

UNIVERSIDAD POLITÉCNICA DE PUEBLA

Programa Académico de Maestría en Ingeniería



“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MÓVIL DIFERENCIAL  
PARA EXPERIMENTACIÓN”

Tesis de Grado de Maestría

Juan Ramón Quezada Paredes

Juan C. Bonilla, Puebla

Enero 2013

UNIVERSIDAD POLITÉCNICA DE PUEBLA

Programa Académico de Maestría en Ingeniería



“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MÓVIL DIFERENCIAL  
PARA EXPERIMENTACIÓN”

Tesis de Grado de Maestría

Juan Ramón Quezada Paredes

Comité Evaluador

Dr. Antonio Benítez Ruiz

Asesor

MC. Azgad Casiano Ramos

Sinodal

Dr. Jesús Manuel Muñoz Pacheco

Sinodal

Juan C. Bonilla, Puebla

Enero 2013

# I Índice

I	Índice .....	3
1	Planteamiento del problema de investigación .....	5
1.1	Introducción.....	5
1.2	Definición del problema.....	6
1.3	Objetivo general .....	7
1.4	Objetivos específicos .....	8
1.5	Justificación.....	8
1.6	Alcances y limitaciones .....	9
2	Marco teórico .....	10
2.1	Modelado cinemático .....	11
2.2	Modelado dinámico .....	16
3	Diseño y construcción del móvil .....	20
3.1	Sensores .....	20
3.2	Microcontroladores.....	24
3.3	Motores .....	26
3.4	Comunicación .....	27
3.5	Alimentación del robot móvil .....	29
3.6	Planos de construcción del móvil.....	30
3.6.1	Móvil con Arduino uno .....	30
3.6.2	Móvil con Arduino mega .....	36
3.7	Tarjeta electrónica del móvil.....	37
4	Control del móvil .....	41
4.1.	Modelado de un motor CD .....	41

4.2.	Identificación del Sistema .....	45
4.3.	Diseño del Controlador .....	51
4.3.1.	Controlador P .....	53
4.3.2.	Controlador I.....	54
4.3.3.	Controlador PD.....	55
4.3.4.	Controlador PI .....	56
4.3.5.	Controlador PID.....	57
5	Pruebas y resultados experimentales.....	60
5.1.	Implementación de los sensores.....	61
5.2.	Implementación del control PID .....	66
5.3.	Implementación de trayectorias .....	70
6	Conclusiones.....	76
7	Bibliografía.....	78
8	Apéndice A .....	81
9	Apéndice B .....	84
B.1.	Gráfica de los datos obtenidos de los motores .....	84
B.2.	Controlador Proporcional .....	84
B.3.	Controlador Integral .....	85
B.4.	Respuesta al escalón unitario del motor CD .....	86
B.5.	Controlador Proporcional Derivativo .....	86
B.6.	Controlador Proporcional Integral .....	87
B.7.	Controlador Proporcional, Integral y Derivativo.....	88

# 1 Planteamiento del problema de investigación

---

## 1.1 Introducción

La robótica permite integrar áreas de conocimiento como la matemática, la física, la mecánica, la electrónica y las ciencias de la computación. Específicamente la computación aporta técnicas y procesos para almacenar y procesar la información, con ayuda de algunos de los campos de las ciencias de la computación como Inteligencia artificial, visión, aprendizaje automático, procesamiento digital de imágenes se enriquece la robótica con técnicas para la operación de los robots [27].

La evolución de la robótica está relacionada con el uso y desarrollo de mecanismos electromecánicos, debido a las ventajas que ofrece en la automatización de tareas.

La investigación y desarrollo de los robots móviles ha crecido rápidamente; después de la creación del primer robot móvil llamado Newt (1977), siguieron los robots móviles Hilare (1979), Stanford (1979), Shakey (1984), Raibert (1986), Sojourner rover (1997), Dante II (1999) [1].

La robótica es un tema multidisciplinario que se puede dividir de acuerdo a su configuración: poliarticulados, móviles, andróides, zoomórficos e híbridos [3].

La característica común de los robots poliarticulados es la de ser sedentarios y mover sus elementos terminales en un determinado espacio de trabajo con uno o más sistemas de coordenadas y un número limitado de grados de libertad. Los robots móviles tienen una gran capacidad de desplazamiento, basados en plataformas dotadas de un sistema locomotor rodante. Los robots andróides intentan reproducir parcial o totalmente la forma y comportamiento cinemático de los humanos. Los robots zoomórficos se caracterizan por sus sistemas de locomoción que imitan a diversos

seres vivos excepto a los androides. Los híbridos son robots de difícil clasificación al utilizar características de las clasificaciones anteriores.

Un robot móvil se define como un sistema electromecánico capaz de desplazarse en forma autónoma de un punto a otro en un determinado espacio de trabajo [7].

El objetivo principal de la robótica móvil es el desarrollo de robots autónomos que puedan desenvolverse en ambientes dinámicos. En robótica móvil, se presentan tres problemas generales a resolver: la navegación, el reconocimiento del entorno o construcción de mapas de entorno y la auto-localización.

Este tipo de robots se clasifican de acuerdo al entorno donde interactúan como: robots móviles marítimos, robots móviles aéreos y robots móviles terrestres.

Los robots móviles terrestres se pueden clasificar por el tipo de locomoción que utilizan en: locomoción por ruedas, orugas o patas. Los robots terrestres con ruedas destacan por su eficiencia en el consumo de energía, son apropiados para superficies lisas y firmes; son menos complejos que los robots terrestres con patas y orugas.

Los robots móviles terrestres son un tema interesante debido al estudio multidisciplinario (física, matemática, electrónica, control y computación) necesario para su implementación; las aplicaciones de éstos son: exploración planetaria, minera, reconocimiento de terreno, inspección, vigilancia, asistencia médica, ocio, entretenimiento, investigación, desarrollo, industria metalmecánica, también se utilizan como plataformas móviles en aplicaciones que requieren la integración con otros tipos de robots como pueden ser los manipuladores.

## **1.2 Definición del problema**

Este trabajo se diseñó un conjunto de robots móviles en configuración diferencial; este conjunto de robots dispondrán de un sensor codificador, un sensor de ultrasonido, un sensor visual y un módulo inalámbrico, estos sensores recibirán la información, el móvil la procesara y realizara una o varias trayectorias en un ambiente controlado.

Existen diversas plataformas móviles existentes, por ejemplo RIO (National Instruments), MINDSTORMS NXT 2.0 (LEGO), SRV-1 (SURVEYOR), 3Pi (Pololu), etcétera.

Estas plataformas tienen diversas aplicaciones y costos todas tienen derechos de autor con lo cual no proporcionan toda la información de la realización de las funciones o los iconos ocupados, por ello se depende totalmente del software proporcionado por el fabricante del hardware si el software tiene limitadas prestaciones, se puede esperar una actualización del mismo o la creación de funciones que acepte el software de la plataforma con la respectiva dificultad que ello representa. Es obvio que se requiere el desembolso para la adquisición de una o varias plataformas las cuales no siempre cubren todos los requerimientos de los proyectos.

Una forma de comprobar algoritmos es la realización de simulaciones, aunque las simulaciones proveen un acercamiento al comportamiento de los robots es necesario la corroboración de las simulaciones con las implementaciones físicas.

Lo anterior con lleva complicaciones importantes de la adecuación de una simulación a una implementación, si no se tiene una plataforma a modo no será posible pasar de una simulación a una implementación, por ello se decidió crear un conjunto de robots móviles en configuración diferencial.

### **1.3 Objetivo general**

- Implementar un robot móvil diferencial para tareas de navegación y exploración.

## **1.4 Objetivos específicos**

- Diseñar una plataforma de robot móvil diferencial con un microcontrolador.
- Caracterizar dos sensores; un codificador y un ultrasonido, así como un módulo inalámbrico y la interfaz de una cámara.
- Implementar un algoritmo de control para el seguimiento de trayectorias.
- Implementar una rutina de navegación en un ambiente controlado evitando obstáculos.

## **1.5 Justificación**

Contar con una plataforma móvil experimental para evaluar diferentes algoritmos o técnicas que consideren: evasión de obstáculos, planeación de trayectorias, seguimiento de trayectorias, navegación tanto en ambientes estáticos como en dinámicos y exploración con ello haría posible la comparación de dichas técnicas, los convenientes e inconvenientes de cada una con la amplia aplicación didáctica que esto representa.

Este proyecto propone diseñar e implementar una plataforma económica para un robot móvil diferencial; hay plataformas de este tipo en el mercado, pero su interacción con el entorno es limitada al no disponer de sensores que nos ayuden a ubicarse, de lo anterior se decidió construir una plataforma que integre los elementos disponibles en la universidad y que maximicen las funcionalidades del móvil.



Un robot móvil diferencial puede servir de apoyo en dos carreras ofertadas por UPPue: la ingeniería en electrónica y de telecomunicaciones y la ingeniería mecatrónica. En la primera existen 5 materias relacionadas al tema y en la segunda 10 materias relacionadas al tema, por lo que su posible impacto como material didáctico es amplio y proporciona un soporte para proyectos más ambiciosos (móvil Arckeman, Segway, agregar un manipulador al móvil).

## **1.6 Alcances y limitaciones**

Los alcances esperados son:

1. Construir e instrumentar un robot móvil diferencial.
2. Caracterizar los sensores de ultrasonido SRF05 y odométrico Pololu enc01a, así como la interfaz para el Xbee 1mW y la cámara CMUcam3.
3. Utilizar el modelo dinámico de un robot móvil diferencial, el cual se integrará a la tarjeta de desarrollo Arduino para el proceso de locomoción.
4. Simular algunas leyes de control con los parámetros conocidos del robot móvil diferencial, así como trayectorias de desplazamiento.
5. Implementar un algoritmo de navegación en el robot móvil diferencial.
6. Probar diferentes estrategias de control lineal y no lineal e implementarlas en el sistema Arduino Uno y Arduino Mega.

# 2 Marco teórico

---

Existen diversas configuraciones para los robots móviles terrestres con ruedas [4], estas dependen principalmente de la aplicación, no obstante, de manera general se tienen las siguientes configuraciones de tracción: Ackerman, triciclo clásico, diferencial, skid steer, síncrona y omnidireccional Figura 2.1.

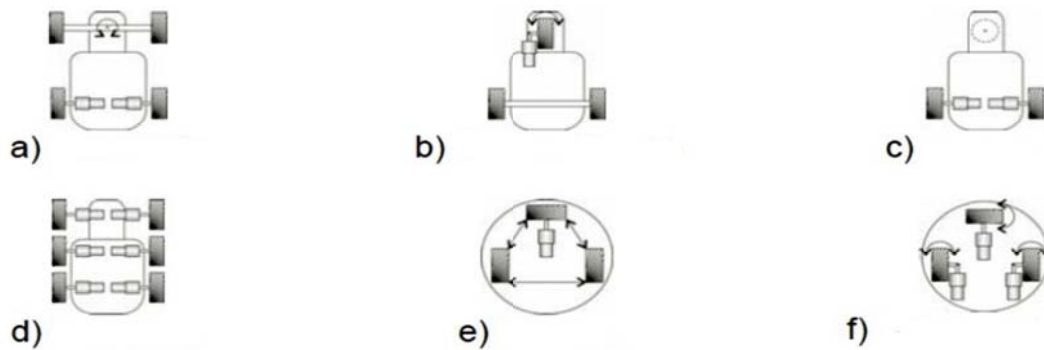


Fig. 2.1 Configuraciones más utilizadas para un robot móvil.

En la Figura 2.1 se observa la configuración a) Ackerman, b) triciclo clásico, c) diferencial, d) skid steer, e) síncrona y f) omnidireccional.

Hay cuatro tipos de ruedas para la locomoción [1] las cuales son: convencionales, tipo castor, ruedas de bolas y omnidireccionales, como se observa en la Figura 2.2.



Fig. 2.2 Tipos de ruedas a ocupar.

La configuración del robot a desarrollar es: un robot móvil terrestre con dos ruedas convencionales, con dos ruedas de bola y tracción diferencial.

Esta configuración es la más documentada y abordada por su sencillez en el modelo dinámico.

Con el objeto de hacer más tratable el problema del modelado en las configuraciones cinemáticas, se suelen establecer algunas suposiciones de diseño y de operación [1].

En las suposiciones de diseño generalmente se toman tres:

1. El material no contiene partes flexibles, por lo que se puede considerar como un cuerpo rígido.
2. Las ruedas tienen a lo más un eslabón de dirección (reducir complejidad del modelado).
3. Los ejes de dirección son perpendiculares a la superficie (los movimientos son en un solo plano).

En las suposiciones de operación son tres:

1. Se descarta toda irregularidad en la superficie.
2. La fricción de traslación es lo suficientemente grande para que no haya desplazamiento de traslación.
3. La fricción rotacional es lo suficientemente pequeña para no exista desplazamiento rotatorio.

## **2.1 Modelado cinemático**

La cinemática es la ciencia que estudia el movimiento sin considerar las fuerzas que lo ocasionan, se puede dividir en directa e inversa. La cinemática directa consiste en calcular la posición y orientación del efector final, la cinemática inversa consiste en calcular los ángulos articulares dada la posición y orientación.

En robótica móvil la cinemática directa consiste en calcular la posición y orientación del vehículo respecto a un marco de referencia fijo.

Las ecuaciones del modelo cinemático del robot diferencial móvil [7] [8], requieren que las llantas traseras sean idénticas, paralelas, no deformables y unidas por un eje, además, usa una rueda omnidireccional frontal con lo cual se garantiza que la plataforma se encuentre solo en el plano XY. El móvil diferencial diseñado no está unido por un eje y utiliza dos ruedas de bola que garantizan que la plataforma se encuentre sobre el plano XY, por lo que el modelo cinemático utilizado se calculara como en [7], [8].

En la Figura 2.3 se muestra el robot móvil diferencial en el plano XY, donde  $(x, y)$  es la posición del punto medio del eje que une a las dos llantas traseras,  $\varphi$  es el ángulo que forma el eje de simetría del móvil respecto al eje X positivo,  $w$ ,  $w_i$  y  $w_d$  son las velocidades angulares del punto medio, la llanta izquierda y derecha respectivamente,  $r$  es el radio de las llantas,  $v$  es la velocidad lineal del punto medio y  $l$  es la separación entre el punto medio y la rueda de tracción.

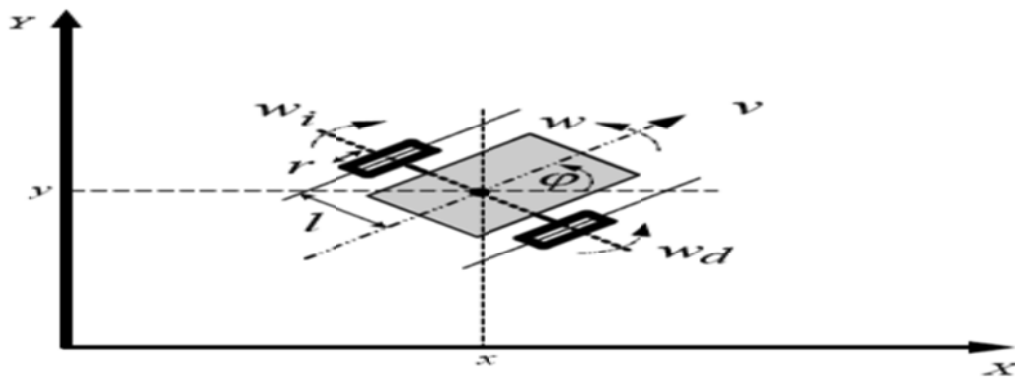


Fig. 2.3 Diagrama del robot móvil.

La velocidad lineal  $v$ , está formada por las proyecciones de  $v$  con los ejes X e Y, es decir  $\dot{x}$  y  $\dot{y}$ . La velocidad angular  $w$  es igual a la razón de cambio de  $\varphi$  con respecto al tiempo [28], la cual se denota como  $\dot{\varphi}$ . De la Figura 2.3 se puede expresar  $\dot{x}$ ,  $\dot{y}$ ,  $\dot{\varphi}$  como:

$$\dot{x} = v \cos(\varphi) \quad (2.1)$$

$$\dot{y} = v \sen(\varphi) \quad (2.2)$$

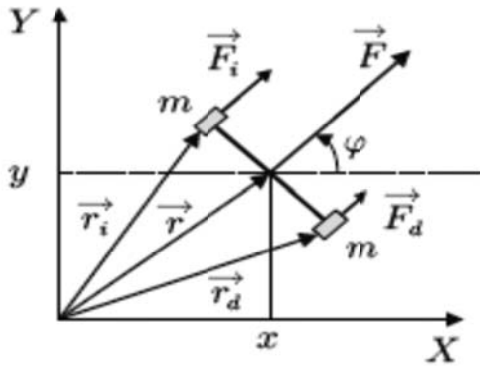
$$\dot{\varphi} = w \quad (2.3)$$

Las variables de interés son  $v$  y  $w$ , se obtienen de la diferencia de velocidades angulares izquierda y derecha  $w_i$  y  $w_d$ , que son las velocidades de las ruedas.

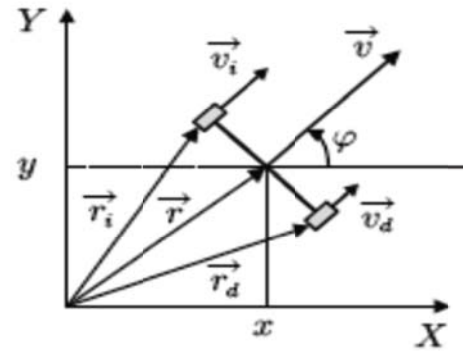
Si se derivan (2.1), (2.2) y (2.3), para encontrar la relación de  $v$  y  $w$  con las velocidades angulares  $w_i$  y  $w_d$ ; y se supone que sobre cada una de estas actúa una fuerza  $F_i$  y  $F_d$ , estas fuerzas provocan las velocidades  $v_i$  y  $v_d$  respectivamente. Usando como coordenadas generalizadas a  $(x, y, \varphi)$  y los diagramas de fuerza y velocidades de la Figura 2.4 (a) y (b), se encuentra que la posición de los puntos de contacto de las llantas en el plano XY está dada por (2.4) y (2.5).

$$r_i = (x - l \sin \varphi) \hat{x} + (y + l \cos \varphi) \hat{y} \quad (2.4)$$

$$r_d = (x + l \sin \varphi) \hat{x} + (y - l \cos \varphi) \hat{y} \quad (2.5)$$



(a) Diagrama de fuerzas



(b) Diagrama de velocidades

Fig. 2.4 Diagrama de fuerza y velocidad de un robot móvil diferencial.

Donde  $\hat{x}$  y  $\hat{y}$  son vectores unitarios de las componentes asociadas a los ejes XY,  $r_i$  y  $r_d$  indican la posición en el plano XY del punto de contacto de la rueda izquierda y derecha respectivamente.

Además, se encuentra que si  $m$  es la masa de cada una de las llantas, entonces la energía cinética del sistema está dada por (2.6) [30].

$$T = \frac{m}{2} \left( |\dot{r}_i|^2 + |\dot{r}_d|^2 \right) = m(\dot{x}^2 + \dot{y}^2 + l^2 \dot{\varphi}^2) \quad (2.6)$$

Las ecuaciones de movimiento del sistema físico están dadas por:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{x}} \right) - \frac{\partial T}{\partial x} = Q_x \quad (2.7)$$

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{y}} \right) - \frac{\partial T}{\partial y} = Q_y \quad (2.8)$$

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\varphi}} \right) - \frac{\partial T}{\partial \varphi} = Q_\varphi \quad (2.9)$$

Donde  $Q_x$ ,  $Q_y$ ,  $Q_\varphi$  son las componentes de la fuerza generalizada, las cuales estan dadas por:

$$Q_x = \vec{F}_i \cdot \frac{\partial \vec{r}_i}{\partial x} + \vec{F}_d \cdot \frac{\partial \vec{r}_d}{\partial x} = (\vec{F}_i + \vec{F}_d) \cdot \hat{x} \quad (2.10)$$

$$Q_y = \vec{F}_i \cdot \frac{\partial \vec{r}_i}{\partial y} + \vec{F}_d \cdot \frac{\partial \vec{r}_d}{\partial y} = (\vec{F}_i + \vec{F}_d) \cdot \hat{y} \quad (2.11)$$

$$Q_\varphi = \vec{F}_i \cdot \frac{\partial \vec{r}_i}{\partial \varphi} + \vec{F}_d \cdot \frac{\partial \vec{r}_d}{\partial \varphi} = l(\vec{F}_i + \vec{F}_d) \cdot (\cos\varphi \hat{x} + \sen\varphi \hat{y}) \quad (2.12)$$

Usando (2.10) a (2.12) y la segunda ley de Newton las ecuaciones de movimiento están dadas por:

$$2m\ddot{x} = (\vec{F}_i + \vec{F}_d) \cdot \hat{x} \quad (2.13)$$

$$2m\ddot{y} = (\vec{F}_i + \vec{F}_d) \cdot \hat{y} \quad (2.14)$$

$$2ml^2\ddot{\varphi} = l(\vec{F}_i + \vec{F}_d) \cdot (\cos\varphi \hat{x} + \sen\varphi \hat{y}) \quad (2.15)$$

Puesto que  $F_i$  y  $F_d$  son las fuerzas que actúan sobre la llanta izquierda y derecha respectivamente, proporcionando las velocidades  $\vec{v}_i$  y  $\vec{v}_d$ , entonces

$$\vec{F}_i = m\vec{v}_i = m\dot{v}_i(\cos\varphi \hat{x} + \sen\varphi \hat{y}) \quad (2.16)$$

$$\vec{F}_d = m\vec{v}_d = m\dot{v}_d(\cos\varphi \hat{x} + \sen\varphi \hat{y}) \quad (2.17)$$

Usando (2.16) y (2.17), sustituyendo de (2.13) a (2.15), después de realizar una integración respecto al tiempo se encuentra:

$$2\dot{x} = (v_i + v_d)\cos\varphi + c_x \quad (2.18)$$

$$2\dot{y} = (v_i + v_d)\sen\varphi + c_y \quad (2.19)$$

$$2l\dot{\varphi} = (v_d - v_i) + c_\varphi \quad (2.20)$$

Donde  $c_x$ ,  $c_y$  y  $c_\varphi$  son tres constantes de integración.

Si no hay deslizamiento de rodadura se puede imponer que:

$$v_i = rw_i \quad (2.21)$$

$$v_d = rw_d \quad (2.22)$$

Si las constantes de las ecuaciones (2.18) a (2.20) son cero resulta:

$$\dot{x} = \frac{(w_d+w_i)r}{2} \cos\varphi \quad (2.23)$$

$$\dot{y} = \frac{(w_d+w_i)r}{2} \operatorname{sen}\varphi \quad (2.24)$$

$$\dot{\varphi} = \frac{(w_d-w_i)r}{2} \quad (2.25)$$

De las ecuaciones (2.23) y (2.24) se obtiene la magnitud de la velocidad  $v$  lineal.

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} = \frac{(w_d+w_i)r}{2} \quad (2.26)$$

La magnitud de la velocidad angular  $w$  con respecto al sistema de referencia inercial XY es:

$$w = \frac{(w_d-w_i)r}{2l} \quad (2.27)$$

Las ecuaciones (2.26) y (2.27) se pueden expresar de forma matricial como:

$$\begin{bmatrix} v \\ w \end{bmatrix} = T \begin{bmatrix} w_d \\ w_i \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2l} & -\frac{r}{2l} \end{bmatrix} \begin{bmatrix} w_d \\ w_i \end{bmatrix} \quad (2.28)$$

Si la trayectoria deseada  $X_d(t) = [x_d(t), y_d(t), \varphi_d(t)]$ , el problema de la cinemática inversa consiste en encontrar un perfil de velocidad  $[v_d(t), w_d(t)]$  que genere la posición y orientación deseada al vehículo.

$$\Psi[v_d(t), w_d(t)] = X_d(t) \quad (2.29)$$

La trayectoria deseada se puede escribir en forma matricial ocupando (2.1) a (2.3).

$$\begin{bmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \\ \dot{\varphi}_d(t) \end{bmatrix} = \begin{bmatrix} \cos\varphi_d(t) & 0 \\ \operatorname{sen}\varphi_d(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_d(t) \\ w_d(t) \end{bmatrix} \quad (2.30)$$

## 2.2 Modelado dinámico

La dinámica es el campo dedicado al estudio de fuerzas que se requieren para ocasionar movimiento, en el robot móvil. El modelo dinámico está integrado por las posiciones  $(x, y)$ , la orientación  $(\varphi)$ , las velocidades lineales  $(\dot{x}, \dot{y})$ , la velocidad angular  $(\dot{w})$ , la aceleración lineal  $(\ddot{x}, \ddot{y})$ , la aceleración angular  $(\ddot{w})$ , los pares utilizados  $(\tau_i, \tau_d)$  que representan la fuerza aplicada a la llanta izquierda y a la llanta derecha respectivamente, utilizando Lagrange [30] podemos expresar el modelo dinámico del vehículo en variables de estado [16].

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)\tau + A^T(q)\lambda \quad (2.31)$$

Donde:

$M(q)$  Matriz de inercia

$C(q, \dot{q})$  Matriz de fuerzas de Coriolis y centrífugas

$G(q)$  Vector de pares gravitacionales

$B(q)$  Matriz de transferencia de las entradas

$\tau$  Vector de fuerzas generalizadas

$A^T(q)$  Matriz Jacobiana de las restricciones de movimiento

$\lambda$  Vector de restricciones de fuerzas

Los robots móviles tienen restricciones de movimiento no holónomas al movimiento del robot, definiendo la siguiente notación para la obtención de las ecuaciones de movimiento y las ecuaciones de restricción de movimiento.

$l$  distancia entre eje de simetría del vehículo y la rueda de tracción

$d$  distancia a lo largo del eje de simetría del punto medio del eje de las ruedas hasta el centro de las masas del vehículo

$r$  radio de las ruedas

$m$  masa total del vehículo



$m_p$  masa de la plataforma del vehículo

$m_R$  masa de la rueda incluido el motor

$I_p$  momento de inercia de la plataforma, tomada alrededor del eje vertical que intersecta el eje de simetría del punto medio del eje de las ruedas hasta el centro de masa del vehículo

$I_R$  momento de inercia de la rueda incluido el rotor alrededor del eje de la rueda

$I_m$  momento de inercia de cada rueda de tracción incluido el eje del motor tomado sobre el diámetro de la rueda

El vehículo presenta tres restricciones [17] dos no holónomas y una holónoma, la primera es que el vehículo no se puede mover de forma lateral, solo hacia adelante y hacia atrás.

$$\dot{x}\text{sen}\varphi - \dot{y}\text{cos}\varphi = 0 \quad (2.32)$$

Las otras restricciones se asocian a sentido del giro y al no deslizamiento de las ruedas de tracción.

$$\dot{x}\text{sen}\varphi + \dot{y}\text{cos}\varphi + l\dot{\varphi} = r\dot{\theta}_d \quad (2.33)$$

$$\dot{x}\text{sen}\varphi - \dot{y}\text{cos}\varphi - l\dot{\varphi} = r\dot{\theta}_i \quad (2.34)$$

Donde,  $\theta_d$  y  $\theta_i$  son las posiciones angulares de las ruedas de tracción derecha e izquierda respectivamente. Se restan (2.33) y (2.34) resultando.

$$2l\dot{\varphi} = r(\dot{\theta}_d - \dot{\theta}_i) \quad (2.35)$$

Se integra (2.35) y con constantes adecuadas se obtiene (2.36).

$$\varphi = \frac{r}{2l}(\theta_d - \theta_i) \quad (2.36)$$

Al sumar (2.33) y (2.34) se tiene (2.37).

$$\dot{y}\text{cos}\varphi + \dot{x}\text{sen}\varphi = \frac{r}{2}(\dot{\theta}_d + \dot{\theta}_i) \quad (2.37)$$

Se puede escribir a (2.32) y (2.37) en forma matricial como (2.38)

$$A(q)\dot{q} = 0 \quad (2.38)$$

Las coordenadas generalizadas están definidas por (2.39)

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta_d \\ \theta_i \end{bmatrix} \quad (2.39)$$

La matriz A (q) es la matriz Jacobiana de las restricciones de movimiento, definida como (2.40)

$$A(q) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} = \begin{bmatrix} \text{sen}\varphi & -\text{cos}\varphi & 0 & 0 \\ \text{sen}\varphi & \text{cos}\varphi & \frac{r}{2} & \frac{r}{2} \end{bmatrix} \quad (2.40)$$

La energía cinética de la plataforma incluida las ruedas está dada por (2.41).

$$T = \frac{1}{2}m(\dot{y}^2 + \dot{x}^2) + m_p \frac{r}{2l} (\dot{\theta}_d + \dot{\theta}_i)(\dot{x}\text{cos}\varphi - \dot{y}\text{sen}\varphi) + \frac{1}{2}I_R(\dot{\theta}_d^2 + \dot{\theta}_i^2) + \frac{1}{2}I_p^2(\dot{\theta}_d + \dot{\theta}_i)^2 \quad (2.41)$$

Sabiendo

$$m = m_p + 2m_R \quad (2.42)$$

$$I = I_p + 2m_R l^2 + 2I_m \quad (2.43)$$

Las ecuaciones de movimiento de Lagrange que gobiernan al vehículo no-holonómico son:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \left( \frac{\partial T}{\partial q_i} \right) = \tau_i - a_{1i}\lambda_1 - a_{2i}\lambda_2 \text{ con } i = 1, \dots, 4 \quad (2.44)$$

Donde

$q_i$  son coordenadas generalizadas,  $\tau_i$  son las fuerzas generalizadas,  $a_{1i}$  con  $i = 1, \dots, 4$  son los cofactores de la matriz de restricciones,  $\lambda_1$  y  $\lambda_2$  son los multiplicadores de Lagrange.

A la ecuación de la energía cinética (2.41) se sustituye en la ecuación de movimiento de Lagrange (2.44) obteniendo.

$$m\ddot{y} - m_p d(\ddot{\varphi}\text{sen}\varphi + \dot{\varphi}^2\text{cos}\varphi) = \lambda_1\text{sen}\varphi + \lambda_2\text{cos}\varphi \quad (2.45)$$

$$m\ddot{x} - m_p d(\ddot{\varphi}\text{cos}\varphi - \dot{\varphi}^2\text{sen}\varphi) = -\lambda_1\text{cos}\varphi + \lambda_2\text{sen}\varphi \quad (2.46)$$

$$m_p \frac{r}{2l} d(\ddot{x}\text{cos}\varphi - \dot{x}\dot{\varphi}\text{sen}\varphi) + (I_p^2 + I_{Rr})\ddot{\theta}_d - I_p^2\ddot{\theta}_i = \tau_d - \frac{r}{2}\lambda_2 \quad (2.47)$$

$$-m_p \frac{r}{2l} d(\ddot{x}^2 \cos\varphi - \dot{y}^2 \sin\varphi) - I_p^2 \ddot{\theta}_d + (I_p^2 + I_{Rr}) \ddot{\theta}_i = \tau_i - \frac{r}{2} \lambda_2 \quad (2.48)$$

Se puede reescribir las ecuaciones dinámicas del vehículo usando la ecuación (2.30), los pares gravitacionales  $G(q)$  son cero y las restricciones de movimiento  $A(q)$  están dadas por (2.40), ocupando la forma vectorial (2.30) con las ecuaciones (2.45) a (2.48), se puede escribir en:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = B(q)\tau + A^T(q)\lambda \quad (2.49)$$

Donde:

$$M(q) = \begin{bmatrix} m & 0 & -m_p \frac{r}{2l} d \sin\varphi & m_p \frac{r}{2l} d \sin\varphi \\ 0 & m & m_p \frac{r}{2l} d \cos\varphi & -m_p \frac{r}{2l} d \cos\varphi \\ -m_p \frac{r}{2l} d \sin\varphi & m_p \frac{r}{2l} d \cos\varphi & I_p^2 + I_R & -I_p^2 \\ m_p \frac{r}{2l} d \sin\varphi & -m_p \frac{r}{2l} d \cos\varphi & -I_p^2 & I_p^2 + I_R \end{bmatrix} \quad (2.50)$$

$$C(q, \dot{q}) = \begin{bmatrix} -m_p d \dot{\varphi}^2 \cos\varphi \\ -m_p d \dot{\varphi}^2 \sin\varphi \\ 0 \\ 0 \end{bmatrix} \quad (2.51)$$

$$B(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.52)$$

$$\tau = \begin{bmatrix} \tau_d \\ \tau_i \end{bmatrix} \quad (2.53)$$

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \quad (2.54)$$

Con lo cual se dispone del modelo cinemático y dinámico de un robot móvil en configuración diferencial, de modo que el control se reduce a controlar a los motores de CD(Corriente Directa).

Este modelo es complejo de modelar por lo que el control lo reduciremos a controlar a los motores de CD.

# 3 Diseño y construcción del móvil

---

El diseño del móvil requiere que sea estable, eficiente, con comunicación con el exterior y un control que le permita realizar diversas tareas.

Un móvil necesita de una etapa de potencia, de alimentación, de sensores, de comunicación y de tratamiento de las señales, por lo que el funcionamiento de un móvil se puede representar como un diagrama a bloques tal como se muestra en la Figura 3.1.

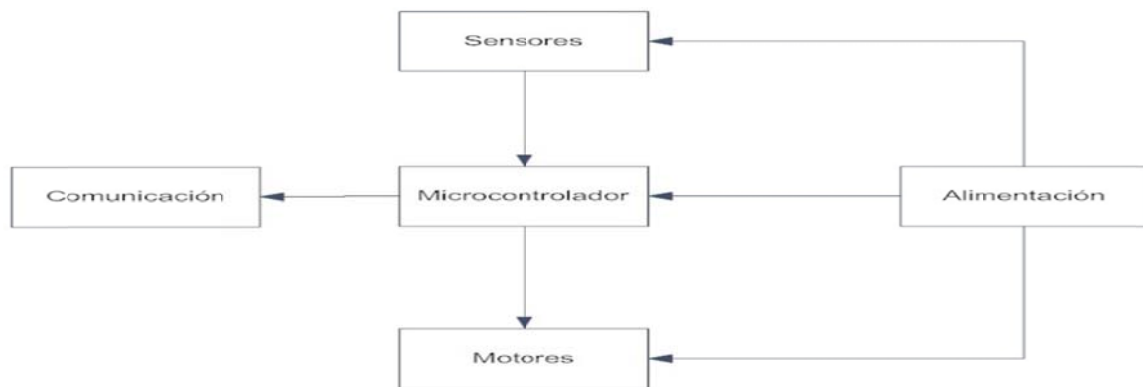


Fig. 3.1 Diagrama a bloques del funcionamiento del móvil.

Con lo cual se puede dividir el diseño mecánico en cuatro bloques principales: sensores, microcontroladores, alimentación y sistema de locomoción.

## 3.1 Sensores

Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, la señal de salida transductible es función de la variable medida [18].

Existen diversos tipos de sensores para la detección en magnitud de varias variables físicas, en particular para robótica móvil la determinación de la posición y orientación del robot dentro del ambiente en el que se encuentra es crucial.

Entre los sensores y técnicas que nos ayudan a posicionar al robot se encuentran: GPS, láseres, odometría, navegación inercial, brújulas magnéticas, posición de referencia de navegación o el modelo de emparejamiento.

Existen dos grupos en las cuales pueden entrar las siete categorías anteriores [10]:

- I. Medidas de posición relativa
  1. Odometría
  2. Navegación inercial
- II. Medidas de posición absolutas
  3. Brújulas magnéticas
  4. Láseres
  5. Sistemas de posicionamiento global
  6. Posición de referencia de navegación
  7. Modelo de emparejamiento

La odometría es el método de navegación más utilizado para el posicionamiento de robots móviles, además de ser de bajo costo, permite velocidades de muestreo altas, es preciso a cortas distancias, sin embargo la idea fundamental de la odometría es la integración de la información del movimiento incremental con el tiempo lo que conduce a la acumulación ilimitada de errores, a pesar de estas limitaciones es una parte importante en un sistema de navegación de un robot móvil.

La odometria se basa principalmente en codificadores, los codificadores son sensores rotativos que transforman un movimiento angular en una serie de pulsos digitales, estos pulsos se pueden ocupar para controlar desplazamientos angulares o lineales si se asocian a cremalleras.

Existen dos tipos de codificadores: absolutos e incrementales.

Los codificadores absolutos producen un código digital único para cada ángulo alrededor del eje rotatorio, con lo cual es posible obtener el desplazamiento absoluto de los motores.

Los codificadores incrementales en su forma más simple no pueden medir el ángulo absoluto del eje, sino el cambio en el ángulo concerniente a cierto dato arbitrario, Si es

necesario conocer la dirección en la cual esta rotando el eje, se le agrega un sensor en un ángulo desfasado de 90 grados con lo cual se crean dos canales de lectura de datos, normalmente conocidos como canal A y canal B. La dirección de rotación se determina con la detección de cual canal se detecta primero y con ello se determina la dirección en la cual esta rotando, este codificador se le conoce como codificador de cuadratura.

El codificador de cuadratura elegido es el codificador enc01a [12], el cual se muestra en la Tabla 3.1.


Componente	Características	Imagen
ENC01A	Voltaje de alimentación: 3.3 - 5 v Codificador incremental de cuadratura	

Tabla 3.1 Codificador enc01a.

El codificador de cuadratura permite el posicionamiento del móvil, pero es necesario conocer si en el posicionamiento existen obstáculos a evitar, para dicha detección se tienen por ejemplo: sensores de ultrasonido, táctiles, infrarrojos, visuales, etc.

Los sensores táctiles son dispositivos mecánicos, los cuales se encuentran normalmente abierto (NO) o normalmente cerrado (NC), la detección se realiza por medio del contacto físico lo cual no es adecuado a nuestras necesidades.

Los sensores infrarrojos son sensores que pueden medir la radiación electromagnética infrarroja de los cuerpos de su campo de visión, su funcionamiento es el siguiente: emiten una luz infrarroja directa por medio de un emisor de infrarrojo (IR), algunos emisores pasan a través de un lente para formar un haz único. Al encontrarse con un obstáculo reflectante, rebota y retorna con cierto ángulo de inclinación, este ángulo depende de la distancia. El rango de detección de los sensores infrarrojos comerciales es 5cm - 80cm aproximadamente.

Los sensores de ultrasonido ocupan una frecuencia superior a los 20KHz, sus aplicaciones son variadas y una de ellas es la medición de distancias, en las cuales de

manera similar a los sensores infrarrojos se realizan emisiones pulsadas que se propagan en un haz cónico estrecho que al encontrarse con un obstáculo, se refleja con la cual el receptor puede saber a que distancia se encuentra dicho objeto gracias al tiempo que le lleva la detección entre la emisión y la recepción. El rango de detección de los sensores de ultrasonido comerciales es de 3cm - 600cm.

Las características del sensor SRF05 a utilizar se muestran en la Tabla 3.2 [14].


Componente	Características	Imagen
SRF05	Voltaje de alimentación: 5 v Frecuencia de trabajo: 40 Hz Tiempo de respuesta: 100 - 25000 $\mu$ s Rango de detección: 3 - 400 cm	

Tabla 3.2 Sensor de ultrasonido SRF05.

El funcionamiento de los sensores de acoplamiento de carga (CCD Charge Coupled devices) o visuales es el siguiente: la luz del objeto ilumina el CCD con lo cual se producen fotones, cuya longitud de onda es de 300 a 1100 nanómetros de acuerdo a la longitud de onda los CCD que están formados por capacitores se cargan variando linealmente en función de la intensidad de iluminación y del tiempo de exposición con lo que puede variar entre iluminación cero (oscuridad) hasta la máxima iluminación (brillante) [18].

Los sensores CCD se emplean en cámaras de TV de estado sólido o dispositivos de formación de imágenes que trabajan con radiación visible o infrarroja, el conocimiento de la ubicación de los fotosensores ofrece ventajas respecto a las cámaras de TV de tubo de vacío, entre las que destacan su bajo consumo, su resolución y sensibilidad.

El procesamiento y análisis de las imágenes en primera instancia requieren la adquisición de una gran cantidad de datos que hacen necesario un equipo dedicado con requerimientos mínimos elevados, es por ello que se decidió hacer uso del proyecto CMUcam, el cual consiste en proveer de la capacidad de visión a pequeños sistemas embebidos a bajo costo y potencia para robots móviles.

Las características de la cámara CMUcam3 se muestran en la Tabla 3.3 [15].


Componente	Características	Imagen
CMUcam3	Voltaje de alimentación: 6 - 15 v Consumo de corriente: 150 mA	

Tabla 3.3 Características CMUcam3.

## 3.2 Microcontroladores

Un microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se destina a gobernar una sola tarea [25].

En el mercado existen cerca de 50 fabricantes de microcontroladores entre los que destacan Microchip y Atmel, debido a su amplia documentación, software gratuito, entre otros beneficios.

Microchip dispone de 296 dispositivos que se encuentran en las familias: PIC10xx, PIC12xx, PIC16xx, PIC18xx, PIC24xx, dsPIC30, dsPIC33 y PIC32xx.

Atmel dispone de 460 dispositivos que se encuentran en las familias: Avr8, Avr32, Avrxmega y Megaavr.

En 2005 Massimo Banzi y otros colaboradores crearon lo que hoy es la plataforma de desarrollo Arduino. Esta plataforma dispone alrededor 21 modelos diferentes de tarjetas con diversas prestaciones.

Los modelos más extendidos son Arduino uno y Arduino mega los cuales se basan en microcontroladores Atmel.

Las características del microcontrolador Arduino uno [18], se muestran en la Tabla 3.3.




Componente	Características	Imagen	
Arduino uno	Microcontrolador	ATmega328	
	Voltaje de alimentación	5v	
	Voltaje de entrada	5v-20	
	Pines de I/O Digital	14	
	Entrada Analógica	6	
	Corriente por pin I/O	40mA	
	Velocidad del reloj	16 MHz	
	SRAM	2 KB	
	EEPROM	1 KB	
	Pines interrupciones	2	

Tabla 3.3 Características de Arduino uno.

Las características del microcontrolador Arduino mega [19], se muestran en la Tabla 3.4.


Componente	Características	Imagen	
Arduino mega	Microcontrolador	ATmega2560	
	Voltaje de alimentación	5v	
	Voltaje de entrada	5v-20	
	Pines de I/O Digital	54	
	Entrada Analógica	16	
	Corriente por pin I/O	40mA	
	Velocidad del reloj	16 MHz	
	SRAM	8 KB	
	EEPROM	4 KB	
	Pines interrupciones	6	

Tabla 3.4 Características de Arduino mega.

Estas tarjetas de desarrollo se han extendido alrededor del mundo gracias a su software gratuito, hardware abierto, una comunidad de desarrolladores, la inclusión en software reconocido como Matlab, Labview, Android, bajo costo y constante mejoramiento, de ahí su elección.

### 3.3 Motores

Los motores son comúnmente utilizados para dotar de movimiento a los robots móviles, existe una amplia gama dependiendo de su empleo, los más utilizados en robótica son los de corriente directa (CD), debido a que su modelado puede ser simplificado a un sistema lineal lo que facilita el control, específicamente los de imán permanente (MCDIP) debido a que el voltaje de control es aplicado al circuito de armadura y el circuito de campo es excitado de manera independiente.

Hablando de MCDIP, se tienen dos tipos: con escobillas y sin escobillas. Al requerir cambios en el sentido de giro ocuparemos motores con escobillas, ya que al cambiar la polaridad de sus terminales cambia el sentido de giro del motor.

Un motorreductor, es decir un motor y una transmisión integrada, le permite obtener el torque necesario para desplazar al móvil, además debe de ser compatible con el codificador de cuadratura y las llantas.

Las características de los motores a utilizar se muestran en la Tabla 3.5 [12].


Componente	Características	Imagen
Motor	Engranaje: 100:1 Velocidad libre: 320 rpm (6 V) Consumo mínimo: 60 mA Par Máximo: 1.8Kg-cm Consumo máximo: 1.6 A	

Tabla 3.5 Motor a utilizar.

Las llantas creadas por Pololu de 42X19 mm, ofrecen la compatibilidad con el codificador enc01a y el motorreductor de 100:1 formando el sistema de locomoción de la Figura 3.2.



Fig. 3.2 Sistema de locomoción.

### 3.4 Comunicación

La comunicación entre dispositivos necesita mínimamente dos elementos; un transmisor y un receptor. Además de que exista un protocolo de comunicación entre ambos. La comunicación puede ser alámbrica o inalámbrica.

Una comunicación inalámbrica hace uso de las ondas electromagnéticas, el rango de frecuencias conocido es de  $3 \text{ a } 3^{27} \text{ Hz}$ , en ese rango existen siete bandas llamadas: radiofrecuencias, microondas, infrarrojo, espectro visible, ultravioleta, rayos x y rayos gamma.

La banda de radiofrecuencias se extiende de los  $3 \text{ a } 3^9 \text{ Hz}$  en ella se encuentran tecnologías como: GSM, GPS, WLAN o WiFi, Bluetooth, radiodifusión (AM, FM), TV, Zigbee, etc. Particularmente Zigbee es un conjunto de protocolos inalámbricos, cuyo objetivo son: la tasa baja de envío y recepción, comunicaciones seguras y bajo consumo.

Esta tecnología, la ocupa entre otros la compañía Maxstream que ha creado 70 módulos de Xbee, los cuales tienen dos variantes de hardware.

Xbee serie 1 provee comunicación punto a punto, así como una red mesh, es decir que cada nodo, además de captar y difundir sus datos sirve de puente para los demás nodos, Xbee serie 2 dispone de varias opciones de red mesh.

Estas variantes de hardware tienen dos versiones de transmisión: regular y profesional, la diferencia entre ellas es el rango de transmisión [26].

Se eligió un Xbee regular de la serie 1 que se muestra en la Tabla 3.6 [13].


Componente	Características	Imagen
Módulo de RF (Xbee)	Voltaje de alimentación: 2.8 – 3.4V Transmisión de datos 30 - 100m Consumo en transmisión: 45 mA (3.3V) Consumo en recepción: 50 mA (3.3V)	

Tabla 3.6 Módulo de RF Xbee.

Se requiere un adaptador que conecte el Xbee al puerto USB de una computadora, para que esta pueda enviar y recibir los datos directamente, como el que se muestra en la Tabla 3.7, el nombre de dicho adaptador es Xbee explorer USB.


Componente	Características	Imagen
Xbee explorer USB	Voltaje de alimentación: 2.8 – 3.4V Transmisión de datos 30 - 100m Consumo en transmisión: 45 mA (3.3V) Consumo en recepción: 50 mA (3.3V)	

Tabla 3.7 Xbee explorer USB.

Cada radio Xbee tiene un *firmware* el cual puede ser actualizado, para ello existe un programa gratuito llamado X-CTU, este programa solo corre en SO Windows, contiene un software de terminal serial, es posible la configuración con algún software de terminal serial por medio de comandos AT (Hyperterminal, PuTTY, Tera term, etc.) y puede configurar la red de Xbee con una dirección de 16 bits.

### 3.5 Alimentación del robot móvil

Suministrar energía eléctrica al móvil se puede realizar por medio de eliminadores o baterías, pero los eliminadores no son una opción debido a que se desea que el móvil sea autónomo, por lo que debemos elegir las baterías adecuadas a nuestro móvil.

Existen baterías con diferentes tipos de tecnología como se muestra en la Tabla 3.8.

Tecnología	Ventajas	Desventajas	Densidad de energía
Plomo ácido (Pb)	Bajo costo	No admite sobrecargas, ni descargas rápidas, sin daño. Peso Excesivo	30 Wh/Kg
Alcalina	Peso ligero	No admite recargas Altamente contaminante	
Níquel Hierro (Ni-Fe)	Bajo costo Larga vida útil	Eficiencia del 65%	40 Wh/Kg
Níquel Cadmio(Ni-Cd)	Recargable 500mAh-1000mAh	Poca capacidad	50Wh/Kg
Níquel Hidruro Metálico (Ni-MH)	Recargable 500mAh- 2800mAh	Baja potencia a frío extremo	80Wh/Kg
Iones de Litio (Li-ion)	Recargable 1500mAh-2800mAh	Poca vida útil con descargas completas	115Wh/Kg
Polímero de Litio (LiPo)	Recargable Tasa de descarga alta	Son delicadas	

Tabla 3.8 Tipos de baterías existentes.

De la Tabla 3.8 se observa que las baterías con mejor desempeño y seguridad son las Ni-MH, por lo que es el tipo de batería elegida.

## 3.6 Planos de construcción del móvil

Tomando en cuenta los componentes mencionados, en las secciones 3.1, 3.2, 3.3 y 3.4 se crearon dos plataformas. La primera con la tarjeta de desarrollo Arduino uno y la segunda con la plataforma Arduino mega. Las medidas están expresadas en milímetros.

### 3.6.1 Móvil con Arduino uno

Con la herramienta CAD Solidworks se realizaron los planos de los componentes de las secciones antes mencionadas, la Figura 3.3 nos muestra el encoder enc01a.

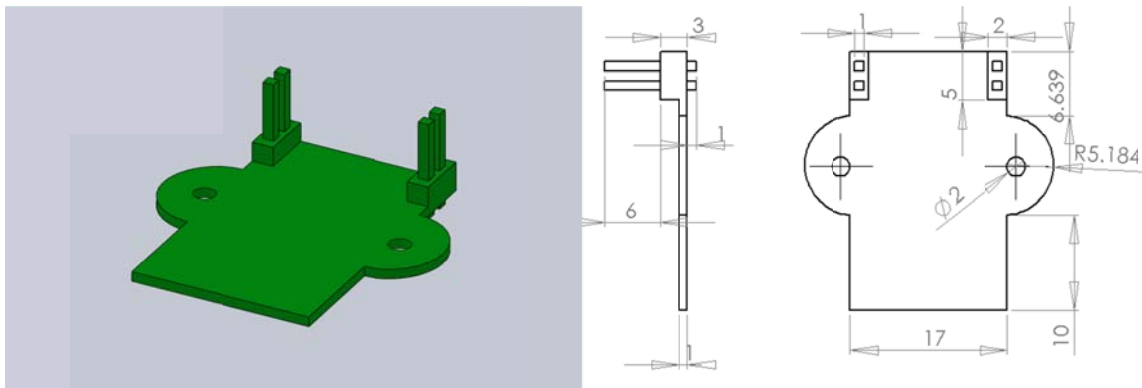


Fig. 3.3 Dibujo CAD de enc01a.

La intención de realizar en la herramienta CAD los diseños de los componentes es para crear un plano del robot móvil a construir y con ello se garantice la repetitividad del diseño, así como la obtención de ciertas medidas utilizables posteriormente y la visualización del centro de gravedad para poder conceptualizar la estabilidad del robot móvil.

La Figura 3.4 muestra la cámara CMUcam3 con la herramienta CAD.

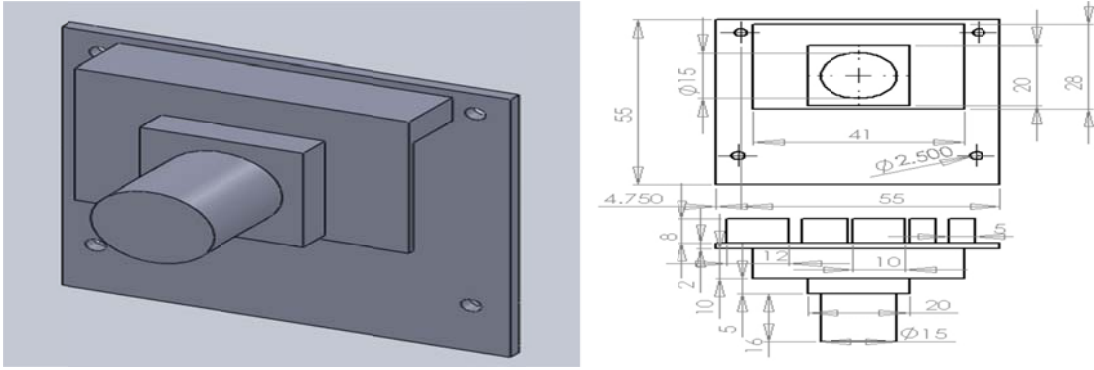


Fig. 3.4 Dibujo CAD de CMUcam 3.

La Figura 3.5 muestra el sensor de ultrasonido SRF05 con la herramienta CAD.

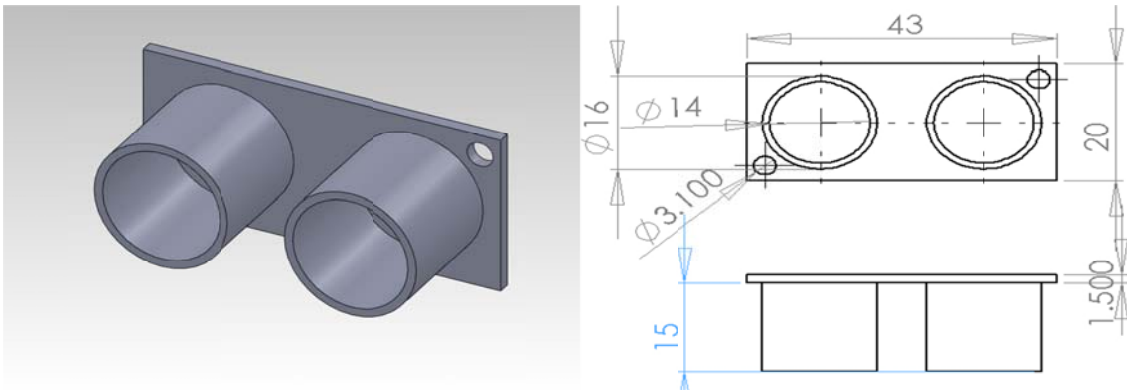


Fig. 3.5 Dibujo CAD de SRF05.

La Figura 3.6 muestra el módulo de comunicación Xbee con la herramienta CAD.

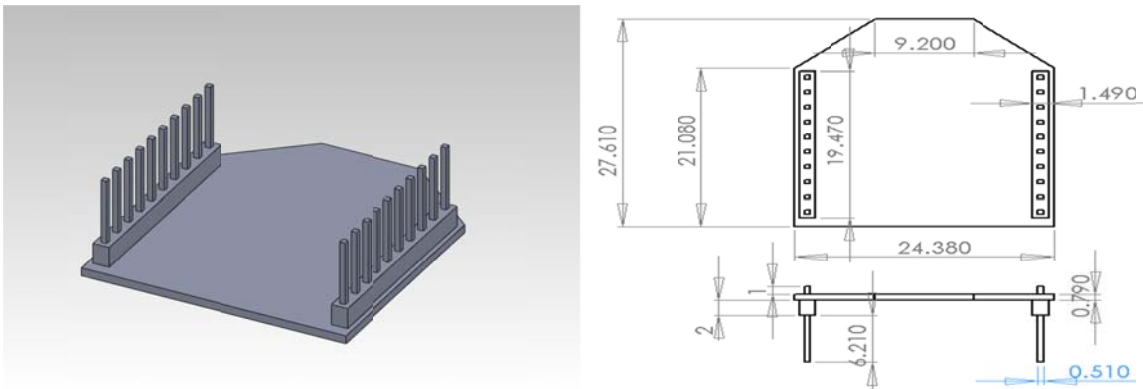


Fig. 3.6 Dibujo CAD del Xbee.

La Figura 3.7 muestra el motor utilizado con la herramienta CAD.

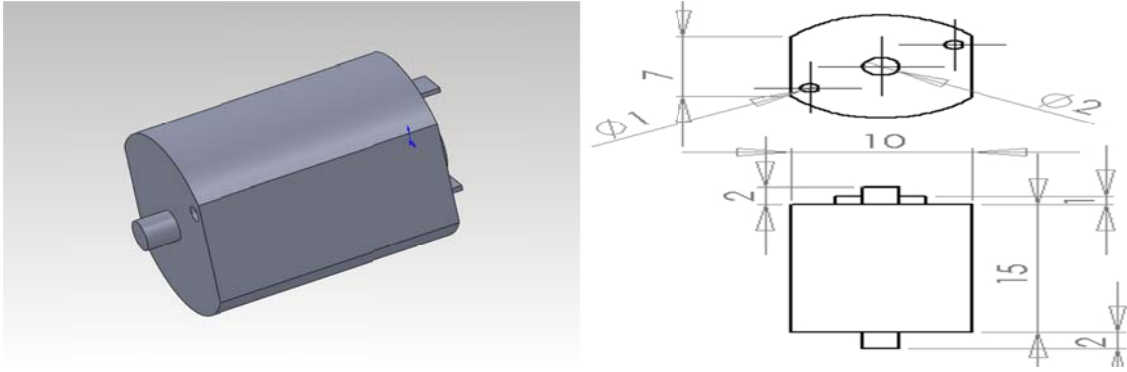


Fig. 3.7 Dibujo CAD del motor utilizado.

La Figura 3.8 muestra la llanta adquirida para el robot móvil.

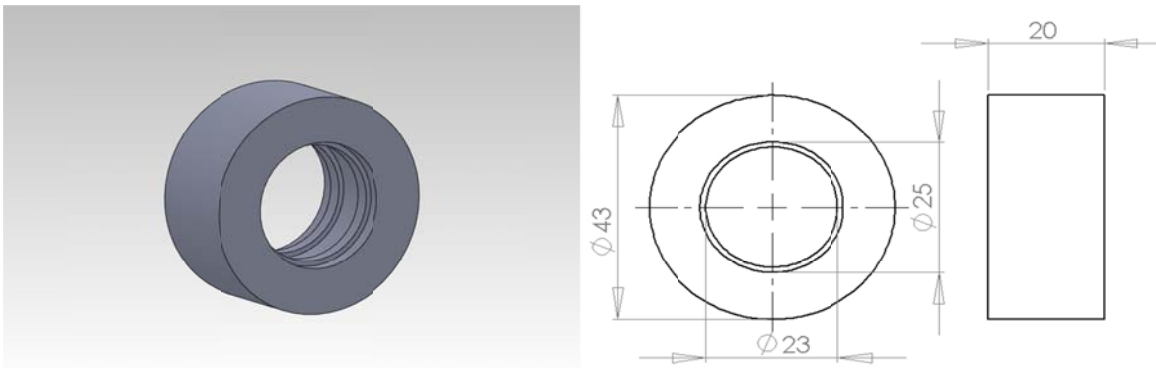


Fig. 3.8 Dibujo CAD de la llanta utilizada.

La Figura 3.9 muestra la tarjeta de desarrollo Arduino uno con la herramienta CAD.

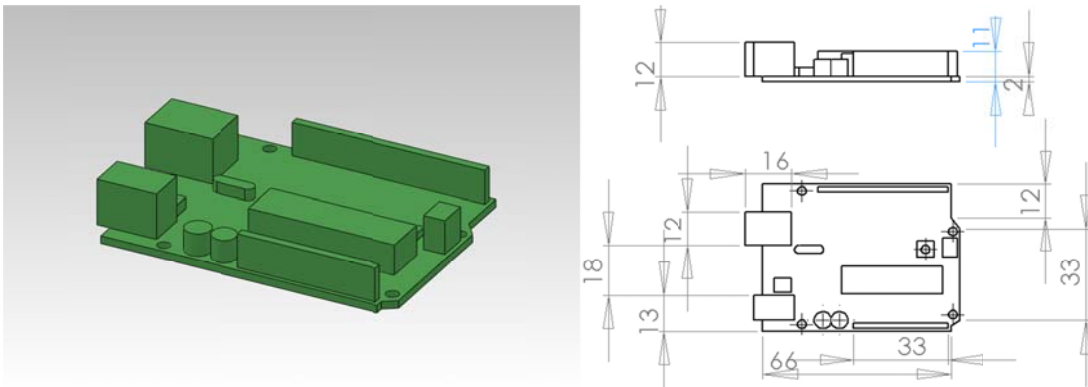


Fig. 3.9 Dibujo CAD de la tarjeta Arduino uno.



Teniendo todos los componentes en la herramienta CAD se procede a realizar la estructura del robot móvil, la cual consta de dos plataformas y soportes para sujetar a los componentes.

En la Figura 3.10 se encuentra el soporte del motor con respecto a la plataforma 1 del móvil.

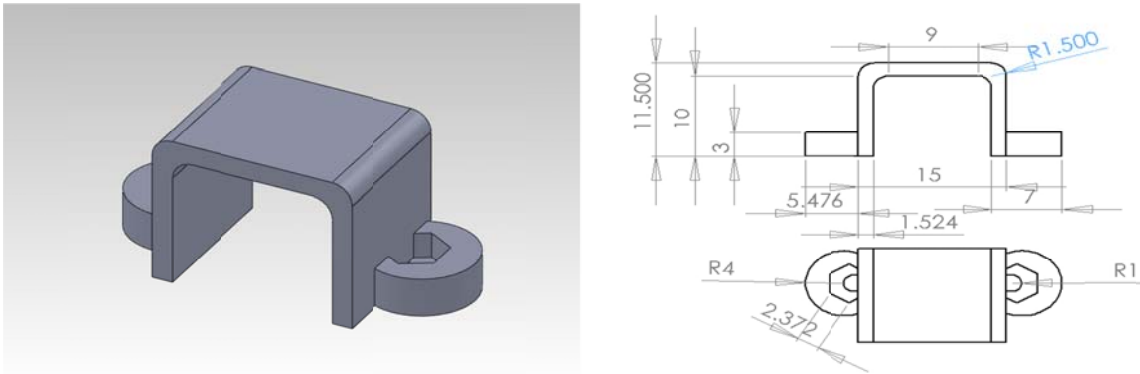


Fig. 3.10 Soporte del motor.

Se requiere una plataforma que sirva para conectar los motores, la plataforma 2, los soportes para CMUcam3 y el soporte Srf05, en la Figura 3.11 se muestra el dibujo CAD de dicha plataforma.

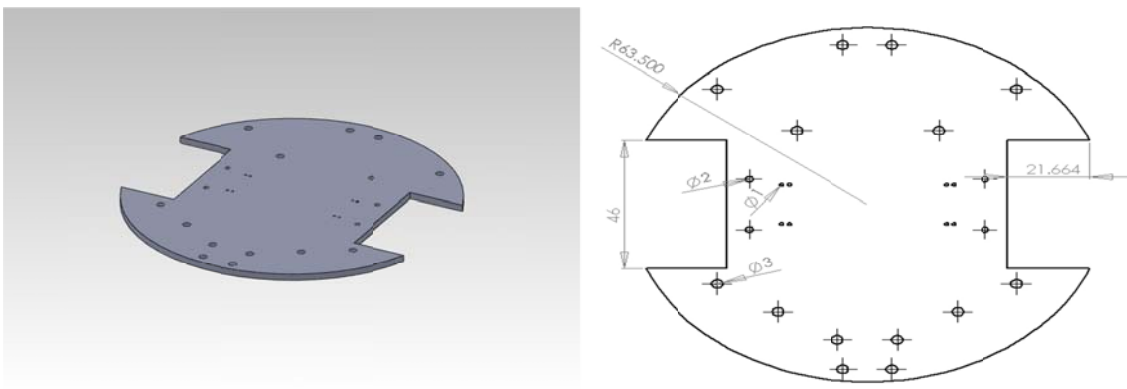


Fig. 3.11 Plataforma 1.

Se requieren soportes para la cámara CMUcam3, con ellos se conectara la plataforma 1 y la cámara, en la Figura 3.12 se muestra el dibujo CAD de dichos soportes.

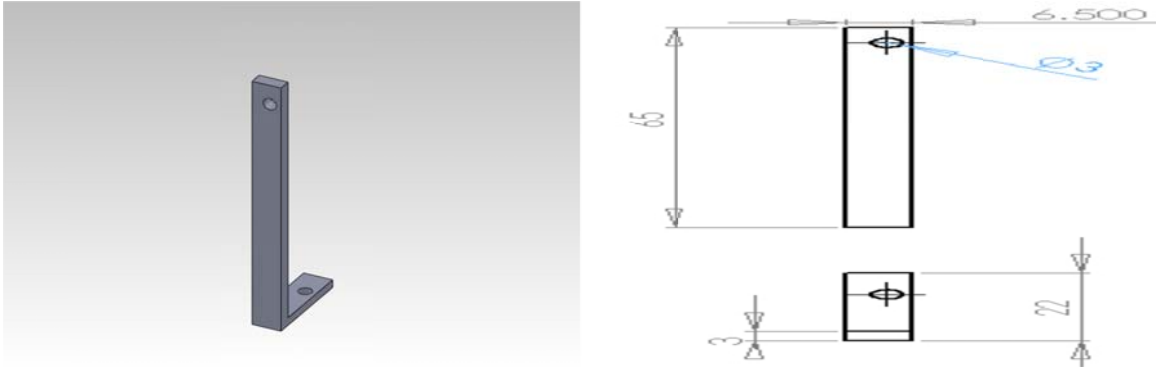


Fig. 3.12 Soporte para CMUcam3.

El sensor de ultrasonido SRF05 necesita un soporte para su conexión con la plataforma 1, con el cual se garantice la sujeción de dicho sensor, en la Figura 3.13 se muestra dicho sensor.

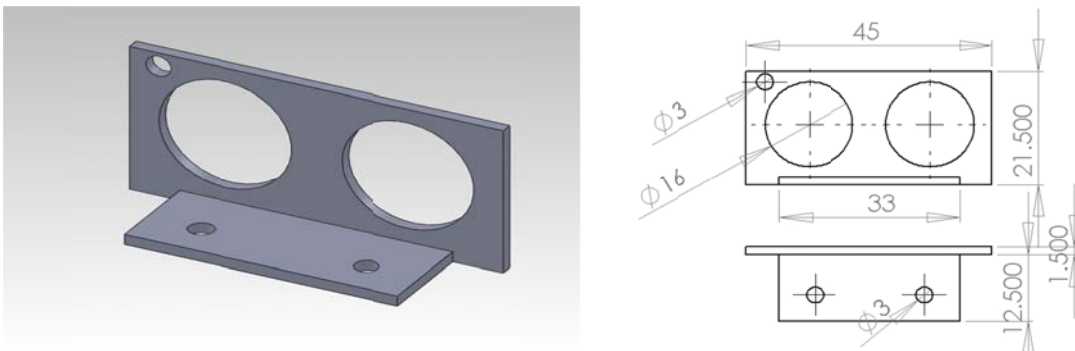


Fig. 3.13 Soporte para SRF05.

Se requiere un soporte para las baterías AA de Ni-MH tal y como se muestra en la Figura 3.14.

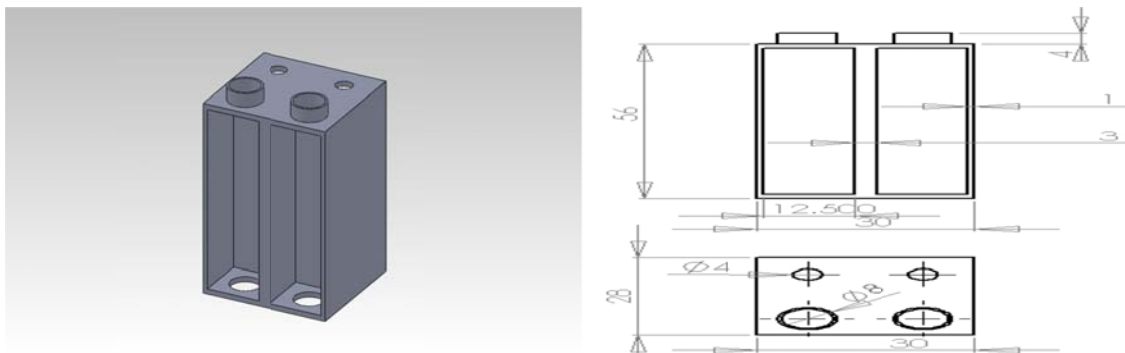


Fig.3.14 Soporte baterías AA.

Por último la plataforma 2, la cual es el soporte entre la tarjeta de desarrollo Arduino uno, una tarjeta electrónica y el robot móvil, el diseño de la plataforma 2 en la herramienta CAD se muestra en la Figura 3.15.

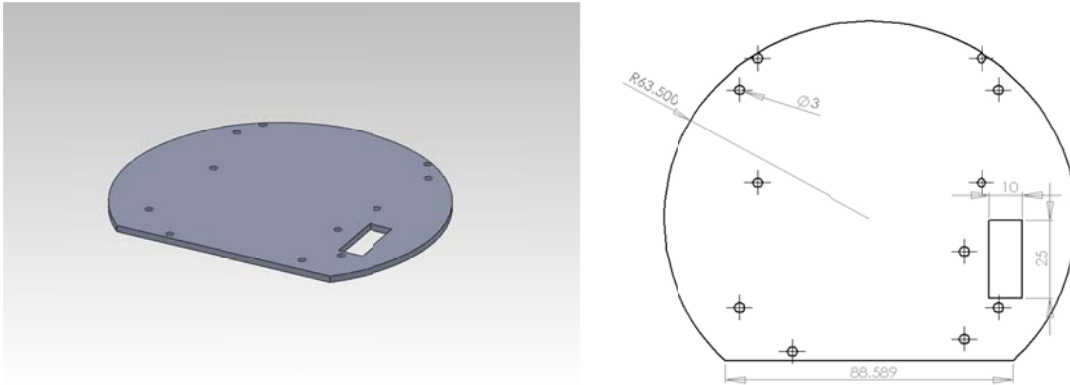


Fig. 3.15 Plataforma 2 del robot móvil.

Una vez que se tienen todas las piezas en la herramienta CAD se procede a ensamblarlas como lo muestra la Figura 3.16 con lo que se ha terminado los planos de construcción del robot móvil para Arduino uno.

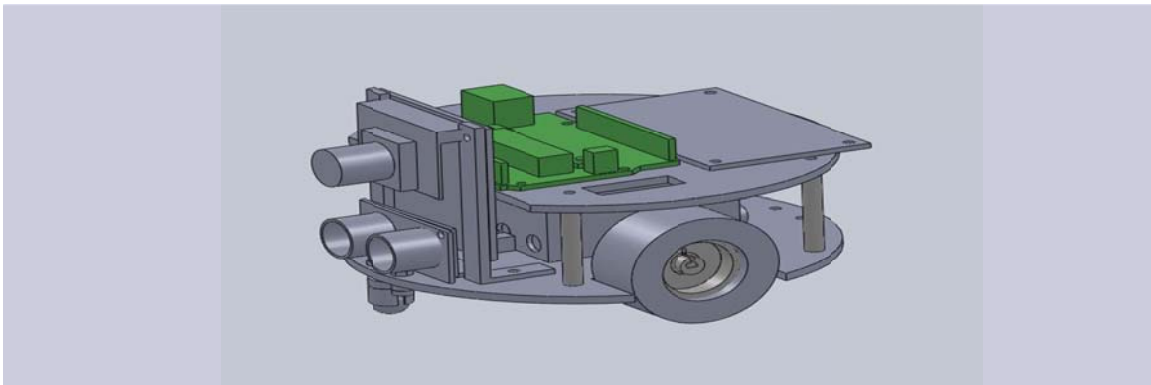


Fig. 3.16 Robot móvil para Arduino Uno.

Cabe mencionar que este móvil fue la primera versión terminada, pero las características técnicas del Arduino uno supusieron una limitante, la cual consiste en la falta de I/O suficientes ya que los codificadores necesitan cuatro entradas con interrupciones y el Arduino uno solo dispone de dos. Se optó por realizar otra versión del móvil con la plataforma Arduino mega.

### 3.6.2 Móvil con Arduino mega

Una de las diferencias del móvil de la tarjeta Arduino mega con respecto al móvil que contiene a la tarjeta Arduino uno es el tamaño, con lo cual las plataformas antes diseñadas no son aplicables para este móvil, aumentando el tamaño del diámetro por 1cm la plataforma 1 para el móvil con Arduino mega se muestra en la Figura 3.17.

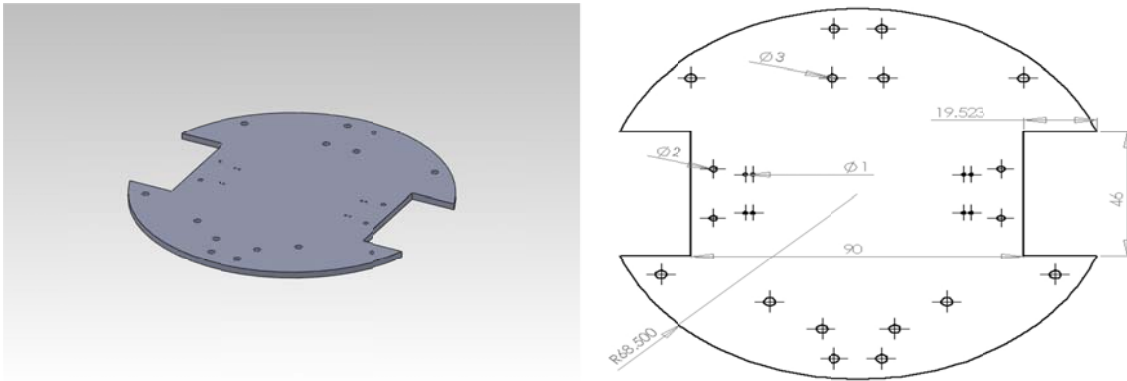


Fig. 3.17 Plataforma 1 Arduino mega.

Igualmente la plataforma 2 sufre modificaciones, para el móvil con la tarjeta de desarrollo Arduino mega, la Figura 3.18 muestra dichas modificaciones.

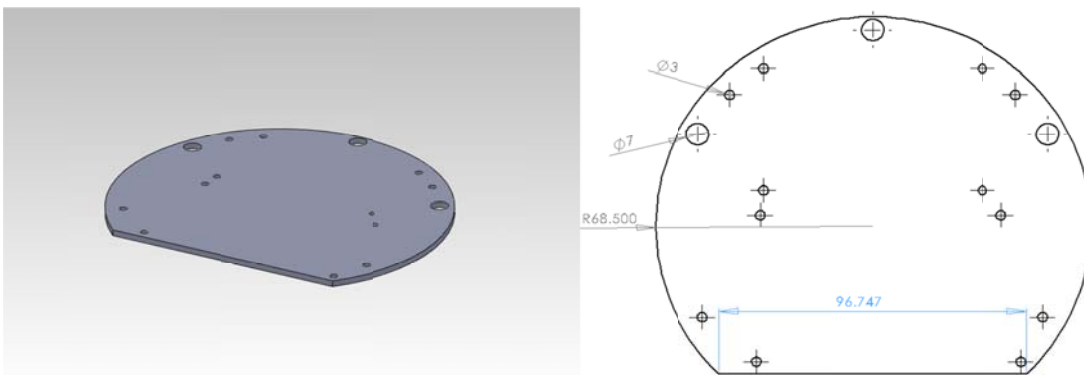


Fig. 3.18 Plataforma 2 Arduino mega.

Con lo anterior podemos ensamblar al móvil para la tarjeta de desarrollo Arduino mega, tal y como se muestra en la Figura 3.19.

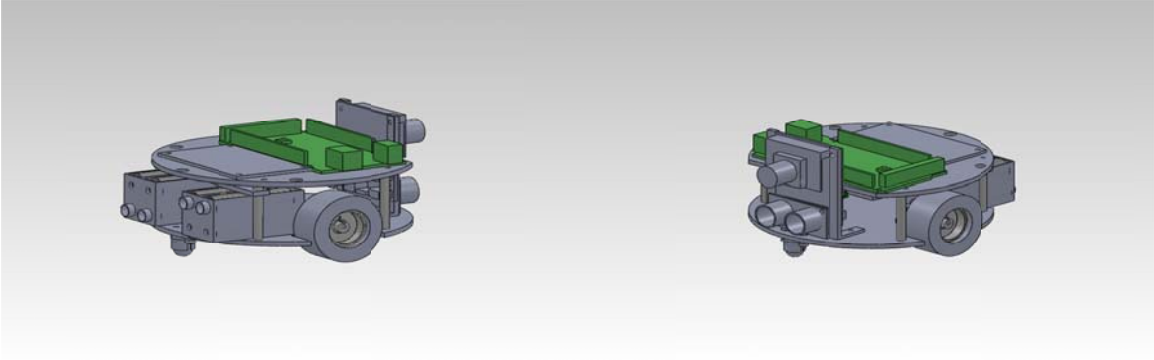


Fig. 3.19 Robot móvil para Arduino mega.

### 3.7 Tarjeta electrónica del móvil

La tarjeta electrónica del robot móvil para Arduino mega tiene dos funciones principales proporcionar el voltaje de alimentación a todos los componentes del robot móvil y conectar la tarjeta de desarrollo Arduino mega con un puente H para proporcionar la potencia necesaria para la puesta en marcha de los motores de CD.

Un puente H es una disposición de cuatro interruptores, como se muestra en la Figura 3.20, en la cual de acuerdo con la activación de los interruptores; el motor tiene un sentido de giro diferente o la inactividad del motor, pero es posible hacer un cortocircuito.

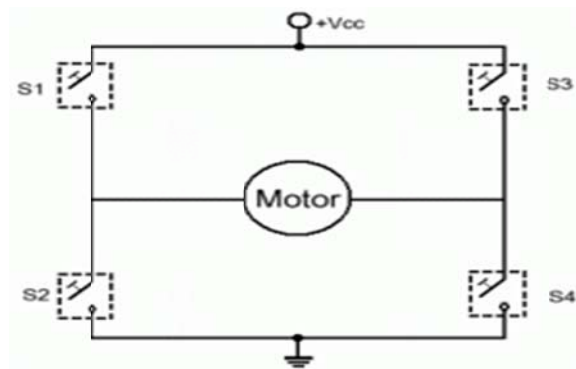


Fig. 3.20 Puente H.

La Tabla 3.9 muestra la Tabla de verdad típica de un puente H.

Motor	Interruptor S1	Interruptor S2	Interruptor S3	Interruptor S4
Apagado	0	0	0	0
No recomendable	0	0	0	1
No recomendable	0	0	1	0
Corto Circuito	0	0	1	1
No recomendable	0	1	0	0
Apagado	0	1	0	1
Giro sentido 1	0	1	1	0
Corto circuito	0	1	1	1
No recomendable	1	0	0	0
Giro sentido 2	1	0	0	1
Apagado	1	0	1	0
Corto circuito	1	0	1	1
Corto circuito	1	1	x	X

Tabla 3.9 Tabla de verdad de un puente H.

Los interruptores pueden ser mecánicos, de estado sólido o circuitos integrados de acuerdo a la Tabla 3.5, el motorreductor 100:1 trabaja con una corriente mínima de 60mA y máxima de 1600mA. El torque desarrollado es de 1.8Kg/cm y el móvil pesa alrededor de .6 Kg, es decir la corriente máxima ocupada por los motores es de 530 mA, por lo que el puente H a ocupar es el circuito integrado L293B el cual soporta hasta 1A.

Cabe mencionar que la tarjeta electrónica ocupa dos fuentes de alimentación una para los codificadores de cuadratura y otra para todos los demás componentes, esto debido a que los codificadores están debajo de los motores de cd y ellos producen demasiada interferencia, la cual afecta el funcionamiento de dicho sensor, por lo que se desacoplo la señal tal y como se muestra en el diagrama esquemático de la Figura 3.21, con la ayuda del software ISIS de Labcenter Electronics.

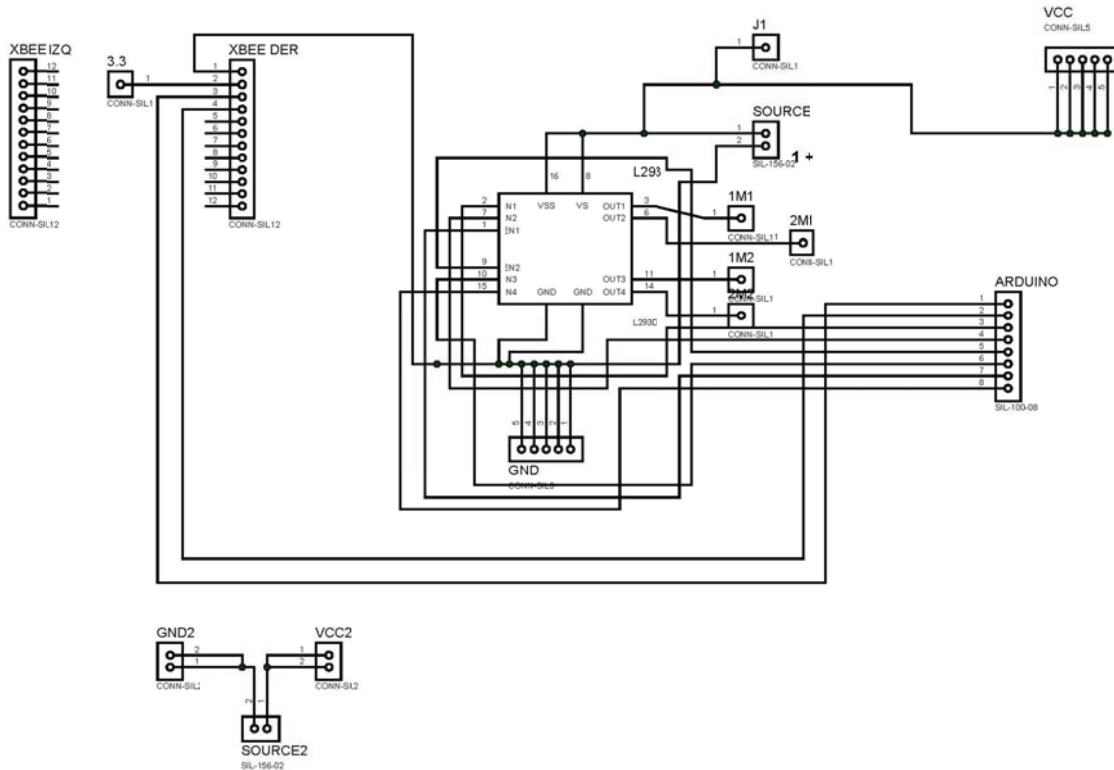


Fig. 3.21 Diagrama de la tarjeta electrónica.

En la Figura 3.22 se muestra la vista Artwork del Software Ares de Labcenter Electronics de la tarjeta de circuito impreso (PCB) desarrollada para el robot móvil.

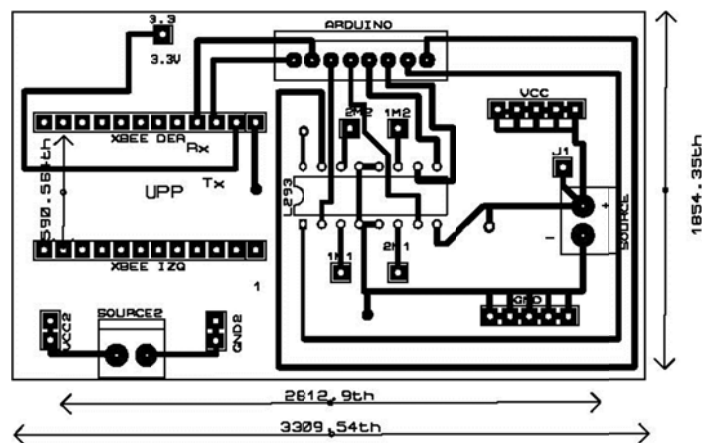


Fig. 3.22 Tarjeta electrónica desarrollada.

La vista 3D de la tarjeta electrónica desarrollada se muestra en la Figura 3.23, en la que se observa la disposición de los dispositivos ocupados.

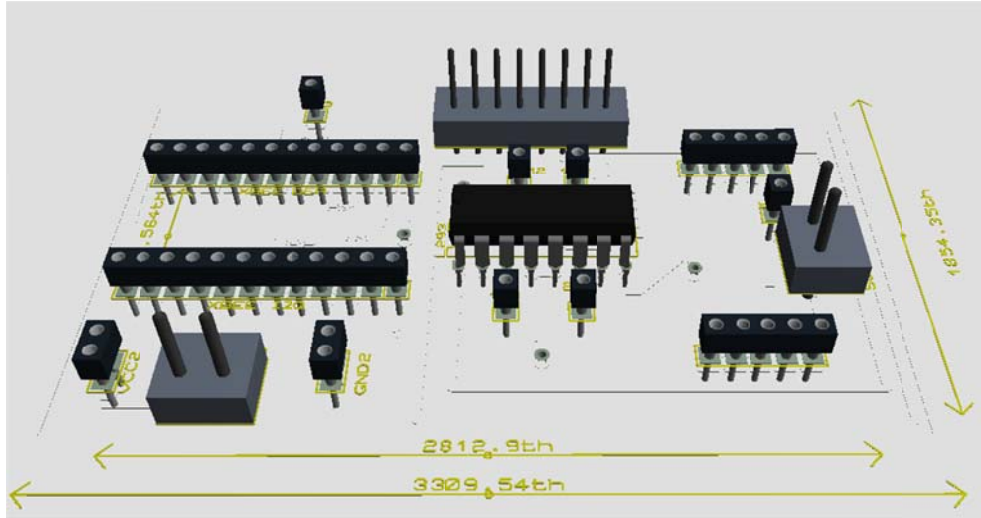


Fig. 3.23 Vista 3D del PCB.

De la Figura 3.23 se observa de izquierda a derecha y de la parte superior a la inferior: la conexión del voltaje de 3.3 V del Xbee explorer con la tarjeta de desarrollo Arduino mega, la tira de pines hembra para el montaje del Xbee explorer, los pines de tierra y Vcc, así como la conexión con la fuente de voltaje para alimentar a los codificadores enc03a, ocho pines los cuales son la conexión para la tarjeta de desarrollo Arduino mega y tienen las siguientes funciones: recepción de datos (1 Rx), transmisión de datos (2 Tx), entradas para la polarización del motor1 (3 y 4 I1), entrada para el uso del pwm por parte del motor1 (5 E1), entradas para la polarización del motor2 (6 y 8 I2) y entrada para el uso del pwm por parte del motor2 (7 E2).

Se tienen los pines para la conexión con los motores con el puente H L293b, además de cinco pines que suministran el segundo voltaje de alimentación, un pin que se conecta con el voltaje de entrada de la tarjeta de desarrollo Arduino mega, la conexión con la segunda fuente de voltaje que alimenta a todos los componentes excepto a los codificadores, por último se tienen cinco pines para la conexión con tierra.

La segunda fuente de alimentación se utilizó debido a que los motores generan mucho ruido a los codificadores, por lo que la solución fue utilizar una fuente de alimentación diferente a la ocupada por los motores.



# 4 Control del móvil

---

## 4.1. Modelado de un motor CD

El motor de corriente directa (CD) es básicamente un transductor de par que convierte energía eléctrica en energía mecánica.

Un motor de CD se puede clasificar en cinco tipos:

- Motor de CD de excitación separada
- Motor de CD en derivación
- Motor de CD de imán permanente
- Motor de CD en serie
- Motor de CD compuesto

Un motor de CD de imán permanente (MCDIP), ofrece beneficios frente a los motores de excitación separada, en derivación, en serie y compuesto cuando las aplicaciones requieren un tamaño pequeño y un caballaje fraccional [19].

Los MCDIP son modelados como se muestra en la Figura 4.1.

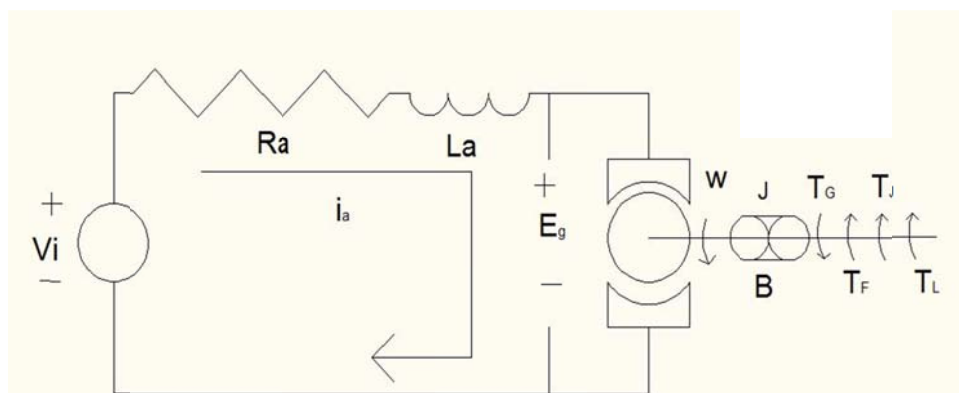


Fig. 4.1 Modelado de un MCDIP.

Donde:

$i_a$  es la corriente de armadura (o corriente del inducido).

$L_a$  es la inductancia de la armadura (o inductancia del inducido).

$R_a$  es la resistencia de la armadura (o resistencia del inducido).

$V_i$  es el voltaje de entrada al motor.

$V_g$  es el voltaje generado por el movimiento de los conductores de la armadura a través del campo magnético producido por los imanes permanentes de los polos del motor.

$\omega$  es la velocidad angular.

$T_G$  par generado.

$T_F$  par debido a la fricción.

$T_J$  par debido a la inercia.

$T_L$  par debido a la carga.

$J$  es el momento de inercia.

$B$  es el coeficiente de fricción viscosa.

El modelo electromecánico estándar de un motor de CD considera al voltaje de armadura y el par debido a la carga como las variables de entrada y a la velocidad como a la variable de salida [20], [21].

Existen tres formas de controlar a un motor de cd:

- Variación del flujo de campo (a través de la resistencia de armadura)
- Variación del voltaje de entrada o de armadura
- Variación de la resistencia del inducido

La variación del flujo de campo, depende de ajustar a la resistencia de flujo también llamada resistencia de armadura, pero en el caso de un MCDIP no es posible realizar una variación del flujo de campo ya que el flujo es fijo.

La variación de la resistencia del inducido es un método en el cual se inserta un resistor en serie al circuito del inducido, pero es inusual esta forma de control y en especial cuando los motores son pequeños ya que dicha resistencia puede tener un efecto nulo o contraproducente.

La variación del voltaje de entrada es el método más utilizado en un motor de CD.

Aplicando la ley de voltajes de Kirchhoff al modelo de la Figura 4.1 se obtiene:

$$V_i(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + V_g(t) \quad (4.1)$$

La ley de inducción de Faraday dice que una vuelta de bobina conductora que gira a través de un campo magnético constante, presenta en sus extremos un voltaje inducido.

En un motor de cd cada bobina del rotor genera un voltaje inducido el cuál es proporcional al flujo magnético y a la velocidad angular  $w(t)$ , es decir

$$V_g(t) = K\phi(t)w(t) \quad (4.2)$$

Al ignorar los cambios del flujo magnético y al ser un MCDIP, el flujo se considera constante modificando (4.2) a

$$V_g(t) = K_e w(t) \quad (4.3)$$

donde  $K_e$  es la constante de velocidad del motor.

El par generado o fuerza generada en la armadura debida al flujo de corriente en las bobinas conductoras, es proporcional a la magnitud de la corriente que fluye en los conductores, la intensidad del campo magnético y a la longitud de los conductores.

Si se considera al flujo constante, el par generado es proporcional a la corriente y está dado por

$$T_G(t) = K_t i_a(t) \quad (4.4)$$

donde  $K_t$  es la constante de par del motor.

El par generado debe de ser igual y opuesto a la suma de los pares de carga, fricción e inercia.

$$T_G(t) = T_F(t) + T_j(t) + T_L(t) \quad (4.5)$$

El par inercial puede ser representado por el producto del momento de inercia total  $J$  y la aceleración angular  $\frac{dw(t)}{dt}$ .

El par debido a la fricción tiene dos componentes, la componente del par que es una función lineal de la velocidad angular o fricción viscosa y los otros términos que involucran fenómenos de fricción, con ello (4.5) se transforma a:

$$T_G = T_F + T_L + J \frac{dw(t)}{dt} + Bw(t) \quad (4.6)$$

donde TF es la suma de las pérdidas de fricción, excepto la fricción viscosa.

Las cuatro ecuaciones (4.1), (4.3), (4.4) y (4.6) son las ecuaciones fundamentales que modelan a los motores de cd, aplicándoles la transformada de Laplace y reacomodando se obtienen:

$$V_i(s) - V_g(s) = (R_a + sL_a)I_a(s) \quad (4.7)$$

$$V_g(s) = K_e \Omega(s) \quad (4.8)$$

$$T_G(s) = K_t I_a(s) \quad (4.9)$$

$$T_G(s) - T_F(s) - T_L(s) = (B + sJ)W(s) \quad (4.10)$$

En la Figura 4.2 se muestra la representación en diagrama de bloques del modelo electromecánico de los MCDIP, el cual consiste de dos entradas:  $V_i(s)$ ,  $T_F(s) + T_L(s)$  y una salida:  $W(s)$  [20].

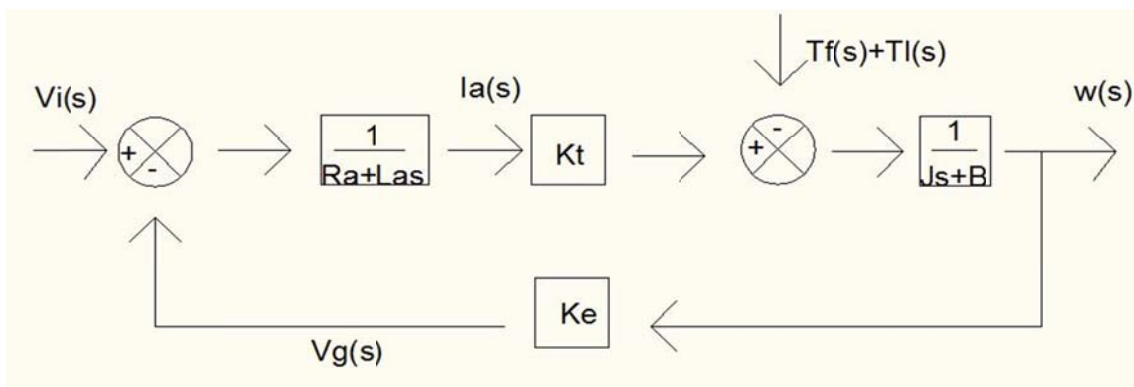


Fig. 4.2.- Modelo en diagrama de bloques de los MCDIP.

Ocupando el diagrama de bloques de la Figura 4.2, la función de transferencia voltaje-velocidad puede representarse como

$$\frac{W(s)}{V_i(s)} = \frac{K_t}{R_a B \left( \frac{L_a}{R_a} s + 1 \right) \left( \frac{J}{B} s + 1 \right) + K_t K_e} \quad (4.11)$$

Donde  $\frac{L_a}{R_a}$  y  $\frac{J}{B}$  son las constantes de tiempo eléctrica y mecánica respectivamente del sistema de segundo orden descrito por función de transferencia.

Normalmente, el valor de  $L_a$  es de magnitud menor que  $R_a$ , de modo que, para valores pequeños de  $L_a \rightarrow 0$ , la constante de tiempo eléctrica es despreciable, ocasionando la reducción de (4.11) a un sistema de primer orden.

$$\frac{W(s)}{V_i(s)} = \frac{K_t}{R_a(Js+B)+K_tK_e} \quad (4.12)$$

Ocupando el diagrama a bloques de la Figura 4.2, la función de transferencia entre el par de carga  $T_F(s) + T_L(s)$  y la velocidad  $W(s)$  se puede representar como:

$$\frac{W(s)}{T_F(s)+T_L(s)} = \frac{-R_a\left(\frac{L_a}{R_a}s+1\right)}{R_aB\left(\frac{L_a}{R_a}s+1\right)\left(\frac{J}{B}s+1\right)+K_tK_e} \quad (4.13)$$

y cuando  $L_a \rightarrow 0$  se reduce (4.13) a un sistema de primer orden.

$$\frac{W(s)}{T_F(s)+T_L(s)} = \frac{-R_a}{R_a(Js+B)+K_tK_e} \quad (4.14)$$

Utilizando el teorema del valor final, la relación en estado estable para un cambio de velocidad debido a un voltaje escalón de las funciones de transferencia (4.12) y (4.14) con un valor  $s=0$ , ocasiona la obtención de los siguientes parámetros [20]:

$$K_m = \frac{K_t}{R_aB+K_tK_e} \quad (4.15)$$

$$T_m = \frac{R_aJ}{R_aB+K_tK_e} \quad (4.16)$$

## 4.2. Identificación del Sistema

La determinación de los parámetros de la función de transferencia del modelo electromecánico del motor puede realizarse mediante varios métodos, los más comunes son: los datos de placa del motor, la medición directa o indirecta (estimación) de los diferentes parámetros.

Para realizar una medición directa o indirecta es necesario aplicar una señal de excitación del sistema, normalmente una señal que puede ser: escalón, impulso, sinusoidal, pseudoaleatoria, rampa, parábola o ruido blanco. Una vez que es aplicada dicha señal el sistema responde de una forma determinada.

La respuesta del sistema puede ser determinada por medio de dos métodos generales: métodos no paramétricos y métodos paramétricos.

Entre los métodos no paramétricos se tienen: análisis de la respuesta transitoria, análisis de frecuencia, análisis de correlación, análisis espectral [22].

Los métodos paramétricos minimizan algún(os) criterio(s) de error entre el modelo y el proceso se dividen en dos modelos paramétricos: los modelos adaptados y los modelos de caja negra [23].

En los modelos adaptados los parámetros representan valores desconocidos que tienen una interpretación física para un modelo específico, los modelos caja negra son generales y a menudo no tienen una interpretación física.

Se ocupara por la sencillez, así como por la documentación al método no paramétrico de análisis de la respuesta transitoria.

En la Figura 4.3 se muestra el diagrama de bloques del proceso de identificación.

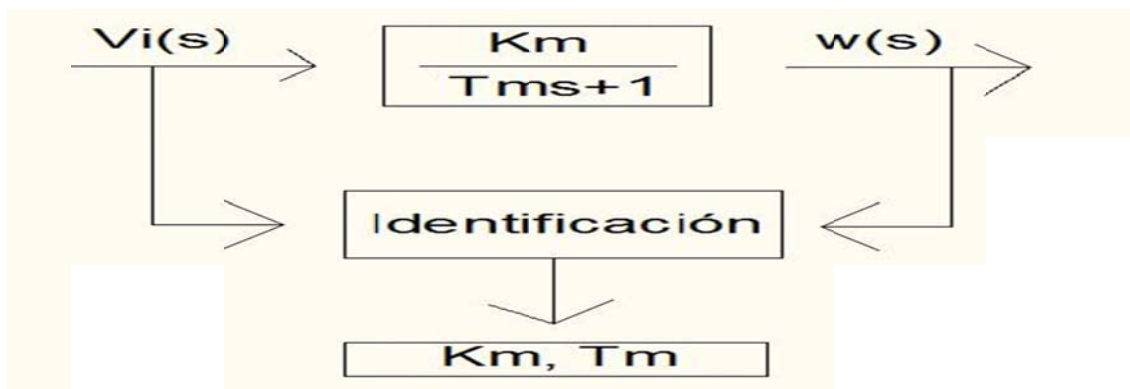


Fig. 4.3.- Proceso de identificación para un sistema de primer orden.

El proceso de la Figura 4.3 consiste en aplicar una excitación de entrada de tipo escalón y observar el comportamiento del sistema para estimar los parámetros del modelo.

Para la identificación experimental de los parámetros del sistema, se considera que el sistema se encuentra libre de perturbaciones, de esta manera, la función de transferencia describe el comportamiento total del sistema mediante (4.12) y ocupando (4.15) y (4.16) se puede describir (4.12) en:

$$\frac{W(s)}{V_i(s)} = \frac{K_m}{T_m s + 1} \quad (4.17)$$

A (4.17) se le aplica una señal escalón unitario de amplitud A, se obtiene su transformada inversa de Laplace para encontrar su representación temporal. La velocidad angular queda determinada como:

$$w(t) = AK_m - AK_m e^{-\frac{t}{T_m}} \quad (4.18)$$

Los parámetros condensados  $K_m$  y  $T_m$  son los valores que determinan, el comportamiento del sistema ante un voltaje de entrada.

Al tratarse de un modelo lineal, se espera que la respuesta del sistema, cuando ha alcanzado su estado estacionario, haya alcanzado la amplitud del voltaje de excitación.

Para encontrar de manera experimental los valores de  $K_m$  y  $T_m$  se aplica, partiendo del reposo un voltaje al MCDIP el tiempo suficiente para que este alcance su respuesta permanente, una vez que esto sucede, se suprime la alimentación hasta que el motor se encuentre nuevamente en reposo para enseguida volver a aplicarle el voltaje pero invirtiendo su polaridad de manera que el motor gire hacia el lado contrario y se repite el procedimiento varias veces para asegurar la repetitividad de la respuesta.

Si el sistema ha alcanzado su respuesta permanente, es decir su velocidad angular se mantiene constante en respuesta al escalón, significa que el tiempo de (4.18) tiende a infinito por lo que.

$$\lim_{t \rightarrow \infty} w(t) = AK_m - AK_m e^{-\frac{t}{T_m}} = AK_m \quad (4.19)$$

despejando  $K_m$  (4.19) se tiene

$$K_m = \frac{w(\infty)}{A} \quad (4.20)$$

Donde  $w(\infty)$  se obtiene al medir la velocidad angular cuando el sistema ha alcanzado su estado estacionario y  $A$  es el máximo voltaje aplicado.

De (4.18) se observa que la salida, es decir la velocidad angular  $w(t)$  es inicialmente cero y al final su valor es unitario. Si el tiempo  $t = T_m$ , el valor de  $w(t)$  es 63.2% del cambio total de la velocidad angular que experimentará debido a una entrada escalón unitario [22].

Ocupando (4.18), derivándola con respecto al tiempo y evaluándola en  $t = 0$  se obtiene

$$\frac{dw(t)}{dt} = \frac{AK_m}{T_m} e^{-\frac{t}{T_m}} = \frac{AK_m}{T_m} \quad (4.21)$$

Donde el valor resultante es la pendiente de la recta tangente, por lo que la recta es de la forma:

$$y(t) = \frac{AK_m t}{T_m} \quad (4.22)$$

Si se supone que  $t = T_m$ , significa que el punto en donde la velocidad angular  $w(T_m)$  es 63.2% del cambio total de la velocidad, es decir su velocidad esta dada por:

$$w(T_m) = 0.632w(\infty) \quad (4.23)$$

Para establecer el valor de  $T_m$  del sistema se alimenta al motor con 6V, con ayuda de los codificadores de cuadratura se obtiene el incremento de los pulsos contados entre el incremento de los instantes de tiempo (4.24), el paso sucede cada 100ms.

$$w(t) = \frac{\Delta \text{pulsos contados}}{\Delta t} \quad (4.24)$$

Sabiendo que los codificadores requieren 48 cuentas para realizar una revolución se divide los pulsos contados entre 48 y con ello sabremos la velocidad angular por segundo del motor en cuestión.

Con (4.23) y la Tabla A.1 se sabe que para una señal escalón unitario de 6V se tienen 3.27 rad/s, es decir  $w_I(\infty) = 3.27 \text{ rad/s}$ , para el motor izquierdo y  $w_D(\infty) = 3.2 \text{ rad/s}$ , para el motor derecho.

Sustituyendo en (4.20) se obtiene

$$K_{mI} = .545 \frac{\text{rad}}{\text{s} \cdot \text{V}} \quad (4.25)$$



$$K_{mD} = .533 \frac{\text{rad}}{\text{s} \cdot \text{V}}. \quad (4.26)$$

Ocupando (4.23)

$$w(T_{mI}) = 0.632w(\infty) = 2.06\text{rad/s} \quad (4.27)$$

$$w(T_{mD}) = 0.632w(\infty) = 2.02\text{rad/s} \quad (4.28)$$

En el instante de tiempo  $t = T_{mI}$  la velocidad angular es  $w(t) = w(T_{mI}) = 2.06\text{rad/s}$ .

Con (4.24) y los datos obtenidos de los motores que se muestran en la Tabla A.2, se encuentra el valor de  $T_{mI}$  observando que valor de tiempo le corresponde con la velocidad angular  $w(T_{mI}) = 2.06\text{rad/s}$ , en la Figura 4.4 se muestra la velocidad angular del motor izquierdo con incrementos de 10ms, el código se encuentra en el apéndice B.1.

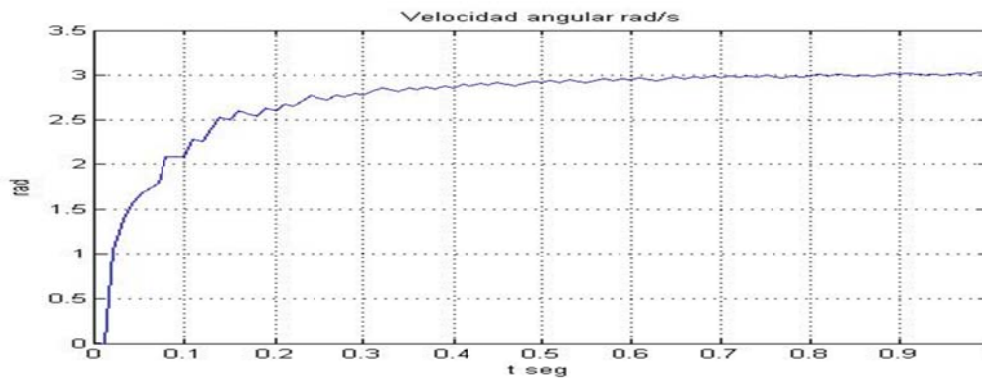


Fig. 4.4 Comportamiento de la velocidad angular del motor izquierdo.

Se observa para  $t = .1\text{s}$ ,  $w(t) \approx \frac{2.06\text{rad}}{\text{s}} \rightarrow T_{mI} = .1$

De manera similar se encuentra el valor de  $w(T_{mD}) = 2.02\text{rad/s}$ , con los datos obtenidos en la Tabla A.3, en la Figura 4.5 se observa el comportamiento de la velocidad angular del motor derecho con incrementos de 10ms, el código ocupado es el mismo para obtener la velocidad angular del motor izquierdo en el apéndice B.1.

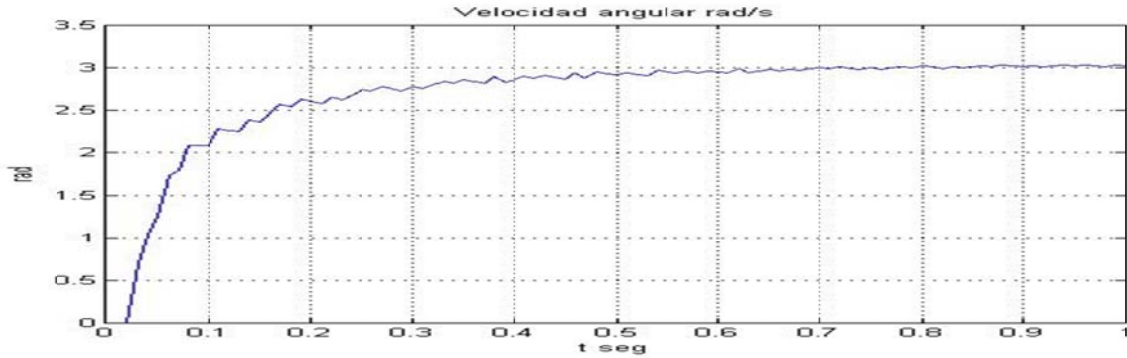


Fig. 4.5 Comportamiento de la velocidad angular del motor derecho.

Se observa para  $t = .09s, w(t) \approx \frac{2.02rad}{s} \rightarrow T_{mD} = 0.09s$

Una vez determinados  $K_m$  y  $T_m$ , mediante el método gráfico de identificación experimental, la función de transferencia que describe el comportamiento del sistema del motor izquierdo es:

$$\frac{w(s)}{V_i(s)} = \frac{K_{mI}}{T_{mI}s+1} = \frac{0.545}{.1s+1} \quad (4.29)$$

Y para el motor derecho es:

$$\frac{w(s)}{V_i(s)} = \frac{K_{mD}}{T_{mD}s+1} = \frac{0.533}{.09s+1} \quad (4.30)$$

Conociendo la función de transferencia de los motores de CD, se puede diseñar un algoritmo de control.

En la Figura 4.6 se muestra el resultado de la función de transferencia (4.29) representativa del motor izquierdo al aplicarle un escalón unitario de 6V.



Figura 4.6 Simulación de la función de transferencia del motor izquierdo.

En la Figura 4.7 se muestra el resultado de la función de transferencia (4.30) representativa del motor derecho al aplicarle un escalón unitario de 6V.

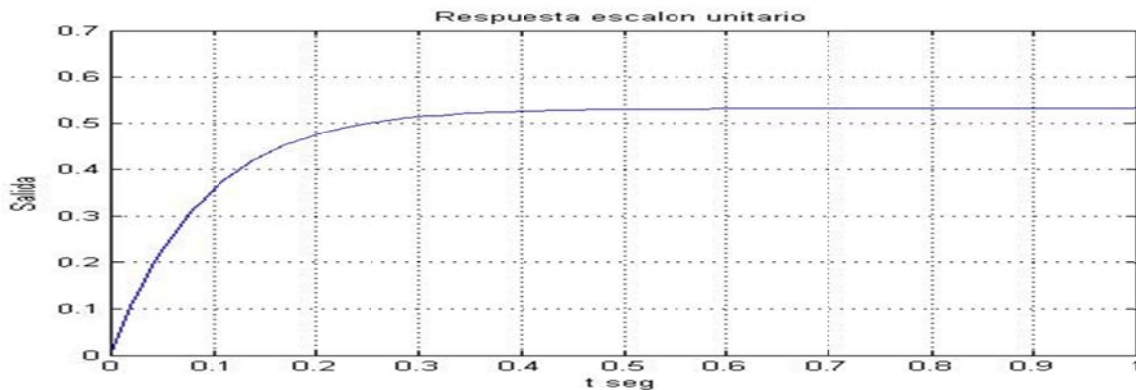


Figura 4.7 Simulación de la función de transferencia del motor derecho.

### 4.3. Diseño del Controlador

Existen muchos modelos y teorías para el control automático, el controlador Proporcional, Integral, Derivativo (PID), es el más utilizado con fines académicos e industriales, ya que es posible aplicarlos en la mayoría de los sistemas de control [31].

La estructura estándar para un controlador PID es la siguiente

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (4.31)$$

la cual puede expresarse como

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (4.32)$$

y al aplicar la transformada de Laplace se tiene

$$U(s) = K_p E(s) + \frac{K_i}{s} E(s) + K_d s E(s) \quad (4.33)$$

donde  $U$  es la señal de control,  $E$  es el error y los parámetros del controlador  $K_p, K_i, K_d$  son las ganancias proporcional, integral y derivativa respectivamente.

Mediante la correcta selección de estos valores es posible ajustar la respuesta del sistema a controlar. El método más conocido para la sintonización de las ganancias son las reglas Ziegler-Nichols, estas reglas se basan en las características de la respuesta transitoria de la planta dada y con ello la obtención de las ganancias  $K_p, K_i, K_d$ .

En la Figura 4.8 se muestra el diagrama a bloques de control de un motor de cd, la variable de entrada es una velocidad de referencia  $w_r(t)$  y de salida una velocidad final  $w_f(t)$ , con lo anterior podríamos hacer un control de velocidad, sin embargo es posible realizar un control de posición si consideramos que la entrada es una distancia de referencia  $d_r(t)$  y de salida una distancia final  $d_f(t)$ , entre estas variables el cambio radica en el tratamiento de los pulsos de los codificadores de cuadratura,  $E$  es la señal de error,  $U$  la señal control,  $K_m$  y  $T_m$  los parámetros obtenidos del proceso de identificación del motor de CD y  $V_m$  es el voltaje de alimentación del motor de CD.

Es necesario sintonizar al controlador PID con respecto a la planta obtenida en (4.29) y (4.30), con ello saber los valores de  $K_p, K_i, K_d$ , una vez que se sepan los valores se les aplica una amplificación de la señal con lo cual se tendrá la suficiente potencia para alimentar a la planta una vez hecho esto, el codificador registrara los cambios en la distancia y con ello se obtendrá el valor del error que a medida que el tiempo aumente el error debe disminuir.

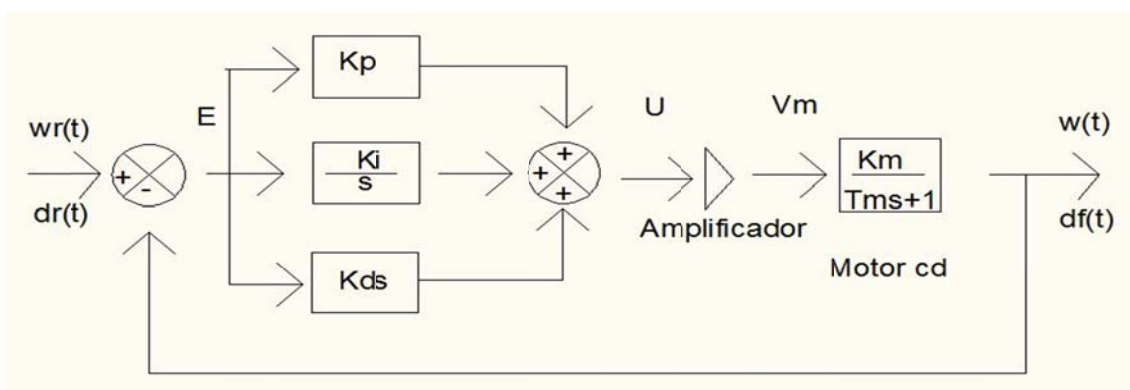


Fig. 4.8 Diagrama a bloques del control de un motor de CD.

De (4.33) se puede expresar un PID como:

$$\frac{K_d s^2 + K_p s + K_i}{s} \quad (4.34)$$

En base a la Figura 4.8 podemos encontrar el sistema de control a simular en Matlab dado por la Figura 4.9.

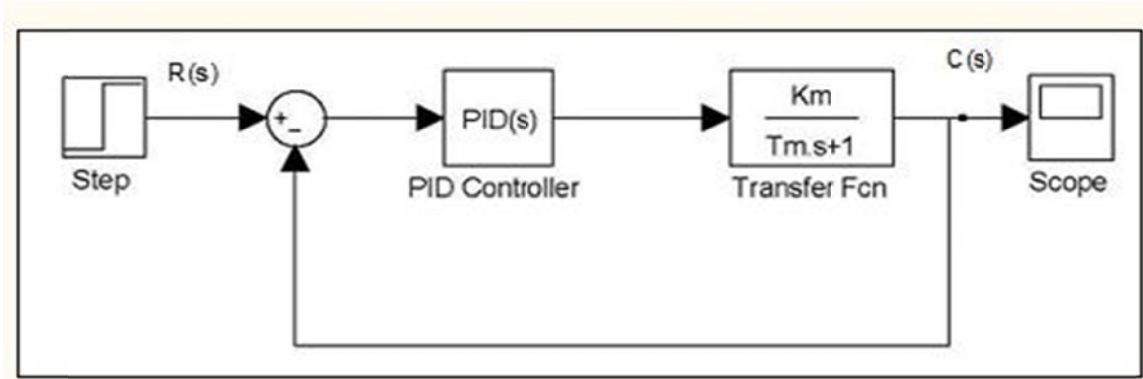


Figura 4.9 Diagrama a bloques del sistema a controlar.

Ocupando las reglas de los diagramas a bloques y las ecuaciones (4.29) y (4.34) resulta:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{K_d K_m s^2 + K_p K_m s + K_i K_m}{(K_d K_m + T_m) s^2 + (K_p K_m + 1) s + K_i K_m} \quad (4.35)$$

Para comprobar cual controlador o controladores son adecuados se simulan los comportamientos del controlador P, un controlador I, un controlador PD, un controlador PI y un controlador PID.

### 4.3.1. Controlador P

De la función de transferencia (4.35), se hacen cero los términos que contengan  $K_d$  y  $K_i$  con lo cual resulta:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{K_p K_m}{T_m s + K_p K_m + 1} \quad (4.36)$$

Esta función de transferencia es simulada en Matlab, el código se encuentra en el Apéndice B.2.

El programa cambia el valor de  $K_p$  en el intervalo [0,5] con incrementos de 0.05 al aplicarle una señal escalón unitario con amplitud 1 de entrada. Se compara el valor de la sobreelongación máxima con la combinación de  $K_p$  que satisfaga algún criterio de sobreelongación, en este caso que sea menor del 5%, es decir que la respuesta del sistema al escalón unitario no sobrepase la amplitud de 1.05.

La simulación comprueba que no es adecuado este controlador para esta función de transferencia.

### 4.3.2. Controlador I

De la función de transferencia (4.35), se hacen cero los términos que contengan  $K_d$  y  $K_p$  con lo cual resulta:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{K_i K_m}{T_m s^2 + s + K_i K_m} \quad (4.37)$$

Esta función de transferencia es simulada en Matlab con el código que se encuentra en el Apéndice B.3.

El programa cambia el valor de  $K_i$  en el intervalo [0,5] con incrementos de 0.05 con una señal escalón unitario de entrada. Se compara el valor de la sobreelongación máxima con la combinación de  $K_i$  que satisfaga algún criterio de sobreelongación, en este caso que sea menor del 5%, sin subamortiguamiento.

El programa nos arroja como ganancia  $K_i$ , el valor

$$K_i = 5 \quad (4.38)$$

Con lo cual (4.35) se puede expresar como:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{2.725}{0.1s^2 + s + 2.725} \quad (4.39)$$

Al aplicarle un escalón unitario a la función de transferencia compensada con el controlador I (4.39) se observa su comportamiento en la Figura 4.10 (izquierda), comparándola con la respuesta de la función de transferencia sin compensar (4.29) en

la Figura 4.10 (derecha), el código se encuentra en el Apéndice B.4, cabe mencionar que esto es válido para el motor del lado izquierdo.

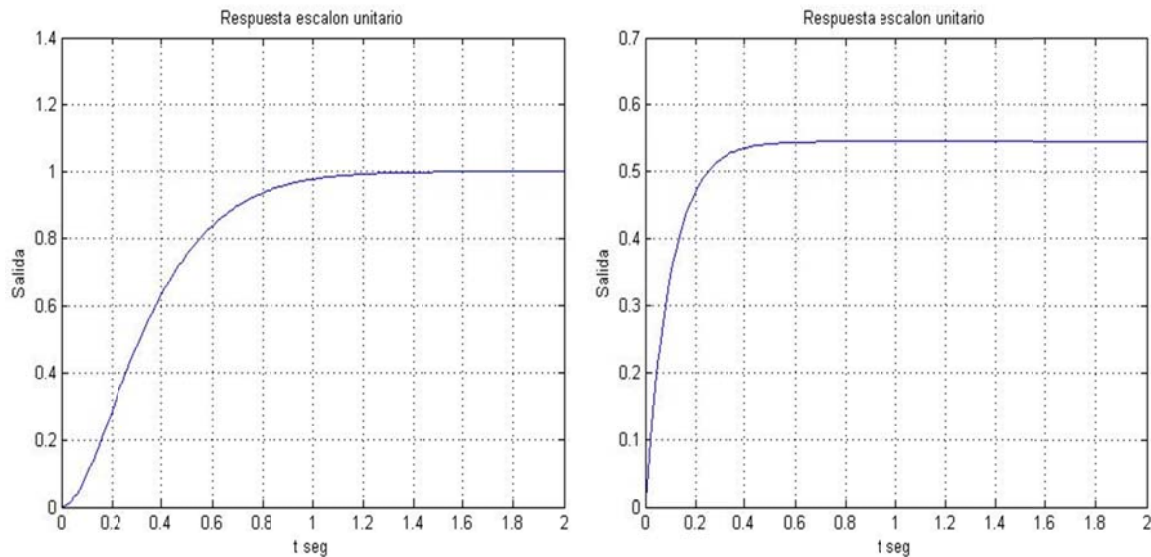


Fig. 4.10 Comparación de las respuesta con controlador I y sin él.

De la Figura 4.10 se observa que el transitorio de la respuesta de la función de transferencia con el controlador I, es mayor, pero en el estado estable no se percibe ningún error a diferencia de la respuesta del motor sin controlador I que responde rápido, pero no llega al valor de la señal aplicada.

La simulación indica que este controlador responde adecuadamente, pero su respuesta es lenta.

### 4.3.3. Controlador PD

De la función de transferencia (4.35), se hace cero el término  $K_i$  con lo cual resulta:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{K_d K_m s + K_p K_m}{(K_d K_m + T_m) s + K_p K_m + 1} \quad (4.40)$$

Esta función de transferencia es simulada en Matlab, el código se encuentra en el Apéndice B.5.

El programa cambia el valor de  $K_p$  y  $K_d$  en el intervalo [0,5] con incrementos de 0.05 con una señal escalón unitario de entrada. Se compara el valor de la sobreelongación máxima con la combinación de  $K_p$  y  $K_d$  que satisfaga algún criterio de sobreelongación, en este caso que sea menor del 5%, sin subamortiguamiento.

La simulación comprueba que no es adecuado este controlador para esta función de transferencia.

#### 4.3.4. Controlador PI

De la función de transferencia (4.35), se hace cero el término  $K_d$  con lo cual resulta:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{K_p K_m s + K_i K_m}{T_m s^2 + (K_p K_m + 1)s + K_i K_m} \quad (4.41)$$

Esta función de transferencia es simulada en Matlab, el código se encuentra en el Apéndice B.6.

El programa cambia el valor de  $K_p$  y  $K_i$  en el intervalo [0,5] con incrementos de 0.05 con una señal escalón unitario de entrada. Se compara el valor de la sobreelongación máxima con la combinación de  $K_p$  y  $K_i$  que satisfaga algún criterio de sobreelongación, en este caso que sea menor del 5%, sin subamortiguamiento.

El programa nos arroja como ganancias  $K_p$  y  $K_i$ , los valores

$$K_p = 0.1 \quad (4.42)$$

$$K_i = 4.45 \quad (4.43)$$

Con lo cual (4.35) se puede expresar como:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{0.0545s + 2.425}{0.1s^2 + 1.055s + 2.425} \quad (4.44)$$

Al aplicarle un escalón unitario a la función de transferencia compensada con el controlador PI (4.44) se observa su comportamiento en la Figura 4.11 (izquierda), comparándola con la respuesta de la función de transferencia sin compensar (4.29) en



la Figura 4.11 (derecha), el código se encuentra en el Apéndice B.3, cabe mencionar que esto es válido para el motor del lado izquierdo.

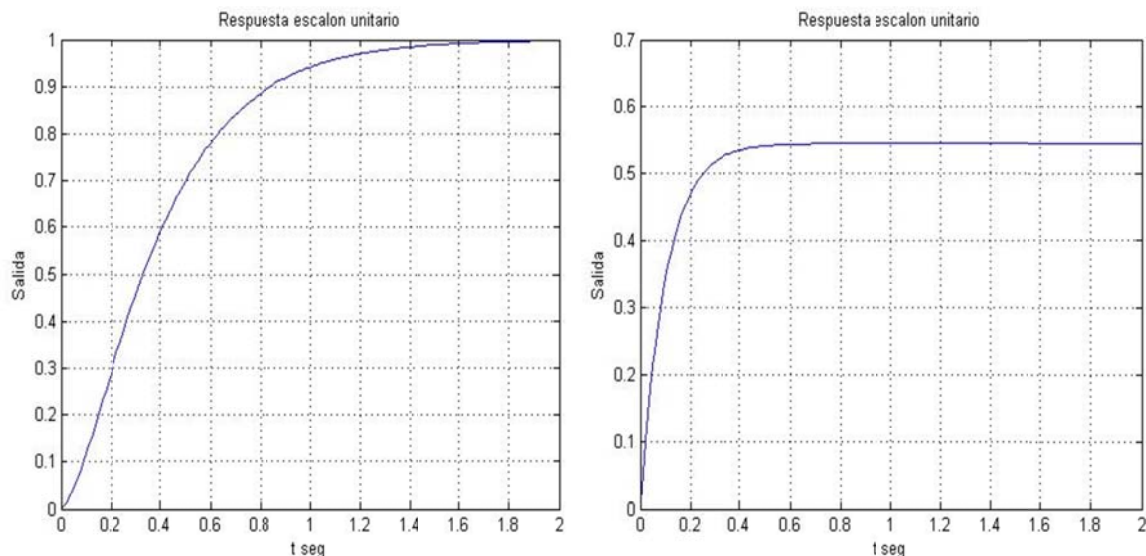


Fig. 4.11 Comparación de las respuesta con controlador PI y sin él.

Se observa que el transitorio de la respuesta de la función de transferencia del controlador PI con respecto al controlador I es más rápido, alcanza en el estado estable a la señal aplicada a diferencia del motor que responde rápido, pero no llega al valor de la señal aplicada.

La simulación indica que este controlador tiene un mejor desempeño que el controlador I.

#### 4.3.5. Controlador PID

Se toma a la función de transferencia (4.35), se simula en Matlab, el código se encuentra en el apéndice B.7.

El programa cambia el valor de  $K_p$ ,  $K_i$  y  $K_d$  en el intervalo [0,5] con incrementos de 0.05 con una señal escalón unitario de entrada. Se compara el valor de la sobreelongación

máxima con la combinación de  $K_p, K_i$  y  $K_d$  que satisfaga algún criterio de sobreelongación, en este caso que sea menor del 5%, sin subamortiguamiento.

El programa nos arroja como ganancias  $K_p, K_i$  y  $K_d$ , los valores

$$K_p = 0.1 \quad (4.45)$$

$$K_i = 4.9 \quad (4.46)$$

$$K_d = 0.1 \quad (4.47)$$

Con lo cual (4.35) se puede expresar como:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{0.0545s^2+0.0545s+2.671}{0.1545s^2+1.055s+2.671} \quad (4.48)$$

Al aplicarle un escalón unitario a la función de transferencia compensada con el controlador PID (4.48) se observa su comportamiento en la Figura 4.12 (izquierda), comparándola con la respuesta de la función de transferencia sin compensar (4.29) en la Figura 4.12 (derecha), el código se encuentra en el Apéndice B.3, cabe mencionar que esto es válido para el motor del lado izquierdo.

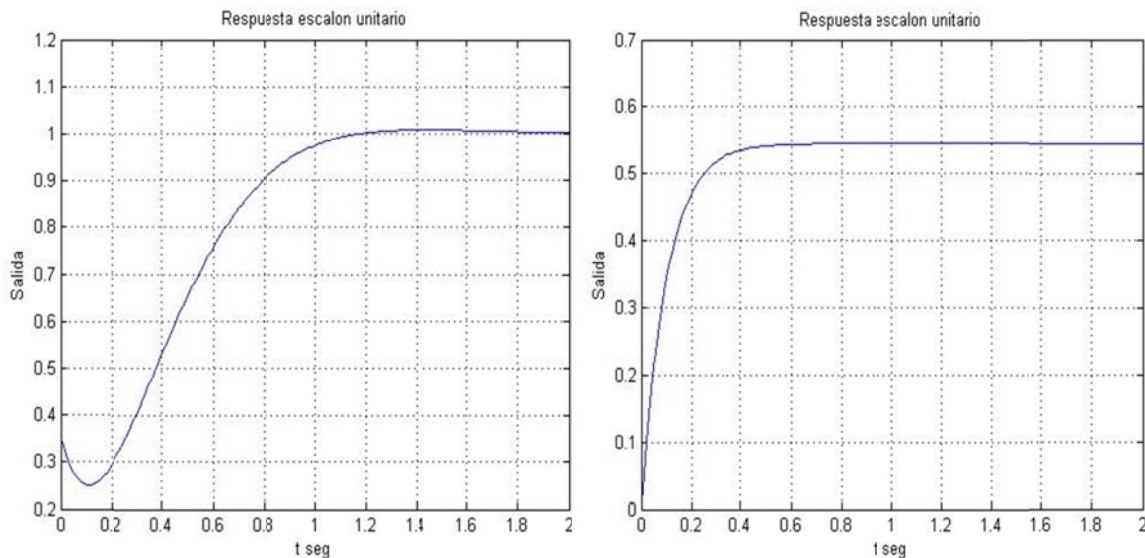


Fig. 4.12 Comparación de las respuesta con controlador PID y sin él.

Se observa que el transitorio de la respuesta de la función de transferencia del controlador PID con respecto al controlador PI es más rápido, pero presenta una oscilación en los primeros instantes y un pequeño sobreamortiguamiento.

Se tiene que el controlador I, controlador PI y el controlador PID con diversas ganancias pueden ser los que controlen a la planta, se ocupara el PID ya que una vez que se tenga implementado el PID, tener los otros controladores se limita a eliminar términos y cambiar las ganancias en el programa.

# 5 Pruebas y resultados experimentales

---

En el capítulo 4 se encontró la función de transferencia de los motores y la sintonización de las ganancias de diversos controladores, con ello hemos obtenido la respuesta del sistema a una determinada entrada, pero para la implementación práctica del sistema se requiere programar los resultados antes obtenidos para comprobarlos y utilizarlos en el móvil.

El programa que se implementó para la tarjeta de desarrollo Arduino mega se centró en utilizar un menú desplegable por medio de un software de comunicación serial, por ejemplo X-CTU, en este menú se tienen opciones tales como la prueba de los componentes independientes, la obtención de los datos ocupados para la determinación de la función de transferencia de los motores y la realización de las rutinas de trayectoria implementadas.

En la Figura 5.1 se muestra en la terminal X-CTU el menú desplegable de opciones para el robot móvil, se observa que dispone de 18 opciones, al oprimir la letra del teclado que corresponde a alguna opción entra en funcionamiento.

El Xbee de transmisión que se encuentra en el móvil junto con el Xbee de recepción que se encuentra en la PC establece la comunicación entre el móvil y la PC por medio de la terminal X-CTU.

El menú se despliega automáticamente después de responder a la solicitud enviada por el usuario, en X-CTU las palabras marcadas en rojo indican que han sido recibidas por el PC por medio del Xbee y las letras en azul indican que el usuario ha enviado una solicitud por medio del teclado al Xbee.

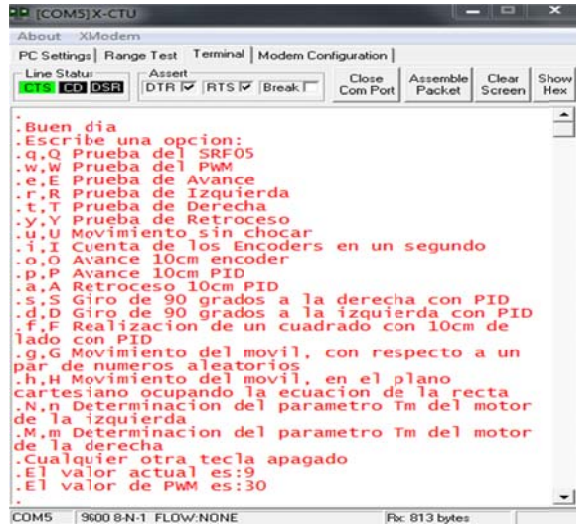


Fig. 5.1 Menú desplegable en terminal X-CTU.

## 5.1. Implementación de los sensores

La opción Q del menú prueba el funcionamiento del sensor de ultrasonido SRF05, el cual nos indica la distancia existente entre el móvil y un objeto detectado por dicho sensor. Este sensor dispone de dos modos de operación: el primero ocupa dos pines para la transmisión y recepción de pulsos que el sensor interpreta como la distancia recorrida. La segunda ocupa un pin para la transmisión y recepción de un tren de pulsos, en la Figura 5.2 se observa el tren de pulsos que fue programado para obtener la distancia del móvil con respecto a un objeto.

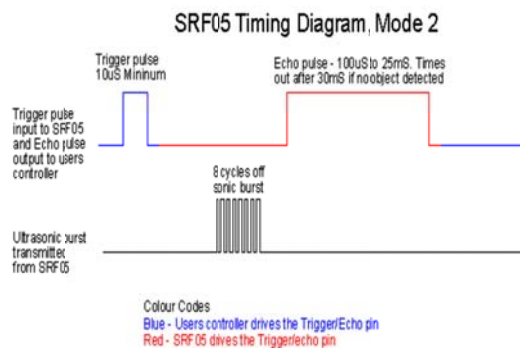


Fig. 5.2 Tren de pulsos programados

En la Figura 5.3 se muestra la petición de medición de distancia por parte del usuario y su posterior resultado.

```
Cualquier otra tecla apagado
El valor actual es:9
El valor de PWM es:30
q10cm
```

Fig. 5.3 Distancia del móvil respecto a un objeto

La opción W del menú realiza una prueba del funcionamiento del pulso con modulación (PWM) en esta prueba se entra a dos ciclos el primero inicializa a los motores con un ciclo útil de 0% con incrementos de 1% cada 100ms, el segundo inicia con un ciclo útil del 100% con decrementos de 1% cada 100ms, en la Figura 5.4 se muestra algunos valores de dicha prueba.

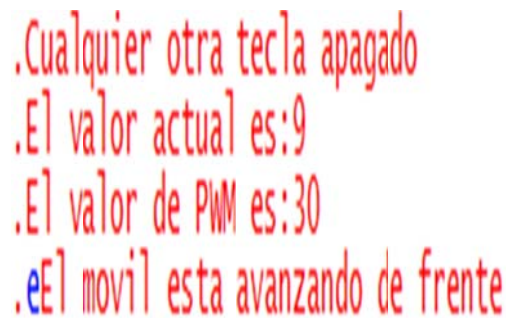
```
... 6% Enable1
... 6% Enable2
... 7% Enable1
... 7% Enable2
... 8% Enable1
... 8% Enable2
... 9% Enable1
... 9% Enable2
... 0% Enable1
... 0% Enable2
... 1% Enable1
... 1% Enable2
... 2% Enable1
... 2% Enable2
... 3% Enable1
... 3% Enable2
... 4% Enable1
... 4% Enable2
... 5% Enable1
... 5% Enable2
```

Fig. 5.4 Prueba del pwm

Recordando que la forma de control más utilizada para un motor de CD es por medio de la variación del voltaje de entrada a dicho motor. El PWM cambia el valor del ciclo útil, es decir modifica el tiempo en el cual el pulso se encuentra apagado y encendido, con lo cual cambia el voltaje promedio aplicado, con ello se varía el voltaje que llega al motor y por consiguiente su respuesta.

Las cuatro opciones siguientes consiste en controlar al móvil de forma manual, es decir el usuario controla los movimientos del robot móvil de forma similar a un radiocontrol.

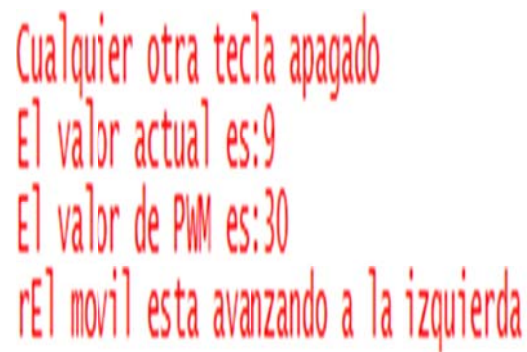
La opción E del menú controla el avance del móvil al frente hasta que otra tecla sea oprimida para detenerlo o cambiar la orden, cabe mencionar que el valor por defecto del ciclo útil es del 30%, en la Figura 5.5 se muestra esta opción.



```
.Cualquier otra tecla apagado
.El valor actual es:9
.El valor de PWM es:30
.eEl movil esta avanzando de frente
```

Fig. 5.5 Prueba de avance

La opción R del menú consiste en que el móvil avance hacia la izquierda hasta que otra tecla sea oprimida para detenerlo, en la Figura 5.6 muestra esta opción.



```
Cualquier otra tecla apagado
El valor actual es:9
El valor de PWM es:30
rEl movil esta avanzando a la izquierda
```

Fig. 5.6 Prueba hacia la izquierda

La opción T del menú consiste en que el móvil avance hacia la derecha hasta que otra tecla sea oprimida para detenerlo, en la Figura 5.7 muestra esta opción.

.Cualquier otra tecla apagado  
.El valor actual es:9  
.El valor de PWM es:30  
.tEl móvil esta avanzando a la derecha

Fig. 5.7 Prueba hacia la derecha

La opción Y del menú consiste en que el móvil retroceda hasta que otra tecla sea oprimida para detenerlo, en la Figura 5.8 muestra esta opción.

.Cualquier otra tecla apagado  
.El valor actual es:9  
.El valor de PWM es:30  
.yEl móvil esta avanzando hacia atras

Fig. 5.8 Prueba retroceso

La opción U del menú ocasiona que el robot móvil realice movimientos utilizando la información obtenida por el sensor de distancia SRF05, de acuerdo al valor de la distancia el móvil puede realizar alguna de las cuatro opciones programadas:

Si el objeto se encuentra a una distancia mayor de 20cm el móvil avanza, si el objeto se encuentra entre 10 y 20 cm el móvil realiza un movimiento hacia la izquierda, si la distancia está en el rango de 3 a 10 cm el móvil realiza un movimiento hacia atrás, sino se encuentra en alguna de esas opciones el móvil se detiene.



La opción I del menú realiza una prueba del funcionamiento de los codificadores de cuadratura, la cual consiste que el motor de la izquierda funcione durante un segundo, después de ese tiempo se detiene y muestra el valor capturado por parte de un contador que esta inicializado en cero cuando estaba detenido el motor y durante su funcionamiento contabiliza cada cambio en la señal de entrada del codificador de cuadratura de la izquierda, se muestra el valor final contabilizado y procede de la misma forma con el motor de la derecha, con esto se puede corroborar el funcionamiento correcto de los codificadores de cuadratura, en la Figura 5.9 se muestra lo antes mencionado.

```
.Cualquier otra tecla apagado
.El valor actual es:9
.El valor de PWM es:30
.i
.0
.
.101
.
.0
.
.90
```

Fig. 5.9 Prueba de los codificadores de cuadratura

La opción O del menú realiza un avance de 10cm, este avance es contabilizado gracias a las cuentas de los codificadores de cuadratura, debido a que las llantas ocupadas contienen un estriado, el cual sirve para contabilizar 12 cambios si contabiliza solo los flancos de subida, 24 cambios si contabiliza tanto los flancos de subida, así como los de bajada o hasta 48 cambios si se ocupan los dos sensores que contiene el codificador de cuadratura, estos contabilizan los flancos de subida y bajada.

Sabiendo que el diámetro de las llantas es de 13.08 cm, es decir que los cambios se producen cada 1.09cm para los 12 cambios, 0.545cm para los 24 cambios y hasta 0.272cm para 48 cambios, por lo que el mínimo avance debe de ser 0.272cm para que el codificador de cuadratura pueda detectarlo y contabilizarlo.

Para poder contabilizar dichos cambios se ocuparon cuatro interrupciones en la tarjeta de desarrollo Arduino mega que dispone de seis interrupciones para contabilizar 48 cambios por revolución, en la tarjeta de desarrollo Arduino uno solo dispone de dos interrupciones por lo que en la tarjeta de desarrollo Arduino uno se pueden contabilizar hasta 24 cambios por revolución únicamente, es decir el avance mínimo es de 0.545cm para una tarjeta Arduino uno.

Se realiza una regla de tres para avanzar cualquier distancia, el desplazamiento debe ser igual para ambos motores, debido a esto se tiene tres opciones con el fin de asegurar un desplazamiento idéntico: Se tienen dos distancias una asociada al motor 1 y otra al motor 2, si el valor de ambas distancias es menor que 10 cm el móvil avanza, si solo el valor de la distancia 1 es igual a 10cm el móvil se desplaza a la izquierda, si el valor de la distancia 2 es igual a 10cm el móvil se desplaza a la derecha, en la Figura 5.10 se muestra los valores obtenidos por el móvil.

```
.Cualquier otra tecla apagado
.El valor actual es:9
.El valor de PWM es:30
.o0
.0
.Cuentas del encoder1
.39
.Cuentas de encoder2
.37
.distancia1
.10.06
.distancia2
.10.06
```

Fig. 5.10 Avance de 10cm con los codificadores

## 5.2. Implementación del control PID

La opción P del menú realiza un avance de 10cm utilizando un controlador PID, este controlador cambia el valor del PWM aplicado a los motores y con ello controla el avance con respecto a una distancia deseada que en este caso es de 10cm y el error existente de la distancia actual, tal y como se mostró el diagrama a bloques de la Figura 4.8.

Para la implementación del controlador PID se utilizó el diagrama de flujo de la Figura 5.11 [32].

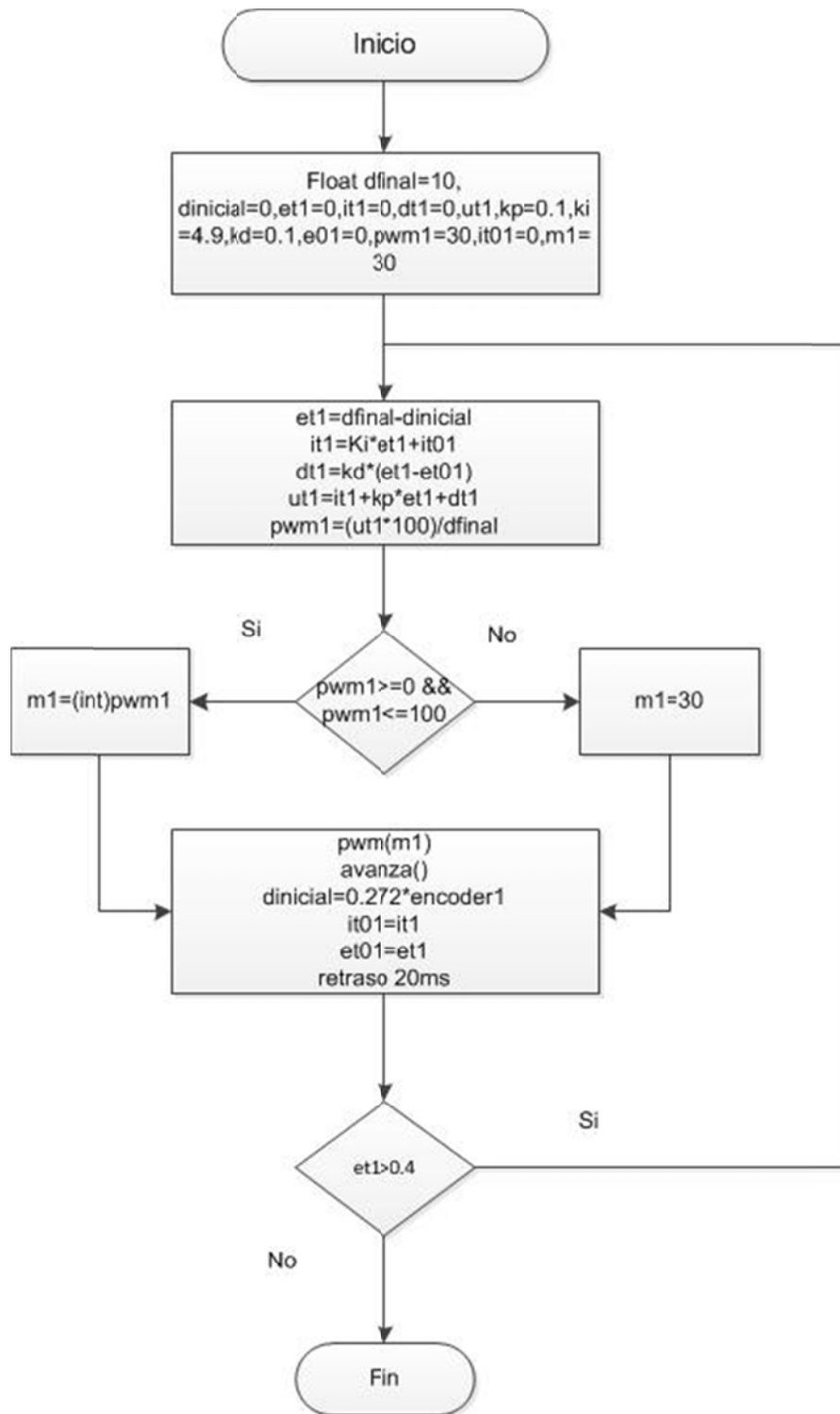


Fig. 5.11 Diagrama de flujo del PID

Este diagrama de flujo implementa un control PID de la distancia que el usuario asigne (10cm), para el avance del motor de la izquierda o m1. Se calcula el error y con él se obtienen los valores de los términos integral, derivativo y proporcional, se suman estos términos obteniendo la señal de control  $u(t)$  esta señal es acondicionada para su transferencia a los motores. Si la señal se encuentra en el rango de 0 a 100 se utiliza una subrutina que convierte ese porcentaje en una señal PWM para su salida hacia el puente H y por ende a los motores, si no se encuentra el valor en el rango de 0 a 100 el porcentaje de la señal PWM es por defecto 30. Se calcula el nuevo valor de la distancia inicial dependiente del conteo que se encuentre en el encoder, se reasigna el valor del error inicial y el término integral para la siguiente iteración, esta iteración espera 20 ms antes de llevarse a cabo si es que la condición del error es mayor a .4cm sino lo es termina el programa. La Figura 5.12 muestra un avance de 10cm y el error existente en cada uno de los codificadores conforme varia el tiempo.

```
.Cualquier otra tecla el movil esta apagado
.El valor actual es:9
.El valor de PWM es:30
.p
.El error M1 es:
.0.21
.La salida del PID M1 es:
.10.61
.
.El error M2 es:
.0.21
.La salida del PID M2 es:
.10.06
```

Fig. 5.12 Avance de 10cm ocupando un PID.

La opción A del menú realiza un retroceso de 10cm utilizando un controlador PID, este controlador cambia el valor del pwm aplicado a los motores y con ello controla el retroceso con respecto a una distancia deseada y el error existente de la distancia actual con el mismo algoritmo que se mostró en la Figura 5.11, la Figura 5.13 muestra un retroceso de 10cm y el error existente en cada uno de los codificadores.

```

.Cualquier otra tecla el movil esta apagado
.El valor actual es:9
.El valor de PWM es:30
.a
.El error M1 es:
.1.30
.La salida del PID M1 es:
.8.98

.El error M2 es:
.0.21
.La salida del PID M2 es:
.9.79

```

Fig. 5.13 Retroceso de 10cm ocupando un PID.

La opción S del menú realiza un giro de 90 grados a la derecha utilizando un controlador PID, este controlador cambia el valor del PWM aplicado a los motores y con ello controla el giro de 90 grados a la derecha con respecto a una distancia deseada y el error existente de la distancia actual.

Al tener un movimiento circular y requerir medir la distancia a lo largo de la circunferencia a partir del origen se tiene que la longitud de un arco de radio r es [29]:

$$s = r\theta \quad (5.1)$$

Modificando (5.1) para su utilización en grados se tiene.

$$s = \frac{r*\theta*\pi}{360^\circ} \quad (5.2)$$

Del diseño del robot móvil se observa que el radio es de 10.8 cm, con lo cual la longitud del arco es aproximadamente de 8.47cm esta distancia garantiza un giro de 90 grados, en la Figura 5.14 se observa el resultado de dicha opción.

```

.Cualquier otra tecla el movil esta apagado
.El valor actual es:9
.El valor de PWM es:30
.S
.El error M1 es:
.0.32
.La salida del PID M1 es:
.8.43

```

Fig. 5.14 Giro 90 grados a la derecha.

La opción D del menú realiza un giro de 90 grados a la izquierda utilizando un controlador PID, este controlador cambia el valor del PWM aplicado a los motores y con ello controla el giro de 90 grados a la izquierda con respecto a una distancia deseada y el error existente de la distancia actual.

La longitud del arco es aproximadamente de 8.47 cm esta distancia garantiza un giro de 90 grados, en la Figura 5.15 se observa el resultado de dicha opción.

```
.Cualquier otra tecla el movil esta apagado
.El valor actual es:9
.El valor de PWM es:30
.d
.El error M2 es:
.0.05
.La salida del PID M2 es:
.8.43
```

Fig. 5.15 Giro 90 grados a la izquierda.

### 5.3. Implementación de trayectorias

La opción F del menú realiza un cuadrado desde un punto de origen avanza 10cm, gira 90 grados a la derecha, avanza 10cm gira 90 grados hacia la derecha, avanza 10cm gira 90 grados y avanza 10cm para llegar aproximadamente al punto de origen, en la Figura 5.16 se muestra la trayectoria antes mencionada.

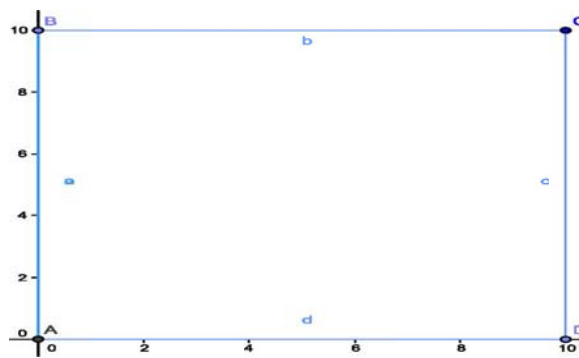


Fig. 5.16 Cuadrado implementado por el móvil

La Figura 5.17 muestra parte de los resultados de la implementación del cuadrado. Para realizar esta rutina se manda a llamar a las rutinas PID avanza y PID derecha, la primera para el avance de 10cm y la segunda para el giro de 90 grados.

```
.El error M1 es:  
.4.29  
.La salida del PID M1 es:  
.5.98  
.  
.El error M2 es:  
-0.34  
.La salida del PID M2 es:|  
.10.61  
.  
.  
.El error M1 es:  
.0.32  
.La salida del PID M1 es:  
.8.16  
.  
.  
.El error M1 es:  
.4.29  
.La salida del PID M1 es:  
.5.98  
.  
.El error M2 es:  
-0.21  
.La salida del PID M2 es:  
.10.34
```

Fig. 5.17 Implementación del cuadrado.

La opción G del menú consiste en la generación de un punto aleatorio entre 0 a 100 centímetros y con él se crea el par ordenado  $(x_2, y_2)$ , el móvil se considera que se encuentra en el origen del plano cartesiano  $(x_1, y_1) = (0,0)$ , por lo que siempre el móvil se encontrara en el primer cuadrante del plano cartesiano.

Teniendo los dos puntos en cuestión se procede a calcular la distancia con respecto a la coordenada x y a la coordenada y, se considera que la orientación del móvil es paralelo a la coordenada y, con lo cual se procede a avanzar la distancia obtenida para la coordenada y, una vez alcanzada se gira 90 grados a la derecha y se procede a avanzar hasta alcanzar el valor de la coordenada x, en la Figura 5.18 se muestra la trayectoria antes mencionada.

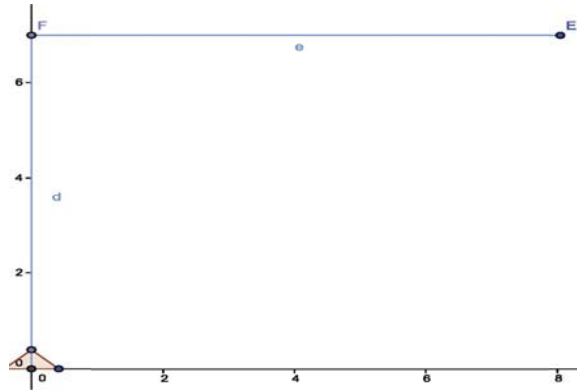


Fig. 5.18 Trayectoria implementada por el móvil.

La Figura 5.19 muestra los resultados obtenidos al implementar esta trayectoria, se obtiene el punto aleatorio en el plano cartesiano cuyo valor oscila de 0 a 100, se calculan las distancias y se avanza de acuerdo a la coordenada y, una vez alcanzada la distancia gira 90 grados a la derecha y avanza hasta llegar al valor de la coordenada x.

```

.g89.00
.21.00
.El error M1 es:
.8.76
.La salida del PID M1 es:
.12.51
.El error M2 es:
.0.33
.La salida del PID M2 es:
.20.94
.
.El error M1 es:
.0.32
.La salida del PID M1 es:
.8.43
.
.El error M1 es:
.34.06
.La salida del PID M1 es:
.54.94
.El error M2 es:
.0.06
.La salida del PID M2 es:
.89.22

```

Fig. 5.19 Implementación de la trayectoria 1.

La opción H del menú al igual que la anterior genera un punto aleatorio entre 0 y 100 y con ello se crea el par ordenado  $(x_2, y_2)$ , el lugar donde se encuentra el móvil se considera el origen del plano cartesiano  $(x_1, y_1) = (0,0)$ , la orientación del móvil es paralela a la coordenada x.

Teniendo los dos puntos en cuestión se obtiene el tercer punto  $(x_2, 0)$ , con dichos puntos se crea un triángulo rectángulo, se calcula el cateto adyacente, el cateto



opuesto y la hipotenusa de dicho triángulo rectángulo, la Figura 5.20 se muestra el escenario antes descrito.

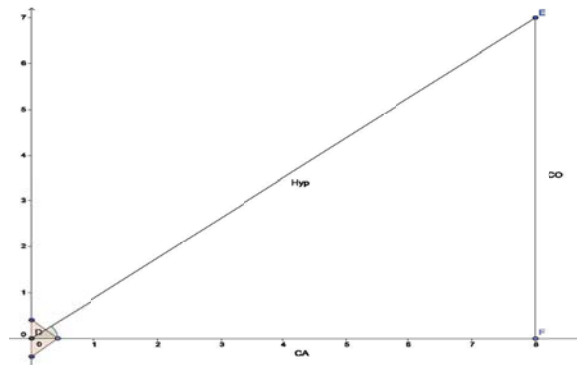


Fig. 5.20 Escenario de la trayectoria 2.

La trayectoria 2 consiste en calcular el ángulo  $\theta$ , el móvil gira a la izquierda hasta el valor del ángulo  $\theta$  y procede a avanzar el valor de la hipotenusa, se utiliza la función trigonométrica seno para un triángulo rectángulo que está definida como:

$$\text{Seno } \theta = \frac{co}{hyp} \quad (5.3)$$

Se despeja el valor del ángulo de (5.3)

$$\theta = \text{Arcseno} \left( \frac{co}{hyp} \right) \quad (5.4)$$

En el lenguaje de programación de Arduino llamado Processing no contempla las funciones trigonométricas inversas por lo que se utilizó la serie de Taylor para la función trigonométrica inversa seno es decir:

$$\text{Arcseno } x = \sum_{n=0}^{\infty} \frac{(2n)!}{4^n (n!)^2 (2n+1)} x^{2n+1} \quad |x| < 1 \quad (5.5)$$

Una vez obtenido el ángulo con la serie de Taylor, se convierte a grados ya que se encuentra en radianes, se realiza el giro de acuerdo a los grados obtenidos, para un posterior avance por la hipotenusa, en la Figura 5.21 se muestra los valores obtenidos de la implementación de la trayectoria 2.

```

h
M1 valor de x2 es:
M3 3.00
M1 valor de y2 es:
M3 8.00
M1 valor de CA es:
M3 3.00
M1 valor de C0 es:
M3 8.00
M1 valor de HY es:
M3 16.24
M1 valor de Sin(angulo) es:
M3 7.77
M1 valor de x2=hy*cos(angulo) es:
M3 3.35
M1 valor de y2=hy*sin(angulo) es:
M3 27.70
M1 49.85
M2 El error M2 es:
M3 0.35
M1 La salida del PID M2 es:
M3 5.17
M1
M2 El error M1 es:
M3 1.96
M1 La salida del PID M1 es:
M3 35.63

```

Fig. 5.21 Implementación de la trayectoria 2.

La opción N del menú alimenta al motor de la izquierda por 1 segundo con un ciclo útil del 100% contabilizando el cambio de entrada lógica por parte de los sensores del codificador de cuadratura y con ello se obtiene el desplazamiento y/o la velocidad angular por parte del motor de la izquierda, para la identificación del valor de  $T_{mI}$  ocupado en la función de transferencia del sistema, en la Figura 5.22 se muestra parte de los valores contabilizados cada 10ms por el codificador de cuadratura.

```

n
0
1000
10
0
20
0
30
0
40
1
50
2
60
2

```

Fig. 5.22 Cambios del codificador de la izquierda.

La opción M del menú alimenta al motor de la derecha por 1 segundo con un ciclo útil del 100% contabilizando el cambio de entrada lógica por parte de los sensores del codificador de cuadratura y con ello se obtiene el desplazamiento y/o la velocidad angular por parte del motor de la derecha, para la identificación del valor  $T_{mD}$  ocupado

en la función de transferencia del sistema, en la Figura 5.23 se muestra parte de los valores contabilizados cada 10ms por el codificador de cuadratura.

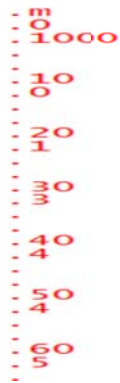


Fig. 5.23 Cambios del codificador de la derecha.

# 6 Conclusiones

---

Se diseñó, construyó y controló un robot móvil diferencial que servirá para el desarrollo de pruebas tanto de control, como la programación de trayectorias con ayuda de algoritmos tales como búsqueda por anchura, búsqueda por profundidad, controladores lineales, no lineales, difusos, arboles, redes neuronales, redes bayesianas, etc.

Se utilizó los desarrollos matemáticos de los modelos cinemáticos y dinámicos de los robots móviles diferenciales, así como el de los motores de CD para tener una base sólida de la cual partir en el desarrollo del móvil.

Se realizó una serie de planos en los cuales se plasmó los diversos componentes que el móvil debería de contener para su implementación, al mismo tiempo se detectaron a tiempo varios errores en el diseño, así como consideraciones no previstas al inicio del proyecto y gracias a la herramienta CAD se observó que el centro de gravedad se encontrara adecuado para la estabilidad del móvil.

La construcción del vehículo contemplo la elección de actuadores y sensores para la adquisición de las señales necesarias para un control de trayectoria.

Se utilizó la técnica de identificación grafica para obtener los parámetros de la función de transferencia que representa a los motores de cd, con el posterior uso de algunos controladores para la sintonización de las ganancias.

Se simularon algunos controladores como P, I, PD, PI, y PID con la planta obtenida por los motores de CD con la consecuente eliminación y elección del mejor de ellos.

Las pruebas realizadas al móvil comprueban que el codificador de cuadratura con su baja resolución produce mucho más errores que los deseados, por lo que en trabajos futuros la modificación del móvil para el uso de algún otro codificador ayudaría a un mejor desempeño en el seguimiento de trayectorias por parte del móvil.

Falta la implementación de más trayectorias, pero las presentadas son un buen comienzo ya que la iteración de la trayectoria 2 en principio el móvil podría realizar

cualquier trayectoria con  $n$  puntos adquiridos que simulen una función en el plano cartesiano.

# 7 Bibliografía

---

- [1] Silva Ortigoza R., García Sánchez J. R, Barrientos Sotelo V. R., Molina Vilchis M. A., Una Panorámica de los Robots Móviles, Revista Telematique, Vol. 6 No.6 2007.
- [2] Craing John J., Robótica, Pearson Educación, Tercera Edición, México, 2006.
- [3] Abdala Castillo S; Ñeco Carbeta R., Caracterización de un robot manipulador articulado. Tesis de Maestría, Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) Morelos 2003.
- [4] Azcon B., Análisis y diseño del control de posición de un robot móvil con tracción diferencial. Tesis profesional, Escola Tècnica Superior d'Enginyeria (ETSE) de la Universitat de Valencia 2003.
- [5] Borenstein J., Feng L., Correction of Systematic Odometry Errors in Mobile Robots. International Conference on Intelligent Robots and System, Pittsburgh, Pennsylvania, pp. 569-574, 1995.
- [7] Silva Ortigoza R., García Sánchez J. R, Barrientos Sotelo V. R., Molina Vilchis M. A., Modelado y Control de un Robot Móvil tipo Newt en la Tarea de Seguimiento de Trayectoria, Revista Telematique, Vol. 7 No.2 2008.
- [8] Sira Ramirez H., Agrawal S.K., Differentially Flat Systems, Marcel Dekker, New York, 2004.
- [9] Borenstein J., Everett H. R., Feng L., Where am I? Sensors and Methods for Mobile Robot Positioning. University of Michigan, 1996.
- [10] Borenstein J., Everett H.R., Feng L., Wehe D., Mobile Robot Positioning & Sensor and Techniques. Journal of Robotic Systems, Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231 – 249, 1996.
- [11] Panzieri S., Pascucci F., Setola R. Simultaneous Localization and Map Building algorithm for Real Time Applications. Departament Informatica e Automatize, University Roma Via della Vasca Navale, 79, 00144, Roma Italy 2003.

- [12] Pololu 3pi Robot User's Guide 2001–2009.
- [13] Xbee RF Module Product Manual v.1.x.
- [14] Hoja de datos de SRF05.
- [15] Hoja de datos de CMUCAM3.
- [16] Gholipour A., Yazdanpanah M.J., Dynamic tracking control of nonholonomic mobile robot with model reference adaptation for uncertain parameters. Control and Intelligent Processing Center of Excellence, Electrical and Computer Engineering Department, University of Tehran, IRAN, 2002.
- [17] Yamamoto Y., Yun X., Internal Dynamics of a Wheeled Mobile Robot. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and systems, pages 1288-1294, Yokohama, Japan, July, 1993.
- [18] Ramón Pallas Areny, Sensores y acondicionadores de señal, Marcombo y Alfaomega ediciones, Cuarta Edición, México 2007.
- [18] <http://arduino.cc/en/Main/ArduinoBoardUno>
- [19] Stephen J. Chapman, Maquinas Electricas, Mc Graw Hill Interamericana, Cuarta Edición, México 2005.
- [20] Muhammad H. Rashid, Electrónica de potencia, Pearson Educación, Tercera Edición, México, 2004.
- [21] Benjamin C. Kuo, Sistemas de control automático, Prentice Hall Hispanoamericana, Séptima Edición 1996.
- [22] Katsuhiko Ogata, Ingeniería de control moderna, person educación, cuarta edición, Madrid, 2003.
- [23] Lennart Ljung, Torkel Glad, Modeling of Dynamics Systems, Prentice Hall, New Jersey, 1994.
- [24] Richard L. Burden, J. Douglas Faires, Analysis numérico, Cengage Learning, Septima Edicion, 2001.

- [25] José M. Angulo Usategui, Ignacio A. Martínez, Microcontroladores PIC Diseño practico de aplicaciones, Mc Graw Hill Interamericana, Tercera Edición, España, 2003.
- [26] Robert Faludi, Building wireless sensor networks, O'Reilly media, 2010.
- [27] Fernando Reyes, Robótica control de robots manipuladores, Alfaomega, Primera edición, 2011.
- [28] Robert Resnick, David Halliday, Kenneth S. Krane, Fisica Vol. 1, Grupo Patria Cultural, Decimo Edición, México, 2001.
- [29] Marcelo Alonso, Edward J. Finn, Física Vol.1, Fondo Educativo Interamericano, Primera Edición, México 1970.
- [30] Donald T. Greenwood, Principles of Dynamics, Prentice Hall, Segunda Edición, USA, 1988
- [31] Norman S. Nise, Sistemas de control para ingeniería, CECSA, Tercera Edición, México, 2006
- [32] Eduardo Garcia Breijo, Compilador C CCS y simulador PROTEUS para microcontroladores PIC, Alfaomega, Primera Edición, México, 2008



# 8 Apéndice A

---

En la Tabla A.1 se muestran los valores que se obtuvieron al aplicarle una señal escalón unitario de 6V durante 10s a los dos motores.

Tiempo (s)	Motor derecha		Motor Izquierda	
	Cuentas	Velocidad (rad/s)	Cuentas	Velocidad (rad/s)
1	147	3.06	147	3.06
2	151	3.14	151	3.14
3	152	3.16	154	3.2
4	152	3.16	154	3.2
5	153	3.16	156	3.25
6	153	3.18	156	3.25
7	154	3.18	157	3.27
8	154	3.2	157	3.27
9	154	3.2	157	3.27
10	154	3.2	157	3.27

Tabla A.1 Velocidad de los motores

En la Tabla A.2 se muestran los valores que se obtuvieron al aplicarle una señal escalón unitario de 6V durante 1s con un paso de 0.01s al motor izquierdo.

tiempo s	Velocidad rad/s	Cuenta	tiempo s	Velocidad rad/s	Cuenta	tiempo s	Velocidad rad/s	Cuenta
0	0.0000	0	0.21	2.6786	27	0.43	2.9070	60
0.01	0.0000	0	0.22	2.6515	28	0.44	2.8883	61
0.02	1.0417	1	0.23	2.7174	30	0.45	2.9167	63
0.03	1.3889	2	0.24	2.7778	32	0.46	2.8986	64
0.04	1.5625	3	0.25	2.7500	33	0.47	2.8812	65
0.05	1.6667	4	0.26	2.7244	34	0.48	2.9080	67
0.06	1.7361	5	0.27	2.7778	36	0.49	2.9337	69
0.07	1.7857	6	0.28	2.7530	37	0.5	2.9167	70
0.08	2.0833	8	0.29	2.8017	39	0.51	2.9412	72
0.09	2.0833	9	0.3	2.7778	40	0.52	2.9247	73
0.1	2.0833	10	0.31	2.8226	42	0.53	2.9481	75
0.11	2.2727	12	0.32	2.8646	44	0.54	2.9321	76
0.12	2.2569	13	0.33	2.8409	45	0.55	2.9167	77
0.13	2.4038	15	0.34	2.8186	46	0.56	2.9390	79
0.14	2.5298	17	0.35	2.8571	48	0.57	2.9605	81
0.15	2.5000	18	0.36	2.8356	49	0.58	2.9454	82
0.16	2.6042	20	0.37	2.8716	51	0.59	2.9661	84
0.17	2.5735	21	0.38	2.8509	52	0.6	2.9514	85
0.18	2.5463	22	0.39	2.8846	54	0.61	2.9713	87
0.19	2.6316	24	0.4	2.8646	55	0.62	2.9570	88
0.2	2.6042	25	0.41	2.8963	57	0.63	2.9431	89
			0.42	2.8770	58	0.64	2.9622	91

Tabla A.2 Valores de la velocidad angular de 0 a 640ms

En la Tabla A.3 se muestran los valores que se obtuvieron al aplicarle una señal escalón unitario de 6V durante 1s con un paso de 0.01s al motor derecho.

tiempo s	Velocidad rad/s	Cuenta	tiempo s	Velocidad rad/s	Cuenta	tiempo s	Velocidad rad/s	Cuenta
0	0.0000	0	0.21	2.5794	26	0.43	2.8963	57
0.01	0.0000	0	0.22	2.6515	28	0.44	2.8770	58
0.02	0.0000	0	0.23	2.6268	29	0.45	2.9070	60
0.03	0.6944	1	0.24	2.6910	31	0.46	2.8883	61
0.04	1.0417	2	0.25	2.7500	33	0.47	2.8704	62
0.05	1.2500	3	0.26	2.7244	34	0.48	2.9438	65
0.06	1.7361	5	0.27	2.7778	36	0.49	2.8812	65
0.07	1.7857	6	0.28	2.7530	37	0.5	2.9514	68
0.08	2.0833	8	0.29	2.7299	38	0.51	2.9337	69
0.09	2.0833	9	0.3	2.7778	40	0.52	2.9167	70
0.1	2.0833	10	0.31	2.7554	41	0.53	2.9412	72
0.11	2.2727	12	0.32	2.7995	43	0.54	2.9247	73
0.12	2.2569	13	0.33	2.8409	45	0.55	2.9088	74
0.13	2.2436	14	0.34	2.8186	46	0.56	2.9707	77
0.14	2.3810	16	0.35	2.8571	48	0.57	2.9545	78
0.15	2.3611	17	0.36	2.8356	49	0.58	2.9390	79
0.16	2.4740	19	0.37	2.8153	50	0.59	2.9605	81
0.17	2.5735	21	0.38	2.9057	53	0.6	2.9454	82
0.18	2.5463	22	0.39	2.8312	53	0.61	2.9661	84
0.19	2.6316	24	0.4	2.8646	55	0.62	2.9514	85
0.2	2.6042	0	0.41	2.8963	57	0.63	2.9372	86
			0.42	2.8770	58	0.64	2.9906	89

Tabla A.3 Valores de la velocidad angular de 0 a 640ms

# 9 Apéndice B

---

## B.1. Gráfica de los datos obtenidos de los motores

```
clc
f=load('DatosTm1.txt');
[N,M]=size(f);
t=f(:,1);           %variable tiempo t se le asigna los valores de la 1 columna
w=f(:,2);           %variable velocidad angular se le asigna los valores de la 2
columna
cuenta=f(:,3);      %variable cuenta se le asigna la tercer columna

plot(t,w)
grid;
title('Velocidad angular rad/s')
xlabel('t seg');
ylabel('rad');
```

## B.2. Controlador Proporcional

```
clc;
clear all;
t=[0:0.01:5];
for kp=0:0.05:5;
    num1=[kp];
    den1=[1];
    tf1=tf(num1, den1);

    num2=[0.545];
    den2=[.1 1];
    tf2=tf(num2, den2);

    tf3=tf1*tf2
    sys=feedback(tf3, 1)
    y=step(sys, t);
    m=max(y);

    if m <1.05 & m>0.5 %Sobreeelongación
        plot(t,y)
        grid;
```

```

        title('Respuesta escalón unitario')
        xlabel('t seg');
        ylabel('Salida');
        sol=[kp;m]
        break;
    end %if
end %for kp

```

### B.3. Controlador Integral

```

clc
clear all;
t=[0:0.01:2];

for ki=0.1:0.05:5;
    num1=[ki];
    den1=[0 1 0];
    tf1=tf(num1, den1);

    num2=[0.545];
    den2=[.1 1];
    tf2=tf(num2, den2);

    tf3=tf1*tf2
    sys=feedback(tf3,1)
    y=step(sys, t);
    m=max(y);

    if m <1.01 & m>1 %Sobreeelogacion
        plot(t,y)
        grid;
        title('Respuesta escalón unitario')
        xlabel('t seg');
        ylabel('Salida');
        sol=[ki;m]
        break;
    end %if
end %for ki

```

## B.4. Respuesta al escalón unitario del motor CD

```
clc
clear all;
t=[0:0.01:2];
num=[0.545];
den=[0.1 1];
sys=tf(num, den);
y=step(sys, t);
plot(t,y)
grid;
title('Respuesta escalón unitario')
xlabel('t seg');
ylabel('Salida');
```

## B.5. Controlador Proporcional Derivativo

```
clc
clear all;
t=[0:0.01:2];
for kp=0.1:0.05:5;
    for kd=0.1:0.05:5;
        num1=[kd kp 0];
        den1=[1];
        tf1=tf(num1, den1);

        num2=[0.545];
        den2=[.1 1];
        tf2=tf(num2, den2);

        tf3=tf1*tf2;
        sys=feedback(tf3, 1)
        y=step(sys, t);
        m=max(y);

        if m < 1.05 & m > 1 %Sobreeeologacion
            plot(t,y)
            grid;
            title('Respuesta escalón unitario');
            xlabel('t seg');
            ylabel('Salida');
            sol=[kp;kd;m]
```

```

        break;
    end %if
end %for kd
if m <1.05 & m>1;
    sol=[kp;kd;m];
    break;
end %if
end %for kp

```

## B.6. Controlador Proporcional Integral

```

clc
clear all;
t=[0:0.01:2];
for kp=0.1:0.05:5;
    for ki=0.1:0.05:5;
        num1=[kp ki];
        den1=[0 1 0];
        tf1=tf(num1, den1);

        num2=[0 0.545];
        den2=[.1 1];
        tf2=tf(num2, den2);

        tf3=tf1*tf2;
        sys=feedback(tf3,1)
        y=step(sys, t);
        m=max(y);

        if m <1.05 & m>.998 %Sobreeologacion
            plot(t,y)
            grid;
            title('Respuesta escalón unitario')
            xlabel('t seg');
            ylabel('Salida');
            sol=[kp;ki;m]
            break;
        end %if
    end %for ki
    if m <1.05 & m>.998;
        sol=[kp;ki;m];
        break;
    end %if
end %for kp

```

## B.7. Controlador Proporcional, Integral y Derivativo

```
clc
clear all;
t=[0:0.01:2];
for kp=0.1:0.05:5;
    for ki=0.1:0.05:5;
        for kd=0.1:0.05:5;
            num1=[kd kp ki];
            den1=[0 1 0];
            tf1=tf(num1, den1);
            num2=[0 0.545];
            den2=[1 1];
            tf2=tf(num2, den2);

            tf3=tf1*tf2;
            sys=feedback(tf3,1)
            y=step(sys, t);
            m=max(y);

            if m <1.05 & m>1 %Sobreeologacion
                plot(t,y)
                grid;
                title('Respuesta escalón unitario')
                xlabel('t seg');
                ylabel('Salida');
                sol=[kp;ki;kd;m];
                break;
            end %if
        end %for kd
    end %for ki
    if m <1.05 & m>1;
        sol=[kp;ki;kd;m];
        break;
    end %if
end %for ki
if m <1.05 & m>1;
    sol=[kp;ki;kd;m]
    break;
end %if
end %for Kp
```