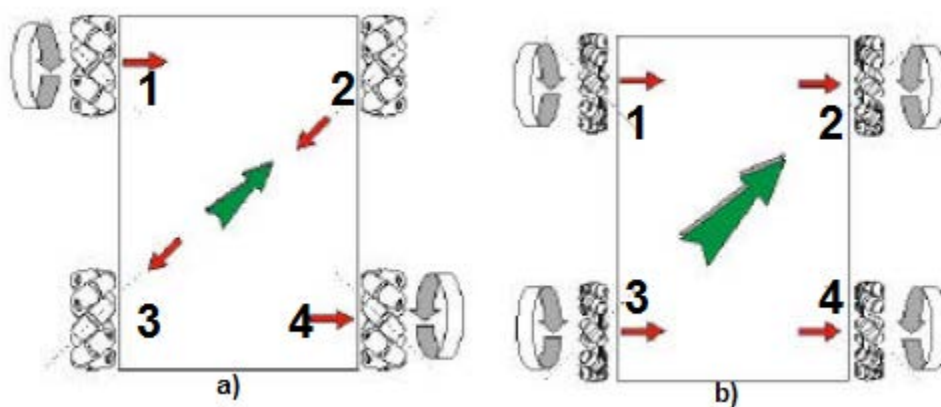


Fig. 6.2 a) Plataforma "AB" b) Plataforma con rodillos de llantas 2 y 3 a 45° [13].

En la figura 6.2b se observa una configuración muy diferente a la propuesta y estudiada en este trabajo, ya que los rodillos de la rueda trasera izquierda y la delantera derecha están colocados en un ángulo de 45°. Con esta configuración se lograría que las ruedas agregaran velocidad en un movimiento diagonal (flecha verde) ya que en nuestra configuración, las ruedas mencionadas no permiten este movimiento, lo que causa una resistencia u oposición al movimiento diagonal, haciendo que los motores de las ruedas 1 y 4 consuman más energía.

- **Disposición de los rodillos.** Las ruedas que utilizamos para el robot están basadas en el diseño original de Ilion, tal como se mencionó en el capítulo 2. Este tipo de ruedas presenta problemas al moverse en un área donde existen pendientes ya que los rodillos no hacen contacto con la superficie, tal como lo muestra la figura 6.2c. Esto

tiene como consecuencia una fuerza de fricción alta que provoca un esfuerzo mayor por parte del motor para poder desplazarla.

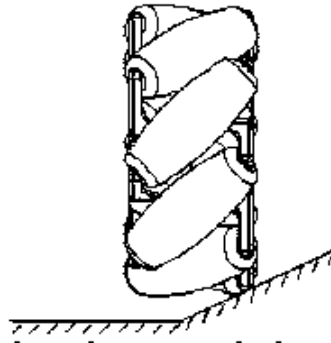


Fig. 6.2c Rueda Mecanum sobre una superficie inclinada [13].

Para solucionar este tipo de problemas se propone utilizar una rueda con los rodillos sujetos por la parte de en medio, con lo cual se resuelve dicho problema. Es importante mencionar que debe tener algún tipo de tope que no les permita, a los rodillos, salirse del eje de giro.

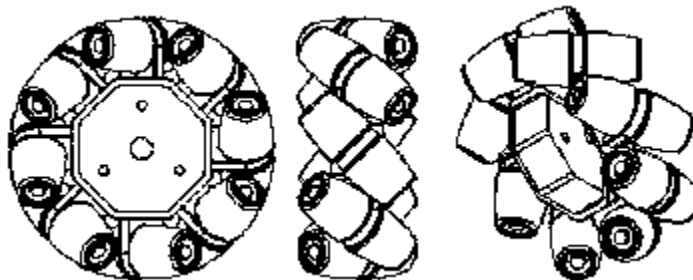


Fig. 6.2d Rueda Mecanum con rodillos montados en el centro [13].

- **Sistema de suspensión.** Siguiendo con la parte de la mecánica, el robot no posee ningún sistema de suspensión, es sabido que dichos sistemas permiten absorber las irregularidades del terreno por el cual circula el vehículo, lo cual aumenta la comodidad y control de este. Por lo tanto, creemos que es necesario agregar dicho sistema al robot, esto permitiría reducir aún más los errores observados en las pruebas del capítulo 6.

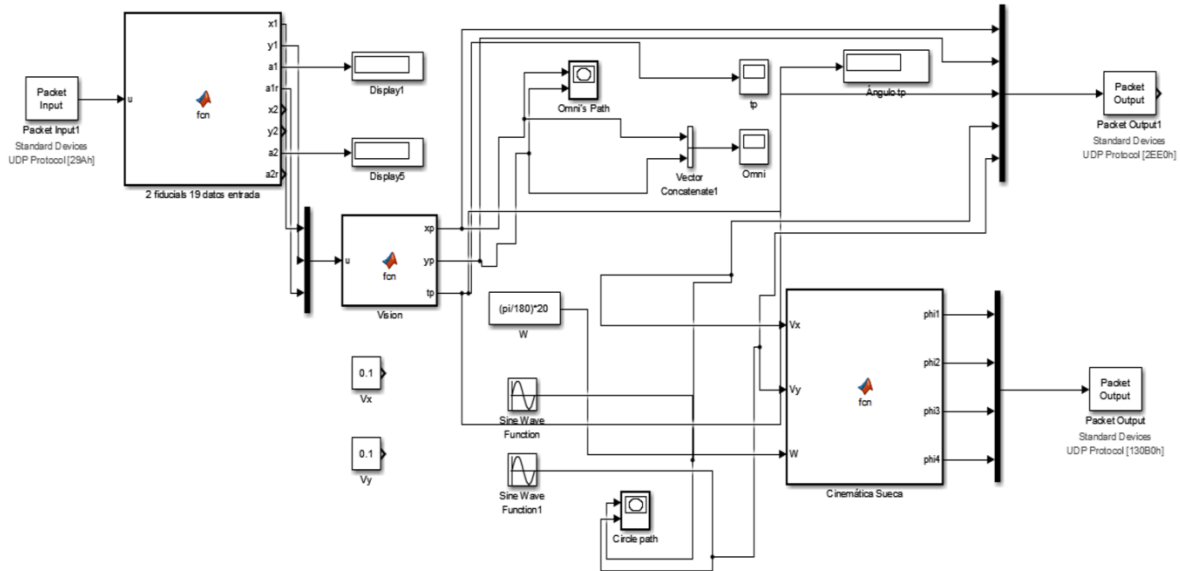
- **Coordinación Plataforma Móvil Omnidireccional con el Brazo Robótico.** Pedro Mariano [8] menciona en su tesis que realizó un experimento donde hay una coordinación entre la plataforma y el brazo que él usa, por cuestiones de tiempo no pudimos experimentar. Sería muy interesante realizar un experimento como el que él realizó, utilizando la operación pick & place; pero sin un pre-docking, ya que nuestro robot puede llegar desde cualquier posición y orientación a cualquier punto mientras no existan obstáculos en el camino. La sugerencia aquí, es que mientras el robot llega rápidamente al objetivo, el brazo se vaya acomodando para recoger el objeto.
- **Dinámica.** Ya corroborando en esta tesis que la cinemática que se usó es muy eficiente tanto para trabajos en lazo abierto como en lazo cerrado se sugiere que ahora sea estudiada y experimentada la dinámica que se emplea en un robot de este tipo, para que el error disminuya aún más dentro de un entorno más complejo como una casa o espacios más estrechos.
- **Visión.** En cuanto a visión se requiere una cámara que sea más sensible para captar los movimientos del móvil de una manera más precisa y así disminuir el error, ya que se lograrían mejores resultados al momento de realizar las pruebas. Incluso se puede pensar en cámaras conectadas en red para tener un área más grande de trabajo.
- **Sensores de distancia.** Una propuesta puede ser, usar sensores de distancia para poder medir qué tan lejos, o qué tan cerca estamos de algún objetivo y poder reducir, aún más, el error con ayuda de la cámara al momento del desplazamiento y evadir obstáculos que aparezcan de repente o ya se encuentren ahí.

Apéndices

Apéndice I. Diagramas de bloque y programación de las simulaciones para las pruebas en Lazo Abierto y Lazo Cerrado

Lazo Abierto

Diagrama de bloques



Función de Matlab para el cálculo de la cinemática

```
function [phi1,phi2, phi3, phi4]= fcn(Vx,Vy,W)
%datos del robot
r=0.025;
b=0.112;
d=0.078;

phi1t= (Vx/r) - (Vy/r) - (((b+d)/r)*W);
phi2t= (Vx/r) + (Vy/r) - (((b+d)/r)*W);
phi3t= (Vx/r) - (Vy/r) + (((b+d)/r)*W);
phi4t= (Vx/r) + (Vy/r) + (((b+d)/r)*W);
%normalización a datos de motores
phi1t = phi1t * 127 / 60;
phi2t = phi2t * 127 / 60;
phi3t = phi3t * 127 / 60;
phi4t = phi4t * 127 / 60;
%moviendo el cero
phi1t = phi1t + 128;
phi2t = phi2t + 128;
phi3t = phi3t + 128;
phi4t = phi4t + 128;
```

```
if phi1t>255
    phi1t = 255;
end
if phi1t<0
    phi1t = 0;
end

if phi2t>255
    phi2t = 255;
end
if phi2t<0
    phi2t = 0;
end

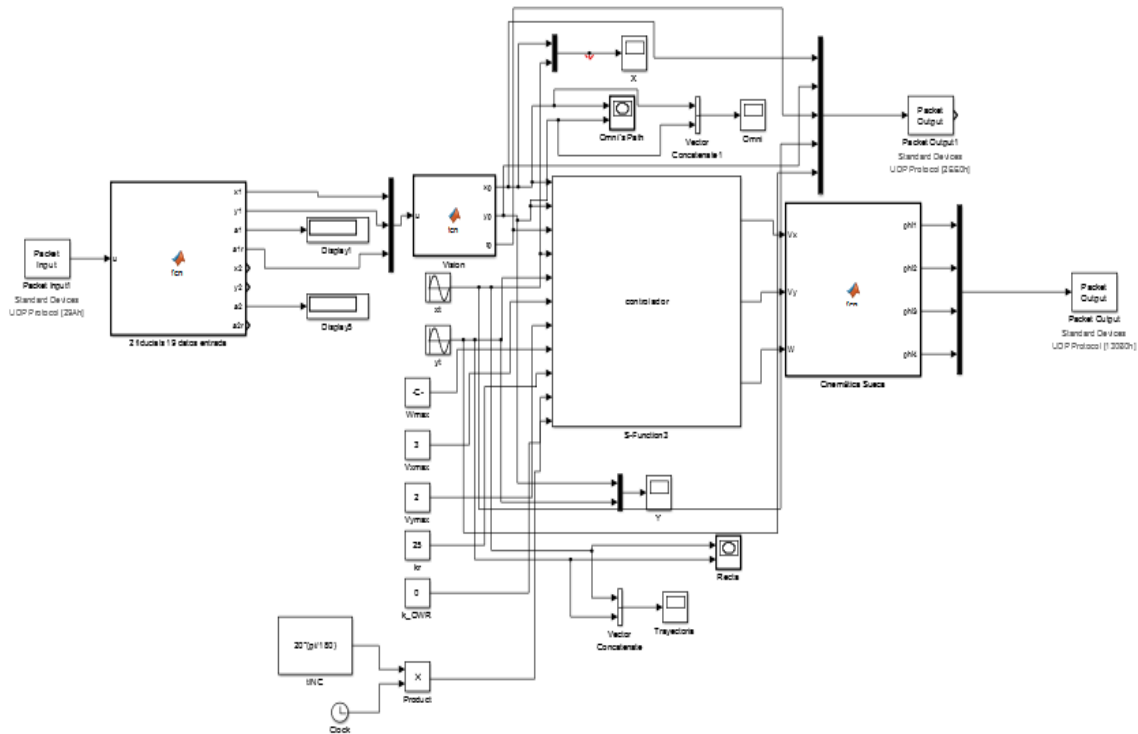
if phi3t>255
    phi3t = 255;
end
if phi3t<0
    phi3t = 0;
end

if phi4t>255
    phi4t = 255;
end
if phi4t<0
    phi4t = 0;
end

phi1 = phi1t;
phi2 = phi2t;
phi3 = phi3t;
phi4 = phi4t;
```


Lazo Cerrado. En este bloque solo se muestra el código para el controlador y reactivision, ya que el bloque de la cinemática es el mismo que el anterior.

Diagrama de bloques



Función S del controlador

```

/*
 * File :pruebaVel.c
 * Abstract:
 *     An example C-file S-function for multiplying an input by 2,
 *     y = 2*u
 *
 * Real-Time Workshop note:
 *     This file can be used as is (noninlined) with the Real-Time Workshop
 * C rapid prototyping targets, or it can be inlined using the Target
 * Language Compiler technology and used with any target. See
 * matlabroot/toolbox/simulink/blocks/tlc_c/timestwo.tlc
 * the TLC code to inline the S-function.
 *
 * Copyright 1990-2013 The MathWorks, Inc.
 */

```

```

#define S_FUNCTION_NAME controlador
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include "math.h"

```

```

/*=====
 * Build checking *
 *=====*/

/* Function: mdlInitializeSizes
=====
 * Abstract:
 *   Setup sizes of the various vectors.
 */
staticvoidmdlInitializeSizes(SimStruct *S)
{
ssSetNumSFcnParams(S, 0);
if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
return; /* Parameter mismatch will be reported by Simulink */
}

if (!ssSetNumInputPorts(S, 11)) return;
ssSetInputPortWidth(S, 0, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 1, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 2, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 3, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 4, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 5, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 6, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 7, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 8, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 9, DYNAMICALLY_SIZED);
ssSetInputPortWidth(S, 10, DYNAMICALLY_SIZED);

ssSetInputPortDirectFeedThrough(S, 0, 1);

if (!ssSetNumOutputPorts(S,3)) return;
ssSetOutputPortWidth(S, 0, DYNAMICALLY_SIZED);
ssSetOutputPortWidth(S, 1, DYNAMICALLY_SIZED);
ssSetOutputPortWidth(S, 2, DYNAMICALLY_SIZED);

ssSetNumSampleTimes(S, 1);

/* specify the sim state compliance to be same as a built-in block */
ssSetSimStateCompliance(S, USE_DEFAULT_SIM_STATE);

ssSetOptions(S,
              SS_OPTION_WORKS_WITH_CODE_REUSE |
              SS_OPTION_EXCEPTION_FREE_CODE |
              SS_OPTION_USE_TLC_WITH_ACCELERATOR);
}

/* Function: mdlInitializeSampleTimes
=====
 * Abstract:
 *   Specifiy that we inherit our sample time from the driving block.

```

```

*/
staticvoidmdlInitializeSampleTimes(SimStruct *S)
{
ssSetSampleTime(S, 0, INHERITED_SAMPLE_TIME);
ssSetOffsetTime(S, 0, 0.0);
ssSetModelReferenceSampleTimeDefaultInheritance(S);
}

/* Function: mdlOutputs
=====
* Abstract:
*   y = 2*u
*/
staticvoidmdlOutputs(SimStruct *S, int_Ttid)
{
floati,dg,drep,dres,tg,trep,te;
float      rho01,rhoc1,rhor1,n1;
float      x01,y01;
floatvxpp,vypp,wpp;
floatvxmax,vymax,wmax;
floatxp,yp,tP,tINC;
floata,b,c,d,r;
floatxd,yd,kr,k_OWR;
float      pi=3.141516;

//variables de entrada
//Posición y orientación del robot móvi diferencial
InputRealPtrsType  uPtrs0 = ssGetInputPortRealSignalPtrs(S,0);
InputRealPtrsType  uPtrs1 = ssGetInputPortRealSignalPtrs(S,1);
InputRealPtrsType  uPtrs2 = ssGetInputPortRealSignalPtrs(S,2);
//Posición deseada para el robot móvil diferencial
InputRealPtrsType  uPtrs3 = ssGetInputPortRealSignalPtrs(S,3);
InputRealPtrsType  uPtrs4 = ssGetInputPortRealSignalPtrs(S,4);
//Wmax
InputRealPtrsType  uPtrs5 = ssGetInputPortRealSignalPtrs(S,5);
// vxmax
InputRealPtrsType  uPtrs6 = ssGetInputPortRealSignalPtrs(S,6);
// vymax
InputRealPtrsType  uPtrs7 = ssGetInputPortRealSignalPtrs(S,7);
// Variables de control
//kr
InputRealPtrsType  uPtrs8 = ssGetInputPortRealSignalPtrs(S,8);
//k_OWR
InputRealPtrsType  uPtrs9 = ssGetInputPortRealSignalPtrs(S,9);
InputRealPtrsType  uPtrs10 = ssGetInputPortRealSignalPtrs(S,10);

//Variables de salida
real_T      *vxp      = ssGetOutputPortRealSignal(S,0);
real_T      *vyp      = ssGetOutputPortRealSignal(S,1);
real_T      *wp       = ssGetOutputPortRealSignal(S,2);

//Posición y orientación del robot móvil omnidireccional
xp  = *uPtrs0[0];
yp  = *uPtrs1[0];
tP  = *uPtrs2[0];
//Posición deseada

```

```

xd    = *uPtrs3[0];
yd    = *uPtrs4[0];
// Wmax deseada
wmax  = *uPtrs5[0];
//vxmax
vxmax = *uPtrs6[0];
// vymax
vymax = *uPtrs7[0];
// Variables de control
kr     = *uPtrs8[0];
k_OWR = *uPtrs9[0];
tINC   = *uPtrs10[0];

//Obstáculos
x01= 5000;
    y01= 5000;
    rho01=20;
    rhor1=4;
    n1=100;

//Cálculo de las velocidades de entrada
dg     = sqrt(pow((xd-xp),2) + pow((yd-yp),2));
    rho01 = sqrt(pow((x01-xp),2) + pow((y01-yp),2));

if (yd==yp&&xd==xp){
    tg = tP;
}
else {
    tg = atan2((yd-yp),(xd-xp));
}

if (rho01 > rho01+rhor1){
    drep=0;
    trep=2*tg;
}
else {
    drep= (n1/2)*pow(( (1/(rho01-rhor1)) - (1/rho01) ),2) ;
    trep= atan2((y01-yp),(x01-xp));
}

dres=drep+dg;
te = (trep-tg) - tP;

if (dres<= k_OWR){
    vxpp = 0;
    vypp = 0;
}
elseif (dres<= k_OWR + kr) {
    vxpp = vxmax*dres*cos(te)/(kr + k_OWR);
    vypp = vymax*dres*sin(te)/(kr + k_OWR);
}

```

```

else {
vxpp = vxmax*cos(te);
vypp = vymax*sin(te);
}
wpp=wmax*sin(-tP+tINC);

*vxp =vxpp;
*vyp =vypp;
*wp  =wpp;

}

/* Function: mdlTerminate
=====
* Abstract:
*   No termination needed, but we are required to have this routine.
*/
staticvoidmdlTerminate(SimStruct *S)
{
}

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file?
*/
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfuns.h"      /* Code generation registration function */
#endif

```

Función de Matlab para el bloque de reactivision

```
function [x1,y1,a1,a1r,x2,y2,a2,a2r]= fcn(u)

aux=0;
e=zeros(19);
for i=1:1:19
    y=u(i);
    if y==77
        aux=i;
    else

end

end

end
r=19-aux;
for j=1:1:r
    e(j)=u(aux+j);
end
for k=1:1:aux
    e(r+k)=u(k);
end
x1=(e(1)-48)*100+(e(2)-48)*10+(e(3)-48);
y1=(e(4)-48)*100+(e(5)-48)*10+(e(6)-48);
a1=(e(7)-48)*100+(e(8)-48)*10+(e(9)-48);
a1r=a1*3.1416/180;

x2=(e(10)-48)*100+(e(11)-48)*10+(e(12)-48);
y2=(e(13)-48)*100+(e(14)-48)*10+(e(15)-48);
a2=(e(16)-48)*100+(e(17)-48)*10+(e(18)-48);
a2r=a2*3.1416/180;
```

Función de Matlab para el bloque de acondicionamiento de los datos de visión

```
function [xp,yp,tp]= fcn(u)

x    = u(1);
y    = u(2);
th   = u(3);

%Coordenadas del robot y del objetivo
xs   = (x/100)*(220)-110;
ys   = (y/100)*(-170)+85;

xp = xs+10;
yp = ys+10;
tp = th;
```

Programación de código en Matlab para la simulación del robot

```
clc
clear all
close all

filename = 'PruebasFinales.xlsx'; %%%Leer archivo de Excel
sheet = 2; %%% Leer hoja donde se almacenaron los datos
R = xlsread(filename,sheet); %%%Crear una matriz con los datos leídos

t = R(2:4200,1);
xp = R(2:4200,2);
yp = R(2:4200,3);
tP = R(2:4200,4);
xt = R(2:4200,5);
yt = R(2:4200,6);

%%%% Funciones paramétricas de las Curvas de Lissajous %%%
A = 0;
B = 0;
wx = 0*(pi/180);
phaseX = 0*(pi/180);
wy = 0*(pi/180);
phaseY = 0*(pi/180);
thetaP = 0;
xtt = A*sin(wx.*t+phaseX);
ytt = B*sin(wy.*t+phaseY);
tPt= thetaP.*t;
tt = t;

mmo = [-4 -4 -9 -9 -11 -11 -9 -9 -4 -4 4 4 9 9 11 11 6 6 11 11 11 9 9 4 4
-4;...
-9 -11 -11 -9 -9 9 9 11 11 9 9 11 11 9 9 4 4 -4 -4 4 -9 -9 -11 -11
-9 -9]; %Plataforma Móvil
P = [-0.5 -0.5 0.5 0.5 -0.5 ; -0.5 0.5 0.5 -0.5 -0.5];%Punto P
Pm = [-11 -11 11 11 -11;...
-9 9 9 -9 -9]; %Base

Hold on

%%%% Graficar puntos %%%

for t=0:499:4200
%%Matriz de rotación%%
Rot = [cos(tPt(t+1)) -sin(tPt(t+1));...
sin(tPt(t+1)) cos(tPt(t+1))];

%%Operación de rotación
r = Rot*mmo; %Plataforma
```

```

rp = Rot*P; %Punto P
rpm = Rot*Pm; %Base

%%Traslación de los puntos de Plataforma y punto P

px = 40*cos(-wx.*tt(t+1));
py = 40*sin(-wy.*tt(t+1));

for i=1
for j=1:1:26
r(i,j) = r(i,j) + px;
end
end

for i=2
for j=1:1:26
r(i,j) = r(i,j) + py;
end
end
%Dibujar Plataforma rotada y trasladada
plot(r(1,:),r(2,:), 'LineStyle', '-', 'LineWidth', 2, 'Color', 'k')

%%%%%% Dibujar Punto P y Base rotados y trasladados %%%%%%%%%%%
for i=1
for j=1:1:5

rp(i,j) = rp(i,j) + px;
rpm(i,j) = rpm(i,j) + px;
end
end

for i=2
for j=1:1:5

rp(i,j) = rp(i,j) + py;
rpm(i,j) = rpm(i,j) + py;
end
end
%%%%%Dibujarpunto P%%%%%%%%%
plot(rp(1,:),rp(2,:), 'color', 'r', 'LineWidth', 4)
%%%%% Dibujar Base %%%%%%%%%
plot(rpm(1,:),rpm(2,:), 'LineStyle', '-', 'LineWidth', 2, 'Color', 'b')
gridon

end

%%%%% Trayectoria a seguir %%%%%
holdon
plot(xtt,ytt, 'LineStyle', '-.', 'LineWidth', 2, 'Color', 'm')
xlabel('X [cm]')
ylabel('Y [cm]')

```


Apéndice II. Programación de Arduino

```
#include <Wire.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#define i2cAddress 0x58
#define i2cAddress2 0x59
#define speed1 0x00           // speed to first motor
#define speed2 0x01
// the IP address for the shield:
IPAddress ip(192, 168, 0, 110);
*/
char ssid[] = "RobotMovil"; // your network SSID (name)
char pass[] = "holamundo";  // your network password (use for WPA, or
use as key for WEP)

int keyIndex = 0;           // your network key Index number (needed only
for WEP)
int status = WL_IDLE_STATUS;
inta,b,c,d;
unsigned int localPort = 35000; // local port to listen on

char packetBuffer[255]; //buffer to hold incoming packet
WiFiUDP udp;
void setup() {
Wire.begin();
//Initialize serial and wait for port to open:
//Serial.begin(9600);

// check for the presence of the shield:
if (WiFi.status() == WL_NO_SHIELD) {
//Serial.println("WiFi shield not present");
// don't continue:
while(true);
}
// WiFi.config(ip); // ipforzada
// attempt to connect to Wifi network:
while ( status != WL_CONNECTED) {
//Serial.print("Attempting to connect to SSID: ");
//Serial.println(ssid);
// Connect to WPA/WPA2 network. Change this line if using open or WEP
network:
status = WiFi.begin(ssid, pass);
// wait 10 seconds for connection:
delay(10000);
}
// start the server:
// you're connected now, so print out the status:
//printWifiStatus();
Udp.begin(localPort);
}

void loop() {
//Serial.print("Entre al loop ");
// if there's data available, read a packet
int packetSize = Udp.parsePacket();
if(packetSize)
```

```

    {
intlen = Udp.read(packetBuffer,255);
if (len>0) packetBuffer[len]=0;
    a=packetBuffer[0];
    b=packetBuffer[1];
    c=packetBuffer[2];
    d=packetBuffer[3];

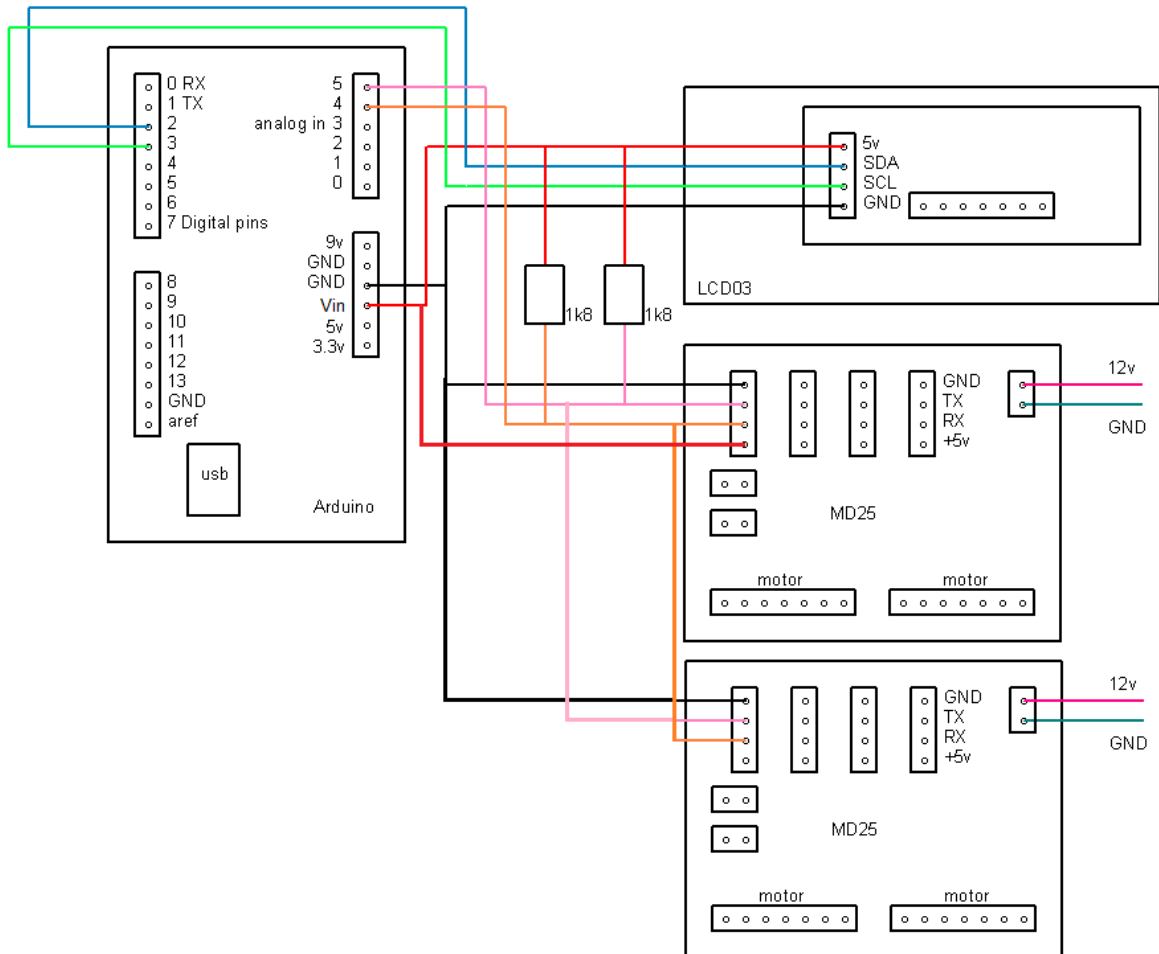
digitalWrite(13,HIGH);
Wire.beginTransmission(i2cAddress); // Drive motor 1
at speed value stored in x
Wire.write(speed1);
Wire.write(b);
Wire.endTransmission();
    //Serial.println(a);
Wire.beginTransmission(i2cAddress); // Drive motor 2
at speed value stored in x2
Wire.write(speed2);
Wire.write(c);
Wire.endTransmission();
    //Serial.println(b);
Wire.beginTransmission(i2cAddress2); // Drive motor 1
at speed value stored in x
Wire.write(speed1);
Wire.write(a);
Wire.endTransmission();
    //Serial.println(a);
Wire.beginTransmission(i2cAddress2); // Drive motor 2
at speed value stored in x2
Wire.write(speed2);
Wire.write(d);
Wire.endTransmission();
    //Serial.println(b);

digitalWrite(13,LOW);
    }
}

void printWifiStatus() {
    // print the SSID of the network you're attached to:
    // Serial.print("SSID: ");
    //Serial.println(WiFi.SSID());
    // print your WiFi shield's IP address:
IPAddress ip = WiFi.localIP();
    //Serial.print("IP Address: ");
    //Serial.println(ip);
    // print the received signal strength:
long rssi = WiFi.RSSI();
    //Serial.print("signal strength (RSSI):");
    //Serial.print(rssi);
    //Serial.println(" dBm");
}

```

Apéndice III. Conexión de las tarjetas MD25 y Arduino



Nota: La pantalla LCD es opcional. Si se desea no ponerla, solo hay que desconectarla. El módulo de Arduino Wifi se coloca encima del Arduino.

Referencias

- [1] Wit, C. C. d., B. Siliciano, et al. (1997). Theory of Robot Control. B. W. Dickinson. Great Britain, Springer: 392.
- [2] Baturone, A. O. (2001). Robótica manipuladores y robots móviles. Barcelona, España, Marcombo.
- [3] Suárez, A. E. A. and A. M. B. Sánchez (2015). Plataforma Móvil Omnidireccional de cuatro Llantas Sucas en Configuración "AB". Mecatrónica. México, Universidad Nacional Autónoma de México: 98 págs.
- [4] Campion, G. (1996). Structural Properties and Classification of Kinematic and Dynamic models of Wheeled Mobile robots. IEEE Transactions on Robotics and Automation. Vol. 12: 47-60.
- [5] Calandín, L. I. G. Modelado Cinemático y Control de robots Móviles con Ruedas. Ingeniería de sistemas y Automática. España, Universidad Politécnica de Valencia. Doctorado: 300 páginas.
- [6] Martínez, R. S. S. (2009). Diseño de un Robot móvil Omnidireccional. Montevideo, Uruguay, Universidad de la República.
- [7] Walsh, G., D. Tilbury, et al. (1994). "Stabilization of Trajectories for Systems with Non-holonomic Constrains." IEEE Transactions of Automatic Control 34: 216-226.
- [8] Gúzman, L. E. S. (2014). Seguimiento de Trayectorias con un Robot Móvil de Configuración Diferencial. Ing. USBMed. Bogotá, Colombia. Vol. 5: 9 páginas.
- [9] Zhang, Y., W. Li, et al. (2014) Analysis and Verification of Repetitive Motion Planning and Feedback Control for Omnidirectional Mobile Manipulator Robotic Systems.
- [10] Rojas, R. "A short history of omnidirectional wheels."
- [11] Patrick F. Muir, C. P. N. (1987). "Kinematic Modeling for Feedback Control of an Omnidirectional Wheeled Mobile Robot." IEEE: 1772-1778.

[12] V. F. Muñoz Martínez, "Modelado cinemático y dinámico de un robot móvil omnidireccional", Instituto Andaluz de Automática Avanzada y Robótica.

[13] Diegel, O., et al. (2002). Improved Mecanum Wheel Design for Omnidirectional Robots. Australasian Conference on Robotics and Automation. Auckland, ARAA: 117-121.

[14] Pro, V. P. R. (2015). "Ruedas mecanum." Tomado de <http://www.vexrobotics.com.mx/vexpro/mecanum-ruedas.html>.

[15] Kuka (2015). "OmniMove." 2015, from <http://www.kuka-robotics.com/en/>.

[16] Robotics, F. (2015). "FingerTech Mecanum Wheels." 2015, Tomado de <http://www.fingertechrobotics.com/proddetail.php?prod=ft-mecanum-set-v2>.

[17] Robot-Electronics (2015). "EMG30, mounting bracket and wheel specification." 2015, Tomado de <http://www.robot-electronics.co.uk/htm/emg30.htm>.

[18] Bass, L. (2013). "Mecanum Wheel." 2015, tomado de <https://grabcad.com/library/mecanum-wheel-3>.

[19] Zippy (2015). "Bateria recargable de litio." 2015, Tomado de <http://www.tucargadorsolar.com/Baterias-Recargables/Bateria-Recargable-de-litio-12v-6800mAh.html#>.

[20] Arduino (2015). "Arduino WiFi Shield." 2015, Tomado de <http://www.arduino.cc/en/Main/ArduinoWiFiShield>.

[21] Arduino (2015). "Arduino Uno." 2015, Tomado de <http://www.arduino.cc/en/Main/ArduinoBoardUno>.

[22] Robot-Electronics (2015). "Drive Systems." 2015, Tomado de http://www.robot-electronics.co.uk/acatalog/Drive_Systems.html.

[23] Silvestre, E. R. R., R. Morales-Guerrero, et al. (2010). "Modelado, simulación y construcción de un robot móvil de ruedas tipo diferencial." Lat. Am. J. Phys. Educ. Vol. 4(Mecatrónica).

[24] Viboonchaicheep, Pakpoom, Et. Al., (2003), "Position Rectification Control for Mecanum Wheeled Omnidirectional Vehicles", Japón.

- [25] Cooney, J.A., Et. Al., (2004), "Visual dead-reckoning for motion control of a Mecanum-wheeled mobile robot", *Mechatronics* 14, 623–637, Nueva Zelanda.
- [26] Shimada, Akira, Et. Al., (2005), "Mecanum-wheel Vehicle Systems Based on Position Corrective Control", Japón.
- [27] Doroftei, Ioan, Et. Al., (2007), "Omnidirectional Mobile Robot – Design and Implementation", fuente: *Bioinspiration and Robotics: Walking and Climbing Robots*, Rumania, 511-528.
- [28] Han, Kyung-Lyong, Et. Al., (2010), "The Sources of Position Errors of Omnidirectional Mobile Robot with Mecanum Wheel", Korea.
- [29] Batista, Víctor Costella, Et. Al., (2013), "Sistema de Control con Lógica Difusa y Diseño de Plataforma Autónoma con Ruedas Mecanum a Partir de su Arquitectura", fuente: *RIDTEC* 9(2), Panamá, 30-38.
- [30] Craig, J. J. *Introduction to Robotics. Mechanics and Control*. M. J. Horton. United States of America, Pearson Prentice Hall: 399.
- [31] Lih, L. C. and S. Y. Hao (2013). "Modeling and Adaptive Control of an Omni-Mecanum-Wheeled Robot." *Intelligent Control and Automation*: 166-179.
- [32] Angeles, J. *Fundamentals of Robotics Mechanical Systems. Mechanical Engineering Series*. F. F. Ling, Springer: 460.
- [33] Villela, V. J. G. and R. M. Parkin (2005). Evadiendo obstáculos con robots móviles. *Revista Digital Universitaria*. México, DGSCA-UNAM. Vol. 6: 2-9.
- [34] Tanenbaum, A. S. and D. J. Wetherall (2012). *Redes de Computadoras*. L. M. C. Castillo. México, Pearson Educación: 816.
- [35] Martínez, A. C. (2012). Plataforma Móvil Omnidireccional a partir de dos Robots Móviles Diferenciales. *Mecatrónica*. México, Universidad Nacional Autónoma de México: 138 págs.
- [36] Escobedo, P. M. C. (2012). Manipulador Móvil: Estudio sobre la coordinación de movimientos de un manipulador serial acoplado. *Mecatrónica*. México, Universidad Nacional Autónoma de México: 151 págs.

[37] González, V. J. V. (2006). Research on a semiautonomous mobile robot for loosely structured environments focused on transporting mail trolleys. Mechanical and Manufacturing Engineering. United Kingdom, Loughborough University. Doctor of Philosophy: 337 págs.

[38] Weisstein, E. W. (2015). "Lissajous Curve." 2015, Tomado de <http://mathworld.wolfram.com/LissajousCurve.html>.