



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Caracterización y control de una
plataforma omnidireccional**

TESIS

Para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

Jean Carlo Andrade Ramírez

DIRECTOR DE TESIS

M. en A. Adalberto Joel Durán Ortega



Ciudad Universitaria, CDMX, 2017



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Agradezco a la **Universidad Nacional Autónoma de México** por ser mi segundo hogar y brindarme todas las facilidades para mi formación profesional.

A la **Facultad de Ingeniería**.

Al **Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS)**.

A la **DGAPA, UNAM**, en lo que respecta al apoyo proporcionado a través del Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT), utilizado para la realización de esta tesis.

Al **Dr. Pedro Acevedo Contla** por su gran apoyo en la realización de esta tesis.

Y un agradecimiento especial a mi director **M. en A. Adalberto Joel Durán Ortega** quien con su conocimiento y constante asesoramiento fue parte esencial durante cada etapa de la realización de esta tesis, sin su apoyo este proyecto no habría podido ser posible.

Dedicatoria

Es con mucho orgullo y afecto que dedico este trabajo escrito a mis padres Georgina Ramírez Godoy y José Fidel Andrade Herrera quienes a través de su apoyo, su confianza y su guía durante toda mi vida he podido concluir esta pequeña etapa de todas las experiencias que me faltan por vivir. Agradezco de sobremanera todas las lecciones de vida, consejos y buenos deseos que me entregaron durante todo mi desarrollo como persona demostrando que no importa lo duro que sea el camino, el destino es lo que debe motivarnos para seguir adelante y jamás rendirse aunque las cosas no sean fáciles.

A mi hermano José Ricardo Andrade Ramírez quien es un ejemplo vivo de coraje, de responsabilidad y de amor por su familia. A través de todos los años que pasamos juntos aprendí a ver el mundo con diferentes lentes, a cuestionarme lo aprendido, a buscar el porqué de algo y la lección más importante que un hermano puede enseñarle a otro: dos personas unidas al nacer jamás estarán separadas sin importar la distancia.

A mi abuela María de los Dolores Godoy Navarro y a mi tía Patricia Ramírez Godoy, que aunque no llegaron acompañarme en este gran momento de mi vida, sé que nunca dudaron de mí y que su amor fue y sigue siendo motivo para continuar luchando por mis sueños, siendo testigo mi abuelo Fernando Ramírez López quien siempre ha estado al pendiente de mi formación académica y personal en conjunto de mi tía María Elena Ramírez Godoy quien siempre ha visto por su familia y jamás la ha abandonado. .

A mis abuelos Carlota Amalia del Carmen Herrera Éboli y José Fidel Andrade Quintero quienes fueron pilares importantes durante mi niñez y juventud además de seguir siendo una parte importante en mi vida.

A mi familia en general quienes han estado a mi lado desde pequeño cuando más lo he necesitado, quienes jamás dudaron en brindarme su apoyo y aquellos que con sólo palabras de aliento consiguieron avivar la llama de mi espíritu para superar todo obstáculo que se me presentó durante el camino. Son todos ellos los pilares de mi carácter, de mi esencia y de mi alma.

A José Alberto Morales Calva quien estuvo y ha estado a mi lado desde el principio del camino para convertirme en ingeniero. Me ha visto llorar y me ha secado las lágrimas, me ha visto reír y ha compartido la alegría, me ha visto luchar y me ha dado las armas que necesitaba, me ha visto tambalear y me ha ofrecido su hombro como apoyo. Agradezco el día que nuestras vidas se cruzaron y espero que jamás se separen.

A Berenice Caballero Valverde, Haxel Lázgare Rendón y María Elena Vaillard Ceballos quienes han sido mis más grandes amigos durante toda la carrera y en quienes he podido encontrar siempre palabras de aliento. Son ellos los que consiguieron con sus bromas y comentarios alegrar mis días aún en los peores momentos y me han demostrado la existencia de una amistad sincera. Sé que siempre podré contar con ellos sin importar la situación en la que me encuentre.

A la Universidad Nacional Autónoma de México y a la gran Facultad de ingeniería que no sólo han sido mi segundo hogar, han sido el lugar que me demostró que los sueños se hacen realidad, que las buenas personas existen en verdad y que no necesitas viajar tan lejos para gozar de una educación de calidad. Por mucho considero que la preparación que obtuve dentro de esta institución supera a la que pude aspirar en otras universidades y que de no haber conseguido entrar a la UNAM en mi primer intento, hubiera insistido hasta el cansancio por ser parte de la máxima casa de estudios del país y llegar hasta donde estoy ahora.

Y finalmente agradezco a todas aquellas personas que han entrado a mi vida aunque fuese por un pequeño instante, pero que sin su aparición mi destino pudo haber sido muy distinto.

Hay un mundo allá fuera desconocido, enorme y maravilloso, no podré conocerlo completamente. Sin embargo, nunca dejaré de recorrerlo.

Jean Carlo Andrade Ramírez

Tabla de contenidos

Capítulo uno	7
Generalidades	7
Objetivo general.....	7
Objetivos particulares	7
Hipótesis	7
Justificación y alcance	8
Antecedentes del proyecto	9
Capítulo dos	12
Introducción	12
Robots móviles omnidireccionales	12
Ruedas más comunes en robots móviles y sus características	13
Aplicaciones.....	16
Capítulo tres	18
Caracterización del motor e implementación en el sistema	18
Selección del microcontrolador	19
Caracterizando el motor a rotor libre	21
Primera ensayo.	21
Segunda ensayo.	22
Tercer ensayo.....	23
Puesta en marcha de la plataforma omnidireccional.....	24
Capítulo cuatro	29
Diseño del control	29
Métodos de sintonización.	29
Respuestas obtenidas de los motores	32
Control de los motores	36
Diseño del control.....	37
Capítulo cinco	40

Diseño del circuito impreso e implementación de control en Arduino	40
Código del control en Arduino	42
Capítulo seis	45
Ajuste de los parámetros del control	45
Ajuste para el movimiento hacia adelante y hacia atrás	45
Análisis de velocidades Tangenciales.....	47
Análisis de desviaciones	50
Adelante.....	50
Atrás	50
Derecha.....	51
Izquierda.....	51
Giro hacia la izquierda.....	51
Giro hacia la derecha	52
Diagonal superior derecha.....	52
Diagonal superior izquierda.....	52
Diagonal inferior derecha	53
Diagonal inferior izquierda.....	53
Ajuste final de la constante K_p	53
Capítulo siete.....	56
Resultados	56
Conclusiones.....	63
Bibliografía	65
Índice de imágenes	68
Índice de tablas.....	70
Anexo uno: Código Arduino.....	71
Declaración de variables.....	71
Funciones.....	75

Capítulo uno

Generalidades

Objetivo general

Dotar con un desplazamiento suave y uniforme a la plataforma omnidireccional conformada por cuatro ruedas de tipo Mecanum dispuestas en una configuración AB, caracterizando su funcionamiento e implementando un sistema de control en lazo cerrado que permita regular la velocidad angular de cada rueda.

Objetivos particulares

- 1.- Poner en funcionamiento los sensores ópticos de cada rueda para que sirvan como referencia de entrada para el controlador.
- 2.- Diseñar el control del desplazamiento con base en un método de sintonización PID.
- 3.- Implementar el control diseñado para el desplazamiento de la plataforma verificando la eficiencia del mismo.
- 4.- Asegurar el desplazamiento moderado y sin cambios bruscos de dirección por la plataforma en al menos dos velocidades distintas.

Hipótesis

El desplazamiento uniforme de la plataforma dependerá directamente de la diferencia que exista entre la velocidad angular de cada una de sus ruedas.

Justificación y alcance

La robótica es un campo en continuo crecimiento y con expectativas de incrementarse en el futuro próximo. Se han realizado anuncios por parte de Amazon y DHL en los que se plantea el envío de paquetes usando drones. Por otra parte, se están realizando concesiones de licencia para coches que no requieren de un conductor para desplazarse a cualquier sitio dentro de las ciudades, siendo prueba de que cada día la robótica autónoma está y estará más presente en nuestras vidas [1].

Dentro de las ramas de la robótica existe una en particular de gran interés actualmente y es la robótica móvil, considerada como un área de la tecnología avanzada. Los productos desarrollados con base en esta área tienen bastas aplicaciones en control, programación, inteligencia artificial, percepción e instrumentación y son una buena forma para dar soluciones tecnológicas innovadoras orientadas al desarrollo de mejores robots e incrementar la gama de posibilidades que existen actualmente para su uso. Así, este campo de la investigación está desarrollándose en todo momento, quedando aún mucho que recorrer [2].

En este trabajo se aborda el caso de un robot móvil omnidireccional cuya importancia de estudio se basa en la facilidad que tienen para cambiar de dirección y sentido para evadir obstáculos, mayor movilidad en espacios reducidos lo que incrementa su uso como vehículos de carga y de un desplazamiento que es independiente a su orientación. Por otro lado, el estudio de las plataformas con configuración mecánica omnidireccional es de gran relevancia para la teoría del control debido al diseño e implementación de los algoritmos que su comportamiento requiere.

El alcance de la presente investigación es la elaboración de un algoritmo de control que permite el desplazamiento uniforme del vehículo en todas las direcciones posibles dentro de la gama comprendida por este tipo de robots móviles, tomando como referencia su funcionamiento y los elementos que lo conforman. El diseño del controlador se basará en el método de la curva de reacción de Ziegler-Nichols, el cual es un método gráfico de sintonización que se aplica directamente a las curvas reales del comportamiento de los motores al ser sometidos a una prueba escalón y cuyos resultados ofrecen un control más robusto aplicando una metodología no compleja.

Antecedentes del proyecto

Como previa investigación al trabajo se realizaron una serie de pruebas y recolección de datos en una plataforma de pequeña escala manufacturada en el laboratorio. La plataforma se diseñó con cuatro ruedas de tipo Mecanum manufacturadas mediante el uso de la técnica de impresión 3D. Cada rueda fue colocada sobre un eje y este eje fue puesto en un rodamiento empotrado a un soporte fijo sobre una placa de acrílico el cual tiene acoplado un motor de corriente directa con un sistema de engranes que aumentan el par de salida del motor. Adicionalmente se le colocaron imanes a cada eje de las ruedas y un sensor magnético. Todo el conjunto de elementos se muestra en la figura 1 y fue empleado como un pequeño banco de pruebas donde los resultados obtenidos fueron usados como una base teórica para ser aplicada en el trabajo actual. Dentro de dichos conocimientos adquiridos se incluye: el análisis y entendimiento de la mecánica y su interacción con los sensores.



Figura. 1. Banco de pruebas.

Los primeros ensayos realizados en el banco de pruebas consistieron en determinar la combinación de los giros a realizar para que la plataforma se dirigiese a una u otra dirección teniendo en cuenta la configuración de las ruedas que más adelante se abordará con más detalle. Para este tipo de ensayos se requirió de un circuito integrado conocido como puente H que se encarga de manipular el giro de un motor de corriente directa como los utilizados en este sistema a través de 4 interruptores internos basados en transistores dependiendo de la combinación binaria de entrada que este reciba. La combinación binaria fue programada en un microcontrolador de la familia Arduino (Atmega328) y sirvió para realizar todas las combinaciones posibles que la gama de movimientos de este tipo de robots móviles contempla.

Fue entonces que mediante esta serie de ensayos se comprobó que la dirección del desplazamiento estará dada por las diferentes combinaciones de giro proporcionadas por las ruedas, siendo estas combinaciones las que deben llevar un control. Los resultados obtenidos se encuentran en la tabla 1.

Dirección	Motor A	Motor B	Motor C	Motor D
Adelante	↓	↓	↓	↓
Atrás	↑	↑	↑	↑
Derecha	↑	↓	↑	↓
Izquierda	↓	↑	↓	↑
Diagonal superior derecha	↑	Apagado	↑	Apagado
Diagonal superior izquierda	Apagado	↑	Apagado	↑
Diagonal inferior derecha	Apagado	↓	Apagado	↓
Diagonal inferior izquierda	↓	Apagado	↓	Apagado
Giro horario	↓	↑	↑	↓
Giro antihorario	↑	↓	↓	↑

Tabla 1. Tabla de combinaciones de giro.

Con la realización de otro tipo de ensayos donde se utilizaron dos sensores magnéticos y cuatro imanes integrados en los ejes de las ruedas colocados a noventa grados de distancia uno con respecto al otro en un arreglo de zigzag se determinó el sentido de giro de las ruedas. La configuración de los imanes con respecto a los sensores determina digitalmente la dirección de giro que cada rueda está llevando a cabo para conseguirlo los sensores emiten una señal de salida entre cero y cinco volts dependiendo si están o no en contacto con el imán colocado en el eje. Posteriormente, esta señal de salida se pasó a través de un transistor que garantizaba una señal en cuadratura que será interpretada digitalmente como un 1 o un 0 dependiendo del estado del sensor. La figura 2 muestra la señal desfasada entregada por cada sensor ejemplificando la lógica que el desfase utiliza para determinar el sentido de giro de la rueda.

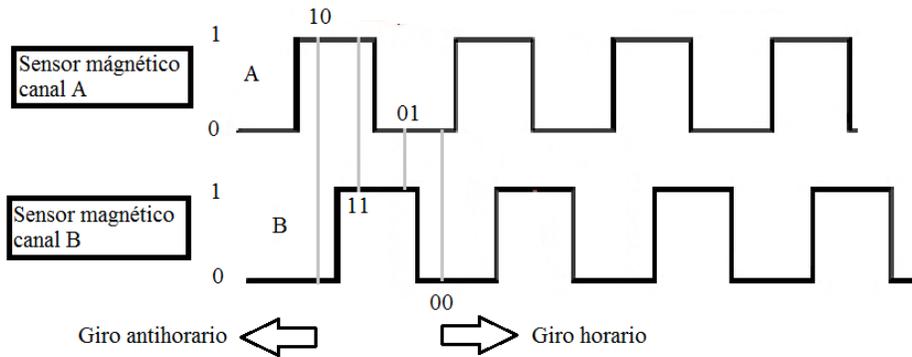


Figura. 2. Desfase de señales.

La lógica consiste en medir cuál de las señales provenientes de los dos sensores es recibida en primer lugar y medir una secuencia determinada de valores binarios que de cumplirse indicarán en qué dirección gira la rueda y al medir la frecuencia de estos pulsos se puede obtener la velocidad angular la cuál posteriormente se puede transformar en velocidad lineal. Esto servirá para controlar no sólo el sentido de giro de las ruedas también permitirá controlar su velocidad.

Secuencia sentido horario		Secuencia sentido anti horario	
Sensor A	Sensor B	Sensor A	Sensor B
1	0	0	0
1	1	0	1
0	1	1	1
0	0	1	0

Tabla 2. Sentido de giro.

De acuerdo con la tabla anterior depende del orden en que los valores sean recibidos como entrada en el microcontrolador para su interpretación como la dirección en la cual gira cada una de las ruedas de la plataforma.

Capítulo dos

Introducción

Los robots móviles son aquellos que están provistos de un sistema de automoción, tomando como referencia los que usan ruedas u orugas (véase figura 3), que les permiten desplazarse de un sitio a otro. Este tipo de robots normalmente reciben información a través de sensores integrados en el mismo y son empleados sobre todo para el transporte de mercancías en cadenas de producción y/o almacenes. También se utilizan estos robots en lugares de difícil acceso, áreas peligrosas para los seres humanos como es el caso de la inspección de centrales nucleares o sitios muy distantes como la exploración en otros planetas.



a)

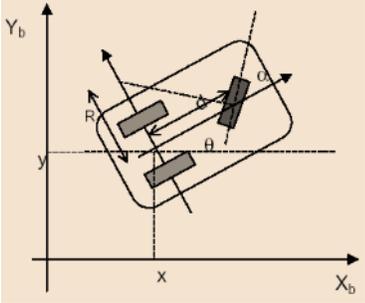
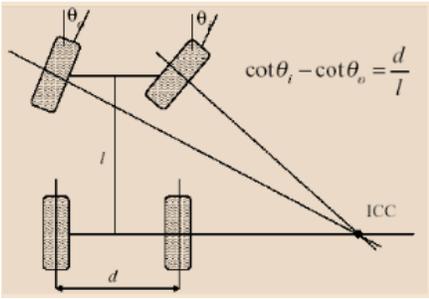
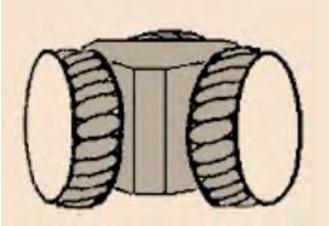
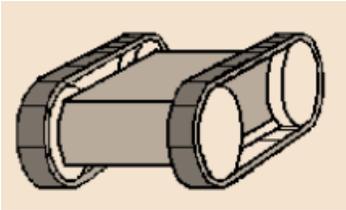


b)

Figura. 3. a) Ruedas: usadas en superficies lisas. b) Orugas: empleadas sobre superficies irregulares.

Robots móviles omnidireccionales

Los robots de desplazamiento omnidireccional son un tipo particular de robots móviles los cuales se pueden definir como aquellos que cuentan con movilidad en cualquier dirección sin tener que rotar previo al comienzo del desplazamiento, entendiéndose, que este incluye a todas las combinaciones de traslaciones como las laterales y las rotaciones. Esta condición se debe a que la disposición de las ruedas brinda al robot tres grados de libertad en el plano [3]. Este tipo de robots brindan mayores ventajas frente a otros cuyo desplazamiento se realiza de igual forma sobre un mismo plano, pero cuyo sistema de automoción se basa en ruedas convencionales o en orugas. En la tabla 3 se realiza una comparación entre los tipos de robots móviles más comunes donde se muestran las ventajas y desventajas de cada uno.

Tipo	Diagrama	Ventajas	Desventajas
Triciclo		-No hay deslizamiento.	-Se requiere guiado no holónomo*.
Ackerman		-Fácil de implementar. -Un sistema simple de 4 barras controla la dirección.	-Restricciones no holónomas*.
Ruedas omnidireccionales		-Permiten movimientos complicados (reducen restricciones cinemáticas).	-El movimiento en línea recta no está garantizado por lo que se requiere control. -Implementación ligeramente complicada
Bandas		-Sistema simple de controlar.	-Resultados pobres en odometría. -No se dispone de modelo preciso de giro. -consume mucha potencia al girar.

***Holónomo:** las restricciones en las velocidades son integrables, es decir es posible expresarlas exclusivamente en términos de las posiciones.

Tabla 3. Ventajas y desventajas de robots móviles comunes. Basado en [4].

Ruedas más comunes en robots móviles y sus características

En el apartado anterior se describió y realizó una comparativa entre los diferentes tipos de robots móviles que basan su desplazamiento en llantas y bandas. En este apartado se abordará la

descripción, características y limitaciones de las ruedas más comunes comenzando con las ruedas convencionales simples y terminando con las ruedas de tipo Mecanum.

Las ruedas convencionales simples solamente permiten el trasladarse en la dirección de la velocidad de la rueda (a), por lo que el eje de rotación no puede moverse salvo en la dirección del desplazamiento. Para resolver este problema, se introduce un corrimiento del eje de giro respecto de la rueda, creando dos tipos de ruedas nuevas: la rueda convencional de desplazamiento lateral (b) y la rueda convencional con desviación delantera (c) [5]. En la figura 4, se puede apreciar la distribución de velocidades para cada tipo de rueda.

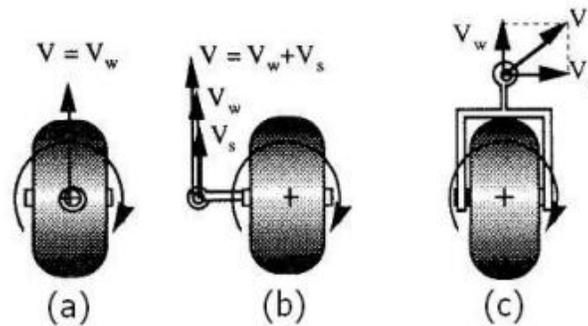


Figura. 4. (a) En rueda convencional simple, (b) En rueda convencional de desplazamiento lateral, (c) En rueda convencional con desviación delantera [6].

En [6] se hace un análisis sobre los tipos de ruedas (b) y (c) en el que se resume lo siguiente: las ruedas de tipo (b) permiten la rotación del eje sobre el punto de contacto entre la rueda y el suelo. Sin embargo, no es posible mover el eje en cualquier dirección sin primero posicionarse correctamente. Las ruedas tipo (c) permiten el movimiento del eje de rotación en cualquier dirección sin posicionar la rueda anteriormente. Esto se observa dado que la dirección de la velocidad del eje (V_s) y de la rueda (V_w) son perpendiculares. La composición de dichas velocidades da como resultado un vector de velocidad V con cualquier dirección. Dando como resultado, que este tipo de ruedas brinde al robot la capacidad de un movimiento omnidireccional.

Existen otros tipos de ruedas que buscan tener una componente activa que produzca tracción en una dirección y una componente pasiva en otra. Las ruedas más conocidas de esta clase son: las ruedas universales, ruedas esféricas y ruedas Mecanum. Las ruedas universales poseen rodillos pasivos ubicados en la periferia de la rueda principal que brindan una componente pasiva y activa a la rueda. Por su parte, las ruedas esféricas, no ponen ninguna restricción

cinemática ya que estas no presentan un eje principal de rotación. La rueda Esférica puede realizar cualquier movimiento en una dirección arbitraria, sin embargo, este tipo de ruedas están limitadas a espacios interiores con pisos o superficies de deslizamiento regulares u homogéneas. Las ruedas de tipo Mecanum, ruedas usadas en la plataforma, son ruedas especiales que permiten una amplia gama de desplazamientos en un mismo plano las cuales tienen instalados sobre su periferia unos rodillos pasivos que forman un ángulo de 45° respecto a la rueda (véase figura 5). La disposición de los rodillos hace que la fuerza ejercida sobre la rueda se descomponga en dos vectores de fuerza: uno perpendicular al eje del rodillo, el cual genera un momento y produce su rotación, y otro paralelo al eje del rodillo, que genera un momento al vehículo. Por lo que es posible, utilizando este tipo de ruedas en una distribución determinada crear un vehículo con movimientos omnidireccional sin tener que emplear complejos mecanismos de dirección, solamente controlando el par y el sentido de giro de cada rueda de manera independiente [7].

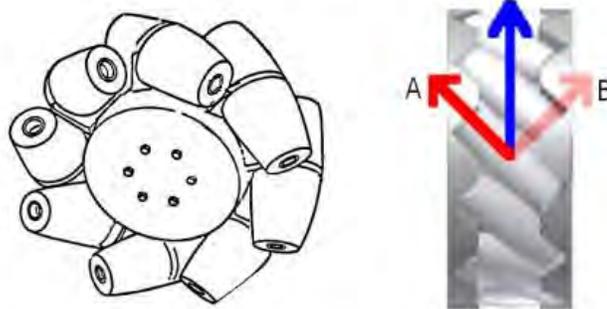


Figura. 5. Ruedas tipo Mecanum [5].

Considerando las componentes A y B resultantes de la descomposición de la fuerza que se muestra en la figura 5, se determina que únicamente es la componente A la que brinda un momento al sistema. Por otro lado, la componente B únicamente proporciona un momento al rodillo que produce su rotación. Lo que implica que el momento otorgado por esta rueda al vehículo posee una dirección con un ángulo respecto de la rueda [5].

Los robots móviles con cuatro ruedas de esta clase se pueden clasificar en dos tipos según la configuración de sus ruedas: el tipo AB y el tipo BB. La manera en que se distingue una configuración de otra depende del ángulo de los rodillos en cada rueda con respecto a un sistema de referencia sobre el vehículo y su combinación dentro del mismo. La figura 6a muestra la diferencia entre la rueda A y la rueda B, se observa que el ángulo de los rodillos de la rueda B llamado β es de 45° , mientras que el ángulo α de la rueda A es de 135° con respecto al sistema de referencia

propuesto. A este tipo de configuración se le conoce como AB. La figura 6b muestra que todos los ángulos con respecto al sistema de referencia son β , por lo que se denomina tipo BB.

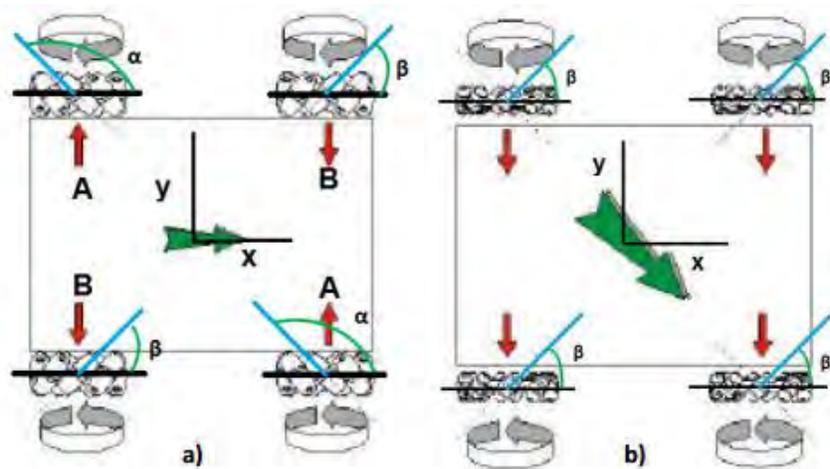


Figura. 6. a) Tipo "AB", b) Tipo "BB" [8].

Aplicaciones

Gracias a las ventajas de los robots con estas características, surgen una gran cantidad de aplicaciones, dentro de las cuales se pueden identificar industriales, académicas, exploración, entre otras. La empresa *KUKA*, dedicada a la fabricación de robots industriales y sistemas de soluciones automatizadas de fabricación cuenta con una línea de robots móviles de este tipo para el desplazamiento de cargas pesadas.



Figura. 7. Plataforma de carga pesada de la empresa KUKA [9].

Por otro lado, la empresa *esclatec* está en proceso de investigación para crear una silla con ruedas omnidireccionales capaz de ser manipulada a través de un dispositivo móvil o Tablet, con el fin de favorecer la inserción laboral de personas con discapacidad. Actualmente ya cuentan con un prototipo que se encuentra en trabajos de mejoramiento y optimización.



Figura. 8. Silla omnidireccional [10].

Las aplicaciones previas son sólo la punta del iceberg para lo que representa la implementación de vehículos omnidireccionales. Hoy en día, muchas ideas y prototipos se encuentran en el proceso de investigación y desarrollo. Sin embargo, es un hecho, que este tipo de robots han llegado y lo seguirán haciendo con una fuerte influencia en nuestra vida diaria.

Capítulo tres

Caracterización del motor e implementación en el sistema

La plataforma con la que se estará trabajando utiliza cuatro ruedas de tipo Mecanum de ocho pulgadas de diámetro dispuestas en una configuración AB que están colocadas junto con los motores y una caja de engranes en un chasis de aluminio de 76.5 [cm] de largo y 60.5 [cm] de ancho mismo que lleva encima una lámina de acrílico perforada. Todos los elementos anteriores están dentro de un marco de aluminio de 94 [cm] por lado. La figura 9 muestra todos los componentes de la plataforma.



Figura. 9. Plataforma omnidireccional a caracterizar.

La plataforma trabaja con motores de 12 [V] PM25R (véase figura 10a) y para cada motor se usa un Talón SR mostrado en la figura 10b que se encarga de suministrar la potencia a los motores para regular la velocidad y sentido de giro dependiendo el pwm que reciba por parte del microcontrolador.



Figura. 10. Motor de 12 [V] y Talon SR

Dentro de las especificaciones del motor, obtenidas de la página del distribuidor (<http://www.andymark.com/Motor-p/am-0255.htm>), destacan las siguientes características:

- Voltaje: 12 [V] DC.
- RPM máximas sin carga: 5,310 (+/- 10%).
- Corriente a rotor libre: 2.7 [A].
- Máxima potencia: 337 Watts (a 2655 rpm, 172 [oz-in] y 68 [A]).
- Par a rotor bloqueado: 2.42 [N-m] o 343.4 [oz-in].
- Corriente rotor bloqueado: 133 [A].

Por su parte, de la hoja de datos del Talon SR, se destacan las siguientes características del dispositivo:

- Voltaje de entrada: 6-28 VDC.
- Corriente continua: 60 A (arriba de 40A se recomienda el uso de ventilador).
- Corriente pico: 100 A.
- Señal de entrada PWM: 0.9-2 ms.
- Resolución de entrada: 10-bit.
- Resolución de salida: 10-bit.

Selección del microcontrolador

El control del desplazamiento de la plataforma se basa en el uso de un microcontrolador, mismo que de acuerdo al algoritmo programado ejecutará las instrucciones para el desplazamiento de la plataforma. Su uso se fundamenta en el hecho de que están diseñados para reducir el costo económico y el consumo de energía de un sistema. El tipo a usar lo determina la aplicación final que se busca para el mismo y es esta la que determina el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos que serán utilizados.

Dentro de los más usados se encuentran los de la familia de Arduino que es una compañía que produce hardware libre, diseña y manufactura placas compuestas por circuitos impresos en las que integran uno de sus microcontroladores y manejan su propio entorno de desarrollo (IDE) en

donde se programa cada placa siendo compatibles con distintos tipos de software como: Adobe Flash, Processing, Max/MSP, Pure Data, etc.

De las distintas y variadas placas que conforman la familia Arduino, el microcontrolador seleccionado para la programación es el Atmega2560 integrado en la placa Arduino MEGA y mostrado en la figura 11 el cual proporciona una velocidad de procesamiento, una capacidad de memoria y una variedad de pines digitales y analógicos aplicables para el proyecto. Además, su programación se facilita usando el entorno de programación propio de Arduino.



Figura. 11. Arduino MEGA.

El sensor destinado para determinar la velocidad angular de cada rueda, es el sensor E4P OEM de US DIGITAL mostrado en la figura 12, el cual es un encoder óptico miniatura de dos canales que proporciona una señal cuadrada para cada uno, misma que será traducida en términos de la velocidad angular correspondiente para cada rueda.



Figura. 12. Sensor E4P OEM.

Dentro de las especificaciones del sensor, se menciona, que el rango de ciclos por revolución que el encoder dará como señal de salida, estará entre los 100 y 160 ciclos del sensor.

Caracterizando el motor a rotor libre

Las características de los motores y la etapa de potencia usadas en este proyecto hacen importante el realizar una serie de ensayos para la caracterización de los motores con los que se estará en operación y dependiendo de estos se aplicarán los conocimientos adquiridos en el trabajo previo para el diseño y programación del control.

Primera ensayo.

El primer ensayo realizado al motor tiene como objetivo el caracterizar los rangos de operación del motor, esto es, encontrar los rangos de voltaje para los cuales el motor tendrá un giro horario, anti horario y en qué momento permanecerá inmóvil para el cambio de dirección. En este caso se utilizó el circuito armado sobre una placa de pruebas protoboard cuyo esquema se muestra en la figura 13. El circuito permite, mediante la variación de voltaje ocasionada por el potenciómetro, que la salida del pwm del Arduino MEGA cambie su ciclo de trabajo y con ello el motor puede cambiar su velocidad, su sentido de giro o llegar a detenerse.

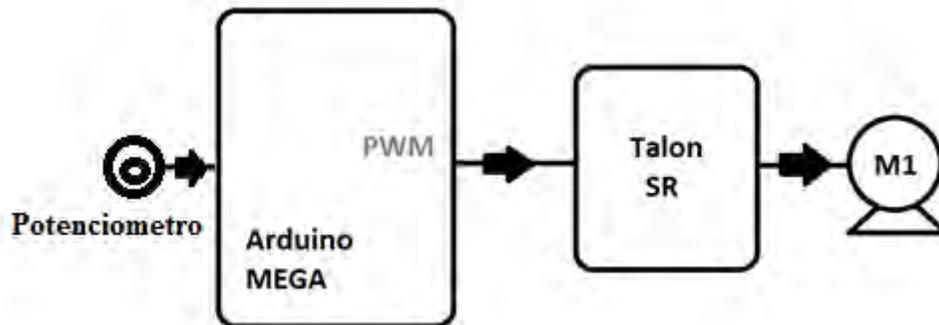


Figura. 13. Esquema del circuito.

Una vez concluido el ensayo, se determinó el rango de operación del motor como se muestra en la Tabla 4 donde se muestran los rangos de voltaje y la acción que realizará el motor estando dentro del mismo.

RANGO DE VOLTAJE [V]	ACCIÓN DEL MOTOR
$V > 3.69$	Motor girando en sentido anti horario
$3.59 < V < 3.69$	Motor detenido
$V < 3.59$	Motor girando en sentido horario

Tabla 4. Rango de acción según voltaje de entrada.

El siguiente ensayo consiste en medir con un osciloscopio el PWM correspondiente a cada rango para poder caracterizar de esa manera los pulsos necesarios a mandar dentro del software de Arduino para poder variar con ello la velocidad del motor.

Segunda ensayo.

El segundo ensayo realizado en el motor determina la variación del ancho de pulso de acuerdo a la variación de voltaje de entrada del potenciómetro. El ensayo se realizó variando medio volt en medio volt iniciando desde cero obteniendo los datos que se muestran en la tabla 5.

Variación de voltaje [V]	Tiempo del pulso en alto [ms]	Tiempo del pulso en bajo [ms]
0	Señal completamente en cero volts	
0.5	0.2	1.8
1	0.4	1.6
1.5	0.6	1.4
2	0.8	1.2
2.5	1	1
3	1.2	0.8
3.5	1.4	0.6
4	1.6	0.4
4.5	1.8	0.2
4.8*	Señal completamente en cinco volts	

Tabla 5. Variación de pulso según voltaje.

*El valor máximo medible fue de 4.8 sin generar ruido en la respuesta.

La tabla anterior muestra el periodo de funcionamiento de nuestro pwm que es de 2 [ms], de la misma forma muestra como el valor máximo de voltaje analógico que puede ser suministrado con el potenciómetro es de 4.8 [V], después de este valor el motor produce se frena o acelera bruscamente, siendo un comportamiento no deseado en su funcionamiento.

En la gráfica mostrada en la figura 14 se puede apreciar como los tiempos altos y bajos del pwm son complementarios entre si dentro de su periodo de funcionamiento. Sin embargo, y a pesar de este comportamiento, comparando los resultados obtenidos en esta prueba con los de la anterior se observa que el tiempo de pulso alto para el cambio de sentido de giro no se encuentra cuando los dos tiempos son iguales, se encuentra entre los valores de 1.4-1.6 [ms].

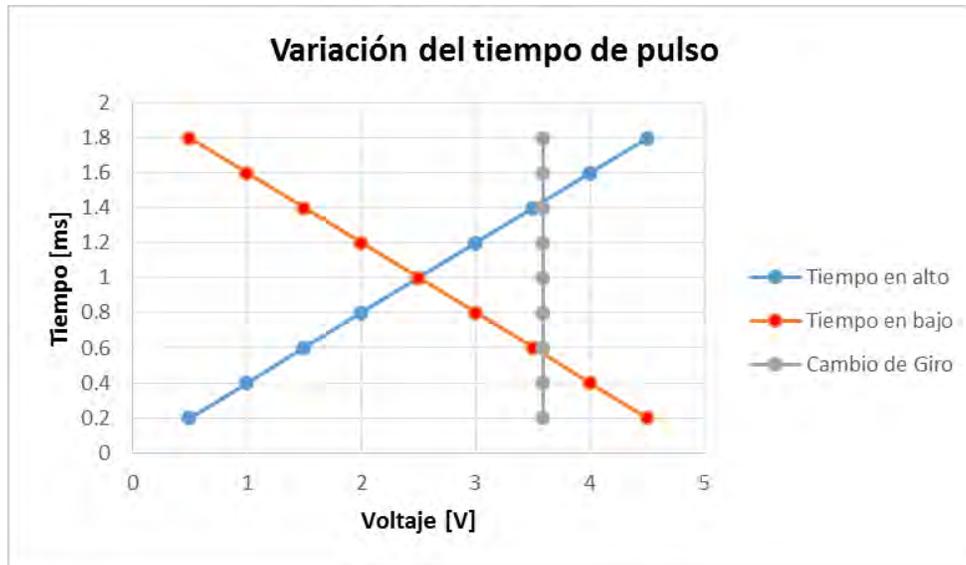


Figura. 14. Variación de pulso con respecto al voltaje.

Tercer ensayo.

En este caso, se calculó la velocidad angular del motor dada una variación de voltaje de entrada y para poder realizar las mediciones se agregó al circuito un encoder independiente de los utilizados por la plataforma el cual al estar acoplado cerca del eje del motor nos desplegará una señal cuadrada en el osciloscopio y con base en esta señal se calculará la velocidad angular. El esquema del circuito armado se muestra en la figura 15.

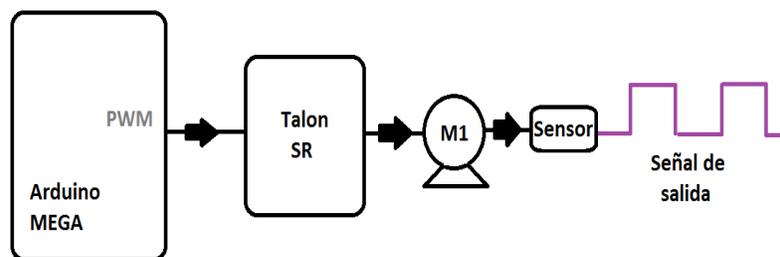


Figura. 15. Esquema del circuito con encoder.

Los datos obtenidos en la prueba se muestran en la tabla 6 que, además serán graficados en la figura 16 para tener un comportamiento gráfico de la velocidad del motor dentro del rango de voltaje al cual fue sometido durante las pruebas, cabe destacar que la velocidad máxima alcanzada puede considerarse dentro del margen aceptado por el distribuidor y que se alcanza mayor velocidad al girar en sentido horario debido a que el valor de voltaje donde el motor se detiene está más cercano a 5[V].

Voltaje [V]	Periodo del pulso [ms]	RPM
2	11	5454
2.2	11	5454
2.5	11	5454
3	19	3158
3.2	27	2222
3.6	0	0
4	27	2222
4.2	19	3158
4.5	17	3529
4.6	13	4615
4.7	13	4615
4.8	13	4615

Tabla 6. Velocidades angulares según voltaje y ancho de pulso.

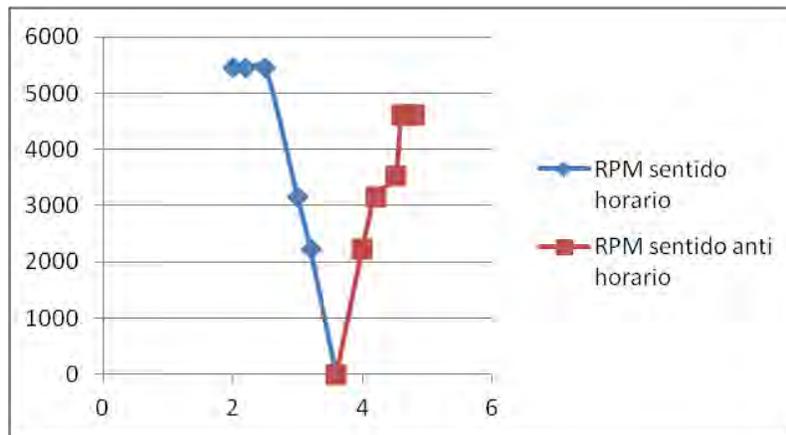


Figura. 16. Rango de velocidades según el voltaje

Puesta en marcha de la plataforma omnidireccional

Una vez hechas las pruebas en un motor independiente lo siguiente es probar el funcionamiento de los motores integrados en la plataforma, comparar su desempeño contra los datos obtenidos en las pruebas de caracterización y así tener una base de acción para el control posterior que requiere la plataforma en su desplazamiento. Los motores de la plataforma están acoplados a una caja de engranes y esta a su vez se encuentra acoplada a una rueda de tipo Mecanum.

La puesta en marcha de la plataforma omnidireccional se realizó dejando sus ruedas fuera del contacto con el suelo, esto para evitar su desplazamiento y tener un entorno de experimentación más controlado. Durante la primera puesta en marcha se apreciaron ciertos comportamientos particulares, unos más esperados que otros. En primer lugar los motores tenían una diferente escala de funcionamiento, mientras que en las pruebas realizadas al motor de 12[V] mostraron que el voltaje requerido del pwm para que el motor se detuviera estaba en el rango de 3.59 a 3.69 [V], para los motores de la plataforma el rango cambió de 3.4 a 3.8 [V], siendo 3.76 [V] el valor en que los cuatro motores permanecen detenidos (véase figura 17). Otro comportamiento esperado era las diferentes velocidades de giro de cada rueda. A pesar de mandar el mismo pulso a cada motor, cada rueda giraba a velocidades angulares diferentes. Finalmente un comportamiento no esperado fue que una de las ruedas giraba en sentido opuesto pese a que el Talón SR indicaba que el sentido de giro debía ser el mismo de las otras tres ruedas, para solucionar este comportamiento se implementó una función en el programa que corregiría el sentido de giro.

$$\left\{ \begin{array}{ll} \text{Sentido antihorario} & \text{Voltajes} > 3.8 \\ \text{Motores detenidos} & \text{Voltaje} = 3.76 \\ \text{Sentido horario} & \text{Voltajes} < 3.4 \end{array} \right.$$

Figura. 17. Rango de operación de los motores.

La función que se requiere para que el motor cuyo giro es diferente al de los otros tres motores dé como resultado un valor en el rango opuesto de voltaje pero manteniendo la misma diferencia se logra con la ecuación (1). Por ejemplo, si a los tres motores que giran en el mismo sentido se les envía 0.5 [V] arriba del valor neutro, al motor que gira en sentido inverso se le debe mandar menos 0.5 [V] de ese valor. El valor neutro está dado por la referencia en el que el voltaje consigue que los cuatro motores se encuentran detenidos, en este caso es 3.76 [V].

$$\text{Voltaje de salida} = 3.76 - (\text{Voltaje requerido} - 3.76) \quad (1)$$

La función anterior corrige el sentido de giro del cuarto motor aun cuando el sentido de giro cambie completamente, esto quiere decir que no importa qué valor de voltaje se decida enviarle a los motores para cambiar su sentido y velocidad, la función hará que el cuarto motor se compense y gire en el mismo sentido y a la velocidad que le corresponda para el valor de voltaje seleccionado.

Previo al cálculo de la velocidad angular de cada rueda se realizó una prueba adicional en la que se buscó el valor de los pulsos por encoder que representaban una revolución. Por datos del

fabricante se sabe que dicho valor se encuentra dentro del rango de 100 a 160 pulsos, lo que llevó a experimentar primero con los valores límites del rango y de acuerdo a los resultados se hicieron otras dos mediciones que mostraron el valor de 150 como el número de pulsos que representan una revolución para nuestro sistema. En la figura 18 se puede apreciar un diagrama con las cuatro mediciones y sus resultados.

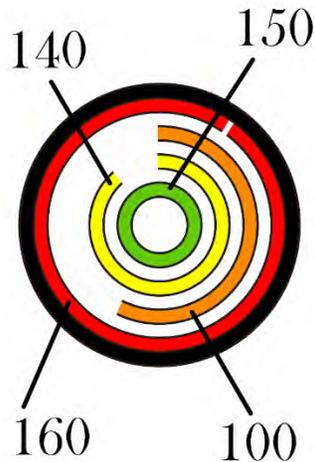


Figura. 18. Pulsos del encoder y su rotación correspondiente.

Teniendo en cuenta este dato se obtuvo una gráfica de la velocidad angular para cada rueda a lo largo de un periodo de tiempo en el que se incrementó paulatinamente el pwm enviado. Las gráficas que pertenecen al motor A y B se muestran en la figura 19. Así mismo, estas gráficas serán usadas posteriormente para el diseño y el cálculo del control PID en cada motor.

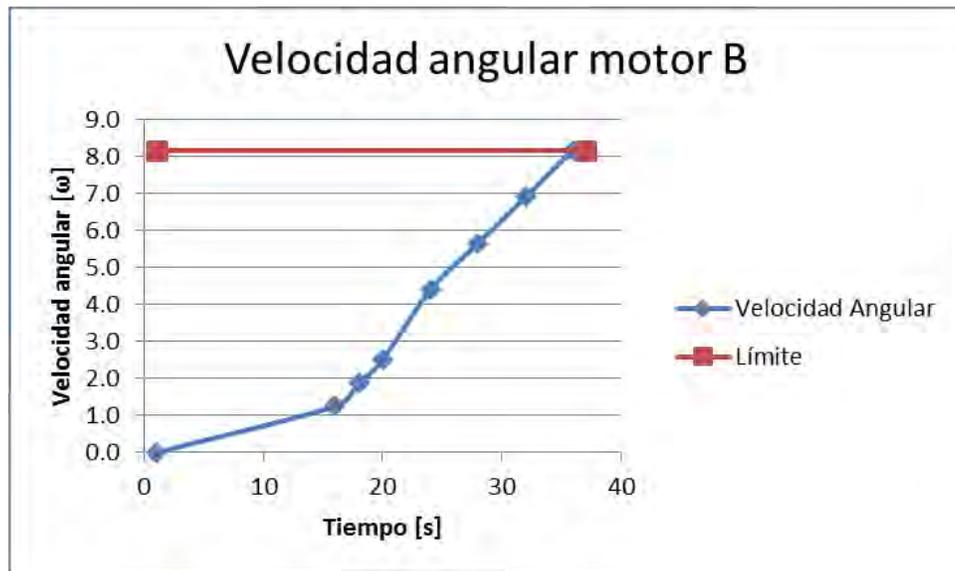
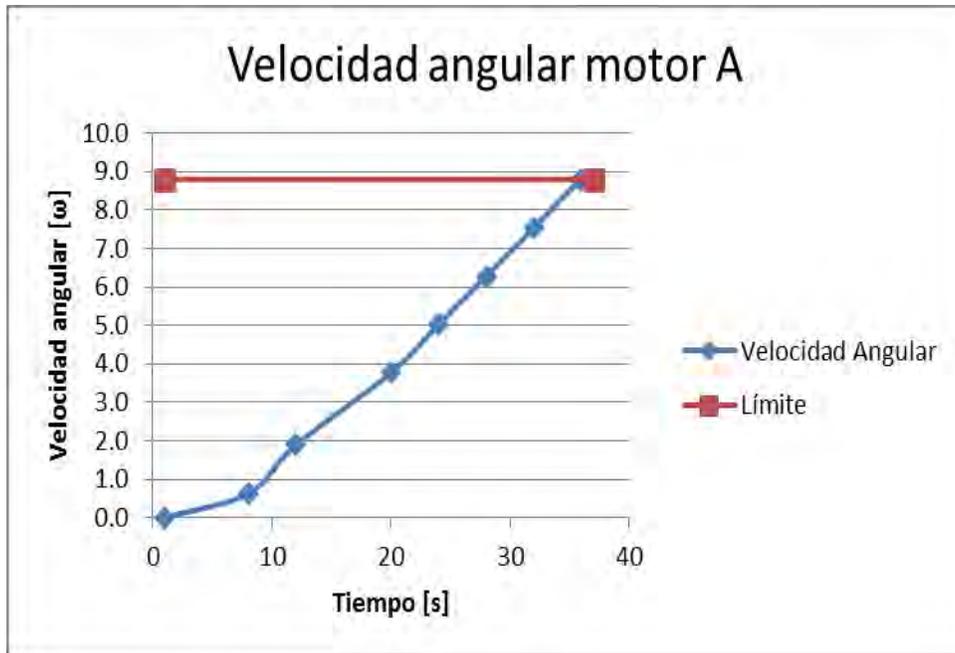


Figura. 19. Velocidades angulares de las rueda A y B.

Al igual que en la figura 19 en la figura 20 se observan las gráficas correspondientes a la velocidad angular de los motores C y D que igualmente serán usadas para el diseño y los cálculos necesarios para su control.

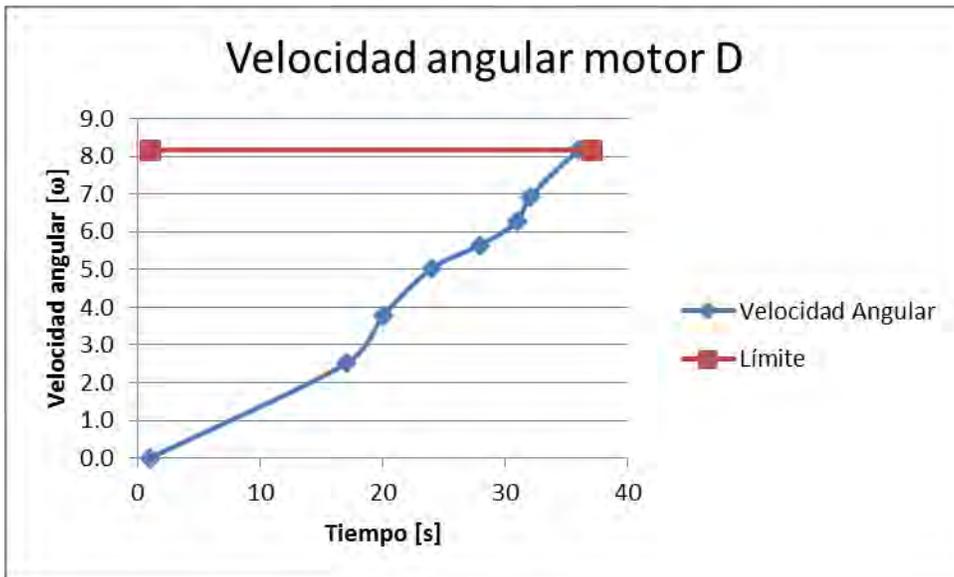
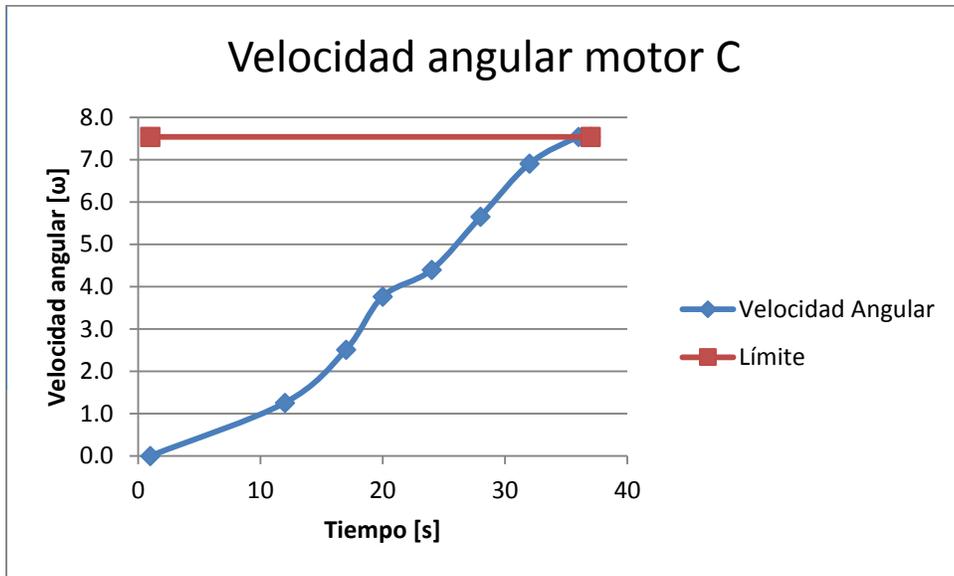


Figura. 20. Velocidades angulares de las rueda B y D.

Capítulo cuatro

Diseño del control

Diversas teorías y técnicas de control han sido desarrolladas en las últimas décadas, no obstante, los controladores Proporcional-Integral-Derivativo (PID) son utilizados en los casos prácticos con mayor frecuencia, de hecho, en los ambientes industriales actualmente representan más del 98% de los controladores implementados. Su estructura simple hace de ellos una herramienta relativamente fácil de sintonizar y adicionalmente proporcionan un rendimiento aceptable para un amplio rango de procesos. En particular los controladores PID son adecuados para sistemas cuya dinámica puede ser modelada de forma aceptable por una ecuación de primer o segundo orden [11].

En la línea del tiempo del control el primer trabajo de gran valor que llegó a la luz fue la máquina de vapor de James Watt, con su regulador de velocidad centrífuga para el control de la velocidad de su máquina. Posteriormente en el siglo XVIII Minorsky trabajó en controladores automáticos para el guiado de embarcaciones. Hazen en 1934 introdujo el término *servomecanismos* en los sistemas de control de posición y Nyquist, dos años antes, diseñó un procedimiento capaz de determinar la estabilidad de un sistema en lazo cerrado.

Durante la década de los cuarenta y cincuenta muchos sistemas de control industrial utilizaban controladores PID. A comienzos de la década de los cuarenta Ziegler y Nichols establecieron reglas para sintonizar estos controladores: las denominadas reglas de sintonización de Ziegler-Nichols. A finales de los años cuarenta y principios de los cincuenta se desarrolló por completo el método del lugar de las raíces propuesto por Evans [12].

Métodos de sintonización.

La sintonización de los controladores PID consiste en la determinación del ajuste de sus parámetros K_c , T_i y T_d para lograr un comportamiento del sistema de control aceptable y relativamente robusto de conformidad con algún criterio de desempeño establecido [13]. Existen varios métodos y algoritmos que se han desarrollado para la sintonización de controladores. A pesar de ello, estos pueden dividirse en dos grandes grupos dependiendo del tipo de controlador: si se trata de un controlador en lazo cerrado o abierto.

Métodos de sintonización de lazo abierto:

- **Criterio de Cohen-Coon:** denominado también como método de ajuste de lazo abierto. En el cual se aplica un escalón directamente a la planta sin considerar el

controlador. La respuesta obtenida al aplicar el estímulo escalón es la fuente de información para completar los parámetros del controlador [14]. El método parte del usado por Ziegler-Nichols. En el desarrollo de su método de sintonización Ziegler y Nichols no consideraron que el proceso fuera auto regulado. Cohen y Coon introdujeron, entonces, un índice de auto regulación definido y plantearon nuevas ecuaciones de sintonización [15].

- **Método de López:** se basa en criterios integrales con ecuaciones para el cálculo de los parámetros del controlador. Definiendo una función de costo, la cual se calcula al derivar una función que depende del error y del tiempo. De esta ecuación se obtiene un valor que caracteriza la respuesta del sistema, y entre menor sea el valor, mejor será el desempeño del sistema de control.
- **Método de Sung, O, Lee, Lee y Yi:** El procedimiento está basado en el mejor modelo de segundo orden más tiempo muerto que se pueda obtener para lazos de control que funcionan como reguladores. El criterio de desempeño corresponde a la minimización de la ITAE y el controlador un PID-Ideal [15].
- **Método de Ziegler-Nichols:** conocido también como el método de la oscilación sostenida. Es un algoritmo basado en la respuesta en frecuencia del proceso que determina una ganancia proporcional crítica, un período de oscilación sostenida, las constantes de tiempo integral y derivativa que se fijan en función del período de la oscilación mantenida, la ganancia proporcional que se fija en función de la ganancia crítica y la constante de tiempo derivativa que es igual a un cuarto de la constante de tiempo integral. Con base a ellas, Ziegler y Nichols propusieron una serie de expresiones que permitían sintonizar el control PID [16].

Métodos de sintonización de lazo cerrado:

- **Método Relay-Feedback:** es un método sencillo que consiste únicamente en aplicar escalones on-off a la salida del controlador con una amplitud h . Ésta altura debe ser del 5 al 10% del valor de salida nominal del controlador. El lazo comenzará a oscilar alrededor del SetPoint. La máxima amplitud de la señal de la variable de proceso se utiliza para determinar la ganancia última. El periodo de la señal de corresponderá al periodo último del sistema. Este método presenta la ventaja de especificar solo un parámetro (la altura del paso h), el tiempo que toma hacer el test es corto y la información obtenida es muy útil para la

sintonización del controlador utilizando finalmente las reglas de sintonía de Ziegler-Nichols o Tyreus-Luyben.

- **Método de Tyreus-Luyben:** fue diseñado específicamente para aplicaciones de procesos y es menos agresivo que el método de Ziegler y Nichols puesto que reduce los sobre picos. Esto se consigue con menores valores para la ganancia y mayores tiempos integrales. Éste método es el más empleado por investigadores en el área de control de procesos [17].

A pesar de la existencia de varios métodos que permiten determinar los elementos necesarios para construir un controlador PID se consideraron en principio tres métodos para elegir el que mejor se adaptaba a nuestro sistema. Los métodos evaluados fueron: el método de Cohen y Coon, el método de Sung et al. y la metodología propuesta por Ziegler y Nichols. Con base en [15] el método de Sung et al. es el mejor método si se requiere de una respuesta rápida con el menor error máximo posible. Sin embargo, la robustez no es tan buena como en el caso de Ziegler y Nichols y, siendo que la curva de reacción de este método se asemeja a la que nuestro sistema presenta, se ha decidido seguir esta última metodología para la sintonización de los parámetros del controlador.

En nuestro caso la respuesta obtenida por el sistema al someterlo a una prueba de escalón (figuras 19 y 20), comparando la velocidad angular del motor con respecto al tiempo se asemeja a la mostrada en la figura 21, la cual representa la curva propuesta por el método seleccionado y de la cual se pueden estimar gráficamente tanto la función de transferencia de la planta como las constantes PID del controlador.

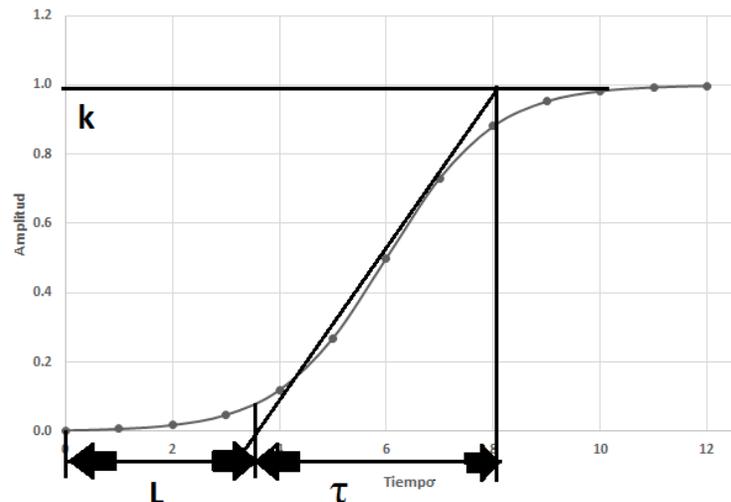


Figura. 21. Curva general de una respuesta en S.

Los parámetros que se necesitan obtener de la gráfica son: el tiempo de retardo L , la constante de tiempo τ y la recta constante K . Estos parámetros se obtienen al trazar una línea tangente a la curva con forma de S en el punto de inflexión de esta. Posteriormente, conforme a la tabla 7 y a la tabla 8 se podrán determinar las constantes PID del controlador.

Tipo de controlador	K_p	T_i	T_d
P	$\frac{\tau}{L}$	∞	0
PI	$0,9 \frac{\tau}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{\tau}{L}$	$2L$	$0,5L$

Tabla 7. Método Ziegler y Nichols con tiempos de integración y derivación.

Tipo de controlador	K_p	K_i	K_d
P	$\frac{\tau}{L}$	0	0
PI	$0,9 \frac{\tau}{L}$	$0,27 \frac{\tau}{L^2}$	0
PID	$1,2 \frac{\tau}{L}$	$0,6 \frac{\tau}{L^2}$	$0,6 \frac{\tau}{L}$

Tabla 8. Método Ziegler y Nichols con constantes PID.

Este método de sintonización como ya se ha mencionado nos permite determinar una aproximación de la función de transferencia del sistema, la forma de hacerlo es mediante la ecuación 2, la cual está en función del tiempo de retardo L , la constante de tiempo τ y la recta constante K .

$$Y(s) = \frac{K e^{-Ls}}{\tau s + 1} \quad (2)$$

Respuestas obtenidas de los motores

En la tabla 9 se muestran los parámetros obtenidos para cada uno de los motores utilizando el método de la curva de reacción de Ziegler-Nichols. En ella se muestra todos los parámetros calculados para su respectivo motor.

	MOTOR A	MOTOR B	MOTOR C	MOTOR D
K	8,8	8,2	7,5	8,2
τ	26	22	23	23
L	8	13	9	11

Tabla 9. Parámetros obtenidos en las pruebas.

A modo de explicación del método se usará el motor A y su gráfica obtenida en la figura 19 para realizar todo el proceso de cálculo. En la figura 22 se observan las líneas de referencia trazadas sobre la gráfica de la velocidad angular del motor. De esta gráfica obtenemos L, τ y K que corresponden a los mostrados en la tabla 9.

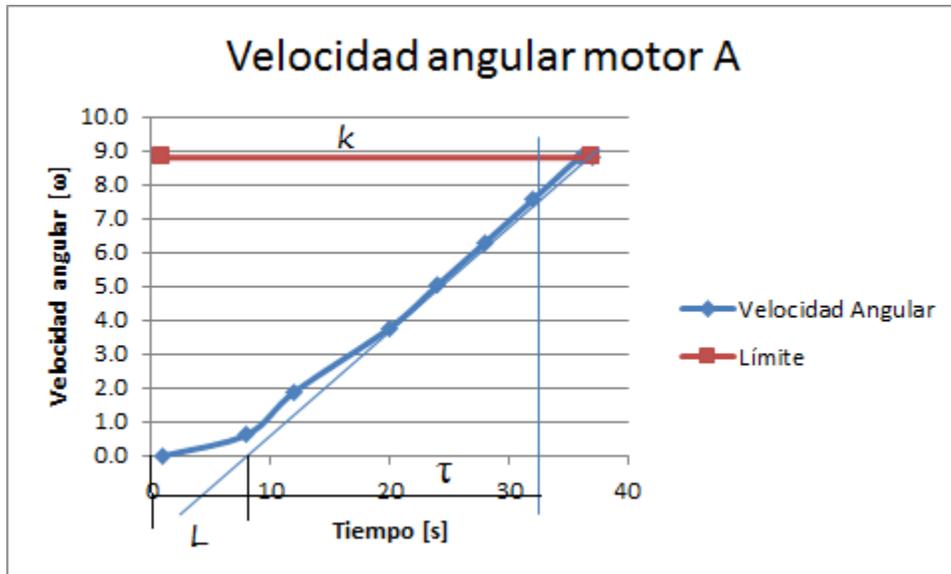


Figura. 22. Respuestas del motores A con los valores K, L y τ .

De los valores obtenidos y, usando las ecuaciones propuestas por el método (ecuaciones 3, 4, 5 y 6), se obtienen las constantes K_p , K_i , K_d y la función de transferencia teórica que representa al motor A.

$$K_p = 1,2 \frac{\tau}{L} = 1,2 \frac{26}{8} = 3.9 \quad (3)$$

$$K_i = 0,6 \frac{\tau}{L^2} = 0,6 \frac{26}{8^2} = 0.24 \quad (4)$$

$$K_d = 0,6 \frac{\tau}{L} = 0,6 \frac{26}{8} = 1.95 \quad (5)$$

$$Y(s) = \frac{K e^{-Ls}}{\tau s + 1} = \frac{8,8 e^{-8s}}{26s + 1} \quad (6)$$

La tabla 10 muestra los resultados del cálculo realizado para los cuatro motores usando la misma metodología explicada para el motor A para obtener las constantes y su respectiva función de transferencia que de acuerdo a la teoría se debe asemejar a la función real de cada motor.

MOTOR A				Función de transferencia
Tipo de controlador	Kp	Ti	Td	$Y(s) = \frac{8,8 e^{-34s}}{26s + 1}$
P	3.250	0.000	0.000	
PI	2.925	26.667	0.000	
PID	3.900	16.000	4.000	
Tipo de controlador	Kp	Ki	Kd	
P	3.250	0.000	0.000	
PI	2.925	0.110	0.000	
PID	3.900	0.244	1.950	
MOTOR B				
Tipo de controlador	Kp	Ti	Td	$Y(s) = \frac{8,2 e^{-13s}}{22s + 1}$
P	1.692	0.000	0.000	
PI	1.523	43.333	0.000	
PID	2.031	26.000	6.500	
Tipo de controlador	Kp	Ki	Kd	
P	1.692	0.000	0.000	
PI	1.523	0.035	0.000	
PID	2.031	0.078	1.950	
MOTOR C				
Tipo de controlador	Kp	Ti	Td	$Y(s) = \frac{7,5 e^{-9s}}{23s + 1}$
P	2.556	0.000	0.000	
PI	2.300	30.000	0.000	
PID	3.067	18.000	4.500	
Tipo de controlador	Kp	Ki	Kd	
P	2.556	0.000	0.000	
PI	2.300	0.077	0.000	
PID	3.067	0.170	1.533	
MOTOR D				
Tipo de controlador	Kp	Ti	Td	$Y(s) = \frac{8,2e^{-11s}}{23s + 1}$
P	2.091	0.000	0.000	
PI	1.882	36.667	0.000	
PID	2.509	22.000	5.500	
Tipo de controlador	Kp	Ki	Kd	
P	2.091	0.000	0.000	
PI	1.882	0.051	0.000	
PID	2.509	0.114	1.255	

Tabla 10. Parámetros del controlador para los motores.

Control de los motores

La implementación del control para cada motor requiere de tres partes básicas: la función del control PID, la función de transferencia del motor y la función de transferencia general del motor que englobe estas dos y que deberá involucrar una retroalimentación para pasar de un sistema en lazo abierto a uno en lazo cerrado. En la figura 23 se muestra el diagrama de bloques de nuestro sistema para cada motor.

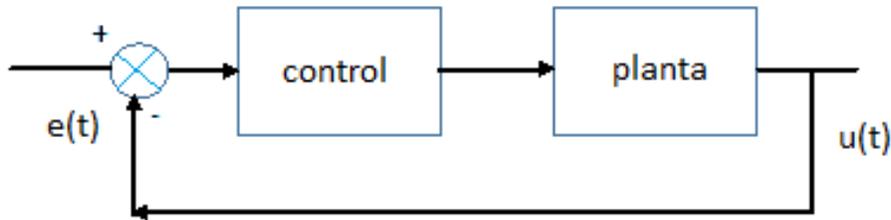


Figura. 23. Diagrama de bloques del sistema de una planta con un control y retroalimentación

Previamente, usando el método de sintonización propuesto, se calcularon los parámetros y la función de transferencia de cada motor, faltando por definir las funciones PID de cada motor y para determinar las funciones de los controladores se requiere utilizar la siguiente ecuación.

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d * s \quad (4)$$

De esa expresión podemos obtener la tabla 11 que muestra las funciones de control para su respectivo motor.

MOTOR A	$G_{cA}(s) = \frac{0.488s^2 + 0.975s + 0.015}{s}$
MOTOR B	$G_{cB}(s) = \frac{0.488s^2 + 0.776s + 0.011}{s}$
MOTOR C	$G_{cC}(s) = \frac{0.431s^2 + 0.863s + 0.013}{s}$
MOTOR D	$G_{cD}(s) = \frac{0.406s^2 + 0.812s + 0.012}{s}$

Tabla 11. Funciones de los controladores PID

Lo anterior nos lleva a obtener la tabla 12 en la que se muestra la función de transferencia general aplicable para cada motor incluyendo la retroalimentación.

MOTOR A	$H_{cA}(s) = \frac{8.8e^{-34s} (0.488s^2 + 0.975s + 0.015)}{8.8e^{-34s} (0.488s^2 + 0.975s + 0.015) + (26s + 1)s}$
MOTOR B	$H_{cB}(s) = \frac{8.2e^{-34s} (0.488s^2 + 0.776s + 0.011)}{8.2e^{-34s} (0.488s^2 + 0.776s + 0.011) + (23s + 1)s}$
MOTOR C	$H_{cC}(s) = \frac{8.2e^{-34s} (0.431s^2 + 0.863s + 0.013)}{8.2e^{-34s} (0.431s^2 + 0.863s + 0.013) + (23s + 1)s}$
MOTOR D	$H_{cD}(s) = \frac{8.2e^{-34s} (0.406s^2 + 0.812s + 0.012)}{8.2e^{-34s} (0.406s^2 + 0.812s + 0.012) + (23s + 1)s}$

Tabla 12. Funciones de transferencia generales.

Finalmente con lo anterior podemos dibujar el diagrama general del control para nuestro sistema, el cual se observa en la figura 24.

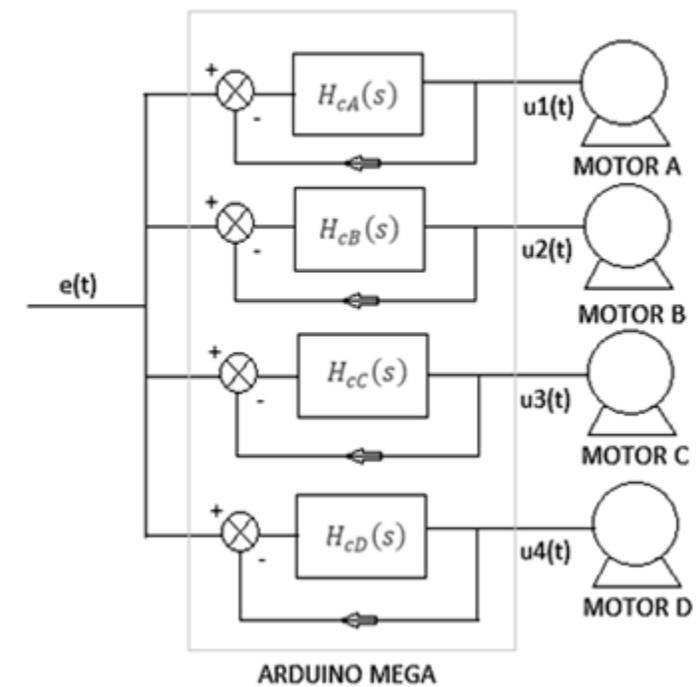


Figura. 24. Diagrama general del sistema con control por motor.

Diseño del control

El método de la curva de reacción de Ziegler-Nichols nos ha ayudado a calcular las constantes que deberán ser usadas para un control proporcional, integral y derivativo, a su vez nos ha ayudado a calcular una función de transferencia que de acuerdo al método debe acercarse a la función real de la planta. Los cálculos previos son cruciales para poder implementar un control PID, esto porque

son las constantes que involucra el control las que definirán que tan rápido se alcanza el estado estable y la acotación del sistema.

Iniciaremos el diseño recordando la acción que cada control realiza y con base en ello se definirá la forma en la que el control será implementado. El primer control, el control proporcional, es una acción que compara los cambios entre una entrada (referencia) y una salida (lectura del sistema) en el tiempo. Esta acción trabaja inmediatamente por lo que el control proporcional trabajará en cambios presentes. La figura 25 muestra como el control proporcional actúa sobre una señal de error que en este caso simplemente es la diferencia de la referencia y la lectura hecha por el sensor de retroalimentación. La señal de control proporcional denominada u_p simplemente será la misma señal, pero modificada por el producto de la constante k_p y es de acuerdo a este valor la variación que la señal presentará.

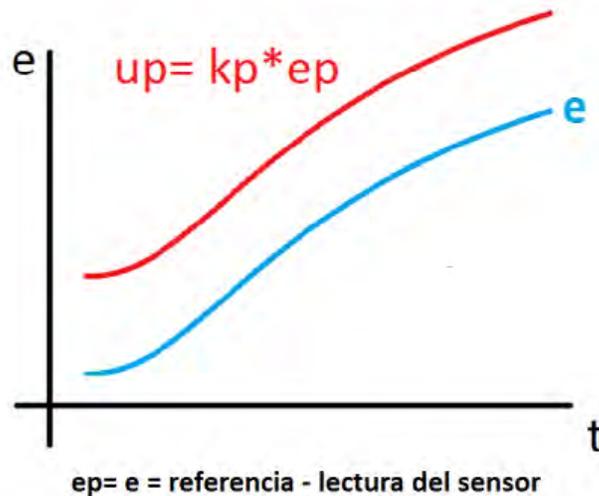


Figura. 25. Acción proporcional.

El control integral tiene como acción de controlador responder a un error acumulado en el tiempo cambiando la señal de salida tanto como se necesite para eliminar completamente el error. Por este motivo se dice que el control integral actuará en el pasado. El error que la acción integral debe de medir es un error acumulativo, esto quiere decir, que hará una comparación entre el valor del error actual con la suma de todos los errores previos y su acción de control, denominada u_i , dará como resultado una señal que sume el área bajo la curva de la señal de error en el tiempo t en el que se realice la medición multiplicada por la constante k_i . Su comportamiento puede observarse en la figura 26.

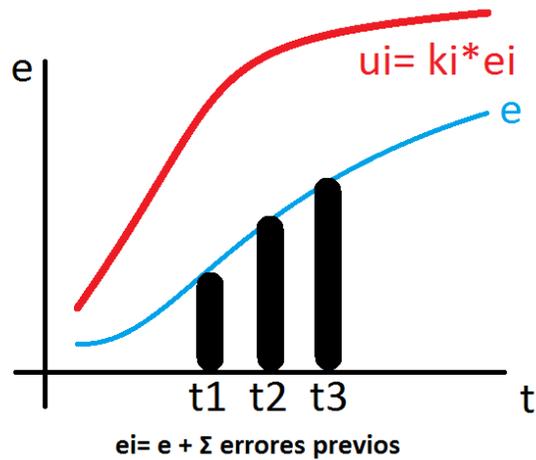


Figura. 26. Acción integral.

La acción de un control derivativo reacciona a que tan rápido cambia la entrada respecto al tiempo alterando la señal de salida en proporción con la diferencia de cambio de entrada. Esta acción se dice que trabaja hacia el futuro. El error para la acción derivativa se calcula recordando que la derivada es la pendiente a la curva y, con base en esta, se tiene que el error derivativo será la diferencia del valor actual y el valor anterior dividida entre el tiempo de muestreo como muestra la figura 27 y su señal de control, definida como u_d , será el producto de este error derivativo y la constante k_d .

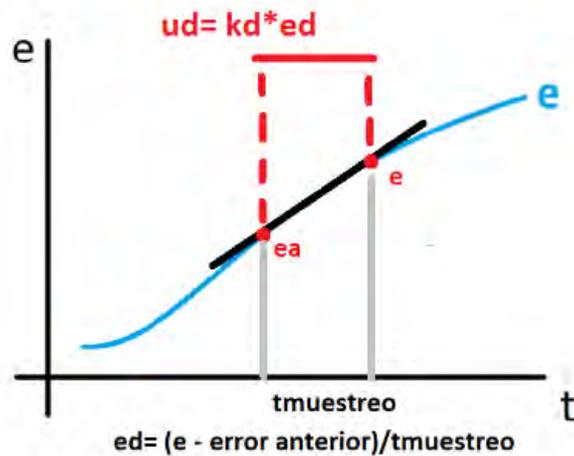


Figura. 27. Acción derivativa.

Capítulo cinco

Diseño del circuito impreso e implementación de control en Arduino

El diseño del circuito impreso requerido para el controlador de la plataforma se realizó mediante el uso del software de diseño ExpressPCB que brinda una mayor libertad para el diseño de pistas, componentes y manipulación del circuito, además es un software libre y de bajo peso. El circuito final puede observarse en la figura 28 y después de ello se hace un desglose de sus partes.

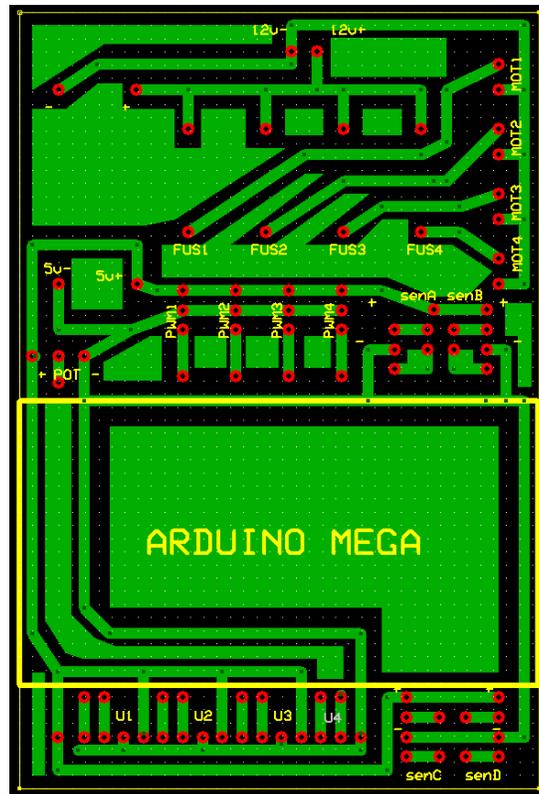
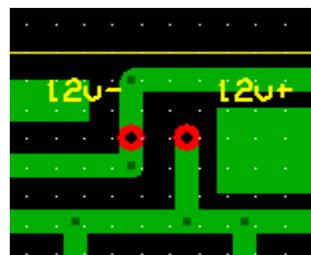


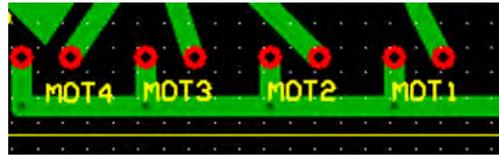
Figura. 28. Circuito completo del controlador.

El circuito del controlador se divide en las siguientes partes:

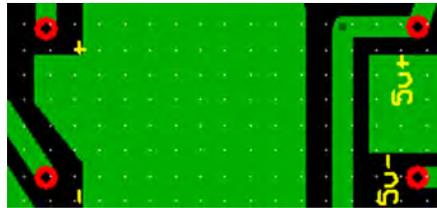
- ➔ Entrada de una batería de 12 [V] como fuente de alimentación.



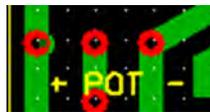
→ Salidas para alimentar los motores.



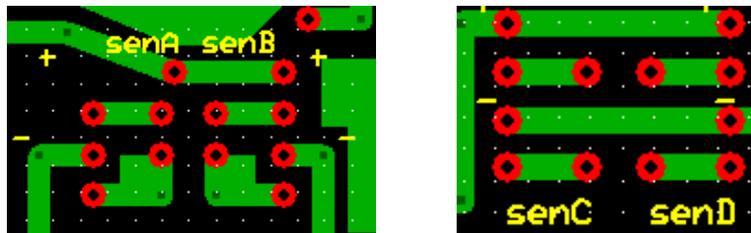
→ Pines de conexión para un regulador de 5 [V].



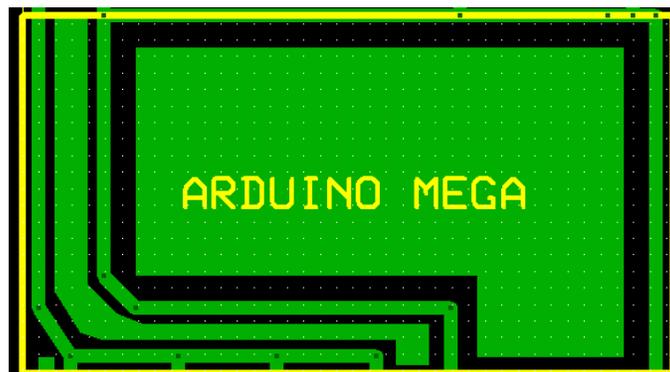
→ Entrada habilitada para un potenciómetro.



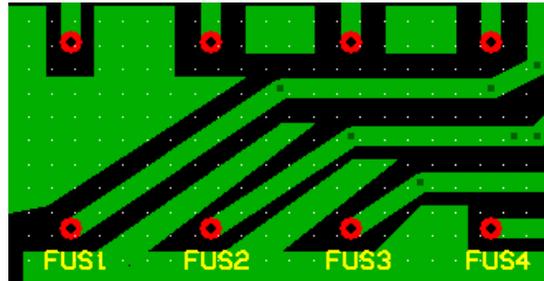
→ Entradas para los encoders ópticos de cada rueda.



→ Espacio para el microcontrolador Arduino MEGA.



→ Sección de fusibles para evitar daños a los motores.

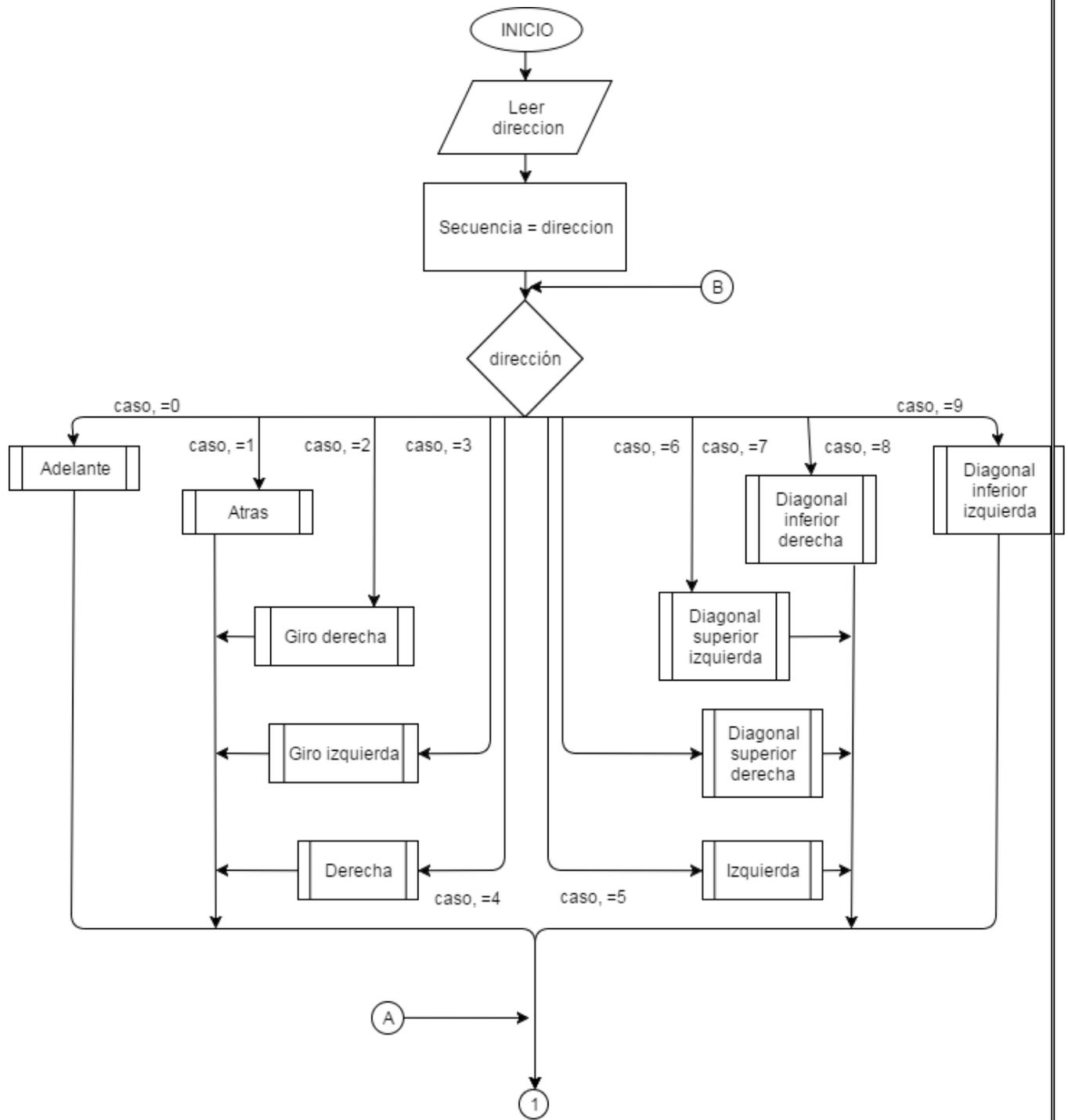


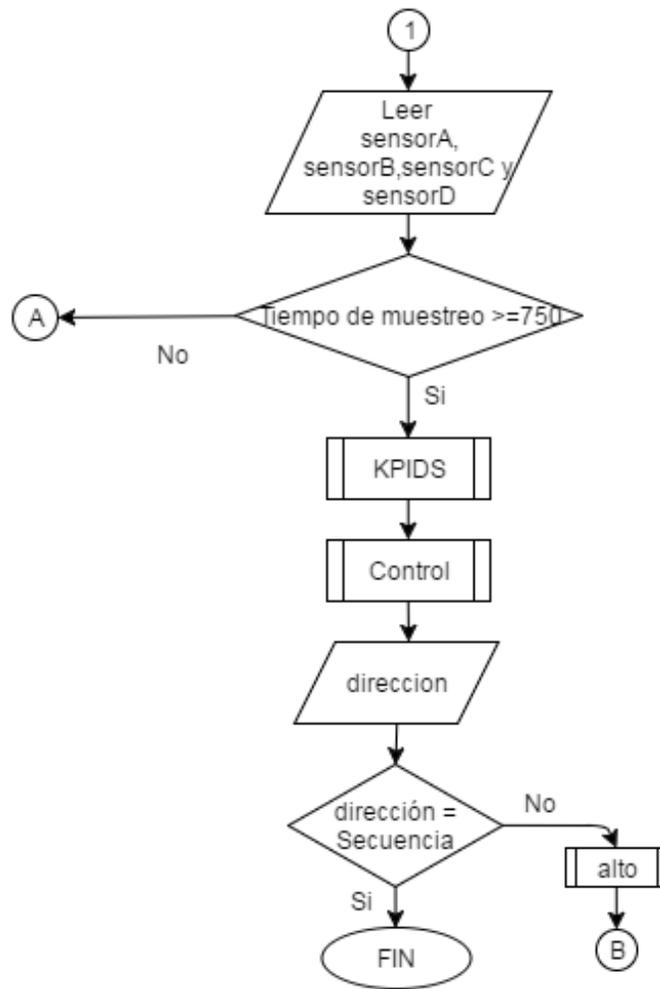
→ Entradas habilitadas para agregar sensores ultrasónicos.



Código del control en Arduino

El código principal básicamente realizará la tarea de lectura de los pulsos del encoder en cada rueda, los transformará en velocidad angular y estos valores serán los que pasen por el proceso de control para generar nuevos pulsos de salida que estabilicen a la plataforma. Así mismo, el código se apoyará de distintas funciones que lo harán más intuitivo. Estas funciones se encargarán de realizar el frenado, cambio de dirección, la selección de constantes y el cálculo del control. A continuación se puede apreciar el diagrama de flujo que representa a grosso modo el código realizado.





Una vez diseñado, manufacturado, terminado el circuito y la programación del controlador en Arduino el siguiente paso son las pruebas de funcionamiento del control que se realizaron sobre la plataforma omnidireccional para estudiar sus resultados.

Capítulo seis

Ajuste de los parámetros del control

Si bien el procedimiento para el cálculo de los parámetros del control se realizó de manera detallada manteniendo la mayor precisión posible al momento de la implementación en la puesta en marcha de la plataforma omnidireccional se presentaron desviaciones en su desplazamiento. Este error se puede deber a que las pruebas de caracterización de los motores se realizaron a rotor libre, no se consideró el posible deslizamiento de las ruedas en el suelo o deberse a la mecánica misma de la plataforma, cuestiones que pueden generar ruido en la retroalimentación del sistema.

En este capítulo se realizarán una serie de compensaciones y ajustes a los valores encontrados de las constantes del controlador para cada rueda, esto para asegurar el menor error posible en su desplazamiento. Los ajustes y compensaciones se llevarán a cabo para todas las direcciones contempladas en la gama de movimientos asegurando con ello que el desplazamiento sea el adecuado.

Ajuste para el movimiento hacia adelante y hacia atrás

Mediante el uso del método de sintonización de Ziegler-Nichols se consiguieron calcular los parámetros necesarios para el control de cada rueda. Sin embargo, la plataforma presentó una serie de desviaciones a pesar del control implementado y debido a estas desviaciones se buscó la manera de compensar este error.

Dentro de la programación del control se estableció como referencias una velocidad angular y un pwm inicial, estos dos parámetros son los que el controlador debe mantener para asegurar un desempeño adecuado. Complementando las referencias anteriores, las constantes K_p , K_i , y K_d son también parámetros importantes que se encargan del buen o mal desempeño del control.

En consecuencia y. con base a lo anterior, se tiene como parámetros para modificar las constantes del controlador y los parámetros de referencia que son el pwm y la velocidad angular de cada rueda. Como primera prueba de variación se propuso una compensación en la velocidad angular de cada rueda, esta compensación, dependiendo el valor y la rueda que se elija incrementará el pwm de la salida del control para darle mayor velocidad angular a la rueda. La ecuación que realiza esa acción se observa en la expresión (5). Donde $controlX$ es la señal de control enviada a la rueda, $PWMRX$ es el pwm de referencia, $compX$ es la compensación en la velocidad angular y finalmente UX es el valor calculado durante el proceso de control de la plataforma.

$$\text{control}X = \text{PWMRX} + \text{comp}X + UX \quad (5)$$

De acuerdo con la expresión anterior y partiendo de los resultados obtenidos en el cálculo del control se obtuvieron los resultados mostrados en la tabla 13 y 14 para los desplazamientos hacia delante y hacia atrás respectivamente. La cuales comparan los resultados obtenidos en el cálculo del control contra los valores modificados para asegurar el comportamiento adecuado de la plataforma.

Rueda A			Rueda B		
Parámetro	Valor calculado o de referencia	Valor corregido	Parámetro	Valor calculado o de referencia	Valor corregido
Kp	3.9	2.1	Kp	2	1.45
Ki	0.24	0.01	Ki	0.08	0.01
Kd	1.95	0.005	Kd	1.95	0.005
PWMRA	193	193	PWMRA	193	193
PWMRB	193	193	PWMRB	193	193
PWMRC	193	193	PWMRC	193	193
PWMRD	193	193	PWMRD	193	193
compA	2	2	compA	2	2
compB	2	2	compB	2	2
compC	2	2	compC	2	2
compD	2	2	compD	2	2
Rueda C			Rueda D		
Parámetro	Valor calculado o de referencia	Valor corregido	Parámetro	Valor calculado o de referencia	Valor corregido
Kp	3	1.45	Kp	2.5	2.1
Ki	0.17	0.01	Ki	0.12	0.01
Kd	1.53	0.005	Kd	1.26	0.005
PWMRA	193	193	PWMRA	193	193
PWMRB	193	193	PWMRB	193	193
PWMRC	193	193	PWMRC	193	193
PWMRD	193	193	PWMRD	193	193
compA	2	2	compA	2	2
compB	2	2	compB	2	2
compC	2	2	compC	2	2
compD	2	2	compD	2	2

Tabla 13. Valores de los parámetros para el movimiento hacia adelante.

Rueda A			Rueda B		
Parámetro	Valor calculado o de referencia	Valor corregido	Parámetro	Valor calculado o de referencia	Valor corregido
Kp	3.9	1.6	Kp	2	1.8
Ki	0.24	0.01	Ki	0.08	0.01
Kd	1.95	0.005	Kd	1.95	0.005
PWMRA	193	193	PWMRA	193	193
PWMRB	193	193	PWMRB	193	193
PWMRC	193	193	PWMRC	193	193
PWMRD	193	193	PWMRD	193	193
compA	2	2	compA	2	2
compB	2	2	compB	2	2
compC	2	2	compC	2	2
compD	2	2	compD	2	2
Rueda C			Rueda D		
Parámetro	Valor calculado o de referencia	Valor corregido	Parámetro	Valor calculado o de referencia	Valor corregido
Kp	3	1.8	Kp	2.5	1.6
Ki	0.17	0.01	Ki	0.12	0.01
Kd	1.53	0.005	Kd	1.26	0.005
PWMRA	193	193	PWMRA	193	193
PWMRB	193	193	PWMRB	193	193
PWMRC	193	193	PWMRC	193	193
PWMRD	193	193	PWMRD	193	193
compA	2	2	compA	2	2
compB	2	2	compB	2	2
compC	2	2	compC	2	2
compD	2	2	compD	2	2

Tabla 14. Valores de los parámetros para el movimiento hacia atrás.

Análisis de velocidades Tangenciales

El análisis previo ofreció una nueva vía para mejorar el control diseñado previamente. Sin embargo, usar el mismo análisis para los movimientos restantes implicaría un tiempo mayor de investigación.

Es por ello, que con el fin de lograr un reajuste que permita un desplazamiento más cercano al ideal

en un corto periodo de tiempo se analizarán con detalle las velocidades lineales de la plataforma omnidireccional. En el capítulo tres se menciona la sintonización de las constantes de control y se aborda sólo el estudio de la velocidad angular de cada rueda, no obstante, este análisis parece poseer cierto grado de ruido que produce un desvío en el desplazamiento. La extensión del análisis a las velocidades lineales pretende entender mejor el porqué de estas desviaciones y hallar una manera de solucionarlas y con ello robustecer el control previamente diseñado.

En la figura 29 se ilustra la plataforma omnidireccional con un sistema de coordenadas que parte de su centro y que servirá como referencia para el estudio. V_{XT} es la velocidad tangencial en cada rueda medida en metros por segundo siendo X la letra de la rueda correspondiente. V_{XR} es la velocidad tangencial del rodillo medida en metros por segundo siendo X la letra de la rueda correspondiente. ω_z es la velocidad angular de la plataforma (la que puede generar una desviación en los movimientos) y está medida en radianes por segundo.

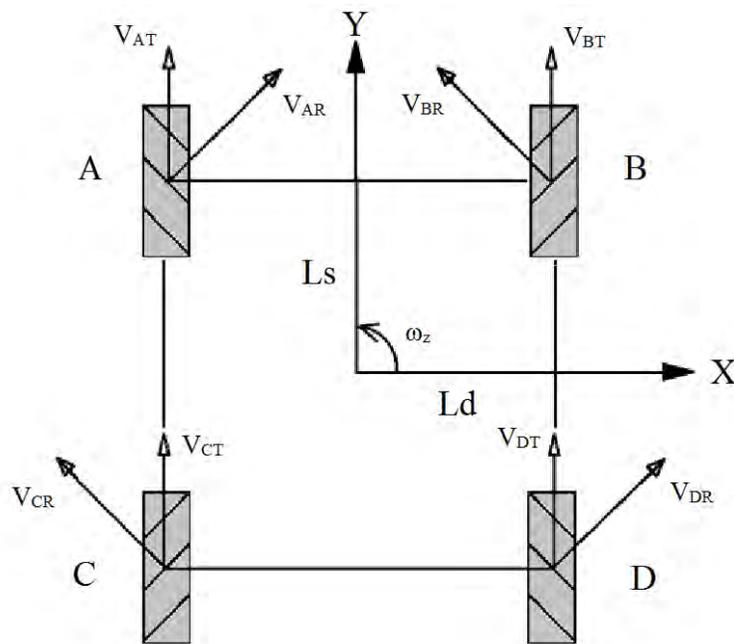


Figura. 29. Plataforma con las velocidades tangenciales, basado en [7].

Usando las ecuaciones cinemáticas propuestas en [18], las cuales proponen que la velocidad de la rueda se puede expresar en términos de la velocidad del vehículo se obtienen las siguientes ecuaciones.

$$\begin{aligned}
V_{AT} &= Vx + Vy - (Ld + Ls) * \omega z \\
V_{BT} &= -Vx + Vy + (Ld + Ls) * \omega z \\
V_{CT} &= -Vx + Vy - (Ld + Ls) * \omega z \\
V_{DT} &= Vx + Vy + (Ld + Ls) * \omega z
\end{aligned} \tag{6}$$

Donde Vx y Vy son las velocidades correspondientes a cada eje coordenado, Ld es la distancia en metros desde el centro de la plataforma al centro del eje de las ruedas laterales y Ls es la distancia medida en metros desde el centro de la plataforma al eje de las ruedas superiores. Las ecuaciones anteriores pueden expresarse matricialmente como se muestra en (7).

$$\begin{bmatrix} V_{AT} \\ V_{BT} \\ V_{CT} \\ V_{DT} \end{bmatrix} = \begin{bmatrix} 1 & 1 & -(Ld + Ls) \\ -1 & 1 & (Ld + Ls) \\ -1 & 1 & -(Ld + Ls) \\ 1 & 1 & (Ld + Ls) \end{bmatrix} \begin{bmatrix} Vx \\ Vy \\ \omega z \end{bmatrix} \tag{7}$$

Resolviendo y despejando los términos de Vx , Vy y ωz obtenemos la ecuación en forma matricial (8).

$$\begin{bmatrix} Vx \\ Vy \\ \omega z \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ Ld+Ls & Ld+Ls & Ld+Ls & Ld+Ls \end{bmatrix} \begin{bmatrix} R\omega_A \\ R\omega_B \\ R\omega_C \\ R\omega_D \end{bmatrix} \tag{8}$$

Donde R es el radio de las ruedas medido en metros y ω_x es la velocidad angular correspondiente a cada rueda en radianes por segundo.

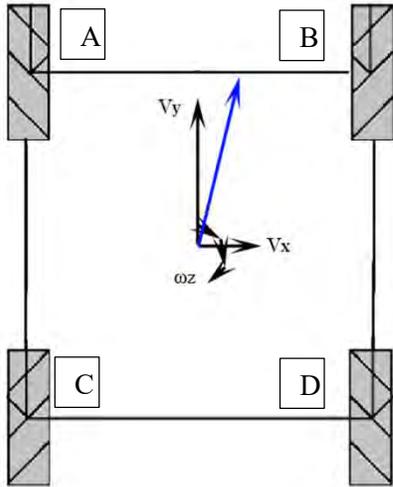
De (8) se puede comprobar los casos de desplazamiento básicos para una plataforma omnidireccional presentados en el capítulo uno.

- Si $\omega_A = \omega_B = \omega_C = \omega_D = \omega$, entonces $Vx=0$, $\omega z=0$ y $Vy=R*\omega$. Esto representa el desplazamiento sobre el eje y .
- Si $\omega_A = \omega_D = \omega$, $-\omega_B = -\omega_C = -\omega$, entonces $Vx=R*\omega$, $\omega z=0$ y $Vy=0$. Esto representa el desplazamiento sobre el eje x .
- Si $\omega_A = \omega_C = \omega$, $-\omega_B = -\omega_D = -\omega$, entonces $Vx=0$, $\omega z = -(R*\omega)/(Ld+Ls)$ y $Vy=0$. Esto representa la rotación en sentido horario de la plataforma.
- Si $\omega_A = \omega_D = \omega$, $\omega_B = \omega_C = 0$, entonces $Vx=(R*\omega)/2$, $\omega z=0$ y $Vy=(R*\omega)/2$. Esto representa el desplazamiento diagonal superior derecha.

Análisis de desviaciones

Usando las ecuaciones previamente descritas se obtendrán los valores aproximados de la componente de la velocidad lineal sobre el eje X y Y, de igual forma se obtendrá la velocidad angular de la plataforma para cada dirección de desplazamiento.

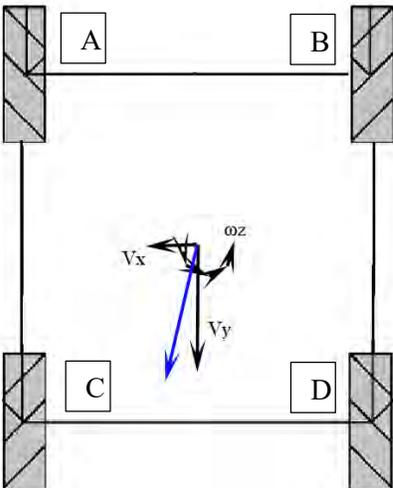
Adelante



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma sufre de una desviación hacia la derecha mostrada por la flecha azul y un giro horario. Por lo tanto requiere modificar los parámetros de los motores B y D.

Vx	Vy	ω_z
0.13	0.74	0.74

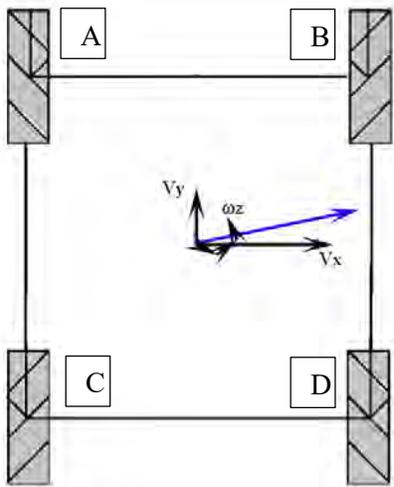
Atrás



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma sufre de una desviación hacia la izquierda mostrada por la flecha azul y un giro anti horario. Por lo tanto requiere modificar los parámetros de los motores A y C.

Vx	Vy	ω_z
-0.15	-0.71	2.37

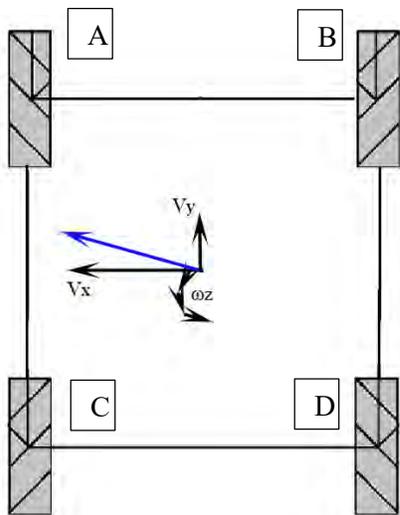
Derecha



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma tiene una desviación diagonal hacia adelante y una rotación anti horaria. Por lo tanto requiere modificar los parámetros de los motores A y D.

V_x	V_y	ω_z
0.74	0.13	0.74

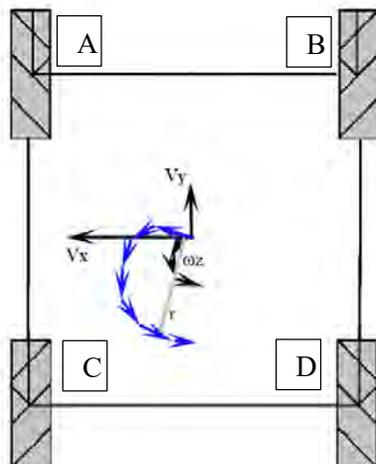
Izquierda



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma tiene una desviación diagonal hacia adelante y una rotación anti horaria. Por lo tanto requiere modificar los parámetros de los motores B y C.

V_x	V_y	ω_z
-0.99	0.13	0.74

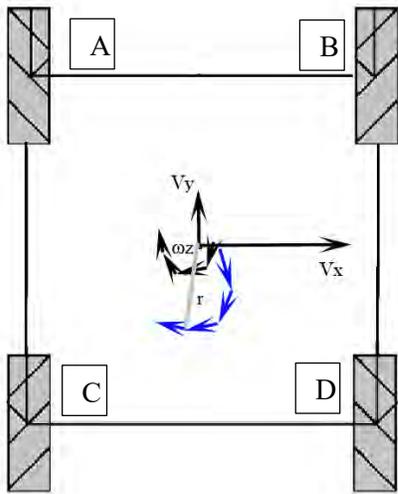
Giro hacia la izquierda



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma al girar tendrá una desviación que la hará rotar con un radio r . Por lo tanto requiere modificar los parámetros de los motores B y D.

V_x	V_y	ω_z
-0.74	0.13	0.74

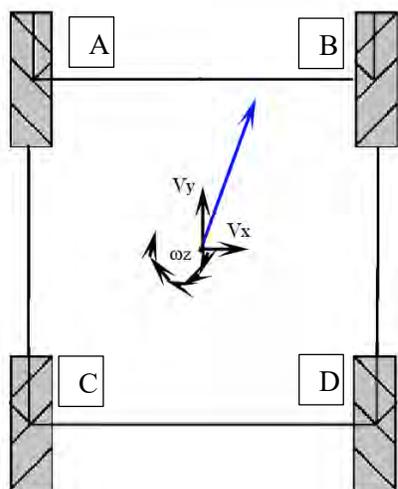
Giro hacia la derecha



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma al girar tendrá una desviación que la hará rotar con un radio r . Por lo tanto requiere modificar los parámetros de los motores A y C.

V_x	V_y	ω_z
0.74	0.13	-0.74

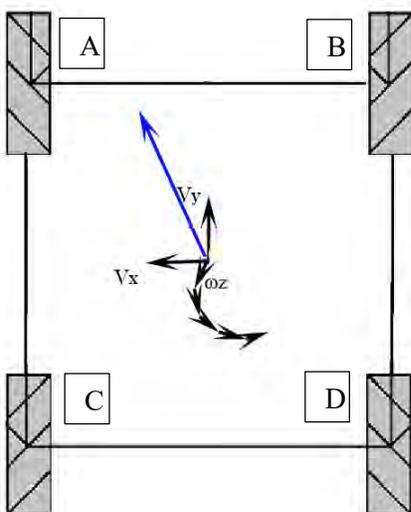
Diagonal superior derecha



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma no se desplazará a 45° y por la velocidad angular alta rotará en sentido horario. Por lo tanto requiere modificar los parámetros del motor A.

V_x	V_y	ω_z
0.13	0.43	-2.54

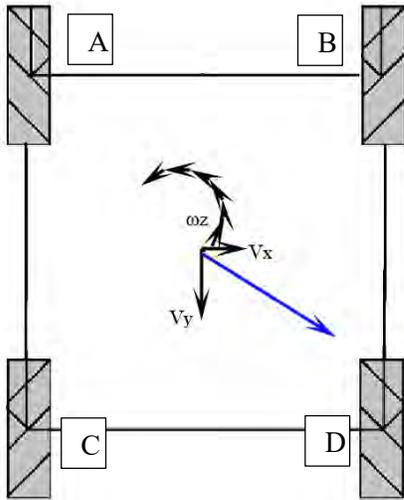
Diagonal superior izquierda



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma no se desplazará a 135° y por la velocidad angular alta rotará en sentido anti horario. Por lo tanto requiere modificar los parámetros del motor B.

V_x	V_y	ω_z
-0.13	0.43	2.54

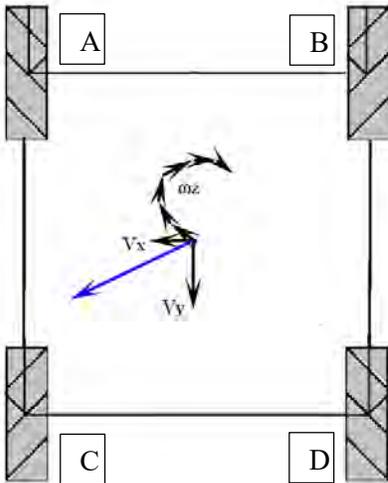
Diagonal inferior derecha



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma no se desplazará a 315° y por la velocidad angular alta rotará en sentido anti horario. Por lo tanto requiere modificar los parámetros del motor B.

Vx	Vy	ω_z
0.13	-0.43	2.52

Diagonal inferior izquierda



Los resultados obtenidos en la tabla y mostrados en la imagen, muestran como la plataforma no se desplazará a 225° y por la velocidad angular alta rotará en sentido horario. Por lo tanto requiere modificar los parámetros del motor A.

Vx	Vy	ω_z
-0.13	-0.43	-2.52

Ajuste final de la constante Kp

Una vez terminado el análisis de las desviaciones y considerando los resultados obtenidos se han determinado que motores deben modificar sus constantes proporcionales para conseguir un mejor desplazamiento. Por ejemplo, en el caso hacia delante se percibe una desviación hacia la derecha y, son entonces, los motores B y D los que requieren la modificación. De esta misma forma se ha determinado que motores deben modificar sus parámetros para las demás direcciones. La forma para encontrar la nueva Kp es relativamente sencilla y se determina usando las ecuaciones del control PID mostradas en la figura 30.

$$\begin{aligned}
ep_{actual} &= \text{referencia} - \text{lectura} \\
ed &= \frac{(ep_{actual} - ep_{anterior})}{\text{tiempo de lectura}} \\
ei_{actual} &= (Ki * ep_{actual}) + ei_{anterior} \\
Uc &= Kp * ep_{actual} + Kd * ed + ei_{actual}
\end{aligned}$$

Figura. 30. Ecuaciones del control PID.

Donde ep es el error proporcional, ed es el error derivativo, ei el error integral y Uc es la señal de control. De acuerdo con la última ecuación se puede despejar la Kp quedando como se muestra en (9).

$$Kp = \frac{Uc - (Kd * ed + ei_{actual})}{ep_{actual}} \quad (9)$$

Con la ecuación anterior se puede obtener la constante proporcional Kp de cualquiera de los motores haciendo una serie de consideraciones. Las consideraciones a realizarse son las siguientes: los valores de Ki y Kd serán los mismos para todos los casos, se trabajará con un valor de Kp inicial de acuerdo al cálculo del control previo y a partir de ese valor se tomarán dos lecturas de cada sensor para determinar el nuevo Kp con la ecuación establecida en (9) y finalmente se considerará una Uc de 2.92 como el valor esperado para la salida del control. De acuerdo con estas consideraciones para el motor A desplazándose hacia adelante el cálculo sería el siguiente.

$$ep_{anterior} = 2.19; ep_{actual} = 0.73 \quad (10)$$

$$Ki = 0.01; Kd = 0.005; Uc = 2.92 \quad (11)$$

$$ed = \frac{(0.73 - 2.19)}{0.75} = -1.95 \quad (12)$$

$$ei_{anterior} = (0.01 * 2.19) = 0.0219 \quad (13)$$

$$ei_{actual} = (0.01 * 0.73) + 0.0219 = 0.0292 \quad (14)$$

$$Kp = \frac{2.92 - ((0.005 * -1.92) + 0.0282)}{0.73} = 3.97 \quad (15)$$

De la misma forma se realiza el cálculo para cada motor y en cada dirección. En la tabla 15 se visualizan los resultados obtenidos para cada motor y para cada desplazamiento de la plataforma.

Movimiento	Constante Kp del motor A	Constante Kp del motor B	Constante Kp del motor C	Constante Kp del motor D
Adelante	3.9750	1.3150	1.9817	1.3161
Atrás	1.3161	1.9800	1.3150	1.9817
Derecha	1.3139	0.9833	3.9750	1.9833
Izquierda	1.3150	1.9817	1.9800	1.3161
Giro derecha	0.9825	1.3161	0.9833	1.9800
Giro Izquierda	1.9800	1.3161	1.3161	1.9800
Diagonal Inferior Derecha	0.9825	1.9800	1.3161	1.9800
Diagonal Inferior Izquierda	1.9817	1.3139	1.9817	1.3150
Diagonal Superior D.	1.9800	0.9817	1.9817	1.3161
Diagonal Superior I.	1.3150	1.9817	0.9817	1.9833

Tabla 15. Constantes Kp calculadas.

Utilizando las constantes Kp calculadas se realizarán las últimas pruebas de funcionamiento de la plataforma y se evaluarán los resultados obtenidos con respecto a las primeras pruebas realizadas con las constantes calculadas por el método de la curva de reacción de Ziegler-Nichols.

Capítulo siete

Resultados

Las primeras pruebas realizadas en la plataforma mostraron que el control empezaba a generar una mejora en su desplazamiento. Sin embargo, este aún presentaba una desviación considerable y para alcanzar el comportamiento ideal aún se requería de una serie de ajustes, correcciones y compensaciones para mejorar el desempeño de la plataforma.

Los resultados obtenidos en estas primeras pruebas se muestran en las figuras de la 31 a la 40 y muestran la comparativa entre el desplazamiento ideal, el real con control y el real sin control para los diferentes tipos de desplazamiento. En ellas se puede apreciar como el control comienza actuar para compensar y corregir el desplazamiento. Sin embargo, este no es suficiente provocando que para cada uno de los casos el desplazamiento siga siendo distante al esperado.

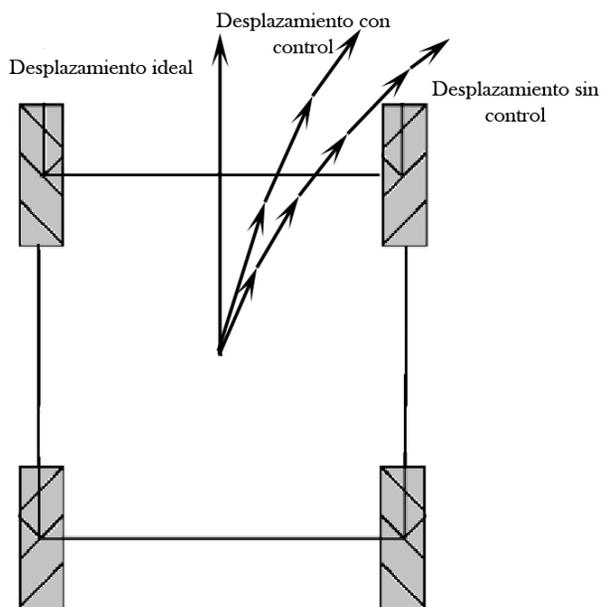


Figura. 31. Prueba de desplazamiento hacia adelante.

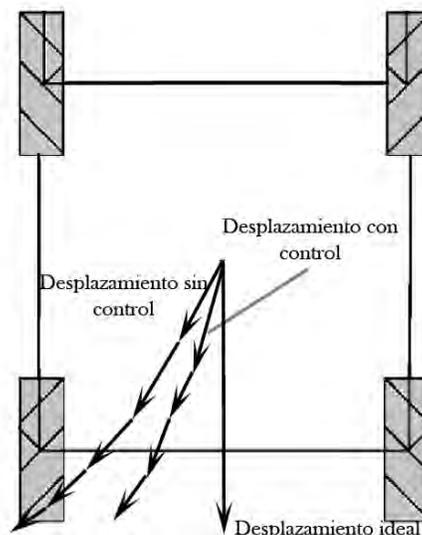


Figura. 32. Prueba de desplazamiento hacia atrás.

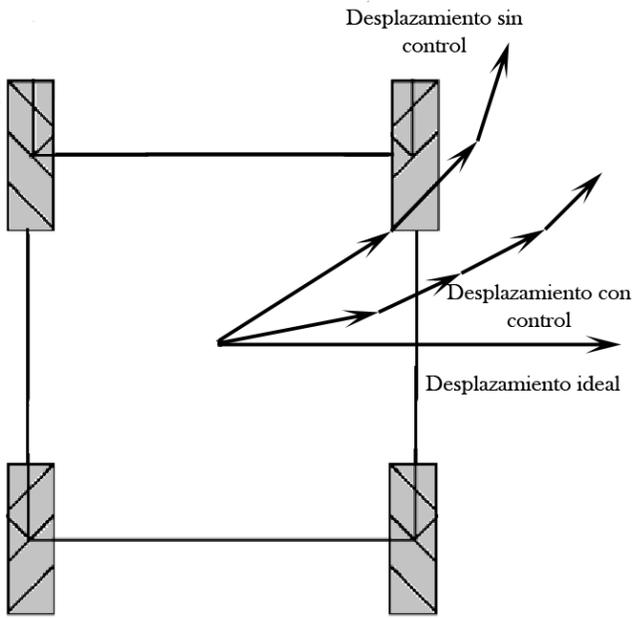


Figura. 33. Prueba de desplazamiento hacia la derecha.

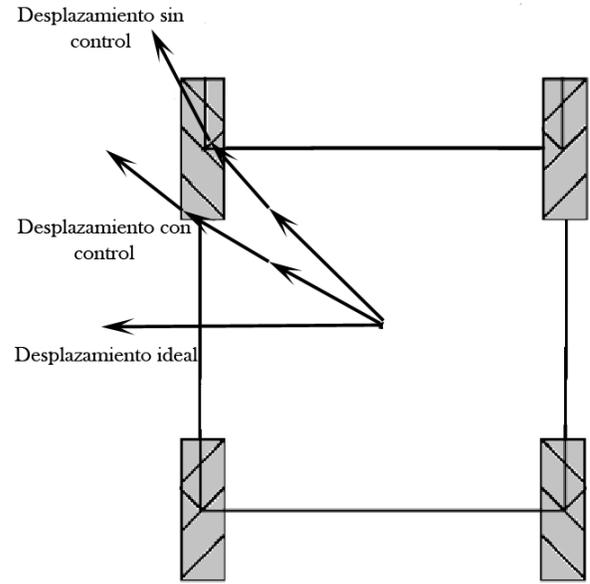


Figura. 34. Prueba de desplazamiento hacia la izquierda.

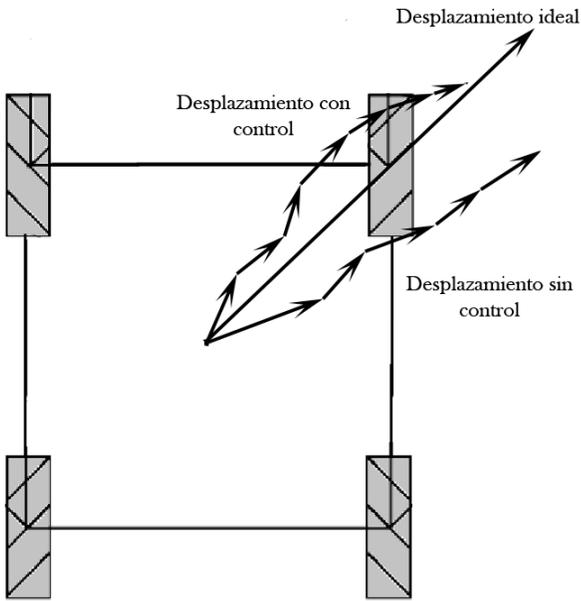


Figura. 35. Prueba de desplazamiento hacia diagonal superior derecha.

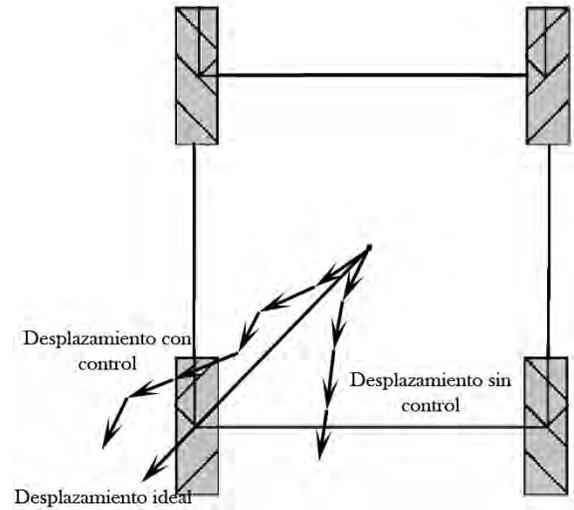


Figura. 36. Prueba de desplazamiento hacia diagonal inferior izquierda.

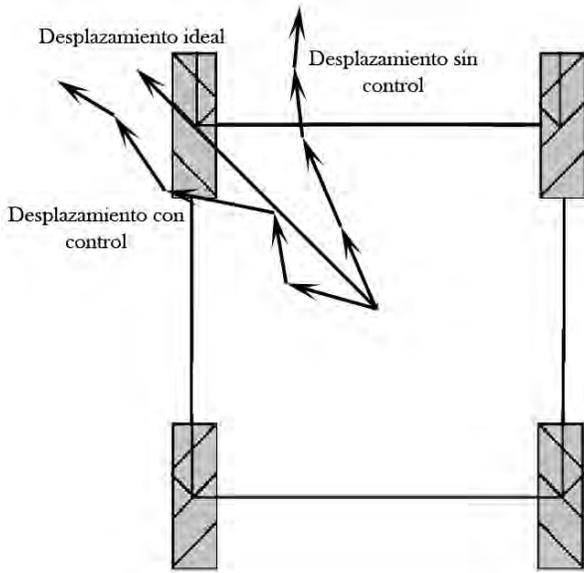


Figura. 37. Prueba de desplazamiento hacia diagonal superior izquierda.

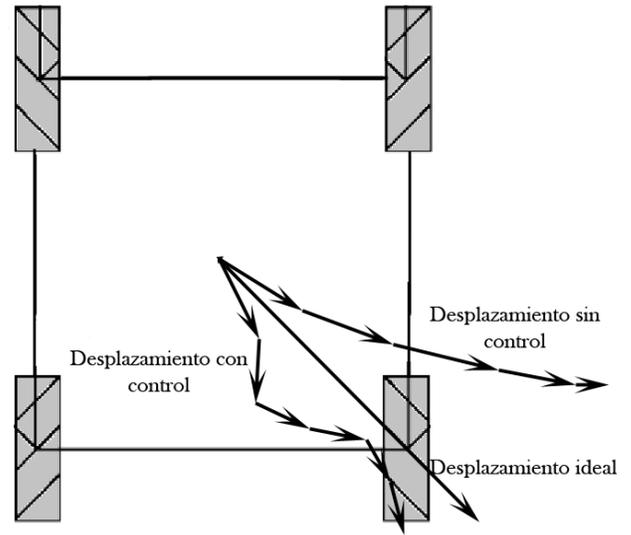


Figura. 38. Prueba de desplazamiento hacia diagonal inferior derecha.

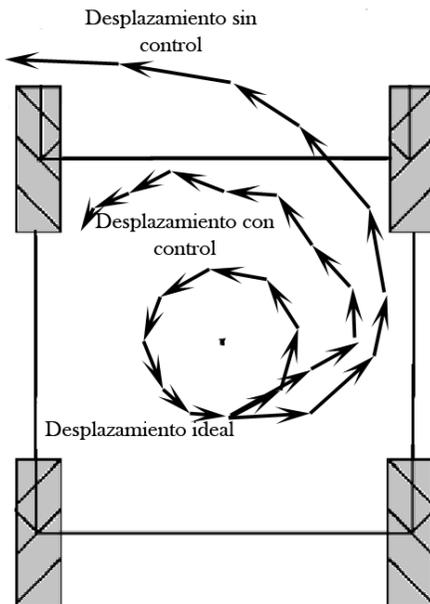


Figura. 39. Prueba de desplazamiento giro izquierda.

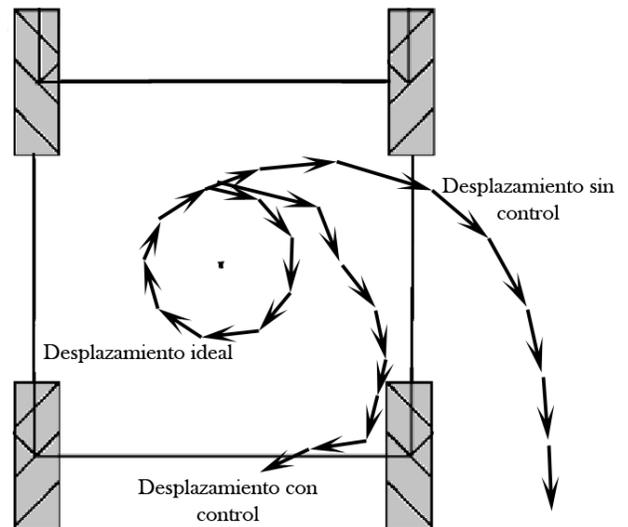


Figura. 40. Prueba de desplazamiento giro izquierda.

Al seguirse presentado una desviación considerable en el desplazamiento se usaron los datos obtenidos en las tablas 13 y 14 para la compensación de los errores en los movimientos hacia atrás y hacia adelante los datos obtenidos aseguraron un desplazamiento uniforme de la plataforma en estas dos direcciones alcanzando un rango de desviación menor mostrado en la figura 41. Lo realizado para los movimientos hacia adelante y hacia atrás es lo mismo que se pretende lograr para

las ocho direcciones faltantes que la plataforma contempla en su gama de movimientos. Sin embargo, el proceso para determinar los parámetros requeridos sería muy tardado por lo que se propuso el análisis de las desviaciones.

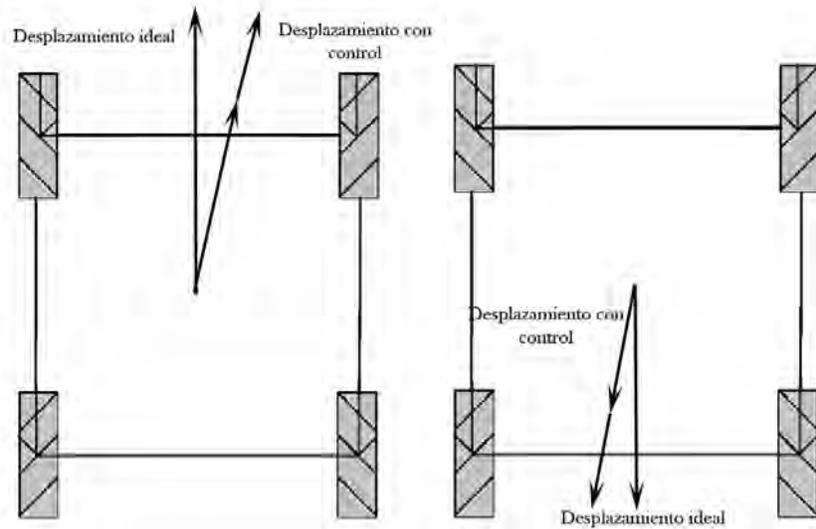


Figura. 41. Ajuste realizado al control direcciones adelante y atrás

Las pruebas finales realizadas en la plataforma se hicieron usando el análisis de las desviaciones y los nuevos valores de la K_p calculados. Los resultados mostraron que el control empleado para esta etapa de experimentación tuvo una mejora considerable con respecto a los resultados obtenidos en un principio y a pesar de que en estos aún se presenta una pequeña desviación en el desplazamiento de la plataforma el comportamiento mostrado se considera aceptable y deja a la plataforma en las condiciones básicas para operar en aplicaciones reales como su uso para robot de servicio.

Las siguientes figuras muestran la comparativa entre el desplazamiento ideal, el real con los parámetros de la primera etapa del control y el real con los parámetros de esta nueva etapa de control para los diferentes tipos de desplazamiento. En ellas se puede apreciar como el control actual compensa y corrige aún más el desplazamiento mostrado en comparación con la primera etapa manteniendo un margen de desviación y un comportamiento aceptable para cada caso.

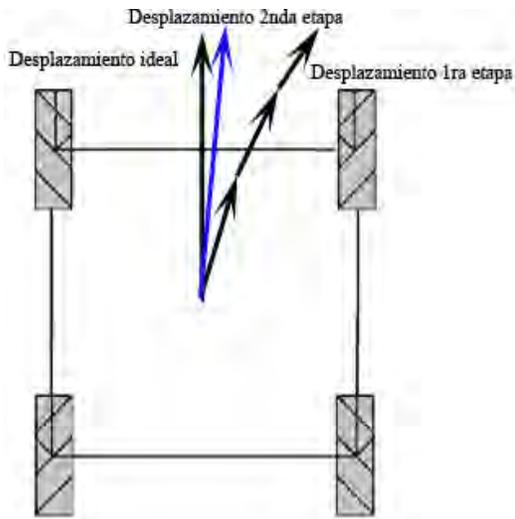


Figura. 42. Segunda prueba de desplazamiento hacia adelante.

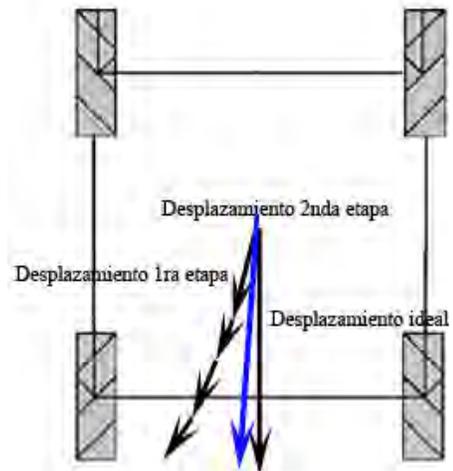


Figura. 43. Segunda prueba de desplazamiento hacia atrás.

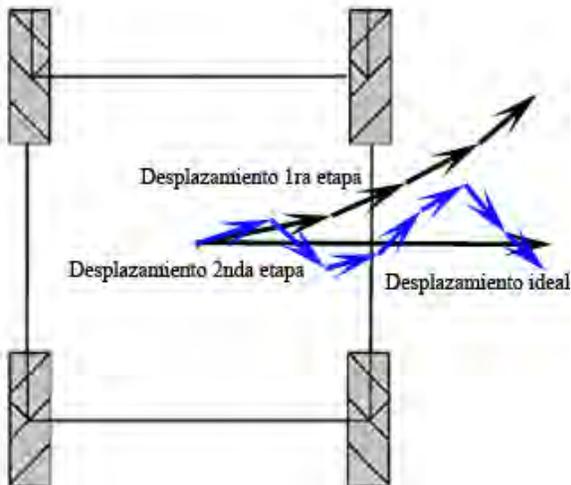


Figura. 44. Segunda prueba de desplazamiento hacia la derecha.

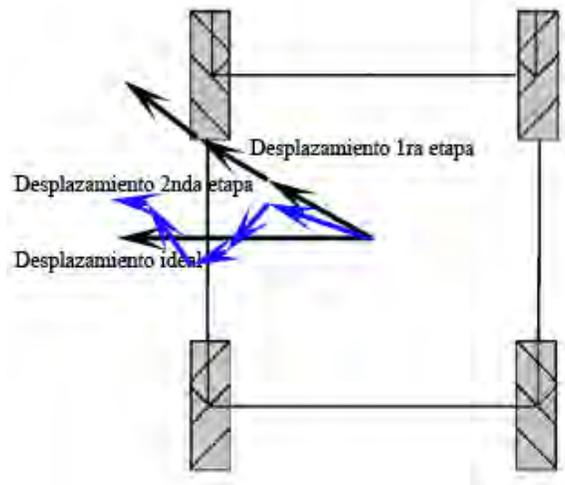


Figura. 45. Segunda prueba de desplazamiento hacia la izquierda.

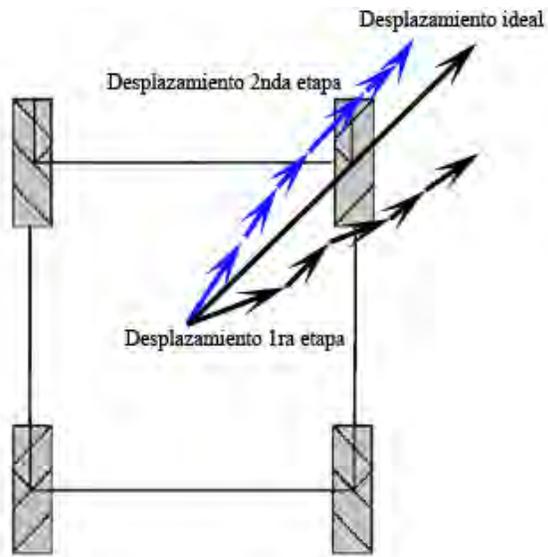


Figura. 46. Segunda prueba de desplazamiento hacia diagonal superior derecha.

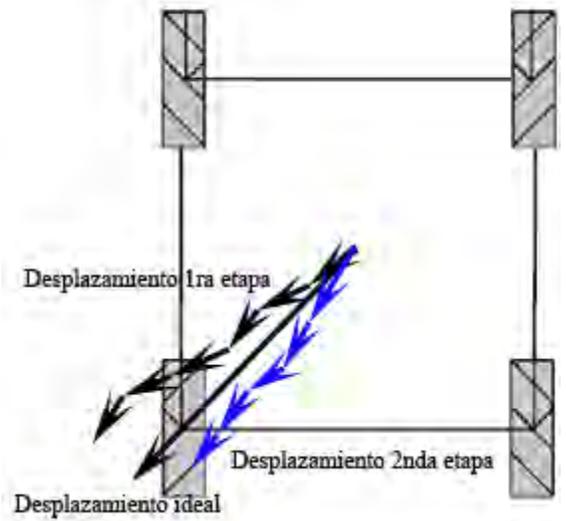


Figura. 47. Segunda prueba de desplazamiento hacia diagonal inferior izquierda.

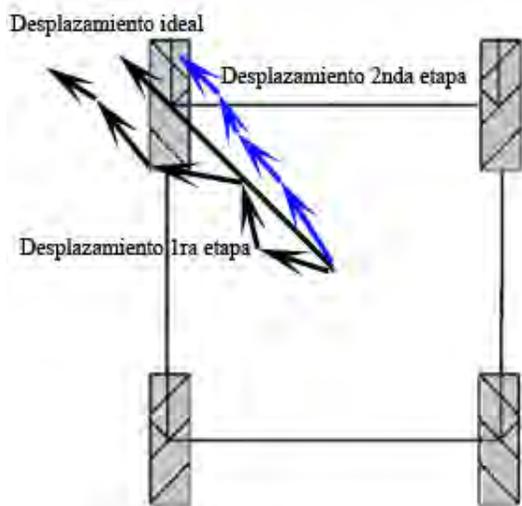


Figura. 48. Segunda prueba de desplazamiento hacia diagonal superior izquierda.

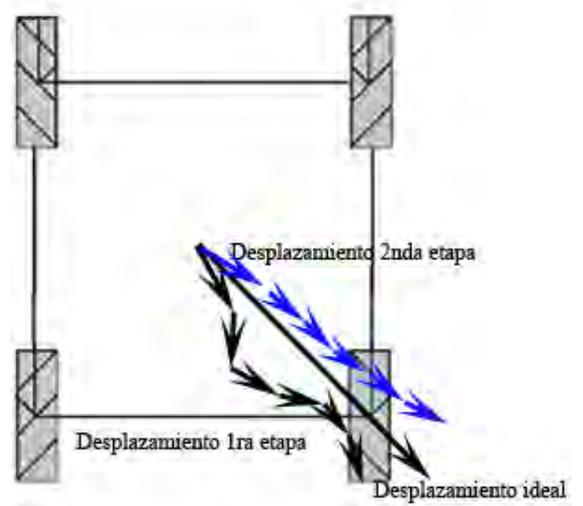


Figura. 49. Segunda prueba de desplazamiento hacia diagonal inferior derecha.

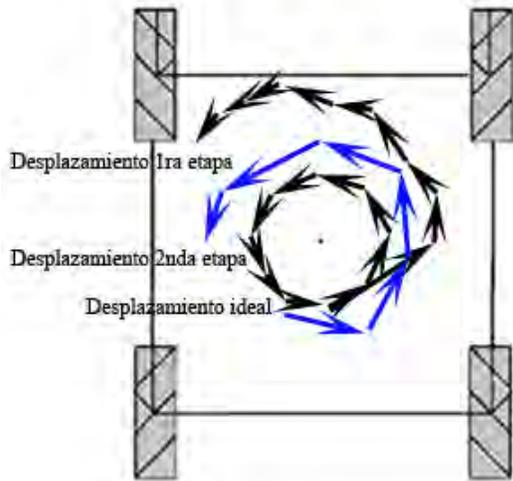


Figura. 50. Segunda prueba de desplazamiento giro izquierda.

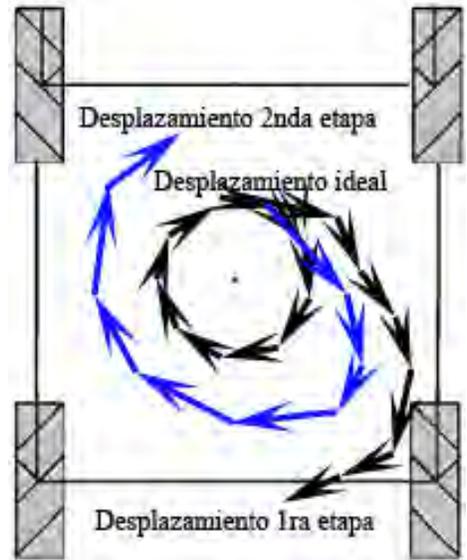


Figura. 51. Segunda prueba de desplazamiento giro izquierda.

Conclusiones

Se consigue estudiar un robot móvil con configuración mecánica omnidireccional entendiendo el cómo funciona, cuales son los elementos que interactúan en su desplazamiento y la elaboración de un algoritmo de control que permite su desplazamiento en todas las direcciones posibles dentro de la gama de movimientos para este tipo de robots móviles permitiendo su suave desplazamiento.

Se diseña e implementa un sistema de control para la plataforma omnidireccional de cuatro ruedas tipo Mecanum en configuración AB basado en el método de la curva de reacción de Ziegler y Nichols cumpliendo con el segundo de los objetivos particulares propuestos. El primer objetivo particular quedó alcanzado al utilizar los sensores ópticos de cada rueda como referencia para el controlador utilizando la señal de salida de cada uno de los encoders integrados en el mecanismo de las ruedas. La señal recibida por los sensores es traducida en términos de la velocidad angular y son dichos sensores los que otorgan al controlador la información necesaria para poder desempeñar su función y ejecutar las instrucciones para desplazar la plataforma.

En las pruebas realizadas experimentalmente se observa que la primera iteración del control no ofrece los resultados ideales para el desplazamiento en cada dirección y se sigue manteniendo una diferencia notable entre las velocidades angulares de cada rueda provocando que la plataforma se desvíe de su trayectoria alejándola del comportamiento deseado del vehículo con configuración mecánica omnidireccional, lo que comprueba nuestra hipótesis: el desplazamiento uniforme de la plataforma depende de la velocidad angular de sus ruedas. El error presentado en el desplazamiento tiene dos posibles orígenes: el primero se debe a que en los ensayos de caracterización los datos obtenidos tuviesen cierto nivel de ruido no considerado durante el proceso de diseño del controlador o como segunda instancia deberse a la mecánica misma de la plataforma que interfiere con el funcionamiento y es esta la que genera ruido en la retroalimentación del sistema. Las desviaciones que se presentan durante esta primera etapa de experimentación muestran que aún se requiere de un ajuste en los parámetros del control para que este pueda ejecutar las instrucciones programadas de una manera que acerque a la plataforma a su desplazamiento ideal. Esto lleva a analizar las desviaciones con mayor detalle realizando el proceso de cálculo de las velocidades lineales sobre los ejes X y Y así como a la determinación de la velocidad angular propia de la plataforma producida por la variación de la velocidad angular de cada una de las ruedas y que es uno de los factores que produce la desviación en nuestro sistema. Este nuevo análisis arroja como resultado cuales de los motores y en qué proporción deben ser modificados para que la desviación inicial de la plataforma se reduzca en mayor proporción.

La segunda iteración del control ofrece resultados más favorables corrigiendo aún más las desviaciones de la plataforma y acercándola a su desplazamiento ideal al mantener los valores de las velocidades angulares dentro de un margen de error menor al de la primera etapa de experimentación. Con base en este análisis se consigue observar analíticamente como la velocidad angular de cada rueda influye en la dirección final de la plataforma lo que refuerza nuestra hipótesis inicial y es a través de una segunda etapa de experimentación, donde se alcanza a reducir las desviaciones en mayor proporción acercándose al desplazamiento ideal. Sin embargo, aún se aprecia una ligera desviación para algunos de los desplazamientos y un movimiento sinuoso para los desplazamientos laterales.

Es a través de las dos etapas de experimentación realizadas y los resultados presentados en cada una de ellas que se alcanza el tercer objetivo particular al implementar el control para el desplazamiento de la plataforma diseñado a través del método de sintonización propuesto y verificando la eficiencia del mismo sobre nuestro sistema.

Debido a la complejidad presentada durante el análisis, el tiempo requerido para el proceso de experimentación y el ajuste de las dos iteraciones del control no fue posible que se desarrollaran de una manera formal las pruebas de análisis con diferentes velocidades de referencia y sus respectivos ajustes de desplazamiento para cumplir con el cuarto de los objetivos particulares, lo que lleva a dejar este proceso de la investigación como estudios a realizarse a futuro contemplando la información sobre lo ya descrito.

De lo previamente descrito se concluye, que nuestro sistema es viable para ser utilizado en aplicaciones reales dentro o fuera del laboratorio y se ha alcanzado la mayor parte de objetivos propuestos en este trabajo al dotar con un desplazamiento suave a la plataforma implementando un sistema de control en lazo cerrado usando una de las metodologías propuestas, el cual permite la manipulación de la velocidad angular de cada rueda.

Bibliografía

- [1] S.a. (s.f). *Introducción a la robótica móvil autónoma*. Enero 15, 2016, de RobotLab Sitio web: <http://www.cursosrobotlab.com/robotica/>
- [2] Bermúdez, G. (2002). ROBOTS MÓVILES.TEORIA, APLICACIONES Y EXPERIENCIAS. *TECNURA10*. Colombia.
- [3] Rojas R. (2005). *Omnidirectional control*. In *Robotik*, Alemania: Freie Universität Berlin,
- [4] S.a. (s.f.). *Sistemas de Locomoción de robots móviles*. Abril 5, 2016, Sitio web: http://www.esi2.us.es/~vivas/ayr2iaei/LOC_MOV.pdf
- [5] Martínez, S., Sisto, R. (2009). *Control y comportamiento de robots omnidireccionales*. (Tesis de licenciatura). Universidad de la República, Uruguay.
- [6] Wada, M., Mori, S. (1996). Holonomic and omnidirectional vehicle with conventional tires. *International Conference on Robotics and Automation*, 4, 3671-3676.
- [7] Cock, L. (2015). Diseño de una arquitectura de control para vehículos eléctricos omnidireccionales. (Tesis de maestría). Universidad EAFIT, Colombia.
- [8] Diegel, O., et al. (2002). Improved Mecanum Wheel Design for Omnidirectional Robots. *Australasian Conference on Robotics and Automation*. Auckland, ARAA: 117-121.
- [9] S.a. (s.f.). *Una manera nueva de entender la movilidad*. Febrero 4, 2017, de esclatec Sitio web: <http://esclatec.com/productos/silla-omnidireccional/>
- [10] S.a. (2016). *KMP omniMove*. Febrero 4, 2017, de KUKA Sitio web: http://www.kuka-robotics.com/mexico/es/products/mobility/mobile_platform_omnimove/start.htm
- [11] Larson H. (1992). *Introducción a la teoría de probabilidades e inferencia estadística*, México: LIMUSA. **Cita en el texto:** (Larson, 1992, p. 465-534).
- [12] Ogata, K. (2010). *Ingeniería de control moderna*. Madrid: Pearson educación, S.A.. Cita en el texto: (Ogata, 2010, p. 2).
- [13] Alfaro, V. (2002). Métodos de sintonización de controladores PID que operan como reguladores. *Ingeniería* 12, 21-36. **Cita en el texto:** (Alfaro, 2002, p.21).

- [14] Fernández, F. (2015). Uso y determinación de los parámetros de un controlador PID mediante el métodos de “Ensayo y Error”, Ziegler–Nichols y Cohen–Coon. **Cita en el texto:** (Fernández, 2015, p.6).
- [15] Alfaro, V. (2002). Métodos de sintonización de controladores PID que operan como reguladores. *Ingeniería* 12, 21-36.
- [16] Calvo, J. et al. (2006). Ajuste empírico de reguladores PID en cadena cerrada con Matlab. *Técnica Industrial* 261, 42-46. **Cita en el texto:** (Calvo, 2006, p. 43).
- [17] Blanco, J. (2011). Automatización y control regulatorio de una columna de destilación extractiva a nivel planta piloto para la producción de etanol anhidro. (Tesis de maestría). Universidad Nacional de Colombia Facultad de Ingeniería, Bogota.
- [18] Thomas, R. (2011). *Omni-Directional Mobile Platform for the Transportation of Heavy Objectcs.* (Tesis de maestría). Massey University, Nueva Zelanda.
- [19] West M., Asada H. (1992). *In Desing of a Holonomic Omnidirectional Vehicle*, (Vol. 3, Pp. 97-103). Francia: International conference on robotics and automation.
- [20] S.a. (s.f.). *Sensores de efecto Hall.* Enero 15, 2016, de Universidad Nacional de San Luis Sitio web: <http://www0.unsl.edu.ar/~interfases/labs/lab09.pdf>
- [21] S.a. (s.f.). *Sensores ultrasónicos.* Enero 15, 2016, de Sensorstecnic & semiconductors Sitio web: <http://www.sensorstecnic.net/es/productos/category/105/sensores-y-transmisores/sensores-ultrasonicos>
- [22] S.a. (febrero 3, 2012). *ARDUINO + BLUETOOTH.* Mayo 3, 2016, de DIYmakers Sitio web: <http://diymakers.es/arduino-bluetooth/>
- [23] Navarro, K. (2014). *Sensor Ultrasónico HC-SR04 + Arduino.* Marzo 3, 2016, de Panama Hitek Sitio web: <https://panamahitek.com/sensor-ultrasonico-hc-sr04-arduino/>
- [24] Fierro, R., et al. (2016). Simulación y ensamblaje de un prototipo para control y navegación de un robot móvil. *Jornadas de Ingeniería Eléctrica y Electrónica (FIEE).*
- [25] Ramírez, G. (2013). Método de aprendizaje simple para navegación de minirobots móviles rodantes. *DYNA* 70 (138), 59-66.

- [26] Bastidas, L. (2014). *Control pid de velocidad de un motor dc*. Abril 5, 2016, de Prezi Sitio web: <https://prezi.com/hhmoedi5drfq/control-pid-de-velocidad-de-un-motor-dc/>
- [27] S.a. (2013). *Método de Ziegler-Nichols*. Abril 5, 2016, de Picuino Sitio web: <https://sites.google.com/site/picuino/ziegler-nichols>
- [28] González, S. (2013). *Control PID con sintonización difusa y control difuso*. (Tesis de maestría). Instituto Politécnico Nacional, Mexico.
- [29] Hernández, I. (2016). *Manipulador móvil omnidireccional, su coordinación de movimientos en ambientes inteligentes*. (Tesis de licenciatura). Universidad Nacional Autónoma de México, México.
- [30] Peña M., Andrade J., et al. (2016). Omnidirectional Platform Control and Displacement Characterization. *IEEE ICA/ACCA 2016*. Curicó, Chile, 19-21.

Índice de imágenes

Figura. 1. Banco de pruebas.	9
Figura. 2. Desfase de señales.	11
Figura. 3. a) Ruedas: usadas en superficies lisas. b) Orugas: empleadas sobre superficies irregulares.	12
Figura. 4. (a) En rueda convencional simple, (b) En rueda convencional de desplazamiento lateral, (c) En rueda convencional con desviación delantera [6].	14
Figura. 5. Ruedas tipo Mecanum [5].	15
Figura. 6. a) Tipo “AB”, b) Tipo “BB” [8].	16
Figura. 7. Plataforma de carga pesada de la empresa KUKA [9].	16
Figura. 8. Silla omnidireccional [10].	17
Figura. 9. Plataforma omnidireccional a caracterizar.	18
Figura. 10. Motor de 12 [V] y Talon SR.	18
Figura. 11. Arduino MEGA.	20
Figura. 12. Sensor E4P OEM.	20
Figura. 13. Esquema del circuito.	21
Figura. 14. Variación de pulso con respecto al voltaje.	23
Figura. 15. Esquema del circuito con encoder.	23
Figura. 16. Rango de velocidades según el voltaje	24
Figura. 17. Rango de operación de los motores.	25
Figura. 18. Pulsos del encoder y su rotación correspondiente.	26
Figura. 19. Velocidades angulares de las rueda A y B.	27
Figura. 20. Velocidades angulares de las rueda B y D.	28
Figura. 21. Curva general de una respuesta en S.	31
Figura. 22. Respuestas del motores A con los valores K, L y τ	33
Figura. 23. Diagrama de bloques del sistema de una planta con un control y retroalimentación	36
Figura. 24. Diagrama general del sistema con control por motor.	37
Figura. 25. Acción proporcional.	38
Figura. 26. Acción integral.	39
Figura. 27. Acción derivativa.	39
Figura. 28. Circuito completo del controlador.	40
Figura. 29. Plataforma con las velocidades tangenciales, basado en [7].	48
Figura. 30. Ecuaciones del control PID.	54
Figura. 31. Prueba de desplazamiento hacia adelante.	56
Figura. 32. Prueba de desplazamiento hacia atrás.	56
Figura. 33. Prueba de desplazamiento hacia la derecha.	57
Figura. 34. Prueba de desplazamiento hacia la izquierda.	57

Figura. 35. Prueba de desplazamiento hacia diagonal superior derecha.	57
Figura. 36. Prueba de desplazamiento hacia diagonal inferior izquierda.	57
Figura. 37. Prueba de desplazamiento hacia diagonal superior izquierda.	58
Figura. 38. Prueba de desplazamiento hacia diagonal inferior derecha.	58
Figura. 39. Prueba de desplazamiento giro izquierda.	58
Figura. 40. Prueba de desplazamiento giro izquierda.	58
Figura. 41. Ajuste realizado al control direcciones adelante y atrás	59
Figura. 42. Segunda prueba de desplazamiento hacia adelante.	60
Figura. 43. Segunda prueba de desplazamiento hacia atrás.	60
Figura. 44. Segunda prueba de desplazamiento hacia la derecha.	60
Figura. 45. Segunda prueba de desplazamiento hacia la izquierda.	60
Figura. 46. Segunda prueba de desplazamiento hacia diagonal superior derecha.	61
Figura. 47. Segunda prueba de desplazamiento hacia diagonal inferior izquierda.	61
Figura. 48. Segunda prueba de desplazamiento hacia diagonal superior izquierda.	61
Figura. 49. Segunda prueba de desplazamiento hacia diagonal inferior derecha.	61
Figura. 50. Segunda prueba de desplazamiento giro izquierda.	62
Figura. 51. Segunda prueba de desplazamiento giro izquierda.	62

Índice de tablas

Tabla 1. Tabla de combinaciones de giro.....	10
Tabla 2. Sentido de giro.	11
Tabla 3. Ventajas y desventajas de robots móviles comunes. Basado en [4].....	13
Tabla 4. Rango de acción según voltaje de entrada.	21
Tabla 5. Variación de pulso según voltaje.	22
Tabla 6. Velocidades angulares según voltaje y ancho de pulso.....	24
Tabla 7. Método Ziegler y Nichols con tiempos de integración y derivación.....	32
Tabla 8. Método Ziegler y Nichols con constantes PID.....	32
Tabla 9. Parámetros obtenidos en las pruebas.....	33
Tabla 10. Parámetros del controlador para los motores.	35
Tabla 11. Funciones de los controladores PID.....	36
Tabla 12. Funciones de transferencia generales.....	37
Tabla 13. Valores de los parámetros para el movimiento hacia adelante.	46
Tabla 14. Valores de los parámetros para el movimiento hacia atrás.	47
Tabla 15. Constantes K_p calculadas.....	55

Anexo uno: Código Arduino

El código de programación básicamente estará dividido en cuatro grandes categorías: la definición de las variables, la inicialización de puertos de salida y entrada, el ciclo principal a ejecutarse con las instrucciones requeridas y una serie de funciones que realizarán ciertas tareas repetitivas de una manera más ordenada, clara e intuitiva. Las funciones permitirán crear segmentos modulares de código, de forma que se puedan realizar ciertas rutinas y retornar al área de código desde donde se realizó la llamada a dicha función.

Declaración de variables.

En esta parte de código se deben definir el tipo y la cantidad de variables que se van a utilizar a lo largo del proceso, estas variables serán globales y podrán usarse tanto en el ciclo principal como en las distintas funciones que se creen dentro del código. En la siguiente parte del código pueden verse las variables a usarse.

//Variables destinadas a medir el tiempo necesario para ejecutar las instrucciones de control

volatile unsigned long tiempoAnterior;

unsigned long tiempoActual;

volatile double dT;

double limite=750;

long duracion;

int freno=75; // Variable que determina el tiempo de frenado de la plataforma

//Variables que representan el valor inicial de las constantes del control PID para cada motor

double kpA=0;

double kiA=0;

double kdA=0;

double kpB=0;

double kiB=0;

double kdB=0;

double kpC=0;

double kiC=0;

double kdC=0;

double kpD=0;

```

double kiD=0;
double kdD=0;
double ref=2.92; //Valor de pulsos de referencia para el control
//En esta sección se define las variables que guardarán los errores usados en el control PID
double errorA=0;
double errorAntA=0;
double errorSumaA = 0;
double errorDerA=0;
double errorB=0;
double errorAntB=0;
double errorSumaB = 0;
double errorDerB=0;
double errorC=0;
double errorAntC=0;
double errorSumaC = 0;
double errorDerC=0;
double errorD=0;
double errorAntD=0;
double errorSumaD = 0;
double errorDerD=0;
//Variables de salida del control
double controlA=198;
double controlB=198;
double controlC=198;
double controlD=198;
int val=765; //Valor analógico para motores detenidos
//Pines de conexión para los pwm mandados a los Talones SR
int pwmA = 6;
int pwmB = 3;

```

```
int pwmC = 2;
int pwmD = 5;
//Variables que guardan el resultado previo a la salida del control
double UA=0;
double UB=0;
double UC=0;
double UD=0;
//Variables usadas para la medición de pulsos de los encoders en las ruedas
boolean anteriorA = 0;
boolean actualA = 0;
double contadorA = 0;
double kountA=0;
boolean anteriorB = 0;
boolean actualB = 0;
double contadorB = 0;
double kountB=0;
boolean anteriorC = 0;
boolean actualC = 0;
double contadorC = 0;
double kountC=0;
boolean anteriorD = 0;
boolean actualD = 0;
double contadorD = 0;
double kountD=0;
//Pines de conexión para las salidas del canal A de los cuatro encoders
int pinSensorA = 51;
int pinSensorB = 49;
int pinSensorC = 47;
int pinSensorD = 45;
```

```

//Valores que definen la referencia para el frenado de las ruedas
double valA= 0;
double valB= 0;
double valC= 0;
double valD= 0;
int incomingByte = 0; //Valor que determina la función de dirección a seguir por la plataforma
int secuencia = 0; //Valor que determina las instrucciones a seguir para el control

```

La siguiente sección del código representa la inicialización de pines y variables para el estado inicial del proceso.

```

void setup() {
Serial.begin(9600); // Establece la velocidad de datos del puerto serie
tiempoAnterior=0; // Definimos el tiempo anterior en 0 [s]
// Definimos las entradas
pinMode(pinSensorA,INPUT);
pinMode(pinSensorB,INPUT);
pinMode(pinSensorC,INPUT);
pinMode(pinSensorD,INPUT);
// Definimos las salidas
pinMode(pwmA, OUTPUT);
pinMode(pwmB, OUTPUT);
pinMode(pwmC, OUTPUT);
pinMode(pwmD, OUTPUT);
//Iniciamos los motores en un estado de alto total
analogWrite(pwmA, val/4);
analogWrite(pwmB, val/4);
analogWrite(pwmC, val/4);
analogWrite(pwmD, val/4);
delay(2000); //Espera de 2[s] antes de iniciar el código (se hará sólo al encender el Arduino)
}

```

Funciones.

Antes de comenzar a explicar el ciclo principal de trabajo es importante describir algunas de las funciones que se han creado para realizarlo de una manera más ordenada y una vez entendiendo qué hace cada una de ellas pasar al código que se realizará hasta apagar el sistema.

boolean debounceA: función que lee el estado del encoder de la rueda A, y devuelve un valor para ser usado dentro del ciclo principal. De forma análoga, se realiza una función para las otras ruedas.

```
boolean debounceA(boolean dato_antriorA)
{
    boolean dato_actualA = digitalRead(pinSensorA);
    if (dato_antriorA != dato_actualA)
    {
        delay(10);
        dato_actualA = digitalRead(pinSensorA);
    }
    return dato_actualA;
}
```

KPIDS: es el proceso en el que de acuerdo al valor de la secuencia se le asignarán diferentes valores a las constantes del control que permitan su desplazamiento de manera uniforme y que es parte complementaria de la función *Controlador*. En el ejemplo mostrado sólo se visualiza el caso 1 que corresponde a la dirección de adelante, sin embargo, en la programación existe un caso para cada dirección.

```
void KPIDS()
{
    switch(secuencia)
    {
        case 1: //ADELANTE;
            kpA=1.8;
            kiA=0.01;
            kdA=0.005;
```

```

    kpB=1.6;
    kiB=0.01;
    kdB=0.005;
    kpC=1.6;
    kiC=0.01;
    kdC=0.005;
    kpD=1.8;
    kiD=0.01;
    kdD=0.005;
    break;
}
}

```

ADELANTE: en esta sección del código se realiza la acción de acelerar las ruedas hasta que estas alcancen un valor deseado de referencia. El valor propuesto para nuestro caso es el de 790, el cual representa la referencia analógica de nuestro sistema para la velocidad angular deseada y mientras el valor de la variable sea menor a este las ruedas seguirán aumentando su aceleración. Existe una función similar para cada dirección.

```

void ADELANTE()
{
    while(val<790) //proceso de aceleración
    {
        val++;
        delay(100);
        analogWrite(pwmA, (765-(val-764))/4);
        analogWrite(pwmB, val/4);
        analogWrite(pwmC, val/4);
        analogWrite(pwmD, val/4);
    }
    secuencia = 1; //Definimos que Adelante es la secuencia 1
}

```

```
/*Reiniciamos las variables del control en caso de tener algún valor por realizar una secuencia previa*/
```

```
errorA=0;  
errorAntA=0;  
errorSumaA = 0;  
errorDerA=0;  
errorB=0;  
errorAntB=0;  
errorSumaB = 0;  
errorDerB=0;  
errorC=0;  
errorAntC=0;  
errorSumaC = 0;  
errorDerC=0;  
errorD=0;  
errorAntD=0;  
errorSumaD = 0;  
errorDerD=0;
```

```
}
```

Controlador: se puede considerar como la función más importante, ya que en ella se realizan todos los cálculos necesarios para el control de la velocidad angular de cada rueda dependiendo la dirección en la que la plataforma se desplace y se encarga de que los valores del pwm siempre se encuentren en el rango de operación requerido para que las ruedas no cambien su sentido de giro de una manera sorpresiva o que excedan un límite de velocidad establecido. Para efectos de ejemplificación sólo se describirá el caso del motor A y una dirección. Sin embargo, en el código se contempla cada una de las direcciones y el control correspondiente para cada rueda.

```
void controlador(){
```

```
    KPIDS(); //Determinamos los valores de las constantes del control a usar
```

```
    kountA=((0.73)*contadorA); /*Definimos el valor actual del control con base al contador del encoder A*/
```

```

errorA=ref-kountA; //Calculamos el error proporcional
errorDerA=(errorA-errorAntA)/(limite/1000); //Calculamos el error derivativo
errorSumaA += (kiA*errorA); //Calculamos el error integral
UA=kpA*errorA+(errorSumaA)+kdA*errorDerA; //Realizamos el cálculo de la señal de control
controlA=UA; // Asignamos a una variable la señal de control calculada
//Comparamos la señal de control entre los límites inferior y superior permitidos
if ((controlA)>235){
    controlA=235;
}
if ((controlA)<191){
    controlA=191;
}
//De acuerdo a la dirección deseada se determinarán las instrucciones a seguir.
switch(secuencia)
{
case 1:
    analogWrite(pwmA,(765-((floor(controlA)*4)-764))/4);
    analogWrite(pwmB,floor(controlB));
    analogWrite(pwmC,floor(controlC));
    analogWrite(pwmD,floor(controlD));
    break;
    /*El comando floor nos redondea el resultado del control para dejarlo en entero
}
errorAntA=errorA; //Definimos el error anterior como el error actual
tiempoAnterior=tiempoActual; //Declaramos que el tiempo anterior será el tiempo actual
contadorA=0; //Reiniciamos el contador del encoder A
}

```

Alto: en esta función, dependiendo la dirección anterior se selecciona el proceso necesario para el frenado de la plataforma, en el código de ejemplo se muestra las instrucciones necesarias para la dirección de adelante.

```
void alto(){
    valA= floor(controlA)*4;
    valB= floor(controlB)*4;
    valC= floor(controlC)*4;
    valD= floor(controlD)*4;
    switch(secuencia)
    {
        case 1:
            while(valA>765||valB>765||valC>765||valD>765)
                {
                    if(valA>765) {valA--;}
                    if(valB>765) {valB--;}
                    if(valC>765) {valC--;}
                    if(valD>765) {valD--;}
                    delay(freno);
                    analogWrite(pwmA, (765-(valA-764))/4);
                    analogWrite(pwmB, valB/4);
                    analogWrite(pwmC, valC/4);
                    analogWrite(pwmD, valD/4);
                }
            break;
    }
}
```

Imprimir: esta función se creó para imprimir sólo cuando sea deseado y valores que uno elija. De forma predeterminada, la función imprimir manda al puerto serie del Arduino manda los resultados obtenidos en el cálculo de la salida de control y la comparación entre esta y la referencia del sistema. Se muestra el caso de la rueda A

```

void imprimir(){
    Serial.print("controlA: ");
    Serial.print(floor(controlA));
    Serial.print(' ');
    Serial.print("referencia: ");
    Serial.print(ref);
    Serial.print(' ');
    Serial.print("errorA: ");
    Serial.print(errorA);
    Serial.println(' ');
}

```

Ciclo principal: esta parte del código es la que se repetirá hasta apagar el sistema, en ella se realiza la selección de la dirección a seguir, los llamados a las funciones anteriores y los procesos necesarios de medición para los encoders.

```

void loop() {
    if (Serial.available() > 0) //Esperamos recibir un byte de entrada por el puerto serial
    {
        incomingByte = Serial.read(); // lee el byte de entrada en sistema ascii
    }
    switch(incomingByte) // seleccionamos la dirección a seguir dependiendo la entrada
    {
        case 48:// Correspondiente al 0 decimal
            alto();
            Serial.println("Adelante");
            ADELANTE();
            break;
        /*Se realiza el mismo proceso hasta el 57 (0-9 decimal) para todas las direcciones
        case 83:// Corresponde a la tecla s
            alto();

```

```

    secuencia=0; //Reiniciamos el valor de la secuencia esperando una nueva dirección
    break;
case 115:// Corresponde a la tecla s minúscula
    alto();
    secuencia=0; //Reiniciamos el valor de la secuencia esperando una nueva dirección
    break;
}
incomingByte=0; //Volvemos a cero el valor del puerto serial

//LECTURA DEL ENCODER A
actualA = debounceA(anteriorA);
if ( anteriorA == 0 && actualA == 1)
{
    contadorA++;
    delay (100);
    //Serial.println(contador);
}
anteriorA = actualA;
/*Repetimos el mismo proceso para los demás encoders*/
tiempoActual = millis(); //Medimos el tiempo actual del proceso
dT=(double)(tiempoActual - tiempoAnterior) ; //Comparamos el tiempo actual con el anterior
if(dT>=limite){
    controlador(); // Realizamos el proceso de control
}
}
}

```