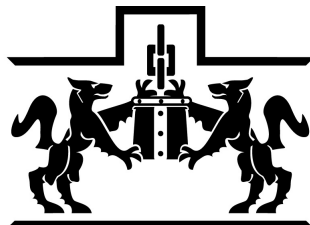


# Universidad Iberoamericana

Estudios con Reconocimiento de Validez Oficial por Decreto Presidencial  
Del 3 de abril de 1981



LA VERDAD  
NOS HARÁ LIBRES

UNIVERSIDAD  
IBEROAMERICANA

CIUDAD DE MÉXICO ®

## “Coordinación de movimiento de sistemas multiagentes heterogéneos”

TESIS

Que para obtener el grado de

DOCTOR EN CIENCIAS DE LA INGENIERÍA

Presenta

**ALEXANDRO LÓPEZ GONZÁLEZ**

Director: Dr. Eduardo Gamaliel Hernández Martínez

Codirector: Dr. Guillermo Fernández Anaya

Lectores: Dr. Alberto Luviano Juárez

Dr. Jesús Alberto Meda Campaña

Dr. José Emiliano Martínez Ordaz

# “Coordinación de movimiento en sistemas robóticos multiagentes heterogéneos”

Universidad Iberoamericana



Alexandro López González

13 de enero de 2017

# Resumen

En control de formación establece estrategias para la convergencia de un grupo de robots móviles a patrones geométricos predefinidos. En el acercamiento tradicional de consenso posicional, se asume que cada robot conoce la posición espacial exacta de otros robots del grupo con respecto a un marco de coordenadas seleccionado, y se diseñan leyes de control con base en las posiciones relativas deseadas.

Para obtener un esquema de coordinación más descentralizado en esta tesis se presenta una alternativa basada en distancias y orientación, considerando que los robots podrían estar equipados con sensores para dichas variables. Usando técnicas Lyapunov se desarrolla un algoritmo de control, que usa funciones potenciales atractivas-repulsivas basadas en distancia y orientación, permitiendo la formación de robots de primer orden a distancias deseadas entre estos, respetando topologías de conectividad representadas por medio de grafos no dirigidos. Esta ley de control incluye términos repulsivos para evitar colisiones entre robots y un término tipo integral que evita la necesidad de normalización en las velocidades de las ruedas del robot. Se realizan simulaciones numéricas y experimentos con robots tipo unicycle.

El esquema basado en distancias y orientación se extiende para incluir robots de primer y segundo orden, en un algoritmo de control planar, permitiendo la formación de robots heterogéneos en cuanto a su modelo cinemático o dinámico, que incluye términos repulsivos para evitar colisiones entre robots. Se comprueba la convergencia a las distancias deseadas para el caso de grafos no dirigidos y se realizan simulaciones numéricas con los modelos cinemáticos y dinámicos, de los robots unicycle, omnidireccional y cuadróptero. Adicionalmente se realizan experimentos con robots omnidireccionales y unicycles.

También se presenta el diseño y experimentos con la plataforma de robots heterogéneos desarrollada. La plataforma incluye un núcleo de control que puede ser usado para diferentes tipos de sistemas de locomoción. La plataforma se usa para demostrar la validez de las leyes de control, implementando dos robots con ruedas, un robot unicycle y un robot omnidireccional.

# Dedicatorias

Dedico esta tesis a mis padres que a lo largo de toda mi trayectoria académica me han brindado los recursos, el apoyo, y los conocimientos necesarios para poder llevar a cabo mis metas. Mi madre, quien siempre ha sido mi confidente y mi soporte. Mi padre, mi maestro y compás moral.



También dedico este trabajo a mi Jema, quien me ha brindado su amor, apoyo y comprensión.

♪Tu y yo en el balcón  
que asoma a los plantíos de plátano  
los patrones se han muerto y tu...  
aún sigues trapeando el piso de ajedrez  
ven acá que desde aquí se logran ver  
todas las tierras que ahora nuestras son  
y un zambo el heredero será  
pues de un indio y una negra eso da ♪

# Declaración

Yo Alexandro López González en calidad de autor de la tesis, “Coordinación de movimiento en sistemas robóticos multiagentes heterogéneos” autorizo para su utilización en la Biblioteca Francisco Xavier Clavigero en formato electrónico o impreso y sin fines de lucro, cuyo titular es la Universidad Iberoamericana Ciudad de México, por un período ilimitado a contar desde la fecha de la publicación de la presente, con expresa renuncia a los derechos derivados de su explotación en dichos medios y durante el mencionado período.

# Agradecimientos

Agradezco a mi asesor de tesis, Dr. Eduardo G. Hernández , por su apoyo, esfuerzo, dedicación y amistad. Sus enseñanzas, sus conocimientos, sus indicaciones, su persistencia, su paciencia y su motivación han sido fundamentales para mi formación como investigador.

Agradezco también a mi compañero y amigo, Mtro. Pablo Paniagua Contro, por su trabajo, sus enseñanzas, su apoyo y jovialidad, que permitieron la realización de este documento.

Por su participación indispensable en la realización de esta tesis agradezco a:

Dr. Guillermo Fernández Anaya

Dr. José Job Flores Godoy

Dr. Enrique Ferreira Vázquez Dr. Alberto Luviano Juárez

Dr. Jesús Alberto Meda Campaña

Dr. José Emiliano Martínez Ordaz

Dra. Anabel Arrieta Ostos

# Índice general

<b>1. Introducción</b>	<b>10</b>
1.1. Planteamiento general del problema . . . . .	12
1.2. Objetivos . . . . .	13
1.2.1. Generales . . . . .	13
1.2.2. Específicos . . . . .	13
1.3. Justificación . . . . .	14
1.4. Contribución . . . . .	15
1.5. Estructura . . . . .	15
<b>2. Antecedentes</b>	<b>16</b>
2.1. Los Sistemas de Robots Multiagentes . . . . .	16
2.1.1. Coordinación de movimiento de robots móviles heterogéneos . . . . .	18
2.1.2. Control de formación basado en posiciones y en distancias . . . . .	19
2.1.3. Desarrollo de plataformas de robots heterogéneos . . . . .	21
2.2. Fundamentos teóricos . . . . .	21
2.2.1. Los sistemas de robots multiagentes . . . . .	21
2.2.2. Teoría de grafos . . . . .	22
2.2.3. Algoritmo de consenso de primer orden . . . . .	27
2.2.4. Algoritmo de consenso de segundo orden . . . . .	28
2.2.5. Formación de primer orden con posiciones relativas deseadas . . . . .	29
2.2.6. Formación basada en distancias . . . . .	31
2.2.7. Evasión de colisiones . . . . .	32
2.2.8. Modelado de robots móviles no holonómicos . . . . .	33
<b>3. Control de formación basado en distancia y orientación</b>	<b>41</b>
3.1. Planteamiento del problema . . . . .	41
3.2. Control y prueba de convergencia . . . . .	42
3.3. Simulaciones numéricas . . . . .	47
3.4. Experimentos . . . . .	49
3.4.1. Resultados experimentales . . . . .	49
<b>4. Formación de robots heterogéneos basada en distancia y orientación</b>	<b>53</b>
4.1. Formación basada en distancias y orientaciones para robots de primer y segundo orden . . . . .	53
4.2. Simulaciones numéricas . . . . .	56
4.2.1. Resultados . . . . .	56
4.3. Trabajo experimental . . . . .	57

<b>5. Plataforma experimental</b>	<b>60</b>
5.1. Plataforma Irobot® Create . . . . .	60
5.2. Desarrollo de la plataforma de robots móviles heterogéneos . . . . .	61
5.2.1. Estructura . . . . .	63
5.2.2. Núcleo de control . . . . .	65
5.2.3. Implementación del sistema de control . . . . .	66
<b>6. Conclusiones y trabajo futuro</b>	<b>68</b>
6.1. Conclusiones . . . . .	68
6.2. Trabajo futuro . . . . .	69
<b>A. Diagramas de ensamble estructural de robots</b>	<b>71</b>
<b>B. Diagrama y códigos del núcleo de control</b>	<b>74</b>
B.1. Diagrama del nucleo de control HUEX . . . . .	75
B.2. Códigos del núcleo de control HUEX . . . . .	76
<b>C. Archivos en Matlab de simulaciones y experimentos</b>	<b>79</b>
C.1. Plataforma experimental heterogénea . . . . .	79



# Índice de figuras

2.1. Grafo detallado . . . . .	22
2.2. Grafo con autoaristas y nodos aislados . . . . .	23
2.3. Grafo de conectividad A . . . . .	24
2.4. Rigidez en Grafos . . . . .	24
2.5. Grafo completo y bidireccional . . . . .	25
2.6. Grafo de persecución cíclica . . . . .	26
2.7. Grafo de persecución cíclica bidireccional . . . . .	26
2.8. Grafo de líder bidireccional . . . . .	27
2.9. Grafo de conectividad B . . . . .	30
2.10. Robot tipo Uniciclo . . . . .	34
2.11. Rueda Sueca o mecanum . . . . .	35
2.12. Robot tipo Omnidireccional . . . . .	35
2.13. Movimientos del robot omnidireccional . . . . .	36
2.14. Robot Cuadricóptero . . . . .	37
2.15. Niveles de control para el cuadricóptero . . . . .	40
3.1. Distancia relativa y orientación entre dos robots . . . . .	42
3.2. Simulación numérica usando 4 robots . . . . .	48
3.3. Experimento uno con $\varrho_p = 0.3$ y $\varrho_e = 0$ . . . . .	50
3.4. Experimento dos con $\varrho_p = 0.3$ y $\varrho_e = 0.005$ . . . . .	52
4.1. Simulación de formación . . . . .	58
4.2. Experimentos de formación con robot unicycle y omnidireccional . . . . .	59
5.1. Plataforma de experimentos con 3 Irobot Create . . . . .	61
5.2. Esquema de la plataforma de experimentos . . . . .	62
5.3. Implementación de control . . . . .	62
5.4. Núcleo Octagonal . . . . .	63
5.5. Planos de SRM heterogéneo . . . . .	64
5.6. Construcción de unicycles y omnidireccionales . . . . .	64
5.7. Ruedas de unicycles y omnidireccionales . . . . .	65
5.8. Placa HUEX . . . . .	65
5.9. Núcleo de control HUEX . . . . .	66
5.10. sistema de captación de movimiento Vicon . . . . .	66
5.11. Marcadores en robots y visualización en Vicon . . . . .	67
6.1. Modelos en desarrollo . . . . .	70
A.1. Diagrama de robot unicycle . . . . .	72
A.2. Diagrama de robot omnidireccional . . . . .	73

B.1. Diagrama conexión de la placa HUEX . . . . .	75
---	----

# Capítulo 1

## Introducción

La robótica es una rama de la ingeniería que aborda la investigación y desarrollo de sistemas mecánicos especializados denominados robots. Las disciplinas de la ingeniería que se involucran en la robótica son al menos, pero no limitadas a la mecánica, la física, la eléctrica, la electrónica, la industrial, el control, la inteligencia artificial y la computación. Robot es una entidad artificial mecánica mientras que el bot es una entidad artificial virtual. El término robot nace de la obra Robots Universales Rossum, escrita por *Karel Čapek* en 1920, a quien se le atribuye el término robot, derivado de la palabra checa robota, que significa trabajador forzado. La inepción de la robótica como el área de estudio de los robots se le atribuye al escritor Isaac Asimov.

Existen innumerables definiciones de robot, que atribuyen un amplio rango de características. Dos de las características más usadas son:

- Un robot debe tener la capacidad de modificar su ambiente, o su estado (posición, configuración, etcétera) dentro de dicho ambiente.
- Un robot debe tener algún grado de toma de decisión autónoma.

Note que un robot no necesita de forma obligatoria el uso de sensores, se pueden programar lazos de control sin adquisición de variables. Los robots tienen muchas formas físicas y existen muchas maneras de clasificarlos, una de ellas es:

- Robots manipuladores
  - Cadena cinemática abierta
  - Cadena cinemática cerrada (Paralelos)
- Robots móviles (clasificados por tipo de locomoción)
  - Ruedas (AGVs)
  - Patas (bipedos, cuadrúpedos, etcétera)
  - Alas (UAV, ala fija, ala rotatoria)
  - Propelas (Marinos, submarinos)
- Locomoción especial
  - Inspirados biológicamente

- Reconfigurables

Como lo denota *Karel Čapek* el objetivo primordial para el estudio de los robots es la automatización de tareas tediosas o peligrosas para el humano. Otro motivador para el estudio de los robots es el aspecto económico, la repetitividad de tareas, la precisión, el costo y el tiempo de trabajo, son algunos de los aspectos que generan ahorro económico en las industrias.

Históricamente el acercamiento más fuerte a la robótica es el de los manipuladores de cadena cinemática abierta, conocidos comúnmente como brazos robot. El éxito de estos robots en plantas de producción ha sido moderado debido a que las líneas de producto se adaptan a los robots para poder trabajar de forma efectiva. Una vez funcionando, la cantidad de trabajadores necesarios se reducen al mínimo. Sin embargo este acercamiento es económicamente viable en pocas situaciones. Este hecho ha impulsado estudio de otros tipos de robots, los cuales podrían adaptarse de forma más sencilla a el ambiente humano, sin necesidad de modificaciones.

La robótica móvil estudia robots que usan diversos tipos de locomoción para desplazar toda su estructura física en espacios bidimensionales o tridimensionales. La variedad de métodos de locomoción es muy amplia, robots con ruedas, patas, alas, propelas, robots que saltan, trepan o que se arrastran, entre muchos otros. La ventaja que tienen los robots móviles radica en que pueden navegar espacios naturales o artificiales distintos sin necesidad de modificar el ambiente para su funcionamiento. Esto representa una ventaja en tareas que el humano realiza periódicamente, de forma tal que un robot móvil podría sustituir al humano en tareas físicas. Un ejemplo de esto es la aspiradora robótica Roomba<sup>®</sup> que puede realizar la tarea de aspirar los hogares del humano de forma automática e independiente.

Dos de los problemas que enfrenta la robótica móvil son el consumo energético y la complejidad de tareas. El consumo energético es un problema porque al no tener un suministro continuo de energía por medio de un cable, como los robots móviles sufren de tiempos cortos de operación. Segundo, la complejidad de una tarea que por necesidad, aumenta la complejidad del robot, por lo que los costos aumentan junto con el diseño mecánico, electrónico y computacional. Ambos problemas, la complejidad y la energía, se abordan por medio de la optimización. Con la mejora de los componentes y procedimientos de maquinado, el precio baja de forma drástica, y la tecnología de baterías mejora mientras que el consumo de energía se reduce. Aún así, se han buscado soluciones para estos problemas, una de ellas son los sistemas robóticos multiagente.

Los sistemas robóticos multiagente (SRM) son grupos de robots móviles cuya misión es la de realizar tareas complejas de forma conjunta, cooperando para llevar a cabo su finalidad. Las ventajas que se buscan en un SRM son:

- Los robots son más simples y por lo tanto más económicos.
- Los robots poseen leyes descentralizadas, es decir que tienen tareas simples que en conjunto formarían parte de la solución del objetivo final.
- Existe mayor tolerancia a fallas. Si un robot se ve impedido para llevar a cabo su tarea, otros pueden sustituirlo y así completar la tarea.

Los SRM son un área muy activa de investigación y desarrollo tecnológico, que afronta una gran variedad de retos, entre ellos comunicación, coordinación, asignación de tareas, mapeo, localización y muchos otros. Pero las soluciones planteadas

son bastas y de diversas fuentes, como el enfoque de inspiración biológica que busca estudiar e imitar las sociedades naturales, como hormigas, termitas o abejas, para brindar soluciones. Por otro lado, el acercamiento de comportamientos emergentes que brinda a los robots las mejores capacidades posibles de comunicación para llevar a cabo sus tareas utilizando herramientas de sistemas de control y teoría de grafos.

Basado en la idea de los SRM, este proyecto aborda el estudio de estrategias de coordinación de movimiento bajo el enfoque de comportamientos emergentes para sistemas robóticos multi-agentes heterogéneos (SRMH). Cada robot tiene un conocimiento parcial acerca de la posición y orientación de ciertos miembros del grupo y capacidades físicas distintas, por lo tanto las estrategias de control son descentralizadas y el objetivo es obtener un comportamiento global deseado a través de simples interacciones locales. Este proyecto se enfoca en dos casos de locomoción generales: terrestre y aérea. Las estrategias de coordinación se basan en modelos matemáticos de ecuaciones diferenciales y control no lineal donde se analiza la convergencia a las posiciones o distancias deseadas, la evasión de colisiones y el seguimiento grupal de trayectoria. El trabajo experimental se realizó en una plataforma heterogénea y flexible, integrada por módulos de control y actuadores, diseñada para la implementación de los algoritmos estudiados en esta tesis.

## 1.1. Planteamiento general del problema

Considere un sistema de robots multiagente (SRM) compuesto por agentes heterogéneos, SRMH, que difieren en:

- El tipo de motricidad, terrestre o aérea.
- Modelo cinemático o dinámico de postura.

El objetivo de coordinación de movimiento consiste en construir arquitecturas de control que satisfagan:

- Convergencia a patrones de formación
- Evasión de colisiones entre robots.

La coordinación de movimiento estará sujeta a las siguientes restricciones físicas y de control:

- Información de posición obtenida a través de un sistema de posicionamiento espacial.
- Espacio de trabajo limitado y conocido.
- Monitoreo en computadora remota para la observación del comportamiento grupal.

La plataforma experimental propuesta de SRMH debe satisfacer las siguientes condiciones:

- Contener un núcleo básico de control que incluya los elementos fundamentales de control de un robot móvil, no importando su motricidad.

- Módulos intercambiables de actuadores y su etapa de potencia que provean diferente movilidad.
- Sistema de comunicación similar o compatible que permita el intercambio transparente de información entre robots.

El presente proyecto se limita a estudiar agentes móviles terrestres y aéreos, debido a su posible similitud para construir un sólo núcleo de control compatible. Específicamente abordan las siguientes configuraciones básicas:

- Robots del tipo unicycle.
- Robots móviles omnidireccionales con ruedas tipo Mecanum.
- Robots aéreos de ala rotatoria.

En la siguiente sección se plantean los objetivos de la tesis sujetos a las restricciones antes mencionadas.

## **1.2. Objetivos**

### **1.2.1. Generales**

- Desarrollar estrategias de coordinación de movimientos en SRMH, usando herramientas analíticas de sistemas dinámicos y control para coordinar movimientos de robots terrestres y aéreos.
- Diseñar una plataforma experimental flexible de SRMH capaz de incluir robots terrestres y aéreos para validar las estrategias de control.

### **1.2.2. Específicos**

- OP1 Análisis matemático de la cinemática y dinámica de los agentes móviles (omnidireccionales, unicycles y cuadricópteros).
- OP2 Extensión de estrategias de formación y evasión de colisiones del caso homogéneo al caso heterogéneo.
- OP3 Análisis matemático de arquitecturas de control para la coordinación de movimiento y su adecuación con el control individual de los agentes de acuerdo a su motricidad.
- OP4 Diseño mecánico, selección de elementos actuadores, y sistema electrónico de control de la plataforma experimental.
- OP5 Construcción de los agentes y los diferentes sistemas de motricidad.
- OP6 Pruebas de la plataforma para validar la estrategia de control

### 1.3. Justificación

En la actualidad existe una gran explosión mediática con respecto a los drones aéreos usados en fotografía, eventos deportivos y con fines bélicos. La atención de las industrias, corporaciones gubernamentales, asociaciones civiles y público en general crece de forma sostenida y se enfoca en otros tipos de robots y avances en automatización.

En la medida en que el mercado comienza a encontrar áreas de oportunidad donde los robots representan una ventaja económica o de seguridad, se proponen diferentes soluciones a esos problemas específicos. Esto ofrece un panorama favorecedor para los científicos e ingenieros que se encargan de diseñar y construir toda clase de robots.

Conforme aumenten las iteraciones de solución de problemas específicos se optimizarán los robots desarrollados. Para cada problema el robot que lo solucione de manera óptima será un robot muy específico. Sin embargo para realizar tareas conjuntas y resolver problemas distintos de forma simultánea se requerirán cualquiera de dos opciones; un robot de propósito general, o un grupo de robots de propósito específico trabajando de forma conjunta.

El primer acercamiento presenta el problema del costo. Un robot de alta complejidad tendrá un costo más elevado. Adicionalmente, tendrá poca tolerancia a fallas, por lo que si sufre una avería o algún impedimento de cualquier índole, no podrá continuar sus tareas.

El segundo acercamiento exhibe la ventaja del costo. Varios sistemas sencillos tendrán un costo menor al sistema complejo. Adicionalmente, el sistema podría soportar fallas de varios de sus miembros sin perder efectividad en las tareas a desarrollar. Entonces, al tener un grupo de robots distintos, especializados en tareas específicas, se puede lograr interacciones de sistemas más complejos de forma más eficiente.

Un ejemplo de esto son los vehículos automatizados, como aquellos diseñados por Google, Tesla, Ford y muchas otras empresas, que tienen la finalidad de resolver el complejo problema de movilidad de millones de personas, aumentando la seguridad de los usuarios. Con el crecimiento de estas tecnologías, se buscará tener sistemas aislados de menor costo, vehículos accesibles a los compradores, y heterogéneos, autos, autobuses, camiones etcétera, que brinden seguridad y transporte eficiente a millones de personas.

De forma evidente la implementación de sistemas de robots heterogéneos requiere aún mucho desarrollo e investigación. En cuanto al estudio de los SRM heterogéneos, se puede afirmar que no existen muchas plataformas de experimentación adecuadas para su estudio. Este hecho impide el estudio de los sistemas heterogéneos. En la sección 2.1.3 se expande la investigación de las plataformas.

Para el desempeño adecuado de los robots en los ambientes humanos, estos deben tener conocimiento de su entorno. Una variable básica es la ubicación espacial del robot. Para conocer la ubicación espacial, el acercamiento clásico, Euclidiano, requiere el establecimiento de un marco de referencia con tres planos mutuamente ortogonales y origen ubicado en la intersección de los tres planos. Con este acercamiento se logra ubicar en el espacio cualquier cuerpo usando tres magnitudes, usualmente  $(x, y, z)$  y tres orientaciones angulares  $(\phi, \theta, \psi)$ , denominadas ángulos de Euler.

El principal problema con este acercamiento es el seleccionar en todo ambiente el origen y medir las seis variables principales con respecto a este marco de coordenadas. Adicionalmente la abstracción Euclidiana de la ubicación espacial no es natural al humano, por lo que la interacción simultánea de los robots y los humanos presenta dificultades.

Una alternativa al acercamiento clásico es el uso de coordenadas polares en dos dimensiones y su extensión a tres dimensiones en coordenadas cilíndricas. En este acercamiento se usa únicamente un eje de referencia, desde el cual se describe la ubicación de un cuerpo con el vector que describe la magnitud con respecto al eje de referencia y la orientación angular con respecto a este mismo. Esta forma de ubicación puede hacer uso de la distancia como magnitud referenciada a un eje de referencia arbitrario. La estimación de distancia entre objetos y el ambiente es la forma más normal de ubicación espacial para el humano.

## 1.4. Contribución

En esta tesis se plantea una metodología de coordinación de movimiento basado en distancias y orientación, aplicable a grupos heterogéneos de robots. La contribución teórica del uso de distancia y orientación permite la implementación más portátil en sistemas físicos, ya que no se requiere de un sistema de posicionamiento global.

El uso de esta estrategia de control en robots heterogéneos abre posibilidades para la resolución de problemas complejos, en los que varios robots diferentes lleven a cabo tareas específicas que contribuyan a la resolución del problema.

La plataforma diseñada presenta características adecuadas para el control de los robots propuestos, y la expansión a otros modelos de robots. El núcleo de control permite programar diferentes robots en una sola plataforma con la misma comunicación y protocolos, diferenciados por sus actuadores, modelos matemáticos y algoritmos de control.

De esta forma se plantea una posible solución al problema del posicionamiento espacial por medio de la representación polar, usando la distancia como variable de interés. Se aplican estas estrategias a SRM heterogéneos, lo que brinda variedad y flexibilidad en los SRM. Adicionalmente, la eficacia de la plataforma construida, permite probar cualquier algoritmo planeado para SRM homogéneos y heterogéneos.

## 1.5. Estructura

En el capítulo dos de este trabajo se presenta un estado del arte de los SRM y SRMH, así como el marco teórico necesario para el desarrollo de esta tesis. En el tercer capítulo se expone el principal resultado teórico de este documento, formación basada en distancia y orientación, y en el capítulo cuatro se extiende el resultado al caso heterogéneo. El quinto capítulo muestra el desarrollo de la plataforma experimental y en el sexto capítulo se plantean conclusiones y trabajo futuro. Se incluyen en los anexos características de la plataforma de robots desarrollada. En los anexos A, B y C se describen diagramas de ensamble estructural, códigos del núcleo de control y archivos de la interfaz con MatLab, respectivamente.



# Capítulo 2

## Antecedentes

### 2.1. Los Sistemas de Robots Multiagentes

Los Sistemas de Robots Multiagentes se definen como un grupo de agentes autónomos coordinados por un sistema de comunicación para realizar una tarea de forma cooperativa [93, 44]. El área de los SRM extiende el estudio de un solo robot a un número mayor de robots interrelacionados, generando gran cantidad de problemas de estudio. Por ejemplo, la descomposición de tareas complejas, asignación de tareas para los agentes, interacción de los robots en ambientes conocidos, desconocidos o con obstáculos y evitando colisiones entre sí mismos.

Los SRM han encontrado un amplio campo de aplicaciones en exploración terrestre, oceánica y espacial debido a sus características de paralelismo, robustez, escalabilidad, homogeneidad o heterogeneidad, flexibilidad y alta complejidad de tareas que pueden resolver [93, 16]. Además diseñar y construir un grupo de robots con menor grado de complejidad puede ser más fácil, barato, flexible y con mayor tolerancia a fallas [92]. Algunas aplicaciones que se estudian en la actualidad van dirigidas hacia la navegación autónoma, reconocimiento de espacios de trabajo, monitoreo y vigilancia, sustitución de personal en ambientes peligrosos, robots jugadores de soccer, entre muchas más [65, 66, 16] surgiendo como una nueva área de estudio para el sector de ingeniería.

Desde el final de la década de 1980, se han planteado diversas áreas de análisis y múltiples retos para los SRM, y aunque el estado del arte ha aumentado considerablemente no se puede afirmar que un área o tema se encuentre ampliamente desarrollado [93, 73, 5]. De las distintas áreas de interés mencionadas en [73, 5, 16], se pueden identificar tres principales: coordinación de movimiento, asignación de tareas y comunicación.

La comunicación es uno de los aspectos más importantes de la investigación de sistemas multiagentes, se buscan estrategias para la optimización de la interacción de los agentes desarrollando redes de sensores, protocolos de comunicación o canales de comunicación dependientes del tipo de sensor, infrarrojo, de proximidad, de radiofrecuencia, sistemas de posicionamiento global o visión por computadora. La comunicación puede ser implícita y explícita [63, 79]. La comunicación implícita es un método que transmite variables de información expresadas a través del ambiente y la comunicación explícita se da cuando los robots pasan directamente información entre ellos [90].

En la asignación de tareas se busca distribuir las tareas entre los robots de forma

eficiente, de forma que no solo la tarea principal se lleve a cabo, si no que las tareas sean bien distribuidas entre los agentes [9, 37]. En la asignación de tareas se estudian aspectos como la delegación de autoridad y control, tolerancia a fallas, modelado de capacidades de robots heterogéneos, resolución de conflictos, etcétera.

La coordinación de movimientos aborda el control de tráfico, consenso, control de formación y el avance de formación o marcha [18], siendo el objetivo principal que un robot mantenga su posición relativa con respecto a ciertos miembros del grupo, evitando colisiones y siguiendo ciertas trayectorias. El planteamiento se reduce a un problema geométrico en configuración en el espacio con un tiempo determinado.

Estas tres líneas de investigación se redireccionan de acuerdo a dos grandes enfoques de coordinación. La primera relacionada con la inspiración biológica o inteligencia de enjambre [93, 50, 82], que analiza y reproduce el comportamiento de sociedades cooperativas biológicas como abejas y hormigas en controles basados en reglas locales simples, considerando agentes homogéneos. El otro enfoque es conocido como de comportamientos emergentes, e incluye diferentes estrategias basadas en sistemas dinámicos y control, gráficas de comunicación y asignación de roles de grupo. Bajo este enfoque se han estudiado:

- Coordinación a través de la asignación de líderes de grupo [19, 46].
- Mapeo y localización para mapear el ambiente físico usando los sensores de los robots para construir modelos espaciales y ubicar al robot en el modelo espacial generado [88].
- Transporte y manipulación de objetos, sujetándolos, empujándolos o rodeándolos [91, 28, 84] .
- Robots reconfigurables modulares que son robots autónomos con morfología variable [93].
- Aprendizaje con algoritmos supervisados y sin supervisión para la toma de decisiones [2, 76].

De acuerdo a [5], los problemas fundamentales de coordinación de movimiento son el control de formación, control de avance en formación o marcha y evasión de obstáculos. El control de formación y marcha es una parte indispensable de los SRM, ya que la gran mayoría de sus aplicaciones requiere de control en la coordinación de movimiento de los robots en el área de trabajo para desplazarse y posicionarse de forma estratégica para el trabajo cooperativo. Por ejemplo, en tareas de exploración, vigilancia, detección de perímetros, dispersión y búsqueda se trata de desplegar a los robots de forma que maximicen el área que abarca cada robot, realizando patrones de movimiento y posicionamiento para evitar colisiones. Además la formación puede cambiar si algún robot falla en su tarea [10, 92]. En el caso de manipulación de objetos y manejo de materiales se busca que los robots lleguen a la posición deseada para sujetar el objeto y trasladarlo posiblemente en forma grupal [5, 6, 16].

Los requerimientos que se busca satisfacer con una ley de control para formación y marcha son: convergencia a la formación deseada desde cualquier posición inicial de los agentes, evasión de colisiones entre agentes, evasión de obstáculos y convergencia grupal a la trayectoria de marcha [18]. Algunos de los esquemas en la literatura incluyen desde los basados en comportamientos de enjambre [10], hasta

modelos cinemáticos y dinámicos enlazados a través de gráficas de comunicación [60] y movidos por campos vectoriales. Otros estudios apenas comienza a incorporar estrategias no convencionales como algoritmos genéticos [18] .

La literatura de control de formación, marcha y evasión de colisiones es vasta. Sin embargo, es importante notar que las estrategias han sido desarrolladas para agentes homogéneos, sin considerar su posible comunicación con agentes de otra morfología y capacidades. Haciendo revisión de cada caso homogéneo, se pueden identificar estudios sobre coordinación de:

- Robots puntuales u omnidireccionales [52, 53].
- Robots con ruedas del tipo unicycle [26, 81].
- Robots tipo carro [15, 27].
- Cuadricópteros [12].
- Submarinos [39].
- Robots con patas [35, 1].

### **2.1.1. Coordinación de movimiento de robots móviles heterogéneos**

Los SRMH son aquellos donde los agentes difieren en características físicas y capacidades [45, 50]. Las diferencias a nivel físico mecánico pueden ser de locomoción, actuadores, sensores, tamaños, etc. En cuanto a la electrónica y programación pueden variar por los componentes electrónicos y sus leyes de control, algoritmos de comportamiento, entre otros [38, 29, 74]. La ventaja de utilizar un SRMH es precisamente el aprovechamiento de esas diferencias, para la realización de tareas complejas. Por ejemplo, robots con diferentes sistemas de sujeción, permiten cargar diferentes tipos de piezas de trabajo. El sensado por agentes aéreos, combinado con robots terrestres, proporcionaría un mapeo más completo de un espacio de trabajo desconocido. En [75] se muestra una simulación con robots homogéneos con diferencias de medición que los hace heterogéneos.

El caso de robots heterogéneos es una de las metas de ingeniería de coordinación de los SRM. La idea de coordinar diferentes entes robóticos en espacio abierto, es el reto que se ha planteado el llamado SoS (System of Systems en inglés) surgido en Estados Unidos [8, 86, 49]. El SoS se encuentra en desarrollo en diversas áreas, principalmente la militar [67], donde se busca una coalición de sistemas independientes (satélites, aeronaves, vehículos terrestres, submarinos y barcos) operados y coordinados para funcionar como unidad autónoma de forma eficiente y efectiva para tácticas de ataque y monitoreo de campos abiertos [47, 13]. El SoS se compone de sistemas independientes amigables al usuario, cada uno con diferentes plataformas de control y comunicación, tiempos de operación arbitrarios y sin relación para el desarrollo de cada robot y el acoplamiento interdependiente entre sistemas. Note que la optimización de cada sistema no garantiza la optimización global del SoS, y la operación combinada de los sistemas constituye el cumplimiento del objetivo de todo el SoS. La presente tesis se inspira en el SoS, tratando de abordar desde el punto de vista académico la problemática de los SRMH, planteando posibles soluciones para el SoS.

Para llevar a cabo una tarea, los SRMH necesitan interactuar de forma física y en comportamiento colectivo, por lo que la integración de diferentes plataformas de hardware resulta en alta complejidad para el diseño y la implementación. Para que los agentes cooperen, se requiere hardware de comunicación compatible entre agentes y sensores con capacidad de reconocimiento mutuo. El diseño no plantea un reto trivial, por ejemplo un dispositivo de comunicación deberá ser ligero y energéticamente eficiente para un robot volador, a diferencia de un dispositivo de comunicación para un robot terrestre que podría ser más pesado y con mayor capacidad de alcance [38, 85, 54, 29, 74].

El control comportacional en todo SRMH es complicado. Para cada agente deben ser diseñadas reglas individuales de control buscando un comportamiento colectivo general para todos los agentes. La complejidad radica en el control básico de cada agente para sus sensores, actuadores y sistemas de comunicación y el control superior que regula la dinámica de interacción grupal. Con agentes heterogéneos, la dificultad aumenta ya que se diseñan agentes con mayor grado de especialización buscando eficiencia en el sistema. Estos robots diferentes pueden tener sensores y actuadores adicionales que le permiten trabajar en otras acciones o configuraciones grupales. El estudio de robots heterogéneos debe incorporar al mismo tiempo la asignación de tareas de acuerdo a capacidades y la coordinación de movimientos para asegurar que el robot adecuado para la tarea se posicione en al área adecuada para realizarla eficazmente.

### 2.1.2. Control de formación basado en posiciones y en distancias

La coordinación de movimiento en SRM involucra estrategias de control basadas en posiciones relativas para cada robot respecto a los otros miembros del equipo [5].

En control de formación, el movimiento de los robots depende de la información disponible sobre la posición, desplazamiento o distancia con respecto a otros robots [70]. El problema tradicional de formación es una extensión del llamado *problema de consenso* estudiado ampliamente en [77], donde los robots son modelados como integradores simples con una topología de comunicación fija, descrita por un grafo de comunicación [21]. Entonces, las leyes de control se diseñan para converger a patrones geométricos predefinidos, evitando colisiones entre robots [18]. Las propiedades del grafo de formación, incluida la matriz Laplaciana, son usadas para estudiar la convergencia global, estabilidad grupal, seguimiento de trayectoria grupal, entre otros.

En el algoritmo tradicional de formación basada en consenso, las leyes de control usan gradientes negativos de funciones potenciales atractivas y repulsivas (F-PAR) para generar campos vectoriales que garantizan la convergencia y la evasión de colisiones, respectivamente [77, 44]. Las entradas de control de cada robot se vuelven una función de la posición de aquellos robots que puede detectar. por lo que se requiere el uso de sensores de posición con respecto a un marco de referencia global. La convergencia con grafos no dirigidos se ha estudiado en [24] y para grafos definidos arbitrariamente en [41]. Adicionalmente funciones de navegación surgidos de la combinación atractiva y repulsiva de campos vectoriales para evitar mínimos locales han sido propuestas en [26] y se pueden extender a los casos de robots con restricciones no holonómicas como se muestra en [44, 11] usando linealización entrada-salida

para robots de tipo unicycle. Muchas estrategias de formación han sido usadas con robots móviles en tareas de seguridad y vigilancia, exploración y mapeo, simulación de agentes biológicos, detección de perímetros, etcétera [10, 5].

El control basado en posiciones se estudia en [43] en tiempo discreto para integrador simple e integrador doble en [40] con periodo constante de muestreo. En algunos trabajos, la comunicación entre robots puede aparecer o desaparecer dependiendo de la proximidad de otros robots, por lo que el grafo de comunicación se vuelve variable. La formación basada en robots vecinos se estudia en trabajos como [48], en tiempo discreto y continuo. Control híbrido de formación con comportamiento de parvada se muestra en [95], donde un protocolo elige la creación y destrucción de lazos de comunicación entre robots, un controlador de movimiento regula las distancias entre agentes y las velocidades de los robots modelados como doble integrador usando el gradiente negativo de una función potencial artificial.

Para implementar el algoritmo tradicional de formación basada en consenso evitando el uso de sistemas de posicionamiento global, algunos trabajos incorporan sensores locales de desplazamiento relativo, como medidores de distancia láser, en los robots y diseñan observadores para estimar sus posiciones, como en [4]. Si se considera que el centroide de los robots se mueve a lo largo de una trayectoria, las leyes de control requieren la información de la trayectoria del centroide y el intercambio preciso inalámbrico de la posición exacta de los otros robots. Un trabajo similar se presenta en [83], donde un método de odometría funciona como estimador de posición y orientación. La ley de control es proporcional con respecto a los errores de posición de todos los robots, requiriéndose el intercambio de comunicación usando una red TCP/IP. El retraso de las comunicaciones se toma en cuenta en la ley de control para probar su convergencia.

Recientemente, las metodologías de formación basada en distancias parecen ser la alternativa a las basadas en posición. El objetivo es que los robots convergan a las distancias deseadas definidas entre pares de robots. Por ejemplo, en [34, 94] algoritmos reactivos inspirados en comportamiento biológico se implementan para formar a los robots con respecto a sus vecinos cercanos, sin especificar la posición en un patrón de formación. En los esquemas basados en grafos de formación, algunas funciones atractivas y repulsivas en función de distancias se examinan en [24] analizado para el caso de grafos no dirigidos. Esquemas de seguimiento de líder se estudian en [3] para el caso de robots de doble integrador.

Note que el acercamiento de formación basada en distancias puede generar configuraciones distintas que satisfacen las restricciones de distancia. Si se requiere una formación específica, el grafo de comunicación deberá ser rígido como se muestra en [71, 57], donde al menos  $(n - 3)$  aristas de comunicación para  $n$  robots deberán ser definidas. Finalmente, las leyes de control se obtienen usando matrices de distancia euclidiana, en lugar de funciones potenciales para un caso general en [68].

Otros trabajos como [22, 23] proponen esquemas líder-seguidor para robots tipo unicycle usando la distancia y orientación del líder, lo que implementa una topología especial de comunicación. Control híbrido basado en distancia y orientación se usa en [89] para disminuir el error durante ciertos movimientos del líder. Se estudian en [33] aproximaciones de la velocidad del líder para mantener estable la formación durante el seguimiento de trayectoria, por medio de estimación óptima de la distancia medida por sensores locales o métodos adaptivos en formaciones cuyo grafo tiene un árbol de expansión como en [51]. También, se estima la distancia usando cámaras

en [32] incluyendo los efectos de cuantificación y retardos en la medición. Finalmente, estimadores locales para medidas de distancia inconsistentes y sus efectos en formaciones rígidas se analizan en [36].

Es necesario notar que, en todos los trabajos mencionados anteriormente, aunque las funciones potenciales dependen explícitamente de la distancia entre robots, las leyes de control se obtienen como el gradiente negativo de funciones potenciales específicas, por lo que los robots requieren saber la posición absoluta de algunos robots, lo cual reduce la descentralización para esta estrategia de control.

### 2.1.3. Desarrollo de plataformas de robots heterogéneos

A la fecha muy pocas plataformas de SRMH han sido creadas con el propósito de estudios sobre robots heterogéneos [50, 29]. La gran mayoría de los desarrollos de robot heterogéneos se realiza únicamente en simulaciones como en [75, 58, 64, 45, 54]. En muchas ocasiones la heterogeneidad se estudia únicamente para aprovechar diferentes robots disponibles en laboratorios y probar algún algoritmo de control [56, 85, 55, 74].

La plataforma de SRMH más interesante creada hasta el momento es "Swarmanoid", dirigida por Marco Dorigo en el laboratorio IRIDIA de la Université Libre de Bruxelles [29]. La plataforma está formada por tres tipos de robots que cooperan en la misión de localizar un libro en un estante y transportarlo a un lugar predeterminado. La plataforma tiene las desventajas de que las capacidades de los tipos de robots les impiden ser usados en otro contexto y hay poca oportunidad de adaptarlos. Adicionalmente, el estudio del comportamiento es únicamente desde un punto de vista biológico.

Una aplicación concreta que demuestra la potencialidad de los SRMH es la mostrada en [84] en que se presenta el uso de tres agentes, uno aéreo, otro marino y el último submarino, interactuando con especialistas humanos para realizar monitoreo de arrecifes de coral.

## 2.2. Fundamentos teóricos

En esta sección se presentan los fundamentos conceptuales y matemáticos necesarios para el desarrollo de esta tesis. Primero se abordan generalidades de los SRM, enseguida los fundamentos de teoría de grafos como elementos descriptivos de topologías de formación y comunicación. A continuación se incluye la descripción de los problemas básicos de consenso y su planteamiento matemático en los casos posicionales y de distancias. Para finalizar se incluye la descripción de los modelos matemáticos de los robots. Los conceptos y derivaciones matemáticas han sido adecuados de varios artículos y libros ya citados, en especial de [77] y [70], por lo que no se hará mención de las fuentes de información durante el marco teórico, a no ser que se trate de una referencia nueva.

### 2.2.1. Los sistemas de robots multiagentes

Un Sistema de robots multiagente es un sistema de dos o más agentes robóticos, físicos o virtuales que realizan tareas interactuando entre ellos y su ambiente,

compartiendo información de su estado o el estado del ambiente. Esto deviene en el axioma básico del consenso:

**Axioma.** La información compartida es una condición necesaria para la cooperación.

Cumplíéndose el axioma anterior, el consenso es un acuerdo o conformidad en una variable, producido por negociación y consentimiento, por parte de los miembros de un grupo de robots.

Para lograr consenso debe existir una variable de interés o “estado de información” y procesos adecuados para negociar el estado de esa variable y lograr acuerdo con respecto a los valores de esa variable, denominados “algoritmos de consenso”. La variable de interés puede ser:

- Forma o figura de la formación.
- Tiempo de encuentro.
- Longitud del perímetro monitoreado.
- Distancia entre agentes.
- Dirección del movimiento.

Los algoritmos de consenso son diseñados para ser distribuidos, es decir se evita la centralización para reducir el uso de recursos, su escalabilidad a mayor cantidad de robots y aumentar la tolerancia a fallas del sistema completo.

### 2.2.2. Teoría de grafos

Para el estudio de los SRM la herramienta matemática mas útil es la teoría de grafos. La teoría de grafos es el campo de las matemáticas que estudia las propiedades e interacciones de los grafos, los grafos  $\mathcal{G}$  son las estructuras conformadas por el par nodo–arista  $(\mathcal{V}_n, \mathcal{E}_n)$ .

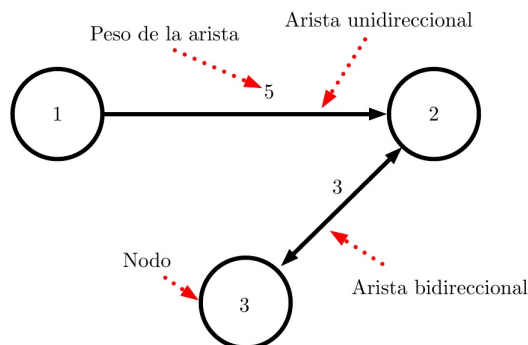


Figura 2.1: Grafo detallado

La ventaja matemática de los grafos es su representación en forma matricial. Por ejemplo la figura 2.1 se puede representar por medio de la “matriz de adyacencia” binaria:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (2.1)$$

Se asigna de forma binaria uno si es que el nodo se conecta o recibe información del nodo adyacente, cero en la situación contraria.

Los grafos pueden tener muchas propiedades y características, algunas pertinentes para el estudio de los SRM que son:

- Un grafo dirigido es aquel en que las aristas tienen un sentido de orientación.
- Un grafo completo es aquel en que cada nodo se encuentra unido al resto de los nodos.
- Un grafo con árbol de expansión directo es aquel en que un nodo de inicio o raíz, puede encontrar una ruta de nodos conexos por aristas dirigidas para llegar a cualquier nodo del grafo.
- Un grafo puede ser conexo si no existen nodos aislados o fuertemente conexo si para cada par de nodos existe unión bidireccional.
- Un grafo con pesos es aquel que tiene asociado a cada arista un peso.
- Un grafo puede ser no rígido, rígido o globalmente rígido como se demuestra en [7].

En el contexto de SRM ideales dos características de los grafos que se excluyen son las auto aristas y los nodos aislados mostrados en la figura 2.2, por representar comunicación interna y falta de comunicación en un robot respectivamente.

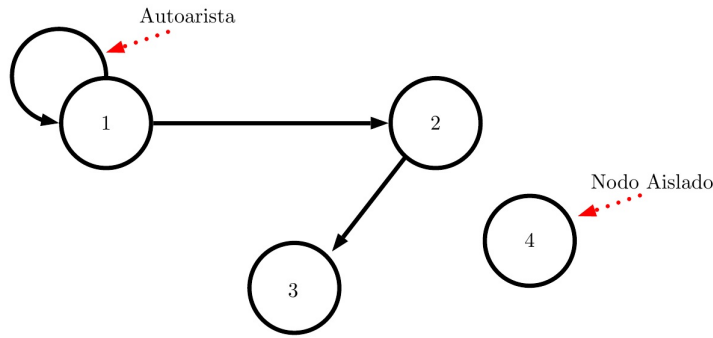


Figura 2.2: Grafo con autoaristas y nodos aislados

En un SRM con  $n$  vehículos un grafo dirigido  $\mathcal{G}$  es el par  $(\mathcal{V}_n, \mathcal{E}_n)$  donde  $\mathcal{V}_n = \{1, \dots, n\}$  es el conjunto de nodos finitos y no vacío.  $\mathcal{E}_n \subseteq \mathcal{V}_n \times \mathcal{V}_n$  es el conjunto de aristas definidas por el par de nodos  $(i, j)$ . La arista  $(i, j)$  denota, en un grafo dirigido, que el vehículo  $j$  está conectado o recibe información del vehículo  $i$ . Como se denota anteriormente los pares de autoaristas  $(i, i)$  o  $(j, j)$  no se consideran. Adicional a la condición de exclusión de nodos aislados se requiere la condición de árbol de expansión directo. La matriz de adyacencia  $A_n = [a_{ij}] \in \mathbb{R}^{n \times n}$  de un grafo dirigido  $\mathcal{G} = (\mathcal{V}_n, \mathcal{E}_n)$  donde  $a_{ij}$  es el peso de la arista  $(i, j) \in \mathcal{E}_n$  se define como:

$$A_n = [a_{ij}] = \begin{cases} a_{ij}, & \text{si } (i, j) \in \mathcal{E}_n \\ 0, & \text{si es de otra forma} \end{cases} \quad (2.2)$$



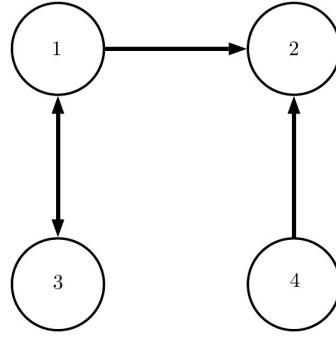


Figura 2.3: Grafo de conectividad A

Considere el grafo del ejemplo uno mostrado en la figura 2.3, la matriz de adyacencia será:

$$A_n = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.3)$$

Una característica clave para el consenso basado en distancias es la rigidez de los grafos. Para un grafo bidimensional  $\mathcal{G} = (\mathcal{V}_n, \mathcal{E}_n)$ , se define  $l$  como el número de aristas. Entonces la condición para garantizar rigidez se establece si se cumple que  $l \geq 2n - 3$ .

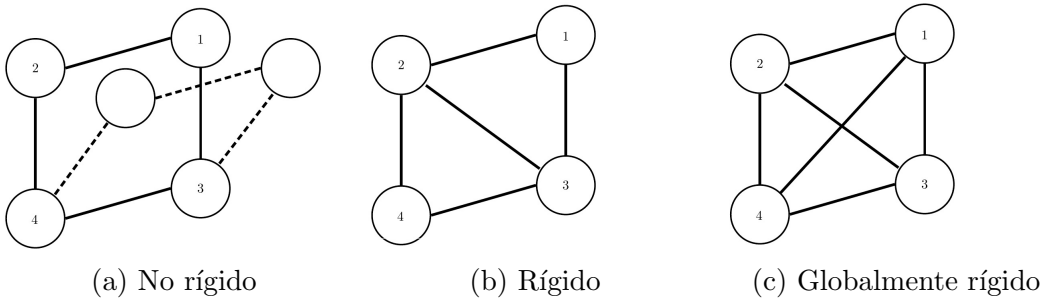


Figura 2.4: Rigidez en Grafos

En la figura 2.4 se muestran tres grafos, el de la figura 2.4a no es rígido por que varias configuraciones cumplen las distancias deseadas, es decir para  $n = 4, l = 4$  se tiene que  $l < 2n - 3$ . Para el grafo de la figura 2.4b se comprueba que la función es rígida, para  $n = 4, l = 5$  se obtiene que  $l = 2n - 3$ , esta configuración sin términos repulsivos podría encontrar una solución sobreponiendo, colisionando, los agentes 1 y 4. El grafo de la figura 2.4c es globalmente rígido ya que para  $n = 4, l = 6$  se obtiene que  $l > 2n - 3$ , esta configuración no permite ninguna flexibilidad.

### Matriz Laplaciana

La Matriz Laplaciana presenta muchas características topológicas de un grafo, se define de la siguiente manera:

$$\mathcal{L}_n = \Delta - A_n \quad (2.4)$$

considerando a  $A_n$  la matriz de adyacencia y definiendo  $\Delta$  como:

$$\Delta = [d_{ij}] \in \mathfrak{R}^{n \times n} \begin{cases} \sum_{i=1, j \neq i}^n a_{ij}, i = 1, \dots, n \\ 0, \text{ si } (i \neq j) \end{cases} \quad (2.5)$$

se nota que  $\Delta$  es básicamente la matriz diagonal de cardinalidad de conexiones. Para el grafo del ejemplo uno mostrado en la figura 2.3 se presenta la matriz Laplaciana:

$$\begin{aligned} \mathcal{L}_n &= \Delta - A_n \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (2.6)$$

Para un grafo no dirigido la matriz  $\mathcal{L}_n$  es simétrica, y para un grafo dirigido  $\mathcal{L}_n$  no es simétrica por necesidad, por lo que en este caso suele llamarse matriz Laplaciana no simétrica o matriz Laplaciana dirigida. En ambos casos la suma de las filas en la matriz  $\mathcal{L}_n$  es cero, por lo que 0 es un eigenvalor de  $\mathcal{L}_n$  asociado al eigenvector unitario, el vector columna  $\mathbf{1}_n$  de unos. Note que  $\mathcal{L}_n$  es diagonalmente dominante y no tiene elementos negativos en la diagonal. Usando el teorema de los discos de Gershgorin, referenciado en [59], sabemos que para un grafo no dirigido todos los eigenvalores diferentes de cero en  $\mathcal{L}_n$  son positivos ( $\mathcal{L}_n$  es semidefinida positiva), mientras que para un grafo dirigido todos los eigenvalores diferentes de cero de  $\mathcal{L}_n$  tienen partes reales positivas. Entonces, todos los eigenvalores de  $-\mathcal{L}_n$  tienen partes reales negativas. Para un grafo no dirigido, el valor de 0 es un eigenvalor simple de  $\mathcal{L}_n$  si y solo si el grafo es conexo. Para un grafo dirigido, el valor de 0 es un eigenvalor simple de  $\mathcal{L}_n$  si y solo si el grafo es fuertemente conexo.

En las figuras 2.5, 2.6, 2.7 y 2.8 se muestran cuatro tipos distintos de topología, el grafo completo bidireccional, el de persecución cíclica, el de persecución cíclica bidireccional y el de líder bidireccional, cuyas respectivas matrices Laplacianas generalizadas se muestran en las ecuaciones (2.7) a (2.10).

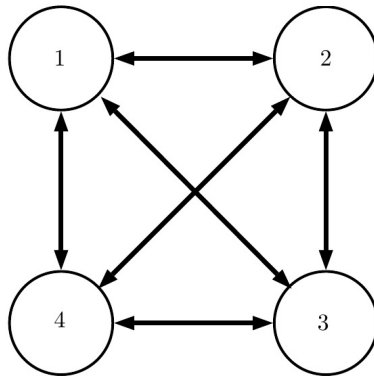


Figura 2.5: Grafo completo y bidireccional

$$\mathcal{L}_n = \begin{bmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & & \cdots & -1 \\ \vdots & & \ddots & & \vdots \\ -1 & -1 & -1 & \cdots & n-1 \end{bmatrix} \quad (2.7)$$

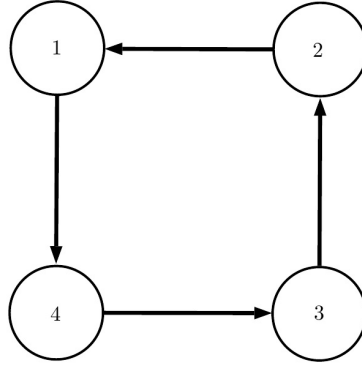


Figura 2.6: Grafo de persecución cíclica

$$\mathcal{L}_n = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \\ -1 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \quad (2.8)$$

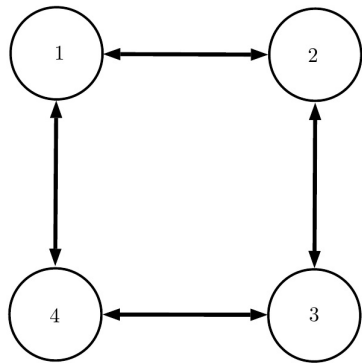


Figura 2.7: Grafo de persecución cíclica bidireccional

$$\mathcal{L}_n = \begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & -1 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & 0 & \cdots & -1 & 2 & -1 \\ -1 & 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \quad (2.9)$$

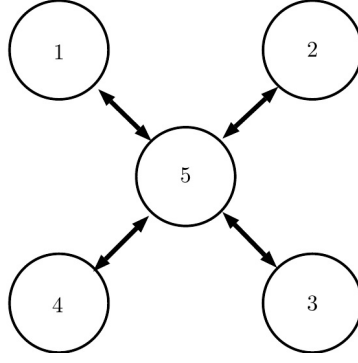


Figura 2.8: Grafo de líder bidireccional

$$\mathcal{L}_n = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & -1 \\ 0 & 1 & 0 & 0 & \cdots & -1 \\ 0 & 0 & 1 & 0 & & -1 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \\ -1 & -1 & -1 & \cdots & -1 & n-1 \end{bmatrix} \quad (2.10)$$

### 2.2.3. Algoritmo de consenso de primer orden

La idea básica para el algoritmo de consenso es la de imponer dinámicas similares en los estados de cada robot. Suponga  $N = \{R_1, \dots, R_n\}$  con  $n$  robots de primer orden con modelo cinemático de un solo integrador, representado por

$$\dot{\vec{z}}_i = u_i, \quad i = 1, \dots, n \quad (2.11)$$

que se mueven en el plano con posiciones  $\vec{z}_i(t) = [x_i(t), y_i(t)]^T$ , cuya comunicación puede ser modelada por medio del grafo  $\mathcal{G} = (\mathcal{V}_n, \mathcal{E}_n)$  definido en la sección 2.2.2. El algoritmo de consenso general está dado por

$$u_i(t) = - \sum_{j=1}^n a_{ij} [\vec{z}_i(t) - \vec{z}_j(t)], \quad i = 1, \dots, n \quad (2.12)$$

donde  $a_{ij}$  es el elemento  $(i, j)$  de la matriz de adyacencia  $A_n$  definida en la sección 2.2.2.

La ecuación (2.12) se puede escribir de forma matricial de la siguiente manera

$$\dot{\vec{z}}(t) = -\mathcal{L}(t)\vec{z}(t) \quad (2.13)$$

en las ecuaciones (2.12) y (2.13),  $\vec{z} = [\vec{z}_1, \dots, \vec{z}_n]^T$  es el estado de información, en este caso posiciones planares, y  $\mathcal{L}(t)$  es la matriz Laplaciana no simétrica asociada al grafo  $\mathcal{G} = (\mathcal{V}_n, \mathcal{E}_n)$  definido en la sección 2.2.2.

Para las ecuaciones (2.12) y (2.13), se alcanza consenso cuando para todos  $\vec{z}_i(0)$  y para todo  $i, j = 1, \dots, n$ ,  $|\vec{z}_i(t) - \vec{z}_j(t)| \rightarrow 0$ , cuando  $t \rightarrow \infty$ .

Note que la ecuación (2.12) y su forma matricial (2.13) garantizan que los estados de información de los robots sean los mismos, pero no establece un valor específico común. Los algoritmos mostrados en las secciones 2.2.5 y 2.2.6 que corresponden a dos problemas básicos de consenso; formación y formación basada en distancias,

se basan en este algoritmo básico pero especifican de forma más precisa los valores deseados en posición o distancia para el grafo de comunicación.

## Análisis de convergencia para los algoritmos de consenso

Considerando que para el algoritmo de consenso (2.12) se tiene una topología de comunicación invariante en el tiempo, la matriz de adyacencia es constante, y por lo tanto  $\mathcal{L}_n$  es constante, como se muestra en la sección 2.2.2, cero siempre es un eigenvalor de  $-\mathcal{L}_n$  y todos los eigenvalores diferentes a cero en  $-\mathcal{L}_n$  tienen parte negativa real. Adicionalmente el vector columna  $\mathbf{1}_n$  de unos, es un eigenvector asociado al eigenvalor cero, que nos muestra que el espacio generador  $\mathbf{1}_n$  está contenido en el kernel de  $\mathcal{L}_n$ . Por lo tanto, si el eigenvalor en cero de  $\mathcal{L}_n$  es simple, entonces  $\bar{z}(t) \rightarrow \bar{z}\mathbf{1}_n$ , donde  $\bar{z}$  es un escalar constante, lo que implica que  $|\bar{z}_i(t) - \bar{z}_j(t)| \rightarrow 0$  cuando  $t \rightarrow \infty$  para todo  $i, j = 1, \dots, n$ . En consecuencia la convergencia del algoritmo de consenso se enfoca en asegurar que el eigenvalor en cero de  $\mathcal{L}_n$  sea simple.

### 2.2.4. Algoritmo de consenso de segundo orden

En [78] se muestra la generalización del algoritmo de consenso para órdenes superiores, y se especifica que el consenso de primer y segundo orden son casos especiales del algoritmo mostrado. Debido al tipo de robots usados, a lo largo de este documento basta con el uso del consenso de primer y segundo orden.

Para analizar el consenso de segundo orden, se muestra en [77] los estados de información con dinámica de doble integrador. Entonces considere un SRM denotado por  $N = \{R_1, \dots, R_n\}$  con  $n_1$  agentes que se mueven en el plano, estos son robots de primer orden, con modelo cinemático de un solo integrador, y  $n_2$  agentes de segundo orden, representados por medio de modelo dinámico. Entonces  $n = n_1 + n_2$  y

$$\begin{aligned} \dot{\bar{z}}_i &= u_i, & i &= 1, \dots, n_1 \\ \dot{\bar{z}}_i &= p_i, & \dot{p}_i &= u_i, & i &= (n_1 + 1), \dots, (n_1 + n_2) \end{aligned} \quad (2.14)$$

Note que para el caso de robots de segundo orden,  $\bar{z}_i$  es el estado de información, es decir la posición,  $p_i$  es la velocidad y  $u_i$  la entrada de control asociada al  $i$ -ésimo robot. Entonces el algoritmo de control para robots de segundo orden está dado por

$$u_i(t) = \sum_{j=1}^{n_1+1} a_{ij} [(\bar{z}_i - \bar{z}_j) + (p_i - p_j)], \quad i = 1, \dots, n \in \mathfrak{R} \quad (2.15)$$

esta ecuación representa la aceleración del  $i$ -ésimo robot, y se logra consenso cuando para cada robot, cada  $\bar{z}_i$  y  $p_i$  para todo  $i = 1, \dots, n$ ,  $|\bar{z}_i(t) - \bar{z}_j(t)| \rightarrow 0$  y  $|p_i(t) - p_j(t)| \rightarrow 0$ , cuando  $t \rightarrow \infty$ . El sistema en lazo cerrado puede ser escrito de forma matricial

$$\begin{bmatrix} \dot{\bar{z}}_i \\ \dot{p}_i \end{bmatrix} = (\Theta \otimes I_m) \begin{bmatrix} \bar{z}_i \\ p_i \end{bmatrix} \quad (2.16)$$

siendo

$$\Theta = \begin{bmatrix} 0_{n \times n} & I_n \\ -\mathcal{L}_n(t) & -\mathcal{L}_n(t) \end{bmatrix} \quad (2.17)$$

con la matriz laplaciana  $\mathcal{L}_n$  definida en la sección 2.2.2,  $I_m$  e  $I_n$  matrices identidad de dimensiones adecuadas  $m$  y  $n$  respectivamente.

## 2.2.5. Formación de primer orden con posiciones relativas deseadas

Considere un SRM denotado por la ecuación (2.11) con las características descritas en la sección 2.2.3.  $N_i$  denota los sets de posiciones detectables por el robot  $R_i$ . Debe existir conectividad, por lo que  $N_i \neq \emptyset \quad \forall i = 1, \dots, n$ . Sea  $\mathcal{C}_n$  un conjunto de vectores de posición deseada, tal que  $c_{ij} = [h_{ij}, v_{ij}]^T \in \mathcal{C}_n$  es un vector que denota la posición deseada del robot  $R_i$  con respecto al robot  $R_j$  en una formación particular. Entonces definimos la posición relativa deseada de cada  $R_i$  en la formación por

$$z_i^* = \frac{1}{n_i} \sum_{j=N_i} (\vec{z}_j + c_{ji}) \quad (2.18)$$

donde  $n_i$  es la cardinalidad de  $N_i$ . Entonces la posición relativa deseada de  $R_i$  puede considerarse un promedio de las posiciones  $\vec{z}_i$  con respecto a los elementos de  $N_i$ . A continuación se muestra un ejemplo de ley de control para este algoritmo basado en funciones potenciales atractivas.

**Planteamiento del problema.** El objetivo de control es diseñar una ley de control  $u_i = f_i(N_i)$  para cada robot  $R_i$ , tal que

$$\lim_{t \rightarrow \infty} (\vec{z}_i - z_i^*) = 0, \quad i = 1, \dots, n. \quad (2.19)$$

Si el grafo  $\mathcal{G} = \{\mathcal{V}_n, \mathcal{E}_n, \mathcal{C}_n\}$ , que incluye nodos  $\mathcal{V}_n$ , aristas  $\mathcal{E}_n$  y vectores de posición deseada  $\mathcal{C}_n$ , es dirigido, conexo y tiene un árbol de expansión directo, éste puede ser representado por medio de la matriz de adyacencia y de la matriz Laplaciana. Entonces podemos definir una función potencial atractiva

$$\gamma_i = \|\vec{z}_i - z_i^*\|^2 \quad (2.20)$$

con  $z_i^*$  definido en (2.18) tenemos que

$$\gamma_i = \|\vec{z}_i - \sum (\vec{z}_j - c_{ji})\|^2 \quad (2.21)$$

Por lo que usando esta función potencial se propone la ley de control de la forma

$$u_i = -\frac{1}{2} \mathcal{k} \frac{\partial \gamma_i}{\partial \vec{z}_i}, \quad i = 1, \dots, n, \mathcal{k} \in \mathfrak{R} \quad (2.22)$$

en donde al sustituir la derivada de (2.20) se obtiene

$$u_i = -\mathcal{k} (\vec{z}_i - z_i^*) \quad (2.23)$$

La ecuación (2.23) es una ley de control basada en el error, con una ganancia  $\mathcal{k} > 0 \in \mathfrak{R}$ . Como ejemplo considere tres agentes  $\vec{z}_1, \vec{z}_2, \vec{z}_3$  con modelo cinemático dado por la ecuación (2.11). La conectividad se representa con  $N_1 = \{z_2\}, N_2 = \{z_3\}, N_3 = \{z_1\}$

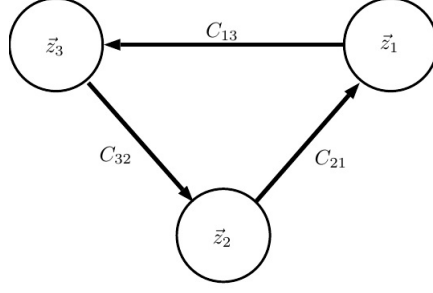


Figura 2.9: Grafo de conectividad B

Las posiciones relativas deseadas son  $z_1^* = \vec{z}_2 + c_{21}$ ,  $z_2^* = \vec{z}_3 + c_{32}$ ,  $z_3^* = \vec{z}_1 + c_{13}$ , con lo que las funciones potenciales se definen como

$$\begin{aligned}\gamma_1 &= \|\vec{z}_1 - \vec{z}_2 - c_{21}\|^2 \\ \gamma_2 &= \|\vec{z}_2 - \vec{z}_3 - c_{32}\|^2 \\ \gamma_3 &= \|\vec{z}_3 - \vec{z}_1 - c_{13}\|^2\end{aligned}\tag{2.24}$$

y el control para cada robot en lazo cerrado será

$$\begin{aligned}\dot{\vec{z}}_1 &= u_1 = -k_1(\vec{z}_1 - \vec{z}_2 - c_{21}) \\ \dot{\vec{z}}_2 &= u_2 = -k_2(\vec{z}_2 - \vec{z}_3 - c_{32}) \\ \dot{\vec{z}}_3 &= u_3 = -k_3(\vec{z}_3 - \vec{z}_1 - c_{13})\end{aligned}\tag{2.25}$$

El algoritmo de control de la ecuación (2.23) también se puede representar de forma matricial

$$\dot{\vec{z}} = -k [\mathcal{L} \otimes I_2] \vec{z} - C\tag{2.26}$$

donde  $\vec{z}$  es el vector de estados,  $\mathcal{L}$  es la matriz Laplaciana,  $I_2$  es la matriz identidad de dimensión  $2 \times 2$ ,  $\otimes$  denota el producto Kronecker y  $C = [\sum_{j \in N_1} c_1, \dots, \sum_{j \in N_n} c_n]^T$ . Para información sobre el producto Kronecker refiérase a [59]. Entonces para el ejemplo de la figura 2.1 la forma matricial está conformada por la matriz de adyacencia

$$A_d = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}\tag{2.27}$$

la matriz diagonal  $\Delta$  es

$$\Delta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\tag{2.28}$$

por lo que la matriz Laplaciana será

$$\mathcal{L} = \Delta - A_d = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}\tag{2.29}$$

sustituyendo en la ecuación (2.26)

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = -k \left[ \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \otimes I_2 \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} - \begin{bmatrix} c_{21} \\ c_{32} \\ c_{13} \end{bmatrix} \right]\tag{2.30}$$

La ecuación (2.30) representa el sistema en lazo cerrado mostrado en forma matricial.

## 2.2.6. Formación basada en distancias

Para el caso de consenso basado en distancias se desea que los robots del SRM converjan a las distancias deseadas  $d_{ij}^*$  entre los robots  $i$  y  $j$ . Entonces en el caso de integradores de primer orden se establece lo mencionado en el apartado 2.2.3. Considerando un SRM de robots de primer orden como el mostrado en la ecuación (2.11) y utilizando la notación de grafos mostrada en la sección 2.2.2, entonces sea un grafo  $\mathcal{G} = (\mathcal{V}_n, \mathcal{E}_n, \mathcal{D}_n)$  donde  $\mathcal{V}_n = \{1, \dots, n\}$  es el conjunto de nodos finitos y no vacío.  $\mathcal{E}_n \subseteq \mathcal{V}_n \times \mathcal{V}_n$  es el conjunto de aristas definidas por el par de nodos  $(i, j)$ , y sea  $\mathcal{D}_n = \{d_{ji}\}, \forall (j, i) \in E$  el vector de distancias deseadas para los robots  $i$  y  $j$ . Se define la distancia entre agentes como  $d_{ij} = \|\vec{z}_i - \vec{z}_j\|$  y la distancia deseada entre agentes como  $d_{ij}^*$ , considere que se debe de cumplir  $d_{ij} = d_{ji}$  y  $d_{ij}^* = d_{ji}^*$ . Sea  $\beta_{ij} = \|\vec{z}_i - \vec{z}_j\|^2$  la distancia euclidiana al cuadrado del robot  $R_i$  con respecto al robot  $R_j$  en una formación particular, note que  $\beta_{ij} = \|\vec{z}_i - \vec{z}_j\|^2 = (z_i - z_j)^T(z_i - z_j)$ , y que  $\beta_{ij} = \beta_{ji}$ . Entonces se usa la ecuación de consenso (2.12) como base para plantear una estrategia de control.

**Planteamiento del problema.** El objetivo de control es diseñar una ley de control  $u_i = f_i(N_i)$  para cada robot  $R_i$ , tal que

$$\lim_{t \rightarrow \infty} (d_{ij} - d_{ij}^*) = 0, \quad i = 1, \dots, n. \quad (2.31)$$

de tal forma que los agentes converjan a la distancia deseada. Para cada agente se define una función potencial  $\gamma(\beta_{ij})$  y una función  $\rho_{ij}$  definida como

$$\rho_{ij} = \frac{\partial \gamma(\beta_{ij})}{\partial \beta_{ij}} \quad (2.32)$$

note que  $\rho_{ij} = \rho_{ji}$ . Entonces se propone una ley de control

$$u_i = -\frac{1}{2}k \sum \rho_{ij} (\vec{z}_i - \vec{z}_j) \quad (2.33)$$

Una posible función potencial es

$$\gamma(\beta_{ij}) = ((\beta_{ij} - (d_{ij}^*)^2)^2) \quad (2.34)$$

con lo que

$$\rho_{ij} = 2 * (\beta_{ij} - (d_{ij}^*)^2) \quad (2.35)$$

entonces la ley de control es

$$u_i = -k \sum ((\beta_{ij} - (d_{ij}^*)^2) (\vec{z}_i - \vec{z}_j)) \quad (2.36)$$

La formación basada en distancias se puede extender al caso del doble integrador, de forma análoga al caso de posiciones, como se muestra en [70] basado en [69] y [72]. Uno de los problemas de la formación basada en distancias es que al controlar la distancia y no las posiciones, baja conectividad en el grafo de comunicación no garantiza la convergencia. Un grafo globalmente rígido garantiza la convergencia, lo cual se muestra en el apartado 2.2.2.



## 2.2.7. Evasión de colisiones

Al tratar de lograr en cualquier ley de consenso, basado en (2.12), en que  $u_i = f_i(N_i)$  para cada robot  $R_i$ , se busca que

$$\lim_{t \rightarrow \infty} (\bar{z}_i - z_i^*) = 0, \quad i = 1, \dots, n. \quad (2.37)$$

lo cual cumplirá la condición de convergencia a la posición deseada. Pero se puede agregar la siguiente condición

$$\| \bar{z}_i(t) - \bar{z}_j(t) \| \neq 0, \quad \forall t \geq 0, i \neq j \quad (2.38)$$

esta condición garantiza la no colisión entre robots. Considerando la función potencial atractiva mostrada en (2.2.5), el sistema

$$\gamma_i = \| \bar{z}_i - \bar{z}_j - c_{ji} \|^2 \quad (2.39)$$

siendo las funciones  $\gamma_i$  siempre positivas y con un mínimo en  $\gamma_i = 0$ , se propone la ley

$$u_i = -\frac{1}{2} \mathcal{K} \frac{\partial \gamma_i}{\partial \bar{z}_i}, \quad i = 1, \dots, n, \mathcal{K} \in \mathfrak{R} \quad (2.40)$$

Esta ley de control cumple la condición (2.37) pero no la condición (2.38) de no colisión entre agentes. Para cumplir con esta condición se usa una función potencial repulsiva. Como se define en la sección 2.2.6, el cuadrado de la distancia entre agentes está dada por  $\beta_{ij} = \| \bar{z}_i - \bar{z}_j \|^2$ , entonces para todos los agentes  $R_i$ , están en peligro de colisión cuando se encuentran en el conjunto  $M_i = \{ \in N \mid \beta_{ij} \leq d^2 \}$  donde  $d$  es el diámetro de la zona de influencia,  $M_i$  cambia con el movimiento de los agentes. La ley de consenso que incluye potenciales repulsivos y atractivos puede estar dada por:

$$u_i = -\frac{1}{2} \mathcal{K} \frac{\partial \gamma_i}{\partial \bar{z}_i} - \sum_{j \in M_i} \frac{\partial V_{ij}}{\partial \bar{z}_i}, \quad i = 1, \dots, n \quad (2.41)$$

donde  $\gamma_i$  está definida en (2.39) y  $V_{ij}$  es una función repulsiva que cumple tres condiciones

- $V_{ij}$  se incrementa continuamente cuando  $\beta_{ij} \leq d^2$  y  $\beta_{ij} \rightarrow 0$
- $\lim_{\beta_{ij} \rightarrow 0} V_{ij} = \infty$
- $V_{ij} = 0$  para  $\beta_{ij} \geq d^2$ ,  $\frac{\partial V_{ij}}{\partial z_i} = 0$  para  $\beta_{ij} = d^2$

estas condiciones garantizan que la función potencial repulsiva solo aparezca en el radio de influencia del agente, y únicamente con respecto a otros agentes. Las siguientes tres funciones también satisfacen las condiciones anteriores

$$\begin{aligned} V_{ij} &= \begin{cases} \eta \left( \frac{1}{\beta_{ij}} - \frac{1}{d^2} \right)^2, & \text{si } \beta_{ij} \leq d^2 \\ 0, & \text{si } \beta_{ij} > d^2 \end{cases} \\ V_{ij} &= \begin{cases} \eta \left( \frac{1}{\beta_{ij}} - \frac{1}{d^2} \right)^r, & \text{si } \beta_{ij} \leq d^2 \\ 0, & \text{si } \beta_{ij} > d^2 \end{cases}, r = 2, 3, 4... \\ V_{ij} &= \begin{cases} \eta \left( \frac{(\beta_{ij} - d^2)^2}{\beta_{ij}} \right), & \text{si } \beta_{ij} \leq d^2 \\ 0, & \text{si } \beta_{ij} > d^2 \end{cases} \end{aligned} \quad (2.42)$$

donde  $\eta > 0$ . Note que es posible reescribir  $\frac{\partial V_{ij}}{\partial \vec{z}_i} = 2\frac{\partial V_{ij}}{\partial \beta_{ij}}(\vec{z}_i - \vec{z}_j)$ , ya que  $\beta_{ij} = \beta_{ji}$ , se satisface que  $V_{ij} = V_{ji}$  y  $\frac{\partial V_{ij}}{\partial \beta_{ij}} = \frac{\partial V_{ji}}{\partial \beta_{ji}}$ ,  $\forall i \neq j$ , lo que denota que la función potencial repulsiva posee una propiedad de antisimetría

$$\frac{\partial V_{ij}}{\partial \vec{z}_i} = -\frac{\partial V_{ji}}{\partial \vec{z}_j}, \forall i \neq j \quad (2.43)$$

La gran desventaja de incluir funciones repulsivas y atractivas en una ley de control para SRM es que los agentes pueden quedar atrapados en puntos de equilibrio no deseados. La presente tesis plantea una alternativa de diseño de funciones con comportamiento atractivo y repulsivo combinados. Esto permitirá que bajo ciertas condiciones, los equilibrios no deseados puedan ser evitados.

### 2.2.8. Modelado de robots móviles no holonómicos

El objetivo de esta sección es dar detalle de los modelos matemáticos usados para los robots usados en las simulaciones y experimentos. El objetivo del modelado es describir propiedades estructurales en el modelo cinemático o dinámico, tomando en cuenta las restricciones de movilidad derivadas de las restricciones. Se pueden obtener diferentes modelos que denotan las características de los robots.

- Modelo cinemático de postura, es la descripción mas sencilla que brinda información del modelo, captura las características de maniobrabilidad del robot.
- Modelo dinámico es el caso mas general, da una descripción que incluye fuerzas generalizadas.

Las suposiciones básicas para el modelado de los robots con ruedas son que la estructura es un cuerpo rígido, no poseen ruedas deformables y se mueven en un plano horizontal. Los dos robots con ruedas se modelan en base a las ecuaciones de movimiento de cuerpo rígido, simplificado en la metodología presentada en [20]. El modelo de postura se describe en términos de las coordenadas  $x, y$  con respecto al origen y que el ángulo de orientación es  $\theta$ , con lo que la postura del robot está dada por

$$\vec{z} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (2.44)$$

y la matriz de rotación de la orientación con respecto al marco de referencia es

$$R(\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.45)$$

Se asume que durante el movimiento el plano de cada rueda permanece vertical y rueda a lo largo de su eje horizontal. Los robots presentados en este documento incluyen dos tipos de rueda, la convencional para el unicycle y la sueca o mecanum para el robot omnidireccional de cuatro ruedas. Ambas se consideran ideales, es decir que el contacto con el suelo se reduce a un solo punto en el plano. Para el caso de la rueda convencional, solo se mueve en el eje de dirección del giro, sin deslizamiento a lo largo de los otros ejes. Para el caso de la rueda sueca dos ejes tienen desplazamiento y uno se encuentra fijo.

## Uniciclo

El robot tipo uniciclo típico tiene dos ruedas fijas con un eje de rotación común y una rueda loca o *caster*. Su denominación en la metodología presentada en [20] es de tipo (2,0), debido a que posee una rueda fija (o cualquier cantidad de ruedas fijas con eje común), una rueda *caster* y cero ruedas direccionables. Su configuración lo hace pseudoholonómico, requiriendo el movimiento sincronizado de ambas ruedas hacia adelante o atrás para moverse en esa dirección, y el movimiento diferencial de las ruedas para girar. De acuerdo a la figura 2.10, el modelo cinemático del robot uniciclo está dado por

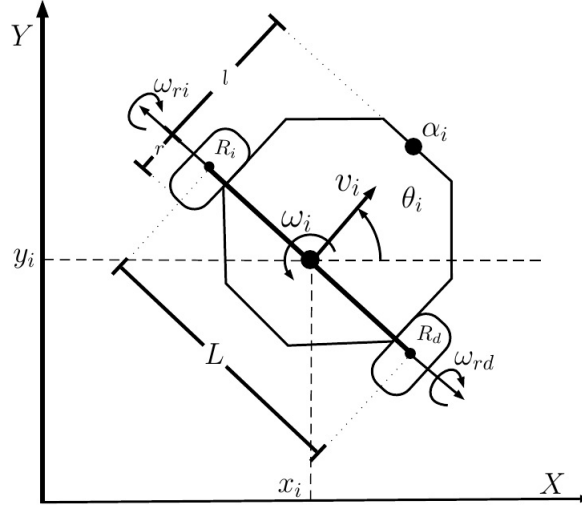


Figura 2.10: Robot tipo Uniciclo

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (2.46)$$

donde  $v_i$  es la velocidad lineal del punto central del robot, ubicado sobre el eje entre las ruedas,  $\omega_i$  es la velocidad angular. Se sabe, como se muestra en [14], que el sistema (2.46) no puede ser estabilizado por alguna ley de control continua e invariante en el tiempo. Debido a esta restricción, se simplifica el análisis seleccionando como salida de control las coordenadas  $\alpha_i = [x_i + l \cos \theta_i, y_i + l \sin \theta_i]^T$ , mostradas en la figura 2.10. Dado que la salida de control  $\alpha_i$  se denota como

$$\dot{\alpha}_i = \underbrace{\begin{bmatrix} \cos \theta_i & -l \sin \theta_i \\ \sin \theta_i & -l \cos \theta_i \end{bmatrix}}_{A(\theta_i)} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (2.47)$$

por lo que si se define una ley de control  $[v_i, \omega_i]^T = A^{-1}(\theta_i)u_i$ , la ecuación (2.47) se reduce a  $\dot{\alpha}_i = u_i$ , donde la entrada de control  $u_i$  se selecciona de acuerdo con la dinámica deseada para las coordenadas  $\alpha_i$ . Las velocidades de las ruedas se calculan con base en la velocidad lineal  $v_i$  y angular  $\omega_i$  como

$$\begin{bmatrix} \omega_{rd} \\ \omega_{ri} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & \frac{L}{2} \\ 1 & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (2.48)$$

Estas velocidades angulares  $\omega_{rd}$  y  $\omega_{ri}$  correspondientes a las ruedas son las que se alimentan directamente a los motores que mueven el robot uniclo.

### Omnidireccional

De acuerdo a la clasificación de [20], el robot omnidireccional de cuatro ruedas es del tipo (3,0) lo que indica que se puede mover a cualquier dirección del plano, teniendo cuatro ruedas con rodillos a cuarenta y cinco grados de inclinación con respecto a la cara de la rueda mostrada en la figura 2.11. Esto le permite desplazarse y rotar en todas las direcciones rotando las ruedas como se muestra en la figura 2.13.

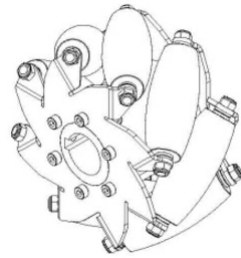


Figura 2.11: Rueda Sueca o mecanum

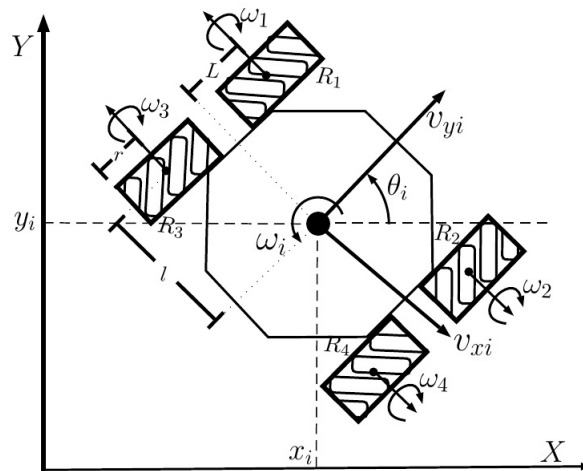


Figura 2.12: Robot tipo Omnidireccional

El modelo cinemático que describe los movimientos del robot omnidireccional se basa en la figura 2.12, y se muestra a continuación

$$\dot{\vec{z}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix}}_{R(\theta_i)} \begin{bmatrix} v_{xi} \\ v_{yi} \end{bmatrix} \quad (2.49)$$

$$\dot{\theta}_i = \omega_i$$

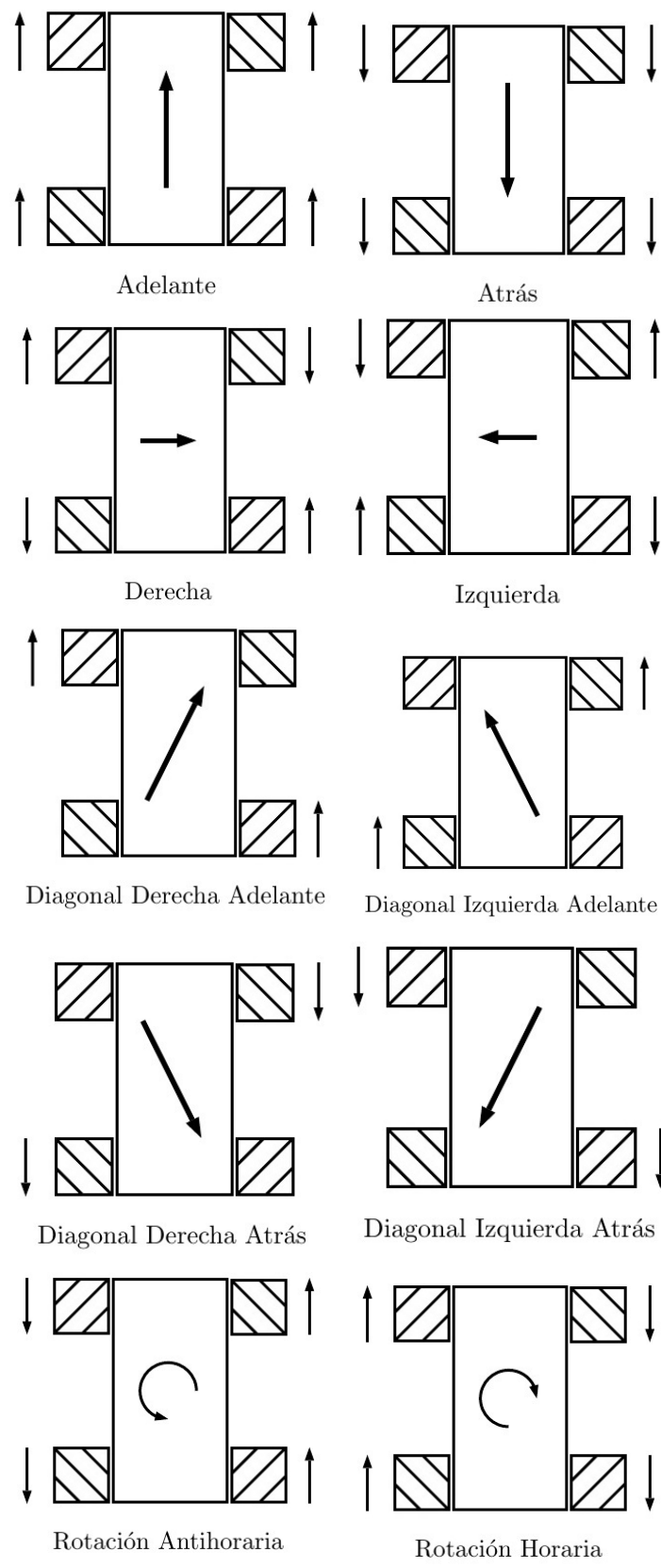


Figura 2.13: Movimientos del robot omnidireccional

donde  $v_{xi}$  y  $v_{yi}$  son las componentes de la velocidad lineal del punto central del robot y  $\omega_i$  es la velocidad angular, entonces se puede definir la ley de control como  $[v_{xi}, v_{yi}]^T = R^{-1}(\theta_i)u_i$ , donde las coordenadas traslacionales de (2.49) se reducen a  $\dot{\vec{z}}_i = u_i$ , donde la entrada de control deseada es  $u_i$ . Note que la dinámica de la orientación está desacoplada, por lo que se puede definir una ley de control para  $\omega_i$ , logrando una orientación deseada en  $\theta_i$ .

Para el robot omnidireccional las velocidades de las ruedas mostradas en la figura 2.12 dependen de las velocidades, lineal  $v_i$  con sus componentes  $v_{ix}$  y  $v_{iy}$  y angular  $\omega_i$ , y se calculan como

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 1 & -(L+l) \\ -1 & 1 & (L+l) \\ -1 & 1 & -(L+l) \\ 1 & 1 & (L+l) \end{bmatrix} \begin{bmatrix} v_{xi} \\ v_{yi} \\ \omega_i \end{bmatrix} \quad (2.50)$$

donde  $L$  y  $l$ , mostradas en la figura 2.12, son las distancias de las ruedas al centro del robot. Estas velocidades angulares,  $\omega_1$  hasta  $\omega_4$  son las que se alimentan directamente a los motores que mueven el robot omnidireccional.

## Cuadricóptero

El cuadricóptero, cuadirotor o cuadróptero es una aeronave de ala rotatoria que consta de cuatro rotores eléctricos con hélices que al girar le brindan propulsión para su sostén en el aire. Velocidades diferenciales en los cuatro rotores permiten el movimiento en los ángulos de roll, pitch y yaw, por lo tanto, desplazamiento espacial tridimensional. Detallado en [42], el cuadricóptero puede ser modelado como un cuerpo rígido con masa, sujeto a fuerzas inerciales y gravitacionales en un espacio de tres dimensiones. La combinación de las velocidades angulares de las hélices producen, de forma simultanea, una fuerza de empuje aplicada al centro de masa y tres momentos de inercia.

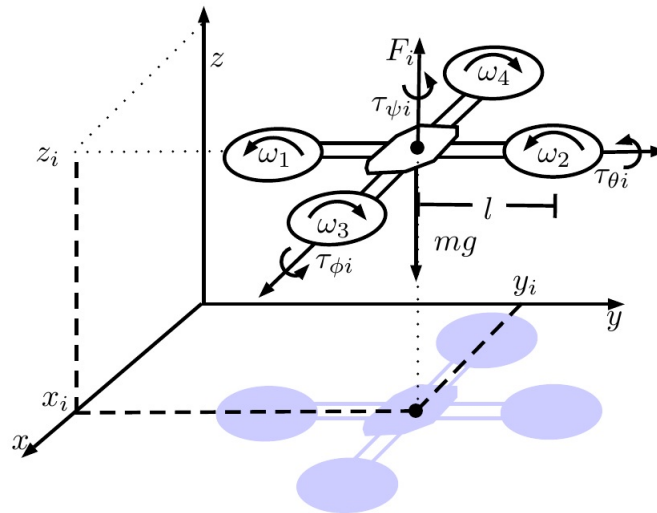


Figura 2.14: Robot Cuadricóptero

De acuerdo a la figura 2.14, la posición del marco de referencia del cuadricóptero  $\Gamma_b$  con respecto al marco de referencia global  $\Gamma_e$  está dada por  $\vec{z}_i = [x_i(t), y_i(t), z_i(t)]^T$ .

Las rotaciones al rededor de los ejes estan dadas por  $\eta = [\phi, \theta, \psi]^T$ . conocidos como ángulos de Euler, roll, pitch y yaw, respectivamente. Las velocidades lineal y angular en el marco de referencia del robot se definen con  $V = [u, v, w]^T$  y  $\Omega = [p, q, r]^T$ , a correspondencia. El modelo cinemático relaciona las fuerzas traslacional y rotacional de ambos marcos de referencia. Por lo que, las velocidades traslacionales están dadas por

$$\dot{\tilde{z}} = R_b^e V \quad (2.51)$$

con  $R_b^e$  siendo el resultado de la multiplicación entre las matrices de rotación

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & -\cos \phi \end{bmatrix}, R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Se conoce que las matrices de rotación  $R_x$ ,  $R_y$  y  $R_z$  no son conmutativas y difieren en la descripción de la secuencia de ángulos de Euler. Por ejemplo en [80], los autores usan la matriz  $R_b^e = R_z R_y R_x$ , y en [30] trabajan con  $R_b^e = R_x R_y R_z$ . Claramente la ley de control cambia dependiendo del orden seleccionado de rotaciones, por lo que para simplificar se usa  $R_b^e = R_z R_y R_x$  dado entonces por

$$R_b^e = \begin{bmatrix} c\psi c\theta & -c\theta s\psi & s\theta \\ c\phi s\psi + c\psi s\phi s\theta & c\phi c\psi - s\phi s\psi s\theta & -c\theta s\phi \\ s\phi s\psi - c\phi c\psi s\theta & c\psi s\phi + c\phi s\psi s\theta & c\phi c\theta \end{bmatrix} \quad (2.52)$$

donde  $c\phi = \cos \phi$ ,  $c\theta = \cos \theta$ ,  $c\psi = \cos \psi$  y  $s\phi = \sin \phi$ ,  $s\theta = \sin \theta$ ,  $s\psi = \sin \psi$ . La velocidad del vector  $\eta$  referenciado a  $\Gamma_b$  está dada por

$$\dot{\eta} = W_n^{-1} \Omega \quad (2.53)$$

con

$$W_n = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.54)$$

por medio del enfoque Euler-Lagrange mostrado en [17], el modelo dinámico del cuadricóptero está dado por

$$m \ddot{\tilde{z}} = R_b^e \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (2.55)$$

$$\tilde{J} \ddot{\eta} = \tau - C(\eta, \dot{\eta}) \dot{\eta} \quad (2.56)$$

siendo  $m$  la masa del vehículo,  $F$  la fuerza de empuje principal aplicada a la masa,  $\tilde{J} = JW_n$  con  $J$  la matriz simétrica de términos inerciales,  $g$  es la constante gravitacional,  $C(\eta, \dot{\eta}) = J\dot{W}_n$  es el término de Coriolis y  $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T$  son los

torques generados por los motores  $\Gamma_b$ . De acuerdo a [80], la dinámica de los ángulos de orientación puede ser simplificada realizando un cambio de variable

$$\tau = \tilde{J}\tilde{\tau} + C(\eta, \dot{\eta})\dot{\eta} \quad (2.57)$$

donde  $\tilde{\tau} = [\tilde{\tau}_\phi, \tilde{\tau}_\theta, \tilde{\tau}_\psi]^T$  el vector auxiliar de torque. Por lo que sustituyendo la matriz de rotación  $R_b^e$  dada en (2.52) y los torques (2.57), el sistema dinámico (2.55) se puede reducir a

$$m\ddot{z} = \begin{bmatrix} F \sin \theta \\ -F \cos \theta \sin \phi \\ F \cos \theta \cos \phi - mg \end{bmatrix} \quad (2.58)$$

$$\ddot{\eta} = \tilde{\tau} \quad (2.59)$$

Observe que la selección de  $R_b^e$  implica que el ángulo yaw  $\psi$  no aparece en la dinámica traslacional. Esta propiedad es ventajosa y será explotada mas adelante durante la implementación de las leyes de control.

La fuerza de empuje es  $F = f_1 + f_2 + f_3 + f_4$  donde  $f_i, i = 1, \dots, n$  son las fuerzas individuales de empuje generadas por el  $i$ -esimo motor como se muestra en la figura 2.14. Considerando que todas las propelas tienen la misma dimensión y paso, note que los rotores 1 y 2 giran en sentido de las manecillas del reloj, produciendo cambios en  $\phi$  y desplazamiento en el eje  $Y$ . Sea  $\ell$  el largo del brazo del cuadricóptero, entonces  $\tau_\phi = \ell(f_2 - f_1)$ . De forma análoga los rotores 3 y 4 rotan en sentido contrario a las manecillas del reloj, modificando el ángulo  $\theta$  y el movimiento en el eje  $X$ . Entonces,  $\tau_\theta = \ell(f_4 - f_3)$ . El torque en el ángulo yaw se produce por la diferencia de pares de rotores 1 - 2 y 3 - 4, y está dado por  $\tau_\psi = \frac{b}{k}[-f_1 - f_2 + f_3 + f_4]$ , donde  $k$  es la constante que caracteriza a cada arreglo rotor-propela, y  $b$  es denominada constante de arrastre del cuerpo rígido como se muestra en [62]. De forma resumida la relación de fuerza de empuje y las entradas del sistema de (2.58)-(2.59) se pueden escribir de forma matricial como

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\ell & \ell & 0 & 0 \\ 1 & 0 & -\ell & \ell \\ -\frac{b}{k} & -\frac{b}{k} & \frac{b}{k} & \frac{b}{k} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (2.60)$$

En un modelo extendido, las fuerzas individuales  $f_i$  dependen de la velocidad del rotor, e inciden factores como las características de la propela (largo, ángulo de ataque y deflección), turbulencia del aire y otros factores ambientales. Estas consideraciones aerodinámicas se modelan en un actuador no lineal. De acuerdo a [30], descartando efectos transitivos y perturbaciones, las fuerzas  $f_i$  pueden modelarse como

$$f_i = k\omega_i^2, i = 1, \dots, 4 \quad (2.61)$$

donde  $\omega_i$  es la velocidad angular del rotor  $i$ , y  $k$  es la constante de empuje antes mencionada. Esta relación estática no es restrictiva ya que la dinámica de los actuadores es rápida con respecto al movimiento de la aeronave. en el caso de motores trifásicos, usados de forma comercial, el control electrónico de velocidad (ESC por sus siglas en inglés) resuelve la etapa de potencia y el control de velocidad. Los proveedores brindan en las especificaciones la fuerza del motor (e indirectamente la



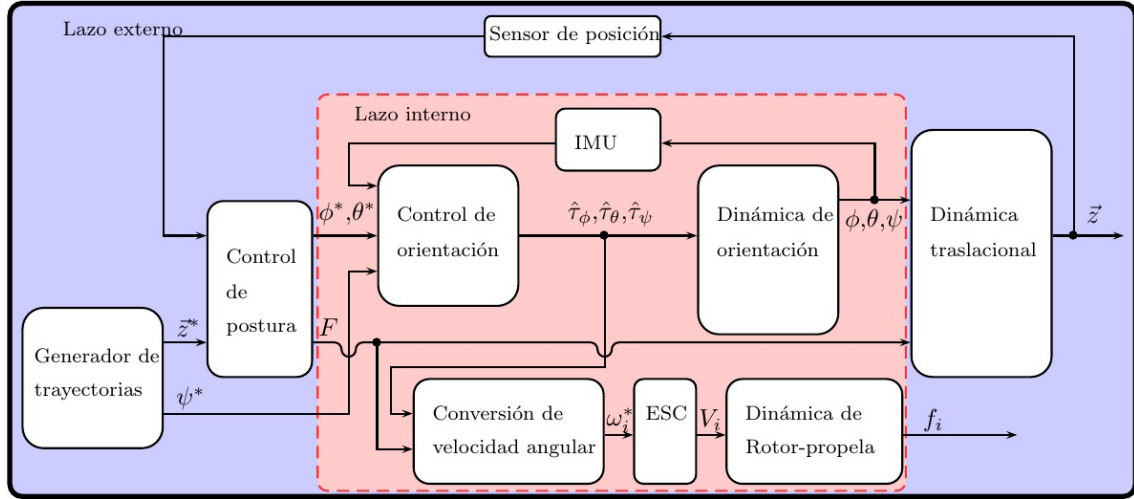


Figura 2.15: Niveles de control para el cuadricóptero

constante  $k$ ), la cual calculan de forma experimental con diferentes configuraciones de propelas. Claramente una velocidad angular menor requiere menor consumo de corriente.

Como se detalla en [42], el control del cuadróptero puede ser dividido en dos niveles jerárquicos, control de postura y control de orientación, asumiendo que la dinámica de la rotación converge más rápido que la dinámica de la traslación. El esquema de control se muestra en la figura 2.15. Entonces, la ley de control  $F = \frac{m(u_z+g)}{\cos\theta\cos\phi}$ ,  $\eta_i^* = [\phi_i^*, \theta_i^*, \psi_i^*]^T = \left[ \arctan\left(-\frac{u_{iy}}{u_{iz}+g}\right), \arctan\left(\frac{u_{ix}\cos\phi_i}{u_{iz}+g}\right), \psi_i^* \right]$ , linealiza la dinámica traslacional a  $\ddot{z}_i = [u_{ix}, u_{iy}, u_{iz}]^T$  y genera los puntos de ajuste para los ángulos de orientación, que se alcanzan por medio de una ley interna de control  $\tau_i = \ddot{\eta}_i^* - k_1(\dot{\eta}_i - \dot{\eta}_i^*) - k_2(\eta_i - \eta_i^*)$ . Se usan entonces las coordenadas  $x$  y  $y$  del cuadróptero como las entradas de control que se pueden controlar por medio de las leyes de consenso, mientras que la coordenada  $z$  se puede controlar independientemente de las otras.

# Capítulo 3

## Control de formación basado en distancia y orientación

A lo largo de este capítulo se presenta una metodología de formación basada en el uso de distancia entre robots con referencia a un eje coordenado arbitrario. Usando técnicas de Lyapunov y considerando que los agentes cuentan con sensores para medir la orientación y la distancia, se plantean métodos definidos por funciones potenciales atractivo-repulsivas basadas en distancia, para grafos no dirigidos y que incluyen condiciones de evasión de colisiones. Adicionalmente se incluye un término de tipo integral que elimina los efectos de zona muerta en los actuadores para evitar el uso de técnicas de normalización. Esta técnica de control se implementa en simulaciones con robots tipo omnidireccional, para luego extenderse al caso de robots no holonómicos tipo unicycle con experimentos en tiempo real.

Todo lo referente a este capítulo se encuentra publicado en la revista *Advanced Robotics*, volumen 30 número 14 correspondiente al año 2016, siendo éste artículo la referencia [61].

### 3.1. Planteamiento del problema

Considere, como se muestra en la sección 2.2.6,  $N = [R_1, \dots, R_n]$  robots con posiciones  $\vec{z} = \vec{z}_1, \dots, \vec{z}_n$ , con  $\vec{z}_i = [x_i, y_i]$ ,  $i = 1, \dots, n$  respectivamente, con modelo cinemático de primer orden como se muestra en la ecuación (2.11), dado por

$$\dot{\vec{z}}_i = u_i \in \mathbb{R}^2, \quad i = 1, \dots, n \quad (3.1)$$

donde  $u_i$  es la velocidad de  $R_i$ . Para todos los robots  $R_i$ , con  $i = 1, \dots, n$ . Considere la notación de grafos mostrada en la parte 2.2.2, entonces sea un grafo  $\mathcal{G} = (\mathcal{V}_n, \mathcal{E}_n, \mathcal{D}_n)$  el que describe la topología de comunicación y las distancias deseadas entre robots.

Suponiendo que el grafo tiene completitud, se cumple que  $d_{ij} = d_{ji}$  para todo  $(j, i) \in E$ . Afirmando que el grafo es conexo, y garantizando la existencia de un árbol de expansión directo, se obtienen las características deseables de la matriz Laplaciana mostrada en la sección 2.2.2. Como se define en el apartado 2.2.6 sea

$$\begin{aligned} \beta_{ij} &= \|\vec{z}_j - \vec{z}_i\| \\ &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\ &= \sqrt{(\vec{z}_j - \vec{z}_i)^T (\vec{z}_j - \vec{z}_i)} \end{aligned} \quad (3.2)$$

la distancia entre robots  $(i, j)$ . Note que  $\beta_{ij} = \beta_{ji}$  es simétrica. Como  $\vec{z}_i$  y  $\vec{z}_j$  son variables dependientes del tiempo,  $\beta_{ij}$  es diferenciable con respecto al tiempo, cuya derivada es

$$\begin{aligned}
\dot{\beta}_{ij} &= \frac{1}{2} \left( (x_j - x_i)^2 + (y_j - y_i)^2 \right)^{-\frac{1}{2}} \left( 2(x_j - x_i)(\dot{x}_j - \dot{x}_i) + 2(y_j - y_i)(\dot{y}_j - \dot{y}_i) \right) \\
&= \frac{(x_j - x_i)(\dot{x}_j - \dot{x}_i) + (y_j - y_i)(\dot{y}_j - \dot{y}_i)}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} = \frac{(\vec{z}_j - \vec{z}_i)^T (\dot{\vec{z}}_j - \dot{\vec{z}}_i)}{\sqrt{(\vec{z}_j - \vec{z}_i)^T (\vec{z}_j - \vec{z}_i)}} \\
&= \frac{(\vec{z}_j - \vec{z}_i)^T (\dot{\vec{z}}_j - \dot{\vec{z}}_i)}{\beta_{ij}} \tag{3.3}
\end{aligned}$$

Considere  $\theta_{ij}$  como el ángulo del vector  $\vec{z}_j - \vec{z}_i$ , con respecto a un eje arbitrario. Este ángulo  $\theta_{ij}$  podría ser medido por medio de un magnetómetro usando el polo magnético de la tierra. Note que por simetría  $\theta_{ji} = \pi + \theta_{ij}$  de acuerdo a la figura 3.1.

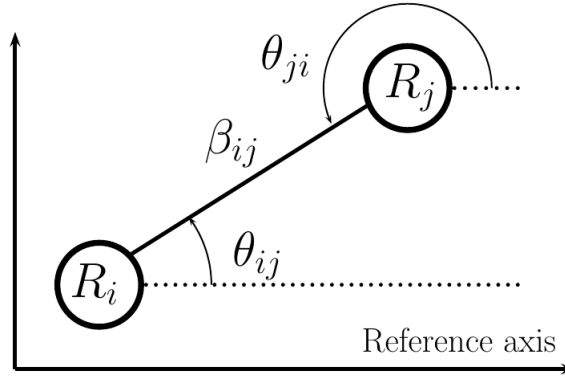


Figura 3.1: Distancia relativa y orientación entre dos robots

**Planteamiento** El objetivo de control es crear una ley de control descentralizada para cada robot  $u_i$ , que dependa de  $\beta_{ij}$ ,  $\theta_{ij}$  y  $d_{ij} \forall j \in N_i$ , que cumpla  $\lim_{t \rightarrow \infty} \beta_{ij} - d_{ij} = 0, \forall j \in N_i$  evitando colisiones entre robots, es decir,  $\beta_{ij} \neq 0, \forall i, j \in N$ .

Este acercamiento de distancia y orientación es más descentralizado que el control tradicional basado en distancias ya que los robots no requieren saber la posición  $\vec{z}_i$  en el plano de otros miembros de SRM. En la siguiente sección se muestran los resultados de esta propuesta.

## 3.2. Control y prueba de convergencia

Para analizar la estrategia de control y realizar la prueba de convergencia se inicia definiendo el error de distancia entre dos robots  $R_i$  y  $R_j$  dado por

$$\varphi_{ij} = \beta_{ij} - d_{ij}, \tag{3.4}$$

con  $d_{ij} \in \mathbb{R}$ . Un grafo bien establecido garantiza que  $d_{ij} = d_{ji}$ . Por lo que se cumple la condición de simetría para  $\varphi_{ij} = \varphi_{ji}$ . Usando este error de distancia, la siguiente definición establece las funciones potenciales implementadas con términos repulsivos para evitar colisiones.

**Definición** Una Función Potencial Atractiva-Repulsiva basada en distancias es cualquier función  $V_{ij} : \mathbb{R} \rightarrow \mathbb{R}$  definida para un par de robots  $R_i$  y  $R_j$  que satisface las siguientes condiciones

- $V_{ij}$  es una función suave en  $\varphi_{ij}$ .
- $V_{ij} > 0$  y  $V_{ij} = 0$  solo si  $\varphi_{ij} = 0$ .
- $V_{ij} \rightarrow \infty$  cuando  $\beta_{ij} = \varphi_{ij} + d_{ij} \rightarrow 0$ .
- $\frac{\partial V_{ij}}{\partial \varphi_{ij}} = \varphi_{ij} \Gamma_{ij}$  con  $\Gamma_{ij}$  siendo una función suave de  $\varphi_{ij}$ .

dos ejemplos de  $V_{ij}$  son

$$V_{ij} = \frac{(\beta_{ij} - d_{ij})^2}{\beta_{ij}} = \frac{\varphi_{ij}^2}{\varphi_{ij} + d_{ij}} \quad (3.5)$$

$$V_{ij} = \left[ \frac{1}{\beta_{ij}} - \frac{1}{d_{ij}} \right]^{2n} = \left[ \frac{\varphi_{ij}}{(\varphi_{ij} + d_{ij}) d_{ij}} \right]^{2n} \quad (3.6)$$

La función (3.5) está dada por [25], mientras que a la función (3.6) se le conoce como *función potencial de Khatib* dada en [44]. La derivada con respecto al tiempo de  $V_{ij}$  se puede expresar como

$$\dot{V}_{ij} = \left( \frac{\partial V_{ij}}{\partial \varphi_{ij}} \right) \left( \frac{\partial \varphi_{ij}}{\partial \beta_{ij}} \right) \dot{\beta}_{ij} \quad (3.7)$$

Note que la ecuación (3.4),  $\partial \varphi_{ij} / \partial \beta_{ij} = 1$ . Entonces, sustituyendo (3.3) y (3.2), la ecuación (3.7) se vuelve

$$\dot{V}_{ij} = \varphi_{ij} \Gamma_{ij} \left[ \frac{(z_j - z_i)^T (\dot{z}_j - \dot{z}_i)}{\beta_{ij}} \right]. \quad (3.8)$$

De acuerdo a la figura 3.1, la diferencia entre coordenadas  $\vec{z}_i$  y  $\vec{z}_j$  se pueden expresar en términos de distancia y ángulo de orientación como

$$\vec{z}_j - \vec{z}_i = \beta_{ij} \begin{bmatrix} \cos \theta_{ij} \\ \sin \theta_{ij} \end{bmatrix}, \quad (3.9)$$

pudiendo reescribir la ecuación (3.8) de la siguiente manera

$$\dot{V}_{ij} = \varphi_{ij} \Gamma_{ij} [\cos \theta_{ij} \quad \sin \theta_{ij}] \begin{pmatrix} \dot{z}_j \\ -\dot{z}_i \end{pmatrix} = \varphi_{ij} A_{ij} \begin{pmatrix} \dot{z}_j \\ -\dot{z}_i \end{pmatrix}, \quad (3.10)$$

donde

$$A_{ij} \equiv [\Gamma_{ij} \cos \theta_{ij} \quad \Gamma_{ij} \sin \theta_{ij}]. \quad (3.11)$$

En el siguiente teorema se presentan los resultados principales que aseguran la convergencia global de la ley de control propuesta.

**Teorema 1** Considere un grupo de robots con modelo cinemático dado por (3.1), comunicados por un grafo no dirigido y rígido. Defina una ley de control para cada robot de la forma

$$u_i = K_i (\varrho_p \varphi_i + \varrho_e \omega_i), i = 1, \dots, n \quad (3.12)$$

donde  $\varrho_p \in \mathbb{R}$ ,  $\varrho_e \in \mathbb{R}$ ,  $\varrho_p > 0$ ,  $\varrho_e > 0$  son ganancias constantes,

$$\begin{aligned} \varphi_i &= [a_{i1}\varphi_{i1}, \dots, a_{in}\varphi_{in}]^T \\ \omega_i &= [a_{i1}\omega_{i1}, \dots, a_{in}\omega_{in}]^T, \end{aligned} \quad (3.13)$$

con  $a_{ij}$  definido en la sección 2.2,

$$K_i = A_i^T, \text{ with } A_i = \begin{bmatrix} A_{i1} \\ \vdots \\ A_{in} \end{bmatrix} \in \mathfrak{R}^{n \times 2}, i = 1, \dots, n. \quad (3.14)$$

y  $A_{ij}$  presentado en la ecuación (3.11), con

$$\omega_i = [\omega_{i1}, \dots, \omega_{in}]^T = \int M_i \varphi_i dt, \text{ with } M_i = \varrho_e A_i A_i^T, i = 1, \dots, n. \quad (3.15)$$

Suponga que

- Los robots no colisionan en  $t = 0$ .
- Los robots no son colineales en  $t = 0$ .
- El equilibrio ocurre cuando  $\varphi_{ij} = 0$ ,  $i, j = 1, \dots, n$  a lo largo de las trayectorias de los robots.

Entonces, en el sistema en lazo cerrado (3.1) y (3.12) los robots convergen a la formación deseada, es decir  $\lim_{t \rightarrow \infty} \varphi_{ij} = 0$ ,  $\forall j \in N_i$  evitando colisiones entre robots, lo que significa que  $\beta_{ij} > 0$ ,  $\forall i \neq j$ ,  $\forall t > 0$ .

**Prueba** La dinámica de los robots en lazo cerrado (3.1)-(3.12) está dada por

$$\dot{z}_i = K_i (\varrho_p \varphi_i + \varrho_e \omega_i), \quad i = 1, \dots, n \quad (3.16)$$

Considere una función candidata Lyapunov

$$V = \frac{1}{2} \sum_{i=1}^n \left( \sum_{j=1}^n V_{ij} + \omega_i^T \omega_i \right) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n V_{ij} + \frac{1}{2} \sum_{i=1}^n (\omega_i^T \omega_i) \quad (3.17)$$

donde  $V_{ij}$  es la Función Potencial Atractiva-Repulsiva basada en Distancias. Note que  $V$  siempre es positiva, y se desvanece cuando se logra el objetivo, es decir cuando  $\varphi_{ij} = 0$ ,  $\forall j \in N_i$ . La derivada con respecto al tiempo de  $V$  se muestra a continuación

$$\dot{V} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \dot{V}_{ij} + \sum_{i=1}^n [\omega_i^T \dot{\omega}_i] \quad (3.18)$$

Sustituyendo (3.10) en (3.18) se obtiene que

$$\dot{V} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n [\varphi_{ij} A_{ij} (\dot{z}_j - \dot{z}_i)] + \sum_{i=1}^n [\omega_i^T \dot{\omega}_i]. \quad (3.19)$$

Expandiendo el término del lado derecho, la ecuación (3.19) se reduce a

$$\dot{V} = \frac{1}{2} \left( \sum_{i=1}^n \sum_{j=1}^n [\varphi_{ij} A_{ij} \dot{z}_j] - \sum_{j=1}^n \sum_{i=1}^n [\varphi_{ij} A_{ij} \dot{z}_i] \right) + \sum_{i=1}^n [\omega_i^T \dot{\omega}_i]. \quad (3.20)$$

Como se observa en la figura 3.1 que  $\cos \theta_{ij} = -\cos \theta_{ji}$  y  $\sin \theta_{ij} = -\sin \theta_{ji}$ , se sabe que  $A_{ij} = -A_{ji}$ . Como el grafo de comunicación es no dirigido, es decir  $(i, j) \in E$ , entonces  $(j, i) \in E$  y considerando las características de simetría en  $\varphi_{ij} = \varphi_{ji}$ , la ecuación (3.20) se reduce a

$$\begin{aligned} \dot{V} &= \frac{1}{2} \left( \sum_{i=1}^n \sum_{j=1}^n [\varphi_{ij} A_{ij} \dot{z}_j] + \sum_{j=1}^n \sum_{i=1}^n [\varphi_{ji} A_{ji} \dot{z}_i] \right) + \sum_{i=1}^n [\omega_i^T \dot{\omega}_i] \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n [2\varphi_{ij} A_{ij} \dot{z}_j] + \sum_{i=1}^n [\omega_i^T \dot{\omega}_i] \\ &= \sum_{i=1}^n \sum_{j=1}^n [\varphi_{ij} A_{ij} \dot{z}_j] + \sum_{i=1}^n [\omega_i^T \dot{\omega}_i] \end{aligned} \quad (3.21)$$

Si reemplazamos la dinámica de los robots dada en (3.16) y  $\dot{\omega}_i = M_i \varphi_i$  en la ecuación (3.22), se tiene

$$\begin{aligned} \dot{V} &= \sum_{i=1}^n \sum_{j=1}^n [\varphi_{ij} A_{ij} K_j (\varrho_p \varphi_j + \varrho_e \omega_j)] + \sum_{i=1}^n [\omega_i^T M_i \varphi_i] \\ &= \varrho_p \sum_{i=1}^n \sum_{j=1}^n [\varphi_{ij} A_{ij} K_j \varphi_j] + \varrho_e \sum_{i=1}^n \sum_{j=1}^n [\varphi_{ij} A_{ij} K_j \omega_j] + \sum_{i=1}^n [\omega_i^T M_i \varphi_i] \\ &= \varrho_p \sum_{j=1}^n \left[ \left( \sum_{i=1}^n [\varphi_{ij} A_{ij}] \right) K_j \varphi_j \right] + \varrho_e \sum_{j=1}^n \left[ \left( \sum_{i=1}^n [\varphi_{ij} A_{ij}] \right) K_j \omega_j \right] \\ &+ \sum_{i=1}^n [\omega_i^T M_i \varphi_i] \end{aligned} \quad (3.22)$$

Note que el término simétrico  $\sum_{i=1}^n [\varphi_{ij} A_{ij}] = \sum_{i=1}^n [\varphi_{ji} A_{ij}]$  puede ser compactado a

$$\begin{aligned} \sum_{i=1}^n [\varphi_{ji} A_{ij}] &= [\varphi_{j1} \quad \varphi_{j2} \quad \dots \quad \varphi_{jn}] \begin{bmatrix} A_{1j} \\ A_{2j} \\ \vdots \\ A_{nj} \end{bmatrix} \\ &= -[\varphi_{j1} \quad \varphi_{j2} \quad \dots \quad \varphi_{jn}] \begin{bmatrix} A_{j1} \\ A_{j2} \\ \vdots \\ A_{jn} \end{bmatrix} = -\varphi_j^T A_j \end{aligned} \quad (3.23)$$

y la ecuación (3.23) se reduce a

$$\dot{V} = -\varrho_p \sum_{j=1}^n [\varphi_j^T A_j K_j \varphi_j] - \varrho_e \sum_{j=1}^n [\varphi_j^T A_j K_j \omega_j] + \sum_{i=1}^n [\omega_i^T M_i \varphi_i] \quad (3.24)$$

Cambiando los subíndices  $j$  por  $i$  en los primeros dos términos de la ecuación (3.24) y usando el valor de  $K_i$  propuesto en (3.14), resulta que

$$\dot{V} = -\varrho_p \sum_{i=1}^n [\varphi_i^T A_i A_i^T \varphi_i] - \varrho_e \sum_{i=1}^n [\varphi_i^T A_i A_i^T \omega_i] + \sum_{i=1}^n [\omega_i^T M_i \varphi_i] \quad (3.25)$$

Observando el término

$$\varphi_i^T A_i A_i^T \varphi_i = (A_i^T \varphi_i)^T (A_i^T \varphi_i) = \|A_i^T \varphi_i\|^2 \quad (3.26)$$

se puede reemplazar el valor  $M_i$  mostrado en (3.15) y usando (3.26), la ecuación (3.25) cambia a

$$\begin{aligned} \dot{V} &= -\varrho_p \sum_{i=1}^n \|A_i^T \varphi_i\|^2 - \varrho_e \sum_{i=1}^n [\varphi_i^T A_i A_i^T \omega_i] + \varrho_e \sum_{i=1}^n [\omega_i^T A_i A_i^T \varphi_i] \quad (3.27) \\ &= -\varrho_p \sum_{i=1}^n \|A_i^T \varphi_i\|^2 - \varrho_e \sum_{i=1}^n [\varphi_i^T A_i A_i^T \omega_i] + \varrho_e \sum_{i=1}^n [\varphi_i^T A_i A_i^T \omega_i]^T \end{aligned}$$

Note que los últimos dos términos de sumatoria en el lado derecho se cancelan mutuamente debido a que producen el mismo valor real. Consecuentemente, la derivada con respecto al tiempo de la función de Lyapunov es la función semidefinida negativa dada por

$$\dot{V} = -\varrho_p \sum_{i=1}^n \|A_i^T \varphi_i\|^2 \quad (3.28)$$

Note que la ley de control (3.12) es un control proporcional-integral modificado donde las ganancias de las partes proporcional e integral son funciones que dependen de la información de distancia y ángulo. Aunado a esto, la adición de términos  $a_{ij}$  en (3.13) elimina las componentes del error de distancia entre aquellos robots que no poseen comunicación entre ellos.

Observe que  $\dot{V} = 0$ , solo cuando se satisface de forma simultánea que  $A_i^T \varphi_i = 0$ ,  $i = 1, \dots, n$  independientemente de los valores de  $\omega_i$ ,  $i = 1, \dots, n$ . Por la suposición inicial del Teorema 1 mostrado en 3.2, el único valor posible de equilibrio es  $\varphi_{ij} = 0$ ,  $i, j = 1, \dots, n$ . Entonces, usando el teorema de invarianza de LaSalle [87], se tiene estabilidad asintótica en este equilibrio. Por lo tanto, los robots convergen a la distancia deseada.

Por otro lado, la función de Lyapunov se incrementa infinitamente cuando  $\beta_{ij} \rightarrow 0$ . Debido a que los robots no colisionan en  $t = 0$  por la suposición inicial y  $\dot{V}$  es semidefinida negativa, es claro que los robots pueden acercarse, pero  $\beta_{ij} \forall i, j \in N$  nunca será cero, entonces los robots no colisionan entre sí.

**Observación 1** Es evidente que un equilibrio no deseado aparece en el caso de que los robots se encuentran colineales. Para el análisis de esta situación, asuma que los robots se encuentran colineales y orientados en el eje de referencia. En este caso, los posibles valores de los ángulos  $\theta_{ij}$  son 0 o  $\pm\pi$ . Por lo que, la segunda columna de las matrices  $A_i$  que contiene solo funciones senoidales se hace cero. Entonces,  $\text{rank}(A_i^T) = 1, \forall i$ , implica que existe  $\varphi_i \neq 0, \forall i$  tal que  $A_i^T \varphi_i = 0$  generando configuraciones de equilibrios no deseados. La suposición inicial restringe esta situación, garantizando que no se encuentren colineales en condiciones iniciales. De cualquier forma, se requiere un estudio detallado de esta situación en trabajos futuros.

**Observación 2** Note que si  $\varrho_e = 0$ , el término integral desaparece. En este caso, la derivada respecto al tiempo de la función de Lyapunov es equivalente. Sin embargo, la relevancia del término integral aparece en la práctica durante la implementación, donde el término integral ayuda a la eliminación de la zona muerta en los actuadores para el caso de valores muy pequeños en la ley de control. La estructura de este controlador evita el uso de normalización en las leyes de control que podría modificar el desempeño original de la ley propuesta.

### 3.3. Simulaciones numéricas

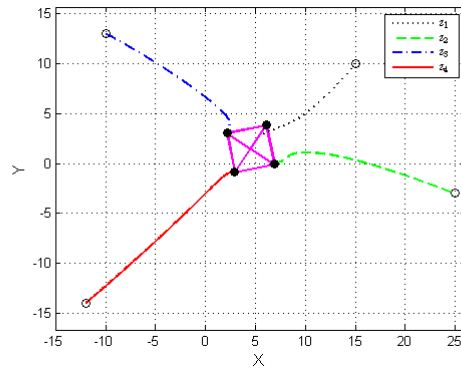
Se lleva a cabo una simulación numérica del sistema en lazo cerrado (3.1)-(3.12) donde  $n = 4$ ,  $\varrho_p = 0.2$  y  $\varrho_e = 0.005$  los resultados se muestran en la figura 3.2. El grafo de conectividad está dado por

$$\mathcal{L}(G) = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix} \quad (3.29)$$

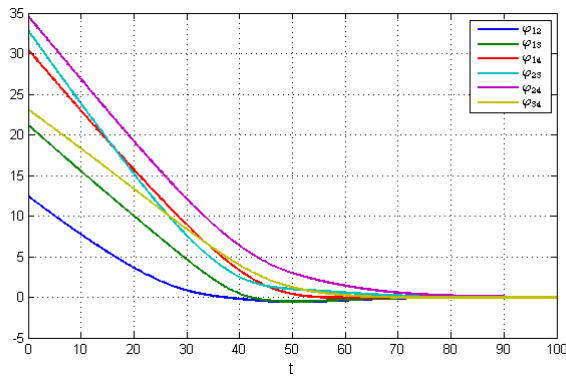
donde las distancias relativas deseadas son  $d_{12} = d_{21} = 4$ ,  $d_{13} = d_{31} = 4$ ,  $d_{14} = d_{41} = 5.6569$ ,  $d_{23} = d_{32} = 5.6569$ ,  $d_{14} = d_{24} = 4$  y  $d_{34} = d_{43} = 4$ . La función potencial atractiva repulsiva esta dada por la ecuación (3.5). Las posiciones iniciales de los robots son  $z_{10} = [15, 10]$ ,  $z_{20} = [25, -3]$ ,  $z_{30} = [-10, 13]$  y  $z_{40} = [-12, -14]$ .

La convergencia de los robots a las distancias deseadas en el plano se muestran en la figura 3.2a, que se comprueba por la convergencia a cero de los errores de distancia  $\varphi_{ij}$  mostrados en la figura 3.2b. Se muestran las entradas de control en la figura 3.2c y finalmente la función de Lyapunov decreciente se muestra en la figura 3.2d.

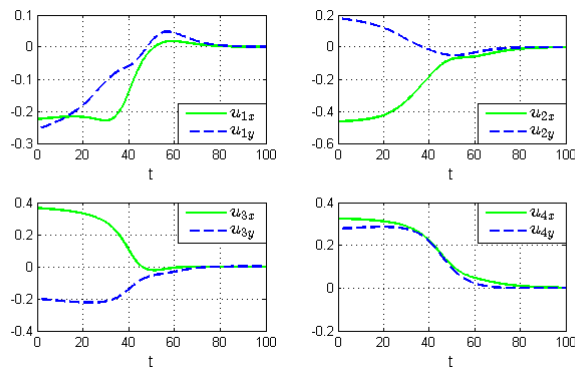




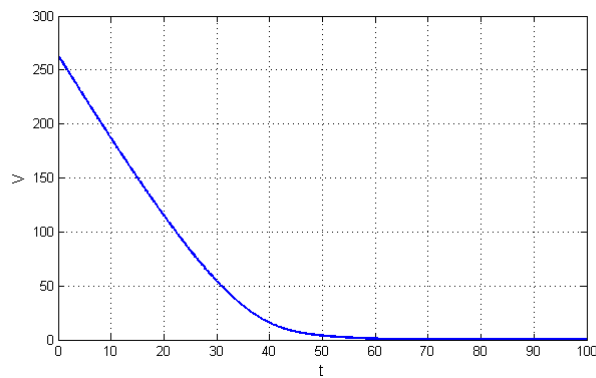
(a) Trayectorias



(b) Errores de distancia



(c) Entradas de control



(d) Función de Lyapunov

Figura 3.2: Simulación numérica usando 4 robots

## 3.4. Experimentos

Para realizar los experimentos se usó la plataforma detallada en la sección 5.1. Esta plataforma se basa en el uso de robots unicyclos IRobot® modelo Create, por lo que a continuación se explica la extensión del algoritmo al caso de robots no holonómicos del tipo unicyclo.

La ley de control propuesta se extiende al caso de robot no holonómicos, considerando el caso del robot unicyclo mostrado en el apartado 2.2.8 con la dinámica del punto  $\alpha_i$  dada por

$$\dot{\alpha}_i = A_i(\theta_i) \begin{bmatrix} v_i \\ w_i \end{bmatrix}, i = 1, \dots, n \quad (3.30)$$

lo que permite diseñar una estrategia de control usando la posición de  $\alpha_i$ , a través de la linealización por retroalimentación dada por

$$\begin{bmatrix} v_i \\ w_i \end{bmatrix} = A_i^{-1}(\theta_i) \dot{\alpha}_{id}, i = 1, \dots, n \quad (3.31)$$

Entonces, la ley basada en distancias definida en (3.12) puede ser extendida a la dinámica de las coordenadas  $\alpha_i$  dadas en (3.30) para cada unicyclo usando

$$\begin{bmatrix} v_i \\ w_i \end{bmatrix} = \frac{1}{2} A_i^{-1}(\theta_i) (K_i (\varrho_p \tilde{\varphi}_i + \varrho_e \tilde{\omega}_i)) \quad (3.32)$$

con  $\tilde{\varphi}_i$  y  $\tilde{\omega}_i$  similares a las funciones  $\varphi_i$  y  $\omega_i$  definidas en (3.12) pero en función de las coordenadas del punto  $\alpha_i$ . Cabe mencionar que el sistema en lazo cerrado (3.30)-(3.32) es reducido al caso de robots puntuales estudiado en la sección 3.2. Consecuentemente, la ley de control (3.32) maniobra las coordenadas  $\alpha_i$  a las distancias deseadas evitando colisiones entre robots. Sin embargo, esta linealización entrada-salida deja los ángulos de orientación  $\theta_i$  sin controlar.

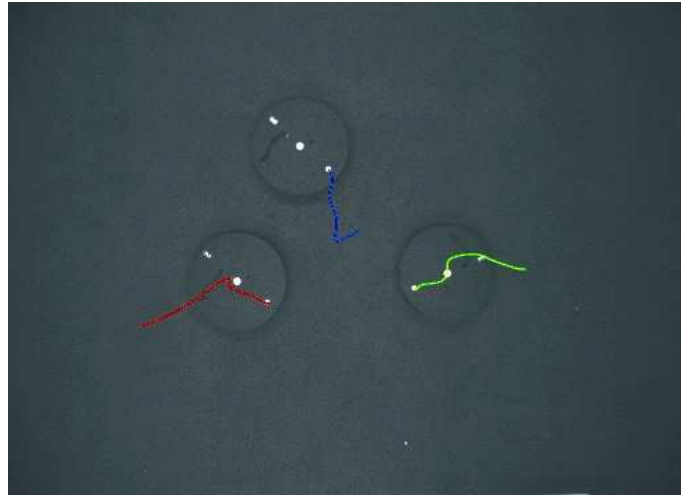
### 3.4.1. Resultados experimentales

El primer experimento mostrado en la figura 3.3 para el sistema en lazo cerrado mostrado en (3.30)-(3.32) tiene  $n = 3$ ,  $\varrho_p = 0.3$  y  $\varrho_e = 0.0$ . Note que el término integral tiene un valor nulo y por lo tanto no tiene efecto en la ley de control. La función potencial atractiva repulsiva basada en distancias está dada por (3.5) y el grafo de conectividad basado en distancia es

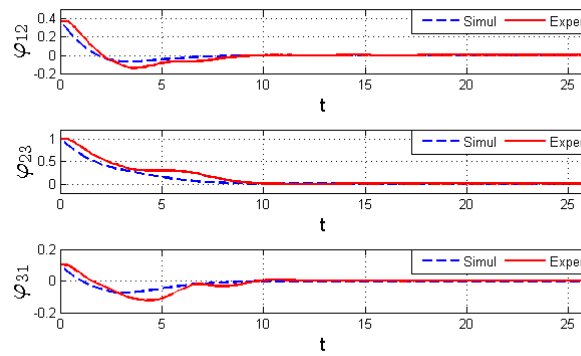
$$\mathcal{L}(G) = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad (3.33)$$

Las distancias relativas deseadas son  $d_{12} = d_{21} = 0.6m$ ,  $d_{13} = d_{31} = 0.6m$ ,  $d_{14} = d_{41} = 0.6m$ ,  $d_{24} = d_{42} = 0.6m$  y  $d_{34} = d_{43} = 0.6m$ . Las condiciones iniciales en metros son  $z_1(0) = [1.3253, 1.2010]$ ,  $z_2(0) = [0.6691, 0.7919]$ ,  $z_3(0) = [2.1410, 1.0912]$  y los ángulos iniciales en radianes  $\theta_1(0) = -0.7149$ ,  $\theta_2(0) = -2.5747$  y  $\theta_3(0) = -1.7126$ .

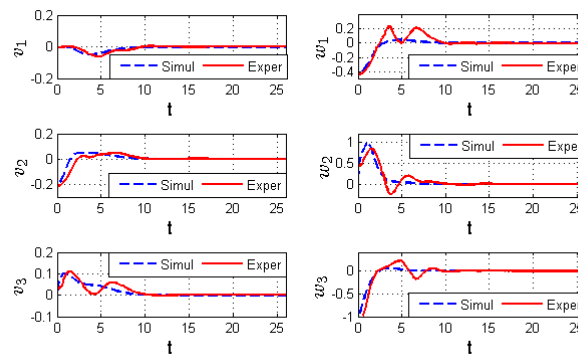
Las trayectorias de las coordenadas  $\alpha_i$ , grabadas por el sistema de visión, se muestran en la figura 3.3a que converge asintóticamente a las distancias relativas deseadas. Esto se prueba en la convergencia a cero de los errores de distancia mostrados en la figura 3.3b. Finalmente, la figura 3.3c muestra las entradas de control lineal y angular para cada robot.



(a) Trayectorias de los robots en el plano



(b) Errores de distancia



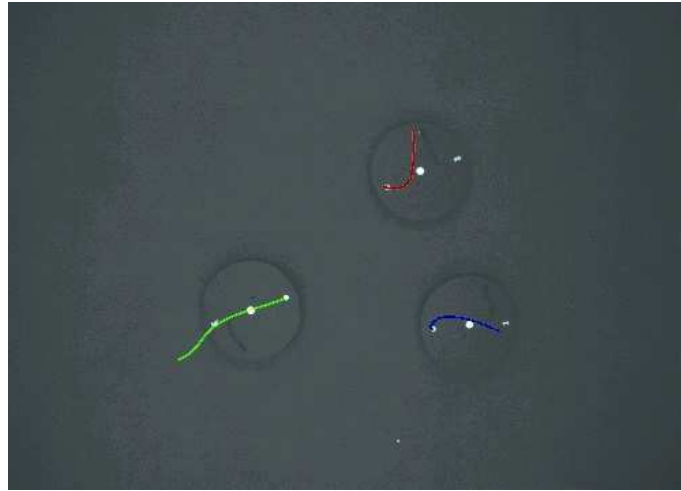
(c) Velocidades lineales y angulares

Figura 3.3: Experimento uno con  $\varrho_p = 0.3$  y  $\varrho_e = 0$

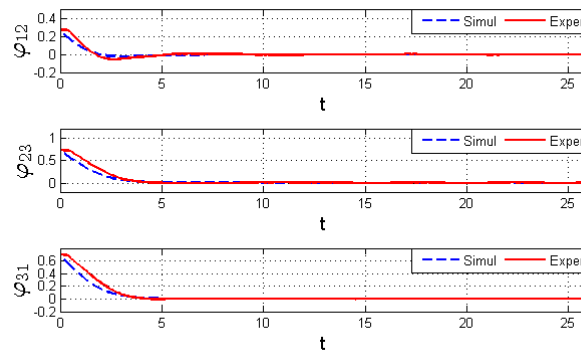
Un segundo experimento se presenta en la figura 3.4, ahora con  $\varrho_p = 0.3$  y  $\varrho_e = 0.005$ , es decir, con el efecto de los términos integrales. Conservando las distancias relativas deseadas y la misma función potencial atractiva repulsiva basada en distancias que en el experimento anterior. Las posiciones iniciales de cada robot son  $z_1(0) = [2.1603, 0.6028]$ ,  $z_2(0) = [1.8245, 1.4127]$ ,  $z_3(0) = [0.6212, 0.6241]$  y los ángulos  $\theta_1(0) = 2.6070$ ,  $\theta_2(0) = 2.6318$  y  $\theta_3(0) = -0.5179$ . La figura 3.4 muestra las trayectorias de los robots, la convergencia de los errores de distancia y las entradas de control. Note que las leyes de control cumplen con los objetivos de control.

### **Análisis de los resultados**

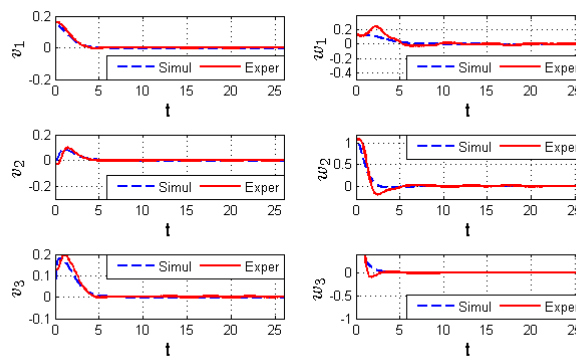
La definición del problema garantiza que los agentes no requieren la posición global para asegurar el consenso. El acercamiento de control propuesto es una ley proporcional e integral modificada que usa una familia de funciones potenciales atractivas repulsivas basadas en distancias, que usa únicamente distancia y orientación relativa. El término integral mejora la convergencia del error de distancia eliminando la zona muerta de los motores. La comparación entre las figuras 3.3c y 3.4c demuestra un mejor desempeño de la velocidad angular y lineal cuando el término integral se implementa. Por último, se pueden notar algunas diferencias entre la simulación y los experimentos, estas se deben a restricciones dinámicas no consideradas en el modelo, como efectos inerciales, fricción en ruedas, y resolución de la cámara. Sin embargo la simulación y los experimentos son muy próximos en ambos experimentos.



(a) Trayectorias de los robots en el plano



(b) Errores de distancia



(c) Velocidades lineales y angulares

Figura 3.4: Experimento dos con  $\rho_p = 0.3$  y  $\rho_e = 0.005$ .

# Capítulo 4

## Formación de robots heterogéneos basada en distancia y orientación

En esta sección se plantea una estrategia de control planar para robots móviles de primer y segundo orden. Usando funciones potenciales atractivo-repulsivas basadas en distancias (F-PARD) y orientación. Aplicable a topologías de comunicación no dirigidas (bidireccionales), la estrategia de control es descentralizada. Además el resultado se aplica a robots no holonómicos usando linealización entrada-salida. Esto permite la simulación e implementación con robots heterogéneos. Todo lo referente a este capítulo se presentó en el congreso *Multi-Conference on Systems and Control* del *IEEE*, en Argentina Buenos Aires, del 19 al 22 de septiembre de 2016 referenciado en [61].

### 4.1. Formación basada en distancias y orientaciones para robots de primer y segundo orden

Tomando las condiciones antes presentadas en la sección 2.2.4, considere un grupo de robots compuesto por agentes de primer y segundo orden, definidos por la ecuación (2.14), con  $n_1$  robots de integrador simple y  $n_2$  robots de integrador doble, con  $n = n_1 + n_2$ , es decir

$$\dot{\vec{z}}_i = u_i, \quad i = 1, \dots, n_1 \quad (4.1)$$

$$\dot{\vec{z}}_i = p_i, \quad \dot{p}_i = u_i, \quad i = (n_1 + 1), \dots, (n_1 + n_2) \quad (4.2)$$

donde  $u_i \in \mathbb{R}^2$  es el vector de entrada del robot  $R_i$ .

Se tiene un grafo de comunicación, no dirigido, con árbol de expansión directo y conexo  $\mathcal{G} = (\mathcal{V}_n, \mathcal{E}_n, \mathcal{D}_n)$  definido en el apartado 2.2.2 y ampliado en la sección 3.1.

Sabiendo que la distancia entre dos robots es  $\beta_{ij} = \beta_{ji} = \|\vec{z}_j - \vec{z}_i\|$  y el ángulo  $\theta_{ij}$  medido hasta  $\beta_{ij}$  con respecto a un eje arbitrario como se muestra en la figura 3.1. Por simetría  $\theta_{ji} = \pi + \theta_{ij}$ . La derivada de  $\beta_{ij}$  es

$$\dot{\beta}_{ij} = \frac{(\vec{z}_j - \vec{z}_i)^T (\dot{\vec{z}}_j - \dot{\vec{z}}_i)}{\beta_{ij}} \quad (4.3)$$

Y considerando la forma polar  $[\vec{z}_j - \vec{z}_i]^T = \beta_{ij} [\cos \theta_{ij}, \sin \theta_{ij}]$ , la ecuación (4.3)

se puede escribir como

$$\dot{\beta}_{ij} = [\cos \theta_{ij}, \sin \theta_{ij}] \left( \dot{z}_j - \dot{z}_i \right) \quad (4.4)$$

**Planteamiento** El objetivo de control es diseñar una ley de control  $u_i$ , para todo  $R_i$ ,  $i = 1, \dots, n$ , que usando los valores de distancia entre robots  $\beta_{ij}$  y ángulo de orientación  $\theta_{ij}$   $\forall j \in N_i$ , llegue a consenso a las distancias deseadas  $d_{ij}$  tal que  $\lim_{t \rightarrow \infty} \beta_{ij} - d_{ij} = 0$ ,  $\forall j \in N_i$  evitando colisiones entre robots, es decir  $\beta_{ij} \neq 0$ ,  $\forall i, j \in N$  con la posibilidad de controlar robots no holonómicos heterogéneos.

Considerando las funciones potenciales atractivas definidas en la parte 3.2, la ley de control (3.12) puede ser ampliada al caso combinado de robots de primer y segundo orden a través del siguiente teorema.

**Teorema 2** Considere el grupo de robots definidos por las ecuaciones (4.1) y (4.2), comunicados por un grafo  $\mathcal{G} = (\mathcal{V}_n, \mathcal{E}_n, \mathcal{D}_n)$  basado en distancias, conexo y no dirigido y una F-PARD específica. Defina la ley de control como

$$u_i = \varrho_p A_i^T \varphi_i, i = 1, \dots, n_1 \quad (4.5)$$

$$u_i = A_i^T \varphi_i - \varrho_d p_i, i = (n_1 + 1), \dots, (n_1 + n_2) \quad (4.6)$$

donde  $\varrho_p \in \mathbb{R}$ ,  $\varrho_d \in \mathbb{R}$ ,  $\varrho_p > 0$ ,  $\varrho_d > 0$  son ganancias constantes,  $A_i = [A_{i1}, \dots, A_{in}]^T \in \mathbb{R}^{n \times 2}$ ,  $i = 1, \dots, n$ ,  $\varphi_i = [a_{i1}\varphi_{i1}, \dots, a_{in}\varphi_{in}]^T$  con  $a_{ij}$  definido en la sección 2.2.2 y  $p_i$  la velocidad variable de los robots de segundo orden definidos en (4.2).

Suponga que los robots no son colineales ni colisionan en  $t = 0$ , y el equilibrio ocurre solo cuando  $\varphi_{ij} = 0$ ,  $i, j = 1, \dots, n$ . Entonces, en el sistema en lazo cerrado (4.1), (4.2), (4.5) y (4.6), los robots convergen a la formación deseada, es decir  $\lim_{t \rightarrow \infty} \varphi_{ij} = 0$ ,  $\forall j \in N_i$  evitando colisiones entre robots, o sea  $\beta_{ij} > 0$ ,  $\forall i \neq j$ ,  $\forall t > 0$ .

Note que en las leyes de control (4.5) y (4.6), la adición de los términos  $a_{ij}$  en los vectores  $\varphi_i$ ,  $i = 1, \dots, n$  aniquila las componentes del error de distancia entre robots que no poseen comunicación en el grafo de formación. También, a manera de simplificación, el término integral incorporado en (3.12) no es considerado en este análisis.

**Prueba** Considere una función candidata de Lyapunov

$$V = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n V_{ij} + \frac{1}{2} \sum_{i=n_1+1}^n [p_i^T p_i] \quad (4.7)$$

donde  $V_{ij}$  es la F-PARD semidefinida positiva. Note que  $V$  siempre es positiva, y se desvanece en el equilibrio deseado, esto ocurre solo cuando  $\varphi_{ij} = 0$ ,  $\forall j \in N_i$  y  $p_i = 0$  para los robots de segundo orden. La derivada con respecto al tiempo de  $V$  es

$$\dot{V} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \dot{V}_{ij} + \sum_{i=n_1+1}^n [p_i^T \dot{p}_i] \quad (4.8)$$

por lo que, sustituyendo (3.10) en (4.8),

$$\dot{V} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \left[ \varphi_{ij} A_{ij} (\dot{z}_j - \dot{z}_i) \right] + \sum_{i=n_1+1}^n [p_i^T \dot{p}_i] \quad (4.9)$$

$$\begin{aligned} \dot{V} &= \frac{1}{2} \left( \sum_{i=1}^n \sum_{j=1}^n \left[ \varphi_{ij} A_{ij} \dot{z}_j \right] - \sum_{i=1}^n \sum_{j=1}^n \left[ \varphi_{ij} A_{ij} \dot{z}_i \right] \right) \\ &+ \sum_{i=n_1+1}^n [p_i^T \dot{p}_i] \end{aligned} \quad (4.10)$$

como el grafo de comunicación es no dirigido,  $\varphi_{ij} = \varphi_{ji}$  y  $A_{ij} = -A_{ji}$ , entonces  $\dot{V}$  puede reducirse a

$$\begin{aligned} \dot{V} &= \frac{1}{2} \left( \sum_{i=1}^n \sum_{j=1}^n \left[ \varphi_{ij} A_{ij} \dot{z}_j \right] + \sum_{j=1}^n \sum_{i=1}^n \left[ \varphi_{ji} A_{ji} \dot{z}_i \right] \right) \\ &+ \sum_{i=n_1+1}^n [p_i^T \dot{p}_i] \end{aligned} \quad (4.11)$$

$$\begin{aligned} &= \frac{1}{2} \left( 2 \sum_{i=1}^n \sum_{j=1}^n \left[ \varphi_{ij} A_{ij} \dot{z}_j \right] \right) + \sum_{i=n_1+1}^n [p_i^T \dot{p}_i] \\ &= \sum_{j=1}^n \left[ \sum_{i=1}^n (\varphi_{ji} A_{ij}) \dot{z}_j \right] + \sum_{i=n_1+1}^n [p_i^T \dot{p}_i] \end{aligned} \quad (4.12)$$

advértase que  $\sum_{i=1}^n (\varphi_{ji} A_{ij}) = -\sum_{i=1}^n (\varphi_{ji} A_{ji}) = -\varphi_j^T A_j$ . En ese caso,

$$\dot{V} = -\sum_{i=1}^n \left[ \varphi_i^T A_i \dot{z}_i \right] + \sum_{i=n_1+1}^n [p_i^T \dot{p}_i] \quad (4.13)$$

reemplazando (4.1), (4.2), (4.5) y (4.6) en (4.13), entonces

$$\begin{aligned} \dot{V} &= -\sum_{i=1}^{n_1} \left[ \varphi_i^T A_i u_i \right] - \sum_{i=n_1+1}^n \left[ \varphi_i^T A_i p_i \right] \\ &+ \sum_{i=n_1+1}^n [p_i^T u_i] \end{aligned} \quad (4.14)$$

sustituyendo las leyes de control (4.5) y (4.6), se vuelve

$$\begin{aligned} \dot{V} &= -\sum_{i=1}^{n_1} \left[ \varphi_i^T A_i (\varrho_p A_i^T \varphi_i) \right] - \sum_{i=n_1+1}^n \left[ \varphi_i^T A_i p_i \right] \\ &+ \sum_{i=n_1+1}^n \left[ p_i^T (A_i^T \varphi_i - \varrho_d p_i) \right] \end{aligned} \quad (4.15)$$

$$\begin{aligned} \dot{V} &= -\varrho_p \sum_{i=1}^{n_1} \left[ (A_i^T \varphi_i)^T (A_i^T \varphi_i) \right] - \sum_{i=n_1+1}^n \left[ p_i^T A_i \varphi_i \right]^T \\ &+ \sum_{i=n_1+1}^n \left[ p_i^T A_i^T \varphi_i \right] - \varrho_d \sum_{i=n_1+1}^n \left[ p_i^T p_i \right] \end{aligned} \quad (4.16)$$



perciba que el segundo y tercer término se cancelan mutuamente por que producen el mismo valor real. Por lo tanto, el valor final de  $\dot{V}$  es

$$\dot{V} = -\varrho_p \sum_{i=1}^{n_1} \|A_i^T \varphi_i\|^2 - \varrho_d \sum_{i=n_1+1}^n \|p_i\|^2 \quad (4.17)$$

Note que  $\dot{V}$  es semidefinida negativa. Dada la suposición inicial del teorema 2 dado en 4.1, la única posibilidad de equilibrio sucede cuando  $\varphi_i = 0$ , para  $i = 1, \dots, n_1$  y también  $p_i = 0$ , para  $i = (n_1 + 1), \dots, (n_1 + n_2)$ . Para estos términos secundarios, usando el teorema de invarianza de LaSalle, vease [87], observe que  $\dot{z}_i = p_i = 0$ , luego los valores de  $z_i$ ,  $i = (n_1 + 1), \dots, (n_1 + n_2)$  convergen a un valor constante. Además  $p_i = 0$  obliga a que  $\dot{p}_i = u_i = A_i^T \varphi_i = 0$ ,  $i = (n_1 + 1), \dots, (n_1 + n_2)$ . En consecuencia, la única posibilidad de equilibrio se da cuando todos los errores de distancia son cero, es decir, cuando  $\varphi_i = 0$ ,  $i = 1, \dots, n$ . Entonces se tiene estabilidad asintótica a estos equilibrios que representan la convergencia de los robots a la formación deseada.

En otro aspecto, la función de Liapunov se incrementa de forma infinita cuando cualquier  $\beta_{ij} \rightarrow 0$ . Como los robots no colisionan en  $t = 0$  debido a la suposición inicial y  $\dot{V}$  es semidefinida negativa, es claro que los robots pueden estar cercanos, pero  $\beta_{ij} \forall i, j \in N$  nunca será igual a cero, y por lo tanto los robots no colisionan.

**Observación 3** Finalmente, un equilibrio no deseado aparece en el caso en que  $A_i^T \varphi_i = 0$ , para  $i = 1, \dots, n_1$ , y  $\varphi_i \neq 0$ . Esto sucede cuando los robots son colineales. De cualquier forma, dadas las suposiciones iniciales los robots no se encuentran inicialmente atrapados en este mínimo local.

## 4.2. Simulaciones numéricas

El enfoque de control se extendió a robots no holonómicos implementando una linealización entrada-salida. Esto muestra que las leyes de formación basada en distancias pueden extenderse al caso heterogéneo, principal objetivo de esta tesis. Para las simulaciones realizadas se usan dos robots de primer orden  $n_1 = 2$ , uno omnidireccional y un unicycle, y uno de segundo orden  $n_2 = 1$ , un cuadricóptero. En la sección 2.2.8 se muestran a detalle los modelos cinemáticos de postura de los dos robots con ruedas y el modelo dinámico simplificado del cuadricóptero.

### 4.2.1. Resultados

Usando los modelos del unicycle, omnidireccional y cuadricóptero, se realizó una formación con robots heterogéneos, usando las leyes de control (4.5) y (4.6). Con un grafo completo de comunicación se tiene que  $N_1 = \{z_2, z_3\}$ ,  $N_2 = \{z_1, z_3\}$  y  $N_3 = \{z_2, z_3\}$ . La F-PARD es  $V_{ji} = \frac{(\beta_{12} - d_{12})^2}{2\beta_{12}}$ , que deviene en  $\Gamma_{ji} = \frac{\beta_{12} + d_{12}}{2\beta_{12}^2}$ . Los valores de las ganancias son  $\varrho_p = 1$ ,  $\varrho_d = 1$  y las distancias deseadas  $d_{12} = d_{23} = d_{31} = 5$ , diseñadas para formar un triángulo equilátero. Las posiciones iniciales de los robots son  $z_1(0) = [11, 10]$ ,  $z_2(0) = [10, 10]$ ,  $z_3(0) = [-10, -10]$ . El resto de las condiciones iniciales de las variables no usadas en el control de formación son  $\theta_2(0) = 0$ , para el robot omnidireccional, y  $z_3(0) = 0$ ,  $\eta_3(0) = [0, 0, 0]^T$  para el cuadricóptero. La ley de control para la coordenada  $z_3$  es  $u_{3z} = -k_p(z_3 - z_3^*)$ , con  $z_3^* = 6$  como la altura

deseada y  $\psi_3^* = 0$ . se puede observar en la figura 4.1a que los robots convergen a la formación deseada, y el cuadricóptero llega a la altura deseada. Se muestra la convergencia de los errores de distancia  $\varphi_{ij}$  en la figura 4.1b. Finalmente, las entradas de control generadas por el algoritmo de formación se muestran en la figura 4.1c.

### 4.3. Trabajo experimental

Se realizaron experimentos usando dos robots, uno tipo unicycle y otro tipo omnidireccional, los diseños y características de la plataforma se presenta a detalle en el capítulo 5.

Las ganancias de control se propusieron como  $\rho_p = 1$  usando la misma F-PARD de las simulaciones. Con las distancias deseadas siendo  $d_{12} = d_{23} = d_{31} = 0.6m$ . Para el primer robot,  $R_1$  tipo unicycle mostrado en la figura 2.10,  $l = 5cm$  y las velocidades de las ruedas como se muestra en (2.2.8) se calculan como

$$\begin{bmatrix} \omega_{rd} \\ \omega_{ri} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & \frac{L}{2} \\ 1 & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} \quad (4.18)$$

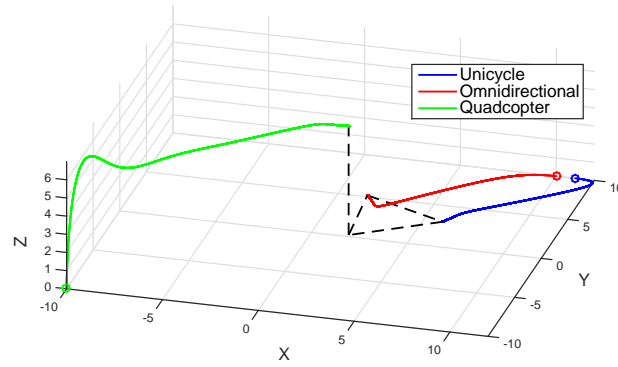
con  $L = 17.5cm$  y  $r = 2.5cm$ . Para el segundo robot  $R_2$ , tipo omnidireccional mostrado en la figura 2.12, como se muestra en (2.2.8), las velocidades de cada rueda se calculan como

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 1 & -(L+l) \\ -1 & 1 & (L+l) \\ -1 & 1 & -(L+l) \\ 1 & 1 & (L+l) \end{bmatrix} \begin{bmatrix} v_{x2} \\ v_{y2} \\ \omega_2 \end{bmatrix} \quad (4.19)$$

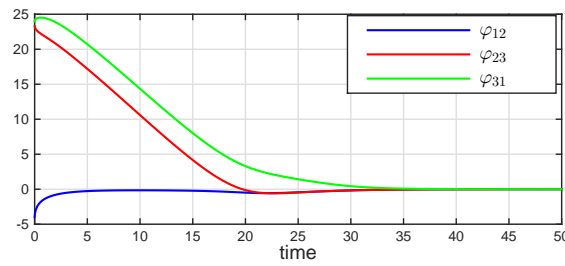
siendo  $L = 10cm$ ,  $l = 5cm$  y  $r = 2.75cm$ . Las posiciones iniciales, medidas en metros y radianes, de los robots son  $[x_1(0), y_1(0), \theta_1(0)] = [-1.4578, 0.2872, -1.5934]$  para el robot unicycle y  $[x_2(0), y_2(0), \theta_2(0)] = [0.9209, 0.2381, -0.0159]$  para el robot omnidireccional. El ángulo de orientación para  $R_2$  se eligió como  $\theta_2^* = 0.5\pi$ .

#### Análisis de los resultados

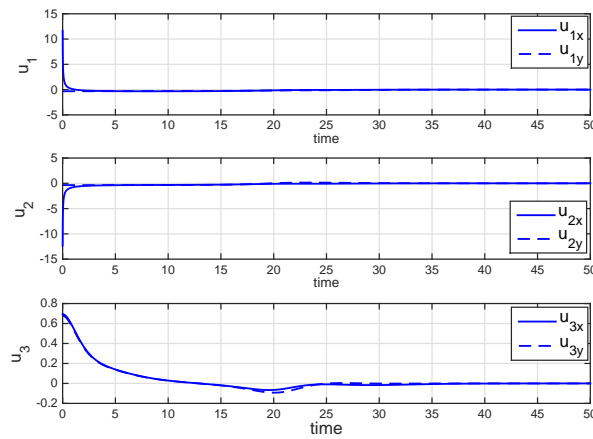
Observe en la figura 4.2a que los robots convergen a la distancia deseada. Se muestra la convergencia del error de distancia  $\varphi_{12}$  en la figura 4.2b. La convergencia de los ángulos de orientación se presenta en la figura 4.2c, donde el ángulo  $\theta_2$  converge a  $\theta_2^*$ . Por último, las entradas de control generadas por el control de formación se muestran en la figura 4.2d. Se normalizó la ley de control en cada robot, para evitar movimientos rápidos, y para eliminar la zona muerta de los actuadores en la vecindad de la formación requerida.



(a) Trayectorias de los robots

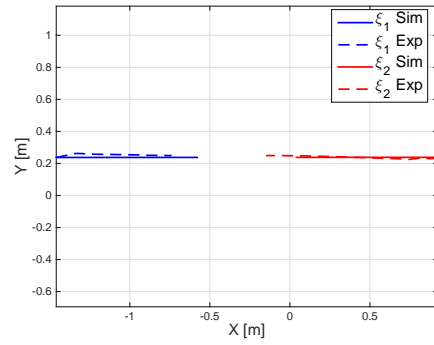


(b) Errores de formación

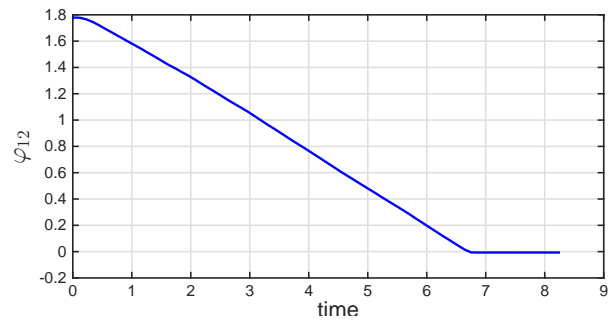


(c) Entradas de control

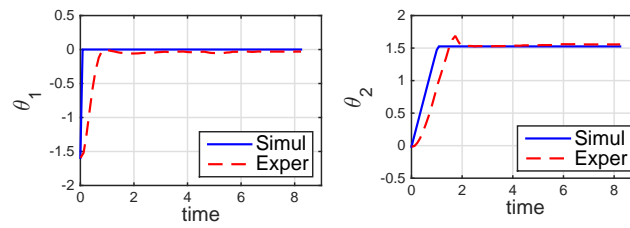
Figura 4.1: Simulación de formación



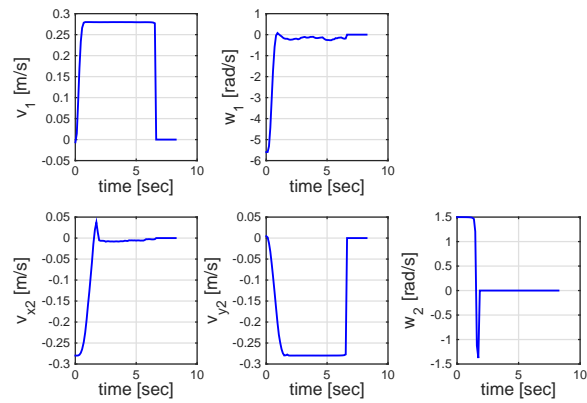
(a) Trayectorias de los robots en el plano



(b) Error de formación



(c) Ángulos de orientación



(d) Entradas de control

Figura 4.2: Experimentos de formación con robot unicycle y omnidireccional

# Capítulo 5

## Plataforma experimental

### 5.1. Plataforma Irobot<sup>®</sup> Create

Para el trabajo experimental, se usa la plataforma experimental mostrada en la figura 5.2, conformado por tres robots tipo unicycle de la marca IRobot<sup>®</sup> modelo Create, diseñados con propósitos de investigación. Los robots cuentan con diferentes sensores locales, puertos de entrada salida que pueden ser conectados a otros dispositivos o sensores. Para la comunicación inalámbrica se agregaron módulos receptores bluetooth por comunicación serial.

Para estimar la posición y orientación de cada robot en el espacio de trabajo, se usó una cámara monocromática instalada en el techo a una distancia de  $2,5m$  modelo Genie HM-1024 con conexión Ethernet manufacturada por Teledyne Dalsa<sup>®</sup>. Dicha cámara trabaja a 117 cuadros por segundo con resolución de  $1024 \times 768$  píxeles. El sistema era capaz de cubrir un área de aproximadamente  $3m \times 2m$  lo que se traduce en una resolución de  $3mm^2$  por pixel. Para aumentar el desempeño de la cámara, el piso del área de trabajo se cubrió con alfombra negra y los robots se cubrieron con placas negras con solo dos círculos blancos, ubicados en las coordenadas  $(x_i, y_i)$  y  $\alpha_i$  dadas en (2.2.8), respectivamente, como se muestra en la figura 5.1.

Finalmente, una PC Core i5 con 4GB RAM realiza el procesamiento de video usando Matlab<sup>®</sup> R2013a para extraer la posición y orientación de cada robot y cierra el lazo enviando por bluetooth las velocidades angulares a las ruedas de los robots.

La plataforma experimental constituye una opción flexible para simular patrones de formación centralizados y descentralizados. Por medio de la adecuada selección de la información arrojada por el sistema de visión se pueden estudiar diferentes topologías de comunicación y leyes de control. Para los experimentos realizados, la medición de distancia es simulada usando la información del sistema de visión, por lo que los sensores locales de proximidad de los robots no fueron usados.

La implementación del algoritmo de control se resume en la figura 5.3. Se asume que la cámara se encontraba calibrada. El primer paso, antes del lazo de control, es adquirir la posición inicial de los marcadores en los robots. Debido a que los marcadores blancos tienen las mismas dimensiones, se asignó una etiqueta a cada uno de forma manual con el ratón o *mouse* de la computadora, desde  $R_1$  hasta  $R_n$ . Una vez terminada la asignación, un algoritmo calcula la posición inicial y la orientación de cada robot definidas por la posición de los marcadores detectados. Durante el resto del experimento, el algoritmo de visión registra la nueva ubicación de los marcadores en regiones predefinidas cercanas a la última posición conocida. Este



Figura 5.1: Plataforma de experimentos con 3 Irobot Create

método permite una adquisición veloz de datos, evitando la búsqueda exhaustiva en el plano entero de la imagen.

El lazo de control se inicializa con las posiciones conocidas de las coordenadas  $(x_i, y_i)$  y  $\alpha_i = (p_i, q_i)$  convertidas a metros y estimando la orientación de los robots a través de

$$\theta_i = \arctan\left(\frac{q_i - y_i}{p_i - x_i}\right), \quad \forall i. \quad (5.1)$$

después, las distancias asociadas a cada robot y sus orientaciones son calculadas por

$$d_{ij} = \|\alpha_j - \alpha_i\|, \quad \forall i, \forall j \in N_i \quad (5.2)$$

$$\theta_{ij} = \arctan\left(\frac{q_i - q_j}{p_i - p_j}\right), \quad \forall j \in N_i. \quad (5.3)$$

posteriormente, el algoritmo de control determina las señales de control  $v_i$  y  $w_i$  de acuerdo a la ecuación (3.32), y las velocidades angulares de las ruedas del robot usando las expresiones

$$\begin{aligned} w_{ri} &= \frac{v_i}{r} + \frac{L}{2r}w_i \\ w_{li} &= \frac{v_i}{r} - \frac{L}{2r}w_i, \end{aligned} \quad (5.4)$$

donde  $w_{ri}$  y  $w_{li}$  son las velocidades derecha e izquierda de cada robot, respectivamente,  $L = 0,25m$  es la distancia entre ruedas del IRobot's y  $r = 0,03m$  el radio de las ruedas. Estas velocidades angulares deseadas se envían vía bluetooth a cada robot. El control local de cada robot regula la velocidad de las ruedas de acuerdo con los valores deseados. El lazo de control puede funcionar de forma indefinida hasta que el usuario termina el experimento, guardando detalles de tiempo, coordenadas, orientación, y entradas de control para cada robot. Estos experimentos son comparados con las simulaciones numéricas en condiciones similares.

## 5.2. Desarrollo de la plataforma de robots móviles heterogéneos

La plataforma de los robots Irobot Create funcionó muy bien para los experimentos de robots homogéneos en un plano, sin embargo el problema de la heterogeneidad

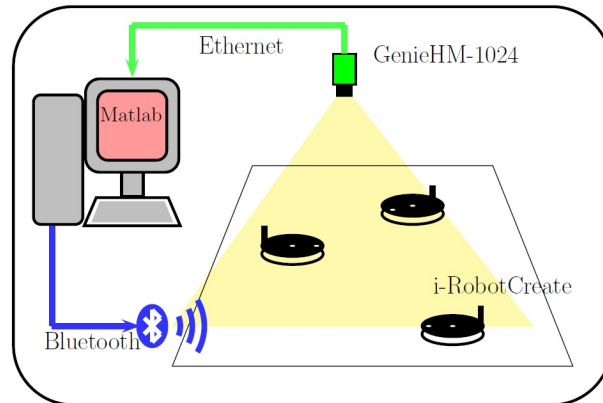


Figura 5.2: Esquema de la plataforma de experimentos

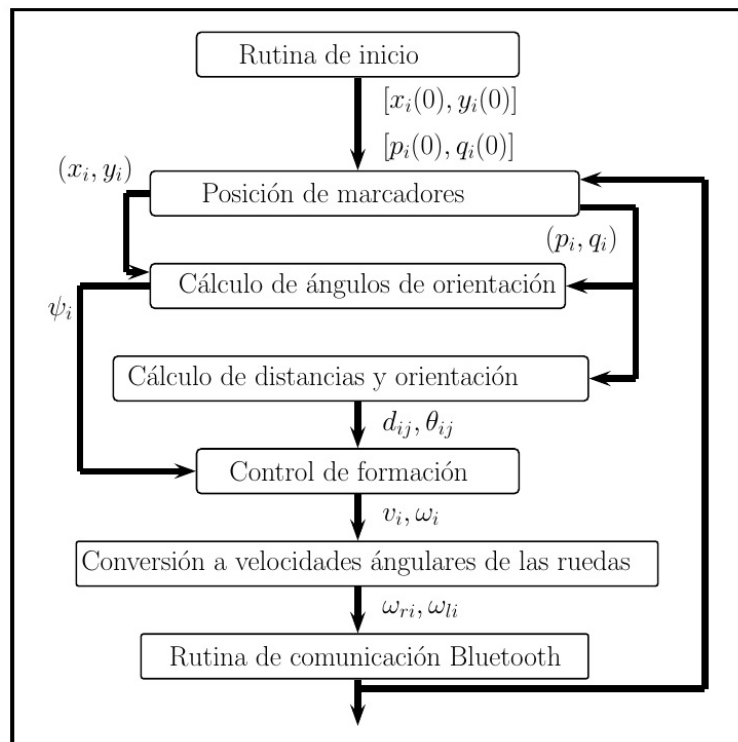


Figura 5.3: Implementación de control

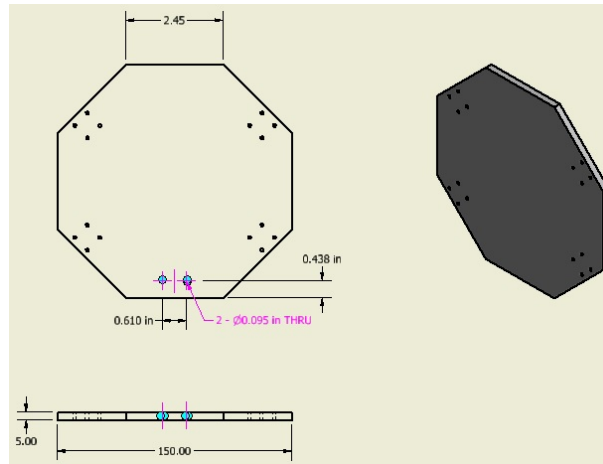


Figura 5.4: Núcleo Octagonal

requería el diseño de una plataforma mas compleja. En este capítulo se detalla la estructura de la plataforma diseñada.

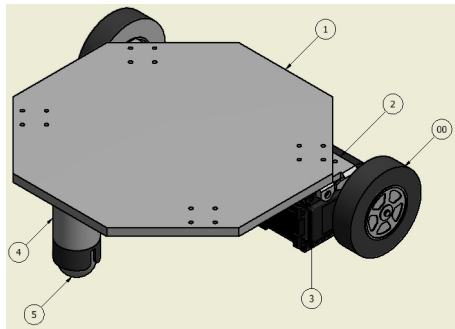
### 5.2.1. Estructura

Durante el diseño de la plataforma se planteó la existencia de un núcleo de control que pudiera mover todos los actuadores y unirse a los diferentes tipos de locomoción de cada robot. Para lograr esto se diseñó un núcleo octagonal mostrado en la figura 5.4 que aloja todos los componentes electrónicos y realiza la función de estructura central de soporte. A este nucleo se adjuntan todos los componentes que corresponden a los robots unicyclo y omnidireccional como se muestra en las figuras 5.5a y 5.5b. Se muestran adicionalmente planos para un posible robot cuadricóptero y un posible cuadrúpedo en las figuras 5.5c y 5.5d respectivamente. La viabilidad de los experimentos con los diseños de cuadrúpedos y quadricópteros se analiza en el capítulo 6.2. Detalles de los componentes para el unicyclo y el omnidireccional se muestran en el apéndice A.

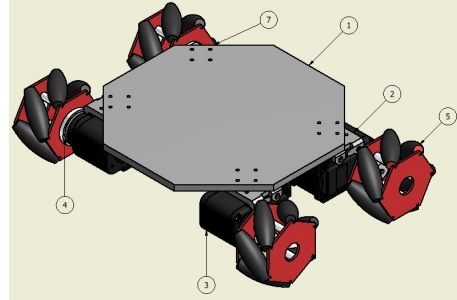
Se hicieron varias pruebas con diferentes materiales para la construcción de los prototipos. Las pruebas no incluyen análisis de cargas o estrés debido a las condiciones de carga mínimas, es decir, los pesos y fuerzas experimentados por la estructura y componentes no fueron significativos como para requerir un análisis mas profundo del comportamiento del material a las fuerzas que actuan sobre éste. Los tres materiales seleccionados fueron, madera MDF, acrílico y Nylon. Debido a su fragilidad la madera MDF se eliminó de forma inmediata. Se fabricaron prototipos funcionales en Nylon, sin embargo, el costo y maquinado requerido orientaron la selección hacia el acrílico. En la figura 5.6a se muestra el robot omnidireccional en Nylon blanco. Los prototipos de omnidireccional y unicyclo en acrílico se muestran en la figura 5.6b.

Los actuadores seleccionados fueron los servomotores Dynamixel<sup>®</sup> AX – 12W, que brindan rotación continua y comunicación serial. Las ruedas del robot unicyclo son las *BIOLOID<sup>®</sup> FP04-F17* con un diámetro de 52mm mostradas en la figura 5.7a. Las ruedas seleccionadas para el robot omnidireccional fueron las *FingerTech Mecanum Wheels* con diámetro de 52mm mostradas en la figura 5.7b.

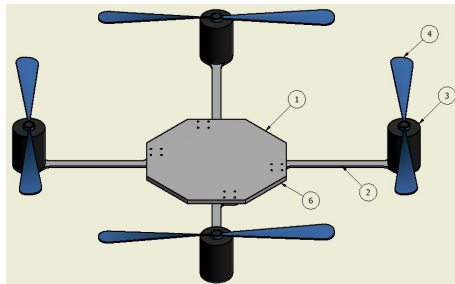




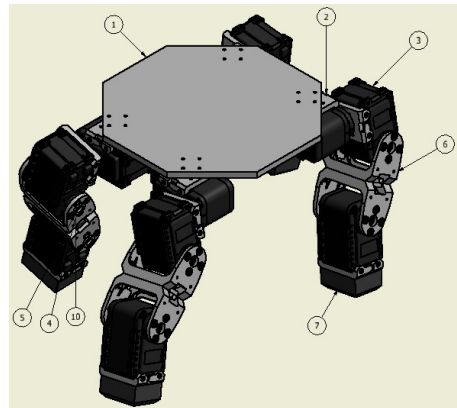
(a) Ensamble de robot uniclo



(b) Ensamble de robot omnidireccional

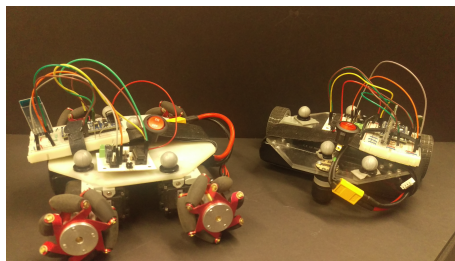


(c) Posible ensamble de robot cuatricóptero



(d) Posible ensamble de robot cuadrúpedo

Figura 5.5: Planos de SRM heterogéneo

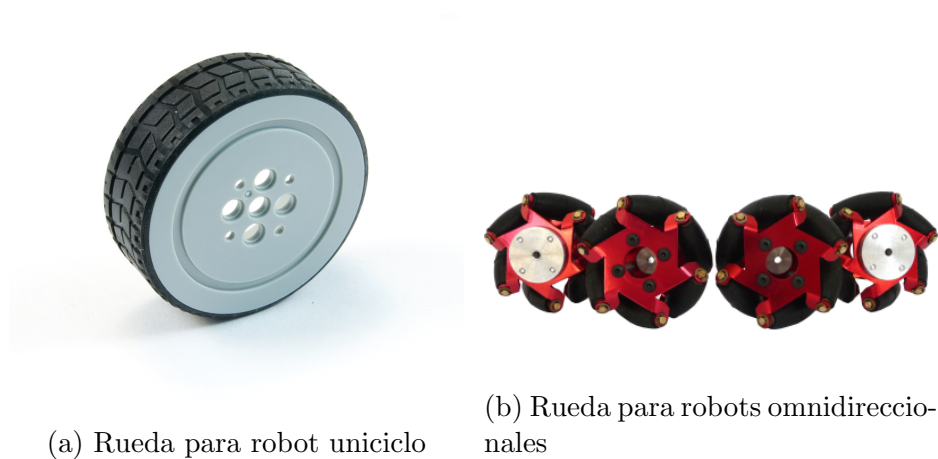


(a) Robot uniclo acrílico y omnidireccional Nylon



(b) Robots uniclo y omnidireccional construidos en acrílico

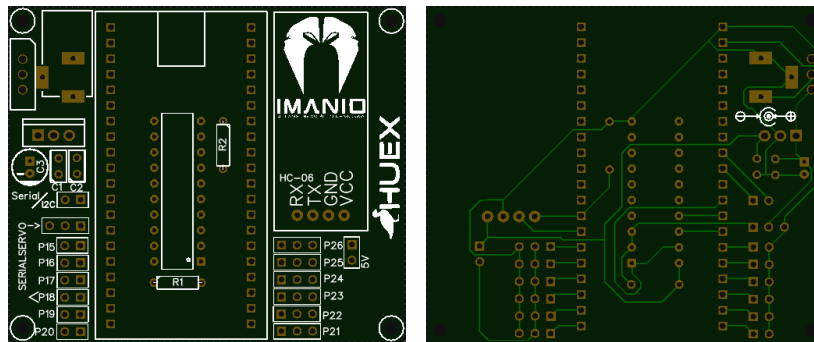
Figura 5.6: Construcción de unicyclos y omnidireccionales



(a) Rueda para robot unicycle

(b) Rueda para robots omnidireccionales

Figura 5.7: Ruedas de unicyclos y omnidireccionales



(a) Vista frontal de la placa

(b) Vista trasera de la placa

Figura 5.8: Placa HUEX

## 5.2.2. Núcleo de control

El núcleo de control es un circuito que puede controlar los servomotores Dynamixel AX – 12W, así como producir PWMs necesarios para cualquier tipo de servomotor o speed control para motores trifásicos. El núcleo de control llamado *HUEX*, del náhuatl *huexocauhtli* que significa pato, incluye la etapa de potencia necesaria para suministrar potencia a los servomotores y a los componentes por medio de un regulador L7805CV. Incluye un buffer SN74LS241N para realizar la comunicación serial del servomotor. Tiene un módulo de comunicación Bluetooth HC-06 y una tarjeta de desarrollo mbed NXP<sup>®</sup> basado en el microcontrolador LPC1768. El esquema completo del circuito se muestra en la figura B.1 correspondiente al apéndice B, en la figura 5.8 se incluye imágenes de la placa manufacturada por la empresa IMANIO.

En la figura 5.6a se muestra la versión experimental del núcleo de control. El núcleo terminado se muestra en la figura 5.9a y montado en un robot omnidireccional en la figura 5.9b.

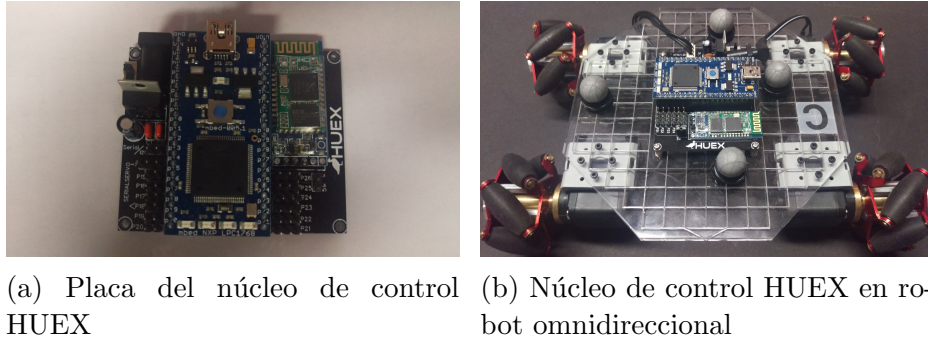


Figura 5.9: Núcleo de control HUEX

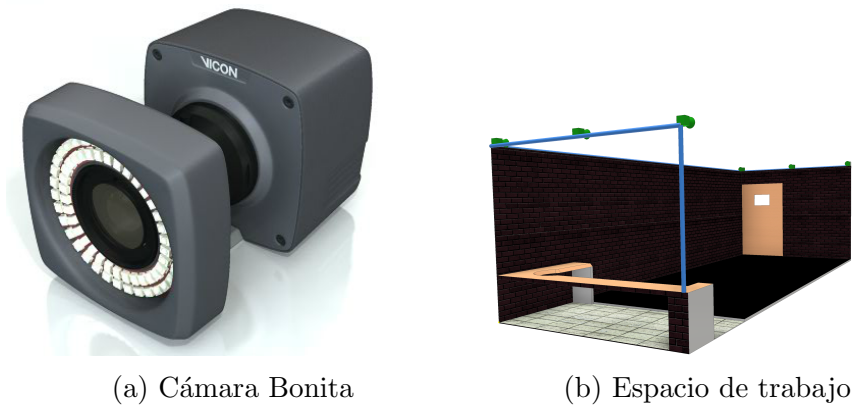


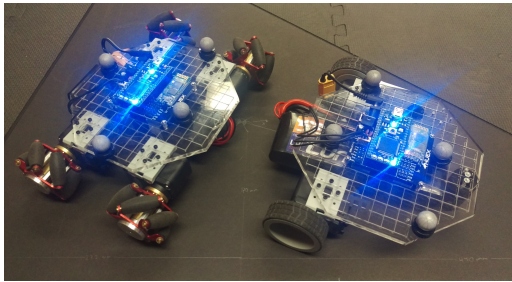
Figura 5.10: sistema de captación de movimiento Vicon

### 5.2.3. Implementación del sistema de control

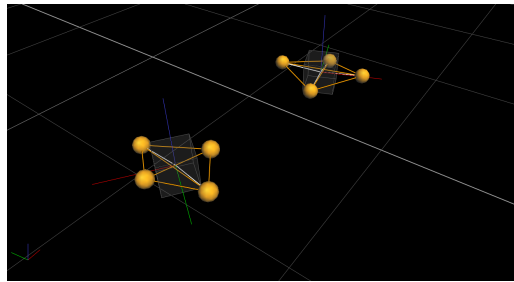
Al requerir la posición y orientación de los robots, para comprobar los métodos de control en robots moviéndose, ya no en un plano, mas bien en el espacio, se mudó del sistema de visión al de captación de movimiento con cámaras infrarrojas. El sistema de captación de movimiento Vicon<sup>®</sup> compuesto por 6 camaras modelo Bonita<sup>®</sup> mostradas en la figura 5.10a, permite maximizar la ubicación espacial en el área de trabajo mostrada en la figura 5.10b, 3 × 7 metros, y ubicar el centroide de los robots con una resolución de  $\pm 5mm$ . En la figura 5.11a, se muestran los robots con los marcadores esféricos que identifican a los robots en el sistema Vicon. La posición retroalimentada a través de los marcadores esféricos por el programa Tracker<sup>®</sup> del sistema Vicon está relacionado con el centroide del robot, por lo que se busca que el centroide coincida con el centro del robot o el punto de análisis  $\alpha$  para el robot unicycle, como se muestra en la figura 5.11b

El programa Tracker registra la posición y orientación del objeto y envía la información en tiempo real a MatLab R2015a en donde se realizan los algoritmos de control y se envían las velocidades de las ruedas a cada robot. El proceso tarda 100ms en completarse.

Se usa una PC con Windows 7 profesional SP1, con procesador Intel<sup>®</sup> i7 4770 a 3.4 GHz, 16 GB de RAM, y sistema operativo de 64 bits. Los programas usados para la interfaz son Vicon Tracker V3.3, y MatLab R2015a.



(a) Robots con marcadores esféricos



(b) Vista de los marcadores

Figura 5.11: Marcadores en robots y visualización en Vicon

# Capítulo 6

## Conclusiones y trabajo futuro

### 6.1. Conclusiones

La robótica presenta soluciones de alta tecnología a problemas humanos, en gran variedad de ámbitos, desde la industria hasta la educación, pasando por la medicina y el hogar, entre muchos otros.

Los robots móviles presentan ventajas sobre sus contrapartes fijas (brazos robóticos), debido a su posibilidad de navegar e interactuar con espacios físicos existentes. Esta ventaja ha creado una enorme área de estudio que presenta muchísimos desafíos. Entre estos desafíos se encuentra el costo, la complejidad de los robots móviles, y la complejidad de las tareas. Una solución planteada en robótica para combatir estos problemas es el uso de varios robots más sencillos y económicos, que trabajando juntos, realicen la tarea deseada. A estos sistemas se les llama sistemas robóticos multiagente.

Entonces, en el marco de los SRM de robots móviles este documento se enfoca en tres aspectos de estudio. Primero, una estrategia de coordinación que no depende de la posición espacial exacta, sino que se basa en la distancia entre robots y la orientación respecto a un eje de referencia arbitrario. Segundo, la coordinación de robots distintos, heterogéneos, con diversas capacidades motrices con la finalidad de armonizar sus interacciones. Tercero, la creación de una plataforma de robots heterogéneos, con un núcleo básico de control que puede conectarse a diversos tipos de actuadores.

La estrategia de formación basada en distancia y orientación que usa F-PARDs mostrada en el capítulo 3 se comprobó usando simulaciones numéricas en 3.3 y luego experimentos en 3.4. La estrategia desarrollada es una opción que soluciona el problema del uso de ejes de referencia cartesianos, podría ser usada para coordinar robots y humanos en distancias respecto a cualquier eje de referencia arbitrario.

El control para robots de primer y segundo orden presentado en el capítulo 4, demostrado por medio de simulaciones numéricas en 4.2 y experimentos en 4.3, permite la interacción de robots distintos en formación. La heterogeneidad es deseable, ya que diferentes agentes pueden tener especialización distinta, y realizar partes especiales en una tarea. De forma tal que el algoritmo podría ser extendido para incluir equipos multiagente conformados por robots, humanos y caninos para la búsqueda sistemática de personas extraviadas en zonas naturales o para búsqueda y rescate en zonas de desastres.

Con el núcleo de control diseñado HUEx, se logró coordinar, como se muestra en

4.3, dos robots distintos, un unicycle y un omnidireccional, para aprovechar las capacidades de locomoción del omnidireccional y la simplicidad del unicycle y converger a las distancias deseadas. Este núcleo de control permite la comunicación necesaria para el cumplimiento del axioma 2.2.1m condición básica para el consenso. también brinda la posibilidad de ampliación a diversos tipos de locomoción, manteniendo las características esenciales que representan a un SRM.

Entonces, dado lo establecido en el capítulo 1, se cumplió el objetivo general de diseño de estrategias de coordinación de movimiento para SRMH, las cuales se presentan en los capítulos 3 y 4. Además se cumplió el objetivo de diseño de la plataforma experimental de SRMH, la cual se muestra en el capítulo 5. Estos objetivos se cumplieron llevando a cabo los objetivos específicos, OP1 dado en 2.2.8, OP2 en el capítulo 4, OP3 a lo largo del capítulo 2, los pasos para lograr OP4 y OP5 se muestran en el capítulo 5 y OP6 en la sección 4.3.

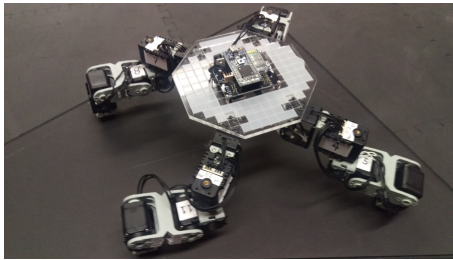
## 6.2. Trabajo futuro

Durante los trabajos realizados surgieron muchas ideas y varios problemas que vislumbran oportunidades de desarrollo.

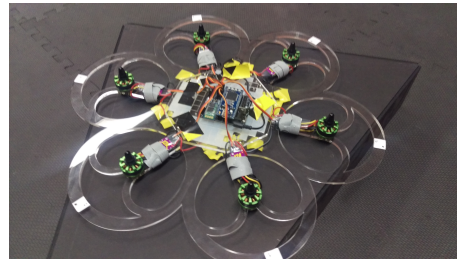
Un área de análisis futuro se presenta en el estudio de las leyes de control planteadas y sus restricciones. Se encontró durante el desarrollo de las leyes de control un mínimo local, el problema de la configuración colineal, detallado en las secciones 3.2 y 4.1. Aunque las restricciones planteadas en 3.2 y 4.1, permiten evitar casi por completo este problema, siempre existe la posibilidad de ocurrencia. El caso de orientación colineal se pudo plantear en los cálculos y simulaciones, pero fue imposible reproducirlo durante los experimentos. Una posible solución se presenta en [31] donde los autores usan una ley que contiene términos especiales, los cuales aparecen cuando se presenta la situación de la colinealidad y permiten cambiar las condiciones para evitarla.

Como se mostró en los capítulos 4 y 5, el uso de Vicon en espacio cerrado es una antesala de prueba a la coordinación descentralizada de las estrategias de control. En este caso el trabajo futuro sería en el área de sensores. Usar sensores de distancia, láser o ultrasónicos y sensores de orientación, magnetómetro o IMU, aumentaría la independencia de los marcos coordenados cartesianos. Otros aspectos que aumentarían la autonomía deseada serían el procesamiento de las leyes de control dentro del mismo robot, sin usar una computadora central, y la comunicación independiente entre robots, sin necesidad de procesamiento por parte de una computadora central. De forma evidente estos objetivos representarían, cada uno, un proyecto en sí mismos.

La plataforma heterogénea diseñada es el primer paso de una serie de escalones en camino a un núcleo de control con alto procesamiento independiente, que pudiese incluir sensores adecuados y redundantes para coordinar de forma eficaz a cualquier número de agentes. En la figura 6.1 se muestran dos robots en proceso de desarrollo que se controlan por el núcleo de control diseñado. El 6.1a es un cuadrúpedo y el 6.1b un hexacóptero. Estos y muchos otros sistemas de locomoción, en diferentes ambientes podrían ser controlados y coordinados por medio del núcleo HUEX. El trabajo futuro será seguir ampliando las capacidades del núcleo HUEX y explorar nuevas posibilidades de locomoción.



(a) Robot cuadrúpedo



(b) Robot hexacóptero

Figura 6.1: Modelos en desarrollo

# Apéndice A

## Diagramas de ensamble estructural de robots



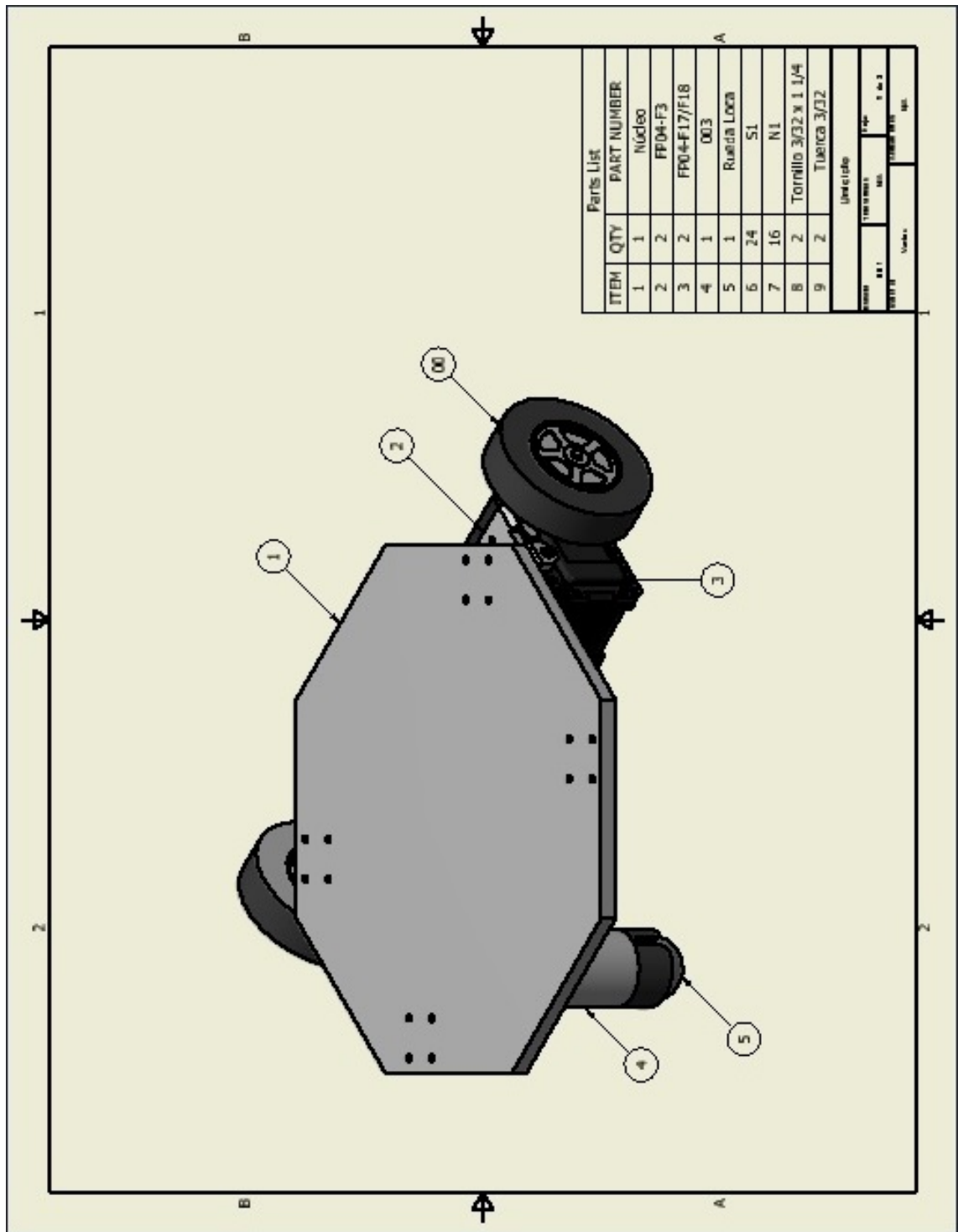


Figura A.1: Diagrama de robot uniciclo





# Apéndice B

## Diagrama y códigos del núcleo de control

### B.1. Diagrama del nucleo de control HUEX

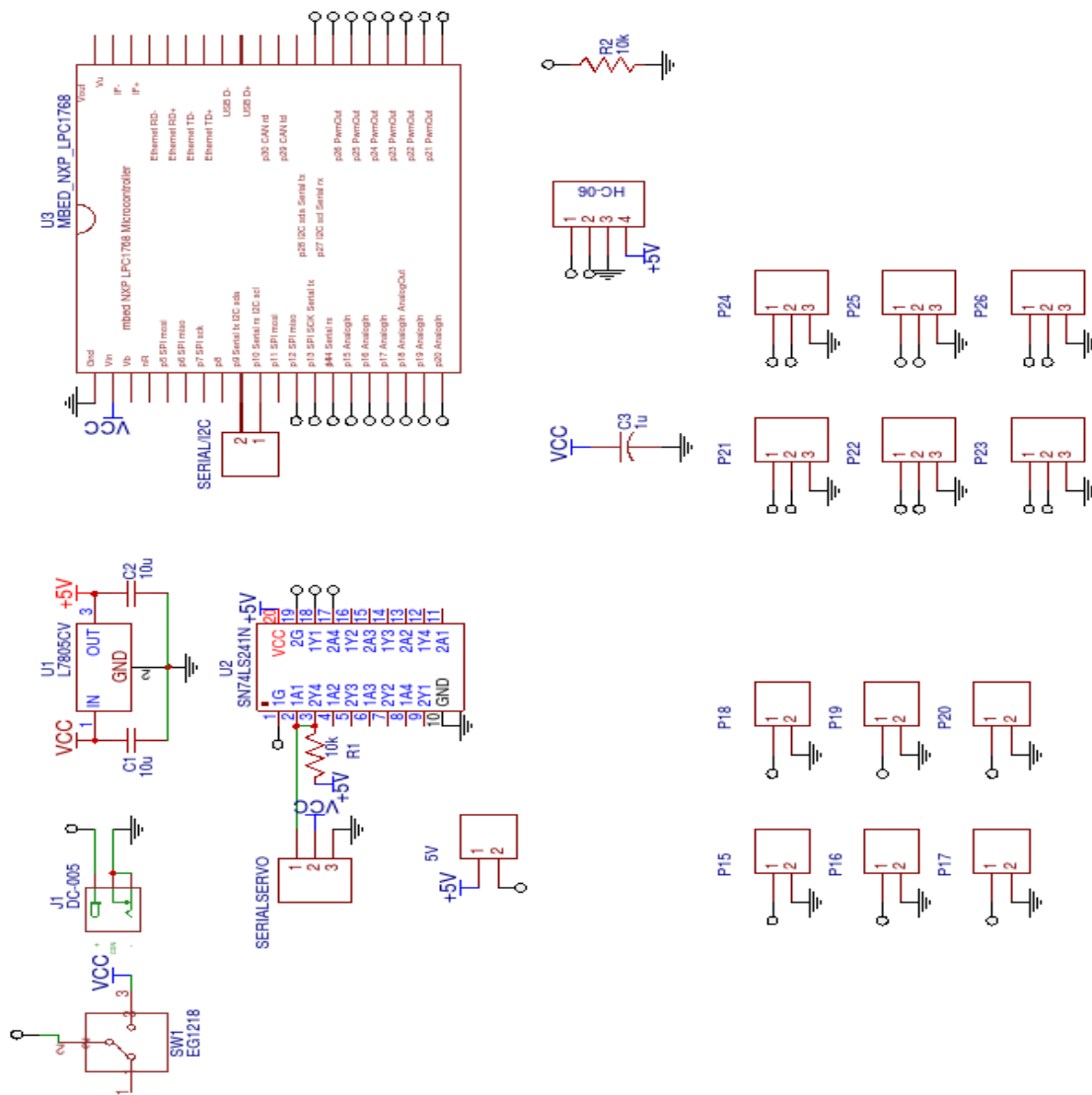


Figura B.1: Diagrama conexión de la placa HUEX

## B.2. Códigos del núcleo de control HUEX

Para el buen funcionamiento del núcleo de control HUEX se diseñó una librería en C++ para mover los servomotores Dynamixel. El diseñador de esta librería es el Licenciado Pablo Paniagua Contro. Dicha librería se puede encontrar en <https://developer.mbed.org/users/papaco/>. A continuación se muestran los programas para el control de los robots unicycle y omnidireccional.

---

```
//Programa para el robot unicycle
#include "DynamixelAX12.h"
#include "mbed.h"

Serial serialBT(p28,p27);
Dynamixel din_1(1),din_2(2);
DigitalOut led_1(LED1);
DigitalOut led_2(LED2);
DigitalOut led_3(LED3);
DigitalOut led_4(LED4);
void Joystick(char incom);
bool FS=false;//FailSafe

void SerInterrupt(){
    led_1 = !led_1;
    char incom=serialBT.getc();
    if(!FS)
        Joystick(incom);
    else
        if(incom=='P'){
            FS=false;
            led_4=0;
        }//if
}//SerInterrupt

void FailSafe(){
    FS=true;
    din_1.SetVelocity(0);
    din_2.SetVelocity(0);
    led_4 = 1;
}//FailSafe

int main(void){
    serialBT.baud(9600);
    serialBT.attach(&SerInterrupt,Serial::RxIrq);
    din_1.SetMode(true);
    din_2.SetMode(true);
    FailSafe();
    while(1){}//while(1)
}//Main

void Joystick(char incom){
    if((incom=='1')||(incom=='2')){
```

```

short val=((short)serialBT.getc())|(((short)serialBT.getc())<<8);
if(val<0)
    val=1024-val;
switch(incom){
    case '1':
        din_1.SetVelocity(val);
        break;
    case '2':
        din_2.SetVelocity(val);
        break;
} //switch
} else{
    while(serialBT.readable()){serialBT.getc();}
    FailSafe();
} //else
return;
} //Joystick

```

---

```

//Programa para el robot Omnidireccional
#include "DynamixelAX12.h"
#include "mbed.h"
#include <math.h>

Serial serialBT(p28,p27);
Dynamixel din_1(1),din_2(2),din_3(3),din_4(4);
DigitalOut led_1(LED1);
DigitalOut led_2(LED2);
DigitalOut led_3(LED3);
DigitalOut led_4(LED4);
void Joystick(char incom);
bool FS=false; //FailSafe

void SerInterrupt(){
    led_1 = !led_1;
    char incom=serialBT.getc();
    if(!FS)
        Joystick(incom);
    else
        if(incom=='P'){
            FS=false;
            led_4=0;
        } //if
} //SerInterrupt

void FailSafe(){
    FS=true;
    din_1.SetVelocity(0);
    din_2.SetVelocity(0);
    din_3.SetVelocity(0);
    din_4.SetVelocity(0);
    led_4 = 1;

```

```

}//FailSafe

int main(void){
    serialBT.baud(9600);
    serialBT.attach(&SerInterrupt,Serial::RxIrq);
    din_1.SetMode(true);
    din_2.SetMode(true);
    din_3.SetMode(true);
    din_4.SetMode(true);
    FailSafe();
    while(1){}//while(1)
}//Main

void Joystick(char incom){
    if((incom=='1')||(incom=='2')||(incom=='3')||(incom=='4')){
        short val=((short)serialBT.getc())|(((short)serialBT.getc())<<8);
        if(val<0)
            val=1024-val;
        switch(incom){
            case '1':
                din_1.SetVelocity(val);
                break;
            case '2':
                din_2.SetVelocity(val);
                break;
            case '3':
                din_3.SetVelocity(val);
                break;
            case '4':
                din_4.SetVelocity(val);
                break;
        }//switch
    }else{
        while(serialBT.readable()){serialBT.getc();}
        FailSafe();
    }//else
    return;
}//Joystick

```

---

# Apéndice C

## Archivos en Matlab de simulaciones y experimentos

### C.1. Plataforma experimental heterogénea

Para los experimentos mostrados en 4.3 con la plataforma heterogénea se usaron los siguientes archivos en Matlab. El primer archivo es la función de control que implementa el algoritmo de control de robots heterogéneos, extendido para cuatro robots, dos unicíclos y dos omnidireccionales. Los siguientes dos archivos corresponden a las funciones que calculan las velocidades de las ruedas y las envían a los robots.

---

```
// Programa principal de control

%*****
%                               Last update 07/09/2016 *
%*****
%                               Heterogeneous                               *
%*****

%
%       Orobots(1)=OMNIBOT('OMNIBOT_A');
%       Orobots(2)=OMNIBOT('OMNIBOT_B');
%       Orobots(1).createBT;
%       Orobots(2).createBT;
%       Urobots(1)=UNIBOT('UNIBOT_E');
%       Urobots(2)=UNIBOT('UNIBOT_F');
%       Urobots(1).createBT;
%       Urobots(2).createBT;
%*****

function het(Orobots,Urobots)
    j=vrjoystick(1);
    track=TRACKER();
    track.on;
    Orobots(1).connect;
    Orobots(2).connect;
    Urobots(1).connect;
    Urobots(2).connect;
    Orobots(1).lock;
```



```

Orobots(2).lock;
Urobots(1).lock;
Urobots(2).lock;
try
    %%                                Loop
    while(~button(j,1))
        Orobots(1).update(track);
        Orobots(2).update(track);
        Urobots(1).update(track);
        Urobots(2).update(track);
        alpha=[[Orobots(1).pos.x,Orobots(1).pos.y];...
        [Urobots(1).pos.x,Urobots(1).pos.y];...
        [Urobots(2).pos.x,Urobots(2).pos.y];...
        [Orobots(2).pos.x,Orobots(2).pos.y]];
        A=500;
        B=500;
        D12=sqrt(A^2+B^2);
        D13=sqrt(A^2+B^2);
        D14=2*B;
        D23=2*A;
        D24=sqrt(A^2+B^2);
        D34=sqrt(A^2+B^2);
        Beta12=norm(alpha(2,:)-alpha(1,:));
        Beta13=norm(alpha(3,:)-alpha(1,:));
        Beta14=norm(alpha(4,:)-alpha(1,:));
        Beta23=norm(alpha(3,:)-alpha(2,:));
        Beta24=norm(alpha(4,:)-alpha(2,:));
        Beta34=norm(alpha(4,:)-alpha(3,:));
        phi12=Beta12-D12;
        phi13=Beta13-D13;
        phi14=Beta14-D14;
        phi23=Beta23-D23;
        phi24=Beta24-D24;
        phi34=Beta34-D34;
        phi1=[phi12;phi13;phi14];
        phi2=[phi12;phi23;phi24];
        phi3=[phi13;phi23;phi34];
        phi4=[phi14;phi24;phi34];
        theta12=atan2(alpha(2,2)-alpha(1,2),alpha(2,1)-alpha(1,1));
        theta21=atan2(alpha(1,2)-alpha(2,2),alpha(1,1)-alpha(2,1));
        theta13=atan2(alpha(3,2)-alpha(1,2),alpha(3,1)-alpha(1,1));
        theta31=atan2(alpha(1,2)-alpha(3,2),alpha(1,1)-alpha(3,1));
        theta14=atan2(alpha(4,2)-alpha(1,2),alpha(4,1)-alpha(1,1));
        theta41=atan2(alpha(1,2)-alpha(4,2),alpha(1,1)-alpha(4,1));
        theta23=atan2(alpha(3,2)-alpha(2,2),alpha(3,1)-alpha(2,1));
        theta32=atan2(alpha(2,2)-alpha(3,2),alpha(2,1)-alpha(3,1));
        theta24=atan2(alpha(4,2)-alpha(2,2),alpha(4,1)-alpha(2,1));
        theta42=atan2(alpha(2,2)-alpha(4,2),alpha(2,1)-alpha(4,1));
        theta34=atan2(alpha(4,2)-alpha(3,2),alpha(4,1)-alpha(3,1));
        theta43=atan2(alpha(3,2)-alpha(4,2),alpha(3,1)-alpha(4,1));
        gamma12=(Beta12+D12)/(2*Beta12^2);

```

```

gamma13=(Beta13+D13)/(2*Beta13^2);
gamma14=(Beta14+D14)/(2*Beta14^2);
gamma23=(Beta23+D23)/(2*Beta23^2);
gamma24=(Beta24+D24)/(2*Beta24^2);
gamma34=(Beta34+D34)/(2*Beta34^2);
k1=[gamma12*cos(theta12) gamma12*sin(theta12);...
    gamma13*cos(theta13) gamma13*sin(theta13);...
    gamma14*cos(theta14) gamma14*sin(theta14)];
k2=[gamma12*cos(theta21) gamma12*sin(theta21);...
    gamma23*cos(theta23) gamma23*sin(theta23);...
    gamma24*cos(theta24) gamma24*sin(theta24)];
k3=[gamma13*cos(theta31) gamma13*sin(theta31);...
    gamma23*cos(theta32) gamma23*sin(theta32);...
    gamma34*cos(theta34) gamma34*sin(theta34)];
k4=[gamma14*cos(theta41) gamma14*sin(theta41);...
    gamma24*cos(theta42) gamma24*sin(theta42);...
    gamma34*cos(theta43) gamma34*sin(theta43)];
kp0=500;
kpU=30;
kTH=.2;
thS=pi/2;
eps=0.0001;
u1=kp0*k1'*phi1;
u1=[u1;(-kTH*(Orobots(1).pos.psi-thS))]
u2=kpU*k2'*phi2
u3=kpU*k3'*phi3
u4=kp0*k4'*phi4;
u4=[u4;(-kTH*(Orobots(2).pos.psi-thS))]
%%                               Omnidirectional 1 -- Robot1
A=[cos(Orobots(1).pos.psi) -sin(Orobots(1).pos.psi)
    0;sin(Orobots(1).pos.psi) cos(Orobots(1).pos.psi) 0;0 0 1];
errorpos=[u1(1); u1(2)];
errorang=u1(3);
if(abs(norm(errorpos))<=50)
    N=0;
else
    N=7000;
end

if(abs(errorang)<=0.0873)
    Nw=0;
else
    Nw=10;
end
FNormpos=N/sqrt(u1(1)^2+u1(2)^2+eps)*[u1(1); u1(2)];
FNormang=Nw/sqrt(u1(3)^2+eps)*u1(3);
FNorm=[FNormpos; FNormang];
U1=A\FNorm;
vx=U1(1);
vy=-U1(2);
w=U1(3);

```

```

Orobots(1).move(vx,vy,w)
%%                               Uniciclo 1 --> Robot2
A=[cos(Urobots(1).pos.psi)
   -Urobots(1).l*sin(Urobots(1).pos.psi);sin(Urobots(1).pos.psi)
   Urobots(1).l*cos(Urobots(1).pos.psi)];
if(abs(norm(u2))<=10)
    N=0;
else
    N=6000;
end
FNorm2=N/sqrt(u2(1)^2+u2(2)^2+eps)*u2;
U2=A\FNorm2;
Urobots(1).move(U2(1),U2(2));
%%                               Uniciclo 2 --> Robot3
A=[cos(Urobots(2).pos.psi)
   -Urobots(2).l*sin(Urobots(2).pos.psi);sin(Urobots(2).pos.psi)
   Urobots(2).l*cos(Urobots(2).pos.psi)];
if(abs(norm(u3))<=10)
    N=0;
else
    N=6000;
end
FNorm3=N/sqrt(u3(1)^2+u3(2)^2+eps)*u3;
U3=A\FNorm3;
Urobots(2).move(U3(1),U3(2));
%%                               Omnidirectional 2 -- Robot4
A=[cos(Orobots(2).pos.psi) -sin(Orobots(2).pos.psi)
   0;sin(Orobots(2).pos.psi) cos(Orobots(2).pos.psi) 0;0 0 1];
errorpos=[u4(1); u4(2)];
errorang=u4(3);
if(abs(norm(errorpos))<=50)
    N=0;
else
    N=7000;
end

if(abs(errorang)<=0.0873)
    Nw=0;
else
    Nw=10;
end
FNormpos=N/sqrt(u4(1)^2+u4(2)^2+eps)*[u4(1); u4(2)];
FNormang=Nw/sqrt(u4(3)^2+eps)*u4(3);
FNorm=[FNormpos; FNormang];
U4=A\FNorm;
vx=U4(1);
vy=-U4(2);
w=U4(3);
Orobots(2).move(vx,vy,w);
pause(.05);
end
end

```

```

catch
    disp(ERROR);
end
Orobots(1).move(0,0,0);
Orobots(2).move(0,0,0);
Urobots(1).move(0,0);
Urobots(2).move(0,0);
Orobots(1).lock;
Orobots(2).lock;
Urobots(1).lock;
Urobots(2).lock;
Orobots(1).disconnect;
Orobots(2).disconnect;
Urobots(1).disconnect;
Urobots(2).disconnect;
track.off;
end

```

---

```

// Funcion para el calculo y envio de velocidades al robot unicycle

```

```

%*****
%
%                               Last update 06/09/2016 *
%*****
%                               UNIBOT                               *
%*****

```

```

classdef UNIBOT < handle
    properties
        type=0;           % 0 as in Unicycle
        L=175;            %Longitud entre llantas
        r=25;             %Radio de las ruedas
        l=50;            %Longitud al punto alpha
        BTport;          %Bluetooth serial port
        pos;             %Current position of the robot
        name;            %Robot object name as it is on tracker
        j=vrjoystick(1); %Joystick
        SPx=0;           %Home x
        SPy=0;           %Home y
    end
    methods
        function obj = UNIBOT(name)
            obj.name=name;
            obj.pos=COORDINATES();
        end
        function createBT(obj)
            obj.BTport=Bluetooth(obj.name,1);
        end
        function connect(obj)
            fopen(obj.BTport);
        end
        function disconnect(obj)

```

```

        fclose(obj.BTport);
    end
    function lock(obj)
        fwrite(obj.BTport,'P','char');
        pause(1);
    end
    function update(obj,t)
        obj.pos=t.get(obj.name);
    end
    function setHome(obj,x,y)
        obj.SPx=x;
        obj.SPy=y;
    end
    function rc(obj)
        while(~button(obj.j,1))
            w=-axis(obj.j,1)*100;
            v=-axis(obj.j,2)*20000;
            move(obj,v,w);
            pause(.1);
            if(button(obj.j,2))
                obj.lock();
            end
        end
    end
    function home(obj,t)
        error=100;
        k=1;
        eps=.0001;
        while(abs(norm(error))>10)
            obj.update(t);
            A=[cos(obj.pos.psi)
                -obj.l*sin(obj.pos.psi);sin(obj.pos.psi)
                obj.l*cos(obj.pos.psi)];
            error=[obj.pos.x-obj.SPx;obj.pos.y-obj.SPy];
            F=-(k*error);
            if(abs(norm(error))<=10)
                N=0;
            else
                N=6500;
            end
            FNorm=N/sqrt(F(1)^2+F(2)^2+eps)*F;
            U=A\FNorm;
            move(obj,U(1),U(2));
            pause(.05);
            if(button(obj.j,1))
                break;
            end
        end
        obj.move(0,0);
    end
    function move(obj,v,w)

```

```

        w1=round(((v/obj.r)-(w*(obj.L/obj.r))));
        w2=round(((v/obj.r)-(w*(obj.L/obj.r))));
        if(w1>1000)
            w1=1000;
        end
        if(w1<-1000)
            w1=-1000;
        end
        if(w2>1000)
            w2=1000;
        end
        if(w2<-1000)
            w2=-1000;
        end
        fwrite(obj.BTport,'1','char');
        fwrite(obj.BTport,w2,'int16');
        fwrite(obj.BTport,'2','char');
        fwrite(obj.BTport,w1,'int16');
    end
end
end
end

```

---

```

// Funcion para el calculo y envio de velocidades al robot omnidireccional

```

```

%*****
%
%                               Last update 06/09/2016 *
%*****
%                               OMNIBOT *
%*****

```

```

classdef OMNIBOT < handle
    properties
        type=1;           % 1 as in omnidirectional
        L=100;            %Longitud entre llantas
        r=27.5;          %Radio de las ruedas
        l=50;             %Longitud entre ejes
        BTport;          %Bluetooth serial port
        pos;              %Current position of the robot
        name;             %Robot object name as it is on tracker
        j=vrjoystick(1); %Joystick
        SPx=0;            %Home x
        SPy=0;            %Home y
        SPw=0;            %Home w
    end
    methods
        function obj = OMNIBOT(name)
            obj.name=name;
            obj.pos=COORDINATES();
        end
        function createBT(obj)
            obj.BTport=Bluetooth(obj.name,1);
        end
    end
end

```

```

end
function connect(obj)
    fopen(obj.BTport);
end
function disconnect(obj)
    fclose(obj.BTport);
end
function lock(obj)
    fwrite(obj.BTport, 'P', 'char');
    pause(1);
end
function update(obj,t)
    obj.pos=t.get(obj.name);
end
function setHome(obj,x,y,w)
    obj.SPx=x;
    obj.SPy=y;
    obj.SPw=w;
end
function rc(obj)
    obj.connect;
    obj.lock;
    while(~button(obj.j,8))
        vx=axis(obj.j,4)*20000;
        vy=axis(obj.j,2)*20000;
        w=-axis(obj.j,1)*100;
        move(obj,vx,vy,w);
        pause(.1);
        if(button(obj.j,2))
            obj.lock();
        end
    end
    obj.lock;
    obj.disconnect;
end
function home(obj,t)
    error=100;
    eps=.0001;
    k=.1;
    while(abs(norm(error))>10)
        obj.update(t);
        A=[cos(obj.pos.psi) -sin(obj.pos.psi) 0;sin(obj.pos.psi)
            cos(obj.pos.psi) 0;0 0 1];
        error=[-obj.SPx+obj.pos.x;-obj.SPy+obj.pos.y;-obj.SPw+obj.pos.psi];
        F=[-k*error(1);-k*error(2);-0.5*k*error(3)];
        if(abs(norm([error(1); error(2)]))<=10)
            N=0;
        else
            N=6000;
        end
        if(abs(error(3))<=0.0873)

```

```

        Nw=0;
    else
        Nw=10;
    end
    FNormpos=N/sqrt(F(1)^2+F(2)^2+eps)*[F(1); F(2)];
    FNormang=Nw/sqrt(F(3)^2+eps)*F(3);
    FNorm=[FNormpos; FNormang];
    U=A\FNorm;
    obj.move(U(1),-U(2),U(3));
    pause(.1);
    if(button(obj.j,1))
        break;
    end
end
obj.move(0,0,0);
end
function move(obj,vx,vy,w)
    J=(1/obj.r)*[1 1 -(obj.L+obj.l);-1 1 (obj.L+obj.l);-1 1
        -(obj.L+obj.l);1 1 (obj.L+obj.l)]*[vx;vy;-w];
    w1=J(1);
    w2=J(2);
    w3=J(3);
    w4=J(4);
    if(w1>1000)
        w1=1000;
    end
    if(w2>1000)
        w2=1000;
    end
    if(w3>1000)
        w3=1000;
    end
    if(w4>1000)
        w4=1000;
    end
    if(w1<-1000)
        w1=-1000;
    end
    if(w2<-1000)
        w2=-1000;
    end
    if(w3<-1000)
        w3=-1000;
    end
    if(w4<-1000)
        w4=-1000;
    end
    fwrite(obj.BTport,'1','char');
    fwrite(obj.BTport,w1,'int16');
    fwrite(obj.BTport,'2','char');
    fwrite(obj.BTport,-w2,'int16');

```



```
        fwrite(obj.BTport,'3','char');
        fwrite(obj.BTport,-w3,'int16');
        fwrite(obj.BTport,'4','char');
        fwrite(obj.BTport,w4,'int16');
    end
end
end
```

---

# Bibliografía

- [1] Y. Aiyama, M. Hara, T. Yabuki, J. Ota, and T. Arai. Cooperative transportation by two four-legged robots with implicit communication. *Robotics and Autonomous Systems*, 29(1):13 – 19, 1999.
- [2] E. Alpaydin. *Introduction to machine learning*. The MIT Press, 2004.
- [3] B. Anderson, Z. Lin, and M. Deghat. Combining distance-based formation shape control with formation translation. In L. Qiu, J. Chen, T. Iwasaki, and H. Fujioka, editors, *New Trends in Control and Networked Dynamical Systems*, volume 1, pages 121–130. IET, 1st edition, 2012.
- [4] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino. Decentralized time-varying formation control for multi-robot systems. *Journal of Robotics Research*, 33(7):1029–1043, 2014.
- [5] T. Arai, E. Pagello, and L. Parker. Guest editorial advances in multirobot systems. *Robotics and Automation, IEEE Transactions on*, 18(5):655–661, 2002.
- [6] Y. Asahiro, H. Asama, I. Suzuki, and M. Yamashita. Improvement of distributed control algorithms for robots carrying an object. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 608–613 vol.6, 1999.
- [7] L. Asimow and B. Roth. The rigidity of graphs. *Transactions of the american mathematical society*, 245, 1978.
- [8] C. Azani. System of systems architecting via natural development principles. In *System of Systems Engineering, 2008. SoSE '08. IEEE International Conference on*, pages 1–6, 2008.
- [9] K. R. Baghaei and A. Agah. Task allocation methodologies for multi-robot systems, 2003.
- [10] T. Balch and R. Arkin. Behavior-based formation control for multirobot teams. *Robotics and Automation, IEEE Transactions on*, 14(6):926–939, 1998.
- [11] F. Belkhouche and B. Belkhouche. Modelling and controlling a robotic convoy using guidance laws strategies. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 35(4):813–825, 2005.
- [12] A. Bemporad and C. Rocchi. Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 7488–7493, 2011.

- [13] J. Boardman and B. Sauser. System of systems the meaning of.
- [14] R. W. Brockett. Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*, pages 181–191. Birkhauser, 1983.
- [15] K. cai Cao, G. Xiang, and H. Yang. Formation control of multiple nonholonomic mobile robots. In *Information Science and Technology (ICIST), 2011 International Conference on*, pages 1038–1042, 2011.
- [16] Y. Cao, A. Fukunaga, A. Kahng, and F. Meng. Cooperative mobile robotics: antecedents and directions. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, pages 226–234 vol.1, 1995.
- [17] L. R. G. Carrillo, A. E. D. López, R. Lozano, and C. Pégard. Quad rotorcraft control: vision-based hovering and navigation, 2012.
- [18] Y.-Q. Chen and Z. Wang. Formation control: a review and a new consideration. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3181–3186, 2005.
- [19] M. De le Torre and A. Stentz. A market approach to multirobot coordination. *Research Showcase, Carnegie Mellon University*, 2001.
- [20] C. C. de Wit, G. Bastin, and B. Siciliano, editors. *Theory of Robot Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1996.
- [21] J. Desai. A graph theoretic approach for modeling mobile robot team formations. *Journal of Robotic Systems*, 19(11):511–525, 2002.
- [22] J. Desai, J. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 2864–2869, 1998.
- [23] J. Desai, J. Ostrowski, and V. Kumar. Modelling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 6:905–908, 2001.
- [24] D. Dimarogonas and K. Kyriakopoulos. Distributed cooperative control and collision avoidance for multiple kinematic agents. In *IEEE Conference on Decision and Control*, pages 721–726, 2006.
- [25] D. V. Dimarogonas and K. H. Johansson. Stability analysis for multi-agent systems using the incidence matrix: Quantized communication and formation control. *Automatica*, 46(4):695–700, 2010.
- [26] K. Do. Formation tracking control of unicycle-type mobile robots with limited sensing ranges. *Control Systems Technology, IEEE Transactions on*, 16(3):527–538, 2008.
- [27] W. Dong, Y. Guo, and J. Farrell. Formation control of nonholonomic mobile robots. In *American Control Conference, 2006*, pages 6 pp.–, 2006.

- [28] Z. dong Wang, Y. Takano, Y. Hirata, and K. Kosuge. A pushing leader based decentralized control method for cooperative object transportation. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 1035–1040 vol.1, 2004.
- [29] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. Christensen, A. Decugniree, G. Di Caro, F. Ducatelle, E. Ferrante, A. Forster, J. Guzzi, V. Longchamp, S. Magnenat, J. Martinez Gonzales, N. Mathews, M. Montes de Oca, R. O’Grady, C. Pinciroli, G. Pini, P. Retornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stutzle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, page in press, 2013.
- [30] J. Escareno, S. Salazar-Cruz, and R. Lozano. Embedded control of a four-rotor uav. In *2006 American Control Conference*, pages 6–pp. IEEE, 2006.
- [31] E. D. Ferreira-Vazquez, E. G. Hernandez-Martinez, J. J. Flores-Godoy, G. Fernandez-Anaya, and P. Paniagua-Contro. Distance-based formation control using angular information between robots. *Journal of Intelligent & Robotic Systems*, pages 1–18, 2015.
- [32] B. Fidan, V. Gazi, S. Zhai, and N. Cen. Single-view distance-estimation-based formation control of robotic swarms. *IEEE Transactions on Industrial Electronics*, 60(12):5781–5791, 2013.
- [33] R. Fierro, A. Das, V. Kumar, and J. Ostrowski. Hybrid control of formations of robots. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 157–162, 2001.
- [34] J. Fredslund and M. Mataric. General algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation*, 18(5):837–846, 2002.
- [35] M. Friedmann, K. Petersen, and O. von Stryk. Adequate motion simulation and collision detection for soccer playing humanoid robots. *Robotics and Autonomous Systems*, 57(8):786 – 795, 2009. Humanoid Soccer Robots.
- [36] H. Garcia de Marina, M. Cao, and B. Jayawardhana. Controlling rigid formations of mobile agents under inconsistent measurements. *IEEE Transactions on Robotics*, 31(1):31–39, 2015.
- [37] B. Gerkey and M. Mataric. Pusher-watcher: an approach to fault-tolerant tightly-coupled robot coordination. In *Robotics and Automation, 2002. Proceedings. ICRA ’02. IEEE International Conference on*, volume 1, pages 464–469 vol.1, 2002.
- [38] T. Gunn and J. Anderson. Dynamic heterogeneous team formation for robotic urban search and rescue. *Procedia Computer Science*, 19(0):22 – 31, 2013.

The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), the 3rd International Conference on Sustainable Energy Information Technology (SEIT-2013).

- [39] S. Guo, X. Ye, and B. Gao. Motion planning of underwater multi-microbot system. In *Mechatronics and Automation, 2008. ICMA 2008. IEEE International Conference on*, pages 690–695, 2008.
- [40] T. Hayakawa, T. Matsuzawa, and S. Hara. Formation control of multi-agent systems with sampled information – relationship between information exchange structure and control performance –. In *45th IEEE Conference on Decision and Control*, pages 4333–4338, 2006.
- [41] E. Hernández-Martínez and E. Aranda-Bricaire. Decentralized formation control of multi-agent robots systems based on formation graphs. *Studies in Informatics and Control*, 21(1):7–16, 2012.
- [42] E. Hernandez-Martinez, G. Fernandez-Anaya, E. Ferreira, J. Flores-Godoy, and A. Lopez-Gonzalez. Trajectory tracking of a quadcopter uav with optimal translational control. *IFAC-PapersOnLine*, 48(19):226 – 231, 2015.
- [43] E. Hernandez-Martinez, J. Flores-Godoy, and G. Fernandez-Anaya. Decentralized discrete-time formation control for multirobot systems. *Discrete Dynamics in Nature and Society*, 2013:1–8, 2013.
- [44] E. Hernández-Martínez and E. Aranda-Bricaire. Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach. InTech.
- [45] L. Iocchi, D. Nardi, M. Piaggio, and A. Sgorbissa. Distributed coordination in heterogeneous multi-robot systems. *Autonomous Robots*, 15(2):155–168, 2003.
- [46] L. Iocchi, D. Nardi, and M. Salermo. Reactivity and deliberation: a survey on multi-robot systems. *Dipartimento di Informatica e Sistemistica, Universita La Sapienza*, 2001.
- [47] D. Jackson, G. Sedrick, and K. Tayeb. Algorithmic development of effectiveness prediction for system of systems. In *System Theory, 2009. SSST 2009. 41st Southeastern Symposium on*, pages 164–168, 2009.
- [48] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- [49] M. Jamshidi. System of systems engineering – new challenges for the 21st century. *Aerospace and Electronic Systems Magazine, IEEE*, 23(5):4–19, 2008.
- [50] D. Jung and A. Zelinsky. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots*, 8:269–292, 2000.
- [51] S. Kang, M. Park, B. Lee, and H. Ahn. Distance-based formation control with a single moving leader. In *American Control Conference*, volume 1, pages 305–310, 2014.

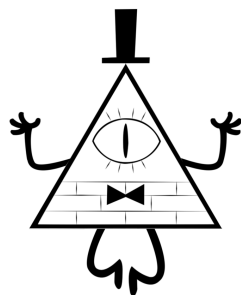
- [52] K. Kanjanawanishkul. Formation control of omnidirectional mobile robots using distributed model predictive control. In *Robot Communication and Coordination, 2009. ROBOCOMM '09. Second International Conference on*, pages 1–7, 2009.
- [53] K. Kanjanawanishkul and A. Zell. A model-predictive approach to formation control of omnidirectional mobile robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2771–2776, 2008.
- [54] T. Kato, K. Watanabe, and S. Maeyama. A formation method for heterogeneous multiple robots by specifying the relative position of each robot. In *SICE Annual Conference 2010, Proceedings of*, pages 3274–3277, 2010.
- [55] R. Kaushik, Y. Feng, W. Morris, J. Xiao, and Z. Zhu. 3d map construction using heterogeneous robots. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 1230–1235, 2008.
- [56] J. Kiener and O. Von Stryk. Cooperation of heterogeneous, autonomous robots: A case study of humanoid and wheeled robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 959–964, 2007.
- [57] L. Krick, M. Broucke, and B. Francis. Stabilization of infinitesimally rigid formations of multi-robot networks. In *IEEE Conference on Decision and Control*, pages 477–482, 2008.
- [58] M. Kumar, D. Garg, and V. Kumar. Self-sorting in a swarm of heterogeneous agents. In *American Control Conference, 2008*, pages 117–122, 2008.
- [59] P. Lancaster and M. Tismenetsky. *The Theory of Matrices: With Applications*. Computer Science and Scientific Computing Series. Academic Press, 1985.
- [60] Z. Lin, M. Broucke, and B. Francis. Local control strategies for groups of mobile autonomous agents. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 1, pages 1006–1011 Vol.1, 2003.
- [61] A. Lopez-Gonzalez, E. D. Ferreira, E. G. Hernandez-Martinez, J. J. Flores-Godoy, G. Fernandez-Anaya, and P. Paniagua-Contro. Multi-robot formation control using distance and orientation. *Advanced Robotics*, 30(14):901–913, 2016.
- [62] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3):20–32, Sept 2012.
- [63] M. McPartland, S. Nolfi, and H. A. Abbass. Emergence of communication in competitive multi-agent systems: a pareto multi-objective approach. In *Proceedings of the 2005 conference on Genetic and evolutionary computation, GECCO '05*, pages 51–58, New York, NY, USA, 2005. ACM.

- [64] K. Miao and M. Li. Role assignment strategy for heterogeneous robots based on fuzzy evaluation and fuzzy inference. In *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*, pages 1–5, 2006.
- [65] NATO. Multi-robot systems in military domains. NATO RTO Technical report.
- [66] I. Navarro and F. Matía. An introduction to swarm robotics. Hindawi publishing corporation.
- [67] O. of the Deputy Under Secretary of Defense for Acquisition and Technology. Systems engineering guide for systems of systems, August 2008.
- [68] K.-K. Oh and H.-S. Ahn. Formation control of mobile agents based on inter-agent distance dynamics. *Automatica*, 47(10):2306–2312, 2011.
- [69] K.-K. Oh and H.-S. Ahn. Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space. *International Journal of Robust and Nonlinear Control*, 24(12):1809–1820, 2014.
- [70] K.-K. Oh, M.-C. Park, and H.-S. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.
- [71] R. Olfati-Saber and R. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, pages 346–352, 2002.
- [72] R. Olfati-Saber and R. M. Murray. Distributed Cooperative Control of Multiple Vehicle Formations using Structural Potential Functions. *IFAC Proceedings Volumes*, 35(1):495 – 500, 2002. 15th IFAC World Congress.
- [73] L. Parker. Current state of the art in distributed autonomous mobile robotics. Oak Ridge National Laboratory.
- [74] L. E. Parker. Heterogeneous multi-robot cooperation. Technical report, MIT, 1994.
- [75] L. Pimenta, V. Kumar, R. Mesquita, and G. Pereira. Sensing and coverage for a network of heterogeneous robots. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3947–3952, 2008.
- [76] J. Pugh and A. Martinoli. Inspiring and modeling multi-robot search with particle swarm optimization. In *In Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, 2007.
- [77] W. Ren and R. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, London, 2008.
- [78] W. Ren, K. Moore, and Y. Chen. High-order consensus algorithms in cooperative vehicle systems. In *2006 IEEE International Conference on Networking, Sensing and Control*, pages 457–462, 2006.

- [79] P. Rybski, A. Larson, H. Veeraraghavan, M. LaPoint, and M. Gini. Communication strategies in multi-robot search and retrieval. In *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems*, pages 301–310, Toulouse, France, 2004.
- [80] J. Santiaguillo-Salinas and E. Aranda-Bricaire. Trajectory following for a a.r. drone four-rotor helicopter using ros (in spanish). In *In the Proc. of XVI Congreso Latinoamericano de Control Automatico*, pages 606–611. CLCA, 2014.
- [81] J. Shao, G. Xie, J. Yu, and L. Wang. A tracking controller for motion coordination of multiple mobile robots. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 783–788, 2005.
- [82] A. Sharkey. Robots, insects and swarm intelligence. *Artificial Intelligence Review*, 26(4):255–268, 2006.
- [83] L. Sheng, Y. Pan, and X. Gong. Consensus formation control for a class of networked multiple mobile robot systems. *Journal of Control Science and Engineering*, 2012(1):1–12, 2012.
- [84] F. Shkurti, A. Xu, M. Meghjani, J. Gamboa Higuera, Y. Girdhar, P. Giguere, B. Dey, J. Li, A. Kalmbach, C. Prahacs, K. Turgeon, I. Rekleitis, and G. Dudek. Multi-domain monitoring of marine environments using a heterogeneous robot team. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1747–1753, 2012.
- [85] R. Simmons, D. Apfelbaum, D. Fox, R. Goldman, K. Haigh, D. Musliner, M. Pelican, and S. Thrun. Coordinated deployment of multiple, heterogeneous robots. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 3, pages 2254–2260 vol.3, 2000.
- [86] J. Simpson and C. H. Dagli. System of systems: Power and paradox. In *System of Systems Engineering, 2008. SoSE '08. IEEE International Conference on*, pages 1–5, 2008.
- [87] J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, United States of America, 1991.
- [88] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002. to appear.
- [89] J. Toibero, F. Roberti, P. Fiorini, and R. Carelli. Hybrid formation control for non-holonomic wheeled mobile robots. In I. S. S. Lee and M. Ki, editors, *Lecture Notes in Control and Information Sciences*, volume 370, pages 21–34. Springer, 1st edition, 2008.
- [90] V. Trianni, T. H. Labella, and M. Dorigo. Evolution of direct communication for a swarm-bot performing hole avoidance. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stutzle, editors, *Ant Colony Optimization and Swarm Intelligence – Proceedings of ANTS 2004 – Fourth International Workshop*, volume 3172 of *Lecture Notes in Computer Science*, pages 130–141. Springer, Berlin, Germany, 2004.



- [91] S. Yamada and J. Saito. Adaptive action selection without explicit communication for multirobot box-pushing. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 31(3):398–404, 2001.
- [92] H. Yamaguchi. A distributed motion coordination strategy for multiple non-holonomic mobile robots in cooperative hunting operations. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 3, pages 2984–2991 vol.3, 2002.
- [93] M. Yogeswaran and S. Ponnambalam. Swarm robotics:an extensive research review. InTech.
- [94] J. Yu and S. M. LaValle. Distance optimal formation control on graphs with a tight convergence time guarantee. In *IEEE Conference on Decision and Control*, pages 4023–4028, 2012.
- [95] M. Zavlanos, H. Tanner, A. Jadbabaie, and G. Pappas. Hybrid control for connectivity preserving flocking. *IEEE Transactions on Automatic Control*, 54(12):2869–2875, 2009.



La realidad es una ilusión,  
el universo es un holograma,  
compra oro, adios!