

Universidad de Alcalá  
Departamento de Electrónica  
Escuela Politécnica Superior

MÁSTER UNIVERSITARIO EN INGENIERÍA  
ELECTRÓNICA



**Trabajo Fin de Máster**

Propuesta de control de formación de robots para  
transporte de carga única

ESCUELA POLITECNICA  
SUPERIOR

**Autor:** Melisa Pilla Barroso

**Director/es:** Felipe Espinosa Zapata

2022

UNIVERSIDAD DE ALCALÁ  
Escuela Politécnica Superior  
Departamento de Electrónica

**Máster Universitario en Ingeniería Electrónica**

Trabajo Fin de Máster

**Propuesta de control de formación de robots para  
transporte de carga única**

Autor: Melisa Pilla Barroso

Director/es: Felipe Espinosa Zapata

**Comisión Evaluadora:**

Presidente: José Luis Martín Sánchez

Vocal 1º: Julio Pastor Mendoza

Vocal 2º: Fco. Javier Rodríguez Sánchez

**Fecha:** 07 de septiembre de 2022



*Este trabajo lo dedico a mis padres Gloria y Guillermo que han estado siempre conmigo ayudándome a cumplir mis metas. A mis hermanos y familia por apoyarme siempre, los quiero mucho.*

*Agradezco a Felipe Espinoza y su guía constante en el desarrollo de este trabajo.*



## Resumen

En este trabajo se propone una solución de control colaborativo donde una formación de robots es capaz de transportar una carga de dimensiones conocidas. La formación es vista como una estructura rígida con un robot virtual en el centro y los demás a su alrededor. Primero se genera una trayectoria con continuidad tangencial y de curvatura. Después, el control no lineal de seguimiento de trayectoria se aplica al robot virtual y se obtienen sus velocidades lineal y angular, a partir de las cuales y con las ecuaciones del movimiento del sólido rígido se calculan las velocidades del resto de robots. Finalmente se hace el control local de seguimiento de velocidades de cada uno de ellos.

**Palabras clave:** Generación de trayectorias, control no lineal, Lyapunov based controller (LBC), formación de robots, sólido rígido.

## Abstract

In this work, a collaborative control solution is proposed where a formation of robots can transport a load of known dimensions. The formation is seen as a rigid structure with a virtual robot in the center and the others around it. First, a trajectory with tangential and curvature continuity is generated. Then, the non-linear trajectory tracking control is applied to the virtual robot and its linear and angular velocities are obtained, from which, and with the motion equations of a rigid body, the velocities of the rest of the robots are calculated. Finally, the local speed tracking control of each of them is carried out.

**Keywords:** Trajectory tracking, non-linear control, Lyapunov based controller (LBC), robot formation, rigid body.



# Índice de Contenidos

Resumen .....	vi
Abstract.....	vi
Índice de Figuras .....	x
1. INTRODUCCIÓN.....	12
1.1. Objetivos.....	13
1.2. Revisión de trabajos previos .....	14
1.3. Retos .....	15
2. PARAMETRIZACIÓN DEL PROBLEMA .....	17
2.1. Geometría del objeto.....	19
2.1.1. Sólido Rígido.....	20
2.1.2. Traslación .....	20
2.1.3. Rotación respecto a un eje .....	21
2.1.4. Movimiento plano general.....	22
2.2. Punto de referencia y puntos activos .....	22
2.2.1. Restricciones dinámicas de unidades tractoras .....	23
3. GENERACIÓN DE TRAYECTORIAS .....	26
3.1. Técnicas de suavizado de trayectorias basadas en interpolación.....	27
3.1.1. Interpolación polinomial.....	27
3.1.2. Curvas de Bézier.....	27
3.1.3. Spline cúbico .....	27
3.1.4. B-splines .....	27
3.1.5. Curvas NURBS .....	28
3.1.6. Curvas de Dubins .....	28
3.1.7. Clotoide .....	28
3.1.8. Hipocicloide .....	29
3.1.9. Suavizado de trayectorias basado en optimización de funciones .....	29
3.1.10. Trayectorias de robots con evasión de obstáculos .....	29
3.2. Análisis de herramientas disponibles en Matlab.....	30
3.3. Solución con plano definido .....	36
4. SOLUCIÓN DE CONTROL COOPERATIVO .....	43
4.1. Control no lineal de la unidad de referencia .....	44



4.2.	Distribución de control .....	46
4.2.1.	Control local de seguimiento de velocidades .....	48
4.3.	Ensayos: Simulación de la solución de control de la formación de robots.....	49
4.4.	Requerimientos para la implementación con robots reales.....	52
5.	CONCLUSIONES Y TRABAJOS FUTUROS .....	54
5.1.	Conclusiones .....	54
5.2.	Trabajos Futuros .....	54
	BIBLIOGRAFÍA .....	56

# Índice de Figuras

---

Figura 1. Configuraciones de los robots móviles con ruedas [4].	12
Figura 2. Formación triangular, posición de los robots con respecto al robot virtual.	17
Figura 3. Ejemplos de formaciones de robots.	20
Figura 4. Traslación del sólido rígido.	21
Figura 5. Vectores posición en un sólido rígido [17].	21
Figura 6. Rotación del sólido en torno a un eje fijo.	22
Figura 7. Descomposición del movimiento plano de un sólido rígido [17].	22
Figura 8. Vectores posición de la formación de robots rectangular.	23
Figura 9. Robot de tracción diferencial [20].	23
Figura 10. Relación entre variables de las componentes dinámica y cinemática del modelo del robot.	24
Figura 11. Valores de saturación de $v$ y $\omega$ .	25
Figura 12. Continuidad paramétrica [3].	26
Figura 13. Trayectoria suavizada utilizando curvas de Dubins.	28
Figura 14. Perfiles de velocidad trapezoidal y posiciones.	31
Figura 15. Generación de trayectoria con perfiles de velocidad trapezoidal.	31
Figura 16. Posiciones y velocidades obtenidas con la función bsplinepolytraj.	32
Figura 17. Generación de trayectorias basada en B-splines.	33
Figura 18. Generación de trayectoria, agregando más puntos de control.	33
Figura 19. Posiciones y velocidades de la nueva trayectoria generada.	34
Figura 20. Posiciones y velocidades de la trayectoria generada con la función waypointTrajectory.	35
Figura 21. Trayectoria suavizada generada con el método basada en interpolación de Hermite y clotoides.	35
Figura 22. Mapa del entorno para la formación de robots.	36
Figura 23. Trayectoria generada en el entorno de trabajo.	37
Figura 24. Posición y velocidades en el tiempo.	37
Figura 25. Formación de robots que se desplazarán por la trayectoria generada.	38
Figura 26. Cuadrícula de ocupación de la función vehicleCostmap [22].	39
Figura 27. Mapa del entorno con su área de inflación.	39
Figura 28. Verificación de colisiones entre la trayectoria y los obstáculos.	40
Figura 29. Nueva trayectoria generada con evasión de obstáculos.	41
Figura 30. Perfiles de velocidad y posición de la nueva trayectoria generada.	42
Figura 31. Estructura de la solución de control cooperativo para el transporte de una carga.	43
Figura 32. Esquema de control no lineal de seguimiento de trayectoria.	44
Figura 33. Descripción gráfica del seguimiento de trayectoria y la relación entre variables.	45
Figura 34. Relación entre los vectores posición de los robots de la formación.	47
Figura 35. Sistema de seguimiento con realimentación de estados y control integral.	48
Figura 36. Sistema de seguimiento más el estimador de estados.	49

Figura 37. Error de seguimiento de trayectoria .....	50
Figura 38. Consignas de velocidad lineal y angular para cada uno de los dos robots de la formación.....	50
Figura 39. Error de seguimiento de velocidades. ....	51
Figura 40. Desplazamiento de la formación de robots teniendo como referencia la trayectoria del robot virtual. ....	52
Figura 41. Formación de robots controlados por un centro remoto para el traslado de un objeto. ....	53

# CAPÍTULO 1

## 1. INTRODUCCIÓN

---

Uno de los aspectos fundamentales de un robot móvil es la capacidad de navegar de forma autónoma por un determinado entorno ya sea por el aire, agua o tierra. La autonomía de un robot móvil se refiere a la capacidad de realizar tareas de planificación, percepción y control sin intervención o con la mínima intervención humana [1],[2]. Sus aplicaciones pueden ser muy variadas desde transporte de cargas delicadas, desplazamiento por lugares peligrosos donde se requiere evadir obstáculos, tareas colaborativas con otros vehículos, etc.

Para que un robot pueda navegar de forma automática por un determinado ambiente de trabajo necesitará disponer de cierta información como el mapa de su entorno, así como su posición inicial y de destino. También será necesario conocer las restricciones, tanto las propias del robot: dimensiones, velocidades y aceleraciones máximas, autonomía, capacidad de reacción, etc.; como del área de trabajo: alteraciones de superficie, cambios de iluminación, posibles obstáculos (estáticos y dinámicos) que pudieran presentarse, etc.[3].

Adicionalmente, un robot móvil cuenta con sensores para monitorear continuamente su posición y orientación, para detectar obstáculos, e incluso actualizar el mapa de su entorno. Algunos de los más habituales son cámaras de vídeo para la estimación de la pose y la actualización del mapa de entorno; lidar o sónar para la detección de obstáculos [2], etc.

El desplazamiento de un robot móvil dependerá del medio de locomoción a utilizarse, existen varias opciones como las ruedas, patas y orugas. Los robots móviles con ruedas son los que más se han estudiado y sus configuraciones son muy variadas, así se tienen: Ackerman, triciclo clásico, tracción diferencial, skid steer, síncrona y tracción omnidireccional [4]. En la Figura 1 se muestran las configuraciones mencionadas.

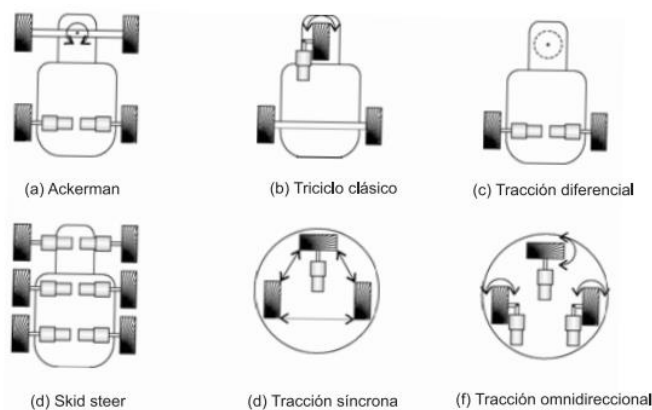


Figura 1. Configuraciones de los robots móviles con ruedas [4].

La configuración más utilizada es la de tracción diferencial (figura 1c) principalmente por tener una cinemática sencilla. Esta estructura cuenta con dos ruedas acopladas en un único eje y controladas de manera independiente con lo que se consigue la tracción y dirección requerida [5]. En este trabajo se utilizan las unidades móviles Pioneer P3-DX, que son robots de tracción diferencial lo que facilita el paso de la velocidad lineal y angular del punto de referencia a las velocidades de giro de cada una de las ruedas activas.

Cuando se habla del sistema de planificación de un robot móvil se hace referencia a la planificación de la misión, de la trayectoria e incluso a la evasión de obstáculos. Para la planificación de la trayectoria será necesario conocer la pose actual del robot, la de destino y la de los puntos (al menos algunos) intermedios de paso. También interesa definir trayectorias de curvatura continua para evitar cambios bruscos en la actuación de control y reducir errores. Definida la trayectoria se ha de implementar la solución de control que minimice el error de seguimiento teniendo en cuenta las restricciones de velocidad y aceleración máximas [1].

En entornos industriales una de las aplicaciones de los robots móviles es realizar tareas colaborativas para el traslado de cargas pesadas y de grandes dimensiones. Este tipo de tareas generalmente requieren que los robots estén dispuestos en formaciones específicas y que mantengan dichas formaciones durante su desplazamiento.

Varios autores coinciden en que se pueden considerar 3 enfoques para el control de la formación de robots, control basado en comportamientos donde cada robot deberá cumplir un comportamiento en específico, control basado en una estructura virtual rígida y el más común, control basado en una arquitectura líder-seguidor [6], [7].

En este trabajo se propone el control de la formación de robots donde la organización de los robots es vista como una estructura virtual rígida. En primer lugar, se determina un robot líder virtual ubicado en el centro del objeto a transportar, punto que sirve de referencia para el seguimiento de trayectorias del conjunto. El controlador global se encarga de planificar la trayectoria del líder y de aplicar técnicas de control no lineal basado en Lyapunov para su seguimiento. Como resultado se obtienen las velocidades  $V, W$  del robot de referencia a partir de las cuales se calculan las velocidades del resto de robots. Las velocidades  $V_i, W_i$  de los robots de la formación se calculan considerando el punto de anclaje de estas unidades móviles como puntos de un sólido rígido cuya ubicación con respecto al punto de referencia del objeto (líder virtual) es conocida. Cada unidad implementa de forma local el control lineal de seguimiento de velocidades.

## 1.1. Objetivos

El objetivo general del TFM es diseñar una solución de control para que un conjunto de robots móviles transporte, de forma cooperativa y coordinada, una carga que, por peso o tamaño, excede las capacidades de transporte de una única unidad móvil.

Para ello se proponen los siguientes objetivos particulares:

1. Revisión de la cinemática de un sólido rígido sometido a movimientos de traslación y rotación.
2. Definir la organización de los robots (número y posición relativa) en función del tamaño de la carga única.
3. Automatización de la generación de trayectoria no lineal a partir de waypoints, garantizando continuidad en las velocidades.
4. Diseño de control no lineal basado en Lyapunov para el seguimiento de la trayectoria de referencia por parte del robot líder virtual.
5. Traslación de velocidades del robot virtual al resto de los robots de la formación.
6. Estudio de viabilidad y requisitos para su implementación en un caso real.

## 1.2. Revisión de trabajos previos

En [8] se propone un esquema de control para un sistema de transporte colaborativo donde se utilizan robots móviles P3-DX organizados de acuerdo a la forma del objeto a transportar. Se aplica la estrategia de formación líder – seguidor, el líder virtual es el encargado de planificar la ruta y definir la trayectoria a seguir. Después, cada unidad seguidora determina su propia trayectoria de acuerdo con la planificación del robot líder de modo que cada robot se mantenga siempre a una distancia  $d$  y un ángulo  $\alpha$  con respecto al líder. Para el seguimiento de trayectorias se comparan dos modelos de control, un control de seguimiento asintóticamente estable y un controlador de seguimiento de modo deslizante.

En [9] al igual que en el trabajo anterior se utiliza un esquema de control líder – seguidor con la diferencia que no se utiliza la información de la posición y orientación del líder para determinar la ubicación del seguidor. La posición del seguidor se define en términos de la estructura local de los seguidores. En cuanto a la técnica de control utilizada no solo se considera el modelo cinemático del sistema sino también el modelo dinámico, de modo que aparecen variables de fuerza y momento para modelar el sistema.

En [10] se propone un nuevo método para el control de formaciones basado en la distancia y restricciones del ángulo. Se propone una ley de control en cascada que utiliza la información de ángulo y distancia entre las unidades móviles para mantener la formación. Primero se proporciona un esquema de control para la evasión de obstáculos y después se propone una ley de control basada en el ángulo y distancia con el objetivo de mantener no solo la forma sino también la dirección deseada durante el desplazamiento.

Un enfoque similar se plantea en [11], se utiliza una ley de control de la formación basada en la distancia. Debido a que las leyes de control usan solo información local, las leyes de control están completamente descentralizadas. La formación cuenta con 3 tipos de unidades, líder y dos seguidores. Únicamente el líder conoce el valor de la velocidad de referencia, por esta razón, el primer y segundo seguidor estiman la velocidad de referencia utilizando un método de control adaptativo. La formación deseada se logra solo usando información de distancia relativa entre agentes

Debido al gran tráfico de red en sistemas multi robot, en [12] se propone una estrategia de control de formación basada en el cambio de estado. Se crea una estructura modular con los algoritmos tradicionales de formación de robots: líder-seguidor, estructura virtual y basado en comportamientos. Después se diseña un mecanismo de activación de cambio de estado. Con la selección razonable del algoritmo de control se busca reducir la complejidad del algoritmo y reducir la carga en las comunicaciones entre robots. La estrategia propuesta en este trabajo ha sido aplicada en robots reales.

En [13] se propone un método de navegación colaborativo para transportar cargas basado en comportamientos. Se plantean dos modos de comportamiento: seguimiento de paredes (wall-following mode – WFM) y el modo orientado a objetos (goal oriented mode – GOM), la aplicación de uno u otro dependerá de las condiciones del entorno. Se utilizan un controlador neuro difuso tipo 2 basado en DGABC (dynamic group artificial bee colony). El controlador propuesto presenta ventajas, tales como: un número reducido de parámetros y permite reducir ruido y perturbaciones de los sensores. Se consigue además que la ruta sea más suave y que el cambio entre modos sea automático.

En [14] lo que se busca es mantener la formación de robots en un círculo, a diferencia de los trabajos antes mencionados se ha implementado un esquema de control y estimación de estado no cooperativo y descentralizado que solo se basa en mediciones de distancia inciertas e intermitentes para equilibrar el patrón de la formación. Se han utilizado distintos sensores con mayor o menor precisión para probar la efectividad del control. El algoritmo de control se lleva a cabo en una secuencia de pausa y avance, donde, durante la pausa cada robot utiliza un estimador para identificar su seguidor más cercano y evaluar su posición relativa. Durante el desplazamiento, cada robot acelera hasta alcanzar la distancia deseada y luego se mueve a una velocidad constante.

### 1.3. Retos

Tras la revisión del estado del conocimiento los principales retos del TFM son:

- Dada la necesaria cooperación entre robots, el suavizado de rutas (garantizando continuidad en velocidades y aceleraciones de referencia) es fundamental. Para ello, en este trabajo se genera la trayectoria de la unidad líder a partir de waypoints usando b-splines y se evalúa en todo momento que el robot de referencia (robot P3-DX previamente identificado) no supera los límites de velocidad y aceleración. En tal caso se realizan los ajustes necesarios de suavizado de la trayectoria.
- Los robots reales están fijados al objeto a transportar y se ha de mantener la formación durante el desplazamiento evitando deslizamientos durante el transporte. En este trabajo la formación es vista como una estructura rígida. El cálculo de las velocidades  $V_i$ ,  $W_i$  de los robots se hacen a partir de las ecuaciones que describen el movimiento plano general del sólido rígido. De este modo se consigue que la ubicación de las unidades seguidoras con respecto al líder virtual sea constante durante el desplazamiento de la formación.

- Se diseña una solución de control no lineal basada en Lyapunov para el seguimiento estable de la trayectoria de referencia por parte de la unidad líder ubicada en el punto de referencia del objeto a transportar. Las velocidades lineal y angular, calculadas para el control de la unidad líder son recalculadas para cada uno de los robots del conjunto. Cada uno de los robots reales implementa su propio control local de seguimiento de velocidades.

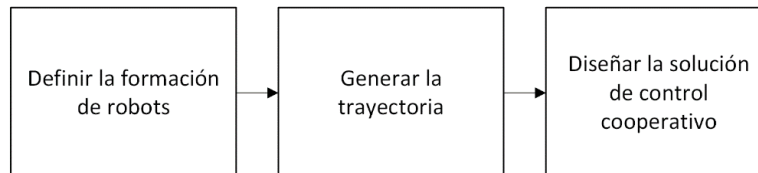
Para abordar estos retos y validar las propuestas mediante simulación se analizarán las herramientas disponibles en Matlab 2022.



# CAPÍTULO 2

## 2. PARAMETRIZACIÓN DEL PROBLEMA

De manera general el trabajo propuesto está dividido de la siguiente forma:



- La primera parte consiste en determinar la ubicación de los  $N$  robots dentro de la formación, lo que dependerá de las dimensiones y peso del objeto a transportar. En el capítulo anterior se mencionó que la formación de robots estará organizada como una estructura rígida con un robot virtual de referencia. De modo que será importante mantener la formación de los robots durante el traslado de la carga, desde el punto inicial hasta su punto de destino. Conociendo la geometría del objeto se determinará la formación, pudiendo tomar una forma poligonal, regular o irregular.

Cada robot queda caracterizado por su punto de enlace con el objeto a transportar. El robot virtual se ubica en el punto de referencia del objeto a transportar y el resto de los robots se ubicarán a una distancia  $|\vec{r}|$  con respecto al robot virtual (ver figura 2). Interesas mantener la magnitud y dirección entre los puntos del sólido durante el desplazamiento.

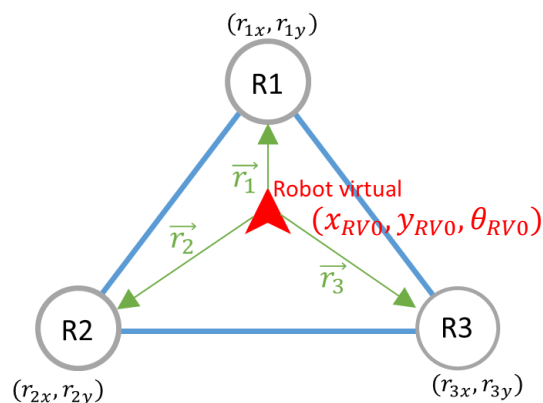


Figura 2. Formación triangular, posición de los robots con respecto al robot virtual.

En este punto será necesario conocer los siguientes parámetros de entrada:

- Posición y orientación inicial del robot virtual  
 $(x_{RV0}, y_{RV0}, \theta_{RV0})$

- Posición de los N robots de la formación.

$$(r_{1x}, r_{1y}), (r_{2x}, r_{2y}), \dots, (r_{Nx}, r_{Ny})$$

Como resultado se obtendrán los siguientes parámetros de salida.

- Pose inicial de los robots de la formación.

$$(x_{R1}, y_{R1}, \theta_{R1}), (x_{R2}, y_{R2}, \theta_{R2}), \dots, (x_{RN}, y_{RN}, \theta_{RN})$$

Donde:

$$\theta_{R1} = \theta_{R2} = \dots = \theta_{RN} = \theta_{RV0}$$

- Magnitud y dirección de los N robots con respecto al robot virtual.

$$\vec{r}_1 = (r_{1x} \vec{i} + r_{1y} \vec{j})$$

$$\vec{r}_2 = (r_{2x} \vec{i} + r_{2y} \vec{j})$$

⋮

$$\vec{r}_N = (r_{Nx} \vec{i} + r_{Ny} \vec{j})$$

- b. La segunda parte comprende la planificación de la trayectoria del robot virtual. Interesa generar una trayectoria de curvatura continua, evitando cambios bruscos y áreas puntiagudas. Es importante conocer el punto inicial y final de la trayectoria y algunos de sus puntos intermedios (waypoints). No solo interesa obtener los puntos de la trayectoria de referencia, sino también la velocidad lineal y angular de referencia que garantice una trayectoria suave (derivadas continuas de pose, velocidad y aceleración).

Los parámetros iniciales son:

- Pose inicial y final del robot virtual.

$$(x_{RV0}, y_{RV0}, \theta_{RV0})$$

$$(x_{RVf}, y_{RVf}, \theta_{RVf})$$

- Vector de waypoints.

$$[x_{wp1}, y_{wp1}, \theta_{wp1}; x_{wp2}, y_{wp2}, \theta_{wp2}; \dots; x_{wpk}, y_{wpk}, \theta_{wpk}]$$

$$k = \# \text{ waypoints}$$

- Duración de la trayectoria

$$t_{final}$$

Los parámetros de salida son:

- Trayectoria de referencia del robot virtual (vector con la posición y orientación de la trayectoria a seguir)

$$[x_{RV_r1}, y_{RV_r1}, \theta_{RV_r1}; x_{RV_r2}, y_{RV_r2}, \theta_{RV_r2}; \dots; x_{RV_rn}, y_{RV_rn}, \theta_{RV_rn}]$$

$n = \# \text{ puntos de la trayectoria discretizada con periodo } Ts$

- Referencias de velocidad lineal y angular del robot virtual para todos los puntos de la trayectoria

$$\left[ v_{RV_{r1}}, \omega_{RV_{r1}}; v_{RV_{r2}}, \omega_{RV_{r2}}; \dots; v_{RV_{rn}}, \omega_{RV_{rn}} \right]$$

Si el área de trabajo incluye restricciones físicas, se habrá de garantizar que el objeto a transportar salva dichas restricciones.

- c. La última parte corresponde al diseño de control cooperativo. Una vez se ha generado la trayectoria se diseña el seguimiento de la trayectoria aplicando teoría de control no lineal basado en Lyapunov. A partir de este punto se deducen las velocidades (lineal y angular) del resto de los robots de la formación utilizando las ecuaciones del movimiento del sólido rígido. Finalmente, se diseña el control local de cada robot.

Los parámetros de entrada son:

- Trayectoria y velocidades de referencia del robot virtual

$$\left[ x_{RV_{r1}}, y_{RV_{r1}}, \theta_{RV_{r1}}; x_{RV_{r2}}, y_{RV_{r2}}, \theta_{RV_{r2}}; \dots; x_{RV_{rn}}, y_{RV_{rn}}, \theta_{RV_{rn}} \right]$$

$$\left[ v_{RV_{r1}}, \omega_{RV_{r1}}; v_{RV_{r2}}, \omega_{RV_{r2}}; \dots; v_{RV_{rn}}, \omega_{RV_{rn}} \right]$$

El control no lineal ha de proporcionar las velocidades de actuación de la unidad virtual para cada uno de los instantes de muestreo  $j$

$$\left[ v_{RV_{uj}}, \omega_{RV_{uj}} \right], \quad 1 \leq j \leq n$$

Y, a partir de ahí, se calculan las velocidades de actuación de los robots de la formación

$$\left[ v_{Ri_{uj}}, \omega_{Ri_{uj}} \right], \quad 1 \leq i \leq N, \quad 1 \leq j \leq n$$

## 2.1. Geometría del objeto

Como se mencionó anteriormente la ubicación de los robots con respecto al objeto a transportar dependerá de su geometría. Si el objeto presenta una forma regular, los vértices servirán de referencia para la ubicación de los robots. En el centro se colocará el robot virtual y en sus extremos los demás robots de la formación. Por otro lado, si el objeto presenta una forma irregular la formación se aproximará a una forma geométrica regular. El número de robots se determinará de acuerdo con el tamaño y peso del objeto. Algunos ejemplos de la formación de robots para el traslado colaborativo se muestran en la figura 3.

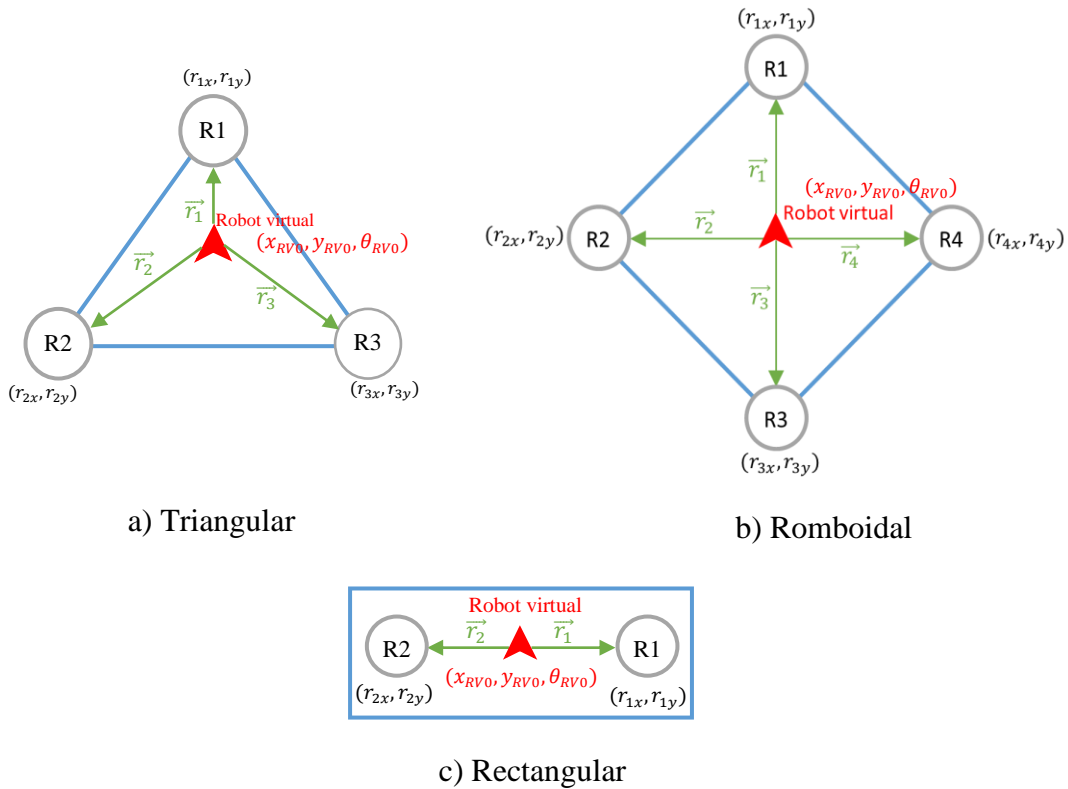


Figura 3. Ejemplos de formaciones de robots.

El objetivo de este trabajo es transportar un objeto o carga de grandes dimensiones utilizando varios robots que están organizados para trabajar de forma colaborativa. La formación de los robots es vista como una estructura rígida buscando mantener la misma formación durante todo el trayecto. Se utiliza la teoría del movimiento plano general del sólido rígido para garantizar que los robots seguidores mantengan la misma ubicación durante todo el traslado. Se admite que la solución mecánica adoptada para enlazar cada robot con el objeto a transportar permite la adaptación de giro sin deslizamiento.

A continuación, se hace una breve revisión de la teoría relacionada con la cinemática del sólido rígido sometido a movimientos de traslación y rotación.

### 2.1.1. Sólido Rígido

Se define al Sólido Rígido como “un sistema de partículas en el que la distancia relativa entre ellas permanece constante con el tiempo” [15]. Durante su movimiento el sólido rígido no modifica su forma ni volumen.

Los movimientos elementales del sólido rígido son la traslación y rotación.

### 2.1.2. Traslación

La traslación se caracteriza porque los puntos del sólido siguen la misma trayectoria (rectilínea o curvilínea) de forma paralela manteniendo la misma velocidad y aceleración en cada instante [16]. La velocidad de un punto del sólido es la misma que la velocidad de su centro de masas.

$$\vec{v}_A = \vec{v}_B = \vec{v}_C$$

$$\vec{a}_A = \vec{a}_B = \vec{a}_C$$

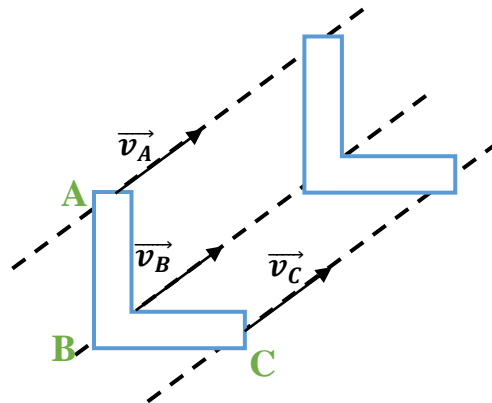


Figura 4. Traslación del sólido rígido.

Considerando dos puntos A y B pertenecientes al sólido rígido (ver figura 5) sus vectores posición se relacionan:

$$\vec{r}_B = \vec{r}_A + \vec{AB}$$

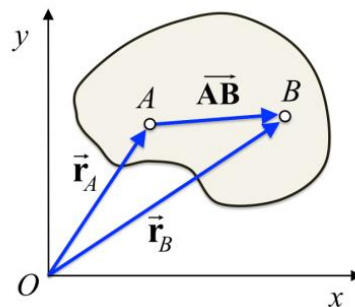


Figura 5. Vectores posición en un sólido rígido [17].

Durante la traslación el vector  $\vec{AB}$  permanece constante en modulo (ya que la distancia entre dos puntos de un sólido rígido es invariable) y en dirección (pues durante la traslación cualquier recta perteneciente a un sólido rígido mantiene su dirección en el movimiento) [18].

### 2.1.3. Rotación respecto a un eje

La rotación se efectúa de modo que un punto del sólido queda fijo mientras los otros puntos rotan con respecto a dicho punto siguiendo trayectorias circulares [16]. Las velocidades lineales son diferentes y existe una velocidad angular  $\omega$  y una aceleración  $a$  que modifican la orientación del sólido en el espacio.

$$\vec{v}_B = 0$$

$$\vec{v}_A = \vec{\omega} \times \vec{BA}$$

$$\vec{v}_C = \vec{\omega} \times \vec{BC}$$

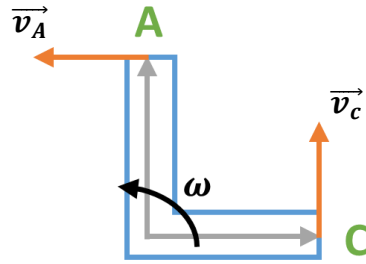


Figura 6. Rotación del sólido en torno a un eje fijo.

### 2.1.4. Movimiento plano general

Un sólido rígido realiza un movimiento plano cuando se mueve de modo que sus puntos describan trayectorias planas, paralelas entre si [19]. El movimiento general del sólido rígido se compone de una traslación y una rotación simultáneas por tanto cualquier movimiento plano se puede descomponer en una traslación de un punto cualquiera A y de forma simultánea una rotación alrededor de A [17], como se muestra en la figura 7.

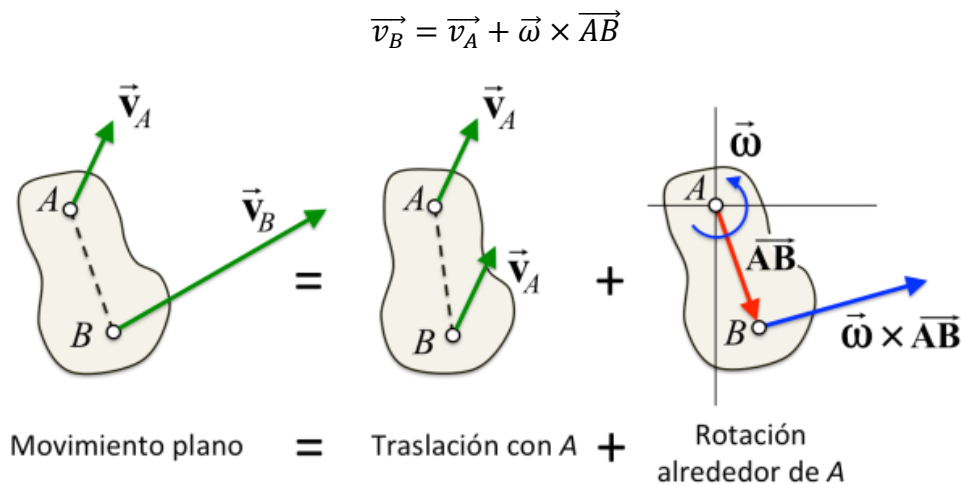


Figura 7. Descomposición del movimiento plano de un sólido rígido [17].

“La velocidad angular de un sólido rígido en movimiento plano es independiente del punto de referencia. La velocidad angular del cuerpo en su rotación alrededor del punto B es la misma que en su rotación en torno al punto A” [17].

$$\vec{\omega}_{AB} = \vec{\omega}_{BA}$$

## 2.2. Punto de referencia y puntos activos

Conocidas las coordenadas de posición de los robots de la formación se establece la pose inicial de cada uno de ellos y se calcula el vector posición de cada robot con respecto al robot virtual. Por simplicidad en este trabajo se utiliza la formación rectangular conformado por dos robots alineados con el robot virtual, tal y como se indica en la figura

3c. El robot R1 se ubica a una distancia  $r$ , a la derecha del robot virtual mientras que el robot R2 se ubica a la izquierda conservando la misma distancia que el primer robot.

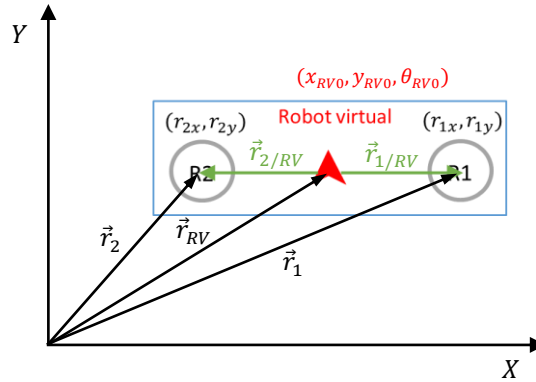


Figura 8. Vectores posición de la formación de robots rectangular.

Para calcular los vectores  $\vec{r}_{1/RV}$  y  $\vec{r}_{2/RV}$  se utiliza la relación entre los vectores posición:

$$\vec{r}_2 = \vec{r}_{RV} + \vec{r}_{2/RV}$$

$$\vec{r}_{2/RV} = \vec{r}_2 - \vec{r}_{RV}$$

$$\vec{r}_1 = \vec{r}_{RV} + \vec{r}_{1/RV}$$

$$\vec{r}_{1/RV} = \vec{r}_1 - \vec{r}_{RV}$$

## 2.2.1. Restricciones dinámicas de unidades tractoras

En este TFM se eligen como unidades tractoras los robots Pioneer P3-DX de la figura 9, con tracción diferencial, formados por dos ruedas convencionales fijas y una rueda omnidireccional de apoyo [20]. Las ruedas activas están separadas una distancia  $b$  y son actuadas por dos motores DC. Un robot de tracción diferencial puede desplazarse hacia atrás o adelante, pero no puede trasladarse lateralmente (robot no holonómico) [5]. El punto de referencia, para caracterizar su movimiento, se ubica en el centro del eje que une las ruedas activas.



Figura 9. Robot de tracción diferencial [20].

La cinemática de estos robots, en un plano horizontal, se describe de la siguiente forma:

$$\dot{x}_0 = v \cos \theta$$

$$\dot{y}_0 = v \sin \theta$$

$$\dot{\theta} = \omega$$

La pose del robot se define mediante su posición  $(x_0, y_0)$  y el ángulo de orientación  $\theta$ . Las velocidades del punto de referencia son:  $v$  lineal y  $\omega$  angular. Estas velocidades están relacionadas con las velocidades lineales de las ruedas activas según las ecuaciones

$$v = \frac{v_R + v_L}{2}$$

$$\omega = \frac{v_R - v_L}{2}$$

Téngase en cuenta que todas las variables indicadas son función del tiempo.

Para lograr un desplazamiento controlado del robot, se deben modificar las señales de entrada de la planta. La dinámica del robot establece la relación entre las señales de actuación  $v_i$  y  $\omega_i$  y las de salida  $v_o$  y  $\omega_o$ , tal y como se indica en la figura 6.

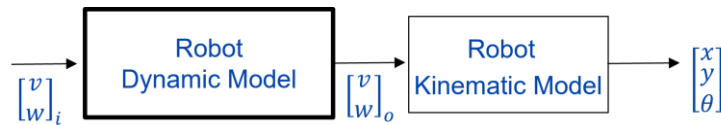


Figura 10. Relación entre variables de las componentes dinámica y cinemática del modelo del robot.

Una vez identificado experimentalmente el modelo dinámico del robot, se conocen los límites de  $v_i$  y  $\omega_i$  que evitan la saturación de la planta, tal y como se muestra en la figura 11.

Los valores máximos y mínimos de la velocidad lineal son  $\pm 0.7495$ , mientras que para la velocidad angular los límites son  $\pm 1.73619$ . Estos valores se deben tener en cuenta para que, en el diseño de la trayectoria, no se generen valores de actuación fuera del alcance de las unidades de transporte.



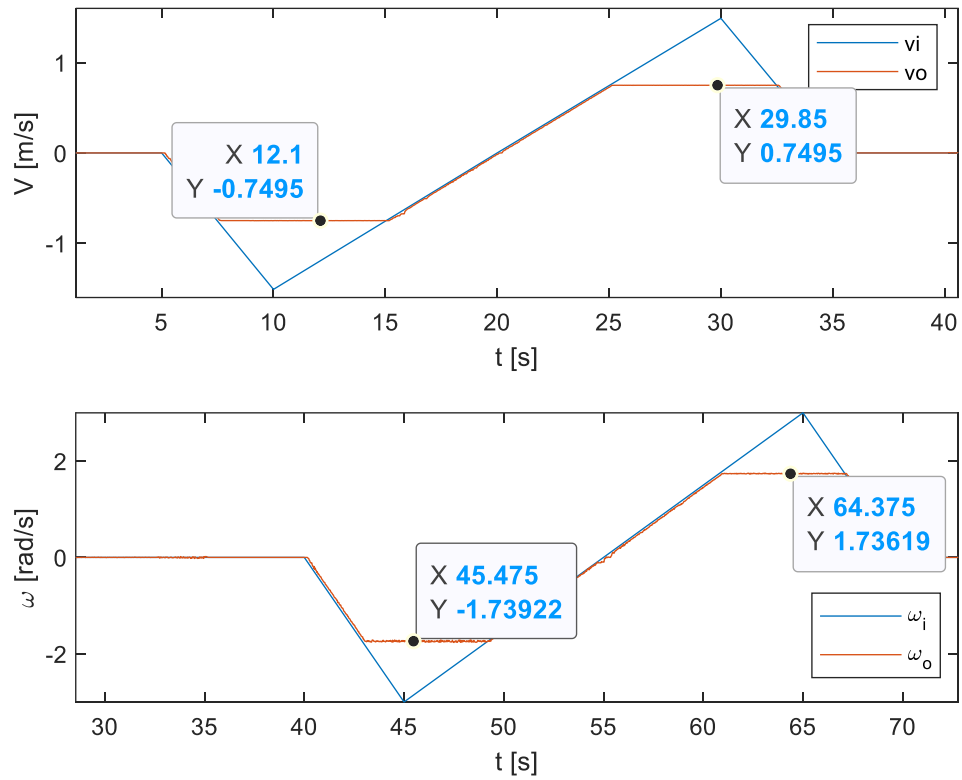


Figura 11. Valores de saturación de  $v$  y  $\omega$ .

Para la generación de trayectorias es importante tener en cuenta la velocidad del robot móvil, pues existe una relación entre la curvatura y la velocidad máxima a la que puede recorrerse dicho camino.

# CAPÍTULO 3

## 3. GENERACIÓN DE TRAYECTORIAS

Para que un robot móvil pueda desplazarse desde su punto inicial hasta un punto final se requiere definir la trayectoria a seguir. Al generar la trayectoria del robot es importante conocer tanto las características dinámicas como las características cinemáticas del robot. Es deseable conseguir una trayectoria suave y continua evitando curvas muy cerradas o secciones puntiagudas.

Para obtener caminos suavizados la generación de trayectorias ha de cumplir dos requisitos: continuidad y seguridad. La continuidad se refiere a la continuidad tangencial o de curvatura y la seguridad hace referencia a la distancia que mantendrá de los obstáculos. En el seguimiento de trayectorias la continuidad se asocia con dos conceptos: continuidad geométrica y paramétrica. Por un lado, la continuidad geométrica garantiza que los segmentos de la trayectoria estén conectados y que las direcciones del vector tangente en cada punto sean iguales. La continuidad paramétrica por otro lado asegura que tanto la dirección como la magnitud del vector tangente sean iguales. En general, si una curva tiene continuidad  $C^n$  su  $n$ -ésima derivada también será continua [3].

En la figura 12 se observan distintas trayectorias, iniciando con la figura 12a donde se muestra una trayectoria discontinua. Después, la figura 12b presenta una trayectoria con continuidad de tipo  $C^0$ , donde todos sus segmentos se encuentran conectados, pero presenta una discontinuidad de pendiente en el punto de unión indicado.

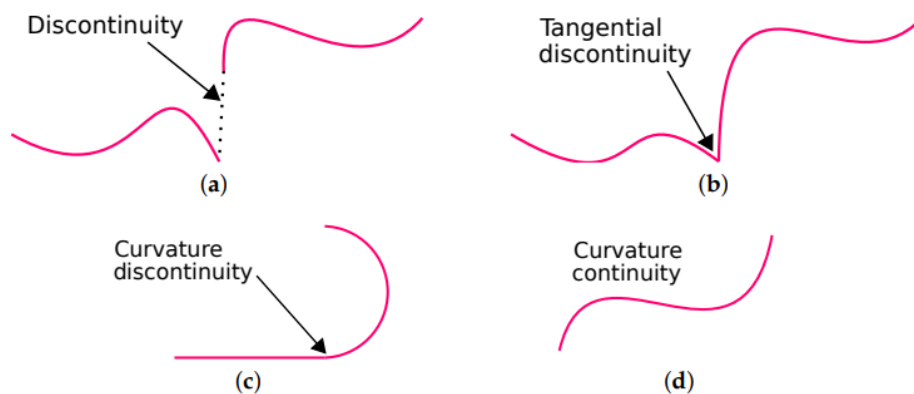


Figura 12. Continuidad paramétrica [3].

En la figura 12c se muestra una trayectoria con continuidad  $C^1$ , los segmentos se conectan y además la dirección del vector tangente es igual en el punto de unión. La trayectoria es continua pero no así su curvatura, en el punto de unión la curvatura cambia repentinamente de cero a un valor finito. La trayectoria de la figura 1d tiene continuidad  $C^2$ , donde existen valores diferenciales de segundo orden en cada punto de la ruta lo que

garantiza que exista continuidad de curvatura haciendo que la trayectoria sea más suave que los ejemplos anteriores.

En términos del movimiento del robot, la continuidad  $C^1$  hace que el movimiento sea continuo, se conserva la velocidad, mientras que en la curvatura  $C^2$  se garantiza la conservación de la aceleración [3].

## 3.1. Técnicas de suavizado de trayectorias basadas en interpolación

A continuación, se resumen las principales características de las técnicas de suavizado más utilizadas [3].

### 3.1.1. Interpolación polinomial

Algunas de las técnicas más comunes de interpolación polinomial son la interpolación de Lagrange y la interpolación de Hermite. El suavizado de trayectorias basado en métodos de interpolación polinomial son algoritmos muy antiguos y presentan ciertas limitaciones como el alto costo computacional y el fenómeno de Runge, que es un problema muy común en polinomios de alto grado. Este fenómeno demuestra que los polinomios de alto grado no son, en general, aptos para la interpolación, pues el error de interpolación tiende a infinito cuando crece el grado del polinomio. Recientemente se han propuesto algoritmos que buscan solucionar este problema con algoritmos basados en splines.

### 3.1.2. Curvas de Bézier

Este método hace uso de puntos de control para definir curvas con ciertas características además que se pueden conectar entre ellas para obtener la forma deseada. Las curvas de Bézier se describen a través de las funciones polinomiales de Bernstein. Tienen un menor costo computacional, pero tiende a crecer si las curvas deseadas son de alto grado. En la mayoría de los trabajos propuestos se ha conseguido planificar rutas en un entorno con restricciones optimizando los puntos de control. En ciertas aplicaciones las curvas de Bézier se han utilizado para conectar un segmento recto con un curvo.

### 3.1.3. Spline cúbico

Es una función polinomial por partes muy simple y a la vez flexible. El spline cúbico presenta mayores ventajas frente a la interpolación polinomial pues produce resultados similares incluso en polinomios de bajo grado a la vez que evita el fenómeno de Runge. El spline cúbico tiene la característica que son los splines de menor grado con continuidad tipo  $C^2$ .

### 3.1.4. B-splines

Cualquier función spline de un determinado grado se puede expresar como una combinación lineal de B-splines (o splines base) del mismo grado. Un B-spline es una

generalización de las curvas de Bézier. Diversos trabajos se han basado en esta técnica debido a que permite generar trayectorias suaves libres de obstáculos.

### 3.1.5. Curvas NURBS

El suavizado de rutas basado en NURBS ha demostrado ser una herramienta muy poderosa para la generación de trayectorias incluso en mapas 3D. Algunas de sus aplicaciones incluyen planificación de trayectorias de manipuladores, robots móviles autónomos, evasión de obstáculos, robótica de enjambre, etc. Una de sus desventajas es que requieren de almacenamiento adicionales y errores en su inicialización pueden provocar graves problemas.

### 3.1.6. Curvas de Dubins

Utiliza arcos circulares y segmentos rectos para encontrar el camino más suave, de curvatura acotada, que une dos puntos de un plano. En la figura 13 se muestra una trayectoria conformada por segmentos rectos (rojo) y segmentos circulares (verde). Las curvas de Dubins son ampliamente utilizadas para la evasión de obstáculos estáticos y dinámicos. Este tipo de algoritmos no presentan elevados costos computacionales y se pueden combinar con otros métodos basados en curvas especiales. Sin embargo, no garantiza continuidad de curvatura.

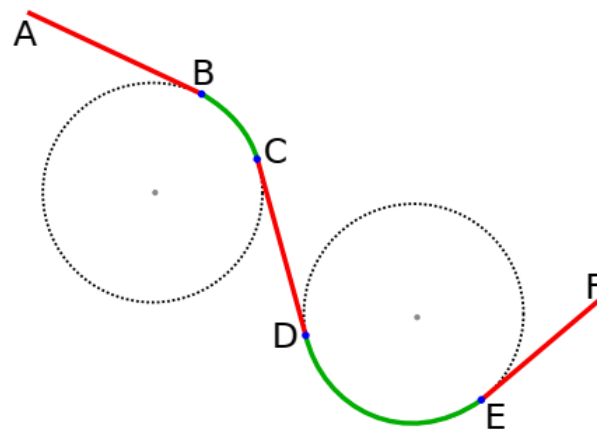


Figura 13. Trayectoria suavizada utilizando curvas de Dubins.

### 3.1.7. Clotoide

Una clotoide también conocida como espiral de Euler o espiral de Cornu es una curva parametrizada en el plano complejo. La curva de Cornu tiene la propiedad de que su curvatura es proporcional a la distancia a lo largo de la trayectoria de la curva. Este tipo de curvas se caracterizan en que su continuidad de curvatura es muy fácil de obtener y se utilizan generalmente como curvas de transición junto con otras curvas.

### 3.1.8. Hipocicloide

Una hipocicloide es una curva geométrica producida por un punto fijo  $P$  que se encuentra en la circunferencia de un círculo de radio  $r_s$  que gira dentro de un círculo de mayor tamaño, de radio  $R_L$ . Generalmente este tipo de curvas está definido por una característica conocida como cúspide, que es una esquina aguda donde la curva no es diferenciable. La hipocicloide tiene la propiedad de generar curvas con cualquier tipo de ángulo, esta propiedad suele utilizarse para mantener rectos los segmentos rectos y solo suavizar los segmentos con giros muy cerrados. Una de las desventajas de este tipo de curvas es que no garantiza la continuidad de tipo  $C^2$  y requiere de curvas de transición como las clotoides para garantizar la continuidad de curvatura.

### 3.1.9. Suavizado de trayectorias basado en optimización de funciones

Mediante la optimización de ciertos parámetros como la velocidad, la aceleración y otros se ha logrado obtener trayectorias suaves en entornos con obstáculos. Entre los trabajos más destacables que utilizan la optimización de funciones está el planificador de banda elástica de tiempo (TEB planner) y además cuenta con integración ROS. El planificador TEB optimiza las trayectorias de los robots mediante la modificación posterior de una trayectoria inicial generada por un planificador global. Algunos parámetros considerados en la optimización de la trayectoria incluyen la longitud de la trayectoria, el tiempo de ejecución, la separación de obstáculos, el paso por puntos intermedios, las restricciones dinámicas, cinemáticas y geométricas de los robots y obstáculos dinámicos.

### 3.1.10. Trayectorias de robots con evasión de obstáculos

Al generar trayectorias suavizadas para entornos con obstáculos es muy común considerar los obstáculos como objetos estáticos con una posición bien conocida. En ambientes de trabajo reales aparecerán objetos en movimiento en donde la trayectoria generada deberá realizarse en tiempo real para evitar colisiones al mismo tiempo se debe mantener el suavizado de rutas.

Si consideramos el método de B-spline para la generación de trayectorias será necesario agregar más puntos de control para evitar obstáculos lo que se consigue fácilmente en el caso de obstáculos estáticos. Sin embargo, con obstáculos dinámicos se deberán considerar características como la velocidad y orientación, el sistema sensorial utilizado permitirá reaccionar ante los cambios en la detección y generar nuevos puntos de control para rehacer la trayectoria.

Diversos trabajos se han propuesto para la generación de trayectorias con obstáculos dinámicos, la mayoría de los cuales no toma en cuenta las limitaciones dinámicas del robot. Se destaca un método llamado banda elástica cronometrada donde se toma en cuenta las restricciones del robot como las velocidades y aceleraciones que limitan el movimiento del robot. Otros métodos utilizan cuadrículas de certeza para la

representación de obstáculos y campos potenciales para la navegación del robot. El método VFH (Vector Field Histogram) y sus variantes proporcionan trayectorias suaves verificando si la trayectoria evita obstáculos. Otros factores importantes para la generación de trayectorias aparte de rutas suaves y evasión de obstáculos son la longitud de la ruta, la precisión, el costo computacional y la confiabilidad.

## 3.2. Análisis de herramientas disponibles en Matlab

Matlab cuenta con varias herramientas para la generación de trayectorias como Robotics System Toolbox y Navigation Toolbox.

Robotics System Toolbox proporciona un conjunto de herramientas y algoritmos para diseñar, simular, probar y desplegar aplicaciones de robots móviles y manipuladores. Incluye algoritmos de generación, seguimiento y control de trayectorias, así como algoritmos para la detección de colisiones, mapeo y localización de robots móviles. Cuenta también con una librería de modelos de robots que se pueden importar fácilmente y ser utilizados en simulaciones junto con los ejemplos de referencia disponibles [21].

Dentro de esta toolbox se incluyen varias funciones para la generación y seguimiento de trayectorias como funciones polinomiales, B-splines y perfiles de velocidad trapezoidal.

<code>bsplinepolytraj</code>	Genera trayectorias polinómicas usando B-splines
<code>cubicpolytraj</code>	Genera trayectorias polinómicas de tercer orden
<code>quinticpolytraj</code>	Genera trayectorias de quinto orden
<code>rottraj</code>	Genera trayectorias a través de matrices de rotación
<code>transformtraj</code>	Genera trayectorias utilizando transformaciones
<code>trapveltraj</code>	Genera trayectorias con perfiles de velocidad trapezoidal

Una de las estrategias comúnmente utilizada para generar trayectorias es utilizando perfiles de velocidad trapezoidal pues genera un perfil de velocidad con aceleraciones y desaceleraciones lineales y secciones de velocidad constante. Sin embargo, presenta cambios bruscos en la aceleración debido a las discontinuidades que presenta.

En Matlab esta función se encuentra con el nombre `trapveltraj`. Genera una trayectoria a partir de un conjunto de *waypoints*. Genera posiciones, velocidades y aceleraciones para cada muestra de tiempo *tSamples* dadas, de acuerdo con el número de muestras *numSamples* especificado.

`[q,qd,qdd,tSamples,pp] = trapveltraj(wayPoints,numSamples)`

Dado el conjunto de waypoints (101 muestras)

$$wpts = [0 \ 0; 3 \ 5; 6 \ 5; 10 \ 10]'$$

Se obtiene el perfil de velocidades y posición.

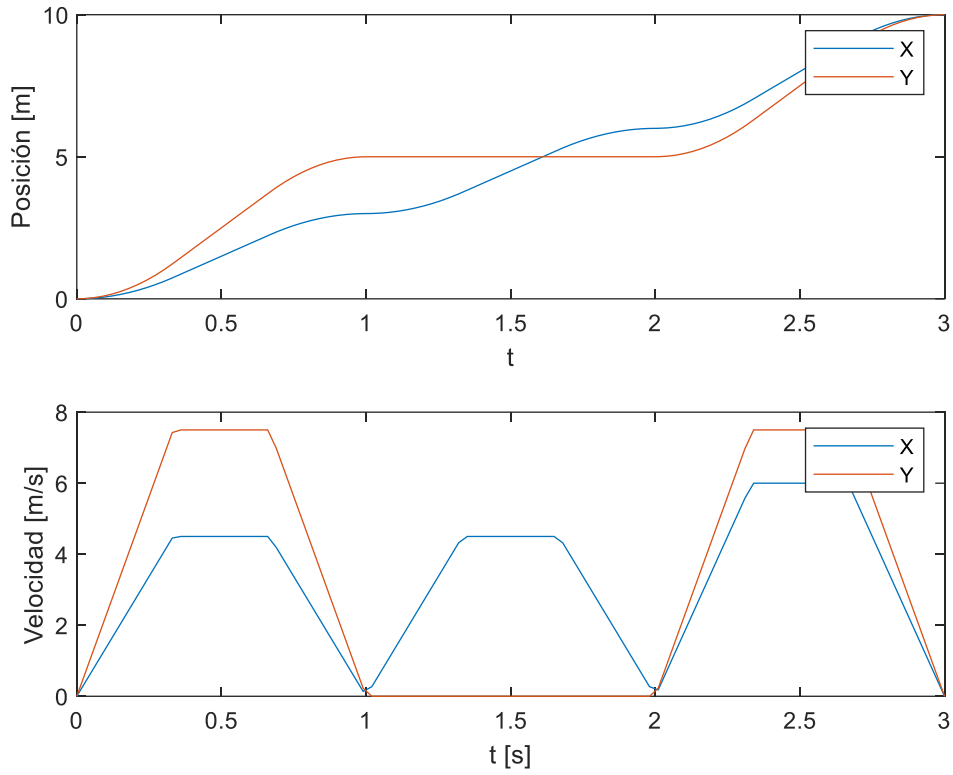


Figura 14. Perfiles de velocidad trapezoidal y posiciones.

La trayectoria obtenida se observa en la figura 15.

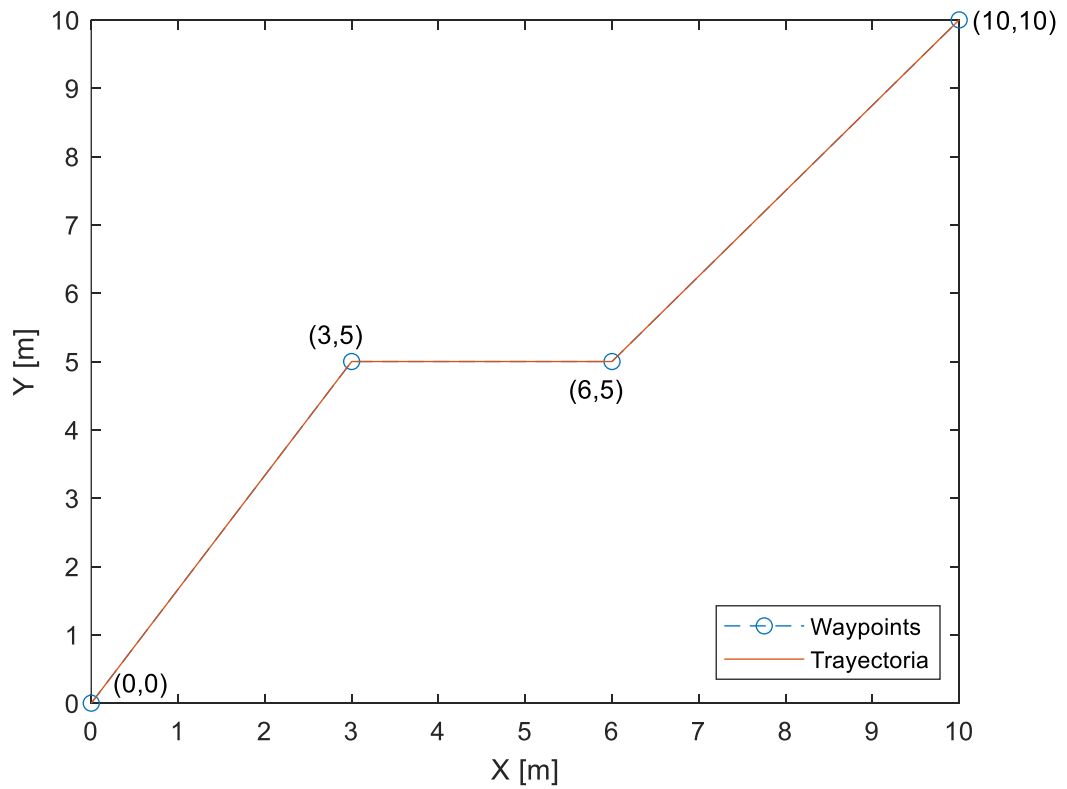


Figura 15. Generación de trayectoria con perfiles de velocidad trapezoidal.

El problema con este método es que no garantiza una trayectoria suave. Las trayectorias generadas suelen estar compuestas por segmentos rectos evidenciando discontinuidades entre segmentos como se tiene en la figura 15.

Utilizar un método de generación de trayectorias basado en B-splines garantiza la obtención de trayectorias suavizadas, este tipo de splines tienen continuidad tipo  $C^2$ , es decir, se tiene continuidad tangencial y de curvatura. La función `bsplinepolytraj` genera una trayectoria que se encuentra dentro del polígono de control definido por puntos de control o waypoints. La trayectoria se muestra uniformemente de acuerdo con el intervalo de tiempo especificado en `tInterval`. De manera similar al caso anterior con la función se obtienen las posiciones, velocidades y aceleraciones para cada instante de tiempo.

`[q,qd,qdd,pp] = bsplinepolytraj(controlPoints,tInterval,tSamples)`

Una ventaja de este método es que permite especificar el intervalo de tiempo (duración total de la trayectoria) así como el número de muestras, con lo que se especifica de manera indirecta el periodo de muestreo  $t_s$  de la trayectoria.

Para el mismo número de waypoints.

$$wpts = [0 \ 0; 3 \ 5; 6 \ 5; 10 \ 10]'$$

La duración de la trayectoria es de 100 segundos con un tiempo de muestreo de 0.1s, el número de muestras que tendrá la trayectoria es de 1001 muestras. Las posiciones y velocidades obtenidas son las siguientes:

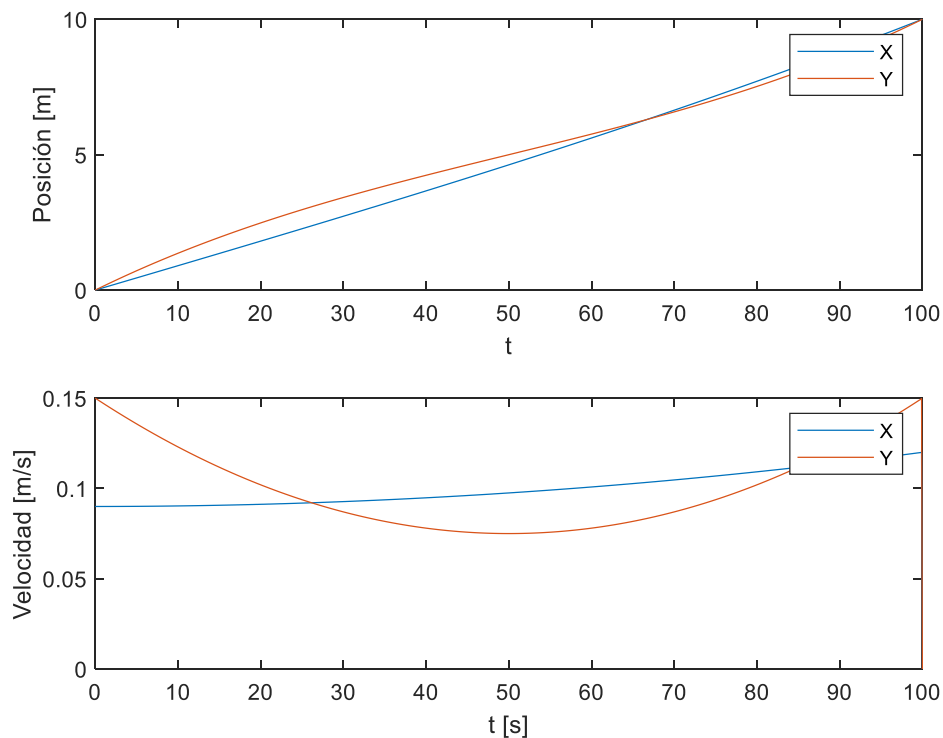


Figura 16. Posiciones y velocidades obtenidas con la función `bsplinepolytraj`.



La trayectoria generada es la siguiente.

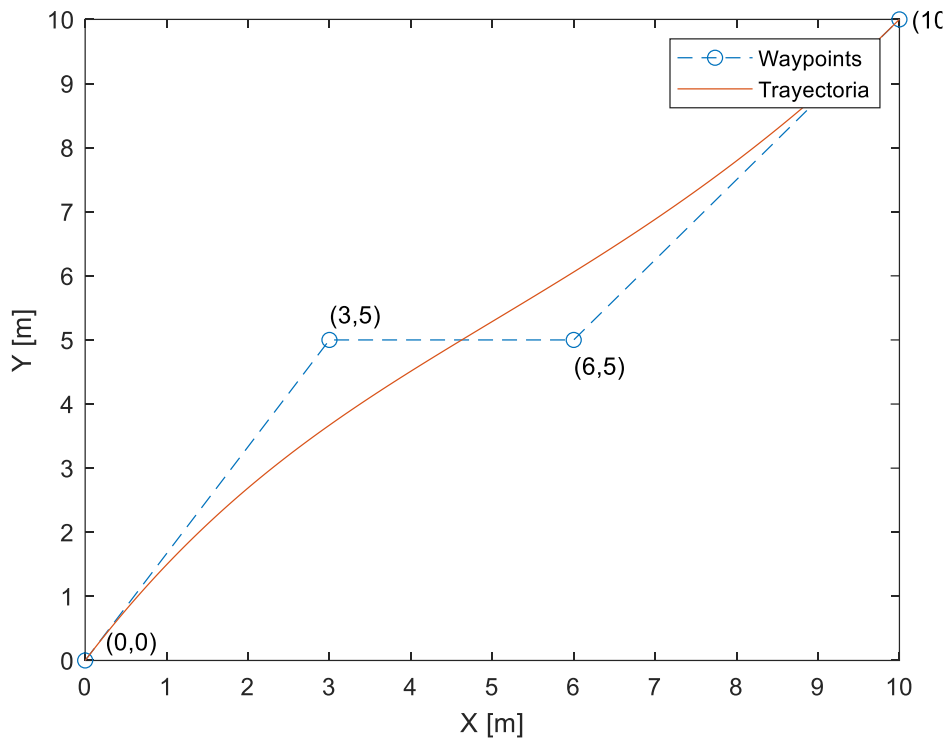


Figura 17. Generación de trayectorias basada en B-splines.

En la figura 17 se observa uno de los retos que se presenta al utilizar interpolación con B-splines, la trayectoria no pasa por los puntos de control establecidos. Este problema puede ser fácilmente corregido añadiendo más puntos de control, usualmente se utilizan matrices que permiten calcular puntos al interior del polígono de control.

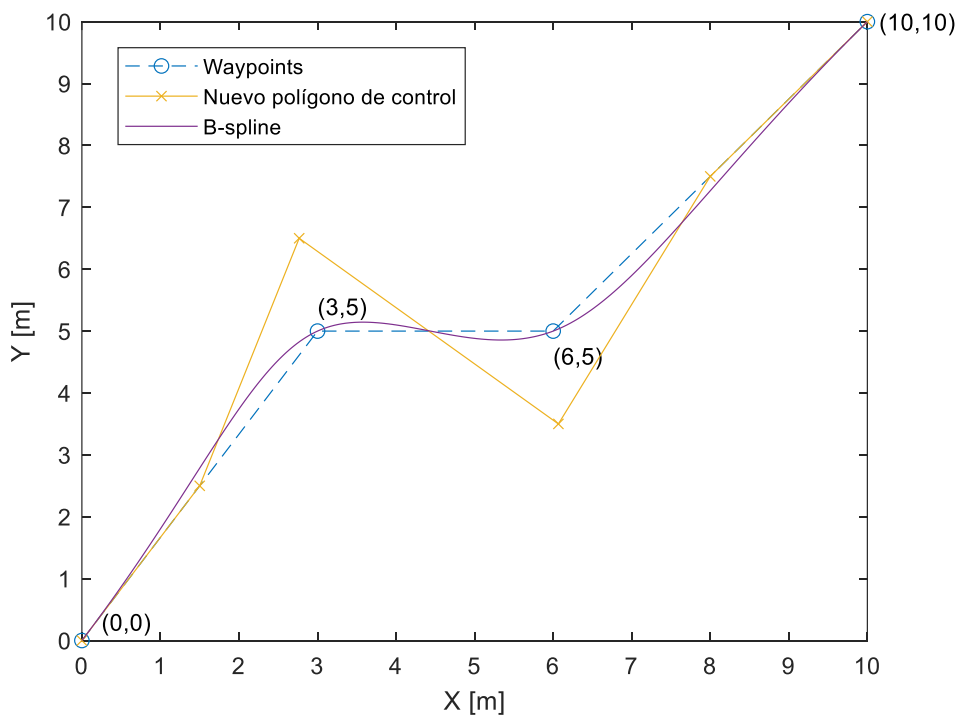


Figura 18. Generación de trayectoria, agregando más puntos de control.

Las posiciones y velocidades también se han modificado como se observa en la siguiente figura.

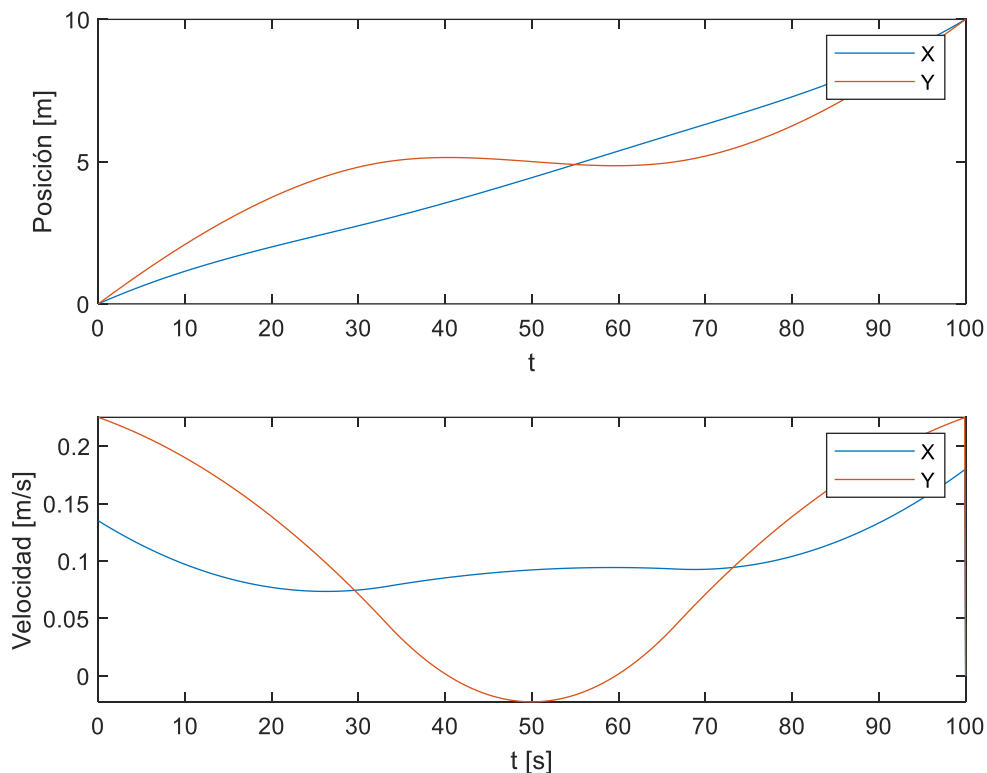


Figura 19. Posiciones y velocidades de la nueva trayectoria generada.

Otra de las herramientas de Matlab que permite generar trayectorias es Navigation Toolbox que proporciona algoritmos para planificación del movimiento, localización y mapeo simultáneos (SLAM) además de navegación inercial. Permite crear mapas de entorno en 2D y 3D con algoritmos de SLAM. Proporciona también modelos de sensores y algoritmos de localización. Para la generación de trayectorias cuenta con las siguientes funciones:

<code>waypointTrajectory</code>	Generación de trayectorias con waypoints
<code>kinematicTrajectory</code>	Generación de trayectorias basado en valores de velocidad

La función `waypointTrajectory` crea un objeto que define una trayectoria suavizada. El algoritmo usa técnicas para el suavizado de la trayectoria como clotoides y la interpolación de Hermite con el fin de minimizar las discontinuidades de curvatura. La función permite especificar algunos parámetros como el tiempo de llegada, la velocidad y orientación de cada waypoint. Con esta función se pueden obtener los parámetros de la posición, orientación, velocidad lineal y angular además de la aceleración.

`trajectory = waypointTrajectory(Waypoints,TimeOfArrival)`

En la figura 20 se observan los valores de velocidad y posición para los waypoints que ya se especificaron anteriormente, para un tiempo de llegada de 100 segundos.

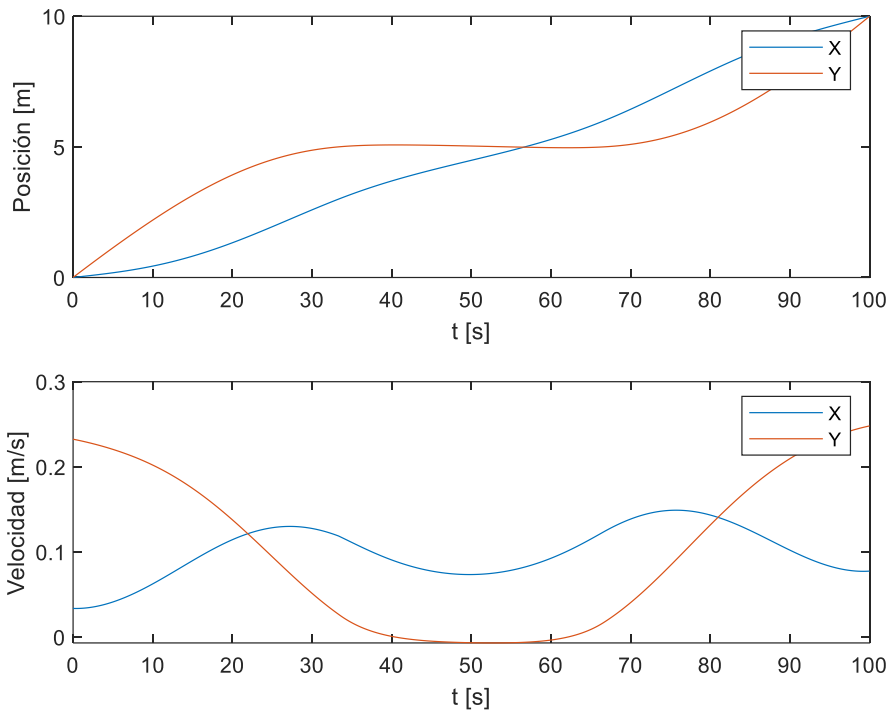


Figura 20. Posiciones y velocidades de la trayectoria generada con la función waypointTrajectory.

La trayectoria resultante es la de la figura 21, que presenta una forma continua y suave, pasando por los waypoints especificados.

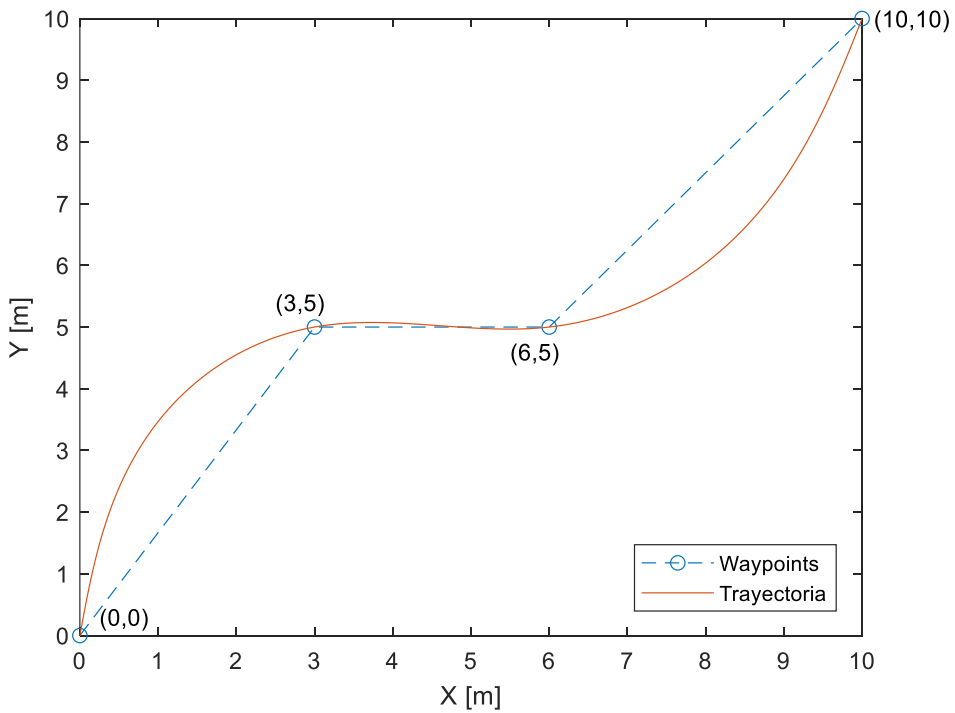


Figura 21. Trayectoria suavizada generada con el método basada en interpolación de Hermite y clotoides.

Los ejemplos que se han desarrollado utilizan tres diferentes funciones de Matlab para la generación de tres trayectorias diferentes. Las trayectorias obtenidas han permitido analizar las ventajas y desventajas de cada una de las estrategias utilizadas y determinar la que garantiza una trayectoria con continuidad tangencial y de curvatura, siendo las últimas estrategias las que proporcionan una trayectoria lo suficientemente suave para que el robot sea capaz de realizar su seguimiento. Se ha optado por utilizar la función `waypointTrajectory` por el simple hecho que proporciona adicionalmente los valores de velocidad angular necesarios para la siguiente sección de este trabajo que consiste en el seguimiento de la trayectoria.

### 3.3. Solución con plano definido

Para la generación de la trayectoria en primer lugar se considera el espacio de trabajo con obstáculos estáticos, la ubicación de los obstáculos es bien conocida. Se tiene el siguiente mapa del entorno.

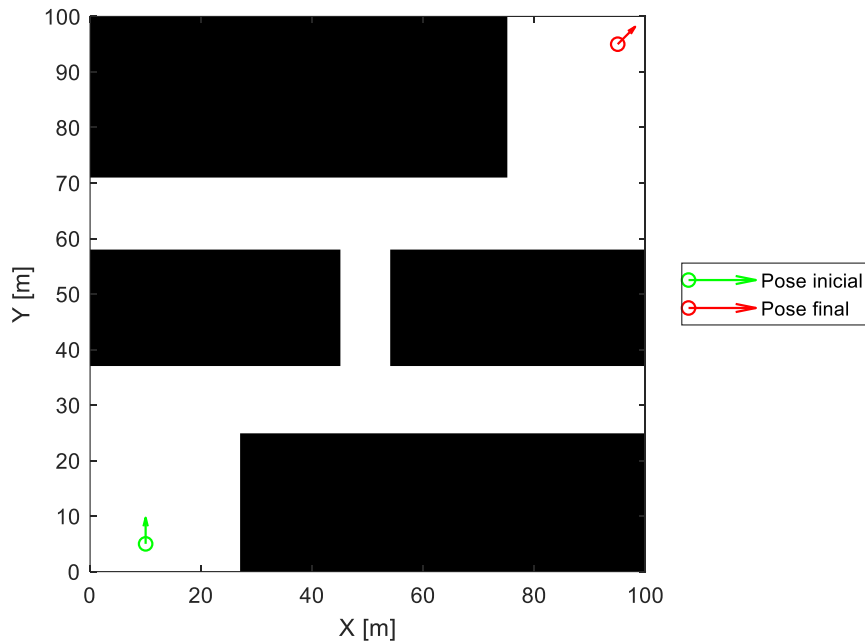


Figura 22. Mapa del entorno para la formación de robots.

La pose inicial del robot  $(x, y, \theta)$  es  $(10, 5, \frac{\pi}{2})$  y el punto de destino es  $(95, 95, \frac{\pi}{4})$ . Conocido el mapa del entorno de la figura 22 se establecen un conjunto de waypoints para la generación de la trayectoria.

$$waypoints = \begin{bmatrix} 10 & 5 \\ 10 & 10 \\ 5 & 22 \\ 20 & 30 \\ 50 & 32 \\ 53 & 65 \\ 85 & 63 \\ 95 & 95 \end{bmatrix}$$

Con los waypoints conocidos se genera una trayectoria suave que pasa por todos los puntos.

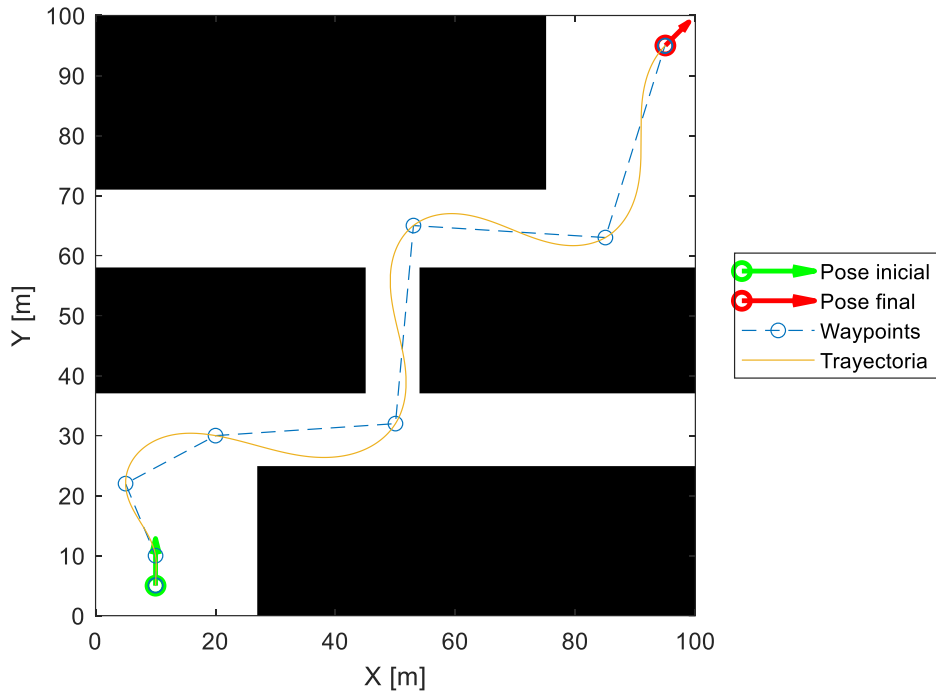


Figura 23. Trayectoria generada en el entorno de trabajo.

La velocidad lineal y angular debe cumplir los límites (ver figura 11) obtenidos con el modelo de la planta, para evitar que entre en estado de saturación.

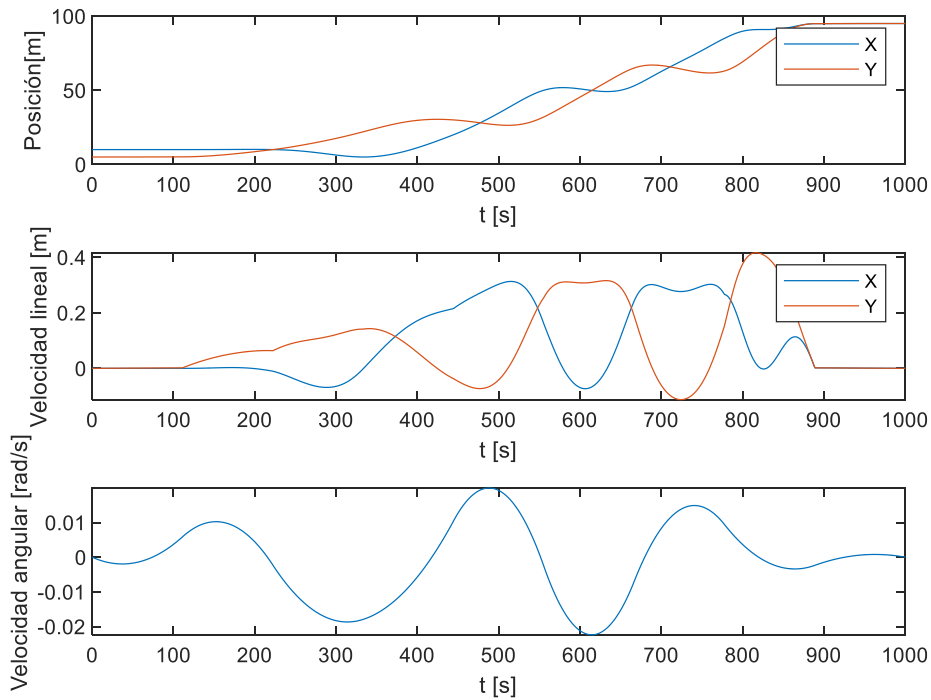


Figura 24. Posición y velocidades en el tiempo

Al observar la figura 23, la trayectoria obtenida no choca con las paredes u obstáculos del mapa. Sin embargo, hay que tener en cuenta que esta trayectoria corresponde a la que seguirá el robot virtual, no se ha considerado aun que el desplazamiento será realizado por un conjunto de robots organizados en una formación particular. Los robots de la formación se ubicarán a cierta distancia alrededor del robot virtual pues éste se ubicará al centro de la formación. Dependiendo del tamaño del objeto a transportar existe la posibilidad de que se den colisiones con los obstáculos.

La formación de robots es la siguiente

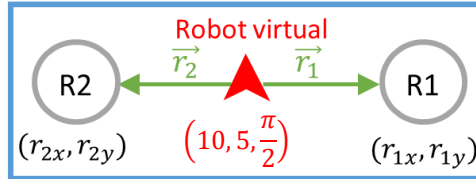


Figura 25. Formación de robots que se desplazarán por la trayectoria generada.

Se conoce la pose inicial del robot virtual

$$(x_{RV0}, y_{RV0}, \theta_{RV0}) = \left(10, 5, \frac{\pi}{2}\right)$$

Los robots R1 y R2 de la formación se ubican a cada lado del robot virtual manteniendo una distancia de 1.5m y tendrán la misma orientación del robot virtual.

Entonces, la pose inicial de R1 y R2 es

$$(x_{R1}, y_{R1}, \theta_{R1}) = \left(8.5, 5, \frac{\pi}{2}\right)$$

$$(x_{R2}, y_{R2}, \theta_{R2}) = \left(11.5, 5, \frac{\pi}{2}\right)$$

Dentro de las toolbox de Matlab antes mencionadas existe una herramienta que permite crear un mapa del entorno teniendo en cuenta las dimensiones de la unidad móvil, en este caso se utilizará esta herramienta considerando a la formación de robots como una única estructura rígida.

La función `vehicleCostmap` crea un mapa de costos que representa el espacio alrededor de un vehículo. Contiene información sobre el entorno, como obstáculos o áreas que el vehículo no puede atravesar. Para verificar colisiones, el mapa de costos redimensiona los obstáculos de acuerdo con el radio de inflación que se especifique en la propiedad `CollisionChecker`.

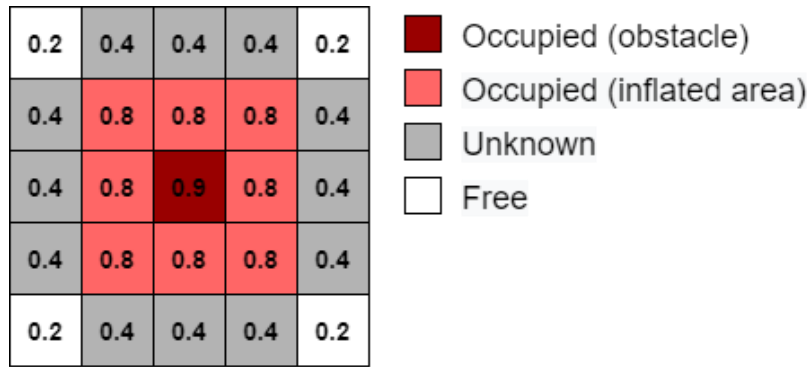


Figura 26. Cuadrícula de ocupación de la función `vehicleCostmap` [22].

El mapa de costos se representa como una cuadrícula de celdas en 2D, a menudo denominada cuadrícula de ocupación o mapa de ocupación [22]. Cada celda de la cuadrícula tiene un valor entre  $[0, 1]$  que representa el costo de navegar a través de esa celda de la cuadrícula. El estado de cada celda de la cuadrícula puede variar entre libre, ocupado o desconocido, según lo determinen las propiedades `FreeThreshold` y `OccupiedThreshold`.

Primero se crea el mapa con la función `vehicleCostmap` y se modifica la propiedad `CollisionChecker` que permite verificar si existen colisiones entre los obstáculos del mapa y la trayectoria generada. Después se establece un radio de inflación, lo que se pretende conseguir con esto es generar un área alrededor de los obstáculos que impidan que cualquiera de los robots de la formación colisione con los obstáculos o paredes del mapa. El radio de inflación es de 2m, pues se considera la distancia (1.5m) entre los robots R1 y R2 con el robot virtual añadiéndole 0.5m de margen.

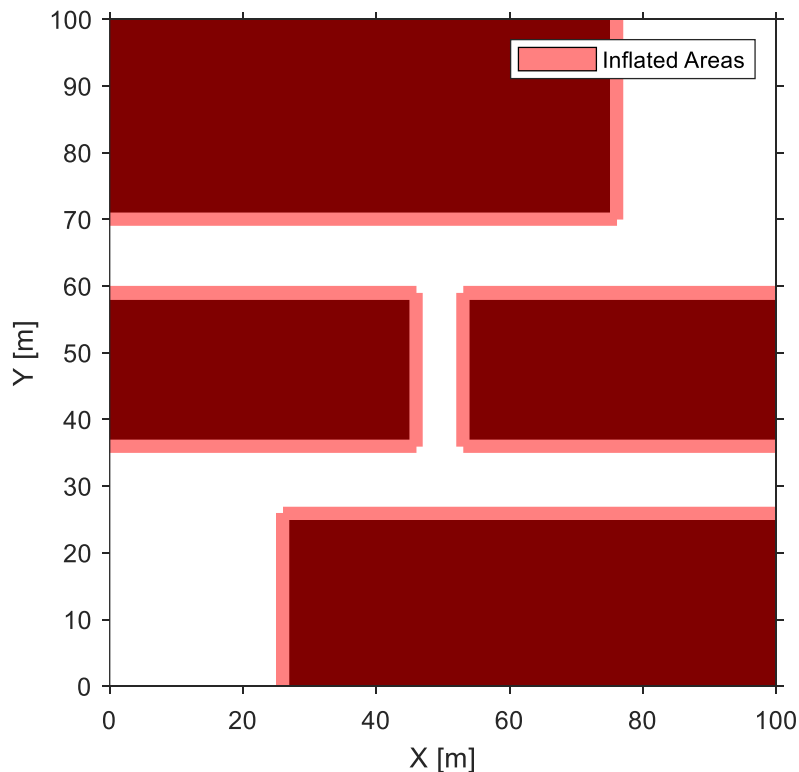


Figura 27. Mapa del entorno con su área de inflación.

Las áreas más oscuras son los obstáculos del mapa, mientras que las de color claro son áreas de inflación, que en este caso son áreas por donde la trayectoria no puede pasar.

Para validar la trayectoria se evalúa cada waypoints y se comprueba que se encuentran en las áreas libres. Cada waypoint tiene un valor de 1 lo que indica que no se ha producido ninguna colisión con los obstáculos.

```
isValid = 10x1 logical array
    1
    1
    1
    1
    1
    1
    1
    1
    1
    1
    1
```

Después se valida la trayectoria. En la figura 28 se muestran dos zonas críticas, la de la parte inferior muestra una colisión entre la trayectoria y el obstáculo. La función de verificación de colisiones devuelve el último punto válido de la trayectoria antes de la colisión.

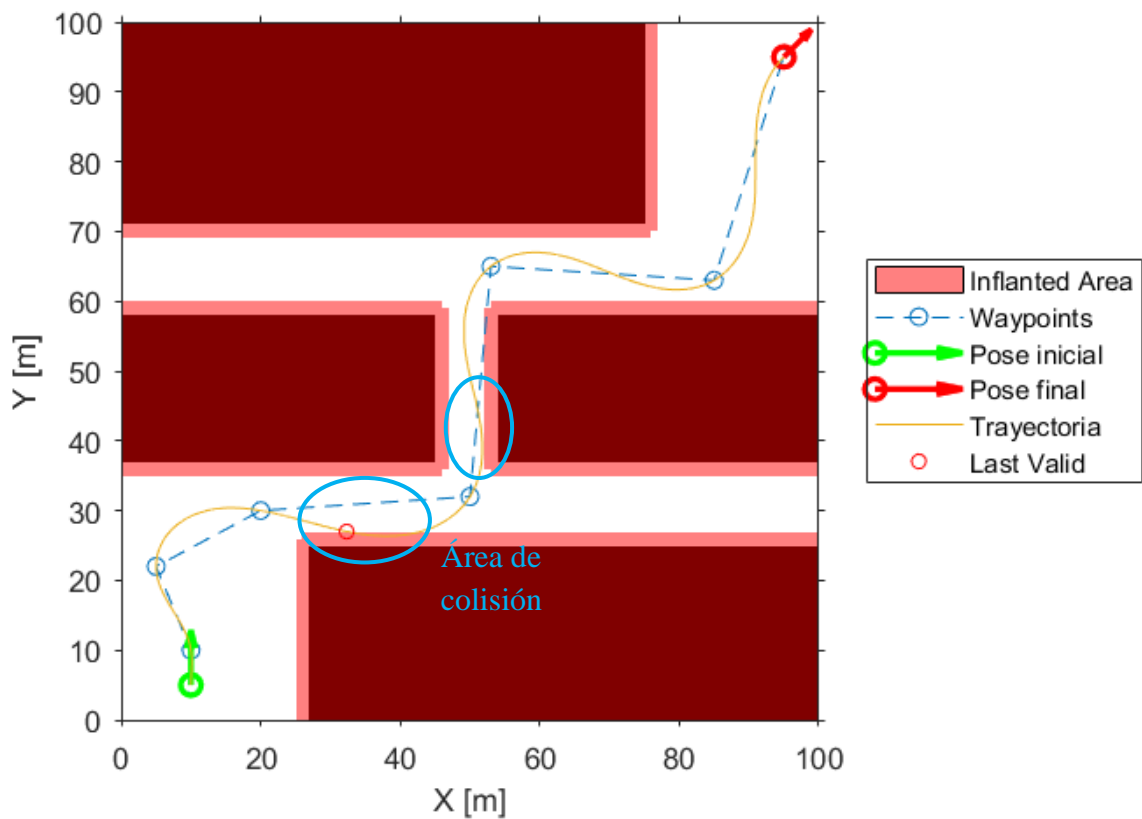


Figura 28. Verificación de colisiones entre la trayectoria y los obstáculos.



A la derecha se tiene otra área crítica, aunque no se ha producido ninguna colisión sería prudente modificar el waypoint más cercano para obtener una mejor trayectoria. De modo que para el nuevo conjunto de waypoints se agrega un punto extra y se modifica uno de ellos.

$$waypoints = \begin{bmatrix} 10 & 5 \\ 10 & 10 \\ 5 & 22 \\ 20 & 30 \\ \mathbf{32} & \mathbf{29} \\ \mathbf{48} & \mathbf{32} \\ 53 & 65 \\ 85 & 63 \\ 95 & 95 \end{bmatrix}$$

En la figura 29 se muestra la nueva trayectoria generada con los nuevos waypoints, se ha evitado toparse con los obstáculos con lo que se garantiza que toda la formación podrá desplazarse siguiendo la trayectoria y evitando los obstáculos del mapa.

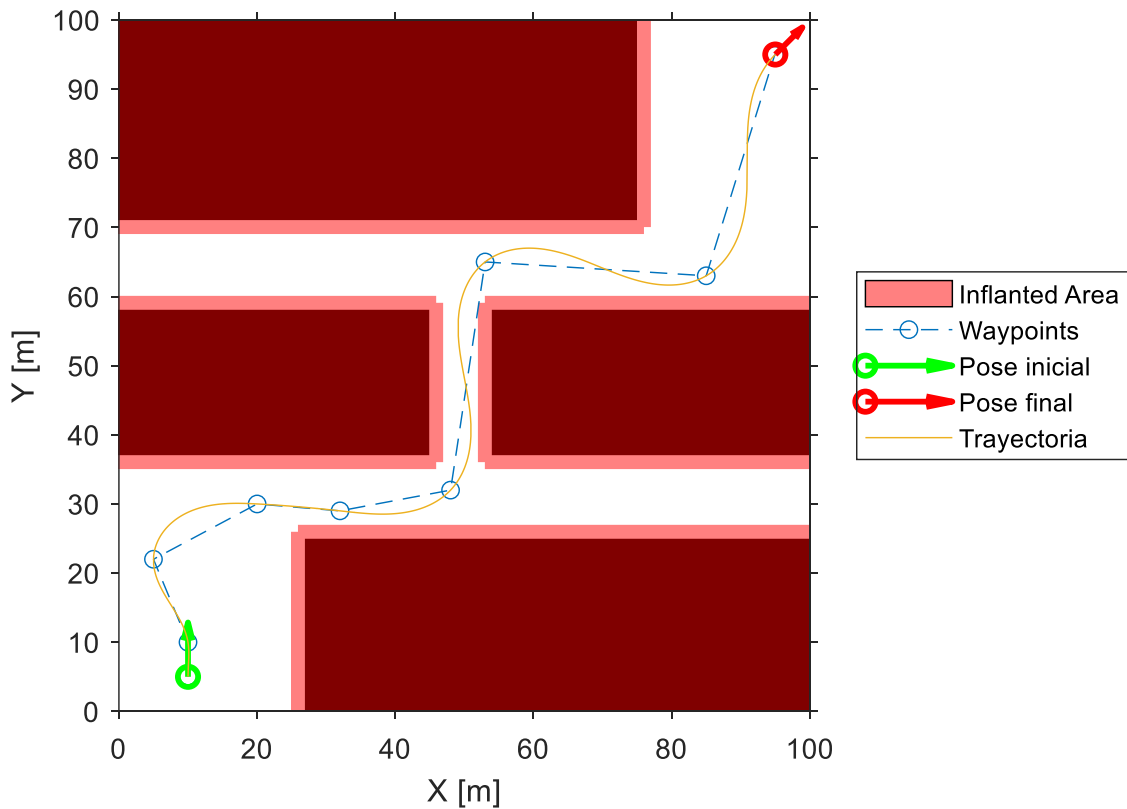


Figura 29. Nueva trayectoria generada con evasión de obstáculos.

Finalmente se calculan nuevamente los perfiles de velocidad y los valores de posición de la nueva trayectoria. Los valores de velocidad lineal y angular obtenidos no superan los límites con lo que se evita la saturación del sistema.

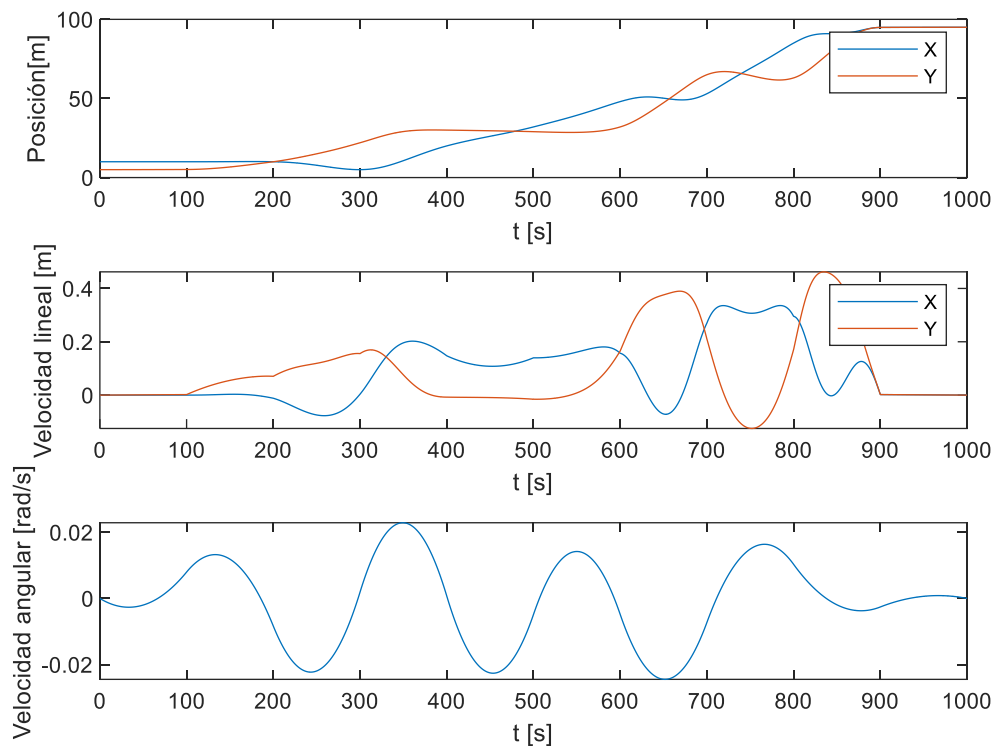


Figura 30. Perfiles de velocidad y posición de la nueva trayectoria generada.

# CAPÍTULO 4

## 4. SOLUCIÓN DE CONTROL COOPERATIVO

La propuesta de control cooperativo que se ha planteado en este trabajo consiste primero en establecer una formación de robots que será vista como una estructura rígida. En el punto de referencia de la estructura se ubicará el robot virtual y a su alrededor se organizarán los demás robots de acuerdo con la forma, tamaño y peso del objeto a transportar. Los N robots de la formación se mantendrán en todo momento a una determinada distancia del robot virtual, es decir, la posición y dirección de cada robot con respecto al robot virtual permanecerá constante durante el traslado de la carga para evitar deslizamientos.

Después, conocidas las características de las unidades móviles y del mapa del entorno se hace la generación de la trayectoria para el robot virtual. Es necesario que la trayectoria cumpla con las condiciones de continuidad necesarias para garantizar un desplazamiento sin cambios bruscos ni deslizamientos.

Con los requisitos previos establecidos la solución de control cooperativo que se plantea se divide en dos partes.

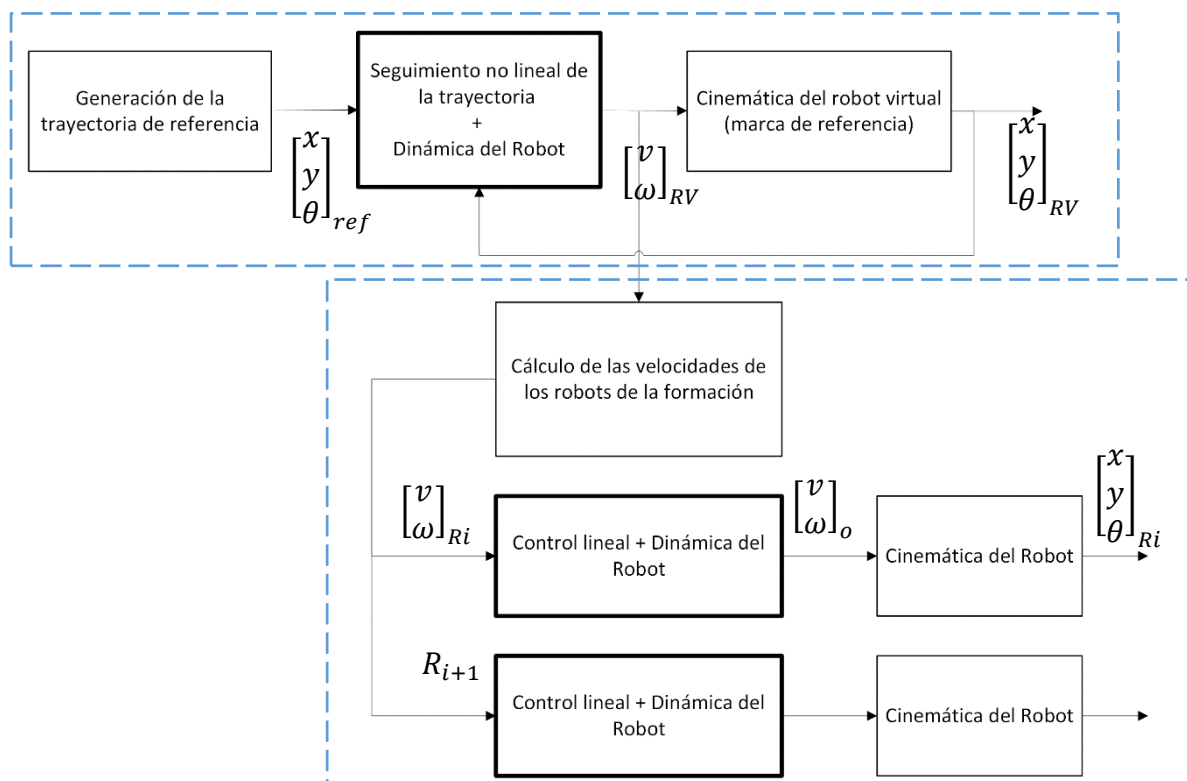


Figura 31. Estructura de la solución de control cooperativo para el transporte de una carga.

- Diseño de control no lineal basado en Lyapunov para el seguimiento de la trayectoria de referencia por parte del robot líder virtual. Se obtienen las

velocidades lineal  $v$  y angular  $\omega$  de entrada al robot virtual representado por su cinemática, y ubicado en el punto de referencia del objeto.

- Traslado de las velocidades del robot virtual al resto de los robots de la formación utilizando las ecuaciones del movimiento del sólido rígido. Se obtienen  $N$  vectores con las velocidades lineal y angular de cada robot de la formación. A continuación, se aplica el control lineal de seguimiento de velocidades para cada uno de los robots de la formación, representado por su modelo dinámico.

## 4.1. Control no lineal de la unidad de referencia

En el capítulo anterior se generó la trayectoria a seguir por la unidad virtual de referencia. La trayectoria de referencia proporciona valores de posición y orientación  $[x \ y \ \theta]_{ref}$  que serán aplicados periódicamente al lazo de control no lineal de seguimiento de trayectorias.

Como se mencionó anteriormente el control de seguimiento de trayectoria se diseña para un robot virtual cuyo modelo cinemático se describe con las siguientes ecuaciones.

$$\dot{x}_0 = v \cos \theta$$

$$\dot{y}_0 = v \sin \theta$$

$$\dot{\theta} = \omega$$

El modelo cinemático incluye operadores no lineales: seno y coseno. En este contexto, estrategias de control no lineal como las basadas en Lyapunov (LBC) resultan adecuadas. Lo que se busca con el control no lineal basado en Lyapunov es, a partir del error de pose (posición y orientación) obtener las consignas de velocidad lineal y angular que posteriormente servirán como entrada para el control lineal de velocidades de cada robot de la formación.

El objetivo del control de seguimiento de trayectoria consiste en obtener una ley de control  $u = [v \ \omega]'$  que logre alcanzar, en cada muestra  $k$ , un punto de referencia dinámico  $q_{r,k} = [x_{r,k} \ y_{r,k} \ \theta_{r,k}]'$ , partiendo de la pose actual de la unidad controlada  $q_k = [x_k \ y_k \ \theta_k]'$  y de las velocidades de referencia lineal y angular  $u_{ref} = [v_r \ \omega_r]'$ .

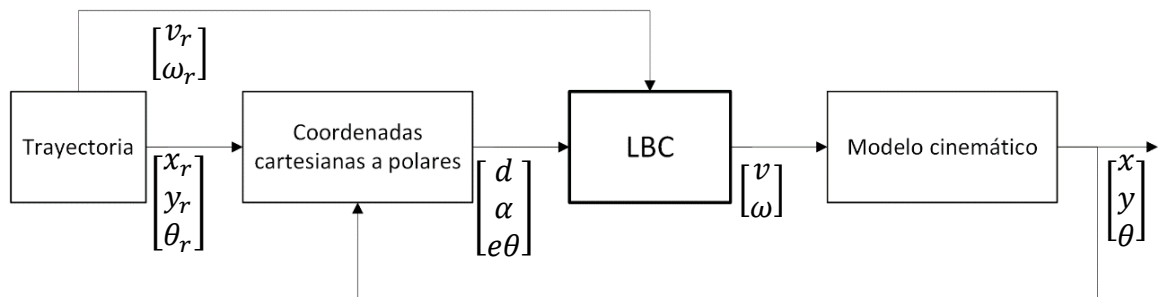


Figura 32. Esquema de control no lineal de seguimiento de trayectoria.

El controlador se basa en el concepto de guiar al robot a su orientación en cada instante por lo que la orientación de referencia toma un papel de importancia. Se aplica una variante [23] de la solución propuesta por Amoozgar [24] para el seguimiento de la trayectoria.

En el diagrama de la figura 33 se describe como el robot desde su posición y orientación actual  $q$  debe realizar una trayectoria que le permita alcanzar el punto objetivo de la trayectoria de referencia. Se define una distancia  $d$ , entre la pose actual y el punto de referencia. Se tiene también la diferencia de la posición coordenadas  $x$  e  $y$ , definida como  $e_x$  y  $e_y$ . El resto de las variables se deducen de la misma figura y se declaran a continuación.

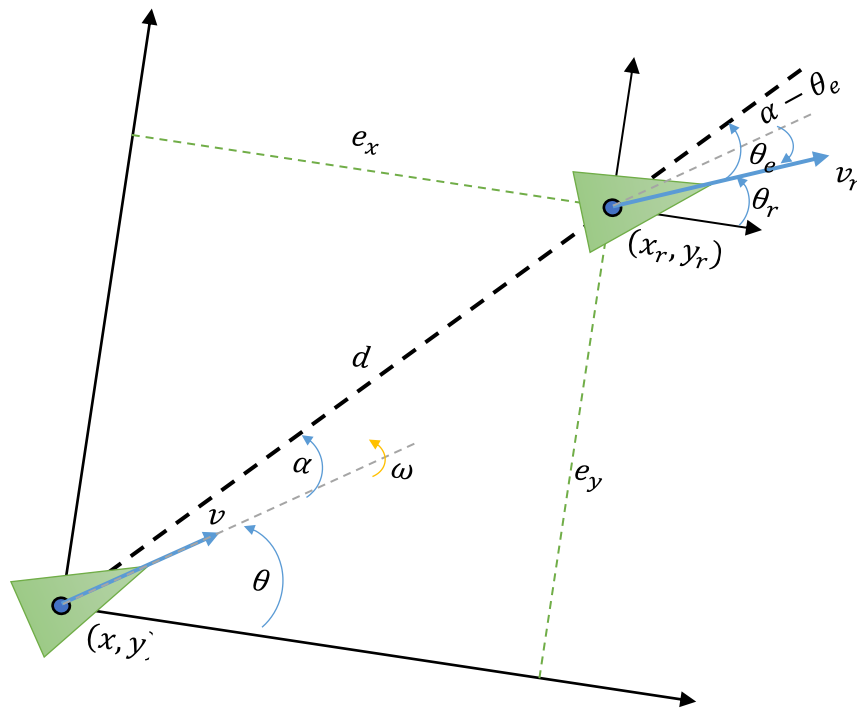


Figura 33. Descripción gráfica del seguimiento de trayectoria y la relación entre variables.

El error de seguimiento se define en coordenadas polares

$$q_e = [d \quad \alpha \quad \theta_e]'$$

Donde  $d$ ,  $\alpha$  y  $\theta_e$  se definen como

$$d = \sqrt{e_x^2 + e_y^2}$$

$$\alpha = \text{atan2}(e_y, e_x) - \theta$$

$$\theta_e = \theta_r - \theta$$

La derivada del error de seguimiento es

$$\begin{aligned} \dot{q}_e &= [\dot{d} \quad \dot{\alpha} \quad \dot{\theta}_e]' \\ \dot{d} &= v_r \cos(\alpha - \theta_e) - v \cos(\alpha) \\ \dot{\alpha} &= \frac{v}{d} \sin \alpha - \omega - \frac{v_r}{d} \sin(\alpha - \theta_e) \\ \dot{\theta}_e &= \omega_r - \omega \end{aligned}$$

Después, se define una función de Lyapunov  $V$ , que cumpla con la condición de estabilidad de Lyapunov, es decir, una función continua, definida positiva cuya derivada sea continua y definida negativa. Se establece la siguiente función de Lyapunov.

$$\begin{aligned} V(q) &= \frac{1}{2} d^2 + 1 - \cos(\theta_r - \theta) \\ V(q) &= \frac{1}{2} d^2 + 1 - \cos(\theta_e) \\ \dot{V}(q) &= d\dot{d} + \dot{\theta}_e \sin(\theta_e) \end{aligned}$$

Reemplazando las ecuaciones de  $\dot{d}$  y  $\dot{\theta}_e$  se define una función en términos de  $v$  y  $\omega$ .

$$\begin{aligned} \dot{V} &= f(v, \omega) \\ \dot{V} &= dv_r \cos(\alpha - \theta_e) - dv \cos(\alpha) + (\omega_r - \omega) \sin(\theta_e) \end{aligned}$$

La ley de control propuesta que cumple con el objetivo se define como

$$\begin{aligned} v &= dK_{v2} \cos(\alpha) + v_r \cos(\theta_e) \\ \omega &= \omega_r + d v_r \sin(\alpha) + K_{w2} \sin(\theta_e) \end{aligned}$$

Finalmente se hace el análisis de estabilidad, en donde se verifica que la derivada de la función  $V$  sea definida negativa para valores positivos de las ganancias  $K_{v2}$  y  $K_{w2}$ .

$$\dot{V} = -K_{v2}d^2 \cos^2(\alpha) - K_{w2} \sin^2(\theta_e) < 0$$

## 4.2. Distribución de control

Para el control de la formación de robots primero se calculan las velocidades lineales y angulares de cada unidad que son las entradas para el control lineal de velocidades (local) de la formación de robots. Conocida la formación de robots de la figura 34, se obtienen los vectores de posición de los robots R1 y R2.

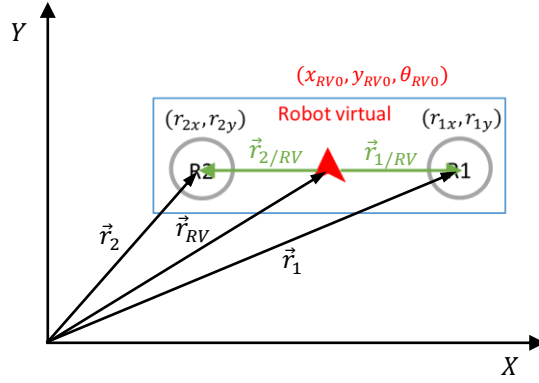


Figura 34. Relación entre los vectores posición de los robots de la formación.

$$\vec{r}_2 = (r_{2x}\vec{i} + r_{2y}\vec{j}), \quad \vec{r}_1 = (r_{1x}\vec{i} + r_{1y}\vec{j})$$

$$\vec{r}_2 = \vec{r}_{RV} + \vec{r}_{2/RV}$$

$$\vec{r}_{2/RV} = \vec{r}_2 - \vec{r}_{RV}$$

$$\vec{r}_{2/RV} = (r_{2x} - x_{RV})\vec{i} + (r_{2y} - y_{RV})\vec{j}$$

$$\vec{r}_1 = \vec{r}_{RV} + \vec{r}_{1/RV}$$

$$\vec{r}_{1/RV} = \vec{r}_1 - \vec{r}_{RV}$$

$$\vec{r}_{1/RV} = (r_{1x} - x_{RV})\vec{i} + (r_{1y} - y_{RV})\vec{j}$$

Después se aplica la ecuación que describe del movimiento plano general del sólido rígido y se obtienen las velocidades  $v, \omega$ , de las unidades robóticas.

$$\vec{v}_{Ri} = \vec{v}_{RV} + \vec{\omega}_{RV} \times \vec{r}_{i/RV}$$

$$(v_{Rix}\vec{i} + v_{Riy}\vec{j}) = (v_{RVx}\vec{i} + v_{RVy}\vec{j}) + \omega_{RV} \vec{k} \times (rx_{i/RV}\vec{i} + ry_{i/RV}\vec{j})$$

$$(v_{Rix}\vec{i} + v_{Riy}\vec{j}) = (v_{RVx}\vec{i} + v_{RVy}\vec{j}) + (\omega_{RV}rx_{i/RV}\vec{j} - \omega_{RV}ry_{i/RV}\vec{i})$$

$$v_{Rix} = v_{xRV} - \omega_{RV}ry_{i/RV}$$

$$v_{Riy} = v_{yRV} + \omega_{RV}rx_{i/RV}$$

$$v_{Ri} = \sqrt{v_{Rix}^2 + v_{Riy}^2}$$

$$\omega_{Ri} = \omega_{RV}$$

## 4.2.1. Control local de seguimiento de velocidades

En un sistema lineal de seguimiento de referencias constantes es necesario incluir integradores en el lazo de control con el fin de eliminar el error de seguimiento en régimen permanente. En la figura 35 se muestra el diagrama de bloques para un sistema de seguimiento con realimentación de estados y control integral [25].

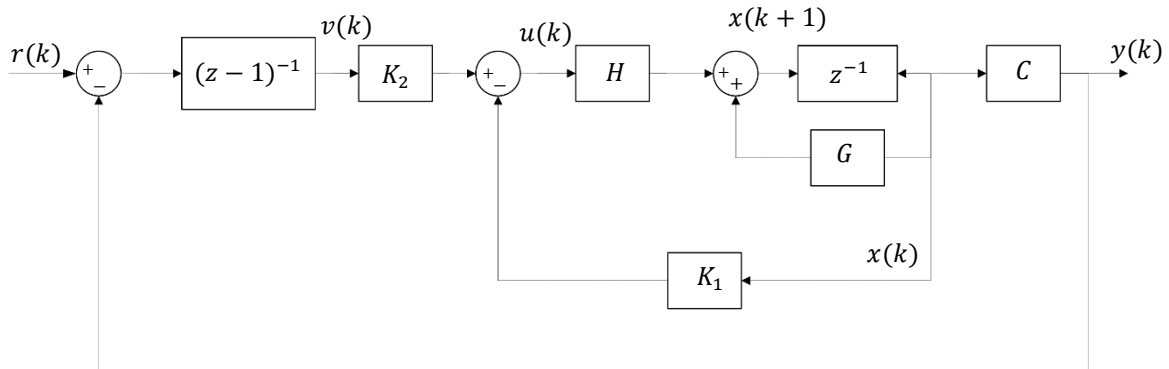


Figura 35. Sistema de seguimiento con realimentación de estados y control integral.

Donde  $x(k) \in \mathbb{R}^n$  es el vector de la planta,  $u(k) \in \mathbb{R}^m$  es el vector de control,  $y(k) \in \mathbb{R}^p$  es el vector de salida,  $v(k) \in \mathbb{R}^p$  el vector de estado asociado al integrador y  $r(k) \in \mathbb{R}^p$  el vector de referencia.

Se puede comprobar la validez de las expresiones:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} G - HK_1 & -HK_2 \\ -C - DK_1 & DK_2 + I_p \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 \end{bmatrix} r(k)$$

La ecuación de salida se escribe como:

$$y(k) = [C - DK_1 \quad -DK_2] \begin{bmatrix} x(k) \\ v(k) \end{bmatrix}$$

Para el cálculo de las ganancias  $K_1$  y  $K_2$  del servosistema se ha utilizado el método de reubicación de autovalores [25].

Para la estimación del estado de la planta se utiliza el filtro de Kalman, conocidas las matrices de covarianza de ruido de estado  $Q$  y de medida  $R$  [26].

Con todos, en la figura 36 se muestra el sistema global de seguimiento de velocidades con estimador de estados.



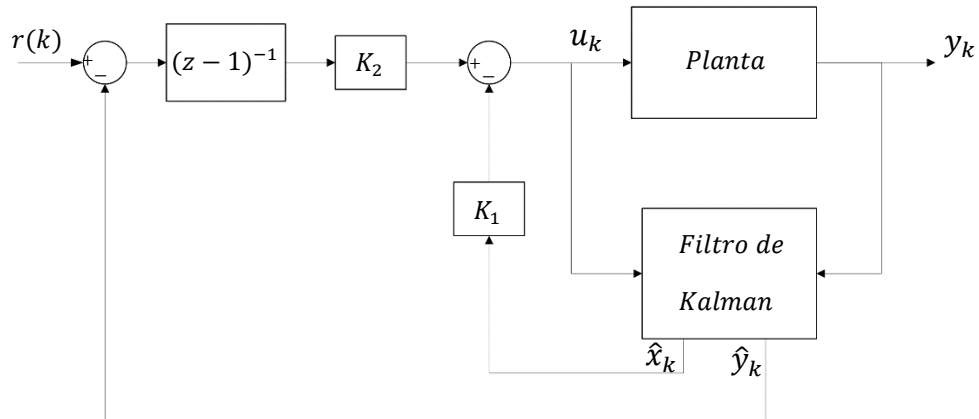


Figura 36. Sistema de seguimiento más el estimador de estados.

### 4.3. Ensayos: Simulación de la solución de control de la formación de robots

Para la simulación del control LBC de seguimiento de trayectoria se han establecido los valores de  $K_{v2} = K_{w2} = 1$ , y la pose inicial del robot virtual coincide con el punto inicial de la trayectoria generada.

$$(x_{RV0}, y_{RV0}, \theta_{RV0}) = \left(10, 5, \frac{\pi}{2}\right)$$

El punto final de la trayectoria que será el punto donde se detiene el robot al finalizar el recorrido es:

$$(x_{RVf}, y_{RVf}, \theta_{RVf}) = \left(95, 95, \frac{\pi}{4}\right)$$

Los resultados, simulados en Matlab/Simulink, del error en el seguimiento de trayectoria del robot virtual se muestran en la figura 37. De acuerdo con la trayectoria generada el tiempo de ejecución es de 1000 segundos con un periodo de muestreo de 0.025s. Como se esperaba el error de seguimiento de trayectoria tiende a 0 aunque hay presencia de transitorios debido a la no linealidad del sistema.

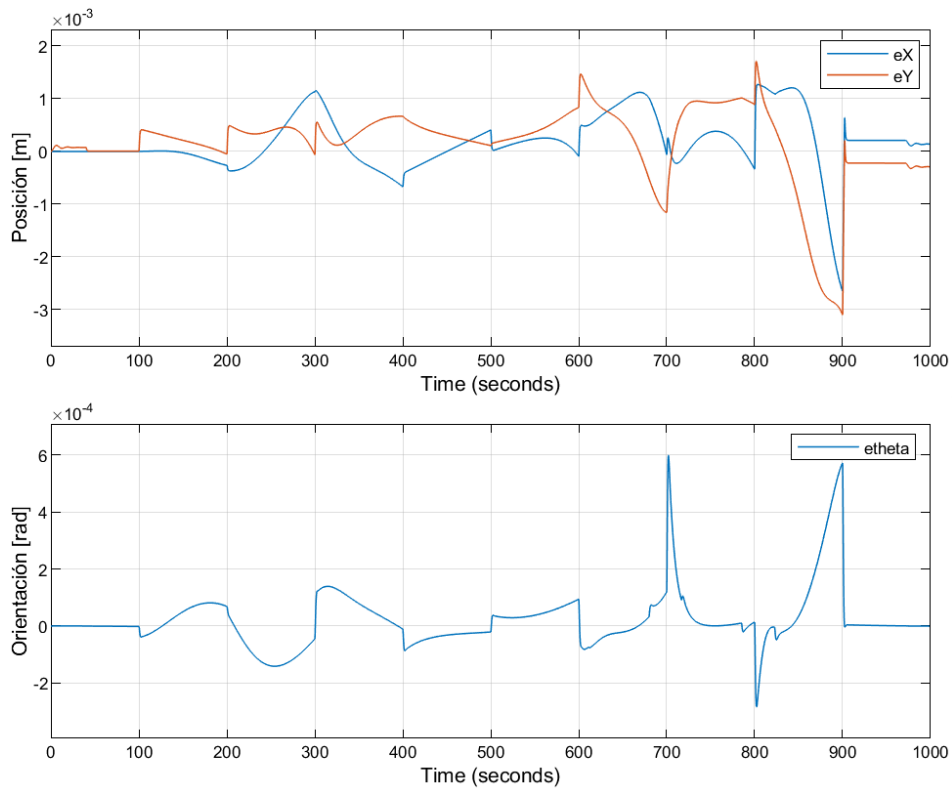


Figura 37. Error de seguimiento de trayectoria

Los valores de velocidad lineal y angular del robot virtual y de los robots R1 y R2 se muestran en la figura 38, todos ellos se encuentran dentro de los límites para evitar la saturación del sistema. La velocidad es la misma pues se desplazan de manera paralela describiendo trayectorias curvilíneas.

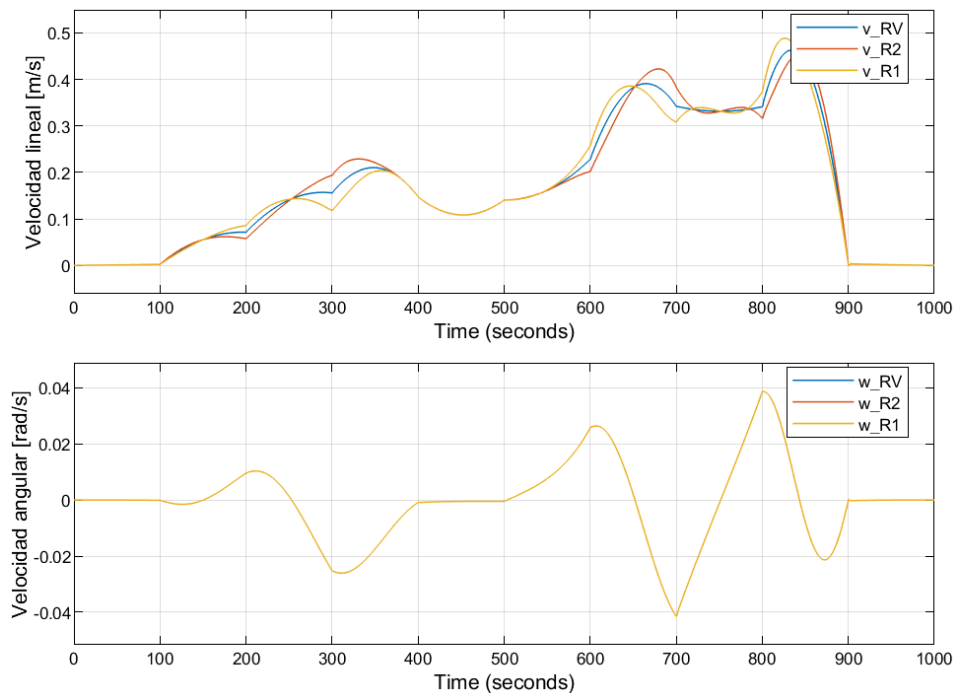


Figura 38. Consignas de velocidad lineal y angular para cada uno de los dos robots de la formación.

Para calcular las velocidades de R1 y R2 se ha implementado un bloque de funciones utilizando las ecuaciones del movimiento plano general del sólido rígido y las velocidades del robot virtual. Las ecuaciones están expresadas de forma vectorial de modo que se han obtenido los valores de velocidad en x y velocidad en y. De igual ha sido necesario calcular los valores de posición de los robots con respecto al robot virtual.

Con las consignas de velocidades conocidas se aplica el control lineal de seguimiento de velocidades de manera independiente a cada robot. Como se mencionó en la sección anterior se ha diseñado un controlador de estados realimentados que incluye un integrador y un estimador de estados.

En Matlab/Simulink se ha utilizado el bloque de Kalman Filter para implementar el estimador de estados, con el que se obtienen los valores de  $\hat{y}_k$  y  $\hat{x}_k$ , además permite incluir las características del ruido de medida y de proceso con las matrices de covarianza  $R$  y  $Q$

En la figura 39 se observa el error de seguimiento de velocidades para cada robot. Valores muy cercanos a 0 con lo que se espera que la formación de robots se desplace en lo posible manteniendo una distancia constante entre ellos.

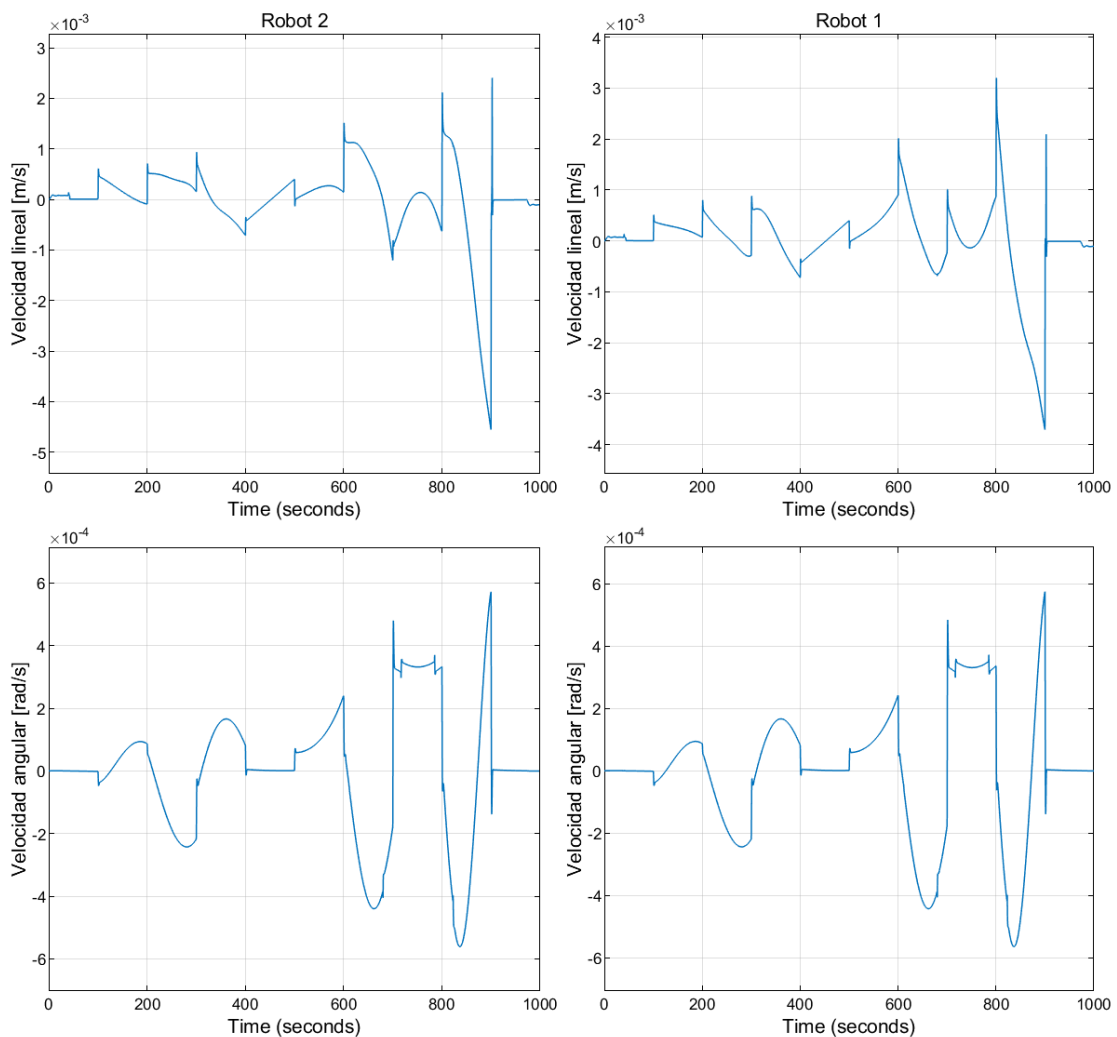


Figura 39. Error de seguimiento de velocidades.

Finalmente, las velocidades obtenidas son la entrada a la cinemática del robot y se obtienen los vectores de pose para cada robot. En la figura 40 se muestra como los robots R1 y R2 se desplazan por la trayectoria generada, la orientación de R1 y R2 es igual en un mismo instante de tiempo y los robots mantienen en todo momento una distancia de 3m entre ellos, cada uno se mantiene a una distancia de 1.5m con respecto al robot virtual pues este se encuentra al centro de la formación.

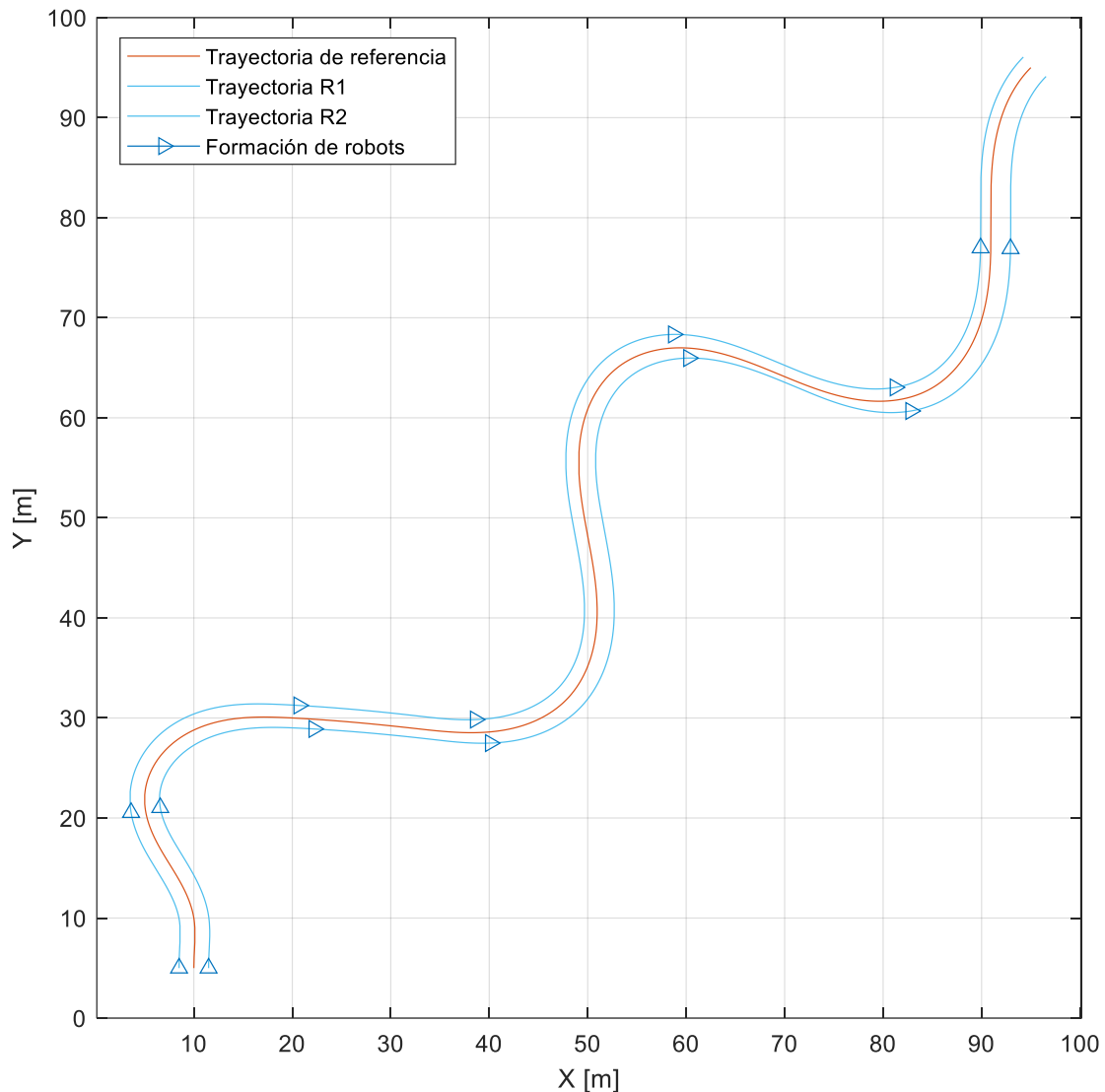


Figura 40. Desplazamiento de la formación de robots teniendo como referencia la trayectoria del robot virtual.

#### 4.4. Requerimientos para la implementación con robots reales

Aunque en este trabajo no se ha implementado el control de la formación de robots para el transporte de un objeto utilizando robots reales, es interesante hacer una breve

descripción de algunos de los requerimientos necesarios para que esto sea posible en futuros trabajos.

El esquema del posible sistema de implementación se muestra en la figura 41.

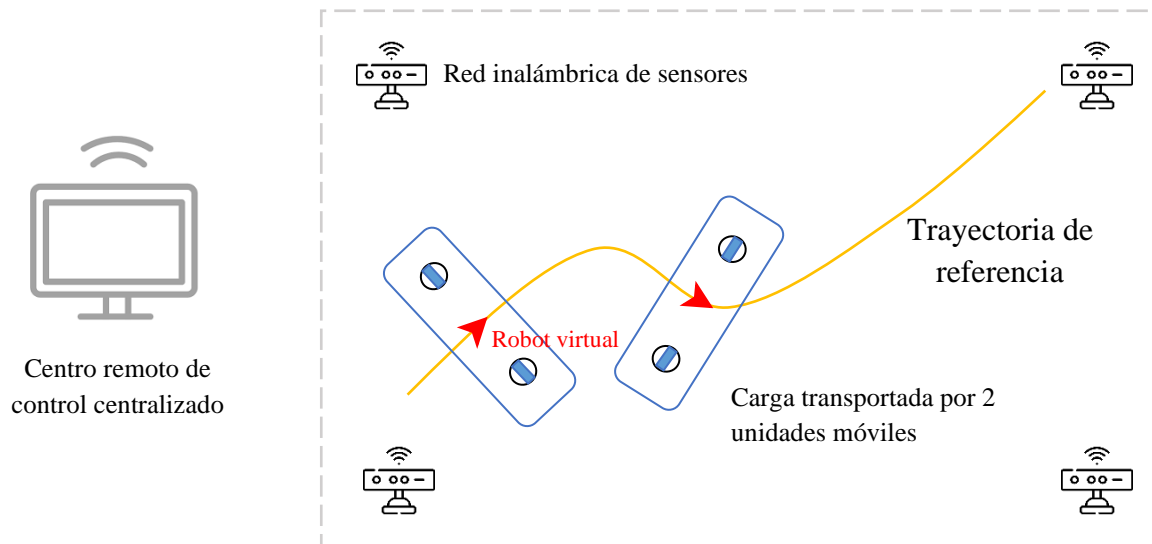


Figura 41. Formación de robots controlados por un centro remoto para el traslado de un objeto.

La formación de robots estará conformada por dos unidades móviles de tracción diferencial Pioneer P3-DX, que cuentan con odometría embarcada. Por otro lado, el objeto a transportar debe tener una marca de referencia que permitirá conocer en todo momento la posición y orientación actual del objeto, y que además coincide con la pose del robot virtual.

Un centro remoto será el responsable de controlar la formación de robots, primero, con la generación de la trayectoria de referencia y después con el control no lineal del robot virtual para el seguimiento de trayectoria. A partir de ahí se calculan las señales  $v, \omega$  de cada robot real. Se contará también con unas cámaras conectadas en red con el centro remoto de control, de esta forma se registra periódicamente la posición y orientación de la marca de referencia. Las unidades móviles deberán incorporar nodos de comunicación bidireccional con el centro remoto.

De manera general el sistema deberá contar con los siguientes elementos [27]:

- PC: Centro remoto y driver de comunicación.
- Robot: con PC embarcado para recibir consignas y enviar respuesta al centro remoto.
- Red Wifi para para enlace con los robots y con las cámaras del escenario de ensayo. Comunicación por sockets.

Herramientas de software y hardware necesarias:

- Hardware: PC, robot P3-DX, cámaras.
- Software: Matlab 2019a/Simulink/Simulink Coder, ssh Putty.

# CAPÍTULO 5

## 5. CONCLUSIONES Y TRABAJOS FUTUROS

---

En este capítulo se exponen las conclusiones del trabajo realizado poniendo particular atención en sus puntos principales: generación de trayectorias continuas y control cooperativo de para el traslado de cargas. Después se proponen futuros trabajos derivados del presente trabajo.

### 5.1. Conclusiones

- La generación de la trayectoria ha sido el punto de partida de este trabajo fin de máster. Conocer las limitaciones de la unidad móvil, así como las características del mapa del entorno ha facilitado generar una trayectoria que permite que el robot se desplaza por el espacio de trabajo evitando colisiones. Para que el desplazamiento del robot sea suave (funciones continuas de velocidad lineal y aceleración), la trayectoria generada ha tenido que cumplir con continuidad tangencial y de curvatura. Después de comparar distintas herramientas disponibles en Matlab para la generación de trayectorias, se ha utilizado una estrategia basada en clotoides e interpolación de Hermite que partiendo de un grupo de waypoints genera una trayectoria con continuidad  $C^2$ . Aprovechando las herramientas de Matlab se ha realizado también una verificación de colisiones que permite comprobar que la trayectoria no choca con las paredes u obstáculos del mapa. Por último, se ha comprobado que los valores requeridos de velocidad lineal y angular están dentro del rango disponible en los robots comerciales de referencia.
- Se ha tomado un enfoque en donde la formación de robots es vista como una estructura rígida de modo que el control no lineal de seguimiento de trayectoria se aplica únicamente al robot virtual, en donde, se ha de corregir el error de seguimiento de la única marca que identifica al objeto. A la vez el LBC proporciona las velocidades lineal y angular  $v, \omega$  del robot virtual a partir de las cuales se calculan las velocidades de cada robot de la formación. Este calculo se facilita utilizando las ecuaciones del movimiento de traslación y rotación de un sólido rígido. Finalmente, el control local de cada robot se encarga del seguimiento lineal de velocidades.

### 5.2. Trabajos Futuros

- El paso inmediato es comparar los resultados de simulación con una implementación real utilizando la infraestructura disponible en los laboratorios

del Departamento de Electrónica de la EPS. De esta forma se podrán evaluar aspectos no considerados en el modelado como un comportamiento no ideal de la red de comunicación inalámbrica.

- En seguimiento de rutas uno de los retos más desafiantes es la evasión de obstáculos en movimiento, especialmente en entornos parcialmente conocidos y cambiantes. Para ello se requieren estrategias de control inteligente (borroso, neuronal, etc) además de procesadores embarcados con suficiente carga computacional.

# BIBLIOGRAFÍA

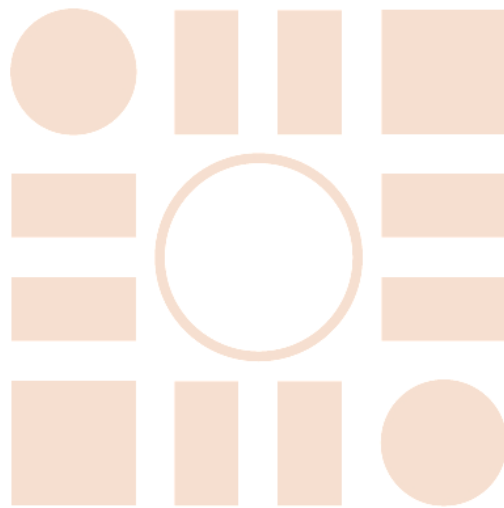
---

- [1] «Robótica: Manipuladores y Robots Móviles - Aníbal Ollero Baturone - Google Libros».  
[https://books.google.es/books?id=TtMfuy6FNCcC&dq=robot+m%C3%B3vil&hl=es&source=gbs\\_navlinks\\_s](https://books.google.es/books?id=TtMfuy6FNCcC&dq=robot+m%C3%B3vil&hl=es&source=gbs_navlinks_s) (accedido 3 de agosto de 2022).
- [2] J. González-Jiménez y A. Ollero, «Estimación de la Posición de un Robot Móvil», vol. 29, pp. 3-18, 1996.
- [3] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, y C.-C. Peng, «Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges», *Sensors*, vol. 18, n.º 9, Art. n.º 9, sep. 2018, doi: 10.3390/s18093170.
- [4] V. Barrientos, J. Sánchez, y R. Ortigoza, «Robots Móviles: Evolución y Estado del Arte», *Polibits*, vol. 35, pp. 12-17, ene. 2007, doi: 10.17562/PB-35-3.
- [5] L. Guzmán, M. Villa, y E. Vásquez, «Seguimiento de trayectorias con un robot móvil de configuración diferencial», *Ing. USBMed*, vol. 5, pp. 26-34, jun. 2014, doi: 10.21500/20275846.298.
- [6] F. Roberti, J. Marcos Toibero, R. Frizera Vassallo, y R. Carelli, «Control Estable de Formación Basado en Visión Omnidireccional para Robots Móviles No Holonómicos», *Rev. Iberoam. Automática E Informática Ind. RIAI*, vol. 8, n.º 1, pp. 29-37, ene. 2011, doi: 10.1016/S1697-7912(11)70005-X.
- [7] E. Fabregas y G. F. Castro, «RFCSIM Simulador Interactivo De Robotica Movil Para Control De Formacion Con Evitacion De Obstaculos», p. 6, 2014.
- [8] A. Yufka y M. Ozkan, «Formation-Based Control Scheme for Cooperative Transportation by Multiple Mobile Robots», *Int. J. Adv. Robot. Syst.*, vol. 12, n.º 9, p. 120, sep. 2015, doi: 10.5772/60972.
- [9] Yu. N. Zolotukhin, K. Yu. Kotov, A. S. Maltsev, A. A. Nesterov, M. A. Sobolev, y M. N. Filippov, «A relative measurement based leader-follower formation control of mobile robots», en *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, jul. 2015, vol. 02, pp. 310-313.
- [10] S. A. Barogh y H. Werner, «Cascaded formation control using angle and distance between agents with orientation control (Part 1)», en *2016 UKACC 11th International Conference on Control (CONTROL)*, ago. 2016, pp. 1-6. doi: 10.1109/CONTROL.2016.7737625.
- [11] S.-M. Kang y H.-S. Ahn, «Design and Realization of Distributed Adaptive Formation Control Law for Multi-Agent Systems With Moving Leader», *IEEE Trans. Ind. Electron.*, vol. 63, n.º 2, pp. 1268-1279, feb. 2016, doi: 10.1109/TIE.2015.2504041.
- [12] L. Qiang, W. Heng, L. Huican, Q. Shuqi, D. Nanxun, y L. Bing, «Formation control of multi robot based on UWB distance measurement», en *2018 Chinese Control And Decision Conference (CCDC)*, jun. 2018, pp. 2404-2408. doi: 10.1109/CCDC.2018.8407528.
- [13] C.-H. Lin, S.-H. Wang, y C.-J. Lin, «Interval Type-2 Neural Fuzzy Controller-Based Navigation of Cooperative Load-Carrying Mobile Robots in Unknown Environments», *Sensors*, vol. 18, n.º 12, Art. n.º 12, dic. 2018, doi: 10.3390/s18124181.
- [14] V. Mwaffo, J. S. Curry, F. L. Iudice, y P. De Lellis, «Pause-and-Go Self-Balancing Formation Control of Autonomous Vehicles Using Vision and Ultrasound Sensors», *IEEE Trans. Control Syst. Technol.*, vol. 29, n.º 6, pp. 2299-2311, nov. 2021, doi: 10.1109/TCST.2020.3036892.
- [15] S. Burbano, *Fisica General*. Editorial Tebar, 2003.



- [16] J. L. T. Franz, *272 Exámenes de Física: Resueltos y Comentados (Primeros Cursos de Universidad)*. Editorial Tebar, 1994.
- [17] A. Beléndez, «1. Cinemática plana del sólido rígido», sep. 2017, Accedido: 18 de agosto de 2022. [En línea]. Disponible en: <http://rua.ua.es/dspace/handle/10045/95218>
- [18] Universidad Tecnológica Nacional, «CINEMATICA DEL CUERPO RIGIDO». [https://frh.cvg.utn.edu.ar/pluginfile.php/40820/mod\\_folder/content/0/6\\_Cuerpo%20R%C3%ADgido\\_Cinem%C3%A1tica/cinematica%20del%20cuerpo%20r%C3%ADgido.pdf?forcedownload=1](https://frh.cvg.utn.edu.ar/pluginfile.php/40820/mod_folder/content/0/6_Cuerpo%20R%C3%ADgido_Cinem%C3%A1tica/cinematica%20del%20cuerpo%20r%C3%ADgido.pdf?forcedownload=1)
- [19] C. G. Fernández, *Fundamentos de mecánica*. Reverte, 2012.
- [20] O. E. Ramos, «Cinemática de Robots Móviles», p. 52.
- [21] «Robotics System Toolbox». <https://es.mathworks.com/products/robotics.html> (accedido 26 de agosto de 2022).
- [22] «vehicleCostmap - MathWorks». <https://es.mathworks.com/help/driving/ref/vehiclecostmap.html> (accedido 27 de agosto de 2022).
- [23] F. Espinosa y C. Losada, «Design techniques for nonlinear control systems. Apuntes de Diseño de Sistemas Electrónicos de Control.» Universidad de Alcalá.
- [24] M. H. Amoozgar y Y. M. Zhang, «Trajectory tracking of Wheeled Mobile Robots: A kinematical approach», en *Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, jul. 2012, pp. 275-280. doi: 10.1109/MESA.2012.6275574.
- [25] «Sistemas de control en tiempo discreto - Katsuhiko Ogata - Google Libros». [https://books.google.es/books?id=aYFUs17m0YQC&printsec=frontcover&dq=Sistemas+de+Control+en+Tiempo+Discreto+ogata&hl=es&sa=X&redir\\_esc=y#v=onepage&q=Sistemas%20de%20Control%20en%20Tiempo%20Discreto%20ogata&f=false](https://books.google.es/books?id=aYFUs17m0YQC&printsec=frontcover&dq=Sistemas+de+Control+en+Tiempo+Discreto+ogata&hl=es&sa=X&redir_esc=y#v=onepage&q=Sistemas%20de%20Control%20en%20Tiempo%20Discreto%20ogata&f=false) (accedido 1 de septiembre de 2022).
- [26] F. Espinosa y C. Losada, «Stochastic estimators: KF. Apuntes de Diseño de Sistemas de Control.» Universidad de Alcalá.
- [27] C. Losada y F. Espinosa, «Link Remote-Center - P3-DX robot. Apuntes de Diseño de Sistemas Electrónicos de Control». Universidad de Alcalá.

Universidad de Alcalá  
Departamento de Electrónica  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá