



CENTRO DE INVESTIGACIÓN Y DE
ESTUDIOS AVANZADOS DEL INSTITUTO
POLITÉCNICO NACIONAL.

UNIDAD ZACATENCO
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN DE MECATRÓNICA.

Control de formación líder-seguidor de un
conjunto de robots móviles

T E S I S

Que presenta

Ing. Jonathan Daniel Gallegos Sanlucar.

Para obtener el grado de
Maestro en Ciencias

En la especialidad de
Ingeniería eléctrica.

Directores de la Tesis:
Dr. Rafael Castro Linares
Dr. Martín Velasco Villa

Ciudad de México

Febrero, 2019

Al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, a todos los que hicieron posible concluir mi formación de maestría, a los profesores en mi vida escolar, por la formación que me han dado. Indudablemente a mi familia y amigos, es gracias a ustedes que es posible el presente trabajo.

Jonathan Daniel Gallegos Sanlucar.

Agradecimientos

A mis padres Oliva y Juan por el apoyo incondicional durante toda mi vida, son mi más grande admiración, personas que a pesar de las adversidades lograron salir adelante con una sonrisa. A mi hermano Juan y su familia a la cual quiero mucho, gracias por el apoyo en todo momento.

A mis tíos Ángela, Lilia, Aracely, Margarita, especialmente a la doctora Eva y a toda la familia. Por el soporte en los momentos difíciles y por la compañía en momentos gratos.

A mis compañeros y amigos, gracias por su amistad, consejos y muestras de cariño. Especialmente agradezco a Claudia Angélica, Emanuel y Mauricio por el apoyo durante la maestría.

A los médicos que durante este periodo he visitado, por su ayuda y comprensión en todo momento, especialmente a los pertenecientes a la familia, Ángel, Verónica y Cristóbal por su ayuda incondicional.

Al CONACyT por la beca otorgada para realizar mis estudios de maestría.

A los doctores Martín Velasco Villa y Rafael Castro Linares por la oportunidad de trabajar con ellos, su guía durante las clases, sus consejos y su paciencia durante el periodo de trabajo de tesis.

A los auxiliares de la sección, Andrés González Rodríguez, Martha Chávez González e Igor Morett Valenzuela por la ayuda relacionada con el ensamble y puesta en marcha de los robots así como en el soporte para compras de material necesario.

Al CINVESTAV y personal que forman parte de la Sección de Mecatrónica por el apoyo recibido para concluir mis estudios de maestría.

Abstract

The following master's thesis presents the theoretical and experimental development of formation strategy based on the leader-follower method while using the SBO scheme which represents the separation, direction, and angular deviation orientation for a set of three mobile robots. It includes the theory and information needed to confirm the tools used to get the control law and to perform the experiments.

For the development of the controller, with the leader-follower scheme while using SBO scheme, the kinematic models of the robots were used. This development is based on three stages. First on *Backstepping* techniques where a virtual control is defined. The second is based on sliding modes where sliding surfaces were defined. Finally passivity techniques were implemented where an equivalent stability analysis is constructed. This strategy was designed for managed bounded disturbances.

In the first experiment, a straight trajectory for the leader was defined which indirectly obtains the desired trajectory of the follower. In the second experiment, a comparison of two controllers described in the training strategy is made while it is compared with the SBO scheme. In the third experiment, the trajectory is changed into a circular path. The three robots, in the same experiment with the necessary adjustment in the controllers, are used.

The results, in simulation for the experiments, were used to adjust the controller's gains with thus avoiding saturation of the controls in real time experiments. Finally, the proposed controller was validated on an experimental platform. This platform use ROS (*Robot Operating System*), programs in python, vision systems for localization, and computers to be carried out.

Resumen

Este trabajo presenta el desarrollo teórico y experimental de una estrategia de formación basada en la metodología líder-seguidor, utilizando el esquema SBO (*Separation Bearing Orientation*), el cual representa la separación, dirección y desviación angular de la orientación respectivamente, para un conjunto de tres robots móviles.

Se incluye la teoría e información necesaria para validar las herramientas utilizadas para obtener la ley de control y para la implementación de los experimentos. Para el desarrollo teórico del controlador con la formación líder-seguidor, con el esquema SBO, se utilizaron los modelos cinemáticos de los robots. Este desarrollo está basado en técnicas de *Backstepping*, donde se define un control virtual; al definir las superficies de deslizamiento se inducen los modos deslizantes; y pasividad para un análisis de estabilidad equivalente. Esta estrategia se diseñó teniendo en cuenta perturbaciones acotadas.

Los resultados en simulación fueron utilizados para el ajuste de las ganancias del controlador para los experimentos y evaluar propuestas de posibles soluciones. Finalmente se validó el controlador propuesto en la plataforma experimental, la cual utiliza ROS (*Robot Operating System*), programas en python, sistemas de visión para localización y ordenadores para su funcionamiento. En un primer experimento se validó la plataforma experimental, seguido de esto, se definió una trayectoria recta para el líder, con lo cual indirectamente se obtiene la trayectoria deseada del seguidor y lograr formaciones secuenciales. Se realizó otro experimento para realizar una comparación del control propuesto con uno existente en la literatura, al mismo tiempo que se compara con el uso del esquema SBO.

En el último experimento se cambia la trayectoria por una circular y se utilizan los tres robots en el mismo experimento, así como el ajuste necesario en los controladores. Finalmente se presentan conclusiones relacionadas al trabajo de tesis.

Índice general

1. Introducción	1
1.0.1. Topologías fijas y variantes en el tiempo	2
1.0.2. Problema de control líder-seguidor	2
1.0.3. Estrategias robustas en la literatura	3
1.1. Antecedentes	4
1.2. Planteamiento del problema	5
1.3. Objetivos	5
1.3.1. Objetivo general	5
1.3.2. Objetivos específicos	6
1.4. Contribución de la tesis	6
1.5. Organización de la tesis	6
2. Marco teórico	9
2.1. Conceptos de robótica móvil	9
2.1.1. Tipo de llantas	9
2.1.2. Restricciones cinemáticas	12
2.1.2.1. Condición de rodamiento puro	13
2.1.2.2. Condición de no deslizamiento	13
2.1.3. Robot tipo (2,0)	13
2.1.4. Locomoción diferencial	14
2.2. Modelo del robot	15
2.2.1. Restricción del modelo	15
2.3. Técnicas de modos deslizantes	15
2.3.1. Modos deslizantes	16
2.3.1.1. Control equivalente	16
2.3.1.2. Condición de alcanzabilidad	16
2.4. Conceptos importantes de pasividad	17
2.4.1. Removiendo el obstáculo del grado relativo	17
2.5. Técnica de <i>Backstepping</i>	18
2.5.1. Pasivación por retroalimentación con <i>Backstepping</i>	18

3. Estrategias de control para la formación	19
3.1. Seguimiento y formación de trayectorias	19
3.1.1. Errores de seguimiento	20
3.1.2. Solución al problema de seguimiento	21
3.1.3. Estabilidad en lazo cerrado	22
3.2. Solución al problema de formación. Utilización de errores relativos	24
3.2.1. Esquema líder-seguidor	24
3.2.2. Ecuación de los errores	25
3.2.3. Control de la trayectoria	26
3.2.4. Esquema SBO	32
3.2.5. Controles deseados	33
3.2.6. Control de la orientación	34
4. Plataforma experimental	39
4.1. Robots TurtleBot	40
4.1.1. TurtleBot Burger	40
4.1.2. TurtleBot Waffle	41
4.2. Sistema de visión	42
4.2.1. Descripción	43
4.3. Sistemas Computacionales	43
4.3.1. ROS	43
5. Resultados Experimentales	47
5.1. Experimento 1.- Simulación de controladores	48
5.1.1. Controles utilizados	48
5.1.2. Simulación óvalo	48
5.1.3. Simulación flor	51
5.2. Experimento 2.- Validación de la plataforma experimental con un robot	53
5.3. Experimento 3.- Línea recta dos robots	55
5.4. Experimento 4.- Comparación y esquema SBO	59
5.5. Experimento 5.- Tres robots con trayectoria circular	65
6. Conclusiones	71
6.1. Trabajo futuro	72
A. Conceptos de sistemas no lineales	73
A.0.1. Linealización por retroalimentación	73
A.0.2. Grado relativo sistemas SISO	73
A.0.2.1. Sistema de fase mínima	74
A.0.3. Conceptos de pasividad	74

A.0.3.1. Disipatividad	74
A.0.4. Pasividad	74
A.0.4.1. Estabilidad de sistemas pasivos	75
B. Artículo	77
C. Código Experimento	85
D. Instalación TurtleBot3	99
D.0.1. Procedimiento general	99
Bibliografía	103

Introducción

La robótica en la actualidad tiene relevancia a nivel industrial y de investigación científica ya que es utilizada para el transporte de materia con el movimiento del robot sujetando la carga asignada. En éste ámbito la robótica de vehículos terrestres se utiliza en el trasporte de objetos, exploración en ambientes desfavorables para los humanos, reconocimiento de entornos y navegación, entre otras actividades, [1] [2] [3] [4].

El problema general de control de formación en robótica móvil es el de controlar las posiciones y orientaciones relativas de un grupo de robots mientras se controla el movimiento del grupo como un todo. Este tipo de problema de control puede ser implementado en robots móviles de diferente tipo, terrestres, aéreos no tripulados o marítimos. Para la formación de robots móviles se encuentran en la literatura diferentes estructuras que resuelven el problema. Un enfoque es el basado en comportamiento, en el cual la acción de control es definida por pesos promediados del control de cada comportamiento deseado para el agente y donde se puede definir comportamientos como evitar colisión, evadir obstáculos, búsqueda de la meta y mantener la formación, entre otros. Este enfoque es natural para definir varios controles con varios comportamientos; la desventaja de este método es el análisis matemático formal y el análisis de estabilidad [5]. En el enfoque de estructuras virtuales las posiciones deseadas son consideradas como una estructura rígida y para cada robot, la posición deseada está referenciada a un punto de la estructura. Una de las ventajas de este enfoque es la posibilidad de asignar una trayectoria y orientación de la estructura para el desplazamiento del conjunto y, si un robot es perturbado, se mantiene la formación de la estructura. Una de las desventajas es su comportamiento con estructuras que varían con el tiempo o que son reconfiguradas frecuentemente, en la cual no se puede obtener el comportamiento esperado [6].

1.0.1. Topologías fijas y variantes en el tiempo

En [7] se presenta un control de seguimiento de la formación variante en el tiempo para vehículos aéreos no tripulados por medio de la metodología líder-seguidor, en este artículo se presenta una forma de líder virtual para la topología. En [6] se utiliza una topología en la cual se busca mantener una posición fija relativa entre los robots, se utilizan estructuras virtuales en donde se tiene la idea de que cada robot actúa como una partícula que pertenece a un cuerpo rígido. Los autores realizan experimentos con tres robots, se menciona que no es necesario designar a un líder en la formación.

En [8] se presenta un enfoque reconfigurable en tiempo real con el cual se realiza el control centralizado y descentralizado de un conjunto de robots móviles, se analiza también la evasión de obstáculos en el controlador. Se analiza una estrategia de conmutación en la formación en tiempo real.

En [9] se desarrolla un control para la formación de robots móviles, reconfigurable en tiempo real, realizando un control dinámico y cinemático para los robots de la formación, se analiza la estabilidad asintótica y se calcula un tiempo en el cual se asegura la convergencia de los errores en tiempo finito.

En [10] se resuelve el problema de sincronización para un grupo de robots móviles por medio de una ley de control continua variante en el tiempo teniendo en cuenta la condición de rango de Brockett. Se presentan los desarrollos de dos diseños de controladores propuestos, el primero emplea una estrategia de seguimiento de trayectorias preestablecidas y el segundo toma como base las técnicas de sincronización. La convergencia de los errores de seguimiento para las técnicas propuestas, es formalmente demostrada y sus respectivos desempeños se evalúan mediante simulaciones numéricas y experimentos en tiempo real.

1.0.2. Problema de control líder-seguidor

Otro esquema importante es el llamado líder-seguidor como su nombre lo indica, la formación se realiza por medio de estos dos elementos. En la estructura líder-seguidor algunos robots se diseñan como seguidores y otros como líderes en la formación. El líder tiene una trayectoria propia y el seguidor tiene una trayectoria que depende de los estados del líder para generar su posición deseada. Una de las ventajas de la estructura es que la formación puede ser mantenida si el seguidor sufre alguna perturbación; sin embargo, si el líder es perturbado también se reflejará en la formación del seguidor.

En [8] se resuelve el problema líder-seguidor por medio de dos enfoques el cual utiliza la distancia deseada y el ángulo deseado para la posición del seguidor, se continua con otro enfoque donde se obtienen las posiciones deseadas por medio

de la definición de dos parámetros de distancias entre dos líderes y un seguidor. Se utilizan gráficos de control para la transición de las formaciones de los robots, en el artículo realizan la enumeración de estos gráficos en forma sistemática y utilizan una matriz de transición para representar el cambio entre gráficos.

En [11] se utiliza el enfoque líder-seguidor en la representación polar con el uso de cámaras panorámicas. Analizan la observabilidad por medio de una red de cámaras móviles. Aplican el filtro extendido de Kalman para estimar la configuración de los robots y así poder aplicar su control entrada salida, por medio de linealización por retroalimentación.

En [12] se trata el problema de formación, considerando la dinámica de los robots como un doble integrador, con el uso de tres agentes por medio del diseño de un controlador por el método de integrador *Backstepping* y funciones potenciales.

1.0.3. Estrategias robustas en la literatura

En [13] se utiliza un robot móvil de cuatro ruedas, se diseña un controlador para las tres partes del robot donde se puede aplicar una retroalimentación, la cinemática, la dinámica y el modelo de los motores de las ruedas, con la ayuda de la metodología de *Backstepping*, se realizan pruebas de estabilidad para las tres etapas consideradas, obteniéndose una solución globalmente asintóticamente estable.

En [14] se aborda el problema de formación considerando un controlador no lineal por medio de la condición de persistencia a la excitación para la velocidad angular deseada, en ese mismo sentido se busca separar el control para el seguimiento de posición y el control para la orientación. Se realiza la prueba de estabilidad asintótica y se considera este enfoque robusto a perturbaciones.

En [15] se utiliza un enfoque de control cuasi-continuo previamente desarrollado y se rediseña para perturbaciones acotadas bajo restricciones, y con el uso de un observador basado en retroalimentación discontinua, se presenta una prueba formal que garantiza estabilidad asintótica en ciertas condiciones.

En [17] se diseñó un controlador basado en técnicas de *Backstepping*, equivalentes pasivos y modos deslizantes para el manejo de incertidumbres del modelo. Una de las diferencias de [17] con el trabajo de tesis desarrollado se encuentra en las ecuaciones de los errores, en este trabajo de tesis se agrega una separación deseada variante en el tiempo y se modifica el control de la orientación por medio del esquema SBO; se considera que el error angular no es cero para el desarrollo de estas ecuaciones. Referente al controlador implementado no se presenta una

discontinuidad con la velocidad angular deseada en cero por lo tanto no es necesario realizar un análisis para las dos condiciones como sí lo realizan los autores en [17].

1.1. Antecedentes

La organización y control de un conjunto de robots móviles ha tenido un gran auge en la actualidad con el surgimiento de dispositivos manipulables por medio de señales de radiofrecuencia y el uso común de tecnologías como sistemas de localización global. Uno de los ejemplos más claros son los vehículos autónomos los cuales cuentan con diferentes maneras de conocer los obstáculos y su posición para realizar un trayecto determinado. Para subsanar estos problemas, en la literatura se encuentran diferentes enfoques como el basado en comportamiento, los métodos de estructuras virtuales y el enfoque líder-seguidor. Con esto en mente se pretende hacer mención de los trabajos más relevantes que aportaron conocimiento importante para la realización de la tesis. El trabajo realizado en [23] presenta el diseño y análisis de la estrategia líder-seguidor de un conjunto de robots móviles con el uso de cámaras a bordo para la obtención de las posiciones de los robots. Este no es un enfoque global de las posiciones, ya que plantea que solo se conocen las distancias y ángulos relativos entre sí para el cálculo de la ley de control. Utiliza una ley de control basada en modos deslizantes, técnicas de pasivación y técnicas de *Backstepping*. En [23] el control experimental se ajustó, para la no utilización de integradores en la velocidad lineal y angular. La aportación a este trabajo es que provee una estructura similar del control para realizar formaciones con los robots .

En [16] se utiliza la formación líder-seguidor con un control de *horizonte reducido* para resolver el problema de formación de múltiples robots no holonómicos, el control que presentan tiene la finalidad de obtener una rápida convergencia del error de seguimiento. Utilizan dos tipos de esquemas, el SBO (*separation-bearing-orientation*) para dos robots y el SSO (*separation-separation-orientation*) para una formación de tres robots, con el fin de la obtención de las posiciones deseadas para el control utilizado. Los autores mencionan que con este tipo de esquema es posible el control explícito de la desviación angular entre líder y seguidor. Los autores mencionan que es posible realizar una formación cuando los robots se mueven hacia atrás. Finalmente proponen una modificación al control con el esquema de horizonte reducido para optimizar la convergencia de los errores, presentan resultados experimentales para una línea recta que conduce al líder. La aportación de este trabajo a la presente tesis es el uso de la estructura SBO en

el cual se definen tres parámetros necesarios para la implementación del esquema, los cuales son l^d , φ^d y β^d .

1.2. Planteamiento del problema

El trabajo de tesis presentado, propone el uso de un esquema líder-seguir para controlar un conjunto de robots móviles no holónomos en tareas de formación. Se considera en particular la estrategia SBO que permite controlar la orientación de los robots y lograr actividades de formación eficientes.

- I) El problema principal de este trabajo de tesis consiste en realizar maniobras definidas bajo el esquema líder-seguir, con un conjunto de robots móviles por medio de una retroalimentación basada en el esquema SBO.
- II) Utilizar las ecuaciones del error de seguimiento por medio del esquema líder-seguir para cumplir con el seguimiento de la trayectoria deseada.
- III) Diseñar un controlador que haga a los errores del sistema asintóticamente estables o que mantenga los errores en una banda acotada.
- IV) Realizar simulaciones numéricas que ayuden en la evaluación de las retroalimentaciones y al ajuste de las ganancias utilizadas.
- V) Utilizar un conjunto de robots *TurtleBot* de tercera generación mediante la consideración del sistema ROS.
- VI) Complementar el esquema de experimentación mediante el sistema Optitrack y realizar el enlace con el programa de control.

1.3. Objetivos

Una vez establecidos los problemas a resolver se plantean el objetivo general y los objetivos particulares.

1.3.1. Objetivo general

El objetivo del trabajo de tesis es el de diseñar y realizar un controlador para la formación líder-seguir de un grupo de robots móviles con base en el esquema SBO.

1.3.2. Objetivos específicos

- Utilizar las ecuaciones de los errores relativos de seguimiento para la formación.
- Diseñar un controlador que haga al sistema en el error asintóticamente estable.
- Realizar simulaciones numéricas para la validación del modelo y el controlador diseñado.
- Puesta en marcha de la plataforma experimental.
- Realización de un programa de adquisición y envío de datos para la implementación del controlador.
- Obtención y análisis de resultados experimentales.

1.4. Contribución de la tesis

La principal contribución de este trabajo es el diseño de una solución al problema líder-seguidor basada en la estrategia SBO y su implementación en la plataforma experimental basada en ROS.

- Propuesta de un controlador basado en backtepping, pasividad y modos deslizantes.
- La definición de una trayectoria deseada compatible para la orientación angular con el esquema SBO para una rápida convergencia de los errores.
- Desarrollo de un controlador en Python para la implementación del control propuesto.
- Puesta en marcha de los robots *TurtleBot* para obtención de resultados experimentales.

1.5. Organización de la tesis

Este trabajo está dividido en seis capítulos. En el capítulo 1 se encuentra la introducción donde se comenta lo relacionado al trabajo de tesis y se menciona

principalmente las estrategias para resolver el problema de formación. En este capítulo se mencionan trabajos relacionados respecto a metodologías de formación. De forma similar se analizan brevemente los trabajos con mayor aportación al trabajo de tesis. El capítulo 2 muestra los conocimientos base para el entendimiento del controlador y la forma de su diseño, además se mencionan conceptos utilizados en el trabajo de tesis. El capítulo 3 ilustra el desarrollo de la estrategia de control utilizada en la resolución de la problemática planteada. El capítulo 4 muestra la plataforma experimental utilizada y conceptos básicos para llevar a cabo la parte experimental. En el capítulo 5 se encuentra la parte de resultados experimentales relevantes que se produjeron en el desarrollo de la tesis. Finalmente el capítulo 6 presenta las conclusiones del trabajo donde se abordan los temas relevantes y posibles trabajos futuros sobre el tema de investigación.

Marco teórico

En este capítulo se presenta un breve trasfondo teórico de las herramientas matemáticas utilizadas para el desarrollo teórico de la retroalimentación propuesta. Se presenta el modelo de los robots utilizados, así como sus restricciones inherentes. Se presenta teoría para la comprensión de los conceptos utilizados en el desarrollo del controlador. Todo esto con el fin de comprender y llevar a cabo el análisis, desarrollo e implementación del trabajo de tesis.

2.1. Conceptos de robótica móvil

La robótica tiene por objetivo el movimiento y traslado de un robot u objeto de un lugar a otro. En el ámbito de la robótica móvil existen diferentes formas de poner en marcha un robot y generar ese movimiento, pudiendo ser por medio de propulsión, piernas, orugas o ruedas. Para los móviles terrestres el uso de la rueda aumenta la eficiencia del uso de energía y se pueden mover más rápido que otros mecanismos como el de orugas, aunque también surgen limitaciones en su implementación como restricciones inherentes al tipo de locomoción. Para este trabajo de tesis se utilizan móviles con ruedas por lo que se detalla su clasificación empezando por el tipo de llantas utilizadas.

2.1.1. Tipo de llantas

Los robots móviles con ruedas se pueden clasificar en relación al tipo de rueda que los componen, puede ser una rueda convencional o compuesta. Primero se define la rueda convencional la cual cumple con el rodamiento puro sobre un eje, mientras que la rueda compuesta posee un mecanismo de rodamientos o esferas que le permite su movimiento en diferentes sentidos.

2. MARCO TEÓRICO

Para una rueda convencional, como se ve en la Figura 2.1, es necesario definir las distancias de desviación d y b ; conocer si el giro sobre el punto C es permitido, estar al tanto de la determinación de la dirección y la actuación de conducción, esto es, si es conducida o tiene movimiento pasivo.

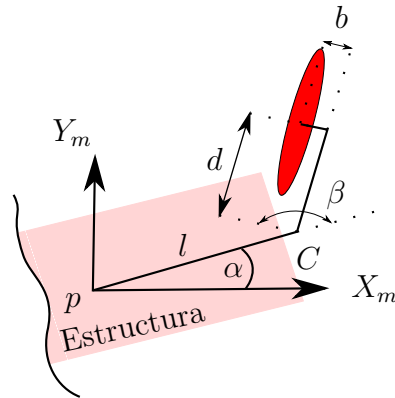


Figura 2.1: Rueda estándar.

De esta configuración general surgen cuatro tipos de ruedas.

- La primera, rueda pasiva fija con $d = 0$.
- La segunda, rueda pasiva de castor con $d \neq 0$.
- La tercera, rueda activa de castor con $d \neq 0$, con actuadores en la rueda y en el eje vertical C .
- La cuarta, rueda activa orientada con $d = 0$ con actuadores en la rueda y en el eje vertical C .

La Figura 2.2 muestra los cuatro tipos de ruedas. Sin embargo en la Figura 2.2b se muestra la segunda y tercer rueda que corresponde a la rueda de castor activa y pasiva.

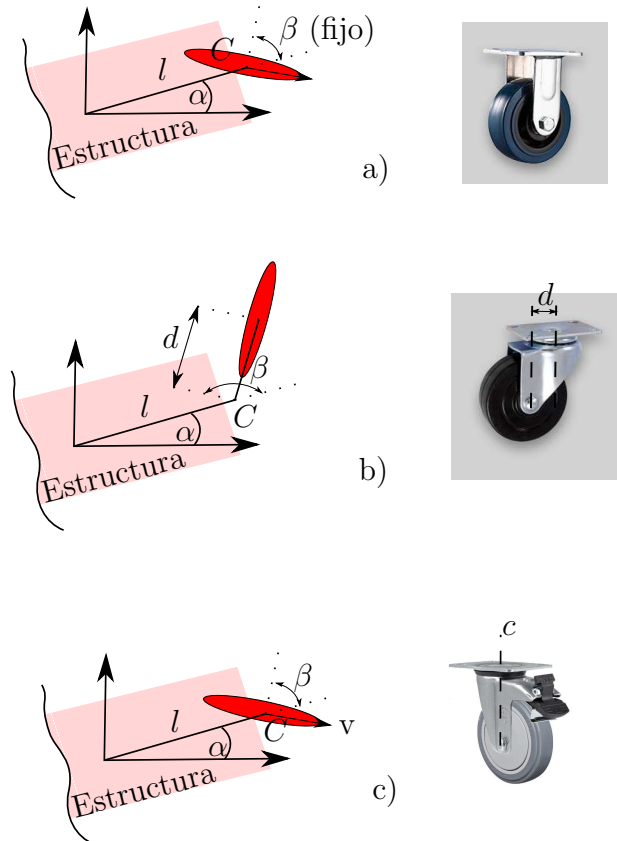


Figura 2.2: Tipos de ruedas convencionales. *a)* Rueda fija pasiva con $d = 0$. *b)* Rueda pasiva o activa de castor con giro en el eje C . *c)* Rueda activa orientable con $d = 0$.

Aunque estas ruedas son confiables y fáciles de construir presentan limitaciones para el movimiento del robot, como por ejemplo la restricción de deslizamiento hacia fuera, o movimiento lateral, de la rueda.

Existen otro tipo de ruedas más complejas como la rueda sueca y la rueda esférica. Estas tienen el objetivo de proveer omnidireccionalidad del móvil.

La rueda sueca presenta la eliminación de la condición no holonómica de velocidad por medio de rodillos como se ve en la Figura 2.3 *a)*. La rueda esférica actuada es difícil de construir y la carga que puede soportar es menor a las convencionales debido a su construcción. La Figura 2.3 *b)* muestra un ejemplo de rueda esférica pasiva sin rodillos, otro de los inconvenientes de estas ruedas es que se contaminan fácilmente y eso podría afectar su funcionamiento [24].

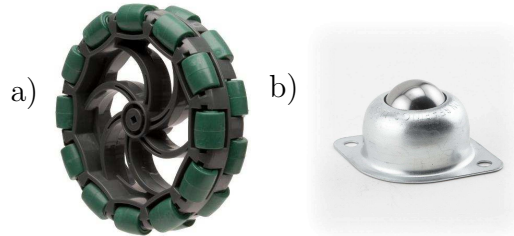


Figura 2.3: Tipos de ruedas especiales. *a)* Rueda sueca. *b)* Rueda esférica .

2.1.2. Restricciones cinemáticas

Se asume que el robot móvil se mueve en un plano, tiene una estructura rígida y llantas no deformables. El vector, que describe los estados del robot, es $\xi = (x \ y \ \theta)^T$ el cual contiene la información de un marco de referencia móvil sobre el robot con respecto a uno inercial. Se asume que durante el movimiento el plano de la rueda es vertical, esto es, las ruedas son perpendiculares al suelo, también las ruedas giran sobre su eje horizontal el cual puede estar fijo o variando en el tiempo. Se asume que el contacto de cada rueda se reduce a un punto. La restricción de velocidad resulta de la suposición de que la velocidad del punto de contacto con el material de la rueda y el piso es cero, esto es, se supone que no existe deslizamiento.

Para las ruedas convencionales se tienen dos condiciones importantes. La de no deslizamiento, esto es que el vector velocidad del centro de la rueda es paralelo al plano de la circunferencia que forma la rueda al moverse, y la condición de rodamiento puro, la cual implica una proporcionalidad de la velocidad angular de la rueda y la velocidad lineal del centro de la misma. Para las ruedas esféricas resulta en una sola condición, esto es que se cumpla la relación de las velocidades lineal y angular en el movimiento. Se muestra ahora la forma de las restricciones cinemáticas para la rueda convencional. El radio de la llanta se denota por r y su ángulo respecto al eje horizontal por φ como se observa en la Figura 2.2b. Los cuatro parámetros α , l , r y d son constantes, mientras que $\varphi(t)$ y $\beta(t)$ son variantes en el tiempo. Se desarrolla la restricción para ésta rueda porque es el caso más general, si es necesaria para los otros dos casos se ajusta el valor de d y $\beta(t)$.

2.1.2.1. Condición de rodamiento puro

La condición de rodamiento puro, mencionada anteriormente, se expresa de la forma siguiente,

$$(-\sin(\alpha + \beta)\cos(\alpha + \beta)l\cos\beta)R(\theta)\dot{\xi} + r\dot{\varphi} = 0 \quad (2.1)$$

con,

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.2)$$

2.1.2.2. Condición de no deslizamiento

La condición de no deslizamiento se presenta de la siguiente manera,

$$(-\cos(\alpha + \beta)\sin(\alpha + \beta)d + l\sin\beta)R(\theta)\dot{\xi} + d\dot{\beta} = 0 \quad (2.3)$$

En general, para un robot de N ruedas. Se denota por el subíndice f para ruedas fijas y S para ruedas direccionales, C para ruedas de castor y SW para ruedas especiales suecas o esféricas. Se tiene que $N = N_f + N_S + N_C + N_{SW}$.

2.1.3. Robot tipo (2,0)

Para los robots móviles, con $N = 3$, existe una clasificación de cinco tipos de vehículos no degenerados, dentro de esa clasificación se encuentra el vehículo tipo (2,0). Este tipo de robots no tiene ruedas direccionales sobre un eje perpendicular a la superficie de rodamiento, tiene una o más ruedas fijas con un eje en común. El *grado de movilidad*, en este caso $m = 2$, tiene el sentido físico de grados de libertad instantáneos o el de velocidades del vehículo que independientemente se pueden asignar sin redireccionar ruedas. El *grado de direccionabilidad*, en este caso $d = 0$, indica el número de ruedas orientables, que se pueden direccionar independientemente en el robot móvil. Un concepto importante es el de *grado de maniobrabilidad*, el cual surge de la adición del *grado de movilidad* y el *grado de direccionabilidad*. Este grado coincide con el número de actuadores posibles en el vehículo con tres ruedas.

La movilidad, en este caso 2, está restringida en el sentido de que, dada una postura ξ , su velocidad debe de cumplir con la condición no holónoma. Un ejemplo típico es la silla de ruedas o los automóviles en una versión simplificada como se indica en [20], donde estos vehículos no se pueden desplazar en cualquier dirección.

2.1.4. Locomoción diferencial

El robot al girar sobre el centro instantáneo de curvatura o rotación, por sus siglas CIC, toma la representación que se muestra en la Figura 2.4. Se tiene que v y w representan las velocidades lineal y angular del móvil, v_L y v_R representan las velocidades lineal de las ruedas izquierda y derecha, es posible realizar algo similar con las velocidades angulares de las ruedas esto es definir w_L y w_R . Se utilizan las variables r y d las cuales representan distancias del radio de la rueda y la distancia del centro del móvil a una rueda respectivamente.

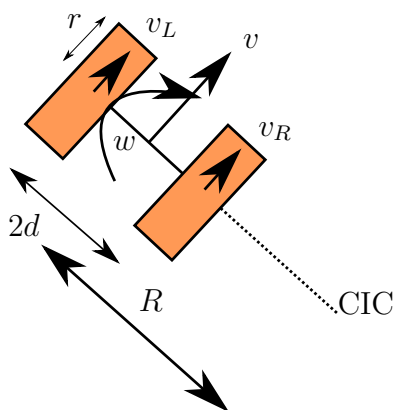


Figura 2.4: Robot diferencial.

Para el robot con movimiento diferencial en la Figura 2.4 se cumple la relación,

$$\begin{aligned} v &= \frac{v_L + v_R}{2} = \frac{rw_L + rw_R}{2} \\ w &= \frac{v_R - v_L}{2d} = \frac{rw_R - rw_L}{2d} \end{aligned} \quad (2.4)$$

o en forma inversa se tiene,

$$\begin{aligned} w_R &= \frac{v + dw}{r} \\ w_L &= \frac{v - dw}{r}. \end{aligned} \quad (2.5)$$

Con esto es posible encontrar una velocidad angular para cada rueda con la velocidad lineal y angular del vehículo.

2.2. Modelo del robot

El modelo del robot utilizado en este trabajo es el descrito por el modelo cinemático de un robot diferencial tipo (2,0) también llamado unicycle. El modelo de la posición y orientación está dado en la forma,

$$\begin{aligned}\dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= w\end{aligned}\tag{2.6}$$

Este modelo cinemático en (2.6) puede ser expresado con respecto a la velocidad de las ruedas con la relación en (2.5). Se observa que solo es necesario encontrar una velocidad lineal y angular para realizar algún movimiento en el vehículo. En el modelo se observa que un cambio en el control angular afecta directamente la orientación del robot. De forma similar un cambio en la velocidad lineal o en la orientación se ve reflejado en las posiciones del robot.

2.2.1. Restricción del modelo

El robot descrito por el modelo (2.6) presenta una restricción no holonómica descrita por,

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0\tag{2.7}$$

ésta restricción es heredada de las restricciones de deslizamiento de las ruedas. El robot se puede mover hacia adelante y atrás, por ejemplo accionando los controles necesarios, pero no puede realizarlo a los lados si se cumple la restricción. Si fuera necesario desplazar al robot hacia un punto lateral, al centro del robot, tendría que realizar una maniobra indirecta para llegar a esa posición y orientación.

2.3. Técnicas de modos deslizantes

El control por modos deslizantes primeramente apareció en el contexto de sistemas con estructuras variables. En la actualidad se utiliza en el control de sistemas de alto orden no lineales con incertidumbres, donde estas técnicas de modos deslizantes son poco sensibles a las variaciones de los parámetros y a perturbaciones. Esta técnica utiliza funciones de conmutación en las entradas del sistema para cumplir los objetivos de control.

2.3.1. Modos deslizantes

Los modos deslizantes surgen cuando se usa una función de conmutación en el control del sistema. La conmutación en alta frecuencia hace que el modo deslizante ocurra. Por ejemplo, el modo deslizante, aparece en sistemas dinámicos de ecuaciones diferenciales donde se presentan funciones discontinuas. Esto implica que, por ejemplo, el control como función de los estados del sistema conmuta a alta frecuencia [18].

2.3.1.1. Control equivalente

Para un sistema de la forma,

$$\dot{x} = f(x, u) \quad x, f \in \mathbb{R}^n \quad u(x) \in \mathbb{R}^m \quad (2.8)$$

se asume que el estado inicial está en la intersección de las discontinuidades, es decir, en la superficie $s(x) = 0$ y el modo deslizante ocurre con las trayectorias confinadas en esta superficie para $t > 0$. Se asume que $\frac{ds}{dt} = \dot{s}(x) = 0$, con ello es posible caracterizar la trayectoria de los estados durante el modo deslizante. Sin tomar en cuenta el control discontinuo se calcula el vector u_{eq} (control equivalente) tal que la derivada del vector en los estados de la trayectoria en (2.8) es igual a cero, esto es,

$$\dot{s}(x) = Gf(x, u) = 0 \quad (2.9)$$

donde $G = \frac{\partial s}{\partial x}$ es una matriz de $m \times n$ con filas de gradientes de funciones $s_i(x)$. Si la solución a (2.9) existe entonces es llamada control equivalente.

Desde una perspectiva geométrica el control equivalente es el que reemplaza al control discontinuo en la intersección de las superficies de conmutación, tal que el vector de velocidad está en una superficie tangencial. Desde un punto de vista físico el control equivalente fue pensado como una idealización, se puede decir que el control equivalente es cercano a la componente lenta del control real el cual es derivado por quitar las componentes de alta frecuencia con un filtro pasa bajas.

2.3.1.2. Condición de alcanzabilidad

La condición de alcanzabilidad es una ecuación diferencial la cual especifica la dinámica de una función de conmutación [25]. La ecuación diferencial de una superficie deslizante $s(x)$ asintóticamente estable se le nombra como una condición de alcanzabilidad. Con la elección de los parámetros en la ecuación diferencial se controla la calidad de la dinámica del sistema en el modo alcanzable como lo

muestra [25]. Posibles asignaciones se pueden realizar a la dinámica de la superficie.

Por ejemplo alcanzabilidad constante, esto es,

$$\dot{s}(x) = Q \operatorname{sgn}(s(x)) \quad (2.10)$$

donde Q es constante. La alcanzabilidad proporcional o con razón fija más una componente, esto es,

$$\dot{s}(x) = Q \operatorname{sgn}(s(x)) - Ks(x) \quad (2.11)$$

donde K es constante. Por último es posible asignar una forma exponencial a la dinámica de la superficie, esto es,

$$\dot{s}(x) = -K|s(x)|^\alpha \operatorname{sgn}(s(x)). \quad (2.12)$$

2.4. Conceptos importantes de pasividad

La herramienta de pasividad es inducida por un diseño por retroalimentación pasiva. Aunque algunos diseños estaban restringidos a sistemas de fase mínima débiles con grado relativo uno. Con los diseños recursivos se remueve esta última restricción. Se presentan los conceptos básicos para el desarrollo de la estrategia de control. El diseño por retroalimentación pasiva comienza con la elección de una salida para un grado relativo uno. Continúa con la búsqueda de una función de almacenamiento $S(x)$ para la razón de suministro $u^T y$. En este contexto la función de almacenamiento puede ser tratada como una candidata de Lyapunov. El fenómeno de *peaking* es un obstáculo para la estabilidad global y semiglobal de sistemas en cascada. Para el diseño de una ley de control es necesaria la construcción de una función de almacenamiento apropiada. Esta construcción se puede realizar por medio de un término cruzado o por composición de las funciones de Lyapunov de la dinámica cero y del sistema controlado. Conceptos importantes se presentan en el apéndice A, los cuales se analizaron de [19].

2.4.1. Removiendo el obstáculo del grado relativo

El uso de *Backstepping* ayuda a remover la restricción del uso de pasividad a sistemas con grado relativo uno, es decir es posible utilizar la técnica en los sistemas de mayor grado relativo, por medio de la definición de controles virtuales en cada paso. El primer paso para realizar esto es aislar el sistema y realizar el procedimiento por *Backstepping* y pasividad al sistema reducido como se menciona en [19].

2.5. Técnica de *Backstepping*

El *Backstepping* es una técnica para el diseño de controles estabilizantes que se utiliza puntualmente en sistemas dinámicos no lineales. Surge en los sistemas irreducibles que pueden ser estabilizados por otros métodos. Debido a su estructura recursiva, es posible diseñar un control para la dinámica más interna tomando en cuenta que el sistema más interno es estabilizable. Es posible ir hacia afuera con el mismo principio surgiendo nuevos controles que estabilizan al sistema en cada etapa. El proceso finaliza encontrando el control más externo donde se utilizan funciones de Lyapunov para lograr estabilidad de los sistemas analizados.

2.5.1. Pasivación por retroalimentación con *Backstepping*

Este diseño nos ayuda a elegir una salida para un sistema con grado relativo uno y que sea un sistema débilmente de fase mínima. La técnica de *Backstepping* exacta es más sencilla de implementar, respecto al *Backstepping* robusto, ya que no considera un control para los parámetros desconocidos. La técnica de *Backstepping* robusto considera las perturbaciones en el diseño del controlador. El primer paso en ambos casos es encontrar salidas pasivantes y funciones de almacenamiento, esta última puede ser utilizada como función candidata de Lyapunov. El segundo paso es definir un control para obtener estabilidad asintótica global en cada paso. El procedimiento para obtener las salidas se realiza desde la ecuación más cercana al controlador, mientras que el diseño del controlador se obtiene de la parte más alejada del control hacia el exterior, el controlador se diseña analizando la función de almacenamiento. El procedimiento es similar en cada caso sin embargo la dificultad no lo es, entre más anidados estén los controles virtuales, se obtienen derivadas de los controles virtuales de mayor orden, lo cual implica dificultad en el diseño como se menciona en [19], las complicaciones se presentan cuando se analiza el enfoque robusto, el cual podría implementar técnicas de *dominación* de términos para tener estabilidad asintótica.

Estrategias de control para la formación

Para la estrategia de control líder-seguidor, se diseñó una retroalimentación de control junto al esquema SBO el cual permite asignar una orientación deseada a cada robot como se describe en [16]. Se plantea primero el problema de seguimiento.

3.1. Seguimiento y formación de trayectorias

Se considera que la formación sigue una trayectoria mientras actúan los robots de forma individual, manteniéndose en la trayectoria y cambiando de posición alrededor de ella. Un ejemplo de formación se presenta en la Figura 3.1, esto está dado en tres etapas, la formación en triángulo, paso por el estrecho en cadena y reagrupación al final nuevamente con la formación triangular.

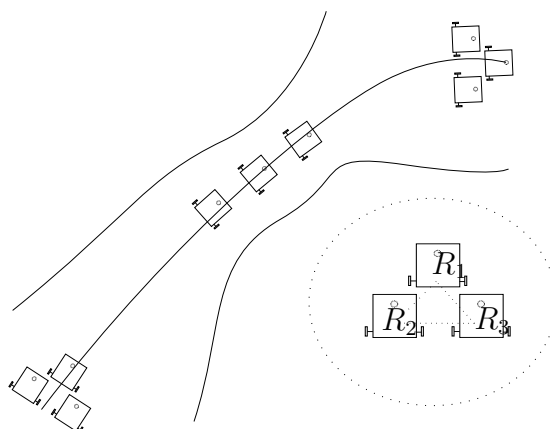


Figura 3.1: Ejemplo de formación general.

3. ESTRATEGIAS DE CONTROL PARA LA FORMACIÓN

El líder de grupo es nombrado como R_1 y los seguidores como R_i con $i = 2, 3..n$ con n el número de robots.

El problema primeramente se centra unicamente en el control de formación. El problema consiste en un grupo de n robots móviles no-holónomos que son controlados para moverse a lo largo de una trayectoria, dirigidos por un líder R_1 , el cual su configuración determina el movimiento de la formación. Los robots R_2 hasta R_n son controlados para seguir al líder por medio de la relación líder-seguidor, con el control de la separación, ángulo de separación y orientación de los seguidores respecto a los líderes.

Considerando robots diferenciales con restricciones no-holónomas, se tiene que las ecuaciones cinemáticas que describen la posición y velocidad del sistema para los robots son,

$$\begin{aligned}\dot{x}_i &= v_i \cos \theta_i \\ \dot{y}_i &= v_i \sin \theta_i \\ \dot{\theta}_i &= w_i\end{aligned}\tag{3.1}$$

donde x_i , y_i en las ecuaciones (3.1) son las coordenadas en el plano, y θ_i es la orientación del robot, además v_i y w_i son las velocidades lineales y angulares para los robots respectivamente.

3.1.1. Errores de seguimiento

Se define la diferencia en las posiciones en la forma, $p_i^e = p_i^d - p_i$, donde $p_i = [x_i, y_i, \theta_i]$ representa las estados actuales de cada robot, p_i^d representa los estados deseados, p_i^d se considera con una dinámica similar a (3.1). Se realiza una representación del vector de diferencias en la posición en las coordenadas del líder para obtener el vector de error de postura, esto se representa con la ecuación 3.2, esto es,

$$\begin{aligned}e_{x,i} &= \cos(\theta_i)(x_i^d - x_i) + \sin(\theta_i)(y_i^d - y_i) \\ e_{y,i} &= -\sin(\theta_i)(x_i^d - x_i) + \cos(\theta_i)(y_i^d - y_i) \\ e_{\theta,i} &= \theta_i^d - \theta_i.\end{aligned}\tag{3.2}$$

Derivando las ecuaciones anteriores se produce,

$$\begin{aligned}\dot{e}_{x,i} &= -\dot{\theta}_i \sin(\theta_i)(x_i^d - x_i) + \cos(\theta_i)(\dot{x}_i^d - \dot{x}_i) \\ &\quad + \dot{\theta}_i \cos(\theta_i)(y_i^d - y_i) + \sin(\theta_i)(\dot{y}_i^d - \dot{y}_i) \\ \dot{e}_{y,i} &= -\dot{\theta}_i \cos(\theta_i)(x_i^d - x_i) - \sin(\theta_i)(\dot{x}_i^d - \dot{x}_i) \\ &\quad - \dot{\theta}_i \sin(\theta_i)(y_i^d - y_i) + \cos(\theta_i)(\dot{y}_i^d - \dot{y}_i) \\ \dot{e}_{\theta,i} &= \dot{\theta}_i^d - \dot{\theta}_i\end{aligned}\tag{3.3}$$

es posible obtener (3.3) en términos de los errores y de las velocidades deseadas. Tomando en cuenta (3.1) para los estados actuales, considerando que similarmente $p_i^d = [x_i^d, y_i^d, \theta_i^d]$ cumple con el modelo cinemático en (3.1), considerando que $(\cos \theta_i)^2 + (\sin \theta_i)^2 = 1$ y con la definición en (3.2) se obtienen las ecuaciones,

$$\begin{aligned}\dot{e}_{x,i} &= w_i e_{y,i} + \dot{x}_i^d \cos(\theta_i) + \dot{y}_i^d \sin(\theta_i) - v_i \\ \dot{e}_{y,i} &= -w_i e_{x,i} - \dot{x}_i^d \sin(\theta_i) + \dot{y}_i^d \cos(\theta_i) \\ \dot{e}_{\theta,i} &= w_i^d - w_i.\end{aligned}\quad (3.4)$$

Considerando que $\theta_i = \theta_i^d - e_{\theta,i}$ la ecuación (3.4) toma la forma,

$$\begin{aligned}\dot{e}_{x,i} &= w_i e_{y,i} + \dot{x}_i^d \cos(\theta_i^d - e_{\theta,i}) + \dot{y}_i^d \sin(\theta_i^d - e_{\theta,i}) - v_i \\ \dot{e}_{y,i} &= -w_i e_{x,i} - \dot{x}_i^d \sin(\theta_i^d - e_{\theta,i}) + \dot{y}_i^d \cos(\theta_i^d - e_{\theta,i}) \\ \dot{e}_{\theta,i} &= w_i^d - w_i\end{aligned}\quad (3.5)$$

con la ecuación anterior y tomando en cuenta la expansión de los términos, $\sin(\theta_i^d - e_{\theta,i})$, $\cos(\theta_i^d - e_{\theta,i})$ en (3.5), así como las restricciones no-holónomas de (2.7), es posible obtener,

$$\begin{aligned}\dot{e}_{x,i} &= (\cos(\theta_i^d) \dot{x}_i^d + \sin(\theta_i^d) \dot{y}_i^d) \cos(e_{\theta,i}) + e_{y,i} w_i - v_i \\ \dot{e}_{y,i} &= (\cos(\theta_i^d) \dot{x}_i^d + \sin(\theta_i^d) \dot{y}_i^d) \sin(e_{\theta,i}) - e_{x,i} w_i \\ \dot{e}_{\theta,i} &= w_i^d - w_i.\end{aligned}\quad (3.6)$$

Tomando en cuenta que la trayectoria deseada cumple con (3.1) es posible encontrar que,

$$v_i^d = \dot{y}_i^d \sin(\theta_i^d) + \cos(\theta_i^d) \dot{x}_i^d.\quad (3.7)$$

Finalmente sustituyendo (3.7) en (3.6) se obtiene la dinámica de los errores como,

$$\begin{aligned}\dot{e}_{x,i} &= v_i^d \cos(e_{\theta,i}) + e_{y,i} w_i - v_i \\ \dot{e}_{y,i} &= v_i^d \sin(e_{\theta,i}) - e_{x,i} w_i \\ \dot{e}_{\theta,i} &= w_i^d - w_i.\end{aligned}\quad (3.8)$$

3.1.2. Solución al problema de seguimiento

A partir de [16] se propone la siguiente retroalimentación,

$$\begin{aligned}v_i &= v_i^d \cos e_{\theta,i} + k_{x,i} e_{x,i} \\ \omega_i &= \omega_i^d + k_{\theta,i} e_{\theta,i} + v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}.\end{aligned}\quad (3.9)$$

3. ESTRATEGIAS DE CONTROL PARA LA FORMACIÓN

Sustituyendo la retroalimentación (3.9) en el sistema en el error (3.8) se obtiene,

$$\begin{aligned}
 \dot{e}_{x,i} &= v_i^d \cos(e_{\theta,i}) + e_{y,i}(\omega_i^d + k_{\theta,i}e_{\theta,i} + v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}) - (v_i^d \cos e_{\theta,i} + k_{x,i}e_{x,i}) \\
 \dot{e}_{y,i} &= v_i^d \sin(e_{\theta,i}) - e_{x,i}(\omega_i^d + k_{\theta,i}e_{\theta,i} + v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}) \\
 \dot{e}_{\theta,i} &= w_i^d - (\omega_i^d + k_{\theta,i}e_{\theta,i} + v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}).
 \end{aligned} \tag{3.10}$$

Finalmente, simplificando términos se tiene la ecuación en lazo cerrado del sistema,

$$\begin{aligned}
 \dot{e}_{x,i} &= e_{y,i}(\omega_i^d + k_{\theta,i}e_{\theta,i} + v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}) - k_{x,i}e_{x,i} \\
 \dot{e}_{y,i} &= v_i^d \sin(e_{\theta,i}) - e_{x,i}(\omega_i^d + k_{\theta,i}e_{\theta,i} + v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}) \\
 \dot{e}_{\theta,i} &= -k_{\theta,i}e_{\theta,i} - v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}.
 \end{aligned} \tag{3.11}$$

3.1.3. Estabilidad en lazo cerrado

Se asume que v_i^d, w_i^d son acotadas en $t \in [0, \infty)$. Además si $v_i^d \neq 0$, el error e_i converge asintóticamente a cero, es decir $e_i \rightarrow 0$ cuando $t \rightarrow \infty$.

Considere la función candidata de Lyapunov como,

$$V(e_i) = \frac{1}{2}(e_{x,i}^2 + e_{y,i}^2 + e_{\theta,i}^2). \tag{3.12}$$

Tomando la derivada respecto al tiempo,

$$\dot{V}(e_i) = e_{x,i}\dot{e}_{x,i} + e_{y,i}\dot{e}_{y,i} + e_{\theta,i}\dot{e}_{\theta,i}$$

con lo cual,

$$\begin{aligned}
 \dot{V}(e_i) &= e_{x,i}e_{y,i}(\omega_i^d + k_{\theta,i}e_{\theta,i} + v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}) - k_{x,i}e_{x,i}^2 + e_{y,i}v_i^d \sin(e_{\theta,i}) \\
 &\quad - e_{x,i}e_{y,i}(\omega_i^d + k_{\theta,i}e_{\theta,i} + v_i^d e_{y,i} \frac{\sin e_{\theta,i}}{e_{\theta,i}}) - k_{\theta,i}e_{\theta,i}^2 - v_i^d e_{y,i} \sin(e_{\theta,i})
 \end{aligned}$$

Finalmente se obtiene,

$$\begin{aligned}
 \dot{V}(e_i) &= -k_{x,i}e_{x,i}^2 - k_{\theta,i}e_{\theta,i}^2 \\
 &= -k_{x,i}e_{x,i}^2 - k_{\theta,i}e_{\theta,i}^2 \leq 0.
 \end{aligned} \tag{3.13}$$

La función $V(e_i)$ es no creciente para todo e_i , por lo tanto $e_{x,i}$ y $e_{\theta,i}$ son acotadas. Debido a que $\frac{d}{dt}(\dot{V})$ es acotada \dot{V} es uniformemente continua. La consideración del Lema de Barbalat permite concluir que, $e_{x,i}, e_{\theta,i} \rightarrow 0$ cuando $t \rightarrow \infty$. Considerando $e_{x,i} = 0$ y $e_{\theta,i} = 0$ se tiene a partir de 3.11 que,

$$0 = -v_i^d e_{y,i} \tag{3.14}$$

Por lo tanto si $v_i^d \neq 0$ se tiene que $e_{y,i} = 0$ cuando $t \rightarrow \infty$.

3.2. Solución al problema de formación. Utilización de errores relativos

3.2.1. Esquema líder-seguidor

Para la estrategia de control se utilizará una definición relativa de los errores por medio de la proyección de la distancia entre los robots, con estas ecuaciones se obtuvo una ley de control retroalimentada para resolver el problema de formación de los robots diferenciales como se muestra en [23].

El control por formación tiene el objetivo de implementar una trayectoria definida por medio de la obtención de posiciones deseadas para el conjunto de robots con una retroalimentación que incluya las proyecciones del conjunto. Este esquema se muestra en la Figura 3.2, donde se presenta una formación de dos robots con el fin de simplificar el análisis para un conjunto de n robots. Se observan dos robots en el plano, el líder y el seguidor.

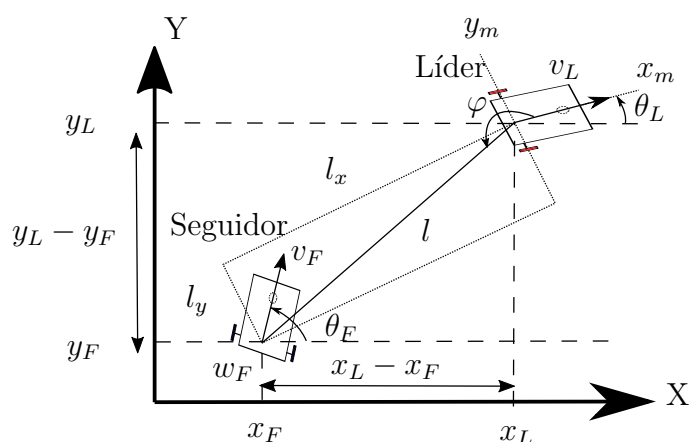


Figura 3.2: Esquema líder-seguidor.

En la Figura 3.2, \mathbf{X} y \mathbf{Y} representan el marco inercial, mientras que x_m y y_m son el marco móvil fijo al cuerpo del líder; se representa la nomenclatura para las posiciones del líder y seguidor con los subíndices L y F respectivamente. La distancia de separación entre ambos vehículos se representa por la letra l y el ángulo deseado entre ellos por φ . Finalmente θ representa el ángulo con respecto a la horizontal (eje \mathbf{X}) dependiendo del móvil indicado, v y w representa la velocidad lineal y angular de los móviles respectivamente.

3.2.2. Ecuación de los errores

El modelo cinemático para el robot líder está dado por,

$$\begin{aligned}\dot{x}_L &= v_L \cos \theta_L \\ \dot{y}_L &= v_L \sin \theta_L \\ \dot{\theta}_L &= w_L\end{aligned}\tag{3.15}$$

donde v_L y w_L representa la velocidad lineal y angular respectivamente. De forma similar para el seguidor se tiene,

$$\begin{aligned}\dot{x}_F &= v_F \cos \theta_F \\ \dot{y}_F &= v_F \sin \theta_F \\ \dot{\theta}_F &= w_F\end{aligned}\tag{3.16}$$

A partir de la Figura 3.2, se define la proyección de la separación l entre los robots como,

$$\begin{aligned}l_x &= -(x_L - x_F) \cos(\theta_L) - (y_L - y_F) \sin(\theta_L) \\ l_y &= (x_L - x_F) \sin(\theta_L) - (y_L - y_F) \cos(\theta_L).\end{aligned}\tag{3.17}$$

Al definir $e_\theta = \theta_F - \theta_L$, junto con las identidades trigonométricas,

$$\begin{aligned}\cos(\theta_F - \theta_L) &= \sin(\theta_L) \sin(\theta_F) + \cos(\theta_L) \cos(\theta_F) \\ \sin(\theta_F - \theta_L) &= \cos(\theta_L) \sin(\theta_F) - \sin(\theta_L) \cos(\theta_F)\end{aligned}$$

se tiene que, a partir de los modelos (3.15) y (3.16), las derivadas con respecto del tiempo de l_x , l_y y e_θ están dadas por,

$$\begin{aligned}\dot{l}_x &= \cos(e_\theta) v_F - v_L + l_y w_L \\ \dot{l}_y &= \sin(e_\theta) v_F - l_x w_L \\ \dot{e}_\theta &= w_F - w_L.\end{aligned}\tag{3.18}$$

Definiendo ahora los errores de seguimiento e_x y e_y como,

$$\begin{aligned}e_x &= l_{xd} - l_x \\ e_y &= l_{yd} - l_y\end{aligned}\tag{3.19}$$

con,

$$\begin{aligned}l_{xd} &= l_d \cos(\varphi_d) \\ l_{yd} &= l_d \sin(\varphi_d)\end{aligned}\tag{3.20}$$

3. ESTRATEGIAS DE CONTROL PARA LA FORMACIÓN

siendo l_d la distancia de separación deseada entre los robots líder y seguidor, mientras que φ_d es el ángulo deseado entre ellos. Se tiene entonces que la dinámica de los errores de seguimiento está dada por,

$$\dot{e}_x = w_L e_y - \cos(e_\theta) v_F + f_1 + \Delta f_1 \quad (3.21a)$$

$$\dot{e}_y = -\sin(e_\theta) v_F - w_L e_x + f_2 + \Delta f_2 \quad (3.21b)$$

$$\dot{e}_\theta = w_F - w_L, \quad (3.21c)$$

con,

$$\begin{aligned} f_1 &= -l_d (\dot{\varphi}_d) \sin(\varphi_d) - w_L l_d \sin(\varphi_d) \\ &\quad + (\dot{l}_d) \cos(\varphi_d) + v_L \end{aligned} \quad (3.22a)$$

$$\begin{aligned} f_2 &= l_d (\dot{\varphi}_d) \cos(\varphi_d) + w_L l_d \cos(\varphi_d) \\ &\quad + (\dot{l}_d) \sin(\varphi_d). \end{aligned} \quad (3.22b)$$

Nótese que f_1 y f_2 son funciones suaves y acotadas ya que dependen de los valores deseados l_d , φ_d y de las señales de control del vehículo líder, v_L , w_L que pueden también considerarse acotadas. Δf_1 y Δf_2 son términos inciertos que pueden estar asociados a errores de modelado o a perturbaciones externas y se supone que están acotados, más precisamente, se supone que satisfacen,

$$\Delta f_1 \leq \eta_1, \quad \Delta f_2 \leq \eta_2 \quad (3.23)$$

donde η_1 y η_2 son constantes reales positivas diferentes de cero.

3.2.3. Control de la trayectoria

La estrategia de control para la formación líder-seguidor se diseñó en base al concepto de equivalencia de sistemas pasivos por retroalimentación (pasivación) en conjunto con las técnicas de *Backstepping* y de modos deslizantes [17]. Como primer paso de diseño se considera la dinámica del error e_x dada por (3.21a) con $\Delta f_1 = 0$, suponiendo que el término $w_L e_y$ es una entrada virtual y denotándolo como $\alpha(e_x)$, es decir,

$$\alpha(e_x) = e_y w_L. \quad (3.24)$$

Se define la función de conmutación,

$$s_1 = e_x + k_1 \int_0^t e_x(\tau) d\tau \quad (3.25)$$

donde k_1 es una constante real. Esta función de conmutación permite definir la llamada superficie deslizante $s_1 = 0$. En esta superficie, $\dot{s}_1 = 0$ y la dinámica del error e_x con esta superficie está dada por,

$$\dot{e}_x + k_1 e_x = 0. \quad (3.26)$$

Entonces al escoger adecuadamente k_1 se tiene que $e_x(t) \rightarrow 0$ cuando $t \rightarrow \infty$. Para que las trayectorias de (3.21a) sean atraídas a la superficie deslizante $s_1 = 0$, se hace la asignación [18],

$$\dot{s}_1 = F_1(s_1) = -\gamma_1 \text{sign}(s_1) \quad (3.27)$$

donde $\gamma_1 > 0$ y $\text{sign}(\cdot)$ es la función signo. A partir de la expresión para s_1 , ecuación (3.25), y la dinámica en (3.21a) con $\Delta f_1 = 0$, se tiene que el control virtual $\alpha(e_x)$ está dado por,

$$\alpha(e_x) = -k_1 e_x - \gamma_1 \text{sign}(s_1) + v_F \cos(e_\theta) - f_1. \quad (3.28)$$

De hecho, es posible dar una condición suficiente que garantiza la atracción a $s_1 = 0$ como se establece en el siguiente resultado.

Teorema 1. *Considérese la dinámica (3.21a) donde el término incierto Δf_1 satisface (3.23), si la ganancia γ_1 en (3.27) satisface,*

$$\gamma_1 > \eta_1 \quad (3.29)$$

entonces el control virtual (3.28) hace que las trayectorias del sistema (3.21a) converjan a $s_1 = 0$.

Demostración. Considérese la candidata a función de Lyapunov,

$$V_1 = \frac{1}{2} s_1^2 \quad (3.30)$$

la cual es definida positiva. Al tomar la derivada de V_1 con respecto al tiempo y sustituyendo el control virtual (3.28) en la dinámica (3.21a) se tiene que,

$$\dot{V}_1 = -s_1 \gamma_1 \text{sign}(s_1) + s_1 \Delta f_1 \quad (3.31)$$

3. ESTRATEGIAS DE CONTROL PARA LA FORMACIÓN

ya que $\text{sign}(x)x = |x|$ y al mayorar (3.31) junto con la condición (3.23) se tiene que V_1 está acotada cómo,

$$\dot{V}_1 \leq -\gamma_1|s_1| + \eta_1|s_1| \quad (3.32)$$

o bien,

$$\dot{V}_1 \leq -(\gamma_1 - \eta_1)|s_1|.$$

Así que si $\gamma_1 > \eta_1$, se tiene la convergencia de s_1 a 0. \square

Considerando ahora la función,

$$y = e_y w_L - \alpha(e_x) \quad (3.33)$$

como salida del sistema (3.21), se tiene que,

$$e_y w_L = y + \alpha(e_x).$$

Sustituyendo el valor anterior en (3.21a), permite escribir e_x como,

$$\dot{e}_x = y + \alpha(e_x) - \cos(e_\theta) v_F + f_1 + \Delta f_1$$

que al sustituir el control virtual $\alpha(e_x)$ se reescribe como,

$$\dot{e}_x = y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1. \quad (3.34)$$

Por otro lado, derivando la ecuación (3.33) con respecto al tiempo y sustituyendo en ella \dot{e}_y de (3.21b), conduce a la expresión,

$$\begin{aligned} \dot{y} &= (-\sin(e_\theta) v_F - w_L e_x + f_2 + \Delta f_2) w_L \\ &+ e_y \dot{w}_L - \dot{\alpha}(e_x) \end{aligned} \quad (3.35)$$

donde,

$$\begin{aligned} \dot{\alpha}(e_x) &= -k_1 \dot{e}_x - \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) \frac{\partial s_1}{\partial t} \\ &+ \dot{v}_F \cos(e_\theta) - v_F \sin(e_\theta) \dot{e}_\theta - \dot{f}_1. \end{aligned} \quad (3.36)$$

Sustituyendo (3.34) en la ecuación anterior y considerando $\dot{e}_\theta = \dot{\theta}_F - \dot{\theta}_L$, permite reescribir $\dot{\alpha}(e_x)$ como,

$$\begin{aligned} \dot{\alpha}(e_x) &= -k_1 [y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1] \\ &- \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) \frac{\partial s_1}{\partial t} + \dot{v}_F \cos(e_\theta) \\ &- v_F \sin(e_\theta) (\dot{\theta}_F - \dot{\theta}_L) - \dot{f}_1. \end{aligned} \quad (3.37)$$

Consecuentemente, \dot{y} toma la forma,

$$\begin{aligned}\dot{y} &= (-\sin(e_\theta) v_F - w_L e_x + f_2 + \Delta f_2) w_L \\ &+ e_y \dot{w}_L - [-k_1 [y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1] \\ &- \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) \frac{\partial s_1}{\partial t} + \dot{v}_F \cos(e_\theta) \\ &- v_F \sin(e_\theta) (\dot{\theta}_F - \dot{\theta}_L) - \dot{f}_1].\end{aligned}$$

Tomando en cuenta la expresión de la dinámica de e_x dada por (3.34), \dot{s}_1 se reescribe como,

$$\begin{aligned}\dot{s}_1 &= \dot{e}_x + k_1 e_x \\ &= y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 + k_1 e_x \\ &= y - \gamma_1 \text{sign}(s_1) + \Delta f_1.\end{aligned}$$

Entonces, la dinámica de y está dada por,

$$\dot{y} = \alpha(\cdot) - \dot{v}_F \cos(e_\theta) + \Delta f \quad (3.38)$$

donde,

$$\begin{aligned}\alpha(\cdot) &= (-\sin(e_\theta) v_F - w_L e_x + f_2) w_L \\ &+ e_y \dot{w}_L + k_1 [y - k_1 e_x - \gamma_1 \text{sign}(s_1)] \\ &+ \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) [y - \gamma_1 \text{sign}(s_1)] \\ &+ v_F \sin(e_\theta) (w_F - w_L) + \dot{f}_1 \\ \Delta f &= \Delta f_2 w_L + k_1 \Delta f_1 + \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) \Delta f_1.\end{aligned} \quad (3.39)$$

A partir de los cálculos anteriores se tiene el siguiente sistema,

$$\dot{e}_x = y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 \quad (3.40a)$$

$$\dot{y} = \alpha(\cdot) - \dot{v}_F \cos(e_\theta) + \Delta f. \quad (3.40b)$$

Se busca ahora una retroalimentación que haga pasivo al sistema (3.40) considerando que $\Delta f = 0$. Con este fin se elige un control auxiliar de la forma,

$$\dot{v}_F = \frac{1}{\cos e_\theta} (\alpha(\cdot) - v) \quad (3.41)$$

donde v es una nueva entrada de control, obteniéndose el sistema,

$$\begin{aligned}\dot{e}_x &= y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 \\ \dot{y} &= v.\end{aligned} \quad (3.42)$$

3. ESTRATEGIAS DE CONTROL PARA LA FORMACIÓN

Considerando ahora que,

$$v = -s_1 + w \quad (3.43)$$

se obtiene el sistema,

$$\dot{e}_x = y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 \quad (3.44a)$$

$$\dot{y} = -s_1 + w. \quad (3.44b)$$

Es posible probar que el control auxiliar \dot{v}_F de (3.41) y v en (3.43) hacen pasivo al sistema en (3.40) desde la entrada w hasta la salida y , considerando $\Delta f = 0$. Para tal efecto, se considera una función de almacenamiento V definida por,

$$V = V_1(s_1) + V_2(y) = \frac{1}{2}s_1^2 + \frac{1}{2}y^2. \quad (3.45)$$

Tomando la derivada temporal a lo largo de las trayectorias del sistema (3.44), y utilizando el hecho de que $\dot{s}_1 = \dot{e}_x + k_1 e_x$ junto a (3.44b), se obtiene,

$$\dot{V} = s_1 (\dot{e}_x + k_1 e_x) + y (-s_1 + w). \quad (3.46)$$

Simplificando términos, se tiene que \dot{V} se puede escribir como,

$$\begin{aligned} \dot{V} &= s_1 (y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 + k_1 e_x) \\ &\quad + y (-s_1 + w) \\ &= s_1 (y - \gamma_1 \text{sign}(s_1) + \Delta f_1) + y (-s_1 + w) \\ &= s_1 y - \gamma_1 s_1 \text{sign}(s_1) + \Delta f_1 s_1 - s_1 y + w y \\ &= -\gamma_1 s_1 \text{sign}(s_1) + \Delta f_1 s_1 + w y \\ &= -\gamma_1 |s_1| + \Delta f_1 s_1 + w y. \end{aligned} \quad (3.47)$$

Tomando en cuenta la cota de Δf_1 , ecuación (3.23), \dot{V} se puede mayorar como,

$$\dot{V} \leq -\gamma_1 |s_1| + \eta_1 |s_1| + w y.$$

Considerando que $\gamma_1 > \eta_1$, se obtiene,

$$\dot{V} \leq -(\gamma_1 - \eta_1) |s_1| + w y.$$

Así que,

$$\dot{V} \leq w y. \quad (3.48)$$

Entonces la retroalimentación que logra hacer pasivo al sistema en (3.40) desde la entrada w hasta la salida y con $\Delta f = 0$, se escribe como,

$$\dot{v}_F = \frac{1}{\text{cose}\theta} (\alpha(\cdot) + s_1 - w). \quad (3.49)$$

Para mantener la propiedad de pasividad cuando $\Delta f \neq 0$, se define una segunda función de conmutación,

$$s_2 = y - \int_0^t v(\tau) d\tau. \quad (3.50)$$

Cuando ocurre un modo deslizante, el sistema se restringe a evolucionar sobre la superficie $s_2 = 0$ o, equivalentemente,

$$\dot{y} = v.$$

Esto se logra por medio del control equivalente para esta función de conmutación, el cual coincide con el control dado por (3.49), es decir,

$$\dot{v}_{Feq} = \frac{1}{\cos\theta} (\alpha(\cdot) + s_1 - w). \quad (3.51)$$

Una vez que las trayectorias son atraídas a la superficie, el control anterior asegura que permanezcan en ella. Se observa que el nuevo estado obtenido es estado cero observable por lo que es posible asignar una retroalimentación de la forma,

$$w = -k_2 y \quad (3.52)$$

la cual estabiliza asintóticamente al sistema (3.40) [18]. Si un punto inicial del sistema $e_x(0)$, $y(0)$ no pertenece a la superficie $s_2 = 0$, es posible atraer las trayectorias a ésta asignando,

$$\dot{s}_2 = \dot{y} - v = F_2(s_2) = -\gamma_2 \text{sign}(s_2) \quad (3.53)$$

obteniendo,

$$\dot{y} = v - \gamma_2 \text{sign}(s_2). \quad (3.54)$$

Para lograr este comportamiento en el sistema (3.40) el control, \dot{v}_F está dado por,

$$\dot{v}_F = \frac{1}{\cos\theta} (\alpha(\cdot) - (v - \gamma_2 \text{sign}(s_2))). \quad (3.55)$$

Sustituyendo v de (3.41) con w de (3.52) y simplificando, \dot{v}_F se escribe como,

$$\dot{v}_F = \frac{1}{\cos\theta} (\alpha(\cdot) + s_1 + k_2 y + \gamma_2 \text{sign}(s_2)). \quad (3.56)$$

O, en forma equivalente,

$$\dot{\xi} = \frac{1}{\cos\theta} (\alpha(\cdot) + s_1 + k_2 y + \gamma_2 \text{sign}(s_2)) \quad (3.57a)$$

$$v_F = \xi. \quad (3.57b)$$

3.2.4. Esquema SBO

Las siglas SBO hacen referencia a los términos *Separation-Bearing-Orientation*; *separation* hace mención a la distancia relativa del líder con el seguidor, *bearing* es igual al ángulo medido desde el eje coordenado x_m asignado al líder hasta la línea de separación del líder con el seguidor, finalmente *orientation* es el ángulo deseado agregado a la orientación deseada, en la Figura 3.3 se observan las variables utilizadas. La descripción de la posición del seguidor por medio de las coordenadas conocidas del líder se utiliza en el diseño de la trayectoria, como se indica en [16]. La trayectoria deseada del seguidor respecto al marco base resulta en,

$$x_F^d = x_L + l_d \cos \varphi_d \quad (3.58a)$$

$$y_F^d = y_L + l_d \sin \varphi_d \quad (3.58b)$$

$$\theta_F^d = \theta_L + \beta_d. \quad (3.58c)$$

El esquema SBO utiliza tres parámetros para poder definir las posiciones deseadas en este esquema como se observa en la Figura 3.3. La distancia de separación l_d , el ángulo entre el líder y el seguidor llamado φ_d y un tercer parámetro el ángulo β_d entre el robot líder con el seguidor.

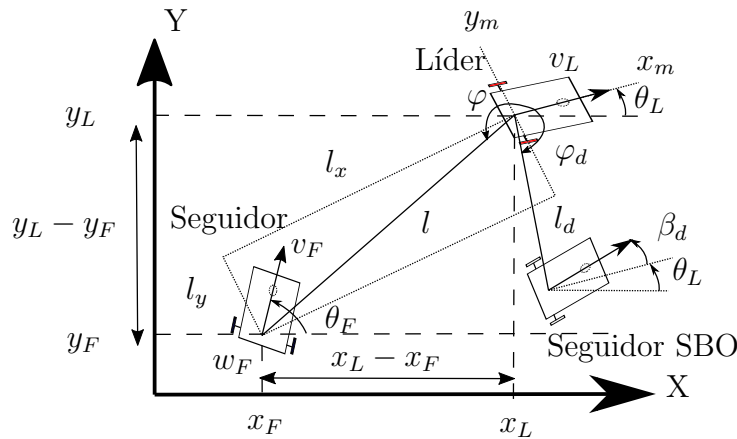


Figura 3.3: Esquema SBO.

Para un esquema donde se indican las posiciones deseadas x_F^d y y_F^d , para cualquier robot seguidor, es posible definir el ángulo deseado de la trayectoria deseada al mismo tiempo que cumple con la restricción no holónoma,

$$\dot{x}_F^d \sin(\theta_F^d) - \dot{y}_F^d \cos(\theta_F^d) = 0$$

esto es,

$$\theta_F^d = \text{atan}\left(\frac{\dot{y}_F^d}{\dot{x}_F^d}\right). \quad (3.59)$$

Se observa que las velocidades de las posiciones deseadas definen el ángulo deseado. En (3.59) se representa la orientación tangente a la trayectoria deseada utilizando las velocidades lineales de las posiciones deseadas, esta orientación carece del termino β_d y no utiliza la orientación del líder. Sin embargo si se utiliza el esquema SBO, presentado en (3.58), es posible asignar una desviación de la orientación β_d respecto a la orientación dada por el líder θ_L , por lo tanto la orientación con el esquema SBO se define como,

$$\theta_{F,SBO}^d = \theta_L + \beta_d. \quad (3.60)$$

El ángulo deseado β_d no puede ser cualquier ángulo arbitrario, por lo que se calcula posteriormente.

3.2.5. Controles deseados

Para la implementación del control es necesario obtener las velocidades deseadas para cada robot i . Se sabe que las velocidades lineal y angular están dadas por las siguientes ecuaciones,

$$\begin{aligned} v_F^d &= \dot{y}_F^d \sin(\theta_F^d) + \cos(\theta_F^d) \dot{x}_F^d \\ w_F^d &= \dot{\theta}_F^d. \end{aligned} \quad (3.61)$$

Derivando la posición deseada (3.58a) se obtiene,

$$\dot{x}_F^d = \dot{x}_L + l_d(-\sin(\varphi_d)\dot{\varphi}_d) + \dot{l}_d \cos(\varphi_d). \quad (3.62)$$

La aceleración correspondiente resulta,

$$\begin{aligned} \ddot{x}_F^d &= \ddot{x}_L + l_d \frac{d}{dt}(-\sin(\varphi_d)\dot{\varphi}_d) + \dot{l}_d(-\sin(\varphi_d)\dot{\varphi}_d) \\ &\quad + \dot{l}_d(-\sin(\varphi_d)\dot{\varphi}_d) + \ddot{l}_d \cos(\varphi_d) \\ &= \ddot{x}_L + l_d(-(\sin(\varphi_d)\ddot{\varphi}_d + \cos(\varphi_d)(\dot{\varphi}_d)^2)) \\ &\quad + \dot{l}_d(-\sin(\varphi_d)\dot{\varphi}_d) + \dot{l}_d(-\sin(\varphi_d)\dot{\varphi}_d) + \ddot{l}_d \cos(\varphi_d) \end{aligned} \quad (3.63)$$

esto es,

$$\begin{aligned} \ddot{x}_F^d &= \ddot{x}_L + -l_d \sin(\varphi_d)\ddot{\varphi}_d - l_d \cos(\varphi_d)(\dot{\varphi}_d)^2 \\ &\quad + 2\dot{l}_d(-\sin(\varphi_d)\dot{\varphi}_d) + \ddot{l}_d \cos(\varphi_d). \end{aligned} \quad (3.64)$$

3. ESTRATEGIAS DE CONTROL PARA LA FORMACIÓN

De manera similar para (3.58a) se obtiene,

$$\dot{y}_F^d = \dot{y}_L + l_d(\cos(\varphi_d)\dot{\varphi}_d) + \dot{l}_d \sin(\varphi_d). \quad (3.65)$$

La aceleración resulta,

$$\begin{aligned} \ddot{y}_F^d &= \ddot{y}_L + l_d \frac{d}{dt}(\cos(\varphi_d)\dot{\varphi}_d) + \dot{l}_d(\cos(\varphi_d)\dot{\varphi}_d) \\ &\quad + \dot{l}_d(\cos(\varphi_d)\dot{\varphi}_d) + \ddot{l}_d \sin(\varphi_d) \\ &= \ddot{y}_L + l_d(\cos(\varphi_d)\ddot{\varphi}_d - \sin(\varphi_d)(\dot{\varphi}_d)^2) \\ &\quad + \dot{l}_d(\cos(\varphi_d)\dot{\varphi}_d) + \dot{l}_d(\cos(\varphi_d)\dot{\varphi}_d) + \ddot{l}_d \sin(\varphi_d) \end{aligned} \quad (3.66)$$

obteniéndose finalmente,

$$\begin{aligned} \ddot{y}_F^d &= \ddot{y}_L + l_d \cos(\varphi_d)\ddot{\varphi}_d - l_d \sin(\varphi_d)(\dot{\varphi}_d)^2 \\ &\quad + 2\dot{l}_d \cos(\varphi_d)\dot{\varphi}_d + \ddot{l}_d \sin(\varphi_d). \end{aligned} \quad (3.67)$$

Partiendo de la posición angular se obtiene la velocidad angular deseada,

$$w_F^d = \dot{\theta}_F^d = \frac{\frac{d}{dt} \left(\frac{\dot{y}_F^d}{\dot{x}_F^d} \right)}{1 + \left(\frac{\dot{y}_F^d}{\dot{x}_F^d} \right)^2} = \frac{\left(\frac{\dot{x}_F^d \ddot{y}_F^d - \ddot{x}_F^d \dot{y}_F^d}{(\dot{x}_F^d)^2} \right)}{1 + \left(\frac{\dot{y}_F^d}{\dot{x}_F^d} \right)^2} \quad (3.68)$$

de donde es posible mostrar que,

$$w_F^d = \frac{\dot{x}_F^d \ddot{y}_F^d - \ddot{x}_F^d \dot{y}_F^d}{(\dot{x}_F^d)^2 + (\dot{y}_F^d)^2}. \quad (3.69)$$

3.2.6. Control de la orientación

Cuando el robot seguidor se encuentra sobre la trayectoria (véase Figura 3.4), una posible orientación deseada se calcula como,

$$\theta_F^d = \arctan \left(\frac{\dot{y}_F^d}{\dot{x}_F^d} \right) \quad (3.70)$$

esta orientación es definida por las componentes de la velocidad lineal deseada.

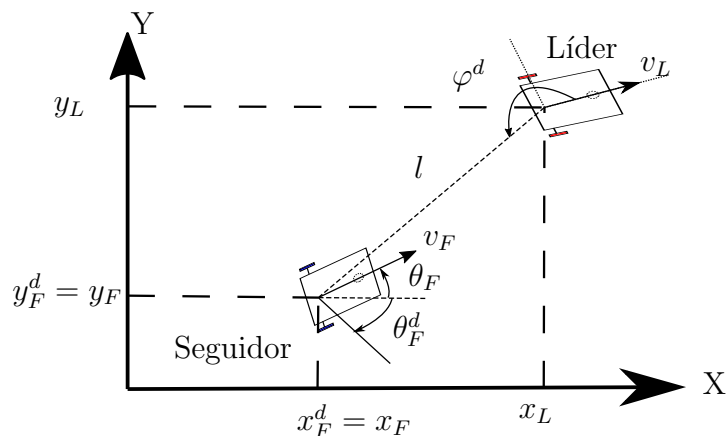


Figura 3.4: Orientación deseada dentro de la trayectoria.

La orientación deseada, cuando el robot se encuentra fuera de la trayectoria, se calcula utilizando la Figura 3.5, es decir,

$$\theta_{FO}^d = \arctan\left(\frac{y_F^d - y_F}{x_F^d - x_F}\right). \quad (3.71)$$

Esta orientación es definida por la posición actual del seguidor y su posición deseada. La orientación representa al ángulo directo desde la posición actual a la posición deseada.

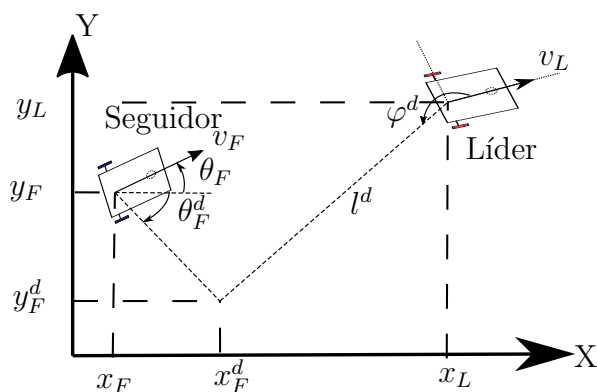


Figura 3.5: Orientación deseada fuera de la trayectoria.

Con estas dos definiciones en cuenta, el problema es llevar una orientación variante en el tiempo a otra por medio de alguna variable que defina las regiones

3. ESTRATEGIAS DE CONTROL PARA LA FORMACIÓN

mencionadas.

Se propone algo similar a lo presentado en Figura 3.6, donde se muestra un ejemplo con dos funciones, una recta aproximándose a una función senoidal, para obtener una función resultante. La gráfica del parámetro que define la transición es la que corresponde a la recta con pendiente negativa, este parámetro define tres regiones. En la Figura 3.6 se observa que de 0-40s la función resultante es una recta; de 40-60s existe una región de transición; y finalmente de 60-100s es donde la gráfica resultante se convierte en una función senoidal. La función constante representa el valor del parámetro donde se desea la región de transición. Esto se desarrolló por medio de las funciones, N y O descritas más adelante.

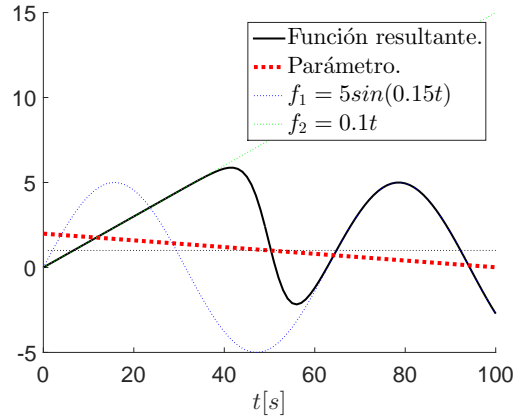


Figura 3.6: Una recta aproximándose a una función senoidal por medio de una función decreciente.

Retomando el planteamiento de las orientaciones, representadas en (3.70) y (3.71), el valor de la posición angular deseada se propone como,

$$\theta_{FA}^d = \arctan \frac{(y_F^d - y_F)N + \dot{y}_F^d O}{(x_F^d - x_F)N + \dot{x}_F^d O} \quad (3.72)$$

donde,

$$\begin{aligned} N &= \frac{1}{2} (\tanh(\epsilon_1(\zeta - \lambda_1)) + 1) \\ O &= \frac{1}{2} (-\tanh(\epsilon_2(\zeta - \lambda_2)) + 1) \end{aligned} \quad (3.73)$$

con

$$\zeta = \sqrt{e_x^2 + e_y^2}. \quad (3.74)$$

Se remarca que (3.72) es diferente a (3.71) y (3.70) debido a que (3.72) utiliza los errores de seguimiento e_x y e_y para definir una norma ζ y con eso establecer regiones “dentro” y “fuera” de la trayectoria, si $\zeta > \lambda_1$ y $\zeta > \lambda_2$ se define la región “fuera” de la trayectoria, si $\zeta \leq \lambda_1$ y $\zeta \leq \lambda_2$ se define la región “dentro” de la trayectoria, con $\lambda_{1,2}$ constantes iguales. $\epsilon_{1,2}$ son constantes para tener una transición suave de una región a otra. Con las funciones, O y N , la definición (3.72) se aproxima a (3.71) cuando el parámetro ζ se encuentra en la zona definida como “fuera” de la trayectoria deseada. De manera similar la ecuación (3.72) se aproxima con (3.70) cuando el parámetro ζ se encuentra “dentro” de la trayectoria deseada, en general (3.72) tiende a (3.70) conforme el parámetro ζ decrementa. Se obtuvo el valor para la variable β_F^d como,

$$\beta_d = \arctan \frac{(y_F^d - y_F)N + \dot{y}_F^d O}{(x_F^d - x_F)N + \dot{x}_F^d O} - \theta_L. \quad (3.75)$$

Por lo tanto,

$$\theta_F^d = \theta_L + \beta_d. \quad (3.76)$$

Finalmente se define el control w_F de la forma,

$$w_F = \dot{\theta}_F^d - k_3 e_{\theta F} \quad (3.77)$$

donde,

$$e_{\theta F} = \theta_F^d - \theta_F.$$

Debido a la necesidad de derivar la función signo, para el cálculo de $\alpha(\cdot)$ en la ecuación (3.39), se utilizó una aproximación por medio de la función tangente hiperbólica, esto es,

$$\text{sign}(s_1) \approx \tanh(\beta s_1)$$

con lo cual,

$$\frac{\partial}{\partial s_1} \text{sign}(s_1) \approx \beta (\text{sech}(\beta s_1))^2$$

donde el parámetro $\beta = 100$ controla la pendiente de la curva y la forma de su derivada.

Plataforma experimental

La plataforma consiste en un conjunto de robots móviles tipo (2,0) formado por los robots *Turtlebot* de tercera generación. Los seguidores del tipo *Burger* y un líder del tipo *Waffle* de la compañía ROBOTIS. La programación de estos robots está basada en ROS (*Robot Operating System*), para sistemas Linux. Se utiliza un programa maestro que enlaza las señales de control con los robots por medio de un programa elaborado en Python, ver apéndice C. La Figura 4.1 muestra la estructura general de la plataforma experimental.

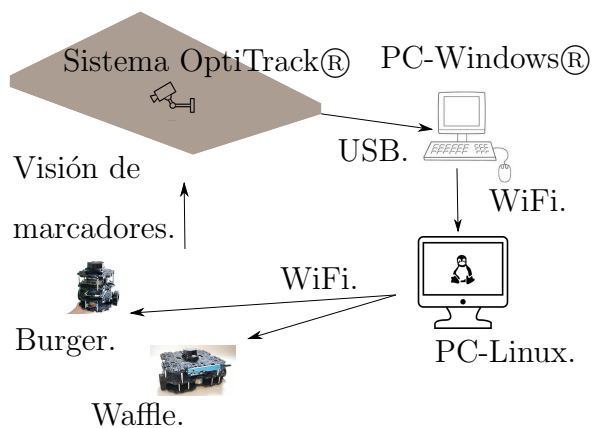


Figura 4.1: Estructura general de la plataforma experimental.

El sistema de localización utilizado fue un conjunto de doce cámaras *Flex 3* de *OptiTrack* ® y las interconexiones con el software *Motive*® de la compañía *NaturalPoint*.

4.1. Robots TurtleBot

Diversas plataformas de robots son utilizadas en el mercado, con el fin de validar los experimentos para sistemas móviles. Para esta tesis se realizó la instalación y puesta en marcha de los Robots *TurtleBot3* correspondientes a la parte experimental del proyecto, ver apéndice D.

TurtleBot es una plataforma robótica estándar que utiliza ROS para la comunicación. Existen tres versiones de la serie *TurtleBot*, la que se describe a continuación es la última generación de los robots a la fecha. Los robots *TurtleBot3* son pequeños y programables es por eso que son usados para investigación, educación o para desarrollo de productos. El objetivo de esta última versión es reducir el tamaño y precio respecto a sus antecesores sin pérdida de funcionalidad. Se describe brevemente los dos tipos de robots utilizados.

4.1.1. TurtleBot Burger

El robot *TurtleBot Burger* se conforma de una estructura de cuatro placas las cuales soportan a los componentes principales. Comenzando desde la parte inferior se tiene la identificación de los motores y su controlador conformado por el módulo *Dynamixel*. Sobre la primer base se encuentra también la batería de LiPo 11.1 v 1800mAh 5C de tres celdas. En el segundo nivel se encuentra la placa *OpenCR* la cual dirige los comandos del módulo *dynamixel* y hace posible la comunicación con la tarjeta principal. La tarjeta donde se instaló el sistema operativo se encuentra en la tercera placa y es la *Raspberry Pi 3* modelo B la cual administra las conexiones con los sensores y actuadores. Finalmente se tiene el módulo LDS el cual es el sensor láser de distancia que permite detectar objetos dentro de cierto rango de visión. La Figura 4.2 muestra la distribución principal de los componentes.

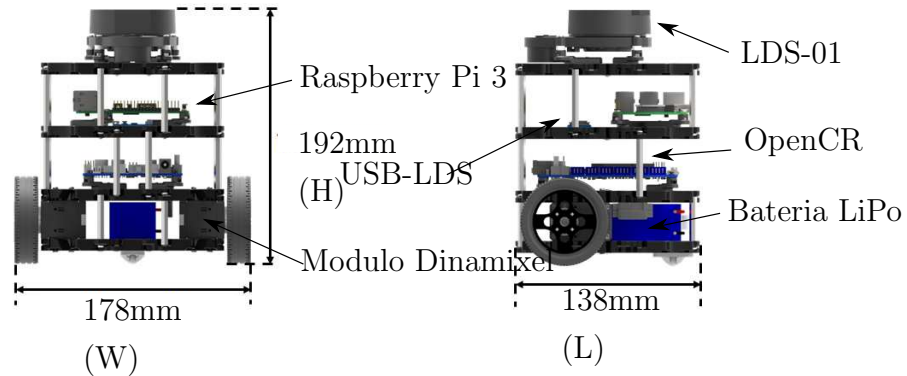


Figura 4.2: Componentes principales del robot *Burger*.

4.1.2. TurtleBot Waffle

El robot *TurtleBot Waffle* se conforma de una estructura de cuatro placas las cuales soportan a los componentes principales. Comenzando desde la parte inferior se tiene la identificación de los motores y su controlador conformado por el módulo Dinamixel. Sobre la primer base se encuentra también la batería de LiPo 11.1 v 1800mAh 5C de tres celdas. La segunda placa soporta a las tarjetas *OpenCR* y a la tarjeta principal *Intel® Joule™*. Finalmente la parte superior soporta al sensor LDS-01 y a la cámara *Intel® Realsense™ R200*. La Figura 4.3 muestra la distribución principal de los componentes.

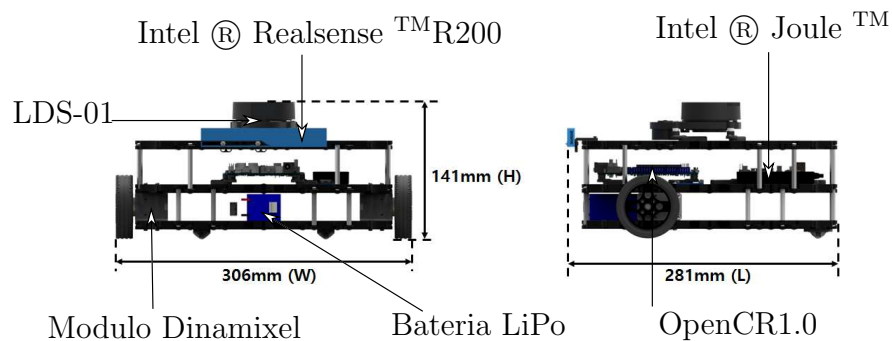


Figura 4.3: Componentes principales del robot *Waffle*.

4.2. Sistema de visión

Existen diferentes usos para la visión artificial, como lo puede ser el reconocimiento de objetos, medición de distancias u orientaciones, la detección de eventos, entre otras aplicaciones. Los sistemas de visión en la práctica son utilizados para saber, por ejemplo si en un vehículo autónomo, se va o no en la posición correcta en el carril de conducción. Esto se puede averiguar por medio de técnicas que analizan el flujo óptico. Otra posible aplicación es la de reconocer personas en sistemas de seguridad. Un uso más frecuente es el uso industrial en la automatización de procesos, en sistemas de calidad y para el control de procesos [22].

Estos sistemas pasan por un proceso similar para ser de utilidad. El proceso general es la adquisición de una imagen en la cámara, posteriormente el procesamiento de la imagen para convertir los datos de los pixeles a conjunto de matrices. Por último se utilizan los algoritmos para realizar la acción deseada ya sea para medir distancias, contar objetos o reconocerlos.

Los sistemas de visión, orientados a la robótica, son utilizados, por ejemplo, para localizar un cuerpo en el plano al mismo tiempo que se conoce su ubicación y orientación respecto a un marco de referencia. De forma similar se puede implementar para cuerpos descritos en tres dimensiones.

El sistema de visión está formado por doce cámaras *Flex 13*, un *OptiHub* e interconexiones necesarias. Esto forma parte del sistema *OptiTrack*TM de la compañía *NaturalPoint Inc*. Un esquema para enviar los datos de las posiciones a la computadora con el software *Motive* se muestra en la Figura 4.4.

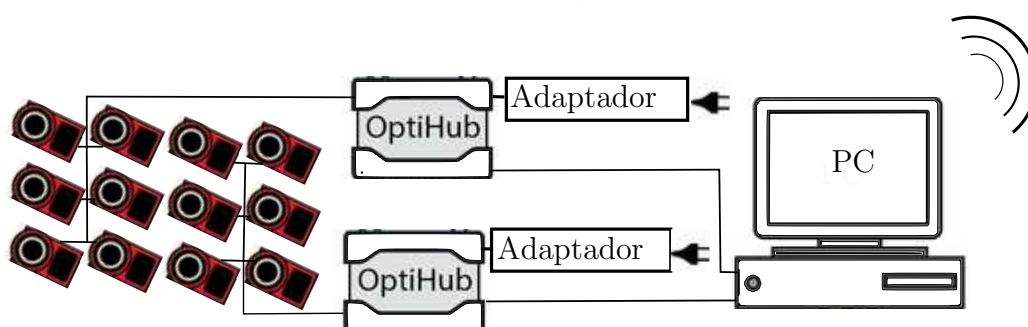


Figura 4.4: Composición del sistema de visión.

4.2.1. Descripción

Las cámaras *Flex 13* tienen la capacidad de procesamiento a bordo para la entrega de datos ya procesados en la computadora central, todo esto con el fin de aumentar la eficiencia del envío de datos. Una de las características importantes es el control de la exposición de luz y la manipulación de la velocidad de muestreo. Estas camaras tienen una resolución de 1.3MP, latencia de 8.3 ms y 30-120 FPS de muestreo.

El adaptador *OptiHub* se utiliza para la sincronización de las cámaras, el dispositivo cuenta con dos modos uno externo y otro interno, este último es el que se utiliza en el experimento, se necesita un adaptador de 12v para cada módulo.

Finalmente en la computadora de escritorio, con sistema operativo Windows 8, se cuenta con el programa *Motive* de la misma compañía. Este software realiza el análisis de los datos provenientes de las cámaras. Este programa cuenta con la opción de envío de datos por medio de un puerto y de esta forma ser leído remotamente en una red local.

4.3. Sistemas Computacionales

Se utilizaron cuatro computadoras, una de ellas la correspondiente a la tarjeta *Intel-Joule* para al robot *Waffle*. Dos de ellas cuentan con la tarjeta *raspberry Pi 3B*, utilizadas con los dos robots *Burger*. Finalmente se tiene una computadora, con un procesador *intel*, que administra el envío de datos a los robots. Estos dispositivos utilizan *Ubuntu 16.04 (Linux)* como sistema operativo.

4.3.1. ROS

El manejo de los robots y sus componentes, tiene relevancia en la industria y en investigación a nivel mundial. Con esta problemática en mente, se han desarrollado herramientas para un manejo más sencillo de los parámetros en un sistema robótico.

Una de estas herramientas es ROS (*Robotic Operating System*), el cual es un sistema operativo robótico que se desarrolla sobre un sistema basado en Linux. Debido a esto ROS hereda ciertas funciones como lo son, el manejo de los archivos, comunicación entre programas y un enfoque orientado a la terminal. Este

4. PLATAFORMA EXPERIMENTAL

sistema operativo robótico presenta diferentes herramientas para el manejo de datos, programas, sensores y actuadores de forma remota en una red local. Esto facilita la interconexión de procesos ejecutados en Python o en lenguaje C, todo esto es posible por medio de un programa maestro que internamente administra los permisos y procesos en la red.

ROS provee los servicios estandarizados de un sistema operativo tales como abstracción del hardware, control de dispositivos de bajo nivel, paso de mensajes entre procesos y mantenimiento de paquetes.

El sistema ROS se basa en una arquitectura de grafos donde el procesamiento toma lugar en los nodos que pueden recibir, mandar y multiplexar mensajes de sensores, controles, estados, planificaciones y actuadores. La librería está orientada para un sistema Linux.

ROS es software libre bajo términos de licencia BSD. Esta licencia permite el uso comercial y su uso en la investigación científica. Las contribuciones de los paquetes en ROS están bajo una variedad de licencias.

Los conceptos de oyente (*listener*) y comunicador (*talker*) en el sistema ROS son de gran importancia ya que con esto es posible intercambiar información. El nodo o programa oyente recibe los datos de cierto tópico, el cual debe estar indicado en el nodo receptor, en este proceso se realiza la suscripción a un tópico. El nodo comunicador envía, a cierta frecuencia, los datos por medio de la publicación en el tópico en cuestión. La Figura 4.5 muestra un proceso simplificado del envío y recepción de datos por dos nodos.

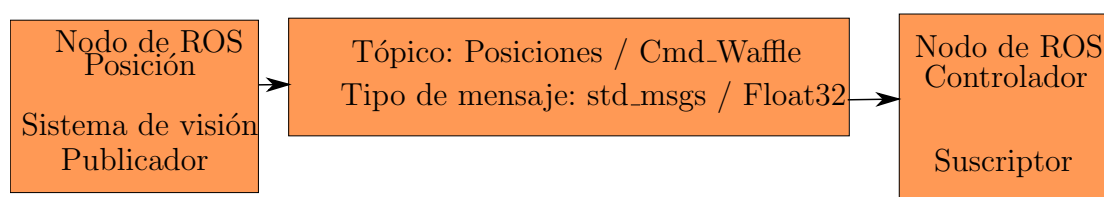


Figura 4.5: Sistema básico de funcionamiento de recepción y envío de datos por medio de ROS.

El nodo ROS máster es el encargado de la administración de las conexiones por medio de P2P entre procesos. La conexión de estos procesos se da por medio de los tópicos, en los cuales necesariamente se tienen que definir dos características importantes, el nombre del tema y el tipo de mensaje que se maneja.

La estructura general de los programas implementados en ROS se muestra en la Figura 4.6, donde se observa el uso de tres programas principales. El primero

es el sistema de visión, el segundo es el controlador y el tercero es el *driver* de los robots para la respectiva plataforma. El proceso comienza con el nodo *ROS Master* el cual administra las conexiones en la red. Se continúa con el nodo del sistema de visión para utilizar los datos del sistema *OptiTrack*, aquí las posiciones de los robots ya se encuentran en la red de ROS y pueden ser utilizadas por cualquier miembro. El nodo *driver* es el encargado de realizar las conexiones con la tarjeta SBC (*Single Board Computer*) o *OpenCR* (*Open-source Control Module for ROS*) y preparar al robot para la recepción de los datos de las velocidades. Finalmente se describe el controlador, el cual por medio de un buffer recibe los datos de la red, los procesa y envía las señales de control a los robots.

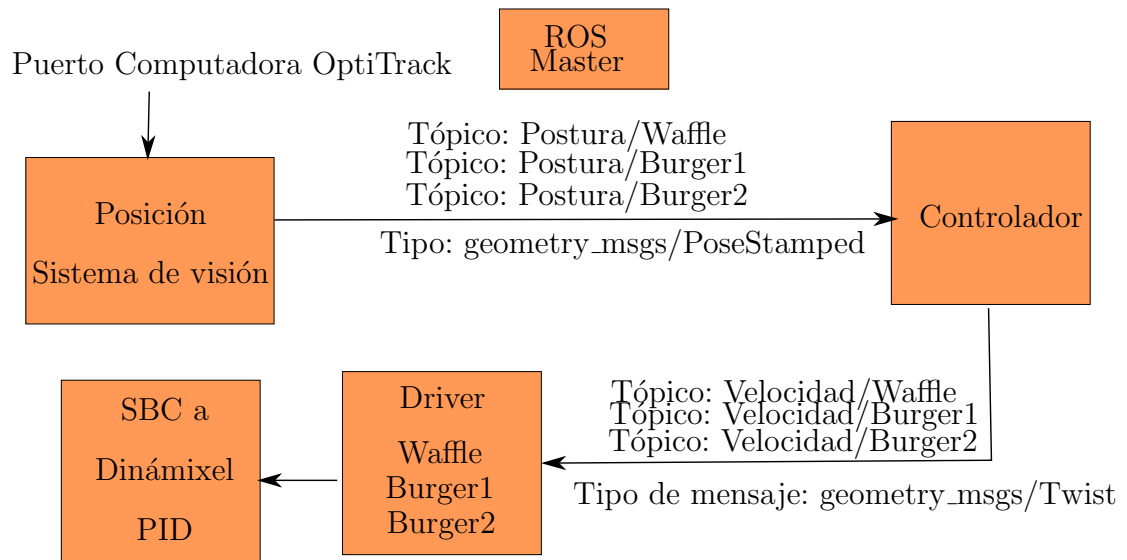


Figura 4.6: Sistema General del desarrollo en ROS.

Los tipos de mensajes utilizados se muestran en la Figura 4.6. Primero se describe el mensaje para recibir los datos, (**PoseStamped**), el cual está conformado por el encabezado (*std_msgs/Header header*), este encabezado tiene una secuencia de tiempo para volver único al mensaje, y la posición (*geometry_msgs/Pose pose*), esta última está conformada por un punto, (x, y, z) , y una orientación, (x, y, z, w) , en cuaterniones. Con esto es posible obtener la posición y orientación de los robots. El siguiente mensaje es el que se utiliza para enviar las señales de control, en forma de velocidad lineal (*Twist geometry_msgs/Vector3 linear*) y

4. PLATAFORMA EXPERIMENTAL

velocidad angular (`geometry_msgs/Vector3` angular).

Finalmente se muestra la plataforma experimental en la Figura 4.7, donde *a)* muestra los tres robots *turtlebot* de tercera generación, dos robots tipo *Burger* y un robot tipo *Waffle*, en *b)* se observa el sistema de visión artificial para la obtención de las posiciones de los robots, en la figura *c)* se muestra la computadora externa con el programa *Motive* donde se procesa la información y se reescribe en un puerto para ser leída por otros dispositivos, en *d)* se observa la computadora donde se ejecutó el programa maestro y el controlador que manda las señales de las velocidades a los robots.



Figura 4.7: Plataforma experimental.

La plataforma de los robots *TurtleBot* se encuentra en constante actualización. Sin embargo se sigue desarrollando código para sensores, actuadores o dispositivos de utilidad en la robótica para implementarse en el ambiente de ROS.

Resultados Experimentales

El primer experimento muestra la comparación de dos controladores para el seguimiento de trayectoria para un solo robot. Se realiza una comparación con los controladores para dos trayectorias diferentes, un ovalo y una trayectoria en forma de flor, con resultados en simulación.

El segundo experimento se realizó con el fin de ajustar la plataforma experimental para tener un comportamiento similar tanto en simulación como en forma experimental, la trayectoria utilizada es un óvalo.

El tercer experimento, una simulación numérica y resultados experimentales, muestra para una estrategia de formación con dos robots resultados sobre una línea recta para el líder y variaciones en la formación, dando posiciones deseadas variantes en el tiempo para el seguidor.

El cuarto experimento muestra una comparación nuevamente para diferentes controladores, se observa cómo se comportan los controladores con y sin el esquema SBO. Los resultados se muestran con una línea recta para el líder y con posiciones variantes en el tiempo para los dos robots seguidores.

El quinto experimento muestra el comportamiento para una trayectoria circular para el líder junto con posiciones variantes en el tiempo para los dos seguidores.

5.1. Experimento 1.- Simulación de controladores

Para este experimento se compararon dos controladores, descritos en (5.1) junto a (3.9), para dos trayectorias diferentes. La primera una especie de óvalo descrito en la Tabla 5.1 y la segunda una flor de tres pétalos descrita en la Tabla 5.2 con resultados en simulación para un robot. Este experimento se realizó para verificar el funcionamiento de ambas retroalimentaciones y elegir un control para los experimentos.

5.1.1. Controles utilizados

El primer control utilizado en este experimento fue el documentado en [10], este controlador fue implementado para el seguimiento de trayectorias.

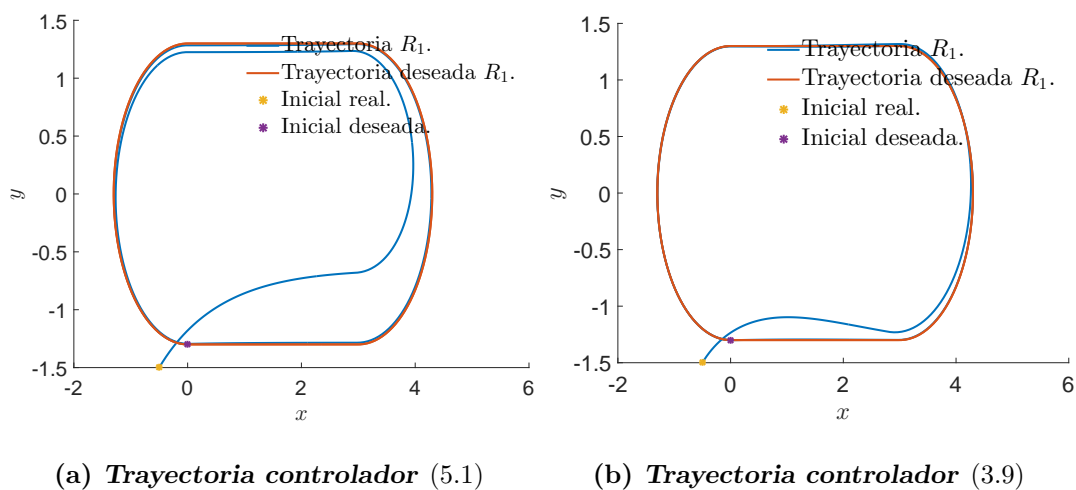
$$\begin{aligned} v_i &= (\dot{x}_i^d - k_1 \tilde{x}_i) \cos(\theta_i) + (\dot{y}_i^d - k_2 \tilde{y}_i) \sin(\theta_i) \\ w_i &= \dot{\theta}_i^d - k_3 \tilde{\theta}_i \end{aligned} \quad (5.1)$$

De forma similar el segundo control implementado en este experimento, vea (3.9), fue el utilizado en [16], este controlador se utiliza para el seguimiento de trayectorias.

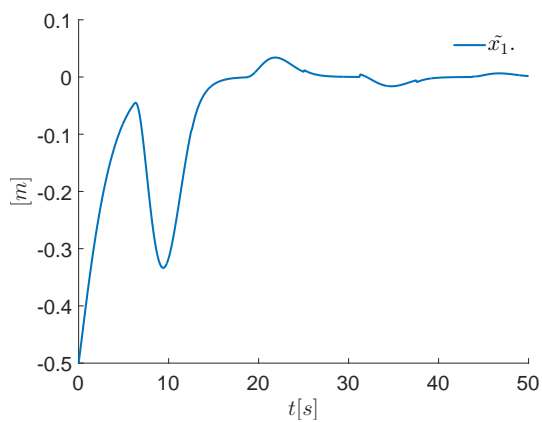
5.1.2. Simulación óvalo

El primer segmento del óvalo es la recta horizontal inferior. El segundo segmento es el formado por un medio círculo alineado a la derecha. El tercer segmento es el formado por la recta superior y el último segmento es un medio círculo alineado a la izquierda, las ecuaciones se encuentran en la Tabla 5.1.

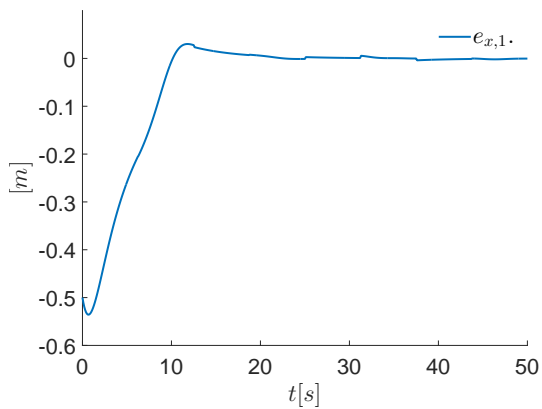
Segmento 1	Segmento 2
$x_i^d = q_1$	$x_i^d = r \cos(q_2 - \frac{\pi}{2}) + l$
$y_i^d = -r$	$y_i^d = r \sin(q_2 - \frac{\pi}{2})$
$\theta_i^d = 0$	$\theta_i^d = q_2$
$q_1 = \alpha_1 t_{s1}$	$q_2 = \alpha_2 t_{s2}$
Segmento 3	Segmento 4
$x_i^d = -q_3 + l$	$x_i^d = r \cos(q_4 + \frac{\pi}{2})$
$y_i^d = r$	$y_i^d = r \sin(q_4 + \frac{\pi}{2})$
$\theta_i^d = \pi$	$\theta_i^d = q_4 + \pi$
$q_3 = \alpha_3 t_{s3}$	$q_4 = \alpha_4 t_{s4}$

Tabla 5.1: Trayectoria deseada óvalo.

Figura 5.1: Trayectoria en forma de óvalo.

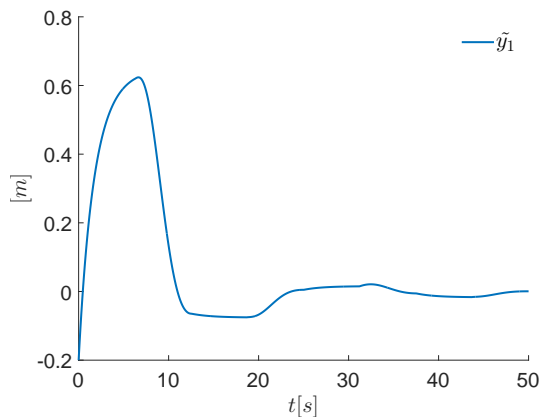
5. RESULTADOS EXPERIMENTALES



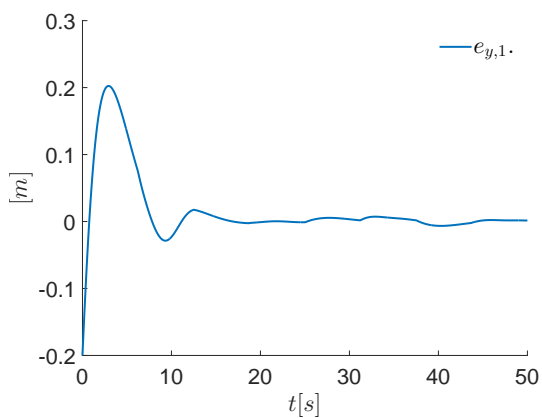
(a) Error \tilde{x}_1 control (5.1).



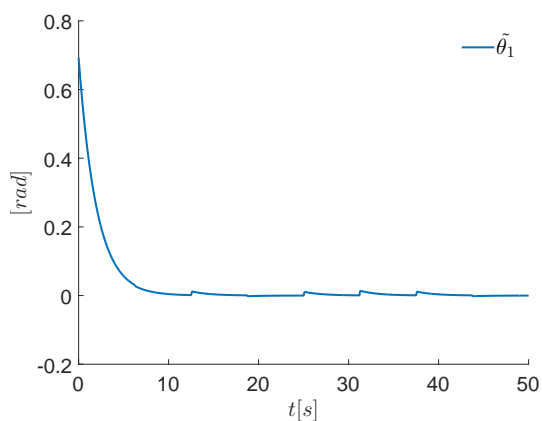
(b) Error $e_{x,1}$ control (3.9).



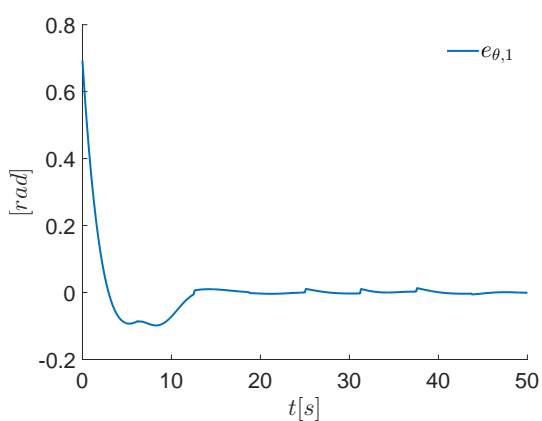
(c) Error \tilde{y}_1 control (5.1).



(d) Error $e_{y,1}$ control (3.9).



(e) Error $\tilde{\theta}_1$ control (5.1).



(f) Error $e_{\theta,1}$ control (3.9).

Figura 5.2: Errores de seguimiento en trayectoria en forma de óvalo para R_1 .

En la Figura 5.2 se observan los errores para los controles especificados, se observa que el control (5.1) tiene problemas en las trayectorias rectas, con $w_i^d = 0$. Esto se refleja en la componente del error para \tilde{y}_1 en la Figura 5.2 (c). Para el controlador (3.9) converge en las partes rectas y curvas.

5.1.3. Simulación flor

En la trayectoria anterior se observó el desempeño de los controladores primero en una línea recta y después en una parte circular, con esta trayectoria se desea comparar ahora los controladores donde funcionaron correctamente, en una parte completamente curva esa fue la motivación de esta trayectoria descrita en la Tabla 5.2.

Segmento
$x_i^d = r \cos(kq) \cos(q)$
$y_i^d = r \cos(kq) \sin(q) + r$
$\theta_i^d = \tan^{-1}\left(\frac{\dot{y}_i^d}{\dot{x}_i^d}\right)$
con, $r = 1, k = 3, w = 1$ y $q = wt$

Tabla 5.2: Trayectoria deseada flor con k pétalos.

En la Figura 5.3 se observan las trayectorias deseadas y las actuales.

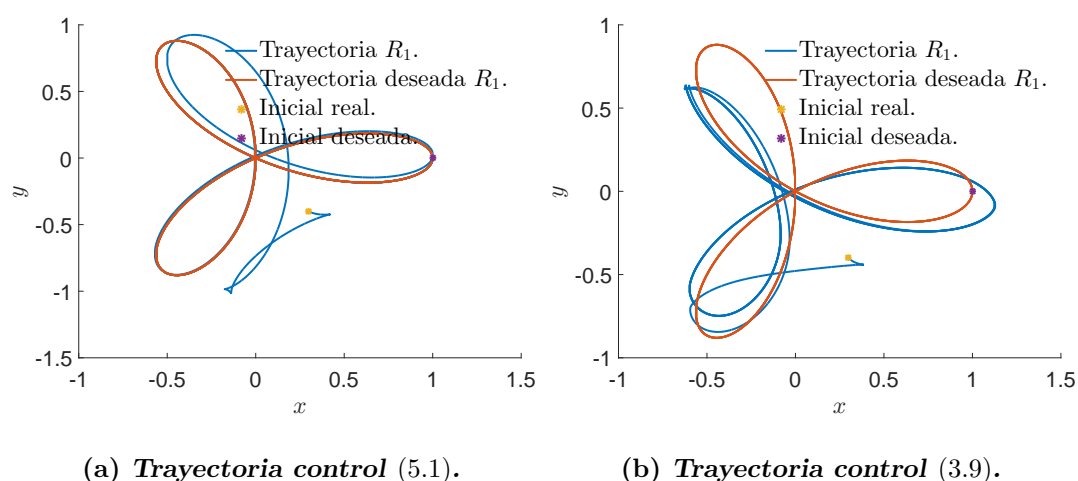


Figura 5.3: Trayectorias en forma de flor.

5. RESULTADOS EXPERIMENTALES

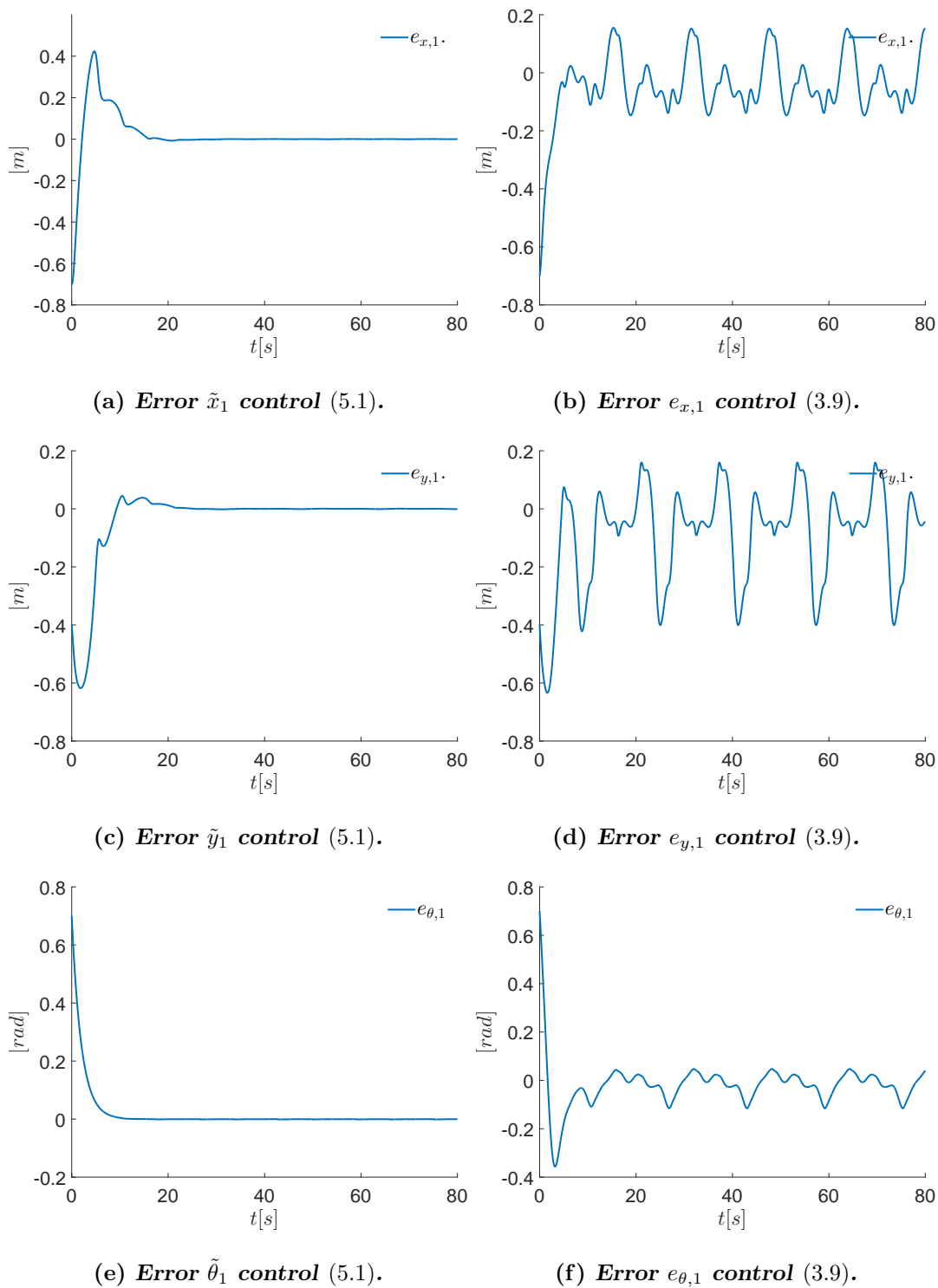


Figura 5.4: Errores de seguimiento con trayectoria en forma de flor.

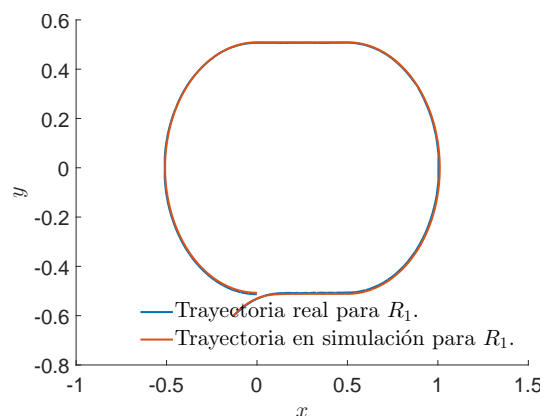
En la Figura 5.4 se observa que el control (5.1) responde correctamente durante todo el trayecto como se esperaba, sin embargo el control (3.9) presenta dificultades para llegar a la posición deseada, cabe destacar que las ganancias se dejaron de la misma forma en ambos experimentos por lo que se puede concluir que para el segundo controlador es necesaria una correcta sintonización, se aclara que la trayectoria es demandante en el sentido que se exige recorrer cinco veces el trayecto deseado. Estos fueron los primeros resultados que se obtuvieron en simulación para un solo robot.

5.2. Experimento 2.- Validación de la plataforma experimental con un robot

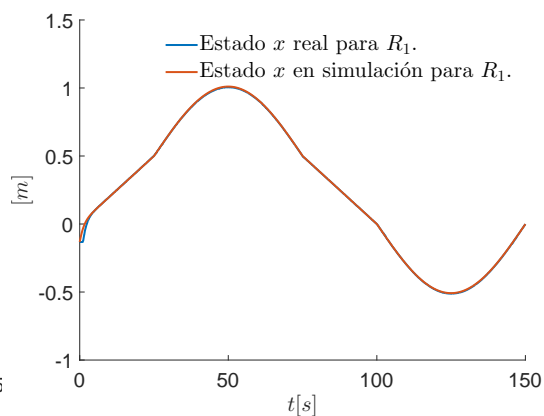
Debido a que la plataforma era reciente en el laboratorio, se analizó el comportamiento de los datos de la simulación y los datos reales obtenidos para el robot *TurtleBot3 Burger*. Para una trayectoria en forma de óvalo son necesarios 4 segmentos, de forma similar se utiliza la trayectoria de la Tabla 5.1. El control utilizado para este experimento es el descrito en (5.1). Se obtuvieron los datos experimentales y se contrastaron con los generados en simulación, obteniéndose los resultados de la Figura 5.5, esto con el fin de validar el modelo en (2.6).

Se observa que al principio se tiene un transitorio en las señales, una posible explicación para esto es debido a que existe un controlador interno que posee el robot para asegurar la velocidad deseada. Se observa que los estados tienen una variación menor en comparación con los controles, estos últimos presentan cierta cantidad de ruido y un transitorio en los cambios ocurridos en la trayectoria.

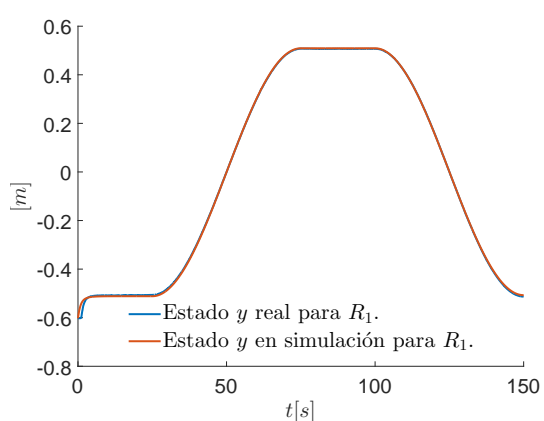
5. RESULTADOS EXPERIMENTALES



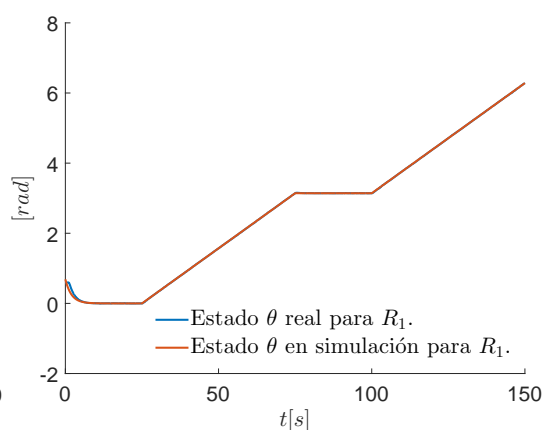
(a) *Superposición de las trayectorias simulada y real.*



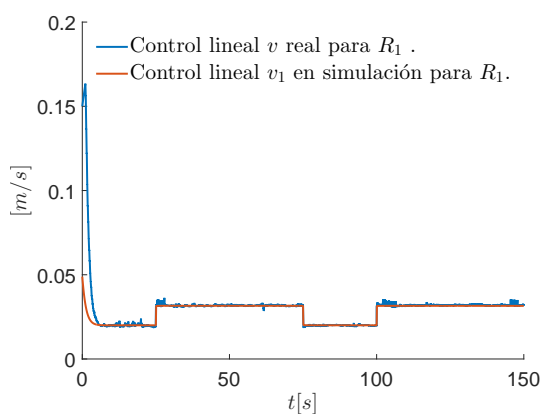
(b) *Comparación del estado x .*



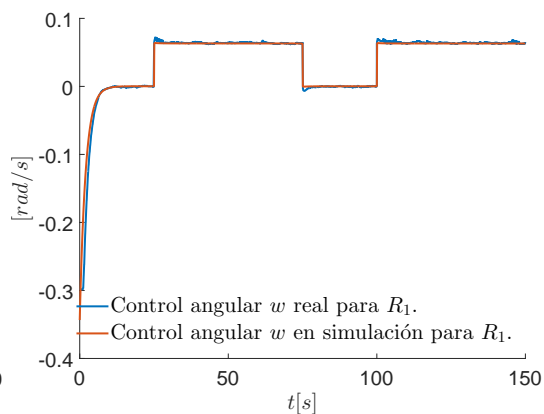
(c) *Comparación del estado y .*



(d) *Comparación del estado θ .*



(e) *Comparación del control v .*



(f) *Comparación del control w .*

Figura 5.5: Validación de la plataforma experimental.

5.3. Experimento 3.- Línea recta dos robots

Se utilizaron dos robots en esta prueba, para ello se implementó el control para la velocidad lineal descrito en (3.57) y el control angular de (3.77) junto a la definición (3.72) para el robot seguidor. Para el líder se designó una trayectoria deseada dada por una línea recta, esto fue así con el propósito de realizar una formación secuencial y para posteriormente comparar, esta trayectoria, con un controlador funcional en líneas rectas, se tiene que,

$$x_L^d = \alpha t, y_L^d = r, \theta_L^d = 0 \quad (5.2)$$

donde $r, \alpha \in \mathbb{R}$. Para corroborar el funcionamiento del controlador, ante posiciones deseadas variantes en el tiempo, se eligió,

$$\begin{aligned} l_d(t) &= l_0 + r_0 \sin(w_0 t) \\ \varphi_d(t) &= a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0. \end{aligned} \quad (5.3)$$

Los valores para la distancia deseada fueron $w_0 = 0.2$ [rad/s], $r_0 = 0.05m$ y $l_0 = 0.5m$. Para el polinomio $\varphi_d(t)$ se eligieron las condiciones iniciales como,

$$\begin{aligned} \varphi_d(0) &= \frac{\pi}{2}, \varphi_d(t_f) = \pi \\ \dot{\varphi}_d(0) &= \dot{\varphi}_d(t_f) = \ddot{\varphi}_d(0) = \ddot{\varphi}_d(t_f) = 0 \end{aligned} \quad (5.4)$$

esta prueba se realizó con un lapso de $t_{total} = 50s$. Es necesario el tiempo total del experimento para calcular las constantes a_{0-5} y realizar un cambio suave del ángulo deseado $\varphi_d(t)$ de $\frac{\pi}{2}$ a π radianes. El efecto deseado es posicionar de forma secuencial o alineada al seguidor atrás del líder para la trayectoria asignada. En la Tabla 5.3 se observan las condiciones iniciales y los valores utilizados en la evaluación experimental.

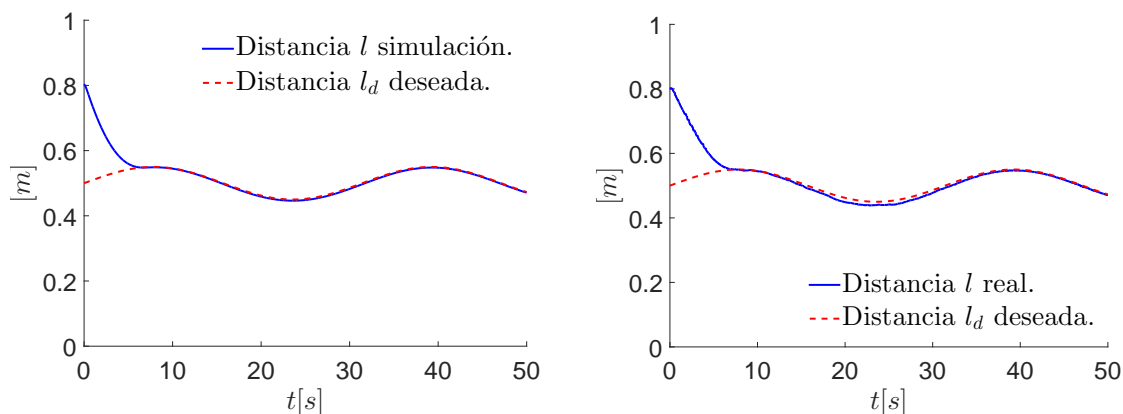
$x_L(0)$	$y_L(0)$	$\theta_L(0)$	$x_F(0)$	$y_F(0)$	$\theta_F(0)$
-0.52	0.001	-0.02	-1.09	0.56	-0.09

k_1	k_2	k_3	γ_1	γ_2
0.001	10.0	5.0	0.05	0.01

Tabla 5.3: Condiciones iniciales y ganancias del controlador.

5. RESULTADOS EXPERIMENTALES

En la Figura 5.6(a), se muestra la evolución de la distancia $l(t)$, la cual se implementó con una función senoidal descrita en (5.3), y su correspondiente valor deseado $l_d(t)$ en simulación. En la Figura 5.6(b) se presenta el resultado de la plataforma experimental, descrito para la distancia entre el líder y el seguidor.



(a) *Evolución de las distancias en la simulación.*

(b) *Evolución de las distancias en el experimento.*

Figura 5.6: Distancias, l y l_d , en simulación y en experimentación.

En la Figura 5.7(a), se muestra la evolución del ángulo $\varphi(t)$ y su valor deseado $\varphi_d(t)$ en simulación. En la Figura 5.7(b) se presenta el resultado de la plataforma experimental para los ángulos φ y φ_d .

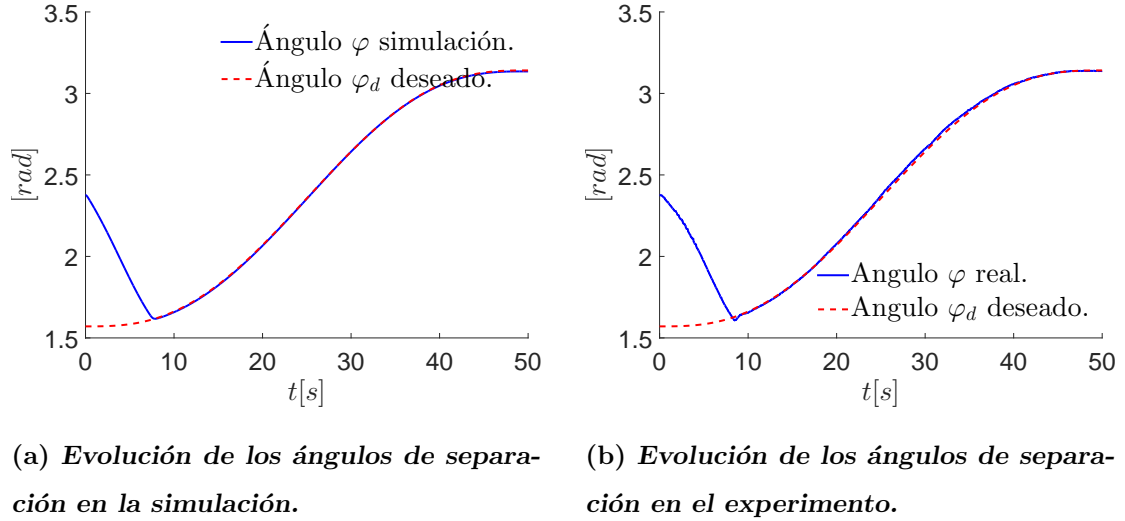


Figura 5.7: Ángulo φ en simulación y en experimentación.

La Figura 5.8(a) muestra la evolución del control v_F en simulación. La Figura 5.8(b) presenta el resultado de la plataforma experimental para el control angular, se observa que se presenta nuevamente ruido en la señal de control.

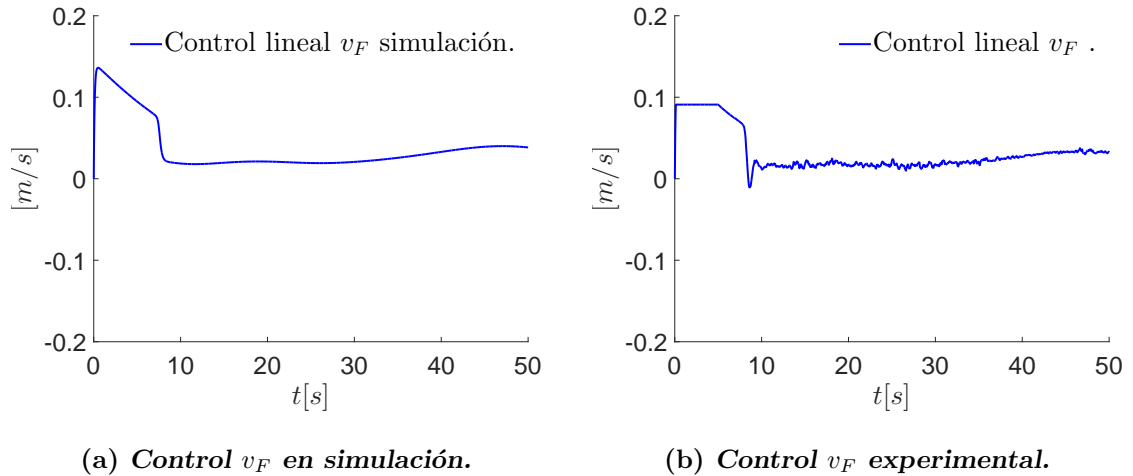


Figura 5.8: Control v_F en simulación y en experimentación.

La Figura 5.9(a) muestra la evolución del control w_F en simulación. La Figura 5.9(b) presenta el resultado de la plataforma experimental para el control angular, se observa que se presenta ruido mayor que la señal de control descrita en 5.8(b).

5. RESULTADOS EXPERIMENTALES

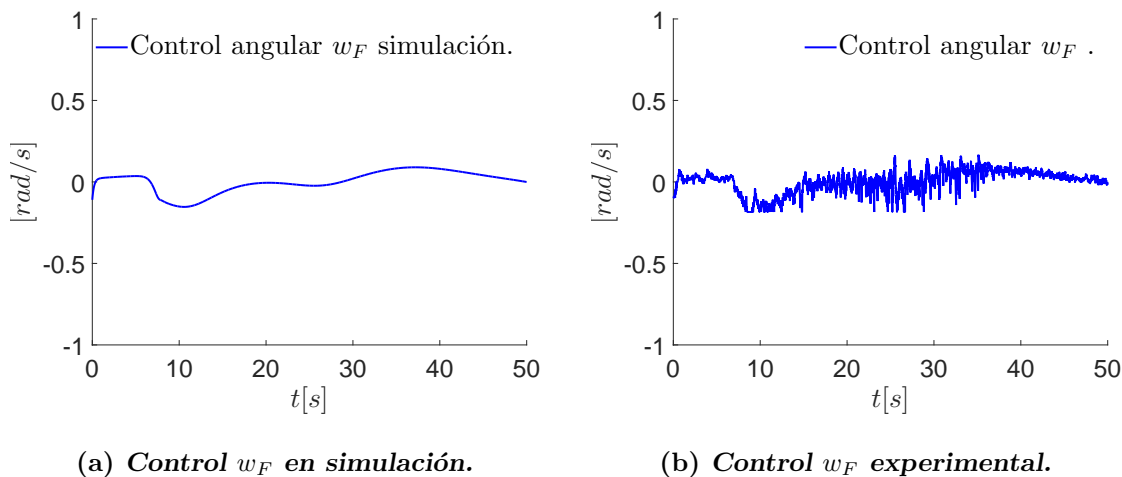


Figura 5.9: Control w_F en simulación y en experimentación.

Se observa en la Figura 5.10(a) la trayectoria de los dos robots y la trayectoria deseada para el robot seguidor R2 en simulación. La Figura 5.10(b) muestra la trayectoria de la formación líder-seguidor en el plano $x - y$, para los resultados en la plataforma experimental.

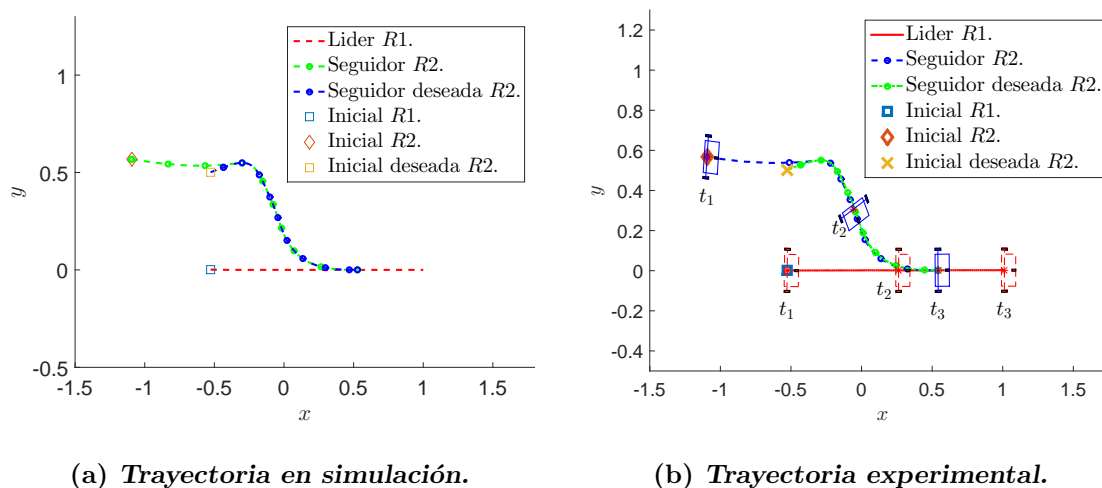


Figura 5.10: Trayectoria en simulación y en experimentación.

En la Figura 5.11 se muestran imágenes de la evaluación experimental de los robots.

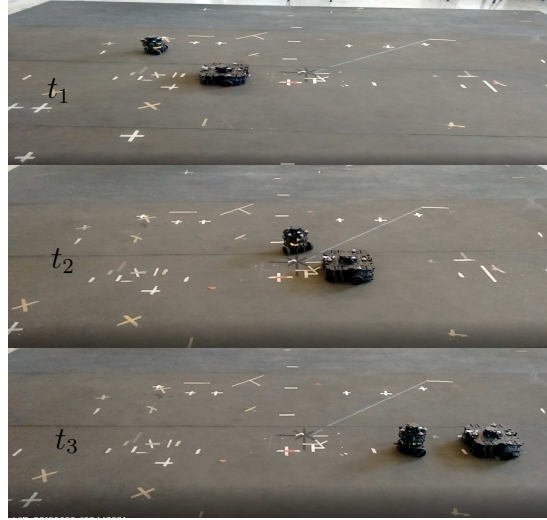


Figura 5.11: Robots en diferentes tiempos del experimento.

Se observa en las Figuras 5.10(b) y 5.11, correspondientes a los resultados experimentales, con tiempos $t_1 = 0s$, $t_2 = 25s$ y $t_3 = 50s$. Estos son instantes diferentes del experimento, los cuales tienen el fin de mostrar cómo es la posición y orientación de los dos robots.

En este experimento se observa la convergencia de los estados actuales a los estados deseados, vistos en las Figuras 5.6 y 5.7, esto en un tiempo menor a 10s.

5.4. Experimento 4.- Comparación y esquema SBO

Se utilizaron tres robots en esta prueba, para ello se implementó el control para la velocidad lineal descrito en (3.57) y el control angular de (3.77) para los robots R_2 y R_3 , se presenta la comparación con el controlador descrito en (3.9), para una estrategia de formación. Primeramente se muestran las trayectorias en el plano, los robots se dirigen de una posición izquierda a una derecha como se indica en la Figura 5.12, con tiempos $t_1 = 0s$, $t_2 = 25s$ y $t_3 = 50s$. Para este experimento en algunas pruebas se usó el esquema SBO y en otras no fue así. Cuando se utiliza el esquema SBO se define junto con la posición deseada en (3.72). Cuando no se utiliza el esquema SBO se define la posición angular deseada como (3.59).

5. RESULTADOS EXPERIMENTALES

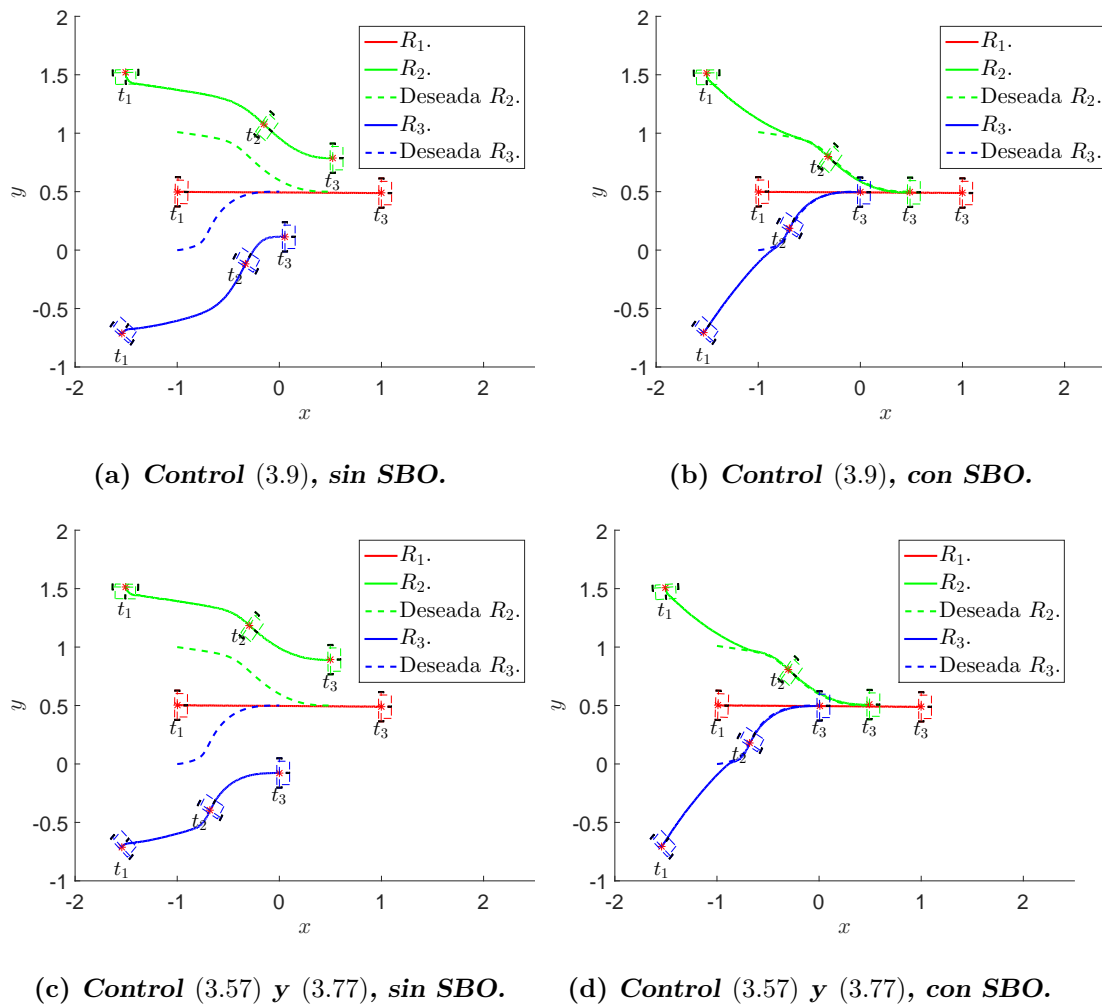


Figura 5.12: Trayectorias.

En la Figura 5.13 se observan las orientaciones de los robots, esto con el fin de complementar la Figura 5.12.

5.4 Experimento 4.- Comparación y esquema SBO

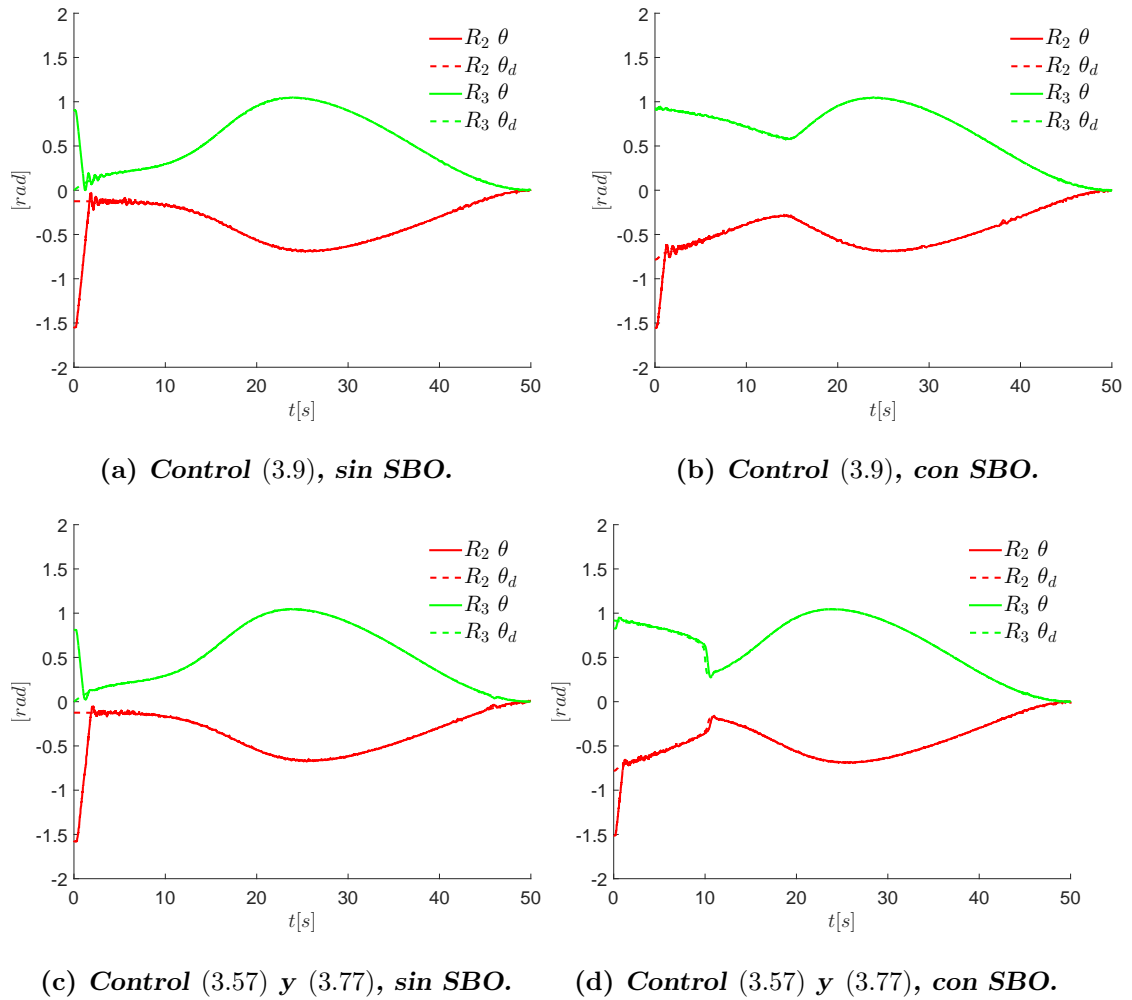


Figura 5.13: Orientaciones.

En la Figura 5.14 se observan las variables involucradas en el esquema SBO para el robot 2.

5. RESULTADOS EXPERIMENTALES

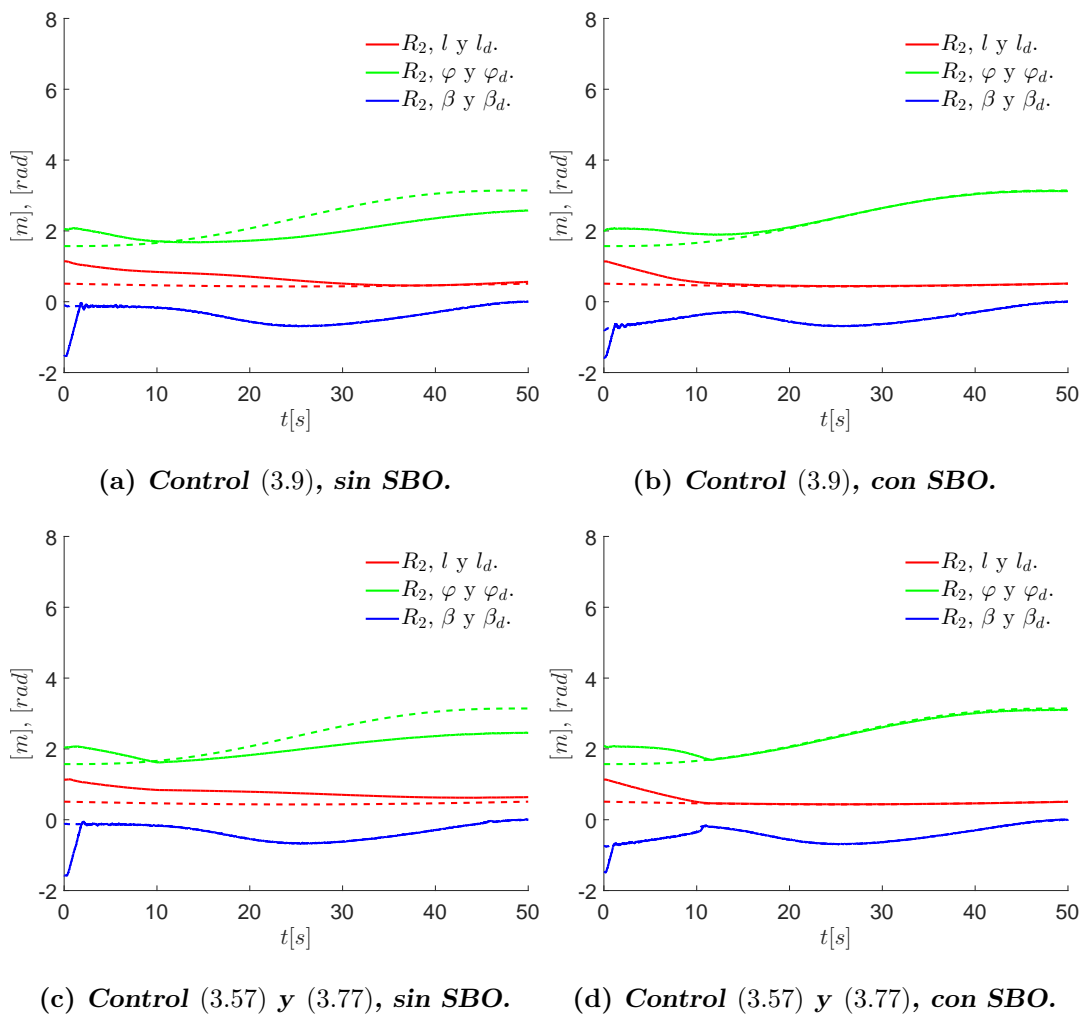


Figura 5.14: SBO Robot 2.

En la Figura 5.15 se observan las variables involucradas en el esquema SBO para el robot 3.

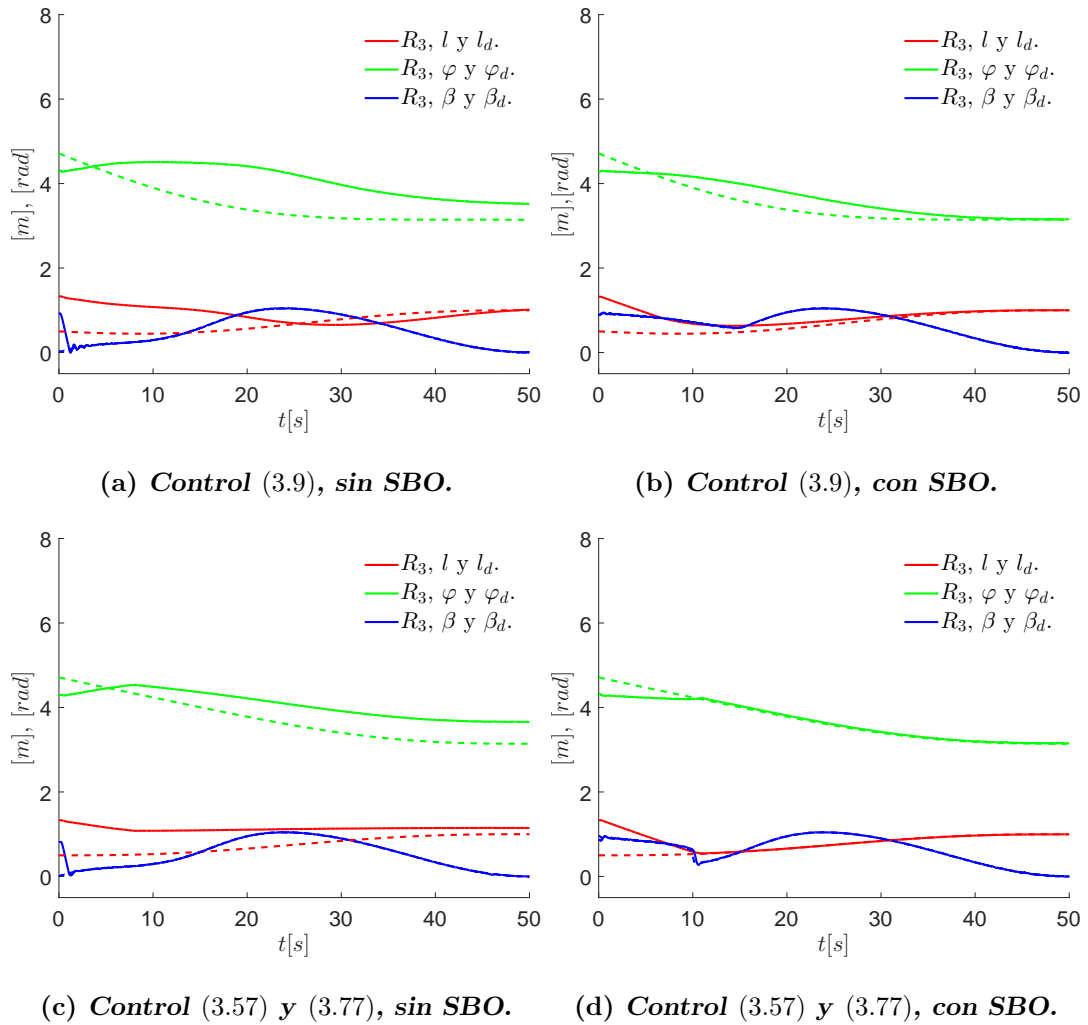


Figura 5.15: SBO Robot 3.

En la Figura 5.16 se presentan las señales de control reales para el segundo robot.

5. RESULTADOS EXPERIMENTALES

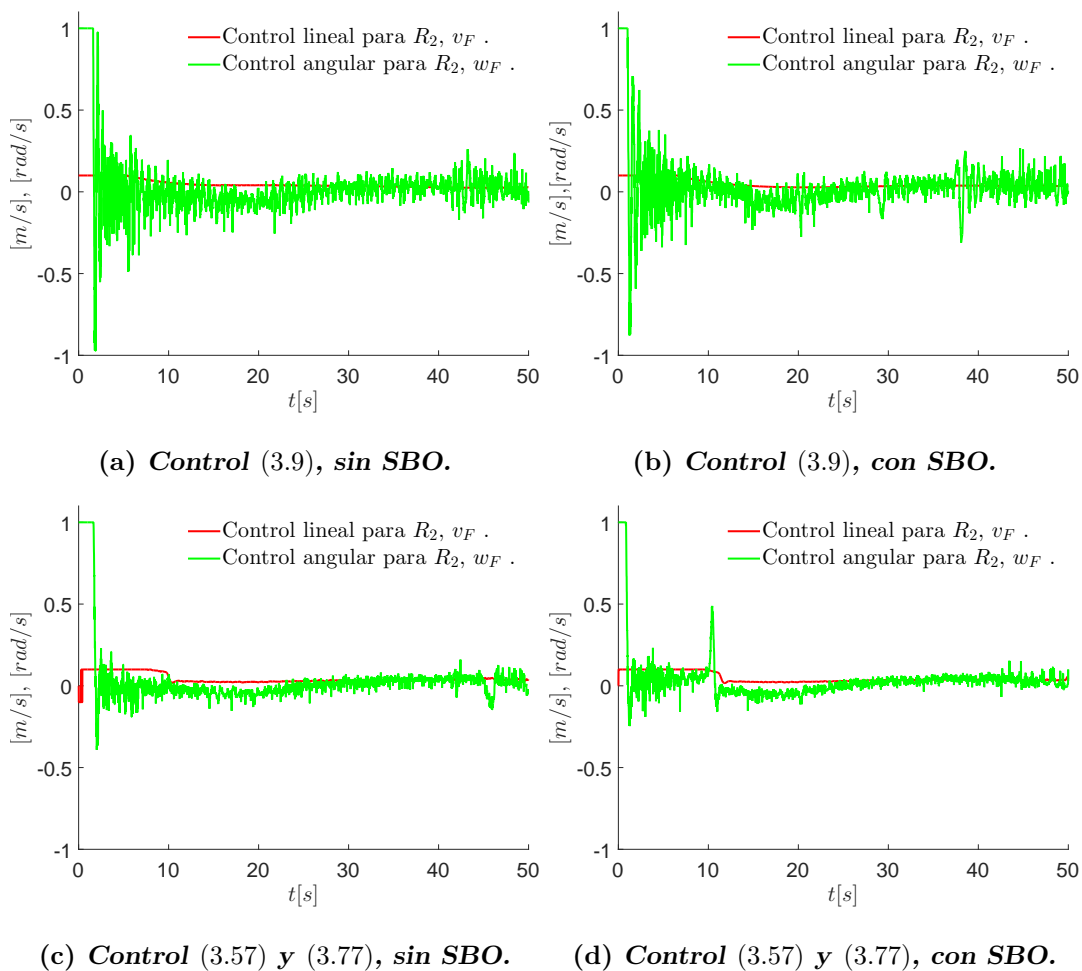


Figura 5.16: Controles Robot 2.

En la Figura 5.17 se presentan las señales de control reales para el tercer robot.

5.5 Experimento 5.- Tres robots con trayectoria circular

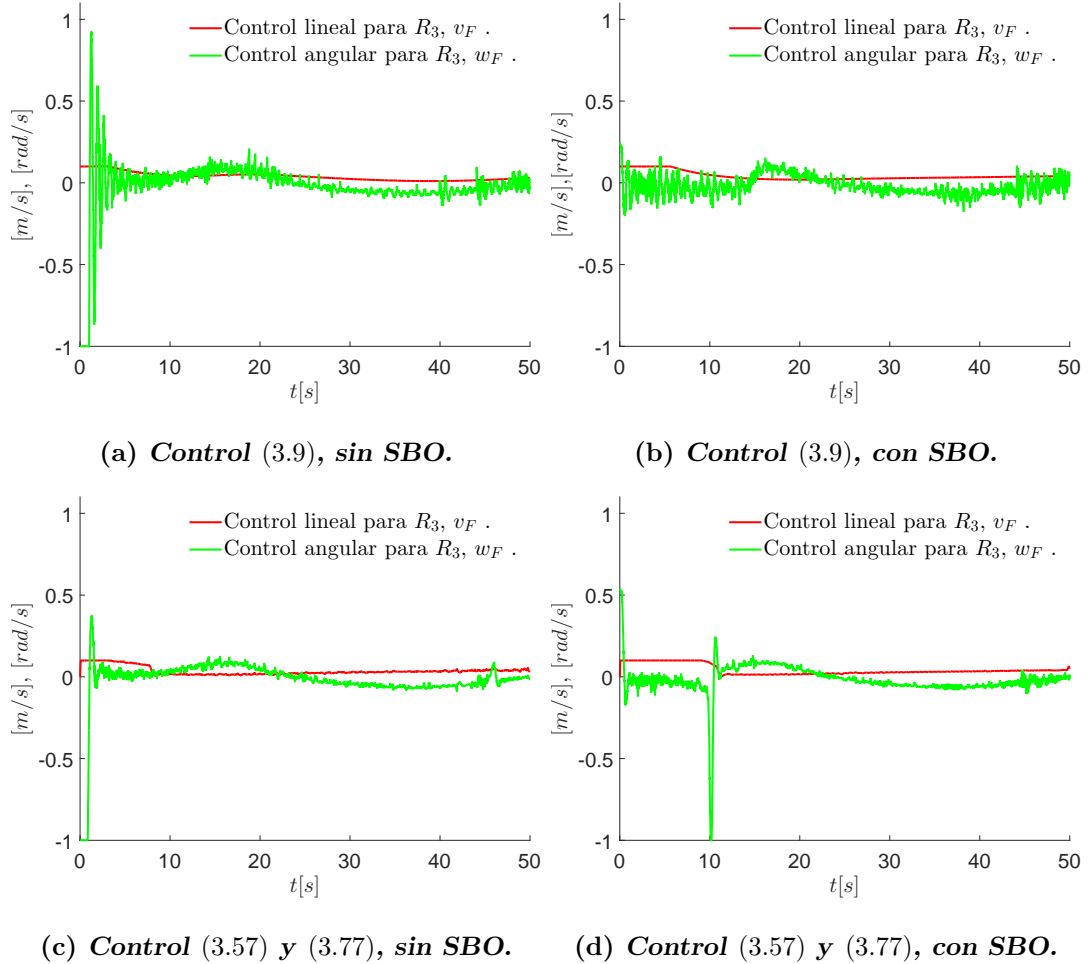


Figura 5.17: Controles Robot 3.

Como se puede observar al utilizar el esquema SBO junto con la definición en (3.72) se logran resultados satisfactorios en tiempo y comportamiento para ambas retroalimentaciones.

5.5. Experimento 5.- Tres robots con trayectoria circular

Los resultados presentados son para los tres robots, utilizando el control por modos deslizantes, *Backstepping* y equivalencia a sistemas pasivos descrito en

5. RESULTADOS EXPERIMENTALES

(3.57) y (3.77), junto al esquema SBO y la definición de (3.72). La Figura 5.18 muestra las trayectorias del experimento con la trayectoria circular para el líder, se implementa el control con tres robots variando los parámetros de distancia y ángulo, además se muestran los tiempos $t_1 = 0s$, $t_2 = 25s$, $t_3 = 50s$ y $t_4 = 100s$. El objetivo de este experimento fue el de crear una formación triangular al final, al mismo tiempo que los robots emulan el proceso de adelantar un vehículo.

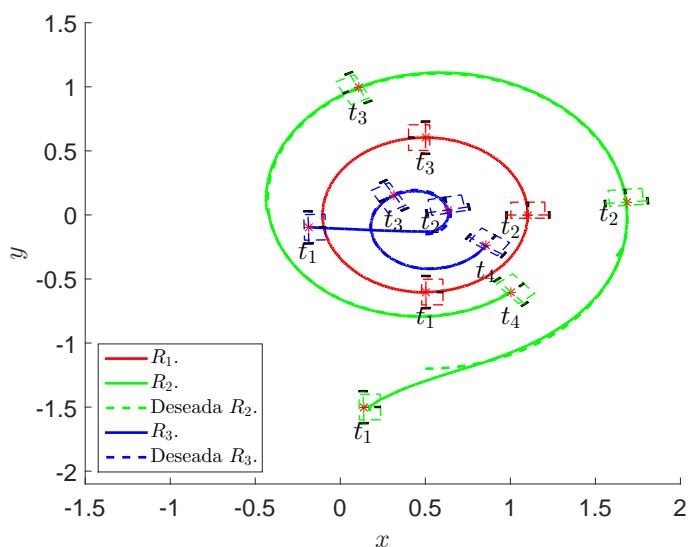


Figura 5.18: Trayectorias de los tres robots en el plano.

La Figura 5.19 muestra las orientaciones de los robots seguidores.

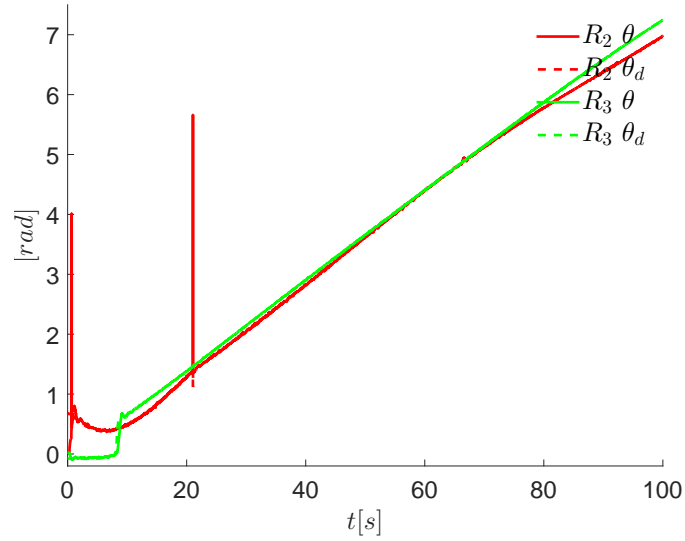


Figura 5.19: Orientaciones de los robots seguidores.

La Figura 5.20 muestra los errores relacionados con el robot seguidor R2.

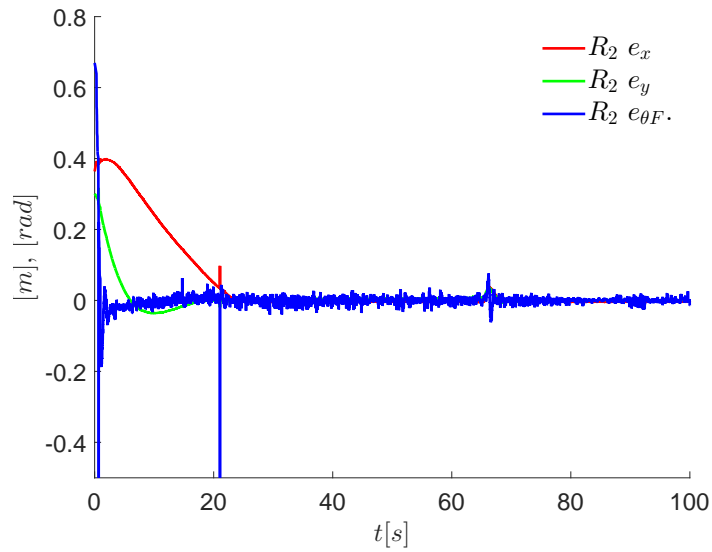


Figura 5.20: Errores para el segundo robot.

La Figura 5.21 muestra las variables relacionadas con el ángulo deseado, distancia deseada y desviación del ángulo para el esquema SBO para el segundo robot.

5. RESULTADOS EXPERIMENTALES

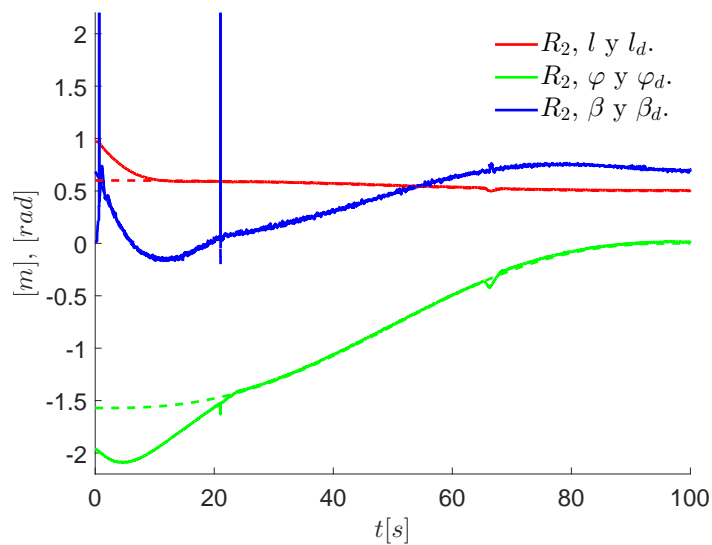


Figura 5.21: Variables SBO para R_2 .

La Figura 5.22 muestra el control para el tercer robot.

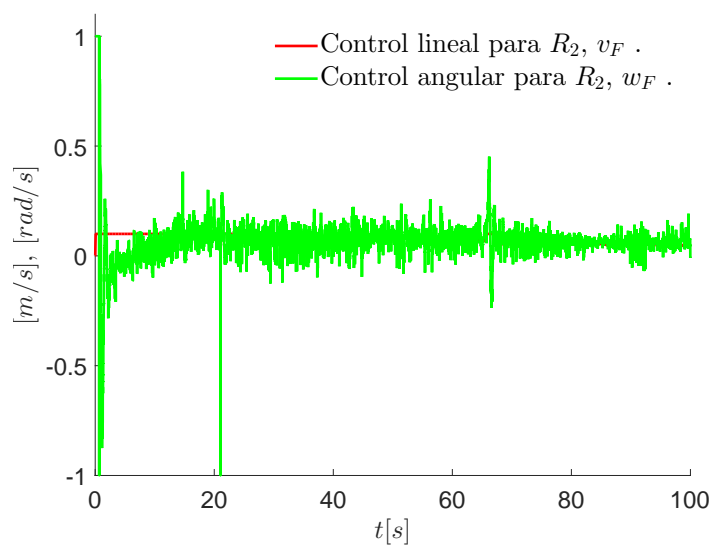


Figura 5.22: Controles R_2 .

La Figura 5.23 muestra los errores relacionados con el robot seguidor R_3 .

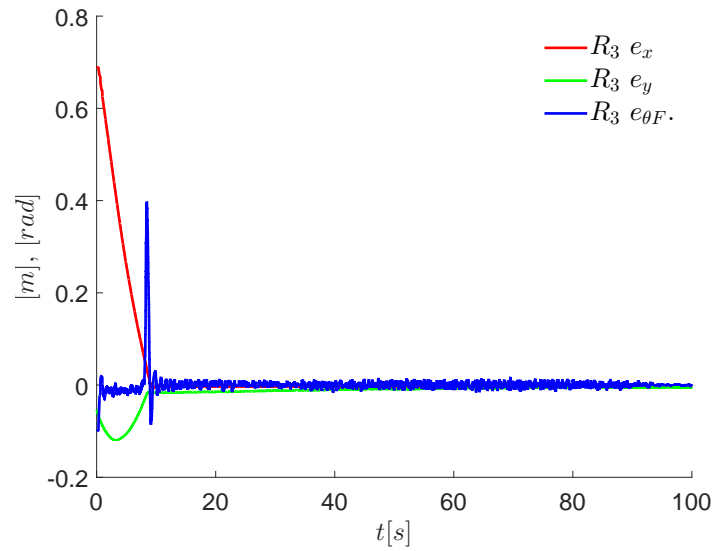


Figura 5.23: Errores para el tercer robot.

La Figura 5.24 muestra las variables relacionadas con el esquema SBO para el tercer robot.

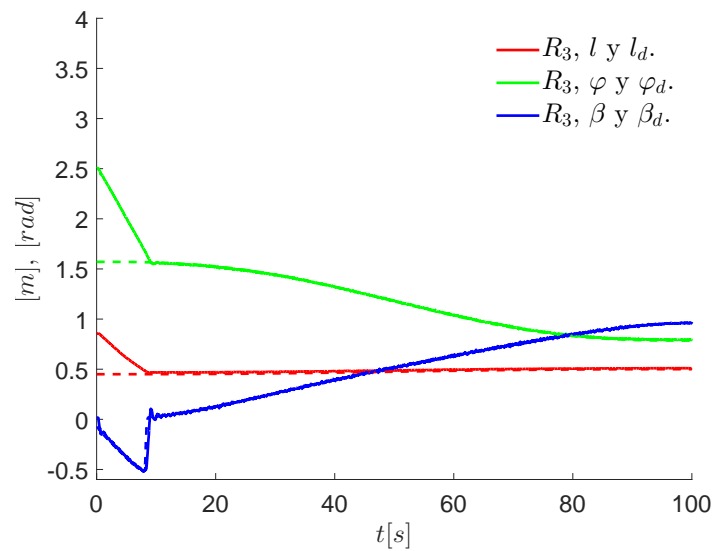


Figura 5.24: Variables SBO para R_3 .

La Figura 5.25 muestra las señales de control para el tercer robot.

5. RESULTADOS EXPERIMENTALES

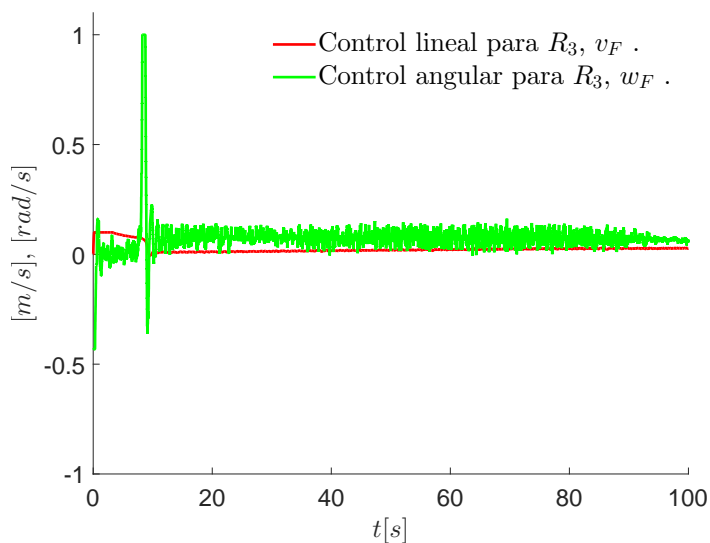


Figura 5.25: Controles R3.

Se puede observar que la retroalimentación funciona correctamente para el control de la formación, con los tres robots sobre una trayectoria circular para el líder. En este experimento también se observa un comportamiento diferente al experimento anterior, debido a que el segundo robot es el más externo en la formación los controles tienen diferentes rangos de velocidades. Esto ocasionó que se tuviera una mayor velocidad, en un principio se recorría el trayecto en un tiempo aproximado de 50 segundos, sin embargo la demanda del controlador para el robot era demasiado elevada. Fue por eso que se decidió implementar el experimento a una velocidad menor para tener un accionamiento de los controles dentro de los parámetros recomendados en la velocidad lineal, $0.1m/s$, y con ello evitar daños a la plataforma experimental.

Conclusiones

La importancia del esquema desarrollado radica en su implementación práctica y resultados obtenidos. La ventaja que presenta esta estrategia es la convergencia de los errores teniendo en cuenta que son sistemas subactuados y con restricciones de movimiento. Este trabajo presenta el diseño y evaluación experimental de un controlador para una formación líder-seguidor de dos y tres robots móviles diferenciales. El diseño del controlador está basado en la metodología de *Backstepping*, de modos deslizantes y de pasividad para la velocidad lineal del robot seguidor con el fin de realizar un controlador robusto a incertidumbres acotadas.

Se implementó la retroalimentación propuesta en una plataforma compuesta por tres robots móviles y un sistema de localización basado en visión artificial. Se presentan resultados experimentales con diferentes tipos de controladores y se muestran comparaciones entre controladores con el fin de evaluar su desempeño bajo ciertas condiciones. Se desarrolló un controlador en Python utilizando la herramienta modular ROS. Se realizó la modificación del controlador propuesto para trayectorias variantes en el tiempo. Implementación del esquema SBO y se realizó una modificación de la trayectoria deseada, para tener una convergencia de los errores más rápida que cuando no se utiliza esta definición. Uno de los logros alcanzados fue la publicación del artículo, presentado en el apéndice B, en el Congreso Mexicano de Robótica ComRob 2018. En la parte experimental se lograron errores reducidos en el orden de un centímetro respecto a las variaciones de la formación. El sistema de visión implementado es de naturaleza variable, es decir depende en gran medida de la luminosidad para un correcto funcionamiento, esto ocasionó que se tuviera ruido y perturbaciones en algunos experimentos. Esto sucedió a pesar de haber realizado los ajustes necesarios en el programa o de calibrar el equipo. Se puede verificar que existen tiempos en los que el sistema deja de enviar datos por que no reconoce al robot como un cuerpo rígido, generando espurios en las gráficas. Los inconvenientes del controlador fueron para la

implementación, debido a que se necesita realizar mayores cálculos computacionales, respecto a los utilizados por otros controladores, sin embargo se tiene en cuenta que es un sistema no lineal y el controlador arrojó buenos resultados.

6.1. Trabajo futuro

Como trabajo a futuro podría implementarse evasión de obstáculos por medio de algún método descrito en la literatura para formaciones variantes en el tiempo, esto podría realizarse ya que la trayectoria deseada se puede modificar en tiempo real, debido al esquema utilizado. Otra de las posibilidades es la implementación de este control de forma descentralizada donde el procesamiento de las señales de control se realice a bordo de los robots para una formación de tres robots, realizando los experimentos de esta tesis y comparando la afectividad de ambos. Construir la plataforma experimental para poder realizar ajustes a voluntad con el fin de mejorar los resultados o de tener resultados con alto grado de repetitividad. Un posible trabajo a futuro podría ser la implementación de este controlador, junto al esquema SBO propuesto en esta tesis, para una plataforma experimental diferente por ejemplo en formación de vehículos aéreos no tripulados junto a robots móviles creando un conjunto no homogéneo de robots. Se podría indagar en utilizar un controlador angular diferente para lograr resultados más rápidos en los experimentos o realizar una modificación general para el esquema líder-seguidor. Implementar la estrategia en algún enfoque industrial, de servicio, de seguridad o comercial.

Conceptos de sistemas no lineales

De [19] y [21] se describen conceptos importantes, los cuales se mencionan por la relevancia para el desarrollo del trabajo de tesis.

A.0.1. Linealización por retroalimentación

El método de linealización exacta por retroalimentación [21] es una herramienta de control la cual utiliza un cambio de coordenadas para un sistema no lineal afín a la entrada y una realimentación que transforma al sistema no lineal a una forma lineal equivalente. Este método incluye la linealización entrada-estado y la linealización entrada-salida. Este es un enfoque geométrico en el cual la linealización puede ser total o parcial, por medio del estado difeomorfo y la realimentación linealizante. Sin embargo esta linealización puede resultar en esfuerzo de control desperdiciado y llevar a sistemas no robustos. En ocasiones se elimina la parte no lineal estable del sistema y se agregan términos desestabilizantes. Un pequeño error en el modelado puede llevar a pérdida de estabilidad en el sistema.

A.0.2. Grado relativo sistemas SISO

El grado relativo de un sistema SISO (Single Input Single Output) en la forma,

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}\tag{A.1}$$

con $u \in \mathbb{R}$ entrada, $y \in \mathbb{R}$ salida del sistema, donde f , g y h , definidas en un dominio $D \subset \mathbb{R}^n$, son suficientemente suaves, tiene grado relativo $r \in W$ en el

punto x_0 si,

$$\begin{aligned} L_g L_f^k h(x) &= 0 \quad \forall x \text{ en una vecindad de } x_0 \text{ y toda } k \leq r - 2 \\ L_g L_f^{r-1} h(x_0) &\neq 0. \end{aligned} \tag{A.2}$$

En forma simple el grado relativo se puede interpretar como el número de veces que se deriva direccionalmente la salida para obtener el control del sistema.

A.0.2.1. Sistema de fase mínima

Cuando el sistema (A.1) se convierte a la forma normal, la dinámica de los ceros o la dinámica inobservable puede ser estable asintóticamente en el punto de interés, en este caso el sistema es llamado de fase mínima. Si la dinámica cero es estable en el sentido de Lyapunov el sistema se dice débilmente de fase mínima.

A.0.3. Conceptos de pasividad

Para los conceptos de pasividad se supone que existe un sistema H con la propiedad de que la energía del sistema solo puede ser incrementada por la administración de energía externa.

A.0.3.1. Disipatividad

Asuma que $w : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ está asociada con el sistema H y es llamada razón de suministro, la cual es localmente integrable para cada $u \in U$, que satisface $\int_{t_0}^{t_1} |w(u(t), y(t))| dt < \infty$ para todo $t_0 \leq t_1$. Sea X un conjunto conectado de \mathbb{R}^n conteniendo el origen. Se dice que el sistema es disipativo en X con una razón de suministro $w(u, y)$ si existe una función $S(x)$, $S(0) = 0$, tal que para toda $x \in X$ se cumple,

$$S(x) \geq 0 \text{ y } S(x(T)) - S(x(0)) \leq \int_0^T w(u(t), y(t)) dy \tag{A.3}$$

para toda $u \in U$ y toda $T \geq 0$ tal que $x(t) \in X$ para todo $t \in [0, T]$. La función $S(x)$ es la llamada función de almacenamiento.

A.0.4. Pasividad

Un sistema es pasivo si tiene una función de almacenamiento semidefinida positiva $S(x)$ y una razón de suministro bilineal $w(u, y) = u^T y$, donde x es el

estado del sistema, u es la entrada y y es la salida. Satisfaciendo la desigualdad,

$$S(x(T)) - S(x(0)) \leq \int_0^T w(u(t), y(t)) dt \quad (\text{A.4})$$

para toda u y $T \geq 0$. Pasividad es la propiedad de que el incremento en el almacenamiento S no es más grande que la integral de la cantidad suministrada. En forma derivativa se tiene,

$$\dot{S}(x) \leq w(u, y) \quad (\text{A.5})$$

entonces la propiedad de pasividad es que la razón de incremento del almacenamiento no es mayor que la razón de suministro. En otros términos, cualquier incremento en el almacenamiento en un sistema pasivo es debido a fuentes externas. La relación entre pasividad y la estabilidad en el sentido de Lyapunov puede ser establecida por medio de la función de almacenamiento $S(x)$ como función candidata de Lyapunov.

A.0.4.1. Estabilidad de sistemas pasivos

La definición de pasividad no requiere que la función de almacenamiento $S(x)$ sea definida o semidefinida positiva. Para esto se tiene en cuenta el término *estabilidad asintótica condicional*, que es una propiedad de la solución del sistema para condiciones iniciales junto a condiciones perturbadas condicionadas a un conjunto Z . Específicamente se obtiene cuando existe estabilidad condicional y la solución es atractiva condicional al origen. A pesar de ser más débil que la propiedad de estabilidad, este concepto nos ayuda a probar estabilidad con funciones candidatas de Lyapunov semidefinidas positivas.

Definición A.0.1 (Estado cero detectable y estado cero observable).

Considere el sistema H con entrada cero, esto es $\dot{x} = f(x, 0)$, $y = h(x, 0)$, siendo $Z \subset \mathbb{R}^n$ el conjunto más grande invariante positivo contenido en $x \in \mathbb{R}^n | y = h(x, 0) = 0$. Se dice que H es estado cero detectable (ECD) si $x = 0$ es asintóticamente condicionalmente estable hacia Z . Si $Z = 0$, se dice que H es estado cero observable (ECO).

Cuando se usa la propiedad de ECD para establecer un resultado global, se asume que $x = 0$ es GAS condicional a Z . Uno de los beneficios de esta propiedad es que pasividad y estabilidad están conectadas incluso cuando la función de almacenamiento $S(x)$ es solo semidefinida positiva. El beneficio destacado es que con la retroalimentación simple $U = -y$ se logra estabilidad asintótica.

Teorema A.0.1 (Pasividad y estabilidad). *Sea el sistema H pasivo con una función de almacenamiento S una vez diferenciable C^1 y con $h(x, u)$ siendo C^1 en u para toda x . Entonces se tiene las siguientes propiedades.*

- *Si S es definida positiva, entonces el equilibrio, $x = 0$, de H con, $u = 0$, es estable.*
- *Si H es ECD, entonces el equilibrio $x = 0$ de H con $u = 0$ es estable.*
- *Cuando no hay salida, $y = h(x)$, entonces la retroalimentación $u = -y$ logra estabilidad asintótica de $x = 0$ si y solo si H es ECD.*

Cuando la función de almacenamiento S es radialmente no acotada estas propiedades permanecen globales.

Apéndice B

Artículo

Control de Formación Líder-Seguidor Robusto de un Conjunto de Robots Móviles Diferenciales

J. D. Gallegos-Sanlucar, R. Castro-Linares , M. Velasco-Villa

Resumen—Este artículo presenta el desarrollo teórico y experimental de una estrategia de formación basada en la metodología líder-seguidor de un conjunto de dos robots móviles. En la formación se utilizaron los modelos cinemáticos de los robots para el desarrollo teórico del controlador el cual está basado en técnicas de backstepping, modos deslizantes y pasividad. Se definió una trayectoria para el líder, con lo cual indirectamente se obtiene la trayectoria deseada del seguidor. Los resultados en simulación fueron utilizados para el ajuste de las ganancias del controlador y así evitar la saturación de los controles. Finalmente se validó en una plataforma experimental el controlador propuesto.

I. INTRODUCCIÓN.

La robótica en la actualidad tiene relevancia a nivel industrial y de investigación científica ya que es utilizada para el transporte de materia con el movimiento del robot sujetando la carga asignada. En éste ámbito la robótica de vehículos terrestres se utiliza en el transporte de objetos, exploración en ambientes desfavorables para los humanos, reconocimiento de entornos y navegación, entre otras actividades. [1] [2] [3] [4]. Para la formación de robots móviles se encuentran en la literatura diferentes estructuras que resuelven el problema. Un enfoque es el basado en comportamiento, en el cual la acción de control es definida por pesos promediados del control de cada comportamiento deseado para el agente y donde se pueden definir comportamientos como evitar colisión, evadir obstáculos, búsqueda de la meta y mantener la formación, entre otros. Este enfoque es natural para definir varios controles con varios comportamientos; la desventaja de este método es el análisis matemático formal y el análisis de estabilidad [5]. En el enfoque de estructuras virtuales las posiciones deseadas son consideradas como una estructura rígida y para cada robot, la posición deseada está referenciada a un punto de la estructura. Una de las ventajas de este enfoque es la posibilidad de asignar una trayectoria y orientación de la estructura para el desplazamiento del conjunto y, si un robot es perturbado, se mantiene la formación de la estructura. Una de las desventajas es su comportamiento con estructuras que varían con el tiempo o que son reconfiguradas frecuentemente, en la cual no se puede obtener el comportamiento esperado [6]. En la estructura líder-seguidor algunos robots se diseñan como seguidores y otros como líderes en la

J. D. Gallegos-Sanlucar, R. Castro-Linares , M. Velasco-Villa están adscritos a la Sección de Mecatrónica del Departamento de Ingeniería Eléctrica del CINVESTAV-IPN, Av. IPN 2508, Col. San Pedro Zacatenco, CP 07360, Ciudad de México, México.

{jonathan.gallegos, rcastro, velasco}@cinvestav.mx

*Se agradece al Conacyt por la beca de maestría otorgada a J. D Gallegos-Sanlucar.

*Trabajo parcialmente apoyado por el proyecto Conacyt 254329.

formación. El líder tiene una trayectoria propia y el seguidor tiene una trayectoria que depende de los estados del líder para generar su propia posición deseada. Una de las ventajas de la estructura es que la formación puede ser mantenida si el seguidor sufre alguna perturbación; sin embargo, si el líder es perturbado también se reflejará en la formación del seguidor [7] [8] [9]. Este trabajo retoma las formaciones de robots móviles por medio de la formación líder-seguidor, la cual utiliza para su implementación una distancia deseada, un ángulo deseado y las posiciones del líder. Se desarrolló un controlador para la velocidad lineal y la velocidad angular del seguidor. El controlador para la velocidad lineal se diseñó en base a la técnica de backstepping junto con técnicas de modos deslizantes para rechazar perturbaciones, [10]. Una de las diferencias con el trabajo realizado en [10] se encuentra en las ecuaciones de los errores, al agregar una separación deseada variante en el tiempo, en éste trabajo se considera que el error angular no es cero para el desarrollo de estas ecuaciones. Referente al controlador implementado no se presenta una discontinuidad con la velocidad angular deseada en cero por lo tanto no es necesario realizar un análisis para las dos condiciones. El controlador de este artículo presenta una discontinuidad cuando la orientación del líder y el seguidor son perpendiculares, además se utiliza una aproximación diferente de la función signo, entre otras diferencias. Finalmente se implementó el control propuesto por medio de una plataforma experimental.

II. ANÁLISIS DEL ESQUEMA LÍDER-SEGUIDOR.

El control por formación tiene el objetivo de obtener una posición deseada para el conjunto de robots por medio de una retroalimentación que incluya las posiciones deseadas del conjunto. Se analiza el esquema Líder-Seguidor el cual requiere de una distancia y un ángulo deseado para poder describir la posición del móvil. Este esquema se muestra en la Figura 1, donde se presenta una formación de dos robots con el fin de simplificar el análisis para un conjunto de n robots.

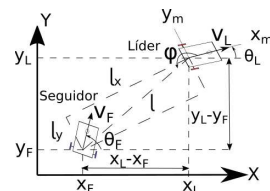


Figura. 1. Esquema Líder-Seguidor.

En la Figura 1, X y Y representan el marco inercial, mientras que x_m y y_m son el marco móvil fijo al cuerpo del líder; se representa la nomenclatura para las posiciones del líder y seguidor con los subíndices L y F respectivamente. La distancia de separación entre ambos vehículos se representa por la letra l y el ángulo deseado entre ellos por φ . Finalmente θ representa el ángulo respecto a la horizontal (eje X) dependiendo del móvil indicado y v representa la velocidad lineal de los móviles.

El modelo cinemático para el robot líder está dado por,

$$\begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{\theta}_L \end{bmatrix} = \begin{bmatrix} \cos\theta_L \\ \sin\theta_L \\ 0 \end{bmatrix} v_L + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_L, \quad (1)$$

donde w_L representa la velocidad angular. De forma similar para el seguidor se tiene,

$$\begin{bmatrix} \dot{x}_F \\ \dot{y}_F \\ \dot{\theta}_F \end{bmatrix} = \begin{bmatrix} \cos\theta_F \\ \sin\theta_F \\ 0 \end{bmatrix} v_F + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_F. \quad (2)$$

A partir de la Figura 1, se define la proyección de la separación l entre los robots como,

$$\begin{aligned} l_x &= -(x_L - x_F) \cos(\theta_L) - (y_L - y_F) \sin(\theta_L), \\ l_y &= (x_L - x_F) \sin(\theta_L) - (y_L - y_F) \cos(\theta_L). \end{aligned} \quad (3)$$

Al definir $e_\theta = \theta_F - \theta_L$, junto con las identidades trigonométricas,

$$\begin{aligned} \cos(\theta_F - \theta_L) &= \sin(\theta_L) \sin(\theta_F) + \cos(\theta_L) \cos(\theta_F) \\ \sin(\theta_F - \theta_L) &= \cos(\theta_L) \sin(\theta_F) - \sin(\theta_L) \cos(\theta_F), \end{aligned}$$

se tiene que, a partir de los modelos (1) y (2), las derivadas con respecto del tiempo de l_x , l_y y e_θ están dadas por

$$\begin{aligned} \dot{l}_x &= \cos(e_\theta) v_F - v_L + l_y w_L \\ \dot{l}_y &= \sin(e_\theta) v_F - l_x w_L \\ \dot{e}_\theta &= w_F - w_L. \end{aligned} \quad (4)$$

Definiendo ahora los errores de seguimiento e_x y e_y como

$$\begin{aligned} e_x &= l_{xd} - l_x \\ e_y &= l_{yd} - l_y, \end{aligned} \quad (5)$$

donde

$$\begin{aligned} l_{xd} &= l_d \cos(\varphi_d), \\ l_{yd} &= l_d \sin(\varphi_d), \end{aligned} \quad (6)$$

siendo l_d la distancia de separación deseada entre los robots líder y seguidor, mientras que φ_d es el ángulo deseado entre ellos, se tiene entonces que la dinámica de los errores de seguimiento está dada por,

$$\dot{e}_x = w_L e_y - \cos(e_\theta) v_F + f_1 + \Delta f_1 \quad (7a)$$

$$\dot{e}_y = -\sin(e_\theta) v_F - w_L e_x + f_2 + \Delta f_2 \quad (7b)$$

$$\dot{e}_\theta = w_F - w_L, \quad (7c)$$

con,

$$\begin{aligned} f_1 &= -l_d (\dot{\varphi}_d) \sin(\varphi_d) - w_L l_d \sin(\varphi_d) \\ &\quad + (\dot{l}_d) \cos(\varphi_d) + v_L, \end{aligned} \quad (8a)$$

$$\begin{aligned} f_2 &= l_d (\dot{\varphi}_d) \cos(\varphi_d) + w_L l_d \cos(\varphi_d) \\ &\quad + (\dot{l}_d) \sin(\varphi_d). \end{aligned} \quad (8b)$$

Nótese que f_1 y f_2 son funciones suaves y acotadas ya que dependen de los valores deseados l_d , φ_d y de las señales de control del vehículo líder, v_L , w_L que pueden también considerarse acotadas. Δf_1 y Δf_2 son términos inciertos que pueden estar asociados a errores de modelado o a perturbaciones externas y se supone que están acotados, más precisamente se supone que satisfacen

$$\Delta f_1 \leq \eta_1, \quad \Delta f_2 \leq \eta_2 \quad (9)$$

donde η_1 y η_2 son constantes reales positivas diferentes de cero.

III. CONTROL DE LA FORMACIÓN LÍDER-SEGUIDOR.

III-A. Control de la trayectoria.

La estrategia de control para la formación líder-seguidor se diseñó en base al concepto de equivalencia de sistemas pasivos por retroalimentación (pasivación) en conjunto con las técnicas de backstepping y de modos deslizantes. Como primer paso de diseño se considera la dinámica del error e_x dada por (7a) con $\Delta f_1 = 0$, suponiendo que el término $w_L e_y$ es una entrada virtual y denotándolo como $\alpha(e_x)$, es decir,

$$\alpha(e_x) = e_y w_L. \quad (10)$$

Se define la función de conmutación,

$$s_1 = e_x + k_1 \int_0^t e_x(\tau) d\tau \quad (11)$$

donde k_1 es una constante real. Esta función de conmutación permite definir la llamada superficie deslizante $s_1 = 0$. En esta superficie, $\dot{s}_1 = 0$ y la dinámica del error e_x con esta superficie está dada por,

$$\dot{e}_x + k_1 e_x = 0. \quad (12)$$

Entonces al escoger adecuadamente k_1 se tiene que $e_x(t) \rightarrow 0$ cuando $t \rightarrow \infty$.

Para que las trayectorias de (7a) sean atraídas a la superficie deslizante $s_1 = 0$, se hace la asignación [11],

$$\dot{s}_1 = F_1(s_1) = -\gamma_1 \text{sign}(s_1) \quad (13)$$

donde $\gamma_1 > 0$ y $\text{sign}(\cdot)$ es la función signo. A partir de la expresión para s_1 , ecuación (11), y la dinámica en (7a) con $\Delta f_1 = 0$, se tiene que el control virtual $\alpha(e_x)$ está dado por,

$$\alpha(e_x) = -k_1 e_x - \gamma_1 \text{sign}(s_1) + v_F \cos(e_\theta) - f_1. \quad (14)$$

De hecho, es posible dar una condición suficiente que garantiza la atracción a $s_1 = 0$ como se establece en el siguiente resultado.

Teorema 1. Considérese la dinámica (7a) dónde el término incierto Δf_1 satisface (9), si la ganancia γ_1 en (13) satisface,

$$\gamma_1 > \eta_1 \quad (15)$$

entonces el control virtual (14) hace que las trayectorias del sistema (7a) converjan a $s_1 = 0$.

Demostración: Considérese la candidata a función de Lyapunov,

$$V_1 = \frac{1}{2}s_1^2 \quad (16)$$

la cual es definida positiva. Al tomar la derivada de V_1 con respecto al tiempo y sustituyendo el control virtual (14) en la dinámica (7a) se tiene que,

$$\dot{V}_1 = -s_1\gamma_1 \text{sign}(s_1) + s_1\Delta f_1 \quad (17)$$

ya que $\text{sign}(x)x = |x|$ y al mayorar (17) junto con la condición (9) se tiene que V_1 está acotada cómo,

$$\dot{V}_1 \leq -\gamma_1|s_1| + \eta_1|s_1| \quad (18)$$

o bien,

$$\dot{V}_1 \leq -(\gamma_1 - \eta_1)|s_1|.$$

Así que si $\gamma_1 > \eta_1$, se tiene la convergencia de s_1 a 0. ■

Considerando ahora la función,

$$y = e_y w_L - \alpha(e_x) \quad (19)$$

como salida del sistema (7), se tiene que,

$$e_y w_L = y + \alpha(e_x).$$

Sustituyendo el valor anterior en (7a), permite escribir e_x como,

$$\dot{e}_x = y + \alpha(e_x) - \cos(e_\theta) v_F + f_1 + \Delta f_1$$

que al sustituir el control virtual $\alpha(e_x)$ se reescribe como,

$$\dot{e}_x = y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1. \quad (20)$$

Por otro lado, derivando la ecuación (19) con respecto al tiempo y sustituyendo en ella \dot{e}_y de (7b), conduce a la expresión,

$$\dot{y} = (-\sin(e_\theta) v_F - w_L e_x + f_2 + \Delta f_2) w_L + e_y \dot{w}_L - \dot{\alpha}(e_x), \quad (21)$$

donde,

$$\dot{\alpha}(e_x) = -k_1 \dot{e}_x - \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) \frac{\partial s_1}{\partial t} + \dot{v}_F \cos(e_\theta) - v_F \sin(e_\theta) \dot{e}_\theta - \dot{f}_1. \quad (22)$$

Sustituyendo (20) en la ecuación anterior y considerando $\dot{e}_\theta = \dot{\theta}_F - \dot{\theta}_L$, permite reescribir $\dot{\alpha}(e_x)$ como,

$$\begin{aligned} \dot{\alpha}(e_x) = & -k_1 [y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1] \\ & - \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) \frac{\partial s_1}{\partial t} + \dot{v}_F \cos(e_\theta) \\ & - v_F \sin(e_\theta) (\dot{\theta}_F - \dot{\theta}_L) - \dot{f}_1. \end{aligned} \quad (23)$$

Consecuentemente, \dot{y} toma la forma,

$$\begin{aligned} \dot{y} = & (-\sin(e_\theta) v_F - w_L e_x + f_2 + \Delta f_2) w_L \\ & + e_y \dot{w}_L - [-k_1 [y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1] \\ & - \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) \frac{\partial s_1}{\partial t} + \dot{v}_F \cos(e_\theta) \\ & - v_F \sin(e_\theta) (\dot{\theta}_F - \dot{\theta}_L) - \dot{f}_1]. \end{aligned}$$

Tomando en cuenta la expresión de la dinámica de e_x dada por (20), \dot{s}_1 se reescribe como,

$$\begin{aligned} \dot{s}_1 = & \dot{e}_x + k_1 e_x \\ = & y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 + k_1 e_x \\ = & y - \gamma_1 \text{sign}(s_1) + \Delta f_1. \end{aligned}$$

Entonces, la dinámica de y está dada por,

$$\dot{y} = \alpha(\cdot) - \dot{v}_F \cos(e_\theta) + \Delta f, \quad (24)$$

donde,

$$\begin{aligned} \alpha(\cdot) = & (-\sin(e_\theta) v_F - w_L e_x + f_2) w_L \\ & + e_y \dot{w}_L + k_1 [y - k_1 e_x - \gamma_1 \text{sign}(s_1)] \\ & + \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) [y - \gamma_1 \text{sign}(s_1)] \\ & + v_F \sin(e_\theta) (w_F - w_L) + \dot{f}_1 \\ \Delta f = & \Delta f_2 w_L + k_1 \Delta f_1 + \gamma_1 \frac{\partial}{\partial s_1} \text{sign}(s_1) \Delta f_1. \end{aligned} \quad (25)$$

A partir de los cálculos anteriores se tiene el siguiente sistema,

$$\dot{e}_x = y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 \quad (26a)$$

$$\dot{y} = \alpha(\cdot) - \dot{v}_F \cos(e_\theta) + \Delta f. \quad (26b)$$

Se busca ahora una retroalimentación que haga pasivo al sistema (26) considerando que $\Delta f = 0$. Con este fin se elige un control auxiliar de la forma,

$$\dot{v}_F = \frac{1}{\cos e_\theta} (\alpha(\cdot) - v) \quad (27)$$

donde v es una nueva entrada de control, obteniéndose el sistema,

$$\begin{aligned} \dot{e}_x = & y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 \\ \dot{y} = & v. \end{aligned} \quad (28)$$

Considerando ahora que,

$$v = -s_1 + w \quad (29)$$

se obtiene el sistema,

$$\dot{e}_x = y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 \quad (30a)$$

$$\dot{y} = -s_1 + w. \quad (30b)$$

Es posible probar que el control auxiliar \dot{v}_F de (27) y v en (29) hacen pasivo al sistema en (26) desde la entrada w hasta la salida y , considerando $\Delta f = 0$. Para tal efecto, se considera una función de almacenamiento V definida por,

$$V = V_1(s_1) + V_2(y) = \frac{1}{2}s_1^2 + \frac{1}{2}y^2. \quad (31)$$

Tomando la derivada temporal a lo largo de las trayectorias del sistema (30), y utilizando el hecho de que $\dot{s}_1 = \dot{e}_x + k_1 e_x$ junto a (30b), se obtiene,

$$\dot{V} = s_1 (\dot{e}_x + k_1 e_x) + y (-s_1 + w). \quad (32)$$

Simplificando términos se tiene que \dot{V} se puede escribir como,

$$\begin{aligned} \dot{V} &= s_1 (y - k_1 e_x - \gamma_1 \text{sign}(s_1) + \Delta f_1 + k_1 e_x) \\ &\quad + y (-s_1 + w) \\ &= s_1 (y - \gamma_1 \text{sign}(s_1) + \Delta f_1) + y (-s_1 + w) \\ &= s_1 y - \gamma_1 s_1 \text{sign}(s_1) + \Delta f_1 s_1 - s_1 y + w y \\ &= -\gamma_1 s_1 \text{sign}(s_1) + \Delta f_1 s_1 + w y \\ &= -\gamma_1 |s_1| + \Delta f_1 s_1 + w y. \end{aligned} \quad (33)$$

Tomando en cuenta la cota de Δf_1 , ecuación (9), \dot{V} se puede mayorar como,

$$\dot{V} \leq -\gamma_1 |s_1| + \eta_1 |s_1| + w y.$$

Considerando que $\gamma_1 > \eta_1$, se obtiene,

$$\dot{V} \leq -(\gamma_1 - \eta_1) |s_1| + w y.$$

Así que,

$$\dot{V} \leq w y. \quad (34)$$

Entonces la retroalimentación que logra hacer pasivo al sistema en (26) desde la entrada w hasta la salida y con $\Delta f = 0$, se escribe como,

$$\dot{v}_F = \frac{1}{\text{cose}\theta} (\alpha(\cdot) + s_1 - w). \quad (35)$$

Para mantener la propiedad de pasividad cuando $\Delta f \neq 0$, se define una segunda función de conmutación,

$$s_2 = y - \int_0^t v(\tau) d\tau. \quad (36)$$

Cuando ocurre un modo deslizante, el sistema se restringe a evolucionar sobre la superficie $s_2 = 0$ o, equivalentemente,

$$\dot{y} = v.$$

Esto se logra por medio del control equivalente para esta función de conmutación, el cual coincide con el control dado por (35), es decir,

$$\dot{v}_{F_{eq}} = \frac{1}{\text{cose}\theta} (\alpha(\cdot) + s_1 - w). \quad (37)$$

Una vez que las trayectorias son atraídas a la superficie, el control anterior asegura que permanezcan en ella. Se observa que el nuevo estado obtenido es estado cero observable por lo que es posible asignar una retroalimentación de la forma,

$$w = -k_2 y \quad (38)$$

la cual estabiliza asintóticamente al sistema (26) [11]. Si un punto inicial del sistema $e_x(0)$, $y(0)$ no pertenece a la superficie $s_2 = 0$, es posible atraer las trayectorias a ésta asignando,

$$\dot{s}_2 = \dot{y} - v = F_2(s_2) = -\gamma_2 \text{sign}(s_2) \quad (39)$$

obteniendo,

$$\dot{y} = v - \gamma_2 \text{sign}(s_2). \quad (40)$$

Para lograr este comportamiento en el sistema (26) el control, \dot{v}_F está dado por,

$$\dot{v}_F = \frac{1}{\text{cose}\theta} (\alpha(\cdot) - (v - \gamma_2 \text{sign}(s_2))). \quad (41)$$

Sustituyendo v de (27) con w de (38) y simplificando, \dot{v}_F se escribe como,

$$\dot{v}_F = \frac{1}{\text{cose}\theta} (\alpha(\cdot) + s_1 + k_2 y + \gamma_2 \text{sign}(s_2)). \quad (42)$$

O, en forma equivalente,

$$\dot{\xi} = \frac{1}{\text{cose}\theta} (\alpha(\cdot) + s_1 + k_2 y + \gamma_2 \text{sign}(s_2)) \quad (43a)$$

$$v_F = \xi. \quad (43b)$$

III-B. Control de la orientación.

La trayectoria deseada del robot seguidor está dada por,

$$\begin{aligned} x_F^d &= x_L + l_d \cos \varphi_d \\ y_F^d &= y_L + l_d \sin \varphi_d. \end{aligned} \quad (44)$$

La orientación deseada cuando el robot se encuentra fuera de la trayectoria se calcula como se muestra en la Figura 2, es decir,

$$\theta_F^d = \arctan \left(\frac{y_F^d - y_F}{x_F^d - x_F} \right). \quad (45)$$

Cuando el robot seguidor se encuentra dentro de la trayec-

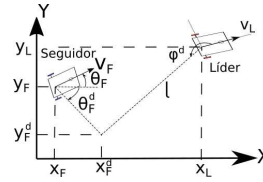


Figura 2. Orientación deseada fuera de la trayectoria.

toria (véase Figura 3), θ_F^d se calcula como,

$$\theta_F^d = \arctan \left(\frac{\dot{y}_F^d}{\dot{x}_F^d} \right). \quad (46)$$

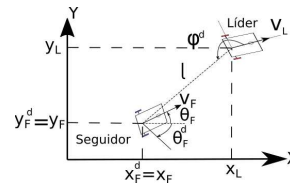


Figura 3. Orientación deseada dentro de la trayectoria.

Para ambos casos se define el control w_F de la forma dada por,

$$w_F = \dot{\theta}_F^d - k_3 e_{\theta F} \quad (47)$$

donde,

$$e_{\theta F} = \theta_F^d - \theta_F.$$

IV. EVALUACIÓN EXPERIMENTAL.

La estrategia de control propuesta anteriormente se evaluará experimentalmente considerando un conjunto de robots móviles tipo (2,0) formado por los robots Turtlebot de tercera generación. El seguidor del tipo Burger y un líder del tipo Waffle de la compañía ROBOTIS. La programación de estos robots se basa en ROS (*Robot Operating System*), para sistemas UNIX (Ubuntu (Linux)), con un programa maestro que enlaza las señales de control a los robots por medio de un programa elaborado en Python. La Figura 4 muestra la estructura general de la plataforma experimental. El sistema

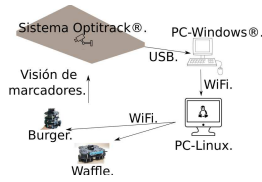


Figura. 4. Estructura general de la plataforma experimental.

de localización utilizado fue un conjunto de doce cámaras *Flex 3* de OptiTrack® y las interconexiones con el software Motive® de la compañía NaturalPoint.

Para el líder se diseñó una trayectoria deseada dada por una línea recta,

$$x_L^d = \alpha t, y_L^d = r, \theta_L^d = 0$$

donde $r, \alpha \in \mathbb{R}$. Para corroborar el funcionamiento del controlador, ante posiciones deseadas variantes en el tiempo, se eligió

$$l_d(t) = l_0 + r_0 \sin(w_0 t) \\ \varphi_d(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0.$$

Para el polinomio $\varphi_d(t)$ se eligieron,

$$\varphi_d(0) = \frac{\pi}{2}, \varphi_d(t_f) = \pi \\ \dot{\varphi}_d(0) = \dot{\varphi}_d(t_f) = \ddot{\varphi}_d(0) = \ddot{\varphi}_d(t_f) = 0$$

con un lapso de 50s para calcular las constantes a_{0-5} y realizar un cambio suave del ángulo deseado $\varphi_d(t)$ de $\frac{\pi}{2}$ a π radianes. El efecto deseado es posicionar de forma secuencial o alineada al seguidor atrás del líder para la trayectoria asignada. En la Tabla I se observan las condiciones iniciales y los valores utilizados en el controlador en la evaluación experimental se muestran en II.

$x_L(0)$	$y_L(0)$	$\theta_L(0)$	$x_F(0)$	$y_F(0)$	$\theta_F(0)$
-0.52	0.001	-0.02	-1.09	0.56	-0.09

Tabla I
CONDICIONES INICIALES.

Debido a la necesidad de derivar la función signo, para el cálculo de $\alpha(\cdot)$ en la ecuación (25), se utilizó una aproximación por medio de la función tangente hiperbólica, esto es,

k_1	k_2	k_3	γ_1	γ_2
0.001	10.0	5.0	0.05	0.01

Tabla II
GANANCIAS DEL CONTROLADOR.

$$\text{sign}(s_1) \approx \tanh(\beta s_1)$$

con lo cual,

$$\frac{\partial}{\partial s_1} \text{sign}(s_1) \approx \beta (\text{sech}(\beta s_1))^2$$

donde el parámetro $\beta = 100$ controla la pendiente de la curva y la forma de su derivada.

IV-A. Ángulo deseado.

El valor de la posición angular deseada se obtiene a partir de,

$$\theta_F^d = \arctan \left(\frac{(y_F^d - y_F)N + \dot{y}_F^d O}{(x_F^d - x_F)N + \dot{x}_F^d O} \right) \quad (48)$$

donde,

$$N = \frac{1}{2} (\tanh(\epsilon_1(\zeta - \lambda_1)) + 1) \\ O = \frac{1}{2} (-\tanh(\epsilon_2(\zeta - \lambda_2)) + 1)$$

con

$$\zeta = \sqrt{e_x^2 + e_y^2}.$$

Se remarca que (48) es diferente a (45) y (46) debido a que (48) utiliza los errores de seguimiento e_x y e_y para definir una norma ζ y con eso establecer regiones “dentro” y “fuera” de la trayectoria, si $\zeta > \lambda_1$ y $\zeta > \lambda_2$ se define la región “fuera” de la trayectoria, si $\zeta \leq \lambda_1$ y $\zeta \leq \lambda_2$ se define la región “dentro” de la trayectoria, con $\lambda_{1,2}$ constantes iguales. $\epsilon_{1,2}$ son constantes para tener una transición suave de una región a otra. Finalmente con las funciones O y N se aproxima la definición (48) con (45) fuera la trayectoria deseada, de manera similar con estas funciones se aproxima la ecuación (48) con (46) dentro de la trayectoria deseada .

IV-B. Gráficas evaluación experimental.

En la Figura 5 se muestran imágenes de la evaluación experimental de los robots.

En la Figura 6 se muestra la trayectoria de la formación



Figura. 5. Robots en diferentes tiempos del experimento.

líder-seguidor en el plano X-Y. En la Figura 7, se muestra la evolución de la distancia $l(t)$ y su correspondiente valor deseado $l_d(t)$. La evolución del ángulo $\varphi(t)$ y su valor deseado $\varphi_d(t)$ se muestran en la Figura 8. La Figura 9 muestra la evolución del control v_F y la Figura 10 muestra el control angular w_F .

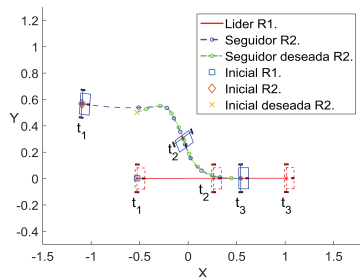


Figura 6. Trayectorias experimentales.

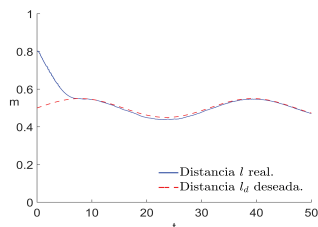


Figura 7. Evolución de la distancia deseada y distancia real.

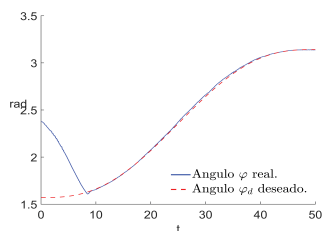


Figura 8. Evolución del ángulo deseado y ángulo real.

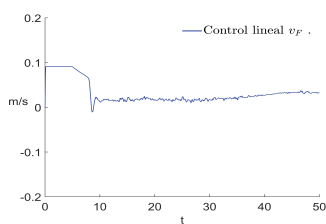


Figura 9. Control experimental v_F para R2.

V. CONCLUSIONES.

Este trabajo presenta el diseño y evaluación experimental de un controlador para una formación líder-seguir de dos robots móviles diferenciales. El diseño del controlador está basado en la metodología de backstepping, de modos

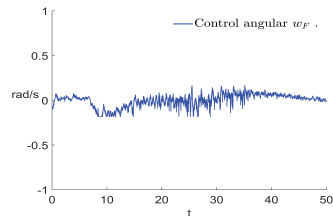


Figura 10. Control experimental w_F para R2.

deslizantes y de pasividad para la velocidad lineal del robot seguidor con el fin de realizar un controlador robusto a incertidumbres acotadas. En la práctica, este control se implementó en una plataforma compuesta por dos robots móviles y un sistema de localización basado en visión artificial. Se plantea como trabajo futuro agregar otro móvil con el fin de lograr formaciones más complejas. Otra de las posibilidades es la implementación de este control de forma descentralizada donde el procesamiento de las señales de control se realice a bordo de los robots.

REFERENCIAS

- [1] S. Kang, C. Cho, J. Lee, D. Ryu, C. Park, K.-C. Shin, M. Kim, "ROBHAZ-DT2: design and integration of passive double tracked mobile manipulator system for explosive ordnance disposal", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003, vol.3, pp. 2624-2629.
- [2] A. Yamashita, T. Arai, J. Ota and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation", IEEE Transactions on Robotics and Automation, 2003, 19(2):223-236.
- [3] S. Berman, Y. Edan, and M. Hamshidi, "Navigation of decentralized autonomous automatic guided vehicles in material handling", IEEE Transactions on Robotics and Automation, 2004, 19(4): 743-749.
- [4] W. Burgard, M. Moors, C. Stachniss and F. E. Schneider, "Coordinated multi-robot exploration", IEEE Transactions on Robotics, 2005, 21(3) : 376-386.
- [5] T. Balch, and R. C. Arkin, "Behaviour-based formation control for multirobot teams", IEEE Trans. Robot. Autom., 1998, 14, pp. 926-939.
- [6] M. A. Lewis, and K.-H. Tan, "High precision formation control of mobile robots using virtual structures", Auton. Robots, 1997, vol 4, pp. 387-403.
- [7] J. P. Desai, V. Kumar and P. Ostrowski, "Modeling and control of formations of nonholonomic mobile robots", IEEE Transactions on Robotics and Automation, 2001, 17(6): 905-908.
- [8] G. L. Mariottini, G. Pappas, D. Prattichizzo, K. Daniilidis, "Vision-based Localization of Leader-Follower Formations", Proc. of the IEEE Conference on Decision and Control, 2005, pp. 635-640.
- [9] J. Chen, D. Sun, J. Yang, and H. Chen, "Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme", The International Journal of Robotics Research, 2009, vol 29, pp. 727-747.
- [10] R. Castro, J. Alvarez, and J. Martinez, "Robot formation control using backstepping and sliding mode techniques", In 6th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE, 2009, pp. 1-6.
- [11] V. Utkin, J. Guldner, and J. Shi, *Sliding mode control in electromechanical systems*, U.S.A., CRC Press, 1999
- [12] R. Sepulchre, M. Jankovic, P. V. Kokotovic, *Constructive Nonlinear Control* London: Springer, 1997.
- [13] C. Canudas de Wit, B. Siciliano, G. Bastin (Eds.), *Theory of Robot Control*, Springer, London, 1996.
- [14] H. K. Khalil, *Nonlinear Systems*, 2nd Edition, Upper Saddle River, NJ, 1996.

Código Experimento

```
1 #!/usr/bin/env python
2 import rospy
3 from math import sqrt , cos , sin , tanh , atan2 , pi , tanh , cosh
4 import tf2_ros
5 import geometry_msgs.msg
6 import turtlesim.srv
7 import sys , select , termios , tty
8 import time
9 import datetime as DT
10 from tf.transformations import euler_from_quaternion ,
    quaternion_from_euler
11 from std_msgs.msg import Float64
12 def miapagado() :
13     try :
14         fh.close()
15     except :
16         print('El archivo ya estaba cerrado gracias por
            preguntar')
17     print "Me voy apagar!"
18     msgf = geometry_msgs.msg.Twist()
```

C. CÓDIGO EXPERIMENTO

```
19     msgf.angular.z =0.0
20     msgf.linear.x = 0.0
21     turtle_vel1.publish(msgf)
22     turtle_vel2.publish(msgf)
23 def controladorR1(x1,y1,theta1,x1d,y1d,theta1d,v1d,w1d):
24     k=0.2
25     k3=0.5
26     x1_tilde=x1-x1d
27     y1_tilde=y1-y1d
28     theta1_tilde=theta1-theta1d
29     errorx1=-cos(theta1)*x1_tilde-sin(theta1)*y1_tilde
30     errory1=sin(theta1)*x1_tilde-cos(theta1)*y1_tilde
31     errorTheta1=-theta1_tilde
32     v1=v1d*cos(errorTheta1)+k*errorx1
33     w1=w1d+k3*errorTheta1+v1d*errory1*(sin(errorTheta1)/
        errorTheta1)
34     return v1,w1,errorx1, errory1, errorTheta1
35 def trayectoriaR1(t):
36     r=0
37     l=1.5
38     t1=50.0
39     if (t>=0.0) & (t<=t1):
40         ts1=t-0.0#tiempo para este segmento.
41         a0=l/t1#Constante para que termine en tiempo
42         q1=a0*ts1#Esta es la funcion que describe la variacion
            del parametro x
43         q1p=a0#primera derivada
44         x_s1=q1-0.5
45         y_s1=r
```

```

46     xp_s1=q1p#derivada1 posicion x
47     yp_s1=0.0#derivada1 posicion y
48     theta_s1=0.0#posicion theta
49     thetap_s1=0.0#derivada1 posicion theta
50     tecla='w'
51     x1d=x_s1
52     y1d=y_s1
53     theta1d=theta_s1
54     x1dp=xp_s1
55     y1dp=yp_s1
56     theta1dp=thetap_s1
57     v1d=sqrt(x1dp**2+y1dp**2)
58     w1d=thetap_s1
59     else:
60         tecla = ' '
61         x1d=0.0
62         y1d=0.0
63         theta1d=0.0
64         x1dp=0.0
65         y1dp=0.0
66         theta1dp=0.0
67         v1d=0.0
68         w1d=0.0
69         #fh.close()
70     return x1d, y1d, theta1d, x1dp, y1dp, theta1dp, v1d, w1d,
        tecla
71 def trayectoriaR2(t, x1, y1, theta1, v1, w1, x2, y2, theta2):
72     x1p=v1*cos(theta1)
73     y1p=v1*sin(theta1)

```

C. CÓDIGO EXPERIMENTO

```
74     v1p=0.0
75     w1p=0.0
76     x1pp=v1p*cos(theta1)-v1*sin(theta1)*w1
77     y1pp=v1p*sin(theta1)+v1*cos(theta1)*w1
78     w=1.0/5.0
79     Am=0.05
80     L=0.5+Am*sin(w*t)
81     Lp=Am*cos(w*t)*w
82     Lpp=-Am*sin(w*t)*(w**2)
83     a0=1.5708
84     a1=1.9984e-17
85     a2=-1.1102e-18
86     a3=0.00012566
87     a4=-3.7699e-06
88     a5=3.0159e-08;
89     fi= a5 * t ** 5 + a4 * t ** 4 + a3 * t ** 3 + a2*t**2 + a1
          *t+a0
90     fip=5 * a5 * t ** 4 + 4 * a4 * t ** 3 + 3 * a3 * t ** 2 +
          2 * a2 * t + a1
91     fipp=20 * a5 * t **3 + 12 * a4 * t ** 2 + 6 * a3 * t + 2 *
          a2
92     x2d=x1+L*cos(fi)
93     x2dp=x1p+L*(-sin(fi)*fip)+Lp*cos(fi)
94     x2dpp=x1pp-L*sin(fi)*fipp-L*cos(fi)*((fip)**2)+2*Lp*(-sin(
          fi)*fip)+Lpp*cos(fi)
95     y2d=y1+L*sin(fi)
96     y2dp=y1p+L*(cos(fi)*fip)+Lp*sin(fi)
97     y2dpp=y1pp+L*cos(fi)*fipp-L*sin(fi)*((fip)**2)+2*Lp*(cos(
          fi)*fip)+Lpp*sin(fi)
```

```

98     w2d=(x2dp*y2dpp-x2dpp*y2dp)/((x2dp)**2+(y2dp)**2)
99     lx=-(x1-x2)*cos(theta1)-(y1-y2)*sin(theta1)
100    ly=(x1-x2)*sin(theta1)-(y1-y2)*cos(theta1)
101    lxd=L*cos(fi)
102    lyd=L*sin(fi)
103    e2x=lxd-lx
104    e2y=lyd-ly
105    e2thetaFL=theta2-theta1
106    d=sqrt(e2x**2+e2y**2)
107    A1=10.0
108    A2=10.0
109    a1=0.05
110    a2=0.05
111    F1=(tanh(A1*(d-a1))+1.0)/2.0
112    F2=(-tanh(A2*(d-a2))+1.0)/2.0
113    theta2d=atan2((y2d-y2)*F1+y2dp*F2,(x2d-x2)*F1+x2dp*F2)
114    f1=L*fip*sin(fi)-w1*L*sin(fi)+Lp*cos(fi)+v1
115    f1p=L*((fip**2)*cos(fi)+fipp*sin(fi))-Lp*fip*sin(fi)-w1*(
        L*cos(fi)*fip+Lp*sin(fi))-w1p*L*sin(fi)-Lp*sin(fi)*fip+
        Lpp*cos(fi)+v1p
116    f2=L*fip*cos(fi)+w1*L*cos(fi)+Lp*sin(fi)
117    return w2d, theta2d, L, fi, e2x, e2y, e2thetaFL, f1, f2, x2d,
        y2d, f1p, w1p
118 def YWAlphaex(e2x, e2y, e2thetaFL, w1, s1, xi, f1, k1):
119     gamma1=0.05#0.05
120     k2=10.0
121     A=100.0
122     alphaex=-k1*e2x-gamma1*tanh(A*s1)+xi*cos(e2thetaFL)-f1
123     y=e2y*w1-alphaex

```

C. CÓDIGO EXPERIMENTO

```
124     w=-k2*y
125     return y, w, gamma1, k2,A
126 def sistemaAuxiliar (e2thetaFL , xi , s1 , s2 , y , w1 , w1p , w2 , e2x , e2y , f2 ,
    f1p , k1 , gamma1 , k2 , A) :
127     gamma2=0.01#0.1 , 1.0
128     alphaPunto=(- sin ( e2thetaFL ) * xi - w1 * e2x + f2 ) * w1 + e2y * w1p + k1 * ( y
        - k1 * e2x - gamma1 * tanh ( A * s1 ) ) + gamma1 * ( A * ( ( 1.0 / cosh ( A * s1 ) )
            ** 2 ) ) * ( y - gamma1 * tanh ( A * s1 ) ) + xi * sin ( e2thetaFL ) * ( w2 - w1 ) +
            f1p ;
129     xip = ( 1.0 / cos ( e2thetaFL ) ) * ( alphaPunto + s1 + k2 * y + gamma2 * tanh ( A
        * s2 ) ) ;
130     return xip
131 def controladorR2 ( w2d , xi , theta2d , theta2 ) :
132     v2 = xi
133     k3 = 5.0
134     error2ThetaF = theta2d - theta2
135     w2 = w2d + k3 * error2ThetaF
136     return v2 , w2 , error2ThetaF
137 if __name__ == '__main__':
138     rospy.init_node ( 'controlador1' ) # nombre
139     tfBuffer = tf2_ros.Buffer () # comunicacion
140     listener = tf2_ros.TransformListener ( tfBuffer ) #
        comunicacion
141     turtle_vel1 = rospy.Publisher ( '/cmd_vel_waffle' ,
        geometry_msgs.msg.Twist , queue_size=1)
142     turtle_vel2 = rospy.Publisher ( '/cmd_vel' , geometry_msgs.
        msg.Twist , queue_size=1)
143     prueba1 = rospy.Publisher ( '/prueba/p1' , Float64 , queue_size
        =1)
```

```
144 prueba2=rospy.Publisher('/prueba/p2', Float64, queue_size
    =1)
145 prueba3=rospy.Publisher('/prueba/p3', Float64, queue_size
    =1)
146 prueba4=rospy.Publisher('/prueba/p4', Float64, queue_size
    =1)
147 prueba5=rospy.Publisher('/prueba/p5', Float64, queue_size
    =1)
148 control_v1=0.0
149 control_w1=0.0
150 control_v2=0.0
151 control_w2=0.0
152 linear_vel1 = 0.0
153 angular_vel1 = 0.0
154 linear_vel2 = 0.0
155 angular_vel2 = 0.0
156 rate = rospy.Rate(100)
157 tecla = 'w'#teclado()#'w'
158 msg1 = geometry_msgs.msg.Twist()
159 msg2 = geometry_msgs.msg.Twist()
160 rospy.on_shutdown(miapagado)
161 m2_s0=0.0
162 m2_s1=0.0
163 theta2=0.0
164 deri2=0.0
165 acumulador2=0.0
166 cuenta2=0
167 antes=DT.datetime.now()
168 ahora=DT.datetime.now()
```

C. CÓDIGO EXPERIMENTO

```
169     taux=0.0
170     t=0.0
171     cuenta=0
172     deltat=0.0
173     fh = open( 'datos.txt', 'a')
174     e2x_ant1=0.0
175     v_ant1=0.0
176     xip_ant1=0.0
177     Integral_e2x=0.0
178     xi=0.0
179     Integral_v=0.0
180     while not rospy.is_shutdown():
181         antes=ahora
182         ahora = DT.datetime.now()
183         try:
184             trans1 = tfBuffer.lookup_transform( 'world', '
                RigidBody1', rospy.Time())
185             trans2 = tfBuffer.lookup_transform( 'world', '
                RigidBody2', rospy.Time())
186         if cuenta==0:
187             cuenta=1
188             print "Me ejecuto una sola vez"
189             taux=0.0
190             Integral_e2x=0.0
191             xi=0.0
192             Integral_v=0.0
193             w2=0.0
194         else:
195             deltat=(ahora-antes).total_seconds()
```

```

196         taux=taux+deltat
197         t=taux
198     except (tf2_ros.LookupException, tf2_ros.
           ConnectivityException, tf2_ros.
           ExtrapolationException):
199         print "Espera aun no obtengo las transformadas"
200         rate.sleep()
201         continue
202     quaternion1=[trans1.transform.rotation.x,trans1.
           transform.rotation.y,trans1.transform.rotation.z,
           trans1.transform.rotation.w]
203     euler1 = euler_from_quaternion(quaternion1)
204     roll1 = euler1[0]
205     pitch1 =euler1[1]
206     yaw1 = euler1[2]
207     anguloAcompensar1=roll1
208     theta1=anguloAcompensar1
209     quaternion2=[trans2.transform.rotation.x,trans2.
           transform.rotation.y,trans2.transform.rotation.z,
           trans2.transform.rotation.w]
210     euler2 = euler_from_quaternion(quaternion2)
211     roll2 = euler2[0]
212     pitch2 =euler2[1]
213     yaw2 = euler2[2]
214     anguloAcompensar2=roll2
215     if cuenta2==0:
216         cuenta2=1
217         print "Me ejecuto una sola vez para otra cosa para
           robot2"

```

C. CÓDIGO EXPERIMENTO

```
218         m2_s0=0;
219         m2_s1=anguloAcompensar2
220         acumulador2=0.0#-0.0301#0#-0.0162994
221         theta2=acumulador2+anguloAcompensar2
222     else :
223         m2_s0=m2_s1
224         m2_s1=roll2
225         #Se obtiene la derivada2
226         deri2=(m2_s1-m2_s0)/0.001
227         umbral2=1000
228         if deri2==0.0:
229             acumulador2=acumulador2
230         elif ((deri2 >0.0) & (deri2 <umbral2)):
231             acumulador2=acumulador2
232         elif ((deri2 <0.0) & (deri2 >-umbral2)):
233             acumulador2=acumulador2
234         elif deri2<-umbral2:
235             acumulador2=acumulador2+2.0*pi
236         elif deri2>umbral2:
237             acumulador2=acumulador2-2.0*pi
238         theta2=acumulador2+anguloAcompensar2
239     print("angulo normal: ",anguloAcompensar2,"Angulo
        transformado",theta2)
240     if tecla == 'w':
241         x1=-trans1.transform.translation.x
242         y1=trans1.transform.translation.z
243         x2=-trans2.transform.translation.x
244         y2=trans2.transform.translation.z
245         x1d, y1d, theta1d, x1dp, y1dp,theta1dp,v1d,w1d,
```

```

                tecla=trayectoriaR1(t)#Deseada1
246          v1,w1,errorx1 ,errory1 ,errorTheta1=controladorR1(x1
                ,y1 ,theta1 ,x1d,y1d,theta1d ,v1d,w1d)#Control1
247          w2d, theta2d , L, fi ,e2x,e2y,e2thetaFL , f1 , f2 , x2d
                , y2d , f1p , w1p=trayectoriaR2(t,x1,y1,theta1 ,v1
                ,w1,x2,y2,theta2)#Deseada2
248          k1=0.001
249          s1=k1*Integral_e2x+e2x
250          y , w , gamma1 , k2 ,A=YWAlphaex(e2x ,e2y ,e2thetaFL ,w1 ,
                s1 ,xi ,f1 ,k1)
251          s2=y-Integral_v
252          xip=sistemaAuxiliar(e2thetaFL ,xi ,s1 ,s2 ,y,w1,w1p,w2
                ,e2x ,e2y ,f2 ,f1p ,k1 ,gamma1 ,k2 ,A)
253          #----- Control -----
254          v2 , w2 , error2ThetaF=controladorR2(w2d,xi ,theta2d ,
                theta2)
255          v=-s1+w#
256          Integral_e2x=0.5*(e2x+e2x_ant1)*deltat+
                Integral_e2x
257          Integral_v=0.5*(v+v_ant1)*deltat+Integral_v
258          xi=0.5*(xip+xip_ant1)*deltat+xi
259          e2x_ant1=e2x
260          v_ant1=v
261          xip_ant1=xip
262          if v2>=0.1:
263              v2=0.1
264          elif v2<=-0.1:
265              v2=-0.1
266          else:

```

C. CÓDIGO EXPERIMENTO

```
267         v2=v2
268     if w2>=0.2:
269         w2=0.2
270     elif w2<=-0.2:
271         w2=-0.2
272     else :
273         w2=w2
274     control_v1=v1
275     control_w1=w1
276     control_v2=v2
277     control_w2=w2
278     factor_waffle_v=1.0
279     factor_waffle_w=1.0
280     factor_burger1_v=1.0
281     factor_burger1_w=1.0
282     linear_vel1 =factor_waffle_v*control_v1
283     angular_vel1=factor_waffle_w*control_w1
284     linear_vel2 =factor_burger1_v*control_v2
285     angular_vel2=factor_burger1_w*control_w2
286     cadena=str(t)+' '+str(e2x)+' '+str(e2y)+' '+str(
        error2ThetaF)+' '+str(v2)+' '+str(w2)+' '+str(
        x2)+' '+str(y2)+' '+str(theta2)+' '+str(x1)+' '
        +str(y1)+' '+str(theta1)+' '+str(x2d)+' '+str(
        y2d)+' '+str(theta2d)+'\n'
287     print t
288     fh.write(cadena)
289     elif tecla == ' ' :
290         angular_vel1 =0.0
291         linear_vel1= 0.0
```

```
292         angular_vel2 =0.0
293         linear_vel2= 0.0
294         print "Me detengo"
295         fh.close()
296     else :
297         angular_vel1 =angular_vel1
298         linear_vel1=linear_vel1
299         angular_vel2 =angular_vel2
300         linear_vel2=linear_vel2
301     msg1.angular.z =angular_vel1
302     msg1.linear.x = linear_vel1
303     msg2.angular.z =angular_vel2
304     msg2.linear.x = linear_vel2
305     pru1 =e2x
306     pru2 =e2y
307     pru3 =error2ThetaF
308     pru4 =v2
309     pru5 =w2
310     prueba1.publish(pru1)
311     prueba2.publish(pru2)
312     prueba3.publish(pru3)
313     prueba4.publish(pru4)
314     prueba5.publish(pru5)
315     turtle_vel1.publish(msg1)
316     turtle_vel2.publish(msg2)
317     rate.sleep()
```


Instalación TurtleBot3

Se describe la instalación y puesta en marcha de los robots Turtlebot3 Burger y Waffle.

D.0.1. Procedimiento general

Se recomienda instalar primero la tarjeta controladora y el programa en la PC remota antes del ensamble de los robots. Los pasos generales son los siguientes.

- Instalación en la PC remota.
 - Instalar Ubuntu 16.04 ROS.
 - Partición e instalación del sistema operativo.
 - Instalar ROS.
 - Agregar los paquetes necesarios para ejecutar los programas del TurtleBot3.
 - Ejecutar

```
$ wget https://raw.githubusercontent.com/ROBOTIS-  
GIT/robotis_tools/master/install_ros_kinetic.sh  
&& chmod 755 ./install_ros_kinetic.sh && bash ./  
install_ros_kinetic.sh  
$ sudo apt-get install ros-kinetic-joy ros-kinetic-  
teleop-twist-joy ros-kinetic-teleop-twist-  
keyboard ros-kinetic-laser-proc ros-kinetic-rgbd
```

D. INSTALACIÓN TURTLEBOT3

```
-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
```

```
$ cd ~/catkin_ws/src/  
$ git clone https://github.com/ROBOTIS-GIT/  
  turtlebot3_msgs.git  
$ git clone https://github.com/ROBOTIS-GIT/  
  turtlebot3.git  
$ cd ~/catkin_ws && catkin_make
```

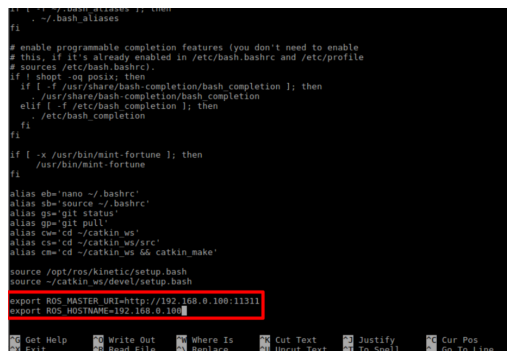
- Instalación SBO Raspberry Pi3.
 - Instalar Ubuntu 16.04 mate.
 - Configuración inicial.
 - Instalar ROS.
 - Agregar los paquetes necesarios para ejecutar los programas del TurtleBot3.
 - \$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_kinetic.sh && chmod 755 ./install_ros_kinetic.sh && bash ./install_ros_kinetic.sh
 - \$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd

```
-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
```

```
$ cd ~/catkin_ws/src/  
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git  
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git  
$ cd ~/catkin_ws && catkin_make
```

- Si en la instalación de ROS el proyecto se construyó adecuadamente se visualizará en la terminal sin errores.

- Configuración de la red



```
fi  
fi  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc)  
if ! shopt -sq posix; then  
  if [ -f /usr/share/bash-completion/bash_completion ]; then  
    . /usr/share/bash-completion/bash_completion  
  elif [ -f /etc/bash_completion ]; then  
    . /etc/bash_completion  
  fi  
fi  
if [ -x /usr/bin/mint-fortune ]; then  
  . /usr/bin/mint-fortune  
fi  
alias eb='nano ~/.bashrc'  
alias sb='source ~/.bashrc'  
alias gp='git status'  
alias gm='git pull'  
alias cm='cd ~/catkin_ws'  
alias ca='cd ~/catkin_ws/src'  
alias cma='cd ~/catkin_ws && catkin_make'  
source /opt/ros/kinetic/setup.bash  
source ~/catkin_ws/devel/setup.bash  
export ROS_MASTER_URI=http://192.168.0.100:11311  
export ROS_HOSTNAME=192.168.0.100
```

Figura D.1: Configuración archivo `/.bashrc`

- .

Bibliografía

- [1] S. Kang, C. Cho, J. Lee, D. Ryu, C. Park, K.-C. Shin, M. Kim, “ROBHAZ-DT2: design and integration of passive double tracked mobile manipulator system for explosive ordnance disposal”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, vol.3, pp. 2624-2629 .
- [2] A. Yamashita, T. Arai, J. Ota and H. Asama, “Motion planning of multiple mobile robots for cooperative manipulation and transportation”, *IEEE Transactions on Robotics and Automation*, 2003, 19(2):223-236.
- [3] S. Berman, Y. Edan, and M. Hamshidi, “Navigation of decentralized autonomous automatic guided vehicles in material handling”, *IEEE Transactions on Robotics and Automation*, 2004, 19(4): 743-749.
- [4] W. Burgard, M. Moors, C. Stachniss and F. E. Schneider, “Coordinated multi-robot exploration”, *IEEE Transactions on Robotics*, 2005, 21(3) : 376-386.
- [5] T. Balch, and R. C. Arkin, “Behaviour-based formation control for multirobot teams”, *IEEE Trans. Robot. Autom.*, 1998, 14, pp. 926-939.
- [6] M. A. Lewis, and K.-H. Tan, “High precision formation control of mobile robots using virtual structures”, *Auton. Robots*, 1997, vol 4, pp. 387-403.
- [7] T. Xiong, Z. Pu, and J. Yi , “Time-varying formation tracking control for multi-UAV systems with nonsingular fast terminal sliding mode”, *32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2017 ,IEEE.
- [8] J. P. Desai, V. Kumar and P. Ostrowski, “Modeling and control of formations of nonholonomic mobile robots”, *IEEE Transactions on Robotics and Automation*, 2001,17(6): 905-908.

BIBLIOGRAFÍA

- [9] J. Zhang, and S. Jayasuriya, “Finite-time settling real-time control for multi-robot formation”, Proceedings of the 48h IEEE Conference on Decision and Control (CDC) ,2009.
- [10] Olivares Cruz, M. Seguimiento de trayectorias libre de singularidades para una clase de robots móviles no holónomos, Tesis de Maestría, Ingeniería Eléctrica, CINVESTAV, 2016.
- [11] G. L. Mariottini, G. Pappas, D. Prattichizzo, K. Daniilidis, “Vision-based Localization of Leader-Follower Formations”, Proc. of the IEEE Conference on Decision and Control, 2005,pp. 635-640.
- [12] C. Yangyang, and T. Juping, “Formation control of three-coleader agents in the plane via backstepping design”, Chinese Control Conference, 2008, IEEE.
- [13] M. Huichao, L. Shurong, and C. Haiyang, “Robust backstepping tracking control for mobile robots”,Proceedings of the 31st Chinese Control Conference , 2012, IEEE.
- [14] A. Loria, J. Dasdemir, and N. A. Jarquin, “Leader–Follower Formation and Tracking Control of Mobile Robots Along Straight Paths”, IEEE Transactions on Control Systems Technology, 2016, vol 24, pp. 727-732.
- [15] C. Canudas de Wit, and H. Khenouf, “Quasi-continuous stabilizing controllers for nonholonomic systems: design and robustness consideration,” Proceedings of the European Control Conference, 1995.
- [16] J. Chen, D. Sun, J. Yang, and H. Chen, “Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme”, The International Journal of Robotics Research, 2009, vol 29, pp. 727-747.
- [17] R. Castro, J. Alvarez, and J. Martinez, “Robot formation control using backstepping and sliding mode techniques”, In 6th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE, 2009, pp. 1-6.
- [18] V. Utkin, J. Guldner, and J. Shi, *Sliding mode control in electromechanical systems*, U.S.A., CRC Press, 1999.
- [19] R. Sepulchre, M. Jankovic, P. V. Kokotovic, *Constructive Nonlinear Control* London: Springer, 1997.

- [20] C. Canudas de Wit, B. Siciliano, G. Bastin (Eds.), *Theory of Robot Control*, Springer, London, 1996.
- [21] H. K. Khalil, *Nonlinear Systems*, 2nd Edition, Upper Saddle River, NJ, 1996.
- [22] R. Klette, *Concise Computer Vision*, Springer, Auckland, 2014.
- [23] González Ramírez, M. Esquema Líder-Seguidor para Formación de Robots Móviles con Cámara a Bordo, Tesis de Maestría, Ingeniería Eléctrica, CINVESTAV, 2013.
- [24] Siciliano, Khatib Editors, *Handbook of Robotics*, Springer, 2008.
- [25] W. Gao and J. C. Hung, “Variable structure control of nonlinear systems: a new approach,” *IEEE Transactions on Industrial Electronics*, vol. 40, no. 1, pp. 45–55, Feb 1993.