



UNIVERSIDAD AUTÓNOMA  
DEL ESTADO DE HIDALGO



INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

ÁREA ACADÉMICA DE COMPUTACIÓN

CENTRO DE INVESTIGACIÓN EN TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

---

MAESTRÍA EN AUTOMATIZACIÓN Y CONTROL

SEGUIMIENTO DE TRAYECTORIAS EN VEHÍCULOS AUTÓNOMOS  
TERRESTRES EMPLEANDO *VISUAL SERVOING*

T E S I S

QUE PARA OBTENER EL GRADO DE MAESTRO EN  
CIENCIAS EN AUTOMATIZACIÓN Y CONTROL

**PRESENTA:**

**ALMA YAROSLAVI MARTÍNEZ CAMPOS**

ASESOR:

DR. HUGO ROMERO TREJO

PACHUCA HGO., MÉXICO JULIO DE 2011





Alma Yaroslavi Martínez Campos

Seguimiento de trayectorias  
en vehículos autónomos  
terrestres empleando *visual  
servoing*



# Agradecimientos

**A Dios:** *por ser siempre parte de mi vida. Gracias por tu guía, tu consuelo, por tu protección y por tomarme de tu mano. Reconozco que todo es gracias a ti.*

**A mis padres Alfonso y Anita:** *por ser mi fortaleza, mi refugio, los pilares en mi vida. Gracias porque siempre me han brindado su apoyo y se sin ninguna duda en mi corazón que Dios no pudo darme mejores padres. Los amo con mi vida.*

**A mis hermanos Mario y Karen:** *por ser ángeles en mi vida, ustedes son mi motivo, mi alegría, . . . mi todo. Gracias por ser mis amigos incondicionales y por estar siempre a mi lado, les amo mucho. Deseo que les vaya bien en todo lo que emprendan, son las personas que más admiro y de quienes siempre estaré orgullosa.*

**A mis sobrinos David y Nairan:** *por transformar mi vida con su luz, por darle paz a mi corazón y por ser mi más grande bendición, sin duda ustedes me alientan para ser mejor cada día y quiero ser un buen ejemplo en sus vidas.*

**A Oscar:** *por tu amor, por tu apoyo incondicional y tu paciencia. Bendigo el momento en el que llegaste a mi vida para hacerme muy feliz, gracias por estar a mi lado y por confiar en mi. Te amo y te amaré para siempre jamás.*

**A mi familia:** *por estar siempre al pendiente de mi, por su preocupación, por sus consejos y por todo el amor que día a día tienen para mí. En especial a mi abuelita Julia Gálvez, a mi tía Candida Campos, a mis tíos Mario Martínez, Jesus Campos y Rodrigo Martínez, a Geslio López, Jose Juan López, Ana Levita Campos, Cecilia Martínez, Yajaira Pérez, Gabriela González, Fabiola Pedraza y al resto de mi familia que no es menos importante.*

**A mis amigos de siempre** *por haberme brindado su apoyo, por escucharme y alentarme a cada momento. En especial a Ahiram Cortés, Ana G. Sánchez, Parivash Gutiérrez, Veronica Estrada, Johovani Suarez, Daniel Olvera, Marco Cobak, Gustavo Gutiérrez, Carlos Jiménez, Edgar Efrén, Juan Valdez, Viridiana Martínez entre otros.*

**A mis amigos del CITIS:** *por haber compartido conmigo su amistad y compañerismo. En especial a Lizeth Trejo, Ivone Trejo, Maxel Carpio, Ricardo Sánchez, Carlos del Castillo, Aurelio Jiménez, Victor Arcos, Selene Mora, Isabel Rincón, Ge-lier Reyes y otros más.*

**A los miembros del jurado:** *por sus comentarios y aportaciones a esta tesis.*

**Al Conacyt:** *por su apoyo a través de la beca No. 20603*



# Resumen

## Seguimiento de trayectorias en vehículos autónomos terrestres empleando *visual servoing*

En esta tesis se realiza la construcción de una plataforma experimental con fines de investigación, dicha plataforma consiste de tres robots móviles terrestres tipo unicycle, donde cada uno tiene dos motorreductores de CD conectados a las llantas de tracción, una rueda loca que proporciona estabilidad al sistema, además se tienen tres tarjetas interconectadas: el Arduino que sirve como cerebro del robot, la etapa de potencia y un XBee empleado en la comunicación inalámbrica. Además, el espacio de trabajo está equipado con una cámara de visión como sensor, donde las imágenes captadas son procesadas por un algoritmo desarrollado empleando las plataformas Dev-C++, LabVIEW 9.0, Matlab 7.8 y Arduino que permite estimar la ubicación de cada uno de los robots tanto en posición como en orientación.

Finalmente, se emplea un esquema de *visual servoing* para controlar el seguimiento de una trayectoria en el espacio de trabajo de los robots móviles, para lo cual se realiza el diseño de las leyes de control empleando el enfoque de Lyapunov, el cual toma en cuenta el modelo matemático de los robots móviles, las posiciones y orientación arrojadas por el algoritmo de estimación, garantizando con esto la estabilidad en lazo cerrado. Los resultados se validan experimentalmente.



# Abstract

## Tracking trajectories on autonomous land vehicles using *visual servoing*

In this thesis an experimental platform for research purposes is built, the platform consists of three unicycle type land mobile robots, each robot has two CD gearmotors connected to the traction wheels, a caster wheel which gives stability to the robot, it also has three interconnected cards: the Arduino that works as the robot's brain, the power stage and an XBee used for wireless communication. Also, the workspace is equipped with a vision camera as a sensor, where the captured images are processed by a developed algorithm using the computer tools Dev-C++, LabVIEW 9.0, Matlab 7.8 and Arduino that estimates the location of each mobile robot in both position and orientation.

Finally, a *visual servoing* scheme is used to control the tracking trajectory in the mobile robot's workspace; this is done by designing control laws using Lyapunov's approach, which takes into account the kinematic model and dynamic of mobile robots, the position and the orientation given by the computer estimation algorithm, guaranteeing the stability of the closed-loop system. We show the experimentally results that were obtained.



# Publicaciones

Como resultado de este trabajo de tesis, se publicaron los siguientes artículos:

## Artículos en congresos nacionales e internacionales

### Aceptado

- **A. Yaroslavi Martínez Campos and H. Romero**, *Diseño, construcción y control visual de robots móviles autónomos terrestres*, Simposio Iberoamericano Multidisciplinario de Ciencias e Ingeniería 2011, Pachuca Hidalgo, México, 21 - 23 Septiembre 2011.

### Aceptado

- **A. Yaroslavi Martínez Campos and H. Romero**, *Estimation of position and orientation for land mobile robots using artificial vision*, Mexican International Conference on Artificial Intelligence MICAI 2011, Puebla, México, 26 Noviembre - 4 Diciembre 2011.

### Aceptado

- **A. Yaroslavi Martínez Campos and H. Romero**, *Tracking reference of land mobile robots using visual servoing*, Mexican International Conference on Artificial Intelligence MICAI 2011, Puebla, México, 26 Noviembre - 4 Diciembre 2011.



# Índice general

Agradecimientos . . . . .	III
Resumen . . . . .	V
Abstract . . . . .	VII
Publicaciones . . . . .	IX
Índice general . . . . .	XI
Índice de figuras . . . . .	XV
Índice de tablas . . . . .	XVII
<b>Acrónimos y abreviaturas</b>	<b>XIX</b>
<b>Notación</b>	<b>XXI</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Planteamiento del problema . . . . .	2
1.3. Solución propuesta . . . . .	2
1.4. Objetivos de la tesis . . . . .	2
1.5. Hipótesis de trabajo . . . . .	3
1.6. Justificación . . . . .	3
1.7. Contribuciones de la tesis . . . . .	3
1.8. Organización de la tesis . . . . .	4
<b>2. Estado del arte</b>	<b>5</b>
2.1. Introducción . . . . .	5
2.2. Clasificación de los robots . . . . .	6
2.3. Áreas de estudio actuales . . . . .	9
2.4. Importancia del control de formación y marcha . . . . .	10
2.5. Trabajo relacionado . . . . .	10
2.6. Comentarios . . . . .	12

<b>3. Diseño y construcción del robot móvil</b>	<b>13</b>
3.1. Introducción . . . . .	13
3.2. Diseño mecánico . . . . .	14
3.2.1. Sistemas del robot móvil . . . . .	15
3.3. Diseño electrónico . . . . .	19
3.3.1. Fuente de alimentación . . . . .	19
3.3.2. Sistema de control . . . . .	21
3.3.3. Interfaz de potencia . . . . .	26
3.4. Construcción del robot móvil . . . . .	27
3.4.1. Prototipo . . . . .	27
3.4.2. Etapa electromecánica . . . . .	28
3.4.3. Módulo electrónico . . . . .	29
3.5. Comentarios . . . . .	30
<b>4. Estimación visual</b>	<b>31</b>
4.1. Introducción . . . . .	31
4.2. Calibración de la cámara . . . . .	32
4.3. Estimación de la ubicación del robot móvil . . . . .	33
4.4. Resultados obtenidos . . . . .	40
4.5. Comentarios . . . . .	41
<b>5. Modelado y control</b>	<b>43</b>
5.1. Introducción . . . . .	43
5.2. Suposiciones de trabajo . . . . .	44
5.2.1. Restricciones cinemáticas . . . . .	45
5.3. Metodología propuesta para la obtención del modelo matemático . . . . .	47
5.4. Diseño de las leyes de control . . . . .	52
5.4.1. Diseño de una ley de control mediante el análisis de Lyapunov . . . . .	53
5.5. Implementación de la etapa de control . . . . .	54
5.6. Comentarios . . . . .	55
<b>6. Resultados experimentales</b>	<b>57</b>
6.1. Introducción . . . . .	57
6.2. Plataforma experimental . . . . .	58
6.3. Experimentos realizados . . . . .	59
6.3.1. Primer experimento: Desplazamiento en el eje $x$ . . . . .	60
6.3.2. Segundo experimento: Desplazamiento en el plano $x, y$ . . . . .	63
6.3.3. Tercer experimento: Desplazamiento en el eje $x$ , desplazamiento en el eje $y$ , ante perturbaciones de iluminación. . . . .	66
6.4. Comentarios . . . . .	69

---

<b>7. Conclusiones y trabajos futuros</b>	<b>71</b>
<b>Glosario</b>	<b>73</b>
<b>A. Plataformas: Dev-C++, LabVIEW, Matlab y Arduino</b>	<b>77</b>
<b>B. Conceptos de estabilidad en el sentido de Lyapunov</b>	<b>81</b>
<b>C. Herramientas de visión artificial</b>	<b>87</b>
<b>D. Herramientas computacionales</b>	<b>93</b>
<b>Bibliografía</b>	<b>99</b>



# Índice de figuras

3.1. Estructura del robot tipo unicycle. . . . .	14
3.2. Diseño del robot móvil en configuración unicycle. . . . .	14
3.3. Placa de acrílico utilizada como base ó chasis para el robot móvil. . .	15
3.4. Placa superior de acrílico empleada para sostener la electrónica em- pleada en el robot móvil. . . . .	16
3.5. Tornillos utilizados para sujetar las placas de acrílico. . . . .	17
3.6. Sistema de tracción. . . . .	18
3.7. Llantas que utiliza el robot móvil. . . . .	18
3.8. Dimensiones de la rueda loca. . . . .	19
3.9. Batería recargable Nickel-Metal de 9V, 300mA. . . . .	20
3.10. Disposición de pines del regulador de voltaje. . . . .	20
3.11. Fuente de alimentación de 5V y 12V. . . . .	21
3.12. Disposición de pines del microcontrolador empleado en el control del robot móvil. . . . .	22
3.13. Características principales de Arduino Duemilanove. . . . .	24
3.14. Conexiones mínimas requeridas para el XBee. . . . .	25
3.15. Diagrama esquemático de la tarjeta electrónica para el robot móvil. .	26
3.16. Disposición de pines del puente H empleado para el control de los motores.	27
3.17. Robot móvil con llantas en configuración unicycle. . . . .	28
3.18. Sistema electromecánico final, vista isométrica. . . . .	28
3.19. Tarjeta Arduino Duemilanove. . . . .	29
3.20. Tarjeta electrónica (diseño en pistas y construcción). . . . .	29
4.1. Patrón de calibración. . . . .	33
4.2. Estructura usada de OpenCV. . . . .	34
4.3. Muestras positivas. . . . .	36
4.4. Ejecución haartraining. . . . .	37
4.5. Detección de objetos (motivos). . . . .	38
4.6. Coordenadas en el plano imagen (espacio de trabajo). . . . .	39
4.7. Ángulo de orientación. . . . .	40

4.8. Resultado final del algoritmo de visión. . . . .	41
5.1. Componentes de un robot móvil en configuración de unicycle. . . . .	44
5.2. Restricción holónoma. . . . .	45
5.3. Restricción no holónoma, complejidad para llegar a una posición final. . . . .	46
5.4. Variables, parámetros y componentes involucradas en el modelado matemático del robot móvil con ruedas. . . . .	47
5.5. Movimiento de una rueda en el plano $(X, Y)$ . . . . .	49
5.6. Robot virtual de referencia. . . . .	52
6.1. Fotografía de la plataforma experimental. . . . .	58
6.2. Motivo colocado sobre un robot móvil. . . . .	59
6.3. Señal de referencia vs señal de posición en el eje $x$ . . . . .	60
6.4. Señal de error en la posición $x$ . . . . .	61
6.5. Señal de posición en el eje $y$ . . . . .	61
6.6. Señal de posición angular de orientación $\theta$ . . . . .	62
6.7. Señales de control $v$ y $\omega$ . . . . .	62
6.8. Señal de referencia vs señal de posición en el eje $x$ . . . . .	63
6.9. Señal de referencia vs señal de posición en el eje $y$ . . . . .	64
6.10. Señal de posición angular de orientación $\theta$ . . . . .	64
6.11. Señales del error en las posiciones $x, y$ . . . . .	65
6.12. Señales de control $v$ y $\omega$ . . . . .	65
6.13. Señal de referencia vs señal de posición en el eje $x$ . . . . .	66
6.14. Señal de referencia vs señal de posición en el eje $y$ . . . . .	67
6.15. Señal de posición angular de orientación $\theta$ . . . . .	67
6.16. Señales del error en las posiciones $x, y$ . . . . .	68
6.17. Señales de control $v$ y $\omega$ . . . . .	68
A.1. Plataformas computacionales empleadas. . . . .	77
A.2. Directorio. . . . .	78
A.3. Contenido de la carpeta: Imágenes. . . . .	78
A.4. Contenido de la carpeta: Programas. . . . .	78
A.5. Contenido de la carpeta: Resultados experimentales. . . . .	79
C.1. Representación de un píxel. . . . .	88
C.2. Organización matricial uniforme de una imagen digital. . . . .	88
C.3. Representación gráfica del modelo de color RGB. . . . .	89
C.4. Etapas típicas en un sistema de <i>Visión Artificial</i> . . . . .	90
C.5. Tipos básicos de distorsión en una lente. . . . .	91
D.1. Estructura OpenCV. . . . .	95

# Índice de tablas

3.1. Tabla de verdad para el control de giro de los motores, en donde L=Estado bajo, H=Estado alto y X=Estado de no importa. . . . .	27
4.1. Algoritmo: Adquisición y visualización de video . . . . .	34
5.1. Algunas diferencias entre un robot móvil no-holonómico y uno holonómico. . . . .	46
5.2. Variables, parámetros y componentes del robot móvil en configuración unicycle. . . . .	48



# Acrónimos y abreviaturas

Siglas	Descripción
AD	Arduino Duemilanove.
CA	Corriente Alterna.
CD	Corriente Directa.
CI	Circuito Integrado.
CMOS	Semiconductor de Óxido Metálico Complementario.
CR	Centro de Rotación.
DMOS	Semiconductor de Óxido Metálico de Doble Difusión.
EEPROM	ROM programable y borrable eléctricamente.
FTDI	El futuro la tecnología de los dispositivos internacionales.
GND	Tierra en los circuitos eléctricos.
IPL	Librería de procesamiento de imágenes de intel.
IPP	Primitivas de desempeño integral de intel.
ISCP	Programación de circuito en serie.
Matlab	Laboratorio de matrices.
OpenCV	Open Source Computer Vision Library.
PWM	Modulación por ancho de pulso.
RGB	Rojo Verde Azul.
RPM	Revoluciones por minuto.
RX	Línea de recepción de datos.
SRAM	Memoria Estática de Acceso Aleatorio.
TTL	Lógica Transistor-Transistor.
TX	Línea de transmisión de datos.
UART	Transmisor-Receptor Asíncrono Universal.
USB	Bus universal en serie.
VIN	Voltaje de entrada



# Notación

Símbolo	Descripción
<i>A</i>	Amper.
<i>cm</i>	centímetro.
<i>g</i>	gramo.
<i>Hz</i>	Hertz.
$K\Omega$	Kilo-Ohm.
<i>KB</i>	Kilo byte.
<i>m</i>	metro.
<i>m/seg</i>	metro/segundo.
<i>mA</i>	mili Amper.
<i>Mhz</i>	Megahertz.
<i>MW</i>	Mega-Watt.
$\Omega$	Ohm.
<i>rad</i>	radián.
<i>rad/seg</i>	radián/segundo.
<i>seg</i>	segundo.
<i>V</i>	Volt.



# Capítulo 1

## Introducción

### 1.1. Motivación

Los robots móviles se caracterizan por su capacidad de desplazarse de forma autónoma en un entorno desconocido o conocido sólo parcialmente. Sus aplicaciones cubren una gran variedad de campos, entre los cuales se incluyen trabajos subterráneos (minería, construcción de túneles, etc.), tareas submarinas (inspección de oleoductos, mediciones, misiones de búsqueda y rescate, etc.), misiones espaciales y exploración planetaria (recogida de muestras, mantenimiento de estaciones orbitales, etc.), vigilancia e intervención de seguridad (desactivación de explosivos, operación en zonas radioactivas, etc.), aplicaciones militares, y muchos otros. En todas estas aplicaciones la justificación más importante para el uso de la robótica directa o teleoperada es la dificultad o imposibilidad de intervención humana.

El calificativo de “autónomo” hace referencia a la capacidad de percibir, modelar, planificar y actuar para alcanzar unos objetivos sin la intervención, o con una intervención mínima de supervisión humana. Esto delimita la frontera entre los robots autónomos y los vehículos o máquinas teleoperadas, donde un operador humano realiza de forma remota las tareas anteriores. De igual manera, no pueden ser considerados robots autónomos, los denominados vehículos autoguiados, los cuales se limitan a seguir caminos preestablecidos (pintados en el suelo, en bandas magnéticas, etc.) o a actuar de forma repetitiva. Este es el caso de los carros filo-guiados que a menudo se encuentran en fábricas y escenarios industriales, y cuya misión consiste en repetir una y otra vez determinados caminos y acciones.

**En este trabajo se realiza la construcción y control no lineal visual de tres vehículos autónomos terrestres (robots móviles homogéneos).**

A continuación se enuncia el problema que se plantea en esta tesis.

## 1.2. Planteamiento del problema

La problemática abordada en este trabajo se resume en:

Cómo controlar el movimiento de cada robot móvil durante la realización de una trayectoria dada en el espacio de trabajo.

## 1.3. Solución propuesta

La solución que se propone al problema planteado se describe a continuación:

- En primer lugar, se realiza la construcción de los tres robots móviles tipo uniclo, empleando para cada uno de estos dos placas de acrílico de  $7 \times 9.5$  cm, con un espesor de 0.3 cm, unidas por 4 tornillos donde se monta la tarjeta de control marca Arduino, que a su vez tiene interconectado otra tarjeta electrónica (etapa de adaptación y potencia) y el módulo de comunicación inalámbrico XBee.
- Posteriormente, se obtiene el modelo cinemático del robot móvil, el cual está descrito por una ecuación diferencial no lineal.
- A continuación, se realiza la calibración de la cámara marca Philips para obtener sus parámetros intrínsecos, los cuales son usados en la etapa de reconocimiento de patrones.
- Enseguida se obtienen las variables físicas: posición  $x, y$  y el ángulo de orientación  $\theta$  mediante la captura y procesamiento de las imágenes (motivos) de cada robot.
- Finalmente, se diseñan las leyes de control para el seguimiento de trayectorias empleando para esto la información obtenida a través de la cámara de visión.

## 1.4. Objetivos de la tesis

Diseñar y evaluar el desempeño de leyes de control no lineal para el seguimiento de trayectorias en vehículos autónomos terrestres, empleando *visual servoing*.

Para lograr el objetivo general, se establecen los siguientes objetivos específicos:

1. Construir un robot móvil tipo unicycle para propósitos de investigación.
2. Desarrollar un algoritmo de visión para estimar la ubicación de tres robots móviles.
3. Desarrollar leyes de control no lineal que hacen uso del modelo cinemático y de las dinámicas del robot móvil, usando la información entregada por la cámara de visión.
4. Evaluar las leyes de control ante cambios en la iluminación en el espacio de trabajo.

## 1.5. Hipótesis de trabajo

Empleando un esquema de *visual servoing* es posible realizar el seguimiento de trayectorias en vehículos autónomos terrestres.

## 1.6. Justificación

Uno de los aspectos importantes en todo centro de investigación y/o desarrollo tecnológico es contribuir al conocimiento, en el caso del Centro de Investigación en Tecnologías de la Información y Sistemas no es la excepción, ya que este trabajo de tesis contribuye al autoequipamiento de un laboratorio de robótica móvil, que tenga prototipos funcionales y económicos con propósitos de investigación en el área de control para la coordinación y cooperación entre robots móviles terrestres, ya que se cuenta con una arquitectura abierta para probar diferentes estrategias de control que puedan ser validadas en la parte experimental.

## 1.7. Contribuciones de la tesis

Las contribuciones principales que presenta este trabajo de tesis se dan a continuación:

1. Contar con tres prototipos funcionales de robots móviles terrestres con autonomía en su sistema de locomoción.
2. Integrar diferentes tecnologías en la instrumentación y control de los robots móviles.

3. Integrar diferentes plataformas de programación (Dev-C++, LabVIEW, Matlab, Arduino) para el monitoreo y control de los robots móviles empleando como único sensor la cámara de visión.
4. Proponer una metodología para la obtención de las variables: posiciones  $x, y$  y el ángulo de orientación  $\theta$  de cada uno de los robots móviles a partir de las imágenes captadas y procesadas.

## 1.8. Organización de la tesis

La tesis se encuentra organizada en siete capítulos y cuatro apéndices, los cuales están distribuidos de la siguiente manera:

- Capítulo 2, *Estado del arte* se describen algunos de los trabajos relacionados con este trabajo de tesis.
- En el Capítulo 3, *Diseño y construcción del robot móvil* se presentan cada una de las piezas utilizadas en la construcción, además se da la electrónica diseñada y empleada para la comunicación y control del robot móvil.
- En el Capítulo 4, *Estimación visual* se da la metodología propuesta para la obtención de las posiciones  $x, y$  y el ángulo de la orientación  $\theta$  de cada uno de los robots móviles terrestres.
- En el Capítulo 5, *Modelado y control* se obtiene el modelo cinemático del robot móvil, el cual es empleado para el diseño de la ley de control.
- En el Capítulo 6, *Resultados* se muestran los resultados arrojados de la implantación de la ley de control para el seguimiento de trayectoria que hace uso de las posiciones y ángulos obtenidos a través de la metodología visual propuesta.
- Finalmente, las conclusiones y trabajos futuros se describen en el Capítulo 7.

# Capítulo 2

## Estado del arte

El objetivo de este capítulo es presentar y describir los trabajos de investigación y desarrollo tecnológico que están directamente relacionados con esta tesis.

El capítulo está organizado de la siguiente manera: En primer lugar, se da una breve introducción sobre robots móviles en la Sección 2.1, mientras que en la Sección 2.2 se describen las clasificaciones de este tipo de robots, al tiempo que se establece la clasificación de los robots móviles con los que se trabaja en el desarrollo de esta tesis. Posteriormente, se dan las áreas de estudio actuales para estos sistemas en la Sección 2.3. La importancia del control de formación y marcha se resume en la Sección 2.4, enseguida los trabajos sobre robots móviles basados en el esquema de comportamiento e instrumentados con sensores tradicionales, de ubicación y de visión artificial se dan en la Sección 2.5. Finalmente, los comentarios de este capítulo se tienen en la Sección 2.6.

### 2.1. Introducción

Un robot móvil es un sistema en el que cabe identificar diversos subsistemas: de percepción, planificación, control de movimientos y locomoción que interaccionan entre sí. El sistema de percepción permite que el robot sea capaz de hacer frente a situaciones cambiantes del entorno, así como a reaccionar ante posibles eventos inesperados, lo que exige la utilización de un sistema sensorial que suministre la información del entorno. Esta información requerida debe permitir al robot realizar tres tareas fundamentales: estimar su posición y orientación, mantener actualizado el mapa del entorno y detectar los posibles obstáculos. Esta idea describe a un sistema robótico dotado de sensores, herramientas y/o manipuladores [17] que le permitan efectuar tareas en una determinada ubicación. El robot móvil utiliza como sistema de locomoción ruedas, orugas, patas ó una combinación de éstas. La definición de robot

móvil incluye la habilidad para operar con autonomía, basada en decisiones hechas por el robot mismo y sin la necesidad de la intervención humana [50].

## 2.2. Clasificación de los robots

Los robots pueden clasificarse de acuerdo a dos factores: arquitectura de grupo y origen de la cooperación. Cada uno de estos factores es analizado a continuación.

### Arquitectura de grupo:

La arquitectura de grupo determina las capacidades y limitaciones del sistema. Según [3] y [18] los tres rasgos importantes que determinan la arquitectura de grupo son: tipo de control, diferenciación de los robots y estructura de comunicación.

1. **Tipos de control.** Los tipos de control de sistemas de robots móviles coordinados pueden ser centralizado o descentralizado. En el primero se asume que existe un control global [48]. La arquitectura descentralizada se divide a su vez en distribuido (o totalmente descentralizado), donde todos los robots son iguales respecto al control y en híbrido (algunos autores lo llaman jerárquico) donde existe un control de más alto nivel en algunos de los robots y que gobierna sobre otros, de esta forma, existen robots líderes dentro del grupo. Las ventajas del esquema descentralizado son una mayor autonomía de los robots móviles (sólo provistos de un controlador local), el decrecimiento de la carga computacional de los algoritmos de control con respecto a un esquema centralizado y su aplicabilidad a grupos de gran escala. Sin embargo, la complejidad del control descentralizado reside en cómo lograr un comportamiento global deseado a través de interacciones locales [31].
2. **Diferenciación de los robots.** La diferenciación de los robots se puede clasificar en homogéneos (cuando los robots móviles y sus capacidades son iguales) y heterogéneos (cuando los robots móviles son distintos y/o tienen capacidades distintas). Un grupo heterogéneo conlleva el problema de la asignación de tareas y los robots tienen una mayor necesidad de modelar las capacidades de los otros robots. Sin embargo, en grupos homogéneos a veces es necesario distinguir entre los diferentes roles (como líder o seguidor).
3. **Estructura de comunicación.** La cual determina las formas posibles de interacciones del grupo. Según [14] estas posibles interacciones pueden resumirse en:

- *Interacción implícita.* También conocida como cooperación sin comunicación, donde no existe una comunicación explícita entre robots sino que el medio ambiente es su medio de comunicación. Un ejemplo es cuando dos robots desean empujar un objeto desde dos puntos diferentes sin comunicación entre ellos. Cuando el objeto empieza a moverse, uno de los robots interpreta que su compañero ha empezado a realizar su parte y entonces empieza a operar.
- *Interacción mediante sensores.* Donde la interacción ocurre cuando un robot sensa a otro sin una comunicación explícita. Un robot deberá diferenciar si lo que sensa es a otro robot o a un obstáculo (*kin recognition*). Los sensores pueden ser de radiofrecuencia o infrarrojos o incluso utilizando visión artificial.
- *Interacción explícita.* Existe comunicación explícita mediante la difusión de mensajes codificados. Muchas disciplinas como el diseño de topologías para redes y protocolos de comunicación son utilizadas para estos fines. Paralelo a este tipo de comunicación se encuentran problemas de optimización de comunicaciones locales y retardos en la comunicación.

**En este trabajo de tesis se trata el problema de seguimiento de trayectorias de robots móviles terrestres con una arquitectura de control descentralizado del tipo distribuido. Los robots son homogéneos y la estructura de comunicación es vía inalámbrica, utilizando una cámara de visión como sensor para la ubicación de los robots.**

El segundo factor de clasificación de sistemas de robots tiene que ver con la base de origen de la cooperación.

### Orígenes de la cooperación:

Según [20], existen dos esquemas en los que se basan los sistemas multi-robots para tener un comportamiento cooperativo: Los *esquemas basados en comportamiento*, proveniente del estudio del comportamiento de especies animales, y los *esquemas basados en modelos* (llamados también comportamientos emergentes) que se fundamentan principalmente en conceptos de física y de teoría del control.

1. **Esquemas basados en comportamiento.** Los esquemas basados en comportamiento imitan a seres biológicos (conocidos en la literatura como *swarms* que incluyen hormigas, peces, aves, etc. [24]), donde existe una organización colectiva a través de un comportamiento natural individual. Algunos trabajos al respecto son [6],[32],[52]. Según [52], cuando algunas especies de animales

pequeños trabajan en grupo, ellos combinan sus sensores para maximizar el rango de detección de depredadores o buscar comida más eficientemente, entre otras actividades. La literatura que trata acerca de los *swarms* [24] es amplia y aplicándose a sistemas multi-robots han mostrado buenos resultados en aspectos como organización por ellos mismos (*self-organization*), tolerancia a fallas, reparaciones y además de necesitar las mínimas capacidades de sensado. Un trabajo de mucha influencia ha sido el de Reynolds en 1987 [11]. Reynolds estudió un grupo de aves en vuelo. Descubrió que volaban agrupadas siguiendo a un líder en común y evitaban colisionar entre ellas. Cada ave tenía una estrategia de control local que en conjunto provocaba un comportamiento global deseado. En resumen la estrategia local de cada ave tenía tres componentes: separación, alineación y cohesión. Las ideas de Reynolds han inspirado trabajos en la formación de grupos de robots móviles (ver [21],[31],[45]). Otros comportamientos sociales de especies biológicas como la búsqueda o el sistema cazador-presa han servido para dotar de inteligencia a los robots en actividades como la evasión y detección de perseguidores. Las estrategias para competición en sistemas multi-robot, tales como en robots de soccer, también han sido inspiradas en especies más grandes de animales, donde el comportamiento no es sólo innato o de supervivencia, sino intencional, con el propósito de maximizar las capacidades individuales. En esta última rama, la inteligencia artificial, la teoría de juegos y la teoría económica han jugado un papel importante.

2. **Esquemas basados en modelos.** En los esquemas basados en modelos [20], la cinemática o dinámica de cada robot móvil y las interacciones entre los robots son modeladas matemáticamente. Se llaman también comportamientos emergentes porque no son copiados de modelos naturales sino que surgen de las técnicas de teoría de control, física, inteligencia artificial, etc. y que suponen comportamientos artificiales en la coordinación de los agentes móviles. Generalmente estas técnicas se aplican sobre sistemas descentralizados. Algunos enfoques que utilizan teoría de control [28],[38] organizan al grupo de agentes de forma que existen diferentes configuraciones de líderes y seguidores (arquitectura híbrida). Algunas técnicas basadas en física utilizan los campos potenciales artificiales locales. En estos trabajos, los obstáculos y otros robots son campos repulsivos y la meta de cada robot es un potencial atractivo. Algunas otras técnicas de análisis dentro de los esquemas basados en modelos se mencionan en las referencias de [24] y algunas de ellas son: Métodos basados en la teoría de estabilidad de Lyapunov, análisis de topología/geometría, modelos de ecuaciones diferenciales, teoría de grafos, predicción del movimiento de cohesión, frecuencia de colisión, fuerzas de atracción y repulsión de partículas, inteligencia artificial distribuida, lógica temporal, entre otros.

**En este trabajo, se emplean estrategias de control no lineal dentro del esquema basado en el modelo.**

## 2.3. Áreas de estudio actuales

Existen diversas áreas de interés con respecto a los sistemas de robots móviles. Según [3],[14] estas áreas de estudio pueden clasificarse en:

1. **Coordinación de movimiento.** En esta área se tratan problemas como control de tráfico, control de formación, avance en formación, seguimiento grupal de trayectoria, alcanzar una meta [61], etc. En algunos trabajos se busca el control de formación utilizando el mínimo de sensores y el mínimo de comunicaciones. La idea principal es que un robot mantenga a otro robot vecino dentro de su campo visual y guarde una posición relativa con respecto a él. La planeación de trayectoria deberá contemplar a todos los robots y su ambiente, en general, es un problema geométrico en configuración con el espacio-tiempo.
2. **Asignación de tareas.** En esta área se estudian aspectos como la delegación de autoridad y control, estructuras de comunicación, robots heterogéneos versus robots homogéneos, realizar acciones locales coherentes, resolución de conflictos, etc.
3. **Comunicación.** Se buscan estrategias para la optimización de la interacción de los robots móviles desarrollando protocolos de comunicación o nuevos canales de comunicación.
4. **Transportación y manipulación de objetos.** Los robots deben cargar, empujar o manipular objetos comunes y muchas veces estas actividades se combinan al mismo tiempo con navegar a un destino en una formación específica. El enfoque de estos problemas está relacionado con la asignación de tareas, tolerancia a fallas y aprendizaje.
5. **Exploración y búsqueda.** El grupo de robots móviles debe levantar objetos dispersos en el ambiente, evocando labores como limpieza de residuos tóxicos, cosecha, búsqueda y rescate, etc.
6. **Localización y mapeo.** Se plantean algoritmos distribuidos para que un grupo de robots navegue en lugares de interés, inicialmente desconocidos, para ser explorados y así, por ejemplo, converger a la mejor ruta de exploración descubierta por algún robot.

## 2.4. Importancia del control de formación y marcha

Una de las áreas de estudio importantes dentro de la problemática actual de los robots móviles es la coordinación de movimiento. Como se ha mencionado anteriormente, en esta área se trata de resolver problemas como control de tráfico, control de formación y avance en formación o marcha, etc. El control de formación y marcha está inmerso en casi todas las aplicaciones de robots móviles, ya que todas estas actividades requieren un control en la coordinación de movimiento de los robots dentro del espacio de trabajo con el propósito de obtener una colocación y/o desplazamiento estratégico para realizar un trabajo en común [61]. Por ejemplo, en las aplicaciones de vigilancia y exploración se requiere que los robots avancen en una formación específica para maximizar sus capacidades de detección y eventualmente reconfigurar el patrón del grupo si uno de los robots sufre una avería [52]. En las operaciones de búsqueda y rescate los robots deben de tener la capacidad de compartir la información que cada uno recopila y eventualmente seguir una trayectoria común o reagruparse en ciertas formaciones [20]. En el caso de manipulación de objetos grandes [14], los robots deben alcanzar posiciones determinadas en el grupo para cargar y mover un objeto hacia algún lugar dentro del espacio de trabajo [3],[22]. Existen algunos requerimientos a satisfacer cuando se propone una ley de control de formación y control de marcha de un grupo de agentes móviles, las cuales pueden resumirse en los siguientes puntos:

- Convergencia a la formación deseada desde cualquier posición inicial de los robots móviles.
- Evasión de obstáculos.
- En el caso de marcha, convergencia a la trayectoria deseada.

La presente tesis está enfocada a desarrollar e implementar estrategias de control no lineal para el seguimiento de trayectorias, considerando un espacio de trabajo libre de obstáculos y dejando como trabajo futuro la parte de formación y marcha. En la siguiente sección se enumeran algunos trabajos relacionados y las contribuciones de esta tesis a la literatura del control de seguimientos de trayectoria así como de formación y marcha.

## 2.5. Trabajo relacionado

Los trabajos de control de formación y marcha descentralizados generalmente se pueden clasificar en dos grandes grupos [20],[52]:

- El primer esquema (basado en comportamientos), los robots móviles se forman siguiendo reglas de comportamiento simples, como conservar distancia entre vecinos, inteligencia de *swarm*, técnicas basadas en física, etc. [11],[24],[35],[52]. En este esquema generalmente se considera a todos los robots móviles con las mismas capacidades de percepción y entonces convergen a patrones de formación sin una posición específica.
- El segundo esquema de formación está basado en gráficas de formación (provenientes de teoría de grafos) las cuales representan las comunicaciones posibles entre agentes para alcanzar la formación deseada [19],[27],[43],[45],[55]. Similar a las gráficas de formación está el enfoque de patrones geométricos con trabajos como [32].

El análisis de las estrategias de formación se centra en mostrar sus dos requerimientos fundamentales: convergencia global a la formación deseada y evasión de colisiones entre agentes [28], [61]. Algunos trabajos buscan probar la convergencia global sin considerar la colisión. El primer caso de estudio es el llamado *agreement problem o consensus problem* [29],[31],[44]. El problema consiste en, dado un grupo de robots móviles, con capacidades de sensado limitadas a un cierto radio de influencia, demostrar que convergen a un punto en común. Se establecen condiciones mínimas de comunicación para que esto se logre, ya que el grupo podría converger a dos o tres puntos diferentes y no a un solo punto en el plano. El segundo caso de análisis de convergencia está enmarcado en el contexto de gráficas de formación [30],[33],[43] para el caso de robots puntuales u omnidireccionales y robots no-holonomos. Otros trabajos complementan el análisis de convergencia al estudiar la infactibilidad de la formación (*formation infeasibility*), que estudia las condiciones para que los robots converjan a un equilibrio, rigidez y estabilidad de la formación (ver referencias de [4],[5],[12],[13],[56],[57],[60]).

Algunas nociones de estabilidad de formación se presentan en [12],[57],[61]. Una de ellas es la estabilidad de cadena, que analiza la propagación de una perturbación en un grupo de robots ya formados. Un sistema estable, en ese sentido, requiere que si uno o varios de los robots es perturbado de su posición en la formación, ese efecto desaparezca cuando es transmitido a los demás miembros del grupo. Otro concepto es estabilidad de acoplamiento, que garantiza que la estrategia de formación funciona aún cuando el número de robots se incrementa, analizando los efectos que conlleva el tratar con un mayor número de robots móviles. El último concepto es la estabilidad líder-formación donde se pone mayor énfasis en cómo el comportamiento del líder afecta los errores de interconexión resultantes en los demás miembros de la formación. Es importante señalar que el análisis de infactibilidad y rigidez de formación quedan fuera de los objetivos de este trabajo de tesis.

Al utilizar un algoritmo de visión en la plataforma de trabajo como se hace en [23] donde los robots móviles son etiquetados en la parte superior con círculos del mismo color para su identificación, hace posible una reducción en el procesamiento al escoger motivos circulares (invariantes a rotaciones) permitiendo únicamente conocer la posición de los robots móviles  $x, y$ . El uso de una cámara omnidireccional catadióptrica combina cámaras convencionales y un espejo para obtener una visión de  $360^\circ$  como se presenta en [10]. El sensado omnidireccional es posible con espejos convexos como: espejos cónicos, esféricos o hiperbólicos, en el este trabajo se localiza el robot por color en la imagen omnidireccional y luego se estima el punto que le corresponde en el espacio asumiendo que el robot solo se mueve en el plano horizontal. Las ecuaciones que relacionan puntos en el espacio con puntos en el plano de la imagen no son tan complicadas; sin embargo, el mapeo inverso es bastante complejo además que ha sido probado únicamente a nivel simulación utilizando un control centralizado.

## 2.6. Comentarios

Los trabajos presentados y descritos en este capítulo son los que están de alguna manera relacionados con el trabajo de tesis, ya que ésta involucra diferentes áreas de la ingeniería. El trabajo que inspiró el desarrollo de esta tesis se presenta en [23], la cual usa motivos del mismo color para determinar la posición de los robots, sin determinar el ángulo de orientación de cada uno de los robots móviles. La diferencia con nuestro trabajo, radica principalmente que proponemos motivos de diferentes colores con la finalidad de determinar las posiciones y orientaciones de los vehículos autónomos terrestres.

# Capítulo 3

## Diseño y construcción del robot móvil

Este capítulo está dedicado al diseño y construcción del robot móvil en configuración unicycle, se detallan todas y cada una de las piezas utilizadas en la construcción, así como también se da a conocer la electrónica empleada para el control del robot móvil.

La organización de este capítulo es la siguiente: En primer lugar, se presenta en la Sección 3.1 una introducción a los robots móviles terrestres tipo unicycle. Posteriormente, se da en la Sección 3.2 el diseño mecánico, el cual está conformado por las piezas mecánicas del robot móvil. En seguida, se muestra en la Sección 3.3 el diseño electrónico que es empleado para instrumentación, comunicación y control del robot móvil. A continuación en la Sección 3.4 se describe la construcción del vehículo móvil terrestre en su configuración unicycle. Finalmente, los comentarios se dan en la Sección 3.5.

### 3.1. Introducción

El robot tipo unicycle es, en general, el elegido por los investigadores a la hora de probar nuevas estrategias de control por tener una cinemática sencilla. Es una estructura que consta de dos ruedas fijas convencionales sobre el mismo eje, controladas de manera independiente y una rueda loca que le confiere estabilidad, como se ve en la Figura 3.1.

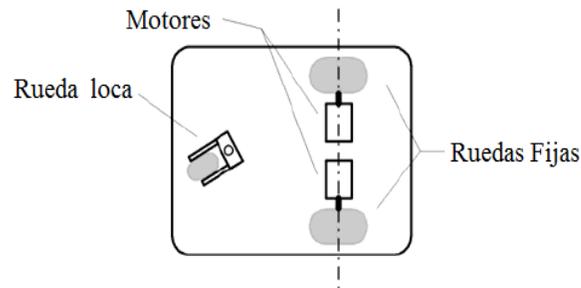


Figura 3.1: Estructura del robot tipo unicycle.

El sistema de tracción asociado al robot le permite independizar las consignas de velocidad lineal y angular respectivamente, las ventajas que se derivan de la estructura mecánica y de la electrónica de control hacen de esta configuración la preferida para robots de laboratorio. Esta es una de las razones por la cual se decide trabajar con este tipo de robots ya que su propósito es para fines de investigación.

## 3.2. Diseño mecánico

En esta parte del diseño se da a conocer cómo se construye el esqueleto del móvil (Figura 3.2), se detallan características de los materiales, la tracción, las llantas y otras piezas utilizadas para el acoplamiento de las partes antes mencionadas.

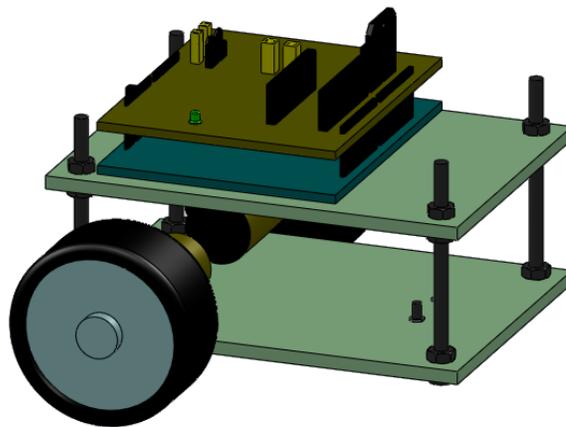


Figura 3.2: Diseño del robot móvil en configuración unicycle.

### 3.2.1. Sistemas del robot móvil

El robot móvil en configuración unicycle consta básicamente de tres sistemas que son:

- a. Sistema de soporte.
- b. Sistema de tracción.
- c. Sistema de dirección.

A continuación se explican cada uno de los sistemas que conforman al robot móvil:

#### (a). Sistema de soporte

El sistema de soporte está conformado por una placa base y una placa superior de acrílico y cuatro tornillos metálicos. Enseguida se describe cada uno de los componentes mencionados:

#### Placa base (chasis)

La placa principal del robot móvil es una lámina de acrílico de  $7 \times 9.5$  cm, con un grosor de 0.3 cm. El diseño propuesto tiene la forma de un rectángulo, como se muestra en la Figura 3.3. La finalidad de dicho diseño radica en la disminución de peso con la desventaja que en el momento de chocar el móvil (en caso que se dé) con algún objeto éste puede atorarse debido a sus esquinas.

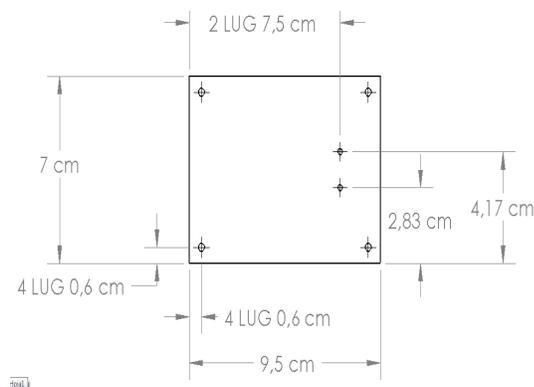


Figura 3.3: Placa de acrílico utilizada como base ó chasis para el robot móvil.

## Placa superior

La placa superior está hecha del mismo material que el chasis (acrílico), cuenta con cuatro perforaciones y fue cortada en la misma forma que la placa principal. Ver Figura 3.4.

Tanto el diseño como el material empleado para estas placas se justifica por el hecho de reducir el peso para el robot móvil. Además de que el acrílico puede permanecer largo tiempo, en la intemperie, sin sufrir daño alguno y es un buen aislante, permitiendo de esta manera evitar posibles cortos circuitos.

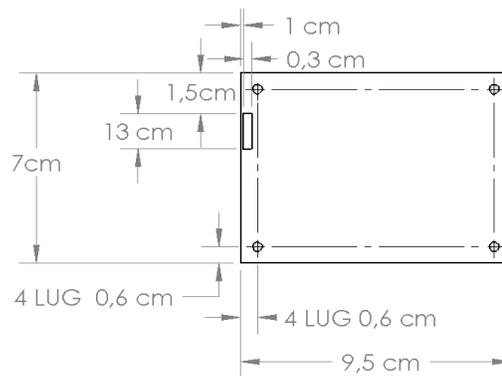


Figura 3.4: Placa superior de acrílico empleada para sostener la electrónica empleada en el robot móvil.

## Tornillos

Se emplean cuatro tornillos metálicos con una altura de 0,5 cm y un diámetro de 0,3 cm para unir la placa superior con el chasis. Se emplea este material por considerarlo más resistente que el acrílico. Sus especificaciones se ilustran de forma más detallada en la Figura 3.5.

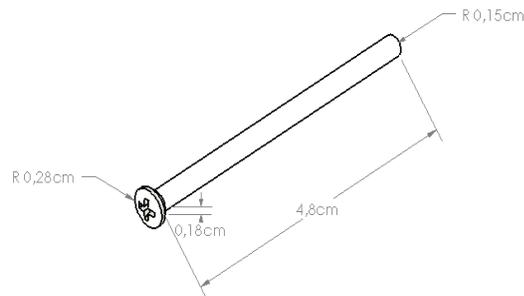


Figura 3.5: Tornillos utilizados para sujetar las placas de acrílico.

### (b). Sistema de tracción

Una de las ventajas de utilizar la configuración unicyclo es en el desplazamiento, ya que al girar las ruedas motrices a la misma velocidad el móvil se desplaza en línea recta, si en algún momento se desea que él mismo gire hacia la derecha o izquierda, lo cual se puede detener o disminuir la velocidad de uno de los motores y el robot móvil da vuelta tomando como eje de giro el punto de contacto de la rueda que esté detenida, en caso de girar en el otro sentido el eje de giro es el que esté en turno sin girar, analizando estos movimientos se observa que el desplazamiento es mínimo al dar vuelta, además que no se requiere de otro motor que haga la función de eje de dirección como es el caso de un robot de cuatro ruedas (por ejemplo el automóvil).

El diseño del sistema de tracción se muestra en la Figura 3.6 y está básicamente conformado por los siguientes elementos:

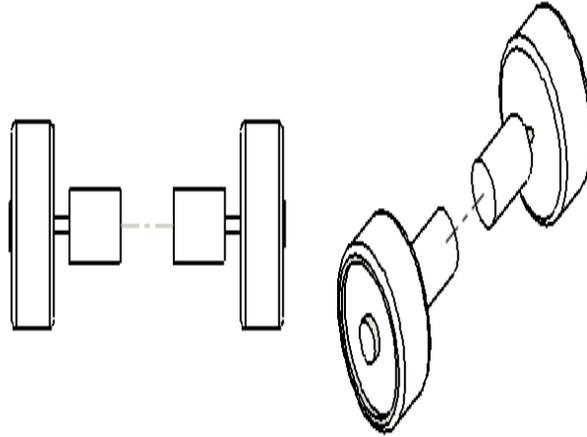


Figura 3.6: Sistema de tracción.

- Dos motoredutores de corriente directa CD. El propósito de estos motoredutores es servir como actuadores en el sistema de tracción. Las características que se conocen de estos son: voltaje de alimentación  $12V$  de CD a  $200\text{ mA}$ , una velocidad nominal de  $220\text{ RPM}$ .
- Dos llantas convencionales. Este par de ruedas presentan la característica de que son rugosas, tienen de diámetro  $4.2\text{ cm}$ , ancho  $1.9\text{ cm}$  y un peso de  $16\text{ g}$ ., ver Figura 3.7.



Figura 3.7: Llantas que utiliza el robot móvil.

### (c). Sistema de dirección

Se emplea una tercer rueda que es de giro libre o “rueda loca” que está adosada a un eje que no es concéntrico permitiéndole así presentar la menor resistencia al desplazamiento del móvil y su función principal es de apoyo. Esta rueda es la encargada de conferir estabilidad mecánica al sistema con un ancho de  $1.905\text{ cm}$  y una altura que puede ser ajustada debido a que cuenta con dos espaciadores, tal como se muestra en la Figura 3.8.

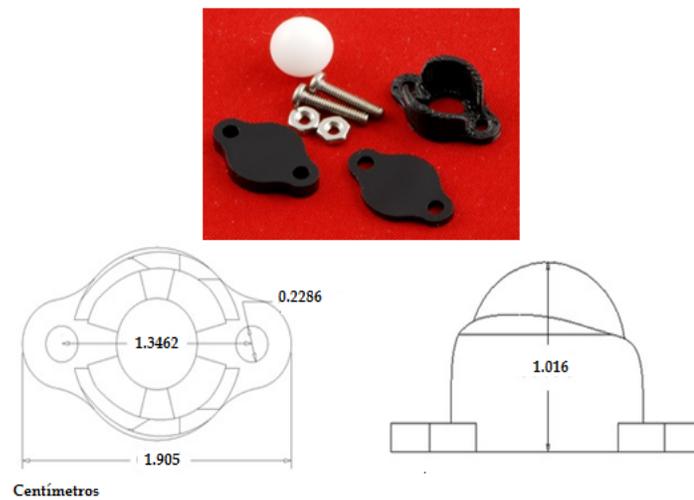


Figura 3.8: Dimensiones de la rueda loca.

## 3.3. Diseño electrónico

En esta etapa se describe el regulador de voltaje para la alimentación de la tarjeta AD y de los motoreductores, así como también se dan a conocer las características del *driver* utilizado para el cambio de giro de los motores. Por otra parte, también se pueden apreciar los elementos básicos que conforman la etapa electrónica del robot móvil, como son: la fuente de alimentación y el microcontrolador principalmente.

### 3.3.1. Fuente de alimentación

Debido a que el robot móvil es autónomo, es decir, no depende de una conexión a CA o de algún otro tipo que involucre la conexión con cables a una fuente externa, existe la necesidad de integrar esta etapa que se encarga de suministrar el voltaje adecuado para el microcontrolador y para los dos motores que se utilizan, así que

se diseñan y construyen dichas fuentes para que cumpla con las exigencias de estos circuitos, para lograr dicho propósito se utiliza una batería recargable, y reguladores de voltaje.

La Figura 3.9 muestra una batería recargable Nickel-Metal, cuadrada de (9V a 300mA) la cual cuenta con las siguientes características:

- Dimensiones  $2.5 \times 4.8 \times 1.5 \text{ cm}$ .
- Para uso general.



Figura 3.9: Batería recargable Nickel-Metal de 9V, 300mA.

### Regulación de voltaje

Para cumplir con el voltaje de alimentación (5V) se utiliza el CI 7805.

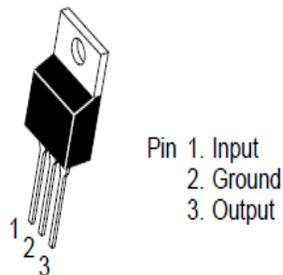


Figura 3.10: Disposición de pines del regulador de voltaje.

Este componente electrónico es un regulador de voltaje positivo de tres terminales, el cual suministra un voltaje de 5V con una corriente de salida máxima de 1A, este componente regula el voltaje utilizado en la alimentación para el microcontrolador. Mientras que para regular el voltaje de alimentación (12V) de los dos motores se utiliza un CI 7812, que cuenta con la misma disposición de pines que el CI 7805. El diseño de las fuentes de tensión se muestra en la Figura 3.11.

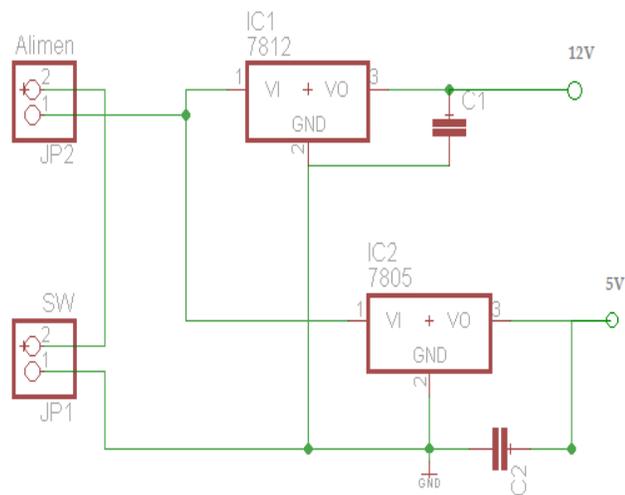


Figura 3.11: Fuente de alimentación de 5V y 12V.

### 3.3.2. Sistema de control

Un elemento importante de este sistema es el microcontrolador de la placa Arduino, este CI es el que recibe por medio del módulo de comunicación inalámbrico XBee los datos del estado de los robots móviles, los cuales son estimados por el sistema de visión. Las funciones que desempeña este microcontrolador es el envío de señales de control para los motores, entre otras. La Figura 3.12 muestra la disposición del microcontrolador ATmega168 de la tarjeta Arduino.

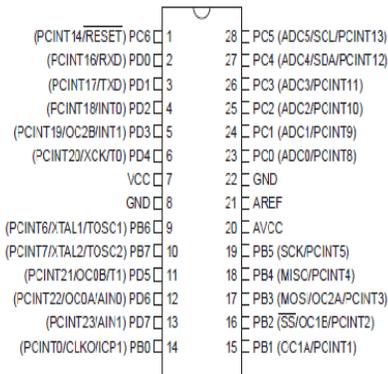


Figura 3.12: Disposición de pines del microcontrolador empleado en el control del robot móvil.

### (a). Microcontrolador Arduino Duemilanove

El Arduino Duemilanove (AD) es una placa con microcontrolador basada en el ATmega168 (datasheet) o el ATmega328 (datasheet). Tiene 14 pines con entradas-salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a  $16\text{Mhz}$ , conexión USB, entrada de alimentación, una cabecera ISCP, y un botón de reset. Contiene todo lo necesario para utilizar el microcontrolador; simplemente se conecta a una computadora a través del cable USB o se alimenta con un transformador o una batería para trabajar con él.

#### Alimentación

El AD puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

Los pines de alimentación son los siguientes:

- VIN. La entrada de voltaje para la tarjeta Arduino cuando se esté utilizando una fuente de alimentación externa (en comparación con  $5V$  de la conexión USB). Se puede suministrar voltaje a través de este pin.
- $5V$ . La fuente entrega un voltaje para alimentar el microcontrolador y otros componentes de la placa. Esta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de  $5V$ .

- 3.3V. Una fuente de 3.3V generados por el CI FTDI, con corriente máxima de 50mA.
- GND. Pines de tierra.

## Memoria

El ATmega328 tiene 32KB (el ATmega168 tiene 16KB) de memoria flash para almacenar código (2KB son usados para el arranque del sistema (bootloader). El ATmega328 tiene 2KB (Atmega168 1KB) de memoria SRAM. El ATmega328 tiene 1KB (ATmega168 512 bytes) de EEPROM, a la cual se puede acceder para leer o escribir con la librería EEPROM.

## Entrada y salida

Cada uno de los 14 pines digitales en el Duemilanove se puede utilizar como entrada o salida, utilizando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Cada pin puede proporcionar o recibir un máximo de 40mA y tiene un resistencia interior pull-up (desconectado por defecto) de 20 a 50KOhm. Además, algunos pines tienen funciones especializadas, como lo son:

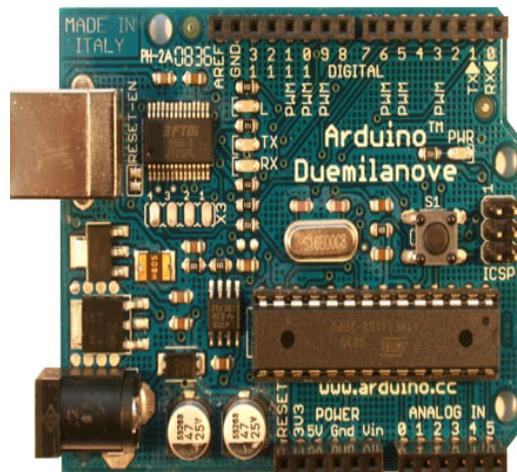
- Serie: 0 ( RX ) y 1 ( TX ) de datos. Usado para recibir (RX) transmitir (TX) datos a través del puerto serie TTL. Estos pins están conectados a los pines correspondientes del chip FTDI USB-to-TTL.
- PWM: 3, 5, 6, 9, 10, y 11. Proporciona una salida PWM (Modulación por ancho de pulso) de 8 bits de resolución (valores de 0 a 255) a través de la función `analogWrite()`.
- LED: 13. Hay un built-in LED conectado al pin digital 13. Cuando el pin es de valor HIGH, el LED está encendido, cuando el pin es LOW, es apagado.

El Duemilanove tiene 6 entradas analógicas, cada uno de los cuales ofrecen 10 bits de resolución (es decir, 1024 valores diferentes).

- AREF. Voltaje de referencia para las entradas analógicas. Se utiliza con la función `analogReference()`.
- Reset. Restablece al microcontrolador.

## Comunicación

El Arduino Duemilanove contiene una serie de facilidades para comunicarse con una computadora, otro Arduino u otros microcontroladores. El ATmega168 y ATmega328 proporcionar UART TTL (5V) de comunicación en serie, que está disponible en los pines digitales 0 (RX) y 1 (TX). El Arduino Duemilanove puede ser programado bajo su propio software Arduino.



Microcontrolador	ATmega328 (ATmega168 en versiones anteriores)
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pines E/S digitales	14 (6 proporcionan salida PWM)
Pines de entrada analógica	6
Intensidad por pin	40 mA
Intensidad en pin 3.3V	50 mA
Memoria Flash	16 KB (ATmega168) o 32 KB (ATmega328) de las cuales 2 KB las usa el gestor de arranque(bootloader)
SRAM	1 KB (ATmega168) o 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) o 1 KB (ATmega328)
Velocidad de reloj	16 MHz

Figura 3.13: Características principales de Arduino Duemilanove.

### (b). Módulo de comunicación XBee

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas *IEEE\_802.15.4*. La Figura 3.14 muestra las conexiones mínimas que necesita el módulo XBee para poder ser utilizado. En seguida de esto, se debe configurar según el modo de operación adecuado para la aplicación requerida, para esta aplicación se configura de “Modo recibir/transmitir”.

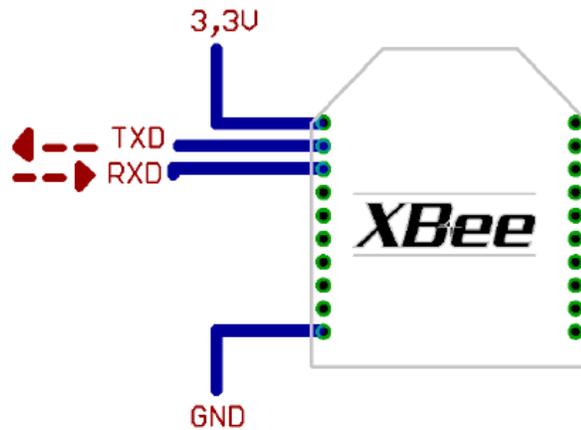


Figura 3.14: Conexiones mínimas requeridas para el XBee.

El módulo requiere una alimentación que va desde 2.8 a 3.4V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún convertidor adecuado para los niveles de voltaje.

El módulo de comunicación utilizado en este trabajo es el **XBee SER1, 1MW, RPSMA ANT CONN, 250KBP**.

### (c). Placa principal

La Figura 3.15 muestra el diseño de la placa principal, donde sus componentes son:

- Alimentación.
- Switch.
- Reguladores de voltaje (7805 y 7812).
- Puente H L298.
- XBee Explorer Regulated.
- XBee SER1, 1MW, RPSMA, con antena.
- Headers (acoplamiento con la placa Arduino).



PWM	Dirección	Frenado	Salida del dispositivo
H	H	L	Giro
H	L	L	Giro
L	X	L	Detenido, hay voltaje en ambas salidas
H	H	H	Detenido, hay voltaje en ambas salidas
H	L	H	Detenido, no hay voltaje en ambas salidas
L	X	H	Nada

Tabla 3.1: Tabla de verdad para el control de giro de los motores, en donde L=Estado bajo, H=Estado alto y X=Estado de no importa.

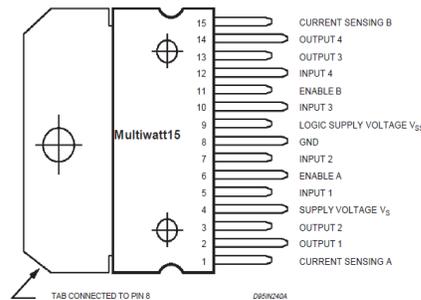


Figura 3.16: Disposición de pines del puente H empleado para el control de los motores.

### 3.4. Construcción del robot móvil

En esta etapa se ilustra la construcción del robot móvil con llantas en configuración uniclo, para esto, se muestran fotografías tomadas tanto del sistema electromecánico como de la electrónica empleada.

#### 3.4.1. Prototipo

Finalmente, se logra construir un prototipo como el que se muestra en la Figura 3.17 con un desempeño aceptable en cada uno de sus sistemas. Para propósitos de la descripción del robot móvil final, se divide esta sección en dos:

- Sistema electromecánico.
- Sistema electrónico.

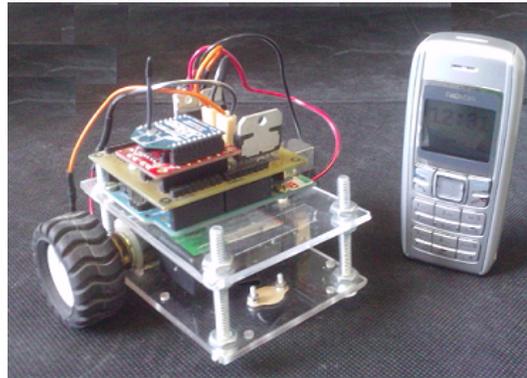


Figura 3.17: Robot móvil con llantas en configuración uniciclo.

### 3.4.2. Etapa electromecánica

Como se vió al inicio de este capítulo, esta etapa se divide en 3 partes, en cuanto al sistema de soporte se refiere, la placa principal (chasis), los postes y la placa superior (acrílico) son mostradas en las Figuras 3.3, 3.4 y 3.5, respectivamente.

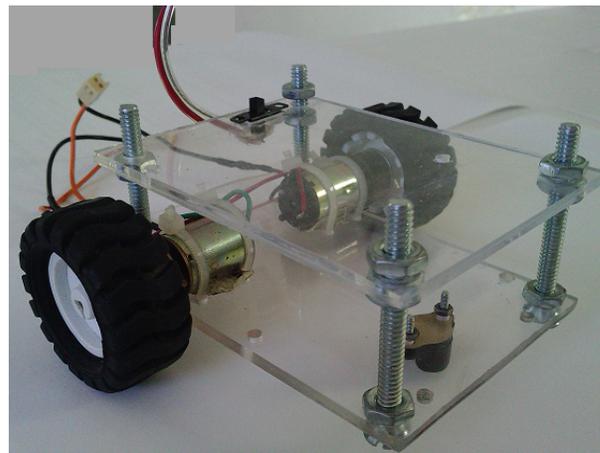


Figura 3.18: Sistema electromecánico final, vista isométrica.

Por otro lado, el sistema de tracción como se puede apreciar en la Figura 3.18, está conformado por dos motoredutores, el cual a su eje se le acoplo una llanta convencional.

### 3.4.3. Módulo electrónico

Una vez concluida la etapa electromecánica del prototipo, se continua con la construcción del módulo electrónico, esta etapa está formada por una tarjeta Arduino Duemilanove que es la que contiene al microcontrolador, una tarjeta diseñada bajo la plataforma EAGLE Layout Editor 5.6.0 que cuenta con su fuente de alimentación, un puente H, como el que se muestra en la Figura 3.16, los cuales son utilizados en el sistema tracción, además esta placa cuenta con conectores para el acople con la tarjeta Arduino, el módulo XBee, la batería y los motores, las Figuras 3.19 y 3.20 ilustran la descripción anterior.

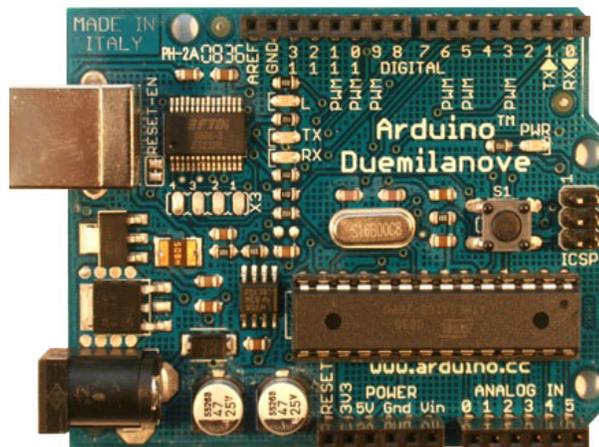


Figura 3.19: Tarjeta Arduino Duemilanove.

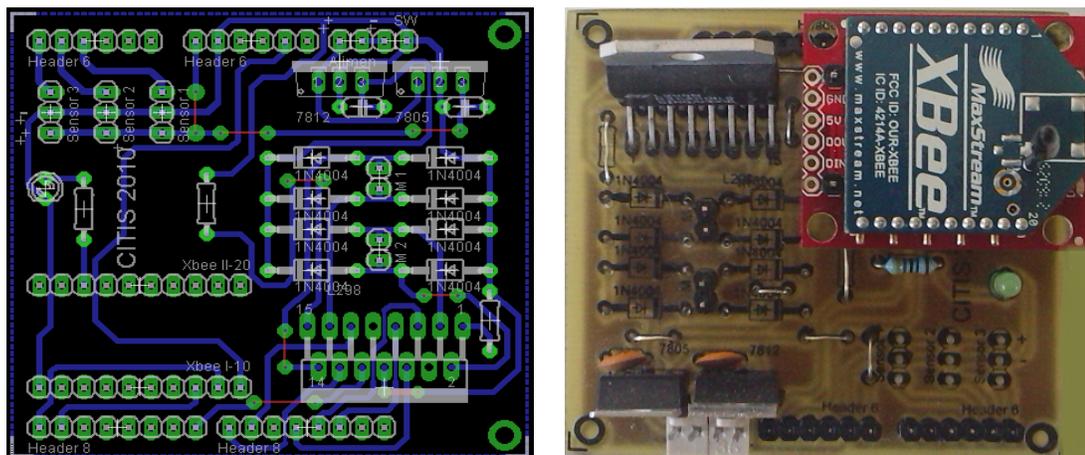


Figura 3.20: Tarjeta electrónica (diseño en pistas y construcción).

### **3.5. Comentarios**

Cabe destacar que no se conocen más características de los motores debido a que no se cuenta con su hoja de especificaciones. Por otra parte, una vez concluido el diseño de cada una de las piezas que conforman el robot móvil, se ha presentado la instrumentación necesaria para tener comunicación electrónica entre las partes que conforman el robot móvil. Una vez que se tiene el diseño se presenta el prototipo final obtenido durante el desarrollo de este trabajo de tesis.

# Capítulo 4

## Estimación visual

El objetivo de este capítulo es mostrar el desarrollo del algoritmo empleado para la estimación de la posición y orientación de cada uno de los tres robots móviles.

Este capítulo está organizado de la siguiente manera: La introducción se presenta en la Sección 4.1, mientras que el método para la calibración de la cámara se da en la Sección 4.2, los algoritmos propuestos para la estimación de la ubicación de cada uno de los robots móviles se muestran en la Sección 4.3, los resultados de la metodología propuesta para la detección de la ubicación de los robots se ilustran en la Sección 4.4. Finalmente, los comentarios referentes a este capítulo se hacen en la Sección 4.5.

### 4.1. Introducción

El advenimiento de la tecnología ha permitido que algoritmos de control cada vez más complejos sean posibles implementarlos en el control de sistemas no lineales. El uso de una cámara de visión como sensor en sus inicios, en los años 70's se empleaban en lazo abierto debido al alto costo computacional que los algoritmos demandan, y en robótica un esquema de visión era imposible de implementarlo en lazo cerrado, normalmente era utilizado con fines de vigilancia, sin embargo en la actualidad ya es posible emplear una cámara de visión como sensor en el lazo de control y de esta manera tener un esquema de *visual servoing* en tiempo real.

Por otro lado el sistema de reconocimiento de objetos en imágenes ha despertado en los últimos años un gran interés debido a sus posibilidades de aplicación en diversos campos.

El objetivo es encontrar un objeto predefinido en una imagen. A veces esta tarea consiste en extraer una cierta característica o rasgo de la imagen, como pueden ser bordes, zonas de color, texturas, contornos, etc. y después usar métodos heurísticos

para encontrar alguna configuración o combinación de esos rasgos específicos en el objeto de interés.

Para objetos complejos, como caras humanas, células, etc., es difícil encontrar reglas que puedan manejar la gran variedad de casos y diferencias que dichos objetos pueden albergar. Por ejemplo, en el caso de la detección de células, tienen las distintas formas de las mismas, variedad de tamaños de sus núcleos, la nitidez de las imágenes tomadas, las distintas tonalidades que provocan los tintes aplicados, etc. A la hora de detectar caras humanas, se encuentran caras giradas, gafas, bigotes, barbas, entre otros.

OpenCV [64] incorpora, gracias a la implementación de R. Lienhart [42], las funciones necesarias tanto para entrenar un clasificador basado en el marco general de detección de objetos de Viola-Jones [39],[40] como también las funciones para el uso del clasificador.

En este trabajo se implementa el algoritmo de detección de Viola-Jone como sistema de detección de objetos en imágenes, que consiste en dividir primero la imagen en subventanas y después evaluar cada una para averiguar si contiene o no al objeto que se busca. La forma en que se decide si esa subventana contiene al objeto depende del método que se utilice, pero el sistema de detección siempre se basa en ciertas características propias del objeto en cuestión. Para decidir si una subventana es el objeto a buscar o no, OpenCV utiliza técnicas “*boosting*”, por ejemplo *Discrete Adaboost*, *Real Adaboost*, *Gentle Adaboost* y *Logitboost*.

Viola-Jone utilizan el *boosting* (AdaBoost) que se basa en la aplicación de sencillas reglas, a las que únicamente se les exige que sean un poco mejores que si se tomara la decisión sobre la subventana de forma aleatoria. La combinación de varias de esas reglas son las que permiten obtener un buen sistema de detección. El *boosting* es el método que permite combinar esas reglas y construir el detector de objetos. El *boosting* se basa en la aplicación de sencillas reglas, a las que únicamente se les exige que sean un poco mejores que si se toma la decisión sobre la subventana de forma aleatoria. La combinación de varias de esas reglas son las que permiten obtener un buen sistema de detección.

## 4.2. Calibración de la cámara

Para la calibración de la cámara se hace uso del Toolbox de Matlab [63] llamado *Camera Calibration* el cual permite obtener la matriz  $K$  de parámetros intrínsecos

de la misma, utilizando para esto el patrón de calibración mostrado en la Figura 4.1, dando como resultado

$$K = \begin{bmatrix} 451.12250 & 0.00000 & 319.50000 \\ 0.00000 & 451.12250 & 239.50000 \\ 0.00000 & 0.00000 & 1.00000 \end{bmatrix} \quad (4.1)$$

donde:

$K(1, 1), K(2, 2)$ : Distancia focal en unidades pixelares.

$K(1, 3), K(2, 3)$ : Punto principal *cm*.

$K(1, 2)$ : coeficiente de asimetría que define el ángulo entre los ejes  $x$  y  $y$  multiplicado por  $K(1, 1)$ .



Figura 4.1: Patrón de calibración.

### 4.3. Estimación de la ubicación del robot móvil

En esta sección se presenta la metodología empleada para estimar la posición y orientación de cada uno de los robots móviles en su espacio de trabajo. Para esto, se han usado los módulos: Cv, CvAux, CxCore, HighGUI y CvCam (ver Apéndice D) la cual tiene la interconexión estructural mostrada en la Figura 4.2.

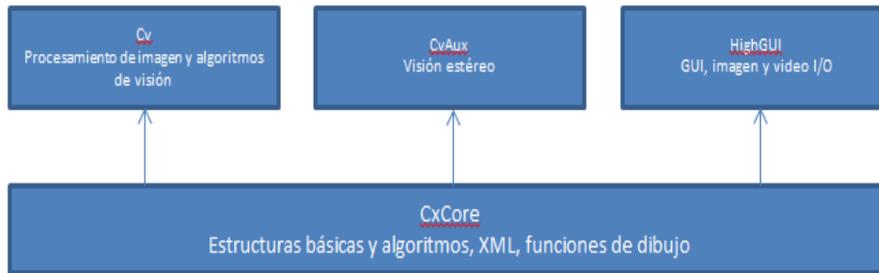


Figura 4.2: Estructura usada de OpenCV.

Pasos empleados en la metodología propuesta para la estimación de la ubicación de los robots móviles:

1. Adquisición y visualización de video.

Para la adquisición y visualización de las imágenes captadas por la cámara empleada (webcam) se realiza un programa en *C++* (mostrado en el Apéndice D) que utiliza la librería OpenCV, este programa permite muestrear las imágenes en tiempo real a modo video. El Algoritmo 4.1 muestra el procedimiento empleado en la adquisición y visualización de video.

<p>Algoritmo 4.1: Adquisición y visualización de video</p>
--

- |   |
|---|
| <ul style="list-style-type: none"> <li>■ Para mostrar una imagen, se ha de seguir el siguiente procedimiento             <ol style="list-style-type: none"> <li>a) <b>Se carga la imagen desde el fichero</b> <code>IplImage *img = cvLoadImage(char [] filename, int type)</code>. siendo <code>Filename</code>: la imagen a cargar. <code>Type</code>: hace referencia a los colores de la imagen <code>CV_LOAD_IMAGE_COLOR = 1</code>, la imagen resultante es en color. <code>CV_LOAD_IMAGE_GRAYSCALE = 0</code>, la imagen es en escala de grises.</li> <li>b) <b>Se crea la ventana donde se muestra la imagen</b> <code>cvNamedWindow(const *char name, int flags)</code>. <code>name</code>: nombre con el que se identificará la ventana. <code>flags</code>: Actualmente, la única bandera es <code>CV_WINDOW_AUTOSIZE</code>, si se establece, entonces el tamaño de la ventana se ajusta automáticamente a la imagen que se muestra.</li> <li>c) <b>Se muestra la imagen</b> <code>cvShowImage(const *char name, const CvArr* img)</code>. <code>name</code>: la ventana donde se va a mostrar la imagen. <code>img</code>: la imagen a mostrar.</li> </ol> </li> <li>■ Mostrar la captura de la webcam:             <ol style="list-style-type: none"> <li>a) <b>Se inicia la webcam</b> <code>capture = cvCaptureFromCAM( int index )</code>. <code>index</code>: el índice asociado a la webcam (normalmente 0)</li> <li>b) <b>Captura por la webcam</b> <code>img = cvQueryFrame (CvCapture *capture)</code>. <code>capture</code>: la estructura creada anteriormente asociada a la webcam.</li> <li>c) <b>Mostrar la captura</b> <code>cvShowImage(const *char name, const CvArr* img)</code>. <code>name</code>: la ventana donde se va a mostrar la imagen. <code>img</code>: la imagen a mostrar.</li> </ol> </li> </ul> |
|---|

## 2. Diseño del clasificador

Para la estimación de la posición y orientación de cada robot móvil se optó por diseñar un clasificador empleando algoritmos existentes dados en [40], una de las desventajas de emplear este tipo de algoritmos es que los parámetros de entrada tienen que cumplir con las especificaciones requeridas en la sintaxis, lo cual implica una serie de pasos y procedimientos previos.

A continuación se explica brevemente los pasos requeridos para poder emplear el algoritmo propuesto en [40]:

- Contar con dos bases de datos de imágenes tomadas con la webcam a diferentes ángulos y diferentes alturas, una con imágenes positivas y otra con imágenes negativas. Se llama positiva a la imagen que contiene los motivos a ser reconocidos por el clasificador, mientras que una imagen negativa es aquella que no contiene los motivos que se desea reconocer. En este clasificador se requiere de ambas. Es importante mencionar que debido a que se tienen tres robots móviles, cada uno representado con un motivo diferente (ver figura 4.3) se tiene como consecuencia la realización de

tres clasificadores. Sin embargo para ahorrar cálculo computacional, solo se diseñaron dos clasificadores, ya que dos de los robots son etiquetados con motivos complementarios.

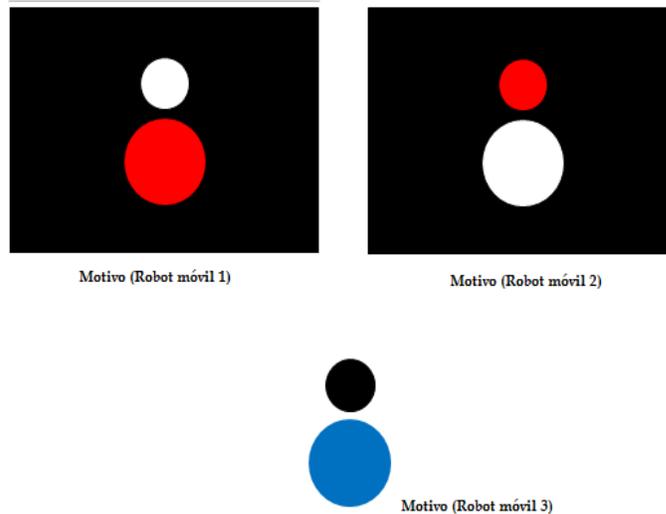


Figura 4.3: Muestras positivas.

- A continuación se realiza un programa en C++ con la finalidad de manipular cada una de las imágenes positivas contenidas en una de las bases de datos, donde se selecciona la región que contiene a los motivos, el algoritmo arroja como resultado las coordenadas de posición  $x$ ,  $y$ , el ancho  $w$  y el alto  $h$  a la que fue tomada la imagen, y toda esta información se registra en un archivo de texto.
- Para las imágenes negativas almacenadas en la otra base de datos solo se genera un archivo de texto, donde se almacena la dirección de ubicación de la imagen.
- Una vez se tiene localizado en cada imagen el objeto de interés se crea un archivo *.vec* que recoge esa información mediante la instrucción:
 

```
createsamples -info positivas/info.txt -vec data/vector.vec -num 800 -w 24 -h 24
```

 siendo:
  - info*: la ubicación del archivo con el índice de las imágenes positivas.
  - vec*: es el nombre del archivo de salida con la muestra generada.
  - num*: cantidad de imágenes positivas
  - w*: ancho de la muestra de salida.
  - h*: alto de la muestra de salida.

- Finalmente, se pasa a la etapa de entrenamiento empleando para esto la salida del paso anterior que es el archivo `.vec` y el archivo de texto que contiene las direcciones de las imágenes negativas. La instrucción empleada para esto es la siguiente:

```
haartraining -data data/cascada vec data/vector.vec -bg  
negativas/infofile.txt npos 420 -nneg 160 -nstages 20 -mem 1024  
-mode ALL -w 24 -h 24
```

siendo:

`-data`: directorio de salida para la generación del entrenamiento.

`-vec`: archivo con muestras generado en la etapa anterior.

`-bg`: localización del fichero con la información de las imágenes negativas.

`npos`: número de imágenes positivas a analizar.

`nneg`: número de imágenes negativas.

`nstages`: número de etapas de entrenamiento. Es directamente proporcional a la calidad del clasificador a generar.

`mem`: memoria dedicada al proceso.

`mode`: modo de ejecución.

`w`: ancho.

`h`: alto.

Mientras se está ejecutando genera una salida de pantalla similar a la que se muestra en la Figura 4.4:

```
[...]  
STAGE TRAINING TIME: 5037.31  
STAGE: 17  
POS:240 240 1.000000  
NEG: 160 1308777143 0.000004  
BACKGROUND PROCESSING  
TIME: 26671.78  
PRECALCULATION TIME: 108.59  
[...]  
STAGE TRAINING TIME: 5389.59  
STAGE: 18  
POS:240 240 1.000000  
NEG: 160 -1465156860 -0.000003  
BACKGROUND PROCESSING  
TIME: 58371.50  
PRECALCULATION TIME: 108.56
```

Figura 4.4: Ejecución haartraining.

Una vez que ha finalizado de ejecutarse se tiene como salida el clasificador para el reconocimiento de los motivos, generando un archivo `.xml`.

### 3. Obtención de las coordenadas y de la orientación

En este último paso de la metodología propuesta para detectar la posición y orientación de cada uno de los robots móviles, se hace uso del archivo *.xml* generado en el paso anterior, junto con las funciones *CvHaarClassifierCascade* y *cvHaarDetectObjects*. A continuación, se describe el algoritmo realizado para este propósito:

(a). Una vez que se tiene el clasificador para cada uno de los motivos montados en los robots móviles, se procede a la implantación en tiempo real, para esto se toman imágenes con la webcam de los robots en el espacio de trabajo, gracias al clasificador *.xml* se toma solo la región de interés que contiene a los motivos, trazando mediante software un rectángulo como se muestra en la Figura 4.5.

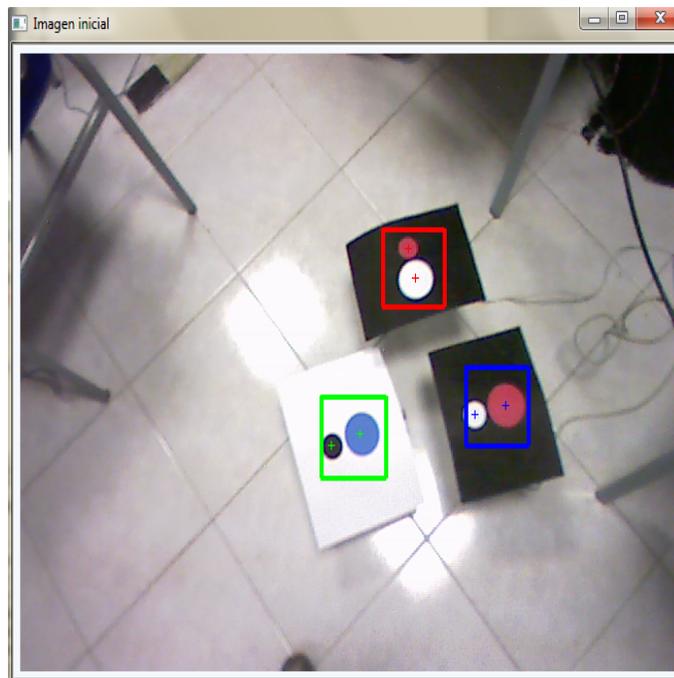


Figura 4.5: Detección de objetos (motivos).

(b). Con la ayuda del rectángulo dibujado en el paso anterior, se obtiene el centroide de coordenadas  $(x, y)$  que son las coordenadas de la posición del robot móvil. Dichas coordenadas se sitúan en el espacio de trabajo y están relacionadas en el plano imagen mostrado en la Figura 4.6 por (4.2):

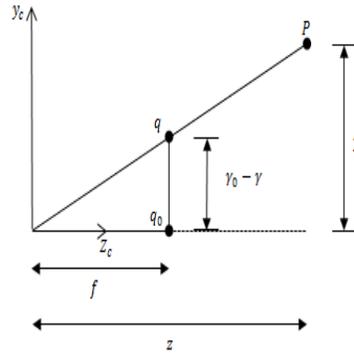


Figura 4.6: Coordenadas en el plano imagen (espacio de trabajo).

$$\frac{\gamma_0 - \gamma}{f} = \frac{y}{z}; y = \frac{(\gamma_0 - \gamma)z}{f} \quad (4.2)$$

$$\frac{\rho_0 - \rho}{f} = \frac{x}{z}; x = \frac{(\rho_0 - \rho)z}{f}$$

(c). Una vez que se conocen las coordenadas  $(x, y)$  en el espacio de trabajo, se procede a obtener la orientación de cada uno de los robots móviles, para ello se han colocado objetos circulares dentro de la imagen de diferentes colores para encontrar su centroide, éste forma la base para encontrar la orientación.

Para detectar un objeto de determinado color se necesita detectar todos los píxeles que lo componen. En OpenCV se puede acceder a las características de la imagen y gracias a estos datos se hace el recorrido de la imagen.

Para ello se buscan en todos los píxeles de la imagen (recorrido de la imagen), calculando si son o no del color deseado. ¿Cómo saber si por ejemplo es rojo?. En primer lugar se observa qué datos característicos tiene dicho color. En este caso, su dato más característico es que el canal con mayor valor es el del rojo. También, se tiene en cuenta que los valores de los demás canales no son muy altos, por lo menos no lo suficiente como para acercarse al valor del canal del rojo, ya que estos valores pueden corresponder a colores como el morado o el naranja.

Con un color que no sea de los tres básicos esto se complica un poco; en este caso lo mejor es abrir la imagen con un editor de imágenes (Paint por ejemplo), y mirar que valores RGB tienen los píxeles del color deseado. Así se pueden obtener las características que determinan dicho color.

Una vez que se tienen todos los píxeles del objeto se lleva un contador de cuántos píxeles se tienen, y una suma de sus posiciones, lo que sirve para calcular lo que es el punto medio del objeto tal y como se muestra en la Figura 4.7. Estas

coordenadas del punto medio son precisas si contamos con una gran cantidad de píxeles del objeto, este dato es de vital importancia debido a que es posible encontrar la orientación del robot móvil, empleando para esto el triángulo rectángulo que se forma con los dos centroides, lo cual resulta que la orientación está dada por

$$\theta = \text{atan2}(y, x) \quad (4.3)$$

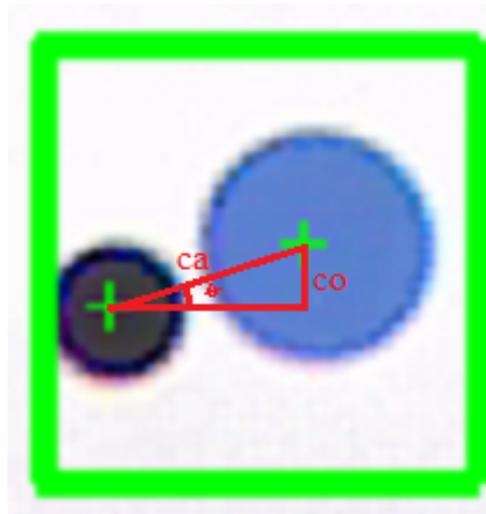


Figura 4.7: Ángulo de orientación.

## 4.4. Resultados obtenidos

Finalmente, el resultado del algoritmo de visión se puede ver la Figura 4.8, donde se hace el reconocimiento de objetos para posteriormente obtener la posición del robot móvil ya que como se mencionó anteriormente las imágenes son montadas sobre los robots.

Esta posición y orientación se imprime en pantalla y en un archivo de texto como se muestra en la Figura 4.8.

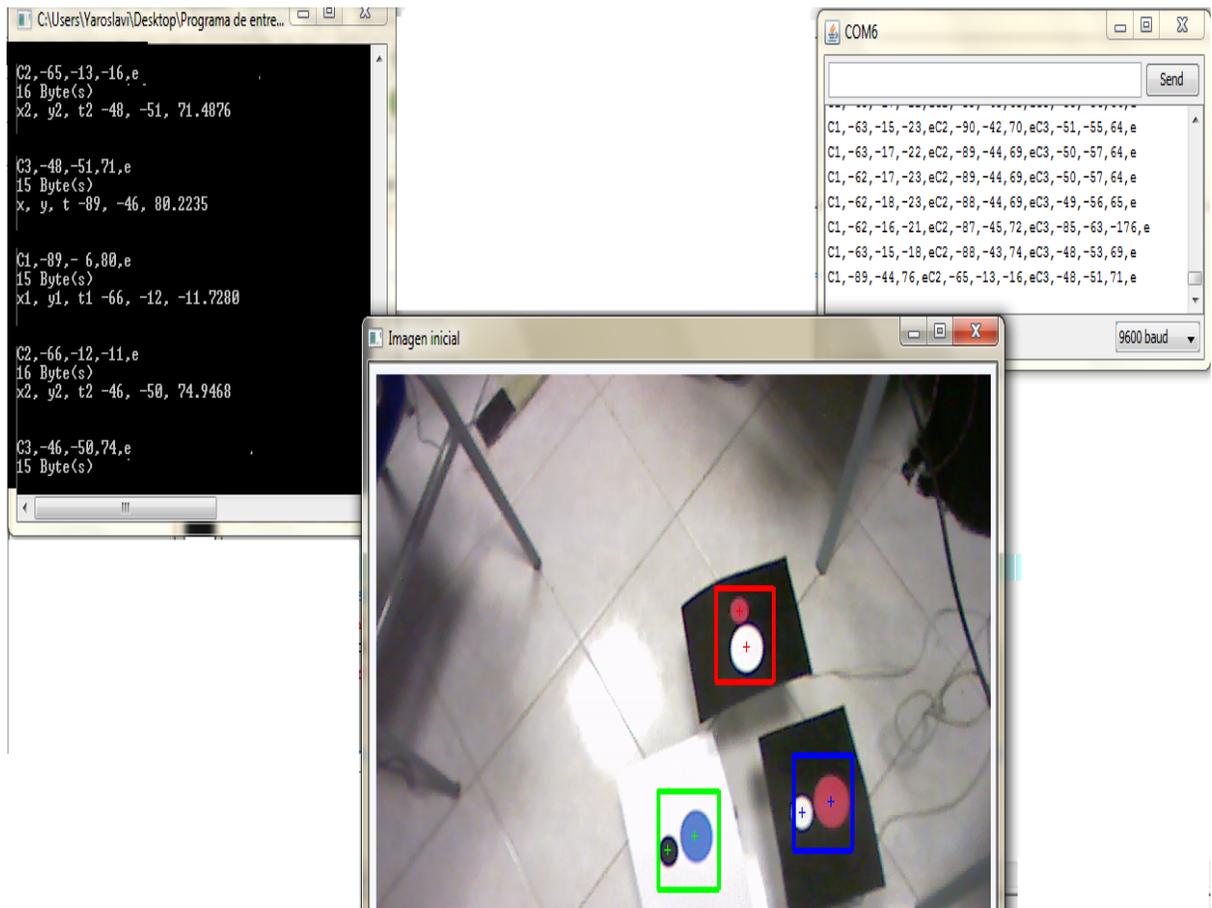


Figura 4.8: Resultado final del algoritmo de visión.

## 4.5. Comentarios

El empleo de un esquema de control que hace uso de una retroalimentación visual requiere de variables físicas estimadas mediante el método presentado en este capítulo y son calculadas en tiempo real para su utilización por la ley de control, es importante comentar que para la detección de las variables de ubicación de los robots dependen fuertemente de la iluminación en el espacio de trabajo. Es por ello que los resultados obtenidos de esta metodología propuesta son de vital importancia para el buen desempeño del controlador en lazo cerrado, el cual es diseñado en el siguiente capítulo.



# Capítulo 5

## Modelado y control

El objetivo de este capítulo es la obtención del modelo matemático del robot móvil en configuración unicycle (modelo cinemático y modelo dinámico) y el diseño de las leyes de control que se le aplican a cada uno de los robots.

La distribución de este capítulo está dada de la siguiente manera: En la Sección 5.1 se da una introducción referente al modelado y control de robots móviles, en la Sección 5.2 se dan las condiciones bajo las cuales se obtiene el modelo matemático en la Sección 5.3 del robot móvil terrestre. Mientras que en la sección 5.4 se obtienen las leyes de control empleando el método de Lyapunov, en la Sección 5.5 se da la implementación de la etapa de control. Finalmente, los comentarios se presentan en la Sección 5.6.

### 5.1. Introducción

Como se mencionó en el Capítulo 1 el diseño de las leyes de control están basadas en el modelo matemático que describe el comportamiento del robot móvil. En general un sistema puede representarse de muchas formas diferentes, por lo que puede tener muchos modelos matemáticos, dependiendo de cada perspectiva. En robótica existen básicamente dos tipos de modelos: el dinámico para los que se requiere el estudio del movimiento considerando las fuerzas (energía y velocidad, representadas como un conjunto de ecuaciones diferenciales) y el cinemático que involucran el estudio del movimiento sin considerar las fuerzas involucradas [49].

La cinemática se refiere, a cómo se mueve el robot, ésta puede ser [8]:

- Directa. En donde dada la posición inicial y los movimientos realizados, se calcula la posición final del robot.
- Inversa. En donde dada la posición inicial y final deseadas, se calcula la serie de movimientos que el robot debe realizar para cumplir con dichas posiciones.

Primeramente se obtiene el modelo cinemático directo que determina la posición y orientación del robot móvil en términos de las variables angulares del mismo. Para posteriormente obtener su modelo dinámico. Para el estudio de la cinemática de un robot móvil con ruedas es muy importante saber cómo está construido, cuántos motores tiene, qué tipo de llantas utiliza y sobre todo qué variables y parámetros está utilizando.

## 5.2. Suposiciones de trabajo

Para el desarrollo de este trabajo se ha elegido una arquitectura de tres ruedas y en específico la configuración en unicycle en donde se emplea una sola rueda delantera para dar estabilidad mecánica al sistema y dos ruedas traseras para la tracción, la arquitectura elegida es mostrada en la Figura 5.1.

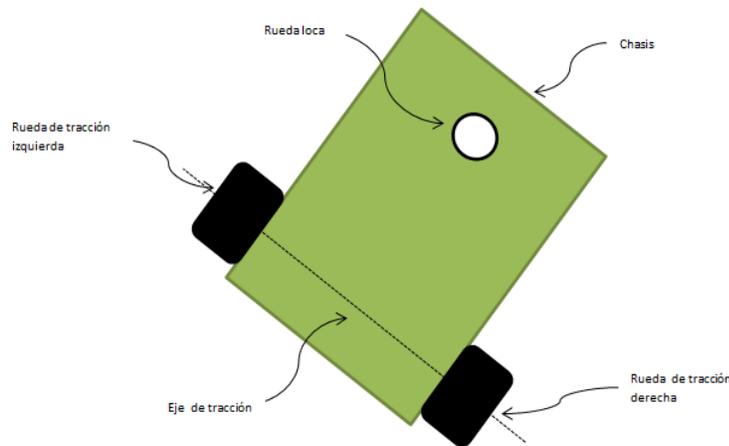


Figura 5.1: Componentes de un robot móvil en configuración de unicycle.

Las suposiciones bajo las cuales se trabaja y se consideran en la obtención del modelo son las siguientes:

- S1.** Se supone que el espacio del robot móvil es plano y sin obstáculos.
- S2.** Se supone que las llantas traseras no sufren derrapamientos.
- S3.** Se supone que el robot móvil no sufre deslizamiento.
- S4.** Se encuentra en un plano tal que toda la energía potencial es constante.

### 5.2.1. Restricciones cinemáticas

En un robot móvil con llantas se pueden presentar cualquiera de las siguientes restricciones:

- Holonómicas. En donde los diferentes grados de libertad están desacoplados, este tipo de restricción la presentan los robots diferenciales y síncronos, los cuales pueden desacoplar la posición de la orientación rotando sobre su eje, ver Figura 5.2.

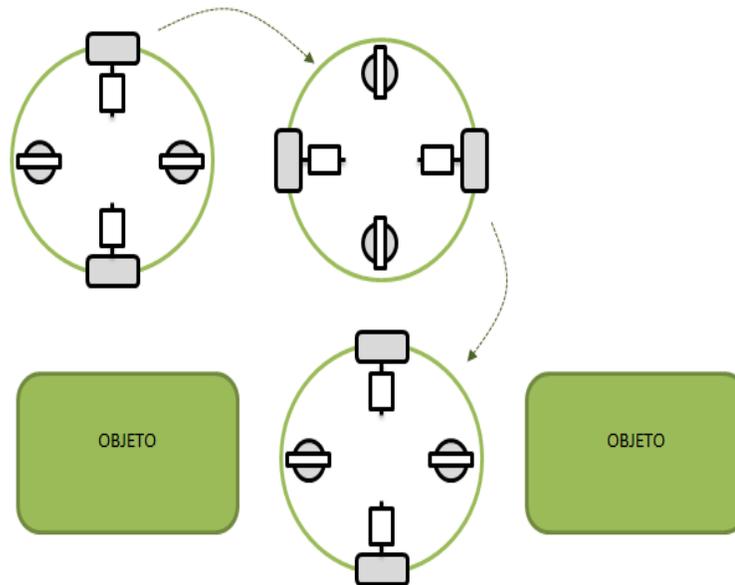


Figura 5.2: Restricción holónoma.

- No-holonómicas [34]. En donde los grados de libertad están acoplados, la presentan las configuraciones de uniciclo, triciclo y Ackerman, los cuales para dar vuelta deben moverse hacia el frente o hacia atrás, es decir, el robot no puede pasar instantáneamente de derecha a izquierda o de izquierda a derecha, necesitando así una serie de movimientos lo cual complica llegar a una posición final deseada, esto se ilustra en la Figura 5.3.

No-holonómico	Holonómico
Diseño y control más sencillo. Tienen menos grados de libertad. Actuadores con control independiente y/o mecánicamente desacoplados. Emplean un mayor número de maniobras. Se generan problemas de planeación de ruta y de implementar un comportamiento de reacción.	Diseño y control más complejo. Los grados de libertad son los mismos a los del entorno en el que se desenvuelve. Se desplazan con mayor facilidad. Planeación de ruta menos complicada. Comportamiento de reacción más sencillo de implementar.

Tabla 5.1: Algunas diferencias entre un robot móvil no-holonómico y uno holonómico.

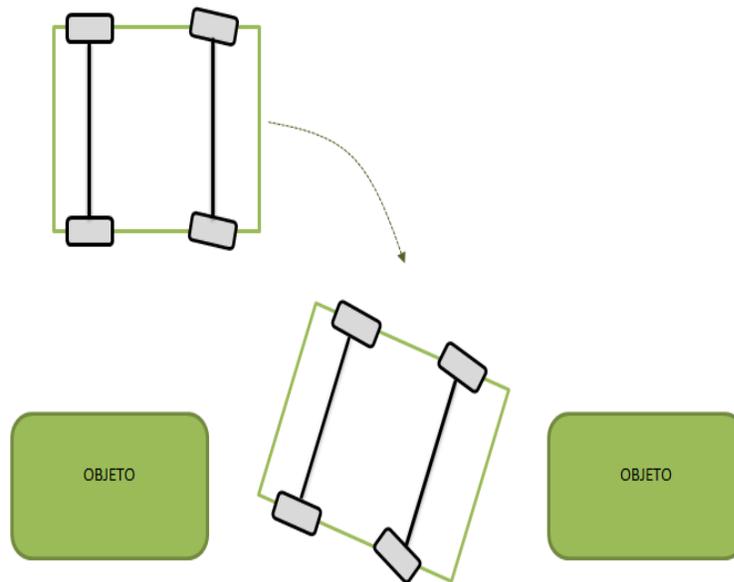


Figura 5.3: Restricción no holónoma, complejidad para llegar a una posición final.

Los robots móviles no-holonómicos son más predominantes debido a que demandan un diseño y un control más sencillo. Por su naturaleza este tipo de robots móviles tienen menos grados de libertad que un holonómico. A su vez, los actuadores de un no-holonómico son por lo regular de control independiente simplificando así su nivel de control [51]. Debido a que tienen menos grados de libertad existen algunos problemas con algunos movimientos que no pueden ser desarrollados. En la Tabla 5.1 presenta un resumen de las diferencias que existen entre estos dos tipos de restricciones.

### 5.3. Metodología propuesta para la obtención del modelo matemático

Para la obtención del modelo cinemático es necesario conocer las características del sistema, como son: variables y componentes del robot móvil, es decir:

- Componentes:
  - Tres ruedas:
    - Dos traseras.
    - Una delantera.
- Variables:
  - Dirección y tracción.
  - Comandos o señales de control:
    - Fuerza de desplazamiento lineal:  $F_l$ .
    - Par angular:  $\tau_a$ .

La Figura 5.4 muestra las variables y componentes del robot móvil y sus descripciones se dan en la Tabla 5.2.

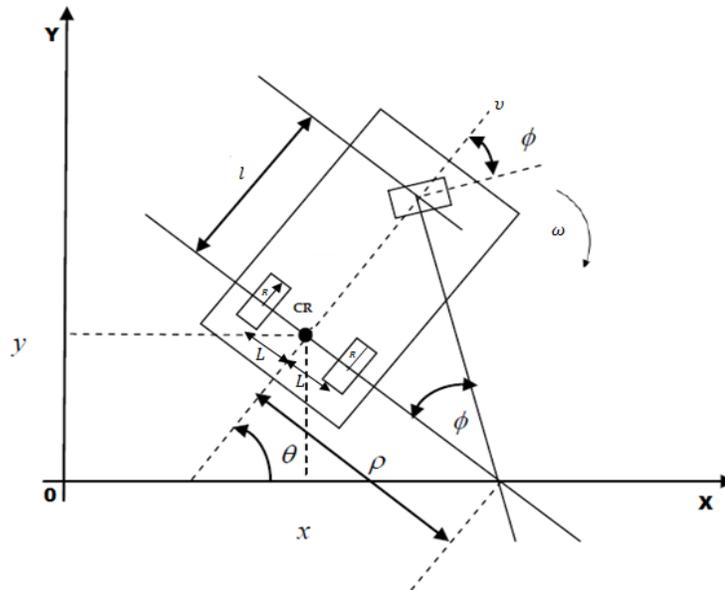


Figura 5.4: Variables, parámetros y componentes involucradas en el modelado matemático del robot móvil con ruedas.

Parámetro	Descripción	Unidad
$l$	Distancia entre los ejes de la rueda delantera y traseras.	$m$
$L$	Distancia de la rueda al CR.	$m$
$CR$	Centro de Rotación.	$m$
Variables		
$(x, y)$	Posición del vehículo.	$m$
$\theta$	Ángulo de orientación.	$rad$
$\rho$	Radio de giro.	$m$
$v$	Velocidad lineal de vehículo.	$m/seg$
$\phi$	Ángulo de dirección del vehículo.	$rad$
$w$	Velocidad angular del vehículo.	$rad/seg$

Tabla 5.2: Variables, parámetros y componentes del robot móvil en configuración unicyclo.

Pasos a seguir en la obtención del modelo matemático del robot móvil tipo unicyclo:

1. Definir un eje de referencia inercial ( $X, Y$ ) el cual se encuentra fijo en el espacio de trabajo del robot como se muestra en la Figura 5.4.
2. Considerar el punto  $(x, y)$  en el plano cartesiano, que es el centro de rotación ( CR ) del eje de unión entre las ruedas traseras motrices, para poder medir la posición del cuerpo del robot móvil. Además, se obtienen las proyecciones del punto  $(x, y)$  sobre el eje de referencia inercial fijo ( $X, Y$ ).
3. A continuación se definen las ecuaciones que describen el modelo cinemático:

Como ya se mencionó, la cinemática es la encargada del estudio del movimiento que trata a éste sin importarle las fuerzas que lo causan. Dentro de la cinemática se estudia la posición, velocidad, aceleración y todas las derivadas de las variables de posición de mayor orden con respecto al tiempo o cualquier otra variable que ayude a encontrar la posición en el espacio cartesiano y la orientación del robot, es decir, las coordenadas cartesianas  $(x, y)$  y el ángulo  $\theta$ . El estudio de la cinemática de los robot móviles se refiere a todas las propiedades geométricas basadas en el tipo de movimiento.

De manera general los sistemas mecánicos no-holonómicos pueden ser descritos por la ecuación basada en la formulación de Euler-Lagrange [53]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)\tau + J^T(q)\lambda \quad (5.1)$$

de igual forma se define la restricción no-holonómica como

$$J(q)\dot{q} = 0 \quad (5.2)$$

donde  $q$  es un vector  $n$  dimensional de variables de configuración,  $M(q)$  es una matriz simétrica definida positiva de dimensión  $n \times n$ ,  $C(q, \dot{q})$  es la matriz de fuerzas centrípetas y de Coriolis de dimensión  $n \times n$ ,  $G(q)$  es el vector de fuerza gravitacional,  $B(\theta)$  es la matriz de transformación de entrada de dimensión  $n \times r$ ,  $\tau$  es el vector  $r$  dimensional de entradas,  $J(q)$  es el vector de restricción de no holonomía de dimensión  $1 \times n$  y  $\lambda$  el multiplicador de Lagrange de restricción de fuerzas. En la Figura 5.4 se muestra la estructura del robot móvil tipo unicyclo que satisface las ecuaciones (5.1) y (5.2) [9].

Para el caso del robot móvil tipo unicyclo, el vector de variables de configuración se define como  $q = [x \ y \ \theta]^T$  donde  $x$  y  $y$  son las coordenadas del punto  $CR$ ; y  $\theta$  es el ángulo de orientación del robot en el marco inercial.

La ecuación de la dinámica del sistema (5.1) se puede reescribir considerando que  $G(q)$  y  $C(q, \dot{q})$  son cero, esto gracias a la suposición **S4**.

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix} \lambda \quad (5.3)$$

Donde  $\tau_1$  y  $\tau_2$  son los pares de los motores derecho e izquierdo;  $m$  e  $I$  representan la masa y la inercia del robot, respectivamente; mientras que  $R$  es el radio de las ruedas y  $L$  es la distancia entre ellas.

El movimiento de una rueda en el plano  $(X, Y)$  sin deslizamiento (ver Figura 5.5) permite reescribir a la restricción de no holonomía (5.2) como

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (5.4)$$

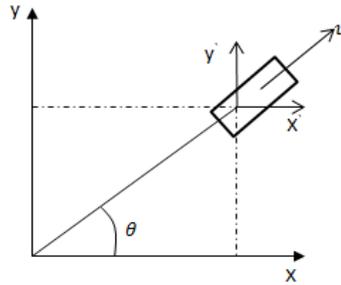


Figura 5.5: Movimiento de una rueda en el plano  $(X, Y)$ .

Esta ecuación (5.4) muestra que la trayectoria del robot es limitada, debido a la condición de no deslizamientos del robot móvil dada en **S3**.

Realizando un cambio de variable en el espacio de los controles como  $F_l = \frac{1}{R}(\tau_1 + \tau_2)$  y  $\tau_a = \frac{L}{R}(\tau_1 - \tau_2)$  lo que implica que la ecuación (5.3) queda dada por

$$\begin{aligned}\ddot{x} &= \frac{F_l}{m} \cos \theta + \frac{\lambda}{m} \sin \theta \\ \ddot{y} &= \frac{F_l}{m} \sin \theta - \frac{\lambda}{m} \cos \theta \\ \ddot{\theta} &= \frac{\tau_a}{I}\end{aligned}\quad (5.5)$$

Donde  $F_l$  y  $\tau_a$  es la fuerza de desplazamiento lineal y el par angular, respectivamente. Para obtener la ecuación (5.5) en su forma normal se utiliza la siguiente transformación [1]

$$\dot{q} = g(q) \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5.6)$$

donde

$$g(q) = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \quad y \quad \dot{g}(q) = \begin{bmatrix} -\dot{\theta} \sin \theta & 0 \\ \dot{\theta} \cos \theta & 0 \\ 0 & 0 \end{bmatrix} \quad (5.7)$$

Al derivar (5.6) con respecto al tiempo se obtiene

$$\ddot{q} = \dot{g}(q) \begin{bmatrix} v \\ \omega \end{bmatrix} + g(q) \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} \quad (5.8)$$

Sustituyendo (5.7) en esta ultima expresión (5.8) se tiene que

$$\begin{aligned}\ddot{q} &= \begin{bmatrix} -\dot{\theta} \sin \theta & 0 \\ \dot{\theta} \cos \theta & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} \quad , \quad \ddot{q} = \begin{bmatrix} -\dot{\theta} \sin \theta v \\ \dot{\theta} \cos \theta v \\ 0 \end{bmatrix} + \begin{bmatrix} \cos \theta \dot{v} \\ \sin \theta \dot{v} \\ \dot{\omega} \end{bmatrix} \\ \ddot{q} &= \begin{bmatrix} -v\dot{\theta} \sin \theta + \dot{v} \cos \theta \\ v\dot{\theta} \cos \theta + \dot{v} \sin \theta \\ \dot{\omega} \end{bmatrix}\end{aligned}\quad (5.9)$$

la ecuación (5.9) puede ser reescrita como

$$\begin{aligned}\ddot{x} &= -v\dot{\theta} \sin \theta + \dot{v} \cos \theta \\ \ddot{y} &= v\dot{\theta} \cos \theta + \dot{v} \sin \theta \\ \ddot{\theta} &= \dot{\omega}\end{aligned}\quad (5.10)$$

Igualando (5.5) y (5.10) se obtiene

$$\begin{aligned}\frac{F_l}{m} \cos \theta + \frac{\lambda}{m} \operatorname{sen} \theta &= -v\dot{\theta} \operatorname{sen} \theta + \dot{v} \cos \theta \\ \frac{F_l}{m} \operatorname{sen} \theta - \frac{\lambda}{m} \cos \theta &= v\dot{\theta} \cos \theta + \dot{v} \operatorname{sen} \theta \\ \frac{\tau_a}{I} &= \dot{\omega}\end{aligned}\quad (5.11)$$

Si se multiplica la primer parte de (5.11) por  $\cos \theta$ , la segunda parte por el  $\operatorname{sen} \theta$  y finalmente se suman los resultados, se obtiene que

$$\dot{v} = \frac{F_l}{m} \quad y \quad \dot{\omega} = \frac{\tau_a}{I} \quad (5.12)$$

donde  $v$  y  $\omega$  son la velocidad lineal y angular del robot móvil.

El modelo cinemático del robot móvil tipo unicycle se deduce a partir de la Figura 5.4,

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \operatorname{sen} \theta \\ \dot{\theta} &= \omega\end{aligned}\quad (5.13)$$

Y en forma matricial se tiene que:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \operatorname{sen} \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}. \quad (5.14)$$

La ecuación del modelo cinemático, ecuación 5.13 describen la posición y orientación del robot móvil respecto a su velocidad lineal y angular respectivamente, pero la dinámica del robot esta dada por 5.12.

Así que el modelo matemático completo (conformado por el modelo cinemático directo el modelo dinámico) resultante es un sistema de quinto orden como el siguiente:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \operatorname{sen} \theta \\ \omega \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{1}{I} \end{bmatrix} \begin{bmatrix} F_l \\ \tau_a \end{bmatrix}. \quad (5.15)$$

En donde:

Variables de estado :  $[x \ y \ \theta \ v \ \omega]^T$

Entradas:  $[F_l \ \tau_a]^T$

$m$ : masa del robot móvil

$I$ : momento de inercia del robot sobre el eje  $z$  en el punto  $(x, y)$

$F_l$ : fuerza aplicada al robot en el punto  $(x, y)$

$\tau_a$ : torque aplicado al robot en el punto  $(x, y)$

Es importante mencionar que se consideran que los tres robots tiene el mismo modelo matemático idealmente hablando. En la siguiente sección se considera el modelo cinemático para cada uno de los robots móviles en el diseño de las leyes de control.

## 5.4. Diseño de las leyes de control

En este apartado se presenta el diseño de una ley de control para el seguimiento de trayectoria mediante el análisis de Lyapunov. El diseño del control se basa en la obtención de la dinámica del error de seguimiento de cada uno de los robots móviles. Para el seguimiento de una trayectoria se considera un robot virtual de referencia (ver Figura 5.6) definiendo a la señal de referencia como el vector de posición  $q_{ri} = [x_{ri} \ y_{ri} \ \theta_{ri}]^T$ , donde  $i$  indica el  $i$ -ésimo robot [54]. Esta trayectoria deberá satisfacer no solo las ecuaciones cinemáticas sino también la restricción de no holonomía:

$$\dot{x}_{ri} = v_{ri} \cos \theta_{ri}, \quad \dot{y}_{ri} = v_{ri} \sin \theta_{ri}, \quad \dot{\theta}_{ri} = \omega_{ri} \quad \dot{x}_{ri} \sin \theta_{ri} = \dot{y}_{ri} \cos \theta_{ri} \quad (5.16)$$

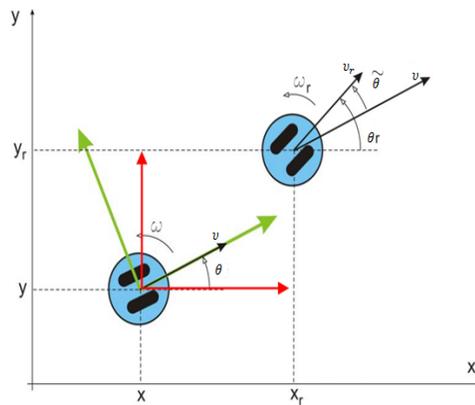


Figura 5.6: Robot virtual de referencia.

La dinámica del error pueden se escritas independientemente del marco inercial fijo, utilizando la siguiente trasformación de coordenadas:

$$\begin{bmatrix} x_{ei} \\ y_{ei} \\ \theta_{ei} \end{bmatrix} = \begin{bmatrix} \cos \theta & \text{sen } \theta & 0 \\ -\text{sen } \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ri} - x \\ y_{ri} - y \\ \theta_{ri} - \theta \end{bmatrix} \quad (5.17)$$

Donde  $q_{ei} = [x_{ei} \ y_{ei} \ \theta_{ei}]^T$  son las variables de error en el sistema de coordenadas del robot móvil.

Al derivar (5.17) y usando (5.4), (5.13) y (5.16) las dinámicas de error son descritas en el nuevo sistemas de coordenadas:

$$\begin{bmatrix} \dot{x}_{ei} \\ \dot{y}_{ei} \\ \dot{\theta}_{ei} \end{bmatrix} = \begin{bmatrix} v_{ri} \cos \theta_{ei} \\ v_{ri} \text{sen } \theta_{ei} \\ \omega_{ri} \end{bmatrix} + \begin{bmatrix} -1 & y_{ei} \\ 0 & -x_{ei} \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (5.18)$$

donde  $[v_i \ \omega_i]^T$  es el vector de control del modelo cinemático. La ley de control se propone con base en(5.18), donde  $(x_{ei}, y_{ei}, \theta_{ei})$  son las variables de estado del sistema.

### 5.4.1. Diseño de una ley de control mediante el análisis de Lyapunov

La función candidata de Lyapunov  $V(x)$  que se propone es:

$$V(\mathbf{x}) = \frac{1}{2}((x_{ei})^2 + (y_{ei})^2) + (1 - \cos \theta_{ei}) \quad (5.19)$$

Las condiciones para que la función candidata de Lyapunov sea una función definida positiva (ver Apéndice B) son:

- a.  $V(x_{ei}, y_{ei}, \theta_{ei}) = V(0, 0, 0) = 0 + 0 + 0 = 0$   
 dado que se busca que

$$\lim_{t \rightarrow \infty} \begin{vmatrix} x_{ei} \\ y_{ei} \\ \theta_{ei} \end{vmatrix} \rightarrow 0$$

- b.  $V(\mathbf{x}) > 0 \ \forall \mathbf{x} \neq 0; V(x_{ei}, y_{ei}, \theta_{ei}) \neq 0 \Rightarrow V(x_{ei}, y_{ei}, \theta_{ei}) > 0$

de lo anterior es fácil ver que la función  $V(\mathbf{x})$  es definida positiva.

Al derivar con respecto al tiempo la función (5.19), resulta:

$$\dot{V}(\mathbf{x}) = x_{ei}\dot{x}_{ei} + y_{ei}\dot{y}_{ei} + \dot{\theta}_{ei} \text{sen } \theta_{ei} \quad (5.20)$$

Sustituyendo (5.18) en (5.20), es decir, a lo largo de las trayectorias del sistema se obtiene:

$$\begin{aligned}\dot{V}(\mathbf{x}) &= x_{ei}(v_{ri} \cos \theta_{ei} - v_i + y_{ei}\omega_i) + y_{ei}(v_{ri} \sin \theta_{ei} - x_{ei}\omega_i) + (\omega_{ri} - \omega_i) \sin \theta_{ei} \\ \dot{V}(\mathbf{x}) &= x_{ei}(v_{ri} \cos \theta_{ei} - v_i) + \sin \theta_{ei}(y_{ei}v_{ri} + \omega_{ri} - \omega_i)\end{aligned}\quad (5.21)$$

De esta última expresión (5.21) se propone a  $v_i$  y  $\omega_i$  de tal forma que  $\dot{V}(\mathbf{x})$  sea definida negativa

$$\begin{aligned}v_i &= v_{ri} \cos \theta_{ei} - k_{1i}x_{ei} \\ \omega_i &= \omega_{ri} + v_{ri}y_{ei} + k_{2i} \sin \theta_{ei}\end{aligned}\quad (5.22)$$

Finalmente, sustituyendo (5.22) en (5.21) se obtiene:

$$\dot{V}(\mathbf{x}) = -k_{1i}x_{ei}^2 - k_{2i} \sin^2 \theta_{ei}\quad (5.23)$$

donde se observa que  $\dot{V}(\mathbf{x})$  es solo semi definida negativa si las constantes  $k_{1i}$  y  $k_{2i}$  dadas por el diseñador son positivas, lo que implica que el sistema es *estable en el sentido de Lyapunov*.

## 5.5. Implementación de la etapa de control

La salida del controlador no lineal de la sección 5.4.1 son la velocidad lineal y angular deseadas para la estabilización cinemática. Estos valores son como las entradas de referencia para el siguiente parte del controlador dinámico.

Dada la ecuación (5.12) y las siguientes transformaciones lineales en el espacio de los controles:

$$\begin{aligned}F_l &= \frac{1}{R}(\tau_1 + \tau_2) \\ \tau_a &= \frac{L}{R}(\tau_1 - \tau_2)\end{aligned}\quad (5.24)$$

se tiene que

$$\begin{aligned}\tau_1 &= \frac{(LF_l + \tau_a)R}{2L} \\ \tau_2 &= \frac{(LF_l - \tau_a)R}{2L}\end{aligned}\quad (5.25)$$

sustituyendo (5.12) en (5.25)

$$\begin{aligned}\tau_1 &= \frac{(L\dot{v}_m + \dot{\omega}I)R}{2L} \\ \tau_2 &= \frac{(L\dot{v}_m - \dot{\omega}I)R}{2L}\end{aligned}\quad (5.26)$$

donde finalmente  $\tau_1$  y  $\tau_2$  son los pares de control (aplicados en los actuadores).

## 5.6. Comentarios

Es importante comentar que para el diseño de la ley de control se emplea la cinemática y dinámica del robot móvil, además que se cuenta con una cámara de visión como sensor en el espacio de trabajo y aplicando la metodología presentada en el Capítulo 4 se obtienen las variables de posición  $x, y$  y el ángulo de orientación  $\theta$  de cada uno de los robots. La ley de control resultante está descrita por la ecuación (5.25), donde se proponen las ganancias  $k_{1i}$  y  $k_{2i}$  para cada uno de los robots y empleada en la etapa experimental cuyos resultados son mostrados en el siguiente capítulo.



# Capítulo 6

## Resultados experimentales

Este capítulo presenta los resultados experimentales de las leyes de control para el seguimiento de trayectoria presentadas en el Capítulo 5. La plataforma de experimentación consiste en tres robots móviles del tipo unicycle.

La organización de este capítulo es la siguiente: En la primera Sección 6.1 de este capítulo se da una breve introducción resaltando la importancia de la metodología propuesta para estimar la ubicación de los robots móviles, en la Sección 6.2 se describe la plataforma experimental y se dan las condiciones bajo las cuales son desarrollados los experimentos, los cuales son mostrados en la Sección 6.3. Finalmente, los comentarios de este capítulo se presentan en la Sección 6.4.

### 6.1. Introducción

En este capítulo se pone a prueba el ingenio debido a la cantidad de dificultades técnicas que se presentaron en el momento de la implantación de las teorías utilizadas en el diseño de las leyes de control, las cuales están basadas en el modelo matemático de los robots móviles, es por ello que se requiere de habilidades de ingeniería para obtener resultados satisfactorios que garanticen el buen desempeño en el comportamiento del sistema en lazo cerrado empleando un esquema de *visual servoing*. Una de las dificultades técnicas que se presentan es en la estimación de los parámetros de ubicación de los robots móviles, lo cual depende del entrenamiento del clasificador, de no ser así los datos arrojados tienen errores y por consecuencia la ley de control que hace uso de ellos es calculada de manera errónea.

## 6.2. Plataforma experimental

La arquitectura del sistema en general, está compuesta de tres robots móviles tipo unicycle, un sistema de visión, un módulo de comunicación y una computadora (Figura 6.1), todos ellos ubicados en un espacio de trabajo de área  $1.5 m^2$  completamente iluminado. Cada robot móvil tiene una dimensión de  $7 cm$  de alto,  $9.5 cm$  de largo y  $7.5 cm$  de ancho con un peso de  $250g$ . Dentro del cuerpo del robot está incluida su fuente de alimentación que consiste de una batería recargable Nickel-Metal, cuadrada de ( $9V$  a  $300 mA$ ), dos motorreductores, un microcontrolador y un módulo de comunicación XBee del protocolo de comunicaciones inalámbrico.



Figura 6.1: Fotografía de la plataforma experimental.

Como se puede observar en la Figura 6.2, en la parte superior de cada robot está colocado un motivo con dos marcas circulares de diferentes colores con un diámetro de  $5 cm$  y  $10 cm$  respectivamente. Estas marcas permiten obtener las posiciones de  $x_i, y_i$  y la orientación  $\theta_i$  de cada robot. El sistema de visión está compuesto por una cámara digital de visión marca *Philips* utilizando una resolución de  $640 \times 480$  pixeles y trabajando con un muestreo de 32 frames por segundo, conectada a un video procesador marca *Intel*, un equipo de comunicación y una computadora. La cámara de visión está montada a  $150 cm$  por encima del espacio de trabajo. La webcam captura las imágenes que posteriormente son usadas por el algoritmo desarrollado que estima la ubicación de los robots móviles.

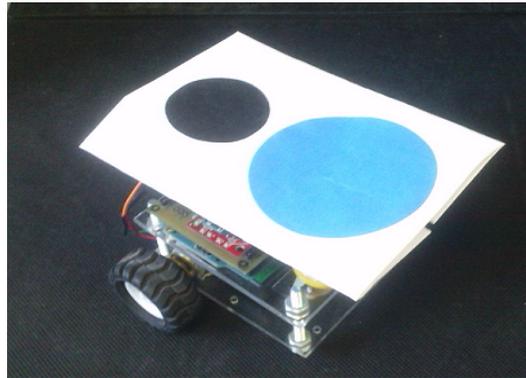


Figura 6.2: Motivo colocado sobre un robot móvil.

### 6.3. Experimentos realizados

En esta sección se presentan los resultados experimentales obtenidos en lazo cerrado empleando un esquema de *visual servoing*, es importante mencionar que los motores empleados como actuadores en los robots móviles comienzan a girar a partir de un voltaje inicial  $v_0 = 3.3V$ , esto debido al sistema de tren de engranes, a la inercia del robot y a la fricción que existe entre la superficie de trabajo y las ruedas empleadas, por lo tanto este voltaje es sumado a la ley de control calculada siempre y cuando el error sea diferente de cero.

Las ganancias  $k_{1i}$  y  $k_{2i}$  propuestas en el controlador son 2.0 y 4.0 respectivamente para todas las pruebas. Es importante resaltar que la ganancia de mayor peso es  $k_{2i}$ , esto se debe a que esta ganancia pondera a la orientación dentro de la ley de control  $\omega_i$ , la interpretación física que se da a esto es que la orientación tiene que responder más rápido que la velocidad lineal. A continuación se describen tres tipos de experimentos que se realizaron considerando diferentes condiciones iniciales, tanto en posición, orientación y iluminación en el espacio de trabajo:

1. En este primer experimento solo se da referencia sobre la coordenada  $x$ , es decir, el robot se mueve únicamente en una línea.
2. En el segundo experimento el robot parte de una condición inicial y alcanza un punto de referencia arbitrario en el espacio de trabajo.
3. En este último experimento se repite el primero, para posteriormente cambiar de referencia a la coordenada en  $y$ , es decir, el robot realiza una trayectoria en forma de  $L$ .

### 6.3.1. Primer experimento: Desplazamiento en el eje $x$ .

Este primer experimento es de vital importancia, debido a que se ponen a prueba todos los algoritmos desarrollados para la estimación de ubicación del robot junto con la ley de control diseñada. Un robot móvil es colocado en alguna posición dentro del espacio de trabajo y se le da una referencia deseada en el eje  $x$ , cuyos resultados obtenidos se muestran en la figura 6.3 de donde se puede observar que el robot inicia en una posición en  $x = 29$  alcanzando la referencia en 15.0 segundos.

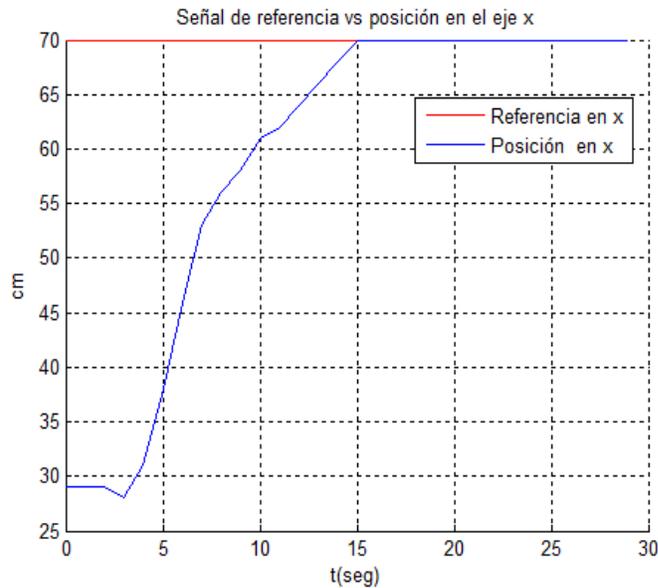


Figura 6.3: Señal de referencia vs señal de posición en el eje  $x$ .

En la gráfica mostrada en la Figura 6.4 se puede observar que la señal de error es acotada y que converge a cero cuando  $t \rightarrow \infty$ .

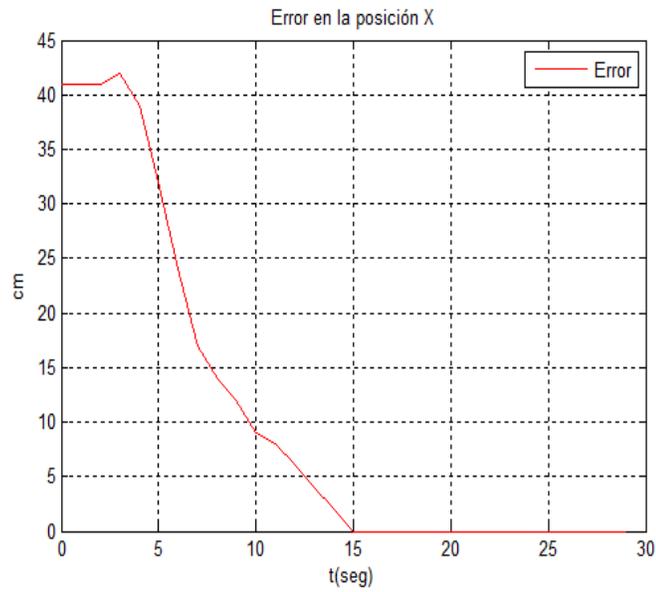


Figura 6.4: Señal de error en la posición  $x$

En las Figuras 6.5 y 6.6 se muestran la posición y orientación del robot móvil respectivamente.

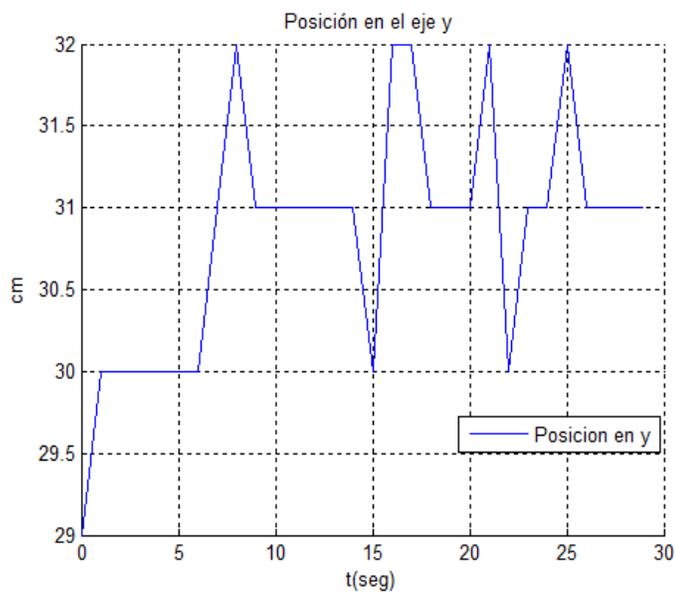


Figura 6.5: Señal de posición en el eje  $y$ .

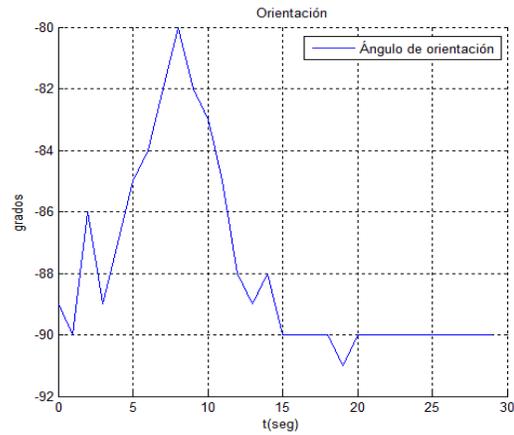


Figura 6.6: Señal de posición angular de orientación  $\theta$ .

En la Figura 6.7 se muestran las señales de control enviadas a los motores del robot móvil, como se puede observar son señales cuya amplitud corresponde al valor entregado a los PWM (sin normalizar), donde se tiene como relación que 3.3V equivale a un valor de 80 en el PWM.

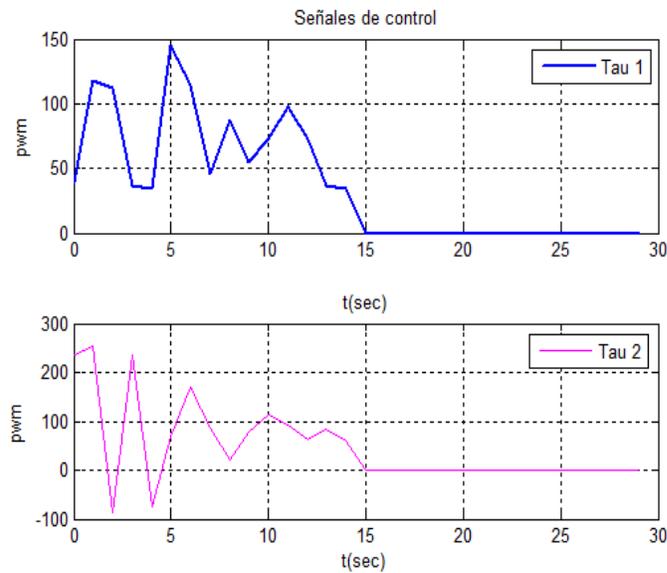


Figura 6.7: Señales de control  $v$  y  $\omega$ .

### 6.3.2. Segundo experimento: Desplazamiento en el plano $x, y$ .

En este segundo experimento se considera una condición inicial arbitraria dentro del espacio de trabajo del robot y se le envía una posición  $(x, y)$  en el plano, cuyos resultados son mostrados en las Figuras 6.8 y 6.9 de donde se puede observar que el robot no alcanza la posición deseada, una de las razones de este desempeño se atribuye principalmente a la alta dependencia de las condiciones de iluminación en el espacio de trabajo, así como a las ganancias  $k_1$  y  $k_2$  propuestas, ya que al variarlas el desempeño del sistema en lazo cerrado sufre cambios significativos.

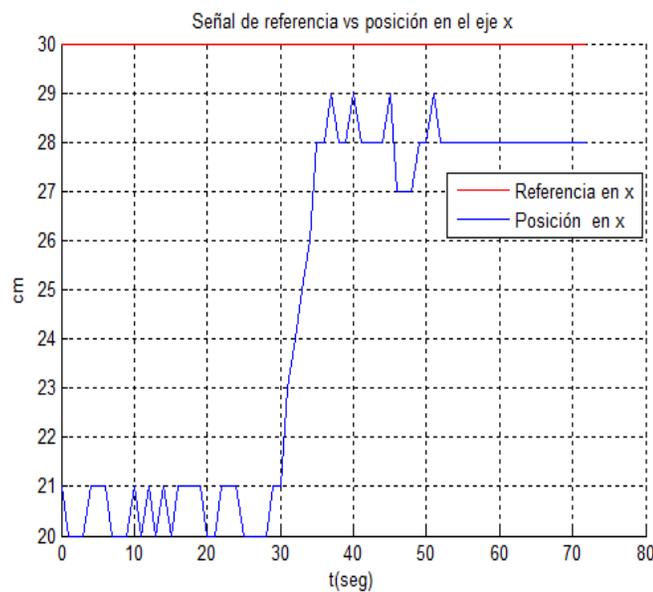


Figura 6.8: Señal de referencia vs señal de posición en el eje  $x$ .

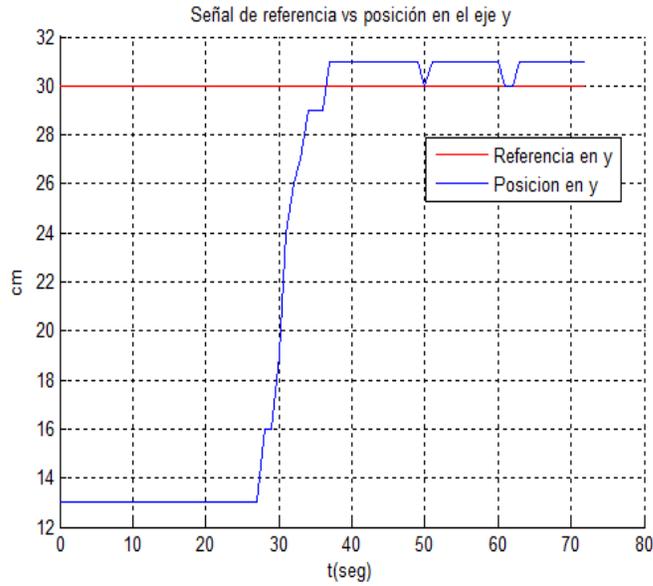


Figura 6.9: Señal de referencia vs señal de posición en el eje  $y$ .

En las Figura 6.10 se muestra la orientación del robot móvil.

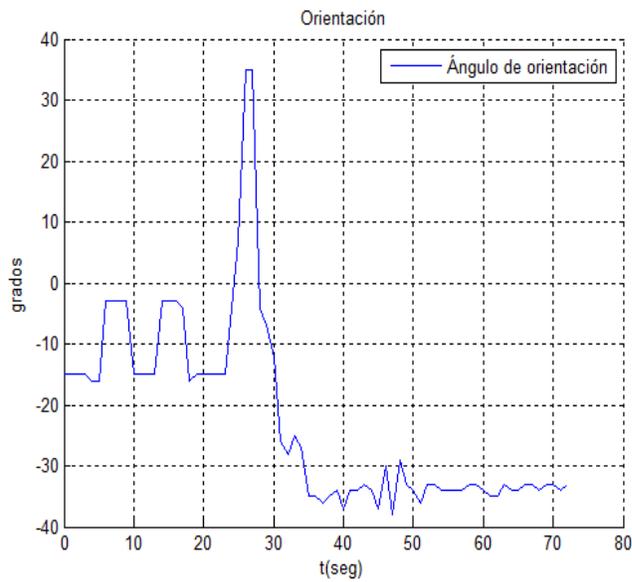


Figura 6.10: Señal de posición angular de orientación  $\theta$ .

Las señales de los errores son mostrados en la Figura 6.11, mientras que las señales

de control se dan en la Figura 6.12.

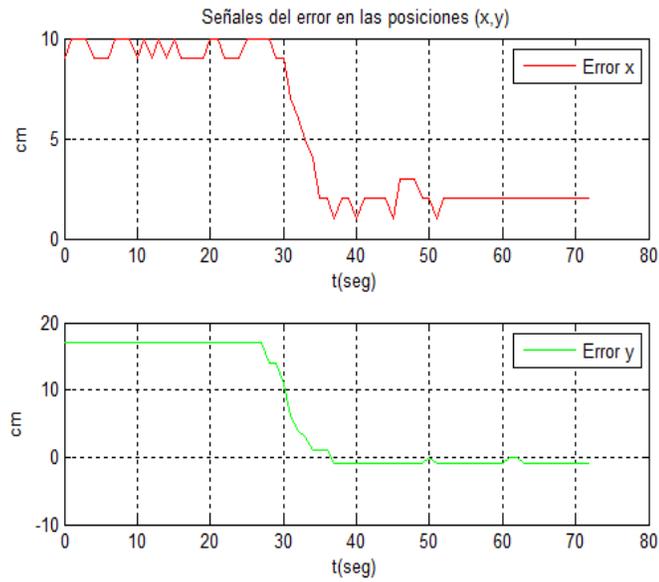


Figura 6.11: Señales del error en las posiciones  $x, y$ .

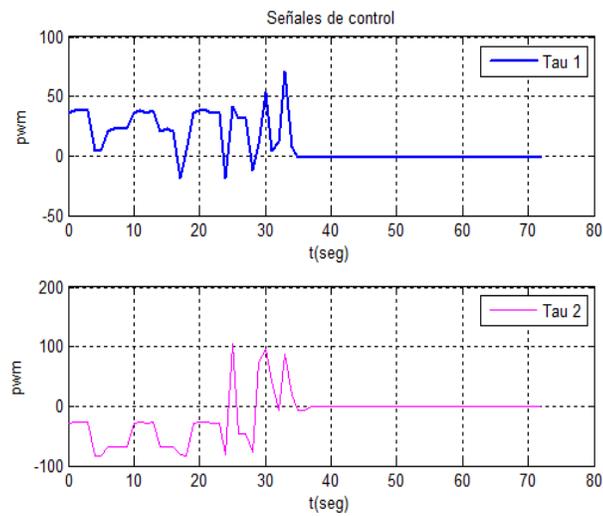


Figura 6.12: Señales de control  $v$  y  $\omega$ .

### 6.3.3. Tercer experimento: Desplazamiento en el eje $x$ , desplazamiento en el eje $y$ , ante perturbaciones de iluminación.

De este último experimento los resultados arrojados ante perturbaciones en la iluminación muestran que el control no presenta un buen desempeño a pesar de variar las ganancias  $k_1$  y  $k_2$ .

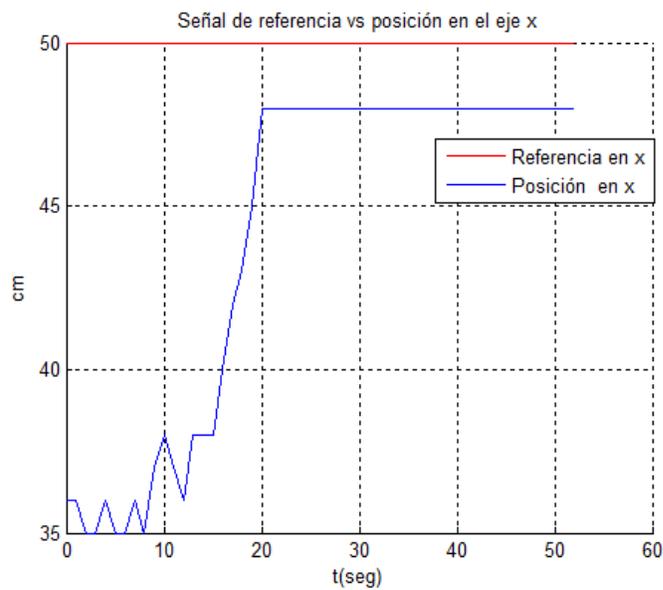


Figura 6.13: Señal de referencia vs señal de posición en el eje  $x$ .

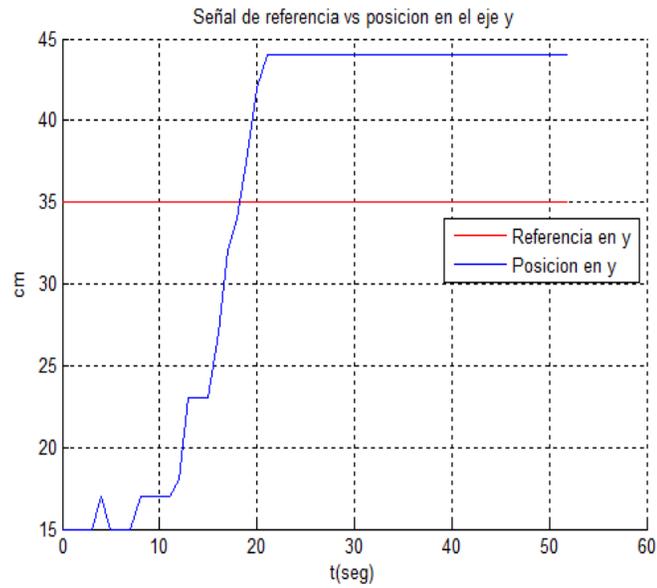


Figura 6.14: Señal de referencia vs señal de posición en el eje  $y$ .

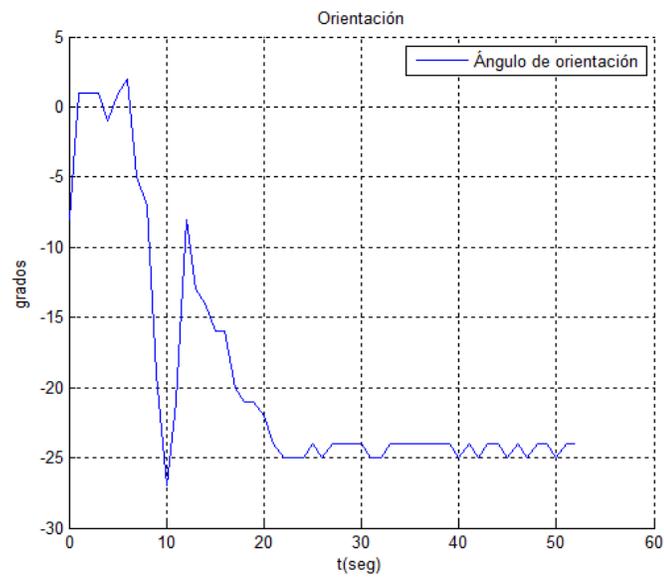


Figura 6.15: Señal de posición angular de orientación  $\theta$ .

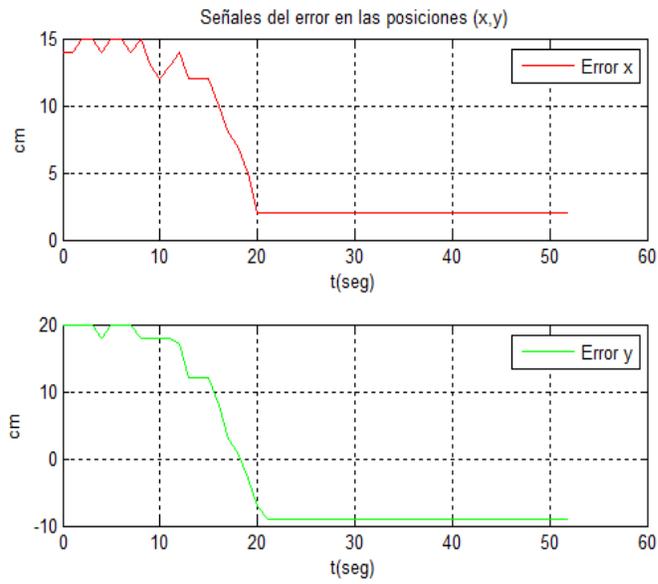


Figura 6.16: Señales del error en las posiciones  $x, y$ .

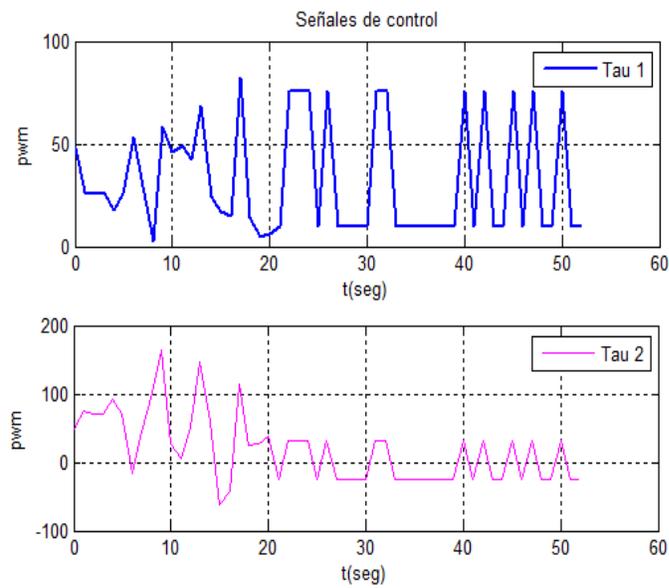


Figura 6.17: Señales de control  $v$  y  $\omega$ .

## **6.4. Comentarios**

De los resultados experimentales obtenidos se puede comentar que la plataforma conformada por tres robots móviles dependen fuertemente de las condiciones de iluminación, así como de una adecuada selección de las ganancias  $k_1$  y  $k_2$ .



# Capítulo 7

## Conclusiones y trabajos futuros

### Conclusiones

De este trabajo de tesis se enuncian los siguientes puntos:

- Para tener la plataforma funcional se integran diferentes áreas del conocimiento como son: mecánica, electrónica, programación, procesamiento de señales e imágenes, control y robótica. Lo que hace del desarrollo de esta plataforma un trabajo multidisciplinario.
- Otra integración que se realizó está presente en el manejo de diferentes plataformas como son: Dev-C++, LabVIEW, Matlab y Arduino en el desarrollo de los algoritmos de procesamiento de imágenes para obtener las posiciones y orientaciones de cada uno de los robots móviles empleadas en las leyes de control.
- Gracias a todas estas integraciones realizadas ha sido posible tener una plataforma experimental para propósitos de investigación que cuenta con un sistema de tres robots móviles terrestres dotados de autonomía que permite que tengan movilidad y desplazamientos libres en su espacio de trabajo.
- El uso de la cámara de visión como sensor permitió emplear un esquema *visual servoing* en el control de los tres robots móviles terrestres, las leyes de control fueron diseñadas empleando el enfoque de Lyapunov, que hace uso del modelo cinemático de los robots, garantizando con esto estabilidad y convergencia en el sentido de Lyapunov.

**La contribución de este trabajo permite concluir que se cumplió satisfactoriamente con el objetivo planteado.**

## Trabajos futuros

A continuación se mencionan algunos trabajos futuros:

- Gracias a que se cuenta con una plataforma experimental funcional de arquitectura abierta es posible proponer diferentes estrategias de control para propósitos de coordinación y cooperación.
- Contemplar condiciones de no colisión entre los robots y objetos dentro del espacio de trabajo.
- Pruebas de seguimiento de trayectorias a nivel experimental.

# Glosario

**Aceleración.** Magnitud que expresa el incremento de la velocidad en la unidad de tiempo. Su unidad en el Sistema Internacional es el metro por segundo cada segundo ( $m/s^2$ ).

**Algoritmo.** Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

**Angulo de rotación.** También conocido como orientación, es el ángulo formado entre el eje de las abscisas y el eje alrededor del cual un cuerpo puede rotar con mínima inercia.

**Autónomo.** Que trabaja por cuenta propia.

**Centro de gravedad.** Punto de aplicación de la resultante de todas las fuerzas de gravedad que actúan sobre las distintas masas materiales de un cuerpo.

**Cinématica.** Parte de la física que estudia el movimiento prescindiendo de las fuerzas que lo producen.

**Comunicación inalámbrica.** Unión, conexión o enlace que no se basa en el contacto físico a través de cables utilizando ondas radioeléctricas mediante emisor y receptor incorporados.

**Controlabilidad.** Se dice que un sistema es controlable en el instante  $t_0$  si es posible llevarlo de cualquier estado inicial  $x(t_0)$  a cualquier otro estado, empleando un vector de control no acotado, en un lapso finito.

**Dinámica.** Parte de la mecánica que trata de las leyes del movimiento en relación con las fuerzas que lo producen.

**Efecto Coriolis.** Es la diferencia entre la aceleración medida del campo de referencia y la aceleración relativa medida con el marco del efector final.

**Eje de referencia inercial fijo.** Es el eje de referencia cuyas coordenadas se encuentran fijas respecto al robot.

**Eje de referencia inercial móvil.** Es el eje de referencia ubicado en una parte del robot móvil.

**Emisor.** Equipo electrónico que codifica el mensaje y lo transmite por medio de un canal o medio hasta un receptor.

**Energía.** Capacidad de realizar un trabajo.

**Estabilidad.** En ciencias, una situación es estable si se mantiene en estado estacionario, es decir, igual en el tiempo y una modificación razonablemente pequeña de las condiciones iniciales no altera significativamente el futuro de la situación.

**Fricción.** Roce de dos cuerpos en contacto.

**Grados de libertad.** Número de coordenadas que describen de manera única la posición de todos los eslabones de un sistema.

**Holónomo.** Un sistema holónomo es aquel en el que sus grados de libertad están desacoplados facilitando su desplazamiento.

**Inercia.** Tendencia de un cuerpo a permanecer en reposo en movimiento lineal uniforme.

**Lagrangiano.** Las ecuaciones de movimiento pueden ser derivadas de manera sistemática independientemente del marco de coordenadas de referencia. El lagrangiano de un sistema mecánico puede ser definido como una función generalizada de coordenadas: la diferencia de energías cinéticas y potenciales.

**Locomoción.** Traslación de un lugar a otro.

**Mecatrónica.** Surge de la combinación sinérgica de distintas ramas de la ingeniería, entre las que destacan: la mecánica de precisión, la electrónica, la informática y los sistemas de control. Su principal propósito es el análisis y diseño de productos y de procesos de manufactura automatizados.

**Microcontrolador.** Circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: CPU, memoria y unidades de entrada/salida.

**Modelo cinemático.** Es aquel que relaciona las coordenadas espaciales con las articulares.

**Modelo dinámico.** Modelo que describe el comportamiento de un sistema dinámico.

**Modelo matemático.** Es la representación por medio de ecuaciones de la dinámica de un sistema. Es el tipo de modelo más importantes para la ciencia y la tecnología.

**Momento de inercia de masa.** Medida de la resistencia del cuerpo a moverse si está parado y a la inversa. Es una resultante de la relación entre la masa del cuerpo y la distancia al centro de giro. La alteración de la resistencia afecta a su vez a la velocidad angular del cuerpo.

**No-holónimo.** Un sistema no-holónimo es aquel que presenta sus grados de libertad acoplados, es decir, que se necesitan una serie de maniobras para alcanzar una posición.

**Prototipo.** Ejemplar original o primer molde en que se fabrica una figura u otra cosa.

**Radio.** Segmento lineal que une el centro del círculo con la circunferencia.

**Receptor.** Equipo electrónico que recibe una señal, código o mensaje emitido por un emisor.

**Robot.** Dispositivo generalmente mecánico, que desempeña tareas automáticamente, ya sea de acuerdo a supervisión humana directa, a través de un programa predefinido o siguiendo un conjunto de reglas generales. Generalmente estas tareas reemplazan, asemejan o extienden el trabajo humano, como ensamblaje en manufactura, manipulación de objetos pesados o peligrosos, trabajo en el espacio, etc.

**Ruido.** Típicamente es una perturbación aleatoria. Señal indeseada presente en un sistema físico.

**Sensor.** Dispositivo que convierte un parámetro físico (como temperatura, presión, flujo, velocidad, posición) en una señal eléctrica. En algunos casos se le considera un sinónimo de transductor, pero un verdadero sensor contiene un sistema de acondicionamiento de la señal, de manera que es mucho más sencillo realizar una medición.

**Señal de referencia.** Es el valor que se desea que alcance la señal de salida.

**Sistema de referencia.** Un sistema de referencia o marco de referencia es un conjunto de convenciones usadas por un observador para poder medir la posición y otras magnitudes físicas de un objeto o sistema físico en el tiempo y el espacio.

**Sistema en tiempo real.** Un STR, es un sistema informático en el que es significativo el tiempo en el que se producen las acciones. Las acciones deben realizarse dentro de un intervalo de tiempo determinado.

**Sistema.** Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado fin.

**Software.** Se dice de todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

**Velocidad angular.** También conocida como frecuencia angular o pulsación, es una medida de la velocidad de rotación. Se mide en radianes por segundo.

**Velocidad lineal.** Es lo que se tarda en recorrer un espacio en línea recta.

**Velocidad.** Magnitud física que expresa el espacio recorrido por un móvil en la unidad de tiempo. Su unidad en el Sistema Internacional es el metro por segundo (m/s).

**Visión artificial.** Conjunto de técnicas de adquisición y tratamiento computacional de imágenes que permiten obtener información del entorno a estudiar mediante métodos visuales.

# Apéndice A

## Plataformas: Dev-C++, LabVIEW, Matlab y Arduino

Las plataformas utilizadas para este trabajo de investigación son: Windows 7, Dev-C++, LabVIEW 9.0, Matlab 7.8 y Arduino (Figura A.1).



Figura A.1: Plataformas computacionales empleadas.

En este apéndice se contiene la distribución de los archivos, imágenes, programas usados en la etapa de experimentos, los cuales están direccionados en el disco compacto adjunto a este trabajo con la distribución mostrada en la Figura A.2.

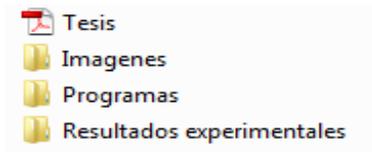


Figura A.2: Directorio.

También se adjunta el archivos en PDF del presente trabajo de tesis.

### **Carpeta: Imágenes**

Dentro de esta carpeta se encuentran las imágenes y fotografías ocupadas en los Capítulos 3, 5, 4 y 6 separadas en sus carpetas correspondientes (Figura A.3).

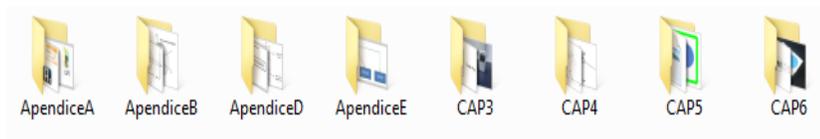


Figura A.3: Contenido de la carpeta: Imágenes.

### **Carpeta: Programas utilizados en la etapa de experimentación**

Dentro de esta carpeta (Figura A.4) se encuentran los programas desarrollados en: Dev-C++, LabVIEW, Matlab y Arduino, estos programas son integrados en la parte experimental.

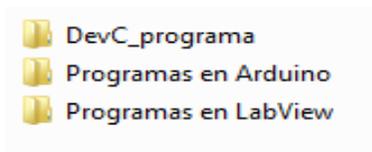


Figura A.4: Contenido de la carpeta: Programas.

### **Carpeta: Resultados experimentales de la implantación de las leyes de control**

En este apartado se muestran las carpetas correspondientes a cada experimento realizado (Figura A.5), conteniendo los programas de Visula C++ y Matlab.

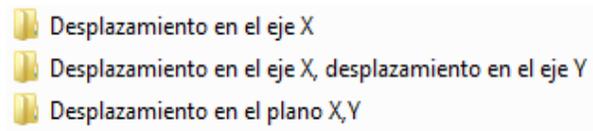


Figura A.5: Contenido de la carpeta: Resultados experimentales.



# Apéndice B

## Conceptos de estabilidad en el sentido de Lyapunov

### Introducción

La estabilidad (asociada a conceptos relacionados con la respuesta dinámica de un sistema) es una propiedad fundamental de los sistemas de control, tan importante como la robustez, el rendimiento y otras características que lo identifican. El análisis de estabilidad de los sistemas no lineales es un problema complejo donde no existe una técnica común que pueda resolver todos los casos que se presentan. La estabilidad de puntos de equilibrio es generalmente llamada estabilidad en el sentido de Lyapunov. Un punto de equilibrio se dice estable si ante pequeñas perturbaciones en las condiciones iniciales, el sistema presenta pequeñas perturbaciones en las trayectorias. Un punto de equilibrio se dice asintóticamente estable cuando no importa en qué punto se inicie dentro de una bola, siempre llega al punto de equilibrio. Un punto de equilibrio se dice exponencialmente estable cuando se impone un tiempo límite para llegar al origen. Los teoremas de estabilidad de Lyapunov dan condiciones suficientes mas no necesarias para la estabilidad de puntos de equilibrio.

### Teoría de Lyapunov

La teoría de Lyapunov estudia la estabilidad de los puntos de equilibrio, en un punto de vista teórico estudia la convergencia (al punto de equilibrio) de la función  $V(x)$ . Esta teoría es la más generalizada de las herramientas para el estudio de la estabilidad de sistemas. Está basada en el trabajo “El problema general de la estabilidad del movimiento” del matemático ruso Alexander Mikhailovich Lyapunov publicado en 1892. La potencia de este método es su generalización, puede aplicarse a sistemas tan-

to variantes en tiempo (sistemas no autónomos) como invariantes en tiempo (sistemas autónomos), independiente del orden del sistema.

Existen dos métodos fundamentales de análisis:

- **PRIMER MÉTODO** (Método indirecto): Utiliza la linealización de un sistema para determinar la estabilidad local en un punto de equilibrio de un sistema no lineal de la forma

$$\dot{x} = f(x) \tag{B.1}$$

en otras palabras, deduce la estabilidad de un sistema no lineal a partir del sistema lineal.

- **SEGUNDO MÉTODO** (Método directo): Este es el método más general para determinar la estabilidad de sistemas no lineales y/o variantes con el tiempo. Contrario al método indirecto de Lyapunov, este método no requiere de la solución explícita de las ecuaciones diferenciales para el análisis; es decir, se determina la estabilidad del sistema sin resolver las ecuaciones de estado. Esto ofrece una gran ventaja porque, por lo general, es muy difícil resolver las ecuaciones de estado no lineales y/o variantes con el tiempo.

Dicha teoría asume tres aspectos importantes:

- Se conoce el modelo del sistema (no se conoce la solución de la ecuación diferencial).
- El modelo se tiene en variables de estado.
- El punto de equilibrio siempre es el origen (si esto no ocurre, entonces se realiza una transformación para satisfacer esta condición).

## Puntos de equilibrio

Es posible que para la trayectoria de un sistema le corresponda un solo punto, llamado un punto de equilibrio. Muchos problemas de estabilidad son formulados naturalmente con respecto a los puntos de equilibrio.

**Definición 1** *Dado un sistema autónomo no lineal ( $f(x)$  no depende del tiempo) descrito por la ecuación diferencial (B.1)*

se dice que  $x^*$  es un punto de equilibrio de (B.1) si el sistema inicia en este punto y permanece en él para siempre [58].

Matemáticamente, esto significa que el vector  $x^*$  satisface

$$0 = f(x^*)$$

## Estabilidad

**Definición 2** *El estado de equilibrio  $x = 0$  se llama estable en el sentido de Lyapunov, o simplemente estable, si para algún  $R > 0$ , existe un  $r > 0$  tal que si  $\|x(0)\| < r$ , entonces  $\|x(t)\| < R$ , para todo  $t \geq 0$ . En otro caso se dice, que el punto de equilibrio es inestable [58].*

**\*De manera implícita la definición asume que la solución de la ecuación diferencial se conoce o existe.**

Esencialmente estabilidad (también llamada estabilidad en el sentido de Lyapunov o Lyapunov estable) significa que la trayectoria del sistema comienzan lo suficientemente cerca al origen, esta evoluciona de manera arbitraria permaneciendo cerca de este punto, en otras palabras, estabilidad significa que ante pequeñas perturbaciones en las condiciones iniciales, el sistema tendrá pequeñas perturbaciones en la trayectoria.

## Estabilidad asintótica y estabilidad exponencial

**Definición 3** *Un punto de equilibrio  $x = 0$  es asintóticamente estable si el punto es estable y además existe un  $r > 0$  tal que  $\|x(0)\| < r$  implica que  $x(t) \rightarrow 0$ , cuando  $t \rightarrow \infty$  [58]*

Es decir, primero tiene que evolucionar y ser estable. Pero además, cuando  $t \rightarrow \infty$  llegue al origen.

**\*No importa en qué punto se inicie en la bola, siempre llega al punto de equilibrio (origen) sin tomar en cuenta cuanto tiempo tarde.**

**Definición 4** *Un punto de equilibrio  $x = 0$  es **exponencialmente estable** si existen dos números estrictamente positivos  $\alpha$  y  $\lambda$  tal que*

$$\forall t > 0, \|x(t)\| \leq \alpha \|x(0)\| e^{-\lambda t} \quad (\text{B.2})$$

*en alguna bola  $Br$  alrededor del origen [58].*

*Lo que significa que el vector de estado de un sistema exponencialmente estable converge al origen más rápido que una función exponencial.*

## Estabilidad global

Lo anterior ha sido formulado para caracterizar el comportamiento local de sistemas no lineales, es decir, como el estado evoluciona después de iniciar cerca del punto de equilibrio.

El concepto de global se introduce a continuación:

**Definición 5** *Si cualquier estado inicial se mantiene asintóticamente (o exponencialmente) estable, el punto de equilibrio se dice estable asintóticamente (o exponencialmente) sin límites. Esto es llamado estabilidad asintótica (o exponencial).*

*Si existe estabilidad asintótica o exponencial para cualquier estado inicial, de forma general (sin límites) también **globalmente** [58].*

*En otras palabras, si en las definiciones dadas de manera local el estado inicial  $x(0)$  no está restringido a que dicho estado esté lo suficientemente cerca del origen (o cumple cualquiera, sea del estado inicial  $x(0)$  de que se parta), entonces se dice que la estabilidad es global.*

**\*Una condición para que la estabilidad sea global, es que el punto de equilibrio debe ser único.**

## Función definida positiva

**Definición 6** *Una función escalar continua  $V(x)$  se llama “definida positiva localmente” si [58]:*

- a.  $V(0) = 0$
- b. Si en una bola  $B_{R_0}$   $x \neq 0 \Rightarrow V(x) > 0$

**Definición 7** *Una función escalar continua  $V(x)$  se llama “definida positiva globalmente” si [58]:*

- a.  $V(0) = 0$
- b.  $V(x) > 0 \forall x \neq 0; x \in \mathbb{R}^n$

## Función de Lyapunov

Las funciones de Lyapunov, son funciones que demuestran la estabilidad de cierto punto fijo en un sistema dinámico o en las ecuaciones diferenciales autónomas. Las funciones que podrían probar la estabilidad de un punto cualquiera de equilibrio son llamadas funciones candidatas de Lyapunov.

No existe un método general para construir o encontrar una función candidata de Lyapunov que demuestre estabilidad, en todo caso, la incapacidad de encontrar una función de Lyapunov no implica automáticamente inestabilidad. Para los sistemas dinámicos (como los sistemas físicos) las leyes de conservación de energía son buena opción como funciones candidatas de Lyapunov.

**Definición 8** Si una bola  $B_{R_0}$ ; la función  $V(x)$  es definida positiva y tiene derivadas parciales continuas, y su derivada con respecto al tiempo a lo largo de las trayectorias del estado del sistema es semi-definida negativa, es decir,

$$\dot{V}(x) \leq 0 \quad (\text{B.3})$$

entonces  $V(x)$  se llama una función de Lyapunov [58].

## Teorema de Lyapunov para estabilidad local

**Teorema 1 (*Estabilidad local*)** Si una bola  $B_{R_0}$ , existe una función escalar  $V(x)$  con primeras derivadas parciales continuas, tal que:

- a.  $V(x)$  es definida positiva (localmente en  $B_{R_0}$ )
- b.  $\dot{V}(x)$  es semi-definida negativa (localmente en  $B_{R_0}$ )

**entonces** el punto de equilibrio  $x = 0$  es estable. Si la derivada  $\dot{V}(x)$  es definida negativa localmente en  $B_{R_0}$ ; **entonces** la estabilidad es asintótica [58].

## Teorema de Lyapunov para estabilidad global

Para afirmar la estabilidad asintótica global de un sistema, naturalmente uno puede esperar que la bola  $B_{R_0}$  que se menciona en la definición local tiene que ser expandida para tener el espacio-estado completo, lo que es realmente necesario pero no suficiente. Una condición adicional sobre la función  $V(x)$  es que tiene que satisfacer:  $V(x)$  debe ser radialmente no acotada, lo que significa que  $V(x) \rightarrow \infty$  como  $\|x\| \rightarrow \infty$  (en otras palabras, como  $x$  tiende a infinito en cualquier dirección). De lo cual se obtiene como resultado lo siguiente:

**Teorema 2 (*Estabilidad global*)** Se asume que existe una función escalar  $V$  del estado  $x$  con primeras derivadas continuas, tal que:

- a.  $V(x)$  es definida positiva
- b.  $\dot{V}(x)$  es semi-definida negativa

c.  $V(x) \rightarrow \infty$  como  $\|x\| \rightarrow \infty$

*entonces* es el punto de equilibrio  $x = 0$  es asintóticamente estable [58].

#### **Ventajas de utilizar Lyapunov**

- Encontramos la estabilidad de un punto del sistema, sin resolver la ecuación diferencial.
- Además de lograr estabilidad también nos permite diseñar controles.

#### **Desventajas de utilizar Lyapunov**

- No existe una metodología para proponer una función candidata de Lyapunov.
- Se tiene que conocer el modelo.

# Apéndice C

## Herramientas de visión artificial

### Introducción

Crear ilusiones ópticas hoy en día es un problema bastante complejo, y sobre todo extenso porque desde el punto de vista de la ingeniería, abarca diferentes áreas. Es necesario tener conocimientos de fotografía para saber cómo una cámara (el “ojo” industrial que tenemos a nuestra disposición) capta imágenes del mundo real de manera que las podemos almacenar, imprimir, o mostrar en una pantalla de una computadora o televisión. También es necesario saber cómo pueden manejarse estas fotografías o videos de manera que sean útiles una computadora. Al fin y al cabo se está manejando información; una información bastante útil ya que establece una relación directa entre lo que vemos y lo que hay en el mundo real. Y esta relación es matemática. El mundo de la imagen es muy extenso, y en los últimos años han aparecido dos áreas especializadas dedicadas a resolver el problema de la integración de objetos sintéticos en imágenes reales: la realidad aumentada y la visión artificial.

### Conceptos generales

Debido a que este proyecto se enmarca en el campo de la visión computacional, es necesario introducir algunos de los elementos básicos de la representación de imágenes [26].

### Píxel

Es la abreviatura de las palabras inglésas “picture element” (Figura C). Es el menor de los elementos de una imagen al que se puede aplicar individualmente un color o una intensidad o que se puede diferenciar de los otros mediante un determinado procedimiento.

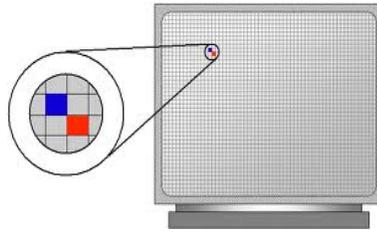


Figura C.1: Representación de un píxel.

## Imagen digital

Una imagen digital se compone de una agrupación de píxeles, cada uno con un valor de intensidad o brillo asociado. Una imagen digital se representa mediante una matriz bidimensional, de forma que cada elemento de la matriz se corresponde con cada píxel en la imagen. Ver Figura C.

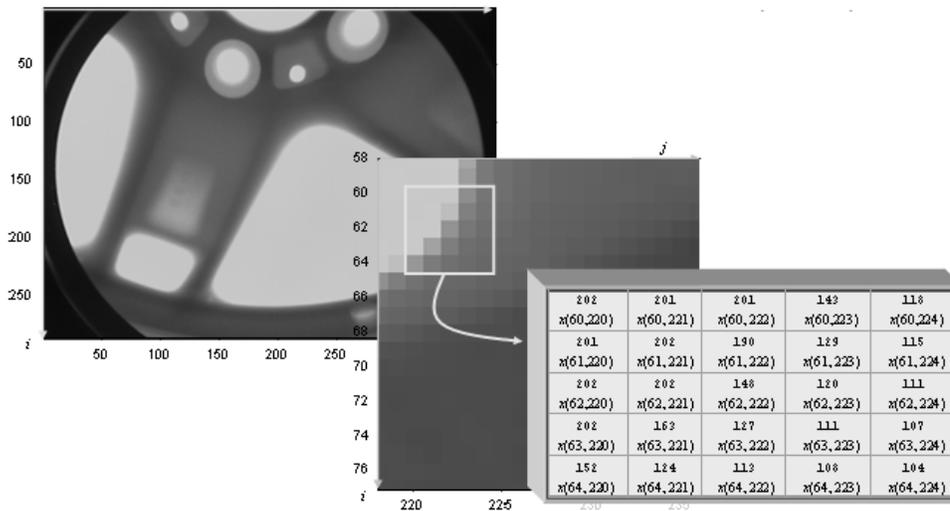


Figura C.2: Organización matricial uniforme de una imagen digital.

## Espacio de color: Modelo RGB

Es uno de los modelos más utilizados por los sistemas informáticos para reproducir los colores en el monitor y en el escáner [46]. Está basado en la síntesis aditiva de las intensidades de luz relativas al rojo, al verde y al azul para conseguir los distintos colores; incluyendo el negro y el blanco.

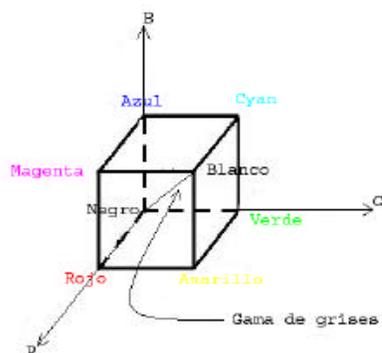


Figura C.3: Representación gráfica del modelo de color RGB.

La representación gráfica del modelo RGB (ver figura C) se realiza mediante un cubo unitario con los ejes R, G y B. El origen (0,0,0) representa el negro y las coordenadas (1,1,1) el blanco. Los vértices del cubo en cada eje R, G y B, de coordenadas (1,0,0), (0,1,0) y (0,0,1) representan los colores primarios rojo, verde y azul. Los restantes tres vértices (1,0,1), (0,1,1) y (1,1,0) al magenta, cian y amarillo respectivamente, colores secundarios y respectivamente complementarios del verde, rojo y azul. La diagonal del cubo representa la gama de grises desde el negro al blanco. En esta diagonal cada punto o color se caracteriza por tener la misma cantidad de cada color primario.

Las imágenes con modelo RGB contienen tres planos de imágenes independientes, uno para cada color primario. Cuando estas tres imágenes son proyectadas a un monitor RGB, la pantalla de fósforo produce una imagen de color compuesto. El procesamiento de imágenes en color, utilizando el modelo RGB, toma sentido cuando las imágenes se expresan naturalmente en términos de tres planos de color. Actualmente, muchas cámaras a color utilizadas para adquirir imágenes digitales, utilizan el formato RGB.

Lo anterior convierte al modelo RGB en un modelo de gran importancia para el procesamiento de imágenes, a pesar de que no deriva en un proceso intuitivo para determinadas aplicaciones como por ejemplo la de comparar colores.

## Visión Artificial

La *Visión Artificial* es una disciplina que tiene como finalidad la extracción de información del mundo físico a partir de imágenes, utilizando para ello un computadora. Se trata de un campo de conocimiento con un objetivo ambicioso y complejo que en los últimos años ha evolucionado de manera drástica con el desarrollo de las

computadoras y sensores de captura de imágenes. Un sistema de *Visión Artificial* actúa sobre una representación de una realidad que le proporciona información sobre brillo, colores, formas, etc. Estas representaciones se pueden agrupar principalmente en imágenes estáticas, escenas tridimensionales e imágenes en movimiento [2].

La *Visión Artificial*, también llamada *Visión Computacional*, pretende con determinadas restricciones reproducir el comportamiento del ser humano. En un sistema de *Visión Artificial* se definen las siguientes cuatro fases bien diferenciadas en el procesamiento de imágenes, mostradas en la Figura C.

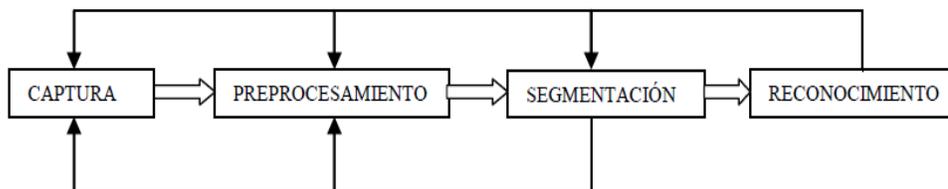


Figura C.4: Etapas típicas en un sistema de *Visión Artificial*.

- La primera fase, consiste en la ***Captura o Adquisición*** de las imágenes digitales mediante algún tipo de sensor, fundamentalmente una cámara. Una imagen que es capturada a través de un sensor, se guarda en la memoria de la computadora como un arreglo o matriz de puntos, llamados *píxeles*. Dependiendo del tipo de imágenes que se traten, cada punto representará la intensidad o brillo de la imagen, si se trabaja con imágenes en blanco y negro; o tres matrices de puntos, uno para rojo, otro para verde y otro para azul si lo hacemos con imágenes a color; su combinación representa los diferentes colores.
- La segunda etapa tiene por objeto el tratamiento digital de imágenes para facilitar las etapas posteriores. En esta fase, conocida como ***Preprocesamiento***, mediante filtros y transformaciones geométricas se tratan de eliminar partes indeseables de la imagen y realzar los detalles de interés para su posterior análisis.
- ***La Segmentación*** es la siguiente fase, y consiste en particionar una imagen en regiones homogéneas con el fin de extraer características de dichas regiones (color, textura, forma, ...) para la posterior etapa de reconocimiento.
- Para finalizar, se llega a la etapa denominada de ***Reconocimiento o Clasificación***. Valiéndose del análisis de ciertas características que se establecen previamente para diferenciar los objetos segmentados, se pretende discernir entre unos y otros.

Diferentes aspectos han ayudado a llevar a la *Visión Artificial* al campo de las aplicaciones industriales como: el aumento de las prestaciones de las computadoras, desarrollo de técnicas algorítmicas eficientes para el abordaje de los problemas, etc. Estos aspectos han permitido realizar el tratamiento de imágenes en tiempo real.

## Calibración de la cámara

Con frecuencia se tiene en cuenta una proyección ideal y perfecta de la cámara (modelo pinhole según la nomenclatura). En la realidad esto no es así, y uno de los factores que más influyen en que la cámara obtenga imágenes deformadas con respecto a la realidad es la distorsión de la lente. Esto sobre todo se observa en lentes de bajo costo como las usadas en la fabricación de webcams y algunas cámaras de vídeo baratas.

Podemos encontrar dos tipos básicos de distorsión de lente: radial y tangencial, como se muestra en la Figura C.

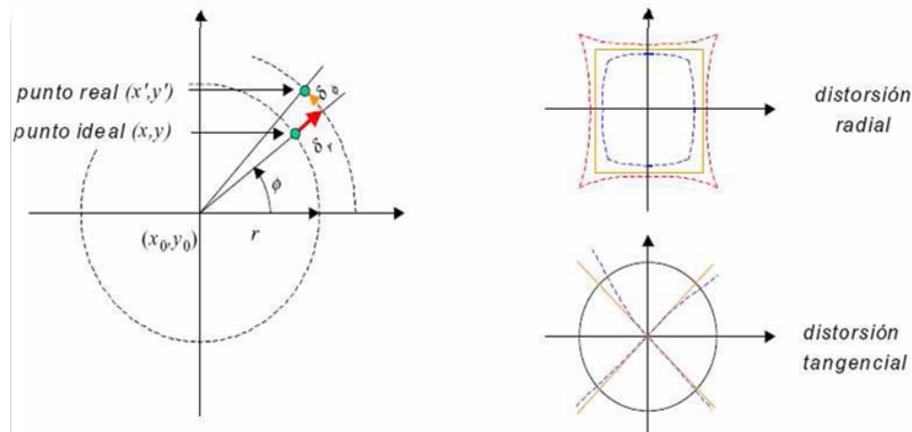


Figura C.5: Tipos básicos de distorsión en una lente.

Se caracteriza la distorsión radial (normalmente la más predominante) a partir de la definición de la ecuación de la circunferencia. Podemos definir las coordenadas de un punto proyectado idealmente (no distorsionado) por  $(x', y')$ . Por tanto para un punto X tenemos:

$$(x', y', 1)^T = [I|0]X_{cam} \tag{C.1}$$

siendo  $X_{cam}$  el punto 3D en coordenadas de cámara, y relacionado con las coordenadas del mundo. Entonces el punto proyectado se relaciona con el punto ideal con

un desplazamiento radial. Por tanto, la distorsión radial se puede modelar como la expresión

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = L\tilde{r} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (\text{C.2})$$

siendo  $(x', y')$  la posición ideal de la imagen (obtenida por una proyección lineal).  $(x_d, y_d)$ , es la posición final de la imagen, después de aplicar distorsión radial.  $r'$  es la distancia radial  $\sqrt{x'^2 + y'^2}$  desde el centro de distorsión.  $L(r')$  es un factor de distorsión, que es función del radio  $r'$ . En coordenadas de pixel, la corrección se escribe como:

$$\begin{aligned} \hat{x} &= x_c + L(r)(x - x_c) \\ \hat{y} &= y_c + L(r)(y - y_c) \end{aligned} \quad (\text{C.3})$$

donde  $(x, y)$  son las coordenadas medidas,  $(\hat{x}, \hat{y})$  son las coordenadas corregidas, y  $(x_c, y_c)$  es el centro de la distorsión radial, con  $r^2 = (x - x_c)^2 + (y - y_c)^2$ . Hay que remarcar que si la relación de aspecto no es la unidad entonces es necesario corregir esto antes de calcular  $r$ .

En cuanto a la elección de la función de distorsión y su centro, la función  $L(r)$  sólo está definida para valores positivos de  $r$  y en  $L(0) = 1$ . Una aproximación a la función que modela la distorsión radial se puede dar a través de un polinomio de Taylor de la forma  $L(r) = 1 + k_1r + k_2r^2 + \dots$ . Los coeficientes para la corrección radial  $k_1, k_2, k_3, \dots, x_c, y_c$  se consideran parte de la calibración interna de la cámara.

El punto principal de la proyección suele usarse como centro inicial para la distorsión radial, aunque no necesariamente deben coincidir exactamente. Hay que apuntar que la corrección radial será más precisa en cuanto se usen más coeficientes  $k_i$  para el modelado de la distorsión. Normalmente suelen utilizarse los dos primeros  $k_1, k_2$  aunque puede tomarse todos los parámetros que se quieran para una estimación más exacta de la distorsión de la lente.

Hay diversos métodos para calibrar una cámara. Pueden utilizarse patrones de calibración o realizar la llamada autocalibración. Los métodos de calibración clásicos se pueden clasificar en función de diferentes criterios [25].

# Apéndice D

## Herramientas computacionales

### Recursos/Herramientas utilizadas

En este apéndice se describen las herramientas computacionales utilizadas para la estimación de posición de los robots móviles.

#### OpenCV

El 13 de Junio de 2000, Intel Corporation anunció que estaba trabajando en la realización de una nueva biblioteca de estructuras/funciones en lenguaje C junto con un grupo de prestigiosos investigadores (Jitendra Malik, University of California Berkeley; Takeo Kanade, Carnegie, Mellon University; Pietro Perona, NSF Engineering Research Center ...) en visión por computadora.

De esta forma nació The Open Source Computer Vision Library (OpenCV), especialmente diseñado para el tratamiento, captura y visualización de imágenes en áreas como interacción hombre-máquina, robótica, monitorización, biométrica, segmentación y reconocimiento de objetos y seguridad y que proporciona bibliotecas de tipos de datos estáticos y dinámicos (matrices, grafos, árboles, etc.). Está optimizada para ser usado bajo procesadores Intel, particularmente para la tecnología MMX, Pentium Pro, Pentium III y Pentium 4, pero puede ser usada bajo cualquier otro tipo de procesadores.

OpenCV es ampliamente utilizado en entornos de vigilancia y seguimiento de objetos. Ha sido utilizado en grandes proyectos como el DARPA Grand Challenge [16] por el equipo ganador y en la herramienta SwisTrack [16] para navegación de robots.

La biblioteca OpenCV es una API de más de 300 funciones escritas en lenguaje C y que tienen las siguientes características:

- Su uso es libre tanto para uso comercial como no comercial.

- No utiliza bibliotecas numéricas externas, aunque puede hacer uso de alguna de ellas en tiempo de ejecución si están disponibles.
- Es compatible con IPL (*Intel Processing Library*) y utiliza IPP (*Intel Integrated Performance Primitives*) para mejorar su rendimiento, si están disponibles en el sistema.

Además, OpenCV dispone de interfaces para otros lenguajes y entornos como:

- EiC - Interprete ANSI C escrito por Ed Breen, actualmente se encuentra en desuso. Hawk y CvEnv son entornos interactivos (escritos en MFC y TCL, respectivamente).
- Ch - Interprete ANSI C/C++ con algunas características de scripting, desarrollado y mantenido por la compañía SoftIntegration® (<http://www.softintegration.com>). Los *wrappers* para Ch están disponibles en *opencv/interfaces/ch*.
- Matlab - Entorno para procesamiento numérico y simbólico desarrollado por *Mathworks*. El interfaz de Matlab para algunas funciones de OpenCV se encuentra disponible en *opencv/interfaces/matlab/toolbox*. En cuanto a la integración con Matlab OpenCV puede utilizar las estructuras nativas de Matlab, y es compatible con el *Image Processing Toolbox*.

### Estructura

Todas las herramientas de alto nivel de OpenCV hacen uso de un paquete de clases C++ y funciones C que usan a su vez funciones muy eficientes escritas en C. Concretamente, el conjunto de funciones suministradas por la librería OpenCV pueden agruparse en los siguientes bloques:

- Estructuras y operaciones básicas.
- Procesamiento y análisis de imágenes: filtros, momentos, histogramas...
- Análisis estructural: geometría, procesamiento del contorno, etc.
- Análisis del movimiento y seguimiento de objetos: plantillas de movimiento, seguidores, flujo óptico, etc.
- Reconocimiento de objetos: objetos propios (*eigne objects*), modelos HMM, etc.
- Calibración de la cámara: *morphing*, geometría epipolar, estimación de la pose, etc.
- Reconstrucción tridimensional (funcionalidad experimental): detección de objetos, seguimiento de objetos tridimensionales, etc.

- Interfaces gráficas de usuarios y adquisición de video.

OpenCV se puede dividir en varias partes bien diferenciadas para su uso, que están a su vez divididas en cabeceras (\*.h) y son las siguientes: Cv, CvAux, HighGUI y CvCam.

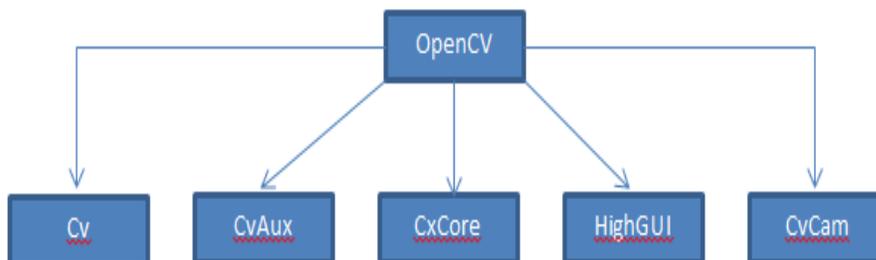


Figura D.1: Estructura OpenCV.

Cada parte está especializada en una cosa en concreto:

- En las bibliotecas **Cv** se encuentran los operadores estándares para el cálculo de gradientes, detección de bordes, esquinas y contornos, conversión de modelos de color, operaciones morfológicas como dilatación, erosión y adelgazamiento, transformadas geométricas (*Hough*), operaciones con momentos (momento central, momentos invariantes, etc.), operaciones con histogramas, y de *matching*. Sobre análisis estructural, se puede encontrar funciones para procesamiento de contornos (momentos, representación jerárquica de contornos...) y computación geométrica. También se encuentran funciones para tratar aspectos de análisis de movimiento y seguimiento de objetos, procesamiento de flujo óptico, reconocimiento de patrones y estimación de la pose de objetos, así como funciones para reconstrucción en 3D (calibración de la webcam, reconocimiento de gestos...).
- En **CvAux** están las funciones en fase de prueba o experimentales, como pueden ser algoritmos para visión estéreo, seguimiento 3D...
- **HighGUI** permite la escritura/lectura de imágenes en numerosos formatos (BMP, JPEG, TIFF, Pxm, Sun Raster, etc.) y la captura de *stream* de video de capturadoras Matros® y cámaras/capturadoras con controlador VFW/WDM (la mayoría del mercado).

También permite la creación de ventanas para visualizar imágenes en ellas. Estas ventanas recuerdan su contenido (no es necesario implementar *callbacks*

de repintado) y además, proporciona mecanismos muy fáciles para interactuar con ellas: *trackbars* que capturan la entrada del teclado y el ratón.

- **CvCam** proporciona una única interfaz de captura y reproducción bajo Linux y Win32; *callbacks* para la gestión de *stream* de video o ficheros AVI y un mecanismo fácil para implementar visión estereo con dos cámaras USB o estereo-cámara.

A partir de la **version 0.9.6**. Estas funciones específicas de cámara se eliminaron para el sistema GNU/Linux, dejando solamente operativas para el trabajo con cámaras el módulo HighGUI.

- **CxCore** dispone de estructuras de datos dinámicas y operadores para estas estructuras. Entre estas estructuras de datos se pueden encontrar vectores (*cvArr*), matrices (*cvMat*), imágenes, grafos, etc.

## C++

En la actualidad, C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales la ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones.

El lenguaje C++ mantiene las ventajas de C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original. La evolución de C++ ha continuado con la aparición de Java, un lenguaje creado simplificando algunas características de C++ y añadiendo otras, que se utiliza para realizar aplicaciones en Internet.

Las principales características del lenguaje C++ son:

- Abstracción (encapsulación).
- Soporte para programación orientada a objetos (polimorfismo).
- Soporte de plantillas o programación genérica (templates).

Se puede decir que C++ es un lenguaje que abarca tres paradigmas de programación: programación estructurada, programación genérica y programación orientada a objetos. Además, añade una serie de propiedades que se encuentran más difícilmente en otros lenguajes de alto nivel:

- Posibilidad de redefinir los operadores.
- Identificación de tipos en tiempo de ejecución (*Runtime Type Information, RTTI*).

C++ está considerado por muchos como el lenguaje más potente, permitiendo trabajar tanto a alto como a bajo nivel. Por otro lado, es uno de los lenguajes que contiene menos automatismos, lo que dificulta su aprendizaje.

## **Dev-C++**

Dev-C++ es un entorno de desarrollo integrado (IDE por sus siglas en inglés) para programar en lenguaje C/C++. Usa MinGW que es una versión de GCC (*GNU Compiler Collection*) como su compilador. Dev-C++ puede además ser usado en combinación con Cygwin y cualquier compilador basado en GCC.

El Entorno está desarrollado en el lenguaje Delphi de Borland. Tiene una página de paquetes opcionales para instalar, con diferentes bibliotecas de código abierto.



# Bibliografía

- [1] A. D. Medeiros, A.P. Araújo, P. N. Guerra, P.J. Alsina. Linear modelling and identification of a mobile robot with differential drive.
- [2] A. Sánchez, L. Esteban, F. Vélez, A. Moreno. Visión por Computador. Dykinson, 2003.
- [3] A.S. Fukunaga Y. U. Cao and A.B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robotics*, 4(1):34-46, 1997.
- [4] A. Williams G. Lafferriere, J. Caughman. Graph theoretic methods in the stability of vehicle formations. In *American Control Conference*, pp. 3729- 3734. IEEE, June 2004.
- [5] B. Fidan B.D.O. Anderson J.M. Hendrickx, Y. Changbin Yu. Rigidity and persistence of meta-formations. In *Conference on Decision and Control*, pp. 5980-5985. IEEE, December 2006.
- [6] B. Francis Z. Lin and M. Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions on Automatic Control*, 50(1):121-127, 2005.
- [7] B. Anthony y FowlerWallace. *Dinámica, mácanica para ingeniería*. Pearson Educación, México, 1999.
- [8] J. Carrera y F Garrido. Software de simulación del robot puma unimation 560 basado en el modelo cinemático dirrecto e inverso. Tesis, 2000.
- [9] C. De la Cruz y R. Carelli. Linealización con realimentación del modelo dinámico de un robot móvil y control de seguimiento de trayectoria. *AADECA*, 2006.
- [10] C. De la Cruz y R. Carelli. Control Centralizado de formación usando una cámara omnidireccional. *COMRob*, Torreón, Coahuila, Mexico, October 2007.
- [11] C. Reynolds. Flocks, birds, and schools: A distributed behavioral model. *Computer graphics*, 21(1):25-34, 1987.

- 
- [12] D. Swaroop and J. K. Hedrick. String stability of interconnected systems. In American Control Conference. pp 1806-1810. IEEE, June 1995.
- [13] D. V. Dimarogonas and Kostas J. Kyriakopoulos. Distributed cooperative control and collision avoidance for multiple kinematic agents. In Conference on Decision and Control. pp 721-726. IEEE, December 2006.
- [14] E. Pagello T. Arai and L. E. Parker. Guest editorial advances in multirobot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655-661, 2002.
- [15] F. Miyazaki T. Nōguchi. Y. Kanayama, Y. Kimura. A stable tracking control scheme for an autonomous mobile robot. *proc. of IEEE Int. Conf. on Robotics and Automation*, 1990.
- [16] G. Bradski, Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. OŠReilly Media, Inc., 2008.
- [17] W. E. Gómez. *Cinemática de robots manipuladores*. Tesis, 2004.
- [18] H. G. Tanner and A. Kumar. Towards decentralization of mutli-robot navigation functions. In *International Conference on Robotics and Automation*, pp 4132-4137. IEEE, 2005.
- [19] H. G. Tanner. Flocking with obstacle avoidance in switching networks of interconnected vehicles. In *International Conference on Robotics and Automation*, pp 3006-3011. IEEE, April 2004.
- [20] H. Yamaguchi. A distributed motion coordination strategy for multiple non-holonomic mobile robots in cooperative hunting operations. *Robotics and Autonomous Systems*, 43(1):257-282, 2003.
- [21] I. Suzuki H. Ando, Y. Oasa and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818-828, 1999.
- [22] I. Suzuki Y. Asahiro, H. Asama and M. Yamashita. Improvement of distributed control algorithms for robots carrying an object. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics*. pp 608-613. IEEE, October 1999.
- [23] J. Bruce and M. Veloso, Fast and accurate vision-based pattern detection and identification. In *IEEE International Conference on Robotics and Automation*. IEEE, May 2003.
- [24] J. C. Hamann W. M. Spears, D. F. Spears and R. Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17(1):137-162, 2004.

- 
- [25] J. I. González. Estudio experimental de métodos de calibración y autocalibración de cámaras. UPGC 2003.
- [26] J.J. Esqueda. Fundamentos de procesamientos de imágenes. Instituto Tecnológico de Ciudad Madero. CONACTEC 2002.
- [27] J. P. Desai. A graph theoretic approach for modeling mobile robot team formations. *Journal of Robotic Systems*, 19(11):511-525, 2002.
- [28] J.P Ostrowski J.P. Desai and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905-908, 2001.
- [29] K. J. Kyriakopoulos D. V. Dimarogonas. On the state agreement problem for multiple unicycles with varying communication links. In *Conference on Decision and Control*. pp 4283-4288. IEEE, December 2006.
- [30] L. McCoy J. Baillieul. The combinatorial graph theory of structured formations. In *Conference on Decision and Control*. pp 3609-3615. IEEE, December 2007.
- [31] M. Broucke Z. Lin and B. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Transactions on Automatic Control*, 49(4):622-629, 2004.
- [32] M. Broucke J. A. Marshall and B. Francis. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 49(11):1963-1974, 2004.
- [33] M. Egerstedt A. Muhammad. Connectivity graphs as models of local interactions. In *Conference on Decision and Control*. pp 124-129. IEEE, December 2004.
- [34] M. Ishikawa and S. Mitsuji. State estimation of non-holonomic mobile robots using nonlinear observers. *IEEE, Robotics and Automation*, 2:1379-1384, 1995.
- [35] M. J. Mataric J. Fredslund. General algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation*, 18(5):837-846, 2002.
- [36] M. L. James. *Engineering Mechanics Dynamics*. John Wiley and Sons, Inc., United States of America, 2001.
- [37] N. Banavar Ravi and Sankaranarayanan Velupillai. *Switched finite time control of a class of underactuated systems*. Springer Verlay, Germany, 2006.

- 
- [38] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Conference on Decision and Control*, pp 2968-2973. IEEE, December 2001.
- [39] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):151-173, May 2004.
- [40] Paul Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, pp. 511-518, 2001.
- [41] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, Inc., New Jersey, 2002.
- [42] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *IEEE ICIP 2002*, volume 1, pp. 900-903, September 2002.
- [43] R. M. Murray J. A. Fax. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465-1476, 2004.
- [44] R. Olfati-Saber and R. M. Murray. Agreement problems in networks with directed graphs and switching topology. In *Conference on Decision and Control*, pp. 4126-4132. IEEE, December 2003.
- [45] R. Olfati-Saber and R. M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *The 15th IFAC World Congress*, Barcelona, Spain. IFAC, July 2002.
- [46] R. Prádanos, J.M. Sanz, M. Domínguez, M. Espinosa. Perspectivas de futuro en la variable color en ingeniería gráfica. En *XIV Congreso Internacional de Ingeniería Gráfica*. Santander. 2002.
- [47] R. Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Florida, USA 1986.
- [48] S. G. Loizou and K. Kyriakopoulos. Close loop navigation for multiple holonomic vehicles. In *International Conference on Intelligence Robots and Systems*, pp. 2861-2866. IEEE, October 2002.
- [49] Sánchez López G. Yamel y Badillo Branco F. Enrique. *Modelación cinemática y dinámica del dispositivo háptico phantom 1.0*. Tesis, 2002.
- [50] Shimon Y. Nof. *Handbook of industrial robotics*, Second Edition. John Wiley and Sons, Inc., United States of America, 1999.

- [51] M.W.Spong and M. Vidyasagar. Robot Dynamics and Control, Wiley, New York, 1989.
- [52] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. IEEE Transactions on Robotics and Automation, 14(3):926-939, 1998.
- [53] Tan Lam Chung Trong Hieu Bui, Tan Tien-guyen and Sang Bong Kim. A simple nonlinear control of a two-wheeled welding mobile robot. International Journal of Control, Automation Systems, 1, 2003.
- [54] V. Fernando, E. Slawiński, A. Rosales, G. Scaglia. Dynamic control by using numerical methods for mobile robots. XII Reunión de Trabajo en Procesamiento de la Información y Control.
- [55] V. Kumar J.P. Desai and J.P. Ostrowski. Controlling formations of multiple mobile robots. In Proc. International Conference on Robotics and Automation. pp 2864 - 2869. IEEE, May 1998.
- [56] V. Kumar H. G. Tanner and G. J. Pappas. Leader-to-formation stability. IEEE Transactions on Robotics and Automation, 20(3):443-455, 2004.
- [57] V. Kumar H. G. Tanner and G. J. Pappas. Stability properties of interconnected vehicles. In International Symposium on Mathematical Theory of Networks and Systems. IEEE, August 2002.
- [58] W. Jean-Jaques and E. Slotine. Applied Nonlinear Control. 1991.
- [59] W.L. Xu. W. Dong. Adaptive tracking control of uncertain nonholonomic dynamic system. IEEE Transactions On Automatic Control, pages 450-454, 2001.
- [60] X. Chen Y. Li. Stability on multi-robot formation with dynamic interaction topologies. In International Conference on Intelligent Robots and Systems, pp. 394-399. IEEE/RSJ, September 2005.
- [61] Y. Q. Chen and Z. Wang. Formation control: A review and a new consideration. In International Conference on Intelligent Robots and Systems. pp 3181-3186. IEEE/RSJ, August 2005.
- [62] Z. Zhang. A flexible new technique for camera calibration. In Transactions on Pattern Analysis and Machine Intelligence. IEEE, 2000.
- [63] <http://www.vision.caltech.edu/bouguetj/calibdoc/>
- [64] Intel.IntelOpenSourceComputerVisionLibrary, v1.1.ore.http :  
[//sourceforge.net/projects/opencvlibrary/](http://sourceforge.net/projects/opencvlibrary/), October2008.